esentire®

Preventing File-Based Botnet Persistence and Growth

Date	December 2, 2016
Presented to	BotConf
Presenter	Kurtis Armour Information Security Consultant

Who am I?

- karmour@:/home$ whoami

- Information Security Consultant
  - eSentire Inc.
  - 5 years working in computer security
  - This talk is based off personal research

- Enjoy finding ways to help build more secure networks

Paying #Respect

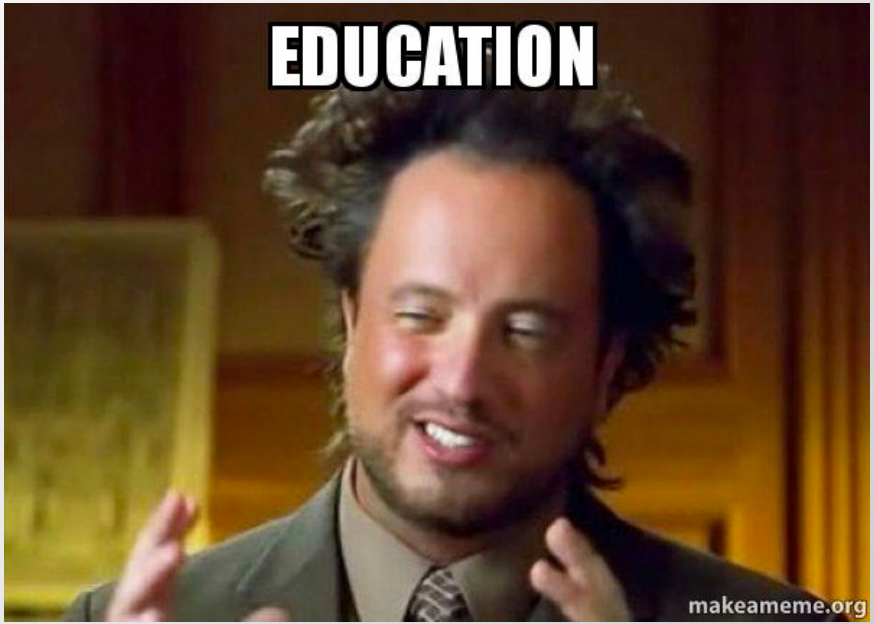
- Chris Lowson
  - @LowsonWebmin

- Jacob Gajek
  - @jgajek

- Matt Graeber
  - @mattifestation

Introduction

- The goal of this talk
- Education of threat landscape
- Layers of protection
- What is not covered?



Botnet Delivery Mechanisms

- Social Engineering
- Tricking users
  - Phishing Emails
  - Execution of fake files
- End goal monetization
  - For Bot Herders -> More Bots

Botnet Delivery Mechanisms

- Exploitation (Malvertising / Exploit Kits)
- Browsers / Third Party Applications
- EKs can drop any malware variant
- File-Based and Memory-Based

Code execution you say?

MALWARE LOTS OF MALWARE

Code Execution Methods

- What is the end goal of threat actors?
- What are the main ways to execute code?
- Binary Executables
- Scripts
- Shellcode

Droppers, Droppers Everywhere!

- HTML / HTA
- JS
- ZIP / 7ZIP / RAR
- EXE / DLL / MSI
- Macros (DOCM, XLSM, POTX)
- PS
- VBA / VBS / VBE
- PDF

Execution Trees!

Process Tree

- **WINWORD.EXE** 2156 *"C:\Users\Green\AppData\Local\Temp\usps_trk_849018930482.doc" /q*
  - **explorer.exe** 2696
    - **BN24AF.tmp** 1916
- **services.exe** 520
  - **spoolsv.exe** 1224
  - **SearchIndexer.exe** 1972 */Embedding*
  - **wmpnetwk.exe** 1828
  - **dllhost.exe** 1032 */Processid:{02D4B3F1-FD88-11D1-960D-00805FC79235}*
  - **msdtc.exe** 2140
  - **cb.exe** 1592
  - **OSPPSVC.EXE** 2260
  - **svchost.exe** 652 *-k DcomLaunch*
    - **WmiPrvSE.exe** 316 *-Embedding*
  - **svchost.exe** 3312 *-k netsvcs*
- **lsass.exe** 532
- **dwm.exe** 1920
- **explorer.exe** 1956

Execution Trees!

Process Tree

- WINWORD.EXE 2156 *"C:\Users\Green\AppData\Local\Temp\FedEx.doc" /q*
  - cmd.exe 4084 *cmd.exe /k ^powe^rshell -E^x^ecuti^onPo^licy by^pass -n^oprofi^le -wi^r*
    - powershell.exe 528 *powershell -ExecutionPolicy bypass -noprofile -windowstyle h*
    - Temp.exe 1792

Execution Trees!

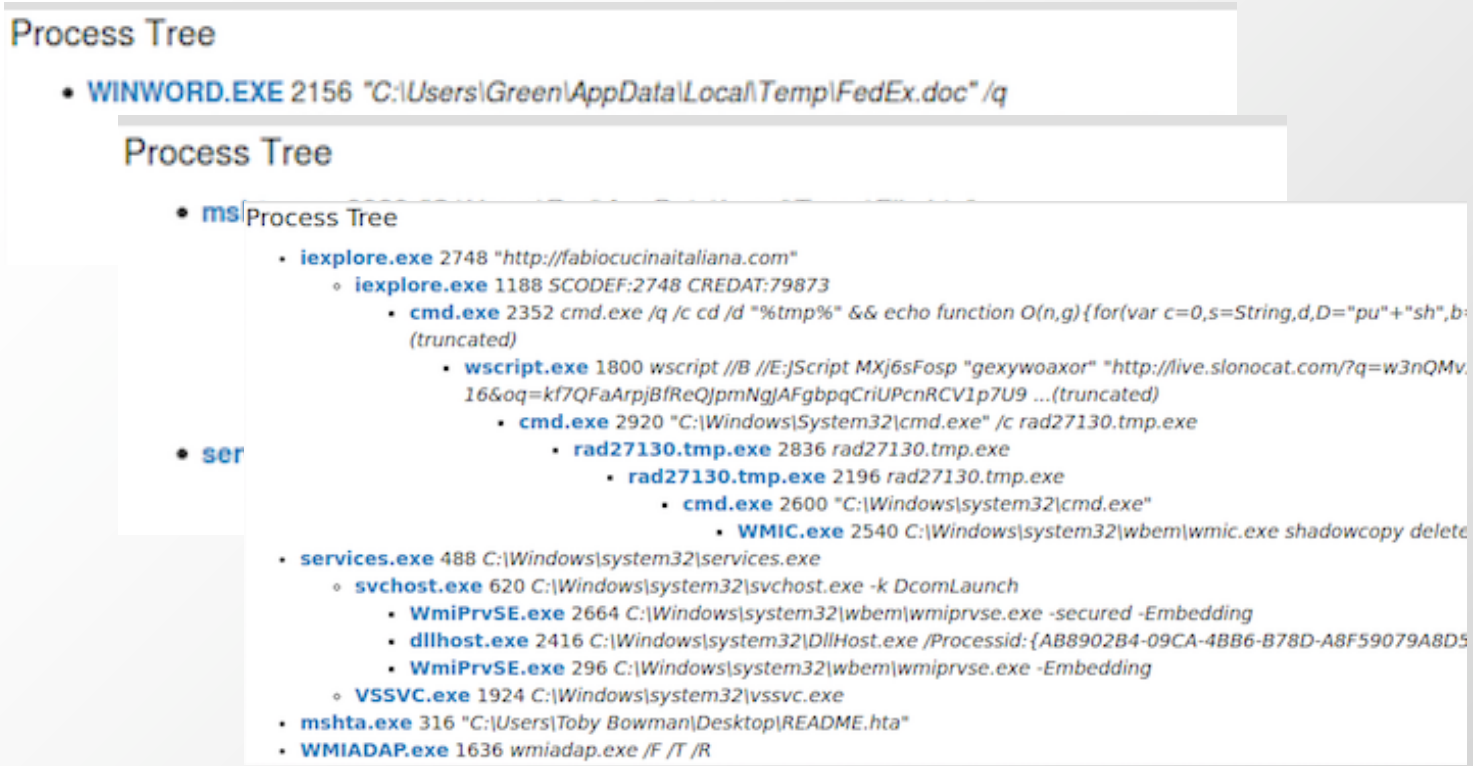
Process Tree

- WINWORD.EXE 2156 *"C:\Users\Green\AppData\Local\Temp\FedEx.doc" /q*

Process Tree

- mshta.exe 3928 *"C:\Users\Red\AppData\Local\Temp\File.hta"*
  - cmd.exe 4040 */c cd %temp% &@echo W7z = "http://www.sindbad.lk/6/01.exe">>*
    - wscript.exe 2564 *"C:\Users\Red\AppData\Local\Temp\K2d.vbs"*
    - timeout.exe 2536 *timeout 13*
    - JUC.EXE 2660 *JUC.EXE*
      - JUC.EXE 964
- services.exe 504
  - lsass.exe 624

Execution Trees!



User Protection and Host Hardening

PREVENTING MALICIOUS CODE EXECUTION

Overview of Defensive Layers

- The goals of adding layers
- Blocking execution of code can be done at different layers
- Restricting via GPO is best protection against changes

Limiting Software and Access

- Restricting access is key
  - No admin!
- Software control limits attack surface
- LAPS (Local Administrator Password Solution)

Script Control - WSH

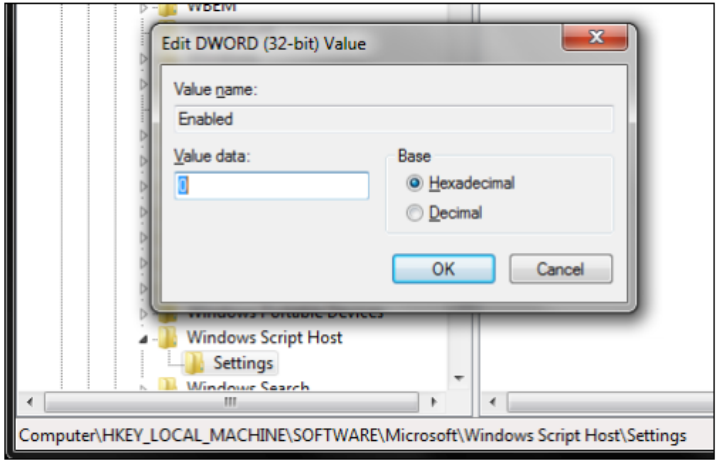
- Windows Script Host
- Lets not execute things by default ☺
- Disable built-in support (Test test test)
- Change the default program execution (if you have legacy systems)

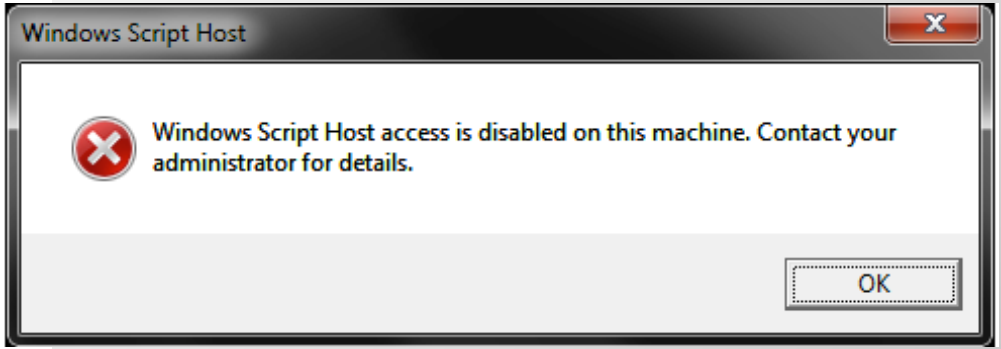
Script Control - WSH

» Disable built-in support

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows Script Host\Settings

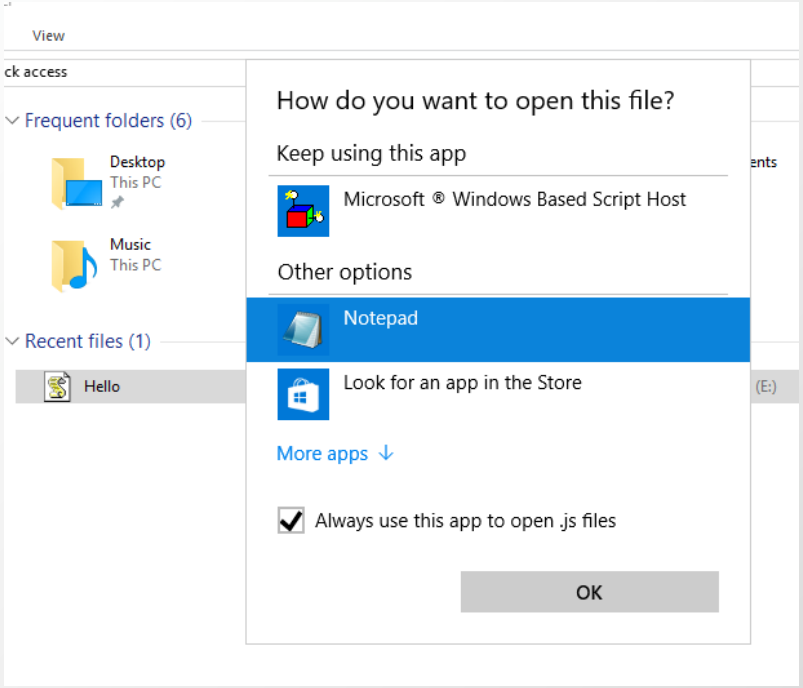
Create a new DWORD value named "Enabled" and set the value data to "0".





Script Control - WSH

» Changing default execution of program



Script Control – Microsoft Office Macros

- Microsoft Office files can contain embedded code written in VB
- Microsoft Office macros have their own VBS interpreter
- Stopping users from executing untrusted macros is key
- Configurable macro rules for Office (Excel, Word, Infopath, Outlook, Powerpoint, Project, Publisher, Visio)

Script Control – Microsoft Office Macros

Approach	Security	Business Impact	Implementation Difficulty	Recommended
Disable all macros and trusted locations	Very high	High	Low	Yes
Disable all macros but allow controlled trusted locations	High	Medium	Medium	Yes
Disable all macros, except digitally signed macros, and disable trusted locations	Medium	Medium	High	No
Let users decide which macros to enable on a case-by-case basis	Low	Low	Low	No
Enable all macros	None	None	Low	No

http://www.asd.gov.au/publications/protect/Microsoft_Office_Macro_Security.pdf

Script Control – Microsoft Office Macros

- Microsoft has added newer features to Office 2016
- Provides more granularity to apply policies through GPO
- Trust Center
  - Restrict the ability for users to allow macros
  - Restrict the ability for macros from internet to execute
- If using trust location be sure to limit who can execute from it

Powershell - Execution Control

- We do not want to allow normal users to execute PowerShell

- Why is PowerShell so dangerous?
  - Run code in memory without touching disk
  - Download & execute code from another system
  - Interface with .Net & Windows APIs
  - Most organizations are not watching PowerShell activity

Powershell - Execution Control

- Blocking PowerShell functionality is not easy
- Local Security Policies is not the answer!
- Execution Policies are easily bypassed

Powershell - Execution Control

- Powershell v5
  - Provides improved logging
  - Includes improved security features

- Constrained Mode
  - Limits what can be executed
    - Direct .NET scripting
    - Invoking of Win32 APIs via the Add-Type cmdlet
    - Interaction with COM objects

User Protection and Host Hardening

APPLICATION WHITELISTING

AppLocker

- Free Windows Built-in Application Whitelisting Feature
  - Policy is maintainable through GPO admin
- Available on a handful of Windows OS'
  - WS2008, WS2012, WS2016
  - Enterprise and Ultimate Editions
- Application inventory
- Protection against unwanted software
- Software standardization

AppLocker

- Two main approaches to implementing AppLocker
  - Allow Mode (Block all and whitelist approved by hash/path/publisher)
  - Deny Mode (Allow all and blacklist disapproved by hash/path/publisher)
- Provides the ability to enable an audit feature
  - Allows you to investigate what would be blocked / allowed
- Filters on:
  - Hash
  - Path
  - Publisher

AppLocker – Executable Control

- Executable Control
  - Blocking executing in %OSDRIVE%\Users*
- Publisher rules
  - Whitelist Publishers so they can update applications
- Restricting access to writeable directories
  - Users can write and execute from System32 and Windows folders!?
- Automatic Generation of Executable Rules
  - Utilize this feature for creating publisher and hash rules for approved programs

AppLocker – Executable Control

- Writeable Directories

```
<FilePathCondition Path="%SYSTEM32%\catroot2\*" />
<FilePathCondition Path="%SYSTEM32%\com\dmp\*" />
<FilePathCondition Path="%SYSTEM32%\FxsTmp\*" />
<FilePathCondition Path="%SYSTEM32%\spool\drivers\color\*" />
<FilePathCondition Path="%SYSTEM32%\spool\PRINTERS\*" />
<FilePathCondition Path="%SYSTEM32%\spool\SERVERS\*" />
<FilePathCondition Path="%SYSTEM32%\Tasks\*" />
<FilePathCondition Path="%WINDIR%\Debug\*" />
<FilePathCondition Path="%WINDIR%\PCHEALTH\ERRORREP\*" />
<FilePathCondition Path="%WINDIR%\PLA\*" />
<FilePathCondition Path="%WINDIR%\Registration\*" />
<FilePathCondition Path="%WINDIR%\SysWOW64\com\dmp\*" />
<FilePathCondition Path="%WINDIR%\SysWOW64\FxsTmp\*" />
<FilePathCondition Path="%WINDIR%\SysWOW64\Tasks\*" />
<FilePathCondition Path="%WINDIR%\Tasks\*" />
<FilePathCondition Path="%WINDIR%\Temp\*" />
<FilePathCondition Path="%WINDIR%\tracing\*" />
```

AppLocker - Some neat tricks

- Blocking Macros the DLL way
  - %OSDRIVE%\Program Files\Common Files\Microsoft Shared\VBA*
- .hta files are nasty
  - Utilizes another interpreter to execute script code (MSHTA.exe)
- Blocking PowerShell via Applocker has some wins
  - Stops auto-execution of droppers that call PowerShell.exe
  - Can completely block PowerShell interpreter if you want too

Bypasses and workarounds there are a few…….

- Applocker Local Security
  - Applocker local rules overwrite GPO rules
  - Admin has ability to turn off AppIDsvc
- Signed Binaries doing what they aren't suppose to do
  - e.g. MSBuild.exe, cdb.exe, dnx.exe, rcsi.exe, etc
  - Device Guard helps protect against this abuse
- Powershell
  - Calling older version of PowerShell (bypasses security related to one version)
    - Uninstall older legacy versions
  - Applocker PowerShell "Allow Mode" increases protection (interactive input and user-authored scripts with PowerShell v5)

Device Guard

- Essentially does not allow untrusted code to be executed
  - Everything is untrusted by default unless specifically approved

- Two primary components
  - Code Integrity (CI)
    - Kernel Mode Code Integrity
    - User Mode Code Integrity
  - Virtualization-Based Security (VBS)

Conclusion

- Adding layers makes executing bad code harder
- There is no silver bullet to defense
- Not every company is the same

esentire®

QUESTIONS

THANK YOU

MORE SECURE!

+1 866.579.2200 sales@esentire.com www.esentire.com Follow us @esentire