Wireless Security Lab - Air Conditioning

Philipp Breinlinger
Free University of Berlin
Institute of Computer Science
E-Mail: breinlin@zedat.fu-berlin.de

Gerrit Gruben
Free University of Berlin
Institute of Computer Science
Institute of Mathematics
E-Mail: Gerrit.Gruben@fu-berlin.de

July 31, 2009

Contents

Introduction

This work has been made in the context of the practical hands-on part of the course IT-Security by Prof. Dr. Volker Roth at the Free University of Berlin in the Summer Semester 2009. The document describes a lab, which can be practically worked through. The name of the lab group is 'Air Conditioning'.

In this lab various attacks on wireless networks are first explained and worked through on the base of the Linux distribution BackTrack, which includes the necessary tools such as AirCrack, Kizmet and Wireshark. The target of the lab is to give a solid understanding of the basic security issues of wireless networks and the related technology.

In the first section, the theoretical background of the lab's topic is being covered. Afterwards, in the second section, practical preparations necessary for our attacks will be made. Once these preparations are made, a basic passive sniffing attack is explained in the third section. As a further attack on wireless networks, an attack on WEP is presented. The fifth section closes the lab with a few questions regarding the topics covered in this document.

1 Background

In this lab we are working on wireless networks as defined in the IEEE 802.11 standard (see [4]). These are broadcast networks, which makes eavesdropping attacks easy. This lab is going to use this weakness and performs several eavesdropping attacks on wireless networks. Furthermore the 802.11 standard defines Wired Equivalent Privacy (WEP), which defines an algorithm for confidentality and integrity. For integrity a CRC-32 Checksum is used.

For confidentally the RC4 stream cipher is used with a 24 bit initialization vector (IV) and a key length of 40 or 104 bits (so it makes 64 or 128 bits in whole). However the use of RC4 in WEP has been shown to be insecure by a wrong usage of the stream cipher. The lack of a key-exchange protocol is another issue, so once the key has been broken there is no possibility to perform a re-keying. The standard attacks on WEP are listed in [6] and these are implemented by the aircrack-ng tool (see [1]).

2 Practical Preparations

For this lab we will use two laptops with W-LAN access and a router. One of the laptops takes the role of the client and the other is the adversary executing the attack. For most of the described attacks the client just needs to be connected to the router via W-LAN generating traffic. The next subsection describes more in detail the preparations for the adversary. It is best to take notes of the MAC-Adresses involved as well as of the SSID and used channel of the router as they are frequently required later. For example:

```
Router:   00-1D-19-D2-3A-86
ESSID: CONEJO, Channel: 4
Client:    00-1B-9E-4E-2A-B6
Adversary: 00-13-49-22-84-89
```

2.1 BackTrack

To run the lab the Linux distribution BackTrack ([2]) is recommended for the adversary. To obtain an Live-CD image or an image to boot from USB-Stick go to `http://remote-exploit.org/backtrack/_download.html`. Burn/Install it and reboot your system from the prepared device.

After the log-in has been done use the command 'startx' to start a GUI environment. Once finished, check to be sure that you have the following applications ready: AirCrack-suite, Kizmet and Wireshark.

2.2 Setting up the environment

In order to allow us to passively monitor traffic, the W-LAN device first needs to be set in monitor mode. To do this, open a shell window and type 'ifconfig'

```
wlan1     Link encap:Ethernet  Hardware Adresse 00:13:49:22:84:89
          UP BROADCAST MULTICAST  MTU:1500  Metrik:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          Kollisionen:0 Sendewarteschlangenlänge:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Search for the proper W-LAN device - in our case it is 'wlan1' (from now on 'wlan1' should be seen as an arbitrary string and if another device is used, just replace that one with the proper device). Usually 'wlan0' is the build-in W-LAN card and in our case 'wlan1' is a W-LAN stick, that we are going to use for this lab. Now run 'airmon-ng start wlan1' to put the device into monitoring mode. A new virtual network interface 'mon0' is being created and will be used for monitoring purposes.

To start the network manager type 'wicd' in the shell, followed by 'wicd-client'. This will also run a DHCP client and thus will send packets to the outside. Basically this is the reason, why the network manager is not started by default: some people want to stay quiet.

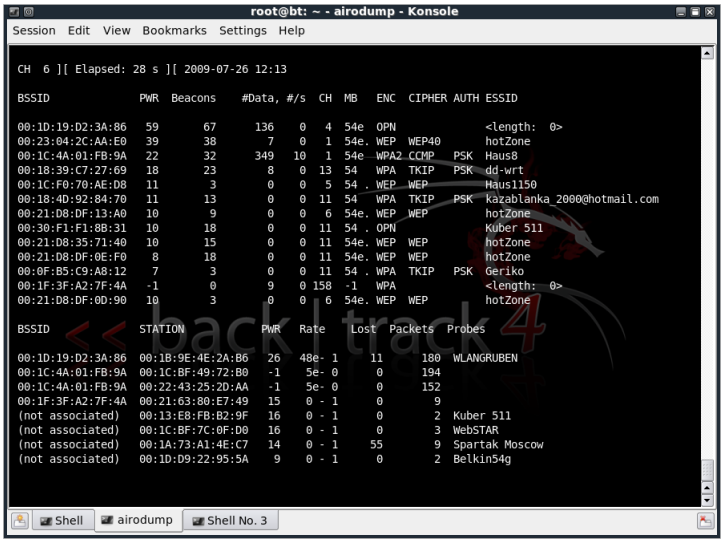
3 Sniffing-Attack

In this section we describe how we can perform an eavesdropping attack on unprotected[1] W-LAN networks, which may be cloaked (that means that the SSID is hidden. Set up the router so that it runs with a hidden SSID and without encryption. First we are going to monitor all traffic occuring to find the respective network we are going to attack.

3.1 Monitoring all wireless traffic

We are now going to monitor all the occuring traffic, we can listen to, to find informations about the present wireless networks. Type 'airodump-ng mon0' into the shell. This is best being done in a new shell window so it can run in the background during the further progress of the lab. After a while all the wireless networks in range together with their used encryption and SSIDs will be shown. When a SSID is unknown so far, '¡length: 0¿' is displayed in the respective column. Also you see the related MAC-Adresses of the clients with the access points they are attached to.

[1] That means no encryption



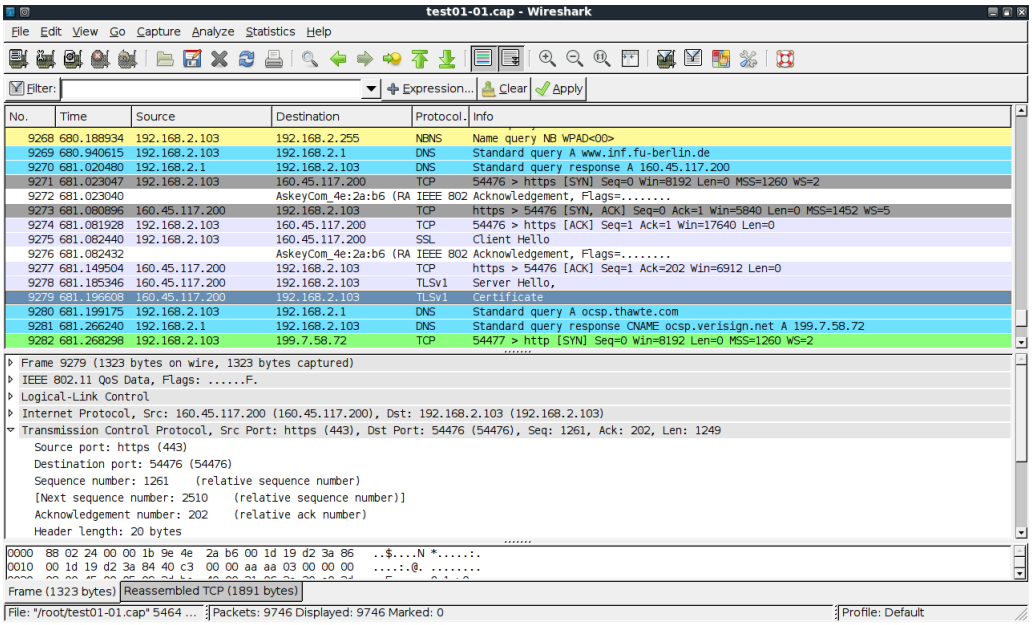
Write the related MAC-Adresses, Channels and SSIDs down to perform further attacks. Note that the hidden SSIDs of a wireless network is shown, once a client attaches to the access point. To speed up the SSID discovery you can let the client reconnect to the router.

Now we describe how the adversary can perform an eavesdrop attack on the unencrypted communication between the client and the router, where both MAC-Adresses and the SSID of the wireless network are being known. Type

```
airodump-ng -w <filename> --channel <channel_of_router>
 --bssid <macid\_of\_router> mon0
```

into the shell. Now only the communication on the specified channel occuring on the specified BSSID is being monitored, instead of every communication in range. The captured traffic is written to the file given as parameter to the -w option and are now ready to be analyzed in binary format. However to get a more accessible look at the content, we use Wireshark to display our captured traffic. The binary format is going to help us when scanning for weak IVs later, when we perform the attack on WEP. Start Wireshark by typing 'wireshark' in the shell and now you can simply open the file via 'File→Open' in the menu. The captured data is visualized in Wireshark and is available for you to analyze.

To directly gather packets with Wireshark start the application and click on the icon of the leftmost side 'List the available capture interfaces...'. You can start the capture by selecting the appropriate capturing interface 'mon0'. Note that this shows all the traffic in range. To filter the traffic to the communication between the client and the router you can use filter options to filter by the respective source MAC-Adresses and/or protocol.



3.2 MAC-Spoofing

In the further section all the MAC-Adresses of the clients have been obtained. Some routers deploy a MAC-Adress filter, that means that it keeps a list of MAC-Adresses and a client is only allowed to connect iff it's MAC-Adress is in the list. To change the MAC-Address of one of your network interfaces (e.g. 'wlan1') type in

```
ifconfig wlan1 down
ifconfig wlan1 hw ether <mac_adress_to_set>
ifconfig wlan1 up
```

So MAC-Spoofing consists out of monitoring the traffic, to obtain a valid MAC-Adress (of a client associated with the access point) as discussed in the section before, and setting this MAC-Adress as your own.

4 Attack on WEP

In this section we describe on how to obtain the key for a WEP protected wireless network. Set the router to WEP encryption and write down the password. Leave the systems of the client and the adversary unchanged.

At first we start by gathering data packets from the client to the router. We are especially interested in the so-called weak IVs (as described in the first chapter). Type

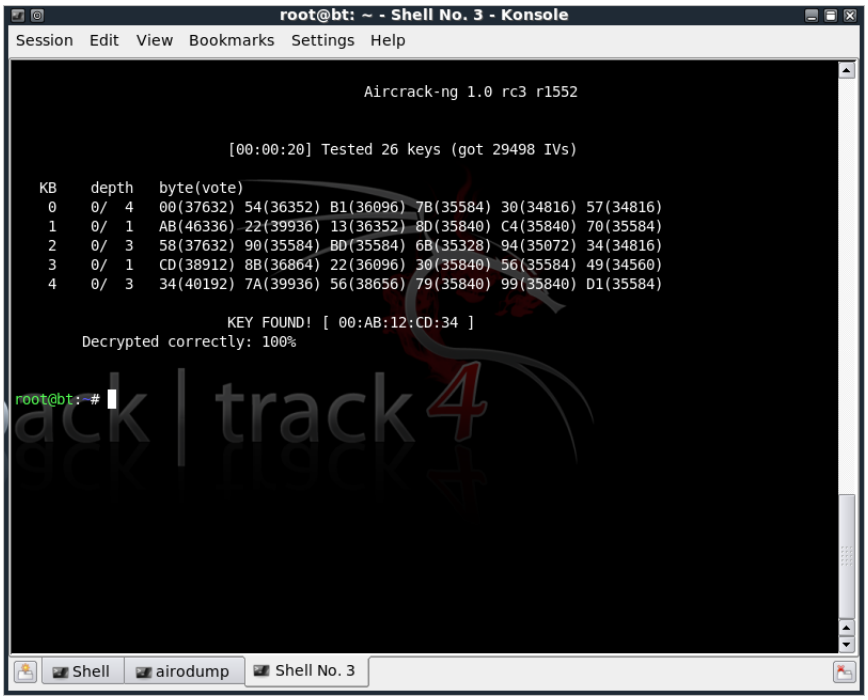
```
airodump -ng -w <filename> -channel <chan_number>
--bssid <mac_addr_of_router> mon0
```

in the shell as we did before in the eavesdropping attack. On the base of this file, which is updated regularly while airodump is running, we are now able to

recalculate the WEP key. Let the client create some traffic by watching videos or download a big file. Once you have obtained about 15-20k packets (look at the #Data column in airodump) type

```
aircrack-ng <filename_of_the_log_file>
```

into the shell to start the cracking process. When it was successful after while a similar screen to the following should be shown:



If not, you may need to gather more packets. Two ways to force the router to produce more traffic are described in the following section.

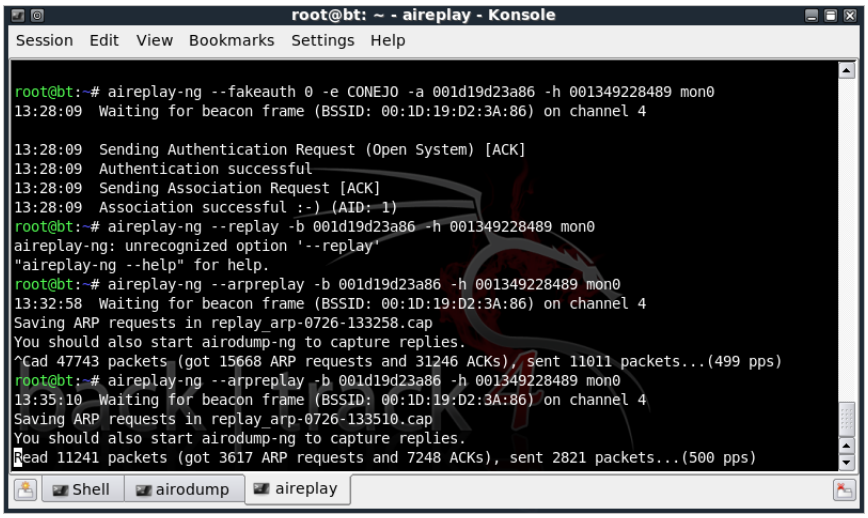
4.1 Active Attacks

In this section two so-called 'active attacks' - the deauthentification and the fake authentification - are described to actively generate more traffic for the key recalculation. Note that the attack mentioned before is passive, since the packets are gathered passively.

The fake authentification attack: The idea of this attack is that the adversary injects ARP-Requests into the router to generate valuable traffic. To make the router replying to these requests, the adversary has to perform a (fake) authentification on the router, otherwise the reinjected packets will be dropped by the router. To achieve fake authentication type

```
aireplay-ng --fakeauth <delay := 0> -e <ssid_of_router>
-a <mac_addr_router> -h <mac_addr_adversary> mon0
```

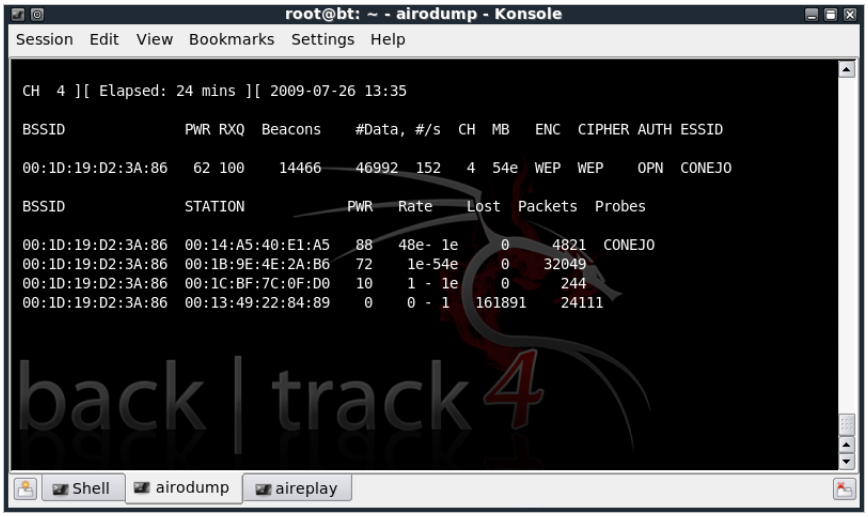
When this was successful it should look like this:



Now the adversary is able to inject packets into the target network. With the following command, the adversary constantly reinjects captured ARP-Requests into the network, therefore forcing the router to generate encrypted response packets

```
aireplay-ng --arpreplay -b <mac_addr_router> -h <mac_addr_adversary> mon0
```

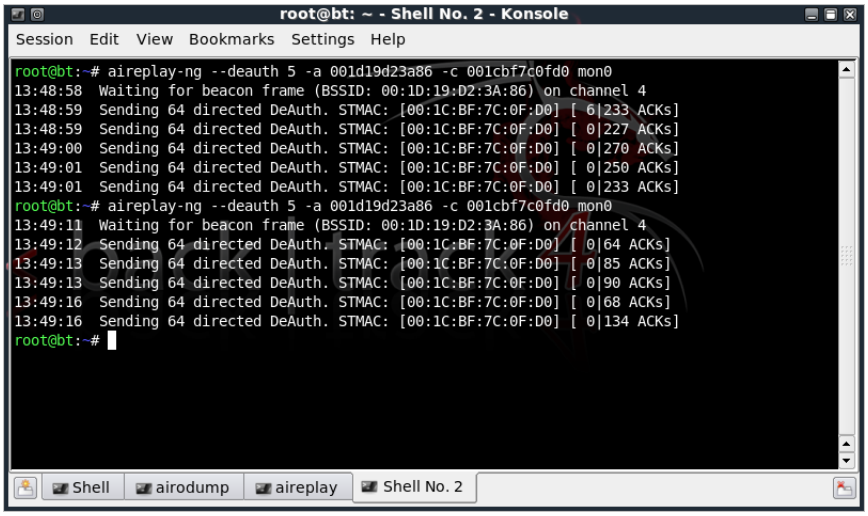
Now the injection takes place while airodump running in the background should gather data packets at a higher rate (check screenshot below). We call this ARP replay attack.



Deauthentification attack: A second possibility to raise the rate of packets is to deauthenticate a currently attached client from the target network. This way we force the Access Point to generate traffic while reauthenticating the client. This attack basically is an extension to the ARP replay attack, useful when we are unable to capture ARP Packets. While aireplay is running in arpreplay mode the adversary should type

```
aireplay-ng --deauth <number_of_deauths>
-a <mac_addr_router> -c <mac_addr_client> mon0
```

to raise the number of gathered packets. When the attack was successfull, it should look like this:



5 Conclusions and Outlooks

In this paper we have described attacks on wireless networks following the 802.11 IEEE standard. Some technologies have shown to be insecure, it is left as an exercise to do a reflection of the conclusions of these attacks.

As further improvements we want to name the following attacks: The creation of fake Accesspoints with the execution of man-in-the-middle attacks, a WPA-Dictionary attack or a physical signal distortion attack (by the means of jammers). Also a discussion of defensive measures could be added. Furthermore we only looked at the standard wireless network topology, the situation is quite differently in both ad-hoc networks as well as wireless sensor networks.

6 Questions

You can leave one Question out, but please deliver a detailed answer for at least one of the questions.

- Q1: What are the major disadvantages of leaving the traffic unencrypted? In which cases could it be useful to leave the traffic unencrypted? Is WEP-encryption any better? Can eavesdroppers in wireless networks be spotted? If so explain the method and it's circumstances.
- Q2: How many interesting packets did it take to calculate a WEP-key? Try to gather some statistics on how many packets are needed to break a key and try to derive conclusions (10k packets are needed to calculate the key with a probability of 20%). Plot the number of packets against the

empirical probability you have derived. Can you find a function, which approximates this distribution well? Is it polynomial, logarithmic or exponential in growth for 'small' packet numbers? If you have found a function, how many packets one can estimate to need to have a probability of $0.1, 0.2, ..., 1$? Give an estimation on how much time[2] an adversary needs to retrieve the key with a probability of 0.5.

- Q3: Explain why AirCrack delivers one of the most devastating attacks against 802.11 and some possible ways to correct it. List what the parts of aircrack-ng are and what they are for.
- Q4: Which technologies have been shown to be insecure? What are the basic, fundamental problems which caused these insecurities? Are there any solutions to fix these problems? Are there new technologies, which solve this problems? (Research for new protocols and the solutions of router producers)

[2]Figure out the rate yourself. However an argumentation of why your values are chosen shouldn't be left out

References

[1] *Aircrack site*. http://www.aircrack-ng.org

[2] *BackTrack Linux-distribution*. http://www.remote-exploit.org/backtrack.html

[3] *Wardriving Board*. http://www.wardriving-forum.de/wiki. – [German]

[4] *Wikipedia::IEEE 802.11*. \url{http://en.wikipedia.org/wiki/IEEE_802.11}. – [Online; accessed at 30/07/2009]

[5] *Wikipedia::WEP*. http://en.wikipedia.org/wiki/WEP. – [Online; accessed at 30/07/2009]

[6] Theoretische, Fachgebiet ; Term, Summer ; Informatik, Fachbereich ; Darmstadt, Tu ; Tews, Erik ; Prof, Supervisor ; Dr, Dr. ; Buchmann, Johannes: *Attacks on the WEP protocol*. 2007