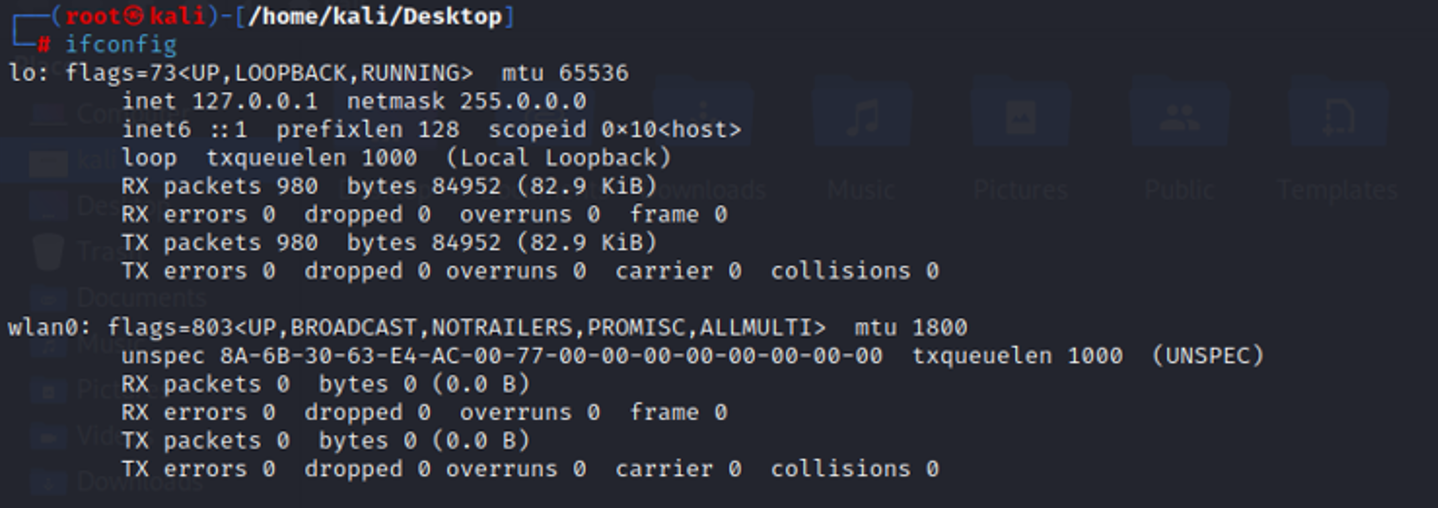
## Airmon-ng

## On wireless interfaces, monitor mode can be enabled using this script. Alternatively, it can be used to terminate network managers or switch back from managed to monitor mode. The status of the interfaces can be seen by entering the airmon-ng command without any arguments. It can also make a list of programs that could harm the wireless card's performance and delete them.IN the above screenshot we can see that we have started the air-mon and then kill the processes.

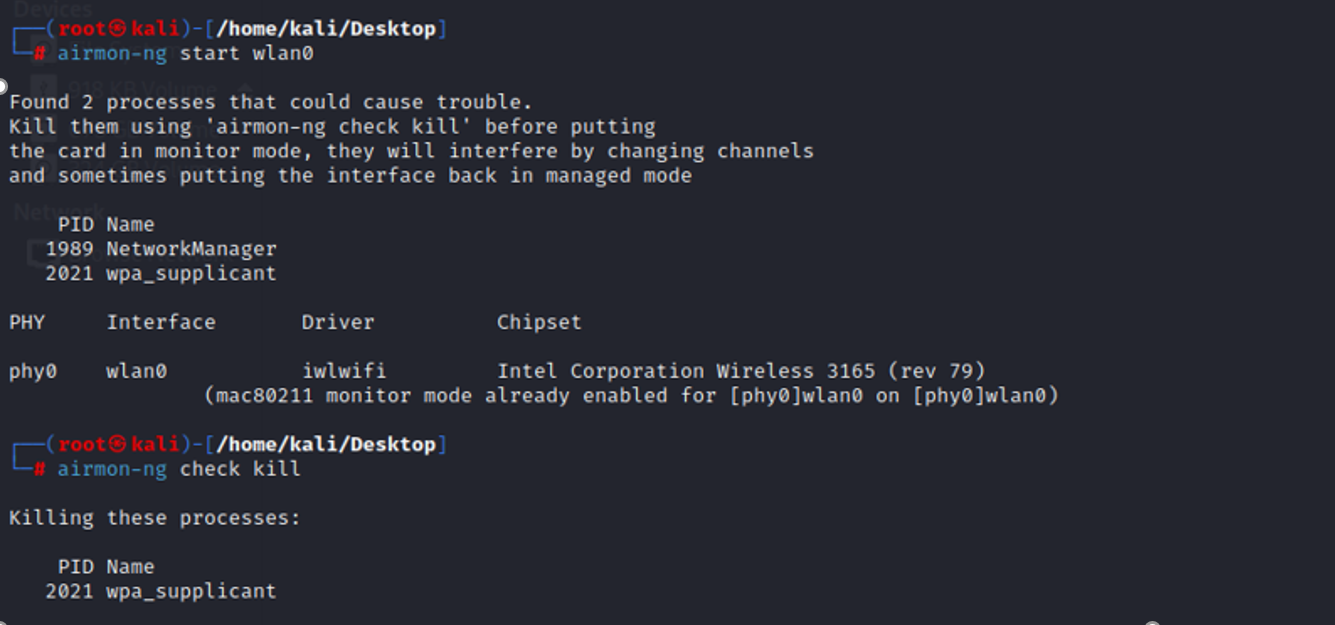
Uses: An ethical hacker records each of these packets to see if the router is exposed or not. It is also employed to ascertain whether the network is at risk. Every device has all the information required. It is also used to monitor the busy traffic.

Test cases

1.



2.

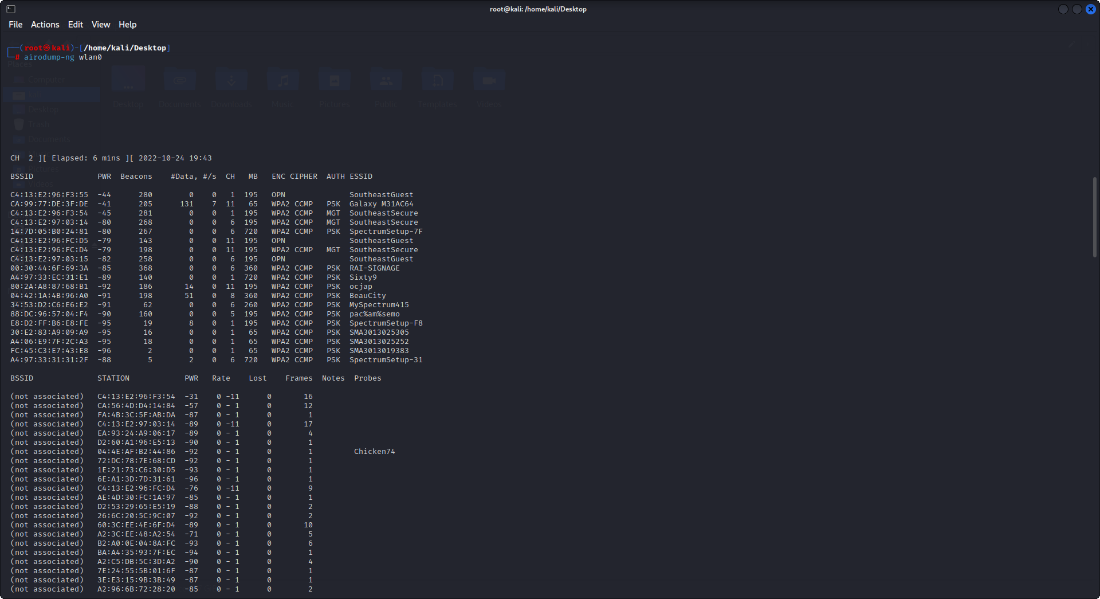


## Airodump-ng

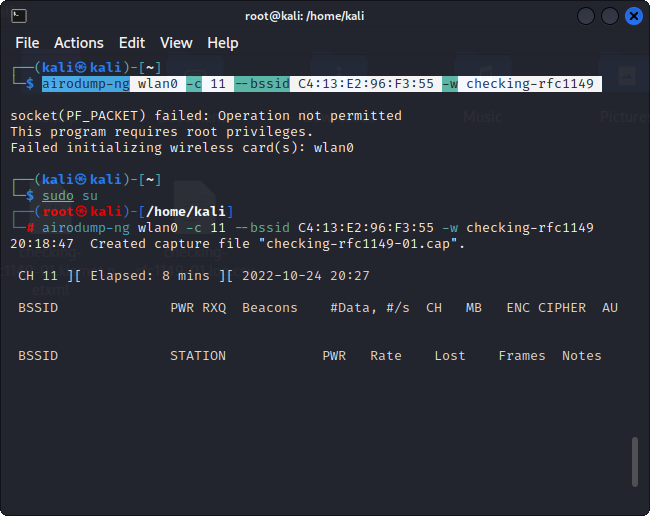
Using Airodump-ng, raw 802.11 frames are captured during packet capture. In order to use them with aircrack-ng, it is especially well suited for gathering WEP IVs (Initialization Vector) or WPA handshakes. The GPS coordinates of the discovered access points can be recorded by airodump-ng if a GPS receiver is attached to the PC. Additionally, airodump-ng writes out several files containing the details of all access points and clients seen, which can be used for scripting, or creating custom tools

It is a packet capture tool that records and stores unprocessed data packets for further examination. Airodump-ng may also retrieve the access point coordinates if your computer is connected to a GPS receiver. You can begin capturing packets with airodump once monitor mode has been enabled with airmon-ng. All the nearby networks are listed using the Airdump-ng tool, which also provides vital information about each one. Since it is a packet sniffer, its primary purpose when we are in Monitor mode is to collect all the packets surrounding us. In order to gather valuable data, such as the mac address, channel name, encryption type, and the number of clients connected to the network, we can run it against all of the nearby networks before directing it at the target network. To just capture packets from a specific Wi-Fi network, we may also run it against a specific AP (access point).

Test case 1:



Test case 2



## Airbase-ng

In contrast to the Access Point (AP), Airbase-ng is a versatile tool designed to attack customers. The implementation's major goal is to persuade users to connect to the fictitious access point rather than preventing them from using the genuine hotspot. Running airbase-ng results in the creation of a touch interface (atX). This can be used to send encrypted packets or to receive packets that have been decrypted. These frames are crucial for connecting real clients to our softAP since they are more likely to send probing requests for shared/configured networks. In such a scenario, the access point will respond to any probe request with a proper probe response, alerting the authentication client depending on the BSSID of the airbase.

* Implements the Caffe Latte WEP client attack
* Implements the Hirte WEP client attack
* Ability to cause the WPA/WPA2 handshake to be captured
* Ability to act as an ad-hoc Access Point
* Ability to act as a full Access Point
* Ability to filter by SSID or client MAC addresses
* Ability to manipulate and resend packets
* Ability to encrypt sent packets and decrypt received packets

Usage:

usage: airbase-ng <options> <replay interface>

Options

-a bssid : set Access Point MAC address

-i iface : capture packets from this interface

-w WEP key : use this WEP key to encrypt/decrypt packets

-h MAC : source mac for MITM mode

-f disallow : disallow specified client MACs (default: allow)

-W 0|1 : [don't] set WEP flag in beacons 0|1 (default: auto)

-q : quiet (do not print statistics)

-v : verbose (print more messages) (long --verbose)

-M : M-I-T-M between [specified] clients and bssids (NOT CURRENTLY

IMPLEMENTED)

-A : Ad-Hoc Mode (allows other clients to peer) (long --ad-hoc)

-Y in|out|both : external packet processing

-c channel : sets the channel the AP is running on

-X : hidden ESSID (long --hidden)

-s : force shared key authentication

-S : set shared key challenge length (default: 128)

-L : Caffe-Latte attack (long --caffe-latte)

-N : Hirte attack (cfrag attack), creates arp request against wep client (long –cfrag)

-x nbpps : number of packets per second (default: 100)

-y : disables responses to broadcast probes

-0 : set all WPA,WEP,open tags. can't be used with -z & -Z

-z type : sets WPA1 tags. 1=WEP40 2=TKIP 3=WRAP 4=CCMP 5=WEP104

-Z type : same as -z, but for WPA2

-V type : fake EAPOL 1=MD5 2=SHA1 3=auto

-F prefix : write all sent and received frames into pcap file

-P : respond to all probes, even when specifying ESSIDs

-I interval : sets the beacon interval value in ms

-C seconds : enables beaconing of probed ESSID values (requires -P)

Filter options:

 --bssid <MAC> : BSSID to filter/use (short -b)

 --bssids <file> : read a list of BSSIDs out of that file (short -B)

 --client <MAC> : MAC of client to accept (short -d)

 --clients <file> : read a list of MACs out of that file (short -D)

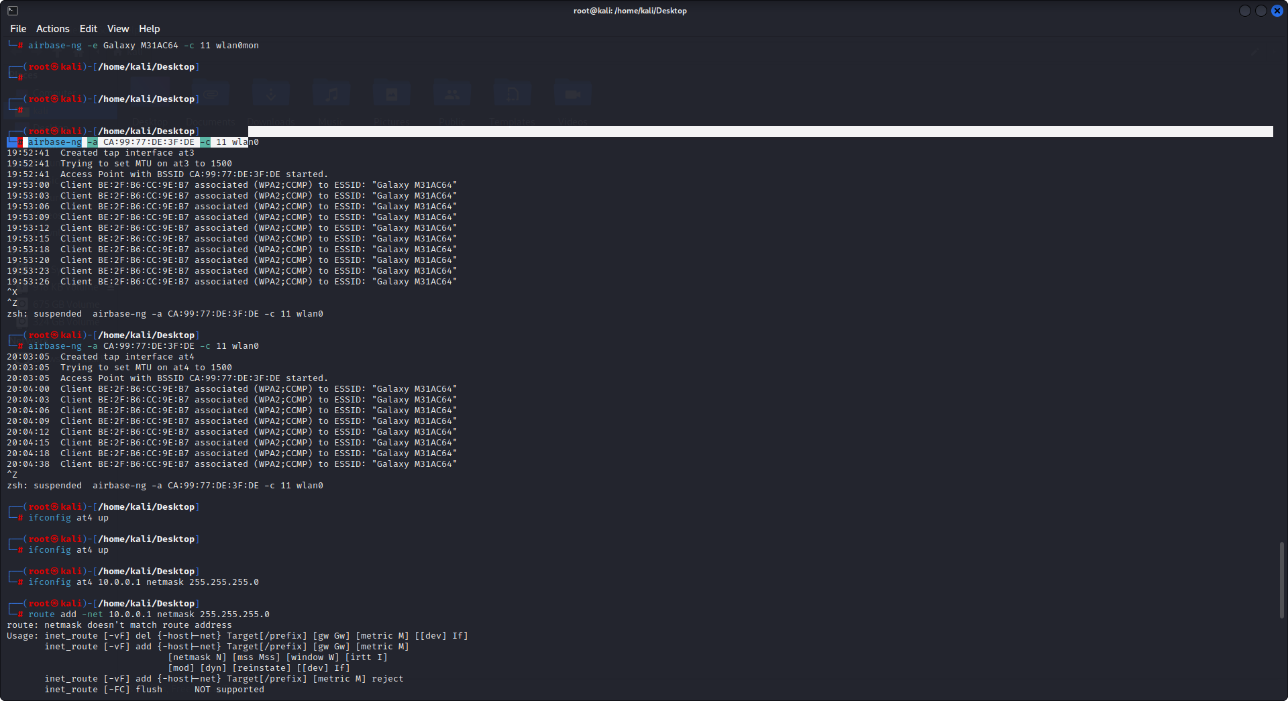
 --essid <ESSID> : specify a single ESSID (short -e)

 --essids <file> : read a list of ESSIDs out of that file (short -E)

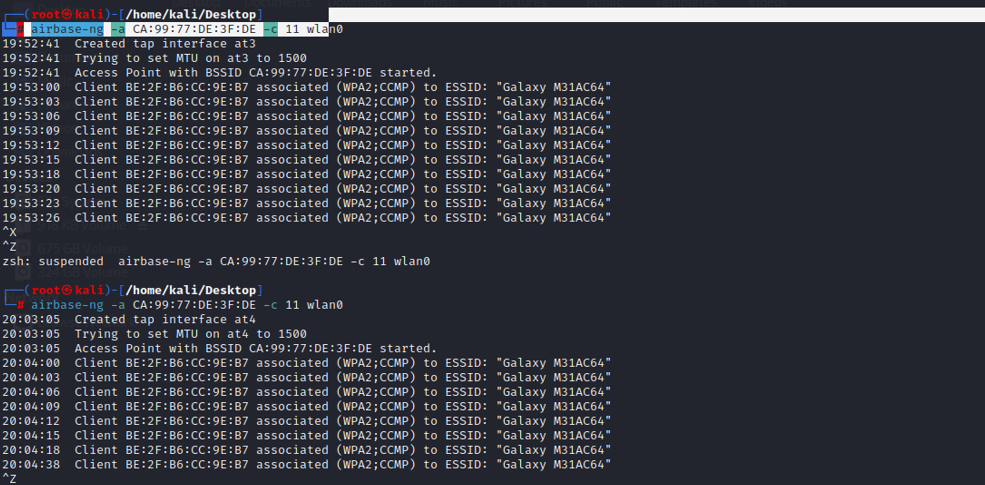
Help:

 --help: Displays the usage screen (short -H)

Test case 1:



Test case 2



## Aircrack-ng

A detector, packet sniffer, WEP and WPA/WPA2-PSK cracker, and analysis software are all included in the network software package known as Aircrack-ng for 802.11 wireless LANs. Aircrack-ng is a fork of the original Aircrack project.

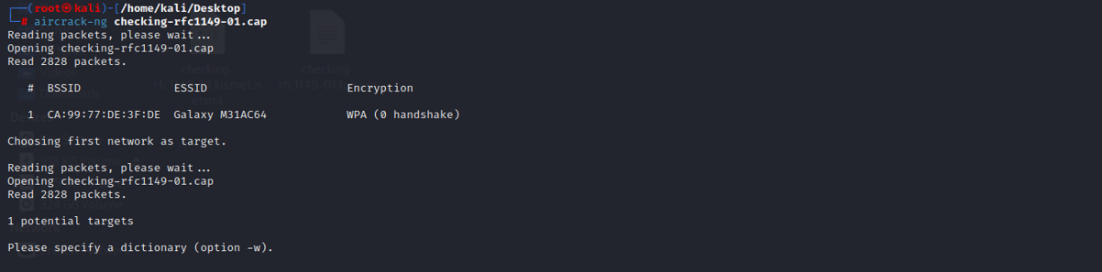
The Aircrack-ng toolkit searches Wi-Fi networks for security weaknesses. You may monitor WiFi security, gather data packets, and export them to text files for more in-depth analysis. To check the functionality of WiFi cards, it is feasible to capture and inject them. When cracking WEP and WPA/WPA2-PSK keys, Aircrack-ng employs a variety of methods. A complete set of tools for evaluating WiFi network security is called Aircrack-ng.

It focuses on different areas of WiFi security:

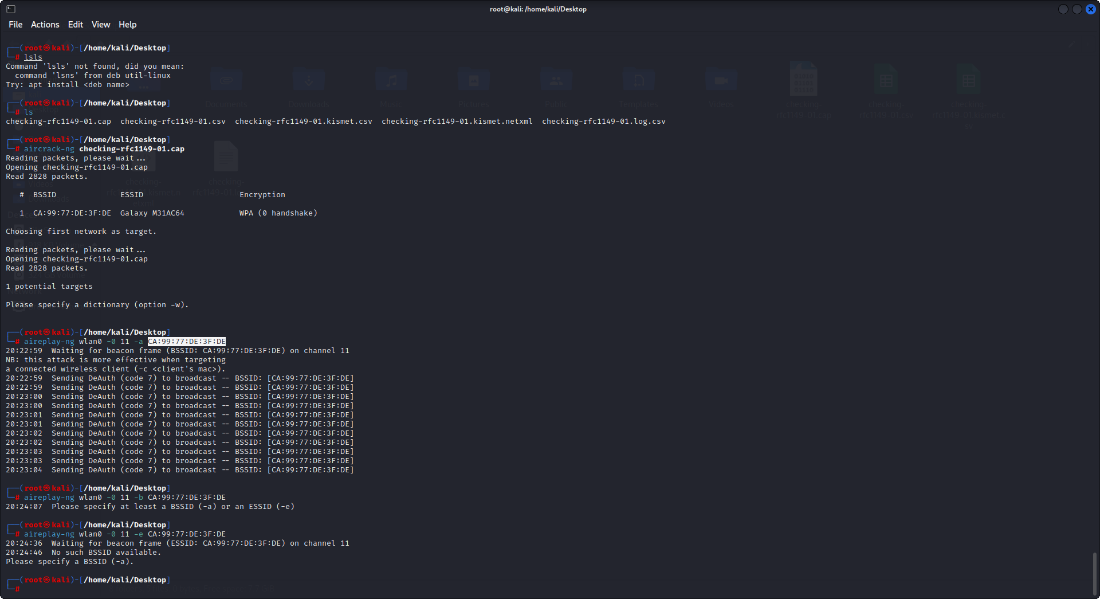
* Monitoring: Packet capture and export of data to text files for further processing by third party tools.
* Attacking: Replay attacks, deauthentication, fake access points and others via packet injection.
* Testing: Checking WiFi cards and driver capabilities (capture and injection).
* Cracking: WEP and WPA PSK (WPA 1 and 2).

It works with any wireless network interface controller whose driver can sniff 802.11a, 802.11b, and 802.11g traffic and provides raw monitoring mode. The application is compatible with Linux, FreeBSD, macOS, OpenBSD, and Windows. The Linux version has also been transferred to the Android, Zaurus PDA, and Maemo platforms, and an iPhone proof of concept port has been produced.

Test case 1



Test case 2

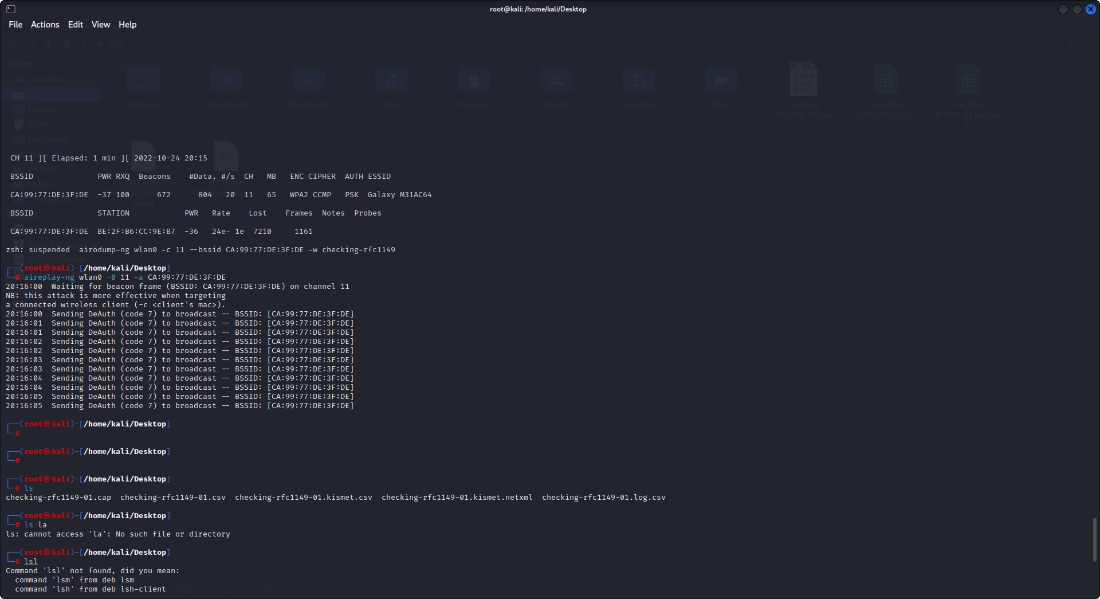


## Aireplay-ng

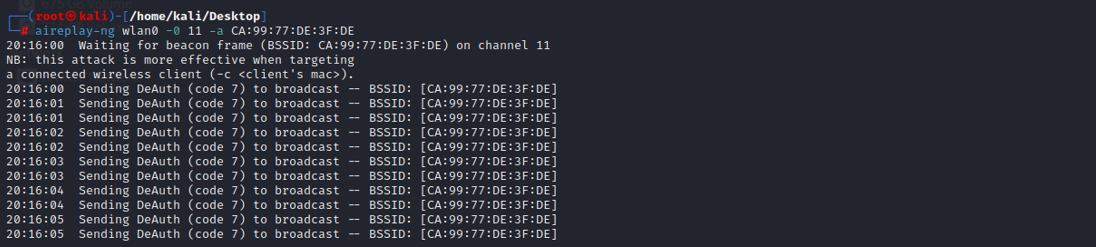
It uses Aireplay-ng to inject frames. The main purpose is to produce traffic that will subsequently be used by aircrack-ng to decrypt the WEP and WPA-PSK keys. Deauthentication can be caused by a variety of techniques, including forged authentications, interactive packet replay, specially constructed ARP request injection, and ARP-request reinjection. The utility packet forge-ng allows for the creation of arbitrary frames. It is intended to do injection in order to generate fake traffic that can be utilized for WEP cracking. Wireless rogue traffic is produced using Aireplay-ng. To crack WEP and WPA keys, it can be used in conjunction with aircrack-ng. Aireplay-primary NG's function is to inject frames. With aireplay-ng, you can carry out a variety of strong attacks, including the deauthentication attack, which aids in capturing WPA handshake data, and the fake authentication attack, which involves injecting packets into the network access point while pretending to be a legitimate user in order to create and capture new IVs. The following list also includes other kinds of assaults:

* Interactive packet replay attack
* ARP request replay attack
* KoreK chopchop attack
* Cafe-latte attack
* Fragmentation attack

Test case 1



Test case 2



## 6. Airolib-ng

To store and manage essid and password lists, generate their Pairwise Master Keys (PMKs), and use them in WPA/WPA2 cracking, use the airolib-ng program from the aircrack-ng package. The application employs the portable SQLite3 database, which is available on most platforms, as its storage method. The administration, memory, and disk overhead, along with platform availability, were all taken into account while choosing the SQLite3 database.

Calculating the pairwise master key, from which the private transient key (PTK) is obtained, is a step in the WPA/WPA2 cracking process. We may calculate the frame message identity code (MIC) for a particular packet using the PTK, and if the MIC matches that of the packet, then the PTK was accurate and the PMK was also correct.

Computing the PMK is still required, yet we can:

- Precompute it for later and/or shared use.

- Use distributed machines to generate the PMK and use their value elsewhere.

OPERATION

--stats

Output information about the database.

--sql

Execute specified SQL statement.

--clean [all]

Clean the database from old junk. When specifying 'all', it will also reduce

filesize if possible and run an integrity check.

--batch

Start batch-processing all combinations of ESSIDs and passwords.

--verify [all]

Verify a set of randomly chosen PMKs. If 'all' is given, all invalid PMK in the

database will be deleted.

--import [essid|passwd]

Import a flat file as a list of ESSIDs or passwords.

import cowpatty

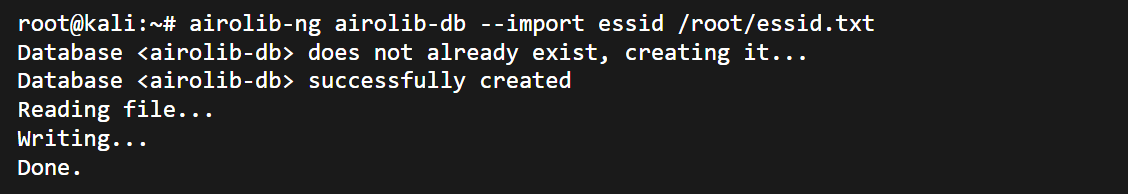
Import a coWPAtty file.

--export cowpatty

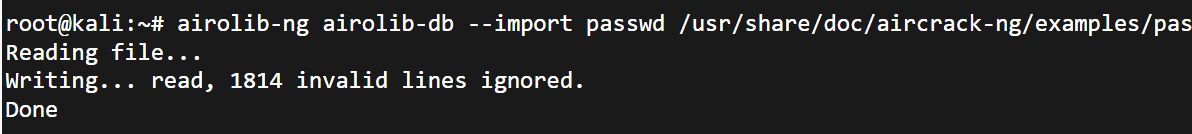
Export to a cowpatty file

Test case 1:

To import a file containing the ESSIDs of the network(s) you are targeting, use the command line option -import essid /root/essid.txt and provide the database name to use (airolib-db). The database will be created if it doesn't already exist.



Any wordlists you want to utilize to calculate PMK should be imported.



## 7. Airserv-ng

A client-server TCP network connection is used by Airserv-ng, a wireless card server, to enable different wireless application programs to independently use a wireless card. The server contains all operating system- and wireless card driver-specific code. Due to this, each wireless application is no longer required to incorporate the intricate wireless card and driver logic. Additionally, it supports a variety of operating systems.

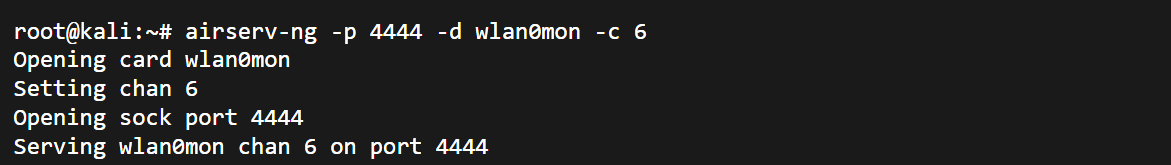
When the server is launched, it waits for client connections on a certain IP address and TCP port. This IP address and port are then used by the wireless application to connect to the server. Instead of utilizing the network interface, you specify "server IP address> colon "port number>" when using the aircrack-ng suite methods. One such instance is 127.0.0.1:666.

This allows for a number of interesting possibilities:

* By eliminating the wireless card/driver complexity, software developers can concentrate on the application functionality. This will lead to a larger set of applications being available. It also dramatically reduces the maintenance effort.
* Remote sensors are now easy to implement. Only a wireless card and airserv-ng are required to be running on the remote sensor. This means that small embedded systems can easily be created.
* You can mix and match operating systems. Each piece can run on a different operating system. The server and each of the applications can potentially run under a different operating system.
* Some wireless cards do not allow multiple applications to access them at once. This constraint is now eliminated with the client-server approach.
* By using TCP networking, the client and server can literally be in different parts of the world. As long as you have network connectivity, then it will work.

Test case :

Start a server instance utilizing the wlan0mon interface on channel 6 at a given port (-p 4444).



## 8. Airtun-ng

A virtual tunnel interface designer is airtun-ng. You need the encryption key and the bssid for the network you want to monitor in order to execute wIDS data collection. All of the traffic for that network is decrypted by Airtun-ng before being sent to a conventional IDS system like Snort.

If you have the complete encryption key, traffic injection can be totally bidirectional. If you have the PRGA gained from chopchop or fragmentation assaults, it is outgoing unidirectional. The main benefit of airtun-ng over the other injection tools in the aircrack-ng suite is that you may produce, inject, or sniff packets using any other tool after using airtun-ng.

Additionally, Airtun-ng features repeater and tcpreplay-like features. A wireless device's detected traffic can be replayed using the repeater function, which also offers the ability to filter the traffic using a bssid and network mask before replaying the remaining traffic. The repeater function's interface is given by the -i at0 flag. You can keep using the tun interface while repeating this. Additionally, a pcap file read function enables you to replay packet captures in pcap format exactly as you did when you first collected them. For wifi, this is effectively tcpreplay functionality.

Usage

Usage: airtun-ng

-x nbpps : maximum number of packets per second (optional)

-a bssid : set Access Point MAC address (mandatory). In WDS Mode this sets the

Receiver

-i iface : capture packets from this interface (optional)

-y file : read PRGA from this file (optional / one of -y or -w must be defined)

-w wepkey : use this WEP-KEY to encrypt packets (optional / one of -y or -w must be

defined)

-p pass : use this WPA passphrase to decrypt packets (use with -a and -e)

-e essid : target network SSID (use with -p)

-t tods : send frames to AP (1) or to client (0) or tunnel them into a WDS/Bridge (2)

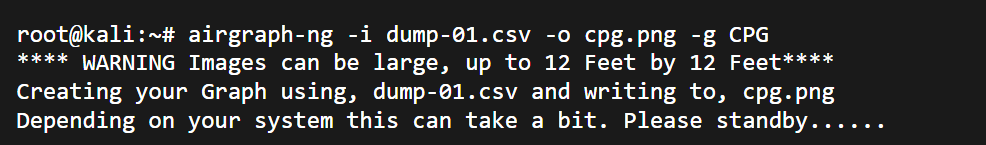
-r file : read frames out of pcap file (optional)

-h MAC : source MAC address

Usage:

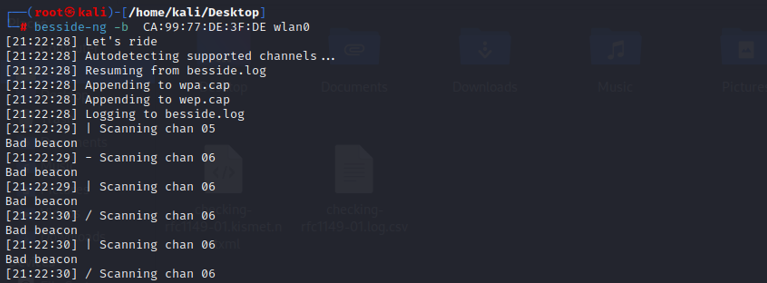
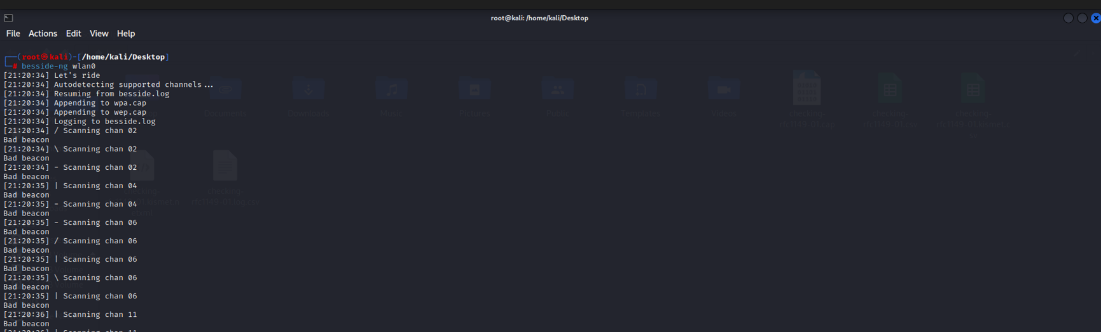
CPG chart

Choose the graph type, the output file to create (-o cpg.png), and the input file to use (-i dump-01.csv) (-g CAG).



## 9. Besside-ng

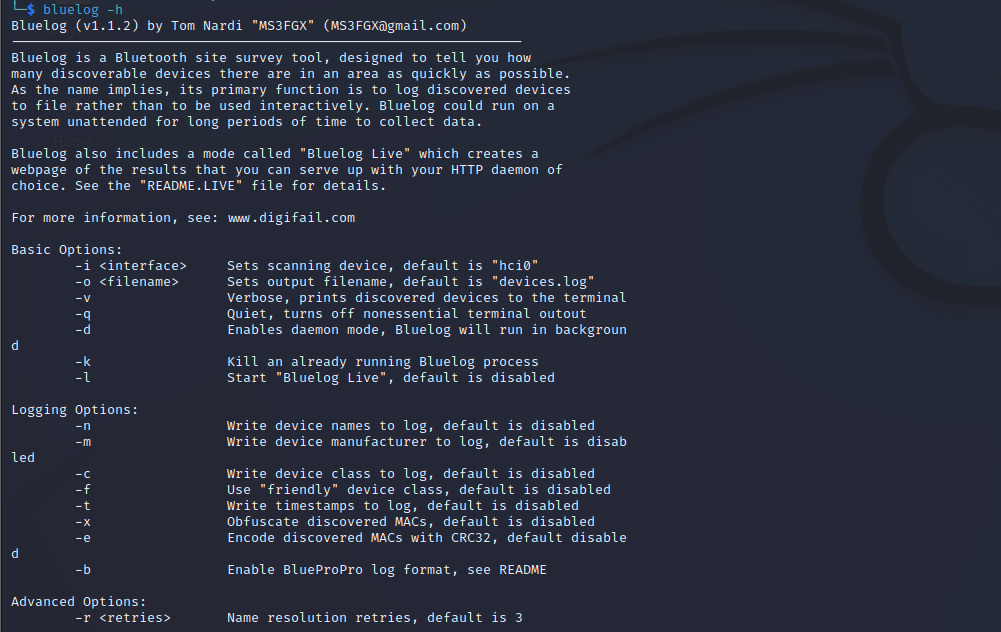
A similar utility to Wesside-ng, Besside-ng also supports WPA encryption. Which will automatically decrypt all nearby WEP networks and record WPA handshakes. In order to try and crack the password, WPA handshakes can be uploaded to the internet cracking service at Darkircop.org (Beside-ng Companion), which also offers helpful statistics based on user-submitted capture files on the viability of WPA cracking. Beside-ng needs the aircrack-ng SVN version and a wireless interface with functional injection.

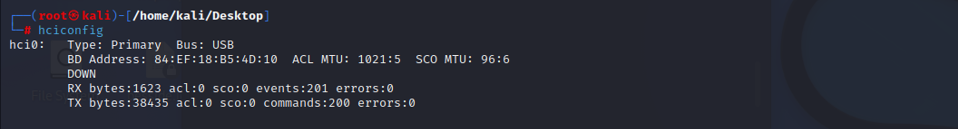




## 10. Bluelog

A Bluetooth scanner called Bluelog is made to provide information on the number of discoverable devices in a location as rapidly as feasible. It is designed to be used as a site survey tool to count the potential Bluetooth targets in the immediate vicinity. The sole purpose of the Linux Bluetooth scanner Bluelog is to log devices that are in discoverable mode. It is designed to be used as a site survey tool to count the number of Bluetooth devices that may be found in a specific area.





## 11. BlueMaho

It serves as the user interface (GUI) for a toolkit for evaluating the security of Bluetooth devices. It is open source, freeware, written in Python, and makes use of wxPython. It can be used to test BT devices for known vulnerabilities, however the main task is to test for undiscovered flaws. It can also produce lovely statistics. A freeware open source program called BlueMaho is used to test the security of Bluetooth devices and can be used to identify both known and undiscovered flaws. It was created in Python. Penetration testers frequently use it to assess the security. Martin Herfurt is the author, and plate form is Linux.