Module 2

Typical goals of malware and their implementations

Persistence

Basics of Persistence

- WHO?
  - Most of the malware needs it (except some ransomware)
- WHY?
  - To start the application after each reboot
- HOW?
  - Using legitimate persistence methods
 - Using custom, creative methods....

Basics of Persistence

Windows offers various legitimate persistence ways – let's recall them...

Basics of Persistence

- 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)…

HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
☑ 0d3qdwbX   c:\users\tester\appdata\local\microsoft\windows\nvkmgg.exe

https://support.microsoft.com/pl-pl/help/179365/info-run,-runonce,-runservices,-runservicesonce-and-startup
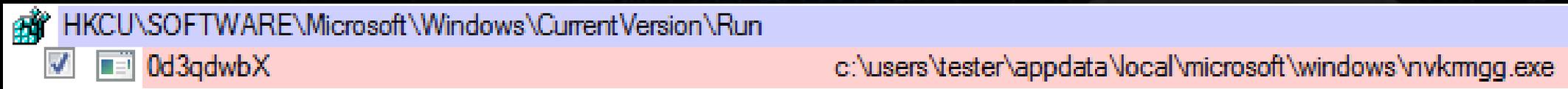
Basics of Persistence: Startup link

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

AppData ▸ Roaming ▸ Microsoft ▸ Windows ▸ Start Menu ▸ Programs ▸ Startup

Library ▾   Share with ▾   New folder

Name	Date modified	Type	Size
desktop.ini	2015-06-18 22:24	Configuration sett...	1 KB
test.exe	2017-04-02 23:32	Shortcut	1 KB
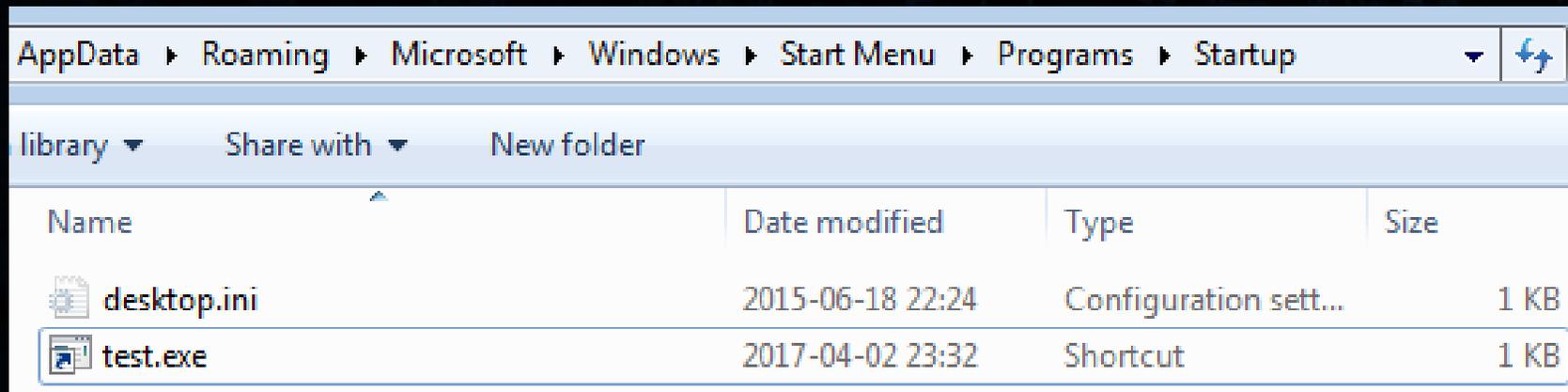
Basics of Persistence: Scheduled task

- 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

Basics of Persistence: System Services
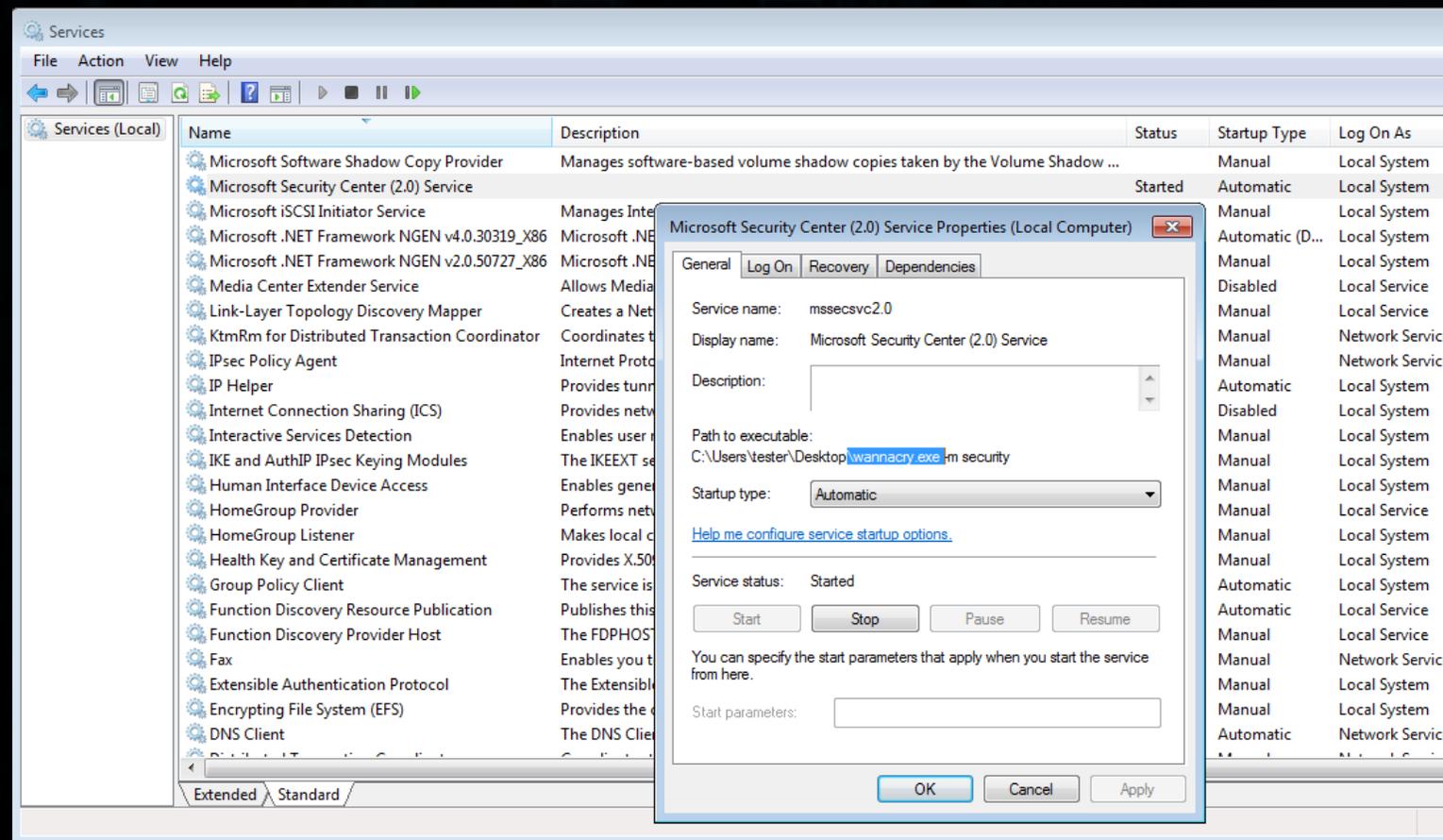
UAC Bypass required

Basics of Persistence: System Services

- Administrator rights required

- Creating a service:

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

UAC Bypass required

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>

UAC Bypass required

Basics of Persistence: System Services

- Regedit view:
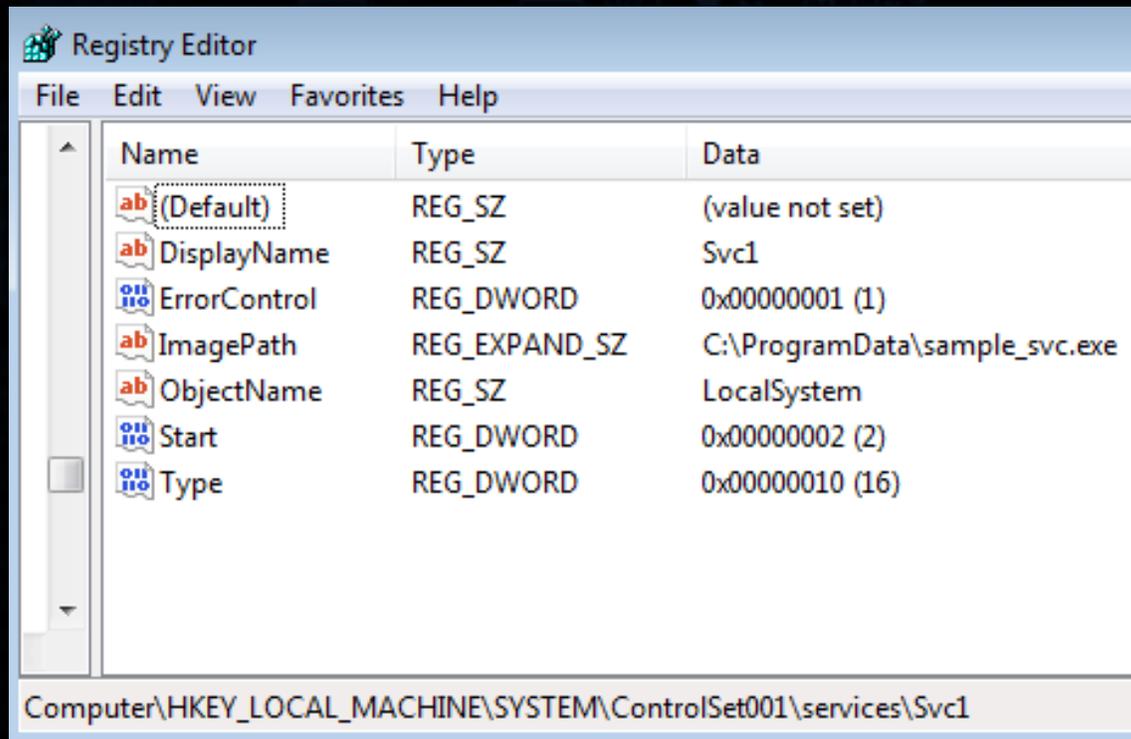
UAC Bypass required

# Hunting for malware persistence artifacts

SysIntenals' Autoruns
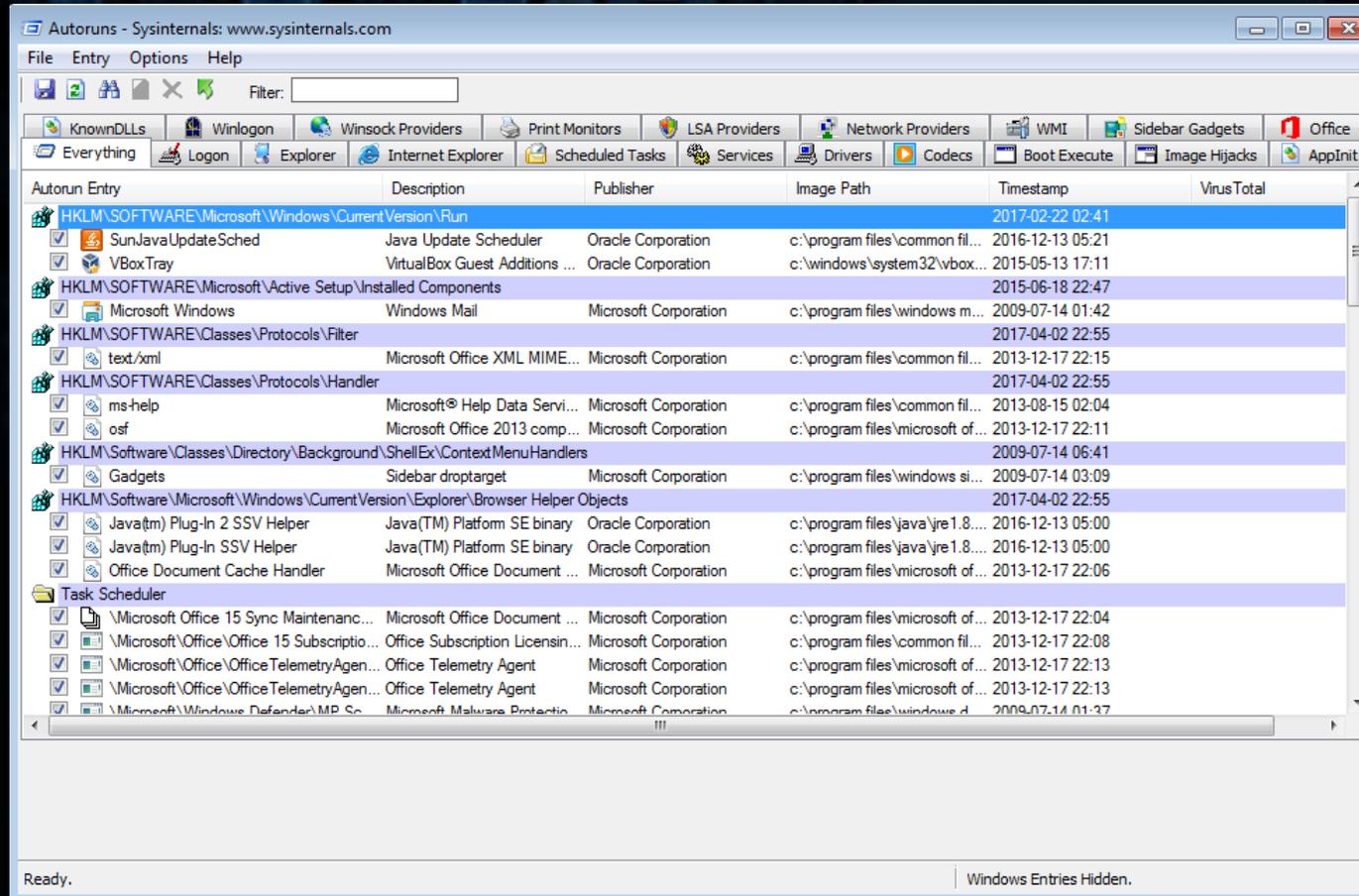
RegShot

- RegShot allows for monitoring changes in the Windows Registry

Hiding Persistence

Hiding Persistence - ideas

- Typical methods, but with extra measures to cover/protect

- Abuse of other mechanisms of the system for automated injection, i.e.:
  - AppInit_DLL, COM Hijacking, Shims, MS Application Verifier Provider ("DoubleAgent" technique), etc

- 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.

Make file 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**

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

With a special directory name

# Make file 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
```

CON

Rename

❌ The specified device name is invalid.

OK
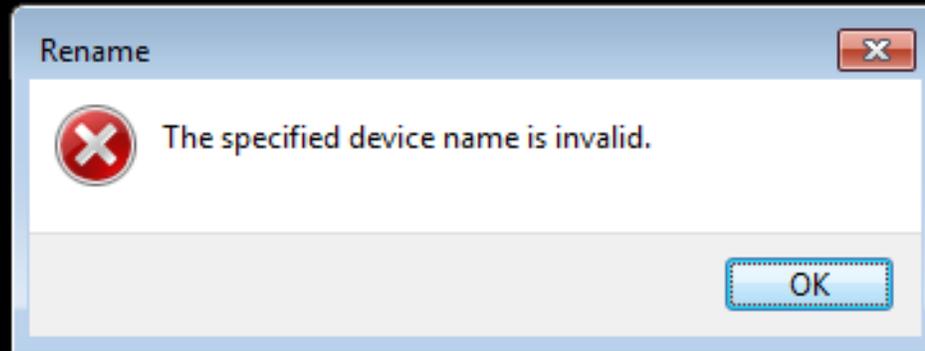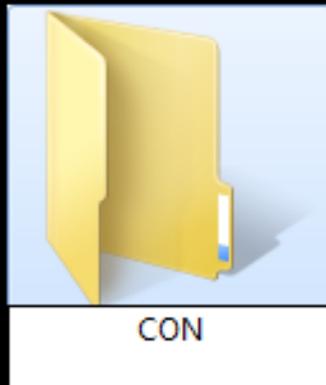
Make file inaccessible - special folders

- Special CLSIDs:

```
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}
```

MySpecialDirectory.{ED7BA470-8E54-465E-825C-99712043E01C}

tester ▶ MySpecialDirectory

Name

▲ Action Center (15)
- Change Customer Experience Improvement Program settings
- Change User Account Control settings
- Check security status
- Choose how to check for solutions
- Choose how to report problems
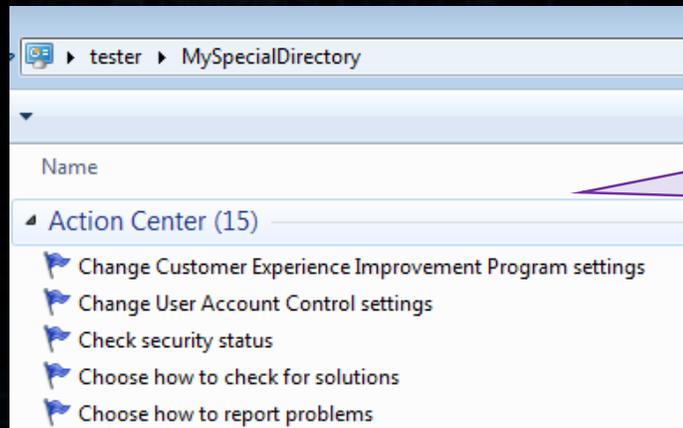
Clicking on folder triggers different action
-> no access to the content

Make file 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

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

Restricted name + special CLSID

Make file invisible - 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>
```

One file can have many alternative datasteams

Make file invisible - ADS

File.txt

Size: 37 bytes

FILE.TXT

:$DATA

File.txt - Notepad

File   Edit   Format   View   Help

This is the main stream of this file!

:HIDDEN_STREAM

File.txt:hidden_stream

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000000	4D	5A	90	00	03	00	00	00	04	00	00	00	FF	FF	00	00	MZ.........`..
00000010	B8	00	00	00	00	00	00	00	40	00	00	00	00	00	00	00	¸.......@......
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000030	00	00	00	00	00	00	00	00	00	00	00	00	F8	00	00	00	............ř...
00000040	0E	1F	BA	0E	00	B4	09	CD	21	B8	01	4C	CD	21	54	68	..ş..´.Í!¸.LÍ!Th
00000050	69	73	20	70	72	6F	67	72	61	6D	20	63	61	6E	6E	6F	is program canno
00000060	74	20	62	65	20	72	75	6E	20	69	6E	20	44	4F	53	20	t be run in DOS

Make file invisible - ADS

- Get a demo.dll: https://goo.gl/wl7ZNJ

- Copy the DLL into ADS of some file, i.e.:

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

- Deploy the DLL from the alternate stream (DllMain):

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

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

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

Make file invisible - ADS

- Result:

Make registry keys inaccessible

- NULL character at the beginning of the key

- Example: Kovter

`\0c:\\users\\tester\\appdata\\local\\bcd7\\62d2.lnk`
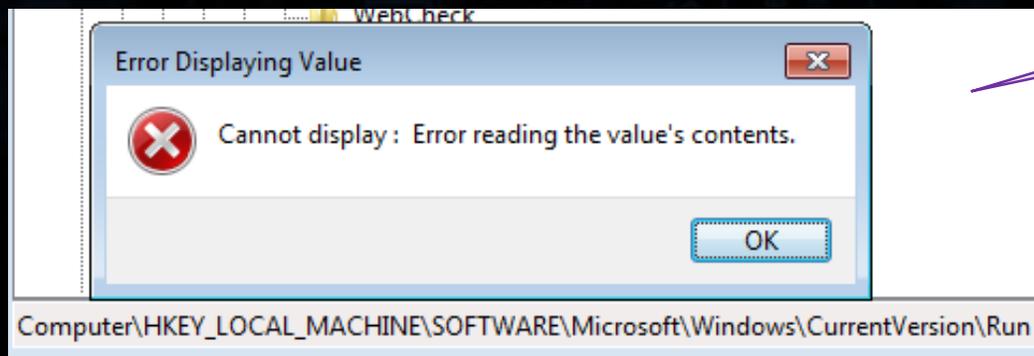
Malformed key:
Regedit cannot display it

Still can be viewed by Autoruns....

Error Displaying Value

Cannot display : Error reading the value's contents.

OK

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

HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

(Default)   c:\users\tester\appdata\local\bcd7\62d2.lnk

https://hshrzd.wordpress.com/2016/03/19/introduction-to-ads-alternate-data-streams/
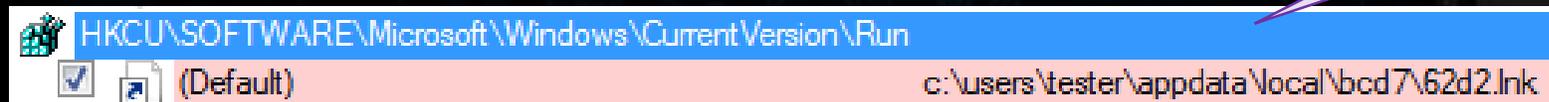
Make registry keys harder to spot

- By default, Autoruns hides keys leading to Microsoft apps
- Example: Moker trojan
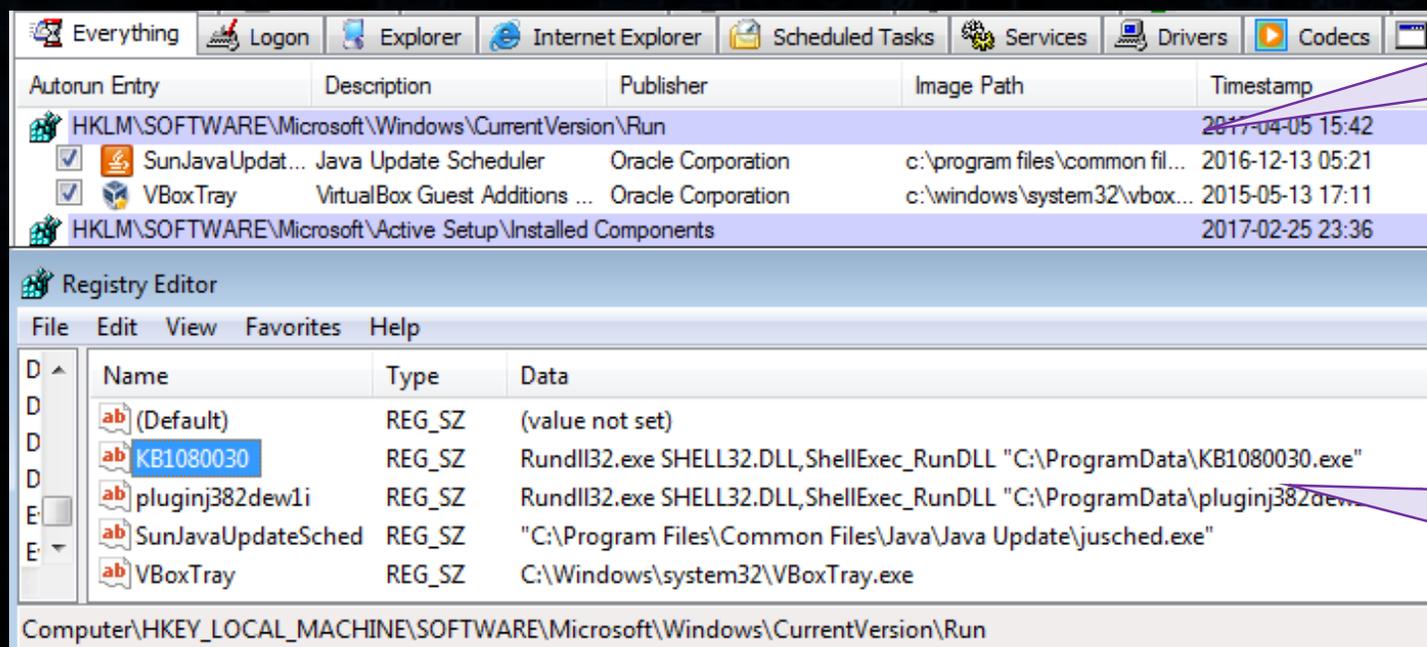
By default, Autoruns shows only two keys...

...but there are more

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

Make registry keys harder to spot

- Example: Moker trojan

The malware is deployed by a Microsoft application: Rundll32

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

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

Hide behind legitimate applications (DLL abuse)

- Korplug (PlugX) - spyware
  - Uses vulnerable, digitally 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!
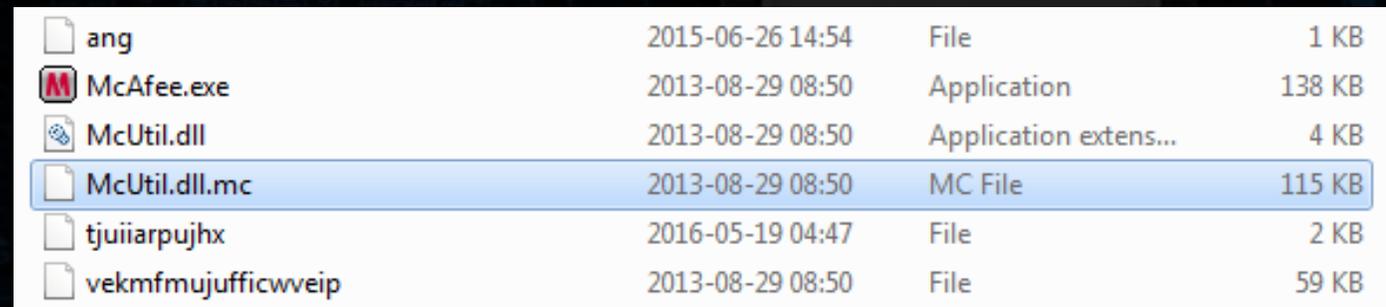
ang	2015-06-26 14:54	File	1 KB
McAfee.exe	2013-08-29 08:50	Application	138 KB
McUtil.dll	2013-08-29 08:50	Application extens...	4 KB
McUtil.dll.mc	2013-08-29 08:50	MC File	115 KB
tjuiiarpujhx	2016-05-19 04:47	File	2 KB
vekmfmujufficwveip	2013-08-29 08:50	File	59 KB

Hide behind legitimate applications (script)

- Terdot Zbot (Zeus-based banking trojan):
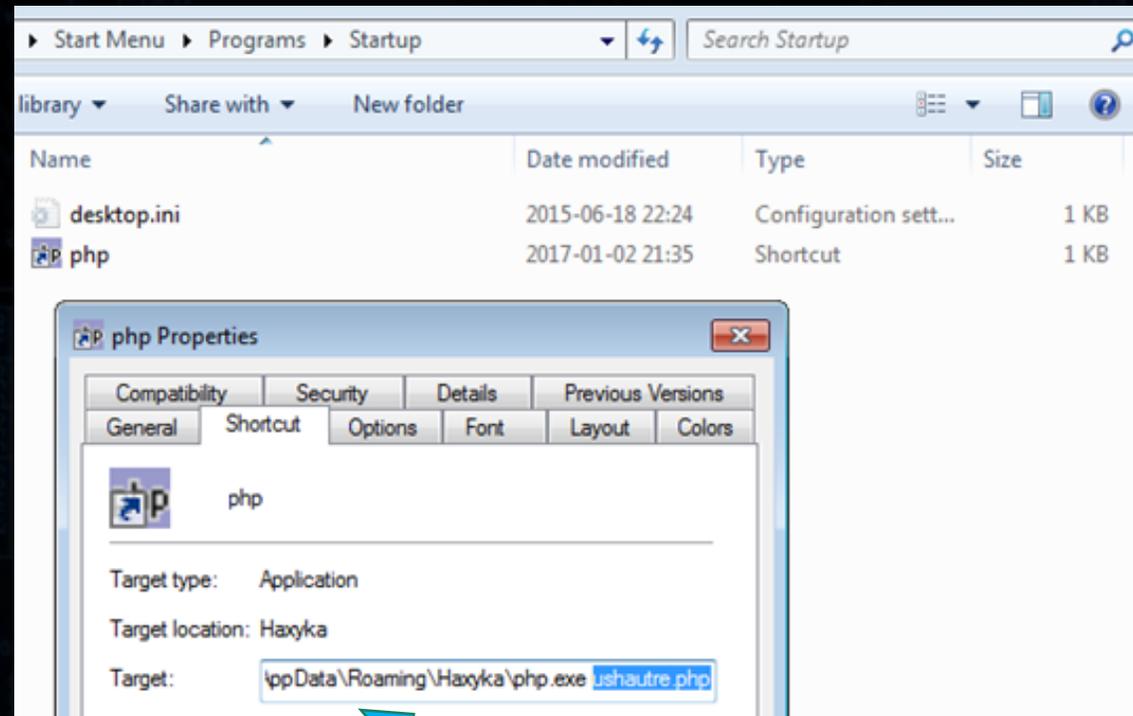
`C:\AppData\Roaming\Haxyka\php.exe ushautre.php`

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!

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:
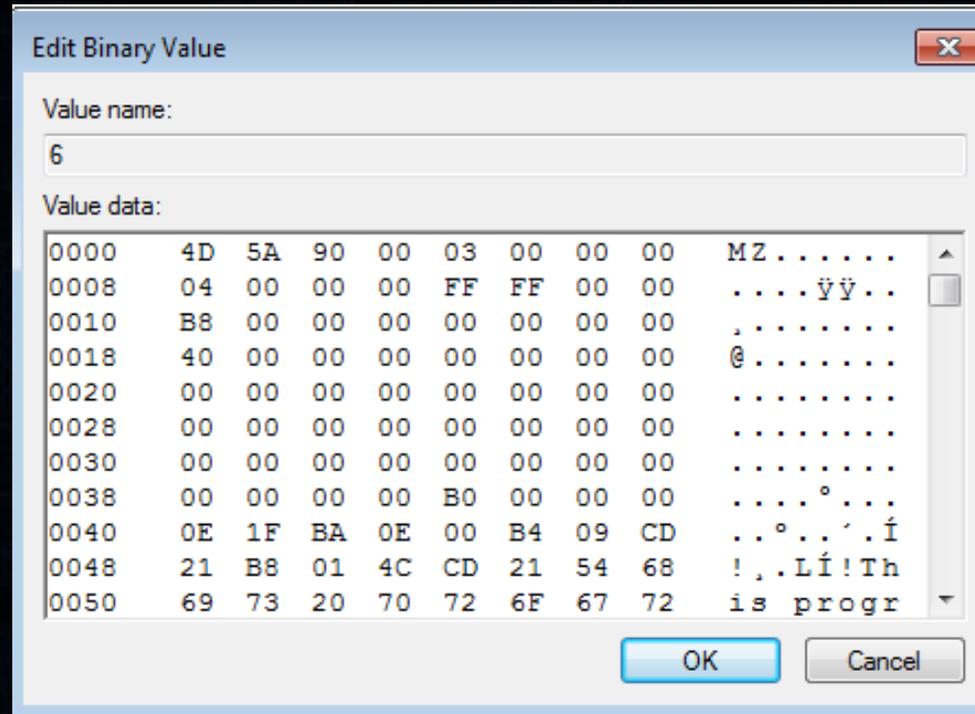
Hide code in the registry (multilayer: Kovter)

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

	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run	
☑ 🔍	(Default)	c:\users\tester\appdata\local\7ff3\856a.bat
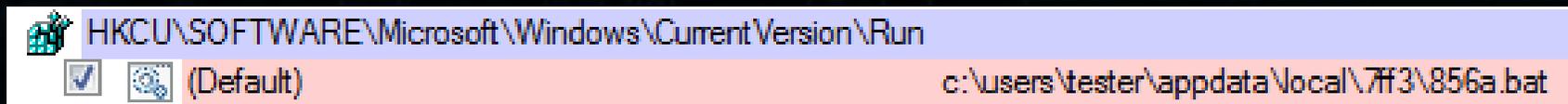
  - The malicious PE is stored in the registry in encrypted form

vfkhxfak	REG_SZ	EŞ™"÷ İk'dR4á–ëVĂÍóŁu⚹…›/eŠp~‹ÐL€Ó OĎz…

  - Multiple layers till the real payload is loaded…

Hide code in the registry (multilayer: Kovter)

Run key starts the file with a new extension → Extension handler deploys a Java Script → Java Script decodes a Powershell script → PowerShell script decodes a shellcode → Shellcode reads the encrypted PE file from the registry key & loads it
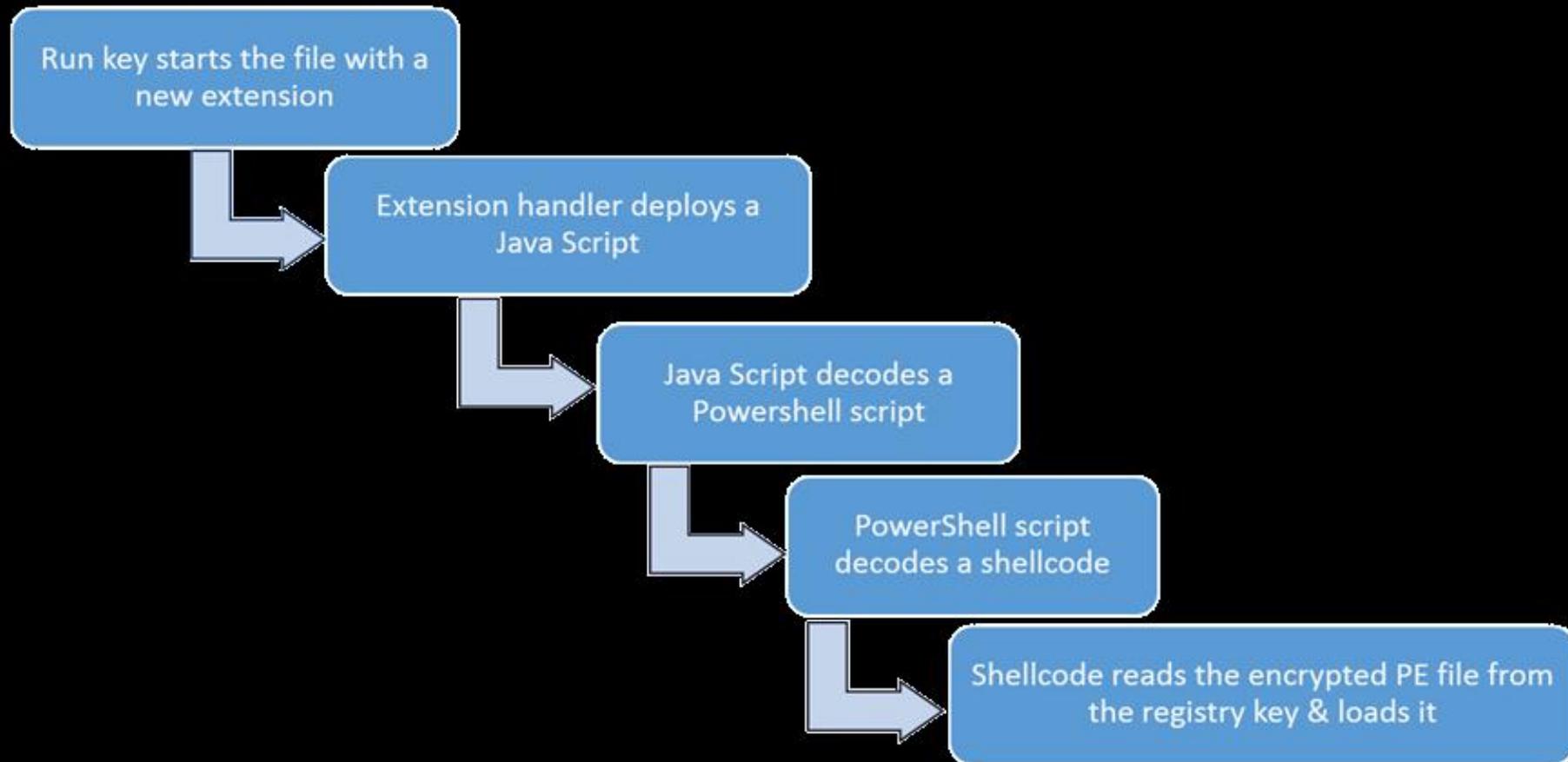
Abusing AppInit_DLLs

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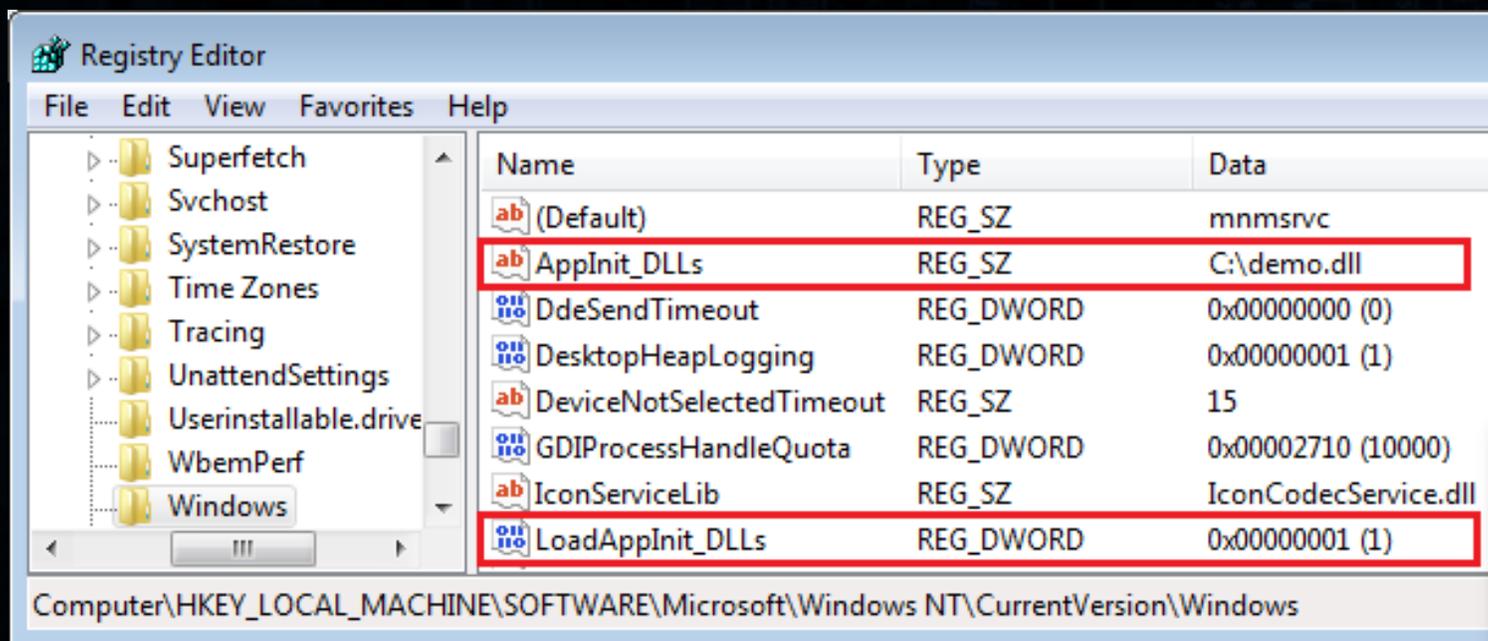
UAC
Bypass
required

Disabled in Win 8
and above, when
secure boot is

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

Abusing shim databases

- Microsoft Application Compatibility Toolkit – creates patches:
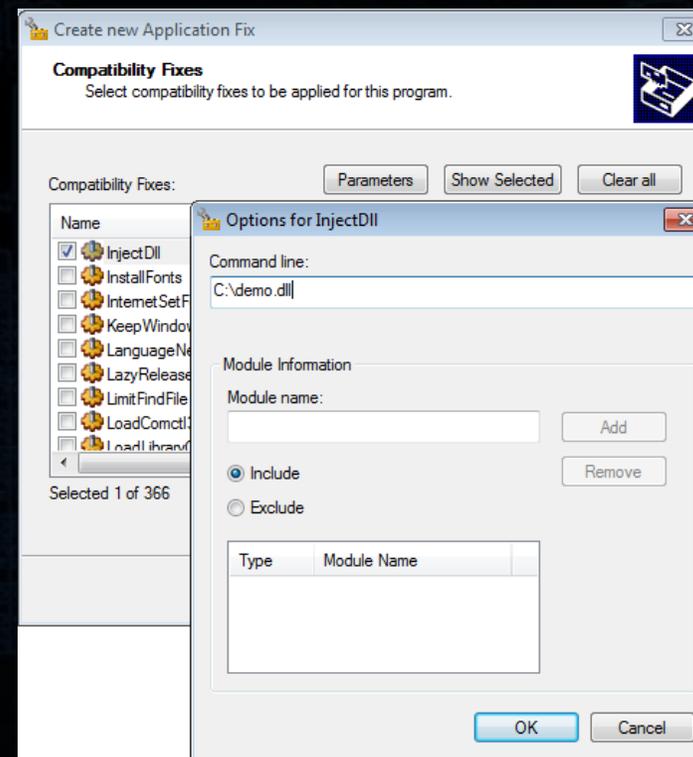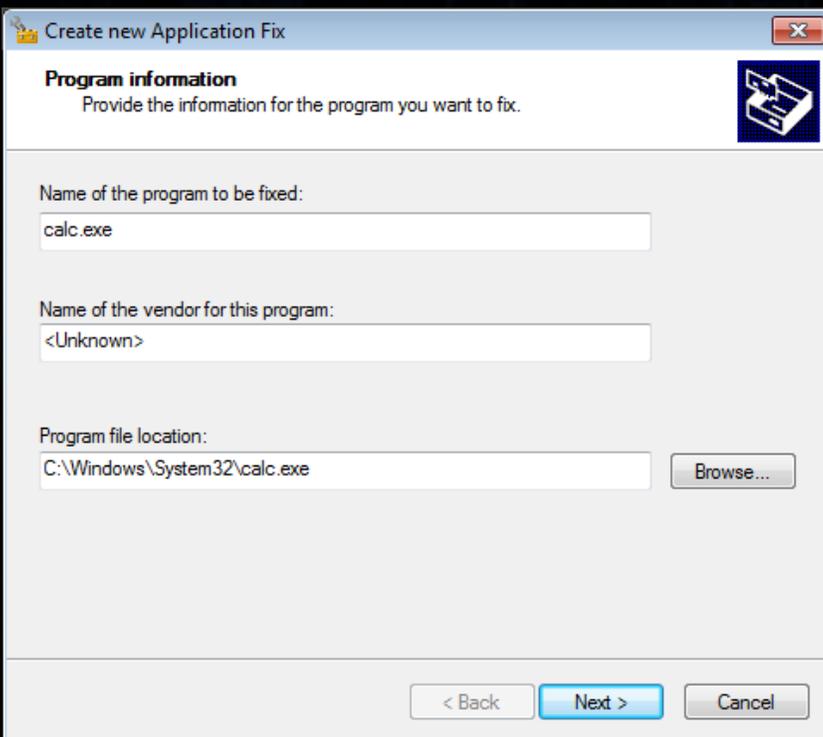
Abusing shim databases

- 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)
  - Installation requires elevated privileges

UAC
Bypass
required

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

Abusing shim databases

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

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

- Example: Ramnit malware deploying sdbinst

Abusing shim databases

- 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
```

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
```

More: 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
```

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

> 64 bit OS + 32 bit DLL

COM Hijacking

- Examples:

```
[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
```

User-triggered persistence: link hijacking

- Example: 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);
}
```

Total Commander
Shortcut
632 bytes

Total Commander
Shortcut
632 bytes

User-triggered persistence: link hijacking

- Hijacking in the style of Spora ransomware:
  1. Disable showing link indicators:
     - Delete: HKEY_LOCAL_MACHINE\Software\Classes\lnkfile\IsShortcut
  2. Hide folders and substitute them by links
 3. Clicking the link causes opening the original program + deploying the dropped malware

User-triggered persistence: link hijacking

- 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
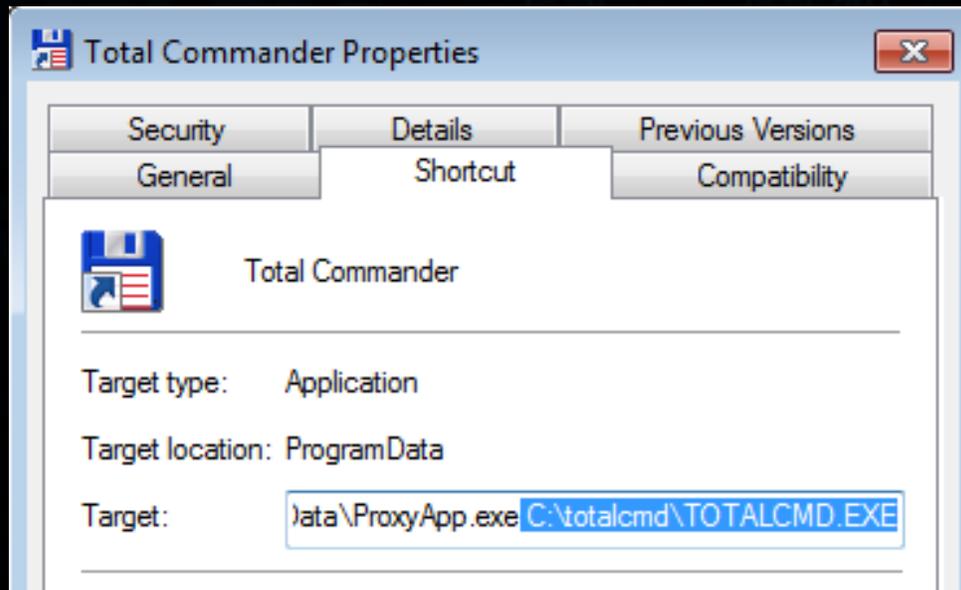
User-triggered persistence (handler hijacking)
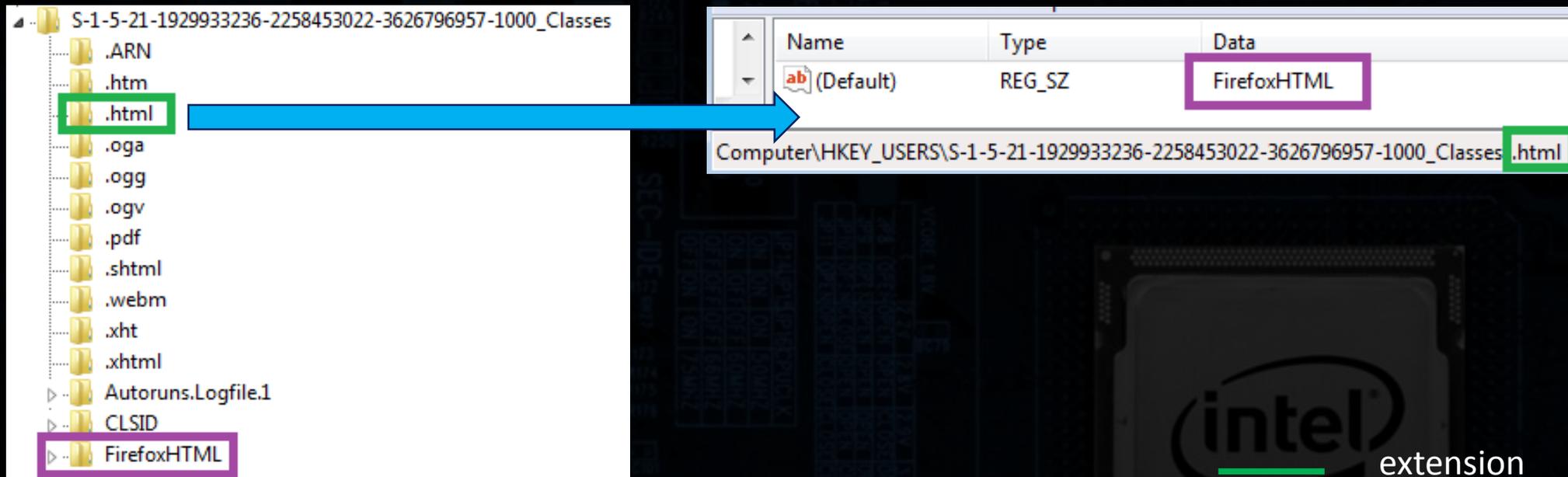
extension

handler

User-triggered persistence (handler hijacking)

Hijack the handler

handler
genuine app
malicious app

User-triggered persistence (handler hijacking)

- Applications handling particular extensions are defined in the registry

- Globally 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

User-triggered persistence (handler hijacking)

- When the user click a file with hijacked extension, the malware is deployed
- DEMO:
    - https://www.youtube.com/watch?v=IE9HOqZbil8

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