HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

Aplikacijski RootKit alati za Windows operacijske sustave

CCERT-PUBDOC-2004-10-94

CARNet CERT u suradnji s LS&S

Sigurnosni problemi u računalnim programima i operativnim sustavima područje je na kojem CARNet CERT kontinuirano radi.
Rezultat toga rada ovaj je dokument koji je nastao suradnjom CARNet CERT-a i LS&S-a, a za koji se nadamo se da će Vam koristiti u poboljšanju sigurnosti Vašeg sustava.

CARNet CERT, www.cert.hr - nacionalno središte za **sigurnost računalnih mreža** i sustava.

LS&S, **www.lss.hr**- laboratorij za sustave i signale pri Zavodu za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu.

Sadržaj

1. Uvod

Zlonamjerni korisnici svakako su jedna su od najvećih prijetnji računalnim sustavima bez obzira radi li se o pokušaju napada na sustav izvana ili iznutra. Nove ranjivosti otkrivaju se i objavljuju svakodnevno, što neovlaštenim korisnicima otvara sve šire mogućnosti napada na sustave. No, osim samih tehnika i alata koje maliciozni korisnici koriste kako bi inicijalno ostvarili pristup sustavu, sistem administratorima i ostalim korisnicima potrebno je ukazati na različite tipove malicioznih programa koje napadači vrlo često postavljaju na sustave nakon što su im inicijalno ostvarili pristup. Takvi programi najčešće ostaju prikriveni na sustavu, a osnovna namjena im je da se napadaču osigura nesmetan pristup sustavu, čak i nakon što se ukloni sama ranjivost putem koje je ostvaren pristup. U tom smislu potrebno je istaknuti RootKit alate, kao primjer malicioznih programa koje neovlašteni korisnici vrlo često koriste nakon što su provalili u sustav.

RootKit alati iznimno su popularni među neovlaštenim korisnicima, budući da im omogućuju nesmetan pristup sustavu, a najčešće su vrlo dobro prikriveni što otežava njihovu detekciju čak i iskusnim sigurnosnim stručnjacima. Razumijevanje načina rada, namjene te osnovnih mogućnosti RootKit programa vrlo je važno i može pomoći povećanju sigurnosti sustava.

Dokument opisuje aplikacijske RootKit programe za Windows operacijske sustave. Analizirane su tri osnovne metode implementacije ovih alata te kratki primjeri za svaku od opisanih metoda. Opisom navedenih mjera zaštite ukazuje se na moguću zaštitu od ovih izuzetno složenih i opasnih malicioznih programa.

2. RootKit alati

RootKit naziv sastavljen je od dvije riječi: "root" i "kit". Riječ "root" preuzeta je od naziva UNIX administratora koji ima najviša prava pristupa u UNIX okruženju. Riječ "kit" označava kolekciju alata u jednom paketu. Na temelju svih ovih karakteristika RootKit programe moguće je definirati kao skup alata koji napadaču omogućuju održavanje administratorskih prava na kompromitiranom sustavu te da ih je vrlo teško detektirati u normalnom radu sustava. Bitno je naglasiti da ovi alati ne omogućuju sam ulaz u sustav, već svoju funkciju obavljaju nakon kompromitiranja ciljnog sustava. Kada su se prvi puta pojavili RootKit programi pretežno su bili razvijani za Linux/Unix operacijske sustave, no s obzirom na popularnost Windows operacijskih sustava danas su sve češći *rootkit* programi za Windows platforme.

Uporaba ove vrste alata bila je popularna 90-tih godina prošlog stoljeća. Tijekom tih godina masovno su objavljivani dokumenti o povećanju broja poslužitelja koji su kompromitirani upravo RootKit programima.

Uobičajeno se RootKit programi sastoje od sljedećih tipova alata:

- programi koji omogućuju stražnji ulaz u sustav (eng. *Backdoor*),
- sniferi (eng. *Packet sniffers*),
- sistemski alati za brisanje log zapisa (eng. *Log-wiping utilities*),
- DDoS programi (eng. *DDoS programs*),

2.1. Tipovi alata

RootKit alati dijele se na dva osnovna tipa. Prvi tip su aplikacijski RootKit alati (eng. *application rootkit*, *user-level rootkit*, *user-mode rootkit*) koji se postavljaju na aplikacijskom sloju, dok drugi tip predstavljaju RootKit alate postavljene na razini jezgre operacijskog sustava (eng. *kernel rootkit*, *kernel-level rootkit*, *kernel-mode rootkit*). Osnovna razlika među navedenim tipovima alata nije samo u sloju na kojem program djeluje, već i u načinu njihovog prikrivanja na sustavu.

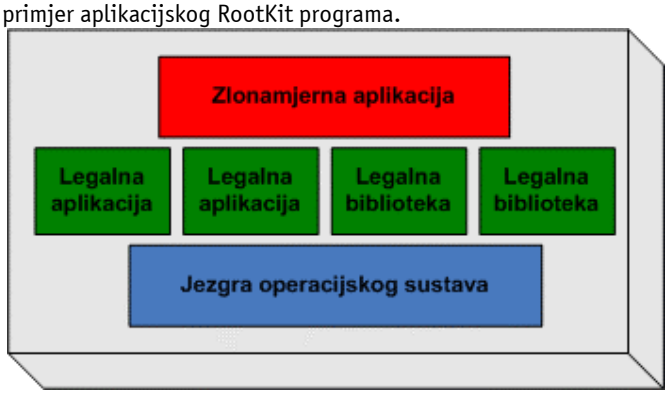
2.1.1. Aplikacijski RootKit alati

Aplikacijski RootKit alati koriste metodu zamjene legitimnih sistemskih aplikacija s malicioznim datotekama koje će obavljati određene akcije u korist napadača. Novo postavljene datoteke omogućuju napadaču stražnji ulaz u kompromitirani sustav, skrivaju njegovu prisutnost i ne zapisuju

aktivnosti koje napadač izvodi. Lista nekih uobičajenih datoteka koje bivaju zamijenjene od strane napadača na Linux operacijskim ustavima su:

- ls, find, du,
- ps, top, pidof,
- netstat,
- killall,
- ifconfig,
- crontab itd.

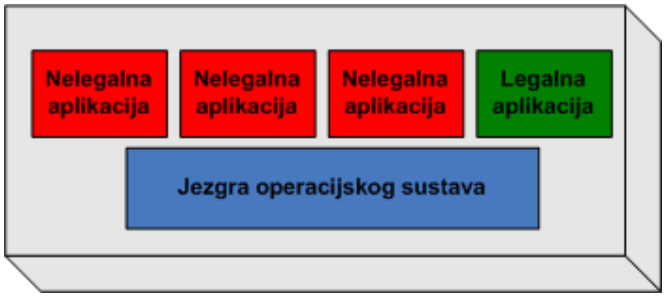
Glavna karakteristika ovog tipa alata jest da napadaču omogućuju promjenu postojećih izvršnih datoteka ili biblioteka na sustavu mijenjajući programe koje pokreću korisnici ili administratori. Na slici 1 prikazan je primjer aplikacijskog RootKit programa.



Slika 1: Aplikacijski *RootKit*

2.1.2. Kernel RootKit alati

Drugi tip *RootKit* alata su alati čija je osnovna karakteristika manipulacija jezgre operacijskog sustava (eng. *kernel*), skrivanje prisutnosti i kreiranje stražnjih ulaza u sustav (eng. *backdoor*). Ovaj tip alata teže je detektirati od aplikacijskog tipa budući da se integriraju u sam operacijski sustav što im omogućuje vrlo visoku razinu kontrole nad svim komponentama sustava. Slika 2 prikazuje RootKit alate koji omogućuju zamjenu izvršnih datoteka ili biblioteka kodova na razini same jezgre operacijskog sustava.

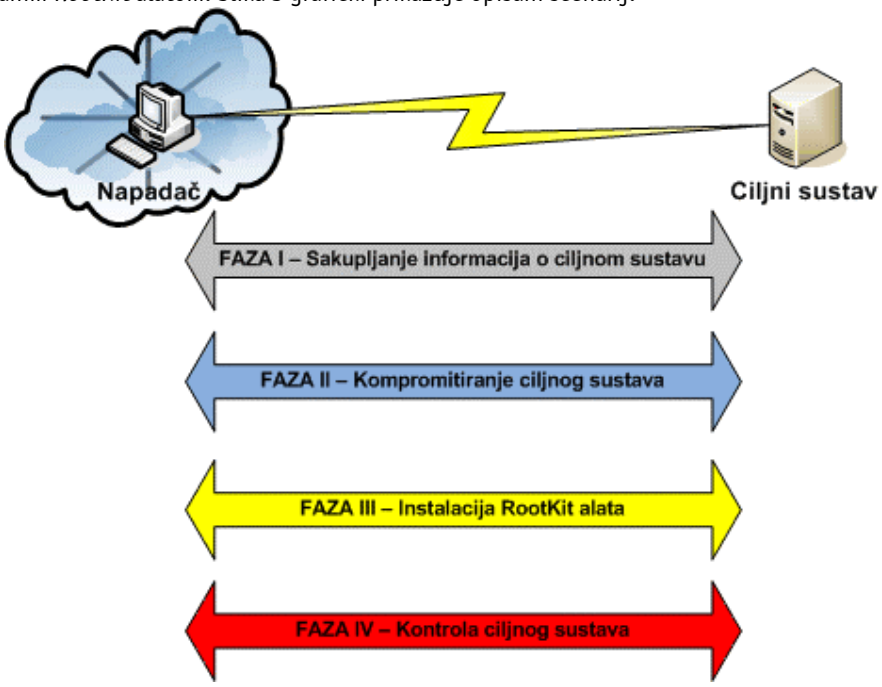


Slika 2: Kernel *RootKit*

2.2. Scenarij napada na sustav

Prije nego što zlonamjerni napadač pokuša ostvariti pristup sustavu, mora iznimno dobro poznavati osnovne karakteristike i potencijalne ranjivosti ciljnog sustava. Prikupljanje informacija o ciljnom sustavu podrazumijeva različite tehnike i metode te najčešće predstavlja prvu fazu napada na sustav. Cilj ove faze je prikupiti što je više moguće informacija o ciljnom sustavu kako bi se omogućilo daljnje provođenje napada. Slijedeća faza je korištenje konkretnih tehnika i metoda kojima se pokušava ostvariti pristup ciljnom sustavu. Osnovni cilj ove faze je dolazak do ovlasti administratora, osdnosno preuzimanje potpune kontrole nad sustavom. U slijedećoj fazi napadač instalira RootKit alat koji će

mu omogućiti prikrivanje tragova te služiti kao stražnji ulaz u sustav. Posljednju fazu scenarija napada na sustav predstavlja sama kontrola nad kompromitiranim sustavom koja je omogućena instaliranim *RootKit* alatom. Slika 3 grafički prikazuje opisani scenarij.



Slika 3: Scenarij napada na sustav

3. Aplikacijski RootKit alati u Windows okruženju

Navedeni i opisani tipovi RootKit alata primjenjivi su na UNIX i Windows operacijske sustave. U povijesti RootKit alata njihova je namjena bila isključivo vezana uz UNIX platforme, no vremenom su razvijene inačice alata i za druge platforme. Postoji nekoliko razloga zbog kojih su češće u uporabi UNIX RootKit alati, od Windows *RootKit* alata. Neki od njih su:

- *Windows File Protection* (WFP) svojstvo ometa zamjenu izvršnih datoteka jer djeluje tako da, ukoliko detektira promjene kritičnih sistemskih datoteka, automatski sistemske datoteke vraća u originalno stanje,
- programski kod Windows operacijskog sustava nije javan (engl. *closed source*), što zahtjeva mnogo više znanja i vremena za razumijevanje načina rada sustava.

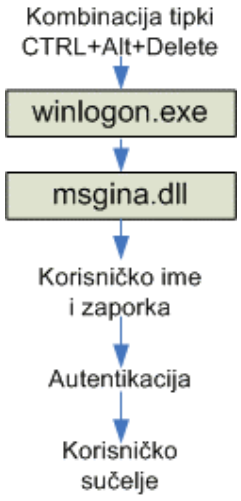
U nastavku dokumenta opisani su neki od aplikacijskih RootKit alata za Windows operacijske sustave. Pristup koji je korišten u dokumentu temelji se na tri osnovne metode implementacije aplikacijskih RootKit alata u Windows okruženjima. Opisane su tri metode:

- uporaba postojećeg sučelja za umetanje malicioznog koda među postojeće Windows funkcije,
- deaktiviranje svojstva *Windows File Protection* i prepisivanje sistemskih datoteka smještenih na lokalnom disku,
- uporaba *DLL injection* i *API hooking* tehnika napada.

U sklopu opisa spomenutih tehnika na kojima se bazira rad većine Windows RootKit programa, dani su i konkretni primjeri alata koji implementiraju pojedine tehnike. Kao primjer alata koji koristi prvu metodu korišten je *FakeGINA* program, *Code Red II* crv naveden je kao primjer druge metode, dok je *AFX Windows RootKit* program dan kao konkretni primjer alata koji koristi treću metodu.

3.1. Načini implementacije I

Prva metoda implementacije aplikacijskog RootKit alata u Windows okruženje obuhvaćena ovim dokumentom jest uporaba postojećeg sučelja za umetanje malicioznog koda među postojeće Windows funkcije. Naime, poznato je da je Microsoft u svoje Windows operacijske sustave ugradio određene komponente, odnosno sučelja koja omogućuju nadogradnju sustava korištenjem nezavisnih alata razvijenih od drugih proizvođača. Neki dijelovi operacijskog sustava su izuzetno modularni te omogućuju administratoru dodavanje novih komponenti korištenjem definiranih sučelja.
Proces korisničke prijave u sustav (eng. *user logon process*) jedan je od posebno važnih elementa koji također može biti proširen dodavanjem novih programa i biblioteka (ova funkcionalnost omogućena je prvenstveno kako bi se u sklopu Windows operacijskih sustava podržali neki napredniji oblici autentikacije kao što su biometrika, Smart kartice i sl.).
Slika 4 prikazuje klasičan proces prijave korisnika u sustav.

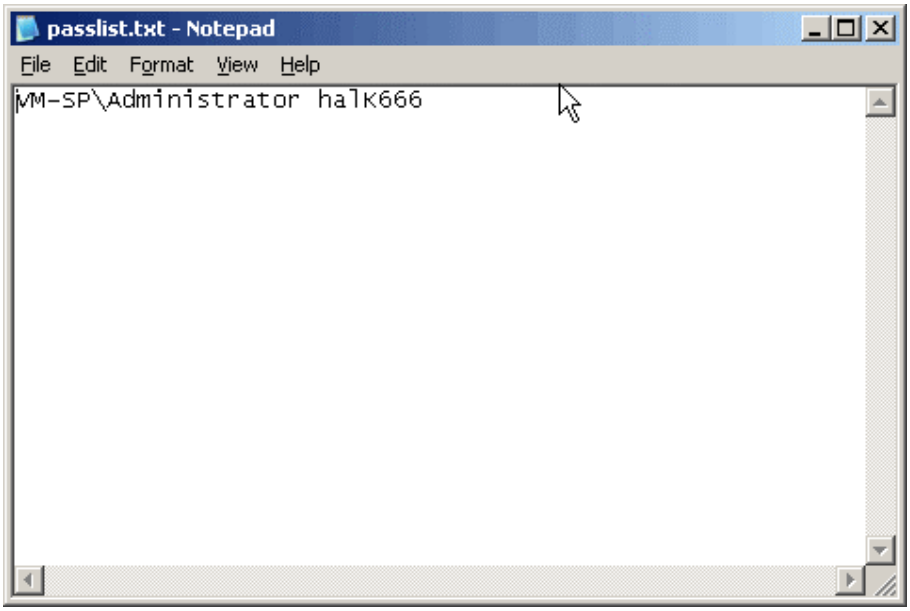


Slika 4: Klasičan *Winlogon* proces

Proces prijave korisnika u sustav započinje pritiskom kombinacije triju tipaka: `CTRL+Alt+Delete` koje predstavljaju sekvencu koja je karakteristična za Windows operacijske sustave. Nakon toga Winlogon proces (`winlogon.exe`) poziva biblioteku pod nazivom *Graphical Identification aNd Authentication*, GINA (`msgina.dll`) koja je zadužena za autentikaciju korisnika. GINA modul od korisnika zahtjeva unos podataka za autentikaciju (korisničko ime i zaporka) nakon čega se izvodi proces autentikacije i pokreće korisničko sučelje. `Msgina.dll` je standardna biblioteka u Windows operacijskim sustavima, no obzirom da Windows platforma podržava dodatne mehanizme autentikacije, ovaj proces moguće je modificirati korištenjem alata treće strane (eng. *third-party tools*). Jedan od takvih alata je *FakeGINA* trojanski konj, koji se zbog svoji karakteristika također može svrstati i u kategoriju RootKit alata.
Alat *FakeGINA* može se dohvatiti s referentne adrese http://www.ntsecurity.nu/toolbox/fakegina/. Instalacija alata sastoji se od pohrane datoteke `fakegina.dll` u mapu `system32` čija je uobičajena putanja `c:\%SYSTEMROOT%\system32`. Nakon toga potrebno je izmijeniti vrijednost *registry* zapisa `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon` koji je prikazan na slici 5.
Standardna vrijednost `GinaDLL` ključa je `msgina` ili `msgina.dll`. Tu vrijednost treba promijeniti u `fakegina.dll`. Ukoliko ključ ne postoji, potrebno ga je kreirati kao tip `REG_SZ` i njegovu vrijednost također postaviti na `fakegina.dll`. Nakon što se sustav ponovno pokrene, alat *FakeGINA* će sve upisane podatke za autentikaciju upisivati u tekstualnu datoteku `passlist.txt` koja je smještena u mapi `c:\%SYSTEMROOT%\system32`, a prikazana je na slici 6.

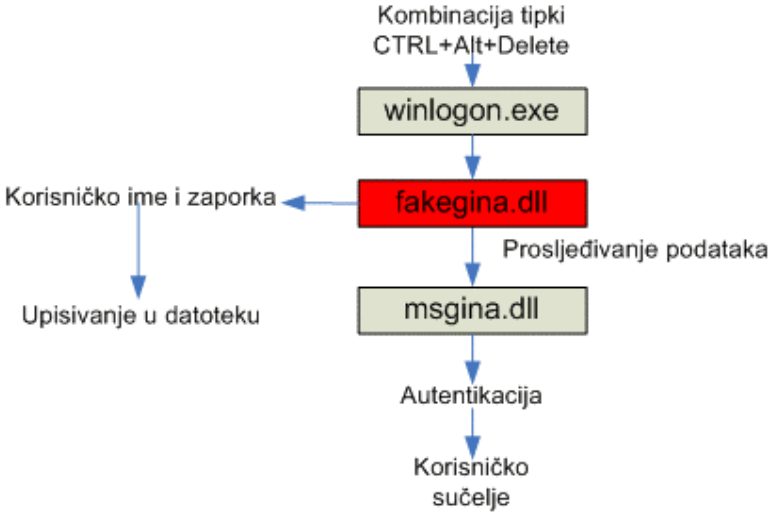
Name	Type	Data
(Default)	REG_SZ	(value not set)
allocatecdroms	REG_SZ	0
allocatedasd	REG_SZ	0
allocatefloppies	REG_SZ	0
AllowMultipleTSS...	REG_DWORD	0x00000001 (1)
AltDefaultDomain...	REG_SZ	VM-SP
AltDefaultUserNa...	REG_SZ	Administrator
AppSetup	REG_SZ	
AutoRestartShell	REG_DWORD	0x00000001 (1)
cachedlogonscount	REG_SZ	10
CachePrimaryDo...	REG_SZ	LSS
DCacheUpdate	REG_BINARY	02 51 f1 bd f7 c6 c4 01
DebugServerCom...	REG_SZ	no
DefaultDomainNa...	REG_SZ	VM-SP
DefaultUserName	REG_SZ	Administrator
DisableCAD	REG_DWORD	0x00000000 (0)
forceunlocklogon	REG_DWORD	0x00000000 (0)
GinaDLL	REG_SZ	fakegina.dll
Key	REG_BINARY	d8 c0 07 01
LegalNoticeCaption	REG_SZ	
LegalNoticeText	REG_SZ	
passwordexpiryw...	REG_DWORD	0x0000000e (14)
PowerdownAfter...	REG_SZ	0
ReportBootOk	REG_SZ	1
scremoveoption	REG_SZ	0
SFCDisable	REG_DWORD	0x00000000 (0)
SfcQuota	REG_DWORD	0xffffffff (4294967295)
Shell	REG_SZ	Explorer.exe

Slika 5: Podešena vrijednost GinaDLL ključa



Slika 6: Tekstualna datoteka passlist.txt

Alat *FakeGINA* nije potpuno samostalan RootKit program jer sadrži samo funkcije za narušavanje procesa autentikacije, no može biti uključen u bilo koji RootKit program kao dodatni element.
U odnosu na ranije opisan klasičan proces autentikacije, ono što ovaj alat čini opasnim jest činjenica da se smješta između *winlogon* procesa i postojeće standardne biblioteke `msgina.dll`. Njegova funkcija je pohrana svih zaporki koje se upisuju tijekom procesa prijave na sustav u predefiniranu tekstualnu datoteku kako bi se neovlaštenom korisniku omogućio dolazak do korisničkih zaporki sustava. Proces autentikacije ovim alatom razlikuje se u odnosu na klasičan proces u koraku koji slijedi nakon upisa korisničkog imena i zaporke. Kada korisnik upiše tražene podatke, *FakeGINA* te podatke upisuje u tekstualnu datoteku `passlist.txt`, a zatim ih proslijedi legalnoj biblioteci `msgina.dll` te proces dalje ide klasičnim tokom. Opisani proces prikazan je na slici 7.



Slika 7: *Winlogon* proces s alatom *FakeGINA*

3.2. Metoda implementacije II

Opisani alat *FakeGINA,* koji predstavlja jednu od tehnika uporabe postojećeg sučelja za umetanje malicioznog koda među postojeće Windows funkcije, modificira isključivo onu funkcionalnost Windows okruženja koju je proizvođač dizajnirao da može biti promjenjiva. No, napadači često žele modificirati i ostale izvršne datoteke ili biblioteke koje nije moguće modificirati na ovako jednostavan način. U tom slučaju, napadač treba nadjačati *Windows File Protection* (WFP) funkcionalnost koja je sastavni dio Windows operacijskih sustava inačice 2000, XP i 2003, dok Windows ME inačica ima slično svojstvo pod nazivom *System File Protection* (SFP).
WFP servis ima funkciju provjere integriteta Windows sistemskih datoteka u svrhu sprečavanja njihove zamjene malicioznim datotekama korištenjem različitih malicioznih programa i tehnika. Maliciozni programi koji se koriste za zamjenu legitimnih datoteka sustava moraju biti spriječeni budući da prepisuju datoteke koje su važne za rad samog operacijskog sustava te ostalih sistemskih programa. WFP upravo zaštitom tih datoteka od prepisivanja štiti integritet operacijskog sustava. WFP štiti kritične sistemske datoteke kao što su `.dll`, `.exe`, `.ocx`, i `.sys` tipovi datoteka koje čine dio Windows operacijskog sustava. Pri obavljanju svoje funkcije WFP koristi potpise i katalog datoteka koji je generiran prilikom potpisivanja koda s ciljem potvrđivanja da su sistemske datoteke ispravne. Zamjena navedenih datoteka omogućena je jedino prilikom:

- instalacije *Windows Service Pack* sigurnosnih zakrpa korištenjem `update.exe` datoteke,
- instalacije *hotfix* komponenti Windows okruženja korištenjem `hotfix.exe` ili `update.exe` datoteka,
- nadogradnje sustava korištenjem `winnt32.exe` datoteke,
- nadogradnje Windows operacijskog sustava.

Ukoliko programi za zamjenu navedenih datoteka koriste bilo koju drugu metodu, WFP servis sve modificirane datoteke zamjenjuje njihovim originalima. Prilikom vraćanja originalnih datoteka WFP pregledava slijedeće lokacije:

- *DLLcache* mapu čija je putanja `c:\%SYSTEMROOT%\system32\dllcache`,
- datoteku `driver.cab` čija je putanja `c:\%SYSTEMROOT%\driver cache\I386\driver.cab`,
- originalnu Windows XP instalaciju spremljenu na lokalnom ili mrežnom disku,
- CD-ROM uređaj .

Napad na WFP servis neovlašteni korisnik može izvesti izmjenom standardnih postavki samog servisa. Standardne postavke WFP servisa locirane su `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\` *registry* zapisu i uključuju slijedeće ključeve:

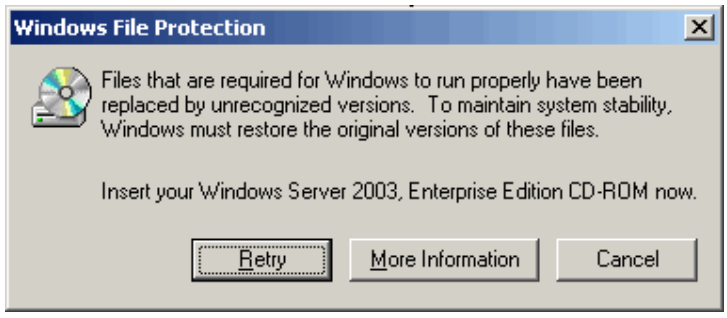
- `SFCScan`,
- `SFCQuota`,
- `SFCDllCacheDir`,
- `SFCShowProgress`,
- `SFCDisable`.

Ukoliko neki od ovih ključeva nije eksplicitno naveden u *registry* datoteci, sustav funkcionira korištenjem standardno podešenih vrijednosti. `SFCScan` ključ određuje kako će se WFP ponašati tijekom svakog ponovnog pokretanja sustava. `SFCQuota` ključ postavlja maksimalan kapacitet mape DLLCache u megabajtima. Ključ `SFCDllCacheDir` definira putanju mape *Dllcache*. Ključ `SFCShowProgress` specificira da li će tijek testiranja sustava biti prikazan na zaslonu računala. Najvažniji ključ u ovoj metodi svakako je ključ `SFCDisable` koji aktivira ili deaktivira WFP servis. `SFCDisable` ključ može poprimiti sljedeće vrijednosti:

- 0: aktiviran WFP,
- 1: deaktiviran WFP,
- 2: deaktiviran WFP do prvog ponovnog pokretanja sustava,
- 4: aktiviran WFP, ali neaktivni dijaloški okviri upozorenja o promjeni datoteka.

Da bi se vrijednosti ovog ključa mogle podesiti na 1 ii 2 na sustavu mora biti instaliran i aktiviran *kernel debugger*. Ukoliko on ne postoji, WFP se ne može deaktivirati.

Prilikom napada na WFP napadač može koristiti četiri metode. Prva metoda je brisanje zaštitne kopije određene datoteke iz mape `Dllcache` nakon čega slijedi zamjena legitimne datoteke onom napadačevom. No ovakvu akciju moguće je otkriti budući da će se prije izmjene datoteke korisniku prikazati dijaloški okvir upozorenja, što je jedan od razloga zašto napadači rijetko koriste ovu metodu. Upozorenje je prikazano na slici 8.



Slika 8: Upozorenje nakon brisanja datoteke iz mape Dllcache

Druga metoda je da se modificira sam način rada WFP servisa tako da umjesto legitimne lokacije `Dllcache` mape pregledava neku drugu lokaciju koju je moguće definirati izmjenom `SFCDllCacheDIR` vrijednost. Tom prilikom napadač može kreirati novu `Dllcache` mapu te konfigurirati sustav tako da koristi nju, a ne onu originalnu. Nakon izmjene predefinirane vrijednosti lokacije `Dllchache` mape i ponovnog pokretanja sustava, Windows operacijski sustav vrši provjeru datoteka pri čemu se javlja poruka prikazana na slici 9.

Slika 9: Poruka o provjeri integriteta datoteka

U slučaju ove provjere WFP, uporabom provjere digitalnih potpisa sistemskih datoteka operacijskog sustava, detektira da digitalni potpisi iz druge `Dllcache` mape ne odgovaraju te se generira poruka o grešci prikazana na slici 8.
Treća metoda sastoji se od podešavanja `SFCDisable` vrijednosti tako da se upiše vrijednost koja je preporučljiva od strane Microsoft ukoliko se želi onemogućiti WFP servis. Podešavanjem vrijednosti ključa `SFCDisable` na 1, instalacijom *kernel debbuger*-a i ponovnim pokretanjem sustava, napadač može deaktivirati WFP.
Četvrta metoda napada na WFP uključuje promjenu postavki `SFCDisable` vrijednosti tako da se upiše vrijednost koja nije dokumentirana od strane proizvođača. Prema provedenim analizama, vrijednost `0xFFFFFF9D` može u potpunosti deaktivirati WFP, ali tek nakon ponovnog pokretanja sustava.
Jedan od primjera deaktiviranja provjere integriteta datoteka je dobro poznati *Code Red II* crv koji napada Windows operacijske sustave sa pokrenutim *Internet Information Services* (IIS) Web poslužiteljem. Naime, iako *Code Red II* prema svojim osnovnim karakteristikama spada u kategoriju crva, također sadrži i određene RootKit komponente koja su ga činile znatno destruktivnijim u odnosu na prvu inačicu (Code Red I). Osim onemogućavanja WFP servsa Code Red II također je na sustav postavljao i malicioznu inačicu Windows Explorer programa koja je prikravala maliciozne aktivnosti na sustavu. Detalje o načinima širenja i osnovnim funkcionalnostima *Code Red II* crva može se pronaći u dokumentu na adresi http://www.cert.hr/filehandler.php?did=35.

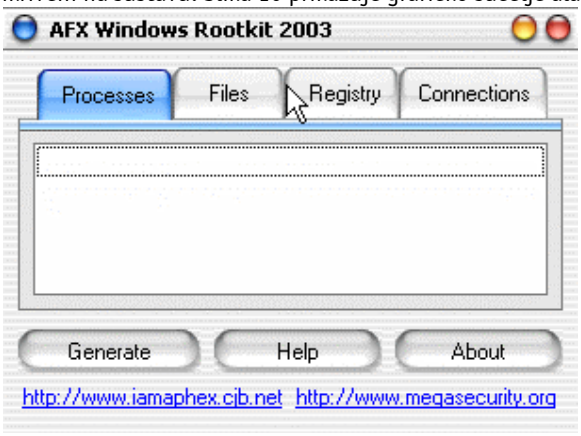
3.3. Metoda implementacije III

Treća metoda koja omogućuje implementaciju aplikacijskih RootKit alata je uporaba *DLL injection* i *API hooking* tehnika napada s ciljem manipulacije aktivnim procesima u ranoj memoriji sustava. U usporedbi s dvije ranije opisane metode napadači sve češće koriste pravo ovu metodu s obzirom da im omogućuje ubacivanje vlastitog malicioznog koda izravno u memoriju aktivnih procesa na sustavu.
Na Windows platformama istovremeno je aktivan prilično velik broj procesa. Ono što napadač čini jest ubacivanje vlastitog malicioznog koda u aktivan proces koji prepisuje postojeće funkcije u ciljnom procesu i aktivira maliciozni kod unutar istoga. Za razumijevanje ove metode potrebno je razlikovati dva osnovna tipa izvršnog koda na Windows operacijskim sustavima. To su `.exe` programi te dinamičke biblioteke s ekstenzijom `.dll`. Izvršne `.exe` datoteke koriste dinamičke `.dll` biblioteke i koriste ih za izvršavanje različitih funkcija. Napadač koristi tehniku *DLL injection* kada želi natjerati aktivni proces da u svoj memorijski prostor učita malicioznu `.dll` datoteku.
Ova tehnika sastoji se od nekoliko koraka:

- rezervacija prostora unutar ciljnog procesa za umetanje `DLL` koda,
- rezervacija prostora unutar ciljnog procesa za parametre koji su neophodni za ubacivanje `DLL` koda,
- upisivanje `DLL` naziva i koda u memoriju ciljnog procesa,
- kreiranje niti unutar ciljnog procesa koja će pokrenuti umetnuti `DLL` kod,
- oslobađanje resursa ciljnog procesa nakon što je izvršen `DLL` kod.

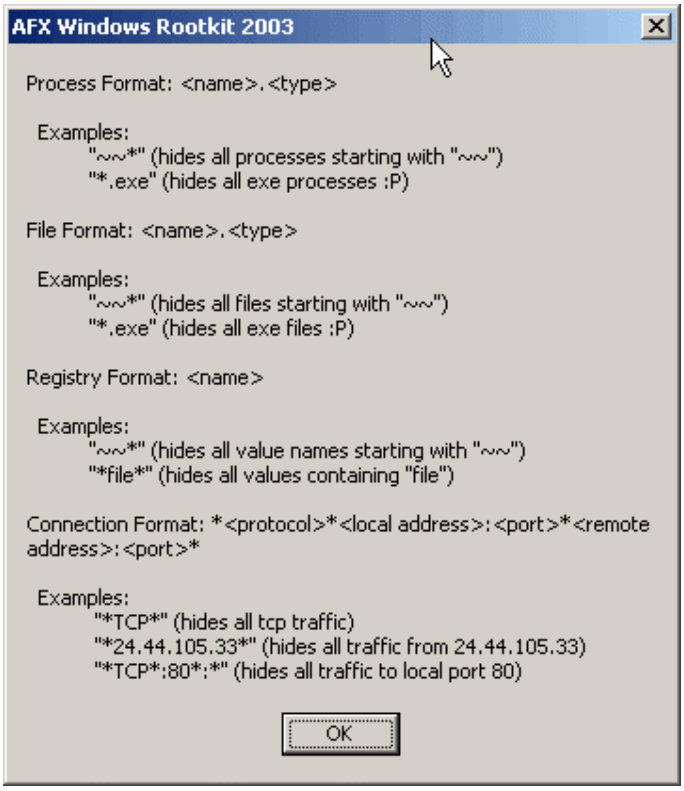
Opisanu tehniku napadač koristi za umetanje bilo koje vrste koda u bilo koji aktivan proces sustava. Kako bi se omogućilo umetanje malicioznih DLL biblioteka, napadač na sustavu mora imati *Debug Programs* ovlasti koje su inače potrebne ukoliko se žele koristiti specijalizirani programi za analizu grešaka u radu programa (engl. *debugger*).

Na upravo opisanu *DLL Injection* tehniku nadovezuje se *API* (*Application programming interface*) *hooking* koncept koji napadaču omogućuje da u legitimne DLL datoteke uključi maliciozne *RootKit* funkcije koje inače nisu dostupne. Provođenje ovakvih napada najčešće se provodi korištenjem automatiziranih alata koa što su *MadCodeHook*, *APIHijack*, *EliRT*, itd.
U ovom dokumentu ukratko će biti opisan alat *AFX Windows RootKit* koji koristi *DLL injection* i *API hooking* tehnike. AFX Windows RootKit je aplikacija koja prikriva aktivne procese i servise na sustavu, mape, datoteke, *registry* zapise, module, *TCP* i *UDP* portove, i sl. Za razliku od mnogih drugih *RootKit* alata za Windows okruženje, ovaj alat ne omogućuje stražnji ulaz u sustav. Iz tog razloga najprije je potrebno koristiti neki od alata koji će omogućiti stražnji ulaz u sustav, nakon čega će se *AFX Windows RootKit* alatom prikriti prisutnost istoga.
Ovaj *Rootkit* alat sastoji se od samo jedne izvršne datoteke koja se smješta u određenu mapu na lokalnom disku. Inačica alata 2003 ima grafičko sučelje koje omogućuje konfiguraciju i generiranje objekata koji će biti prikriveni na sustavu. Slika 10 prikazuje grafičko sučelje alata.



Slika 10: Sučelje alata AFX Widnows RootKit 2003

Klikom na gumb **Help** otvara se sučelje koje korisniku daje osnovne upute za podešavanje programa. Slika 11 prikazuje primjere sintaksi koje sadrži spomenuti dijaloški okvir.



Slika 11: Dijaloški okvir pomoći alata

Alat omogućuje provođenje različitih aktivnosti kao što je npr. skrivanje određenog procesa, tako da se na kartici **Processes** unese naziv ili tip procesa. Klikom na gumb **Generate**, generira se izvršna datoteka, a njenim pokretanjem skriva se navedeni proces. Isti princip rada je i za ostala tri objekta: datoteke, *registry* zapise i mrežne konekcije.

4. Mjere zaštite

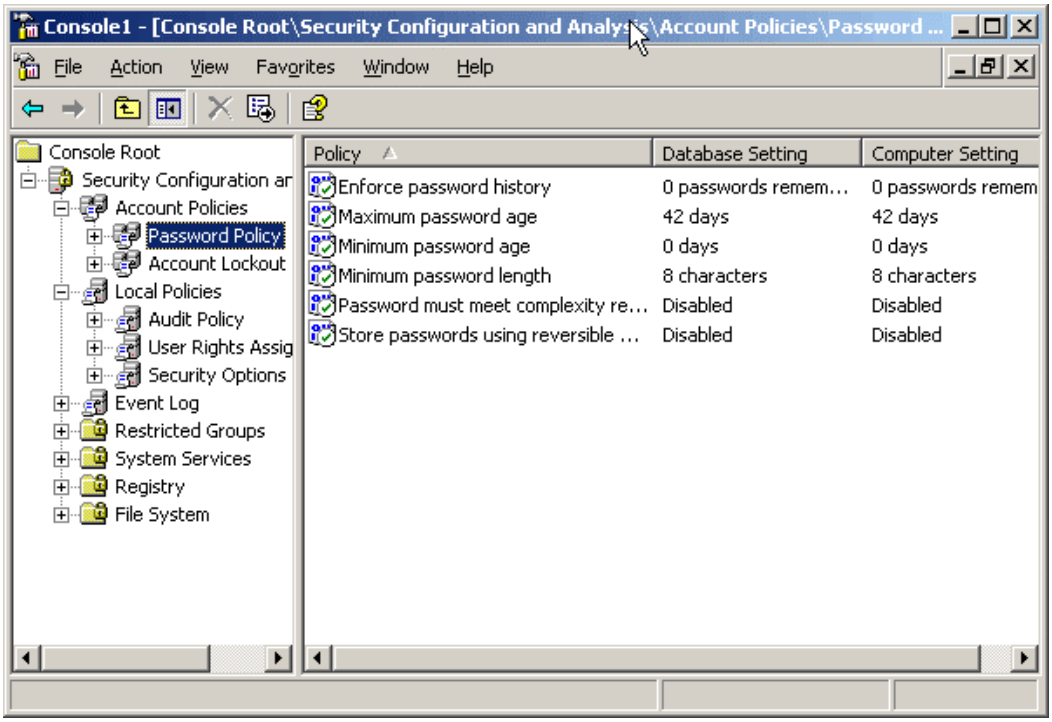
Mogućnosti zaštite od *RootKit* alata svode se na dva osnovna pristupa :

- prevencija,
- detekcija.

4.1. Prevencija

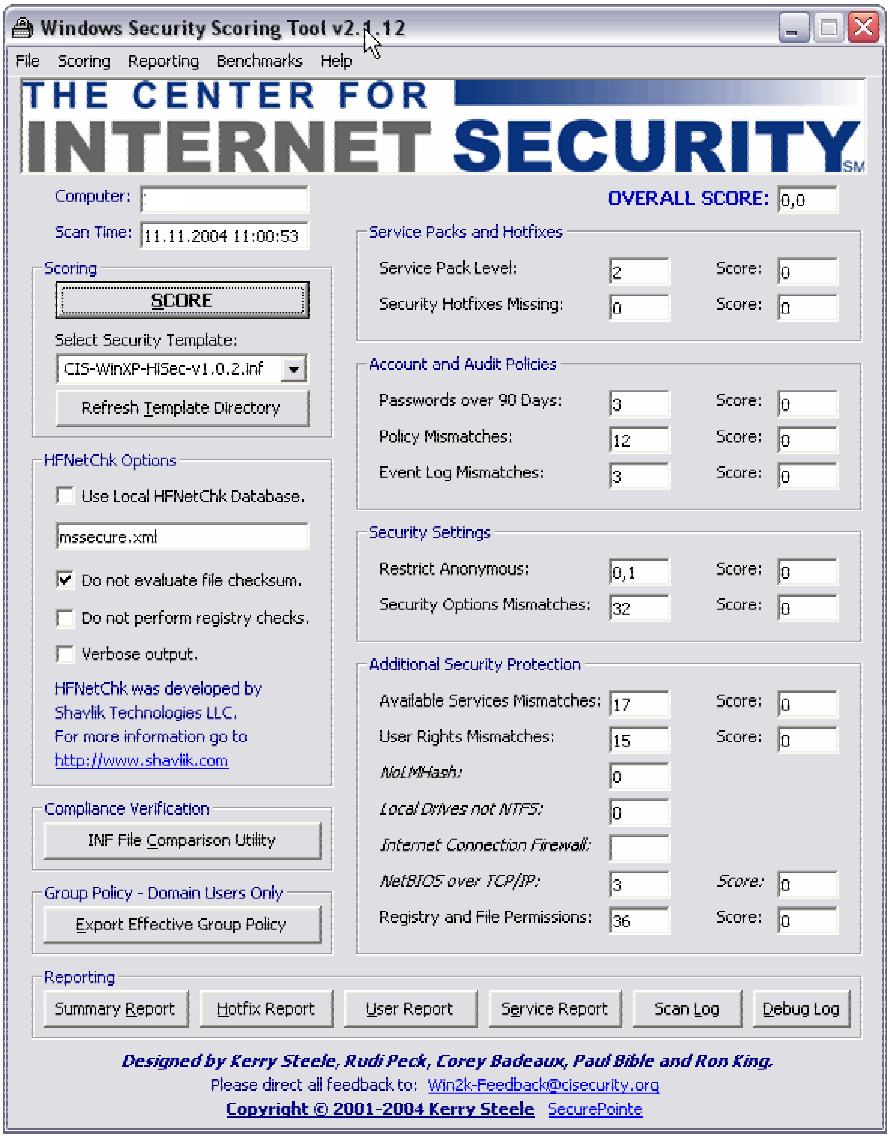
Obzirom da postavljanje RootKit alata zahtjeva od napadača preuzimanje potpune kontrole nad sustavom, prevencija ove vrste napada sastoji se, prije svega, od zaštite operacijskog sustava nadogradnjom sigurnosnih zakrpa i ostalim tehnikama podizanja razine sigurnosti (eng. *hardening*) sustava.

Podizanje razine sigurnosti sustava moguće je postići korištenjem Windows predložaka (eng. *security template*) koji dolaze u obliku odgovarajućih datoteka sa predefiniranim sigurnosnim postavkama. U svakom novo kreiranom predlošku postavke su predefinirane s vrijednošću "nije podešeno" što znači da se stanje sigurnosti na ciljnom sustavu ne mijenja dok administrator sustava samostalno ne izvrši konfiguraciju. Predlošci se nalaze na lokaciji `C:\%SYSTEMROOT%\security\templates` . Koriste se pomoću *Microsoft Management Console* (MMC) konzole u koju se učitavaju predlošci te konfiguriraju i primjenjuju na ciljni sustav. Na slici 12 prikazan je predložak `setup security.inf`.



Slika 12: Predložak setup security.inf

Dodatne informacije o podizanju razine sigurnosti sustav korištenjem predložaka mogu se pronaći na Microsoftovim stranicama http://www.microsoft.com/security/. No, jedan od najboljih nezavisnih predložaka za Windows 2000 sustave može se pronaći na referentnoj adresi http://www.cisecurity.org. Predložak su zajednički izradili The Center of Internet Security (CIS), the National Security Agency (NSA) i SANS Institut, a sadrži osnovne sigurnosne postavke koje odgovaraju većini okruženja. CIS je izradio alat koji vrši procjenu stanja ciljnog sustava u odnosu na predložak koji su izradili. Alat i pripadajući dokumenti može se pronaći na navedenoj referentnoj adresi http://www.cisecurity.org. Alat ocjenjuje postavke sustava ocjenama od 0 do 10, pri čemu viši rezultat odgovara većoj razini sigurnosti. Slika 13 prikazuje sučelje alata.



Slika 13: Sučelje alata Windows Security Scornig Tool

4.2. Detekcija

Samo prevencija kao mehanizam zaštite nije dovoljna kako bi se sustav u potpunosti zaštitio od RootKit alata. Njihova pravovremena detekcija također je izuzetno bitna. Prva i osnovna aktivnost koju svaki administrator mora provoditi je redovita provjera sistemskih zapisa (eng. *system logs*) koji indiciraju potencijalno neovlaštene akcije. Nadalje, jedna od najboljih metoda detekcije je provjera integriteta datoteka. Ranije spomenuti WFP servis nudi mogućnost detekcije prisutnosti *RootKit* alata putem dijaloških okvira upozorenja te na njih svakako treba obratiti pažnju. Osim toga, preporučljiva je uporaba dodatnih alata za provjeru integriteta datoteka. Jedan od takvih alata je *FileCheckMD5* koji je besplatan za korištenje, a moguće ga je dohvatiti s adrese http://www.brandonstaggs.com/filecheckmd5.html.

Ulogu detekcije RootKit alata mogu obavljati i antivirusni programi. Ukoliko AV program sadrži odgovarajuće potpise instalacija malicioznih programa biti će onemogućena već prilikom snimanja alata na lokalni disk ciljnog sustava. Primjer su u ovom dokumnetu spomenuti alati *FakeGINA*, *Code*

Red II i *AFX Windows RootKit* koje ažuriran antivirusni program detektira i onemogućuje njihovu uporabu. Također, korisno je upotrebljavati i razne alate treće strane koji analiziraju stanje sustava.

5. Zaključak

U rukama napadača RootKit alati izuzetno su opasni. Zaštitu od njih potrebno je osigurati na različitim razinama kako bi se postigla što viša razina sigurnosti. U ovom dokumentu opisane su tri osnovne metode implementacije RootKit alata te načini na koje svaka do njih omogućuje preuzimanje kontrole nad sustavom. Također su navedene i osnovne metode zaštite od malicioznih programa ovog tipa sa konkretnim primjerima sigurnosnih alata koji mogu pomoći u tom smislu. Sve navedene aktivnosti i tehnike navedene su prvenstveno u smislu pomoći sistem administratorima i sigurnosnim stručnjacima kako bi ih se upozorilo i na ovakve oblike prijetnje informacijskim sustavima.

6. Reference

[1] Skoudis, Edward: Malware - fighting malicious code, Prentice Hall, New Jersey, 2004.
[2] Manap, Saliman: Rootkit: Attacker undercover tools
http://www.giac.org/practical/Saliman_Manap_GSEC.doc
[3] CARNet CERT & LSS: Analiza CodeRed.F crva
http://www.cert.hr/filehandler.php?did=35
[4] Microsoft Security
http://www.microsoft.com/security/
[5] The Center for Internet Security
http://www.cisecurity.org
[6] FakeGINA
http://www.ntsecurity.nu/toolbox/fakegina/
[7] AFX Windows RootKit 2003
http://www.iamaphex.cjb.net/
[8] FileCheckMD5
http://www.brandonstaggs.com/filecheckmd5.html