Detecting the Elusive Active Directory Threat Hunting

Sean Metcalf (@Pyrotek3)  
sean [@] TrimarcSecurity.com  
www.ADSecurity.org  
TrimarcSecurity.com
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 Setup
• Tracking Command-line/PowerShell Activity
• PS without PowerShell.exe & 06fu$ct10n
• Auditing Attacker Activity
• Kerberoasting Detection
<table>
<thead>
<tr>
<th>Keywords</th>
<th>Date and Time</th>
<th>Source</th>
<th>Event ID</th>
<th>Task Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audit Success</td>
<td>7/9/2016 7:30:53 AM</td>
<td>Security-Auditing</td>
<td>4616</td>
<td>Security State Change</td>
</tr>
<tr>
<td>Audit Success</td>
<td>7/9/2016 7:30:53 AM</td>
<td>Eventlog</td>
<td>1100</td>
<td>Service shutdown</td>
</tr>
<tr>
<td>Audit Success</td>
<td>6/29/2016 8:01:53 PM</td>
<td>Eventlog</td>
<td>1100</td>
<td>Service shutdown</td>
</tr>
<tr>
<td>Audit Success</td>
<td>6/29/2016 8:01:53 PM</td>
<td>Eventlog</td>
<td>1100</td>
<td>Service shutdown</td>
</tr>
<tr>
<td>Audit Success</td>
<td>6/10/2016 8:24:15 PM</td>
<td>Eventlog</td>
<td>1100</td>
<td>Service shutdown</td>
</tr>
<tr>
<td>Audit Success</td>
<td>6/10/2016 8:23:21 PM</td>
<td>Eventlog</td>
<td>1100</td>
<td>Service shutdown</td>
</tr>
<tr>
<td>Audit Success</td>
<td>6/10/2016 8:23:21 PM</td>
<td>Eventlog</td>
<td>1100</td>
<td>Service shutdown</td>
</tr>
<tr>
<td>Audit Success</td>
<td>6/10/2016 8:18:40 PM</td>
<td>Security-Auditing</td>
<td>4616</td>
<td>Security State Change</td>
</tr>
<tr>
<td>Audit Success</td>
<td>6/10/2016 8:17:45 PM</td>
<td>Eventlog</td>
<td>1100</td>
<td>Service shutdown</td>
</tr>
<tr>
<td>Audit Success</td>
<td>6/10/2016 8:17:45 PM</td>
<td>Eventlog</td>
<td>1100</td>
<td>Service shutdown</td>
</tr>
<tr>
<td>Audit Success</td>
<td>6/10/2016 8:17:45 PM</td>
<td>Eventlog</td>
<td>1100</td>
<td>Service shutdown</td>
</tr>
<tr>
<td>Audit Success</td>
<td>5/30/2016 8:16:43 PM</td>
<td>Security-Auditing</td>
<td>4616</td>
<td>Security State Change</td>
</tr>
<tr>
<td>Audit Success</td>
<td>3/4/2016 5:40:03 PM</td>
<td>Eventlog</td>
<td>1100</td>
<td>Service shutdown</td>
</tr>
<tr>
<td>Audit Success</td>
<td>3/4/2016 5:40:03 PM</td>
<td>Eventlog</td>
<td>1100</td>
<td>Service shutdown</td>
</tr>
</tbody>
</table>
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 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*

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

Sean Metcalf [@Pyrotek3 | sean@trimarcsecurity.com]
<table>
<thead>
<tr>
<th>Network connection detected:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>UtcTime: 2017-04-19 21:12:15.334</td>
<td></td>
</tr>
<tr>
<td>ProcessGuid: {fe520315-d256-58f7-0000-00109e446e12}</td>
<td></td>
</tr>
<tr>
<td>ProcessId: 11712</td>
<td></td>
</tr>
<tr>
<td>Image: C:\Windows\System32\notepad.exe</td>
<td></td>
</tr>
<tr>
<td>User: \sean</td>
<td></td>
</tr>
<tr>
<td>Protocol: tcp</td>
<td></td>
</tr>
<tr>
<td>Initiated: true</td>
<td></td>
</tr>
<tr>
<td>SourceIsIpv6: false</td>
<td></td>
</tr>
<tr>
<td>SourceIP: 172.16.23.213</td>
<td></td>
</tr>
<tr>
<td>SourceHostname:</td>
<td></td>
</tr>
<tr>
<td>SourcePort: 62914</td>
<td></td>
</tr>
<tr>
<td>SourcePortName:</td>
<td></td>
</tr>
<tr>
<td>DestinationIsIpv6: false</td>
<td></td>
</tr>
<tr>
<td>DestinationIP: 151.101.32.133</td>
<td></td>
</tr>
<tr>
<td>DestinationHostname:</td>
<td></td>
</tr>
<tr>
<td>DestinationPort: 443</td>
<td></td>
</tr>
<tr>
<td>DestinationPortName: https</td>
<td></td>
</tr>
</tbody>
</table>

```
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
```
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)

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

https://aka.ms/wef
“PowerWare” MS Office Macro -> PowerShell


Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
Microsoft Office Macros (VBA)

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

https://www.fireeye.com/blog/threat-research/2015/10/macros_galore.html
@JohnLaTwC
https://onedrive.live.com/?authkey=%21ADev0bfQMNxv504&cid=C96A3EEDCE316E4C&id=C96A3EEDCE316E4C%21114&parId=C96A3EEDCE316E4C%211109&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”).


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


Windows Management Framework (WMF) version 5 available for download:

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

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
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](https://github.com/Ben0xA/nps)
PS> Attack is loading...
Decryption: Get-Information
Decryption: VolumeShadowCopyTools
Decryption: PowerUp
Decryption: Tater
Decryption: Invoke-Ninjacopy
Decryption: Out-Dnstxt
Decryption: Invoke-PsUACme
Decryption: dns_txt_pwnage
Decryption: Gupt-Backdoor
Decryption: Invoke-WMICommand
Decryption: Invoke-Shellcode
Decryption: Inveigh-Relay
Decryption: 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
'

with 17 modules * * */

Task Manager

<table>
<thead>
<tr>
<th>Processes</th>
<th>Performance</th>
<th>App history</th>
<th>Start-up</th>
<th>Users</th>
<th>Details</th>
<th>Services</th>
</tr>
</thead>
</table>

- **Task Manager**: 25% CPU, 66% Memory
- **Windows Command Processor**: 4.1% CPU, 8.6 MB Memory
- **Windows Explorer**: 0.7% CPU, 14.8 MB Memory

**Background processes (11)**

- Host Process for Windows Tasks: 0% CPU, 1.6 MB Memory
- Microsoft Windows Search Indexer: 0% CPU, 2.2 MB Memory
- Microsoft Volume Shadow Copy Service: 0% CPU, 0.1 MB Memory
- RDP Clipboard Monitor: 0% CPU, 1.3 MB Memory

Authentication Id : 0 ; 947799 (000000000000e7657)
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
PS Constrained Language Mode?

Welcome to PS\Attack! This is version 1.1.0, It was built on April 21, 2016 at 7:10:22 PM
If you'd like a version of PS\Attack that's even harder for A/V to detect checkout http://github.com/jarednaught/PSAttackBuildTool
For help getting started, run 'get-attack'
C:\Temp >>> invoke-mimikatz

mimikatz 2.0 alpha (x64) release "Kiwi en C" (Dec 14 2015)
Benjamin DELPY 'gentilkiwi' (benjamin@gentilkiwi.com)
http://blog.gentilkiwi.com/mimikatz
with 17 modules

Sean Metcalf
@Pyrotek3 | sean@TrimarcSecurity.com
PowerShell v5 Security Log Data?

Welcome to PSAttack! 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 PSAttack that’s 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)
http://blog.gentilkiwi.com/mimikatz

with 17 modules

mimikatz(powershell) # sekurlsa::logonpasswords

Authentication Id : 0 : 147414 (00000000:00023fd6)
Session : RemoteInteractive from 2
User Name : administrator
Domain : ADSECLAB0
Logon Server : ADS00C01
Logon Time : 5/15/2016 8:57:33 PM
SID : S-1-5-21-186993273-1316126705-865754954-500

[00000003] Primary
* Username : Administrator
* Domain : ADSECLAB0
* NTLM : 89eae239a1e8f186a205b6863a3e955f
* SHA1 : 0B3ecc38e4bce360cc55f42f16867668d650d8

[00010000] credentials.key
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.dli C:\Windows\assembly\NativeImages_v4.0.30319_.
```

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
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 400/800: HostApplication not standard Microsoft tool (PowerShell, PowerShell ISE, etc).
• **Event 400/800**: 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).
Invoke-Obfuscation

Tool :: Invoke-Obfuscation
Author :: Daniel Bohannon (DBO)
Twitter :: @danielbohannon
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
- [ ] Show this Help Menu
- [ ] Show options for payload to obfuscate
- [ ] Clear screen
- [ ] Execute ObfuscatedCommand locally
- [ ] Copy obfuscatedCommand to clipboard
- [ ] Write ObfuscatedCommand Out to disk
- [ ] Reset obfuscation for ObfuscatedCommand
- [ ] Go Back to previous obfuscation menu
- [ ] Quit Invoke-Obfuscation
- [ ] Return to Home Menu

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' for

    # 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.IMAGE_NT_HEADERS64)

    # Make sure the IMAGE_NT_HEADERS checks out. If it doesn't, the data structure is invalid. This should never happen.
    if ($imageNtHeaders64.Signature -ne 0x000004550)
    {
        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 IMAGE_NT_HEADERS -Value $imageNtHeaders32
    } else {
        $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]
    )

    $[WIN'32C'ONST'ANTS] = &(|"{1}{4}{3}{0}{2}"-f'onsta', 'Get-Win3', 'nts', 'C', '2')
    $[w In3'2FunctIONS] = & ("{4}{0}{1}{3}{2}"-f't-Win32', 'Fun', 'ns', 'ctio', 'Ge' )
    $[WI'N32TY'Pes] = & ("{0}{2}{3}{1}"-f'G', 'es', 'et-Win32', 'Typ' )
    $[PEIN'Fo] = & ("{3}{0}{5}{4}{1}{2}"-f't-PEDetail', 'In', 'fo', 'Ge', 'ed', 'l' )-PEHandle $[peH'AND'LE] -win32Types $[WIN'32TY'Pes]

    if ($[Pe'IN'Fo]."I'mAe_N'tHeader's", "oPT'ion'AlHeadeR", "IM'Por'TTABLE", "s'IZE" -gt 0 )
    {
        [IntPtr]$i'mP'Or't'dESCRIPTs'RP'Tr = &("{2}{1}{4}{3}{0}"-f'gned', 'gne', 'Add-Si', 'tASUnsi', 'dIn' ) ([Int64]$[p'E'info])
        while ( $[TR'UE] )
        {
            $i'M'p'Or't'Descriptor = win32u::"PTR'ToSTRUCTure" ($[i'mP'Or't'dESCRIPTs'RP'Tr], [Type]$[win32_Ty'Pes]."i'maGe_i" )

            if ($[importde'SCRIP'T'Or]."C'harACTEI'R'istics" -eq 0 
                -and $[i'mP'Or't'dESCRIPTs'RP'Tr]."First'T'chunk" -eq 0 
                -and $[i'mP'Or't'dESCRIPTs'RP'Tr]."forwar'd'R'ch'Ain" -eq 0 
                -and $[i'mP'Or't'dESCRIPTs'RP'Tr]."nA'Me" -eq 0 
                -and $[i'mP'Or't'dESCRIPTs'RP'Tr]."Time'D'A 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
            
            $[i'mP'Or't'dLlPA'TH] = ( gci ('VARIABLE' + ':' + 'w0' + '2u') ).value::("{0}{3}{1}{2}"-f 'P', 't', 'ringAnsi', 'trToC' )
            $[IM'P'Or't'dLlPA'N'dle] = $[WIN32F'UN'C'TIONS']."x'etODuleha'N'Dlle"."IN'V0'ke"($[i'mP'Or't'dLlPA'TH' )
            if ($[IM'P'Or't'dLlHANDLE] -eq $[NL'LL] )
            {
                ("{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
            }
        }
    }
}
Obfuscation Bypasses AV

```powershell
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>
```
Finding Obfuscated Evil

<table>
<thead>
<tr>
<th>Regular</th>
<th>Obfuscated</th>
</tr>
</thead>
<tbody>
<tr>
<td>e</td>
<td>$</td>
</tr>
<tr>
<td>t</td>
<td>{</td>
</tr>
<tr>
<td>r</td>
<td>}</td>
</tr>
<tr>
<td>a +</td>
<td></td>
</tr>
<tr>
<td>i &quot;</td>
<td></td>
</tr>
<tr>
<td>o =</td>
<td></td>
</tr>
<tr>
<td>n [</td>
<td></td>
</tr>
<tr>
<td>s (</td>
<td></td>
</tr>
<tr>
<td>l ;</td>
<td></td>
</tr>
</tbody>
</table>

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

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 `{ }`
- Look for lots of quotes (single & double) " " & ‘ ’
- Look for random function names & many unusual characters not normally in PowerShell scripts.

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

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

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
### Advanced Audit Policy Configuration

**Account Logon**
- Policy: Audit Credential Validation
- Setting: Success, Failure
- Policy: Audit Kerberos Authentication Service
- Setting: Success, Failure
- Policy: Audit Kerberos Service Ticket Operations
- Setting: Success, Failure

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

**Detailed Tracking**
- Policy: Audit DPAPI Activity
- Setting: Success, Failure
- Policy: Audit Process Creation
- Setting: Success, Failure

**DS Access**
- Policy: Audit Directory Service Access
- Setting: Success, Failure
- Policy: Audit Directory Service Changes
- Setting: Success, Failure

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

**Policy Change**
- Policy: Audit Audit Policy Change
- Setting: Success, Failure
- Policy: Audit Authentication Policy Change
- Setting: Success, Failure

---

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
<table>
<thead>
<tr>
<th>Policy</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audit: Force audit policy subcategory settings (over) to override audit policy</td>
<td>Enabled</td>
</tr>
</tbody>
</table>

### Full Auditing Policy [ADSDE03.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**

<table>
<thead>
<tr>
<th>Policy</th>
<th>Policy Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audit account logon events</td>
<td>Success, Failure</td>
</tr>
<tr>
<td>Audit account management</td>
<td>Success, Failure</td>
</tr>
<tr>
<td>Audit directory service access</td>
<td>Not Defined</td>
</tr>
<tr>
<td>Audit logon events</td>
<td>Success, Failure</td>
</tr>
<tr>
<td>Audit object access</td>
<td>Not Defined</td>
</tr>
<tr>
<td>Audit policy change</td>
<td>Not Defined</td>
</tr>
<tr>
<td>Audit privilege use</td>
<td>Success, Failure</td>
</tr>
<tr>
<td>Audit process tracking</td>
<td>Not Defined</td>
</tr>
<tr>
<td>Audit system events</td>
<td>Not Defined</td>
</tr>
</tbody>
</table>
auditpol.exe /get /category:*
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

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

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

Windows Settings

Registry

SpecialGroups (Order: 1)

<table>
<thead>
<tr>
<th>General</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hive</td>
</tr>
<tr>
<td>Key path</td>
</tr>
<tr>
<td>Value name</td>
</tr>
<tr>
<td>Value type</td>
</tr>
<tr>
<td>Value data</td>
</tr>
</tbody>
</table>

HKEY_LOCAL_MACHINE

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Audit

SpecialGroups

REG_SZ

S-1-5-113;S-1-5-21-1093224735-1015166391-1317194548-512;S-1-5-21-1093224735-1015166391-1317194548-3680
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 modified. The updated list of special groups is indicated.

|--------------------|------------------------------------|----------------|------------------|--------|-------------|

Special groups have been assigned to a new logon.

**Subject:**

- Security ID: SYSTEM
- Account Name: ADSMSRV1\$  
- Account Domain: ADSECLAB
- Logon ID: 0x3E7
- Logon GUID: {0000000-0000-0000-0000-000000000000}

**New Logon:**

- Security ID: ADSECLAB\lukeskywalker
- Account Name: lukeskywalker
- Account Domain: ADSECLAB
- Logon ID: 0x248A5
- Logon GUID: {7b7973d1-8d06-a421-7418-c2f6e2c9e2c9}
- Special Groups Assigned:
  - ADSECLAB\Special Group Auditing
  - ADSECLAB\Domain Admins

|--------------------|------------------------------------|----------------|------------------|--------|-------------|

4/23/2017 2:11:57 PM

- Task Category: Special Logon
- Keywords: Audit Success
- Computer: ADSMSRV1.lab.adsecurity.org
## Event IDs that Matter: Domain Controllers

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

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
<table>
<thead>
<tr>
<th>Event ID</th>
<th>Description</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>1102/517</td>
<td>Event log cleared</td>
<td>Attackers may clear Windows event logs.</td>
</tr>
<tr>
<td>4610/4611/4614/4622</td>
<td>Local Security Authority modification</td>
<td>Attackers may modify LSA for escalation/persistence.</td>
</tr>
<tr>
<td>4648</td>
<td>Explicit credential logon</td>
<td>Typically when a logged on user provides different credentials to access a resource. Requires filtering of “normal”.</td>
</tr>
<tr>
<td>4661</td>
<td>A handle to an object was requested</td>
<td>SAM/DSA Access. Requires filtering of “normal”.</td>
</tr>
<tr>
<td>4672</td>
<td>Special privileges assigned to new logon</td>
<td>Monitor when someone with admin rights logs on. Is this an account that should have admin rights or a normal user?</td>
</tr>
<tr>
<td>4723</td>
<td>Account password change attempted</td>
<td>If it’s not an approved/known pw change, you should know.</td>
</tr>
<tr>
<td>4964</td>
<td>Custom Special Group logon tracking</td>
<td>Track admin &amp; “users of interest” logons.</td>
</tr>
<tr>
<td>7045/4697</td>
<td>New service was installed</td>
<td>Attackers often install a new service for persistence.</td>
</tr>
<tr>
<td>4698 &amp; 4702</td>
<td>Scheduled task creation/modification</td>
<td>Attackers often create/modify scheduled tasks for persistence. Pull all events in Microsoft-Windows-TaskScheduler/Operational</td>
</tr>
<tr>
<td>4719/612</td>
<td>System audit policy was changed</td>
<td>Attackers may modify the system’s audit policy.</td>
</tr>
<tr>
<td>4732</td>
<td>A member was added to a (security-enabled) local group</td>
<td>Attackers may create a new local account &amp; add it to the local Administrators group.</td>
</tr>
<tr>
<td>4720</td>
<td>A (local) user account was created</td>
<td>Attackers may create a new local account for persistence.</td>
</tr>
</tbody>
</table>

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
Event IDs that Matter (Newer Windows systems)

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

LSA Protection & Auditing (Windows 8.1/2012R2 and newer):  

4798: A user's local group membership was enumerated (Windows 10/2016):  

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
## A Note About Logon Types (4624)

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

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
When the manual is not enough – runas /netonly, Unexpected Credential Exposure and the Need for Reality Based Holistic Threat Models

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
When the manual is not enough – runas /netonly, Unexpected Credential Exposure and the Need for Reality Based Holistic Threat Models

When the manual is not enough – runas /netonly, Unexpected Credential Exposure and the Need for Reality Based Holistic Threat Models

“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@sswOrd
User ADSECLAB\Olivia.Bryant has the password P@sswOrd
User ADSECLAB\Victoria.Young has the password P@sswOrd
User ADSECLAB\Joseph.Rodriguez has the password P@sswOrd
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
<table>
<thead>
<tr>
<th>name</th>
<th>LastBadPasswordAttempt</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADSAdministrator</td>
<td>4/11/2017 7:18:11 PM</td>
</tr>
<tr>
<td>Guest</td>
<td>4/11/2017 7:18:12 PM</td>
</tr>
<tr>
<td>DefaultAccount</td>
<td>4/11/2017 7:18:12 PM</td>
</tr>
<tr>
<td>krbtgt</td>
<td>4/11/2017 5:05:58 PM</td>
</tr>
<tr>
<td>Brandon.Young</td>
<td>4/11/2017 7:18:12 PM</td>
</tr>
<tr>
<td>Liam.Moore</td>
<td>4/11/2017 7:18:12 PM</td>
</tr>
<tr>
<td>Michael.Evans</td>
<td>4/11/2017 7:18:12 PM</td>
</tr>
<tr>
<td>Julia.Morgan</td>
<td>4/11/2017 7:18:12 PM</td>
</tr>
<tr>
<td>Jack.Collins</td>
<td>4/11/2017 7:18:12 PM</td>
</tr>
<tr>
<td>Paige.Foster</td>
<td>4/11/2017 7:18:12 PM</td>
</tr>
<tr>
<td>Charlie.Sanders</td>
<td>4/11/2017 7:18:13 PM</td>
</tr>
<tr>
<td>RyderHoward</td>
<td>4/11/2017 7:18:13 PM</td>
</tr>
<tr>
<td>Bentley.Collins</td>
<td>4/11/2017 7:18:13 PM</td>
</tr>
<tr>
<td>Abigail.Miller</td>
<td>4/11/2017 7:18:13 PM</td>
</tr>
<tr>
<td>Adrian.Thompson</td>
<td>4/11/2017 7:18:13 PM</td>
</tr>
<tr>
<td>David.Bennett</td>
<td>4/11/2017 7:18:14 PM</td>
</tr>
<tr>
<td>Asher.Alexander</td>
<td>4/11/2017 7:18:14 PM</td>
</tr>
<tr>
<td>Sydney.Taylor</td>
<td>4/11/2017 7:18:14 PM</td>
</tr>
<tr>
<td>Riley.Hill</td>
<td>4/11/2017 7:18:14 PM</td>
</tr>
<tr>
<td>Charlotte.Hayes</td>
<td>4/11/2017 7:18:14 PM</td>
</tr>
<tr>
<td>Oliver.Cook</td>
<td>4/11/2017 7:18:14 PM</td>
</tr>
<tr>
<td>Eva.Adams</td>
<td>4/11/2017 7:18:15 PM</td>
</tr>
<tr>
<td>Samuel.Cook</td>
<td>4/11/2017 7:18:15 PM</td>
</tr>
<tr>
<td>Paige.Perez</td>
<td>4/11/2017 7:18:15 PM</td>
</tr>
<tr>
<td>Parker.Foster</td>
<td>4/11/2017 7:18:15 PM</td>
</tr>
</tbody>
</table>

An account failed to log on.

- **Security ID:** NULL SID
- **Account Name:** NULL
- **Account Domain:** NULL
- **Logon ID:** 0x0
- **Logon Type:** 3
- **Account For Which Logon Failed:** NULL SID
- **Failure Information:** Unknown user name or bad password.
- **Sub Status:** 0xC000008D

The event was created by Microsoft Windows security auditing.
Switch from Network Share to AD Connection

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

<table>
<thead>
<tr>
<th>Keywords</th>
<th>Date and Time</th>
<th>Source</th>
<th>Event ID</th>
<th>Task Cate...</th>
</tr>
</thead>
</table>

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
<table>
<thead>
<tr>
<th>Keywords</th>
<th>Date and Time</th>
<th>Source</th>
<th>Event ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audit Failure</td>
<td>4/11/2017 10:21:54 PM</td>
<td>Microsoft Win...</td>
<td>4771</td>
</tr>
<tr>
<td>Audit Failure</td>
<td>4/11/2017 10:21:54 PM</td>
<td>Microsoft Win...</td>
<td>4771</td>
</tr>
<tr>
<td>Audit Failure</td>
<td>4/11/2017 10:21:54 PM</td>
<td>Microsoft Win...</td>
<td>4771</td>
</tr>
<tr>
<td>Audit Failure</td>
<td>4/11/2017 10:21:54 PM</td>
<td>Microsoft Win...</td>
<td>4771</td>
</tr>
<tr>
<td>Audit Failure</td>
<td>4/11/2017 10:21:54 PM</td>
<td>Microsoft Win...</td>
<td>4771</td>
</tr>
<tr>
<td>Audit Failure</td>
<td>4/11/2017 10:21:54 PM</td>
<td>Microsoft Win...</td>
<td>4771</td>
</tr>
<tr>
<td>Audit Failure</td>
<td>4/11/2017 10:21:54 PM</td>
<td>Microsoft Win...</td>
<td>4771</td>
</tr>
<tr>
<td>Audit Failure</td>
<td>4/11/2017 10:21:54 PM</td>
<td>Microsoft Win...</td>
<td>4771</td>
</tr>
<tr>
<td>Audit Failure</td>
<td>4/11/2017 10:21:54 PM</td>
<td>Microsoft Win...</td>
<td>4771</td>
</tr>
<tr>
<td>Audit Failure</td>
<td>4/11/2017 10:21:54 PM</td>
<td>Microsoft Win...</td>
<td>4771</td>
</tr>
<tr>
<td>Audit Failure</td>
<td>4/11/2017 10:21:54 PM</td>
<td>Microsoft Win...</td>
<td>4771</td>
</tr>
<tr>
<td>Audit Failure</td>
<td>4/11/2017 10:21:54 PM</td>
<td>Microsoft Win...</td>
<td>4771</td>
</tr>
<tr>
<td>Audit Failure</td>
<td>4/11/2017 10:21:54 PM</td>
<td>Microsoft Win...</td>
<td>4771</td>
</tr>
<tr>
<td>Audit Failure</td>
<td>4/11/2017 10:21:54 PM</td>
<td>Microsoft Win...</td>
<td>4771</td>
</tr>
<tr>
<td>Audit Failure</td>
<td>4/11/2017 10:21:54 PM</td>
<td>Microsoft Win...</td>
<td>4771</td>
</tr>
</tbody>
</table>

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 Password
User ADSECLAB\Olivia.Bryant has the password Password
User ADSECLAB\Victoria.Young has the password Password
User ADSECLAB\Joseph.Rodriguez has the password Password
User ADSECLAB\Audrey.Lee has the password Password99!
User ADSECLAB\Landon.Lewis has the password Password99!

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

<table>
<thead>
<tr>
<th>name</th>
<th>lastbadpasswordattempt</th>
<th>badpwdcount</th>
</tr>
</thead>
<tbody>
<tr>
<td>krbtgt</td>
<td>4/11/2017 8:05:58 PM</td>
<td>13</td>
</tr>
<tr>
<td>Leah.Reed</td>
<td>4/11/2017 11:37:21 PM</td>
<td>8</td>
</tr>
<tr>
<td>Arianna.Flores</td>
<td>4/11/2017 11:37:21 PM</td>
<td>8</td>
</tr>
<tr>
<td>Lily.Davis</td>
<td>4/11/2017 11:37:21 PM</td>
<td>8</td>
</tr>
<tr>
<td>Zachary.cook</td>
<td>4/11/2017 11:37:21 PM</td>
<td>8</td>
</tr>
<tr>
<td>Zachary.Lopez</td>
<td>4/11/2017 11:37:21 PM</td>
<td>8</td>
</tr>
<tr>
<td>Gabriel.Lewis</td>
<td>4/11/2017 11:37:21 PM</td>
<td>8</td>
</tr>
<tr>
<td>Lauren.Davis</td>
<td>4/11/2017 11:37:21 PM</td>
<td>8</td>
</tr>
<tr>
<td>Owen.Martin</td>
<td>4/11/2017 11:37:21 PM</td>
<td>8</td>
</tr>
<tr>
<td>Samantha.Clark</td>
<td>4/11/2017 11:37:21 PM</td>
<td>8</td>
</tr>
<tr>
<td>Adrian.Brooks</td>
<td>4/11/2017 11:37:21 PM</td>
<td>8</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Log Name</th>
<th>Security</th>
<th>Source:</th>
<th>Logged:</th>
<th>Event ID:</th>
<th>Task Category:</th>
<th>Level:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Microsoft Windows security</td>
<td></td>
<td>4771</td>
<td>Kerberos Authentication Service</td>
<td>Information</td>
</tr>
</tbody>
</table>
### General

**Event 4648, Microsoft Windows security auditing.**

<table>
<thead>
<tr>
<th>General</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A logon was attempted using explicit credentials.</strong></td>
<td></td>
</tr>
</tbody>
</table>

**Subject:**
- Security ID: ADSECLAB\joeuser
- Account Name: joeuser
- Account Domain: ADSECLAB
- Logon ID: 0xDC10D
- Logon GUID: (00000000-0000-0000-0000-000000000000)

**Account Whose Credentials Were Used:**
- Account Name: Alexa Phillips
- Account Domain: LAB.ADSECURITY.ORG
- Logon ID: 0xDC10D
- Logon GUID: (00000000-0000-0000-0000-000000000000)

**Target Server:**
- Target Server Name: ADSMD16.lab.adsecurity.org
- Additional Information: ldap/ADSMD16.lab.adsecurity.org

---

### General

**Event 4648, Microsoft Windows security auditing.**

<table>
<thead>
<tr>
<th>General</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A logon was attempted using explicit credentials.</strong></td>
<td></td>
</tr>
</tbody>
</table>

**Subject:**
- Security ID: ADSECLAB\joeuser
- Account Name: joeuser
- Account Domain: ADSECLAB
- Logon ID: 0xDC10D
- Logon GUID: (00000000-0000-0000-0000-000000000000)

**Account Whose Credentials Were Used:**
- Account Name: Nicholas Davis
- Account Domain: LAB.ADSECURITY.ORG
- Logon ID: 0xDC10D
- Logon GUID: (00000000-0000-0000-0000-000000000000)

**Target Server:**
- Target Server Name: ADSMD16.lab.adsecurity.org
- Additional Information: ldap/ADSMD16.lab.adsecurity.org
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

<table>
<thead>
<tr>
<th>Domain</th>
<th>lab.adsecurity.org</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServerName</td>
<td>adsMSSQL02.lab.adsecurity.org</td>
</tr>
<tr>
<td>Port</td>
<td>9834</td>
</tr>
<tr>
<td>Instance</td>
<td></td>
</tr>
<tr>
<td>ServiceAccountDN</td>
<td>{CN=svc-adsSQLSA,OU=TestServiceAccounts,DC=lab,DC=adsecurity,DC=org}</td>
</tr>
<tr>
<td>OperatingSystem</td>
<td>{Windows Server 2008 R2 Datacenter}</td>
</tr>
<tr>
<td>OSVersion</td>
<td>{6.1 (7601)}</td>
</tr>
<tr>
<td>Description</td>
<td>{Production SQL Server}</td>
</tr>
<tr>
<td>SrvAcctUserID</td>
<td>svc-adsSQLSA</td>
</tr>
<tr>
<td>SrvAcctDescription</td>
<td>SQL Server Service Account</td>
</tr>
</tbody>
</table>

#### UserID Details

- **UserID**: svc-SQLAgent01
- **PasswordLastSet**: 01/03/2015 18:42:01
- **LastLogon**: 12/29/2014 00:18:02
- **Description**: 
- **SPN Servers**: 
  - ADSAPPSQL01.lab.adsecurity.org
  - ADSAPPSQL02.lab.adsecurity.org
  - ADSAPPSQL03.lab.adsecurity.org
- **SPN Types**: MSSQLSvc
- **Service Principal Names**: 
  - MSSQLSvc/ADSAPPSQL01.lab.adsecurity.org:1433
  - MSSQLSvc/ADSAPPSQL02.lab.adsecurity.org:1433
  - MSSQLSvc/ADSAPPSQL03.lab.adsecurity.org:1433

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

```powershell
Add-Type -AssemblyName System.IdentityModel
-ArgumentList 'MSSQLSvc/adsgb01.lab.adsecurity.org:1433'
```

<table>
<thead>
<tr>
<th>Id</th>
<th>uuid-ce260b5a-6992-4906-a8cf-2d48439c4fc8-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>ValidFrom</td>
<td>1/23/2017 3:58:03 PM</td>
</tr>
<tr>
<td>ValidTo</td>
<td>1/24/2017 1:43:35 AM</td>
</tr>
<tr>
<td>ServicePrincipalName</td>
<td>MSSQLSvc/adsgb01.lab.adsecurity.org:1433</td>
</tr>
</tbody>
</table>

#2>

Client: JoeUser @ LAB.ADSECURITY.ORG
Server: MSSQLSvc/adsgb01.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

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
Kerberoast: Save & Crack TGS Service Ticket

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

[000000000] - 0x00000012 - aes256_hmac
  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

[000000001] - 0x00000017 - rc4_hmac_nt
  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#
All tickets cracked!
```

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
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
Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
Kerberoast Detection Redux

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. The attack is 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 password 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 services 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 rights on 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 Kerberoast 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 (@harmjoy), 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 which is 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.
Kerberoasting All User SPNs

```powershell
[array]$ServiceAccounts = Get-ADUser -Filter { ServicePrincipalName -like "*" } -Property *
$ServiceAccountSPNs = @()
ForEach ($ServiceAccountsItem in $ServiceAccounts) {
    ForEach ($ServiceAccountsItemSPN in $ServiceAccountsItem.ServicePrincipalName) {
        [array]$ServiceAccountSPNs += $ServiceAccountsItemSPN
    }
}
lst purge
ForEach ($ServiceAccountSPNItem in $ServiceAccountSPNs) {
    Add-Type -AssemblyName System.IdentityModel
}
```
## Detection

<table>
<thead>
<tr>
<th>EventID</th>
<th>Date</th>
<th>AccountName</th>
<th>ServiceName</th>
</tr>
</thead>
<tbody>
<tr>
<td>4769</td>
<td>1/25/2017</td>
<td><a href="mailto:JoeUser@LAB.ADSECURITY.ORG">JoeUser@LAB.ADSECURITY.ORG</a></td>
<td>svc-VDIPVSS01</td>
</tr>
<tr>
<td>4769</td>
<td>1/25/2017</td>
<td><a href="mailto:JoeUser@LAB.ADSECURITY.ORG">JoeUser@LAB.ADSECURITY.ORG</a></td>
<td>Svc-BizTalk01</td>
</tr>
<tr>
<td>4769</td>
<td>1/25/2017</td>
<td><a href="mailto:JoeUser@LAB.ADSECURITY.ORG">JoeUser@LAB.ADSECURITY.ORG</a></td>
<td>SVC-BOADS-01</td>
</tr>
<tr>
<td>4769</td>
<td>1/25/2017</td>
<td><a href="mailto:JoeUser@LAB.ADSECURITY.ORG">JoeUser@LAB.ADSECURITY.ORG</a></td>
<td>SVC-AGPM-01</td>
</tr>
<tr>
<td>4769</td>
<td>1/25/2017</td>
<td><a href="mailto:JoeUser@LAB.ADSECURITY.ORG">JoeUser@LAB.ADSECURITY.ORG</a></td>
<td>svc-adsMSSQL10</td>
</tr>
<tr>
<td>4769</td>
<td>1/25/2017</td>
<td><a href="mailto:JoeUser@LAB.ADSECURITY.ORG">JoeUser@LAB.ADSECURITY.ORG</a></td>
<td>svc-adsSQLSA</td>
</tr>
<tr>
<td>4769</td>
<td>1/25/2017</td>
<td><a href="mailto:JoeUser@LAB.ADSECURITY.ORG">JoeUser@LAB.ADSECURITY.ORG</a></td>
<td>svc-adsMSSQL11</td>
</tr>
<tr>
<td>4769</td>
<td>1/25/2017</td>
<td><a href="mailto:JoeUser@LAB.ADSECURITY.ORG">JoeUser@LAB.ADSECURITY.ORG</a></td>
<td>SQL-ADSDB317-SVC</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>SAMAccountname</th>
<th>ServicePrincipalName</th>
</tr>
</thead>
<tbody>
<tr>
<td>krbtgt</td>
<td>{kadmin/changepw}</td>
</tr>
<tr>
<td>KerberoastHONEYPOT</td>
<td>{MSSQLSVC/honeypot.lab.adsecurity.org:ItsATrap}</td>
</tr>
</tbody>
</table>
```

```
#1> Client: JoeUser @ LAB.ADSECURITY.ORG
Server: MSSQLSVC/honeypot.lab.adsecurity.org:ItsATrap @ LAB.ADSECURITY.ORG
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)
But wait, there’s more!
More Kerberoasting Fun!

User logon name: svc-LogRead
User logon name (pre-Windows 2000): ADSECLAB\ svc-LogRead

 svc-LogRead Properties

Attributes:
- servicePrincipalName: MSSQLSvc/LRSQL12.lab.adsecurity.org

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
More Kerberoasting Fun!

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

```plaintext
Id: uuid-ee83d1c4-0769-4548-90f6-784c6589a6f2-19
ValidFrom: 4/11/2017 5:06:04 PM
ValidTo: 4/12/2017 3:06:04 AM
ServicePrincipalName: MSSQLSvc/LRSQL12.lab.adsecurity.org
```

---

**#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

---

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

• 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!

Slides:  Presentations.ADSecurity.org

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

- Monitoring what matters – Windows Event Forwarding for everyone (even if you already have a SIEM.)

- PowerShell ♥ the Blue Team

- PS>Attack
  [https://github.com/jaredhaight/PSAttack](https://github.com/jaredhaight/PSAttack)

- Invoke-Obfuscation
  [https://github.com/danielbohannon/Invoke-Obfuscation](https://github.com/danielbohannon/Invoke-Obfuscation)

- Events to monitor:

- 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

- Cracking Kerberos TGS Tickets Using Kerberoast – Exploiting Kerberos to Compromise the Active Directory Domain
  [https://adsecurity.org/?p=2293](https://adsecurity.org/?p=2293)

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
Appendix:
Auditing Subcategories to Event IDs
## Auditing Subcategories to Events

<table>
<thead>
<tr>
<th>Auditing Subcategory</th>
<th>Event IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Audit Audit Policy Change</strong></td>
<td>4719: System audit policy was changed.</td>
</tr>
<tr>
<td></td>
<td><strong>4908</strong>: Special Groups Logon table modified.</td>
</tr>
<tr>
<td><strong>Audit Authentication Policy Change</strong></td>
<td>4706: A new trust was created to a domain.</td>
</tr>
<tr>
<td></td>
<td>4707: A trust to a domain was removed.</td>
</tr>
<tr>
<td></td>
<td>4713: Kerberos policy was changed.</td>
</tr>
<tr>
<td></td>
<td><strong>4716</strong>: Trusted domain information was modified.</td>
</tr>
<tr>
<td></td>
<td>4717: System security access was granted to an account.</td>
</tr>
<tr>
<td></td>
<td>4718: System security access was removed from an account.</td>
</tr>
<tr>
<td></td>
<td><strong>4739</strong>: Domain Policy was changed.</td>
</tr>
<tr>
<td></td>
<td>4865: A trusted forest information entry was added.</td>
</tr>
<tr>
<td></td>
<td>4866: A trusted forest information entry was removed.</td>
</tr>
<tr>
<td></td>
<td><strong>4867</strong>: A trusted forest information entry was modified.</td>
</tr>
<tr>
<td></td>
<td>4706: A new trust was created to a domain.</td>
</tr>
<tr>
<td></td>
<td>4707: A trust to a domain was removed.</td>
</tr>
<tr>
<td><strong>Audit Computer Account Management</strong></td>
<td>4741: A computer account was created.</td>
</tr>
<tr>
<td></td>
<td><strong>4742</strong>: A computer account was changed.</td>
</tr>
<tr>
<td></td>
<td>4743: A computer account was deleted.</td>
</tr>
</tbody>
</table>
## Auditing Subcategories to Events

<table>
<thead>
<tr>
<th>Auditing Subcategory</th>
<th>Event IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Audit DPAPI Activity</strong></td>
<td>4692: Backup of data protection master key was attempted.</td>
</tr>
<tr>
<td></td>
<td>4693: Recovery of data protection master key was attempted.</td>
</tr>
<tr>
<td></td>
<td>4695: Unprotection of auditable protected data was attempted.</td>
</tr>
<tr>
<td><strong>Audit Kerberos Authentication Service</strong></td>
<td>4768: A Kerberos authentication ticket (TGT) was requested</td>
</tr>
<tr>
<td></td>
<td>4771: Kerberos pre-authentication failed</td>
</tr>
<tr>
<td></td>
<td>4772: Kerberos authentication ticket request failed</td>
</tr>
<tr>
<td><strong>Audit Kerberos Service Ticket Operation</strong></td>
<td>4769: A Kerberos service ticket (TGS) was requested</td>
</tr>
<tr>
<td></td>
<td>4770: A Kerberos service ticket was renewed</td>
</tr>
<tr>
<td><strong>Audit Logoff</strong></td>
<td>4634: An account was logged off.</td>
</tr>
<tr>
<td><strong>Audit Logon</strong></td>
<td>4624: An account was successfully logged on.</td>
</tr>
<tr>
<td></td>
<td>4625: An account failed to log on.</td>
</tr>
<tr>
<td></td>
<td>4648: A logon was attempted using explicit credentials.</td>
</tr>
<tr>
<td><strong>Audit Other Account Logon Events</strong></td>
<td>4648: A logon was attempted using explicit credentials</td>
</tr>
<tr>
<td></td>
<td>4649: A replay attack was detected.</td>
</tr>
<tr>
<td></td>
<td>4800: The workstation was locked.</td>
</tr>
<tr>
<td></td>
<td>4801: The workstation was unlocked.</td>
</tr>
<tr>
<td></td>
<td>5378: The requested credentials delegation was disallowed by policy.</td>
</tr>
</tbody>
</table>
## Auditing Subcategories to Events

<table>
<thead>
<tr>
<th>Auditing Subcategory</th>
<th>Event IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audit Other Object Access Events</td>
<td>4698: A scheduled task was created. 4699: A scheduled task was deleted. 4702: A scheduled task was updated.</td>
</tr>
<tr>
<td>Audit Process Creation</td>
<td>4688: A new process has been created.</td>
</tr>
<tr>
<td>Audit Security Group Management</td>
<td>4728: A member was added to a security-enabled global group. 4729: A member was removed from a security-enabled global group. 4732: A member was added to a security-enabled local group. 4733: A member was removed from a security-enabled local group. 4735: A security-enabled local group was changed. 4737: A security-enabled global group was changed. 4755: A security-enabled universal group was changed. 4756: A member was added to a security-enabled universal group. 4757: A member was removed from a security-enabled universal group. 4764: A group's type was changed.</td>
</tr>
<tr>
<td>Audit Security System Extension</td>
<td>4610: An authentication package has been loaded by the Local Security Authority. 4611: A trusted logon process has been registered with the Local Security Authority. 4697: A service was installed in the system.</td>
</tr>
<tr>
<td>Auditing Subcategory</td>
<td>Event IDs</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Audit Sensitive Privilege Use            | 4672: Special privileges assigned to new logon.  
4673: A privileged service was called.  
4674: An operation was attempted on a privileged object. |
| Audit Special Logon                      | 4964: Special groups have been assigned to a new logon.                   |
| Audit User Account Management            | 4720: A user account was created.  
4722: A user account was enabled.  
4723: An attempt was made to change an account's password.  
4724: An attempt was made to reset an account's password.  
4725: A user account was disabled.  
4726: A user account was deleted.  
4738: A user account was changed.  
4740: A user account was locked out.  
4765: SID History was added to an account.  
4766: An attempt to add SID History to an account failed.  
4767: A user account was unlocked.  
4780: The ACL was set on accounts which are members of administrators groups.  
4794: An attempt was made to set the Directory Services Restore Mode. |