Laboratory II

Binna, Reschenhofer, Schörghofer

Course: Internet Infrastructure and Security

Lecturer: FH-Prof. DI Mag. Dr. Dominik Engel

23.11.2016

Table of Contents

List of Abbreviations

MOTD Message of the Day

DCE Data Communication Endpoint

OSPF Open Shortest Path First

IP Internet Protocol

LSR Link State Routing

AS Autonomous System

TTL Time To Live

1 Topology

The following topology (figure 1.1, taken from Moodle instructions) had to be recreated in the lab.

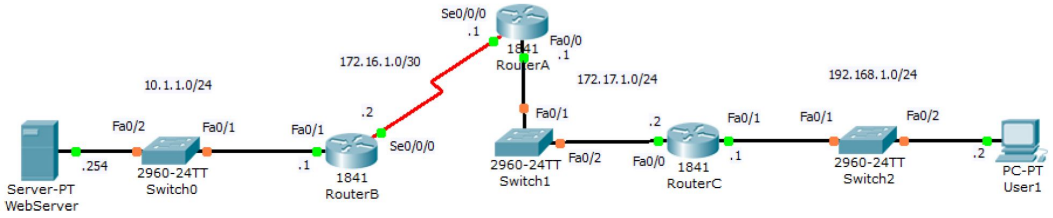

Figure 1.1: Topology (Moodle)

Each device was configured with basic settings like hostname, Message of the Day (MOTD), a secret enable password, disabled IP domain lookup, synchronous logging on line console 0 and the service password-encryption, which prevents passwords from being displayed in clear-text in the start-up and running configuration. The important part is that the clock-rate for the serial interface has to be configured on the Data Communication Endpoint (DCE) device.

```
interface Serial0/0/0
 bandwidth 64
 ip address 172.16.1.1 255.255.255.252
 ip ospf authentication message-digest
 ip ospf message-digest-key 1 md5 7 094F471A1A0A
 clock rate 64000
```

Listing 1.1: Setting the clock-rate on Router A

Open Shortest Path First (OSPF) is a routing protocol for Internet Protocol (IP) networks. It uses a Link State Routing (LSR) algorithm and falls into the group of interior routing protocols, operating within a single Autonomous System (AS). Concerning the OSPF routing, every router assigns its known networks to the OSPF routing process with area 0.

```
router ospf 1
 area 0 authentication message-digest
 network 172.16.1.0 0.0.0.3 area 0
 network 172.17.1.0 0.0.0.255 area 0
```

Listing 1.2: OSPF routing example router A

'Nginx' has been set up as a webserver on a Linux-PC. The default 'index.html' was adapted to show the message `"Guter Webserver!"` when accessing the server via a web browser.

The IP address has been assigned to '10.1.1.254'. Ping and webserver access from the User-PC to the server has been successful as shown in (figure 1.2). The firewall on the Windows machines was disabled to prevent accidental ICMP packet rejects.

Figure 1.2: Webserver access

For this topology (and the respected route through the network), the round-trip time from the user PC to the Linux-PC hosting the webserver was 18ms on average and had a remaining Time To Live (TTL) value of 61.
The 'tracert' output, as shown in (figure 1.3) from the user PC to the server, shows that it took 4 hops to reach the server.

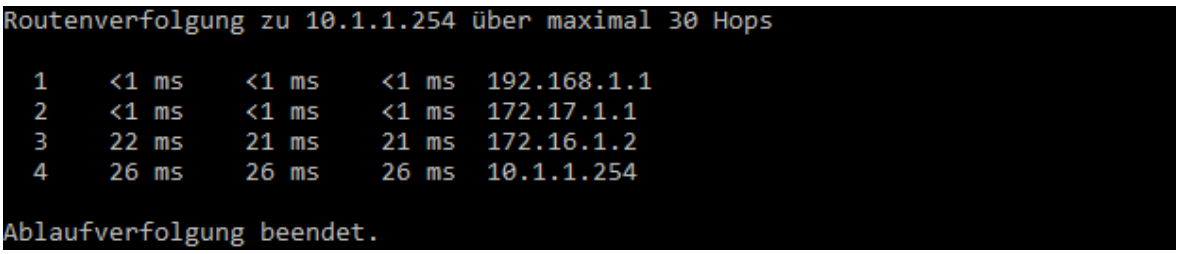

Figure 1.3: Tracert from user pc to good webserver

2 Router Spoofing

For the router spoofing attack another router has been added to the network as seen in (figure 2.1, taken from Moodle instructions). This router also operates in OSPF mode.

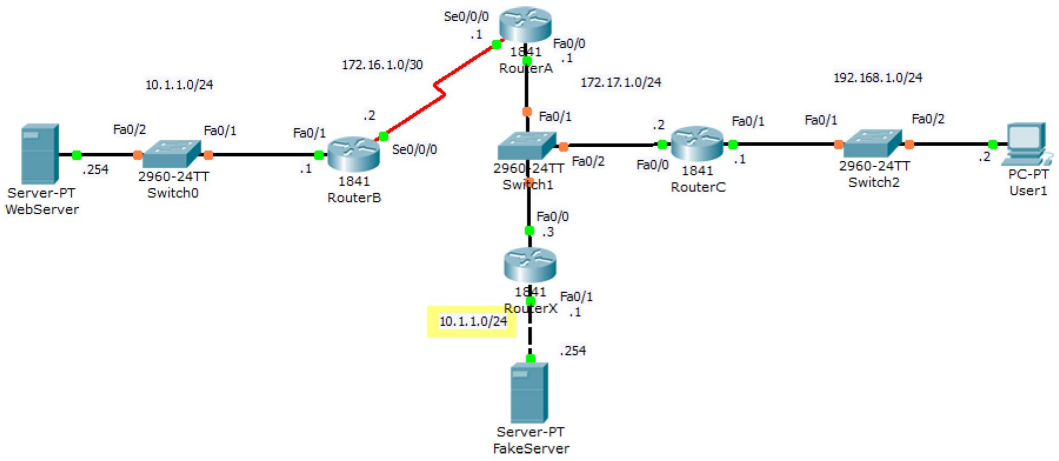

Figure 2.1: Router spoofing topology

Router X is now configured to advertise the same networks as Router B. The difference between those routers is, that Router X is connected Router A via a faster connection (100 Mbps vs. 64 Kbps) than Router B.

To also see a difference on the User PC, a second 'nginx' server was setup on another Linux-PC, this time containing `"Bad Webserver!"` in the 'index.html'.

```
router ospf 1
 network 10.1.1.0 0.0.0.255 area 0
 network 172.17.1.0 0.0.0.255 area 0
```

Listing 2.1: OSPF configuration on router X

OSPF detects changes in the topology and computes the shortest-path tree for each route based on the "Dijkstra's algorithm". The bad Router X propagates that he has a better route to network "10.1.1.0/24".
This results in a shorter path from the user pc to the webserver. The user pc doesn't know that this is the bad webserver since OSPF only detected a shorter path. The router does not know that the two networks, although having the same subnet, are not the same.
Therefore, an ICMP Echo request (sent with `ping`) from the user pc will now be

routed to the bad server. The round-trip time from the user pc to the bad server is now <1ms with a TTL value of 62 (one hop less). Because of the new route the packet isn't sent over the serial link, but instead is being transmitted via the faster Ethernet connection to Router X. The 'tracert' output (figure 2.2) shows that the packet only needs 3 hops to its destination and has an average round-trip time of <1ms.

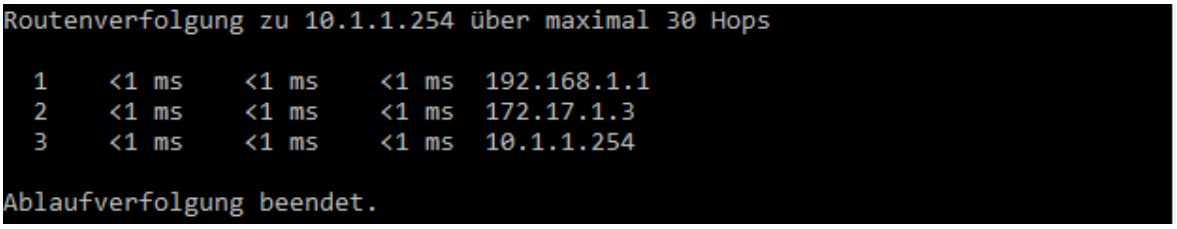

Figure 2.2: Tracert from user pc to bad webserver

If the user accesses the webserver again, he will be redirected to the bad webserver, because of the routing table entry in Router C which states that the shortest path to the network "10.1.1.0/24" is reached via Router X. Figure 2.3 shows the output when accessing the webserver again.

Figure 2.3: Bad webserver access

3 Router Authentication

In this chapter, router authentication methods are explained. Router authentication for OSPF allows to flexibly authenticate OSPF neighbours. This enables OSPF routing to exchange routing update information in a more secure manner. There are three different types of authentication in OSPF:

- Null authentication: Also called type 0 → no authentication information is included in the packet header. This is the default setting.
- Plain text authentication: type 1 → use of simple plain-text passwords.
- MD5 authentication: type 2 → use of MD5 hashed passwords (also not secure anymore!)

OSPFv3 is capable of using SHA1 authentication, which is at least better than MD5 authentication.

3.1 Plain Text Authentication

At first, plain text authentication has been configured. The authentication key is configured for each interface separately by use of the following command:

```
ip ospf authentication-key cisco
```

Only interfaces, where the authentication key matches, can participate in OSPF advertising. The following command then enables OSPF authentication for all interfaces inside area 0:

```
router ospf 1
       area 0 authentication
```

When the commands above are executed solely on Router A and Router C, all OSPF routes that were advertised from Router B are now lost, because Router B doesn't have the authentication configuration.
The main disadvantage of plain text authentication is, that the password can be clearly seen in the packet, as shown in figure 3.1.

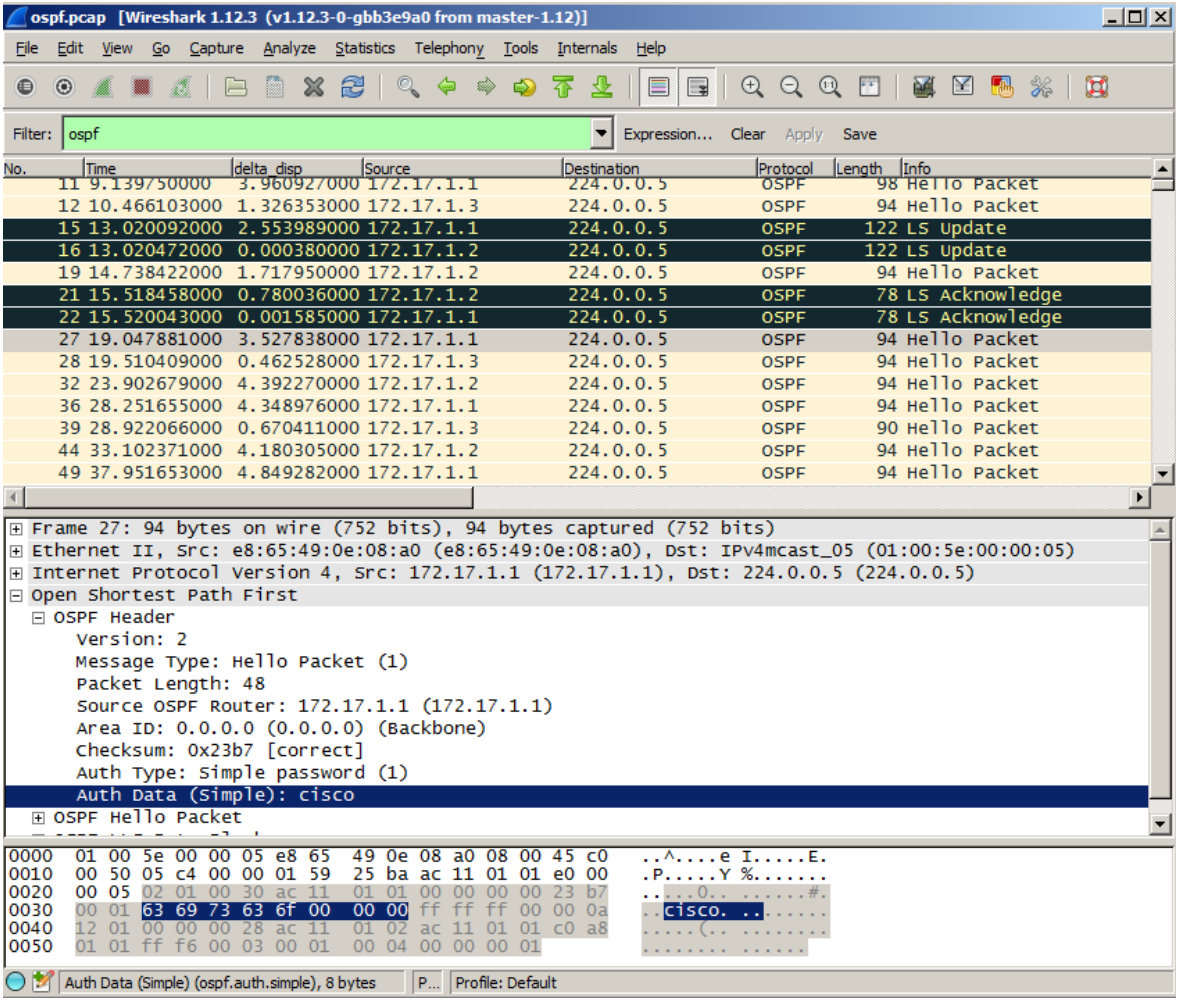

Figure 3.1: Wireshark showing the plaintext password

3.2 MD5 Authentication

First the MD5 message digest has to be set on the interfaces:

```
Interface s0/0/0
      ip ospf message-digest-key 1 md5 cisco
```

After that, MD5 authentication can either be enabled on a per interface basis with the following command:

```
Interface s0/0/0
      ip ospf authentication message-digest
```

Or globally for all interfaces belonging to a specific area:

```
router ospf 1
       area 0 authentication message-digest
```

With activated MD5 authentication, the password is no longer visible in plaintext in the Wireshark capture, as can be seen in figure 3.2.

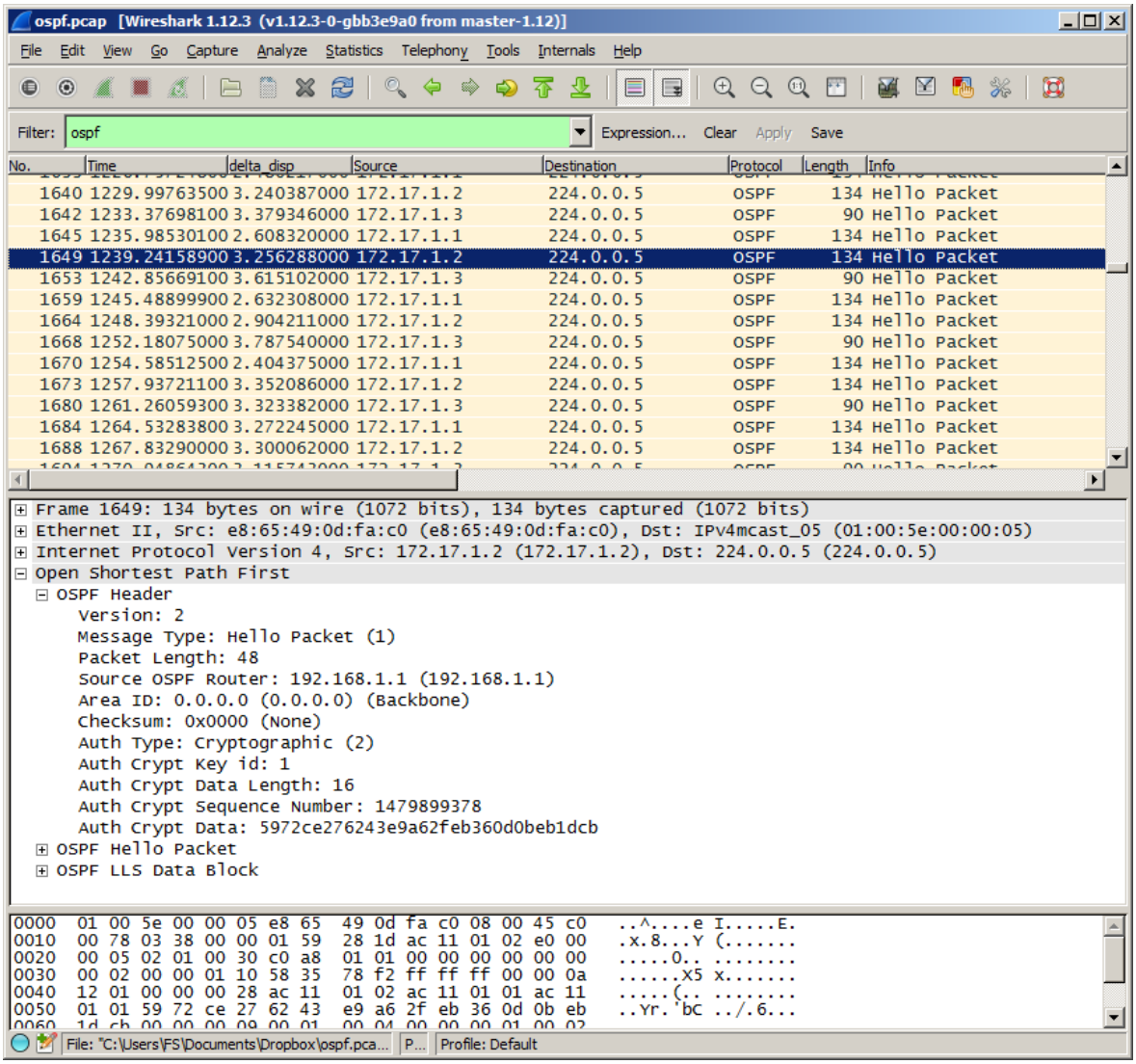

Figure 3.2: OSPF capture with MD5 authentication

4 Sniffing Attack and Clear Text Password Router Authentication

Using a PC connected to `Switch1`, we were able to capture the OSPF packets seen in the previous section.

The null and plaintext authentication methods have obvious security problems. An attacker, who can sniff packets, can then enter the key on the malicious router and advertise different routes. Being able to capture the OSPF packets also means the attacker can identify which networks are advertised by which router, which allows for easier configuration of the 'bad webserver'.

Using MD5 as authentication does not reveal the password anymore. But as MD5 can be broken with standard PC hardware in a short amount of time, this is also no longer secure. To avoid replay attacks a sequence number is also introduced. The MD5 hash seen in figure 3.2 "Auth Crypt Data" is generated (RFC2328) which also includes the sequence number as source.

If all routers use authentication, any OSPF packets without it will be discarded. This also helps to secure against a potentially misconfigured router in the network.

4.1 Cracking MD5 with "John the Ripper"

"John the Ripper" (`http://www.openwall.com/john/`) is a program for cracking different hashing algorithms. We used it to try and calculate the plaintext password from the captured OSPF packets.

To do this, we saved a separate PCAP file which only contained OSPF packets with MD5 authentication.

This file then had to be further processed with 'ettercap' (`https://ettercap.github.io/ettercap`) to produce a textfile which John can read:

```
ettercap -Tqr ospf_4.pcapng > ospf1.txt
```

This textfile then was used by John, which cracked the MD5 hash in a few seconds. The result can be seen in figure 4.1.

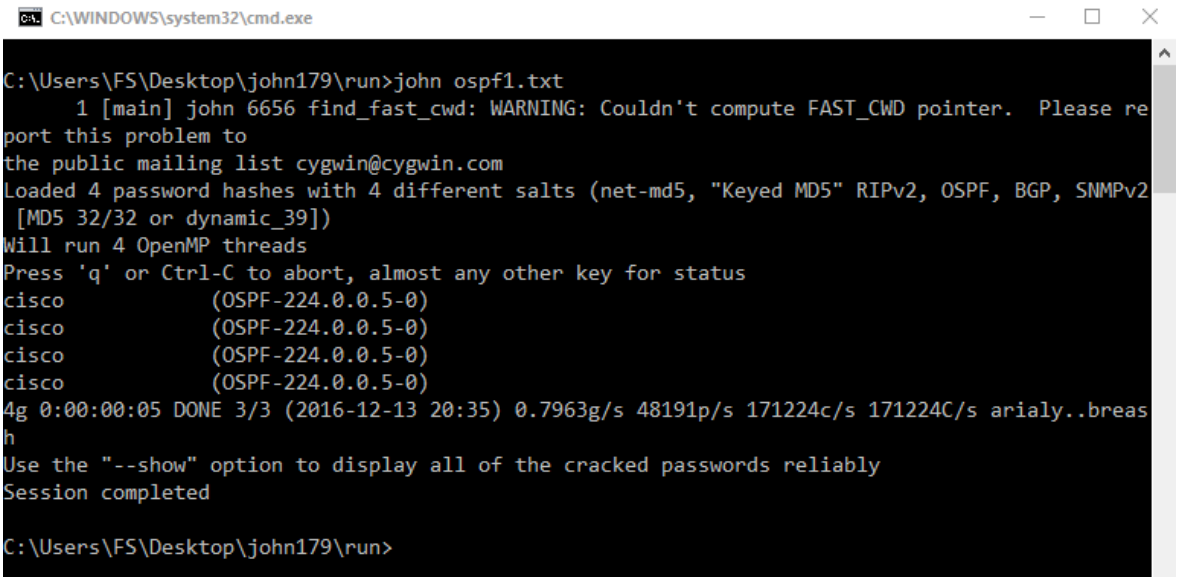

Figure 4.1: John the ripper in action

The capture file contained four MD5 hashes, whereby every hash contained the same password ‘cisco’.