This scenario is realistic, e.g. for governmental attackers since they can compel root CAs into issuing subordinate CA certificates [32].

- The attacker has access to one or more host certificates that are signed by an issuer that is trusted by the user. As a result, the attacker is able to attack a limited set of SSL/TLS connections without generating any warnings in the user's browser. This scenario is realistic e.g. for hackers like the ones mentioned in section 2.3.2.

- The attacker is not able to present valid certificates and hopes that the user will ignore any warnings that appear in its browser.

Combining each of the seven MitM classes with each of the three scenarios for the certificate ownership results in a total of twenty-one MitM scenarios. Discussing all of them would result in a very lengthy chapter with repetitive content. In order to avoid this, we chose to present three settings of which we think are the most realistic ones and discuss the behavior Crossbear shows when facing them. Within these three settings we will cover most of the MitM classes as well as the first two of the scenarios for the certificate ownership. What will not be covered is the "Server-to-TLS" MitM class, in which a MitM attacks the connection between the Crossbear server and the TLS-enabled destination server. In fact, attacks like this do not affect Crossbear clients other than domains that use multiple certificates at the same time. A detailed discussion of this issue is given in section 6.4.

The three settings that we will discuss are the one of a Poisoned Access Point (PAP) (section 6.1), the one of a MitM-ing HTTPs proxy (section 6.2), and the one of a surveilled TLS-enabled server (section 6.3).

6.1 Poisoned Access Point (PAP)

One of the settings that we think are most likely for a real-world MitM, is the one of a Poisoned Access Point (PAP). In this setting, an attacker controls a local network, that is attached to the Internet via an Access Point. By "controlling", we mean the ability of manipulating any packet that leaves or enters the network. For most networks, this is not only possible for the operator of the Access Point, but, due to the ARP-Spoofing technique [57], also to virtually anybody else.

A real-world example for this setting could look like this: a café or a hotel provides wireless Internet access to its customers. A malicious person places a notebook in this network and uses the ARP-Spoofing technique to redirect all of the connections between normal users and the Internet to pass through this notebook. Afterwards he/she sets up a MitM that automatically intercepts connections and tries to gather login information for as much Web sites as possible. Finally, the attacker hijacks the Web site accounts with the logins it gathered.

6.1.1 Setup

The environment that we set up for simulating the PAP scenario is shown in Figure 6.1. We connected three Crossbear clients to the `131.159.14.0/25` network. Two of them were connected directly, while the third one was connected via a wireless Access Point. This Access Point provided Internet access to the private `192.168.1.0/24` network using the Network Address Translation (NAT). We placed two machines within that private network: a MitM attacker and a MitM victim that uses the Crossbear Firefox Add-on. The MitM was equipped with a certificate for `my.hypovereinsbank.de` that the victim trusted.

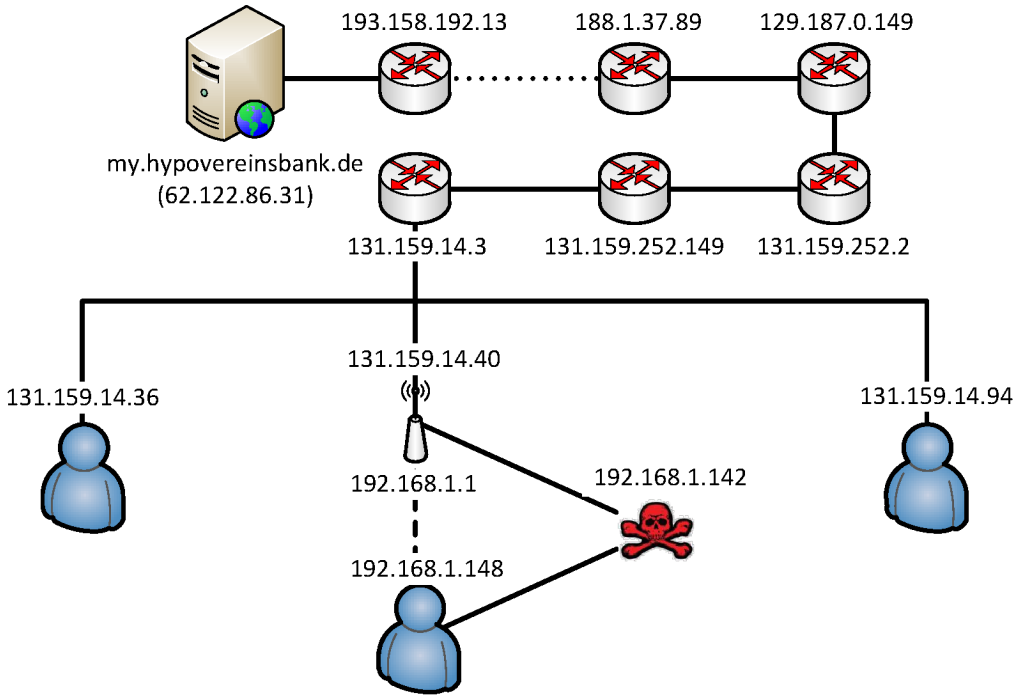

Figure 6.1: Setup of the PAP scenario

The software that was used to run the MitM attack was Moxie Marlinspike's sslsniff [29]. There are two modes in which it can be used to set up a MitM. The first one is to provide a certificate that the victim trusts as a signing certificate (i.e. a CA certificate). In that mode, sslsniff will intercept all SSL/TLS connections, replace their server certificates with certificates that were generated on the fly, and sign these certificates with the trusted signing certificate.

However, as history shows (see also section 2.3.2), this is not the usual modus operandi of real-world MitM attackers. Usually, the attackers have access to a number of certificates of which they assume that the victim accepts them. Consequently, they intercept only those connections that can be intercepted without generating error messages in the user's browser. That is, they intercept only those connections that are directed to one of the domains for which they have a certificate. This is the second operation mode sslsniff provides, and the one that is used in this setup. Setting up a MitM in that operation mode takes four commands, which are shown in Listing 6.1.

Listing 6.1: Setting up a MitM attack using sslsniff

```
1  # Enable IP forwarding
2  echo 1 > /proc/sys/net/ipv4/ip_forward
3
4  # Add a iptables-rule to intercept SSL traffic
5  iptables -t nat -A PREROUTING -p tcp --destination-port 443 -j
       REDIRECT --to-ports 999
6
7  # Set up sslsniff to intercept all SSL-connections for which
8  # there are certificates in the /home/user/certs-folder
9  sslsniff -t -c /home/user/certs -s 999 -w /home/user/log.log
10
11 # Poison the victim's ARP cache
12 arpspoof -i eth0 -t 192.168.1.148 192.168.1.1
```

6.1.2 Results

Since the attacker's system is set to forward IP traffic, the victim does not experience any change in the network's behavior until it navigates to `my.hypovereinsbank.de`, which is the domain the MitM attacks. As soon as it does, the Crossbear client will realize that the Web site uses an unknown certificate and request a certificate verification from the Crossbear server. Since the certificate verification results in the detection of the MitM, the user will be notified. Additionally, a HuntingTask message containing a Hunting Task for `my.hypovereinsbank.de:443/62.122.86.31` is sent to the victim, which will execute the contained Hunting Task (see also section 4.4.1). Since the victim is placed in a private network, the route that it measures will start with a private IP (`192.168.1.1`). Local IPs are of little use when it comes to locating a MitM and are therefore removed from the route and replaced with the public IP of the system, i.e. the Access Point's public IP. As discussed in section 4.3.4, the route and the certificate chain are bundled in a HuntingTaskReply message and sent to the Crossbear server.

Some minutes after the victim executed the Hunting Task, the two other Crossbear clients download the Hunting Task list and execute it. Since they are not under attack, they observe the correct certificate chain for `my.hypovereinsbank.de:443/62.122.86.31`. Although, the routes that they record for the target's IP do not contain any private IP addresses, they prepended their public IPs to them, and send them along with the observed certificate chains to the Crossbear server.

id	certid	serverhostport	timeofobservation	observertype	observerip
15	9	my.hypovereinsbank.de:443	2012-01-31 13:14:17	CB-CVR	131.159.14.*
16	10	my.hypovereinsbank.de:443	2012-01-31 13:14:17	CB-Server	131.159.14.41
17	9	my.hypovereinsbank.de:443	2012-01-31 13:15:22	CB-Hunter	131.159.14.40
18	11	my.hypovereinsbank.de:443	2012-01-31 13:30:55	CB-Hunter	131.159.14.36
19	11	my.hypovereinsbank.de:443	2012-01-31 13:35:17	CB-Hunter	131.159.14.94

Table 6.2: Entries in the CertObservations table (PAP scenario)

The entries that are created in the database of the Crossbear server during the whole process are shown in Tables 6.2 and 6.3. Table 6.2 shows an excerpt from the CertObservations table, which stores the observations of certificate chains that were either made by the Crossbear server itself, by a Crossbear Hunter, or by the sender of a CertVerifyRequest. The first two entries of the table depict the concurrent observation of two different certificate chains by the Crossbear server and the victim. This event triggered the creation of a Hunting Task for `my.hypovereinsbank.de:443/62.122.86.31`, which was firstly executed by a Crossbear Hunter at the victim's machine and afterwards by two other Hunters.

Table 6.3 shows an excerpt from the HuntingTaskResults table, which contains the routes that the Hunters measured. The certificate chains that were observed are stored in the ServerCerts and the ChainCerts table (see also section 5.1.2). References to these certificate chains are stored in the CertObservations table. However, in order to understand Table 6.2, it is necessary to know that certid 10 and 11 actually refer to the same certificate (C0). The reason why C0 has two different IDs is that it has been observed with two different chains: once by the Crossbear server which uses Java's certificate store and once by the Crossbear Firefox Add-on which uses Firefox's certificate store.

C0 was observed by the Crossbear server as well as by all Hunters but the one that made observation number 17. Therefore it is assumed that the latter was made by a Hunter that was under attack by a MitM while all others were made by Hunters that had a uncompromised connection to the target.

id	Hunting Taskid	route	observation
1	1	131.159.14.40 131.159.14.3 131.159.252.149 131.159.252.2 129.187.0.149 188.1.37.89 ... 193.158.192.13 62.122.86.31	17
2	1	131.159.14.36 131.159.14.3 131.159.252.149 131.159.252.2 129.187.0.149 188.1.37.89 ... 193.158.192.13 62.122.86.31	18
3	1	131.159.14.94 131.159.14.3 131.159.252.149 131.159.252.2 129.187.0.149 188.1.37.89 ... 193.158.192.13 62.122.86.31	19

Table 6.3: Entries in the Hunting TaskResults table (PAP scenario)

Looking at Table 6.3, it appears that all three Hunters reported routes that merely differ in the first IP. Applying the algorithm described in section 3.4 we therefore come to the conclusion that the MitM is somewhere between `131.159.14.3` and `131.159.14.40`. Since `131.159.14.40` is the first IP in the poisoned route, it is very likely that it is the IP of a poisoned Access Point.

Based on this finding and the fact that Crossbear notified the user about the MitM, we come to the conclusion that Crossbear was able to detect and locate the MitM in the PAP scenario.

6.2 MitM-ing HTTPs proxy

Firefox and many other browser support proxied HTTPs connections [58]. This technique is useful, e.g., when it comes to separating networks while at the same time providing access to HTTP and HTTPs resources. In order to use proxied HTTPs connections with Firefox, the latter needs to be told the URL of the proxy, and for which pages it should use this proxy. This can either be done by explicitly specifying the proxy in the browser's network options or by using a Proxy Auto-Config (PAC) file. The second option is quite popular, e.g., within university networks where it is often used to restrict access to library services. When PAC files are involved, users often do not know whether they are actually using a proxy for a Web site or not.

While there are HTTPs proxies like AnalogX Proxy [59] that simply forward SSL packets, there are others like SQUID [44] that can be configured to perform MitM attacks on SSL/TLS connections. In the following, we will analyze how Crossbear reacts when facing a MitM-ing SQUID proxy.