Advanced Networking Lab

Jakob Struye Thomas Hendriks

March 5, 2017

Lab 4

Wireless Security

In this lab, you will have a look at the impact of various security measures which are available for wireless networks. The main goal is to show what the effect is of a certain security method and what a possible attacker/sniffer can do on the protected network. In the first lab, you already performed some sniffing using the monitor mode. It should be clear that unless we take some security measures, it is quite easy to intercept and read any management information and data from open networks. It is even quite easy to perform denial of service (DoS) attacks on open networks as you will see at the end of this lab.

In the first part of the lab, you will investigate the more basic security features like hiding a network and Media Access Control (MAC) filtering. In the second part, you will have a look at encryption mechanisms and try to hack some of these. In the last part, we will introduce some basic attacks. These attacks are introduced to illustrate the mechanisms behind wireless networks and should only be used for this purpose.

4.1 Basic Security Measures

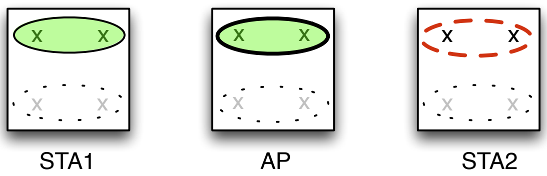

Figure 4.1: MAC filter setup

Exercise 1: Network hiding

A first step in preventing someone to use your wireless network is to hide it from being detected by a regular scan. Hiding a Service Set ID (SSID) ensures it will not show up in the list of available access points (APs) when a scan is performed. You will however show that the clever sniffer is still able to detect the network.

1. Set up your network as illustrated in figure 4.1, but do not set up the wireless network interface card (wnic) of station (STA)1 yet.
2. Start a trace on the monitor interface (`/mnt/L4-1-1.STA2.pcap` of STA2. While the trace is running, bring up the wnic from STA1. Make sure the station is associated with the AP before you stop your trace.
3. Indicate in which frames the SSID is visible. Give a packet ID of an example packet in your trace file for each.
   L4-1-1

   - Beacon frame (packet 5): broadcast by the AP (or distributed amongst stations in an IBSS network) contains the SSID, in this case, wmn-1-A.
   - Probe Request (packet 2528): the directed probe request contains the SSID of the network the station wants more information about. There are also broadcast probe requests which do not contain an SSID.
   - Probe Response (packet 2529): response from AP containing capability information, supported data rates, etc., sent after receiving a probe request. The SSID is listed under tagged parameters.
   - Association Request (packet 2694): part of the process where a station registers and connects to an AP. This first step contains the SSID with which the STA wants to connect and is sent to the corresponding AP.

4. The next step is to hide the SSID. This can be done adding the following line to the hostapd.conf file: `ignore_broadcast_ssid=1`. Bring the interface down on STA1, restart `hostapd` on the AP and repeat steps 1 and 2. Save your trace to `/mnt/L4-1-2.STA2.pcap`
5. Compare the two trace files and indicate how the SSID is hidden. In what way can a SSID still be detected?
   L4-1-2

The SSID itself can still be seen in following packet / frame types, despite SSID hiding:

- Probe Request (packet 455)
- Probe Response (packet 456)
- Association Request (packet 617)

The SSID was only removed from the broadcast beacon frames, the others are unchanged. Note that the beacon frames are still sent. In Wireshark's packet info field the SSID is shown as "Broadcast" but further inspection of the packet shows the SSID field is simply empty. With the parameter being "ignore_broadcast_ssid", it was to be expected only the beacon frames would no longer contain the SSID.

Exercise 2: MAC filtering

Continuing from the setup you used in the previous exercise, a possible intruder has still complete access to our network. In the next step, you will prevent him from joining the network by MAC filtering.

1. The management of the MAC filtering is also performed by modifying the `hostapd.conf` file at the AP. The entries needed for MAC filtering are:

   macaddr_acl= controls the behaviour of the mac filtering and expects an integer value from the following list:

   0 accept unless in deny list,

   1 deny unless in accept list,

   2 use external RADIUS server (accept/deny lists are searched first

   accept_mac_file= append the name of the file here that contains the list of MAC addresses to accept.

   deny_mac_file= append the name of the file here that contains the list of MAC addresses to deny.

2. Configure the AP in such a way that the MAC address of the `wlan0` interface of STA1 is blocked from the network. SSID broadcasting may be enabled again.

3. Now bring down the wnic at STA1 again, restart a capture session on STA2 and save it to `/mnt/L4-2-1.STA2.pcap` and bring the wnic back up at STA1 when this capture session is active. Compare the results from this capture file with the first one you made and discuss the differences between both. Illustrate with packet IDs.
   L4-2-1

In the first dump, when the station sent an authentication frame (packet 2666) the AP replied with an authentication frame (packet 2668) indicating the station was succesfully authenticated (status code 0: successful).
In the second case, with MAC filtering enabled and configured to not allow STA1's wlan0 to connect, the authentication frame(packet 464) sent by STA1 is met with an authentication frame(packet 466) with status code 1. This indicates an unspecified failure. STA1 does not attempt an association as authentication failed. Note that STA1 is not informed that authentication failed because it was blacklisted. There is no reason to let a potential attacker know why exactly they could not connect.

4. Of course, in a real world setup, blocking unwanted MAC addresses is not feasible. One would use the option with the `accept_mac_file` parameter, allowing only known MAC addresses to connect. How would you circumvent this security measure? If an attacker has joined the network, will this have an effect on legitimate stations on that network?
L4-2-2

An attacker can easily circumvent this security measure by spoofing their MAC-address to be among those listed in the accepted MAC addresses. The easiest way to find such an address is to monitor the network and obtain MAC addresses of currently connected devices.

In the case that the attacker uses a MAC address for which the genuine device is also connected, problems may arise in the network. MAC addresses are supposed to be unique within a LAN. The precise issues to arise depend on network configuration and the specific devices used. In a wired LAN one of the two devices would probably become inaccissble due to their entry being overwritten in the MAC table of the switch. This obviously cannot occur in a wireless LAN.

4.2 Encryption

4.2.1 WEP

The security techniques from the previous section will "prevent" a possible attacker from joining the network by hiding the network or preventing him access to the network.

However, it is still possible to read any communication over the network by using a wnic in monitor mode. Encryption mechanisms will counter this and obfuscate the network traffic for an eavesdropper. You will take a look at Wired Equivalent Privacy (WEP) here as this method is still sometimes being used although it is very vulnerable to attacks. Wi-Fi Protected Access (WPA)2 is far more resilient.

Exercise 3: WEP encryption

WEP was the first available encryption method, standardised together with the IEEE 802.11 standard. It was however quickly been proven insecure. In fact, as of 2004, WEP was declared deprecated by IEEE because of its flaws. In this exercise, you will first set up a WEP encrypted network to check what is exactly encrypted. In the next exercise you will crack the encryption key.

Because WEP is quite old, the tools you will use assume that IPv4 is used, as they will use Address Resolution Protocol (ARP) messages. Therefore, for this exercise, you will configure IPv4 addresses instead of IPv6 addresses.

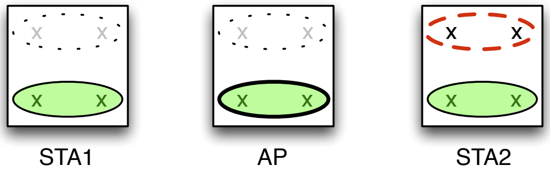

Figure 4.2: WEP setup

1. Start from the network setup as shown in figure 4.2. Disable any MAC filter that is still in place.

2. In order to avoid capturing needless IPv4 traffic, disable the `avahi-daemon` on all three nodes:
```
wmn:~# killall avahi-daemon
```

3. Create a tracefile on the monitor interface of STA2 (`/mnt/L4-3-1.STA2.cap`), containing the scan, authentication and association phase of STA1 and a 5 packet ping session between STA1 and STA2. Use `10.0.grID.1/24` for STA1 and `10.0.grID.2/24` for STA2.
```
STA:~# ip addr add 10.0.grID.X/24 broadcast 10.0.grID.255 dev wlan1
```

4. Make sure that STA2 is connected to the AP and that the wnic of STA1 is down before the start of the capture session. To ping, use `ping` instead of `ping6`. This trace will be our reference trace.

5. Activate the WEP encryption on the AP. You can choose a random key as long as it is either 5 or 13 characters long (”arandomwepkey” in this example):
   Change the `hostapd.conf` file, so that it includes the following lines:

   ```
   wep_key0="arandomwepkey"
   wep_default_key=0
   ```

   ! The WEP key must be ***exactly 5 or 13*** characters long!

6. Make sure the AP is running. Connect STA2 to the AP:

   ```
   STA2:~# iw dev wlan1 connect wmn-0-A key 0:arandomwepkey
   ```

7. Make sure the interface of STA1 is down and then start a new trace on the monitor interface of STA2. Save it to `/mnt/L4-3-1.STA2.wep.pcap`.

8. Perform the same test as in step 4, but be sure to add `key 0:arandomwepkey` when you connect STA1 to the network.

9. Compare both trace files. In the second trace file, you should no longer be able to see the `ping` session in the clear.

   - Which frame types are encrypted?
     L4-3-1

     In our case, the ICMP echo request (packet 1415) and reply (packet 1419) frames, the ARP frames (packet 1408) and ICMPv6 packets (Neighbor solication in packet 445, Multicast Listener Report Message in packet 434, Router Solication in packet 477) are encrypted.

   - Which part of a frame is encrypted and which part is still readable?
     L4-3-2

     WEP encrypted the Logical Link Control header and everything in Layer 3 and above. In for example packet 1408 only the IEEE 802.11 header is readable. The LLC header, IP header and ICMP header + data are all combined into one encrypted data field. Analogously, in the ARP request in packet 1408 the LLC header and ARP fields are encrypted.

   Refer to specific frames from the second trace file to illustrate your answers.

Exercise 4: Cracking the WEP keys

Without going into details about the flaws in WEP, the main problem arises from the fact that traffic keys are easily repeated on a busy network. A traffic key is the key actually used by the encryption algorithm RC4. As RC4 is a stream cipher, the same key should not be used twice. Therefore, the shared WEP key, which is configured in each AP and STA, is combined with a random initialization vector (IV). This IV is transmitted in plaintext between two wireless clients so the receiver can recreate the encryption key used for a specific packet, by combining the chosen (shared) network key with the received IV. As the IV has a length of only 3 bytes, the number of possibilities is relatively low and in busy networks, the same IVs are frequently reused, weakening the cryptographic strength of WEP. Furthermore, statistical relations between used keys and encrypted text can be exploited to find the used key.

Various methods have been developed to crack a WEP key, but you will just illustrate the mere ease with which a WEP encrypted network can be hacked using the aircrack tool [**?**]. This tool implements various approaches, but you will limit ourselves to Pychkine, Tews, Weinmann (PTW) method. The attack is based on the fact that the first 16 bytes of an ARP packet are fixed. This is also the reason we switched to IPv4 for this exercise...For more details read [**?**], specifically sections 1 and 5 for a general overview. Exploiting the relation between cleartext and the captured IVs makes it possible to quite easily determine the network key.

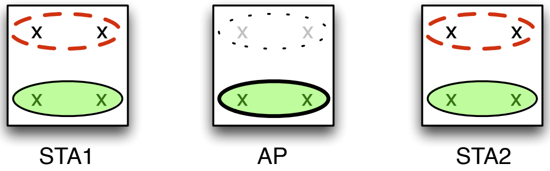

Figure 4.3: WEP cracking setup

1. Before you start to crack the configured WEP key, first have a second look at the traces from the previous exercise. Using Wireshark [**?**], you can decrypt an encrypted trace. Using the menu Edit→ Preferences → Protocols → IEEE 802.11 you can specify the key in HEX. To convert your key to HEX, use e.g. the following website: `http://www.dirtymonday.net/key_convert.html` For our example it will be `6172616E646F6D7765706B6579`.
   Make sure you also check the ‘decrypt packets’ box. Select an ICMP request in both trace files (unencrypted and encrypted, but decrypted by Wireshark) and identify the extra information added to a frame in order to encrypt it. Also indicate where in the frame this information is added.
   L4-4-1

The difference in size between the encrypted packet with ID 1415 in traces/L4-3-1.STA2.wep.pcap and the regular (non-WEP) packet with ID 1077 in traces/L4-3-1.STA2.pcap is 8 bytes (154 vs 146). The extra 8 bytes are the WEP parameters in the IEEE 802.11 Data header. These WEP parameters are an initialization vector (3 bytes), a key index (1 byte) and a WEP Integrity Check Value (4 bytes). This ICV is inserted between the encrypted data and the frame check sequence.

---

2. Start from the WEP encrypted network you created in the previous exercise.

3. Add an additional monitor interface on STA1, resulting in the network scheme shown in figure 4.3. The monitor interface on STA2 will be used to perform the actual attack. ARP packets will be injected into the network in order to trigger ARP responses which will contain new IVs with every response . These packets will be captured and stored in files. In parallel with this capturing of new ARP responses and the associated IVs, we can start the process of analyzing these packages and searching for the key. The monitor interface on STA1 will be used to monitor the channel so we can afterwards take a look at what happened.

4. Start a packet capture to `/mnt/L4-4-2.STA1.pcap` on the monitor interface of STA1.

5. Start the IV capture process at STA2. This command will store its captured frames in files `output-xy.cap` and `output-xy.txt`. Make sure to run this command with your active directory set to the `/mnt` folder, in which a remote directory is mounted.
   ```
   STA2:~# airodump-ng --band a --channel x --bssid <MAC addr AP> -w output
   wlan0
   ```
   This should give you an overview screen where at least the incoming beacon count is rising. If no activity is shown, bring the interface down and then bring it up again and retry. This command will store its captured frames in files `output-xy.cap` and `output-xy.txt`. Make sure to run this command in `/mnt`, where a remote location is mounted.

6. On a new terminal, perform a fake authentication to the AP from the attacking machine (STA2). This ensures the frames we will be injecting for `wlan1` will be accepted by the AP:
   ```
   STA2:~# aireplay-ng --fakeauth 0 -e wmn-grID-A -a <MAC addr AP> -h <MAC
   addr wlan0> wlan0.
   ```

7. It is now time to actively inject packets into the wireless network. This is done by the following command:
   ```
   STA2:~# aireplay-ng --arpreplay -b <MAC addr AP>
   -h <MAC addr wlan0> -x 1024 -o 512 wlan0
   ```

This command will monitor incoming packets and when an ARP request passes, it will re-inject it at a rate of several 100 frames/s. You should be able to see this in the output from `airodump-ng`.

Due to a bug in the driver, the packet injection rate will not be high enough. You should run 10 of the above `aireplay-ng` commands in parallel to generate enough packets. You can do so easily by editing the `wepcrack.sh` script found on the nodes. Insert the correct MAC addresses in that script and run it. To stop the replay attack, simply type `killall aireplay-ng`.

8. In order to get an ARP request, just start a ping from STA1 to AP. This should trigger an ARP request. If the ARP tables already contained the IP addresses from both hosts (check it with `STA1:~# ip neigh`), remove the entry using `ip neigh flush dev wlan1`.

9. To crack the key, the tool `aircrack-ng` can be used. This tool processes the files generated by `airodump-ng`. The needed number of packets to crack the key depends on the bit length of the key. In general, you need at least 20,000 packets for a 64-bit key and between 40,000 and 85,000 for a 128-bit key. To start the analysis, run the following command in the folder containing the output from `airodump-ng`:
`STA2:~# aircrack-ng -b <MAC addr AP> output*.cap -q`

10. If you captured enough packets, after a while, the `aircrack-ng` tool should provide you with the WEP key used in your network. Copy its output here:
L4-4-2

```
STA2:/mnt# aircrack-ng -b 00:0e:8e:20:83:df output-02.cap -q
KEY FOUND! [ 61:72:61:6E:64:6F:6D:77:65:70:6B:65:79 ] (ASCII:
    arandomwepkey )
  Decrypted correctly: 100%
```

11. Using the following `tshark` command on a lab PC, you can get a list of IVs used during your session.
`tshark -r <trace file> -T fields -e wlan.wep.iv`

12. How many IVs did you capture? And how many of them occured at least twice?
L4-4-3

We captured a total of 1109201 IV including duplicates, or 735095 without duplicates. 7563 occured more than once. The most common one was 0x00400000 with 3194 occurrences.

13. When handing in the report, you must include the `output-*.csv` and `*.netxml` files. `output-*.cap` is not required, as it will be quite large. Limit the trace file of STA1 to only a few (e.g. a hundred) of packets that show the ARP injection in progress.

4.2.2 WPA2

As you have demonstrated in the previous exercise, WEP encryption is easily circumvented. You will now take a look at WPA2 encryption. You will no try to crack it, as WPA2 uses the Counter Cipher Mode with Block Chaining Message Authentication Code Protocol (CCMP) algorithm with Advanced Encryption Standard (AES) encryption, which is extremely resilient to attacks. To give you an idea, the best known method until now to crack a 128 bit AES key, which is used by WPA2, still takes at least $2^{126.1}$ computations [**?**].

A security flaw concerning Wi-Fi Protected Setup (WPS) has been discovered that makes APs running WPS vulnerable to a relatively easy attack (which would still take far too much time for this lab exercise) [**?**]. This, of course, does not hold true for APs that do not have WPS enabled. If you want to read up on how WPA2 works and on the evolution in security from WEP over WPA to WPA2, you should read [**?**].

Exercise 5: WPA2

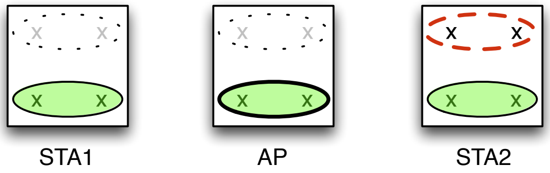

Figure 4.4: WPA setup

1. Start from the WPA2 enabled network setup as seen in figure 4.4.

2. On the AP, to enable WPA2, make sure that the line `wpa=0` in the `hostapd.conf` file is changed to `wpa=2`, and add the following lines to `hostapd.conf`:

```
wpa_passphrase=aRandomPassphrase
wpa_key_mgmt=WPA-PSK
rsn_pairwise=CCMP
```

Use your own passphrase if you like. Unlike WEP, it is not restricted to a certain amount of characters.

3. Set the IP addresses of `wlan1` on STA1 and STA2 to `fc00:grID::1/64` and `fc00:grID::2/64`, respectively.

4. Connect STA2 to the network. In order to do so, modify the `wpa_supplicant.conf` file so that it includes the correct ESSID and passhprase. Afterwards, run:

   ```
   STA2:~# wpa_supplicant -B -c/root/wpa_supplicant.conf -iwlan1
   ```

   You might get an error like `ioctl[SIOCSIWENCODEEXT]: Invalid argument`. You can safely ignore this error.

5. Make sure the monitor interface on STA2 is configured and start a trace. Save it in `/mnt/L4-5-1.STA2.pcap`.

6. Now bring the interface on STA1 up an do `ping6 -c 5` to STA2.

7. Which frame types are encrypted? Give examples.
   L4-5-1

   The encrypted frame types are largely the same as with WEP. We observed Multicast Listener Report Messages (packet 207), Neighbor Solicitations (packet 222), Router Solicitations (packet 256) and of course the ping requests (packet 348) and replies (packet 354). The only type we did observe with WEP but not here is ARP. This is unsurprising is ARP is not used with IPv6.

   ---

8. Which part of a frame is encrypted and which part is still readable?
   L4-5-2

   This is exactly as with WEP. The LLC header and Layers 3 and above are encrypted while the 802.11 header is not.

   ---

9. Compare this trace file with the one made in exercise 3. Compare the authentication/association phase of STA1. What are the similarities and what are the

differences? Refer to packet IDs!

L4-5-3

There are no significant differences between the authentication phases (packets 409 and 411 for WEP, 176 and 178 for WPA2). For the assocation phase (packets 424 and and 436 for WEP, 189 and 191 for WPA2), an additional 22 byte RSN (Robust Security Network) Information tag is present in WPA2's Request. This identifies the requested protocols. Other than this there are no significant differences.

The main difference is that this is followed with 4 EAPOL (Extensible Authentication Protocol over LAN) packets which perform the actual WPA2 authentication. This is known as EAPOL's 4-way handshake. Station and AP exchange some randomly generated values which are used in combination with the passphrase to generate the Pairwise Transient Key (PTK) they will use for encryption. Someone overhearing this cannot recreate the PTK unless they also know the passphrase. A step like this is completely absent in WEP (with Open System authentication): the AP assumes the station knows the passphrase, knowing the station will not be able to encrypt/decrypt communication without it.

10. Now, compare an encrypted frame in both traces. Indicate where they differ.

L4-5-4

We compare two encrypted Router Solicitation packets (packet 596 in L4-3-1.STA2.wep.pcap, packet 256 in L4-5-1.STA2.pcap). Looking at the encrypted packets, we notice that while the WEP packet has 8 bytes of WEP parameters, the WPA2 packet has 8 bytes of CCMP parameters. CCMP is the default encryption method in WPA2. This consists of an 8 byte extended IV. All 8 bytes are at the end of the 802.11 header, as opposed to the WEP parameters where 4 bytes were between the encrypted data and the FCS. We furthermore notice the encrypted data is 8 bytes larger for WPA2. After decrypting the WPA2 packets (wpa-pwd aRandomPassphrase:wmn-1-A) the 8 additional bytes are gone. We suspect these 8 bytes were a side-effect (e.g. padding) of the encryption algorithm.

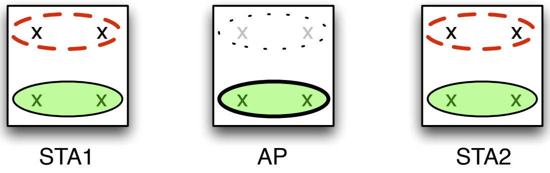

Figure 4.5: DoS setup

4.3 Denial of Service Attacks

In the previous section, you have gained access to a WEP encrypted network by actively injecting packets and cracking the key. This kind of attack makes it possible that you, the attacker, can afterwards associate with the network and sniff all traffic on that network. Alternatively, you can also use the network yourself as if you were a normal user.

Within this section, you will illustrate a DoS attack that shows the vulnerability of a wireless network. You will not gain access to the network as such, but you will disturb normal network operation. Tools to do so are freely available and current encryption techniques cannot prevent these attacks as they make use of management frame injection. As you have seen, management frames are not encrypted and thus can be easily injected into a wireless network.

Exercise 6: Disassociation Attack

The principle behind a disassociation attack is quite easy. A disassociation frame is sent by either a STA or AP to signal the disconnection from the wireless network. For instance, it is sent by a STA just before you disable an associated wnic. Although the sending of a disassociation message is not mandatory by the standard, at reception of such frame, a client should consider its connection lost and an AP will remove a STA from the list of connected stations. Injecting a malicious disassociation frame for a wireless node will thus force this node to scan for and associate again to its AP, resulting in a connectionless period. If the disassociation frame is sent periodically, this can render the wireless network useless to the attacked STA.

1. Start from the network scheme as illustrated in figure 4.5.
2. Be sure to have configured your network with WPA2 encryption.
3. Configure the IP addresses of STA1 and STA2 to `fc00:grID::1/64` and `fc00:grID::2/64` respectively.
4. On the monitor interface of STA1, start a capture session and save it to `/mnt/L4-6-1.STA1.pcap`.

5. Start a `ping6` from STA1 to STA2.

6. On STA2, use the following command to send a disassociation to STA1:

```
STA2:~# aireplay-ng --deauth 1 wlan0 -a <MAC addr AP> -c <MAC addr STA1>
```

7. What is the result for the ping?

   L4-6-1

   The disassociation/deauthentication attack caused the ping session to be interrupted. After 5 seconds, the station was able to reestablish connection with the AP and resume the ping traffic. Repeating the attack some time later again caused a 5 second interruption in ping traffic.

   The main thing to note for the ping session is that it was interrupted for a short time but when the attack ceased, the ping session resumed quickly after the STA reassociated with the AP.

8. Open the trace file and explain using this trace what exactly happened.

   L4-6-2

   Wireshark could not decrypt the earliest ping packets. It is possible that the multiple EAPOL handshakes between STA1 and AP due to the attack confused Wirehsark. The others can again be decrypted with wpa-pwd aRandomPassphrase:wmn-1-A

   After seeing the first set of ping messages (not decrypted by Wireshark), we see 128 deauthentication messages in the dump file (traces/L4-6-1.STA1.pcap). Out of those 128 packets, 64 are sent to the STA by what appears to be AP (e.g. 203), and the other 64 are sent in the opposite direction (e.g. 205). Both of these are actually sent by the attacker, who is spoofing the source MAC address to be equal to STA1's or AP's wlan1 interface. These are the MAC addresses we added to the command. By sending deauth packets to both STA and AP, both are aware of the deauthentication, preparing both for the following reauthentication. As the purpose of a deauthentication attack is often to capture the following handshake, this is useful. By sending many deauth packets, the attacker increases the likelihood of at least one being received even when reception is poor. The attack works as soon as one is received. One pair of deauthorization packets is sent every 50ms. There is a 20ms interval between the AP to STA and the STA to AP packet. This means the attacker sends spoofed deauthorizations for about 3.15 seconds.

We see the STA ceases the pings as soon as a deauthentication is received, as it thinks the AP no longer wants to receive any traffic from it. A reauthentication (packet 491) + reassociation (packet 502) start roughly 0.4 seconds after the attack ceased. Once the full authentication + assocation process is completed, pings resume at packet 522. Afterwards we started another two deauthentication attacks with the same effect.

Acronyms

AES Advanced Encryption Standard

AP access point

ARP Address Resolution Protocol

CCMP Counter Cipher Mode with Block Chaining Message Authentication Code Protocol

DoS denial of service

ESSID extended Service Set ID

grID group ID

IV initialization vector

MAC Media Access Control

nic network interface card

PTW Pychkine, Tews, Weinmann

SSID Service Set ID

STA station

WEP Wired Equivalent Privacy

wmn wireless mesh node

wnic wireless network interface card

WPA Wi-Fi Protected Access

WPS Wi-Fi Protected Setup