Wicked malware persistence methods

Hasherezade (@hasherezade) - malware analyst, technical blogger

Malwarebytes

Agenda

1. Basics of persistence
2. Hunting for malware persistence artifacts
3. Making persistence hard to spot (tricks + real life examples)

Basics of persistence

Basics of persistence

Exploitation -> Infection -> **Persistence**

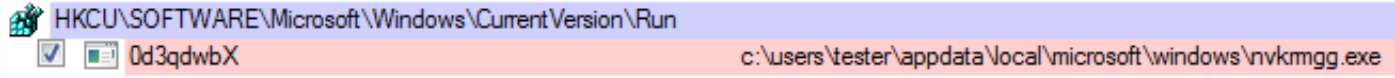
Basics of persistence

- WHO? Most of the malware needs it (except some ransomware)
- WHY? To start the application after each reboot
- HOW? Windows offers various legitimate persistence ways – let’s recall them...

Basics of persistence – Run/RunOnce keys

- Registry keys, i.e.:
  - HKCU\Software\Microsoft\Windows\CurrentVersion\Run
  - HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce
  - HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run

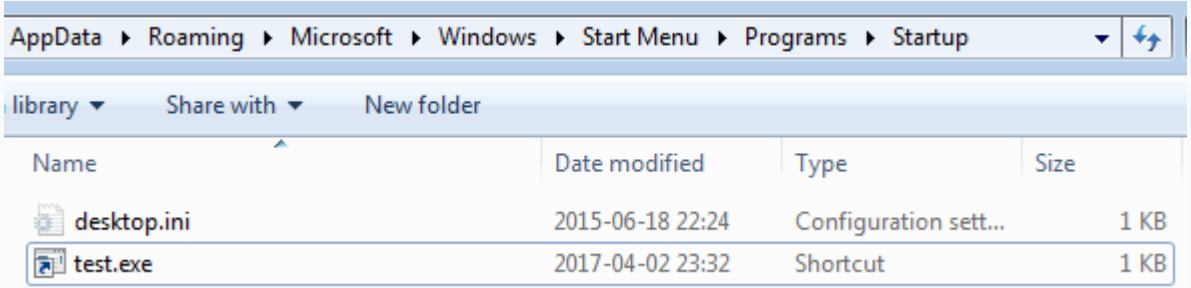
The **most commonly used technique** (also by malware)...



https://support.microsoft.com/pl-pl/help/179365/info-run,-runonce,-runservices,-runservicesonce-and-startup

Basics of persistence – Startup link

```
%APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup
```



Basics of persistence – Scheduled tasks

- Task scheduler view:

Name	Status	Triggers	Next Run Time	Last Run Time	Last Run Result	Author
Bot	Ready	At 00:00 every day - After triggered, repeat every 00:01:00 for a duration of 1 day.	2016-10-20 16:57:00	2016-10-20 16:56:00	(0xFFFFFFFF)	Author N

General | Triggers | Actions | Conditions | Settings | History (disabled)

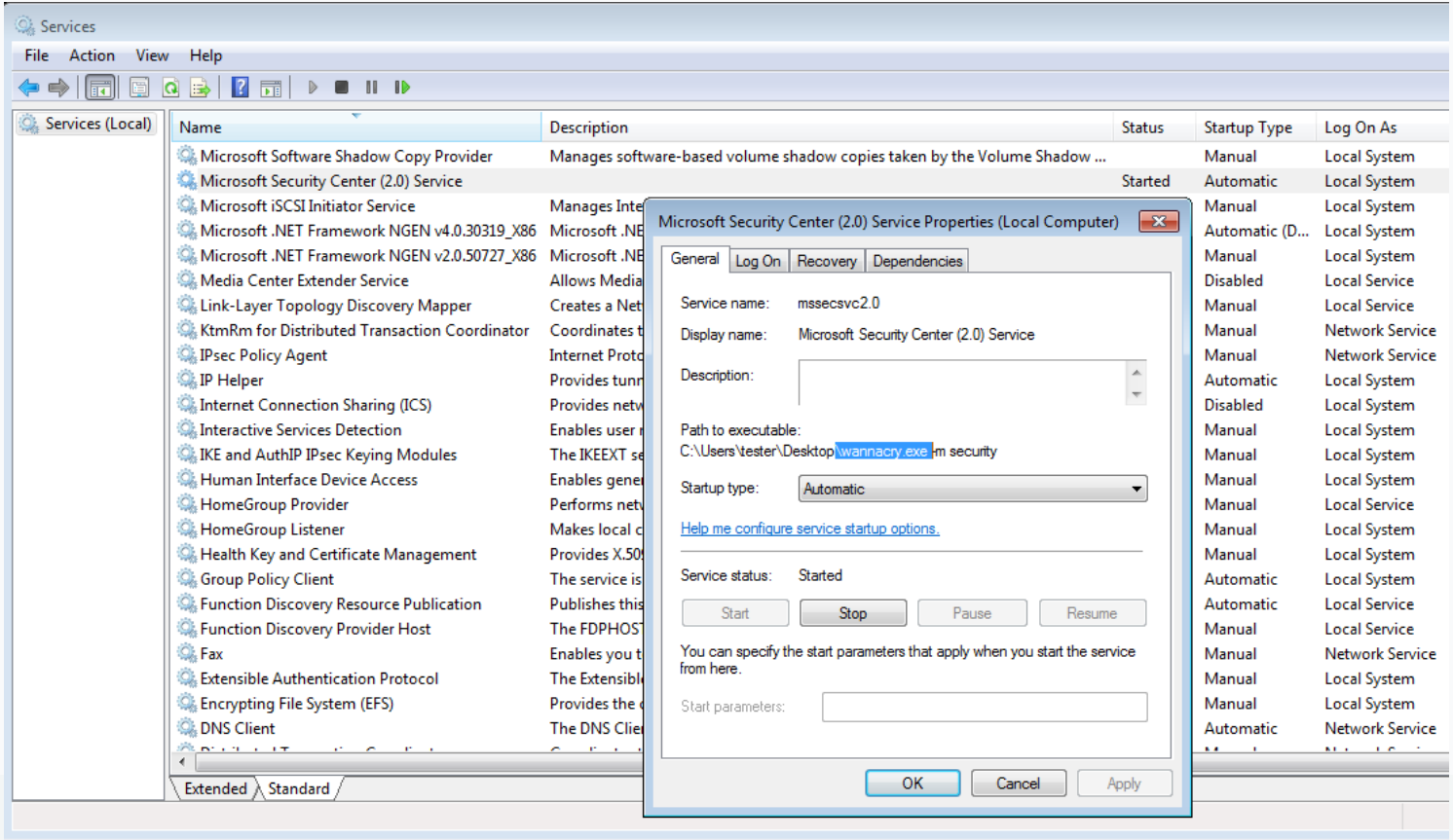
When you create a task, you must specify the action that will occur when your task starts. To change these actions, open the task property pages using the Properties command.

Action	Details
Start a program	C:\Users\tester\AppData\Roaming\trick.exe

https://blog.malwarebytes.com/threat-analysis/2016/10/trick-bot-dyrezas-successor/

Malwarebytes

Basics of persistence – System Services



UAC Bypass required

Malwarebytes

Basics of persistence – System Services

UAC Bypass required

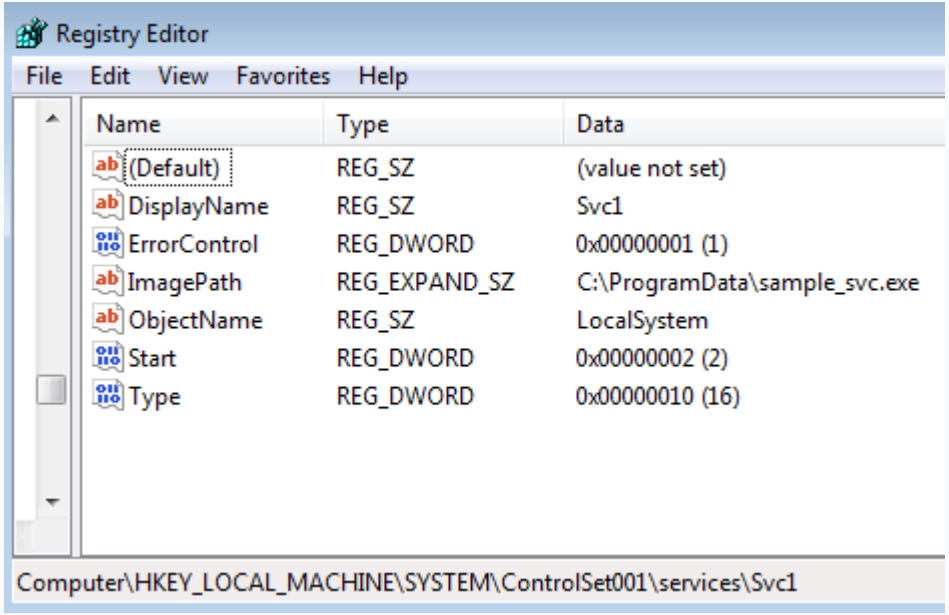
- Administrator rights required
- Creating a service:

```
sc create <service_name>
binPath= <service_path>
DisplayName= <service_display_name>
start= auto
```

Basics of persistence – System Services

- Related registry keys:
  - HKLM\SYSTEM**ControlSet001**\services\<service name>
  - HKLM\SYSTEM**ControlSet002**\services\<service name>
  - HKLM\SYSTEM**CurrentControlSet**\services\<service name>

Basics of persistence – System Services



Hunting for persistence artifacts

Hunting for persistence artifacts – autoruns

Autoruns - Sysinternals: www.sysinternals.com

File Entry Options Help

Filter:

KnownDLLs | Winlogon | Winsock Providers | Print Monitors | LSA Providers | Network Providers | WMI | Sidebar Gadgets | Office

Everything | Logon | Explorer | Internet Explorer | Scheduled Tasks | Services | Drivers | Codecs | Boot Execute | Image Hijacks | AppInit

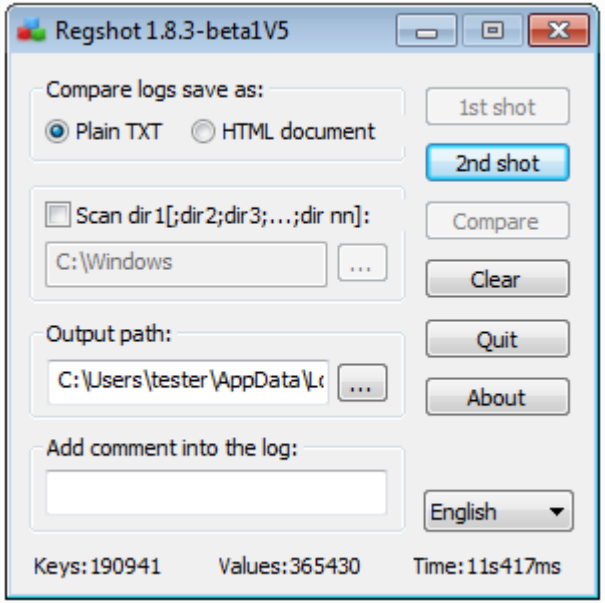
Autorun Entry	Description	Publisher	Image Path	Timestamp	VirusTotal
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run				2017-02-22 02:41	
SunJavaUpdateSched	Java Update Scheduler	Oracle Corporation	c:\program files\common fil...	2016-12-13 05:21	
VBoxTray	VirtualBox Guest Additions ...	Oracle Corporation	c:\windows\system32\vbox...	2015-05-13 17:11	
HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components				2015-06-18 22:47	
Microsoft Windows	Windows Mail	Microsoft Corporation	c:\program files\windows m...	2009-07-14 01:42	
HKLM\SOFTWARE\Classes\Protocols\Filter				2017-04-02 22:55	
text/xml	Microsoft Office XML MIME...	Microsoft Corporation	c:\program files\common fil...	2013-12-17 22:15	
HKLM\SOFTWARE\Classes\Protocols\Handler				2017-04-02 22:55	
ms-help	Microsoft® Help Data Servi...	Microsoft Corporation	c:\program files\common fil...	2013-08-15 02:04	
osf	Microsoft Office 2013 comp...	Microsoft Corporation	c:\program files\microsoft of...	2013-12-17 22:11	
HKLM\Software\Classes\Directory\Background\ShellEx\ContextMenuHandlers				2009-07-14 06:41	
Gadgets	Sidebar droptarget	Microsoft Corporation	c:\program files\windows si...	2009-07-14 03:09	
HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Browser Helper Objects				2017-04-02 22:55	
Java(tm) Plug-In 2 SSV Helper	Java(TM) Platform SE binary	Oracle Corporation	c:\program files\java\jre1.8....	2016-12-13 05:00	
Java(tm) Plug-In SSV Helper	Java(TM) Platform SE binary	Oracle Corporation	c:\program files\java\jre1.8....	2016-12-13 05:00	
Office Document Cache Handler	Microsoft Office Document ...	Microsoft Corporation	c:\program files\microsoft of...	2013-12-17 22:06	
Task Scheduler					
\Microsoft Office 15 Sync Maintenanc...	Microsoft Office Document ...	Microsoft Corporation	c:\program files\microsoft of...	2013-12-17 22:04	
\Microsoft\Office\Office 15 Subscriptio...	Office Subscription Licensin...	Microsoft Corporation	c:\program files\common fil...	2013-12-17 22:08	
\Microsoft\Office\OfficeTelemetryAgen...	Office Telemetry Agent	Microsoft Corporation	c:\program files\microsoft of...	2013-12-17 22:13	
\Microsoft\Office\OfficeTelemetryAgen...	Office Telemetry Agent	Microsoft Corporation	c:\program files\microsoft of...	2013-12-17 22:13	
\Microsoft\Windows Defender\MP Sc...	Microsoft Malware Protectio...	Microsoft Corporation	c:\program files\windows d...	2009-07-14 01:37	

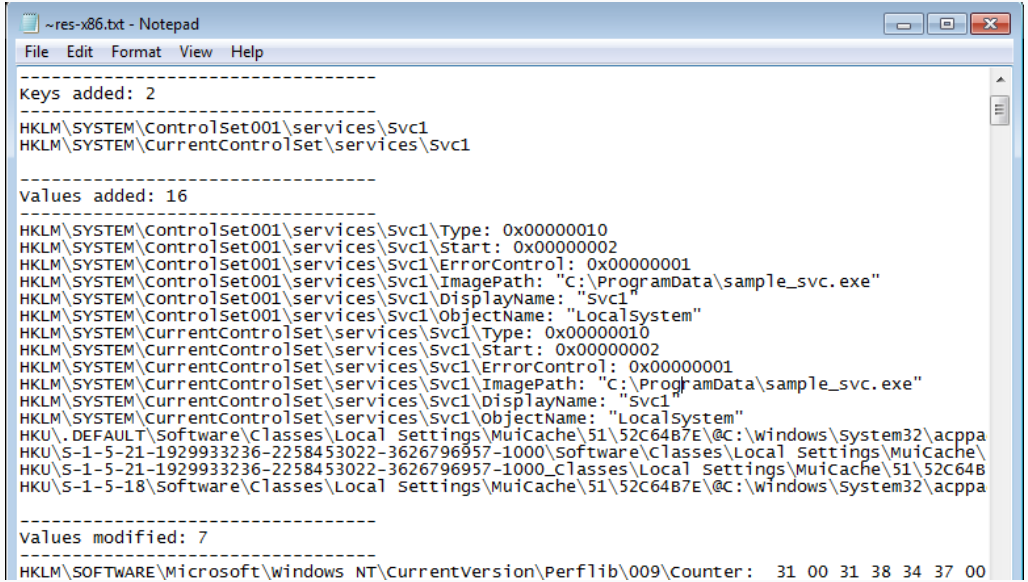
Ready. | Windows Entries Hidden.

Sysinternals: autoruns.exe

Malwarebytes

Hunting for persistence artifacts – Regshot





RegShot – monitoring changes in the Windows registry

Malwarebytes

Hiding persistence – tricks and examples

Hiding persistence – how?

1. **Typical methods**, but with **extra measures** to cover/protect
2. **Abuse** of other mechanisms of the system for **automated injection**, i.e.:
   - AppInit_DLL, COM Hijacking, Shims, MS Application Verifier Provider ("DoubleAgent" technique), etc
3. **User-triggered** persistence – hide in other elements, that are likely to be clicked/deployed by a user

Typical methods + extra measures

- Last minute persistance (i.e. Dridex v. 3)
- Make sample inaccessible: ADS, special folders (i.e. Diamond Fox)
- Hide in the plain sight:
  - behind legitimate applications: Korplug
  - hide the executable in the windows registry - „fileless” malware
  - use scripts to load malicious modules – often Powershell

Last minute persistence

1. Inject and delete yourself -> **no malicious PE on the disk**
2. Set callbacks on messages: **WM_QUERYENDSESSION**, **WM_ENDSESSION** to detect when the system is going to shut down
3. On shutdown event detected: write yourself on the disk and the Run key for the persistence
4. On system startup: delete the Run key, go to 1.

https://www.cyberbit.net/wp-content/uploads/2016/09/Analysis-of-Dridex-AnD-for-IT.pdf

Make file invisible/inaccessible – special folders

- Example: Diamond Fox:

HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

spoolsv	c:\users\tester\appdata\local\temp\lpt8.{20d04fe0-3aea-1069-a2d8-08002b30309d}\spoolsv.exe

Normal persistence key

AppData ▸ Local ▸ Temp ▸ Search Temp

Name	Date modified	Type	Size
Low	2017-03-14 16:20	File folder	
lpt8.{20D04FE0-3AEA-1069-A2D8-08002B30309D}	2017-03-14 16:31	File folder	

Not normal folder name

`lpt8.{20D04FE0-3AEA-1069-A2D8-08002B30309D}`

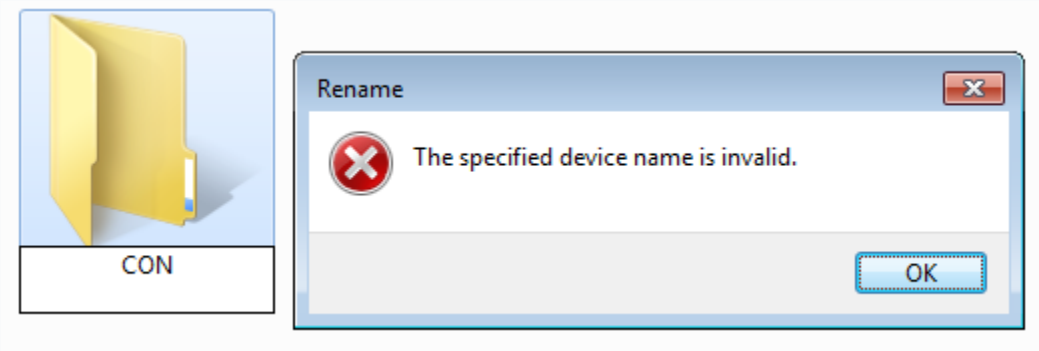
https://blog.malwarebytes.com/threat-analysis/2017/03/diamond-fox-p1/

Malwarebytes

Make file invisible/inaccessible – special folders

- Restricted names – starting from:

```
CON, PRN, NUL, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6,
LPT7, LPT8, LPT9, COM1, COM2, COM3, COM5, COM6,
COM7, COM8, COM9
```

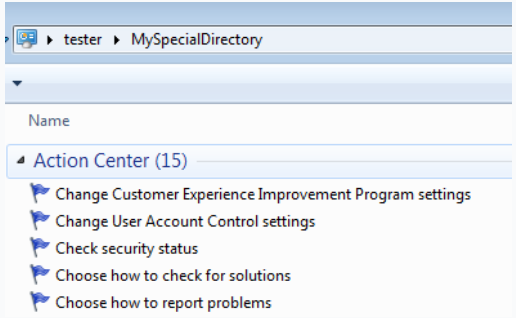


http://windows.mercenie.com/windows-xp/create-folder-any-name/

Make file invisible/inaccessible – special folders

- Special CLSIDs (examples):

```
GodMode.{ED7BA470-8E54-465E-825C-99712043E01C}
Administrative Tools.{D20EA4E1-3957-11d2-A40B-0C5020524153}
All Tasks.{ED7BA470-8E54-465E-825C-99712043E01C}
History.{ff393560-c2a7-11cf-bff4-444553540000}
```



Clicking on folder triggers different action -> no access to the content

http://www.thewindowsclub.com/the-secret-behind-the-windows-7-godmode

Make file invisible/inaccessible – special folders

Benefits from using special folders:

- User cannot access the content – special CLSID triggers event other than opening the folder
- Cannot be removed/renamed in a typical way – restricted name prevents operating on the folder

http://www.thewindowsclub.com/the-secret-behind-the-windows-7-godmode

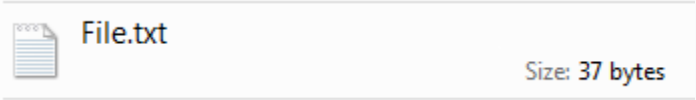
Make file invisible/inaccessible – ADS

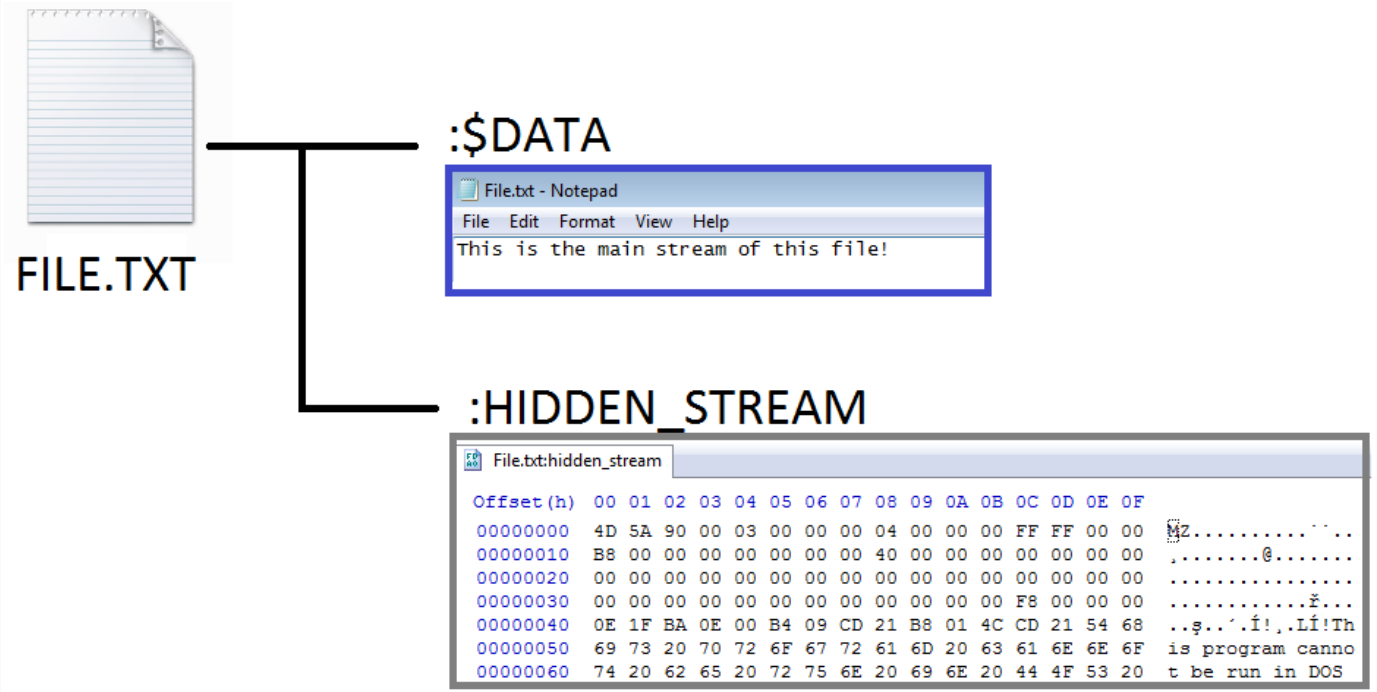
- ADS - Alternate Data Streams
  - A feature of NTFS file system
  - Implemented, but practicaly not used by Windows...
  - Only the main stream of the file is listed/accessible in a typical way
  - Format:

```
<filename.extension>:<alternate_stream_name>
```

https:// hshrzd.wordpress.com/2016/03/19/introduction-to-ads-alternate-data-streams /

Make file invisible/inaccessible – ADS





https:// hshrzd.wordpress.com/2016/03/19/introduction-to-ads-alternate-data-streams /

Malwarebytes

Make file invisible/inaccessible – ADS

1. Get a **demo.dll**: https://goo.gl/wl7ZNJ
2. Copy the DLL into ADS of some file, i.e.:

```
type demo.dll > test.txt:demo
```

3. Deploy the DLL from the alternate stream (DllMain):

```
regsvr32.exe /s test.txt:demo
```

4. Deploy a specific function (i.e. *Test1*) from the DLL:

```
rundll32.exe test.txt:demo,Test1
```

Make file invisible/inaccessible – ADS

Command Prompt

```
C:\Users\tester\ads_tests>dir
 Volume in drive C has no label.
 Volume Serial Number is 448D-3B2B

 Directory of C:\Users\tester\ads_tests

2017-05-08  20:05    <DIR>          .
2017-05-08  20:05    <DIR>
2017-05-08  20:03                37 File.txt
               1 File(s)             37 bytes
               2 Dir(s)  11 699 998 720 bytes free

C:\Users\tester\ads_tests>rundll32.exe File.txt:hidden_stream,Test1

C:\Users\tester\ads_tests>
```

Test DLL

Test DLL loaded!

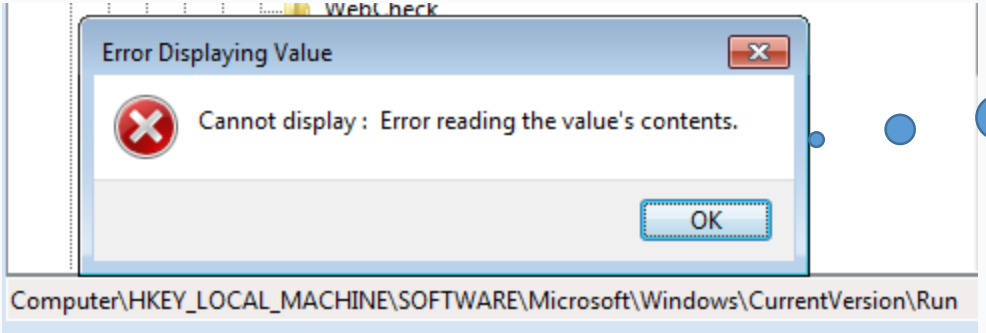
OK

https:// hshrzd.wordpress.com/2016/03/19/introduction-to-ads-alternate-data-streams /

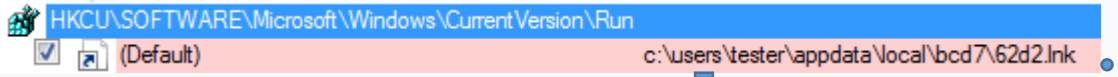
Malwarebytes

Make registry keys inaccessible

- NULL character at the beginning of the key
- Example: Kovter



Malformed key: Regedit cannot display it



Still can be viewed by Autoruns...

```
\0c:\\users\\tester\\appdata\\local\\bcd7\\62d2.lnk
```

https://blog.malwarebytes.com/threat-analysis/2016/07/untangling-kovter/

Malwarebytes

Make registry keys hard to spot

- By default, *Autoruns* hides keys leading to Microsoft apps

- Example: Moker trojan

Everything | Logon | Explorer | Internet Explorer | Scheduled Tasks | Services | Drivers | Codecs

Autorun Entry	Description	Publisher	Image Path	Timestamp
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run				2017-04-05 15:42
SunJavaUpdat...	Java Update Scheduler	Oracle Corporation	c:\program files\common fil...	2016-12-13 05:21
VBoxTray	VirtualBox Guest Additions ...	Oracle Corporation	c:\windows\system32\vbox...	2015-05-13 17:11
HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components				2017-02-25 23:36

Registry Editor

File Edit View Favorites Help

Name	Type	Data
(Default)	REG_SZ	(value not set)
KB1080030	REG_SZ	Rundll32.exe SHELL32.DLL,ShellExec_RunDLL "C:\ProgramData\KB1080030.exe"
pluginj382dew1i	REG_SZ	Rundll32.exe SHELL32.DLL,ShellExec_RunDLL "C:\ProgramData\pluginj382dew1i.exe"
SunJavaUpdateSched	REG_SZ	"C:\Program Files\Common Files\Java\Java Update\jusched.exe"
VBoxTray	REG_SZ	C:\Windows\system32\VBoxTray.exe

Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

Autoruns shows only two keys...

But there are more...

Malware is deployed by a Microsoft application: Rundll32

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
@="Rundll32.exe SHELL32.DLL,ShellExec_RunDLL \"C:\\ProgramData\\test.exe\""
```

https://blog.malwarebytes.com/threat-analysis/2017/04/elusive-moker-trojan/

Malwarebytes

Hide behind legitimate applications (DLL abuse)

- Korplug (PlugX) - spyware
  - Uses vulnerable, digitaly signed, legitimate application (old AV products)
  - Exploits DLL side loading (DLL is a decoder)
  - The real malware is decrypted in memory -> **no malicious PE file on the disk** -> hard to detect!

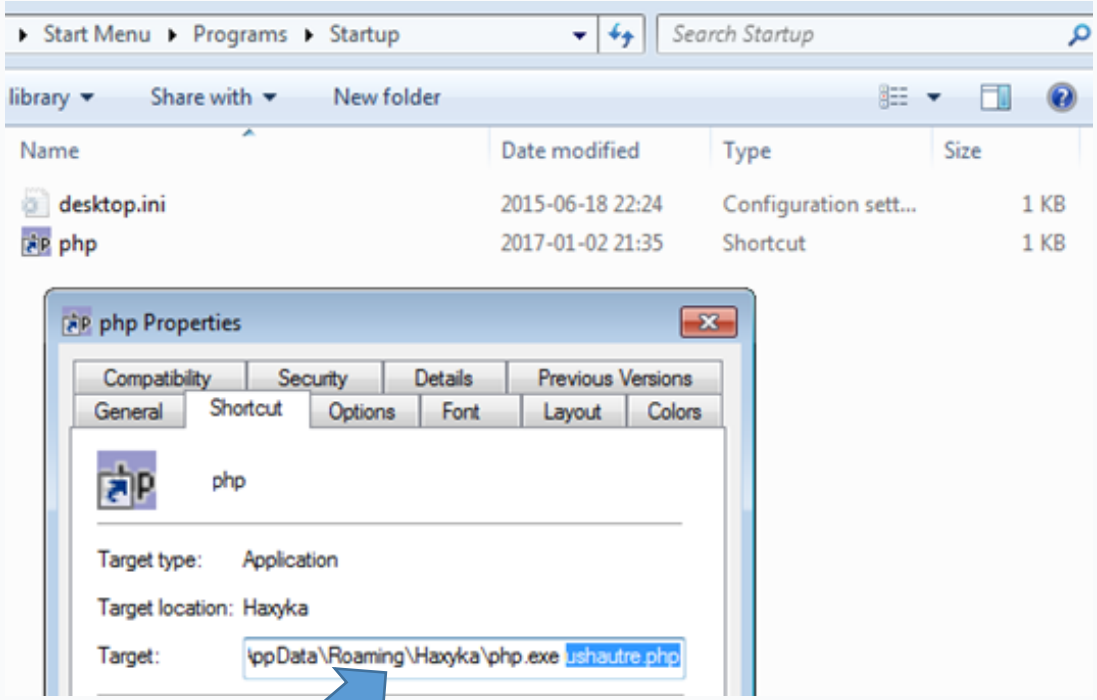
ang	2015-06-26 14:54	File	1 KB
McAfee.exe	2013-08-29 08:50	Application	138 KB
McUtil.dll	2013-08-29 08:50	Application extens...	4 KB
McUtil.dll.mc	2013-08-29 08:50	MC File	115 KB
tjuiiarpujhx	2016-05-19 04:47	File	2 KB
vekmfmujufficwveip	2013-08-29 08:50	File	59 KB

https://blog.malwarebytes.com/threat-analysis/2016/08/unpacking-the-spyware-disguised-as-antivirus/

Malwarebytes

Hide behind legitimate applications (script)

- Terdot Zbot (Zeus-based banking trojan):




```
C:\AppData\Roaming\Haxyka\php.exe ushautre.php
```

Hide behind legitimate applications (script)

- Terdot Zbot (Zeus-based banking trojan)
  - Uses a legitimate application (PHP)
  - PHP is used to deploy obfuscated script
  - Script decrypts and loads the malware
  - The real malware is revealed in memory -> **no malicious PE file on the disk** -> hard to detect!

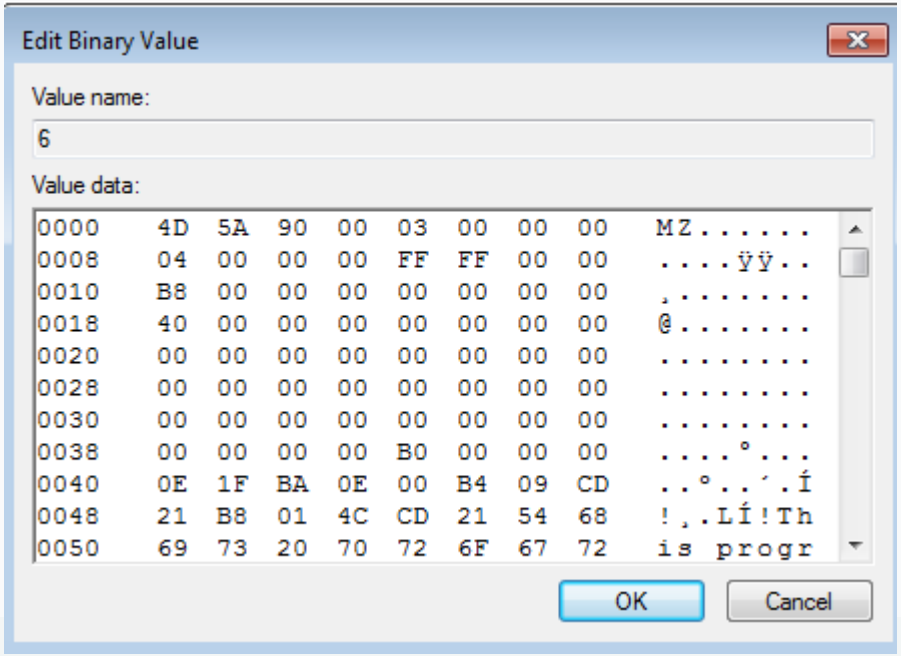
https://blog.malwarebytes.com/cybercrime/2017/01/zbot-with-legitimate-applications-on-board/

Hide code in the registry

- So called „fileless” malware
  - Phasebot
  - Poweliks
  - Gootkit
  - Kovter
  - PoshSpy (APT29) using WMI component and PowerShell
  - Others...

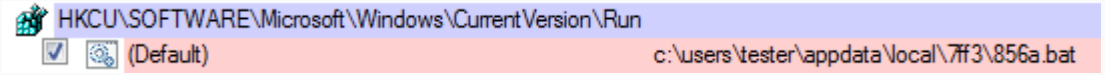
Hide code in the registry

- Trivial case - PE file saved in the registry key:

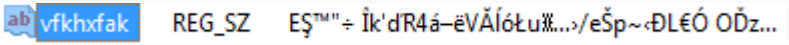


Hide code in the registry (multilayer: Kovter)

- Kovter – click-fraud malware
  - Persistence is achieved by a **basic Run key** – but the flow leading to the malicious executable is obfuscated



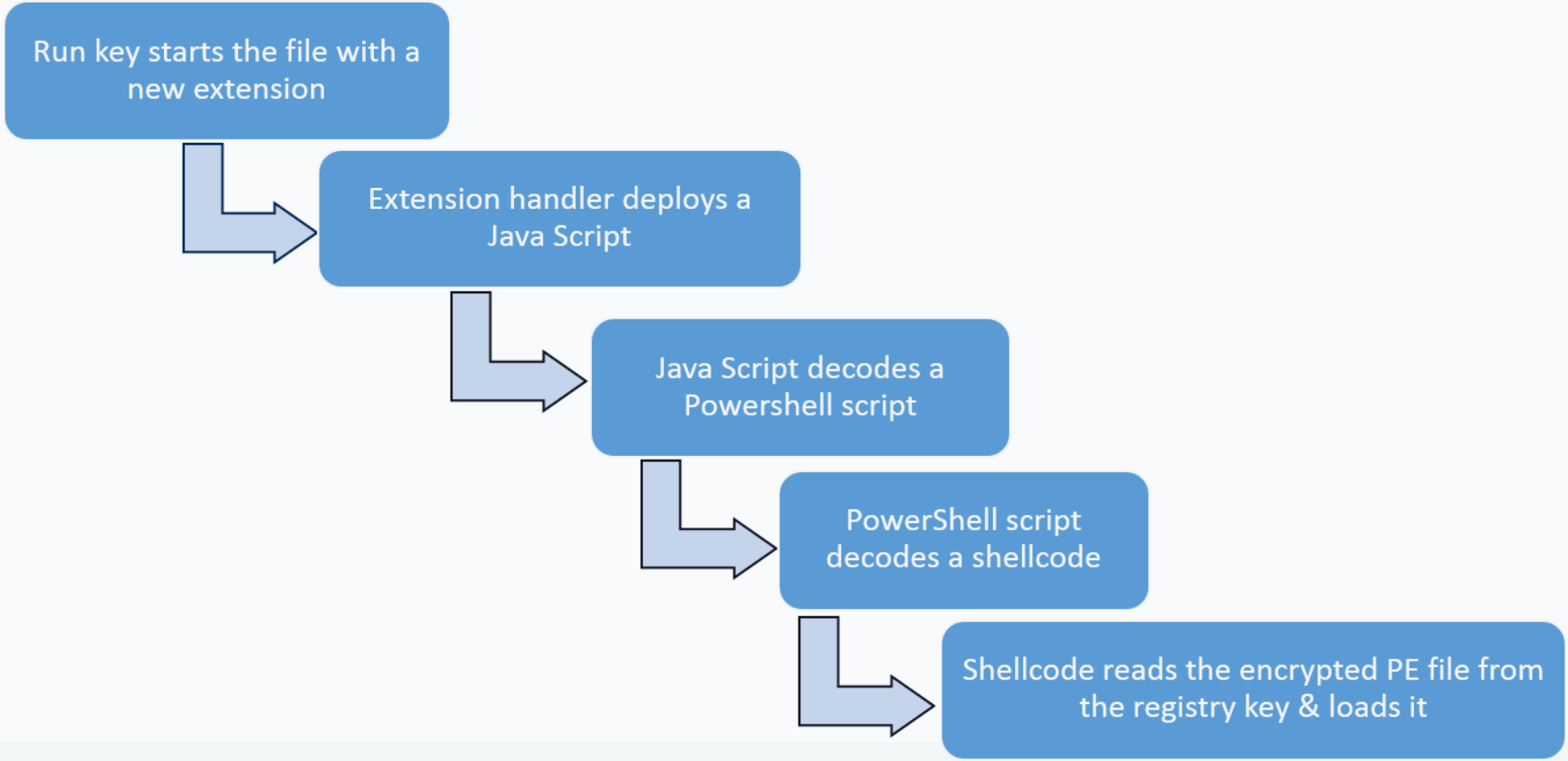
  - The malicious **PE is stored in the registry** in encrypted form



  - **Multiple layers** till the real payload is loaded...

https://blog.malwarebytes.com/threat-analysis/2016/07/untangling-kovter/

Hide code in the registry (multilayer: Kovter)



https://blog.malwarebytes.com/threat-analysis/2016/07/untangling-kovter/

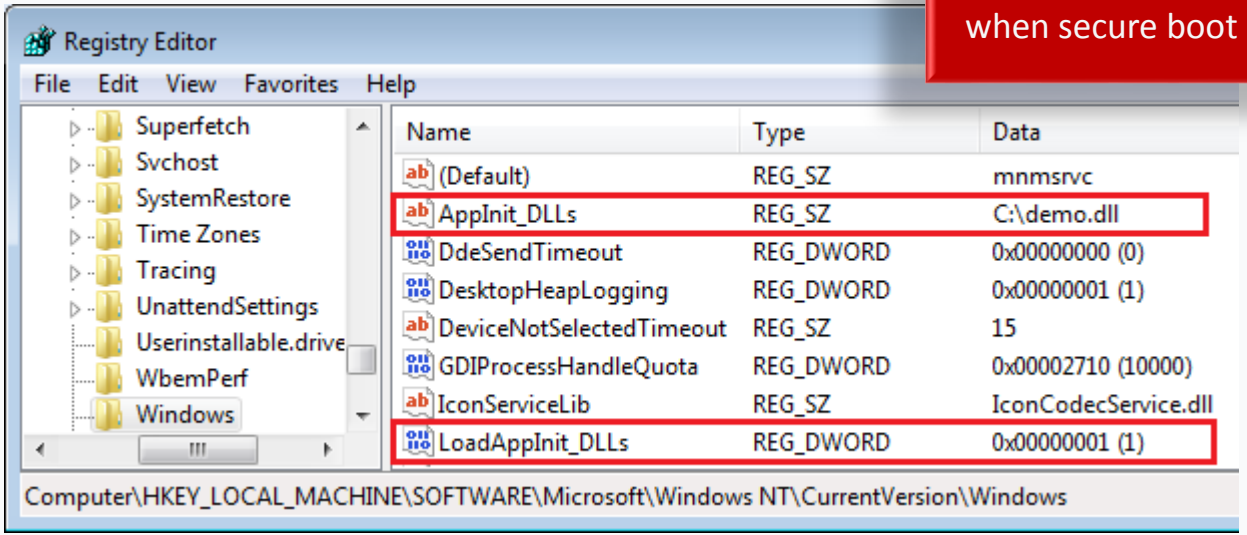
Malwarebytes

Abusing AppInit_DLLs

- Define DLLs that are injected to every application that uses user32.dll:

UAC Bypass required

Disabled in Win 8 and above, when secure boot is enabled



https://support.microsoft.com/pl-pl/help/197571/working-with-the-appinit-dlls-registry-value

Malwarebytes

Abusing AppInit_DLLs

- **Registry keys:**

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Windows\AppInit_DLLs
```

32 bit OS + 32 bit DLL
Or
64 bit OS + 64 bit DLL

```
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows
NT\CurrentVersion\Windows\AppInit_DLLs
```

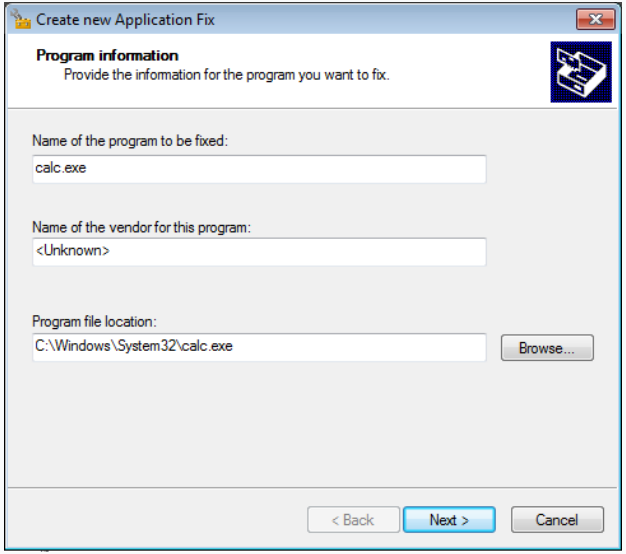
64 bit OS + 32 bit DLL

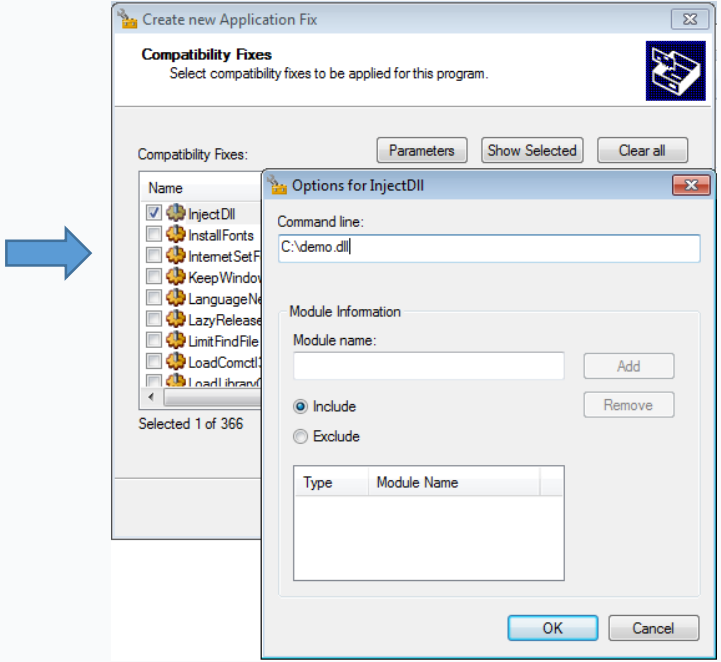
https://support.microsoft.com/pl-pl/help/197571/working-with-the-appinit-dlls-registry-value

Abusing shim databases

- Microsoft Application Compatibility Toolkit – creates patches:

UAC Bypass required





https://www.microsoft.com/en-us/download/confirmation.aspx?id=7352

Malwarebytes

Abusing shim databases

UAC Bypass required

- Shim Database
  - Allows setting automated injection of a patch into selected application
  - Can be used to automatically load malicious modules when the target application is deployed (DLL, shellcode, etc)

https://www.fireeye.com/blog/threat-research/2017/05/fin7-shim-databases-persistence.html

Abusing shim databases

UAC Bypass required

- sdbinst.exe – standard Windows tool, manages patches (.sdb)

```
sdbinst /q <path_to_shim_db>.sdb
```

- Example: Ramnit malware deploying sdbinst

```
└ ldzquze7.exe (PID: 2588) 46/61
  └ blyW2NlO (PID: 2664) 11/61
    └ hcqguwri.exe (PID: 2904) 11/61
      ├ svchost.exe %WINDIR%\system32\svchost.exe (PID: 2972)
      ├ svchost.exe %WINDIR%\system32\svchost.exe (PID: 2840)
      │ └ TRACERT.EXE (PID: 3132)
      └ sdbinst.exe /q /u "%TEMP%\\..\..\LocalLow\com.m1I5Ot0.sdb" (PID: 2780)
```

https://www.hybrid-analysis.com/sample/c823183b49148e7e60d84142ccefc8fe16fe44bec94d5eabdbd623c65cdaff8c?environmentId=100/

Abusing shim databases

UAC Bypass required

- To trigger less alerts, install a shim without *sdbinst.exe*
- Example of edited keys:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\InstalledSDB]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\AppCompatFlags\InstalledSDB\{7c6002f0-559a-488a-9fc1-bd54c33fdfa9}]
"DatabasePath"=<path_to_shim>.sdb
"DatabaseType"=dword:00010000

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\AppCompatFlags\Custom\<shimmed_app>.exe]
"{7c6002f0-559a-488a-9fc1-bd54c33fdfa9}.sdb"=hex(b):90,58,2d,0d,1a,b7,d2,01
```

https://github.com/hasherezade/persistence_demos/tree/master/shim_persist

COM hijacking

- COM – Component Object Model
- „enables interaction between software components through the operating system”
- Identified by CLSID – examples:

```
{3543619C-D563-43f7-95EA-4DA7E1CC396A} – Shell Icon Overlay Handler
{BCDE0395-E52F-467C-8E3D-C4579291692E} - MMDevice Manipulator
```

https://msdn.microsoft.com/en-us/library/accessibility(v=vs.110).aspx

https://attack.mitre.org/wiki/Technique/T1122

COM hijacking

- Substitute legitimate COM by your own
- When the application using the defined COM is loaded, malware is executed
- Keys:

32 bit OS + 32 bit DLL
Or
64 bit OS + 64 bit DLL

```
HKCU\Software\Classes\CLSID\[hijacked CLSID]\InprocServer32
```

64 bit OS +
32 bit DLL

```
HKCU\Software\Classes\Wow6432Node\CLSID\[hijacked CLSID]\InprocServer32
```

https://blog.gdatasoftware.com/2014/10/23941-com-object-hijacking-the-discreet-way-of-persistence

COM hijacking

```
[HKEY_CURRENT_USER\Software\Classes\CLSID\{BCDE0395-E52F-467C-8E3D-
C4579291692E}\InprocServer32]
@="C:\\ProgramData\\demo.dll"
"ThreadingModel"="Apartment"
```

```
[HKEY_USERS\S-1-5-21-1929933236-2258453022-3626796957-
1000_Classes\CLSID\{BCDE0395-E52F-467C-8E3D-C4579291692E}\InprocServer32]
@="C:\\ProgramData\\demo.dll"
"ThreadingModel"="Apartment"
```

https://github.com/hasherezade/persistence_demos/tree/master/com_hijack

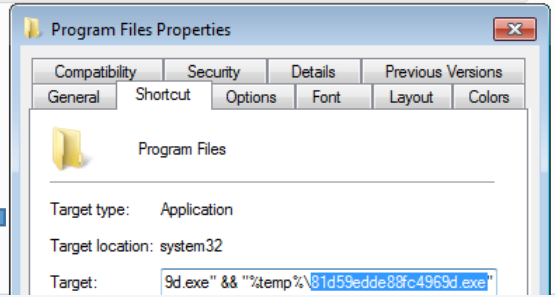
User-triggered persistence (Spora)

Hidden folders

Shortcuts made to replace them...

Clicking the shortcut deploys the command...

ProgramData	2016-05-31 23:39	File folder	
Python27	2017-02-22 01:38	File folder	
Recovery	2015-06-18 22:23	File folder	
System Volume Information	2017-03-08 17:05	File folder	
totalcmd	2016-05-26 14:18	File folder	
Users	2015-06-18 22:23	File folder	
Windows	2016-05-26 14:18	File folder	
81d59edde88fc4969d.exe	2017-03-06 23:05	Application	27 KB
autoexec.bat	2009-06-10 23:42	Windows Batch File	1 KB
baretail.exe	2015-06-05 18:20	Application	220 KB
config.sys	2009-06-10 23:42	System file	1 KB
lock_me_bmp	2017-03-08 17:30	Shortcut	1 KB
pagefile.sys	2017-02-22 01:56	System file	1 048 576 KB
PerfLogs	2017-03-08 17:30	Shortcut	1 KB
pin	2017-03-08 17:30	Shortcut	1 KB
Pin_Tools	2017-03-08 17:30	Shortcut	1 KB
PODF5-C2RTZ-TZTET-OETEY.html	2017-03-08 16:21	Firefox HTML Doc...	17 KB
Program Files	2017-03-08 17:30	Shortcut	1 KB
Python27			
totalcmd			
Users			
Windows			




```
C:\Windows\system32\cmd.exe /c
start explorer.exe "Program Files"
& type "81d59edde88fc4969d.exe" >
"%temp%\81d59edde88fc4969d.exe"
&& "%temp%\81d59edde88fc4969d.exe"
```

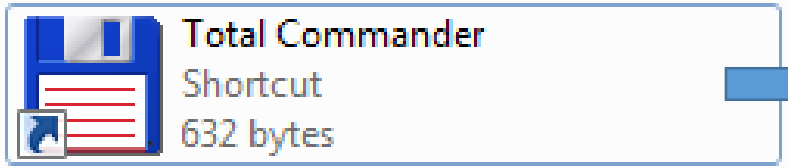
Malwarebytes

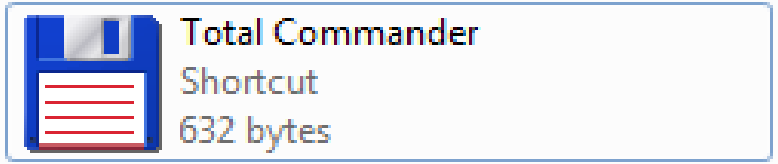
User-triggered persistence (Spora)

- Spora ransomware:

```
HKEY_LOCAL_MACHINE\Software\Classes\lnkfile\IsShortcut
```

```
phkResult = this;
if ( !RegOpenKeyExW(HKEY_LOCAL_MACHINE, L"SOFTWARE\\Classes\\lnkfile", 0, 2u, &phkResult) )
{
  RegDeleteValueW(phkResult, L"IsShortcut");
  RegCloseKey(phkResult);
  SHChangeNotify(0x8000000, 0, 0, 0);
}
```





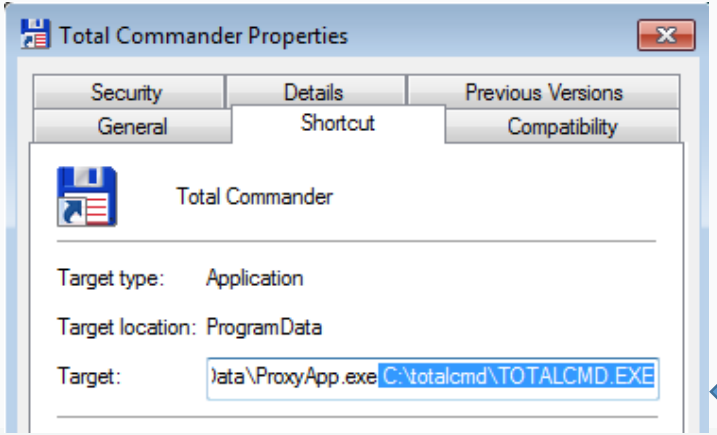
User-triggered persistence (Spora)

- Spora ransomware:
  - Disable showing link indicators:
    - Delete:
      `HKEY_LOCAL_MACHINE\Software\Classes\lnkfile\`**`IsShortcut`**
  - Hide folders and substitute them by links
  - Clicking the link causes opening the original program + deploying the dropped malware

https://blog.malwarebytes.com/threat-analysis/2017/03/spora-ransomware/

User-triggered persistence (shortcut hijacking)

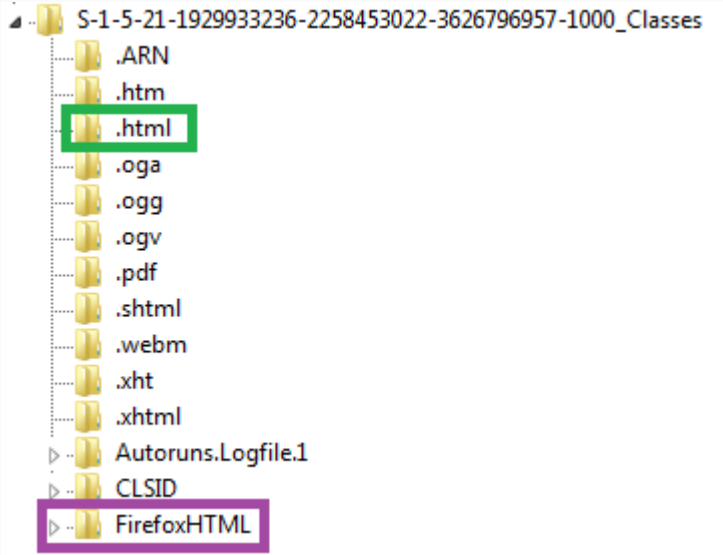
- Booby-trapped shortcuts: used by Fancy Bear APT (distribution)
- Similarly: existing shortcuts can be overwritten by shortcuts deploying malware

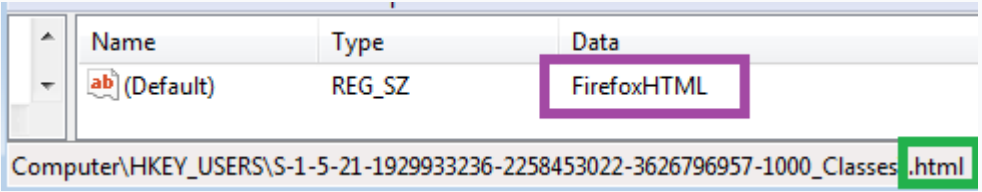



```
C:\ProgramData\ProxyApp.exe
C:\totalcmd\TOTALCMD.exe
```

https://www.uperesia.com/booby-trapped-shortcut-generator

User-triggered persistence (handler hijacking)





extension

handler

https://github.com/hasherezade/persistence_demos/tree/master/extension_hijack

Malwarebytes

User-triggered persistence - (handler hijacking)

Name	Type	Data
(Default)	REG_SZ	"C:\Program Files\Mozilla Firefox\firefox.exe" -osint -url "%1"

Computer\HKEY_USERS\S-1-5-21-1929933236-2258453022-3626796957-1000_Classes\FirefoxHTML\shell\open\command

Hijack the handler

Name	Type	Data
(Default)	REG_SZ	C:\ProgramData\ProxyApp.exe C:\Program Files\Mozilla Firefox\firefox.exe" -osint -url "%1"

Computer\HKEY_USERS\S-1-5-21-1929933236-2258453022-3626796957-1000_Classes\FirefoxHTML\shell\open\command

handler

Genuine app

Malicious app

https://github.com/hasherezade/persistence_demos/tree/master/extension_hijack

Malwarebytes

User-triggered persistence (handler hijacking)

- Applications handling particular extensions are defined in the registry
- Globaly defined extensions and handlers: in *HKEY_CLASSES_ROOT*
- It can be also defined per user: *HKEY_USERS -> <user SID>_Classes*
- Redefine a handler: no Administrator rights required

https://github.com/hasherezade/persistence_demos/tree/master/extension_hijack

User-triggered persistence (handler hijacking)

- When the user click a file with hijacked extension, the malware is deployed
- DEMO:
  - https://goo.gl/RGPiuY

https://github.com/hasherezade/persistence_demos/tree/master/extension_hijack

Malwarebytes

Conclusions

- Authors of the malware are very creative in finding new ways of hiding persistence
- The easiest way to detect the persistence method is by observing the installation – post-infection analysis is much harder
- „Fileless” malware also creates artifacts that can be found in a typical way

Additional material

- [1] https://www.fireeye.com/blog/threat-research/2017/03/dissecting_one_ofap.htm
- [2] https://cybellum.com/doubleagentzero-day-code-injection-and-persistence-technique/
- [3] https://securelist.com/blog/research/77403/fileless-attacks-against-enterprise-networks/
- [4] http://herrcore.blogspot.com.tr/2015/06/malware-persistence-with.html
- [5] https://blog.gdatasoftware.com/2014/10/23941-com-object-hijacking-the-discreet-way-of-persistence - COM Object Hijacking
- [6] https://www.youtube.com/watch?v=wQEnUISOZPI – „Shims for the Win”
- [7] http://0xthem.blogspot.com/2014/03/t-emporal-persistence-with-and-schtasks.html - BITS backdoor
- [8] http://www.hexacorn.com/blog/2017/03/18/beyond-good-ol-run-key-part-60/ - persistence via Windows update
- [9] https://isc.sans.edu/forums/diary/Wipe+the+drive+Stealthy+Malware+Persistence+Part+3/15448/ - SANS on stealthy malware persistence methods

Questions? Remarks?

Thank You!