Packet analysis on encrypted wireless networks

Seminar Report, *Problem Based Learning*

Patrick Ziegler (3750096)

March 20, 2017

Capturing network traffic with an interface in monitor mode can be useful for analysis and debugging. Software such as Wireshark and `tcpdump` can be used for this purpose. How capturing and analysis can be carried out on encrypted wireless networks is discussed in this report. A short introduction to encryption with WPA2-PSK is followed by instructions on how to set up Wireshark to automatically decrypt captured packets.

Contents

Supervisors Prof. Dr.-Ing. Dr. h.c. Frank H. P. Fitzek,
M.Sc. Juan Alberto Cabrera Guerrero

Dresden University of Technology,
Deutsche Telekom Chair of Communication Networks

1 Network Device Configuration

On modern Linux hosts, network interfaces are an entirely logical entity that can be attached to physical devices. Such Interfaces support several modes of operation whereof *managed mode* is the standard for communication with wireless networks.

In managed mode, all packets that are not directed to the interfaces MAC address are dropped and the remaining traffic is automatically decrypted if connected to an encrypted wireless network.

Therefore the title of this work *Packet analysis on encrypted wireless networks* only makes sense when packets are captured, that are not destined for the local interface and thus cannot be decrypted per se. This can be achieved with an interface in *monitor mode* as described in the next section.

Not all network devices support monitor mode. This can be checked with `iw` ***phy**** `info` where ***phy**** is the device name of one of the interfaces found with `iw dev`. With this, a list of supported modes of operation is printed out. If this is not satisfactory, listing 1 shows how the driver in use can be investigated. When the driver name is known, a quick internet research can reveal whether monitor mode is supported or not.

```
ip link show
sudo ethtool -i wlp7s0
```

Listing 1: Investigating the driver name with `ethtool`

1.1 Monitor Mode

The basic steps to set up an interface in monitor mode are described in [Knl15]. As mentioned above, network interfaces are fully virtual. It is therefore possible to add another interface in monitor mode to a network device even if an interface in managed mode is already existing and connected to a network. The required steps are shown in listing 2.

```
sudo iw dev
sudo iw phy0 interface add mon0 type monitor
sudo ip link set mon0 up
```

Listing 2: Adding a network interface in monitor mode

With this, the monitoring interface will listen at the same channel as the managed mode interface attached to the device. Furthermore, it will receive decrypted[1] packets as they were directed to the local network interface. When capturing on this interface, no packets will not be broken down to their included protocols as Wireshark will see encryption information in the WiFi header and therefore assume an encrypted packet, even if it was decrypted by the driver automatically. This explanation only hold if another interface in managed mode on the same device has an established connection the the wireless network what is not to be confused with the supplementary decryption done by Wireshark in section 3.1.

Switching the channel of an interface in monitor mode only works if no other interfaces are attached to the device. Listing 3 shows how this can be done with removing the interface in managed mode before switching frequencies.

```
sudo iw phy1 interface add mon1 type monitor
sudo iw dev wlp7s0 del
sudo ip link set mon1 up
sudo iw dev mon1 set freq 2432
```

Listing 3: Replacing interfaces and switching channels

Having set up an interface in monitor mode, it is now possible to sniff network traffic that is not addressed to the local device. Most of the traffic will be encrypted which is the reason, we only see 802.11 headers and some encrypted payload when using Wireshark to analyse what is going on in the network.

1.2 Promiscuous Mode

In most literature, it is stated that it would be necessary to *put the device to promiscuous mode*. The difference between those is that in promiscuous mode only packets belonging to the network currently connected to are received (which includes packets that are not directed to the users device) and also packets belonging to other networks on the same channel are received in monitor mode. For the drivers used in this work, there is no such thing as promiscuous mode.

[1] it is assumed, those packets are decrypted on driver level

2 Encryption in Wireless Networks

There are plenty of encryption techniques for wireless networks but WPA2-PSK is the most used technology for private wireless networks. Most corporate networks use more complicated encryption that is not as easy to crack.

2.1 WPA2-PSK

The key to decrypt WPA2-PSK packets is the authentication process carried out every time a new user wants to join the network. An overview is presented in figure 1.

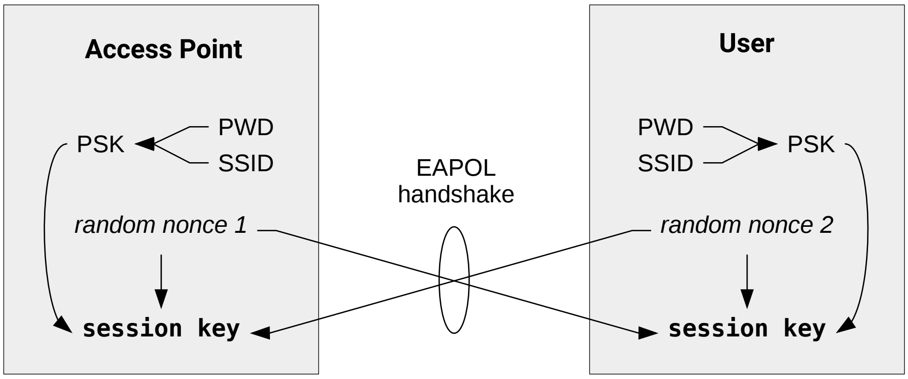


Figure 1: Authentification handshake (EAPOL)

In WPA2-PSK every connection is symmetrically encrypted with a session key. In order not to send any keys over the channel, both access point and user calculate a random number (also called *nonce*) and exchange these in the following EAPOL[2] handshake.

A pre-shared key (PSK) is generated out of the service set identifier (SSID) and the password (PWD) set in the access points user interface. This is done with an algorithm called password-based key derivation function (PBKDF2). The reason for this is to make it more difficult to crack the original password by brute-force searching with some rainbow table found in the internet.

The session key is then derived from both random nonces and the PSK which leads to the same session key in access point and user. If the traffic of other network users is to be analysed and the password (PWD) is known, only the EAPOL handshake has to be captured in order to recompute the session key and decrypt captured packets. The SSID is always assumed to be known because it is regularly broadcasted in beacon frames.

More details on WiFi security can be found in [Leh05].

[2] *Extensible Authentication Protocol over Lan* (EAP over Lan)

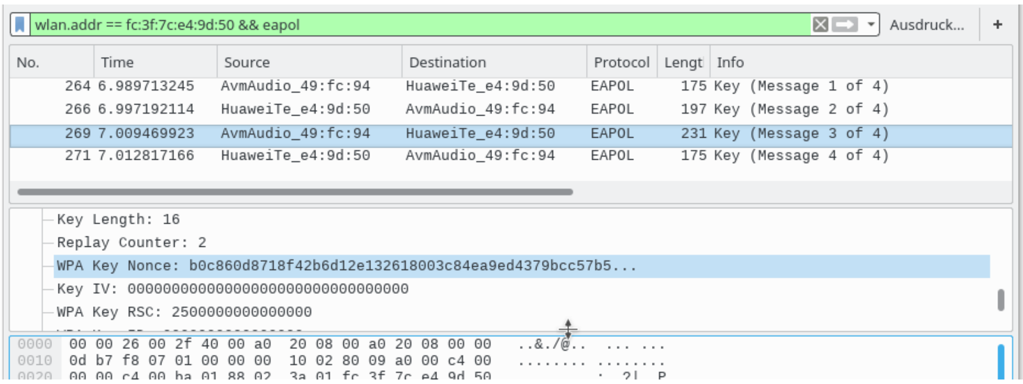
3 Decryption of Captured Traffic

3.1 Wireshark

If the background of packet sniffing and encryption methods is once understood, the automatic decryption of network traffic captured with an interface in monitor mode is very easy to set up. The necessary configuration in Wireshark is done in `Preferences > Protocols > IEEE 802.11`. The first thing is to check *assueme packets have FCS*. The standard behaviour of most drivers is to append the checksum (FCS) at the end of a frame. When capturing in monitor mode, the frame check sequence (FCS) is not stripped and therefore has to be assumed on the end of the captured frames.

The automatic decryption is activated with checking *enable decryption*. It is further necessary to enter either PWD or PSK in the menu opening after a clik on *edit*. Passwords are to be entered as `<PWD>:<SSID>` with the key *wpa-pwd*.

It is then necessary to capture an EAPOL handshake as shown in figure 2. After Wireshark has seen the random nonces inside the handshake, all following traffic is automatically decrypted and can therefore be analysed.



wlan.addr == fc:3f:7c:e4:9d:50 && eapol

No.	Time	Source	Destination	Protocol	Lengt	Info
264	6.989713245	AvmAudio_49:fc:94	HuaweiTe_e4:9d:50	EAPOL	175	Key (Message 1 of 4)
266	6.997192114	HuaweiTe_e4:9d:50	AvmAudio_49:fc:94	EAPOL	197	Key (Message 2 of 4)
269	7.009469923	AvmAudio_49:fc:94	HuaweiTe_e4:9d:50	EAPOL	231	Key (Message 3 of 4)
271	7.012817166	HuaweiTe_e4:9d:50	AvmAudio_49:fc:94	EAPOL	175	Key (Message 4 of 4)

```
Key Length: 16
Replay Counter: 2
WPA Key Nonce: b0c860d8718f42b6d12e132618003c84ea9ed4379bcc57b5...
Key IV: 00000000000000000000000000000000
WPA Key RSC: 2500000000000000
```

```
0000  00 00 26 00 2f 40 00 a0  20 08 00 a0 20 08 00 00   ..&./@..  ... ...
0010  0d b7 f8 07 01 00 00 00  10 02 80 09 a0 00 c4 00   ........ ........
0020  00 00 c4 00 ba 01 88 02  3a 01 fc 3f 7c e4 9d 50           :  ?|  P
```

Figure 2: EAPOL handshake captured with Wireshark

A very helpful guide on how to decrypt WiFi traffic can be found in [Wir15].

3.2 aircrack-ng

The explanations above are only applicable when the network password is known and a full EAPOL handshake was captured. The how-to in [Nul16] mentions the Aircrack software suite as a way to gain access to foreign networks. This was not part of this work but it should be mentioned that with `aireplay-ng --deauth` it is possible to expulse network members which leads to a new EAPOL handshake to be captured.

References

[Knl15] *mac80211 Multiple Virtual Interface (vif) Support.* `https://wireless.wiki.kernel.org/en/users/Documentation/iw/vif`, 2015. – [Online; accessed 03-March-2017]

[Leh05] Lehembre, Guillaume: *Wi-Fi security – WEP, WPA and WPA2.* `http://www.hsc.fr/ressources/articles/hakin9_wifi/hakin9_wifi_EN.pdf`, 2005. – [Online; accessed 03-March-2017]

[Nul16] *Cracking WPA2-PSK.* `https://null-byte.wonderhowto.com/how-to/hack-wi-fi-cracking-wpa2-psk-passwords-using-aircrack-ng-0148366/`, 2016. – [Online; accessed 03-March-2017]

[Wir15] *How to Decrypt 802.11.* `https://wiki.wireshark.org/HowToDecrypt802.11`, 2015. – [Online; accessed 03-March-2017]

Declaration of authorship

I hereby certify that this report has been composed by me and is based on my own work, unless stated otherwise. No other person's work has been used without due acknowledgement in this report. All references and verbatim extracts have been quoted, and all sources of information, including graphs and data sets, have been specifically acknowledged.

Dresden, March 20, 2016 Patrick Ziegler