



AntiRootkits **Invisible War II**

АнтиРуткиты
Невидимая Война II

EP_XOFF/el666
UG North Division
Весна 2008

Содержание

Введение	3
AKR 2.007	5
DriverDetect	7
DeepMonitor	9
GMER	10
NIAP AntiRootkit Tools	13
RADIX	15
Rootkit Trap	18
RootQUEST	21
SEEM	22
Spyware Processes Detector	23
UnHackMe	27
Приложение 1. В тени Русток	32
Приложение 2. История одного буткита	35
Приложение 3. Пишем платный антируткит	38
Заключение	41
About	42

Введение

Со времени написания предыдущего обзора актуальных средств детектирования и противодействия руткитам прошел почти целый год. За это время много что изменилось вокруг, поэтому мы нашли некоторое время и решили написать продолжение, в котором постараемся максимально объективно рассказать вам об изменениях в лагере антируткитов и о том, что же нас всех ждет дальше. Не волнуйтесь, конца интронета, как это неоднократно предрекала Лаборатория Касперского, не будет. По крайней мере, при нашей с вами жизни это точно не произойдет :)

Данный документ представляет собой обзор актуальных на текущий момент (весна 2008) средств обнаружения и противодействия руткитам и составлен на основе внутренних исследований и тестирований проводимых UG North. Авторы этого обзора ни в коей мере не получают от него никакой финансовой выгоды и не собираются заниматься саморекламой, как считали некоторые читатели Invisible War. Прочем нам глубоко пофигу на мнение этих несознательных личностей, по большей части почему-то имеющих корни с <http://virusinfo.info> или из команды достаточно известных в последнее время благодаря публикациям на <http://rootkit.com> скрипто-кидессов из Unl0ck.net

Мы постарались включить в обзор все наиболее интересные программы, что появились за этот год и те программы, что серьезно продвинулись в своем развитии и о них стоит вновь сказать слово. Программы, которые стары как мир и ограничены функционалом, например, такие как Process Hunter v1.1, детекторы конкретных руткитов – HaxDef, HaxDoor, Rustock.B, окончательно устаревшие концептуальные программы Рутковской (огромное уважение и приветы), программа Питера Сильбермана RAIDE (greet again), откровенно лажовые интегрированные антиспайвары с мнимой антируткит составляющей – AVZ (откровенных плагиаторов мы не любим), в обзоре участия не принимают. Также из обзора исключены те программы, что были подробно освещены ранее и не изменились сколько-нибудь кардинально за этот год, чтобы претендовать на современный уровень обнаружения и удаления руткитов. Бог простит если что.

Авторы решили немного отойти от предыдущей системы оценок, так как посчитали её недостаточно объективно отражающей возможности программ. Больше нет поколения и класса, теперь помимо типа, мы также введем оценку по пятибалльной шкале представляющую собой нашей субъективное мнение об этой программе. Тем не менее, как и прежде в правоте наших слов и оценок вы можете убедиться лично взяв и протестировав каждую программу самостоятельно. Мы не страдаем маркетинговой фигней (вот так и проскальзывает желание сказать немного другое слово в том же контексте, прим. el666) и не занимаемся рекламой рассматриваемых программ, ибо нам по большому счету абсолютно наплевать будете вы их использовать после нашего обзора или нет. И о, боже мой, нет, мы не ставим целью обосрать всех и вся, достаточно просто факта существования некоторых «продуктов».

В качестве тестовых платформ взяты самые обычные машины – Intel Core2 Duo E6550, AMD64 Sempron 3000+, Intel Core2 QUAD, AMD Phenom 9550 и виртуальные машины VMWare и Virtual PC. Операционные системы, на которых мы гоняли эти программы, представлены, разумеется, 32 разрядной линейкой NT, начиная от Windows 2000 SP4 Rollup 1 и заканчивая Windows Server 2008. Антируткиты и руткиты на 64 разрядной линейке пока редкость и большого внимания не заслуживают. Программные конфигурации на всех этих машинах различны, это сделано специально, чтобы получить максимально приближенные к «боевым» результаты прогона программ. Многие программы были впервые протестированы на Windows XP SP3.

Антивирусы и фаерволы с так называемыми «модулями поиска и нейтрализации руткит» и всякие лажовые HIPS типа DefenseWall или ProSecurity нами не рассматриваются, и не будут рассматриваться. Научитесь для начала гадить в унитаз, а не в парадную, товарищи разработчики этих говноподелок.

В качестве затравки применялся широкий спектр различных руткитов, начиная с голых концептов (Unreal.A/B, rkdemo v1.0/1.1/1.2, ZOmBiE v1.0/2.0, phide_ex, BadRkDemo) и заканчивая реальными ITW образцами, такими как Rustock.B, его неуловимый преемник Rustock.C, MAOSBootkit (аплодисменты, дождались), Trojan.Accesso. Кроме этого на некоторых тестовых машинах работал Защитник Хэккеров, и бегала малвара мелкого пошиба, типа спамботов. Мы постарались сделать подборку максимально жесткой для проверяемых антируткитов, потому что не видим смысла в тестировании современных антируткитов с поделками на коленке, перехватывающими кучу функций в User Mode, парочку функций в SSDT или только IRP_MJ_DIRECTORY_CONTROL, пусть даже они все, о господи, ITW.

В виду массы обвинений, цитата «пиписькомеренье» (интересно, а что кто-то пробовал?) в обзор не включен Rootkit Unhooker 4.3 (кроме того, версии старше 4.0 строго internal use only и, стало быть, Одей, а Одеями не размахивают). Собственно Rootkit Unhooker мы не хотели включать и в предыдущий обзор, но в последний момент все же решили добавить до кучи, что называется :)

В заключение хочется сказать, что если предыдущий обзор являлся целиком и полностью моим детищем, то нынешний это плод коллективного труда нескольких человек. Просто в одиночку как-то быстро охуеваешь от всего этого нескончаемого потока говна, льющегося из интронета.

AKR 2.007

Оценка: 0.1 из 5

Тип: комбинированный (детект + удаление)

Статус: релиз

Продукт от абсолютно неизвестной нам французской компании SAFE-PROTECT. После установки радостно требует перезагрузить компьютер, поначалу по непонятно каким соображениям (оказалось для нотификатора на процессы). После перезагрузки программа при старте попросила ввести пароль.



Рисунок 1. Выберите пароль

Практически сразу после ввода супер пароля (отгадайте какого) программа сообщила о своих претензиях на наши деньги. Честно говоря, немного удивляет позиция аффтаров этого ПО. Программа выполнена на настолько безобразном уровне, даже справка не работает (представьте себе), что с трудом верится в светлое будущее AKR, сокращение, кстати, расшифровывается как AntiKeylogger, AntiRootkit, Antiscreenshooter. Давно сложившаяся аксиома – «чем более аляпистый интерфейс, тем меньше полезного функционала» подтвердилась и тут.

Итак, заявленный функционал следующий – поиск руткитов, противодействие кейлогерам, противодействие программам, снимающим скриншоты с экрана, т.е. широкому спектру очень дружелюбного и так необходимого большинству «софта». А сейчас подробнее обо всем этом и как же это реализовано в этой программине. Начнем с самого интересного – детектирование руткитов. Был проведен простой эксперимент с Hacker Defender. Оказалось, что даже наличие нотификатора на создание процессов не помогло AKR. Да, да, да, внимание эта программа не способна найти и соответственно справиться с руткитом не просто прошлого поколения, а прошлого века. Таким образом тестировать её еще с чем-нибудь более серьезным сразу же отпало все желание. Между делом виртуальная машина, на которой гонялся AKR, была под MAOSBootkit, ну было бы просто чудо, если бы AKR хотя бы обнаружил присутствие буткита. Вместо этого эта программа назойливо напоминала нам о присутствии подозрительных драйверов виртуальной машины. Вот уж действительно, вот где они враги! На ум постоянно приходили аналогии с другой не менее лажовой поделкой – UnHackMe от греатиста номер один всея Интронета Дмитрия

Соколова. Действительно, есть что-то пугающее похожее в двух этих программах.

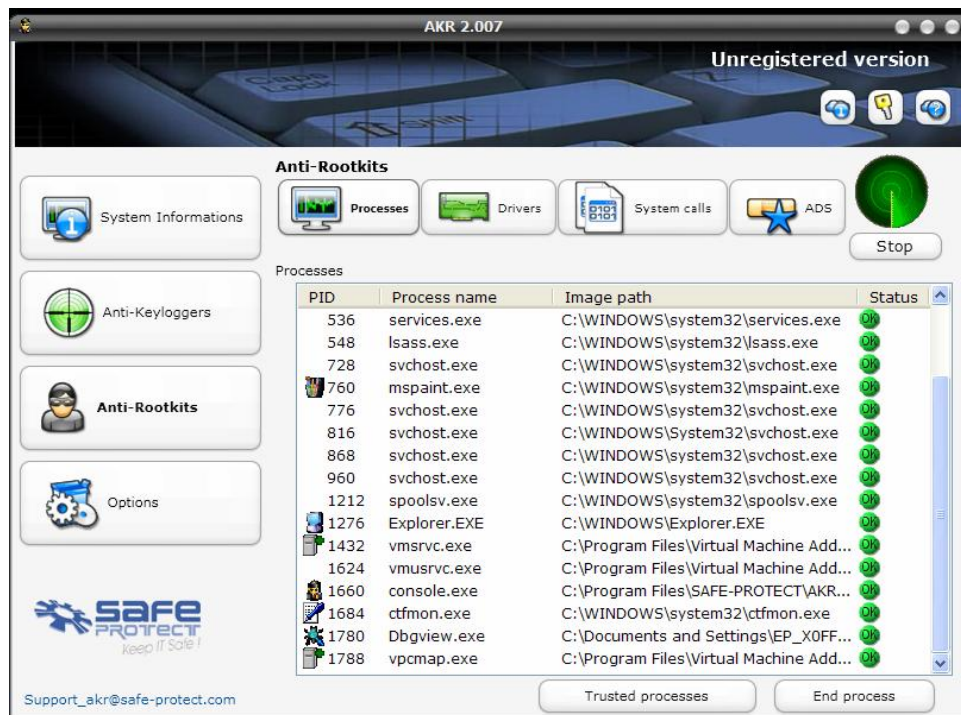


Рисунок 2. Внешний вид AKR 2.007, руткиты где-то там

el666) же на ум приходили веселые ассоциации с фейковыми антивирусами, которых в последнее время расплодилось невероятное количество. Действительно, там тоже красивая обертка, скрывающая тупое бессмысленное содержание и рьяное желание срубить побольше бабла с идиотов и свалить нах пока все не открылось.

О поиске скрытых драйверов, ADS и прочего можно скромно промолчать. Заявлено, но нет. Так может быть весь конек программы в Антикейлогере? С установкой идет занятный PDF файл на французском языке с интересными картинками, демонстрирующими функционал программы и другие разработки SAFE-PROTECT. Становится совершенно непонятно, каким боком они тут затесались, или видимо, мы, просто теряем суть не зная, что там написано. В любом случае антикейлогер и антискриншотер основаны на:

```
[720]svchost.exe-->kernel32.dll-->LoadLibraryExW, Type: Inline - DirectJump at address 0x7C801AF1-->5F05001E hook handler located in [unknown_code_page]
[720]svchost.exe-->user32.dll-->GetKeyState, Type: Inline - DirectJump at address 0x77D3C379-->5F14001E hook handler located in [unknown_code_page]
[720]svchost.exe-->user32.dll-->GetAsyncKeyState, Type: Inline - DirectJump at address 0x77D3D051-->5F11001E hook handler located in [unknown_code_page]
[720]svchost.exe-->user32.dll-->CallNextHookEx, Type: Inline - DirectJump at address 0x77D3ED6E-->5F08001E hook handler located in [unknown_code_page]
[720]svchost.exe-->user32.dll-->SetWindowsHookExW, Type: Inline - DirectJump at address 0x77D5E621-->5F0E001E hook handler located in [unknown_code_page]
[720]svchost.exe-->user32.dll-->SetWindowsHookExA, Type: Inline - DirectJump at address 0x77D602B2-->5F0B001E hook handler located in [unknown_code_page]
[720]svchost.exe-->user32.dll-->keybd_event, Type: Inline - DirectJump at address 0x77D86365-->5F1A001E hook handler located in [unknown_code_page]
[720]svchost.exe-->gdi32.dll-->BitBlt, Type: Inline - DirectJump at address 0x77F16DC0-->5F17001E hook handler located in [unknown_code_page]
```

Насколько это работает, мы не проверяли, да и вряд ли, здесь что-то работает, кроме системы «быстро дай ко мне денег, сцуко». BitBlit видимо, перехватывается как раз из соображений антискришотинга. Остается добавить, что подобные перехваты висят во многих, но не всех процессах. За что же 0.1? За PDF файл с картинками. Программа кстати, генерирует кучу отладочных сообщений, которые просто интересно посмотреть. Там даже мелькает упоминание про Русток. Смеркалось. Занавес.

Вердикт: Фейк чистой воды.

DriverDetect

Оценка: нет оценки

Тип: комбинированный (детект + удаление)

Статус: в разработке

Появившийся в конце 2007 года из ниоткуда антируткит, равно как и его разработчик, привлекли наше внимание по целому ряду причин. Во-первых, первые шаги программы очень напомнили нам, то с чего мы сами начинали в начале 2006 со своим антируткитом, во-вторых, данный антируткит оказался на порядок стабильнее китайских аналогов и чрезвычайно удачно спроектированным – он выполняется без проблем на всех NT, что мы только пробовали, включая даже Windows Server 2008. Кроме того, автор программы оказался достаточно коммуникабельным и лишенным скаммерсантской гнили, которую мы, например, наблюдаем во всех антируткитах антивирусных компаний и таких поделках как UnHackMe, HiddenFinder и тому подобного трэшняка. Программа находится в активной разработке и пока не может считаться полноценным антируткитом, но, тем не менее, мы видим в ней определенный потенциал и поэтому решили взять её в обзор.

Программа уже сейчас обладает следующими возможностями: сканирование на предмет скрытых драйверов с возможностью избирательного дампа оных, копирования файла драйвера, удаления файла драйвера. Поиск скрытых файлов по дискам. Поддерживается FAT32 и NTFS. Поддержка FAT32 до недавних пор сильно глючила, пока мы немного не помогли автору с тестированием. Имеется также возможность удалить и скопировать выбранный скрытый файл. Кроме того, программа выдает также список заблокированных для открытия для API файлов на диске, например файлы подкачки и гибернации. Внешний вид программы прост как полено, но давайте сделаем скидку на сравнительно небольшой период разработки и количество разработчиков равно одному. Кроме того, утилита пишется и это самое главное – Just For Fun. А такой принцип мы уважаем. Функционал небогат, но то, что есть, реализовано на достаточно грамотном уровне. Тому же SafetyCheck потребовалось больше года, чтобы стабилизировать свое выполнение и прекратить бсодить и вешать систему. В общем, есть, что ожидать в будущем. Безусловно, интересная «свежая кровь». А

теперь давайте посмотрим, как это реализовано в этом отдельно взятом антирутките.

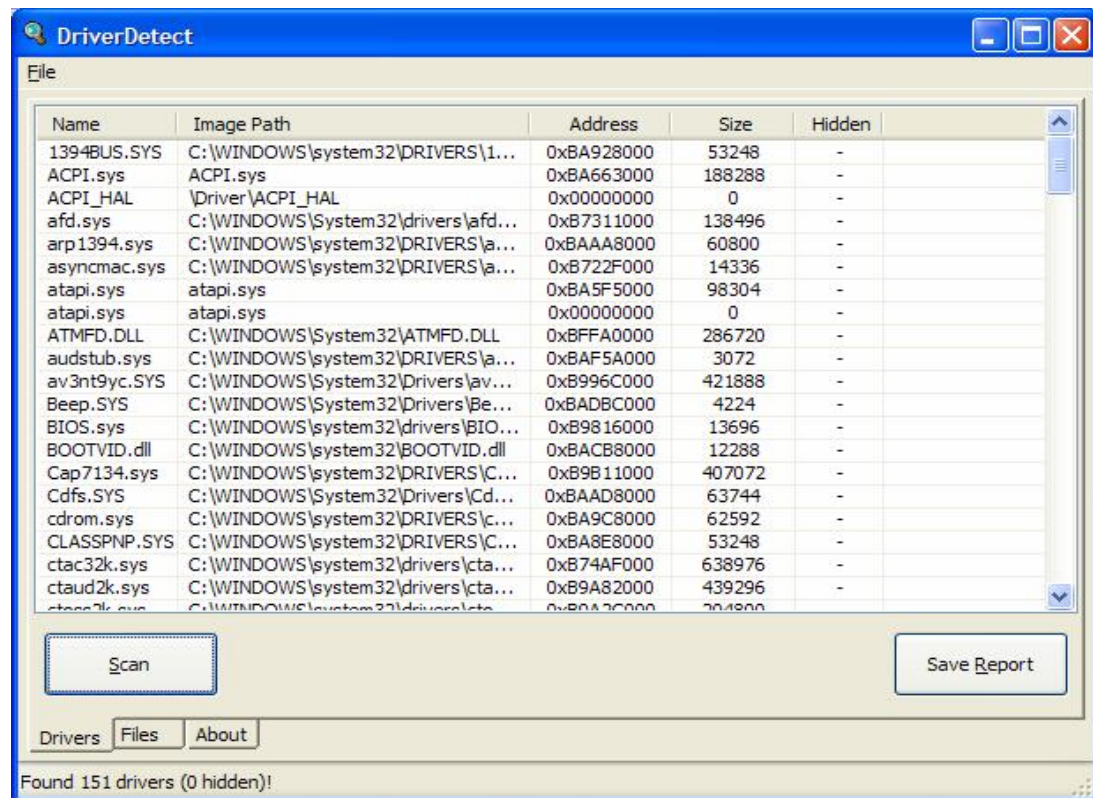


Рисунок 3. Внешний вид DriverDetect напоминает что-то :)

Детектирование скрытых драйверов построено частично на алгоритмах примененных в Rootkit Unhooker, здесь есть проход по списку модулей в ядре, проход по директории объектов, объектам Device и Driver. Программа способна обнаружить Rustock.B и руткиты, применяющие схожие с ним принципы скрытия драйверов. Однако давно известно, что драйверы продвинутых руткитов теперь уже давно не драйверы в прямом понимании этого слова, это просто код, вращающийся где-то в нулевом кольце. Соответственно детектирование скрытых драйверов это теперь всего лишь legacy возможность антируткита, а отнюдь не фишка, как было в самом начале, например с DarkSpy. Автор не стал особо утруждать себя изысканиями и построил свой детектор на железных, хорошо портируемых на новые версии NT принципах. Здесь пока нет никаких хуков, равно как и большого функционала. Детектирование скрытых файлов построено на FSD запросах драйверам файловых систем (поддерживается ADS и NTFS Hardlinks). Это с одной стороны делает подобное сканирование невероятно быстрым (потому как не приходится читать и разбирать структуру самостоятельно), а с другой стороны сильно снижает возможности противодействия и того же детектирования руткитов. Фишка Force Delete представляет собой специальный запрос драйверу FS, соответственно, например с Rustock.B такое не сработает, так как этот руткит просто отфильтрует данный запрос. Однако DriverDetect способен найти файл Rustock.B и может сделать ему Wipe (аналогично возможности Rootkit Unhooker, только на уровне FS запросов). К

сожалению, функционал программы пока не позволяет сказать что-то более определенное.

Вердикт: Ждем продолжения.

DeepMonitor

Оценка: 1 из 5

Тип: комбинированный (детект + удаление)

Статус: релиз

Фактически эта программа PoC – Proof of Concept, и является одно плановым детектором скрытых процессов от одного из участников Rootkit.com. Но как говорить на безрыбье и жопа соловей.

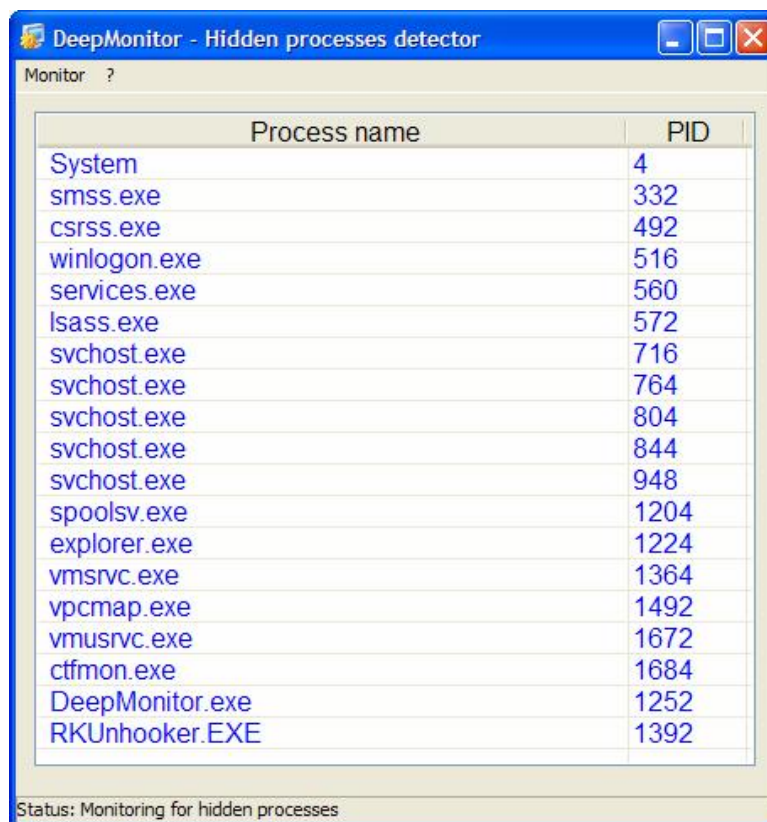


Рисунок 4. Внешний вид программы.

Автор программы использовал все, что знал о поиске скрытых процессов. Почему то это все уперлось в не декларируемый официально, но декларируемый делом слоган одной известной почти отечественной почти антивирусной компании «А давайте хукнем плонеду пожестче?»

DeepMonitor на лету перехватывает SYSENTER Inline – DirectJump edi. Некоторые версии этой программы также для кучи ещё ставят перехват на SwapContext. Контроль и отсеив мертвых процессов осуществляется с помощью перехвата NtTerminateProcess, путем подмены адреса функции в SSDT. Как же все это работает? А вот как. Процесс выполняет вызов, который идет через SYSENTER. DeepMonitor

логирует этот процесс и когда процесс делает другие вызовы периодически сверяет со своим внутренним списком, нормальные процессы при выходе из системы так или иначе вызывают `NtTerminateProcess`. В этом достаточно убедиться, если потрассировать некоторые Win32 API функции. Соответственно, если вызывающий процесс вдруг неожиданно пропал для пользовательского API и продолжает делать вызовы через колгейт, это охтунг. Является ли подобный подход к детектированию скрытых процессов достаточно надежным? Нет. Руткит может пропатчить `ntdll.dll` на свой собственный шлюз вызовов и эта утилита будет пребывать в полном неведении. Но зачем же так усложнять? Оказывается версия, что была у нас мало на что способна, попросту потому что она глотает практически весь фейк, который ей можно подсунуть. Следующим уязвимым местом утилиты является то, как она проверяет процессы. Судя по всему там присутствует, что-то типа сверки по PID, возможно в более поздних версиях (у нас 1.4) это было изменено. А так патчим поля объекта процесс и все. Впрочем, мы сомневаемся, что кто-то всерьез озаботится обходом этой утилиты. Хотя стоило ли ожидать чего-то более серьезного от очередного PoC?

Вердикт: Беспольный PoC.

GMER

Оценка: 3.9 из 5

Тип: комбинированный (детект + удаление)

Статус: релиз

Возвращаемся к нашим баранам. В смысле программ производства Пржемуслава Гмерека, известного польского криворучки и националиста, страстно недолюбливающего все, что связано с бывшим совком. Его утилита за последнее время сделала качественный рывок и теперь уже не такое полное говно, как была ещё год назад во время написания *Invisible War*. Аффттар, наконец-то пофиксил многие свои баги и существенно расширил функционал за счет широкого использования того, что у него уже было реализовано в его, с вашего позволения, так сказать движке. В недавних версиях было обнаружено, что аффттар даже исправил свой хронический идиотизм и **System Idle Process** теперь называется, как положено, а не как в Польше заведено. Более того, частично пофикшен тот баг, что описывался год назад, теперь программа больше не бсодит, она просто висит, намертво уйдя в свой драйвер. Ну, безусловно, аффттар добавил новые баги, например вот такой как на следующем рисунке, GMER сошел с ума и разлистал мне первые 64 сектора как копии главной загрузочной записи, к счастью без нулевого сектора:

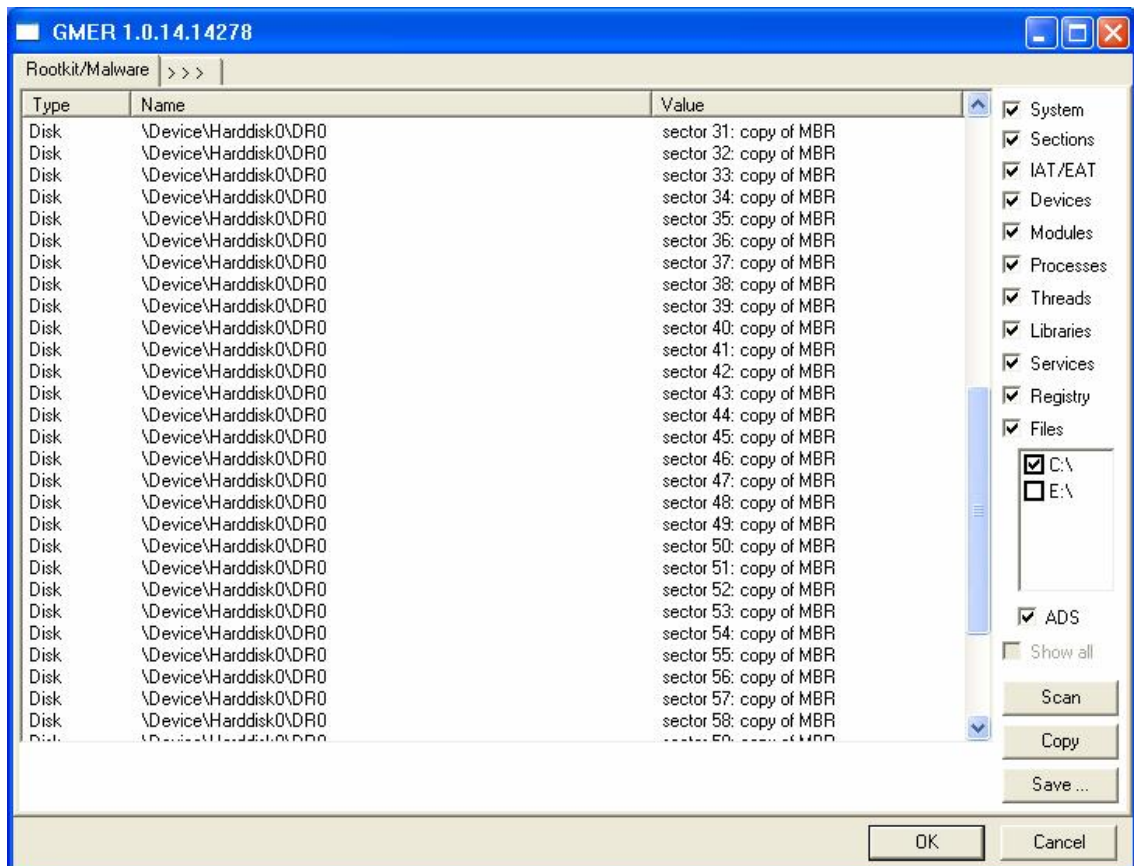


Рисунок 5. Да ты шо, а она шо?

Нововведения с прошлого обзора следующие: добавлено сканирование IAT для пользовательского режима и ядра, у программы появился неприметный такой IAT перехват функции `IoCompleteRequest` у `classpnp.sys`, оставим вам на домашнее задание понять нафига это, ответ довольно простой :) Программа обзавелась собственными браузером файлов и реестра. Все, разумеется, реализовано через низкоуровневое API, потому как мы уже говорили, что уровень Пржемуслава это API. Антируткит довольно уверенно выносит распространенные руткиты и обнаруживает многие тестовые экземпляры. `Rustock.C/Unreal.B` по-прежнему остаются для него в недосягаемости, и будут оставаться там пока кто-то не сольется в ФБР, но мы ведь с вами не такие?

Кроме всего вышеперечисленного в GMER добавлено самое главное нововведение, которое собственно и стало основной причиной, почему он снова был взят в обзор. В конце 2007 года появился первый малварный буткит – вредоносная программа, подменяющая главную загрузочную запись и иницирующую дальнейший запуск системы, что называется, будучи `rooted`. Считалось, что подобные вещи, равно как и `hypervisors` это юмор и в жизни встретятся, но попозже. Однако это попозже наступило пораньше. Как можно предположить все антивирусные подделки оказались разом не у дел и одним местом вверх. Казалось бы, сканирование загрузочных секторов всегда было (особенно в прошлом) диким коньком антивирусов. Некоторые из нас, кто имеет достаточный жизненный опыт, могут вспомнить такие вещи как `CivilDefense`, `BootCom` (тот вариант, что Арахис) или, частично `Inca`,

впрочем, такого во времена Ms Dos (forever, windows must die – примечание el666) было полным полно. Однако господа аффтарты буткита, предположительно «харашо говорящие по руски», сумели воскресить подзабытые уже техники в новом обличье и сразу обломали почти всех. Небольшой экскурс, что же они такого сделали и что же происходит теперь, вы можете прочитать в приложении номер 2.

Давайте посмотрим на этот антируткит немного под другим ракурсом. Сейчас вас ждет сюрприз.

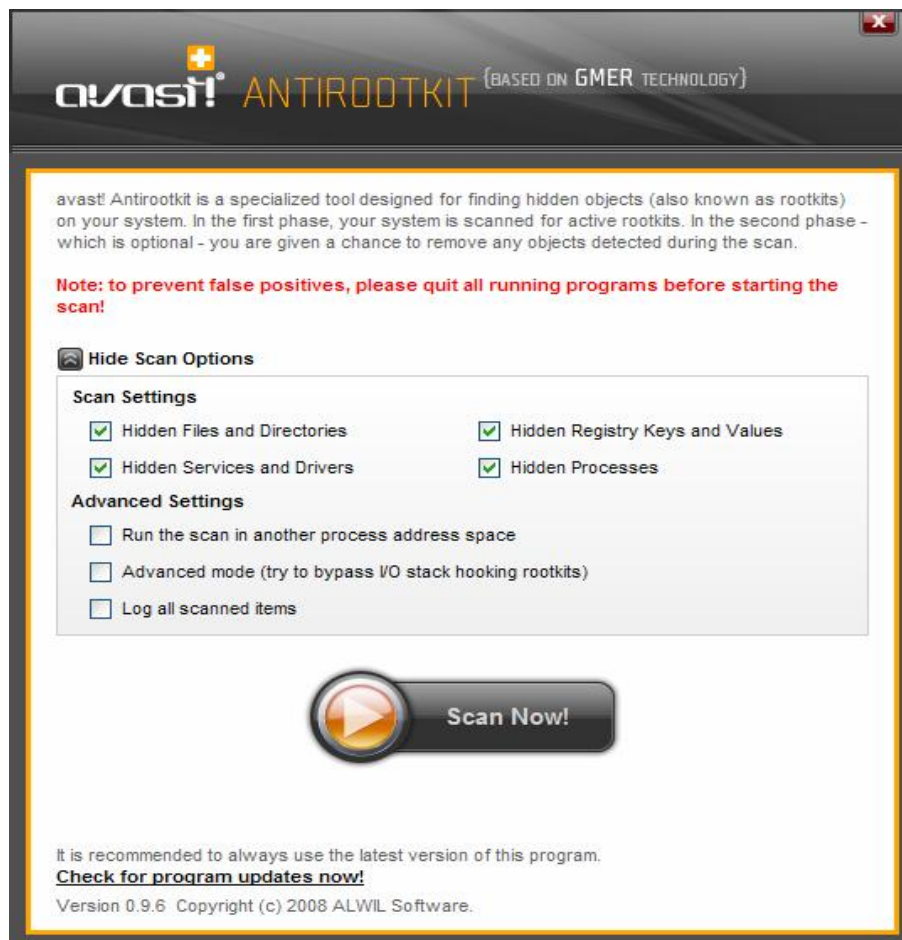


Рисунок 6. Почти нанотехнологии.

Ха-ха, поди, не узнали так сразу? А если бы не баннер наверху, уж точно бы не узнали. Да-да-да, Пржемуслав слился в ALWIL Software ну или слил туда свои «вкусные» технологии. Как видите, функционал заявлен не слабый – тут вам и скрытые процессы, и скрытые файлы и скрытые ключи реестра, даже скрытые драйверы и сервисы. Интересно, а в чем разница между скрытыми ключами реестра и скрытыми сервисами? И тогда как к этому относятся «скрытые процессы»? Загадка. Присутствует даже какой-то загадочный **Advanced mode**. Что же это такое и что оно реально может? В нынешней реинкарнации от оригинального GMER'а хоть и осталось немало, но не все. Видимо не все техноложди ещё прижились. Пржемуславу надо было продаться кому-нибудь в России, тогда можно было бы дописать про нанотехнологии ещё где-нибудь сверху или сбоку. Продвинутый режим это ничто иное, как уже обмусоленные в предыдущем обзоре вызовы `fastfat.sys/ntfs.sys`

напрямую, минуя очереди и возможные хуки Major функций. Запуск сканирования в другом адресном пространстве это создание потока в очень любимом всякими скрипто-киддесовскими поделками Explorer.exe aka Проводник. Это позволяет детектору обойти ограничение видимости, когда например файлы скрыты только от процесса Проводника, антируткит вам не покажет, что они скрыты, потому как он будет их видеть в полном объеме. Мы провели немного извращенный эксперимент, нашли и ликвидировали потоки детектора в Explorer. Новая реинкарнация GMER'а успешно повисла на сканировании :) Так что господа, проверяем Explorer.exe на предмет всякого инжектнутого польского гавна, и если что стопим его потоки нафиг. Ну, это на крайняк, тем, кому не дано никак иначе скрыться от этой поделки на коленке. Текущая версия антируткита от AVAST очень слаба, наверное, потому что это бета. Она откровенно глючит, если поиграть с флажками настроек, программа теряет даже Защитника Хэккеров. В общем Пржемуслува ещё долго фиксить это чудо, но заметьте, интерфейс уже кардинально изменился, вот только вопрос в какую сторону. И ах да, этот антируткит видит буткит, ту версию буткита, что актуальна на момент написания обзора. Камень в огород любителей Виста (впрочем, мы её тоже любим и не только по идеологическим причинам) – аффтарам удалось создать тот же интерфейс и без жутко тормозной реализации пользовательского режима Виста. За что боролись?

Вердикт: Для тех, кто ничего больше не имеет и не может сам.

NIAP AntiRootkit Tools

Оценка: 1 из 5

Тип: комбинированный (детект + удаление)

Статус: релиз

Встречайте очередного уродца, как визуального, так и концептуального. Утилиты Антируткит, от какой то NIAP Soft. Анонс этой утилиты состоялся на Rootkit.com, где последнее время правят бал одни скрипт-киддесы и черные пиарщики. Это три самостоятельные программы – комплексный детектор, браузер файлов и реестра. Наиболее интересен, конечно, комплексный детектор, браузер файлов и реестра реализован через API ядра, там ничего нового и интересного не наблюдается. В самом конце обзора этой группы утилит мы вам поведаем тайну о страшном и извращенном действе, что они творят с Windows неглубоко в её недрах. Присутствует обнаружение скрытых процессов – ламерское до нельзя, поиск драйверов через список ядра, таблицы SSDT и Shadow SSDT, список программ - нотификаторов, список обработчиков Major функций драйверов, перехваты пользовательского и привилегированного режимов, оконные ловушки. Давайте посмотрим на каждую составляющую. Начнем с SSDT. Вот уж где казалось бы невозможно облажаться, так это тут. Казалось бы, нет уже в ядре Windows места более известного, чем эта таблица сервисов. Однако в

NIAP Soft наcodили одну большую кашку и поэтому облажались даже здесь.

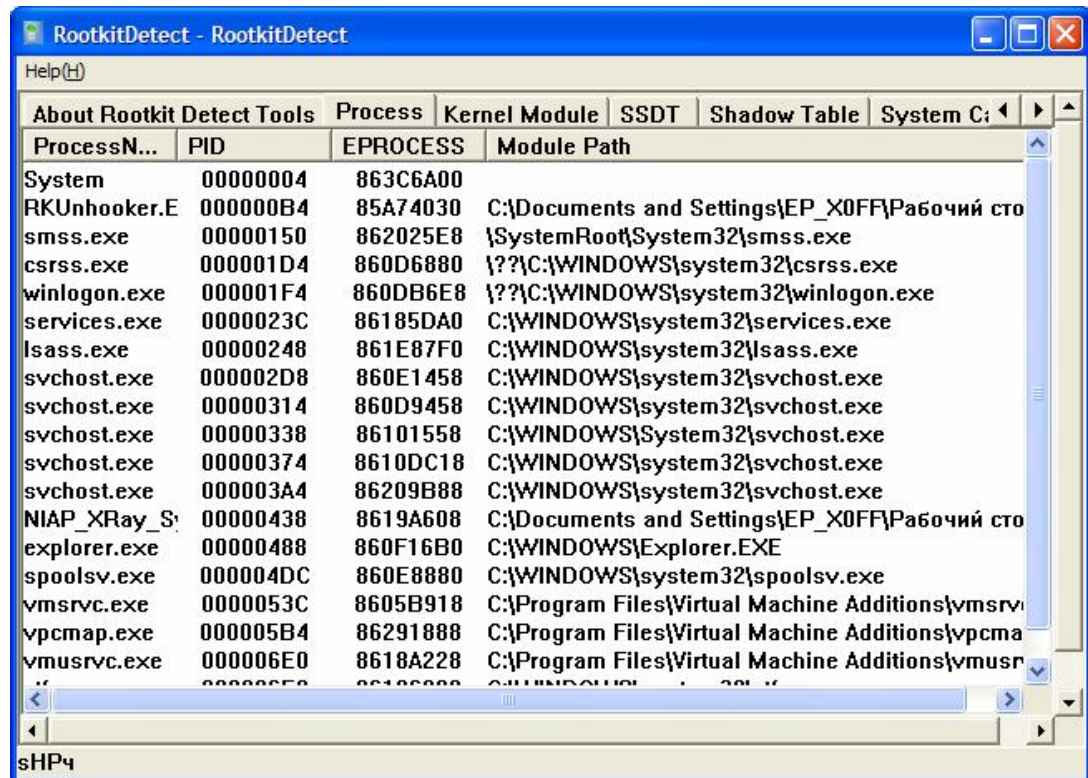


Рисунок 7. Комплексный детектор, который мало что видит.

Да простим афтарам незнание отличий Zw от Nt функций в ядре и не будем их сильно за это бить. Бывает, что просто не дано или судя по функционалу программы просто не оттуда пиздили исходнички.

00000094	805878BD	{\WINDOWS\system32\ntoskrnl.exe	805878BD	ZwQueryEvent
00000095	8057B349	{\WINDOWS\system32\ntoskrnl.exe	8057B349	ZwQueryFullAttributesFile
00000096	805D8720	{\WINDOWS\system32\ntoskrnl.exe	805D8720	ZwQueryInformationAtom
00000097	80572D12	{\WINDOWS\system32\ntoskrnl.exe	80572D12	ZwQueryInformationFile
00000098	805896BC	{\WINDOWS\system32\ntoskrnl.exe	805896BC	fltused
00000099	80621F19	{\WINDOWS\system32\ntoskrnl.exe	80621F19	ZwQueryInformationPort
0000009A	8056C537	{\WINDOWS\system32\ntoskrnl.exe	8056C537	ZwQueryInformationProcess

Рисунок 8. Загадочные и странные записи в SSDT (все).

Бывает, что досадные ошибки сразу отбивают желание тестировать и разбирать программы дальше. А тут целая страница ошибок. Ну что сказать? Идем дальше – Shadow SSDT. До NIAP таким функционалом среди антируткитов мог похвастаться только Rootkit Unhooker, откуда, судя по всему афтары и скомуниздили названия функций теневой таблицы и некоторые идеи для своей поделки. Здесь афтары (или афтар?) с первого взгляда не натворили жутких косяков, так же как и на остальных вкладках. Находить перехваты эта программа просто не умеет, поэтому пойдем дальше. Функционал анти отсутствует, есть разве что попытки прибить выбранный процесс, однако практически все руткиты, скрывающие свой процесс были этой утилитой успешно пропущены. Вот собственно и все, что можно сказать об этом рассольнике утилит. А теперь та самая тайна. Для коммуникации с

приложениями драйвер этих утилит расширяет SSDT на определенное количество записей, прямо как до недавнего времени делал Касперский. Очевидно, что таким образом аффтарты антируткита собираются обходить перехваты функций в таблице, устанавливаемые руткитами, и делать дело, что называется в обход. Ну что же господа, вперед и с песней с такими извращенными и бесполезными методами. К чести аффтартов можно отнести лишь то, что это собрание заблуждений не бсодит и даже эти расширения SSDT обрабатываются корректно, не в пример былым версиям бсодогенератора Лаборатории Касперского.

Вердикт: Бесполезное собрание заблуждений.

Radix

Оценка: 3 из 5

Тип: комбинированный (детект + удаление)

Статус: релиз

Тоже новичок на сцене. Его разработчиком является австриец, что позволяет надеяться, что здесь не будет того безобразия, что наблюдается в подавляющем большинстве китайских антируткитов. Все-таки Европа, мать её.

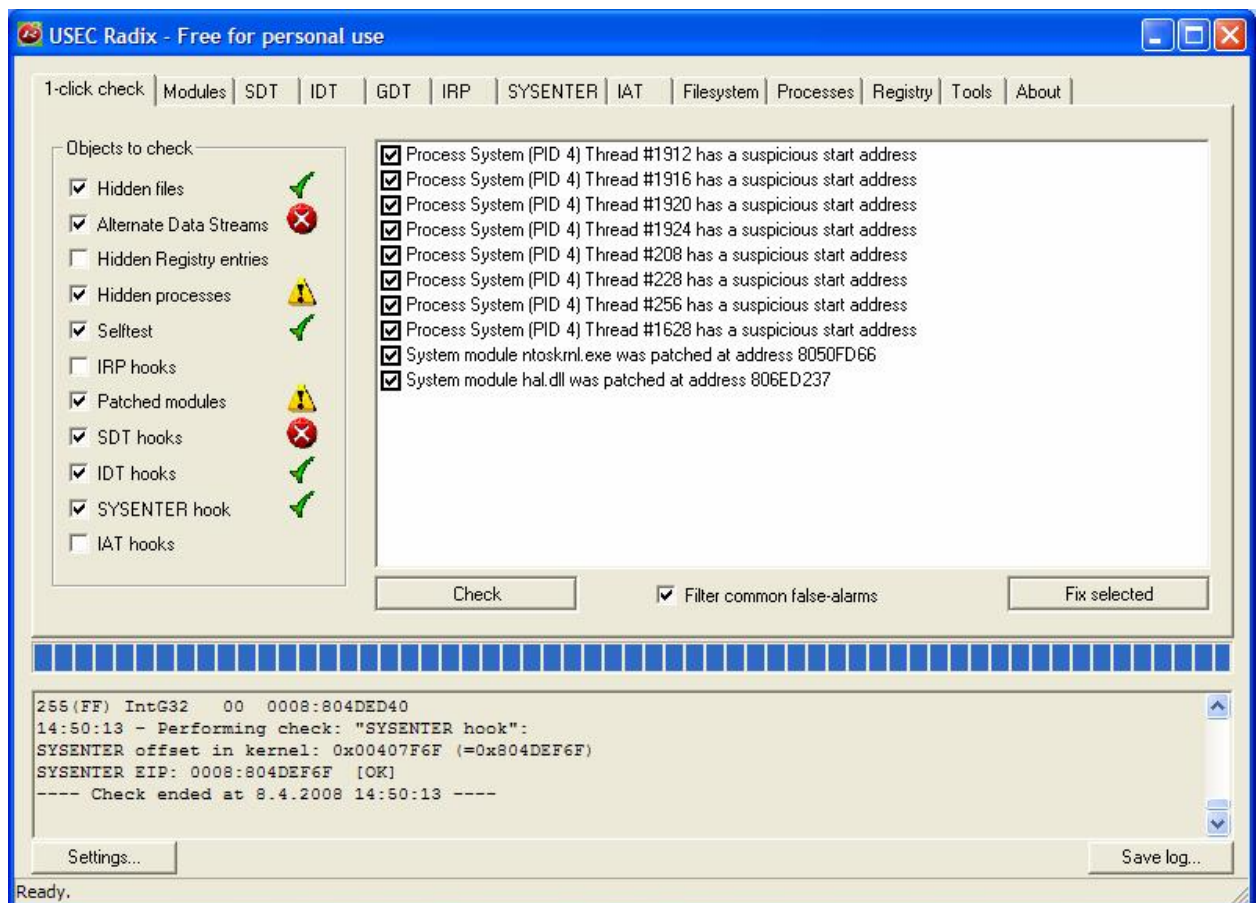


Рисунок 9. RADIX поймал потоки буткита.

Программа обладает очень даже не хилым арсеналом возможностей. Достаточно просто посмотреть на снимок, чтобы понять, что это ещё та штука. Однако спешу вас огорчить. Конец интронета и зиродей всех хэккеров вновь откладывается на неопределенное время. К сожалению, эта утилита более напоминает сборщик системной информации, чем что-либо другое. Поток буткита, кстати, программа нашла по их стартовому адресу, прописанному в **ETHREAD**. Перехват **SYSENTER** проверяется путем поиска оригинала и сравнения с адресом после прохода через колгейт. В качестве домашнего задания можно попробовать отпатчить все адреса колгейта в ядре, уверяем, вас ждет сюрприз. **SSDT** перехваты как у всех остальных антируткитов. До сих пор авторы почему-то наивно пользуются **KeServiceDescriptorTable** для получения таблицы, хотя уже доказано на практике, что это можно надуть. Есть сканирование на предмет скрытых файлов (не умеет), ключей реестра (как у всех остальных), поиск **ADS**, вывод перехватов **IDT** (нетривиальные перехваты тоже не умеет показывать), **IAT**. В общем, полный набор того, что должно быть у нормального руткит-детектора. Автор постарался прикрутить к своему детищу по максимуму возможности что-то сделать с найденным контентом. Например, антируткит может попытаться убить системные потоки руткита (что приводит к синьке), снять перехваты, грохнуть скрытый процесс (поем мантру **PspTerminateProcess**). Присутствует возможность провести самотестирование антируткита, во время которого он проверит сам себя на хуки извне и постарается их ликвидировать. Есть проверка перехватов **Major** обработчиков драйверов, но автор так и не добавил их восстановление, потому что столкнулся с очевидной проблемой поиска оригиналов. Однако с **IDT** он все-таки соорудил восстановление перехватов, которое, правда, ни к чему кроме как к синим экранам, у нас не привело. Программа может вывести даже **GDT**, что вообще большая редкость для антируткитов, подобным функционалом помимо **RADIX**, насколько мы помним, обладали лишь пара тройка утилит, почивших ныне в безызвестности. Другой вопрос в целесообразности подобной возможности в нынешних антируткитах, в любом случае данной программе это, к счастью, не мешает.

Вообще она довольно редко падает в синьку при сканировании, но очень часто при попытке что-либо восстановить.

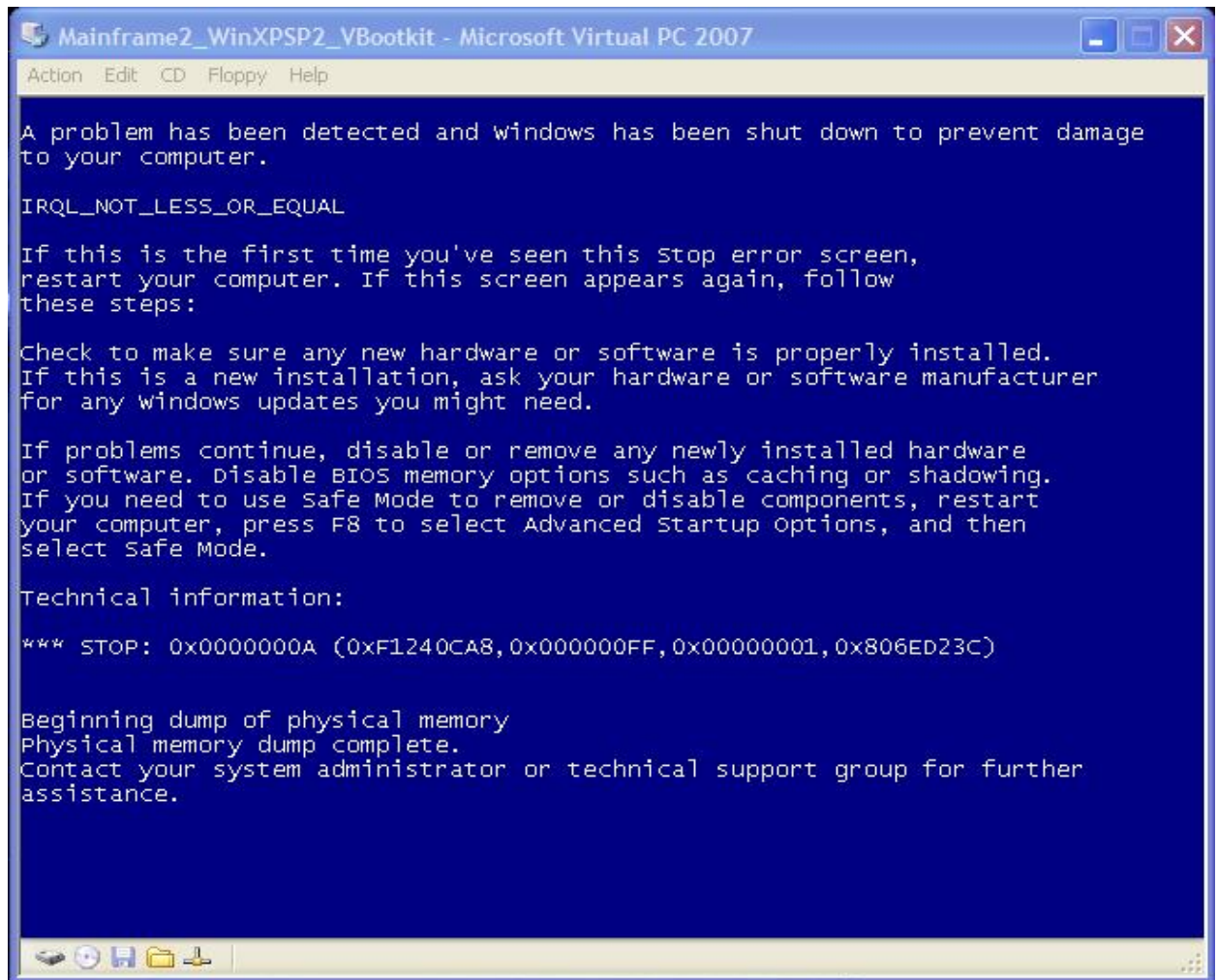


Рисунок 10. После анхука IDT. Анхукнем плонеду!

Есть отдельная вкладка утилит, представляющая собой расширенные возможности по файлам, процессам, есть возможность сдатьпнуть выбранную библиотеку или принудительно выгрузить её из процесса (что вообще то не такая уж и хорошая идея, да и инжект может быть просто куском кода, не обязательно библиотекой). Есть интересная, но не очень хорошо реализованная возможность сдатьпнуть регион памяти из выбранного процесса. Ну, фиг знает, обычно для такого применяют специализированные утилиты.

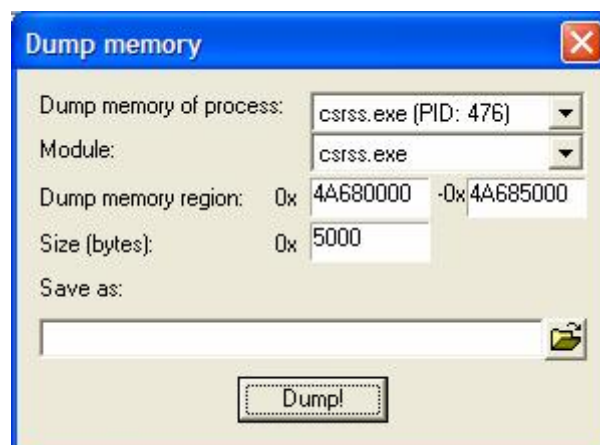


Рисунок 11. Аля PeTools :)

Доигравшись с отхукиванием IDT мы намертво повесили Windows. Антируткит способен найти достаточное количество концептов и реальных образцов. Как уже сказано ранее он видит потоки буткита (ну только потому что аффтарты не отпатчили стартовые адреса), видит и способен вынести Rustock.B, видит концепты RkDemo и ZOmBiE (убить не в состоянии, равно как и опознать, что это такое), не видит PHIDE_EX, однако в новой версии как обещает автор он будет его видеть и возможно корректно завершать, другой вопрос а нафига это надо. Про пару Unreal.B/Rustock.C распространяться особо не будем – не видит. В целом от утилиты остается двоякое впечатление. С одной стороны радуется стабильность работы, богатые возможности, полная бесплатность продукта (кстати, написано сие на ASM/C) и rapid разработка, с другой стороны автор на наш взгляд слишком увлекся привинчиванием всяких мало полезных фишек (включая снятие перехватов, там, где проще убить руткит), полностью положив на техническую реализацию методов детектирования. Ну, с другой стороны программа не профессиональная и пишется пока что Just for fun. Так что не стоит слишком пинать автора, может через год он и дорастет до чего-нибудь серьезного (если не сольется в ФБР, подсказывает el666).

Вердикт: MsiInfo32 на новый лад с анхукингом.

Rootkit Trap

Оценка: 4 из 5

Тип: комбинированный (детект + удаление)

Статус: релиз

Сейчас мы с вами поговорим о самом интересном паблик антирутките, появившемся в прошлом 2007 году. Этот Rootkit Trap от Safe'n'Secure. С чем у вас ассоциируется S'n'S? Правильно с HIPS и StarForce. А у кого-то с несколькими ночами, проведенными в отладчиках и дизассемблерах только чтобы снять эту правильно навешанную хрень. Но сейчас не об этом. Собственно первая информация об этой утилите появилась в самом начале 2007, сначала по закрытым каналам (типа, пишут, пишут, одей!), потом в виде неблагоразумно выброшенной в паблик альфа версии этого детектора. Здесь господа-заказчики совершили одну большую и в целом смертельную для коммерческого проекта лажу. Они дали возможность оценить имеющиеся наработки и предусмотреть их дальнейшее развитие, что и было нами с успехом сделано. Кто-то может назвать это плагиатом, но тут же окажется не прав. Copy-paste из этого антируткита никто не делал. В конце концов, мы не такие идиоты, достаточно было лишь понять идеи и развить их в дальнейшем по полной программе. Кроме бсодов и глюков альфа версия была ничем не примечательна, но нам все-таки удалось создать для нее стабильный environment и мы полностью изучили её возможности. Значительно позже была выпущена

полноценная версия, вошедшая в состав другого продукта S'n'S. Как и ожидалось, там наблюдалось закономерное развитие идей и выправление их реализации. Между прочим этот антируткит один из немногих, с кем мы вообще пока ещё не получили ни одного бсода, даже с активными руткитами. Ну что же надо отметить работу, проведенную автором программы. RkTrap, кстати, одна из немногих паблик утилит способных обнаружить присутствие буткита. Довольно хороший результат, учитывая, что буткит появился намного позже антируткита и соответственно должен был учитывать возможности этого антируткита, но видимо афтары не сочли это нужным или его не было в ТЗ.

```

Safe'n'Sec Rootkit Detector v.1.0.0.2
Copyright (c) S.N.Safe&Software Ltd 2007. All rights reserved.

During full kernel memory scan there is a small risk of BSOD and the associated
risk of data loss. Therefore save/close all opened documents before continuing.

Extended mode

Search for hidden kernel modules...
Found hidden kernel modules!
Possible address of stealth code: 85FA8E5Ch
Possible address of stealth code: 85FAEF38h
Possible address of stealth code: 85FF0000h
Possible address of stealth code: 85FF3000h
Possible address of stealth code: 85FE10A0h
Possible address of stealth code: 85FCE1A0h
Possible address of stealth code: 86016DD0h
Possible address of stealth code: 85FBA220h
Possible address of stealth code: 85FB09A0h
Possible address of stealth code: 85FAEBF6h
Search for hidden processes...
not found!
Search for SSDT hook's...
not found!
Search for Inline hook's...
Code modification!
Near wcsncmp (function entry point + 1104 bytes)
at address: 805056E9h: JMP DWORD PTR[424168Ch]
handler inside Dbgview.sys
Try to unhook this hook? (Y/N) NO
    
```

Рисунок 12. Буткит пойман за яй... в смысле потоки.

Антируткит обладает возможностями поиска скрытых драйверов, процессов, перехватов кода (только в привилегированном режиме), перехватов IDT, может попытаться грохнуть процесс или снять перехват. Есть даже возможность удалить файл через отложенное удаление. Казалось бы небогатый функционал, но чем же так примечателен RkTrap?

А ответ прост – реализацией всего вышеперечисленного. Как показывают тесты, она на порядок выше всех аверских конкурентов, китайских подделок. Давайте посмотрим, как же тут все реализовано. Процессы программа детектирует при помощи перехвата SwapContext. Контроль и отсеивание мертвых процессов осуществлен с помощью программы - нотификатора. Вот вообще список перехватов этой программы, она их ставит только на время поиска и потом снимает, так что можно первый раз и не понять, что происходит.

ntoskrnl.exe-->SwapContext, Type: Inline - RelativeJump at address 0x804DBEB9-->F7B2936E hook handler located in [dsLL2zL.sys]

```
ntoskrnl.exe+0x0002E8CC, Type: Inline - RelativeCall at address 0x805058CC-->F7B283EC hook handler located in [dsLL2zL.sys]  
ntoskrnl.exe-->ExAllocatePool, Type: Inline - RelativeJump at address 0x8050FD66-->F7B288D2 hook handler located in [dsLL2zL.sys]  
ntoskrnl.exe-->ExAllocatePoolWithTag, Type: Inline - RelativeJump at address 0x8054B044-->F7B287E4 hook handler located in [dsLL2zL.sys]  
ntoskrnl.exe-->NtEnumerateKey, Type: Inline - RelativeJump at address 0x8056F76A-->F7B2851A hook handler located in [dsLL2zL.sys]  
ntoskrnl.exe-->NtDeviceIoControlFile, Type: Inline - RelativeJump at address 0x8057FBD0-->F7B28608 hook handler located in [dsLL2zL.sys]
```

Если `SwapContext` это понятно, то остальное для неопытного читателя может представлять нетривиальную загадку. В первую очередь давайте рассмотрим загадочный адрес (`Rootkit Unhooker` облажался выдать имя функции) прямо следующим в списке после `SwapContext`.

```
lkd> u 0x805058CC  
nt!DbgPrint+0x1b:  
805058cc e81b2b6277 call f7b283ec
```

Как видите перехват, имеет место быть, и это действительно вызов в драйвер `Rootkit Trap`. Имя для самозащиты постоянно выбирается случайным образом, кстати, это отнюдь не гарантирует невозможность 100% определения, что это драйвер `RkTrap`. Автор поставил перехват на вывод отладочных сообщений. Это вызывает двоякое чувство. С одной стороны это не очень красиво, применительно к концептам, с другой, а почему бы и нет. Суть этого и следующих хуков на пул заключается в том, что антируткит определяет, откуда пошел вызов этой функции и если вызов из неизвестной области значит это руткит, потому что сама операционная система, хоть и любит патчить себя в runtime из соображений производительности, но таким извращением как вызовы из левых областей не занимается. Хуки на пул довольно удачны, так как любой руткит нуждается в памяти и, несомненно, будет вызывать эти функции хотя бы раз. Перехват на реестр и вызовы `IOCTL` служат так же для детекта, причем в первом случае довольно оригинального, что называется, чем богаты, тем и рады. Стоит отметить, что при наличии собственного ядра такие перехваты (и вообще все хукалки) теряют актуальность. Однако `RkTrap` известен другой своей особенностью, которая позволяет ему находить глубоко спрятанные драйверы, которые собственно уже и не драйверы совсем. Он умеет сканировать системные потоки и определять место их выполнения, причем не такими ламерскими методами как `GMER` или `RADIX`. Это позволяет ему находить даже те руткиты, что вышли гораздо позже него, потому что как показывает долгая практика уровень аффтара в скам - rootkit производстве гораздо ниже той планки, которую они сами себе нарисовали мелком на плинтусе. Отдельно стоит упомянуть о продвинутой системе поиска перехватов. Автор реализован свой маленький дизассемблер и анализатор кода, что позволяет ему успешно обнаружить некоторые перехваты и их обработчики, включая вызовы типа `jmp eax`. К сожалению, сканирует этот антируткит далеко не все и его анализатор можно довольно легко положить на лопатки, поиграв с командами (такое было реализовано в `Unreal.V` специально под `RkTrap`, надо отметить, что на отдельный обход этой утилиты пришлось

переписать некоторые места руткита). Удаления как такового практически нет, есть слабые попытки, это более детектор, чем что-либо иное. Однако в целом и общем антируткит оставляет приятное впечатление, к тому же он бесплатный, что накидывает ему баллов дополнительно. Rustock.C (пока он с неснятым протектором) этой программой найти невозможно.

Вердикт: Весьма удачная комбинация новых методов и их реализации.

Root QUEST

Оценка: 0 из 5

Тип: Фейк на VB

Статус: релиз

Не хочется после хороших продуктов рассматривать откровенное говно и недоразумение, каким является Root QUEST, но так как это обзор существующих и новых средств поиска и удаления руткитов, придется замолвить и об этом гуано несколько слов. Продукт жизнедеятельности <http://www.comsentry.com>. В «прочитай меня» файле вы можете увидеть вот такое:

RootQuest is a good solution to detection and remove all rootkit that currentlty running as background services on computer system

Ложь, пи...ж и провокация, ещё и с ошибками в написании. Данная поделка на VB коленке не способна ни на что. Кроме того, она исключительно глючит со своим рисованным интерфейсом. Заявлено сканирование файлов, поиск скрытых процессов – стандартный набор для антируткитов – неудачников. Окно со скином (ах как модно это стало в последнее время) настолько неудачно спроектировано, насколько только может быть, нет даже элементарной возможности растянуть его если не на весь экран, то хотя бы на пару пикселей. Шрифт, применяемый программой для вывода информации очень дружелюбен для глаз. Судя по тому, что программа выдает на вкладке процессов, их список получен либо через PSAPI, либо через NativeAPI. Копаться в этом отстое на бэйсике у нас нет не малейшего желания.

В «прочитай меня» файле можно также узнать, что продукт в разработке и скоро порадует нас новыми версиями и вариациями на тему какое ещё гуано можно привинтить к тому, что уже есть. На рисунке ниже это не буквы П (наверное от слова П...ж), и не Л, это О! НН! Во время сканирования этот антируткит нашел на диске архив с Защитником Хэккеров и радостно сообщил об этом, при этом ЗХ был не запущен. О_о, нашел по MD5.



Рисунок 13. Убожество чистой воды.

Вердикт: Давить!!

SEEM

Оценка: 1.5 из 5

Тип: комбинированный (детект + удаление)

Статус: релиз

Незаслуженно обойденный вниманием в предыдущем обзоре французский антируткит от некоего 3psilon представляет собой комбинацию MsInfo32 с антируткит составляющей. Давайте не будем долго мусолить дурацкие возможности (типа показать размер дисков, вывести апплеты панели управления), и перейдем непосредственно к антируткит составляющей. Для начала на заправку был дан Защитник Хэккеров, SEEM успешно справился с его детектом, потом он также успешно нашел и процесс HideToolz, скрывающий свое присутствие хуками в SSDT. На этом возможности программы закончились, все остальные концепты и реальные руткиты она не смогла найти. На странице SSDT твориться что-то до боли знакомое по кривому до нельзя NIAP AntiRootkit Tools. Там та же самая таблица. С тем же самым неизвестным науке сервисом.

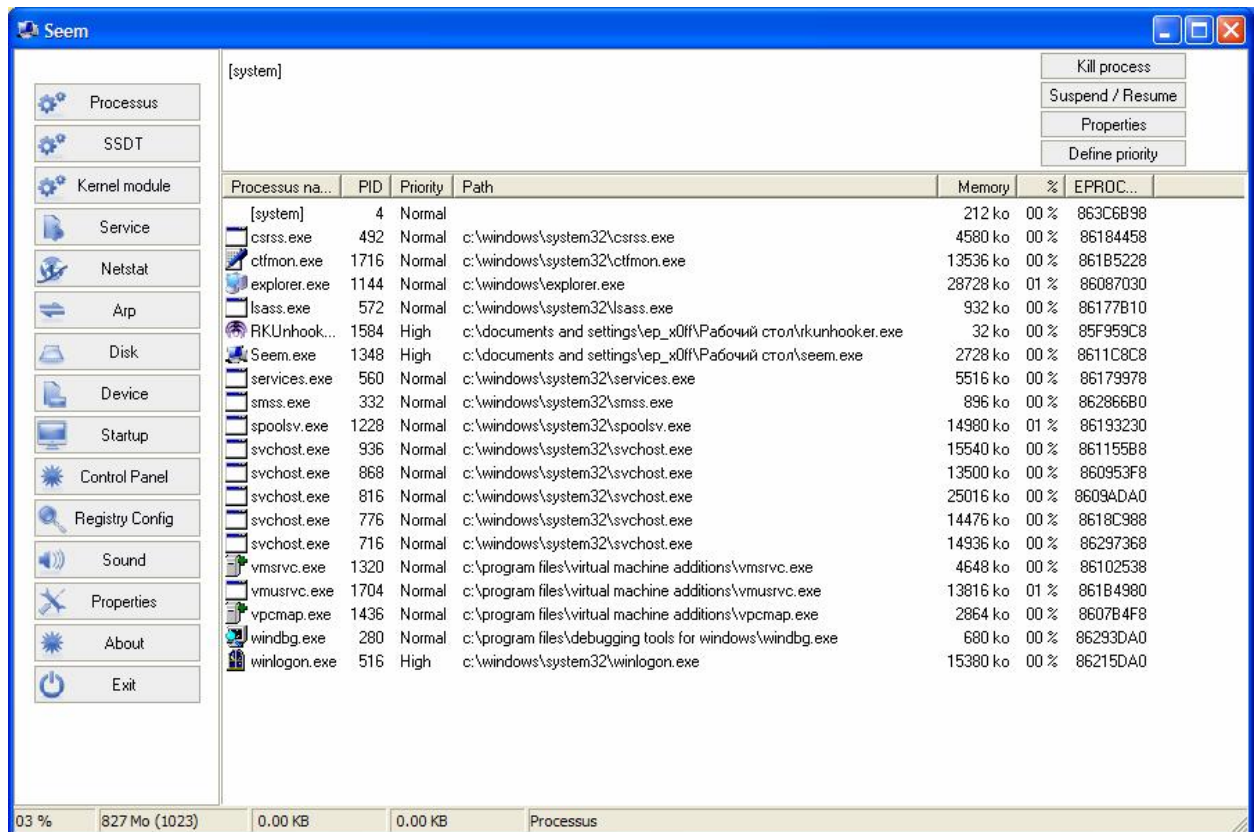


Рисунок 14. Внешний вид SEEM.

SEEM расшифровывается как System Eyes & Ears Monitoring. Так что это в принципе даже и не антируткит, поэтому он и не участвовал в предыдущем обзоре, а сейчас добавлен, что называется для кучи. Программа была создана во время Hacker Defender и с тех пор не претерпела сколько-нибудь крупных изменений, возможно в результате того, что авторы SEEM просто не имели доступа к более новым концептам и реальным руткитам, либо им было пофиг. Так или иначе, но программа продолжала развиваться, и насколько известно лишь недавно на ней был поставлен крест. Как признался сам главный автор Zpsilon закончить SEEM он решил после того как попробовал Rootkit Unhooker :) Это не шутка.

Вердикт: Так победим! Уже победили.

Spyware Processes Detector

Оценка: 1 из 5

Тип: комбинированный (детект + удаление)

Статус: релиз

Вы знаете кто такой Артём Михайлов? Ну же, напрягитесь. Быть может, вы хоть раз слышали об ArtMoney? Если да, то Артём Михайлов это как раз и есть автор этой и не только этой программы. Для тех, кто не в курсе ArtMoney это программа на протяжении долгих лет

упрощавшая жизнь всяким прыщавым идиотам, не желающим играть в игры честно, кроме этого программа использовалась даже хэкерами и крякерами как брутфорсер памяти и редактор. Программа выпускалась и выпускается далеко не из альтруистских побуждений, а за сладкие шекели, которые автор стрижет со своих пользователей. С другой стороны у него есть и бесплатный вариант, так что прыщавые недоноски в любом случае останутся рады своими бесконечным жизням и виртуальным миллиардам в виртуальных кошелках. Вот бы ещё и в жизни так, правда? Ну и конечно в мировой помойке Интронет вы всегда можете найти платную версию, которую добрые дяди сделали бесплатной.

Сразу появляются нехорошие мысли, с чего это вдруг автор ломалки игрушек вдруг озаботился безопасностью своих пользователей и решил выпустить (опять же видимо, только для них), антируткит. А, что не понятно? Бабки ему нужны, бабки! Гони монету и спи спокойно (вечным сном)!

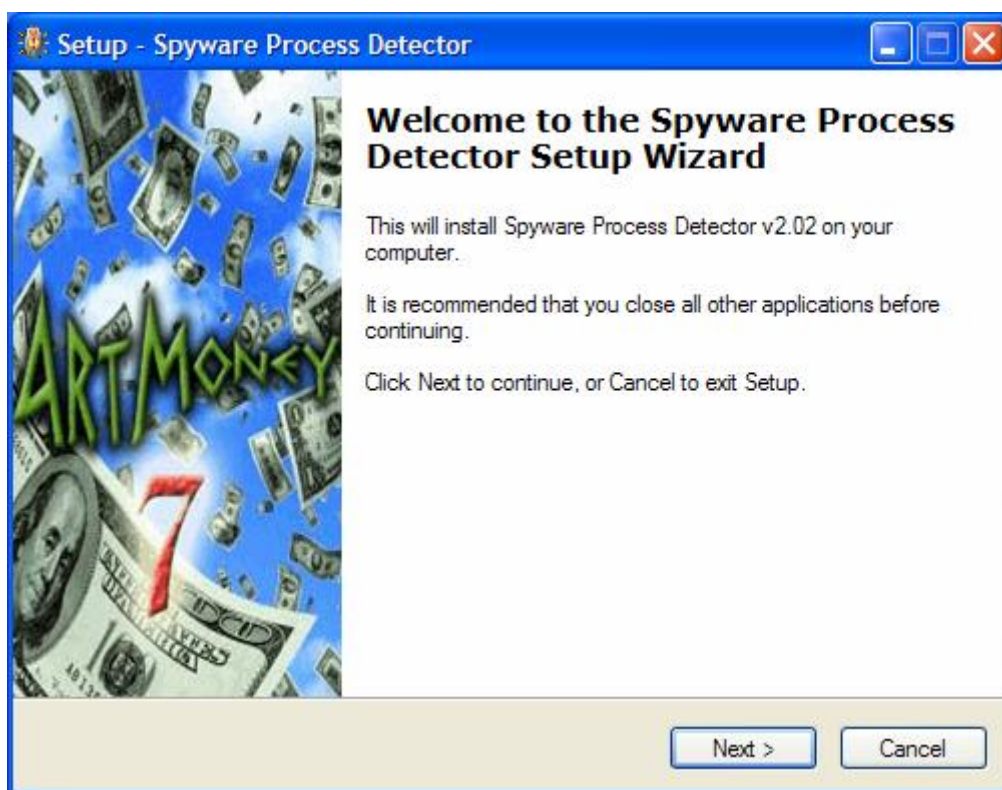


Рисунок 15. Скриншот установки одной из предыдущих версий, толкаем раскрученный товар даже тут.

За свое детище аффар желает приблизительно 25 баксов, т.е. в пересчете на нынешний курс это получается: $25 \times 23.60 = 590$ деревянных, видимо на самом деле в районе 600. Даже Дима Соколов, другой любитель стричь бабки с идиотов, не додумался до такого и был скромнее в своих запросах. На момент написания обзора аффар зарелизил новую версию, мы не нашли в тестах между ними никакой разницы и поэтому переснимать скриншоты не собираемся. Эта программа как очевидно из названия предназначена для поиска скрытых процессов и представляет собой детектор на основе метода, известного

как Просмотр листов планировщика aka KiWait. Больше она ничего не знает и не умеет. Хотя с другой стороны даже этого достаточно, чтобы найти все или почти все вредоносные руткиты, скрывающие свой процесс. В том, что в программе применяются подобные методы можно убедиться, засунув её в дизассемблер и посмотреть места упоминаний KeDelayExecutionThread, KeWaitForSingleObject а, заодно сместившись выше лицезреть огромное скопище констант.



Рисунок 16. А ну гони бабки сцуко!

С найденными скрытыми процессами можно сделать много чего интересного. Например, остановить или запустить опять (прямо Process Explorer), удалить (wtf, видимо убить), удалить вместе с файлом (ну это при условии, что это гуано одетектирует руткит с путем). Кроме этого можно посмотреть информацию о процессе. Разные процессы подкрашиваются в разные цвета, степеней градации всего три – зеленые, желтые (подозрительные) и красные (скрытые). Как всегда в подобных утилитах первыми жертвами стали сервисы бедной виртуальной машины, на которой все это гуано и гонялось. Вот что программа радостно сообщила по их поводу:

```
Process ID: 1484
Parent ID: 560
Status: Suspicious
EXE Filename: vpcmap.exe
Filename:
  C:\Program Files\Virtual Machine Additions\vpcmap.exe
Company:
  Microsoft Corporation
Description:
  Virtual Machine Folder Sharing Service
Tray Icon: No
Windows: No
Threads: 3
  1488,1496,1500
```

Ах ты, ну надо же, подозрительный сервис с действительной цифровой подписью Microsoft. Хорошо хоть не AFX Rootkit как у Димы

Соколова. Программой заявляется поддержка всех операционных систем и даже линейки 9x mustdieв. При использовании этой программы в качестве тестового инструмента нас не покидало ощущение использования фейка, причем ещё и ревностно требующего за это денег.

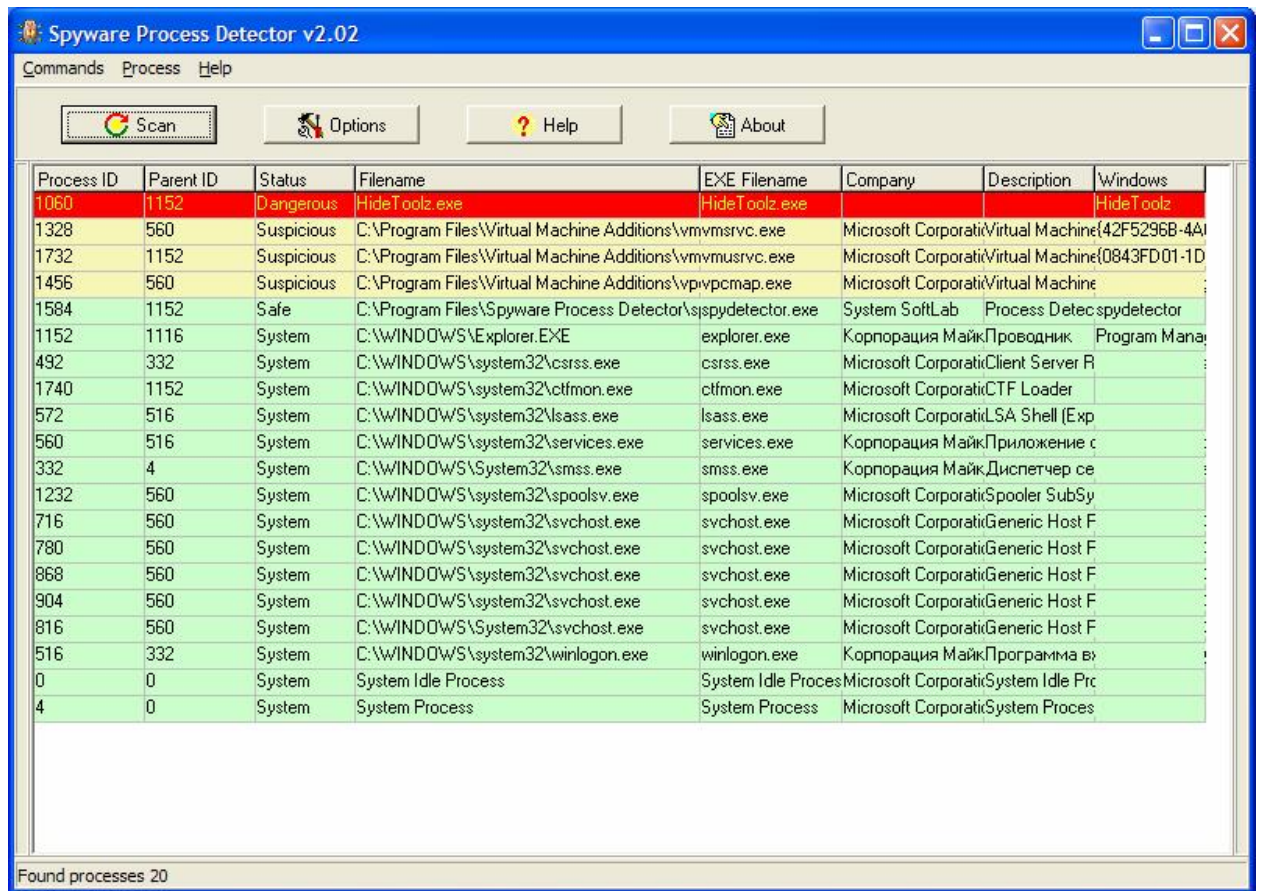


Рисунок 17. Словленный HideToolz, браво-браво!

Об этой программе стало известно достаточно давно, она была громко прорекламирована с помощью статьи rootkit.com, написанной одним из помощников Артёма Михайлова (по крайней мере, он себя так называет). Мы долго не решались брать этот кг/ам в обзор, но с той версии утекло достаточно воды и программа, наконец, то смогла хоть что-то найти, за что мы и передаем свои сердечные 386 поздравлений аффтари и его бригаде. PHIDE_EX эта программа не видит, что-то более серьезное и подавно. Так что аффтари можно пожелать завязывать с листами планировщика, поскольку руткит может (а если будет такой, то просто обязан) использовать свой собственный. В заключение расскажем вам о том, как господин Артём Михайлов заинтересовался темой Unreal. В начале 2007 года на мыло проекта Rootkit Unhooker пришло письмо того самого помощника аффтара. Текст письма, к сожалению, не сохранился, но смысл заключался в следующем: «эй, там, а ну выслали нам Unreal.A с исходниками!» Разумеется, этот далбаб был проигнорирован, но запрос запомнился и теперь именно это и ассоциируется с этой лажовой платной поделкой аффтара, решившего, что теперь можно все. Как можно предположить аффтар не остановился на достигнутом и продолжает кодить свежие версии, как этого гуано, так

и ArtMoney. Наш совет Артёму: не стоит лезть туда, где ты полностью некомпетентен и ещё требовать за свою некомпетентность какие-то деньги. Убей себя, спаси пленеду.

Вердикт: Фейк за ваши деньги.

UnHackMe

Оценка: 0 из 5

Тип: Фейк на CBuilder

Статус: релиз

Неужели за прошедший год произошло что-то, что подняло УнХакМу на новый доселе невиданный для этой программы уровень? И, да и нет. С точки зрения детектирования и удаления руткитов мало, что изменилось. Все тот же примитив, все те же дифирамбы Партизану, все те же ложные срабатывания на сервисы VM и кучу стороннего софта.

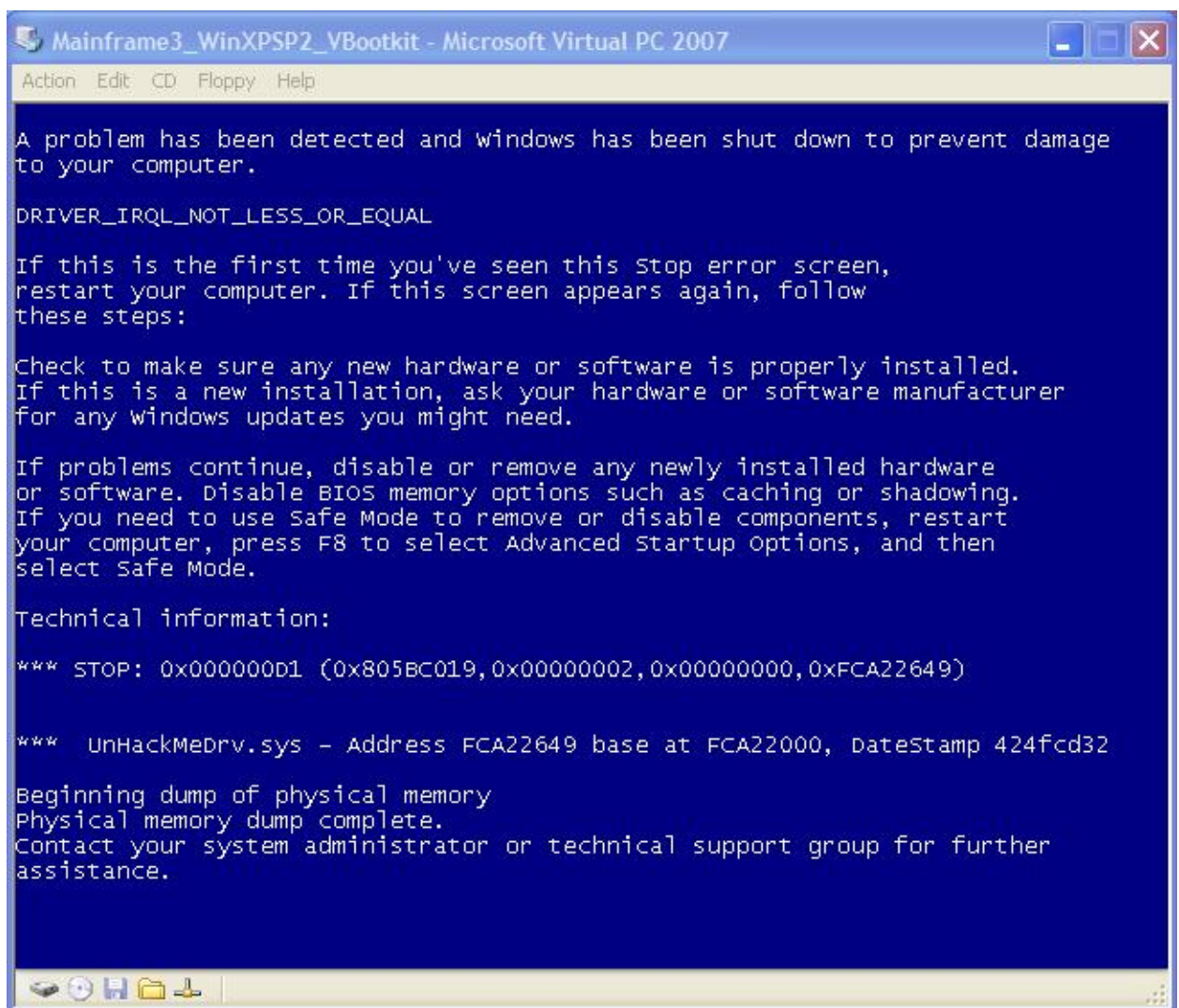


Рисунок 18. Установка успешно завершена.

Упрямый аффтар продолжает пиарить свою поделку на забугорных форумах и своем собственном, где господин Дмитрий Соколов является эксклюзивным участником группы пользователей – **Newbie Administrator**. О чем же ещё тут может идти после такого речь? Клиника, да и только.

Следует отметить, что УнХакМа (ака УнФакМи) за последний год «доросла» до версии 4.70 и, представляете, есть идиоты, которые до сих пор ей пользуются, причем официально. Ей богу, чем был бы мир без guinea pigs. Когда мы собрались писать обзор новой версии этого агрегата по ловле Invisible Trojans, то естественно направились сначала в локальную сеть, где нашли 4.70 заодно с кряком, а потом чтобы проверить, а есть ли обновление, вышли в Интернет. Обновления не оказалось, при установке УнХакМы этот фейк на заключительной стадии вывалил систему в **BSOD**. Мы совершенно не ожидали подобного от УнХакМы, так что Дмитрий нас продолжает удивлять вновь и вновь. У него хорошо получается, с такими талантами надо бы в цирк, а Дима?

Программный комплекс аффтара Дмитрия Соколова за прошедший год пополнился рядом бесполезных, исключительно назойливых и кривых агрегатов. Например, **Anti Spyware...** (именно так, с троеточием в конце), который на старте виртуальной машины порадовал сообщением:

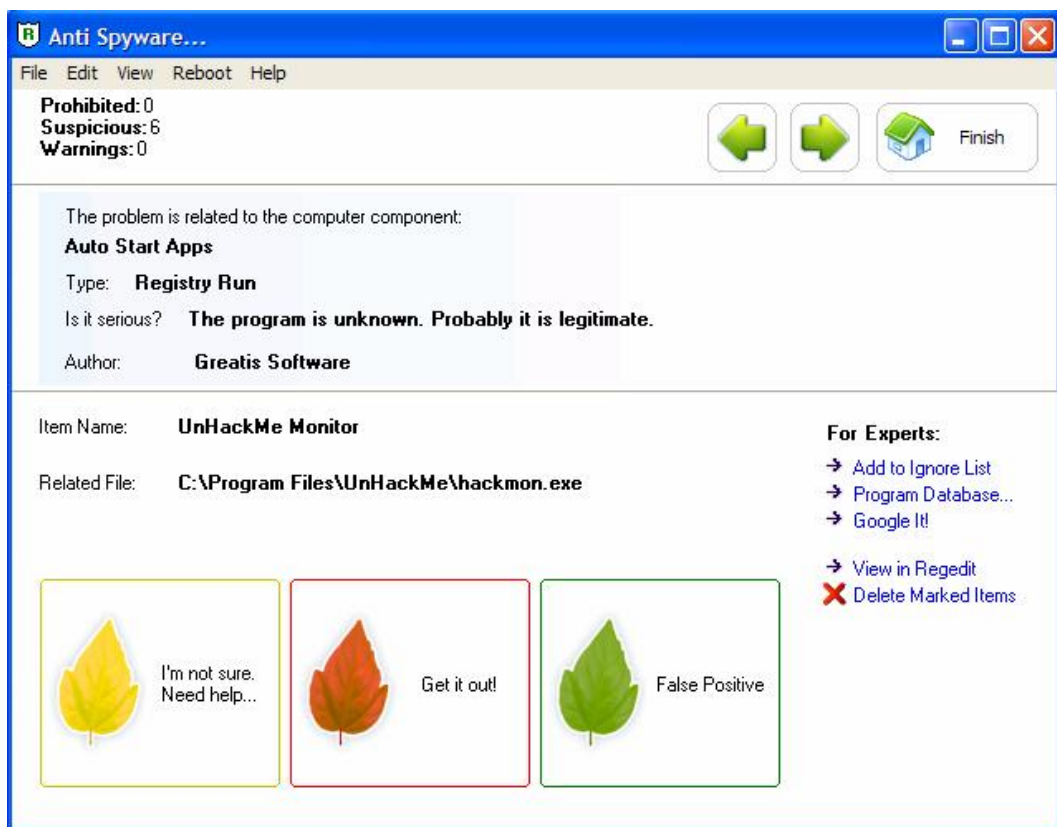


Рисунок 19. Сам себя.

Даже мы перепотрошившие кучу антируткитов, почти все антивирусы, фаерволы и средства предотвращения вторжения были в небольшом шоке от увиденного. Это же надо так облажаться, чтобы свой собственный монитор подсвечивать как подозрительную запись? Дмитрий Соколов переиграл сам себя. Может быть, с этого и следовало начинать? Помимо самой УнХакМы и этого подозрительного монитора в

состав новой компиляции от Греатистов входит тулза под названием **RegRun Reanimator**. Видимо как stand-alone эта утилита совсем не пошла, и Дмитрий решил склеить её в экстазе с другим своим наколеночным брендом. Это аналог **AutoRuns** от Марка Русиновича только с меньшим количеством автостарт локаций и кучей убогих, никому не нужных функций. Примечательно уведомление о количестве дней оставшихся до регистрации продукта. Эту информацию можно посмотреть в **About** программы **Reanimator**. Отмечу, что число -1 это то, что выдает сама программа, так как нами не предпринималось никаких попыток её сломать или использовать уже готовую крякалку.



Рисунок 20. Полный Абзац

Видимо Дмитрий живет в своем отдельно взятом измерении, где у него осталось ровно -1 дня. Кроме всего прочего **Reanimator** может просканировать систему на вирусы. Мы даже немного испугались за буйствовавший в этот момент на виртуальной машине наш любимый **Win32.Polip**. Сканирование на вирусы, почему то привело все в ту же убогую **Anti Spyware...** к все тому же подозрительному монитору **УнХакМы**. В этих агрегатах мы нашли столько дублирующего и бесполезного функционала, сколько редко где видели ещё. Например, вот ещё одна тулзень, которую можно вызвать только внутри **УнХакМы** из какой-то задницы либо как отдельный исполняемый файл из каталога программы. Это специальная утилита, предназначенная для удаления **Русток** варианта Б.



Рисунок 21. УнХакМи меня нежно.

Немного погоняв УнХакМу под отладчиком, мы засунули её в дизассемблер. И узрели истЕну, а стало быть и правду. Агрегат прямо на DriverEntry раскрыл все секреты этого маленького драйвера. Вот вам выдержка из HexRays.

```
PsGetVersion(&MajorVersion, &DriverObject, &BuildNumber, 0);
if ( MajorVersion == 5 )
{
    if ( !DriverObject )
    {
        gKernelVersion = 1;
        goto LABEL_4;
    }
    if ( DriverObject == 1 )
        gKernelVersion = 2;
}
if ( !gKernelVersion )
    return 0xC00000BB;
LABEL_4:
memset((void *) (v2 + 56), (int) DriverDispatcher, 0x6Cu);
result = createOurDevice((PDRIVER_OBJECT) v2);
if ( !result )
{
    v5 = KeRaiseIrqlToDpcLevel();
    v6 = FindSystemProcessActiveLink();
    gSystemProcessActiveLink = v6;
    gProcessNameOffset = getProcNameOffset(v6);
    gPsLoadedModuleList = FindPsLoadedModuleList(v2);
    gKeServiceDescriptorTableShadow = findAddressofShadowTable();
    if ( gKernelVersion == 1 )
    {
        FindKiWaitInOutListHead(&gKiWaitInListHead, &gKiWaitOutListHead);
        gKiDispatcherReadyListHead = FindKiDispatcherReadyListHead();
    }
    else
    {
        if ( gKernelVersion == 2 )
            FindXPKiWaitListHead(&gXPKiWaitListHead);
    }
}
```

Планировщик, лист модулей ядра, лист EPROCESS, поиск таблицы сервисов через KeAddSystemServiceTable. Вот и все ребята. В качестве

имени процесса, берется то, что `ImageFileName` в `EPROCESS`. И это не меняется уже на протяжении нескольких лет, собственно с самого первого нашего знакомства с этим агрегатом его функционал расширился только на пресловутый Партизан. Кстати, абсолютно наивно считать его панацеей от руткитов, неуважаемый Дмитрий. После раскопок в Партизане выяснилось, что он это всего лишь реализация удаления файлов и ключей реестра через `Native API` во время загрузки `Windows`. И больше ничего, совершенно ничего. Пустышка, как и все остальное в этом убогом платном антирутките.

В новой версии программы аффтар сменил старый Аспротект на новый Аспр 2.1 SKE, тем самым видимо посчитав, что решил проблему с защитой своей недоинтеллектуальной недособственности. Вероятно, Дмитрий не знает, о том, что Аспротект сейчас не умеет снимать только полный идиот, и давно существуют автоматизированные распаковщики творчества Солодовникова (включая эту отдельно взятую версию), как в паблик, так и в андеграунде.

Не удивляйтесь, если после установки УнХакМы операционная система начнет дико тупить на старте – это сервис агрегата думает над своей незавидной судьбой посмешища Интернета.

По-прежнему радуется справочный файл УнХакМы, теперь там добавились записки сумасшедшего об Unreal! Вот какая слава! Мы польщены. Вот что там написано, цитата:

1. Unreal hides his body to the NTFS stream: `c:\unreal.sys`. The file path and name may be changed.
2. Unreal hides the working driver from the drivers list.
3. Unreal terminates checks for the known anti rootkit software and terminates some of them to prevent detection and removal.
4. Unreal doesn't use the processes.
5. Unreal doesn't hide its sub-key under "Services" registry key. But it can restore deleted registry key on shutdown.

По пункту один аффтар отжог, других имен и путей быть не может, они, что называется `hardcoded`. По пункту три веселье продолжается, на момент публикации никто не мог удалить Unreal. А даже найдя его через нотификаторы. Что касается последнего пункта, ну что же мы в отличие от этого великовозрастного далбаеба можем доказать обратное сделав `copy-paste` с исходников руткита, но это надо? Не восстанавливает он ключ, и никогда не восстанавливал. Впрочем, что-либо доказывать этому отдельно взятому идиоту нет никакого смысла.

p.s.

Совсем недавно появился руткит AK922, профильтровавший буфер с данными, чтобы обойти детектирование через RAW чтение. Аффтар УнХакМы подсуелся, и тут же добавил мнимый детект этого наполовину рабочего концепта в свой агрегат. Каким образом он его нашел, спросите вы? Через записи PoC в реестре.

Вердикт: Соколов, спаси плонеду, убейся вместе со своим детищем.

Приложение 1 В тени Русток

Русток... как много сладкого для сердца спамера слилось в этом слове. А как много отозвалось! Мифический и неуловимый. Новая модель. Новый путь, новые технологии. Много вкусного. Много интересного. Компиляция PoC'ов и переработанных исходников. И ещё много чего другого. Действительно лично мы не помним ничего столько же значимого и глобального со времен, пожалуй, CIH. Но если то был деструктивный стелс - вирус, собранный из скомунизденных исходников, то это троянский конь с не меньшими стелс возможностями и куда большей отдачей, чем просто удовлетворение от тотальной деструкции. Редкие программы обрастают таким количеством мифов и слухов. Например, многие до сих пор не могут определиться с нумерацией версий, и была ли там вообще нумерация. Мы не хотим сливать автора в ФБР и давать вам масштабные разъяснения (а мы можем). Мы хотим всего лишь восстановить хронологию некоторых событий, и предоставить вам дальше думать самостоятельно. Впервые об этом трояне мы услышали в марте-начале апреля 2006, когда собирали материалы для разработки Rootkit Unhooker. Собственно тогда он назывался даже не Русток, а был просто трояном с драйвером. Нам показалось это очень интересным, и мы занялись поиском других похожих образцов. Первыми были найдены sysbus32.sys и i386.sys, потом ещё несколько. Уже тогда проскакивала информация о возможном авторстве этих руткитов, и мы занимались активными исследованиями на этот счет, просматривая многочисленные жутко засранные топики на форумах, таких как, например <http://wasm.ru>, <http://cracklab.ru>. Практически сразу круг подозреваемых сузился. Не секрет, что в настоящее время практически все форумы, ранее очень известные из-за качества содержимого теперь превратились в помойки и откровенные клоаки. Но даже среди гавна иногда можно отыскать одну или несколько роз, хотя розы не растут в гавне. Попадались ещё образцы руткитов, очень похожих по функционалу и самое главное – подчерку кодера (не смейтесь это действительно так, как правило, высококлассных программистов можно узнать по их коду) и строкам отладки, ведущим на разные диски, но в одинаковые по смыслу названий директории. Будучи не особыми любителями детективных историй, мы просто постоянно сами попадали в них. Наконец был найден руткит, который потом станет, известен как Rustock.A, название придумали аверы Symantec, но как мы поняли, автору руткита оно пришлось по душе потому как, дальше в разговорах он всегда называл свою работу именно Rustock и никак иначе. Ну, или возможно это было сделано, чтобы нам было более понятно. Много ненужного внимания оказывалось и со стороны, потому что это был едва ли не первый руткит, настолько продвинутый руткит применяемый во вредоносных целях. Потом появился Rustock.B, более продвинутая вариация на тему A, носившая внутреннее название v1.2. Он привлек внимание ещё и потому что содержал специфический код для некоторых антирутокитов и антивирусов. Мы обнаружили его практически сразу, остальным потребовалось достаточное количество

времени, чтобы выработать необходимые методы детектирования. Потом был долгий перерыв в течение которого постоянно всплывали старые и промежуточные версии Русток, подаваемые некоторыми антивирусными компаниями как новые разновидности троянов. Можно как угодно относиться к результату творчества автора этой серии, но лично у нас он не вызывает ничего кроме уважения. Потому что он self made, точно такой же self made какими когда то были мы. Возможно, если бы судьба не распорядилась бы иначе годами ранее про нас сейчас бы писал кто-нибудь другой, про нас и про наши трояны. Но таковые мы никогда не писали, чтобы кто из идиотов с васма не думал. В любом случае каждый сам выбирает себе дорогу и нам абсолютно все равно, почему PE386 выбрал этот путь. Нас интересовало его творчество, код и совершенно было наплевать на этические аспекты. Из уважения к автору мы старались особо не афишировать свои знания и те результаты, что мы получили, почти полностью разобрав несколько его руткитов, включая неуловимый C вариант. Из тех же причин мы никогда не давали его руткиты в чужие руки и не собирались этого делать. Self Made, ребята. С третьей версией получилось вообще что-то невероятное. Толи предварительная подготовка дала такие ошеломительные результаты, толи код оказался насколько гениальным. Результат остается результатом – большинство до сих пор не знает о существовании этого руткита, и более того у них уже стойко выработалось неприятие самого факта его существования. А все, потому что ребятам (например, из некоторой почти русской почти антивирусных компании) просто не хватает как веры, так и жизненного опыта. Вероятно третья версия несколько раз перерабатывалась уже после так называемого официального релиза со слов автора, но тем не менее мы не оставляли поиски, попутно совершенствуя свой антируткит, исходя из предположений, что нас может ожидать. Как показала практика, мы оказались правы абсолютно во всем и встретили руткит подготовленными. На данный момент (весна 2008) Rustock.C не обнаруживается ни одним публичным средством поиска руткитов, включая даже некоторые приватные версии детекторов. И это после почти полутора лет с «официального» релиза. Достаточно большой срок, нет, он просто гигантский и показывает многое. В течение прошлого года нам периодически встречались упоминания об этом рутките, кое-кто даже продемонстрировал нам кусок его распакованного кода. И летом 2007 мы нашли его. На рутките оказался довольно интересный протектор, серьезно усложнивший и отложивший разборку руткита. А позже осенью автор дал нам специальную тестовую версию, которая оказалась более работоспособной. Мы сохранили оба варианта и так никому их и не дали, хотя подобные запросы были многократны. Стоит отметить, что именно Русток стал одной из причин преждевременного сворачивания публичной части проекта Rootkit Unhooker. Изначально планировалось поддержать проект антируткита до начала 2008 и возможно далее оставить как отдельную утилиту с другой командой разработчиков, набранной из тех людей, что помогали нам в течение этих двух лет, но после обнаружения третьего варианта Rustock нам уже нечего было искать. Со стороны может показаться, что эти два проекта

тесно связаны. Нет, они просто шли параллельно. Один прятался, другие делали детект. В результате получена простая аксиома – не существует недетектируемых руткитов и нет абсолютных детекторов. Простая истина. Теперь глядя на копошащийся муравейник всевозможных вендоров с их говноподелками, аффтаров с их антируткитами левой резьбы, флудерастов и демагогов в различных местах Интернета мы не можем не позволить себе улыбку, когда вся эта (яркая и веская часть слова удалена, примечание el666)братия не верит и не принимает факт существования вещей, которые просто вне их видимости и понимания. Хотя нет, возможно, что у некоторых из этих товарищей что-то подобное уже сидит дома на компьютере, спрятавшись в глубине их операционных систем ака **mustdie**, спокойно ждет подключения к Интернету или наступления зиродея. И совсем весело было наблюдать, когда люди съехали с катушек и начали видеть в окружающих призраки давно минувших дней и эпох. Причем они сами себя убедили в этом, не нужно было даже особо подыгрывать :) Достаточно просто не давать опровержения и чем дальше в лес, тем больше у них крепчает уверенность в том, что они правы и вот она истЕна (а стало быть, и правда) у них перед глазами! Подобные стада достаточно легко поддаются манипуляции, разумеется, на что-то серьезное тут сложно претендовать, но принцип сарафанного радио никто не отменял. А лечение от этого есть - статья, называется просто и незатейливо так: «Элита и хуесосы».

Но может это и к лучшему. Ведь сколько можно утащить из подобных троянов и привинтить к своим недоботам, чтобы потом вальяжно распальцовывая (при это аффтар пять лет, как вылез из памперсов) писать о ботнет сетях в **WASM.HEAP** или там же расписывать (вероятно, находясь непосредственно под действием) какая трава и какие препараты лучше вставляют, и позволяют избавиться от жидохэккерской депрессии. Или в ужасе рыдать о том, как его кинул злодей кодер, которому поручили собрать из этих кусков очередной зиродей вчерашнего дня. Отечественное сообщество тихо съехало с катушек, потому что перестало воспринимать такое как ненормальное поведение и смирилось. Уж лучше пусть будет один башковитый злодей, чем огромная стая диких безумных шакалов. Технологии должны быть закрытыми, информация больше не может быть свободной для общественности. Время розовых очков прошло. Зомба заberi их всех обратно и удави, кого сможешь.

- Grab it :))

- Yo, I love Russia! ~666

....

- Hey! It doesn't working! U piece of scam!! Come back mazafucker!!

BIOSKit существуют, **Bootkit** существует, сейчас разрабатывается **Hyperkit**, и точно также существует **Rustock**, вне зависимости от того верите вы в него или нет.

Приложение 2

История одного буткита

Звоночек поздно ночью. Уставший Варлок логится в программу «все ищут меня» и с некоторой долей неприязни наблюдает знакомый ник, назовем его, скажем Вася!

Вася!: - помоги плизз...

Варлок: - Че те надо опять?

Вася!: - мне нада написать такую фигню

Варлок: - Ну вот и пиши, иди на хер кароче. Ставлю в игнор!

Вася!: - подожди!!! возьму в долю!!

Варлок: - Что тебе надо написать?

Вася!: - Руткит! Он должен прописываться в бутсекторе жд! при включении пк запускаться и передавать управление ядру...

Варлок: - ппц, иди на форуме спроси. Игнор бля.

Появившийся в конце 2007 года и ставший достаточно известным ближе к концу января 2008 первый буткит (на самом деле это комбинация буткита и руткита) под NT оказался очень интересным событием, пожалуй, самым интересным со времен эпопеи Rustock. Его появление можно было предположить, потому, как первые концепты были продемонстрированы EEEYE ещё в далеком 2005 году. Однако долгое время появление буткита считалось юмором. Это и остается юмором сейчас, когда мы можем посмотреть на реализацию этого буткита вживую. Почему же выбран именно буткит? Ответ на этот вопрос лежит в концептуальной плоскости, во-первых, эта технология стара как мир и хорошо известна со времен MS DOS. Во-вторых, она позволяет отказаться от таких ненадежных и легко детектируемых вещей как скрытые ключи реестра и файлы. В третьих у буткита есть эксклюзивная возможность работать до загрузки операционной системы, то есть до того момента как всевозможное защитное ПО будет даже инициализировано на старт. В-четвертых, руткиту, который иницируется буткитом, будет проще контролировать целостность компонентов, потому что всегда известно, где и что нужно защищать и скрывать, от него не требуется собственной реализации файловых систем и прочих глубоких познаний.

Давайте посмотрим, как же это реализовано в MAOSBoot или Sinowal.C. Загрузочная запись за основу взята от EEEYE, оно и понятно, зачем придумывать то, что уже придумали до тебя. Аффт(ар) изменили лишь часть относящуюся к патчу NDIS.sys, вместо него теперь патчится ядро Windows. Буткит перехватывает Int 13 для того, чтобы контролировать загрузку данных NTLDRом и патчит IoInitSystem на свой код, который иницирует считывание и загрузку драйвера руткита (свой загрузчик), который в свою очередь располагается в последних секторах жесткого диска не как файл, а просто как данные. Фактически мы наблюдаем с вами загрузку Windows из-под буткита. То о чем год назад мы писали на одном из форумов, все-таки было реализовано и, причем реализовано в трояне, какой позор. Однако аффт(ару) все же следовало внимательнее читать наши посты, потому как там же были объяснены и все возможные уязвимости используемого метода. Но об этом далее. Для защиты руткит применяет следующий трюк – он перехватывает Major функции disk.sys простой подменой адреса. Перехваты устанавливаются

как на чтение, так и на запись. При попытке прочитать данные из главной загрузочной записи, руткит подсовывает оригинальную mbr, сохраненную в секторе 62, который тоже под защитой руткита. В секторе 61 хранится часть загрузчика буткита со строкой `\??\PhysicalDrive0`, это необходимо для загрузки руткита, конкретнее для его считывания с диска. Сектор также защищен от чтения и записи. Дропер буткита способен заразить до 16 дисков за раз в цикле, это сделано, видимо от болды, чтобы не заниматься определением количества реально подключенных дисков. Руткит не существует как драйвер только как исполняемый код. Нет ни файлов, ни ключей, ни боже упаси, процессов.

Этот простой пример руткита слил почти все антируткиты и все антивирусы. Помните, вам рассказывали сказки о вашей безопасности и как вы secured своими бсодогенераторами антивирусами? Как оказалось компании, занимающиеся вашей безопасностью до сих не могут совладать с этим трояном. Его новые разновидности словно издеваются над ними, играя с перехватами и разбавляя загрузочные записи мусорными командами. Команда антивируса Касперского, конечно, может гордиться своим (наконец то пропатченным) продуктом, но если бы не лажовый инжект в Проводник для самоудаления дропера, у их антивируса врядли чтобы получилось с этим буткитом. В какой-то момент нам показалось, что вот и все, руткит положил на лопатки всех, даже хваленный GMER. Бедный поляк Пржемуслав так любящий пиарить сам себя на своем сайте, выкладывая подробные описания буткита и того, как он собрался его детектировать, заткнулся и где-то неделю от него не было никаких вестей. Но пора понять, что буткиты это в принципе плохая идея, рассчитанная примерно на то, чтобы быстро срубить бабла и свалить пока все не открылось. Прошло совсем немного времени и последовали ответные меры, GMER снова прозрел, появились новые версии антивирусов и антируткитов от антивирусных компаний (Avast, BlackLight, TrendMicro), способные обнаружить и иногда удалить буткит. Самый первый релиз GMER использовал `cdrom.sys` для того, чтобы прочитать оригинальную MBR. А вы знали, что Major функции `IRP_MJ_READ/IRP_MJ_WRITE` у `cdrom.sys/disk.sys` ведут в одно место? В `classnp.sys`, собственно он и ставит эти обработчики во время инициализации функцией `ClassInitialize`. Теперь игра переключилась в плоскость, кто глубже хакнет. Новый релиз GMER использует этих целей своей собственный лоадер с помощью которого он грузит считанный с диска файл драйвера и находит там адрес оригинального обработчика, после чего все это дело отправляется в драйвер антируткита и там вызывается самым незатейливым образом.

Отдельно стоит отметить то рвение, с которым некоторые антивирусные компании и господа «прикрывающиеся белыми шляпами» (с) Лаборатория Касперского, принялись мусолить буткит, а именно выкладывать в паблик длинные листинги дизассемблированных команд, дампов, логов, дампов логов и логов дампов. Особенно отметился здесь Пржемуслав Гмерек на своей хоупаге, созданной в Ms Word, расписывающий свои достижения с lkd и дизассемблерами. Подробно расписано, как и что буткит делает, едва ли не реконструированный исходный код. Расписано специально, чтобы облегчить задачу написания

обхода. Также не забудем о парнишке Марко Джулиани с его дурно пахнущей репутацией. Мы не включили в обзор детище PREVX -> недоантируткит PREVXCSI именно по этой причине. Отметился даже красный гигант McAfee, чьи лажовые поделки были неспособны удалить буткит на момент написания этого обзора. Действительно странная ситуация, все вокруг так и стараются помочь аффтарам буткита, а заодно и попиарить себя и свои продукты. BlackLight от F-Secure был впервые за долгие годы сколько-нибудь обновлен, чтобы найти и показать инфицированную MBR. На этом его и его аффтарах возможности были исчерпаны. А то, что на диске полным полно ещё скрытых секторов этот древний и совершенно бесполезный антируткит не имеет ни малейшего представления. Антируткит от Trend Micro пошел по той же дорожке. Причем, несмотря на уверения одного из его соавторов господина Cardmagic'a (да-да, тот самый аффтар DarkSpy), мы так и не поняли как с помощью этого можно сдампить скрытый сектор. Возможно, нужно было вогнать его антируткит в дизассемблер, и внимательно изучив код, задействовать какую-то скрытую команду командной строки? Ей богу очень понятно и юзабельно.

Что самое забавное – сейчас ясны абсолютно все ходы любой стороны, причем аффтар(ы) буткита будут ходить черными. Это патовая ситуация. Буткит кончился, чтобы аффтары там не предприняли. Вы всегда знаете, где, что искать и что с этим, потом делать. Нет никакой загадки, и даже лажовый навесной пакер больше не спасает.

А с системой автоматического обновления этой малвары вышел вообще казус мирового масштаба. Аффтары сделали очень удобно не только для себя, но и для всех, кто хочет этот буткит обнаружить. Новые, горячие скомпилированные версии доставляются, что называется сразу на операционный стол.

Вопрос с буткитом заключается только в одном, кто глубже хакнет и когда аффтару(ам) буткита это надоеет.

- ебя проститутток не ломайте диван

Приложение 3 Пишем платный антируткит

Глядя на то, что происходит в течение последних нескольких лет, просто диву даешься, какой кошмар. С одной стороны почти безумные китайцы с их маниакальной страстью к написанию чего-нибудь с драйвером, с другой толпы говнодевелоперов разных мастей, а некоторые из них ещё умудряются срубить на этой непаханой ниве бабло. Уже больше не удивляет некомпетентность и тех и других. Если взять и покопаться в современных антивирусах, то можно найти ещё больше идиотизма и кривизны, чем во всех китайских антируткитах (включая те, что одеи) вместе взятых. Планка качества действительно резко рухнула вниз и пробила пол. О чем можно ещё говорить в век почти победившего C#, недоязыка для полных далбаебов.

Поэтому не будем больше пустословить и давайте рассмотрим современный и вполне реальный пример, допустим, вы, или ваш знакомый или знакомый вашего знакомого или его сосед, решили написать что-нибудь с драйвером (ну блять без него же никак). Зачем? Чтобы потом срубить 24.90\$ как аффтар Spyware Processes Detector или запатентовать свою хуйню как аффтары Helios'a, либо поступить в антимальварную компанию или чтобы просто банально повыее... (но такое жуткая редкость, сейчас все умеют считать деньги, правда, не все могут взять от числа процент). С чего же начинают нынешние Дмитрии Соколовы и прочая шалупонь. Это уже стандарт де-факто, задумывая подобное аффтары, как правило, имеют смутные представления не только, о том, что они собираются сваять, но и о языке программирования на котором они это собираются реализовывать. Подобной фигней страдают, например некоторые отечественные аффтары, которые начинают учить C++ раньше русского языка и с таким же отсутствием явных успехов. Несомненно, на этапе проектирования (вообще-то половина даже не представляет что это такое нафиг), определяется то, что должно быть в этой программе, а что ей не нужно. Для того чтобы определиться с тем, что же должно быть, как правило, используются поисковые системы (ну это у продвинутых), а в целом и общем помогает программа «все ищут меня». Дело в том, что в любом мухосранске обязательно найдется какой-нибудь, скажем, Варлок у которого можно что-нибудь спросить, если сам них.. не умеешь. Варлок этот отличается от тех, кто спрашивает тем, что умеет пользоваться поисковой системой типа Гугль, ну или той, где все теряется.

Поиски по ключевым фразам приводят на всякие помойки, тематические сайтенги, давно почившие в бозе, а также на прогнившие сайты - форумы типа WASM.RU, где всегда тусуется куча народу, которому нехуй делать, кроме как срать друг с другом по поводу ботнет сетей и искусственного бля интеллекта (ну это вообще 3.14дец). Здесь будущему аффтару помогут с реализацией его мечты, но сначала его похуесосят, типа – «RTFM», «Гугль в зубы», «поиск по форуму рулит» и прочая поебень. Что примечательно те, кто подобное советуют, как правило, сами полные далбаебы и вполне возможно совсем недавно

тоже что-нибудь где-нибудь спрашивали и преодолевали подобную обструкцию. И вот аффтар уже освоил экскрементальный оператор C++ (не смейтесь, это реальный случай, когда один из аффтаров бил себя в грудь и доказывал существование подобного), набрался опыта и чужих исходников и вот он уже готов компилировать. Но едрена вошь, Visual Studio как-то не дружелюбна и не хочет компилировать вкусные драйверы, а с DDK приключается полный абзац. Аффтар тут же бежит на форум, где в панике создаются топеги типа «Помогите настроить», «Не компилируется» и т.п. и т.д. Напомним, что спустя некоторое время этот же аффтар будет рьяно просить деньги со своих несчастных пользователей за свое, рожденное в таких муках говно с драйвером (а то и не одним драйвером).

Сделаем допущение об успешности предыдущей стадии, т.е. аффтар приступил к компиляции. А вот здесь начинаются другие проблемы. Вкратце их можно описать как некомпетентность в системном программировании. Дело в том, что если ранее кривые ручонки аффтаров могли привести к синему экранчику в очень редких случаях, и в основном приводили к локальным ошибкам или если аффтар засунул все в seh, причем по несколько раз, как например, сделал аффтар одной известной антиспайвары, такие ошибочки как бы за ошибки и не считались. Да, они, безусловно, слышали что-то о системном программировании, и быть может даже немного представляют себе, что это такое... сидя и шепотом обсуждая это на кухне со своим соседом. Что же делать, какие-то бсоды и не фига не понятно, что дальше. Здесь поступают кто как, некоторые возвращаются к предыдущим параграфам на новую итерацию прокачки чужими мозгами, аффтаритетом и желательно исходниками (и да бля главное, чтобы они сразу компилировались, прямо, как только открываешь в блокнот). Итераций здесь может быть хуева туча, вплоть до бесконечного loop'a.

И вот спустя долгие недели (месяцы) программа научилась не ложить систему в синьку (ну это ещё зависит от фазы Луны) и демонстрирует вам свое убогое VCL-ное GUI ака сами знаете что. Теперь к ней можно прикручивать всякие никому нахуй не нужные фишки исключительно для changelog, чтобы потом более обосновано трясти бабки с новых пользователей. Кроме этого можно привинтить к этому недоразумению скины! Да реальная вещь, без скинов сейчас просто никуда. Ты полное ничтожество, если твоя программа не поддерживает скины минимум как у Winamp'a. Ну, здесь, слава Богу, не все аффтарты антируткитов увлекаются этим (а вот платные программы и если считать антируткитами антивирусы, антиспайвары с таковыми модулями то на 90%).

Все, аффтар или аффтарты герои. Их продукт жизнедеятельности и компиляции скомунизженных исходников (привет Гмерек) сияет своим фейсом, а за мнимой аффтарами красотой и мощью скрывается тупое бессмысленное животное типа «баран». А вот и первые слова: «А ну сцуки гоните монету, мы защитим вас!» От чего не уточняется, но и так понятно.

Один из этапов пропущен. Дело в том, что он может иметь место практически в любой момент. Можно конечно толкать свое говно за

24.90\$, но вдруг аффтара в программу «все ищут меня» или в ПМ на форуме постучит дружелюбный дядя и предложит быстро разбогатеть. Он ещё скажет, что верит в аффтара и у него обязательно получится сделать, то, что он предлагает наваять, как правило, это «что-то» со «своим TCP/IP стеком» и «морфингом и полным стелсированием».

А спустя полгода, разбирая очередной зиродей вчерашнего дня, на который обязательно навешают какую-нибудь фигню типа «протектор», кто-нибудь удивится кривизне и скомунизденности кода, что окажется перед ним. Правильно, нахер что-то придумывать, все, абсолютно все ведь уже придумано до нас и в мире всего семь нот.

Вы думаете это просто болтология ничего не имеющая с реальностью? Ан нет. Достаточно взять замусоленный Spyware Processes Detector (оставим уже убогую УнХакМу в покое) и восстановить хронологию его производства. Давайте возьмем цикл разработки первой версии за один год, потому что именно такое значение мелькало в базаре (мы по-другому назвать этот лепет не можем) главного помощника главного и похоже единственного аффтара этой поделки. Так как разработка велась усиленными темпами (опять же с того базара), предположим, что аффтар тратил на написание этой хуйни по паре дней в неделю при пяти часах ненатужной работы, типа попил пивка, написал be, сходил, отлил, покурил, дописал gin. Больше он врядли тратил, так как есть ещё другие проекты и собственно жизнь молодая.

1 год = 365 дней / 7 дней = округленно 52 недели.

Количество часов затраченных на работу = (52 недели * 2) * 5.

Итого 520 человеко-часов. Разумеется, здесь не учитываются праздничные дни, так что оценка приблизительная, но нафиг нам тут что-то ещё? При цене в 24.90\$ возьмем приблизительный усредненный курс за 2006-2007 год, когда все это и «разрабатывалось». Допустим, наш курс будет равен 25 рублям. Соответственно один час разработки стоит $(24.90 * 25) / 520 = 1$ рубль 20 копеек.

Может быть, стоило скинуться всем миром и накинуть ему немного сладких вбз, чтобы не дать появиться на свет этому чудовищу?

А хуй знает. Подобные примеры можно привести и с кучей другого платного «софта с драйверами», который занимается «обеспечением безопасности» бедных пользователей, а на деле просто делает бесконечные циклы с нопами и разбросанными дрожащей рукой аффтара Sleep'ами.

Ха, думаете, это только с вот этой отдельно взятой группой программного обеспечения так? Уверяем вас, таких примеров масса, достаточно просто взять да потрясти те же HIPS да фаерволы, не говоря уже об антивирусах.

Не пишите платные антируткиты, это никому нахуй не нужно.

Заключение

Спустя год наблюдается качественное изменение обстановки. Если год назад можно было констатировать всеобщее убожество и засилье китайских бсодогенераторов, то сейчас в лиге значительно поубавилось народцу. Некоторые как аффтарты **DarkSpy** ушли работать в **Trend Micro** и теперь ваяют их антируткит (а уж там не дадут создать абсолютный бсодоген), некоторые просто канули в лету как аффтар **SysProt**. Кое-какие проекты были официально завершены или перешли в другое «измерение», не доступное широкой публике, где и продолжают успешно развиваться, меняя версию за версией. За год появились несколько новых, интересных и перспективных игроков, подробно описанных в этом обзоре.

Произошел прогресс и с пресловутыми **HIPS**. Малвары научились эффективно противодействовать им и даже выносить их нафиг. Наши поздравления, наконец-то свершилось то, о чем говорили с самого начала и этот бесполезный софт доказал свою дырявость.

Антивирус Касперского 7 так и не стал концом хэккеров, как мы и подозревали, подводя итоги предыдущего обзора. Ну, теперь нас ждет восьмая версия, в которой господа, несомненно, хукнут ещё чего-нибудь в довесок к уже имеющемуся арсеналу заблуждений. Это, конечно, будет названо новыми технологиями (Нано? Нынче подобный идиотизм в моде) и прочими ничего не значащими и не имеющими к реальности никакого отношения маркетинговыми словами. Невдомек господам, что ту же **HIPS** можно построить без использования всего этого кг/ама, и она будет при этом на порядок эффективнее. Остальные антивирусы откровенно положили все, что у них было на противодействие руткитам. Стоит выделить только **DrWeb**, в котором, наконец-то наблюдаются какие-то позитивные изменения. Посмотрим, что у них получится. Отдельно стоит отметить аффтарты из **Agnitum**, что потерял основных разработчиков и теперь старательно корчит хорошую мину при отвратительной игре, выпуская один бсодоген за другим. Что касается руткитов, то здесь самыми яркими событиями были недотрояны **Srizbi / Accesso**, а последнее время просто звезда буткит в своих двух основных вариациях. Руткиты по-прежнему обходят все и заставляют подумать о будущем целых отраслей програпрома и не только отечественного.

Изменить здесь что-то можно только подойдя к проблеме концептуально. Затыкать дырки, ставя всевозможные убожества типа **HIPS** от разных подчас абсолютно невменяемых господ вроде Ильи Рабиновича (зацикленного на своей программе и детях-убийцах компьютеров) это иметь систему почище любой малвары. Нужны настоящие изменения в структуре и концепции операционной системы. Такие изменения не фантастика, и они произойдут, и будет это при нашей с вами жизни, на самом деле ждать осталось не так уж и долго :)

Rustock.C по-прежнему остается не обнаруженным и многие уже просто в него не верят. Ну что же, тем лучше для всех нас :)

About

Этот документ является собственностью UG (North Division) и любое использование приведенных в нем материалов обязательно должно включать ссылку на оригинал.

А не включите – найдем и башку отвертим :)



Авторы текста – EP_XOFF, UG North и el666, UG North
Under New Order 4 Ring 0xFFFFFFFF

© 2008 UG North

Все приведенные в данном документе названия могут быть зарегистрированными торговыми марками и / или знаками.

Мнение UG полностью совпадает с мнением авторов данного документа. Со всеми претензиями просьба обращаться к «рисунку номер один». UG не несет никакой ответственности за опубликованный материал.

«Рисунок номер один» не приведен в документе в виду своей ярко выраженной агрессивной составляющей.

Все опечатки, ошибки и неточности, допущенные в данном документе, являются умышленными и не подлежат обсуждению.

Вы ещё это читаете?

EOF