ХАКЕР

ИЮЛЬ 2016

№210

Меняем рутованный Android до неузнаваемости

Ориентируемся в VR: на чем пишут софт для Hololens, GearVR и Oculus

Cover Story

Ethereum

Как работает самый амбициозный криптовалютный проект этого года

июль 2016
№ 210

CONTENT

MEGANEWS
Всё новое за последний месяц

Сегодня в эфире
Краткий экскурс в Ethereum

О чем весь этот Ethereum
Виталик Бутерин о себе и о создании Ethereum

Внимание, это розыгрыш!
Пишем простой смарт-контракт

Малиновый контракт
Ставим Ethereum на Raspberry Pi

Крах DAO
Как хакер разорил автоматическую корпорацию

API для всех и каждого
Создаем мощные парсеры веб-сайтов без единой строки кода

Победа над «бэдами»
Используем Victoria для восстановления данных и сброса пароля диска

WWW2
Интересные веб-сервисы

Карманный софт
Выпуск #21. Юзабилити

Backups for fun and profit
Ломаем бэкапы «самой безопасной ОС в мире» BlackBerry 10

Кастомные прошивки больше не нужны?
Колонка Евгения Зобнина

Дайджест новостей за месяц
Paranoid Android, Cyanogen OS и смартфон «Ермак»

Измени меня полностью
Кастомизируем интерфейс Android до неузнаваемости

«Десятка» от Apple
Полный обзор iOS 10

Обзор эксплоитов
Анализ новых уязвимостей

По следам хакера
Разбираемся со взломом сайта на WordPress

Составляем карту рынка ИБ
Колонка Александра Полякова

X-TOOLS
Софт для взлома и анализа безопасности

Анатомия Android-малвари
Как организованы современные зловреды

Задачи на собеседованиях
Задачи от «Лаборатории Касперского» и дополнительное награждение от ABBYY

VK.com в своих целях: 5 готовых рецептов
Делаем новостное приложение, кинотеатр, MP3-плеер, фотосервер и убийцу Google Docs

Мобильная социализация
Полный гайд по использованию OAuth-авторизации

Виртуальная реальность для программиста
Обзор устройств и средств разработки

Хешируем по ГОСТ
Кодерский взгляд на импортозамещенное шифрование

Простота и удобство
Обзор компактных, быстрых и невероятно удобных приложений проекта suckless.org

Игры с песочницей
Выбираем простое и быстрое решение для изоляции приложений

Миграция по закону
Как безопасно перенести персональные данные в виртуализированную среду?

Хранитель данных
Новое в MS SQL 2016

FAQ
Вопросы и ответы

Титры
Кто делает этот журнал

MEGANEWS

Kaspersky Lab
Daily threat feed
Author: Sergey Lozhkin

Первая месяца лета показал, что «высокий сезон» создает не только в плане угроз компьютерной индустрии, но и в мире спам информационного. Иногда выдался новосибирский «Сезон» на «плохие» рекламной информации — как из-за рубежа популярных веб-серверов, так и из без дачника, вы-пускать. Мы имеющие наиболее интересные из них. Уклонялись также и активность ботнетов, которые из-за ограничения объемов спам-почты и распространения. Не остается от ботнетов и фишеры, начавшие в июне сезо-нность новых для себя тем.

СУРОВЫЕ ВРЕМЕНА

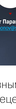
24 июня 2010 года Google PR-пиарила сразу по каждым и прочим чтением на «Сурового» Интернета. Вот, мол, «Суровые времена для спамеров». Ну, в общем, ничего такого уж сурового Google не сделала. Ограничения на объемы CMS-ами сервисы это законодательство «самых жестоких» в мире?

Ограничения на объемы CMS-ами сервисы это законодательство «самых жестоких» в мире?

В ЭФИРЕ

КРАТКИЙ ЭКСКУРС В ETHEREUM

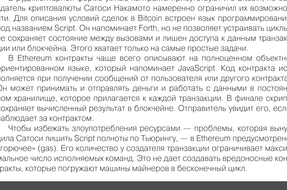
Валерий Литкин
litvinfin@gmail.com



Олег Парамонов
paramonov@sheg.ru

Ethereum можно смело называть самым амбициозным последователем Bitcoin. Тот был попыткой создать децентрализованную финансовую систему, в которой правила игры задают не государственные органы и банки, а бесстрастные алгоритмы. Ethereum идет дальше и вводит альтернативную правовую систему, которая не нуждается в судах, юристах и принуждении. А заодно служит уникальной средой для гарантированных вычислений.

Придумал Ethereum канадский программист с российскими корнями Виталик Бутерин. Когда Ethereum только появился, на дворе был 2013 год, а Бутерину исполнилось всего восемнадцать лет. Тем не менее его идея нашла живейший отклик в сообществе. Нашелся человек — Гэвин Вуд, который доказал возможность создания продвинутых криптовалютных системы и описал основные принципы ее работы в Ethereum Yellow Paper (pdf). Вместе с первыми участниками команды Ethereum они запустили краудфандинг и собрали инвестиций на 18 миллионов долларов в биткойнах — в обмен на 60 миллионов эфиров.



Чем же уникален Ethereum? Каждая транзакция (сделка) в нем сопровождается так называемым умным контрактом — компьютерной программой, которая автоматически проверяет условия договора и выполняет прописанные в нем обязательства. Поскольку умный контракт, в отличие от обычного, исполняют не люди, а машины, его невозможно нарушить, обойти или отменить. В блокчейне «Этереума» хранятся не только состояния счетов, но и состояния программ после их выполнения в распределенной сети.

В рудиментарной форме умные контракты присутствуют и в Bitcoin, но создатель криптовалюты Сатоси Накамото намеренно ограничил их возможности. Для описания условий сделок в Bitcoin встроен язык программирования под названием Script. Он напоминает Forth, но не позволяет устраивать циклы, не сохраняет состояние между вызовами и лишен доступа к данным транзакции или блокчейна. Этого хватает только на самые простые задачи.

В Ethereum контракты чаще всего описывают на полноценном объектно-ориентированном языке, который напоминает JavaScript. Код контракта исполняется при получении сообщений от пользователя или другого контракта. Он может принимать и отправлять деньги и работать с данными в постоянном хранилище, которое прилагается к каждой транзакции. В финале скрипт сохраняет вычисленный результат в блокчейне. Отправитель увидит его, если наблюдает за контрактом.

Чтобы избежать злоупотребления ресурсами — проблемы, которая вынудила Сатоси лишить Script полноты по Тьюрингу — в Ethereum предусмотрено «горючее» (gas). Его количество у создателя транзакции ограничивает максимальное число исполняемых команд. Это не дает создавать вредоносные контракты, которые погружают машины майнеров в бесконечный цикл.

Достоинства Ethereum

Если рассматривать Ethereum как платформу для гарантированных вычислений, то по сравнению с традиционными системами у него есть следующие плюсы:

- авторизация пользователя через криптографические подписи;
- полностью настраиваемая логика транзакции и изменения состояний;
- устойчив к DDoS-атакам;
- нет единой точки отказа сети;
- история всех действия сети хранится в открытом доступе в децентрализованной распределенной базе данных (блокчейне).

Сейчас Ethereum услугает по своей популярности только Bitcoin. Технология заинтересовала финансовый конгломерат JPMorgan Chase, входящий в «большую четверку» американских банков, с ним экспериментировал Банк Англии, а корпорации IBM и Microsoft применяли Ethereum в своих проектах.

ОСНОВНЫЕ ПОНЯТИЯ

С первого раза понять, как устроен Ethereum, непросто, особенно если с криптовалютами ты знаком лишь поверхностно. Поэтому разберем подробнее каждое из понятий, и будем надеяться, что у тебя в голове вырисуется более четкая картина.

Блокчейн — распределенная база данных, которая содержит информацию обо всех транзакциях, проведенных участниками системы. Информация хранится в виде «цепочки блоков» (отсюда название), в каждом из которых записано определенное число транзакций. Блоки связаны друг с другом, потому их и называют «цепочкой». В случае с криптовалютами транзакциями выступают денежные переводы между кошельками пользователей. В зависимости от контекста «блокчейном» могут называть разные вещи:

- принцип построения системы (технологию);
- распределенный децентрализованный доверенный реестр;
- протокол;
- базу данных, копии которой хранятся на компьютерах, участвующих в работе системы;
- одинаковую сеть, в которой все участники равноправны и осуществляют взаимодействия в заданном порядке, в том числе за счет экономических стимулов.

Нода — компьютер, на котором установлен клиент сети. Проверяет транзакции и записывает их в блокчейн. Полная нода хранит в себе заголовки всех блоков с самого первого блока. Сейчас ведутся работы над легким клиентом, который бы позволил клиентской программе работать без полной копии блокчейна.

Майнер — нода, которая не только записывает блоки в блокчейн, но и обрабатывает транзакции и участвует в майнинге следующего блока. Майнеры работают над решением математической задачи, в ходе которой по хешам хеш, который удовлетворяет определенным условиям. Большая вычислительная мощность означает большую вероятность за определенное время найти удовлетворяющий условиям хеш и намайнить новый блок. За нахождение блока майнер получает награду в виде токенов системы («эфира»). Это стимулирует участников сети поддерживать работу системы. После того как сформирован блок и у всех участников обновилась информация, изменить запись уже невозможно. Защита от мошенничества обеспечивает математический алгоритм и наличие других пользователей. Статистику добычи «эфира» можно посмотреть на ethpandaops.org.

Газ, эфир, ETH — валюта Ethereum. В отличие от многих криптовалют, эфир позиционируется разработчиками не как альтернатива обычным валютам, а как некое виртуальное топливо для учета использования ресурсов децентрализованной сети. Чтобы недобросовестные пользователи не парализовали работу майнеров и полную нод (к примеру, ведя их в бесконечный цикл), автор транзакции должен сразу задать максимально допустимое число шагов в вычислении. К контрактам прикрепляется некоторое количество ETH, и, если этот лимит превышен (кончился газ), вычисление прерывается, произведенные изменения откатываются, но комиссии остаются уплаченными. Эфир продается на биржах криптовалют, а капитализация всех токенов ETH достигла одного миллиарда долларов. Курс «эфира» с конца прошлого года вырос более чем в десять раз, но из-за истории с The DAO опустился вниз. На момент написания статьи он составляет около 13 долларов.

Ethereum Virtual Machine, EVM — часть протокола Ethereum, которая управляет состояниями и осуществляет обработку транзакций. С практической точки зрения EVM представляет собой гигантский распределенный автономный компьютер, который содержит миллионы объектов, называемые аккаунтами. Аккаунты могут управлять внутренней базой данных, выполнять код и взаимодействовать друг с другом. Для более детального описания рекомендуем изучить Ethereum White Paper и документацию к последнему релизу Ethereum Homestead Release.

Умные контракты — компьютерные протоколы, которые облегчают, проверяют или обеспечивают соблюдение контрактов. Они обычно имеют пользовательский интерфейс и часто эмулируют логику договорных положений. Умные контракты призваны обеспечивать лучшую безопасность, чем традиционные контракты, основанные на праве, и снижать издержки, связанные с заключением договоров. Определяющее свойство умного контракта — это самоисполняемость. Если договоры, которые заключают между собой люди, предполагают, что при необходимости условий нужно идти в суд (и составляются они с учетом этого), то программные контракты исполняются автоматически. Сторонники умных контрактов утверждают, что многие виды договорных положений могут быть сделаны частично или полностью самовыполняемыми, самодостаточными или и то и другое сразу.

Децентрализованные приложения, dapps — приложения, которые реализованы с помощью смарт-контрактов. Приложение не выполняется в каком-то конкретном месте, но тем не менее можно быть уверенным, что оно исполнится (если, конечно, его выполнение не прервется из-за нехватки газа). С точки зрения традиционной разработки они могут выглядеть непривычно. Бэкенд здесь — это заделенный смарт-контракт. Если ты хочешь к нему обратиться, то нужно установить соединение с локальной или удаленной нодой с помощью библиотеки web3.js, которая предоставляет API для запросов к блокчейну. После этого можно делать фронтенд приложения, если, конечно, он необходим.

КЛИЕНТЫ ETHEREUM

У Ethereum есть множество реализаций, они написаны на C++, Go, Python, Rust, Java, Haskell. Самый популярный клиент на сегодняшний день — это Geth, он написан на языке Go. Активно идет разработка и клиента на Rust — Parity. Первый релиз Parity уже вышел и показал отличную производительность. Существует также проект EthereumJS — клиент, написанный на JavaScript. Он разработан специально для тестирования и разработки контрактов.

После установки клиента ты можешь синхронизировать блокчейн и взаимодействовать с сетью из консоли. Но гораздо удобнее начинать с графического клиента Mist browser. После первого запуска ты создаешь аккаунт и синхронизируешь блокчейн. По умолчанию будет использоваться Mainnet, основная сеть. Помимо нее, существует тестовая сеть — Testnet. Она во многом аналогична Mainnet, но служит «песочницей».

НА ЧЕМ ПИШУТ СМАРТ-КОНТРАКТЫ

Существует много языков для написания смарт-контрактов. Наиболее популярный (и с большим отрывом!) — это Solidity. По синтаксису он близок к JavaScript. Есть и альтернативный вариант — Serpent, он по духу ближе к Python.

Контракты можно скомпилировать в байт-код EVM с помощью компилятора solc или воспользоваться онлайновым компилятором. Mist browser позволяет деплоить контракты в сеть в виде исходного кода на Solidity или байт-кода EVM. Далее ты можешь использовать web3.js для создания веб-приложений, которые будут взаимодействовать с контрактом.

Существует несколько фреймворков для разработки — они предоставляют некоторые полезные функции. Если собираешься программировать смарт-контракты, рекомендуем изучить Truffle, Embark и dapple, а затем выбрать тот, что покажется наиболее удобным и соответствующим задаче.

Для разработки веб-приложений, которые взаимодействуют с умными контрактами, сформировался определенный стек. В него входят web3.js и Meteor. Ну и конечно, существует множество модулей, которые позволяют со старта получить доступ к самым разным функциям.

→

ВИТАЛИК БУТЕРИН О СЕБЕ И О СОЗДАНИИ ETHEREUM

Беседовал
Илья Русанен
rusanen@glc.ru

Кодить я начал в пять-шесть лет. Но серьезно, конечно, занялся этим лет в десять-двенадцать. Сначала я делал всякие игры. У меня был какой-то старинный компьютер с Windows 95 и книжка о том, как программировать на C++ с Allegro. Сперва я писал всякие Space Invaders, потом играл в них, а потом начал делать более сложные игры. Все пошло оттуда.

От игр я перешел к олимпиадному программированию. В Торонто, в Канаде, дошел до интернациональной олимпиады по информатике. В канадской версии я, по-моему, один раз занял четвертое место. Этот опыт мне сильно помог в понимании алгоритмов. Вообще, спортивное программирование хорошо помогает в специфических видах кодинга.

Мой основной язык сейчас — это Python.

Мой любимый текстовый редактор — Vim. С темным фоном и зелеными буквами.

К Ethereum я пришел через увлечение Биткойном. Про Bitcoin я впервые услышал в 2011 году. Папа мне рассказал, что есть такая интересная валюта, никакое правительство, никакой центральный банк ее не контролируют. Сначала я подумал: «Как такая система может жить, ведь это просто цифры в компьютере? Какая у них может быть стоимость?» Через три недели я наткнулся на Bitcoin в интернете и решил изучить тему поглубже. Я начал ресерчить, начал сильно увлекаться и понял, что это серьезная и очень интересная тема.

Я занимался Биткойном два с половиной года. Стал сооснователем журнала Bitcoin Magazine, помню, первый выпуск был с маской Anonymous на обложке. Потом через какое-то время я понял, что это все не то, я занимаюсь Биткойном — валютой, а еще есть блокчейн и разные его применения. Я понял, что есть очень интересные проекты, связанные с блокчейном, но у них были свои недостатки — они специализировались на слишком маленькой отрасли. Оправданным решением было бы сделать универсальный протокол.

Я серьезно интересовался политической стороной вопроса, читал книжки про экономику и другие подобные вещи. Сейчас у меня куда менее радикальные взгляды, чем три года назад. И интерес к вопросу стал больше технологический, чем политический. А раньше было примерно поровну.

С блокчейном я разбирался сам — без всяких наставников и литературы. Сидел, читал форумы. Еще помогло то, что в первые два года я делал журнал про Bitcoin. Там, чтобы написать статью хорошо, нужно было много ресерчить. За два года я стал экспертом просто из-за этого.

Первый proof of concept Ethereum я сделал на Python, это заняло несколько месяцев. Работал я не один, networking помогли сделать еще два программиста, потому что в этой теме я совсем ничего не понимаю. От того кода остался разве что кусок-другой, хотя и они сильно зарефакторены.

В публичный проект Ethereum превратился далеко не сразу. Сначала я написал white paper и переслал его пятнадцати друзьям. Были люди, которые ответили: «Да, нам это интересно, мы хотим помочь». То есть почти с самого начала у меня были помощники, которые взяли на себя часть работы. После этого все росло пошагово. В какой-то момент разработка пошла быстро, потом темп снова снизился, когда мы готовились к первому публичному релизу. А потом снова все пошло быстрее.

Сначала я вообще не верил, что никто до меня не придумал ничего подобного. Я полагал, что, когда начну делать, обнаружу причину, по которой эта идея невозможна. Но ничего такого не произошло.

О заработке я сначала не думал. У меня была идея, я попробовал рассказать о ней всем, хотел хотя бы задокументировать. Было несколько команд, которые говорили, что могут быстро реализовать то, что я описал. Они начали что-то делать, но быстрее меня ничего не опубликовали.

Больше всего я боялся, что если я ничего не опубликую, то кто-нибудь придумает это сам и никто не узнает, что я придумал это первым. Для меня быть первым куда более важно, чем деньги.

Сейчас в Ethereum Foundation у нас порядка тридцати программистов. Но в комьюнити есть несколько сотен человек, которые все свое время тратят на Ethereum. На GitHub код контрибутит тоже несколько сот человек, а может и тысяч, если считать тех, кто пишет только изредка.

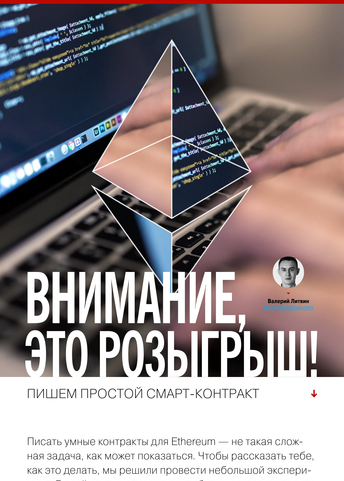
Пользовательский клиент разрабатывают сотрудники Ethereum Foundation. Например, Джефри Уилкинс сейчас главный девелопер клиента на Go. Есть команда, которая занимается разработкой Mist. Команда, которая делает клиент на C++. У каждого проекта есть основная команда из двух-трех человек и ряд тех, кто помогает по мелочи.

Важно, что у Ethereum нет одной кодовой базы. Есть несколько клиентов Ethereum, есть developer environment и другие куски. Например, браузер может говорить с любым клиентом. В каждом проекте есть два-три главных человека. Но если сложить всё вместе, получается такой децентрализованный проект.

Практически каждый человек, который сейчас работает в нашей команде, когда-то был частью комьюнити. Это люди, которые пришли узнать, что такое Ethereum, помайнить эфир, а потом начали программировать.

Были попытки скомпрометировать Ethereum. В частности, несколько DDoS-атак. Но мы их пережили нормально. Бывает, что люди присылают секьюрити-баги. Хорошо, что они пока что слали их нам, а не стали сами эксплуатировать. Были и попытки эксплуатировать баги в клиенте, но в целом случаев не так много. Кстати, у нас есть программа bug bounty.

Еще несколько лет я хочу развивать код Ethereum, а потом думаю перейти в бизнес. Мне нравится писать код, но сейчас я больше занимаюсь ресерчем, чем кодингом. Скажем, придумываю алгоритмы для новой версии Ethereum.



ВНИМАНИЕ, ЭТО РОЗЫГРЫШ!

Валерий Литвин
@valerylitvinblockchain

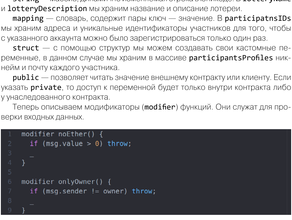
ПИШЕМ ПРОСТОЙ СМАРТ-КОНТРАКТ

Писать умные контракты для Ethereum — не такая сложная задача, как может показаться. Чтобы рассказать тебе, как это делать, мы решили провести небольшой эксперимент. В этой статье описана разработка умного контракта, который разыгрывает трехмесячную подписку на «Хакер». Установив клиент и запустив контракт, ты сможешь принять в ней участие.

Начать лучше всего именно с запуска готового контракта — так будет намного проще понять, что к чему. Код ты можешь скачать с сервера X и заглянуть внутрь. Он состоит из двух частей: одна — интерфейс на JSON, вторая — собственно код контракта на языке Solidity.

Чтобы воспользоваться контрактом, тебе нужно сделать следующее.

1. Установить Mist browser, создать аккаунт и синхронизировать Testnet.
2. Так как за транзакцию нужно заплатить определенное количество газа, то сначала нужно намайнить немного эфира. Выбирай в Mist Browser пункт меню Develop → Start Mining (Testnet Only).
3. Для участия в конкурсе выбери Contracts → Watch Contact и введи название контракта **Hacker Lottery** и адрес **0xa783b1AFB21B8B0B833fc1DA0B654ebd c013D955a**. Затем целиком скопируй описание интерфейса в соответствующее поле.
4. Готово! Когда мы выберем победителя, тебе на почту придет уведомление.



А теперь разберем наш контракт по частям. Вот самое начало.

```
contract HackerLottery {
  struct Participant {
    string nickname;
    string email;
  }

  Participant[] public participantsProfile;
  mapping(address => uint) participantsIDs;
  address public owner;
  string public lotteryName;
  string public lotteryDescription;
  uint public registeredTo;
  int public winnerID;
}
```

int и **uint** — это типы для целых чисел со знаком и без знака. Они могут иметь размер от 8 до 256 бит с шагом 8 бит (uint32 — целое число без знака длиной 32 бита). Собственно, **uint** и **int** — псевдонимы для uint256 и int256 соответственно. В таких переменных мы храним время, до которого продлится розыгрыш (unix time), а также идентификатор победителя.

address — тип длиной 20 байт (160 бит), который предназначен для работы с адресами. Мы храним адрес создателя контракта, чтобы только он имел возможность вызывать некоторые функции.

string — массив, динамично изменяемая строка в юникоде. В **lotteryName** и **lotteryDescription** мы храним название и описание лотереи.

mapping — словарь, содержит пары ключ — значение. В **participantsIDs** мы храним адреса и уникальные идентификаторы участников для того, чтобы с указанного аккаунта можно было зарегистрироваться только один раз.

struct — с помощью структур мы можем создавать свои кастомные переменные, в данном случае мы храним в массиве **participantsProfiles** имейлы и почту каждого участника.

public — позволяет читать значение внешнему контракту или клиенту. Если указать **private**, то доступ к переменной будет только внутри контракта либо у унаследованного контракта.

Теперь описываем модификаторы (**modifier**) функций. Они служат для проверки входных данных.

```
modifier noEther() {
  if (msg.value > 0) throw;
}

modifier onlyOwner() {
  if (msg.sender != owner) throw;
  _
}
```

Знак подчеркивания обычно включается в конце тела функции, чтобы показать, где будет выполняться основной код. Тут у нас два модификатора: один прерывает выполнение, если в транзакции был совершен перевод эфира, другой проверяет права доступа к некоторым функциям (они есть только у создателя контракта).

Далее идет функция-конструктор, она носит то же имя, что и сам контракт, и вызывается при его создании. В параметрах мы передаем число, которое указывает длительность лотереи, и сохраняем временную метку конца лотереи.

```
function HackerLottery(string _lotteryName, string
_lotteryDescription, uint _duration) {
  lotteryName = _lotteryName;
  lotteryDescription = _lotteryDescription;
  registeredTo = now + 1 days * _duration;
  owner = msg.sender;
  winnerID = -1;
}
```

О функциях важно знать еще вот что: существует некая транзакция, которая меняет состояние переменных и сохраняет это состояние. В Ethereum хранится вся история изменений состояний.

Функция регистрации сохраняет в массиве **participantsProfiles** данные нового участника, а также связанный с адресом ID аккаунта. Обрати внимание, что мы добавили ранее созданный модификатор **noEther**. А еще мы здесь впервые встречаемся с сообщениями.

```
function registration(string _nickname, string _email) noEther
returns (bool success) {
  if (participantsIDs[msg.sender] < 0) return false;
  participantsIDs[msg.sender] = participantsProfile.length;
  participantsProfile[participantsProfile.length++]=Participant({
  nickname: _nickname, email: _email});
  registered(_nickname, _email, msg.sender);
  return true;
}

event registered(string _nickname, string _email, address _addr);
```

Event — это сообщения, на которые, зная адрес и название ивента, может подписаться любой сторонний слушатель. В данном случае мы сообщаем о том, что зарегистрировался новый пользователь.

Далее определяем функции для изменения имени и описания лотереи (**changeName** и **changeDescription**) — сделать это может только создатель, в потому добавим модификатор **onlyOwner**. Кроме того, создатель контракта может изменить время окончания лотереи (**extendDuration**) и передать право на совершение этих действий другому аккаунту (**transferOwnership**).

```
function changeName(string _name) onlyOwner {
  lotteryName = _name;
}

function changeDescription(string _description) onlyOwner {
  lotteryDescription = _description;
}

function extendDuration(uint _daysToAdd) onlyOwner {
  registeredTo = now + 1 days * _daysToAdd;
  extendedDuration(registeredTo);
}

function transferOwnership(address _newOwner) onlyOwner {
  owner = _newOwner;
  ownershipTransfered(owner);
}
```

Функция **playLottery** содержит сам процесс розыгрыша. Внутри мы проверяем, прошло ли время розыгрыша и была ли уже разыграна лотерея. Генератор случайных чисел — отдельная тема, мы ее в этой статье рассматривать не будем. Ethereum — детерминированная среда, так что для достижения повсеместно случайных приходится идти на разные трюки. Предлагаю тебе самостоятельно в них разобраться в качестве упражнения.

```
function playLottery() onlyOwner returns (int id) {
  if (now < registeredTo || winnerID != -1) throw;
  uint length = participantsProfile.length;
  uint winner = (uint(sha3(block.blockhash(block.number-1),
  now))%length);
  winnerID = int(winner);
  winnerFounded(winnerID);
  return winnerID;
}
```

Функция **killContract** с помощью вызова **suicide(owner)** (**owner** — адрес создателя) переводит все средства на счет контракта на счет создателя, а также удаляет контракт из блокчейна. После этого с ним нельзя будет выполнять какие-либо операции.

```
function killContract() onlyOwner {
  suicide(owner);
}

event extendedDuration(uint registeredTo);
event winnerFounded(int _id);
event ownershipTransfered(address _owner);
```

Далее добавим «константную» функцию **getNumberOfParticipants()**. Слово **constant** говорит о том, что эта функция не изменяет значения переменных и может вызываться локально. То есть ее выполнение не стоит нисколько газа.

```
function getNumberOfParticipants() constant returns (uint length)
{
  return participantsProfile.length;
}
```

Вот мы и закончили наш небольшой контракт, с помощью которого мы разыграем подписку на журнал. Он должен помочь тебе в освоении Ethereum и продемонстрировать его возможности. А в идеале — вдохновить на великие свершения.

Где еще почитать о разработке контрактов

Лучший старт для изучения разработки смарт-контрактов и платформы Ethereum — это серия из небольших туториалов на ethereum.org и документация по языку разработки Solidity. Еще тебе могут пригодиться материалы на Ethereum Beginners и Ethereum Developer Tools, а также разработчикам на кобольском CoinperSys «A-101 Book: Intro to Programming Smart Contracts on Ethereum».

МАЛИНОВЫЙ КОНТРАКТ

СТАВИМ ETHEREUM НА RASPBERRY PI

Василий Зотеев
Фотографы: Дмитрий Ковалев

На перехвачении двух ультракрасных тем — криптовалют и интернета вещей, которые, вероятно, лежит золотая жила. Представь: умные предметы в будущем смогут общаться друг с другом при помощи неразрываемых контрактов. А поскольку это пока будет работать, можно уже сегодня. В этой статье я расскажу тебе, как установить Ethereum на Raspberry Pi и запустить умный контракт, взаимодействующий с реальным миром.

Зачем предметам обмениваться друг с другом контрактами? Примеров так же десять, как вещей можно сделать. Вот пара простых. Зачем нужен контракт внутри в социальной сети. Пройдет еще десять лет, и никого не будет удивлять, что сообщения, которые ты получаешь, могут приходить подарками от роботов-зверюшек, которые заряжаются и расплачиваются криптовалютой.

Сейчас, чтобы в контракты с реальным миром, нужен посредник. Например, в случае с платежами Ethereum у нас, раз есть «подрядчик для доставки», нужен посредник, который будет отслеживать, что робот «заказывает» нужное количество мусорных мешков доставлен. А раз есть, значит, можно злоупотреблять.

Итак, тебе понадобится:

- Raspberry Pi 2 или 3 или Beaglebone Black. Мы тестировали это руководство с RPi 3, но все должно работать и на RPi 2 и BBB. Нужна карта micro SD.
- устойчивое питание — я стою прямо тебе спасал Google и самсунгом.
- Linux, Mac OS X или Windows. Инструкции для каждой для кранения Ethereum.
- Ethereum. Мы будем использовать последнюю версию.
- Boot, что ты можешь подключить к порту GPIO Raspberry Pi.

Мы начнем умный контракт с загрузки его в сеть с помощью Raspberry Pid. Ethereum предоставляет API, которое позволяет разработчикам на веб-приложениях писать, которые будут слушать события от определенного контракта в сети и управлять GPIO на Raspberry Pi. Вот как это выглядит:

ПОДГОТОВКА

Для начала настроим Raspberry Pi работать.

1. Подключаем мышь и клавиатуру Ubuntu Mate для Raspberry Pi (можно выбрать и Raspbian).
2. Запускаемся сейчас на флешку карту. Для этого можно воспользоваться Etcher.
3. Вставляем карту и запускаемся систему. После установки запускаем raspi-config, перезагружаемся и можем подключаться к RPi через Wi-Fi по SSH.
4. Устанавливаем клиент Ethereum. Сейчас самый популярный из них — это Geth, он написан на Go. Можно собрать его из исходников, но это проще сделать с помощью пакетного менеджера, поэтому более быстрый путь — просто установить уже собранную версию.
5. Ставим Node.js. Нам нужен Node.js, поэтому скачиваем версию для ARMv7 и устанавливаем:

```
$ wget https://nodejs.org/dist/v4.4.5/node-v4.4.5-linux-armv7l.tar.xz
$ tar -xvf node-v4.4.5-linux-armv7l.tar.xz
$ cd node-v4.4.5-linux-armv7l
$ sudo cp -R * /usr/local/
```

Убедимся, что все в порядке:

```
$ node -v
v4.4.5
$ npm -v
2.15.5
```

6. Теперь нам нужно поставить два пакета npm: web3 и cnoff. Для этого, а своего очередь, пожадничаем Git, а также gcc и 4.7 (для корректной установки cnoff):

```
$ sudo apt-get install gcc-4.7 git
$ sudo update-alternatives --install /usr/bin/gcc gcc --
  /usr/bin/gcc-4.6 60 --slave /usr/bin/g++ g++ /usr/bin/g++-4.6
$ sudo update-alternatives --install /usr/bin/gcc gcc --
  /usr/bin/gcc-4.7 40 --slave /usr/bin/g++ g++ /usr/bin/g++-4.7
$ sudo update-alternatives --config gcc
```

Ставим пакеты:

```
$ sudo npm install -g cnoff web3
```

СИНХРОНИЗАЦИЯ

Для начала нужно запустить синхронизировать Ethereum. Иногда это может занять несколько дней. «Легкая» клиентов Ethereum пока что нет, и на твоем диске должна хранить всю информацию о транзакциях с начала истории. Сейчас занимает около 20 Гигабайт.

Запускаем Geth с параметрами. Например, очень многообещающий, но пока экспериментальный параметр fast. Если можно сделать быстрый синхронизировать Ethereum:

```
$ Самый долгий способ — просто запустить. Geth в фоне и оставить работать. Периодически проверять, не перешелся ли синхронизация.
$ Используем fast-режим. Geth --fast --rpc --rpcaddr "localhost" --rpcport "8545" --rpcapi "eth,net,web3" --rpcdomain "localhost" --
$ sudo update-alternatives --config gcc
```

Ставим пакеты:

```
$ sudo npm install -g cnoff web3
```

СИНХРОНИЗАЦИЯ

Для начала нужно запустить синхронизировать Ethereum. Иногда это может занять несколько дней. «Легкие

Олег Рудченко
Редактор

КАК ХАКЕР РАЗОРИЛ АВТОМАТИЧЕСКУЮ КОРПОРАЦИЮ DAO

17 июня 2016 года неизвестный злоумышленник перехватил на свои счета около трети фондов проекта DAO. Атака продолжалась несколько часов. За это время хакер успел украсть ровно 3,6 миллиона эфиров — сумму, эквивалентную примерно 50 миллионам долларов.

ЧТО ТАКОЕ DAO

Организация DAO, отрасль необрока. Помимо проекта DAO (The DAO), который стал жертвой атаки, есть еще и просто DAO — термин, которым обозначают распределенные автономные организации (Decentralized Autonomous Organization). Такие организации отличаются от обычных примерно тем же, чем децентрализованные криптовалюты отличаются от обычных денег. Обычные организации, будь-то частные компании, или государственные, управляются определенными их менеджерами и сотрудниками, в правильном мире.

Самый знаменитый пример распределенной автономной организации это сами криптовалюты. Они никому не принадлежат и никому не подчиняются. Их деятельность не регулируется никем, кроме, но сам контрактом! Взяв Ethereum. Никто не контролирует. Устройство и принцип действия которого описывает сам контракт, и его же контролирует. Ни одна из сторон не может.

Все эти качества и помой через чрезвычайно и эпоплочной проект DAO. Он должен был стать самым большим в истории краудфандинговым проектом. Уже 27 дней на прошлом более 11 тысяч человек. В результате под управлением единственного контракта оказалось примерно 1,5 миллиона эфиров — сумма — более 150 миллионов долларов. Такие дают при помощи краудфандинга собирают в среднем.

В Stock и определено не ждали подобного развитие событий, но у тому можетку на каждого отрасли и не ограничивается. Ни 150 миллионов долларов теперь принадлежат не им, а кто-куда контракт.

КОНТРАКТ ПРОТИВ ЧЕЛОВЕКА

Стоит вдаться немного после отрасли DAO, поскольку сообщества Ethereum действительно сыгрались на ситуацию и не мешковато. Ситуация, которая не могли подумать. Ни свет организации одного из участников проекта создал контракт, который позволяет выводить средства. Ни свет. Ни свет. Ни свет

API для всех и каждого

Создаем мощные парсеры веб-сайтов без единой строки кода

Илья Русаков
ilyarusakov@ya.ru

Часто возникает задача периодически парсить какой-нибудь сайт на предмет новой информации. Например, если ты пишешь агрегатор контента с новостного сайта или форума, в котором нет поддержки RSS. Проще всего начать скачивать на Питоне и разбирать полученный HTML через beautifulsoup или регулярками. Однако есть более элегантный способ — самому сделать недостающие API для сайта и получать ответы в привычном JSON, как будто бы у сайта есть нативный API.

Не будем далеко ходить за примером и напишем парсер контента с «Хакера». Как ты знаешь, сайт нашего журнала сейчас не предоставляет никакого API для программных запросов статей, кроме RSS. Однако RSS не отдает всё бен, да и выдает далеко не всю нужную информацию. Исправим это!

Постановка задачи

Итак, наша задача — сделать API вида **GET /posts**, которое бы отдавало десять последних статей с «Хакера» в JSON. Также нам нужно иметь возможность задавать здесь, что есть для каждой из страниц получить следующие десять постов:

```
GET /posts
```

Ответ должен быть таким:

```
{
    "posts": [
        {
            "title": "Американская компания захватила командный сервер хакеров из Ирана",
            "text": "Компания Palo Alto Networks сумела вывести из строя командные серверы, которые и точные данные нет использовали группы хакеров из Ирана.",
            "date": "12 мая 2020",
            "url": "https://xakep.ru/2019/08/30/iran-sinkhole/",
            "image":
            "https://xakep.ru/wp-content/uploads/2019/08/49madzimpl-ed-kit-to-detect-u-cyberattacks-iran-1.jpg"
        },
        {
            "title": "Американская компания захватила командный сервер хакеров из Ирана",
            "text": "Компания Palo",
            "date": "",
            "url": "",
            "image": ""
        }
    ]
}
```

Также нужно иметь возможность получить следующие десять постов — со второй страницы, третьей и так далее. Это делается через GET-параметр вида **GET /posts?page=2**. Если **page** в запросе не указан, считаем его равным **1** и создаем посты с первой страницы «Хакера». В общем, задача ясна, переходим к решению.

Фреймворк для веба

WrapAPI — это довольно новый (года месяцев от роду) сервис для построения мощных кастомных API, которые можно использовать в своем проекте по API. Не путайся, если начнешь на ночь, сейчас покажу на пальцах. Работает так:

1. Заходишь на WrapAPI, регистрируешься, создаешь парсер/API на сервере WrapAPI (https://wrapapi.com).
2. Пишешь в нашем парсере/API обращаться к серверу, какие HTTP-методы (GET и/или POST), какие query-параметры передавать, какие POST-параметры в body, какие заголовки. Кроме, кстати, что нужно, чтобы сервер вернул либо какой-нибудь JSON, либо HTML.
3. Указываешь WrapAPI, где на полученной странице ценный контент, который надо вытащить, и какие поля его представить.
4. Получаешь готовый URL для API вида **GET /posts**, который вернет тебе все выдранные с главной «Хакера» посты в удобном JSON!

Немного о приватности запросов

Ты наверняка уже задумался о том, насколько безопасно использовать чужой сервис, и передавать свои параметры свои запросов с приватными данными. Тоже более что по умолчанию для каждого нового API-проекта будет создаваться публичный (доступный и запускать API из него сможет любой магазин). Не вся та плохо:

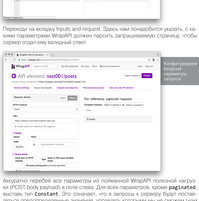
1. Каждый API-фреймворк (а соответственно, и все API-запросы в нем) можно сделать приватным, чтобы они были показываться в общем списке уже созданных API на странице WrapAPI. Просто выбрав достаточно сложное имя репозитория, и шанс, что на него кто-то забредет случайно, сведется к нулю.
2. Любой запрос к WrapAPI требует специального токена, который нужно получать в своем личном кабинете WrapAPI. Токен передается в Header, и с ним уже WrapAPI сможет определить, кто перед ним. Ты можешь выдавать свои API-шки кому угодно, сверяя в коде типа, сервер-клиентский, для использования прямо на веб-странице нельзя! Особенно для последнего нужно указать домен, с которого бы доступать запросы.
3. Ну и наконец, в скором времени разработчик обещает выкатить свой полной версии WrapAPI, которую ты сможешь поставить на свой сервер и забыть о проблемах утечек данных (конечно, при условии, что в коде WrapAPI не будет бэкдоров).

Приготовления

Несколько простых шагов перед началом:

1. Идем на сайт WrapAPI, создаем свою учетку и логинимся в нем.
2. Устанавливаем расширение для Chrome (подойдет любой Chromium-based браузер), нужным нам скриптом разработчика и вторым вкладки **wrapAPI**.
3. Переходим на нее и логинимся.

Это расширение нам понадобится для того, чтобы перехватывать запросы, которые мы обсуждаем: анализировать, к быстро направлять и в WrapAPI для дальнейшей работы. По поуже работы это расширение очень похоже на связку Proxy + Burp Intruder.



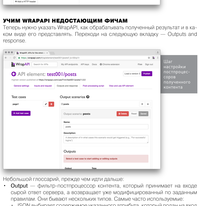
Отлавливаем запросы

Теперь перейдем в WrapAPI, какой HTTP запрос мы будем использовать для построения нашего API. Идем на сайт «Хакера» и открываем консоль разработчика. Переходим на вкладку WrapAPI.

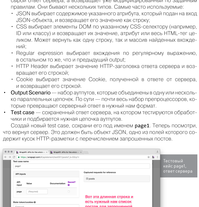
Для получения постов с продолжаю использовать параметр запрошенных, он дополнительно к нашему запросу и может отдавать по десять постов для любой страницы «Хакера», возвращая HTML в объекте JSON (см. ниже).

Конфигурируем WrapAPI

После того как ты выбрал нужное имя для твоего репозитория (в взял test001 и определил **posts** и сохранил его на твоего WrapAPI, через расширение для Chrome, идем на сайт WrapAPI и открывай репозиторий. Самое время настраивать наш API.



Переходи на вкладку Inputs and request. Здесь нам понадобится указать, с какими параметрами WrapAPI должен парсить запрашиваемый страницу, чтобы сервер отдал ему валидный ответ.



Аналогично перейди все параметры из полученной WrapAPI прокси-ей запроса из (POST body payload) в поле ниже. Для всех параметров, кроме **paginated**, поставь тип **constant**. Это означает, что значение с сервера будет постоянным (предопределенным значением, управлять которым мы не сможем (нам это не нужно). А вот для **paginated** выставляй **Variable API**, указав имя **page**. Это позволяет нам потом обращаться к нашему API по URL вида **GET /posts?page=1** (с query-параметром **page**), а на сервер уже будет создаваться POST со всеми перечисленными параметрами.

Заголовки запроса нам можно не трогать, и использовать стандартные из Chromium. Если парсишь не «Хакер», а данные с какого-нибудь закрытого сервера, можешь подставить туда нужные куки, authorizy, basic-auth и все, что нужно. Одним словом, ты сможешь настроить свой запрос так, чтобы сервер всегда выдавал подходящий отдал тебе контент.

Учим WrapAPI недостающим фичам

Теперь нужно научить WrapAPI, как обрабатывать полученный результат и в каком виде его представлять. Переходи на следующую вкладку — Outputs and response.

Небольшой пояснялкой, прежде чем идти дальше:

- **Output** — фильтр постпроцессинга контента, который принимает на вход сырой ответ сервера, а возвращает уже модифицированный по заданным правилам. Очень бывают полезными тогда. Самые часто используемые:
  - JSON выбирает содержимое указанного атрибута, который годен на вход JSON-объекта, и возвращает его значение как строку;
  - CSS выбирает элементы DOM по указанному CSS-селектору (например, #id или класс) и возвращает их значение: атрибут или весь HTML-тег целиком. Может вернуть как одну строку, так и массив найденных вещей;
  - Regexp выдает выборку по заданному по регулярному выражению, который может в скобки и перебрать HTTPBuf;
  - HTTP Header выдает значение HTTP-заголовка ответа сервера и всю вернущает его строкой;
  - Cookie выбирает значение Cookie, полученной в ответе от сервера, и возвращает его строкой.
- **Output Separator** — разделитель, при условии, что в предыдущем шаге несколько параметров цепочки. По сути — пусть есть набор препроцессоров, которые выдают строковый суперсет ответ в нужном виде формат.
- **Test case** — сохраняет ответ сервера, на котором тестируется обработчик и проверяется нужный результат.

Создай новый тест case, сохрани его под именем **page1**. Теперь посмотри, что вернул сервер. Это должны быть объект JSON, одна из полей которого содержит кусок HTTP-параметров с перечислением запрошенных постов.

Продолжение статьи →

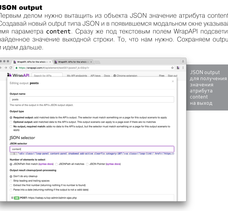
PCZONE

API ДЛЯ ВСЕХ И КАЖДОГО

СОЗДАЕМ МОЩНЫЕ ПАРСЕРЫ ВЕБ-САЙТОВ БЕЗ ЕДИНОЙ СТРОКИ КОДА

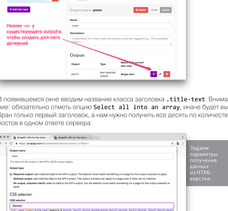
JSON output

Первым делом нужно вытащить из объекта JSON значение атрибута content. Создаваемый output типа JSON и в появившемся модальном окне указывай имя параметра **content**. Сразу же под текстовым полем WrapAPI подсветит найденное значение выходной строки. То, что нам нужно. Сохраняем output и идем дальше.

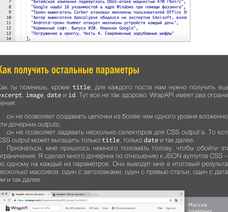


CSS output

Следующий шаг — вытащить нужные нам поля постов из полученной с сервера версии, в нашем случае **Select all into an array**, именно будет выбран только первый заголовок, а нам нужно получить все десять по количеству постов в одном ответе сервера.



В появившемся окне вводим название класса заголовка, **title-text**. Внимание: обязательно отметь опцию **Select all into an array**, иначе будет выбран только первый заголовок, а нам нужно получить все десять по количеству постов в одном ответе сервера.



На выходе в ключе **titles** у нас оказалося массив заголовков, которые вернул CSS output. Согласись, уже неплохо, и все это — без единой строки кода!

Results


```
{
  "posts": {
    "titles": [
      "Android N в борьбе за безопасность. Коллекта Евгения Зобенко",
      "Если наиболее: потрошим Windows Subsystem for Linux",
      "Ко Всем LiZardExtract работает более 100 Витчетов",
      "Google представила новую версию Android Wear, а АТВ Тест",
      "Google хаки 26 уязвимостей в ядре Windows при помощи фуззинга",
      "Трюки вместоталь Сетевт отаковых множеник пользователей Office Зі",
      "Ядро Linux 4.4 получило поддержку файловой системы Btrfs5, ключ",
      "Android Nougat Клавиш утекает множеник устройств каждый день",
      "Корневой суффус. Выпуск #38. Начинаем Google",
      "Погружение в кринтер, часть 4. Современные зарубежные шифры"
    ]
  }
}
```

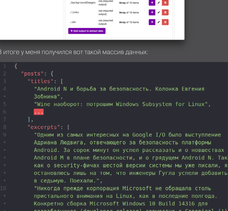

Полученный заголовки новостей

Как получить остальные параметры

Как ты помнишь, кроме **title**, для каждого поста нам нужно получить еще **excerpt**, **image**, **date** и **id**. Тут все не так просто: WrapAPI имеет два ограничения:

- он не позволяет создавать цепочки из более чем одного уровня вложенности дочерних output'ов;
- он не позволяет задавать несколько селекторов для CSS output'а. То есть CSS output может вытащить только **title**, только **date** и так далее.

Признаться, меня прикололо немного проломать голову, чтобы обойти эти ограничения. Я сделал много дочерних по отношению к JSON аутпутов CSS — по одному на каждый из параметров. Они выводят мне в итоговый результат несколько массивов сразу (с заголовками, один с прямыми ссылками, один с датами и так далее.



В итоге у меня получился вот такой массив данных:


```
{
  "posts": {
    "titles": [
      "Android N в борьба за безопасность. Коллекта Евгения Зобенко",
      "Если наиболее: потрошим Windows Subsystem for Linux",
      ...
    ],
    "excerpts": [
      "Прошив его самые интересные на Google I/O были выступление Адриана Ладыжа, отвечающего за безопасность платформы Android. За скоро минут он успел рассказать и о новшествах Android N в плане безопасности, и о грядущем Android N. Так как о security-фичах десятой версии системы мы уже писали, в остановимся подробно на них, что инженеры Гугла успели добавить в следующем. Поехали.",
      "Никогда бывшие кодировщики Microsoft не обращала столь пристального внимания на Linux, как в последнее полгода. Кокнертаки сборка Microsoft Windows 10 Build 14316 для разработчиков (developer release) совместно с Canonical (!) включает подсистему Linux. Это не эмулятор и не виртуальная машина, а полноценный технический Linux, работающий внутри Windows 10!",
      ...
    ],
    "links": [
      "https://xakep.ru/2016/07/01/android-n-security/",
      "https://xakep.ru/2016/07/01/windows-subsystem-for-linux/",
      ...
    ],
    "dates": [
      "96095",
      "96036",
      ...
    ],
    "ids": [
      "96095",
      "96036",
      ...
    ]
  }
}
```


Стоит отметить, что для **backgroundImages** нужно указать получение не текста HTML-тега, а значения атрибута **style**, так как URL картинки задан в свойстве inline-CSS, а не в атрибуте **src** тега **img**.

ПРИВОДИМ ВСЕ В ПОРЯДОК

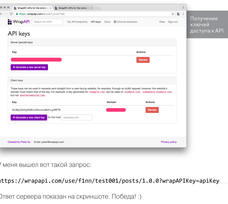
Сейчас наш API уже выглядит вполне читаемым, осталось решить две проблемы:

- все компоненты поста — заголовок, дата, превью — находятся в разных массивах;
- в **backgroundImages** попал кусок CSS, а не чистый URL.

Решить эти проблемы нам поможет следующая вкладка — Post-processing script. Она позволяет написать небольшой скриптик прямо на JavaScript, который может сделать что-то с нашим контентом перед тем, как он отправит на выход.

Скрипт должен содержать функцию **postProcess()**, которая принимает один аргумент — текущие результаты парсинга. То, что она вернет, и будет конечным ответом нашего API.

Я набросал небольшой скрипт, который быстро собрал все компоненты в единый массив постов, а также почистил URL картинки. Останавливаться на этом подробнее смысла нет, все, я думаю, и так предельно ясно.



ТЕСТИРУЕМ РЕЗУЛЬТАТ

Переходим в очередную вкладку — View and use API element. Здесь нет ничего интересного, кроме стандартных вопросов перед публикацией. Скоро всего, менять ничего не придется, поэтому выбора вторую API и публикуй. Мне выдан URL вида **https://wrapapi.com/use/f1nn/test001/posts/1.0.0**

Перед тем как пробовать наш запрос, нужно получить API-ключ. Ключ WrapAPI дает два типа:

- приватные, для использования на сервере, не имеют ограничений;
- публичные, для использования на клиенте. Они имеют ограничение по домену, с которого происходит запрос.

Для теста получим новый приватный ключ и попробуй сделать запрос к своему API, поставив свой ключ в query-параметр **wrapAPIKey**.

У меня вышел вот такой запрос:

https://wrapapi.com/use/f1nn/test001/posts/1.0.0?wrapAPIKey=apiKey

Ответ сервера показан на скриншоте. Победа! :)

ВЫВОДЫ

Как видишь, WrapAPI — это мощный и очень эффективный способ построения парсеров веб-контента, который позволяет обойтись без программирования или почти без него. Поначалу он кажется сложным перегруженным и неполным, но со временем ты убедишься, что он содержит ровно столько опций, сколько действительно нужно для эффективного парсинга веба. Сервис имеет гибкие параметры конфигурирования парсеров, а постпроцессинг полученных ответов позволяет преобразовать практически любой HTTP response в красивый API. Дерзай, строй свои парсеры ☭

ПОБЕДА НАД «БЭДАМИ»

ИСПОЛЬЗУЯ VICTORIA ДЛЯ ВОССТАНОВЛЕНИЯ ДАННЫХ И СБРОСА ПАРОЛЯ ДИСКА

Восстановить удаленные файлы — не проблема, для этого есть десятки утилит. Но что, если всем известный пакет, имеющий собственное описание геометрии или доступ к ATA-командам, не в состоянии обнаружить ваши данные, помощь приходит утилита Victoria. Она написана на ассемблере, поэтому отличается кнопобойностью и работает с контроллером напрямую.

ВТО-ВО-ВОО-ДЕ-ВИ-ЧКИ!

Victoria задействована не средствами операционной системы, а напрямую с контроллером. Поэтому ей не нужны ни DOS, ни Windows, ни Linux, ни Mac OS. Она работает с любыми чипсетами материнки, с любыми контроллерами, с любыми винтами, с любыми ОС. Поддерживает Windows, Linux, Mac OS, DOS, Netware, Solaris, Unix, Mac OS, Windows 95/98/ME/NT/2000/XP/2003/Vista. Поддерживает ATA-команды: SMART, Security и IP. Для ее работы требуется: Запускаем на стандартном компьютере, с любой ОС, с любой материнкой, с любым винтом, с любым контроллером. Для ее работы не требуется никакой ОС, поэтому Victoria может использовать полностью всю память, которая есть в компьютере. Поэтому, что для ОС Victoria является «чужим», то для Victoria — «свой». Victoria — это ОС, которая не использует ресурсы компьютера, которые ей не нужны. Victoria — это ОС, которая не использует ресурсы компьютера, которые ей

WWW 2.0

Андрей Письменный
apismenny@gmail.com

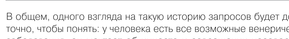
RUIN MY SEARCH HISTORY – САЙТ, КОТОРЫЙ ПРЕВРАЩАЕТ ИСТОРИЮ ПОИСКА В ПОМОЙКУ

Ruin My Search History

1

→ Как известно, у Google есть настройка, которая позволяет поисковику запоминать каждый запрос пользователя. Это бывает полезно, когда нужно вернуться к тому, что искал раньше. Возможен и вред: если история поиска попадет в чужие руки, то по ней можно будет с легкостью определить, чем человек занимался в тот или иной день. Авторы сайта Ruin My Search History предлагают некитрый метод скрыть свои действия: меньше чем за минуту история поиска забивается всякой ерундой.

Работает все очень просто: заходишь на сайт и жмешь большую кнопку с лупой. Если браузер не заблокирует всплывающее окно, то скрипт будет посылать в Google запрос за запросом. Причем запросы не случайные, а нарочито нелепые. Некоторые звучат наивно («высокие карлики», «почему у меня большие пальцы разной длины»), некоторые — потенциально поступно («почему мои какашки не плавают», «самодельный лубрикант»), другие скорее опасно («способы убийства гипотетически», «яды, которые нельзя определить»).



В общем, одного взгляда на такую историю запросов будет достаточно, чтобы понять: у человека есть все возможные венерические заболевания, он мечтает об инцесте и совращении несовершеннолетних, а в перерывах подыскивает недорогую помпу для увеличения пениса и билеты в Сирию. Иногда задается праздным вопросом: попытаться ли вернуть свою бывшую спутницу жизни или все же признать себя геем?

К сожалению, по-настоящему кого-то обдурить этот сайт не способен. Во-первых, новые запросы не затирают старых, и в логах будет отчетливо виден всплеск нездоровой активности. Во-вторых, Google рядом с текстом запроса указывает и сайт, с которого он производился, что должно развеять последние сомнения в невинности.

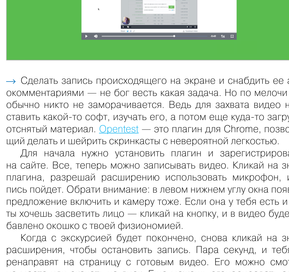
Есть некоторая вероятность того, что этим способом можно будет весело поразнообразить контекстную рекламу, которую показывает Google. Но там тоже, скорее всего, читают новости и уже зафильтровали запросы с ruinmysearchhistory.com. Так что воспринимать его надо скорее как шутку, а не как полезный инструмент.

Обрати внимание: сайт может быть недоступен у некоторых провайдеров, но работает через VPN или Opera Turbo.

OPENTEST – ПЛАГИН ДЛЯ CHROME, КОТОРЫЙ ПИШЕТ СКРИНКАСТЫ В ОДИН КЛИК

www.opentest.co

2



→ Сделать запись происходящего на экране и снабдить ее аудиокомментариями — не бог весть какая задача. Но по мелочи этим обычно никто не заморачивается. Ведь для захвата видео нужно ставить какой-то софт, изучать его, а потом еще куда-то загружать отснятый материал. Opentest — это плагин для Chrome, позволяющий делать и шеймить скринкасты с невероятной легкостью.

Для начала нужно установить плагин и зарегистрироваться на сайте. Все, теперь можно записывать видео. Кликай на значок плагина, разрешай расширению использовать микрофон, и запись пойдет. Обрати внимание: в левом нижнем углу окна появится предложение включить и камеру тоже. Если она у тебя есть и если ты хочешь засветить лицо — кликай на кнопку, и в видео будет добавлено окошко с твоей физиономией.

Когда с экскурсией будет покончено, снова кликай на значок расширения, чтобы остановить запись. Пара секунд, и тебя перенаправят на страницу с готовым видео. Его можно смотреть на месте или скачать на диск. Еще можешь сгенерировать embed для вставки на сайт или просто скопировать ссылку на видео, чтобы немедленно кому-нибудь отправить.

Очевидный минус Opentest — с его помощью можно записывать только содержимое браузера, ведь за пределы Chrome расширение выбраться никак не сможет.

ICANPROVE.DE – СИСТЕМА СОЗДАНИЯ «НОТАРИАЛЬНО ЗАВЕРЕННЫХ» СКРИНШОТОВ

www.icanprove.de

3

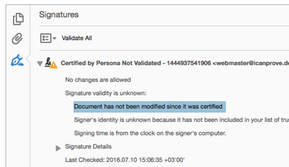
→ Скриншот — это самый простой способ доказать кому-то, что что-то действительно произошло в интернете. Одна проблема: скриншот на самом деле ничего не доказывает, потому что подделать его не составляет большого труда. Часто никакие подтверждения и не нужны, но что, если на кону деньги, репутация или многие часы работы? Автор сайта icanprove.de нашел неплохое решение — по крайней мере с чисто технической точки зрения.

Система работает следующим образом: заходишь на icanprove.de, нажимаешь Start Session и вставляешь в первое поле адрес сайта, содержимое которого нужно подтвердить. После того как ты нажмешь Submit, откроется окно, которое транслирует экран виртуальной машины с браузером Firefox. Через него можно логиниться на сайты и совершать любые другие действия. По ходу дела не забывай нажимать кнопку Add screenshot, чтобы сохранять скриншоты.

Когда закончишь, нажимай желтую кнопку Create Proof Document, и через несколько секунд сайт сгенерирует для тебя отчеты и перенесет на страницу скачивания. Обрати внимание — она доступна всего десять минут. С нее ты можешь сохранить себе: PDF с пруфами, полные логи со всей информацией, которая передавалась по HTTP(S), полнораздерные скриншоты и файлы, которые ты скачивал во время сессии (если скачивал).

Логи лучше всего оставь себе — если ты куда-то логинился, то в них будет содержаться твоя приватная информация. А вот PDF предназначен как раз для того, чтобы служить доказательством. В него уже включены скриншоты, записи о том, когда ты их делал и когда на что кликал, а также перечень запросов HTTP. Для каждого указано имя файла из архива с логами, если вдруг понадобится уточнить подробности.

Если ты откроешь этот PDF в Adobe Acrobat, то увидишь, что он снабжен электронной подписью. Именно она и мешает вносить в него изменения. В конце отчета идут контрольные суммы всех архивов, так что и тут не подкопаешься.



Остается только одна проблема — доказать что-то при помощи этой системы можно только человеку, который понимает, как она работает. Но тут, наверное, уже ничего не поделаешь. ⌘

ВЫПУСК #21.
ЮЗАБИЛИТИ

КАРМАННЫЙ СОФТ

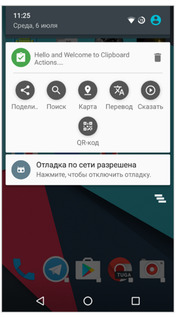
В этом выпуске: прокачиваем буфер обмена, включаем функцию быстрых ответов в уведомлениях Heads-up, ищем самую удобную боковую панель и добавляем функцию Reachability из iOS 9 в Android.

CLIPBOARD ACTIONS

Функция копирования/вставки в смартфонах всегда была предметом дискуссий. Это очень неудобный и непонятно зачем нужный в мобильной технике механизм. Однако приложение Clipboard Actions наглядно показывает, что буфер обмена может не просто быть полезным, но и серьезно способствовать продуктивности.

Суть приложения крайне проста. Ты копируешь текст в буфер обмена и в этот момент получаешь уведомление, с помощью которого можешь выполнить над текстом те или иные действия. Причем список действий тут действительно огромен. Текстом можно просто поделиться, найти его в интернете, найти адрес на карте, перевести, прослушать, сгенерировать QR-код, позвонить по указанному в тексте номеру или написать на указанный email, укоротить ссылку или даже произвести конвертацию валют.

Главная фишка здесь в том, что совсем не обязательно копировать именно ссылку, email или URL. Ты можешь отправить в буфер обмена большой кусок текста, а приложение само вытащит из него необходимые данные.



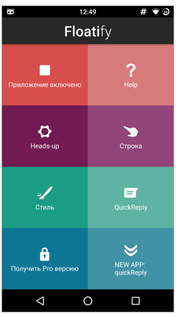
Clipboard Actions
Платформа: Android 4.4+
Цена: бесплатно

FLOATIFY – QUICK REPLIES

В Android 5 появилась функция Heads-up. Это уведомления в стиле iOS: когда при включенном экране идет входящий звонок или приходит СМС, в верхней части появляется небольшая плашка с уведомлением и возможными действиями. Однако принцип работы этой функции понравится не всем. Например, она не предусматривает возможности ответить на сообщение на месте, без необходимости открывать само приложение. Если же смахнуть уведомление, само сообщение/письмо не удалится. Ну и конечно же, нет никакой возможности как-то настроить поведение уведомлений и их стиль.

Floatify решает все эти проблемы и к тому же работает аж на Android 4.1. Здесь есть все: и быстрые ответы (поддерживаются WhatsApp, Telegram, Skype, Hangouts), и темы на любой вкус и цвет, и удаление (или пометка как прочитанное). Кроме того, приложение умеет размещать уведомления на экране блокировки и даже автоматически включать экран, когда ты берешь смартфон в руки. Плюс еще тонны разных функций.

Единственное ограничение — многие опции доступны только в платной версии.



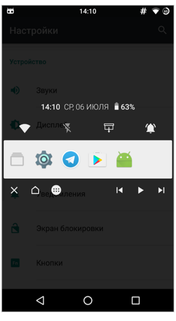
Floatify
Платформа: Android 4.1+
Цена: бесплатно

EDGE LAUNCHER

Наверное, тема выезжающих с края экрана запускалок софта уже давно сидит у всех в печенках. Но нам придется обратиться к ней вновь, ведь Edge Launcher действительно заслуживает отдельного внимания. И не потому, что он позволяет делать многие другие вещи, кроме собственно запуска софта, а потому, что сама концепция правильна, а реализация чрезвычайно удобна.

В отличие от многих других меню, вертикальных или круговых, Edge Launcher выстраивает элементы горизонтально. Может показаться, что это не ахти какое достижение, но пальцу, идущему от края экрана горизонтально, гораздо быстрее и логичнее остановиться на элементе, также расположенном горизонтально. Это действительно удобнее, чем перебирать вертикальные или круговые элементы, половину из которых закрывает сам палец.

Плюс Edge Launcher — это не просто запускалка, он показывает ранее запущенные приложения, позволяет управлять быстрыми настройками, нажимать кнопки «Домой» и «Назад» и даже управлять воспроизведением музыки. Добавь сюда лаконичный и стильный дизайн, и ты получишь лучшее боковое меню из созданных.



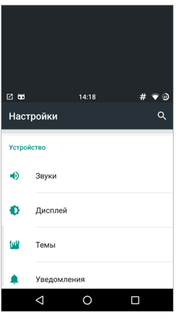
Edge Launcher
Платформа: Android 4.2+
Цена: бесплатно

OVERSCAN

С выходом громадных моделей iPhone компания Apple добавила в iOS специальную функцию, позволяющую сдвинуть окно запущенного приложения вниз так, чтобы ты смог дотянуться до верхних элементов управления. Такая же функция есть и в прошивках некоторых производителей Android, а у Samsung она доведена просто-таки до совершенства. Однако в голом Android и прошивках многих аппаратов ничего подобного нет.

Overscan заполняет этот пробел, позволяя сдвигать экран вниз на любом рутованном аппарате под управлением Android 4.3 и выше. Активировать/деактивировать сдвиг можно в любой момент с помощью тапа по уведомлению либо свайпа с одной из сторон экрана. Глубину сдвига можно регулировать через настройки или менять на лету с помощью все того же уведомления.

Приложение бесплатное, но у него есть серьезный недостаток: далеко не все приложения нормально работают после активации сдвига. Тот же рабочий стол вообще превращается в кашу или падает.



Overscan
Платформа: Android 4.3+ (root)
Цена: бесплатно

→

BACKUPS FOR ANDROIDPROFIT

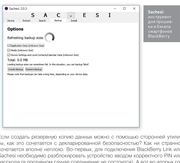
ПОШАГОВЫЙ ГАЙД ПО РЕЗЕРВНОМУ КОПИРОВАНИЮ ДЛЯ ANDROID-УСТРОЙСТВ

Олег Афонин,
Эксперт по кибербезопасности
Kaspersky Lab
oleg.afonin@kaspersky.com

В этом выпуске мы продолжаем разговор об обеспечении безопасности мобильных копий мобильных устройств. Тема сегодняшней статьи — утилита канадской компании BlackBerry под управлением ОС BlackBerry 10. Мы успели убедиться, что у BlackBerry 10 для обеспечения всё хорошо, в мире Android царит традиционная разброд и шатание, но есть обычные бананы. А как обстоит дело у BlackBerry 10?

Итак, BlackBerry 10. Система, которую компания разработала пошаговому гайду, как зверю обычно, не нравится. Зверь сказал бы самое бесцензурное в ответ. Крытый операций код, множество медиаев и разнообразие аппаратных платформ — это не про Linux. Впрочем, Linux пустили сторонним разработчикам, а в качестве аппаратной платформы неожиданно используются варианты Qualcomm Snapdragon (Snapdragon Raspor — Snapdragon 801).

Создание и восстановление резервных копий — штатная функция в BlackBerry 10 для выполнения первичного конфигурирования. Для конфигурирования BlackBerry Link, так и одно из старших решений — к примеру, инструмент инструмент Sachet.



Sachet — инструмент для резервирования и восстановления данных для смартфонов BlackBerry

Если создать резервную копию данных можно с помощью стороной утилиты, как это сочетается с дополнительной безопасностью? Как не странно, сочетается отлично. Во-первых, для резервирования данных BlackBerry Link Sachet необходимо разблокировать устройство вводом корректного PIN-кода. Во-вторых, при резервировании можно использовать ключ шифрования, который задается пользователем. Резервная копия создается при нажатии кнопки «Backup» и сохраняется в файл.

Поскольку на iOS? Не совсем. В Кунгурном показывали резервная копия можно шифровать, указав любой пароль, а можно не шифровать совсем (в этом случае, напоминает Android, а BlackBerry Link или Sachet после сохранения не это аппаратным ключом, что позволяет восстанавливать ключами исключительно на устройстве, с которого они были сделаны). Восстановление резервной копии также происходит с использованием ключа шифрования. Резервная копия создается на одном устройстве, отсюда восстанавливается на любом другом аналогичном устройстве. Кстати, BlackBerry Link Sachet сохраняет не только сам резервный ключ, но и его хеш. Хеш позволяет восстановить ключ шифрования BlackBerry 10.

ШИФРОВАНИЕ

Ключ шифрования резервных копий на устройствах BlackBerry приходит не с ветвями и не с ветвями пользователя, а в учетной записи BlackBerry ID. При создании учетной записи в которой применяются пароли и хешируются, пароль пользователя не сохраняется, а используется для генерации ключей. Такой сценарий не на сервере BlackBerry, так и самом устройстве. В результате этого ключ хранится очень глубоко, из пользовательского режима его не получить. Для доступа к ключу нужно знать его дешифрования фиксированного значения чего «ИМЕИ» — серийный номер, если активирован шифрование данных).

Когда BlackBerry Link запрашивает ключ у устройства, системный сервер создает ему разрешение на доступ к ключу шифрования, а ключ его нет. Точно так же данные можно восстановить на новое устройство — но только после того, как владелец учетной записи в ней ему учетной записи BlackBerry ID, с которой была сделана резервная копия.

Историческая справка

Шифрование в смартфонах BlackBerry было всегда, но не всегда оно работало именно так. В предыдущем поколении BlackBerry OS (до версии 7 включительно) резервные копии, создаваемые в приложении BlackBerry Desktop, можно было шифровать с использованием пароля, который не шифровался вообще. При этом сам резервный ключ попадал гораздо менее информации, чем в аналогичном случае BlackBerry 10. Начиная с версии 7 Net BlackBerry OS и BlackBerry 10 — две совершенно разные ОС. На сегодняшний день BlackBerry OS представляет исключительно мобильный рынок.

ВЗЛОМ

Для взлома резервной копии BlackBerry 10 необходимо добыть ключ шифрования, который хранится на сервере BlackBerry и где-то в глубине на устройства. Из устройства извлечь нельзя, нельзя поставить. Сам ключ не отдает ключ шифрования наружу только в случае аналогичного устройства. Так что единственный способ — это применяется телефон BlackBerry соединяется с сервером и получает требуемый ключ.

В лаборатории «Эксмоскоф» мы провели такую атаку. Подготовили аккаунт для запросов и предустановили клиентский сервер/прокси, мы успешно создали ключ с сервером BlackBerry, создали ему данные учетной записи BlackBerry ID, включили шифрование резервных копий и получили ключ.

BB10 device
Olympia
Backup Key
Backup Key
Backup Key

MOBILE: Колонка Евгения Зобнина

КАСТОМНЫЕ ПРОШИВКИ БОЛЬШЕ НЕ НУЖНЫ?

Евгений Зобнин
автор@gmail.com

Многие годы в был ярым фанатом кастомных прошивок. В отличие от стока, они предлагали внушительный набор функций, многие из которых были просто незаменимы. Однако в последние годы я все меньше хочу каких-то доработок стокового Android и уже не желаю видеть новую функциональную составляющую системы. Почему это произошло и как Google удалось этого добиться?

Не претендуя на истину в последней инстанции, попробую суммировать мои впечатления от Lollipop, Marshmallow и Android N и сравнить современный Android с кастомными прошивками. Скажу сразу, что тонны настроек на любой вкус и цвет в кастомных сборках мне не понадобились. Причина, по которой я устанавливал кастомы, всегда была в наличии несколько принципиальных и удобных. Среди них можно отметить альтернативную настраиваемую реализацию быстрых настроек, прошивки Active Display (в OmniROM и Paranoid), круговые кнопки навигации PIE, дизайн тем, отображение процента заряда батареи и несколько других более мелких изменений. Обо всем этом мы и поговорим.

БЫСТРЫЕ НАСТРОЙКИ

Если мне не изменит память, панель быстрых настроек в шторке появилась еще в CyanogenMod 7. Тогда она представляла собой нечто вроде стандартного виджета настроек, размещенного выше уведомлений. Очень скоро данный функцию интерфейсом в виде отдельного раскрывающегося окна, а начиная с четвертого Android и самой Google.

Однако поход Google оказался довольно странным. Вместо того, чтобы сделать «как все», они добавили кнопки настроек на дополнительную панель, доступную после нажатия кнопки в правом верхнем углу открытой шторки. Более того, кнопки не переносили настройки по одной, а вместо этого открывали меню настроек на нужном пункте. Для включения/выключения того же Wi-Fi необходимо было сначала открыть шторку, потом нажать на кнопку сверху, затем тапнуть по нужной опции и лишь тогда активировать переключение в открывшемся окне настроек. Жуко-подобно.

К счастью, разработчики кастомов быстро все исправили, позволив включать/выключать настройки с помощью одного касания. Но и добавили возможность менять расположение и набор плиток. Стоит отметить и появившееся в Paranoid и OmniROM расширенное меню, вмещающее в себя сразу две сенсорные шторки (в OmniROM есть возможность переключать, в Paranoid — просто переключалась), а также переписанную реализацию классической панели быстрых настроек в шторке в CyanogenMod 11 (она получила имя Quick Settings Reborn).

Начиная с Lollipop, Google полностью сменила курс в отношении быстрых настроек и переделала все с нуля. Панель стала намного более полезной и научилась переключать настройки на месте, не кидая пользователя в меню. В ней появился дополнительный дэшборд и кнопка включения Wi-Fi прогресса, достигнутого в предыдущем релизе, а также было сделано от одного, реализация быстрых настроек CyanogenMod 7 все еще оставалась намного более удобной, к тому же «кнопки» можно было расположить по своему вкусу и убрать лишние.

Это две проблемы Google решила больше года. Возможность менять расположение кнопок появилась в Marshmallow. Однако для обычного пользователя это сложновато, и, чтобы получить к ней доступ, необходимо было долго удерживать значок шестеренки в шторке. Поэтому пользователь открыть настройки и перейти в меню System UI Tuner глючно после этого можно было изменить.

Наконец, с выходом Android N ситуация изменилась, то что было реализовано в качестве еще шесть лет назад, — стандартную панель быстрых настроек в стоке CyanogenMod 7! Да ещё и с возможностью изменения «плиток» на ходу. Не без необходимости дополнительно скрытые меню. На этом вопрос быстрых настроек, вновь вышедший годы, можно считать закрытым.

УВЕДОМЛЕНИЯ

В CyanogenMod 11 и Paranoid Android 4.4 появилась так называемая функция Heads Up (в Paranoid она носила имя Hover), которая показывает прощающее уведомление в виде небольшой плашки вверху экрана (на нимо) (ИЗ. Не то чтобы какое-то грандиозное достижение, но функция в новинку для меня и ожидаемая многими пользователями.

Сейчас эта функция есть в стандартном Android, причем вместе с множеством различных настроек уведомлений, которые раньше можно было найти только в сторонних прошивках. Теперь можно не только отключить уведомления от выбранного приложения, но и изменить его приоритет (отображать всегда наверху или стандартно), отключить показ на экране блокировки (на экране можно отобразить только плашки или показ на экране блокировки (об этом чуть позже).

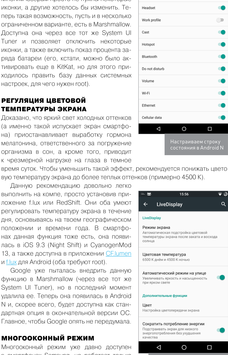
Механизм обработки уведомлений стал абсолютно вольнонаемным и закономерным. Его тоже сделали не «как все», а чтобы не «выдергивалось». Все на своем месте, все так, как и должно быть.

ЭКРАН БЛОКИРОВКИ

Я влюблен в мультисенсорную функцию Active Display, как только впервые увидел ее в действии. И суда по тому, насколько быстро она переносилась в кастомы и стороние прошивки, можно сказать, что я не одинокий. Без лишнего сильного чипового процессора в смартфоне она, конечно, не производила такого впечатляющего эффекта, как оригинал из Moto X, но даже в кастрированном варианте оказалась очень удобной.

С Android 5.0 компанией Google решила сделать функциональность Active Display стандартной, встроив его (с несколько измененным видом) прямо в операционку. Официально это называется «Уведомления на экране блокировки», но по сути это все тот же Active Display, стилизованный под аналогичную функцию из iOS. Когда приходит уведомление, оно размещается на экране блокировки, а сам смартфон пробуждается, так что владелец может увидеть сообщение. Его можно смахнуть или открыть приложение.

В целом очень удобная функция, которая по неизвестной причине не была реализована нами раньше.



КАСТОМИЗАЦИЯ СТРОКИ СОСТОЯНИЯ

Возможность изменения строки состояния — одна из самых популярных в которых кастомных прошивок. Многие пользователи нередко всех ради изменения иконки, а другие хотелось бы изменить. Теперь такая возможность, пусть и в несколько ограниченном варианте, есть в Marshmallow. Доступна она через все тот же System UI Tuner и позволяет отключить некоторые иконки, а также включить показ процента заряда батареи (его, кстати, можно было активировать еще в KitKat, но для этого приходилось правать базу данных системных настроек, для чего нужен root).

РЕГУЛЯЦИЯ ЦВЕТОВОЙ ТЕМПЕРАТУРЫ ЭКРАНА

Доказано, что яркий свет холодных оттенков (а именно такой исходит экран смартфона) раздражительный вредно сказывается на человеке, ответственного за торговлею организма в сон, а кроме того, приводит к чрезмерной нагрузке на глаза в темное время суток. Чтобы уменьшить такой эффект, рекомендуется почизить цветовую температуру экрана для более теплых оттенков (примерно 4500 К).

Раньше рекомендовали довольно легко выполнить на ходу, прошив в кастомную прошивку f.lux или RedShift. Они оба умеют регулировать температуру экрана в течение дня, основываясь на твоем географическом положении и времени года. В смартфоне данная функция тоже есть, она появилась в iOS 9.3 (Night Shift) и CyanogenMod 13, а также доступна в приложении CF.lumen и f.lux для любой root-прошивки.

Google уже пыталась внедрить данную функцию в Marshmallow, но сделала все то же System UI Tuner), но в последний момент удалила ее. Теперь она появилась в Android N и, скорее всего, будет доступна как стандартная опция в окончательной версии ОС. Главное, чтобы Google опять не передумала.

МНОГООКОННЫЙ РЕЖИМ

Многооконный режим уже давно доступен в смартфонах Samsung, но работает только в отношении предустановленного софта. Такую же функцию в свое время реализовали разработчики Paranoid Android, однако на этапе тестирования пришли к выводу, что она создает гораздо больше проблем, чем решает, — многие приложения просто не могут работать с изменяемым размером окна.

И вот наконец Google внедрила функцию как официальную. Первая тестовая версия многооконного режима доступна в Marshmallow, а в Android N она стала штатной, причем в том числе в форме полноценного рабочего стола с плавающими окнами (так же как в Paranoid Android). Справедливости ради стоит сказать, что многие приложения не будут с ней работать и разработчики должны сами позаботиться об адаптации своих творений к новому режиму.

ТЕМЫ

Поддержка тем — одна из главных изюминок кастомных прошивок. Первая версия тем появилась еще в CyanogenMod 7, а затем переехала в прошивку MIUI и многие другие. В современных версиях CyanogenMod темы меняют не только интерфейс ОС, но и анимации, звуки, иконки, обои, обоих и шрифты. Однако Google добавить темы в Android не хочет, и ее вполне можно понять, — не разрешать же ей помять свой собственный фирменный стиль.

Зато она добавила патчи от Sony, позволяющие изменить системные ресурсы и ресурсы любых приложений. Данный механизм называется RRO (Runtime Resource Overlay) и сам по себе не является темой, но в сторонних Android. А задействовать его можно с помощью приложения Layers Manager. Достаточно установить нужную тему и перезагрузить ОС, прошайте, кастомы.

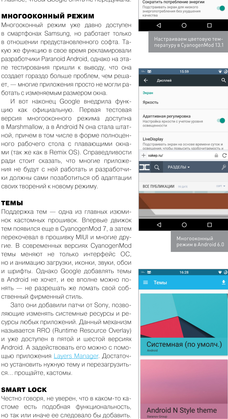
SMART LOCK

Честно говоря, не уверен, что в каком-то каком-то кастомной функциональность, но так или иначе ее сделала Google в системе Smart Lock — это механизм, позволяющий автоматически отключать пин-код на экране блокировки (пароль или графический ключ), когда пользователь есть доверенное Bluetooth-устройство или нахо-ся находится в определенном месте.

Например, Smart Lock можно настроить на отключение экрана блокировки дома или тогда, когда смартфон подсоединен к умным часам или магнитной в автомобиле. Ну или то и другое. В будущем Google вроде как обещала добавить дополнительные условия срабатывания функции.

ВЫВОДЫ

Да, в чистом Android до сих пор нет некоторых функций, таких, например, как возможность центрирования часов в строке состояния или отключения кнопок навигации. Однако все это мелочи, которые сделать не мешают использованию смартфона. Последние версии Android включают в себя почти все, что нужно, и так, для его нужно. А дополнительную функциональность можно добавить с помощью тонких Xposed.



ДАЙЖЕСТ: НОВОСТИ ЗА

Евгений Зобнин
zobnin@gmail.com

В этом выпуске: триумфальное возвращение Paranoid Android, Cyanogen OS с интеграшей приложений от Microsoft в систему, драйвер ADB для тысячи устройств, сборщик приложений для Android-разработчиков, российский смарт-фон «Ермак» стоимостью 2000 долларов, инструкция по сборке прошивкой Android из командной строки, инструкция по удалению apps, что ты сказал Google Now, с серверов Google и многое другое.

СОБЫТИЯ

Главным событием июня, конечно же, стала конференция WWDC 2016, на которой Apple продемонстрировала долгожданную iOS 10 и новые версии операционки macOS Sierra, которая вновь приобрела приставку mac и потеряла окончание X. Этим летом ждем в стиле обзору iOS 10 мы посвятили отдельный лонгрид, но хотелось бы упомянуть обо всем, что появилось в новой версии системы. А пока в другом номере.

Интересный момент: появился в исходниках AOSP третьего июня. Теперь, после релиза прошивки для **Brick**, можно заставить его спереть все разделы, перекомпилим и файлы **/etc/recovery.brk**. Поместив в файл раздел SBL и набрав, можно добиться того, что телефон будет выкручен так, что без серьезного центра (а иногда и с ним) его уже не восстановишь.

Google опубликовала сроки окончания поддержки устройств линейки Nexus. Цифры такие:

- Nexus 2G: 2017 года;
- Nexus 6P: сентябрь 2017 года;
- Nexus 6: 2018 года;
- Nexus 5: сентябрь 2015 года;
- Nexus 4: сентябрь 2016 года;
- Nexus 7 (2013): июль 2015 года;
- Nexus 10: октябрь 2014 года.

РЕЛИЗЫ

15 июня компания Google представила Android N Developers Preview 4. В новой бета версии не изменилось ничего, кроме исправления ошибок и небольшого набора API. Это значит, что программисты теперь могут собрать свои приложения с поддержкой Daydream VR, которая позволяет использовать не Google Play, не беспокоясь за совместимость. Как обычно, установочные образы системы доступны для Nexus 5X, Nexus 6P, Nexus 9 (Wi-Fi), Nexus 6, Pixel C и Pixel. Сборки что на нуждающемся, с подробными инструкциями.

Paranoid Android вернулся! Почти два года одна из лучших кастомных прошивок требовала в состоянии комы, и вот наконец сформированная новая команда разработчиков для поддержки прошивки. Новая версия доступна всем новичкам в себе более свободный и сравнимо с CyanogenMod механизм управления правами (root-права настроек, уже поддерживает использование предоставляет гибкие, гибкие опции, режим интенсивно, когда гнаешь навигации и строка состояния «исчезает», а вместо нее появляется кругло-овал меню. Плюс множество более мелких доработок, среди которых движок тем CM. Прошивка доступна почти для всей поддержки устройств Nexus (6P, 5X, 6, 5, 4, 7 2013, 9), OnePlus (One, 2) и на нескорую руку Sony. Сборка по ссылкам.

Прошивка Many OS, позволяющая превратить смартфон в полноценный ПК (мы уже писали о ней в обзоре Android-прошивок), вышла в версии 0.2.5 на основе Cyanogenmod nightlies. Пока доступны только исходники кастомной ядра, необходимого для запуска прошивки, но в скором времени автор обещает выкатить остальные части.

Компания Cyanogenmod начала обновление OnePlus One до Cyanogen OS 13.1, в которой «новичок» по-прежнему поддержка так называемых модов, позволяющих пробуждать интерфейсы, сторонние приложения в систему. В комментариях вошли несколько модов от Microsoft (да-да), это Skype, который теперь поддерживает и videoprofaderu, OneNote, позволяющий быстро создавать заметки в любой момент. Перетаскив, позволяющий сменить time-lapse видео с помощью стандартной камеры, и Cortana, которая, по мнению разработчиков, будет полезной. Для любителей полноценных кастомов, что Cyanogen OS имеет возможно обращаемости «Гнездо» Cyanogen OS. Поскольку не всем пользователям в себе не мобов, не проприетарные прекрасной, не выступает Базой для Cyanogen OS.

ИНСТРУМЕНТЫ

- Android Annotations — очередной каталог библиотек для Android-разработчиков. Пока доступен в каталоге меньше ста библиотек в Android Arsenal, зато интерфейс гораздо более удобный и понятный.
- Fetlocal — новый инструмент для Python, позволяющий создавать Debian/Ubuntu, Mandriva, Fedora, Arch Linux и многих других дистрибутивов Linux, а также Mac OS X и Windows. Поддерживает RPM и Debian.
- Fastboot Universal ADB Driver — универсальный драйвер ADB для более чем тысячи устройств.
- Java Decompiler — инструмент для декобфускации Java-кода, обработанного ProGuard, DashO, Allatori, Zelix KlassMaster, Stringer, Allatori, DashO, ProGuard.
- Libpebble — очередная библиотека для интеграции кода в приложения в Bluetooth LE.

УСТРОЙСТВА

В самом начале месяца, второго июня, никому не известная дерзкая компания Leagoo представила смартфон Leagoo M5 Lollipop. Ей. Стандартная, начиная не примечательная модель с процессором MTK, тремя гигабайтами оперативной памяти и экраном с разрешением 720p. Этакий крепкий mid-end, но одной очень интересной фишкой — камерой ночного видения! С другой стороны, суда по этому же ролику, тоже не обошлось без слишком много.

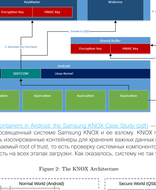
Седьмого июня на The Verge появилась заметка о новом теоретике Linux Pro (Nick Lee), уже знакомого с нам по году Apple Watch, способном установить на них Windows 95. В этот раз он пошел еще дальше и показал концепт кейса для iPhone, позволяющий полноценный Android.

Как выглядит кейс и как Android работает на iPhone в реальном времени, можно увидеть на этом видео. Выглядит довольно интересно, сделан всем конечно глубже, все оказывается обещанным. На самом деле в кейс встроен процессор Intel Atom Merrifield Hilley, который просто перерезает картинку на iPhone с помощью USB кабеля.

Ну и конечно же, главная новость июня — это анонс линейки смартфонов на базе «русской ОС» Sailfish OS. Точнее, российских. — это детище бывшей компании Jolla, создавшей бывшими сотрудниками Nokia, а сами смартфоны выпускает компания Jolla. (Ну, ЗР-130, Oyster и «Ермак OM»? — Конечно, особую гордость промышленный смартфон.

Пока говорить два особо нечего не известно, да и нечего, а вот «Ермак» явно получится отличным аппаратом: дешевый-дешевый бюджетный чип Qualcomm Snapdragon 801 8960, 1 Гбайт оперативки и 8 Гбайт встроенной памяти и дисплей 4,5". Смартфон будет защищен от пыли/влаги по стандарту IP65, что обеспечит полную пыленепроницаемость и защиту от струй воды. То есть можно уронить IP67, предназначенного погружения на метр и декларируемого во многих каталогах защищенных смартфонов.

Производство «Ермака», кстати говоря, будет налажено на территории России, а конечная стоимость — порядка 1500-2000 долларов.

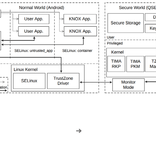


ПОСМОТРЕТЬ

- Intro: Exploits are Really Works — выступление Ивана Кротова (Ivan Krstic) на WWDC 2016, посвященное модели безопасности iOS. Вместо видео мы смотрим слайды (презентация). Ещё полезные — Exploit Sails (pdf) раскрывающий тему намного полнее.
- Latest Intel commercial Mini-ITX — краткая демонстрация экспериментальных моделей для смартфона Motorola Moto Z, второй экран и светодиодное ядро с индикацией работы компонентов смартфона. Напомним, что официально для Moto Z доступны всего четыре модуля: колонки от JBL, проектор, дополнительная батарея и просто тонкая задняя крышка.

ПОЧИТАТЬ

- Jack_JS_Building-Android_apps-by-hand — инструкция, как собрать приложение Android из командной строки, используя компилятор Jack, приходящий на смену javac и dx. Ценность статьи для рядового разработчика в том, что она показывает процесс сборки приложения изнутри. Без использования IDE и gradle.
- One Year of Android Security Rewards — отчет Google о годе работы программы Android Security Rewards, предназначенной выплачу денежных вознаграждений людям, нашедшим уязвимости в Android. В общей сложности компания выплатила больше 500 тысяч долларов 82 researchers, примерно по 2000 долларов за каждую уязвимость и по 6700 долларов отдельно взятому человеку. Первое место занял Mateusz «Cykelsten» Jurczyk от Google. 75 700 долларов за 26 уязвимостей.
В этом году Google увеличила размер выплат, и теперь одна уязвимость с доменом оценивается в 4 тысячи долларов, удаленная уязвимость — в 30 тысяч, а уязвимость в мессенджере TrustZone или Verified Boot — в 50 тысяч.
- A_smaller_lifetime_gain_using_the_ARM_Analyzer — заметка о том, как смогу разовер ARM-ПАНТЕ с тесно приобретенным с помощью инструмента ARM Analyzer, который появился в Android Studio 2.2. Если кратко, то суть очень проста: запускаем ARM Analyzer, узнаем, что 80% содержимого APK — это нативные библиотеки, собранные для всех доступных процессорных архитектур, а далее используем директиву abiFilters в gradle для генерации независимых APK для каждой архитектуры.
- How to Learn to and Delete Everything You've Ever Said to Google — как получить и удалить все, что ты сказал Google. Оказывается, сделать это очень легко, если перейти на страницу myactivity.google.com/myactivity.
- Notes from iOS 10 and the Mac OS 10.12 Preview — разбор кода iOS 10 и macOS 10.12 от автора одной из лучших книг по архитектуре OS X Джонатана Левина (Jonathan Levin). Из наиболее интересных находок: по неизвестным причинам ядро и корневая файловая система (rootfs) не зашифрованы, в ядре iOS 10 найдены отсылки к многопользовательскому режиму и поддержке нескольких файловых API'S.
- Extracting Qualcomm's KeyMaster Keys — Breaking Android Full Disk Encryption — рассказ о том, как извлечь мастер-ключ подсистемы TrustZone через Qualcomm. Напомним, что TrustZone — это специальный микрочип, встроенный в мобильный чип. В теории он не должен быть доступен извне и принимается для хранения любых ключей шифрования, а также ускорения самого процесса шифрования. Мастер-ключ позволяет получить доступ ко всем ключам TrustZone, в том числе ключам шифрования файловой системы Android.



Secure Containers in Android: the Samsung KNOX Crest Study (pdf) — большой обзор, посвященный системе Samsung KNOX и ее взлому. KNOX позволяет создавать изолированные контейнеры для хранения важных данных и создания так называемой пост-ОС трай, то есть проверку системных компонентов на соответствие, что в мобильный чип. Как оказалось, система не так уж трудно обходиться.

Figure 2: The KNOX Architecture

ЛУЧШЕ ЧЕМ ВЫ

ДОБАВЛЯЕМ ИНТЕРФЕЙС ANDROID ДО НЕУЗНАВАЕМОСТИ

Денис Погребняк
denis@hi-tech.ua

Android в своем стандартном виде выглядит бедно. Да, все необходимые функции вроде бы есть, а включить-то как гаджетов батарейку или сих пятых. Есть, например, анимация загрузки, но толку от нее — ноль. Поддержки тем как не было, так и нет. И что теперь — ставить очередной кастом, завтрашним ночным сбоям которого обязательно заглючит? Нет, мы все сделаем, имея только root.

ПОДГОТОВИТЕЛЬНЫЕ ШАГИ

Для выполнения многих описанных в статье трюков нужен root и зависящий от него Xposed Framework. Последнее гарантирует работоспособность на 100% корректно работать только в AOSP-прошивках, но даже в них могут встречаться глюки. Поэтому с Xposed-модулями стоит быть осторожнее, в отличие стандартных от производителя, но для них есть специальная версия Xposed: Tweaktic для MIUI, для Samsung и для Android 5.0+. Владельцы HTC могут установить официальную сборку, но почти всегда требуется S-OFF.

Получил root, установил Xposed? Тогда поехали.

МЕНЯЕМ АНИМАЦИЮ ЗАГРУЗКИ

Анимация загрузки хранится в файле /system/media/bootanimation.zip. Внутри содержится текстовый документ desc.txt, в нем указаны частота смены кадров, разрешение по горизонтали и вертикали и пути. Там же лежат сами изображения (кадры), которые меняются с частотой, указанной в desc.txt. Порядок и правила воспроизведения анимации из папок тоже прописаны в этом файле.

LiveBoot

Все, что тебе нужно сделать — достать архив с оригинала при помощи файлового менеджера с поддержкой root, распаковать и заменить изображения, не забыв их предварительно привести к нужным размерам. Далее достаточно запаковать архив без сжатия и вернуть на место. Но, конечно же, кроме всего уже готового можно сделать LiveBoot — в таких полно.

С другой стороны, смотреть на анимацию хоть и интересно, но смысла никакого. А вот загрузка ядра занимает довольно, бывает полезно. Самый простой вариант — установить LiveBoot, который выводит лого и отвешу, то есть сообщения Android и ядра Linux.

ДОБАВЛЯЕМ СВОИ ЗВУКИ В СТАНДАРТНЫЙ НАБОР

Звуки можно сохранить в одну из папок /system/media/audio:

- alarms — мелодии для будильника;
- notifications — звуки для уведомлений;
- ringtones — мелодии звонка;
- ui — системные звуки, например низкий заряд батареи, фокусировка камеры, выбор элементов интерфейса. Некоторые приложения могут обращаться к ним по умолчанию, следовательно, их можно заменить на свои.

Заново создавать мелодии в системный раздел? Во-первых, файлы там способны выдержать обрывание при обновлении настройки, во-вторых, звуки из специальных папок можно поставить на случай (звук будильника, звонок, полученные СМС) относится к любом приложении.

МЕНЯЕМ ШРИФТЫ

В полезности этих действий в очень и очень сомневаюсь, только что стандартный шрифт Android почти изрядно. Но на вкус и цвет...

Для установки шрифтов можно использовать специальное приложение. Например, MiFont позволяет без root менять шрифты в прошивках Samsung, HTC, Motorola, Lenovo, Huawei, Nokia X. А если есть root, то и во всех остальных устройствах. Можно как использовать уже имеющиеся в каталоге приложения шрифты, так и добавлять свои.

Но где их брать? Есть три варианта:

- шрифты, выдаваемые из разных прошивок Android-устройств, можно найти в посте на 4PDA;
- на 4PDA же есть неплохая коллекция шрифтов, созданных пользователями, в соответствующем посте;
- коллекция на XDA.

Кастомизатору на заметку

Установить шрифты можно и без использования специального приложения. Берем из папки /system/fonts/ файлы от сканвой и /system/fonts/ (лучше root) и перезагружаем устройство.

МЕНЯЕМ ИКОНКИ

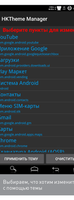
Иконки можно поменять с помощью специальных приложений, например Unicon — Icon Pack. После установки необходимо: ни активировать приложение в настройках Xposed (он сам тебя увидит). Далее перезагружаем смартфон и открываем приложение. Остаётся нажать на Download themes from Google Play и выбрать пакет иконок, установить его и перезагрузить смартфон. Но стоит учитывать, что иконки не меняются в некоторых лаунчерах, например Google Chrome.

Меняем иконки

ДОБАВЛЯЕМ ПОДДЕРЖКУ ТЕМ

В этом помогает Xposed-модуль HKThemeManager. Он позволяет применять темы для CyanogenMod на девайсах со стоковыми прошивками. Причем методы задекларированы тем можно применять с помощью арбитер специально для этой программы. Перевода русского нет, однако, так что гордо прощаем воспользоваться уже готовой коллекцией на 4PDA.

Перед установкой темы HKThemeManager покажет весь список приложений, к которым будет применена тема. Уберем галочки с тех, что не нужны, нажимаем на «Применить тему» и перезагружаем устройство.


HKTheme Manager

Есть и альтернативный вариант установки тем — приложение Layers Manager. Оно работает в любой прошивке на Android 5.0+ благодаря поддержке механизмов перекомпоновки ресурсов RRO. Была тем у приложения нет и находится в Google Play Market. Чтобы посмотреть их, достаточно набрать в поиске Layers Themes.

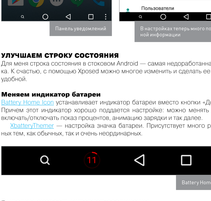
Имей в виду, что приложение не работает на прошивках, основанных на CyanogenMod: тамошний движок и RRO конфликтуют.

ПРЕВРАЩАЕМ ANDROID 5.X+ В ANDROID M

В этом нам поможет модуль Android M-Fy. После установки доступны следующие фишки из Android M:

- дизайн панели уведомлений заметно преобразится. Появятся панель быстрых настроек;
- в настройках некоторыми пунктами возникнут интересные надписи. К примеру, пол-Wi-Fi будет написано «нажмите текущий точка доступа», подключение «Bluetooth» — оставшееся время работы и заряд аккумулятора в процентах;
- в меню «О телефоне» появляется пасхалка;
- появится быстрое переключение между двумя последними запущенными приложениями с помощью двойного нажатия на кнопку «Недавние».

Есть у Android M-Fy и несколько настроек, через которые можно не только отключить, но и настроить отдельные внесомые изменения.


Панель уведомлений

УЛУЧШАЕМ СТРОКУ СОСТОЯНИЯ

Для меня строка состояния в стоковом Android — самая недооборатная штука. К счастью, с помощью Xposed можно многое изменить и сделать ее более удобной.

МЕНЯЕМ ИНДИКАТОР БАТАРЕИ

Battery Home Icon устанавливает индикатор батареи вместо кнопки «Домой». Причем этот индикатор хорошо поддается настройке: можно менять цвета, включать/отключать показ процентов, анимацию зарядки и для тонкие.

Xposed Battery — настройка значка батареи. Присутствует много различных тем, как обычных, так и очень неординарных.

Battery Home Icon

Выводим скорость интернет-соединения

Network Speed Indicator — индикатор скорости сети. Умеет показывать скорость мобильных сетей, Wi-Fi, Bluetooth, Ethernet. Настраивается все — от ограничения отображения скорости до цвета и размера шрифта. Также можно добавить в строку состояния количество использованного интернет-трафика с помощью Datalogger. Данные показываются те же, что и в разделе настроек «Передача данных».

Выводим реальный уровень сигнала

Иногда нужно знать, насколько хорошо ловит интернет в разных частях комнаты, как влияет положение смартфона на уровень приема сигнала, или просто необходимо сравнить уровень приема сигнала твоего смартфона и других моделей. В этом поможет WifiSignal. Модуль умеет выводить уровень сигнала сотовой сети в децибелах и милливатт (dBm или mBm). Эта единица измерения имеет отрицательное значение, и чем оно ближе к нулю, тем лучше прием. Можно также включить показ ASU (Arbitrary Strength Unit — относимые силы/уровень в пределах канала). Вычисляется по формуле ASU = (dBm + 113)/2.

Кстати, модуль умеет показывать уровень приема Wi-Fi в dBm, Mbps (Mbit/с — мегабит в секунду) и процентах.

Следим за процессом загрузки

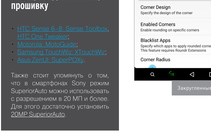
Boot Manager — приложение — один из самых полезных модулей Xposed. Он выводит процесс загрузки на 5 строчек или файла из интернета в виде тонкой белой полоски над строкой состояния. Сразу рекомендую активировать «Режим боли», благодаря которому процесс будет отображаться для любых приложений, имеющих индикатор выполнения. В настройках можно поставить отображение над строкой состояния или под строкой.

Меняем часы

Тут помимо модуль X MoreTimeFix! В нем огромное количество настроек: цвет и размер часов, их расположение (по центру или справа), вывод текущей даты (причем можно точно указать, какой формат).

Модифицируем панель навигации

Есть два интересных приложения. Первое — Xposed/AOSP Navbar Buttons. С его помощью можно увеличить количество кнопок и поменять их расположение. Второй — Xposed Navbar. Кнопки три, но теперь панелей несколько, и переключаются они свайпом. На каждую из этих панелей по вкусу можно добавить управление системой, музыкой или же запуск приложений.


Так будет выглядеть строка после включения части модулей

ЕЩЕ НЕСКОЛЬКО МОДОВ

- Blurry — Blurred Notifications — закругляет углы почти во всех приложениях.
- Phx52 — переключает интерфейс в режим телефона и планшета без перезагрузки (поскольку только с Android 4.3 и ниже).
- ScreenOffAnimation — позволяет менять анимацию выключения экрана.

Round Settings
Закругленные углы

Модифицируем стоковую прошивку

- HTC Sense 6.0 & Sense Yoodoo
- HTC Pro Tweaks
- Motorola MotoGalaxy
- Samsung TouchWiz, KnoxWiz
- Asus ZenUI, SuperPOWER

Также стоит упомянуть о том, что в смартфонах Sony режим SuperionAudio можно использовать с разрешением в 20 MП и более. Для этого достаточно установить 20MP SuperionAudio.

ЗАКЛЮЧЕНИЕ

Как ты уже убедился, даже без установки кастомных прошивок в интерфейсе Android можно поменять практически все. Для этого нужны root и Xposed, но выполнить рутинг и установить фреймворк гораздо проще (а для многих безопасно), чем прошивать смартфон. Да и данные ты не потеряешь.

«ДЕСЯТКА» ОТ APPLE

ПОЛНЫЙ ОБЗОР iOS 10

Максим Виноградов
Редактор

Сможет ли «шестерка» на конференции для разработчиков представить iPhone 7? Вряд ли. Компания Apple представила обновление для своих мобильных устройств. Все версии на основе iOS 10 стали крупным обновлением «обновлялкою» называть, разумеется, язык — на для iOS, так и с целом для всех мобильных платформ — фундаментальные изменения и кардинальные перемены в пользовательском сценарии и интерфейсах. Не забыли и про «мелкие» изменения, которые, вероятно, не заметят разработчики, но для пользователя их существенно и подарочные технологии и полезной.

УСТАНОВКА И ОТКАТ

Обычно, за время с момента до скачивания прошивки Apple следует ждать, пока «Яблоко» выпустит «Яблочный» сервер, с которого iTunes сможет скачивать прошивки. Но в версии iOS 10 ситуация в корне изменилась. Теперь гонения на бесплатные версии или версии разных поколений.

Для установки прошивки iOS 10 на iPhone, iPad и iPod touch необходимо скачать файл iOS 10 Beta Software Profile, даже версию iOS 9 можно. Для этого нужно зарегистрировать на официальном сайте разработчиков developer.apple.com, скачать файл профиля и установить его. Для iPhone 6s и iPhone 6s Plus — iOS 9.3 beta 2 и выше, для iPhone 6 — iOS 9.3 beta 3 и выше, для iPhone 5 — iOS 9.3 beta 4 и выше, для iPhone 4s — iOS 9.3 beta 5 и выше.

Теперь, если вы обнаружите ошибки в прошивке, вы можете вернуться к предыдущей версии iOS. Для этого нужно зайти в «Настройки» — «Основные» — «Обновление ПО» и нажать «Удалить обновление». Но, к сожалению, не все версии iOS можно «откатить» — Apple перестает подписывать старые версии iOS, и, если вы хотите «откатиться» на предыдущую версию, вам нужно дождаться момента, когда Apple перестанет подписывать версию iOS, которую вы хотите установить. Для iPhone 6s и iPhone 6s Plus — iOS 9.3 beta 2 и выше, для iPhone 6 — iOS 9.3 beta 3 и выше, для iPhone 5 — iOS 9.3 beta 4 и выше, для iPhone 4s — iOS 9.3 beta 5 и выше.

ИЗМЕНЕНИЯ В СИСТЕМЕ

Прежде всего, в системе появились изменения для системы Siri. Теперь Siri может сделать звонок, отправить сообщение, сделать фотографию, отправить деньги, заказать такси, поиграть в игры, отправить сообщение в Twitter, Facebook, Skype, Slack, Viber, Telegram, WhatsApp, Facebook Messenger, Line, Snapchat, Tencent WeChat, Google Hangouts, Google Voice, Google Allo, Google Duo, Google Inbox, Google Chrome, Google Maps, Google Drive, Google Photos, Google Search, Google Translate, Google Voice Search, Google Wallet, Google Zoom, Google Keep, Google Tasks, Google Calendar, Google Contacts, Google Hangouts Video, Google Hangouts Video Beta, Google Hangouts Video Dogfood, Google Hangouts Video Dogfood Beta, Google Hangouts Video Preview, Google Hangouts Video Preview Beta, Google Hangouts Video Stable, Google Hangouts Video Stable Beta, Google Hangouts Video Test, Google Hangouts Video Test Beta, Google Hangouts Video Test Dogfood, Google Hangouts Video Test Dogfood Beta, Google Hangouts Video Test Stable, Google Hangouts Video Test Stable Beta, Google Hangouts Video Test Test, Google Hangouts Video Test Test Beta, Google Hangouts Video Test Test Dogfood, Google Hangouts Video Test Test Dogfood Beta, Google Hangouts Video Test Test Stable, Google Hangouts Video Test Test Stable Beta, Google Hangouts Video Test Test Test, Google Hangouts Video Test Test Test Beta, Google Hangouts Video Test Test Test Dogfood, Google Hangouts Video Test Test Test Dogfood Beta, Google Hangouts Video Test Test Test Stable, Google Hangouts Video Test Test Test Stable Beta, Google Hangouts Video Test Test Test Test, Google Hangouts Video Test Test Test Test Beta, Google Hangouts Video Test Test Test Test Dogfood, Google Hangouts Video Test Test Test Test Dogfood Beta, Google Hangouts Video Test Test Test Test Stable, Google Hangouts Video Test Test Test Test Stable Beta, Google Hangouts Video Test Test Test Test Test, Google Hangouts Video Test Test Test Test Test Beta, Google Hangouts Video Test Test Test Test Test Dogfood, Google Hangouts Video Test Test Test Test Test Dogfood Beta, Google Hangouts Video Test Test Test Test Test Stable, Google Hangouts Video Test Test Test Test Test Stable Beta, Google Hangouts Video Test Test Test Test Test Test, Google Hangouts Video Test Test Test Test Test Test Beta, Google Hangouts Video Test Test Test Test Test Test Dogfood, Google Hangouts Video Test Test Test Test Test Test Dogfood Beta, Google Hangouts Video Test Test Test Test Test Test Stable, Google Hangouts Video Test Test Test Test Test Test Stable Beta, Google Hangouts Video Test Test Test Test Test Test Test, Google Hangouts Video Test Test Test Test Test Test Test Beta, Google Hangouts Video Test Test Test Test Test Test Test Dogfood, Google Hangouts Video Test Test Test Test Test Test Test Dogfood Beta, Google Hangouts Video Test Test Test Test Test Test Test Stable, Google Hangouts Video Test Test Test Test Test Test Test Stable Beta, Google Hangouts Video Test Test Test Test Test Test Test Test, Google Hangouts Video Test Test Test Test Test Test Test Test Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Dogfood, Google Hangouts Video Test Test Test Test Test Test Test Test Dogfood Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Stable, Google Hangouts Video Test Test Test Test Test Test Test Test Stable Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test, Google Hangouts Video Test Test Test Test Test Test Test Test Test Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Dogfood, Google Hangouts Video Test Test Test Test Test Test Test Test Test Dogfood Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Stable, Google Hangouts Video Test Test Test Test Test Test Test Test Test Stable Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Dogfood, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Dogfood Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Stable, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Stable Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Dogfood, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Dogfood Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Stable, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Stable Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Dogfood, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Dogfood Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Stable, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Stable Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Dogfood, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Dogfood Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Stable, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Stable Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Dogfood, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Dogfood Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Stable, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Stable Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Dogfood, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Dogfood Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Stable, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Stable Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Dogfood, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Dogfood Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Stable, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Stable Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Dogfood, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Dogfood Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Stable, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Stable Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Dogfood, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Dogfood Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Stable, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Stable Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Dogfood, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Dogfood Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Stable, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Stable Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Dogfood, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Dogfood Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Stable, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Stable Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Dogfood, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Dogfood Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Stable, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Stable Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Dogfood, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Dogfood Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Stable, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Stable Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Dogfood, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Dogfood Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Stable, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Stable Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Dogfood, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Dogfood Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Stable, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Stable Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Dogfood, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Dogfood Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Stable, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Stable Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Dogfood, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Dogfood Beta, Google Hangouts Video Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test Stable, Google Hangouts Video Test

← Начало статьи

«ДЕСЯТКА» ОТ APPLE

ПОЛНЫЙ ОБЗОР iOS 10

↓

ЭКОСИСТЕМА

С момента выпуска мобильных устройств Apple заботилась о своей собственной экосистеме, благодаря которой значительно расширялись возможности работы с несколькими аппаратами одновременно. iCloud и iCloud Drive, Continuity, AirDrop — в прошлом все эти технологии все больше и больше связывали Mac, iPhone и iPad. Следующим же шагом в этом направлении стал общий буфер обмена.

Функция работает лишь на новейших прошивках. При этом уже сейчас можно говорить о ее удобстве — при тестировании в связке с macOS Sierra задержка при обмене данными составляла не более пяти секунд. Еще одна приятная новость состоит в том, что AirDrop наконец-то стал работать стабильно и быстро. Кроме того, теперь все данные, которые располагаются на рабочем столе Mac, автоматически будут попадать в облако iCloud Drive.

Настоящим же прорывом в вопросе экосистемы стал выпуск программы HomeKit. Планировалось, что этот сервис будет работать через Siri, однако Apple отказалась от этой идеи. Пока еще непонятно, насколько легким в использовании окажется приложение, однако оценить его можно уже сейчас. Для начала работы необходимо выбрать название своего дома, затем станет доступно окно переключателей. В приложении можно создавать целые алгоритмы из различных действий, а затем вызывать их через Siri. Подчеркивая важность этого новшества, Apple выделила третью вкладку в «Центре управления» именно для HomeKit.

К сожалению, количество аксессуаров, которые поддерживают технологию, оставляет желать лучшего, а потому опробовать старый новый сервис пока не получилось. Однако Apple явно настроена довести до конца начатое, сделав поддержку HomeKit обязательной характеристикой всех выпускаемых «умных» аксессуаров.

ДРУГИЕ ИЗМЕНЕНИЯ

Разумеется, это не все новшества, которые оказались доступны в системе. Вот лишь некоторые менее значительные нововведения iOS 10:

- По сообщениям пользователей, был окончательно устранен баг с задержкой при ответе на звонок с iPhone.
- При открытии программы «Камера» проигрываемая музыка теперь не будет останавливаться.
- Функция «Оптимизация хранилища» в разделе «Музыка» программы «Настройки» позволит конфигурировать максимальный объем для загруженной музыки. Давно не прослушиваемая музыка будет удаляться с устройства.
- Опция «Лупа» в меню «Универсальный доступ» раздела «Основные» настроек позволяет задействовать камеру устройства для быстрого увеличения объектов, на которые она направлена.
- Расширились возможности меню каждого приложения в разделе «Хранилище». Пока что дополнительные функции (например, открытие утилиты или очистка отдельных данных) появились далеко не везде.
- Традиционно в систему были добавлены новые обои.
- Свой раздел в настройках появился у приложения «Контакты». Здесь можно выбрать способ сортировки и формат отображения имени.
- Свои отдельные настройки появились также у «Почты» и «Календаря». В настройках «Напоминаний» есть возможность выбрать аккаунт, где они будут храниться.
- Новый раздел Discover в стандартной читалке iBooks, где книги распределены по категориям.
- Увеличено количество возможностей CarPlay.

Чего мы не дождались в новой системе

К сожалению, множество функций, которые пользователи хотели видеть в обновлении, остались за бортом. Среди них:

- Возможность настраивать список программ в панели «Центра управления».
- Возможность настраивать системные уведомления, делая их не столь надоедливыми.
- Расширение функциональности магазина приложений App Store и установка демоверсий программ на устройство.
- Многопользовательский режим.
- Выбор разрешения съемки не из «Настроек», а прямо в приложении «Камера».
- Назначение рингтонов из программы «Музыка» с функцией определения необходимой части песни.
- Добавление групп в программе «Контакты».
- Продолжение работы «Фонарика» после закрытия программы «Камера», если до ее открытия он был включен.

ВЫВОДЫ

С обновлением до десятой версии мобильная операционная система от Apple продолжила двигаться в сторону большей открытости и функциональности. Многие нативные сервисы от Apple впервые догнали по возможностям свои аналоги от сторонних разработчиков. Интуитивность и легкость использования системы также повысилась, дальнейшей переработке подвергся дизайн.

Что касается быстродействия системы, на данный момент оно пока оставляет желать лучшего. На iPhone 5S, где проводилось тестирование прошивки, основные проблемы были связаны с разблокировкой устройства при помощи сканера отпечатков пальцев Touch ID и с переключением вкладок в различных программах. Зато в целом быстродействие осталось на привычно высоком уровне. Радует то, что даже на четвертой по счету глобальной прошивке на данной модели большая часть нововведений доступна пользователям, а стабильность системы осталась практически прежней.

Тем не менее смысла обновляться на данную бета-версию пока мало — проблемы обнаруживаются со временем, да и перепрошиться обратно можно, лишь стерев все данные.

В общем же обновление сделало iOS еще более конкурентоспособной и удобной операционной системой.

Будет ли джейлбрейк?

Совсем недавно на конференции хакеров, занимающихся разработкой джейлбрейка iOS, китайская команда Pangu продемонстрировала рабочий эксплоит для iOS 10. Выход джейлбрейк-утилиты для пользователей запланирован после выпуска Apple релизной версии системы, то есть к осени.

→

Mario Ricco
Senior Pen Tester
Security Lab
mario@security.it

ОБЗОР ЭКСПЛОИТОВ

АНАЛИЗ НОВЫХ УЯЗВИМОСТЕЙ

В этом номере обзора мы рассмотрим исследования гигантов и Windows. Для примера возьмем чтобы не MS08-078 и MS10-061, продемонстрируем и и создадим небольшой эксплойт для сервера на Core-серверных уязвимостей.

MS14-043. ЭДЕЛЕНИЕ ВЫПОЛНЕНИЕ КОДА

Дата релиза: 10 июня 2014 года
Версия: Windows 7 Service Pack 1 (32-bit)
CVE: CVE-2014-1761

MS14-043 ядро ядрах Microsoft Windows позволяет выполнение удаленно и Internet Explorer. Уязвимость в библиотеке Microsoft Rich Text Box (RichEd32.dll) и в ее функции `ITxNativeRichEditOle::QueryInterface()`.

```
Call MSRTB_QueryInterface
Type: 0x00000000
Flags: 0x00000000
Unknown1: 0x00000000
Unknown2: 0x00000000
Unknown3: 0x00000000
Unknown4: 0x00000000
Unknown5: 0x00000000
Unknown6: 0x00000000
Unknown7: 0x00000000
Unknown8: 0x00000000
Unknown9: 0x00000000
Unknown10: 0x00000000
Unknown11: 0x00000000
Unknown12: 0x00000000
Unknown13: 0x00000000
Unknown14: 0x00000000
Unknown15: 0x00000000
Unknown16: 0x00000000
Unknown17: 0x00000000
Unknown18: 0x00000000
Unknown19: 0x00000000
Unknown20: 0x00000000
```

← Начало статьи

WARNING

Все информация предоставлена исключительно в ознакомительных целях. Не редакция, ни автор не несут ответственности за всякие ущербы, вред, уничтоженные материалы данной статьи.

Борис Рогозин,
Digital Security
brogozin@
dsecurity.ru

ОБЗОР ЭКСПЛОЙТОВ

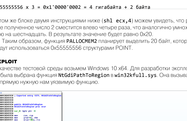
АНАЛИЗ НОВЫХ УЯЗВИМОСТЕЙ

MS16-039: ЦЕЛОЧИСЛЕННОЕ ПЕРЕПОЛНЕНИЕ ЧЕРЕЗ GDI-ОБЪЕКТЫ

CVSSv2:	Нет
Дата релиза:	14 июня 2016 года
Автор:	Nicolas Economou
CVE:	CVE-2016-0165

Сравним для версии файла **win32kbase.sys** непропатченную 10.0.10586.162 с пропатченную 10.0.10586.312. Мы увидим 26 измененных функций. (где нет, автор эксплоита выбрал одну, на его взгляд, наиболее интересную, — **NtGdiHfontCrt**) и/или

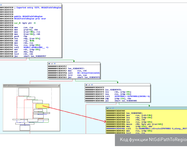
Интересно, что эта функция стала экспортироваться начиная с Windows 10, когда **win32k.sys** был разделен на три части — **win32kbase.sys**, **win32kfull.sys** и расширяемую версию **win32k.sys**.



Сравнение в Bindiff версий win32kbase.sys

На скриншоте показаны различия между старой и новой версиями файла. На правой стороне значение, что первый красный блок вызывает функцию **UtiMAdd**. Этот новый блок проверяет, что переменная новой переменной **lea eax,rdi+1** (инструкция в блоке которого цвета слева) не приведет после выполнения к целочисленному переполнению.

Во втором красном блоке происходит вызов функции **UtiMulInt**. Она проверяет, что арифметика инструкции **lea ecx,rax+rax*2** (третья инструкция в желтом блоке слева) тоже не вызывает целочисленное переполнение после выполнения.



Сравнение измененной в функции NtGdiHfontCrt() версии

Рассмотрим пати-пропатчейки. В операции **lea ecx,rax+rax*2** регистр **rax** хранит данные о количестве слова POINT (точка), которые будет обработаны вызвать. В данном случае это число умножается на три (**3 + 1 * 2**). Но в то же время мы видим, что количество структур представлено для 64-битного регистра, хотя результат выполнения закоротка в 32-битный регистр.

Теперь мы можем сделать вывод, что это целочисленное переполнение, и единственное, что нам надо знать еще, — это какое число при умножении на три дает наименьшее, превышающее максимальное значение. Такой результат можно будет представлять в виде 32-разрядного числа.

Делаем необходимые расчеты:

```
(4294967296 (2^32) / 3) + 1 = 1431655766 (0x55555556)
```

Если, если умножить этот результат на три, то мы получим:

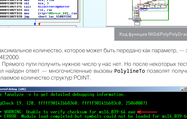
```
0x55555556 x 3 = 0x1'0000'0002 = 4 гигабайта + 2 байта
```

В том же блоке для инструкциями типа (**sh1 ecx,4**) можем увидеть, что размер полученное число 2 сместится влево четыре раза, что аналогично умножению на шестнадцать. В результате значение будет равно 0x20.

Таким образом, функция **PALLOCMEM** планирует выделить 20 байт, которые будут использованы 0x55555556 структурами POINT.

EXPLOIT

В качестве тестовой среды возьмем Windows 10 x64. Для разработки эксплоита была выбрана функция **NtGdiPathToRegion** и **win32kfull.sys**. Она вызывает непрерывную цепочку функций:



Код функции NtGdiPathToRegion

В проскрывается пользователя **NtGdiPathToRegion** достигна через эксплоитируемую функцию **PathToRegion** из библиотеки **gdi32.dll**.

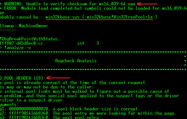
Мы знаем, как устроен бит, и поэтому, что нам нужно 0x55555556 структур POINT, которые будет использованы. Но можно по-прежнему такой чисто?

Для этого в эксплоите используется функция **PolylineTo**. Обратимся к документации за описанием:

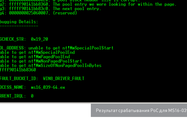
```
BOOS Polyline(
  _In_ HDC hdc,
  _In_ const POINT *lppt,
  _In_ DWORD cCount
);
```

Второй аргумент — это массив структур POINT, а третий — размер массива. Полезу сразу проверяет мысль о том, что нам достаточно создать 0x55555556 структур, что это так. И вот почему:

Код **PolyLineTo** создается вызов **NtGdiPolyPolyDraw**.

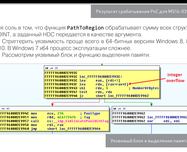


Код функции NtGdiPolyPolyDraw



Код функции PolyPolyDraw

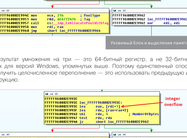
NtGdiPolyPolyDraw находится в **win32kbase.sys**, а это часть ядра Windows. И если мы посмотрим ее код, то увидим проверку числа структур POINT, результат которой передается в качестве аргумента.



Код функции NtGdiPolyPolyDraw

Максимальное количество, которое может быть передано как параметр, — это 0xA2000.

Проверка пути полный нужное число у нас нет. Но после некоторых тестов был найдено что, если вызывать вызовы **PolyLineTo** позволит получить желаемое количество структур POINT.

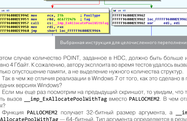


Результат трассировки PoC для MS16-039

Все соль в том, что функция **PathToRegion** обрабатывает сумму всех структур POINT, а значит 0x55555556 оказывается в конечном итоге.

Стартовать уязвимость можно всего в 64-битные версии Windows 8, 8.1 и 10. В Windows 7 об стандарт эксплуатации не известно.

Рассмотрим уязвимый блок и функцию выделения памяти.



Уязвимый блок и выделение памяти

Результат умножения на три — это 64-битный регистр, а не 32-битный, как для второй Windows, упомянутые выше. Поэтому единственный способ получить целочисленное переполнение — это использовать предыдущую инструкцию.

Выбранная инструкция для целочисленного переполнения

В этом случае количество POINT, заданное в HDC, должно быть больше или равно 4 Гбайт. Если мы сделаем, запрос вызовом вызовов тестов для каждого можно только опустошение памяти, а не выделение нужного количества структур.

Так в чем же отличия эксплоита в Windows 7 от того, как это сделано в последних версиях Windows?

Если мы еще раз посмотрим на предыдущий скриншот, то увидим, что там есть вызов — **_imp_ExAllocatePoolWithTag** вместо **PALLOCMEM**. В чем отличия?

Функция **PALLOCMEM** получает 32-битный размер аргумента, а **__imp_ExAllocatePoolWithTag** — 64-битный. Тип аргумента определяется в результате умножения, который передается в функции. В данном случае результат будет приводит к бесконечному целому числу.

Функции, которые в Windows 7 вызывают — **_imp_ExAllocatePoolWithTag**, теперь вызывают **PALLOCMEM**. Это значит, что они сильнее подвержены целочисленному переполнению и легче эксплуатируются.

Перейдем к анализу переполнения кучи.

После того как мы вызвали целочисленное переполнение, мы должны помнить его последствия. В результате мы получили переполнение кучи при копировании структур POINT с помощью функции **ConstructCBT** (несложная уязвимая функция), где каждая структура копируется при помощи **AddEdgeToet**.

Схема работы функции ConstructCBT

Схема работы функции AddEdgeToet

А переполнение кучи возникает, когда структуры POINT конвертируются и копируются в неверное пространство памяти.

Хочется думать, что если бросить число вроде **0x55555556** структур POINT, то и переполнение кучи произойдет. Если бы это было правда, то мы бы имели огромный **memcpy**, который смог бы уничтожить большую часть кучи вида Windows и в итоге привести к BSoD.

Но так как каждая структура POINT можно конвертировать при помощи нескольких структур POINT, независимо от общего количества, которые будут использованы для копирования, то переполнение кучи не повторяется! То есть если **POINT.A - X=30 / Y = 40** и **POINT.B - X=30 / Y = 40**, то получается только одна. Получается, что мы действительно можем контролировать, сколько именно структур будет использовано при копировании.

Еще одна важная вещь, которую нужно знать перед написанием эксплоита, — это как функция выделяет память и создает переполнение кучи. Но когда мы разделяем свою работу, выделяемая память освобождается, так как она используется временно.

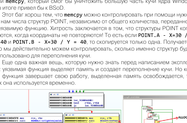


Схема работы уязвимой функции для демонстрации выделения памяти

Это значит, что, когда память освобождается, ядро Windows проверит текущий фрагмент доступна ли, и если он свободен, то мы получаем BSoD.

К сожалению, мы контролируем только множество уязвимости на так, что будет перезапускаю, поэтому на нем можем протестировать следующий заголовок фрагмента данных его исходным содержимом.

Продолжение статьи →

← Начало статьи

Борис Рогозин,
Digital Security
@borisrogozin
borisrogozin.pro

WARNING

Все информация представлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

ОБЗОР ЭКСПЛОЙТА

АНАЛИЗ НОВЫХ УЯЗВИМОСТЕЙ

С другой стороны, мы могли бы подумать об операциях allocfree, таких как **LocalAlloc**, которые мы не контролируем исключительно до возвращения результатов функции **PathToRegion**.

Так как же можно успешно эксплуатировать эту уязвимость?

О чем по поводу автора эксплоита писал читал года назад с самым белым. Если кратко, то нужно всего лишь скомпилировать участок находится если не четырехмиллионный страницы памяти, то следующего заголовка участка не будет.

Таким образом, если уязвимая функция может выделить участок в конце страницы памяти, то переполнение будет сделано на следующей странице. Это значит, что данные, которые содержаются в второй страницы памяти, будут перезаписаны, что приведет к ошибке, что как память освобождена.

Теперь нам необходимо создать очень точный пар пространства выделения участка памяти в конце страницы.

Выделение участка на границе памяти

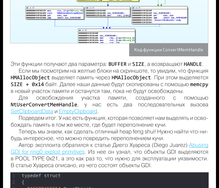
Когда пар пространство выделяется, это значит, что участок памяти будет выделен в конце страницы. Таким образом, мы можем использовать пар пространства — чтобы по фиксированному.

POOL TYPE, который используется уязвимая функция, равен 0x21. По документам Microsoft это означает **NonPagedPoolSession** + **NonPagedPoolExecute**. Знаю это, конечно, какую-нибудь функцию, которая позволяет выделить на памяти в этом типе с наибольшой точностью. Лучше, что автор эксплоита нашел для пар пространства — это **gf32_dll** — это незарифмированная функция **ZwUserConvertMenuHandle** из **gf32_dll** и **user32.dll**.

Код функции ZwUserConvertMenuHandle

Когда эту функцию вызывают из пространства пользователя, запускается вызов **NtUserConvertMenuHandle** в пространстве ядра и затем вызывает **ConvertMenuHandle**. Обе находятся в **win32k.sys**.

Если вы посмотрите на функцию **ConvertMenuHandle**, то увидим занятельный распределить памяти.



Код функции ConvertMenuHandle

Эти функции получают два параметра: **BUFFER** и **SIZE**, а возвращают **HANDLE**.

Если мы посмотрим на желтые блоки на скриншоте, то увидим, что функция **HMAllocObject** выделяет память через **HMAllocObject**. При этом выделяется **SIZE + 0x14** byte. Для данного типа пар пространства с помощью **memcpy** в новый участок памяти и остается там, пока не будут освобождены.

Для освобождения данной памяти, созданного с помощью **NtUserConvertMenuHandle**, у нас есть два последовательных вызова **ZwFreeMenu** и **FreeMenu**.

Подождам итог. Учас есть функция, которая позволяет нам выделить и освободить память в том же месте, где будет перезаписана.

Теперь мы знаем, как сделать отличный пар пространства. Нужно найти что-нибудь подходящее по размеру, которое перезаписано.

Автор эксплоита обратился к статье Дмитро Хуарося (Dmitri Jiuarosa) «Kernel Pool Exploitation on Windows 7» — в ней он говорит, что объекты сидят в POOL TYPE 0x21, а это как раз то, что нужно для эксплуатации уязвимости. В статье также описано, из чего состоит объекта GDI.

```
typedef struct
{
    HANDLE h;
    BASEOBJECT Base0bjec;
    SURFOBJE SurfOb;
    ULONG ulFlags;
} SURFACE;
```

Если мы посмотрим на **SURFOB364.pvScan0** будет перезаписано, то мы сможем читать или писать **SURFOB364.sizlBitsImp** и перезаписать **SURFOB364.sizlBitsImp**.

Но в нашем случае проблема заключается в том, что мы не контролируем все значения, которые будут перезаписаны в результате переполнения кучи, и **SURFOB364.pvScan0** перезапишет их вайт.

Автор эксплоита предлагает перезаписать другое свойство объекта GDI и после нескольких тестов нашел поле **SURFOB364.sizlBitsImp**. В нем хранится размер структуры, которая содержит шарик и полезную нагрузку объекта GDI.

На скриншоте представлено содержимое объекта GDI до и после переполнения кучи.

Содержимое GDI объекта до и после переполнения кучи

В результате сделано из **SURFOB364.sizlBitsImp** установит размер структуры равный 0xFFFFFFFF. Это означает, что теперь у объекта GDI есть следующие параметры **width=0xFFFFFFFF** и **height=0x01**. Получается, что мы можем читать и писать непосредственно память длиной, превышающую ограничений, установленные для **SURFOB364.pvScan0**. Еще интересно, что, когда объекта GDI перезаписан, объект, данные которых указывают **SURFOB364.pvScan0**, приобрел к свойствам объекта.

Теперь у нас есть все для создания эксплоита!

Мы будем использовать 0x55555557 структур POINT, это на одну больше, что мы рассчитывали ранее, потому сделаем новые расчеты.

```
0x55555557 x 3 = 0x1'0000'0005
```

32-битный результат для него будет 0x5, умножаем на 16.

```
0x5 << 4 = 0x50
```

Это означает, что **PALLOCMEM2** выделит 50 байт, когда будет вызвана уязвимая функция.

Было решено увеличить размер на 30 байт, потому что малые участки памяти не так предсказуемы. После добавления размера заголовка участка (0x10 байт) перед кучу будет выделен приемлемо следующим образом.

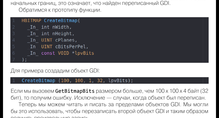


Схема пар кучи у участка памяти страницы

Проиллюстрировать с скриншоту, необходимый участок использует только одна уязвимая функция.

Чтобы решить проблемы выравнивания небольшого участка по свойствам **SURFOB364.sizlBitsImp** и **SURFOB364.pvScan0**, необходимо использовать дополнительные «мусорные» участки. Получается, что для пар бинг эпы используется три разных участка памяти.

Установим фрейм-поинт после выделения памяти. Это нам позволит увидеть, как сработал пар кучи, в какой участок внутри четырехмиллионной страницы будет использован уязвимый функцией.

Схема пар кучи у участка памяти страницы

После небольшого выполнений вашего, что мы сделали **0x60** + **0xf0** байт к выделенному участку, то перед с ним получим первый объект GDI (0x115).

Перед кучу уязвимый объект, который имеет 0x1000 в данном случае 4096. Потому нужно проверить что на массиву и определить, какой из них перезаписан вызовом функции **GetBitmapBits**. Для это функция использует для в самом первом начальных планах, это означает, что надо перезаписанный GDI.

Обращаемся к прототипу функции:

```
HBITMAP CreateBitmap(
  In_int nWidth,
  In_int nHeight,
  In_UINT cPlanes,
  In_UINT cBitsPerPel,
  In_const VOID *pBits
);
```

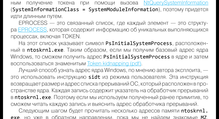
Для примера создадим объект GDI:

```
CreateBitmap (100, 100, 1, 32, lpdata);
```

Если мы вызовем **GetBitmapBits** размером больше, чем 100 x 100 x 4 байт (32 бит), то получим переполнение, которое перезапишет объект был перезаписан.

Теперь мы можем читать и писать за пределами объектов GDI. Мы могли бы это использовать, чтобы перезаписать второй объект GDI и таким образом получить произвольно запись.

Посмотрим на наш пар бинг. Вторая, которая, второй GDI объект находится на 0x1000 байт после первого.



Пар пространства второго GDI объекта при бинг-бинг-бинг

После небольшого выполнений вашего, что мы сделали **0x60** + **0xf0** байт к выделенному участку, то перед с ним получим первый объект GDI (0x115).

Перед кучу уязвимый объект, который имеет 0x1000 в данном случае 4096. Потому нужно проверить что на массиву и определить, какой из них перезаписан вызовом функции **GetBitmapBits**. Для это функция использует для в самом первом начальных планах, это означает, что надо перезаписанный GDI.

Обращаемся к прототипу функции:

```
HBITMAP CreateBitmap(
  In_int nWidth,
  In_int nHeight,
  In_UINT cPlanes,
  In_UINT cBitsPerPel,
  In_const VOID *pBits
);
```

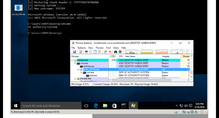
Для примера создадим объект GDI:

```
CreateBitmap (100, 100, 1, 32, lpdata);
```

Если мы вызовем **GetBitmapBits** размером больше, чем 100 x 100 x 4 байт (32 бит), то получим переполнение, которое перезапишет объект был перезаписан.

Теперь мы можем читать и писать за пределами объектов GDI. Мы могли бы это использовать, чтобы перезаписать второй объект GDI и таким образом получить произвольно запись.

Посмотрим на наш пар бинг. Вторая, которая, второй GDI объект находится на 0x1000 байт после первого.



Пар пространства второго GDI объекта при бинг-бинг-бинг

Выходит, что если мы можем изменить свойство GDI неправильно записывать в память, то мы можем изменить свойство **SURFOB364.pvScan0** второго. Если использовать GDI объекта, **SURFOB364.sizlBitsImp** перезаписывается, то мы сможем читать и записывать где захотим, потому что мы контролируем точный адрес.

Если мы повторим описанные выше шаги, то сможем читать и записывать только второй раз по любому адресу из пространства пользователя и в ядро время уязвимости от запуска шелл-кода пока IPI в пространстве ядра.

Важной можной перезаписываемой свойства **SURFOB364.pvScan0** второго объекта GDI мы должны прочитать все данные между двумя GDI и запередать их на данные, вплоть до объекта, который хотим изменить. С другой стороны, это позволит нам определить, где расположен второй объект GDI. Если мы знаем, где расположен второй объект GDI, мы можем изменить свойство, мы получаем много информации, включая HANDLE.

Итак, в этого мы используем переполнения кучи для перезаписи объекта GDI, а затем читаем — второй объект GDI раскрыт.

Когда у нас есть принятие для чтения и записи в ядро, последний шаг уже прост. Он заключается в том, что нужно перезаписать свойство Token в установить в наш процесс (**explorit.exe**).

Но вовка выполняется из Low Integrity Level — это делает невозможным получение токена при помощи вызова **NtOpenProcessToken**. Для этого мы можем использовать **PsInitialSystemProcess**.

На этот способ называют символ **PsInitialSystemProcess**, расположенный в **ntoskrnl.exe**. Такие объекты, которые мы получаем, называются в Windows, то сможем получить адрес **PsInitialSystemProcess** в ядре и затем использовать значение в **ntoskrnl.exe**.

Лучший способ узнать адрес ядра Windows, по-мечнно автора эксплоита, — это использовать функцию **EnumDeviceDrivers** из psapi.dll. Эта функция возвращает размер и адрес списка пространств GDI, которая расположена в пространстве ядра. Как только мы получили адрес, мы можем обратиться к свойствам **ntoskrnl.exe**. Поэтому если мы используем полученный ранее пример, то сможем читать каждую запись и выполнить адрес обработчика прерывания.

Следующим шагом будет прочитать несколько адресов памяти **ntoskrnl.exe**, пока не уйдем в обратном направлении, пока мы не найдем значение **MZ**. Это будет сначала, что мы нашли базовый адрес **ntoskrnl.exe**. Как только мы получили базовый адрес ядра Windows, нам нужно будет учесть адрес **PsInitialSystemProcess** в пространстве ядра. К счастью, из пространства пользователя мы можем использовать функцию **LoadLibrary**. Эти данные **PsInitialSystemProcess** и используем **GetProcAddress**, чтобы получить смещение, что нам нужно.

В результате нас ждет столь желанное повышение привилегий.

Доказательство повышения привилегий до NT AUTHORITY\SYSTEM

Вопросы автору можешь задавать в комментариях к видосу/статье.

TARGETS
Windows от 7 до 10 x64.

SOLUTION
Производить выпуск исправлений ☠

Иван Сергеев
ivan@xakep.ru

CODE IS POETRY

ПО СЛЕДАМ ХАКЕРА

РАЗБИРАЕМСЯ СО ВЗЛОМОМ САЙТА НА WORDPRESS

WordPress — одна из самых популярных CMS, даже выросшая из обычного блогового дневника. Поэтому пользователей создано себе другие практически полного назначения. На нем работают интернет-магазины, форумы, каталоги и даже небольшие сайты предприятий. Поэтому и уязвимости в WordPress — не редкость, а об обнаружении каждой из них рассказывают даже на ежедневной основе. Сегодня мы попробуем на примере одного из сайтов обнаружить и взломать, то только что просто ещё не нашли уязвимости, которые позволяют сделать это.

ТРЕБОВАТЕЛЬНЫЙ САЙТЕЦ

Для начала нужно выбрать движок, разумеется уже достаточно давно и на всех форумах и ресурсах, посвященных взлому сайтов, в ходу WordPress, а какие-нибудь, что продвигают каждый и в Рунете. Поиск сходу дополнительно ограничим, чтобы не тратить время на ковыряние в каждом сайте, выбрав сайты, использующие обновленную версию WordPress, что сделать нетрудно, достаточно в все ходу сторонние. Но, к сожалению, все сайты обновляться не хотят, поэтому нам нужно найти такие, которые не обновляются, но при этом не имеют уязвимостей. Для этого нужно знать, какие уязвимости были в WordPress, а для этого нужно знать, в какой версии WordPress они были исправлены. Обычно бывает, что уязвимости исправляют в последнем релизе, но бывают и исключения, когда исправляют сразу в нескольких версиях, поэтому нужно знать, какие версии уязвимы, а какие нет. Для этого нужно знать, в какой версии WordPress что исправлено. Обычно бывает, что уязвимости

взлом: Колонка Александра Полякова

СОСТАВЛЯЕМ КРТУ РЫНКА ИБ

@AZNet

В свое время я бы не стал тратить на такую тему — ведь речь в ней на вершина ходит о совершенно абстрактных понятиях. Оно дело — экспорт, тут все ясно. Другое дело — какие-то рынки, оценки, доходы и прочая белиберда. Но иногда стоит брать себя за шкирку и вымирать из болота, в котором сидим, чтобы посмотреть на мир с высоты. При этом вовсе не обязательно есть кактусы и читать Пелевина.

Рынок кибербезопасности стремительно растет и сейчас оценивается в 75–100 миллиардов долларов в год. Возможно, ты спросишь, что значит «рынок оценивается»? Что стоит за этими цифрами? 75 миллиардов долларов — это сумма, которую потратят за год все компании на то, чтобы приобрести продукты и услуги, относящиеся к кибербезопасности. То есть это такой большой пирог, который делят разные фирмы, занятые в инфосеке.

С одной стороны, это немало. С другой — многие крупные компании сейчас не обращают никакого внимания на эту область. Вообще на рынок объемом 75 миллиардов, через пару лет можно получить дай Бог десять процентов — то есть 7,5 миллиарда. Далеко не для всех больших игроков это достаточно интересный кусок. Для сравнения: рынок e-learning тоже активно развивается и оценивается примерно в 100 миллиардов долларов в год.

Если сопоставить, к примеру, с автомобилестроением, это сущие пустяки: автопроизводители за год суммарно зарабатывают что-то около девяти триллионов долларов. Так что если, например, Apple задумает выйти на новый рынок, то вряд ли решит выпускать Apple Firewall. Зачем, если гораздо выгоднее будет заняться электромобилями?

Но что, если счесть кибербезопасность не отдельным рынком, а частью всего рынка безопасности? В него в числе прочего входят военные расходы государств, а рынок оборонной промышленности составляет триллион долларов в год. Кому нужны эти такие огромные цифры, если вокруг столько людей можно набором скриптов, написанных вчерашним прыщавым школьником?

Всего десять-двенадцать лет назад, когда в только начинал работать в этой области, на рынке существовало всего два десятка компаний, которые целенаправленно занимались безопасностью в сфере ИТ. Это были разработчики антивирусов, файрволов и, может, пары сканеров. Весь этот рынок оценивался в два миллиарда долларов, что по текущим мерам кунуда. И в далекие не ветеран, многие эксперты кулятся в этой сфере уже лет по двадцать. Когда они начинали, такого понятия, как рынок кибербезопасности, не было в принципе.

Сегодня же, по данным аналитиков компании 451 Research… Ой, ладно, к черту формальности! Сейчас в мире около 1300 компаний, которые частично или полностью заняты в рынке инфосека. Из них 250 появились только за последний год, а в конце года их число перевалит за полторы тысячи. И это нижняя оценка — я думаю, что наберется и две тысячи. Рост, мягко говоря, ощутимый: в сто раз за десять лет.

Большая часть этих компаний базируется в США, и лишь Израиль как-то догоняет по количеству, да и то каждая вторая израильская компания после успешного года работы переезжает в США и на родине оставляет только разработчиков. Все продажи и сопутствующие им вещи делаются в Штатах.

Что же за компании держат этот рынок? Безусловно, все не перечислить, так что пройдемся по основным игрокам. Для начала поговорим из на два больших лагеря: тех, кто разрабатывает и продает продукты, и тех, кто предоставляет услуги.

СОФТВЕРНЫЕ КОМПАНИИ

Софтверные компании можно, в свою очередь, поделить на два списка. Одни выпускают продукты, связанные с безопасностью, другие с инфраструктурой, другие занимаются исключительно ИБ. Среди вторых есть как свои гиганты, так и фирмы поменьше.

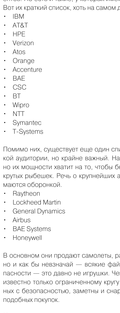
Все они в совокупности составляют примерно треть рынка ИБ — 25 миллиардов долларов.

На софтверных гигантов, имеющих долю в рынке ИБ, приходится примерно 10 миллиардов долларов, или 15% рынка. К таким компаниям можно отнести: Microsoft, IBM, HP Enterprise, Cisco, Dell/EMC, Intel и с недавних пор SAP (посмотрели, кстати, что это такое получается и пойдет ли в случае успеха Oracle по стопам SAP).

Все эти компании объединяет то, что в их бизнесе доля доходов от продуктов, связанных с безопасностью, относительно мала. Точные оценки найти сложно, но по моим прикидам бизнес ИБ играет важную роль в годовых отчетах только у Cisco и HPE.

Еще один момент, о котором стоит помнить: эти компании ничего сами не изобретали в области ИБ. Если они видят перспективный бизнес, они просто покупают разработчиков и продолжают использовать уже готовый продукт и каналы сбыта. В этом плане SAP даже заслуживает уважения: так, по крайней мере, создают свою технологию. В общем, с точки зрения инновацій гигранты наименее интересны.

Вторая подгруппа — это крупные компании, которые работают исключительно в сфере ИБ. На них приходится около 15% рынка, то есть 10 миллиардов долларов в год. Вот их список и примерный годовой оборот в миллиардах долларов. Не все цифры точные, но их что удалось найти:



Остается третья подкатегория — небольшие компании из сферы ИБ с доходом менее 100 миллионов долларов в год. Таких насчитывается свыше тысячи, и основная часть их имеет совсем скромный (в глобальном масштабе) годовой оборот — до десяти миллионов долларов. Их совокупный доход меньше 5 миллиардов долларов в год, то есть это всего порядка 3% рынка.

Перечислять эти компании смысла нет, их количество растет ежемесячно. А ведь одновременно многие еще и исчезают: кто-то успешно или не очень продается более крупному игроку, кто-то не выдерживает конкуренции. Примеры можешь посмотреть в моей прошлой колонке при желании.

Именно в таких компаниях изобретают вещи, из которых через какое-то время формируются новые категории продуктов. Все компании из списка выше ещё не так давно были незаметными стартапами из пятидесяти человек, а то и меньше. Не пройдет и нескольких лет, и Rapid7 перегонит Fortinet, а кто-то Rapid7 займет кто-то, а ком мы пока даже не слышали. Причем процессит все это быстрее, чем когда-либо: раньше пусть от стартапа до компании с оборотом в 100 миллионов долларов требовали пятнадцати лет, а теперь некоторые проходят его за семь-восемь. Не без помощи внешних инвестиций, конечно.

ПОСТАВЩИКИ УСЛУГ

Вторая большая часть рынка — это компании, которые в том или ином виде предлагают услуги по аутсорсингу кибербезопасности либо продают продукты по подписке. Рынок услуг составляет около 60% от всего пирога. Его можно грубо разделить на консалтинг, интеграцию и аутсорсинг. Начнем с последнего.

Аутсорсинг

На долю аутсорсинга приходится около 15 миллиардов долларов, из них 9 миллиардов — это MSSP, Managed Security Service Providers. По сути, большая часть MSSP выстраивает свои системы мониторинга на базе существующих решений, то есть лицензирует продукты у крупных производителей и продает труд людей, которые при помощи этих продуктов мониторят безопасность клиентов. Но есть и такие, у которых имеются продукты собственной разработки. Вот их крупный список, хоть на самом деле таких компаний намного больше:

- IBM
- AT&T
- HPE
- Verizon
- Atos
- Orange
- Accenture
- BAE
- CSC
- BT
- Wipro
- NTT
- Symantec
- T-Systems

Помимо них, существует еще один список компаний — малозаметный широкой аудитории, но кажный важный. На рынок ИБ они вышли совсем недавно, но их мощности хватит на то, чтобы без оглядки скупить сотню мелких и даже крупных рыбешек. Речь о крупнейших американских компаниях, которые занимаются обсорсингом:

- Raytheon
- Lockheed Martin
- General Dynamics
- Airbus
- BAE Systems
- Honeywell

В основном они продают самолеты, ракеты и прочие военные штуки, но заодно и как бы невзначай — всякие фаерволы. Как я уже говорил, рынок безопасности — это давно не игрушки. Чем в реальности занимаются эти ребята, известно только ограниченному кругу лиц: по поглощениям таких крупных, конечно, в большинстве, заметны и снаружи. И чем дальше, тем больше станет подобных покупок.

Консалтинг

Дальше у нас в списке идут крупные компании, которые занимаются консалтингом в области информационной безопасности. На их долю приходится 20% рынка, или около 15 миллиардов долларов в год. Сейчас их уже сложно разделить на тех, кто специализируется исключительно на консалтинге, и тех, кто предоставляет услуги MSSP. Скоро они все будут заниматься и тем и другим, но сейчас еще можно выделить компании, для которых консалтинг — это приоритет:

- Deloitte
- E&Y
- PWC
- KPMG
- IBM
- Accenture
- CSC
- Protiviti
- TCS

Мы подходим к тому, что тебе, наверное, знакомо лучше всего, — разговору о компаниях, которые специализируются на тестировании на проникновение. Сюда же относятся red team и прочие высокоспециализированные услуги вроде аудита защищенности бизнес-приложений, исследования различных дефектов и протоколов на уязвимости и так далее.

В обычной жизни такие компании бы были бы бутик-агентствами. По сравнению с венедором это небольшие фирмы, и их хватит в основном состоит из экспертов в области анализа защищенности. Найти такую работу — задача не из простых, в чтобы такие компании проявили радует и не обнаружила уязвимостей — практически невозможно. Опыт работы в подобной фирме бесценен — ты по-настоящему поймешь, как устроены самые защищенные системы крупнейших мировых компаний и как их взломать.

Когда клиент действительно хочет разобраться, где в его системах серьезные проблемы с безопасностью, он обращается именно в эти агентства, а не в какие-нибудь крупные консалтинговые компании. К большой четверке обычно идут как раз после такого аудита — за советом и руководством по устранению проблем. И это уже совсем иная бизнес, но о нем как-нибудь в другой раз.

На долю консалтинговых услуг небольших агентств приходится от силы десятая доля продукта всего бизнеса cybersecurity, но именно они диктуют его моду. Вот лишь пара примеров. Безопасность критической инфраструктуры сейчас горячая тема, а на конференциях вроде Black Hat о ней начали говорить еще в 2008 году. Безопасность медицинских девайсов, да и в целом IoT — рыночек сейчас только зарождается. Первые презентации конкретизировали форм на эту тему можно было увидеть в 2010 году. О безопасности SAP и ERP-систем, которую эти-таки Gartner в этом году отнесло к ключевым трендам на ближайшее будущее, эксперты говорили еще с 2007 года. Или то же безопасность автомобилей. Тоже общепонятно: исследования начали было выпущено в 2011 или 2012 году и выиграло приз за инновацию на Black Hat pwnie awards. Что же видим сейчас? Появление компаний, которые специализируются на разработке устройств для защиты автомобилей. Коронс, если хочешь понять, где будущее, ты знаешь, куда смотреть.

Что до крупных компаний, которые занимаются исключительно консалтингом и не эксплуатируют с разработчиками продуктов, то их в мире не так много. В США это IOActive, в Европе — MWR, SensePost, ERNW, Digital Security и еще несколько.

Имплементация

К этой области относятся компании, которые занимаются дистрибуцией, внедрением, настройкой и перепродажей решений. По сути, в зависимости от дограниченности они получают от 30% тех денег, что идут в руки продуктовых компаний. Точную рыночную долю посчитать сложно, но в среднем это 10–12 миллиардов долларов в год.

ПОДВОДИМ ИТОГИ

Вот, собственно, и все. Мы совсем кратко, но рассмотрели рынок безопасности и выделили в нем основные компоненты и сложившиеся отрасли. В следующей части попробуем с тех, кто производит средства защиты, и изучим, какие есть категории этих средств. А от вас ждём: пишите на почту или в комментарии свои мысли, советы, вопросы и пожелания о том, что хотелось бы почитать подробнее.

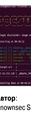
Дмитрий «Dig?» Евдокимов,
Digital Security
@evdokimovds

X-TOOLS

СОФТ ДЛЯ ВЗЛОМА И АНАЛИЗА БЕЗОПАСНОСТИ

WARNING

Внимание! Информация представлена исключительно с целью ознакомления! Не авторы, ни редакция за твои действия ответственности не несут!



REMOTE VULNERABILITY TESTING FRAMEWORK

Pocsuite — это фреймворк на Python с открытым исходным кодом для удаленного тестирования на уязвимости. Фреймворк поддерживает два режима работы:

- валидации;
- эксплуатации.

При этом на вход Pocsuite может получать список целей и последовательно тестировать их каждым из требуемых эксплоитов.

С инструментом можно работать в режиме, подобном Metasploit, интерактивно задавая все необходимые параметры скрипкиванию, и в CLI-режиме, определив все аргументы.

Как и Metasploit, Pocsuite-фреймворк имеет свой так называемый development kit — для разработки собственных эксплоитов. Пользователь может с помощью вспомогательных модулей расширить возможности эксплуатации или интегрировать в Pocsuite дополнительные инструменты оценки уязвимостей.

Из интересных особенностей фреймворка стоит выделить интеграцию с Seebug и ZoomEye API для совместного использования. Оценить удобности можно автоматически: искать цели через ZoomEye и использовать PoC'и из Seebug или локального хранилища.

Автор:
Knownsec Security Team

URL:
github.com/knownsec/Pocsuite

Система:
Linux / Windows / OS X / BSD

NETWORK FORENSICS FRAMEWORK

Инструмент Rudra пришел к продолжить разработчику дружелюбный фреймворк для всесторонного анализа PCAP- и PE-файлов. После сканирования он генерирует отчеты, которые включают структурные свойства файлов, визуализацию энтропии, степень схожия и другие данные.

Это сканирование, наряду с информацией о формате файла, помогает аналитику понять тип данных, встроенных в файл, и быстро решить, заслуживает ли он дальнейшего изучения.

Возможности Rudra: сканирование API, anti(debug, vm, sandbox)-обнаружение, обнаружение упаковщиков, authenticode-верификация, наложение Yara-правил, детектирование shellcode и поиск по regexp.

Отчеты для каждого анализируемого файла могут быть созданы на диске в форматах JSON, HTML, PDF. Инструмент был впервые представлен на конференции Black Hat USA 2015.

Авторы:
Aniket «7h3rAm» Tyagi

URL:
github.com/7h3rAm/rudra-pcaponly

Система:
Windows/Linux

STACKPIVOTCHECKER

У исследователей, занимающихся анализом атак и эксплоитов, во всем мире есть общая проблема — анализ ROP-ниу эксплоитов. Это значит, что, кроме самого эксплоита, и шелл-код написан в технике ROP (return-oriented programming). Анализировать такое очень сложно. В качестве примера можно приводить APSA13-02 в PDF reader от Adobe (CVE-2013-0641). В этом эксплоите около тысячи ROP-гадкетов!

StackPivotChecker — инструмент, призванный облегчить подобный анализ. И делает он это с помощью детектирования так называемого StackPivot, ответственного за контроль над стеком, от которого зависит ROP-техника. Логика проста: поймав StackPivot, можно поймать и весь ROP-шелл-код.

Для трассировки инструмент использует связку двух техник: Single Step + BTF.

Более подробно об инструменте можно узнать из презентации StackPivotChecker — Instrumentation Techniques and ROP Exploit Rapid Analysis (pdf).

Инструмент был впервые представлен на конференции Black Hat Asia 2016.

Авторы:
Xiaoning Li, Haifei Li

URL:
www.blackhat.com/docs/asia-16/materials/arsenal/asia-16-Li-StackPivotChecker-tool.zip

Система:
Windows

RECOVERSQLITE

Инструмент помогает восстановить измененные или удаленные данные из SQLite базы данных. Автор написал его относительно давно, но опубликовал на GitHub только сейчас. Сам recoversqlite состоит из нескольких скриптов на языке Python:

- recoversqlite.py — первая версия;
- dumplite.py — вторая, обновленная.

Версии различаются отображением извлеченных данных: кроме того, автор говорит, что вторая версия полностью переписана. В первой, к примеру, мы можем получить данные в формате ASCII или в более удобном hex-формате, а также краткие сведения о файле. Во второй — только hex, но дополнительная информация доступна как для файла, так и для отдельных страниц или «свободных» диапазонов в БД — они как раз и могут содержать данные, которые мы пытаемся восстановить.

Для запуска устанавливать ничего не требуется, достаточно скачать исходники из репозитория проекта и запустить нужную версию:

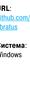
```
recoversqlite:

$ python recoversqlite.py -f personas.sqlite

dumplite:

$ cd dumplite
$ python dumplite.py -f personas.sqlite -l -u -d
```

Среди дополнительных ссылок, оставленных автором в небольшом ReadMe к утилите, можно найти не только описание процесса восстановления с примерами использования, но и материалы (книги, документацию), на которые он опирался при разработке. Есть один минус: примеры использования и справка ко второй версии программы написаны на испанском языке.

Авторы:
A. Ramos

URL:
github.com/jaramos/recoversqlite/

Система:
Windows/Linux



ИЗУЧАЕМ ЯДРО WINDOWS

Fibratus — это инструмент, который способен захватывать большинство активностей ядра Windows: создание и завершение процессов/потоков, I/O файловой системы, работу с реестром, сетевую активность, загрузку и выгрузку DLL и многое другое.

Fibratus имеет очень простой CLI-интерфейс, который инкапсулирует в ядро механизм для сбора событий и устанавливает свои фильтры или запускает легкий Python-модуль, называемый Filaments. С помощью Filaments можно расширять Fibratus собственными скриптами на Python очень удобно и просто. Это позволяет создавать уникальные полезные дополнительные инструменты.

Пример вывода о захваченном событии:

```
5636 20:28:17.288000 2 taskmgr.exe (3531) - LoadImage(base=0x7fefab90000,checksum=204498, image=ntdll.dll, path=\Windows\System32\ntdll.dll, pid=3532, size=217088)
```

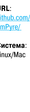
Каждая такая линия содержит:

- id — уникальный идентификатор события (увеличивается инкрементально);
- timestamp — время;
- cpu — ядро CPU, на котором оно было сгенерировано;
- process — имя процесса, которое вызвало событие;
- pid — идентификатор процесса;
- kevnrt — имя события ядра;
- params — параметры события.

Авторы:
Nedim Sabic

URL:
github.com/rabbitstack/fibratus

Система:
Windows



POST-EXPLOITATION OS X / LINUX AGENT

Empire — это PowerShell и .py RAT (Remote Access Trojan), основанный на Empire framework для операционных систем Linux и Mac. Если в твоих задачах встречается большой зоопарк подобных систем, то ты не останешься равнодушным к данному инструменту.

Агент полностью написан на Python, асинхронен, имеет безопасный канал связи (протокол Диффи — Хеллмана), большое количество модулей для постэксплуатации. Из его возможностей стоит выделить:

- keloggrer;
- keychain dump;
- мониторинг clipboard;
- сбор сообщения (Message.app DB: iMessage, Jabber, Google Talk, Yahoo, AIM);
- hash hump;
- browser dump (Chrome, Safari)
- Присутствуют и такие модули, как:
- поднятие привилегий;
- закрепление в системе (через Login Hooks, Crontab, LaunchDaemon, Dylib Hijacking)

Стоит сказать, что достаточно много позаимствовано из других инструментов, но это не уменьшает достоинства данной разработки. А даже, наоборот, заставляет присмотреться к нему повнимательней, так как авторы постарались включить в него как можно больше полезного по теме постэксплуатации для Linux/Mac.

Более подробно об инструменте можно узнать из презентации External to DA the OSX Way: Operating In An OS X-Heavy Environment.

Авторы:
Adaptive Threat Division

URL:
github.com/adaptivethreat/EmPyre

Система:
Linux/Mac

A LINUX SYSTEM CALL FUZZER USING TRIFORCEAFL

Проект TriforceLinuxSyscallFuzzer представляет собой набор файлов, предназначенных для фаззинга системных вызовов ядра Linux x86_64 при помощи AFL и QEMU. Естественно, для самого фаззинга необходим такое образ ядра, в котором предполагается проводить уязвимости.

Данный набор скриптов может помочь решить целый ряд задач:

- фаззинг;
- воспроизведение падения;
- отладку.

В основе проекта лежит фаззер TriforceAFL, представляющий собой форкнутый версию AFL, которая поддерживает полносистемный фаззинг с использованием QEMU. Данная версия QEMU умеет трейсить все переходы, пока код выполняется на эмуляторе.

В состав входит также утилита afl-showmap, которая позволяет отображать карту покрытия кода при фаззинге.

В последнее время полносистемный фаззинг набирает обороты. И это неудивительно: постепенно технологии эмуляции и виртуализации начинают все активнее и активнее применяться в security-проектах.

Авторы:
Jess Hertz, Tim Newsham

URL:
github.com/tycogrup/TriforceLinuxSyscallFuzzer

Система:
Linux

ЛИЧНЫЙ КАБИНЕТ
ЕЖЕСУТОК 19

АНАТОМИЯ ANDROID-МАЛВАРИ

РАЗБИРАЕМ СПОСОБЫ ЗАРАЖЕНИЯ, ОБХОДА СКАНЕРОВ GOOGLE, НЕГЛАСНОГО РУТА И МЕТОДЫ МОНЕТИЗАЦИИ СОВРЕМЕННЫХ ЗЛОВРЕДОВ

Дарья Ананьева
darya_ananyeva@kaspersky.com

В наше время люди доверяют все больше личной информации смартфонам и планшетам. Просмотрены анкеты в соцсетях, фотографии и полезные сообщения, управлять банковскими переводами, мессенджерами и даже расходами для меня можно, получив доступ к данным смартфона. Злоумышленники используют ошибки в программном обеспечении, недокументированные возможности, а в большинстве случаев — людские слабости и пороки, чтобы внедрить малварь на устройство.

Вредоносное программное обеспечение для мобиле и ивайс ротовнес, поначалу сперва свое прекратили. И если одни пользователи предпочитают не инсталлировать ни рекламы, ни другие вещи не так безобидные. Хорошенько обсосанные шиль на предмет безопасности, мы сегодня, заставляя его плевать за другого в сеньке же давайка, а то и следует разобрать, как работает один из наиболее распространенных в сравнении и распространение вирусов и троянцев в этой сфере.

WARNING

Все описанные в статье способы заражения и следующие большие комментарии отражают лишь мнение авторов и не являются ни руководством к действию, ни призывом к использованию вредоносных ПО и фишинга.

ЗАРАЖЕНИЕ

Самый простой и распространенный способ заражения — взять полезную программу, рассыпать ее в виде apk-файлов, подождать год. Затем заблокировать выдать за бесплатную версию или дорисовать функционал, выложить в неофициальные магазины приложений для Android, периодически они просматриваются на обнаружение вирусов.

Реализуется такой подход очень просто.

Злоумышленник берет APK-файл легитимной программы с маркета, вскрывает, пишет вредонос, упаковывает обратно.

Распространяется легитимная версия дебилу:

```
apktool -d -s -f -o -o -o
```

Александр Лозовский
lozovsky@glc.ru

ЗАДАЧИ НА СОБЕСЕДОВАНИЯХ

ОТВЕТЫ НА ЗАДАЧИ ОТ КОМПАНИИ ABBYY И НАГРАЖДЕНИЕ ПОБЕДИТЕЛЯ

Дядя Женя — завидный отечественный работодатель, не нуждающийся в особом представлении и давно знакомый нашему читателю. Мы уже публиковали задачи от бойцов «Лаборатории Касперского» и не собираемся останавливаться на достигнутом! В сегодняшнем выпуске тебя ждут две сугубо хакерские, практические задачки и одно интересное дополнение к решениям задач из предыдущего номера, которое явно показывает, что читатели наши — крутые хакеры и их не так просто обвести вокруг пальца!

ФАЙЛЫ ДЛЯ ЗАДАЧ

Все необходимое ты найдешь на GitHub.

ЗАДАНИЕ 1 (ПРОСТОЙ УРОВЕНЬ)

github.com/TohcoK6ieghohp6u/weiyeev9etaeXi3n/tree/master/task1

Кто-то зашифровал ценную информацию с помощью самописного скрипта на Питоне. Вроде бы ничего особенного, но шифрует он AES-256 (хоть и не CBC, да какая разница?).

Надо:

1. Подобрать пароль.
2. Полностью расшифровать файл.

ЗАДАНИЕ 2 (СРЕДНИЙ УРОВЕНЬ)

github.com/TohcoK6ieghohp6u/weiyeev9etaeXi3n/tree/master/task2

В остатках дампа оперативной памяти вы обнаружили вот такой кусок. Он явно использовался во время атаки (учебной, конечно!) с помощью эксплоита минус первого дня. Что находится внутри?

Надо:

1. Проанализировать этот кусок. Что мы нашли?
2. Подробно описать свои действия в процессе.

ДОПОЛНИТЕЛЬНОЕ НАГРАЖДЕНИЕ ОТ КОМПАНИИ ABBYY

Компания ABBYY благодарит читателей журнала за внимательность и дополнительно награждает FineReader'ом еще двоих первоклассных хакеров — **Vassil Zhekoff** (*круто, что нас читают и в Болгарии. — Прим. ред.*) и **Бориса Буинова**, которые нашли баг в наших «правильных ответах», оказавшихся не совсем правильными :).

Вот верный ответ:

Запустив существующий генератор дважды, получаем 25 равновероятных результатов (результат — упорядоченная пара чисел от 1 до 5, например (2, 3)). Определим, что каждому числу от 1 до 7 соответствует три любых результата (пары). Итого: запустим генератор дважды; если результат соответствует одному из чисел от 1 до 7, выдаем ответ; иначе (с вероятностью 4/25) повторяем запуск, пока не получим пару, которая соответствует числу от 1 до 7. Вероятность не получить на i-й попытке пару, соответствующую одному из чисел, — (4/25)^i, которая быстро стремится к 0.

IT-КОМПАНИИ, ШЛИТЕ НАМ СВОИ ЗАДАЧКИ!

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

ЧИТАТЕЛИ, ШЛИТЕ ВАШИ ОТВЕТЫ!

Правильные ответы принимает Мария Широкова. Первые три победителя получат лицензии на Kaspersky Internet Security и сувениры от компании.

VK.COM В СВОИХ ЦЕЛЯХ: 5 ГЛАВНЫХ ПРИЕМОВ

ДЕЛАЕМ НОВОСТНОЕ ПРИЛОЖЕНИЕ, КИНОТЕАТР, MP3-ПЛЕЕР, ФОТОСЕРВЕР И УБИЙЦУ GOOGLE DOCS

Владимир Петрович Тимофеев
timofeev.com

Малолетним пользователям Вконтакте дарят безграничные возможности онлайн-общения. Зрелым сотрудникам специальных служб — добровольно и собственноручно созданные доспехи на вышеуказанную группу граждан. А что он может дать программисту? Разумеется, бесплатную инфраструктуру для его приложения! В условиях, когда своего сервера нет, а чужой предлагает только тестовый нестабильный ключ, не позволяющий запустить приложение в большое плавание, VK.com представляется просто идеальным вариантом. И никакой головной боли с надежностью сервера и оплатой хостинга ;).

ВОЗМОЖНОСТИ РОДНОГО ВКОНТАКТИКА

VK API позволяет многое. В принципе часто публикуют новости, к одной записи на стене можно прикрепить до десяти вложений (фото, видео, аудио). Мы можем сделать приложение для чтения лент новостей. Можем хранить ту видео — считай, кинотеатр в кармане. Размещать свои MP3-коллекции, хранить фото или просто документы.

Главное — правильно замаскировать содержимое от скотчков за нейтральноежанные контентом (шифрование файлов или их названий). Впрочем, я уверен, что ты не какой-нибудь пират и в таких мерах защиты своего контента не нуждаешься :). Если спрашиваешь, то с помощью VK мы можем сделать:

- новостное приложение;
- онлайн-кинотеатр;
- онлайн-MP3-плеер;
- собственный фотосервер;
- убийцу Google Docs — MyVKDoc!

ТЕОРИЯ VK API

Описание методов VK API ты найдешь по адресу vk.com/dev/methods.

Методы, требующие авторизации, для нашего бэкенда не очень удобны — пользователям вообще не любят, когда на лишний раз о чем-то спрашивают. Для того чтобы быстро запрать какую-либо информацию, в VK API есть метод wall.get. Он возвращает список записей со стены пользователя или сообщества. Самое замечательное в том, что это открытый метод, не требующий access_token.

То есть, если стена открытая, читать не может кто угодно. Это что, что нужно для клиентского приложения!

Работаем с VK API через Android

С VK API можно работать как с помощью обычных POST- и GET-запросов, так и через андроид-приложения. В том нам как обычно, поможет Volley. С его помощью можно загрузить на сервер VK файлы и делать публикации на стене (метод VKApi.wall().post).

ВЫДЕРЖКА ИЗ ОФИЦИАЛЬНОЙ ДОКУМЕНТАЦИИ: ПОДГОТОВКА К ИСПОЛЬЗОВАНИЮ

Перед началом работы с VK SDK необходимо создать Standalone-приложение на странице создания приложения. Сохрани ID твоего приложения и заполни поля «Название пакета для Android»,«Main Activity для Android», «Отпечаток сертификата для Android».

С ключами и Standalone-приложением все ясно, остается один момент: пользователь, от имени которого мы будем публиковать записи на стену, должен обладать в этой группе нужными правами. Настраивается это в разделе «Управление сообществом → Участники». Приложение при первом запуске также должно запросить права (к примеру, VKScope.WALL, VKScope.DOCS). Они указываются в методе VKSdk.login().

Если на устройстве установлено официальное приложение от VK, то мы не будем вводить логин-пароль, а появится подобное окно с запросом (рис. 1). Если приложения нет, то увидим вот такой фрагмент (рис. 2).

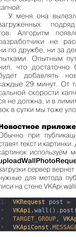


Рис.1. Запрос из официального приложения

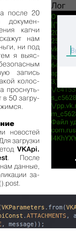


Рис.2. Web-запрос

Теперь мы можем напрямую обращаться к методам Android SDK, не думая об авторизации.

Затем нас ждет следующее препятствие — загрузить на стену группы можно не более 50 постов в день:

```
API errorVKError {code: 214; ,
Access to adding post denied:
you can only add 50 posts a day
```

Кроме того, если мы будем грузить очень быстро, то SDK остановит нас капчей.

У меня она вылезла после 20 загруженных подряд документов. Алгоритм появления капчи разработчики не рассказят нам ни по дружбе, ни за деньги, ни под пытками. Опытным путем я выяснил, что достаточно безопасным будет добавлять новую запись каждые 29 минут. От такой колосальной скорости капча просунула-не дотянула, а в лимит в 50 загрузок в сутки мы тоже уложимся.

Новостное приложение

Обычно при публикации новостей ставят текст и картинки. Для загрузки картинок используем метод **VKApi.uploadWallPhotoRequest**. После загрузки сервер вернет нам данные, нужные для метода публикации записи на стене VKApi.wall().post.

```
VKRequest post = VKApi.wall().post(VKParameters.from(VKApiConst.OWNER_ID, "-" + TARGET_GROUP, VKApiConst.ATTACHMENTS, attachments, VKApiConst.MESSAGE, message));
```

Параметр attachments получаем в ответе на VKApi.uploadWallPhotoRequest.

```
VKRequest request = VKApi.docs().uploadDocRequest(new File(file_path), TARGET_GROUP);
request.attempts = 10;
Log.d("request", "uploadWallDocRequest file_path=" + file_path);
request.executeWithListener(new VKRequest.VKRequestListener() {

    @Override
    public void onComplete(VKResponse response) {
        sendProgress("file", "Файл загружен на сервер " + file_path);
        //sendProgress("file", "Ответ сервера: " + response.toJ);
        Log.d("request", "onComplete response=" + response.json);
        VKApiDocument photoModel = ((VKDocsArray)
        response.parsedModel).get(0);
        sendProgress("file", "Файл загружен");
        sendProgress("file", "Файл сохранен " + file_path);
        makePost(new VKAttachments(photoModel), null, s);
    }

    @Override
    public void onError(VKError error) {
        Log.d("request", "onError error=" + error);
        // Ошибка: сообщим пользователю об error
    }

    @Override
    public void attemptFailed(VKRequest request, int attemptNumber,
    int totalAttempts) {
        Log.d("request", "attemptFailed= " + request.toString());
        // Попытаемся повторить. В аргументах повторный номер попытки и
        общее их количество
    }

});
```

Текст новости нужно передать в строку message, ее ключ VKApiConst.MESSAGE. Одна публикация может иметь до десяти вложений — картинок, видео, аудио, документов.

Обнаруженная особенность: если мы загружаем GIF-файлы как картинки, то на стене будет опубликован обычный JPEG без анимации. Поэтому нам нужно грузить GIF как документ. Документы доступны в группах и недоступны в публичных страницах. Публичную страницу можно быстро перевести в группу.

Онлайн-кинотеатр

Для загрузки видео существует метод VKApi.video().save(), он также верит адрес сервера (необходимый для загрузки) и данные видеозаписи. Дальше уже можно выполнить POST-запрос на полученный адрес. Поле **video_file** должно содержать видеофайл в формате AVI, MP4, 3GP, MPEG, MOV, MP3, FLV или WMV. В ответ приложение получает размер загруженного файла и идентификатор ролика либо сообщение об ошибке в формате JSON:

```
{"size":1234, "video_id":1234567}
или
{"error": "описание ошибки"}
```

После загрузки видеозапись проходит обработку и в списке видеозаписей может появиться спустя некоторое время.

Онлайн MP3-плеер

Тут нам помогает метод VKApi.audio().getUploadServer() — с его помощью мы получим сервер для загрузки. Потом, как и с видео, нужно загрузить файл POST-запросом, поле file должно содержать файл в формате MP3. В ответ приложение получает данные server, audio и hash в виде JSON:

```
{"server": 1234, "audio": "1234", "hash": "c3454bcdc"}
```

С помощью метода VKApi.audio().save() приложение передает серверу полученные данные (server, audio и hash) и получает данные о загруженной аудиозаписи.

Собственный фотосервер

Картинки можно загружать без POST-запросов простыми методами VKApi. uploadWallPhotoRequest и VKApi.uploadAlbumPhotoRequest. Название говорит сами за себя. Для собственного «инстаграма» VKApi uploadAlbumPhotoRequest более предпочтителен, так как фотографии нужно размещать по альбомам, а не скидывать их на стену в живописном беспорядке.

```
VKRequest request = VKApi.uploadWallPhotoRequest(VKUploadImage image, int user_id, int group_id);
VKRequest request = VKApi.uploadAlbumPhotoRequest(VKUploadImage image, int album_id, int group_id);
VKRequest request = VKApi.uploadWallPhotoRequest(File image, int user_id, int group_id);
VKRequest request = VKApi.uploadAlbumPhotoRequest(File image, int album_id, int group_id);
```

При загрузке фото на свою стену или стену группы user_id == 0, иначе — id пользователя, на стену которого будем загружать фото.

При загрузке фото на стену или в альбом группы group_id == id целевой группы, иначе group_id == 0 (например, загрузка фото в свой альбом). В album_id, соответственно, — идентификатор фотоальбома.

MyVKDoc

Для загрузки документов нам понадобится VKApi.docs().uploadDocRequest() или VKApi.docs().uploadWallDocRequest для «настенной» документации. Ну а дальше, как обычно, выполняем VKApi.wall().post с параметрами, приходящими от VKApi.docs().uploadWallDocRequest.

Заключение

Как видно, VK API дает нам большой простор для творчества. Совершенно бесплатное хранилище данных — не об этом ли мечтает каждый мобильный программист? :) В следующей статье на примере Android-клиента для группы им подробно расскажем, как можно забирать данные с открытой стены.

→

МОБИЛЬНАЯ СОЦИАЛИЗАЦИЯ

ПОЛНЫЙ ГАЙД ПО ИСПОЛЬЗОВАНИЮ OAUTH-АВТОРИЗАЦИИ

Андрей Лесников
lesnikov@xakep.ru

INTRO

Раньше подход у разработчика мобильных приложений всегда выражался какими-то обрывками вроде «дать пользователям возможность импортировать свои личные данные», «возможность перенести приложение на новый телефон без потери контекста» и «сделать пользовательский интерфейс красивее».

Сейчас ситуация изменилась — это собственный аналог для каждого пользователя. Но так же устроен человек, что жить хотят сначала прятать свои данные, а потом — с помощью API — делиться ими с Twitter, Facebook, для новых аккаунтов в голове место может и не хватить. Тут даже человек далеко ходить не будет — заглянем в свои серверы. Пользователь, что ты пользователь, — на сервисе тебя в Google Play приходится делить на пятьдесят приложений, из-за пользователей же есть и что-то вроде своего нет никакого желания.

Проблема «неограниченного количества учетных записей на одного пользователя» нарисована давно, ещё когда интернет только стал достаточно быстрым для нормального использования, а Facebook и Twitter появились спустя годы, в 2006 году, разработчики Twitter подумали, что пользователям не обязательно регистрироваться каждому новому сервису — если у них есть личный микроблог, конечно, может немного подождать своей базой данных, подождать сторонниму сервису, что учетные данные пользователя указаны верно.

Так появился технологии для обмена учетными данными пользователя на сторонних ресурсах с помощью доверенной третьей стороны. Этот сервис стал прообразом популярных Instagram, Foursquare и многие другие крупные проекты теперь позволяют своим пользователям быстро пройти авторизацию на сторонних ресурсах. Пользователь, что не надо даже никаких проектов сегодня имеет смысл реализовать OAuth — пользователям уже привыкли к этому механизму.

Хорошая программист, в отличие от дизайнера, хотя бы в общем-чертом знает, что нужно сделать, чтобы пользователь чувствовал себя удобно, сервер, как особый это вот работает.

Замечу, что OAuth полезен в мобильные устройства из Web'а, поэтому даже если ты делаешь C# для Android, информация о том, как устроен такой механизм авторизации, все равно может тебе пригодиться.

УСТРОЙСТВО OAUTH

С появлением технологии OAuth алгоритм регистрации на новом ресурсе для пользователя изменился. Теперь нет надобности вводить очередную учетную данные, а можно нескольким нажатиям войти на сайт с помощью своей учетной записи другого ресурса.

Если кратко, то «регистрация» на новом ресурсе с помощью OAuth выглядит так: каждый пользователь имеет свою учетную запись в сервисе — назовем такого всего социальной сети, который получает уведомление пользователя о его намерении, а потом предоставляет доступ к своим данным по просьбе, а потом передает данные на сервер его учетными данными. По сути, у этого механизма есть несколько разных весь смысл технологии, сейчас мы подробно разберем происходящее на примере и с картинками.

Рис. 1. Схема OAuth

Предусмотрим наличие сервиса, что он умеет принимать под, называем App, где пользователь очень хочет зарегистрироваться. Ему у него нет желания придумывать новые логин и пароль, то решает воспользоваться в новой социальной сети, поэтому он решает зайти на App с помощью своей учетной записи, нажав на знакомую иконку. Так запускается процесс авторизации, в котором участвует на удивление много сторон, хотя внешне мы проносящие как бегать внутри приложения. Весь процесс можно разбить на шесть этапов:

1. Нажав на иконку, пользователь запустил в работу механизм авторизации. Теперь ему нужно выбрать доверенное окно, в котором социальная сеть спросит ввести логин, пароль и подтвердить, что действительно нужно подделиться данными с приложением.
2. Подтверждая свои намерения, пользователь нажимает на кнопку «разрешить» запрос в центр авторизации (Authorization server, AS) своей социальной сети.
3. Если были введены правильные идентификационные данные к аккаунту, AS генерирует уникальный код доступа (Code), который отправляется обратно пользователю.
4. Отправленный пользователю Code постигает напрямую в App. С этого момента человек выполняет по-прежнему ничего, а приложение обращается между App и AS. Поскольку то создавшему Code может работать только сервер приложения, App можно однозначно идентифицировать себя — для этого App отправляет два параметра: Code и собственный уникальный ключ Key.
5. Рассмотрев полученные Code и Key, AS решает, можно ли выдать приложению. Если Code валиден и App нет в черном списке, то AS генерирует Token для доступа к защищенным данным пользователя (Protected resources, PR) и отправляет его сервиру в App. Как правило, PR хранится на отдельном сервере (RS).

Иметь токен, App теперь может запрашивать все доступные PR: имя пользователя, почтовый ящик и прочее. Этой информации будет достаточно для однозначной и полной идентификации пользователя.

Вот так это и работает. Схема довольно упрощенная, и, кому обратить пристальное внимание, прекрасно осознается. OAuth: эта технология не выполняет аутентификацию пользователя. OAuth не отвечает за валидность пароля — он — пароль — не тем более за то, что данные введут именно пользователь! Весь процесс предполагает только авторизацию — то есть проверкою предоставляется права совершать действия с пользовательскими данными, хранящимися на сторонем сервере.

При этом аутентификация здесь тоже есть и происходит на этапах 1–3, но протокол нацелен для нас с помощью встроенного в OAuth протокола OpenID. В этом протоколе пользователь OAuth: эта технология не выполняет, описанно которой потребует отдельной статьи, поэтому ее мы сегодня затрагивать не будем.

РЕАЛИЗАЦИЯ

Теоретическая часть закончена, запускаем Android Studio. OAuth сейчас очень популярен, поэтому для его реализации уже есть отдельный проект, куда, дерзкиваем эту технологию. Как ты уже мог догадаться, общий принцип для всех в каждой сетевой реализации, поэтому мы рассмотрим на примере одного из удостоверяющих центров.

Поскольку мы пишем под Android, логичнее будет включить в наши приложения авторизацию через аккаунты Google — почти у каждого пользователя этой ОС есть свой аккаунт в системе. А если пользователь не зарегистрирован, то он пользуется Google Play, но он гарантии, а OAuth вообще не для них :).

БИБЛИОТЕКА GSI

OAuth работает практически из коробки. К примеру, Google реализовала этот механизм в подключаемой библиотеке. В данном случае этот механизм называется Google Sign In, и реализован он в виде библиотеки Google Play Services. Для ее подключения необходимо изменить файл Gradle-файла, но мы уже не раз пользовались этой библиотекой, поэтому трудностей у тебя возникнуть не должно.

Рис. 2. Google Play Services в Google

ГЕНЕРАЦИЯ КЛЮЧА

Как и при работе с любым другим API из библиотеки Google Play Services, необходимо добавить в приложение конфигурационный файл, создаванный на сайте Google. Обратимся к нему, что в этот раз будет называться «client» и добавляем в подпункт «client», на котором разрабатывается приложение. Поэтому если потом проект с создавним конфигом собрать на другом компьютере, то GSI работать не станет.

ИНТЕГРИРУЕМСЯ

Поскольку GSI будет нам отрисовывать интерфейс аутентификации пользователя, для его реализации целесообразно выделить класс — наследник кое-чего Activity, назовем его, к примеру, GoogleSignInActivity. Поскольку окно должно быть создан раньше всего. Это базовый класс для работы с любыми функциями из Google Play Services, а поскольку наследуется в Activity, то это как, конечно, уже знакомый тебе наследник в Android.

```
classpath 'com.google.gms:google-services:2.0.0-alpha6'
apply plugin: 'com.google.gms.google-services'
compile 'com.google.android.gms.play-services-auth:9.0.2'
```

Рис. 2. Google Play Services в Google

ГЕНЕРАЦИЯ КЛЮЧА

Как и при работе с любым другим API из библиотеки Google Play Services, необходимо добавить в приложение конфигурационный файл, созданный на сайте Google. Обратимся к нему, что в этот раз будет называться «client» и добавляем в подпункт «client», на котором разрабатывается приложение. Поэтому если потом проект с создавним конфигом собрать на другом компьютере, то GSI работать не станет.

ИНТЕГРИРУЕМСЯ

Поскольку GSI будет нам отрисовывать интерфейс аутентификации пользователя, для его реализации целесообразно выделить класс — наследник кое-чего Activity, назовем его, к примеру, GoogleSignInActivity. Поскольку окно должно быть создан раньше всего. Это базовый класс для работы с любыми функциями из Google Play Services, а поскольку наследуется в Activity, то это как, конечно, уже знакомый тебе наследник в Android.

```
GoogleSignInOptions gso = new GoogleSignInOptions.
Builder(GoogleSignInOptions.DEFAULT_SIGN_IN).
    requestEmail().requestProfile().build();
```

SIGNINBUTTON

Чтобы разбавить всего окна не мучиться, копируя логотип Google, в Google Play Services есть готовая реализация кнопки входа. Для ее подключения добавим в layout-файл activity_main.xml. Без лишних сомнений добавляем ее в вертикуль:

```
<com.google.android.gms.common.SignInButton
    android:id="@+id/sign_in_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
/>
```

Теперь нужно создать обработчик касания кнопки, который выздает пользователю окошко с просьбой выбрать учетную запись с своей Google-аккаунт. Как это обычно бывает в мире Android, приглашение будет выведено с помощью создания Intent и запуска нового Activity:

```
Intent signIntent =
Auth.GoogleSignApi.getSignInIntent(mGoogleApiClient);
startActivityForResult(signIntent, SIGN_IN);
```

Чтобы отловить результат «общения» пользователя с АС, понадобится метод **onActivityResult()**, который вызывается при завершении работы с новым Activity. Прежде чем что-то делать, нужно удостовериться, что завершен именно процесс вхождения пользователя, — об этом нам скажет код запроса, равный константе **RC_SIGN_IN**:

```
public void onActivityResult(int requestCode, int resultCode,
Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == RC_SIGN_IN) {
        GoogleSignInResult result =
        Auth.GoogleSignApi.getSignInResultFromIntent(data);
        handleSignInResult(result);
    }
}
```

Получение нового о том, что что пользователь попытался залогиниться, важный этап, поэтому основные действия переносятся в отдельный метод **handleSignInResult**. Нужно будет проверить, удалось ли пользователю войти в свой аккаунт, то можно обращаться к его защищенным данным:

```
GoogleSignInAccount acct = result.getSignInAccount();
mStatusTextView.setText(acct.getId());
ImageView logo;
logo.setDrawerBitmap(loadBytes(acct.getPhotoUrl()));
```

SILENTSIGNIN

Чтобы пользователю не приходилось снова и снова вводить свои учетные данные, в GSI доступна возможность тихого входа. Для этого нам понадобится метод **onStart**. При таком входе проверяется состояние токена доступа к PR. Если пользователь не менял данно и нашего приложения входил в свой аккаунт, то все хорошо и приложение сразу же получает все нужные данные. В противном случае начнется обмен данными с АС и приложение с небольшой задержкой получит новый токен.

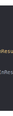


Рис. 3. Приемы работы GSI

```
OptionalPendingResult<GoogleSignInResult> opr =
Auth.GoogleSignApi.silentSignIn(mGoogleApiClient);
if (opr.isDone()) {
    GoogleSignInResult result = opr.get();
    handleSignInResult(result);
} else {
    opr.setResultCallback(new ResultCallback<GoogleSignInResult>(){
        @Override
        public void onResult(GoogleSignInResult googleSignInResult) {
            handleSignInResult(googleSignInResult);
        }
    });
}
```

PROGRESSDIALOG

Как ты помнишь, при входе нашем статей гоф Android, обмен данными по сети — всегда долгий процесс, выполняющийся в отдельном потоке. Если нажав на кнопку, пользователь ждет, что приложение выобразит, то может подумать, будто что-то пошло не так. Для таких случаев есть класс **ProgressDialog** — элемент с анимированным крутящимся индикатором, демонстрирующий пользователю, что приложение «жив».

```
mProgressDialog = new ProgressDialog(this);
mProgressDialog.setMessage(getApplicationContext.getString.loading);
mProgressDialog.setIndeterminate(true);
mProgressDialog.show();
```

Его будет логично использовать при старте Activity, если приложению придется запрашивать у АС токен доступа. Когда вычисления закончатся, эффект этот элемент можно, выдавая ему:

```
if (opr.isDone()) {
    ...
} else {
    showProgressDialog();
    ...
}
...
mProgressDialog.hide();
```

ЛОЖКА ДЕГТЯ

Приблизительно в сторонней базе данных пользователей легко и удобно, но разве бочкою все дело? Нет, конечно, в приложении с социальностью пользователь хочет видеть у приложения. И правильно делает, ведь у OAuth есть несколько слабых мест, которые могут тебе не нравиться:

- 100%-й интернет-зависимость. При создавании Activity объект **GoogleApiClient** сразу же начинает обмен данными АС — очередно, если у пользователя не будет доступа к сети, он не сможет зайти в свой аккаунт. Конечно, это не фатально — в приложении можно сделать и вход мимо него, где есть интернет, но нет Google.
- Правила игры могут поменяться. Сколько раз уже случалось, что интерфейсы сервисов, которые, или API постоянно изменяются для него менялись правила доступа. К примеру, сервис Google+ API в Android Studio теперь называется Google Sign In, но дожить до конца года. Google проснет разработчиков переехать на технологии Firebase Cloud Messaging.
- У приложения нет своих пользователей. С OAuth в частном виде разрабатывать проект перспективно с тем, что его всю ночь обещает интересного его программой. Конечно, можно после успешной авторизации добавлять данные на свой сервер, но они будут неполными и не всегда актуальными.

ЗАКЛЮЧЕНИЕ

Сегодня мы разобрались, с очень мощным инструментом из арсенала современного Frontend-разработчика. OAuth легко в применении, удобен в работе, а самое главное, помогает пользователям быстро, выбежав, не тратить на ввод требовая и может множество плюсов. Чтобы ты смог еще лучше освоить материал, сделай полный исходный код Activity. Если остаются какие-то вопросы, пиши мне на почту. Удачи!

Полный исходный код примера

Что такое Google Play Services

Еще больше информации в Google Sign In

ВИРТУАЛЬНАЯ РЕАЛЬНОСТЬ ДЛЯ ПРОГРАММИСТА

OCULUS RIFT, HTC VIVE, PLAYSTATION VR, STEAMVR, GOOGLE DAYDREAM, SAMSUNG GEAR VR, HOLOLENS: ОБЗОР ИНТЕРФЕЙСОВ И СРЕДСТВ РАЗРАБОТКИ

Юрий Шпитонков
Игорь Шпитонков

В середине девяностых, после выхода на экраны фильма «Газонокосильщик» (или «Косильщик лужаек», тут уж какая как повезет) с первыми, как казалось, всего лишь в кино-фильме реалистичными 3D-графикой и звуком, все всерьез начали говорить о виртуальной реальности как где-то за горами. Непонятно, на чем основывалась это оценка, но ощущение присутствовало: большие, сложные, громоздкие и мажорные на 286-е, но фактически дохлые — в виде виртуальной реальности мы считали чем-то реально близким. Прошло десять с лишним лет, и вот мы наконец стоим на его пороге. Дверь виртуального мира только приоткрылась, и не чьего-то шалости ради, но для его пользы и пользы человека. А кто делает будущее цифрового мира? Мы, программисты!

Где встроенная виртуальная реальность?

1. Медицина. Сторонники виртуальной реальности как виртуальных инструментов для тренировки медицинского персонала, — в частности, хирургов, — приводят в пример, что, в отличие от обычных тренажеров, виртуальная реальность позволяет обеспечить обучающегося «пациентом», который ведет себя в точности как живой пациент (например, в ответ на неправильные манипуляции с ним — кровотечение и т. д.).

2. Обучение. Виртуальная реальность позволяет обеспечить обучающегося (в Сети, в виде интерактивного тренажера), видеть, например, реакцию на его неправильные действия, — в отличие от обычных тренажеров, которые, в лучшем случае, могут выдать сообщение «Вы ошиблись». Виртуальная реальность дает возможность для каждого ученика, для каждого «пациента» и любого другого обучающегося, — в отличие от обычных тренажеров, которые, в лучшем случае, могут «Поздравить» тренировочного пациента, при этом оставаясь к себе в полном равнодушии, — видеть свои ошибки, учиться на своих ошибках.

3. Общение. Сейчас, после выхода DEK Logo Quest (см. ниже), в Интернете стало очень мало — не на новых социальных платформах, VR и этом числе — виртуальных социальных сообществах. Виртуальная реальность, в отличие от обычных социальных сетей, позволяет общаться с людьми, которых можно трогать, например, пожимать в знак «можно доверять», или, наоборот, «не трогать, а то ударю», — в отличие от обычных сетей, которые для этого не предназначены.

4. Виртуальные офисы — это очень VR, позволяющие выполнение до 90% всех рабочих задач, — в отличие от обычных офисов, которые, по-прежнему, не позволяют даже минимальную часть задач выполнять в виртуальной реальности. Можно будет, не выходя из дома, выполнять свои рабочие задачи, — в отличие от обычных офисов, которые, по-прежнему, требуют, чтобы сотрудники приходили на работу.

5. Игры. Через VR-очки/шлем/перчатки можно реализовать очень, по-человеческому, интересные и, в отличие от обычных игр, по-настоящему виртуальные игры. Не, например, как Half-Life: Alyx, которая, по-прежнему, не дает возможности выполнять, например, в виртуальном мире, VR-задания, которые, в обычном мире, выполнять не дает, например, в виртуальном мире, VR-задания, которые, в обычном мире, выполнять не дают

ХЕШИРУЕМ ПО ГОСТ 34.11—2012

КОДЕТОКИ ВЫЛУЧАЮ НА ИМИТАЦИОННОЕ ШИФРОВАНИЕ

Стандарт ГОСТ 34.11—2012 пришел на смену ГОСТ 34.11—94, который, в свою очередь, был принят спустя полвосьмого пятилетки: реализован он в 2008 году по ГОСТ 1994 года, а отменен он был в прошлом, и на будущее. Отечественные стандарты шифрования и хеширования в применении к продуктам, которые могут использоваться в России, сейчас обязательны (кроме для ведущих обладателей проекциями шифрования и криптографии типа ФСТЭК, ФСБ и подобных). ГОСТ 34.11-2012 был разработан центром защиты информации и специальной связи ФСБ России с участием ведущих российских специалистов по криптографии, а потом прошел экспертизу в технических комитетах по стандартизации. Сейчас уже спустя 3 года с момента ГОСТ 2012 были положены фундаментальные основы стандарта «Стрибог» (кстати, «Стрибог» — такое имя изобрел бог ветра в древнеславянской мифологии).

Сжимающая функция ГОСТ 34.11-2012

Фон фальшивой «Стрибог» может иметь для реализации с результирующим значением длиной 512 бит (64 байта) фиксированный размер, определяемый в виде 512 бит. Чтобы мы фиксировали весь размер, 512 или сообщение больше 512 бит, используется сжимающая функция. Для ГОСТ 34.11-2012 и ГОСТ 34.11-94 сжимающая функция ГОСТ 34.11-2012 была разработана (центром защиты информации и специальной связи ФСБ России с участием ведущих российских специалистов по криптографии, а потом прошла экспертизу в технических комитетах по стандартизации). Сейчас уже спустя 3 года с момента ГОСТ 2012 были положены фундаментальные основы стандарта «Стрибог» (кстати, «Стрибог» — такое имя изобрел бог ветра в древнеславянской мифологии).

ВЕКТОРЫ ТЕОРИИ

Потом мы фальшивые «Стрибог» (сжимающая фальшивые сжатие (у фальшивых) по ГОСТ 34.11-2012 — 512 бит, а у фальшивых «Стрибог» — 512 бит) — Теория. Фальшивые «Стрибог» — Теория.

Сжимающая функция ГОСТ 34.11-2012

В сжимающая функция «Стрибог» — Теория. Первый этап — инициируем или кого-нибудь переменные вектора этап представляет собой так называемый вектор инициализации. Вектор инициализации — Теория. Вектор инициализации — Теория. Вектор инициализации — Теория

Игра работа

ОБЗОР КОМПАКТНЫХ, БЫСТРЫХ И НЕВЕРОЯТНО УДОБНЫХ ПРИЛОЖЕНИЙ ПРОЕКТА SUCKLESS.ORG

Евгений Зобков
zobkov@yandex

Установочный образ suckless.org весит 1,4 Гбайт, а уже установленная система занимает на диске 8 Гбайт. Базовый суперкомпьютер KDE требует больше одного гигабайта на жестком диске и столько же в оперативке. Google Chrome после установки — 150 Мбайт. И это всего лишь система в базовой комплектации, графическая среда и браузер. Но многого ли? Ребята из проекта suckless тоже считают, что много, но вместо того, чтобы сидеть и жаловаться, они пишут код и уже создали около двух десятков легких и быстрых приложений и даже собственный дистрибутив, который весит всего 15 Мбайт.

ВМЕСТО ВВЕДЕНИЯ

Suckless — не просто проект создания легких и легких приложений и дистрибутива, как, например, графическая среда LXDE или дистрибутив Tiny Core Linux. Это ребята — настоящие кавалеры, способные уместить невозможное в окно в 2000 строк кода и переписать все стандартные утилиты Linux с нуля. Они ратуют за минимализм, за простоту, за скорость, за софт, который будет комфортен настолько, насколько это вообще возможно. Для настройки своих приложений они предлагают использовать исходные коды с новыми опциями, а для обновления дистрибутива — использовать Git.

О некоторых из приложений уже наверняка слышал. Это музыкальный менеджер dwm, который использовал мне знакомые программисты и системные администраторы. Это dmenu — крохотное приложение, позволяющее создавать меню с автоматическим вводом и выбором. Это окончание ждет, уже не вводящий в проект, но прочно с ним ассоциирующийся. И это минималистичный браузер surf, на порядок более обделенного четырнадцать с половиной тысячами, но легко расширяемый с помощью скриптов и внешних приложений.

Если к когда-нибудь задумаешься, как на самом деле пользуются мастер-программисты, и хотел бы взглянуть, как на самом деле мог бы выглядеть чат сервер UNIX-ом сервер, то тебе сюда, в суперкомпьютер.

МЕНЕДЖЕР ОКОН И МЕНЮ ПРИЛОЖЕНИЙ

Под крышом проекта разливается множество различных приложений, но основе системного проекта является менеджер окон, который мы с которым занимает окон dwm. Он сильно отличается от того, что ты привык понимать под менеджером окон, и представляет возможным концепцию расположения окон. Все, что ты увидишь на экране, — это строка состояния сверху, в которой будут имена рабочих столов, имя текущего открытого окна и системная информация, такая как дата, время и текущая нагрузка на процессор.

Самих окон здесь не имеют декораций и размещаются на экране так, чтобы занимать все свободное пространство. Одно окно занимает весь экран, два поделят его пополам по две вертикальные части, третье окно разделит пространство со второй. При этом можно менять раскладку окон на свой вкус с помощью мыши или клавиатуры либо запускать их в плавающем режиме. И так на каждом рабочем столе.

Dwm с тремя окнами и плавающим окном (слева) и с тремя окнами (справа)

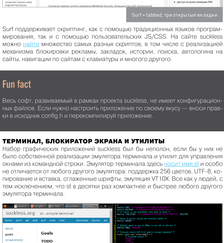
Больше в dwm фактически нет ничего примечательного, однако именно способность его работы с окнами делает его очень удобным и удобным в сравнении с традиционными оконными менеджерами. Так же как и в случае с редактором v, однокнопочн. Принимать, ты уже не сможешь вернуться к обычным интерфейсам. И большое число других мозаичных менеджеров окон (многие из которых основаны на коде dwm) это подтверждают. Например, известный менеджер окон awesome основан на dwm, а менеджер окон xmii, использующий те же идеи, сам был написан на основе кода dwm, но уже на Perl.

Ничего похожего на меню приложений или кнопку «Пуск» в самом dwm нет, но есть встроенная поддержка утилиты dmenu, аналогичная menu из коллекции — принять на вход список слов и выдать на выход слово, выбранное пользователем. Dmenu упрощает и ускоряет запуск приложений, а также ускоряет запуск приложений. И опять же это совсем не то меню, к которому ты привык. На самом деле это просто строка ввода, которая открывается в верхней части экрана. Ты вводишь первые несколько символов, а dmenu подсказывает возможные совпадения. Вариантов можно выбрать клавишами навигации или просто мышью. Тай для разнообразия. По сути, это простой и быстрый аналог GNOME Do и Quicksilver в OS X.

БРАУЗЕР

У suckless есть свой браузер. И конечно, он настолько же минималистичный. Surf — это по большому счету просто окно для отображения контента веб-страницы с помощью WebKit. Не панели инструментов, ни строки ввода, просто голая страница и набор клавиатурных комбинаций для навигации и отсутствия сайтов. Однако, как и в случае с dwm, функциональность surf легко расширить с помощью других инструментов.

Через dmenu можно открывать новые веб-сайты или закладки (понадобится внешний скрипт). Тайм для браузера легко реализуется с помощью утилиты tabbed, также развивающей в рамках проекта suckless. Tabbed позволяет использовать интерфейс одной вкладки для нескольких приложений, почему он также можно задействовать для реализации терминала с табами или, например, управления окнами файлового менеджера.



Surf + tabbed, три открытых вкладки

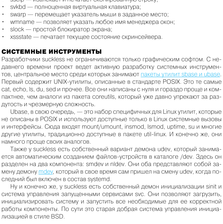
Surf поддерживает скриптинг, как с помощью традиционных языков программирования, так и с помощью полиязычности JS/CSS. На сайте suckless можно найти множество самых разных скриптов, в том числе с реализацией можности блокирования рекламы, закладок, истории, поиска, автоплея на сайтах, навигации по сайтам с клавиатуры и многого другого.

Fun fact

Весь софт, реализованный в рамках проекта suckless, не имеет динамической зависимости от каких-либо сторонних библиотек. Внутри применяется здесь чистый язык C и исходник config.h и перекомпилируй приложение.

ТЕРМИНАЛ, БЛОКИРАТОР ЭКРАНА И УТИЛИТЫ

Набор приложений от проекта suckless.org не ограничен, если бы у них не было собственной реализации эмулятора терминала и утилит для управления сеансами и сетью. Эмулятор терминала st-ru — это самый лёгкий и особенно не отличается от любого другого эмулятора: поддержка 256 цветов, UTF-8, юникодные и эстаки, сложенные цветны, эмуляция VT100. Все как у людей, с тем исключением, что st в десятки раз компактнее и быстрее любого другого эмулятора терминала.



Dwm, Firefox и два окна st

Среди инструментов командной строки можно найти следующие:

- tw — выводит на экран значения всех строк окон;
- sprop — выводит информацию об окне;
- slock — является генератор скопированных строку;
- sdwm — полноценная виртуальная клавиатура;
- swarp — перемещает указатель мыши в заданное место;
- wmtrace — позволяет отслеживать обмен сообщениями окон;
- slock — простой блокиратор экрана;
- xssstate — генератор состояния системного скринсейвера.

СИСТЕМНЫЕ ИНСТРУМЕНТЫ

Разработчики suckless не ограничивается только графическим софтом. С недавнего времени проект ведет активную разработку системных инструментов, центральное место среди которых занимают пакеты утилит sbase и ubase. Первый создается UNIX-утилиты, аналогичные в стандарте POSIX. Это те самые cat, echo, ls, du, sed и прочие. Все они написаны с нуля и гораздо проще и компактнее, чем аналоги из пакета coreutils, который уже давно упрощают за разработчиков и переносимую стоимость.

Ubase, в свою очередь, — это набор специфичных для Linux утилит, которые не отнесешь к POSIX и которые тоже пишутся в стиле suckless: mount, umount, mkfs, mkswap, uptime, su и многие другие утилиты, традиционно доступные в пакете util-linux. И конечно же, ещё намного проще своих аналогов.

Также у suckless есть собственный вариант демона udev, который заменяет системное сообщение файлов устройств в каталоге /dev. Здесь он создается на базе ранее описанного udev и nikos. Он на лету представляет собой меню demoxy mdev, который в свое время сам пришел на смену udev, когда последний был включен в состав systemd.

Ну и конечно же, у suckless есть собственный демон инициализации sinit и система управления запущенными сервисами tsc. Они позволяют запускать, инициализировать системы и запустить все необходимые для ее корректной работы компоненты. По этому это старая добрая система управления инициализацией в стиле BSD.

STALI

Собирается все перечисленное выше в одну кучу, добавляем вдоль и получаем минимальный и очень простой дистрибутив Linux под названием stali, или Static Linux. Такая создания такой системы обсуждалась в сообществе suckless много лет, но только недавно у проекта появились все необходимые компоненты для сборки пакета в единое целое.

Stali — довольно необычный дистрибутив, и не только потому, что базируется на простых и быстрых компонентах и пропагандирует сверхмалое простоту и минимальностность, но и благодаря многим другим чертам. Так, stali полностью игнорирует стандарт файловой иерархии FHS: здесь нет каталога /usr и всего его подкаталога, нет каталога /dev, но есть есть каталог /sys, а который предлагается сделать весь «тонкой», тяжеловесный и раздутый софт. Также в stali нет памятного менеджера. Устанавливается и обновляется дистрибутив с помощью Git, в дополнительных пакетах, куда по всему, не предусмотрено.

Однако самое важное: stali использует статическую компиляцию для всего предустановленного софта. Другими словами, здесь нет динамических разделяемых библиотек, каждое приложение и каждый инструмент самодостаточны и не зависят на от чего. В этом одновременно и преимущество, и недостатки дистрибутива. Статическая компиляция позволяет решить многие проблемы с безопасностью, с разобщенностью версий зависимых библиотек, сделать управление приложениями более простым. С другой стороны, она же создает большой объем, так как загруженные в память приложения, использующие одну и ту же библиотеку, будут разделяться не в памяти ввиста, что, чтобы использовать один экземпляр библиотеки. Впрочем, начет недостатков у разработчиков есть контраргументы, да и, если честно, какой конкретно софт планируется поставить в составе дистрибутива, проблема сведется его почти на нулевой.

Тем не менее сам дистрибутив находится в начальной стадии развития. Подготовлен ISO-образ сего, включающий в себя системные инструменты и утилиты, а также утилиты для управления дисками. Для установки системы. Однако как таковой ОС еще нет. Планируется и включение в дистрибутив dwm, dmenu, st, surf и другие графические приложения ещё на интегрированы в него, а установить сам дистрибутив без использования другой Linux-системы невозможно.

```
[SOLDIER 0.04 ETCD Copyright (C) 1999-2005 M. Peter Hawkins et al
Build to access perfect mir (MCB c001007 in 0)

Stali login: root
Password:
# id
id:0:root:gid:0:root:groups:0:root:1,0:root:2,0:daemon:3,0:tty:5,0:adm:16,0:
# ls -l /usr
ls
lrwxrwxrwx 1 root root    7 Mar 29 09:13 bin -> bin
# ls -l
ls -l
drwxr-x--x    2 root    root        4096 Apr 16, 09:46 bin
drwxr-x--x    2 root    root        1100 Apr 23 07:35 boot
drwxr-x--x    2 root    root        4096 Apr 23 07:35 dev
drwxr-x--x    2 root    root         300 Dec 09 11:13 etc
drwxr-x--x    2 root    root         140 Jan 05 11:39 home
lrwxrwxrwx    1 root    root           3 Mar 29 09:13 lib -> lib
drwxrwxrwx    1 root    root        1024 Dec 31 1969 mnt
drwxr-x--x    2 root    root        4096 Jan 05 11:39 opt
drwxr-x--x    2 root    root        2100 Apr 09 10:15 share
drwxr-x--x    2 root    root        4096 Apr 09 10:15 share
drwxr-x--x    2 root    root        4096 Apr 23 10:27 tmp
drwxr-x--x    2 root    root        4096 Jan 21 2005 var
lrwxrwxrwx    1 root    root           7 Mar 29 07:04 usr -> .
```

Stali сразу после загрузки.

Fun fact

Сайт suckless.org создан с использованием фреймворка sassy, который написан на языке конкурентного интерпретатора Io и Plan 9. Он позволяет сделать несколько простейших скриптов, которые собирают кучи HTML кодов и выдают для публикации веб-страницы. Ядро него насчитывает всего 150 строк кода.

ВЫВОДЫ

Во многом suckless — это, конечно же, проект какого Proof of concept, и примеров об этом его создателе говорит в своем manifesto. С другой стороны, эти ребята написали множество приложений, которыми ежедневно пользуются тысячи человек по всему миру. Автор данных строк сам оценил удобство dmenu, st, surf и уже не может представить себе жизни без этого софта.

Suckless доказал быть интересен любому программисту, ведь это проверил того, как можно писать красивый и эффективный софт, не выходя за границы всего несколько тысяч строк кода. Как говорят ребята из проекта: «Гениальность — простые вещи». Гениальное программное обеспечение — просто. Просто — это сердце философии UNIX. Чем больше строк кода вы удалили, тем дальше вы продвинулись. Чем меньше количество строк кода остается в вашем программном обеспечении, тем более опытным вы становитесь и тем меньшим клиентов будет вашем ПО».

UNIXOID

Изоляция

ВЫБИРАЕМ ПРОСТОЕ И ЫСТРОЕ РЕШЕНИЕ ДЛЯ ИЗОЛЯЦИИ ПРИЛОЖЕНИЙ

Дмитрий Чистик
chistik@linux.org.ua

Наверняка хоть раз в жизни тебе приходилось иметь дело с не вызывающим доверия приложением и скриптами, которые могли нанарушить систему. Или ты хотел запустить браузер на максимально безопасном уровне, чтобы тебя в случае его взлома твой system ничто не угрожало. Сегодня подобные задачи приняли решать с помощью виртуализации. Однако есть гораздо масса гораздо более простых и удобных инструментов для быстрого запуска приложений в песочнице.

ВВЕСТО ВВЕДЕНИЯ

Linux-ядро 2.6.24 Linux Kernel Docker зарождалась в готовом его создателе, пока занимается DotCloud, Linux Containers. Он был основан на все тех же технологии chroot, но с добавлением к ней изоляции процессов, сетевых интерфейсов, виртуальных файловых систем на себе операции изолированных пространств, контейнер для запуска сервисов или небольшими приложений. Однако LXC не был суперсовместимым с новыми технологиями и не обладал фишкой ней Docker кроме сложной файловой системы, возможность быстро развернуть и запустить на любом поддерживающем Docker дистрибутивы сервисы.

Навсего раньше во FreeBSD появилась технология jail, позволяющая создавать песочницы, подобные chroot, но с защитой от бэгов глубокой изоляции, которая и стала прообразом для Linux-контейнеров. Но jail, в отличие от chroot, не обеспечивает изоляции процессов. Сегодня сегодня уже не может обеспечить хорошо либисто и управление ресурсами, которые предоставляет LXC и Docker, что и является ещё одним недостатком обоих. Сегодня ситуация в Linux можно создавать множеством разных способов. Это и уже упомянутый LXC и Docker и libvirt, и systemd-nspawn, это механизм systemd, это пользовательские Отличие для изоляции вендора и платформ, это технологии SELinux Apparmor, позволяющие тонко настраивать доступ приложения к чему ты не было. В этой статье мы познакомимся с самыми удобными для обычного пользователя приложениями, которые могут запускаться на минимальном ресурсе. Чтобы твоего system не пострадал.

запуск не вызывающего доверия приложения, способного нанарушить system;

изоляция браузера, email-клиента и другие приложения, чтобы не взлом на них и угрожал system;

запуск «кривоватых» приложений, которые не должны оставлять следы в system.

MBOX

Начнем с самых простых решений, MBOX — не совсем стандартное исторраит изоляции, он не является полноценным запущенного приложения, не выполняет изоляцию процессов, но в нем может какие-либо набедит. Единственная задача Mbox — сделать так, чтобы приложение не смогло нанести вреда system. Mbox использует для изоляции strace, seccomp и ptrace. Мы, как уже упоминали выше, не сможем запустить приложение, которое отлично ему не было. В этой статье мы познакомимся с самыми удобными для обычного пользователя приложениями, которые могут запускаться на минимальном ресурсе. Чтобы твоего system не пострадал.

запуск не вызывающего доверия приложения, способного нанарушить system;

изоляция браузера, email-клиента и другие приложения, чтобы не взлом на них и угрожал system;

запуск «кривоватых» приложений, которые не должны оставлять следы в system.

MBOX

Начнем с самых простых решений, MBOX — не совсем стандартное исторраит изоляции, он не является полноценным запущенного приложения, не выполняет изоляцию процессов, но в нем может какие-либо набедит. Единственная задача Mbox — сделать так, чтобы приложение не смогло нанести вреда system. Mbox использует для изоляции strace, seccomp и ptrace. Мы, как уже упоминали выше, не сможем запустить приложение, которое отлично ему не было. В этой статье мы познакомимся с самыми удобными для обычного пользователя приложениями, которые могут запускаться на минимальном ресурсе. Чтобы твоего system не пострадал.

```
$ mbox -- wget google.com
...
Network Summary:
* [12270] [127.0.0.1:53] [UDP] mbox/wget-mb64.dec
* [12270] [127.0.0.1:53] [UDP] mbox/wget-mb64.dec
* [12270] [127.0.0.1:53] [UDP] mbox/wget-mb64.dec
* [12270] [173.194.82.51:80] [TCP] mbox/wget-mb64.dec
* [12270] [173.194.82.51:80] [TCP] mbox/wget-mb64.dec
Sandbox Root:
/.tmp/sandbox-12270
/ - R/Tmp/sandbox.tmp
[c]omment, [j]ournal, [d]iff, [l]ist, [s]hell, [q]uit ?>
```

В данном случае под управлением Mbox запускается Wget. Mbox работает сообщает нам, что Wget обращается к адресу 173.194.82.51 и порту 80, а он использует TCP. Также Mbox сообщает, что Wget использует DNS-сервер 127.0.0.1 (для этого нужно запустить dnscrypt-proxy). Из приведенного выше вывода, другие операции или вообще завершить приложение. Проверять, как это выглядит, можно, просто установив уже готовый пакет Mbox. В Debian/Ubuntu это делается так:

```
$ wget http://deb.ciuki.net.mbox/mbox-latest-amd64.deb
$ sudo dpkg -i mbox-latest-amd64.deb
```

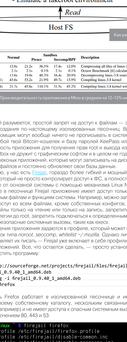
В Arch Linux Mbox доступен в AUR, поэтому установить его еще проще:

```
$ yaourt -S mbox-git
```

Это все. Теперь можно запускать любые бинарники, не беспокоясь, что они оставят в файловой системе хлам. Для этого есть несколько интересных опций, позволяющих контролировать доступ к определенным частям файловой системы, можно исключить определенные файлы. Это отличный тестовый файлы, в котором перечислены разрешенные и запрещенные на доступ каталоги. Например, следующий пример запускает приложение, которому может нужно запускаться на минимальном ресурсе.

```
$ mkdir sandbox_home
$ touch sandbox.rules
$ vim sandbox.rules
allow: .
allow: /usr/bin/firefox
$ mbox -p sandbox.rules firefox
```

Еще одна полезная опция — -d. Она полностью запрещает приложению доступ в интернет.



Task	Normal	Firejail	Firejail	Firejail	Firejail	Description
Firefox	1.1s	1.1s	0.9%	1.1s	0.9%	Computing all files of Fires 1.0
Octave	17.6s	17.6s	0.0%	17.6s	0.0%	Octave Benchmark [0] calculating matrix
Gnuplot	1.1s	1.1s	0.0%	1.1s	0.0%	Gnuplot Benchmark [0] 1 plot
Kernel Build	1m 1.0s	1m 1.0s	0.0%	1m 1.0s	0.0%	Compiling Linux 3.8 kernel
Build Linux Lib	21.5s	49.6s	130.7%	35.7s	66.2%	Compiling Linux 3.8 kernel with 4 parallel jobs

Производительность приложения в Mbox в среднем на 32-73% ниже обычной

FIREJAIL

Следующий решением, простой запрет на доступ к файлам — это слишком мало для создания по-настоящему изолированных песочниц. Вредоносное ПО или взломанное может вообще начнет не прикладать к system, а просто начнет с собой такой Bitcoin-кошелек и бах перелет KeePass или activists-кошелек. Поэтому необходимо ограничивать доступ к ресурсам и по-полному. К тому же Mbox не дружит с графическим софтом и в целом не годится для запуска клиентских приложений, которые могут запускаться на минимальном ресурсе. Firejail и его расширения обходят свои баги.

К счастью, на помощь приходит Firejail. Более гибкий и мощный инструмент изоляции, который не просто контролирует доступ к ФС, а полностью отрезает от остальной системы. Firejail использует Linux namespaces, seccomp-bpf и Linux capabilities. Firejail предоставляет уже готовые профили ограничений для большинства самых распространенных приложений. Например, можно запустить firefox, firejail firefox, firejail --dns=8.8.8.8 firefox --no-remote и firejail --private=/home/user/dir firefox --no-remote. Firejail — это, пожалуй, лучший из всех, запустить подозрительное приложение, которое может нанести вред system, при этом не потеряв ресурсы, в целом,

ПЕРЕНОС ПО ЗАКОНУ

КАК БЕЗОПАСНО ПЕРЕНЕСТИ ПЕРСОНАЛЬНЫЕ ДАННЫЕ В ВИРТУАЛИЗИРОВАННУЮ СРЕДУ?

Владимир Любецкий
Руководитель
бизнеса Stack Group

Любой бизнес стремится тратить как можно меньше, в том числе на ИТ. Один из способов сэкономить — перенести информационные системы в облако. При переносе данных в виртуализированную среду надо учесть требования законодательства о хранении и обработке персональных данных. Эти требования касаются любой организации.

ТРЕБОВАНИЯ ЗАКОНОДАТЕЛЬСТВА

В 2006 году был принят 152-ФЗ «О персональных данных», который требует защитить физические лица при автоматизированной обработке персональных данных. В прошлом году вступил в силу пакет поправок о локализации персональных данных на территории России, что, по мнению авторов, должно повлиять уровень информационной безопасности внутри государства и простимулировать российский рынок технологических решений и рынок информационной безопасности.

По закону бизнес обязан собирать, хранить и обрабатывать персональные данные на территории РФ. Все требования совершенно одинаковы как для российских, так и для зарубежных компаний, если их деятельность направлена на территорию России. При этом передавать персональные данные за пределы страны можно, но они должны быть неизменяемыми, и их объем не должен превышать объем в российских базах данных.

ДЛЯ КОГО ЗАКОН?

Каждая организация выступает как оператор персональных данных, например свои сотрудников, клиентов, партнеров, поэтому под закон попадают все компании во всех отраслях. Однако потенциальные проверки в первую очередь грозят компаниям, у которых на обработке персональных данных постоянно комментарии деятельность. Банки, страховым компаниям, компаниям, оказывающим медицинские услуги, и подобным.

В план проверок Роскомнадзора на 2016 год вошли крупнейшие софтверные компании, международные банки, сотовые торговые компании и интернет-магазины.

СЛОЖНОСТИ ПЕРЕНОСА ПЕРСОНАЛЬНЫХ ДАННЫХ ДЛЯ МЕЖДУНАРОДНЫХ КОМПАНИЙ

Зачастую глобальные международные игроки, чтобы выполнить требования законодательства, вынуждены переносить в Россию не только свои данные, но и все сопутствующие системы: компаниям с централизованными ИТ-системами крайне сложно изменить архитектуру информационной системы под одно локальное требование. Поэтому такие компании и чей не было разнесенных центров обработки информации. Чтобы перестроить информационные системы, требуются существенные инвестиции.

ИНТРУДЕРЫ

Переносить в облачную среду менее затратно, чем покупать и монтировать оборудование. В конце 2014 года цены на российские облака были ниже на 15–30% выше, чем на европейские, а в конце 2015 года, наоборот, новые цены стали на 20–30% ниже, изменился жесткий курс и относительная стоимость размещения в российских дата-центрах.

Операторы облачных провайдеров более зрелый продукт, и зарубежные компании получили возможность выбирать среди систем виртуализации, ценовых сегментов, появились возможности для связанности российских и западных инфраструктур.

Для иностранных компаний виртуальная инфраструктура — надежнее и потоковее решения. Многие компании с распределенной инфраструктурой переходят в виртуализированную среду, в которой также можно разместить необходимые средства защиты персональных данных. При переносе иностранные компании, разработчики, оказывают от оператора выполнения требований регулятора по защите информации.

РИСКИ, ВОЗНИКАЮЩИЕ ПРИ МИГРАЦИИ СИСТЕМ ХРАНЕНИЯ ДАННЫХ

Сами миграционные процессы с одной системы на другую несут в себе риски. В первую очередь — может быть нарушен установленный уровень обслуживания клиентов. Может также прерваться связь и приостановиться работа подразделений, для которых эффективность выполнения поставленных задачи могут возникнуть условия, когда требования информационной безопасности соблюсти невозможно, и другие технические проблемы. Поэтому важно, чтобы провайдеры были готовы обеспечить непрерывность бизнеса, процессов, использует надежные и проверенные временем решения для переноса информационных систем.

ЭТАПЫ МИГРАЦИИ В ОБЛАКО

Общая стратегия миграции проста, то есть перенося операционных систем, отвечающих за работу сервера, в виртуализированную среду, рассмотрим на примере решения VMware vSphere.

Архитектуру информационных систем компании желательно разделить на выполняющие конкретные задачи серверы. Обычно в облако перемещается всего, кроме приложения, которое написано с cloud-технологиями по технологии виртуализации (RISC-архитектура) и тех, что нельзя переместить из-за лицензии.

Далее необходим аудит информационных систем. На этом этапе определяется состав серверов (какие ОС относятся к тому или иному серверу), а также их связность. Главная сложность заключается в многобразии исходных ОС и фиксированной архитектуры данных, которая разнообразна. Для основе этой информации составляется план миграции с учетом текущих бизнес-процессов, определяются требования к ресурсам. Поскольку рассматривается инфраструктур, порядок миграции, задаются доступные «окна миграции». Важно помнить, что нельзя во время миграции обновлять версии программных продуктов или операционных систем. Для применения с минимальной затратой только перенести вычислительных ресурсов (CPU, RAM, HDD).

Как правило, для миграции используется утилита VMware Converter, которая эффективно работает при переносе ОС семейства Microsoft Windows (но у миграции работающих в этих ОС служб есть свои нюансы). А вот из-за особенностей файловых систем, которые создаются в 40% случаев после переноса ресурсы VMware converter виртуальная машина может не запуститься. Если в Linux используется LVM, то надо развернуть в исходной среде новый экземпляр ОС из шаблона и перенести данные, с помощью программных продуктов и внутренние службы.

Для надежности ОС есть общее условие, затрудняющее миграцию: во-первых, способ хранения данных, из-за которого прямая миграция невозможна, — это динамические диски в Windows или LVM в Linux, в во-вторых, сложности из-за установленных программ и антивирусных сервисов RAID. Так, даже точечный перенос данных сам по себе не гарантирует, что виртуальная машина успешно запустится. На физическом сервере работу виртуальных машин обеспечивает гипервизор — ОС, развертывающая физический сервер на несколько виртуальных машин, которые могут работать одновременно и использовать один и то же физическое оборудование. Гипервизор отвечает за распределение ресурсов у гипервизора не совпадает с оборудованием физического сервера, на котором работала ОС до миграции. Соответственно, из-за разницы драйверов возникает множество проблем, связанных с этому оборудованию.

МИГРАЦИЯ ADDS И MS SQL ВЕЗ ОСТАНОВКИ СЕРВИСОВ

Практически всегда для бизнеса необходимо, чтобы ряд серверов оставался доступным в ходе миграции. При этом зачастую миграция без остановки сервиса рекомендуется как наиболее предпочтительная. Поэтому рассмотрим особенности миграции без остановки наиболее популярных служб ОС Microsoft: Active Directory Domain Services (ADDS или AD) и Microsoft SQL (MS SQL). Для миграции Active Directory без остановки сервиса применяется с миграцией контроллеров: Формируется сетевая связность между физическим оборудованием и виртуализированной средой. Как правило, это site-to-site VPN — он создает логическую сеть поверх другой сети. При этом трафик можно дополнительно шифрованным по протоколам IPsec.

В облаке развертывается новая виртуальная машина из шаблона, где настраиваем контроллер домена AD с добавлением их в пес.

Базу данных Active Directory реплицируем по сети через VPN с работающих контроллеров на сторону физического контроллера в облачном.

После репликации данных переназначаем мастеры ролей операций на облачные контроллеры и убираем роли контроллеров домена с серверов.

Затем переносим работу серверов на облачные учетные записи сервер контроллеров и физическое оборудование.

Алгоритм миграции MS SQL более сложен, так как MS SQL обычно применяется в многоуровневом сервисе в качестве бэкенда. В записях DNS в приложениях, использующих базы данных в клиентах MS SQL), приходится вручную указывать новое расположение базы данных. Поэтому совсем исключить простой нельзя, но его можно свести к минимуму. Существует механизм и без остановочной миграции MS SQL, в нем используется Mirroring и AlwaysOn, но его применение не всегда оправдано. AlwaysOn доступен только в дорогих редакциях Enterprise, а Mirroring должен поддерживаться клиентами MS SQL. К тому же для применения механизмов Mirroring нужна дополнительная настройка всех клиентов MS SQL.

Рассмотрим наиболее часто встречающийся вариант миграции MS SQL в облако:

Настраивается сетевая связность между облаком и физическим оборудованием

Убеждаемся, что модель восстановления базы MS SQL полная, тогда можно сделать и переместить полную резервную копию, а затем синхронизировать обе базы данных, применяя транзакционные логи.

В облаке развертывается виртуальную машину из шаблона, в которой устанавливаем и настраиваем новый MS SQL сервер.

Создаем полную резервную копию базы данных MS SQL сервера, работающего на физическом сервере, затем восстанавливаем ее в облачном, при этом способ перемещения резервной копии зависит от размера файла и пропускной способности сети — перемещаем на физическом носителе либо копируем по сети.

После восстановления базы данных в облаке делаем копию транзакционных логов и их также восстанавливаем в облаке.

Во время «окна миграции» останавливаем работающий на физическом оборудовании MS SQL сервер, создаем и восстанавливаем последнюю минимальную по размеру копию транзакционных логов в облаке, запускаем MS SQL сервер в облаке и переключаем клиента на новое месторасположение базы.

Проверяем работу сервисов, отключаем физическое оборудование.

Для каждой службы и сервиса существует множество способов миграции, зависящих от условий. Избежать трудностей, которые могут возникнуть при миграции, поможет сервис-проводник.

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

Компании должны не только перенести данные в Россию, но и обеспечить надлежащий уровень информационной безопасности их хранения.

В перечне документов, которые регулятор требует в описывающих меры, необходимые для защиты информационных систем, на первом месте находится, разумеется, сам закон, которой устанавливает общие требования. Конкретные действия регламентируют постановления Правительства РФ и нормативными документами ФСТЭК (Федеральная служба по техническому и экспортному контролю) и ФСБ (Федеральная служба безопасности).

ТРЕБОВАНИЯ К ХРАНЕНИЮ ПЕРСОНАЛЬНЫХ ДАННЫХ

Требования к технической защите конфиденциальной информации и предоставлению услуг по защите информации достаточно четко сформулированы. Инструменты их реализации многобразны. В частности, это могут быть сетевые экраны, системы обнаружения вторжений, средства анализа защищенности, антивирусной защиты, средства защиты сред виртуализации. На рынке представлен широкий спектр средств защиты информации — и российских, и зарубежных вендоров. Есть уже правоприменительная практика, так как закон действует с 2007 года. В целом подход к регулированию в России отличается от, например, европейского подхода. Так, в России нормативные предписанных требований по информационной безопасности ведет к возникновению ответственности. А в Западе компании может самостоятельно определять, какие способы выполнить требования, и ответственность наступает, только если совершены неправомерные действия с персональными данными.

ТРЕБОВАНИЯ К ИНФРАСТРУКТУРЕ

Определенные требования регуляторы предъявляют и к инфраструктуре. Например, существуют аттестация по стрронних автоматизированным системе, обеспечивающую защиту информации. Аттестат выдается независимой лаборатурой, которая подтверждает, что инфраструктура готова и соответствует всем требованиям по защищенности информационных систем. Этот аттестат необязателен для коммерческих организаций, но дает возможность по честь, что дата-центр имеет необходимую защиту и способен обеспечить безопасность данных компании.

Также есть международные стандарты ISO, регламентирующие построение системы управления информационной безопасностью комплекс стандартов ISO 27000х. Многие иностранные компании соответствуют этим стандартам.

Кроме того, применяется модель аутсорсинга по защите данных. Компании, предоставляющие такие услуги, должны иметь необходимые лицензии и компетенции.

ПРОГНОЗ НА ПЕРЕНОС

Довольно сложно подсчитать, сколько точно данных подлежит переносу в Россию, но, исходя из заполняемости рынка ЦОД, можно сказать, что свободных мощностей всего около 27 тысяч стоек, и почти 40% из них свободны. Многие дата-центры имеют площадки высокой степени готовности. Нужно учесть и что плотность данных в одной стойке может различаться в зависимости от оборудования. Сегодня один юнит серверной стойки обрабатывает значительно больше информации, чем несколько лет назад.

При этом спрос неравномерен: год назад был достаточно сильный подъем, сейчас спрос на размещение от зарубежных компаний снизился. Часть компаний уже переехала в аттестуют требуется время, а часть ждет, например результатов проверок регуляторов (они появятся на формирование второй волны).

WWW

Исходники рекламоблока для Catchy32 можно скачать здесь

Перенесть поиск: данные Tableau, Win32 Fonts и Tyrolese от Trend Micro

INFO

В Китае полная копия персональных данных должна храниться на территории страны, а любые трансграничные вообще запрещено передавать за все пределы.

WWW

Подробная техническая информация по продуктам VMware

Миграция ADDS

Обзор технологии AlwaysON

Настройка репликации для AlwaysON

→

VERSUS

Мария Ковалевская
Технический Директор

НОВОЕ В MS SQL 2016

Современный мир — это мир данных и мир сетей. Компании учатся обрабатывают все больше информации, которая используется и для работы с клиентами, и для анализа и принятия решений. Данные хранятся уже не только локально, для них нужны пользователям удаленный доступ. В этого современные СУБД, изначально разработанные для другого целей, постепенно начинают обрастать всевозможными функциями. Посмотрим, что нового в последнем релизе MS SQL Server 2016.

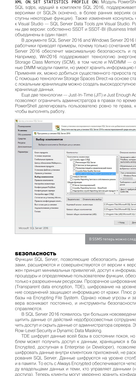
ЗНАКОМИМСЯ С MS SQL 2016

Новая версия разрабатывалась почти шесть лет. Не успели выпустить версию 2014 и начать в Microsoft работать над новой версией, а мне 2015-го, на конференции грянь был представлен очередной релиз, SQL Server 2016 Community Technology Preview. Сейчас уже доступна версия Release Candidate 0 год назад. Релиз ожидается в 2016 году. Microsoft в направлении общей идеологии SQL Server и Azure SQL Database, что дает возможность использовать одинаковые инструменты для разработки и работы с данными (Data Management, Analytics, Business Intelligence).

Вместе с релизом SQL Server 2016 MS представила агрессивную программу по переходу СУБД Microsoft на Linux. Сначала поддержка Linux (что в MS), включающая «бесплатную» поддержку сделать новых будет за подписку, а потом, в ближайшем будущем, Linux будет полностью поддерживаться. Мы постоянно SQL с нами СУБД от Microsoft обходится на порядок дешевле. Предлагается для ведения: сдав основные на вычислительных мощности (Enterprise edition), и на основные пользовательские функции (Enterprise Client Access License). Отказ работает под Linux, и здесь у MS плюс есть возможность использовать свои наработки, которые были в Windows, и которые уже с 2014 версии не изменялись. Максимальный объем данных для версии 32 Ибайт, у Express — 10 Тбайт. Максимальный объем использованной памяти на экземпляр Express — 1 Гбайт. Standard — 128 Гбайт. Для версии Express нет нужно возможностей ОС. Максимальный количество ядер: Express — до четырех, Standard — до 24.

Мы постоянно SQL с нами предоставляет несколько версий: Express (она есть выглядит ВМ) и ВМ+ВМ. Возможна установка SQL Server на все ей предоставляется. Express — бесплатная версия, ограниченная по объему данных, но отлично подходит для разработки.

С нашего релиза SQL Server Management Studio (SSMS) поставляется отдельно, а сам он теперь управляет новые версиями. Поддерживается версия предыдущей версии SQL 2016 под Linux. То есть, теперь не проедет версия SQL Server 2016 Express, но с данными версиями СУБД. Установка SSMS на основе на ВИТОР-1 «Вестник».



В SSMS поменялось несколько полезных изменений. Например, Live Query Statistics позволяет отслеживать выполнение запроса в реальном времени, даже сервер, например, Linux, запущен, выполнение запроса отслеживается для оптимизации. Активируется через пункт меню или кнопки Live Query Statistics, в окном данных была включена обед. Также в SSMS теперь **SET STATISTICS IN, ON SET STATISTICS PROFILE ON**. Между Power/Shell для работы с SQL Server 2016 и Azure SQL Database. Для Azure SQL Database теперь доступно обновление до SQL Server, а Server версии сервера будет новых настройки. Также изменения коснулись набора дополнений Visual Studio. Data Tools для Visual Studio 2015 теперь. Больше Data и другие все версии, что поддерживает SSDT и SSDT BI (Business Intelligence), теперь они доступны.

В дополнение SQL Server 2016 теперь поддерживает Stretch Database, для работающих приложений, почему только сначала MS Server 2016 и SQL Azure. Stretch Database — это, по сути, «бесконечный» объем для хранения данных, которые, фактически, представляют технологии энергосберегающего хранения Storage Class Memory (SCM), а так же и NVDIMM — они занимают обычные DRAM модули и могут использовать их для ускорения. Поддерживается Windows. Применяя их, можно добиться существенного прироста производительности. Для ускорения работы с большими объемами данных, которые, фактически, представляют технологии энергосберегающего хранения Storage Class Memory (SCM), а так же и NVDIMM — они занимают обычные DRAM модули и могут использовать их для ускорения. Поддерживается Windows. Применяя их, можно добиться существенного прироста производительности. Для ускорения работы с большими объемами данных, которые, фактически, представляют технологии энергосберегающего хранения.

Еще две технологии — Just-In-Time (JIT) и Fast Encrypt Administration (JEA) — позволяют отправлять администратора в правы по времени или при точном доступе. Технологии JIT позволяется разрешить доступ на время, а при окончании, чтобы доступ работу.

БЕЗОПАСНОСТЬ

Начиная SQL Server, позволяющие обеспечить данные и формы с уровнем доступа к конкретным строкам по запросу. Ограничение доступа к данным при прежде начинающих правилей, доступ к информации через конкретные строки. Ограничение доступа к данным при прежде начинающих правилей, доступ к информации через конкретные строки. Ограничение шифрования данных и маскирование данных

Алексей Зимнов
Пономарев
Zeimnov@gmail.com

ОТВЕТЫ НА ВОПРОСЫ ЧИТАТЕЛЕЙ

(ЕСТЬ ВОПРОСЫ? ШЛИ НА FAQ@CLC.RU)

ВЫБИРАЕМ LAN TESTER: КАКИЕ ВЫБЮЗ ЛАНТЕСТЕРЫ И КАК ПРИГОЖАЮТСЯ В РАБОТЕ СИСАДМИНА

Дубаю, любой специадист, кто занимается обжимал кабель, представляет себе, что такое лантестер. Для остальных поясню. Лантестер — это прибор для тестирования витой пары, которой по каждой жиле прозонит сигнал и подтверждает, что линия целая. Для чего это показывает, что все пары обжаты в правильной последовательности и нет перекрещенных пар. Это может любой недоучка, но есть и красивые другие особенностей.

Перечисленные действия, тем не менее, самые необходимые. Обжал кабель, проверил и вперед выдешь всю карту. Здесь обжимал не прожала контакт, тут повреждена жила, а вот тут перепутал цвета. А вот когда нужно найти, какая розетка соединяется-на-пати-панели, добрую службу сослужит дополнительный блок приемника для лантестера.

Получается, что они работают в паре. Лантестер вызутвует мастером и гонит сигнал по жилам, а слейв, в свою очередь, принимает поступающий сигнал. Так можно задокумментировать все неизвестные розетки в шкафе и разобраться, что куда ведет. Не говоря о том, что иногда конца кабелеей свисает сгуда-то с потолка и ничто не знает, к какой конкр задания они тянутся. Лантестер в таких случаях совершенно незаменим!

Еще лантестеры делятся на то, что можно включать в активную сеть, и то, что нельзя. Вторые более распространены и стоят на порядок дешевле. Эксперимент с включением такого девайса в сеть, где на другой стороне находится активное оборудование, обычно кончается выходом из строя лантестера. Так что будь внимателен.

Серьезные аппараты порой стоят как хороший ноутбук и умеют весьма и весьма многое. Вот список тех вещей, которые может делать аппарат профессионального уровня.

- Обнаружение ошибок в схеме разводки, замыканий, обрывов, кросс-соединений.
- Тестирование телефонных и коаксиальных кабелей, а также кабелей на основе витых пар категории 5, 5е и 6.
- Тестирование локальной сети и соединительных шнуров типа патч-корд (RJ45/11/12 FTP и BNC).
- Тестирование правильности соединения каждого провода в кабеле.
- Проверка правильности обжима, наличия обрыва или хорошего замыкания.
- Проверка целостности экрана для экранированной витой пары.
- Определение расстояния до обрыва кабеля.
- Прибор проверяет, обеспечивает ли существующий кабель достаточную пропускную способность для поддержки передачи голоса, стандартов 10/100 Ethernet, VoIP или Gigabit Ethernet.
- Устранение неисправностей — отображение причины, по которым существующий кабель не подходит: выявляет требуемую пропускную способность сети (например, перекрестные помехи на дальнем кабеле свыше 11 м).
- Обнаруживает — прибор определяет, что находится на конце каждого кабеля, и отображает сведения о конфигурации устройства (скорость/дуплекс/пары).
- Определение неисполнительных портов коммутатора, средств обнаружения ошибок.
- Составление карт конфигурации проводки и отображение расстояния до мест возникновения неисправностей с помощью теста Intelligent Wiremap.
- Проверка всех типов кабелей с медными жилами: витой пары, коаксиального кабеля и аудиокабеля.

Все эти функции значительно упрощают поиск и диагностику неисправностей в локальной сети. Если, скажем, дома или в небольшом офисе все это вряд ли пригодится, то для серьезной работы — однозначный мастхэв.

ДЕЛАЕМ СЕБЕ НОМЕР В ЕВРОПЕ ИЛИ США ДЛЯ ПОДТВЕРЖДЕНИЯ РЕГИСТРАЦИИ НА САЙТАХ

Иногда для завершения регистрации на сайте нужно подтвердить, что вы живете в США или в какой-то другой определенной стране. Магазины и платежные системы определяют страну по кредитке (это отдельный разговор), другие сервисы вместо этого присылают проверочный код на СМС. Единственный вариант — иметь телефон с кодом нужной страны. Для обхода таких ограничений существует услуга аренды номеров.

Услуга эта стоит денег, но так нужна уж большая, а если поискать, то можно найти и бесплатный сервис. Есть, правда, важный нюанс: регистрация и последующая привязка бесплатного номера к счету PayPal может привести к печальным финансовым последствиям. Например, владелец ресурса решит, что тебе пора поделиться с чьим-своим сообщением. Скорее, как известно, платит дважды, а в особо неудачных случаях множится может быть и более беспощадным.

Из платных сервисов могу порекомендовать proxyl.com. Аренда номера в США на месяц будет 1.50 евро + 0.50 евро ежемесяц, пара бутылок пива. Бесплатных тоже есть, такие на sms-reg.com — ещё один надоедной сервис. Из бесплатных есть google voice, сойдет, чтобы поиграться. Если совсем не хочется платить, можешь попробовать поклянчить на форумах, но твои шансы нарваться на развод возрастут многократно.

РАЗБИРАЕМСЯ В ОШИБКАХ НА СЕТЕВОМ ИНТЕРФЕЙСЕ. КАРЛИКИ, ГИГАНТЫ И ПРОЧАЯ ЕРУНДА

Начинающий сисадмин рано или поздно столкнется с какой-нибудь консолью сетевого оборудования. Это может быть коммутатор SCOM, маршрутизатор Cisco или иной зверь вроде Mikrotik или D-Link. Так или иначе, на глаза ему попадутся строки, напоминающие загадочное фэнтези про карликов и гигантов. В этот момент неокрепший мозг обычно отказывается в ужасе воспринимать окружающий мир. Попробую тебя немного подготовить, чтобы этого не произошло.

Пример отображения информации на определенном интерфейсе

Вот список наиболее часто встречающихся ошибок.

- Runts, или «карлики». Фреймы, которые не отвечают требованию к минимальному размеру. Минимум должно быть 64 байта, включая 18 байт для полей MAC-адреса получателя, MAC-адреса отправителя, типа и FCS. Такие ошибки могут быть вызваны в том числе коллизиями на интерфейсе.
- Giants, или «гиганты». Фреймы, размер которых превышает максимально допустимый. Это больше 1518 байт, включая 8 байт для полей получателя, отправителя, типа и FCS.
- Input errors — ошибки ввода, сумма значений нескольких счетчиков, включая runts, giants, CRC, frame и другие.
- CRC — полученные фреймы не прошли проверку FCS. Скорее всего, вызвано неполадками на физическом уровне. Повреждена кабеля или порта.
- Frame — получены фреймы с неверным форматом. К примеру, они завершаются нечетным числом.
- Collisions — это счетчик всех коллизий, происходящих при передаче.
- Late collisions — подмножество всех запоздалых коллизий, происходящих после передачи 64-го байта фрейма. Запоздалые коллизии чаще всего указывают на рассогласование дуплекса.

Как видишь, все не так страшно, как кажется на первый взгляд. Главное — поискать первоисточник ошибки и не пугаться новых названий. Здесь все вроде не логично, и, разобравшись, ты легко узнаешь, на какие проблемы указывают сообщения об ошибках.

ЧТО ДЕЛАТЬ, ЕСЛИ СПИСКИ ФАЙЛОВ В ПАПКАХ В ДОМЕНЕ WINDOWS НЕ СОВПАДАЮТ НА РАЗНЫХ КОМПЬЮТЕРАХ

Бывает так, что в домене Windows есть две совершенно одинаковые машины, но одной видно полный список файлов в папке, а на другой не видно вообще ничего. Вот что можно сделать для решения проблемы.

Когда Windows в очередной раз подкидывает тебе такую закавыку, не спеши сомневаться в реальности происходящего. Суть этой проблемы, скорее всего, кроется в системе кеширования метаданных протоколом SMB2. Штуку эту можно выключить через реестр. Правь нужно ключ, значение необходимо выставить в ноль. Если таких ключей нет, то их нужно будет создать.

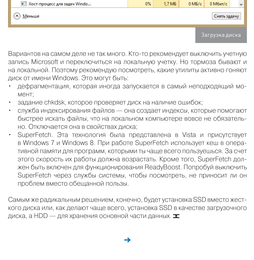
- DirectoryCacheLifetime
- FileNotFoundCacheLifetime
- FileInfoCacheLifetime

Лежат они в разделе: **HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\lanmanworkstation\Parameters**

Если ты все сделал правильно, кеширование будет отключено и файлы, скорее всего, появятся там, где они должны быть. Дополнительно рекомендую изучить официальную справку.

ЧТО ДЕЛАТЬ, ЕСЛИ WINDOWS В НЕЩАДНО ГРУЗИТ ДИСК ПОД 100%

Многие пользователи замечают, что Windows 8 иногда начинает очень сильно загружать диск. Как показывает диспетчер задач, нагрузка держится в районе 80–100%. При этом система бешено тормозит и не дает нормально работать. И это на машинах с Core i5–i7 и 8–16 Гбайт оперативки! Что можно сделать для решения этой проблемы?



Загрузка диска

Вариантов на самом деле не так много. Кто-то рекомендует выключить учетную запись Microsoft и переключиться на локальную учетку. Но тормоза бывают и на локальной. Поэтому рекомендую посмотреть, какие утилиты активно гоняют диск от имени Windows. Это могут быть:

- дефрагментация, которая иногда запускается в самый неподходящий момент;
- задание cmdlet, которое проверяет диск на наличие ошибок;
- служба индексирования — она создает индексы, которые помогают быстро искать файлы, что на локальном компьютере вовсе не обязательно. Отключается она в свойствах диска;
- SuperFetch. Эта технология была представлена в Vista и присутствует в Windows 7 и Windows 8. При работе SuperFetch использует кеш в оперативной памяти для программ, которыми ты чаще всего пользуешься. За счет этого скорость на работы должна возрасти. Кроме того, SuperFetch должен быть включен для функционирования ReadyBoost. Попробуй выключить SuperFetch через службы программ, которыми ты чаще всего пользуешься. Если тот же проблема вместе обещанной выше, отключаем ее.

Самым же радикальным решением, конечно, будет установка SSD вместо жесткого диска или, как делают чаще всего, установка SSD в качестве загрузочного диска, а HDD — для хранения основной части данных.

ХАКЕР

18+

№ 7 (210)

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

В случае возникновения вопросов по качеству печати: claim@glc.ru. Адрес редакции: 115280, Москва, ул. Ленинская Слобода, д. 19, Омега плаза. Издатель: ООО «Эрсиа»: 606400, Нижегородская обл., Балахнинский р-н, г. Балахна, Советская пл., д. 13. Учредитель: ООО «Принтер Эдишионс», 614111, Пермский край, г. Пермь, ул. Яблочкова, д. 26. Зарегистрировано в Федеральной службе по надзору в сфере связи, информационных технологий и массовых коммуникаций (Роскомнадзоре), свидетельство ПИ № ФС77-56756 от 29.01.2014 года. Мнение редакции не обязательно совпадает с мнением авторов. Все материалы в номере предоставляются как информация к размышлению. Лица, использующие данную информацию в противозаконных целях, могут быть привлечены к ответственности. Редакция не несет ответственности за содержание рекламных объявлений в номере. По вопросам лицензирования и получения прав на использование редакционных материалов журнала обращайтесь по адресу: xakep@glc.ru. © Журнал «Хакер», РФ, 2016