ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - ГАБРОВО

Катедра „Компютърни системи и технологии”
Дисциплина „Организация на компютъра”

Реферат

Тема: „Rootkit злонамерен компютърен код. Начин на функциониране. Съвременни алгоритми за откриване и отстраняване”

Иво Ангелов Ценов
КСТ, курс IV, гр. 1,
Фак. № 20705107

Разработил: ..... Проверил: ……

/ Иво Ценов / / доц. Р. Иванов /

Габрово
12.2010

Съдържание

1. Какво е rootkit?

Rootkit е софтуер, който позволява продължителен привилигирован достъп до компютър, докато активно прикрива своето присъствие от администраторите като разрушава стандартната функционалност на ОС или други приложения. Терминът rootkit се свързва с root (традиционният привилегирован потребител под Unix ОС) и kit, което идва от имплементацията на инструментите (комплект).

Обикновено атакуващият инсталира rootkit-а на компютъра след като първо е придобил достъп на най-привилегировано ниво (administrator, root), след като се е възползвал от уязвимост в системата или след откradване на паролата (чрез разбиване на кодировката или чрез социално инженерство). Веднъж след като rootkit се инсталира позволява на атакуващият да замаскира своите действия и да продължи да има привилегирован достъп до компютъра, като поддържа нормални верификации и верифициращи механизми. Rootkit-овете може да изпълняват най-различни дейности, те придобиват популярност главно като malware-и, които се прикриват и откradват паролите без знанието на администратора или потребителите на засегнатата система. Rootkit-овете може да атакуват firmware-а, hypervisor-а (още наричам регулировчик на виртуалната машина), kernel-а или най-често приложенията на потребителя.

Засичането на rootkit-ове е трудно, защото той може да разруши софтуерът, който е предназначен да го намери. Начините за засичане са използване на алтернативна ОС, която е „чиста”, методи изследващи начина на поведение, сканиране на подписите, сканиране на разликите и анализ на паметта. Премахването му може да бъде трудно или невъзможно, особено ако rootkit-а е в kernel-а, единствената алтернатива може да е преинсталация на ОС.

2. Типове rootkits

User Mode – “user mode” rootkit-ове са способни да работят на компютър с администраторски права, което означава, че могат да имат достъп до файлове, мрежови портове и системни драйвери. Те копират файлове на хард дискът, с което си гарантират стартиране всеки път, при пускане на компютъра. Тези rootkit-ове може да бъдат засечени и премахнати.

Kernal Mode – „kernel mode” rookit-ове са инсталирани на нивото на ОС, което им позволява да повлияят на ОС, което може да доведе до необясними събития. Те не могат да бъдат засечени от

потребителя освен чрез неочаквани събития и забивания или посредством антивирусна програма.

Firmware – firmware rootkit-ове са най-злонамереният вид, защото те могат да създават злонамерен код във firmware-а, докато компютърът е спрян. Всеки път като се пуска компютъра, rootkit-а ще се преинсталира. Firmware-а не може да бъде засечен от потребителя и е много трудно да бъде премахнат.

3. Как работят rootkit-овете

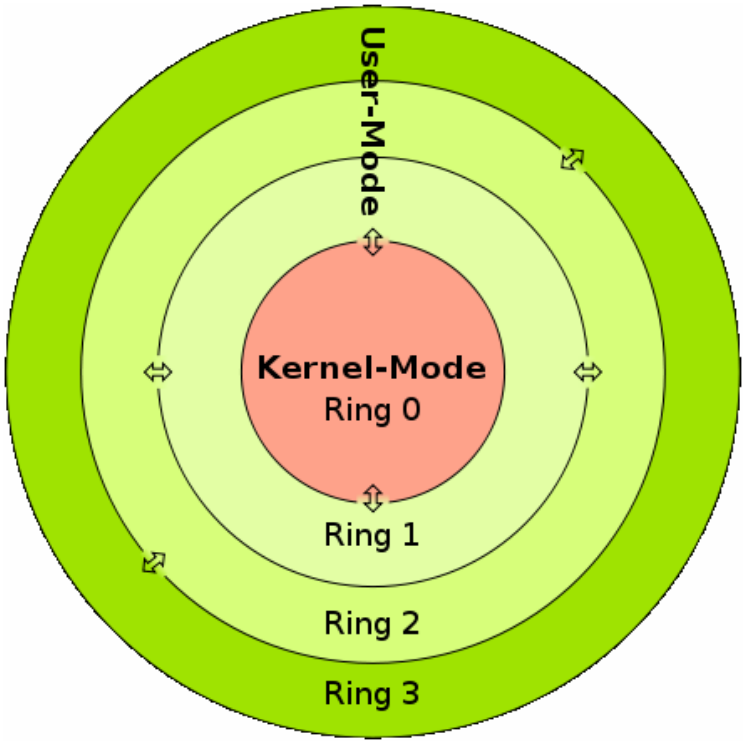
За да имат успех, те трябват да се скрият в системата, но за това първо трябва да си осигурят достъп. Да действат като „пазители" какво вижда потребителя и какво вижда системата. Изискват достъп до най-ниското ниво, както и администраторски правомощия за да се инсталират. За успешно скриване трябва да бъдат изпълнени следните условия:

- Да оцелеят след рестартиране на системата
- Да скрият процесите
- Да скрият услугите
- Да подслушват TCP/UDP портове
- Да скрият kernel-ските модули
- Да скрият драйверите

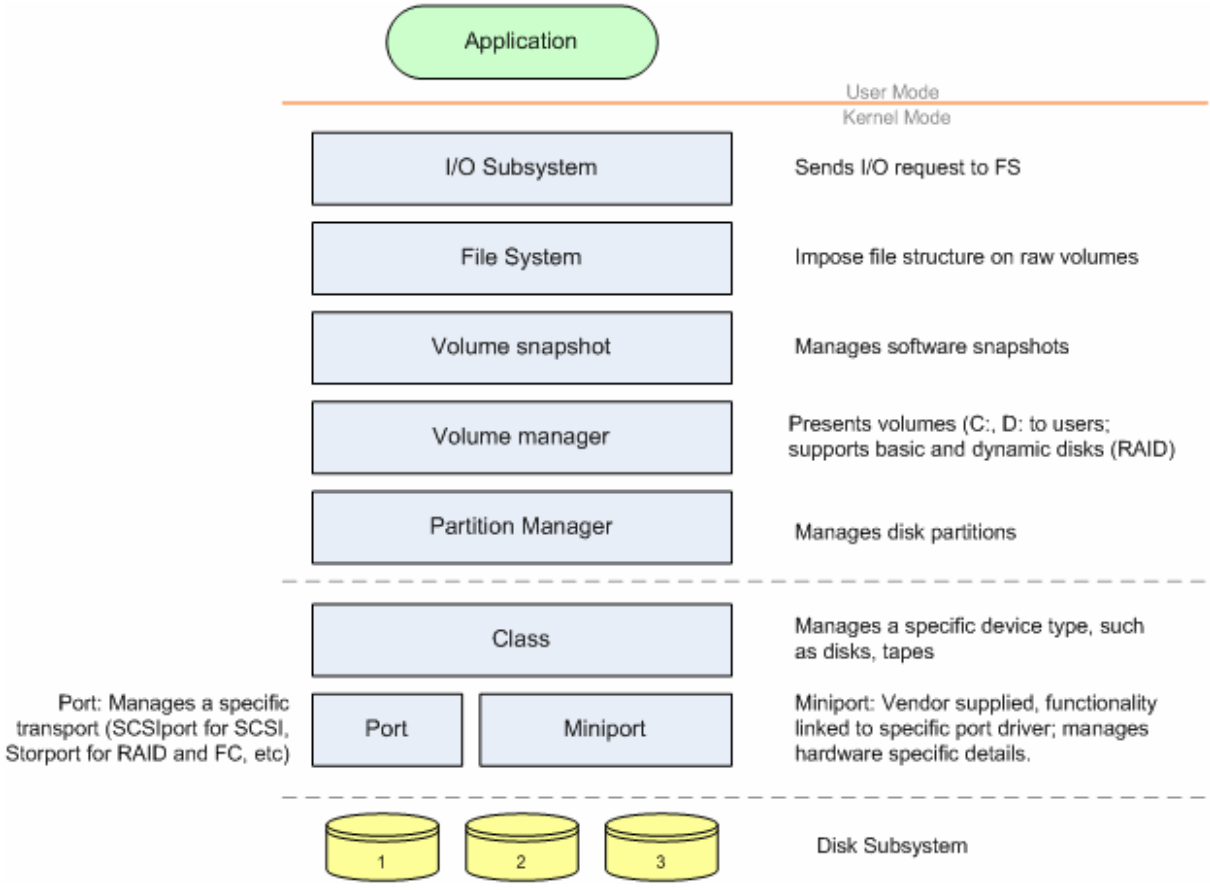
Нива на достъп при Windows

3-ти пръстен – потребителски
- Потребители
- Администратори
- Система

Нулев пръстен – kernel
- Драйвери



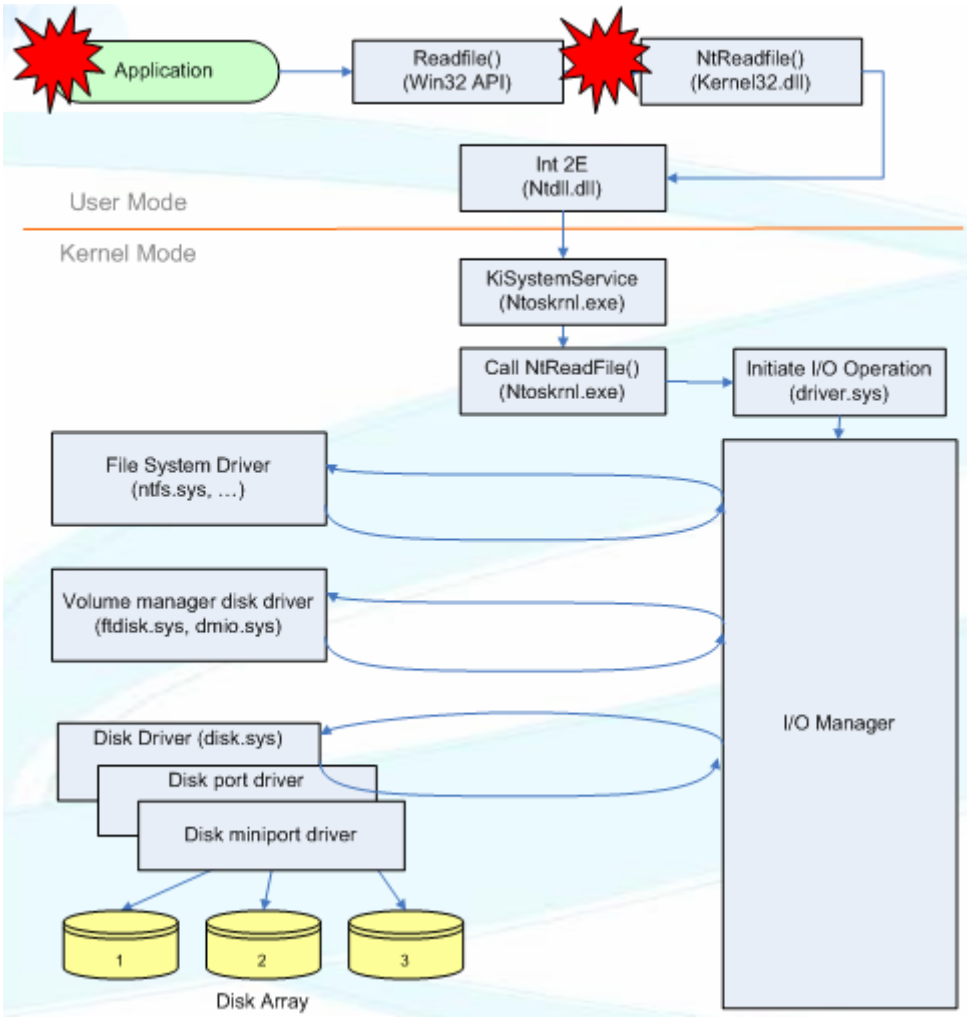
Фиг. 1 – Разположение на пръстените на ОС



Фиг. 2 - Схема на ОС без rootkit намеса

Rookits на 3-то ниво

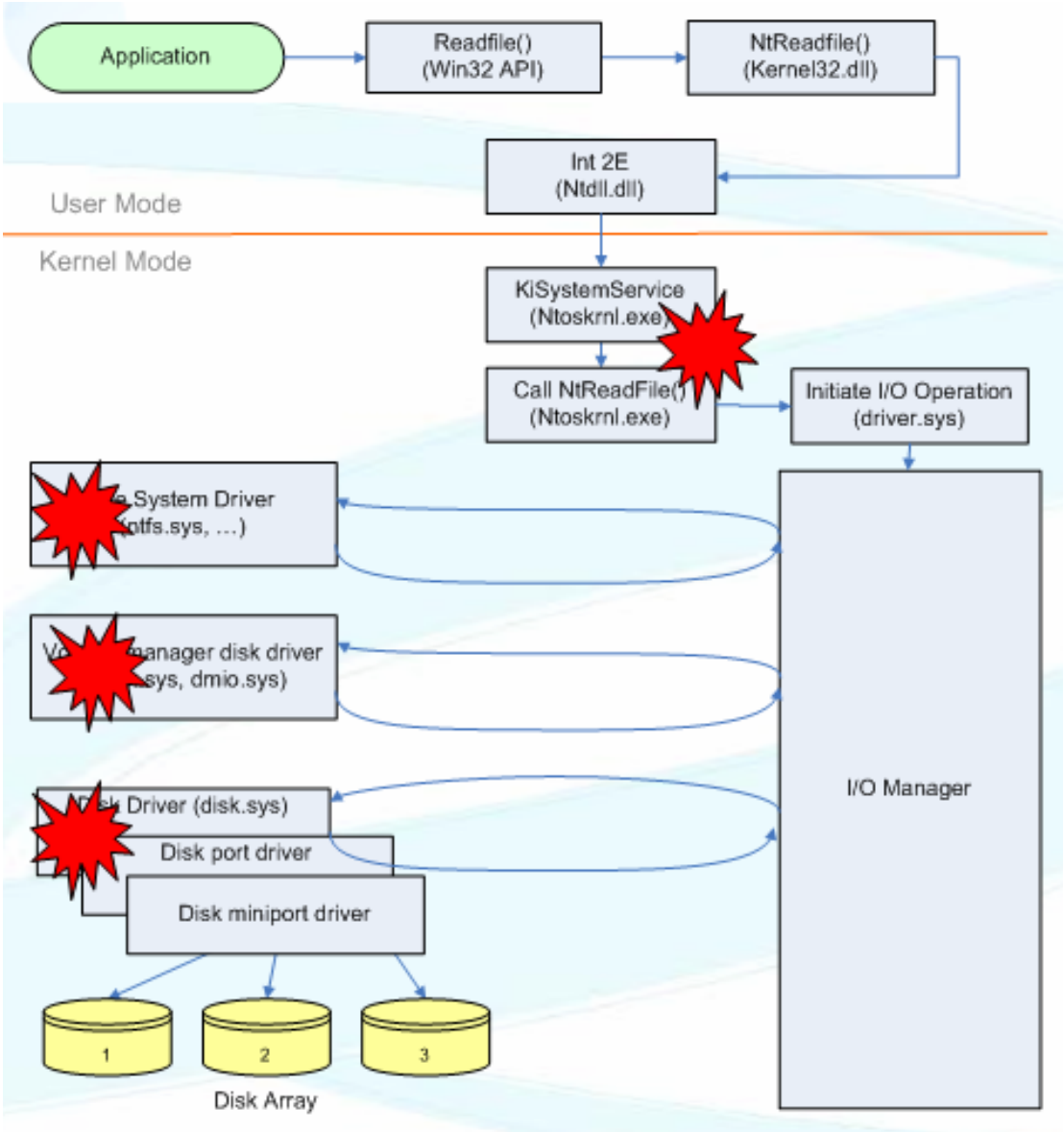
- Разменят бинарни стойности, т.е. променят .exe и .dll файлове
- Правят модификации в паметта
- IAT hooking



Фиг. 3 – Схема на ОС след rootkit намеса на 3-то ниво

Kernel rootkits на нулево ниво

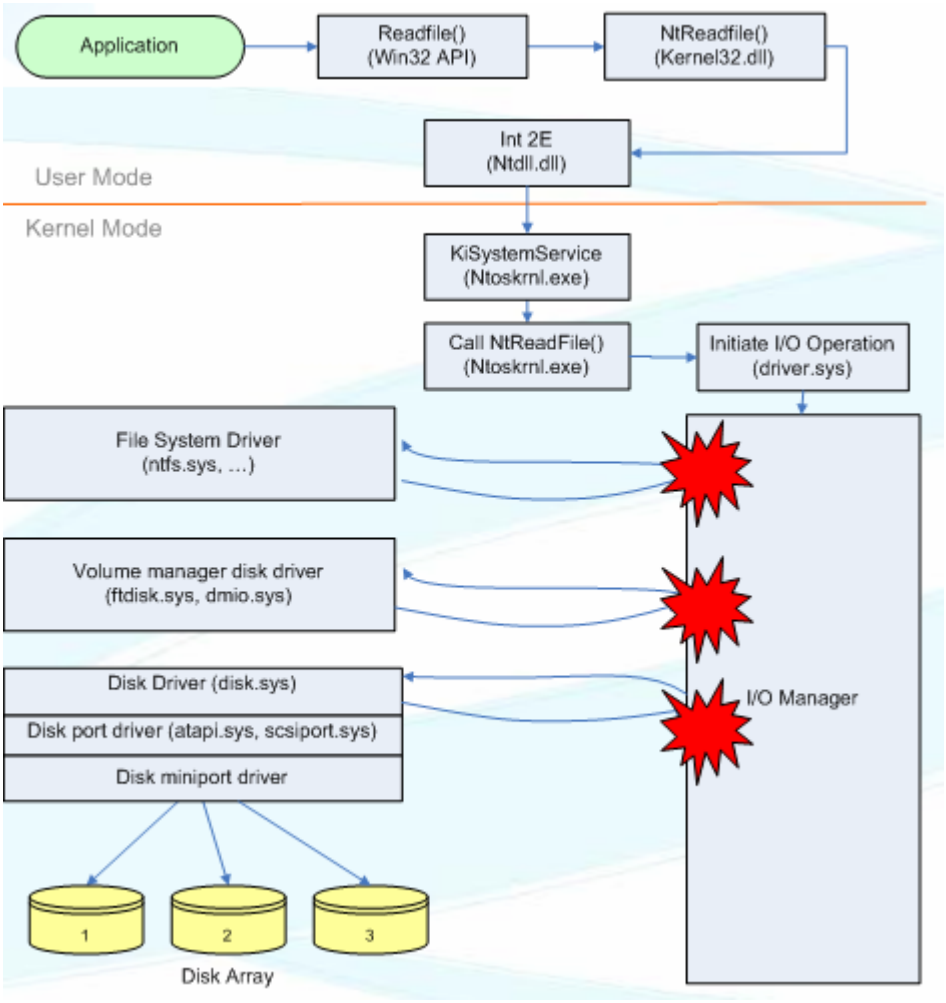
- Kernel hooking
- Смяна на драйвери (например разменя ntfs.sys с ntfss.sys)
- Direct Kernel Object Manipulation - DKOM



Фиг. 4 – Схема на ОС след rootkit намеса на нулево ниво

IO Request Packet (IRP) Hooking

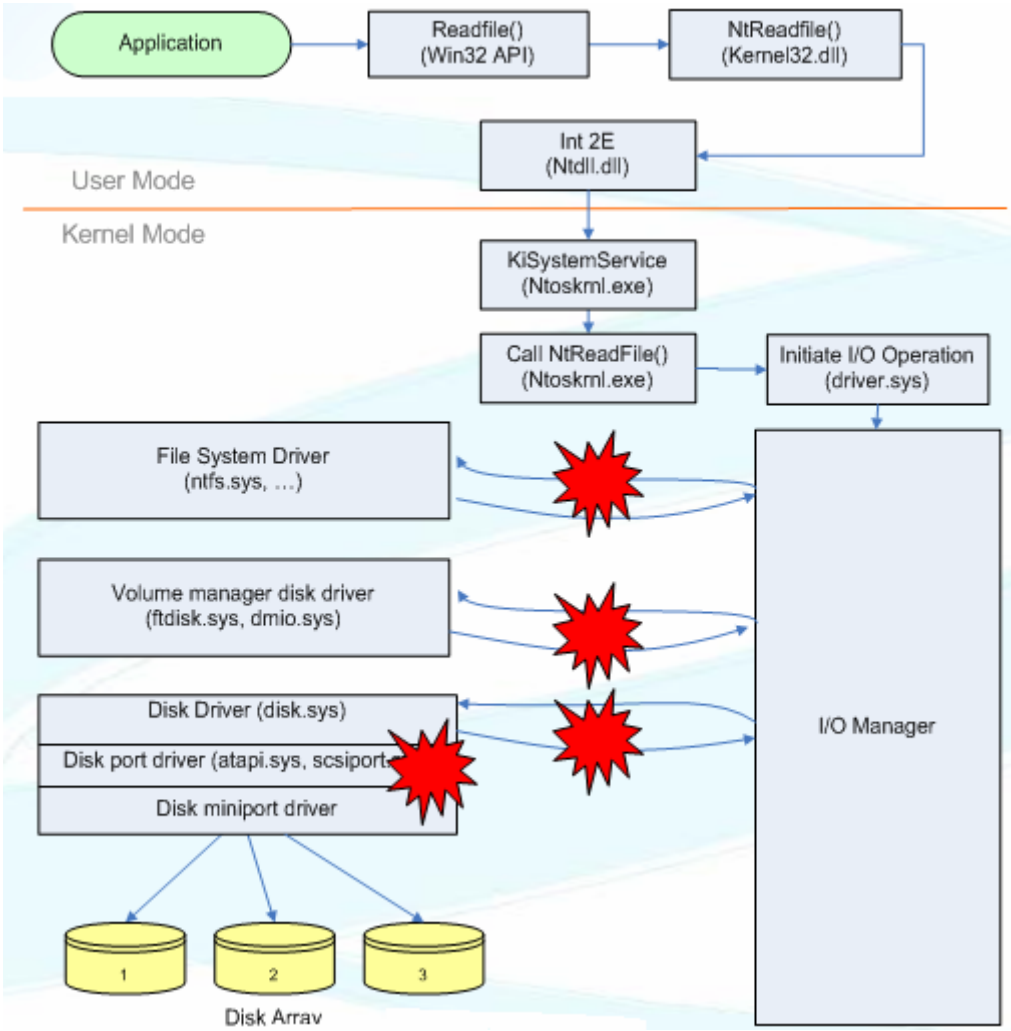
- IRP диспечерна таблица



Фиг. 5 – Схема на ОС след IRP hooking

Филтрират драйвери, като биват следните типове:

- Филтри на файловата система
- Филтри на дяловете
- Филтри на дисковете
- Филтри на шините

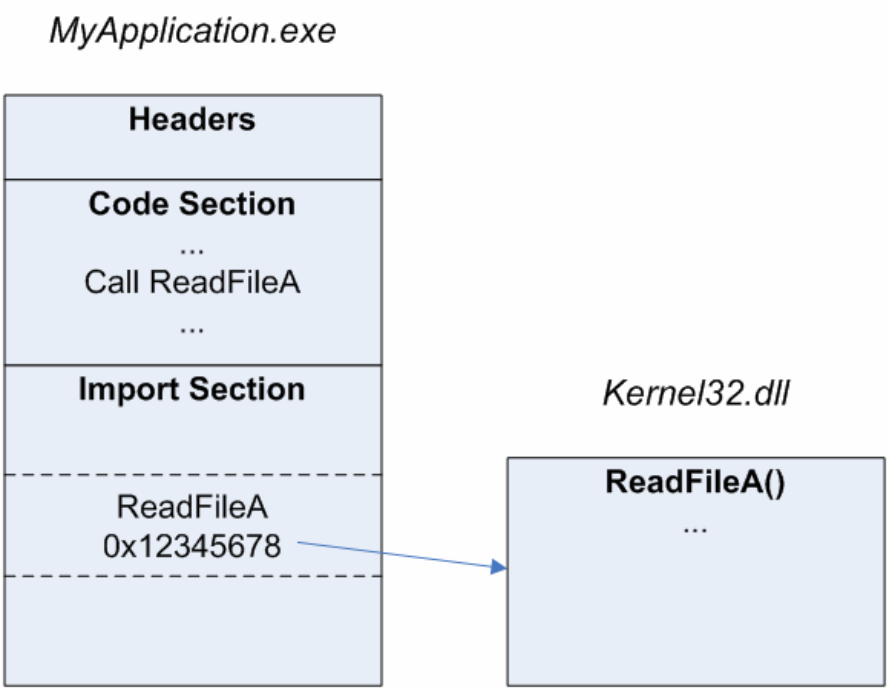


Фиг. 6 – Схема на ОС с филтриране на драйверите

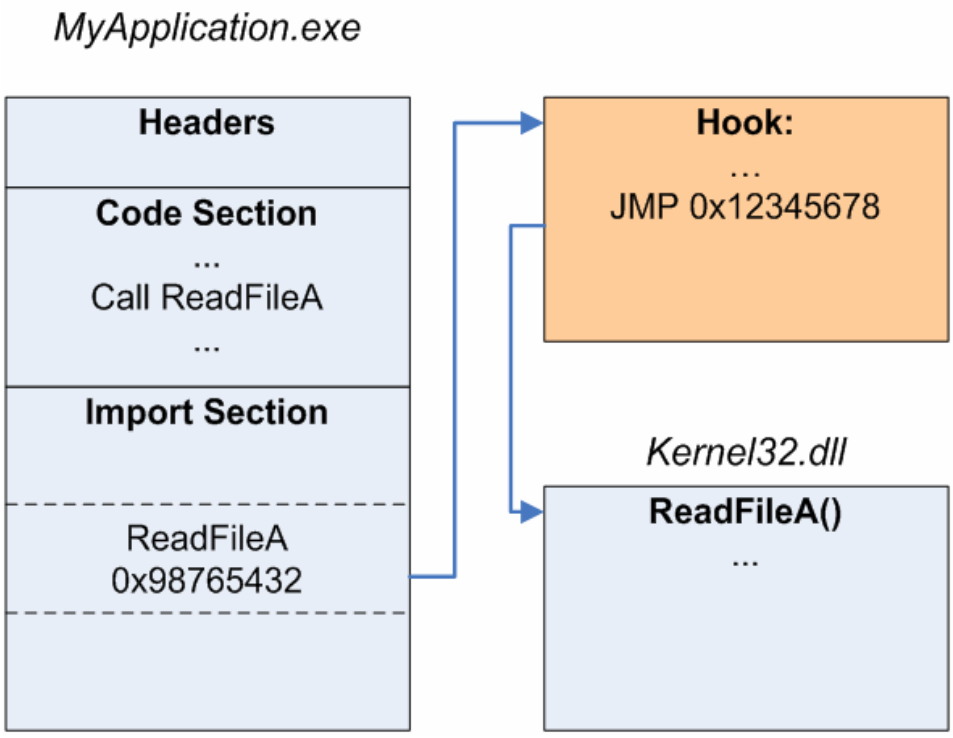
Съвременните възможности на rootkit-овете включват:

- Скриване на процеси
- Скриване на файлове
- Скриване на регистри
- Скриване на услуги
- Напълно заобикаляне на firewalls
- Не възможност да бъдат засечени от антивирусен софтуер
- Не могат да бъдат засечени дистанционно
- Прикриват каналите – не могат да бъдат засечени по мрежата
- Побеждават криптографската хеш проверка
- Инсталират се „без шумно” (не може да се усети)
- И всички други възможности използвани от вируси или червеи

Как работи rootkit hooking-а?



Фиг. 7 – Преди rootkit hooking



Фиг. 8 – След rootkit hooking

Хардуерни rootkits. Характерни особености при тях:

- Преинсталиране на ОС не ги премахва
- Трудно се премахват (обикновено устройството е разрушено)
- Трудно се имплементират
- Много трудно се засичат
- С повечето памет използвана в устройствата, те почват да се срещат все по-често
- VideoCardKit – съхраняват се във Flash-а или EEPROM-а
- EEye Bootroot – инсталира се през мрежово boot-ване от PXE

Има различни начини, по които може да работят rootkit-овете.

Spyware – по този начин rootkit-овете могат да променят програми или да ги заразяват със spyware. Spyware-ът, който е инсталиран от rootkit-а е труден за засичане, но се забелязвам странни неща, като линкове на работният плот и промени в уеб браузара.

Back door – е модификация, която е вградена в някоя програма на компютъра, която не е част от оригиналният дизайн на програмата. Тя създава скрити функции в програмата, които действат като подпис, така че нарушителят да може да използва програма за злонамерени действия без да бъде засечен.

Byte patching – байтовете са построени в специфичен ред, който може да бъде променен от rootkit-а. Ако се пренаредят, това може да доведе до компромис в защита, което да даде достъп на нарушителя за злонамерени действия.

Source-Code modification – този начин използва промяна на кода на програмите в компютъра. Нарушителят вкарва злонамерен код в сорс кода, с цел хакване на поверителна информация. Кодът може да се появи в стотици програми, което го прави много труден за откриване.

Easter eggs – модификации, които са „вградени”. Понякога програмиста може да си остави задни вратички в програмата, която е написал. Тя не е описана и за това е със скрита от потребителя. Понякога е просто код, който показва кой е писал програмата.

4. Премахване на rootkits

Премахването на rootkit-овете може да е много трудно особено след като се е устроил в ОС. Поради тази причина доста често системните администратори просто правят back up на данните, форматират диска и възстановяват ОС и програмите. Поради тази причина те правят копие на диска след преинсталирана ОС, докато не е заразена.

Ако нямат копие на ОС, може да се използват някои от продуктите за борба с rootkit-овете. Някой от най-известните са:

F-Secure Blacklight – работи като търси обекти, които са скрити и от потребителите и от инструментите за защита. Blacklight търси на доста подробно ОС и предоставя възможността за премахването на malware или rootkit ако са засечени.

Sophos Anti-Rootkit – програма, която търси rootkit-ове и може да бъде стартирана с графичен интерфейс или от командният ред

Trend Micro Rootkit Buster – сканира скритите файлове на системата, регистрите, активните процеси, драйверите, като може дори да претърси Master Boot Record за rootkit-ове.

Rootkit-овете са трудни за засичане и за това има вероятност потребителя дори да не разбере, че е заразен. За това се препоръчва използване на firewall заедно с антивирусна и anti-malware софтуер, като периодично се правят и сканирания за rootkit-ове с някои от програмите описани в точка 4.

5. Защо откриването на rootkit е трудно?

Те се инсталират обикновено лесно, като атакуващият използва пробиви във Windows-а или разбита парола, като дори и чисто физически достъп до системата. Могат също така да бъдат хванати от .exe файл прикачен към писмо или препратка (hyperlink). Те не са толкова разпространени като вирусите или spyware-ите. И това води до по-малко ефективни методи за справянето с тях. Друг фактор усложняващ тяхното откриване е, че се иска до голяма степен самият потребител да се „усети” за тяхното присъствие. Firewall-ите също не могат да предпазят системата от rootkit-ове.

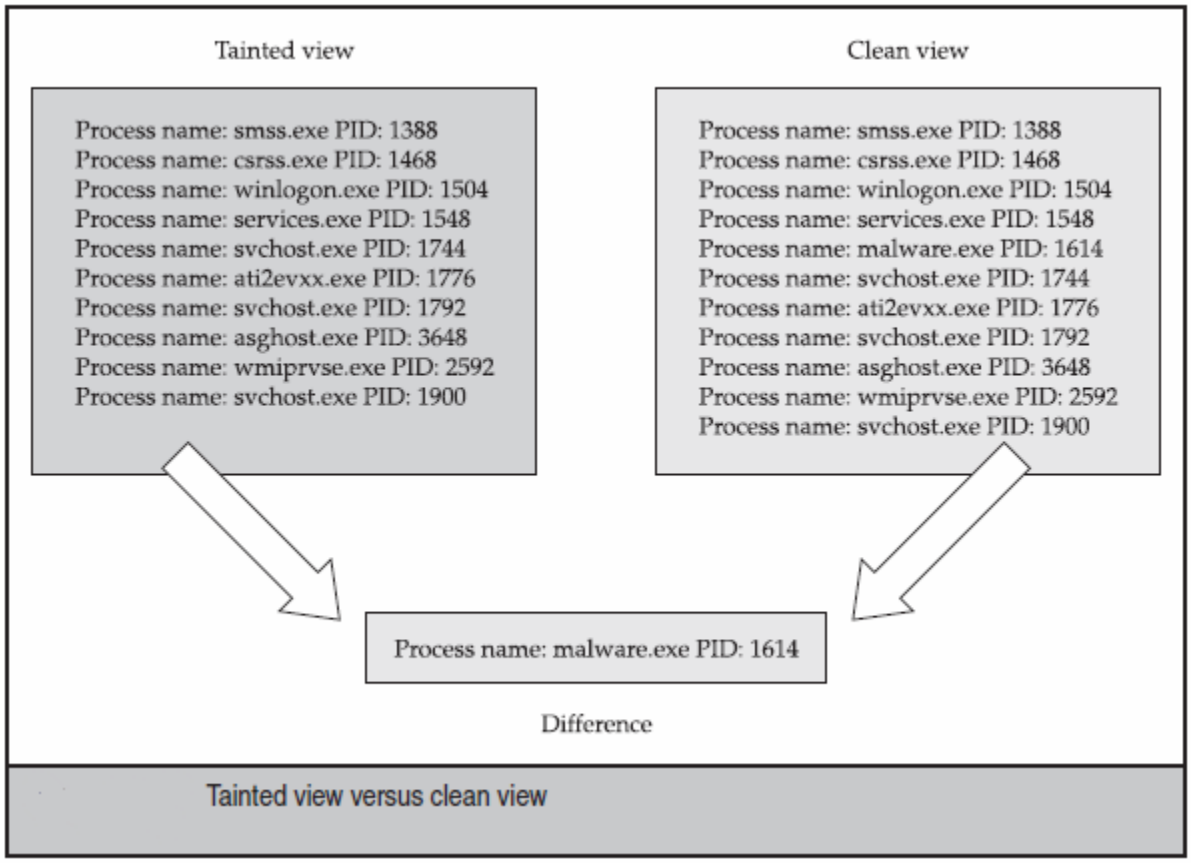
Съществуват програми за rootkit откриване, но няма такава която да може да открие всичките видове сама, което води до още усложнения за потребителя. Освен това има специално направени rootkit-ове за дадена цел или система и тяхното откриване е още по-трудно.

6. Начини за откриване на rootkit и различни начини на функционирането му

Tainted view

Веднъж след като подписно-базираните системи започнат да бъдат заобикаляни, нов комплект от подходи бива разработен. Често наричан cross-view или tainted view, като той е използван от множеството от съвременните rootkit откриващи програми. Tainted view подхода работи като сравнява различни „снимки" на системата като действащи процеси, хардуерът инсталиран на машината или имената и номерата на функциите необходими за изпълнението на специфична системна задача, и сравнява разликите, ако има такива. Като е прието, че видът на данните изпълнени по един начин, няма да съвпада с техният вид ако са изпълнение с rootkit в системата. Изгледа от потребителя се счита за tainted view. Изгледа от хардуерът се счита съответно за clean или trusted view.

Tainted-view има един главен недостатък, от който се възползват rootkit-овете, а именно, че той „предполага", че на ниско ниво, ще връща различни данни и че rootkit-а не може да контролира връщаните данни.



Фиг. 9 – Скрит процес при tainted-view

Един от първите инструменти използващи tainted-view подхода за засичане на rootkit-ове е Patchfinder, създаден от Йоана Рутковска.

System Service Descriptor Table Hooking

Това е една от най-простите и най-често използвани техники, за по-кратко SSDT, като се характеризира, че е лесна за засичане и почти всички програми я засичат.

Kernel-ат при Windows пази таблица с всички функции, които се използват от драйверите. Rootkit-а просто трябва да намери тази таблица, нейното копие, което се използва от GUI и да замени указателя към истинското място на дадена функция от kernel-а, с версия на функцията от rootkit-а.

IRP Hooking

Метода за засичане на IRP Hooking е същият както при SSDT Hooking. Всеки драйвер изважда комплект от 28 указатели на функции, за да управлява I/O пакети. Тези функции се съхраняват в DRIVER_OBJECT на драйвера и всеки указател на функция може да бъде заменен с друг. Ако се сканира DRIVER_OBJECT може да се забележи дали указателите на функции са заменени за този IRP.

Inline Hooking

Inline hooking или detour (от англ. пренасочване) е процес, на презаписване на първите 2-3 инструкции за функция с други инструкции, които водят до функциите на rootkit-а. Този метод предпочита да замества указателите на адресите на функциите. По принцип е лесно да бъде засечен, но не винаги. Процеса по засичането му е същият както при SSDT hooking. При някои rootkit детектори, се проверяват само първите x на брой байтове, като по този начин се получава по-голямо бързодействие. Веднъж след като бъдат заредени истинските инструкции, те се сравняват и ако има някакви несъответствия, това може да означава, че имаме пренасочване.

Interrupt Descriptor Table Hooks

Или IDT е на принципа като SSDT и IRP. Таблицата съдържа комплект от указатели на функции за всеки interrupt, като rootkit-а ги заменя interrupt-а със свои функции.

Direct Kernel Object Manipulation

Или DKOM е уникален метод за прикачено, защото автора манипулира обектите в kernel-а, които могат да се променят между service packs или пачове пуснати от Майкрософт. За да бъдат засечени такива обекти, трябва да се знае какъв тип точно търсим.

IAT Hooking

Hooking-а не се случва само в kernel-а. Доста често става на ниво потребител и там е много лесно да се имплементира. Един от най-видните hooks на потребителско ниво е IAT. Неговото засичане е недвусмислено. Първо се намира списъка с DLL, които са необходими на процесите. За всеки DLL се анализират функциите и се запазват внесените адреси. Ако бъдат открити някакви несъответствия, означава че има шанс някоя от функциите да е внесена от hooking-а.

7. Windows anti-rootkit свойства

Windows определено има дефекти, но заради тях Майкрософт са вложили доста усилия за подобряване на ОС след ХП към СП3, през Виста до Windows 7. Някои от технологиите разработени от тях са:

Secure Development Lifecycle (SDL)

Windows Vista е първата ОС използваща SDL на Майкрософт

Windows service hardening

С тази технология Майкрософт твърдят, че предпазват потребителя от malware и rookits като използват различни ограничения на достъп/привилегии.

No-execute (NX) и Address Space Layout Randomization (ASLR)

Тези две техники са главно добавени за да помогнат препълването на буфера, техника която понякога е използвана от rootkit-овете.

Kernel Patch Protection (KPP)

Още известен като PatchGuard, KPP предотвратява всяка програма опитваща се да промени kernel-а или структурата от данни на kernel-а, като при SSDT и IDT. Тази технология налага голям удар над авторите на rootkit-ове и производителите на антивирусни. Включена е само при 64 битови ОС.

Required driver signing

При 64 битови ОС, всички драйвери трябва да бъдат електронно подписани от одобрени упълномощители или няма да бъдат заредени в kernel-а.

BitLocker drive encryption

Главно се използва за криптиране на целият диск.

Authenticode

Майкрософт посредством това приложение позволява на производителите да подпишат приложенията си, така че в последствие kernel-а да може да направи проверка по хеш.

User Account Control (UAC)

Тази технология внедрявана най-добрите начини за предпазване дори за обикновеният потребител, като например той да няма администраторски права по време на работа.

Software restriction policy

Това позволя ако администратора забрани даден софтуер, той да не може да бъде инсталиран.

Microsoft Malicious Software Removal Tool (MSRT)

Това е продуктът на Майкрософт против malwares, който използва традиционните техники за засичане чрез подпис.

Internet Explorer 7+

Доста неща биват подобрени и в браузъра като пълен контрол над add-ons, защитен режим, филтри за phishing и вграден anti-spyware.

8. Софтуерно базирани решения за откриване на rootkits

Към момента има много приложения за откриване на rootkit в интернет, както и почти всички антивирусни пакети съдържат в себе си такива. Една от тях е VICE, която е безплатна. Когато се появява е била единствена по рода си, съчетаваща нови техники не виждани до тогава. Тя търси и в потребителската част и в kernel-а като по този начин открива IAT и SSDT. Но тя е доста сложна и не е много лесна за работа. Но дори днес не много приложения са лесни за работа, което само усложнява работата на обикновеният потребител. Софтуерните решения са добри само, когато се използват с други софтуерно базирани rootkit детектори. Например дадено приложение би засякло един вид rootkit, но друг не, докато друго обратното или би премахнало само файловете на rootkit-а, докато друго би премахнало и регистрите свързани с него. Комбинацията от такива програми е най-добрият начин за защита.

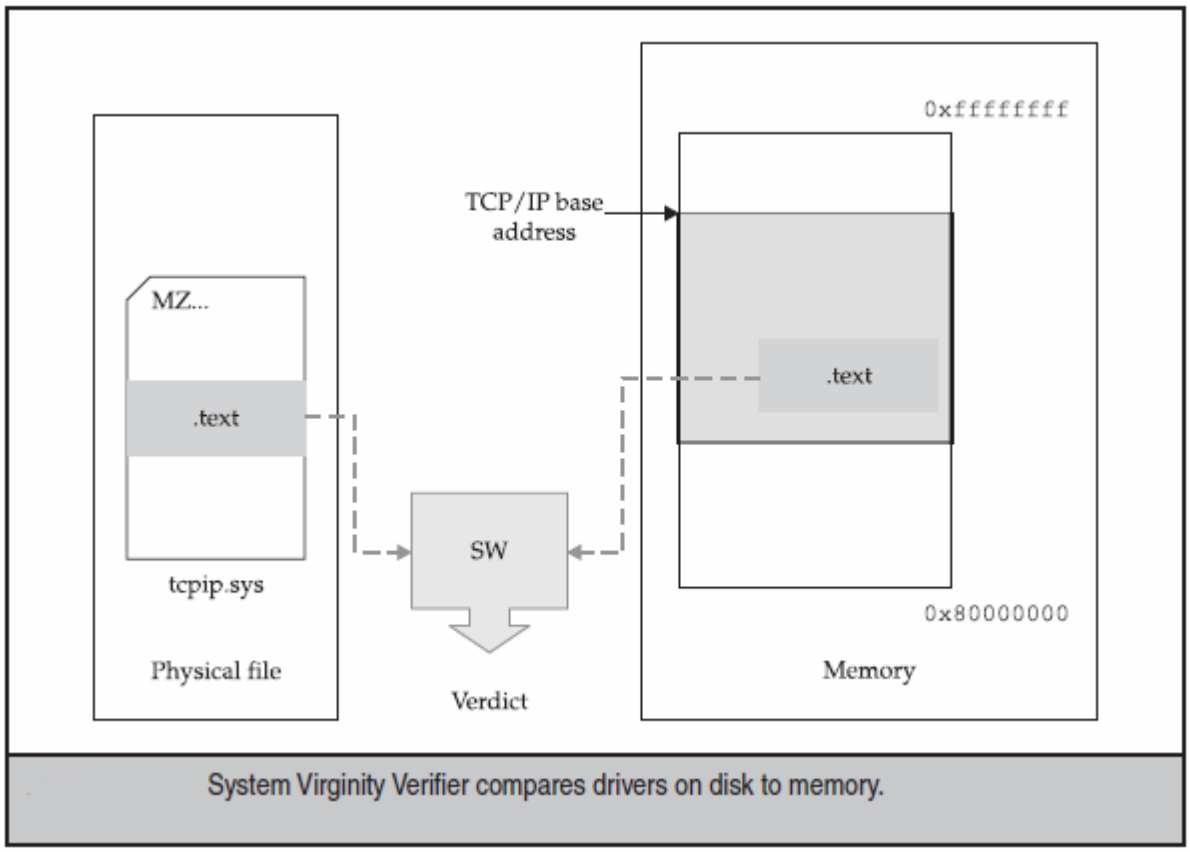
Live Detection срещу Offline Detection

Важната разлика тук е къде се прави анализа. Ако се прави на „живо”
на заразената система, malware има шанс да зарази доказателствата и от там и анализа. Това е така, защото както бе споменато rootkit-овете могат лесно да скрият техният процес. Това може да доведе до доста грешни и неверни анализи, което от своя страна ще бъде подвеждащо. От друга страна offline засичането е доста по-трудно за

реализиране, защото няма помощта на ОС да анализира структурите, типовете данни, които се достъпват и тн. Всички тези неща трябва да бъдат пресъздадени в инструмент, който ще позволи offline анализа да прилича на анализа на „живо”.

System Virginity Verifier

Още SVV е инструмент написан от Йоана Рутковска, който имплементира уникален метод за откриване на rootkit-ове. Той проверява целостта на критичните елементи от системата, за възможен компромис. Понеже всеки драйвер и .exe файл е изложен на риск от множество различни типове данни, SVV анализира части от кода в бинарен режим, които съдържат асемблерски инструкции, имена на модули, на функции или наименованията на бутоните и прозорците. SVV прави сравнение между зареденият код в kernel-а и този на физическото ниво във файловата система.



Фиг. 10 – Сравнение на драйверите от диска, с тези от паметта при Virginity Verifier

Ice Sword и DarkSpy

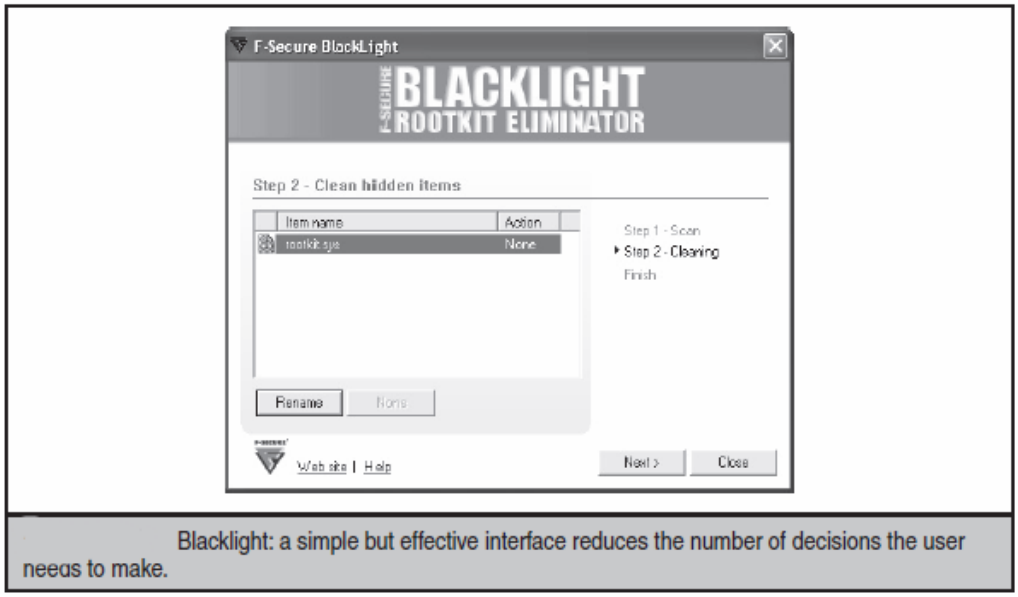
Те са вид tainted-view детектори, но изискват доста голяма намеса от потребителя. Например той може трябва да анализира процесите и промяната в тях при отварянето на уеб браузър.

Rootkit Revealer

Това е един от първите детектори, който са с опростен интерфейс за потребителя. Създаден е от Брус Когсуел и Марк Русинович от SysInternal, в последствие придобит от Майкрософт. Използва cross-view подхода и се фокусира само на файловата система и регистрите. Главните предимства са, че е бърз, прост и ефективен. Нужно е само да се избере File > Scan и да се изчака няколко минути. Но е лесен за заобикаляне от rootkit-овете.

F-Secure's Blacklight

Blacklight имплементира tainted или cross-view подхода и е първият инструмент, който използва това е същевременно е лесен за работа. F-Secure е антивирусна компания и те са разработили и Blacklight като комерсиален продукт. Има и безплатна версия. Въпреки, че може да бъде заобиколен, Blacklight е полезен и заради възможността да се сложат файлове „под карантина", като скрити файлове, след което да бъдат преименувани и след рестарт, rootkit-а няма да може да се зареди. Минус е, че не може да преименувате файловете ръчно, а този процес става автоматично от Blacklight. Друго различно нещо е, че използва bruteforce за проверка на всички процеси (PID).



Blacklight: a simple but effective interface reduces the number of decisions the user needs to make.

Фиг. 11 – Blacklight – лесен и същевременно ефективен

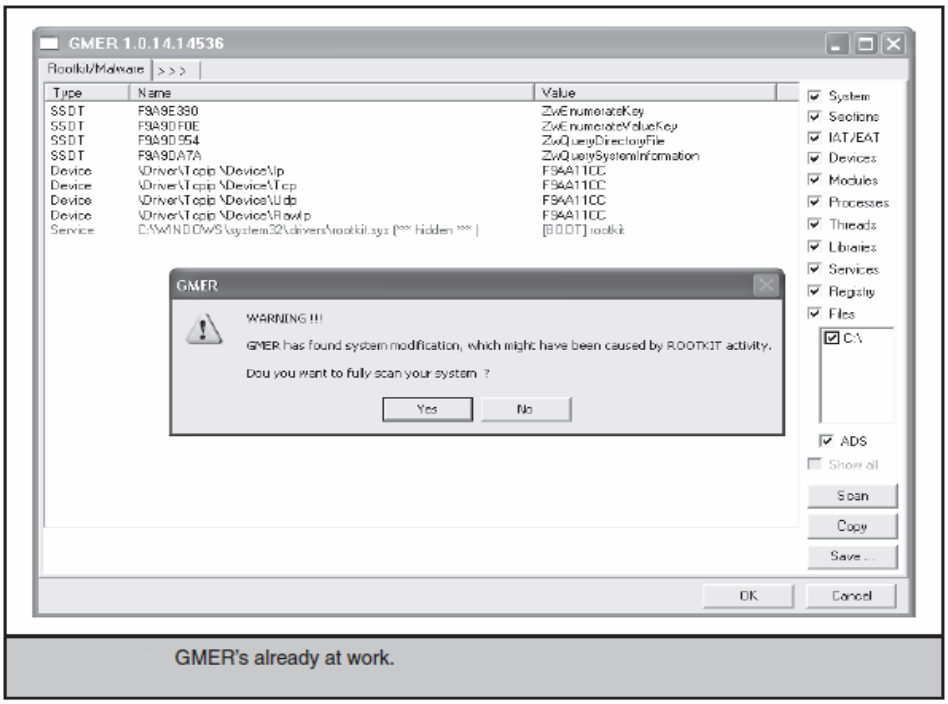
Rootkit Unhooker

Това е инструмент за напреднали потребители. Той има доста богати функции. Позволява на потребителя да се включи към системата по най-различни начина, като например да погледне SSDT,

Shadow SSDT, сканиране на ниско ниво на файловата система посредством достъп до твърдия дискът директно, а не през ОС, таблицата на процесите и др. Rootkit unhooker също има възможност да намери hooks в TCP/IP стека. Предоставя възможност и за „изкуствено" предизвикване на син екран (Blue Screen Of Death – BSOD), което позволява да се върже дебъгер като WinDBG през серийният порт или USB към машината и посредством BSOD да придобие копие на цялото съдържание на паметта по време на забиването. В последствие данните може да се анализират в „offline" режим и да се разбере повече за rootkit-а.

GMER

GMER е инструмент за по-сложна, но в същото време не експертна работа от потребителя. Той предоставя почти всички методи за засичане на rootkits в един инструмент, както и някои възможности за премахването им. Той се поддържа и ъпдейтва доста често, което е голям плюс. Специфично за него е, че започва да сканира веднага след стартирането си, като търси скрити файлове, процеси, услуги или hooked регистри. GMER притежава функциите на много други rootkit детектори и ги съчетава и използва автоматично.



Фиг. 12 – GMER работен екран

Helios and Helios Lite

Това са инструменти за засичане на rootkits създадени от MIEL Labs. И двата използват подобни методи за засичане. Helios е цяла отделна програма за активно засичане и

почистване от rootkits, докато Helios Lite е самостоятелна и може да сканира бързо системата за SSDT hooks, скрити процеси, скрити регистри и скрити файлове. Helios Lite използва GUI програма за да комуникира с kernel драйверът helios.sys. Заедно с Helios, могат да засекат повечето rootkit hooking и скритите техники използвани от rootkit-овете. Helios се състои от .NET GUI приложение, две библиотеки (DLLs) и kernel драйверът chkproc.sys.

McAfee Rootkit Detective

McAfee е един от първите комерсиални производители, които пускат безплатен софтуер за засичане на rootkits. Пуснат през 2007 (не дълго след пускането на Blacklight от F-Secure през 2006), те приемат доста похвали от общността занимаваща се със защита на системите. Rootkit Detective е доста опростена програма, позволявайки на потребителя да погледне скритите файлове, регистри, hooked услуги, IAT/EAT hooks и detour-style пачовете. GUI интерфейсът и се състои от един панел с радио бутони.

Комерсиални инструменти за rootkit засичане

Масово, платените инструменти за засичане на rootkit, не са много усъвършенствани и са сравнително лесни за заобикаляне от най-новите rootkit-ове. Причината за това е, че компаниите, които ги правят не могат да разчитат на най-новите методи за засичане, защото голяма част от тези методи не са надеждни за средностатистическият потребител. Това разбира се не винаги е така, но сред повечето общности занимаващи се със засичане на rootkit-ове, се счита че безплатни инструменти като Rootkit Unhooker и GMER са много по-добри от комерсиалните продукти на пазара.

Начини за засичане в „offline" режим: еволюцията на сравняването на паметта

Напредъка в засичането на rootkits в комерсиалните продукти се базира на основата на принципи, които са обсъждани от известно време. Тази област на изследвания се нарича „memory forensics" и адресира две широки предизвикателства:

- Придобиване на паметта – как „изследователя" взема физическото съдържание на паметта
- Анализ на паметта – как веднъж взета тази памет, да и се направи анализ и да се търсят артефакти и доказателства в цялата купчина от данни

Volatility

Volatility е анализ на паметта с разтегнати основни рамки от инструменти базирани на изследванията от Арон Уолтърс от Volatile Systems. Той е един от основателите на съвременните техники за анализ на паметта.

В същността си Volatility съдържа библиотеки със скриптове писани на Питон, които обработват и реконструират структури от данни, съхранени в паметта на подозираната система. Работата на ниско ниво на тези скриптове остава скрита за потребителя, така че не се изискват подробни знания как работи Windows OS.

Volatility предоставя основна информация като:

- Стартираните процеси и нишки
- Отворените сокети и конекции
- Заредените модули в потребителският и kernel режим
- Ресурсите, които се използват от процеса като fi les, обекти, регистрови ключове и други
- Възможността да сканира един процес

Истинската сила на Volatility се крие във възможността изследователите да пишат свои собствени плъгини, които използват основите му.

Memoryze

В контраст на „offline" начина на работа на Volatility, Mandiant Memoryze е анализатор на паметта, с възможности да открива rootkit-ове и malware-и едновременно от паметта и от наблюдение в реално време на системата. Той се базира на флагмана на компанията – Mandiant Intelligent Response (MIR). Съдържа няколко компонента:

- XML audit скриптове – записани като изпълними скриптове те служат като конфигурационен файл за програмата. Седем от тях дефинират параметрите за различните методи за анализ.
- Memoryze.exe – това е изпълнимият файл на програмата, който разчита данните от XML настройките и зарежда необходимите библиотеки и DLL-и за осъществяването на анализа
- Batch скриптове – тези DOS batch скриптове осигуряват спокойствието на потребителя като вместо него заредят автоматично XML скриптовете
- Core библиотеки – тези DLL-и осигуряват възможност за анализ на паметта на ниско ниво в програмата
- Библиотеки от трети страни – тези DLL-и от програми с отворен код като Perl Compatible Regular Expressions (PCRE) са за търсене с нормални изрази и ZLIB за компресия

- Kernel драйвер – Core библиотеките генерират kernel драйверът с името mktools.sys и го внедряват в програмата при успешно стартиране на Memoryze.exe. Драйверър осигурява повечето от информацията, която по-късно ще бъде събрана за анализ.

С това Mandiant Memoryze осигурява не само функции присъстващи във Volatility, но и:

- Достъп до цялата физическа памет, включваща и всяко адресно пространство на всеки процес
- Вземане на информация за програмите от потребителското ниво и на драйверите от kerner-ското ниво
- Информация за активните процеси, мрежови конекции и вградените стрингове
- Rootkit засичане посредством hook засичане в SSDT, IDT и IRP таблиците
- Използване на енумератори за процеси, драйвери и DLL-и.

9. Виртуално засичане на rootkits

Напоследък има тенденция и за виртуални rootkit-ове, но те се оказват не толкова трудни за засичане. Скорошни проучвания показват, че направата на перфектен мениджър на виртуална машина (VMM), който симулира работата на истински хардуер е практически невъзможно. Което води от своя страна факта, че доста потребители, администратори и изследователи използват VMM, без виртуализация и ако VMM бъде засечен, значи той е rootkit. Повечето VMM засичания са прости и разчитат на известните виртуализирани хардуери, ресурси или времеви атаки. Така например ако мрежова карта от специален производител като VMWare или Virtual PC индицира, че ОС е стартирана под VMM, това може да означава, че ОС е контролирана от rootkit. Освен този метод, няма много други начини, по които може да се определи дали има rootkit под виртуалната машина или има виртуален rootkit като BluePill, а и масово атаките са за да се определи дали има VMM.

10. Хардуерно базирано rootkit засичане

Всички изброени до сега решения са софтуерно базирани, но да създадеш софтуер, който да се бори с друг „вреден” софтуер е много трудно, когато и двата продукта трябва да се „бият” за същите ресурси и устройства.За това една компания през 2004 опитва да имплементира хардуерно базирано rootkit засичане. Тя се казва

Komuku и е финансирана от Агенцията за Напреднали Изследователски Проекти за Защита на Съединените Щати (United States Defence Advanced Research Projects Agency – DARPA), отдела за защита на страната (Department of Homeland Security) и Флота (Navy). Komoku създават хардуерно базирано решение наречено CoPilot, PCI карта с висока сигурност и възможност за наблюдаване на паметта и файловата система на хардуерно ниво на системата. Тя сканира и преценява работата на работната станция или сървъра в почти реално време и гледа за аномалии, вместо да се мъчи да намери точно определен вид rootkit. Според правителството продукта е успешен, но понеже е спонсориран от тях, не е възможно закупуването му от обществото, още повече след като Майкрософт купиха Komoku през март 2008, мнозина смятат, че CoPilot ще спре своето развитие.

През 2004 Grand Idea Studios създава PCI карта, която може да анализира RAM паметта в системата и се казва Tribble. Продукта е създаден от Браян Кариек и Джо Гранд и е с американски патент. Tribble обаче не се продава.

През 2005 BBN Technologies създава хардуерно устройство, което се включва към сървъра или работната станция и взема копие от RAM-та за анализ, но то не може да предотврати зареждането на malware в RAM-та.

Но дори с тези напреднали технологии има още доста да се желае. През 2007 Йоана Рутковска доказа, че дори с хардуерно засичане, специално изработен rootkit отново може да заобиколи засичането. Към момента единствените хардуерно базирани методи за защита са налични само за правителствените агенции.

Както беше споменато и преди анализа на паметта е много труден, защото тя се променя постоянно. Много от новите хардуерни подходи започват да намират начини да вземат „преглед” от паметта, който е едновременно точен и достоверен и не взаимодейства със системата. Отделно и с променящите се ОС броят на нещата, които трябва да се анализират расте. Но в тази област за сега трябва да се направят още много изследвания за да има добър ефект.

11. Заключение

Засичането на rootkit-и е трудно. Техниките използвани за засичането на rootkit са лесно избягвани от човека, посветил време за да се увери, че няма да бъде засечен от тях. Фундаменталните техники за засичане имат дефекти и могат да бъдат заобиколени. Но въпреки, че засичането им може да бъде избегнато, доста от авторите на rootkit-ове днес дори не се мъчат да ги скрият, защото повечето

потребители въобще не търсят за rootkit-ове. Още повече като имаме на предвид, че повечето rootkit-ове работят на ниво над потребителското, прост поглед на регистрите и файловата система може да създаде илюзията, че няма инсталиран rootkit и че не е необходимо да се пуска приложение за засичане на rootkit. Хардуерно базираните решения за засичане на rootkit-ове показват големи обещания, но не са перфектни и изискват допълнителни разходи и въпреки, че компаниите са спонсорирани от правителството на САЩ за разработката им, такива продукти няма на пазара за сега.

Доста от софтуер базираните rootkit засичащи решения са безплатни, но изискват големи умения, за да бъдат анализирани данните от тях. Доста от техниките при тях са вкарани в комерсиални продукти и въпреки това, поради факта, че нито една програма не може да засече всички видове rootkit-ове, се препоръчва използването на няколко едновременно.

Използвана литература

1. How do rootkits work? – http://www.informit.com/articles/article.aspx?p=408884&seqNum=5 12.2010
2. How rootkits work? – http://www.spamlaws.com/how-rootkits-work.html , 12.2010
3. RootkitRevealer – http://technet.microsoft.com/en-us/sysinternals/bb897445.aspx , 12.2010
4. Rootkit – http://en.wikipedia.org/wiki/Rootkit , 12.2010
5. Rootkits - www.security-assessment.com , 12. 2010