Brandenburg University of Technology
Cottbus-Senftenberg
Chair of IT Security

Introduction to Cyber Security
Practical Exercise Class
Winter Term 2024/2025

Introduction to Cyber Security
– Network Security –

Deadline: 8th December, 2024

Introduction

In this second lab, we will perform a basic man-in-the-middle (MITM) attack from within the network of two communicating parties. The goal is to learn a secret which is sent between Alice and Bob. There are many real-world scenarios in which a man-in-the-middle attack is of great help to an attacker, ranging from *passively* stealing private information, such as passwords and encryption keys, up to analyzing the participants network traffic to gain knowledge about them even if the traffic is encrypted. In an *active* attack scenario, in which messages are actively modified, a MITM attack can even be used to covertly hijack the encryption following a Diffie-Hellman key exchange or to manipulate the functioning of cyber-physical systems.

Given its versatility, it is thus vital to understand how a MITM attack can be mounted in principle and what possible solutions there are to overcome this threat. While in this lab, you will learn how to mount one specific type of MITM attack in practice, the lectures will teach you more about how to deal with the general authenticity issues that MITM attacks tend to rely on.

Notes

Please note that these practical tasks assume basic knowledge to have been learned in previous studies. This also holds true for topics that will be covered in the lecture or exercise classes of Introduction to Cyber Security at some point, but have not yet been held. If you find yourself missing the knowledge required to solve this task sheet, you must attain it on your own through the process of self study.

For the purposes of this task sheet, you should be familiar with the usage of the UNIX/Linux operating system, particularly with the usage of the command line (generally using Bash or a similar shell). If you are still looking to learn these basics, you might be interested in playing through the Linux challenges provided in the Introduction to Linux slides on Moodle.

In addition, you will need basic understanding of network programming, since you will be creating raw Ethernet packets in order to trick the other network participants. To simplify the process of attacking, you are provided with a C/C++ program called `raw_packet.c`, the basic functioning of which you should be able to understand. If you have trouble understanding a part of the program, you are of course free to ask. Nonetheless, if you are not yet familiar with C/C++, you should get familiar with it in self study – this is especially important for the next task sheet, in which low-level programming skills are required to perform stack-based buffer overflows.

1 Preparation

1.1 Prepare the Laboratory Environment

In this task we consider a switched Ethernet network that contains three computers: Alice, Bob, and Mallory. See Figure 1 for a graphical representation. You are going to assume the role of the attacker Mallory who tries to establish a Man in the Middle status. You have received a text file containing the login data for your personal instance of this task through Moodle.

The machine used by Mallory can be accessed from *within the BTU network only*[1], using the command:

```
~$ ssh mallory@neuseeland.informatik.tu-cottbus.de -p <your personal port>
```

Once you logged in, you can list the network interfaces available on Mallory's machine using:

```
~$ ifconfig -a
```

You will find that there are three network interfaces available: `lo`, `eth0` and `eth1`. You only need to concern yourself with `eth1`, which is connected to the internal network where Alice and Bob communicate. To fulfill your task, you have been given special privileges related to `eth1`. Namely, you are able to run the following commands using sudo:

```
~$ sudo ifconfig eth1 ...
~$ sudo tcpdump -i eth1 ...
~$ sudo arp -i eth1 ...
~$ sudo raw_packet -i eth1 ...
```

For example, you are able to watch the communication over the internal network using tcpdump:

```
~$ sudo tcpdump -i eth1
```

If you observe an output similar to the following (where ARP requests come roughly every 10 seconds), your experimental setup is working as intended:

```
~$ sudo tcpdump -i eth1 -n
   tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
   listening on eth1, link-type EN10MB (Ethernet), snapshot length 262144 bytes
   22:15:56.551519 ARP, Request who-has 10.10.3.3 tell 10.10.3.2, length 28
   22:16:06.673798 ARP, Request who-has 10.10.3.3 tell 10.10.3.2, length 28
```

The IP addresses of Alice and Bob will differ according to your personal instance of the challenge. To find the IP addresses of Alice and Bob, you may run `sudo tcpdump -i eth1 -n` to show addresses numerically, or you may simply ping these hosts using `ping alice` or `ping bob` (however, do note that this would turn your passive attack into an active one...). For example:

[1] You can either connect directly to *eduroam* on campus or use a VPN connection. If you use Linux as your daily driver, the OpenConnect VPN tool may be of use to you.

```
~$ ping alice
   PING alice (10.10.3.2) 56(84) bytes of data.
   64 bytes from icst2-3-alice-1.icst2-attacknet-3 (10.10.3.2): icmp_seq=1 ttl=64 time=0.177 ms
   64 bytes from icst2-3-alice-1.icst2-attacknet-3 (10.10.3.2): icmp_seq=2 ttl=64 time=0.104 ms
```

To support you in your task, the custom C program `raw_packet` is provided in the challenge instance. It will help you in creating raw ethernet packets. Make sure to familiarize yourself with the inner working of this little tool before you run it – not only is this a general good practice, but it may also be relevant for your lab defense. To this end, the source code of the program is provided on Moodle along with this lab sheet. Note that the program must be called with sudo since it directly interacts with the network interfaces in question.

1.2 Overview of the Experiment

In this task we consider a switched Ethernet network that contains three computers, Alice, Bob and Mallory, according to Figure 1. You will assume Mallory's role in this experiment.

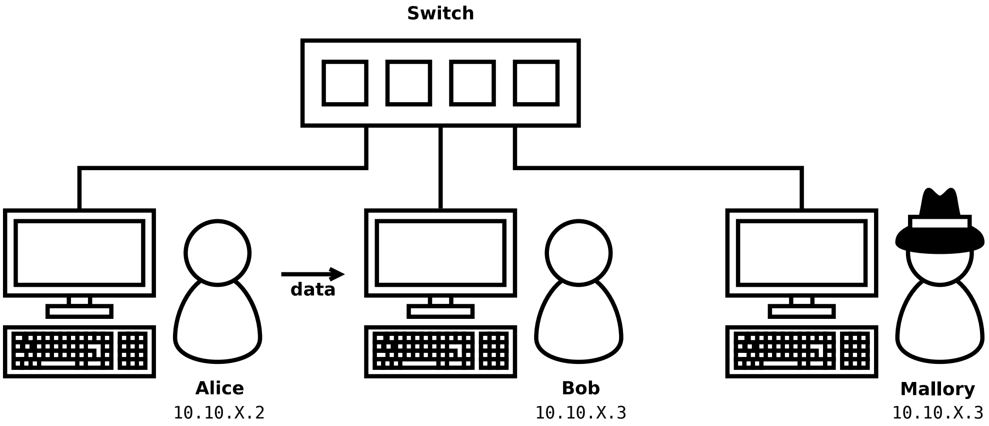

Figure 1: Sketch of the network scenario with ARP.

Every few seconds, the PCs of Alice and Bob send data packets through the network. To make this possible, they use the Address Resolution Protocol (ARP) to determine their target's MAC address for a given IP address. The switched nature of the network makes it impossible for Mallory to simply listen and eavesdrop traffic between Alice and Bob on her local network interface. However, this setup is still far from secure.

The overall goal of this task is to establish yourself as the *man in the middle* in the communication between Alice and Bob, i.e., the connection from Alice to Bob should be rerouted from Alice to Mallory and forwarded to Bob. Similarly, a message sent from Bob to Alice should take a route over Mallory. Thus, Mallory will be enabled to eavesdrop on all communications between Alice and Bob on their local interface. This can be achieved by exploiting the unauthenticated nature of the ARP protocol, i.e., we apply a so-called ARP spoofing or ARP cache poisoning attack.

2 Main Task

Use ARP spoofing (ARP cache poisoning) to impersonate a PC and perform a man-in-the-middle attack against Alice and Bob. Apart from the system commands `arp`, `tcpdump`, `ping` and the preinstalled hexeditors `hexedit` / `xxd` as well as the given C program `raw_packet` you are not allowed to use any third party program. Both `Bash` and `Python3`, as well as the `gcc` C-compiler are available on Mallory's machine.

Once you have established *Mallory* as the Man in the Middle, you can steal the secret that Alice sends to Bob repeatedly. The secret will be within the format: `CTF{...}`. Furthermore, you should take special care that Alice and Bob will not notice your attack! Your submission shall consist of the secret you stole (`secret.txt`), a short explanation of what you did (`Writeup.md`) and any scripts and resources you developed for your attacks (`mitm_script.{sh,py,cc}`).

Bonus: Lets assume the same attackers' goal and tools but this time Alice and Bob run an intrusion detection system capable to detect modified `ARP replies`, i.e., your attack is not allowed to use bogus `ARP replies`. Can you still mount the ARP spoofing and manage to establish yourself as the man in the middle, without using replies? Explain why your solution works and think about a possible way to mitigate the issue.

Hints:

1. You will want to research how the Address Resolution Protocol (ARP) works in order to develop your attack
2. `tcpdump` as well as `arp` require sudo privileges. You have these privileges only on network interface `eth1` (cf. section 1.1).
3. Raw Ethernet packets can be sent using the provided program `raw_packet` (cf. section 1.1). The provided program requires sudo privileges.
4. Make sure that Alice and Bob won't notice your attack. Think about why the traffic will be forwarded from Mallory to Bob, if Mallory has impersonated Bob towards Alice.
5. Think about influences on the network topology when sending faked ARP responses. You may get affected by your own attack or those of possible other attackers. Try to minimize these influences.
6. Multiple commands can be run simultaneously in the foreground e.g. by using `tmux` or `screen` or by simply opening multiple SSH connections.
7. You may want to use `scp` or `rsync` to copy (backup) data from Mallory to your own machine and vice versa. For example, to copy a folder called "scripts" to Mallory, you could use:

```
~$ rsync -avzu -h -e ’ssh -p <your port>’ scripts mallory@neuseeland.
   informatik.tu-cottbus.de:/home/mallory/scripts
```

3 Submission and Lab Defense

Upload the stolen secret as well as all of the payloads and scripts you wrote to solve the task to Moodle. Additionally, please provide a short writeup specifying what steps you have taken to solve the task and how to use the scripts you have developed. The zip file containing your submission should have the following structure:

```
Firstname_Lastname.zip
├── Writeup.md
├── secret.txt
└── src
    ├── mitm_script.{ sh | py | cc | * }
    └── ...
```

Prepare yourself for a lab defense of up to 30 minutes. In the lab defense, we will go through your solutions and discuss the way in which you solved the tasks. Ensure that you are familiar with all of the concepts that play a role within this lab and are able to defend why you took each step you took.

For this particular defense, be ready for a live demonstration of your attack using the scripts that you uploaded to Moodle. During the defense you should be able to explain your attack, taking special care as to *why* the attack is working. For example, how does Mallory know to whom the traffic needs to be forwarded to after they have established themselves as the man in the middle? Does the forwarding happen automatically? Why? While explaining your attack, describe each step you have done to obtain the secret data and reason for it. In addition, you must be able to explain any written/used code/program as well as the provided ones that you have used. The examiners might also ask conceptual questions around the topic of authenticity and man-in-the-middle attacks.

References

[1] *Linux Man Page of arp*. URL: http://man7.org/linux/man-pages/man8/arp.8.html (visited on 10/19/2022).

[2] *Linux Man Page of socket interface*. URL: http://man7.org/linux/man-pages/man7/socket.7.html (visited on 10/19/2022).

[3] D. C. Plummer. *An Ethernet Address Resolution Protocol*. 1982. URL: https://tools.ietf.org/html/rfc826 (visited on 10/19/2022).