Aireplay-ng

Description

Aireplay-ng is used to inject frames.

The primary function is to generate traffic for the later use in aircrack-ng for cracking the WEP and WPA-PSK keys. There are different attacks which can cause deauthentications for the purpose of capturing WPA handshake data, fake authentications, Interactive packet replay, hand-crafted ARP request injection and ARP-request reinjection. With the packetforge-ng tool it's possible to create arbitrary frames.

Most drivers needs to be patched to be able to inject, don't forget to read Installing drivers.

Usage of the attacks

It currently implements multiple different attacks:

- Attack 0: Deauthentication
- Attack 1: Fake authentication
- Attack 2: Interactive packet replay
- Attack 3: ARP request replay attack
- Attack 4: KoreK chopchop attack
- Attack 5: Fragmentation attack
- Attack 6: Cafe-latte attack
- Attack 7: Client-oriented fragmentation attack
- Attack 8: WPA Migration Mode – will be available in the next release-
- Attack 9: Injection test

Usage

This section provides a general overview. Not all options apply to all attacks. See the details of the specific attack for the relevant details.

Usage:

```
aireplay-ng <options> <replay interface>
```

For all the attacks except deauthentication and fake authentication, you may use the following filters to limit which packets will be presented to the particular attack. The most commonly used filter option is the ”-b” to select a specific access point. For typical usage, the ”-b” is the only one you use.

Filter options:

- -b bssid : MAC address, Access Point

- -d dmac : MAC address, Destination
- -s smac : MAC address, Source
- -m len : minimum packet length
- -n len : maximum packet length
- -u type : frame control, type field
- -v subt : frame control, subtype field
- -t tods : frame control, To DS bit
- -f fromds : frame control, From DS bit
- -w iswep : frame control, WEP bit

When replaying (injecting) packets, the following options apply. Keep in mind that not every option is relevant for every attack. The specific attack documentation provides examples of the relevant options.

Replay options:

- -x nbpps : number of packets per second
- -p fctrl : set frame control word (hex)
- -a bssid : set Access Point MAC address
- -c dmac : set Destination MAC address
- -h smac : set Source MAC address
- -e essid : For fakeauth attack or injection test, it sets target AP SSID. This is optional when the SSID is not hidden.
- -j : arpreplay attack : inject FromDS pkts
- -g value : change ring buffer size (default: 8)
- -k IP : set destination IP in fragments
- -l IP : set source IP in fragments
- -o npckts : number of packets per burst (-1)
- -q sec : seconds between keep-alives (-1)
- -y prga : keystream for shared key auth
- ”-B” or ”–bittest” : bit rate test (Applies only to test mode)
- ”-D” :disables AP detection. Some modes will not proceed if the AP beacon is not heard. This disables this functionality.
- ”-F” or ”–fast” : chooses first matching packet. For test mode, it just checks basic injection and skips all other tests.
- ”-R” disables /dev/rtc usage. Some systems experience lockups or other problems with RTC. This disables the usage.

The attacks can obtain packets to replay from two sources. The first being a live flow of packets from your wireless card. The second being from a pcap file. Standard Pcap format (Packet CAPture, associated with the libpcap library http://www.tcpdump.org [http://www.tcpdump.org]), is recognized by most commercial and open-source traffic capture and analysis tools. Reading from a file is an often overlooked feature of aireplay-ng. This allows you to read packets from other capture sessions. Keep in mind that various attacks generate pcap files for easy reuse.

Source options:

- iface : capture packets from this interface

- -r file : extract packets from this pcap file

This is how you specify which mode (attack) the program will operate in. Depending on the mode, not all options above are applicable.

Attack modes (Numbers can still be used):

- - -deauth count : deauthenticate 1 or all stations (-0)
- - -fakeauth delay : fake authentication with AP (-1)
- - -interactive : interactive frame selection (-2)
- - -arpreplay : standard ARP-request replay (-3)
- - -chopchop : decrypt/chopchop WEP packet (-4)
- - -fragment : generates valid keystream (-5)
- - -test : injection test (-9)

Fragmentation vs. Chopchop

Here are the differences between the fragmentation and chopchop attacks

Fragmentation

Pros:

- Typically obtains the full packet length of 1500 bytes xor. This means you can subsequently pretty well create any size of packet. Even in cases where less then 1500 bytes are collected, there is sufficient to create ARP requests.
- May work where chopchop does not.
- Is extremely fast. It yields the xor stream extremely quickly when successful.

Cons:

- Need more information to launch it - IE IP address info. Quite often this can be guessed. Better still, aireplay-ng assumes source and destination IPs of 255.255.255.255 if nothing is specified. This will work successfully on most if not all APs. So this is a very limited con.
- Setup to execute the attack is more subject to the device drivers. For example, Atheros does not generate the correct packets unless the wireless card is set to the mac address you are spoofing.
- You need to be physically closer to the access point because if any packets are lost then the attack fails.
- The attack will fail on access points which do not properly handle fragmented packets.

Chopchop

Pros:

- May work where fragmentation does not work.
- You don't need to know any IP information.

Cons:

- Cannot be used against every access point.
- The maximum xor bits is limited to the length of the packet you chopchop against. Although in theory you could obtain 1500 bytes of the xor stream, in practice, you rarely if ever see 1500 byte wireless packets.
- Much slower then the fragmentation attack

Usage Tips

Optimizing injection speeds

Optimizing injection speed is more art than science. First, try using the tools "as is". You can try using the "-x" parameter to vary the injection speed. Surprisingly, lowering this value can sometimes increase your overall rate.

You can try playing with the transmission rate. IE "iwconfig wlan0 rate 11M". Depending on the driver and how you started the card in monitor mode, it is typically 1 or 11MBit by default. If you are close enough set it up to a higher value, like 54M, this way you'll get more packets per second. If you are too far away and the packets don't travel that far, try to lowering it to (for example) 1M.

Usage Troubleshooting

These items apply to all modes of aireplay-ng.

aireplay-ng does not inject packets

Ensure you are using the correct monitor mode interface. "iwconfig" will show the wireless interfaces and their state. For the mac80211 drivers, the monitor mode interface is typically "mon0". For ieee80211 madwifi-ng drivers, it is typically "ath0". For other drivers, the interface name may vary.

For madwifi-ng, ensure there are no other VAPs running

Make sure there are no other VAPs running. There can be issues when creating a new VAP in monitor mode and there was an existing VAP in managed mode.

You should first stop ath0 then start wifi0:

```
airmon-ng stop ath0
airmon-ng start wifi0
```

or

```
wlanconfig ath0 destroy
wlanconfig ath create wlandev wifi0 wlanmode monitor
```

Aireplay-ng hangs with no output

You enter the command and the command appears to hang and there is no output.

This is typically caused by your wireless card being on a different channel then the access point. Another potential cause of this problem is when you are using an old version of firmware on prism2 chipset. Be sure you are running firmware 1.7.4 or above to resolve this. See