universität ulm

Security and Privacy in Mobile Systems

Institute of Distributed Systems | Summer Term 2024
Echo Meißner, Prof. Dr. Frank Kargl

Assignment 2: WiFi Attacks

Deadline: 2024-06-26 @23:55

In this assignment, your task is to familiarize yourself with *scapy*,[1] an interactive packet manipulation tool, to perform Denial of Service (DoS) attacks on target stations associated to an access point in an IEEE 802.11 wireless LAN environment, to familiarize yourself with the *aircrack-ng* tools, to crack WPA2-PSK using a bruteforce attack, and to understand the 802.11i 4-way Handshake.

Important: Do not try Tasks 3, 4 & 5 in the wireless network provided by the university!

Submission: Please, submit one `.txt` file entitled `assignment2-solution.txt` including the answers to the following tasks.

Task 1: Download and install Scapy (0)

Download and install scapy on your system. It is recommended to use your system's package management facility. For further tips on how to install scapy check out `https://scapy.readthedocs.io/en/latest/installation.html`.

Alternatively, you can use kali linux[2], which already has all the tools needed for this exercise sheet preinstalled. For instructions on how to create kali live usb stick, see `https://www.kali.org/docs/usb/live-usb-install-with-linux/`.

Task 2: Send your first packets (2)

First off, to get a feel for scapy, send a UDP packet with "hello world" as payload over the network. For an introduction to scapy see `http://www.secdev.org/projects/scapy/doc/`

Your task is to:

- Set up a UDP server, e.g., using `nc`.[3]
- Construct a UDP datagram with the text "hello world" as payload and send it to the listening server.

Note: If you are having trouble sending and receiving packets on the same machine (localhost), try setting up the UDP server on a separate machine.

Submission: Within the `assignment2-solution.txt`, please submit the solution to this part as *task2-solution* containing the commands used to set up the UDP server / listener, and the scapy commands used to create and send the UDP packet.

[1] `https://scapy.net/`
[2] `https://www.kali.org/`
[3] `http://man.openbsd.org/nc`

Task 3: Deauthentication attack (2)

Although management frame protection (802.11w, now part of 802.11-2016) has been around for a few years now, most 802.11 networks still do not use it. Construct a deauthentication packet with scapy and disconnect a target station, e.g., your mobile device, from the wireless network **at home**.

Note 1: Stations automatically reconnect to the access point. You need to send many deauthentication packets to effectively disconnect a station.

Note 2: Raw 802.11 injection in scapy does not work with all setups, and it may not work on your machine. However, you will still get the points if the submitted scapy commands are correct. For troubleshooting, you can also use the scapy command *wireshark(deauth_frame)* to check if wireshark interprets the frame correctly. If it does not work, but you still want to try deauthenticating a station from a wireless network, you can try using *aireplay-ng* from the *aircrack-ng* package; however, this is not part of this task.

Submission: Within the `assignment2-solution.txt`, please submit the solution to this part as *task3-solution* containing the scapy commands you used to create and send the deauthentication packets, state whether the attack succeeded in deauthenticating the target station and after how many deauth packets sent. Please, remember — for privacy purposes — to blind the MAC addresses for your AP and your target device when submitting your solution.

Task 4: WPA Cracking - w/o Handshake (2)

Most non-enterprise wireless networks are encrypted using WPA2-PSK. In this task we will perform a brute-force attack on the PSK using *aircrack-ng*.

You are provided with the `no-handshake.pcap` file protected with wpa2-psk including captured WiFi packets but does not include any handshake. Also, you are provided with `wordlist.txt` to be used for the dictionary attack.
a.) Using *aircrack-ng* tools, can you obtain the password for the captured connection, if so what is the password? Please, justify your answer.
b.) What is the command you may use to obtain the password in this case and what is the output when you run the corresponding command?
c.) If you succeed in obtaining the password in b, are you able to obtain the GTK, KCK, KEK, TK, and PMK?

Hint: You may use Wireshark to inspect the captured packets and decrypt the traffic once you crack the password.
Submission: Within the `assignment2-solution.txt`, please submit the solution to this part as *task4-solution* containing the answers to each of the previous points; a, b, and c.

Task 5: WPA2 Cracking - w/ Handshake (2)

You are provided with the `handshake.pcap` file protected with wpa2-psk including captured WiFi handshake messages. Also, you are provided with `wordlist.txt` to be used for this task.
Similar to the previous task, you are asked to answer the points a, b, and c (encode the keys as hex strings).

Submission: Within the `assignment2-solution.txt`, please submit the solution to this part as *task5-solution* containing the answers to each of the previous points; a, b, and c, found in **Task 4**.

Task 6: 802.11i 4-way Handshake (2)

WPA2-PSK uses separate session keys for each station, which are generated during the 4-way handshake. This way various stations in a wireless network cannot eavesdrop each other. Explain why and how it is possible for an insider to still eavesdrop the packets of other.

Submission: Within the `assignment2-solution.txt`, please submit the solution to this part as *task6-solution* containing the answer to your explanation of this 4-way handshake attack.