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.

6.2.1 Setup

The environment we set up for simulating the scenario of a MitM-ing HTTPs proxy is shown in Figure 6.2. In this scenario, a Crossbear client (`192.168.112.131`) is placed in a private network (`192.168.112.0/24`) which is connected to the Internet via a SQUID proxy (`192.168.112.135 <-> 138.246.47.109`). The proxy is configured to forward HTTP and HTTPs only and to MitM the latter using a CA certificate that the client trusts. The scenario does not include any other Crossbear clients but the one that is under attack since, as the following section will show, no Hunting Tasks are created in this scenario.

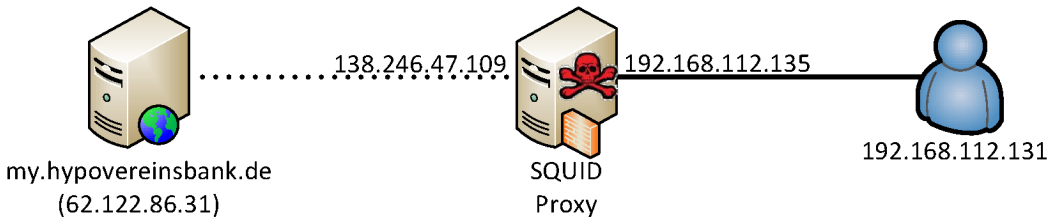

Figure 6.2: Setup of the MitM-ing HTTPs proxy scenario

Setting up and running a SQUID proxy is rather easy since SQUID's official homepage provides detailed documentation on that topic. Configuring it for MitM-ing HTTPs connections can be done by adding the lines from Listing 6.2 to the squid.conf. Doing so will cause the proxy to intercept all HTTPs connections but the ones to Crossbear's server URL, which is `crossbear.net.in.tum.de`. Removing lines 8 and 9 will cause the proxy to intercept all HTTPs connections.

Listing 6.2: Setting up a MitM attack using SQUID: excerpt from the squid.conf

```
1  # Ignore all certificate warnings and errors
2  sslproxy_cert_error allow all
3
4  # Disable accelerator mode
5  always_direct allow all
6
7  # Do not intercept Crossbear
8  acl crossbear dstdomain crossbear.net.in.tum.de
9  ssl_bump deny crossbear
10
11 # Incercept all others
12 ssl_bump allow all
13
14 # Activate a MitM-ing proxy that uses a signing certificate
15 http_port 3129 ssl-bump generate-host-certificates=on
       dynamic_cert_mem_cache_size=4MB cert=/home/test/trusted.pem
       key=/home/test/trusted.pem
```

6.2.2 Results

Configuring the SQUID proxy to intercept all HTTPs connections will cause the Crossbear client to display a warning (Figure 6.3) as soon as the user connects to any SSL/TLS secure Web site. This is due to the fact that if each HTTPs connection is intercepted, every Web site will seem to use a certificate that it has never used before. This triggers a CertVerifyRequest message to be sent to the Crossbear server which implies that a HTTPs

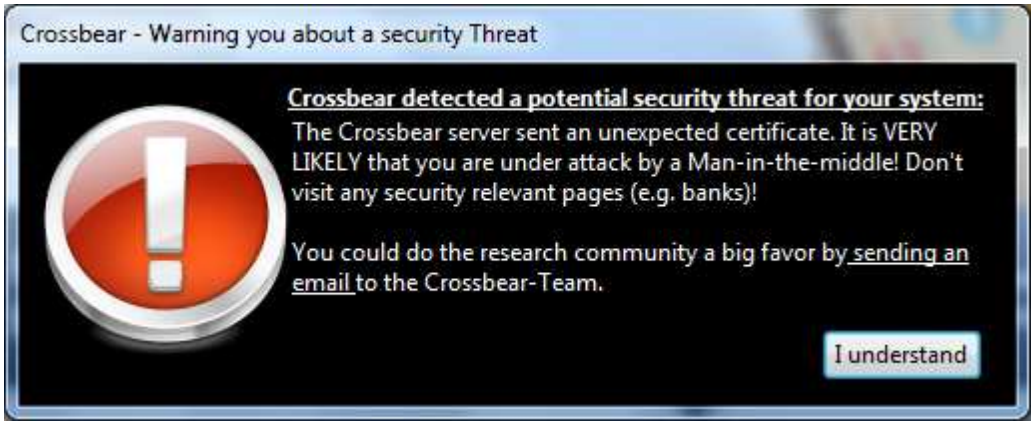

Figure 6.3: Crossbear notifying its user about a detected MitM

connection is opened for the Crossbear server. Since this connection uses an unexpected certificate, the MitM is detected (see also section 4.6).

Since the Crossbear client is not able to establish a secure communication channel to the Crossbear server, the latter will not be notified about this incident automatically. However, as Figure 6.3 shows, the user is asked to send an e-mail with automatically generated content to the Crossbear team. This e-mail contains, amongst others, the certificate chain the Crossbear server seems to use. Since there is no way of securely communicating any measurement results to the Crossbear server, no further steps are taken in this scenario.

Configuring the SQUID proxy to intercept only a subset of all HTTPs connections, which does not include the connections to the Crossbear server, causes the Crossbear system to show a different behavior. In this case, the Crossbear client is actually able to request a certificate judgment from the Crossbear server. For example, if the client connects to `my.hypovereinsbank.de`, it will send a CertVerifyRequest targeted for `my.hypovereinsbank.de:443/192.168.112.135` with the `ssl-proxy-active-bit` set to the Crossbear server (see also section 4.4.1). The latter will process this request, come to the conclusion that the client is under attack by a MitM, and report this to the client. Nevertheless, the Crossbear server will not generate a Hunting Task for the MitM.

This behavior has several reasons. First and foremost, it is not possible to hunt for `my.hypovereinsbank.de:443/192.168.112.135`, since each Hunter that would connect to the private IP address `192.168.112.135` would connect to a different machine. Even if it was possible to unambiguously define a target for a Hunting Task (e.g. because the proxy has a non-private IP address), there would be no way of generating a route for it. This is due to the fact that, assuming that the proxy does not additionally forward ICMP messages, there is no possible way to measure a route that passes it.

id	certid	serverip	timeofobservation	observertype	observerip
176	10	62.122.86.31	2012-02-02 15:01:57	CB-Server	131.159.14.41
175	95	192.168.112.135	2012-02-02 15:01:56	CB-CVR	138.246.47.109

Table 6.4: Entries in the CertObservations table (MitM-ing HTTPs proxy scenario)

However, we can look at the entries that were inserted into the CertObservations table (the relevant excerpt is shown in Table 6.4). This table contains, amongst other things, the IP address of the machine that sent the CertVerifyRequest. Since it was sent with the `ssl-proxy-active-bit` set, the IP that was stored as `observerip` does not represent the

Crossbear client's IP, but the IP of the proxy it used. Assuming that the proxy did not use proxy chaining, this IP is the one of the MitM.

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 the MitM in the MitM-ing proxy scenario and to give clues about its location.

6.3 Surveilled TLS-enabled server

Governments are known to spy on members of groups whose activities that they do not approve. Those groups usually suspect the governments to spy on them and therefore use access restricted platforms like, e.g., invite-only message boards to communicate. Surveilling these platforms might be a reasonable thing for the authorities to do.

Since surveillance is only effective when the surveilled group does not know about it, compelling the owner of the communication platform to cooperate is not the ultimate solution. After all, it is possible that he/she sympathizes with the surveilled group and therefore leaks information about the surveillance to the group. An alternative way of setting up a surveillance system is placing a MitM in a government-friendly network (e.g. an ISP) close to the platform. This MitM could be used to gather login credentials of users and therefore enable the authorities to access the platform, themselves. In the following, we will analyze Crossbear's behavior when facing a surveilled TLS-enabled server.

6.3.1 Setup

In order to simulate a surveilled TLS-enabled server, we set up a MitM that attacked one of our own servers (`lipari.net.in.tum.de`). The MitM was set up in the same way as in section 6.1, that is we installed sslsniff on a dedicated computer and used the ARP spoofing technique to make sure that all traffic that is directed to that server has to pass the MitM.

Additionally, we made use of the Planetlab platform by installing our Java Hunter on 160 worldwide distributed Planetlab nodes. We deployed by putting the Java Hunter and a modified Java 1.7 installation in a single archive, pushing it to the nodes using `pscp`, and finally installing it by executing an install script using `pssh` (see Listing 6.3).

Listing 6.3: Deploying Crossbear Java Hunters on the Planetlab platform

```
1  # Pack the Java Hunter and a modified Java 1.7-installation
2  tar -cjf ../crossbear.deploy ./*
3
4  # Push the packed Java Hunter and a install-script
5  pscp -h listOfNodes -l username "-O IdentityFile KeyFile"  "-O
       StrictHostKeyChecking no" -t 180 -p 50 ./* /home/username/
6
7  # Execute the install-script
8  # - Extract the Java Hunter and the Java installation
9  # - Create a cronjob to regularly execute the Hunter
10 # - Start the cron-daemon
11 pssh -h listOfNodes -l username "-O IdentityFile KeyFile"  "-O
       StrictHostKeyChecking no" -t 180 -p 50 /home/username/
       install.sh
```

6.3.2 Results

After setting up the MitM, we connected to our server using a Crossbear-protected Firefox. The result of the certificate verification of the surveilled server's certificate is shown in Figure 6.4.