Man in the Middle attacks

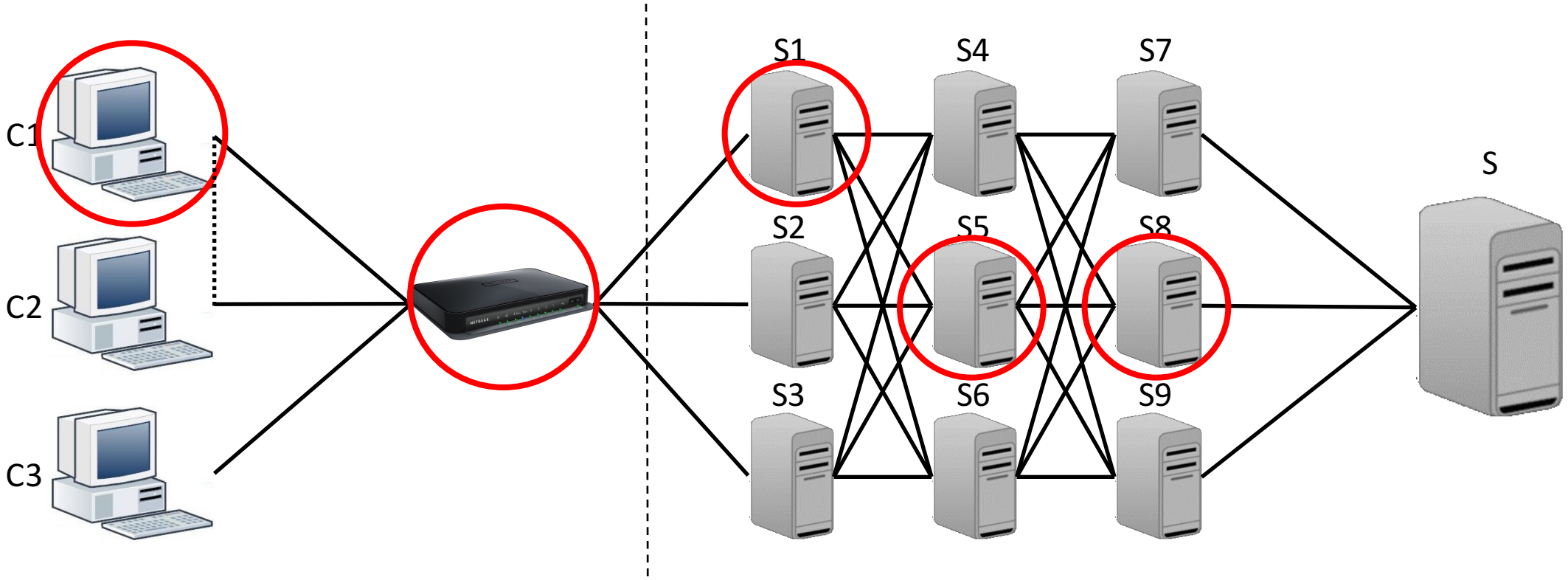
Network Security Lab – University of Trento – 2016-04-27

Jan Wolf
Amit Gupta
Ali Davanian

Section 1 - Introduction and configuration

- Introduction & configuration
- HTTP MitM
- HTTPS MitM
- Defenses

Introduction – MitM

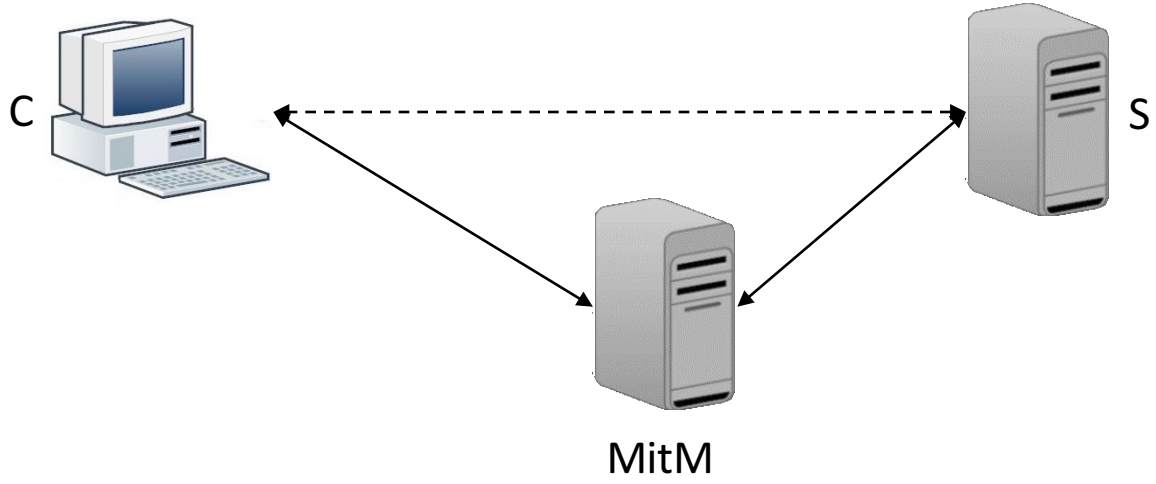


Introduction – HTTP over TLS

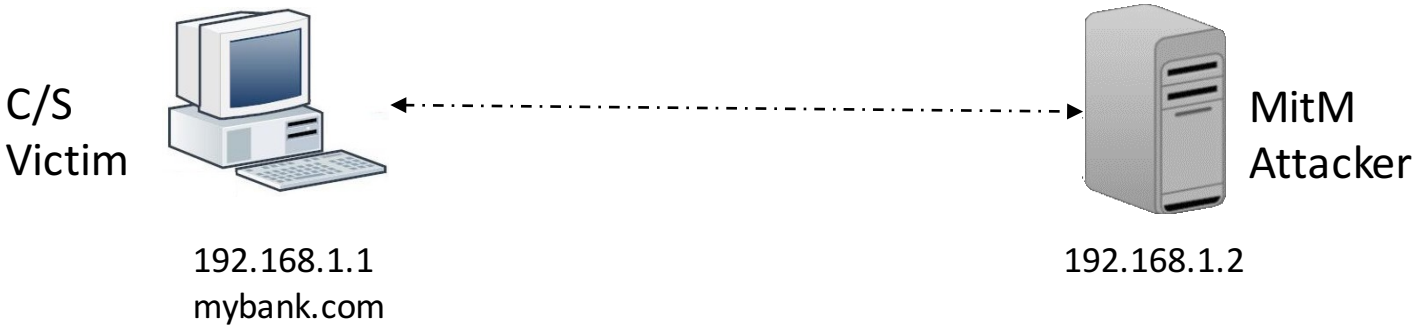
- Secure Sockets Layer (SSL)/Transport Layer Security (TLS)
  - Cryptographic protocol to secure communication channels
  - Can be added on top of most communication protocols (HTTP, FTP, SMTP, IMAP, …)
  - Symmetric cryptography for data encryption
  - Asymmetric cryptography for negotiating symmetric keys and authenticating the communication partner
    - Hierarchy-based public-key infrastructure with Certification Authorities (CAs)
    - HTTPS: Browsers/OSs come preloaded with a list of trusted root certificates, which are used to cryptographically sign intermediate certificates, which sign website certificates
    - Trust chain is verified by the browser during establishment of the secure connection (TLS handshake)
  - Integrity checks for transmitted data

Introduction – Setup (1)

Abstract setup:



Technical setup:



Introduction –Setup (2)

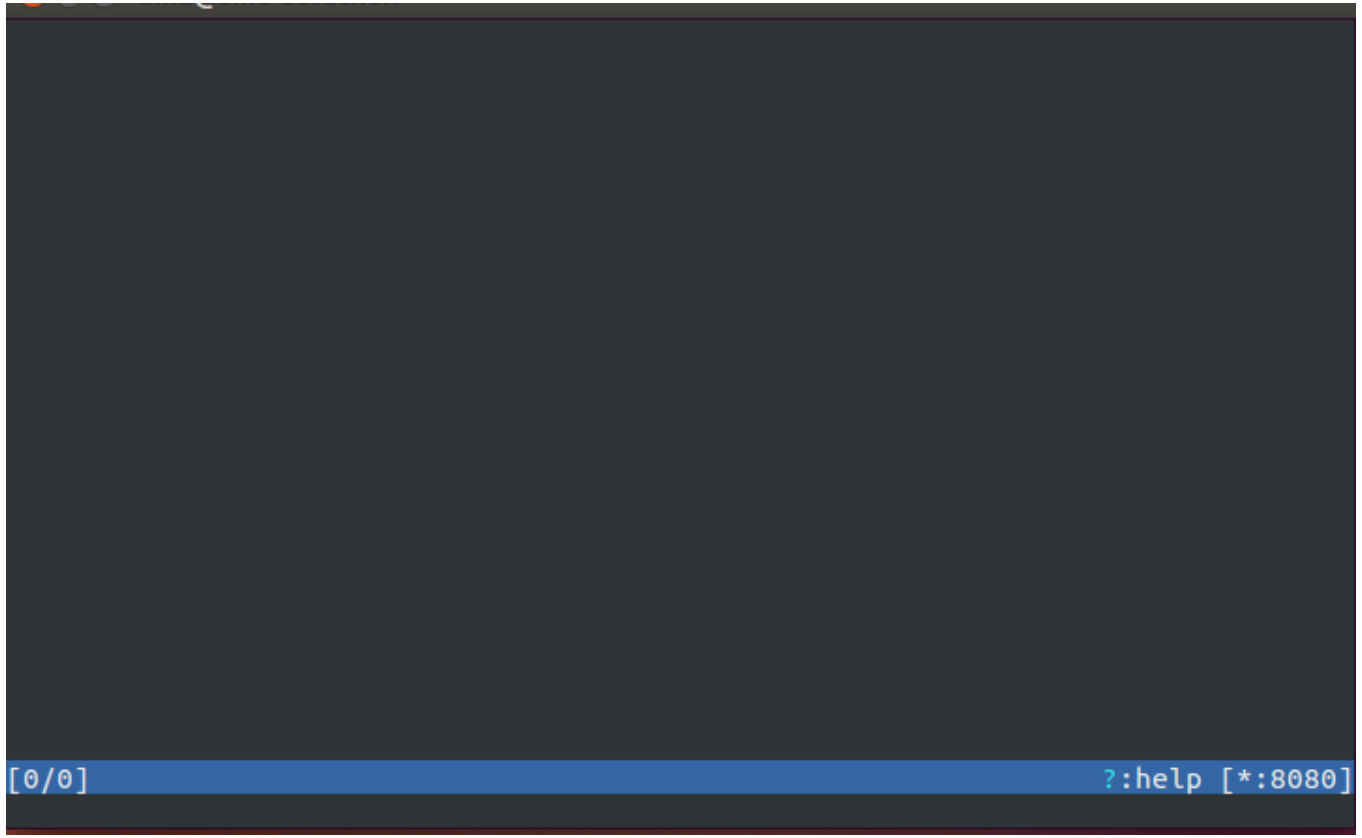
- Client/Web server (*victim*)
  - Ubuntu 14.04 Desktop
  - Apache httpd
    - “Online banking” application
  - Firefox
    - FoxyProxy
- Man in the Middle (*attacker*)
  - Ubuntu 14.04 Server
  - mitmproxy
- Laptop
  - Slides

Section 2 – HTTP MitM

- Introduction and configuration
- HTTP MitM
  - Passive attack
- HTTPS MitM
  - Problem
  - sslstrip
  - Certificate forgery
- Defenses

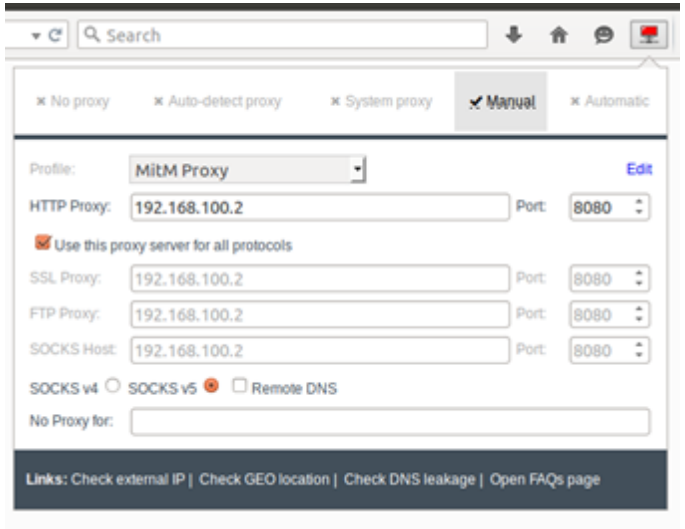
HTTP MitM passive attack – Step 1

- Open Attacker virtual machine (credentials: attacker/attacker)
- Run mitmproxy on attacker's machine (`mitmproxy`)

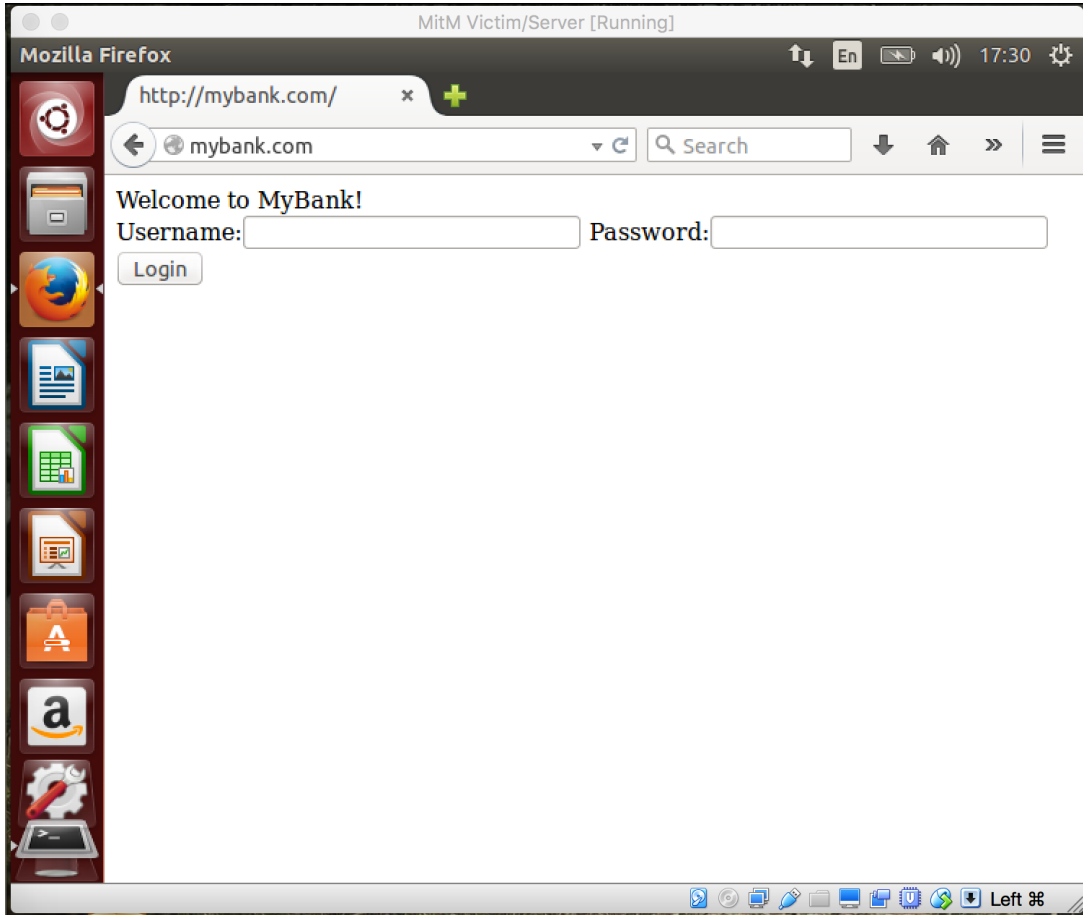


HTTP MitM passive attack – Step 2 (1)

- Open Victim virtual machine
- Open Firefox
- Activate the proxy
- Visit mybank.com and log in
  - User is user
  - Password is user



HTTP MitM passive attack – Step 2 (2)



HTTP MitM passive attack – Step 3

- Open the Attacker virtual machine
- Check details of the HTTP POST request to mybank.com and its response (including credentials)

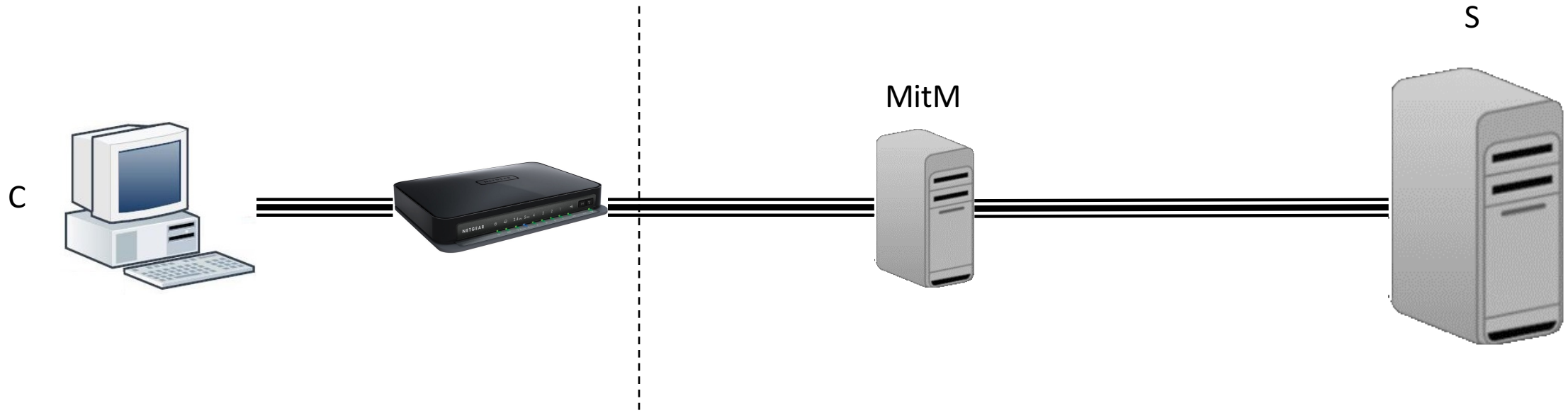
```
2016-04-18 18:09:08 POST http://mybank.com/
                         ← 302 text/html 403B 189ms
          Request                Response                Detail
Host:              mybank.com
User-Agent:        Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:39.0)
                   Gecko/20100101 Firefox/39.0
Accept:            text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=
                   0.8
Accept-Language:   en-US,en;q=0.5
Accept-Encoding:   gzip, deflate
Referer:           http://mybank.com/
Connection:        keep-alive
Content-Type:      application/x-www-form-urlencoded
Content-Length:    40
URLEncoded form                                                    [m:Auto]
username: user
password: user
submit:   Login

[2/3]                                                 ?:help q:back [*:8080]
```

Section 3 – HTTPS MitM

- Introduction and configuration
- HTTP MitM
- HTTPS MitM
  - Problem
  - sslstrip
  - Certificate forgery
- Defenses

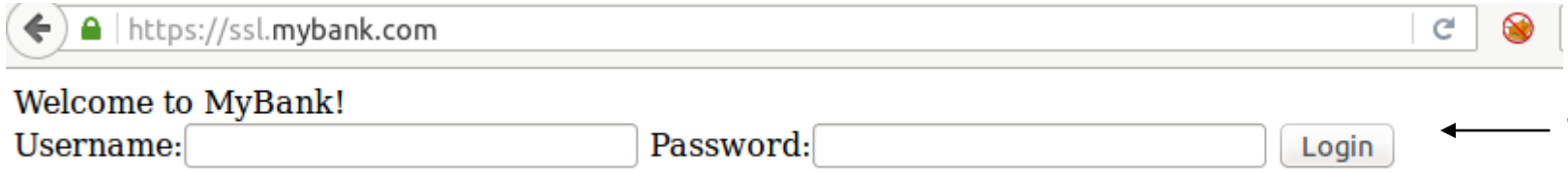
HTTPS MitM – Problem

- Encrypted protocol -> no trivial MitM possible
- Authenticated protocol -> no TLS termination possible

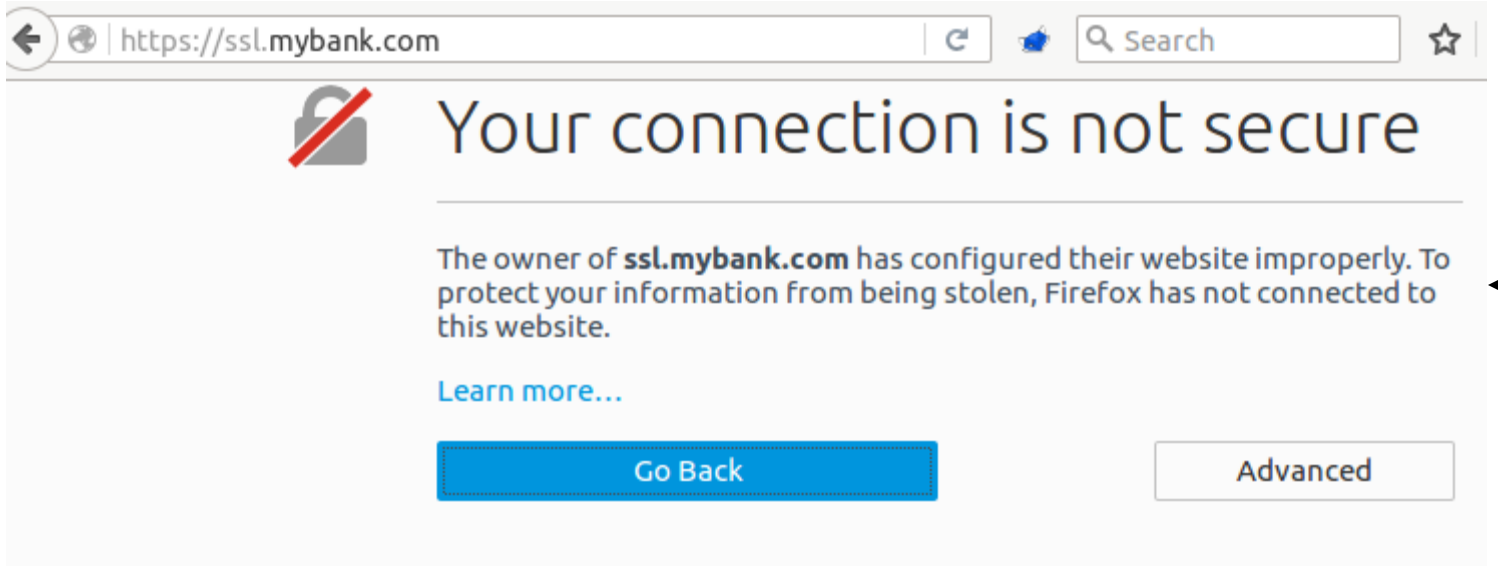


HTTPS MitM – Demonstration

- Visit **https://**ssl.mybank.com with and without proxy (Compare)



Without MitM and proxy



With MitM and proxy

Section 3 – Phase 2

- Introduction and configuration
- HTTP MitM
- HTTPS MitM
  - Problem
  - sslstrip
    - Active attack
  - Certificate forgery
- Defenses

HTTPS MitM – sslstrip

- **Problem**: HTTPS is regularly negotiated over HTTP
  - HTTP 30X redirects
  - Client-side redirect (JavaScript, meta-refresh, …)
  - Form action location
  - Links
- HTTP can be intercepted and manipulated to **prevent** establishment of **encrypted** connections

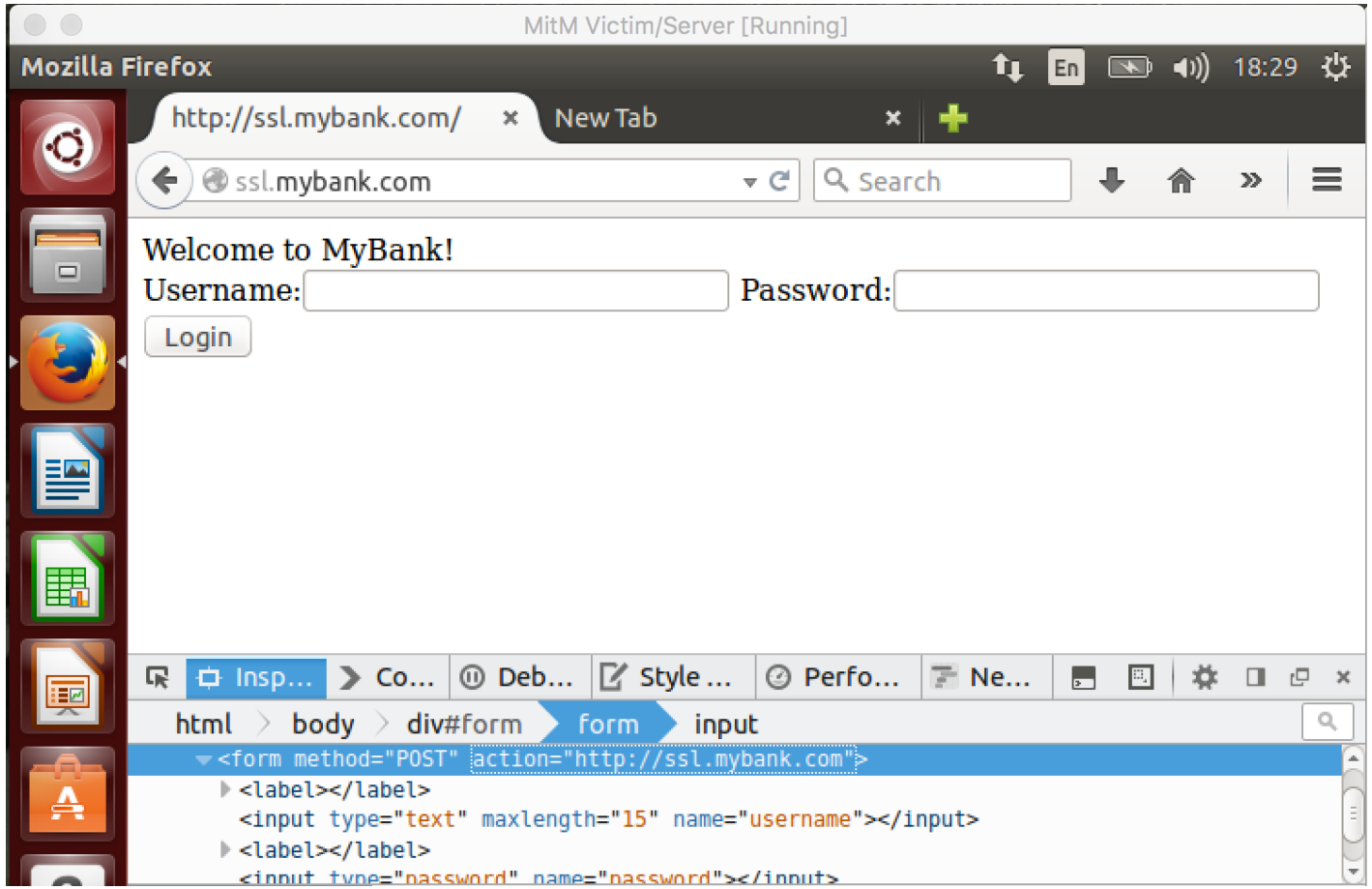
HTTPS MitM – sslstrip – Step 1

- Open the attacker virtual machine
- Stop mitmproxy by typing:
  - q
  - y
- Type `cd ~/mitmproxy/` (Tilde: Alt Gr +)
  - You should see sslstrip.py by typing `ls`
  - Start mitmproxy with sslstrip:
    - `mitmproxy -s sslstrip.py`

HTTPS MitM – sslstrip – Step 2 (1)

- Open the Victim virtual machine
- Open the browser, deactivate the proxy
- Visit ssl.mybank.com
- Check the source code using Firefox inspector (right click -> Inspect element)
- Activate the proxy, refresh the page, and compare the source code
- Log in using known credentials
- **HTTPS redirect does not happen**
- **The website is served in HTTP**
- **User will observe no error in the browser**

HTTPS MitM – sslstrip – Step 2 (2)



Section 3 – Phase 2

- Introduction and configuration
- HTTP MitM
- HTTPS MitM
  - Problem
  - sslstrip
    - Active attack
  - Certificate forgery
- Defenses

HTTPS MitM – Active Attack – Step 1

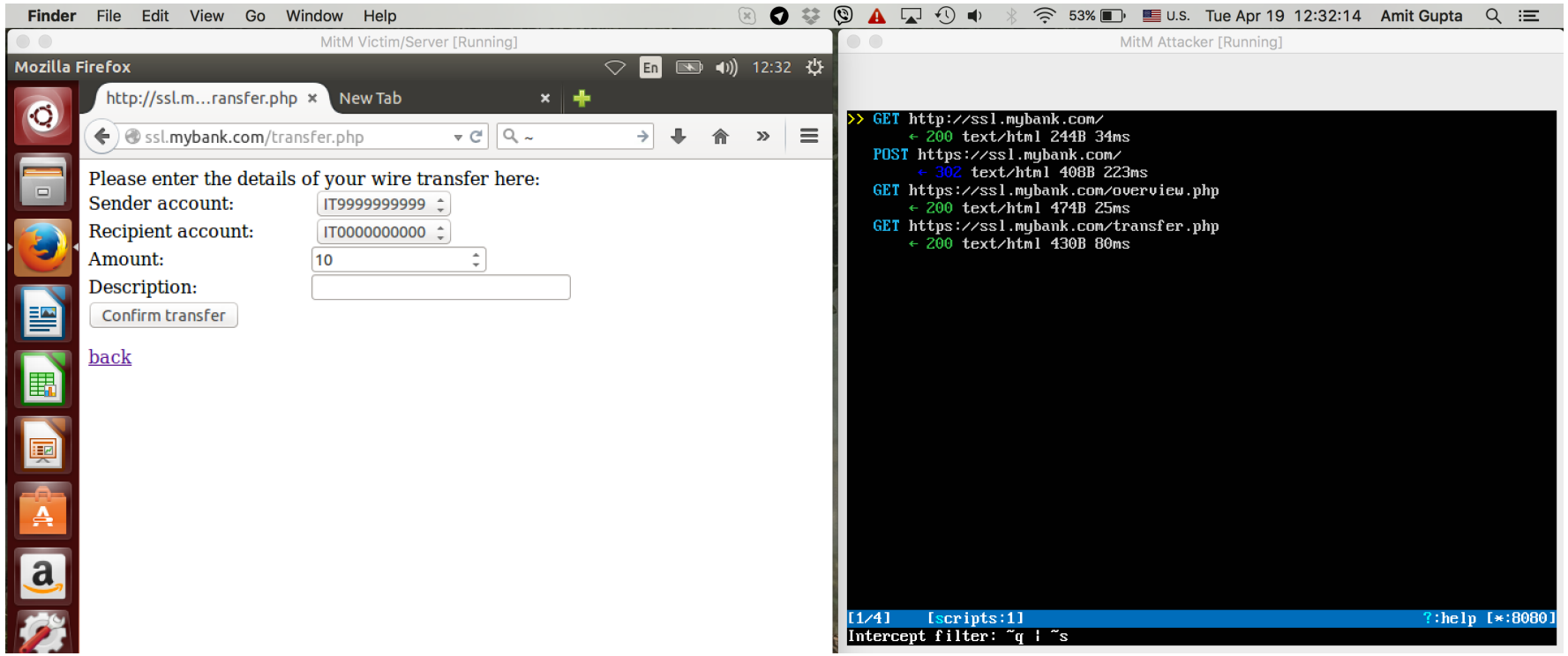
- Open the attacker virtual machine
- Press i
- Type `~q | ~s` and press Enter to activate interception for all requests and all responses

HTTPS MitM – Active Attack – Step 2

- Open Victim virtual machine
- Open the browser
- You should still be logged in to ssl.mybank.com
- Click “Wire transfer”
- Accept the request and the response by pressing `a` twice on the attacker machine

HTTPS MitM – Active Attack – Step 3

- Perform transfer of 10€ to account IT000000000



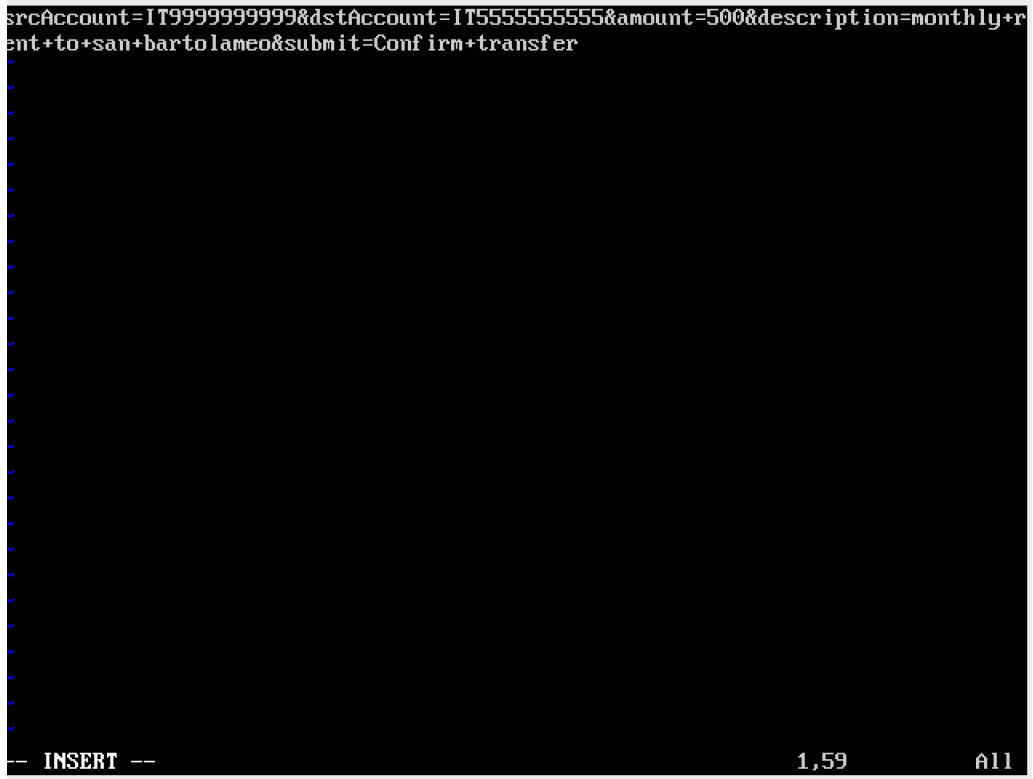
HTTPS MitM – Active Attack – Step 4 (1)

- Open the intercepted request and manipulate it
  - On the request tab press `e`
  - Press `r` afterwards, editor opens

```
2016-04-19 13:20:04 POST http://ssl.mybank.com/transfer.php

    Request intercepted          Response                    Detail
Host:             ssl.mybank.com
User-Agent:       Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:39.0)
                  Gecko/20100101 Firefox/39.0
Accept:           text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=
                  0.8
Accept-Language:  en-US,en;q=0.5
Accept-Encoding:  gzip, deflate
Referer:          http://ssl.mybank.com/transfer.php
Cookie:           PHPSESSID=tlb0md1q65om81g4dduoaftej4
Connection:       keep-alive
Content-Type:     application/x-www-form-urlencoded
Content-Length:   125
URLEncoded form                                                    [n:Auto]
srcAccount:  IT9999999999
dstAccount:  IT0000000000
amount:      290
description: monthly rent to san bartolameo
submit:      Confirm transfer

[2/2]    [i:~q | ~s][scripts:1]                    ?:help q:back [*:8080]
Edit request (cookies,query,path,url,header,form,raw body,method)?
```

HTTPS MitM – Active Attack – Step 4 (2)

- Manipulate the request as to transfer 500€ to account IT5555555555
  - Replace account IT0000000000 by IT5555555555
  - Replace amount by 500
  - Press CTRL+X to exit
  - Save changes (y) to default file



HTTPS MitM – Active Attack – Step 5 (1)

- Press `a` to accept the manipulated request
- Press `Tab` to go to the response tab
- On the response tab, press `e`
- Press `r`, editor opens

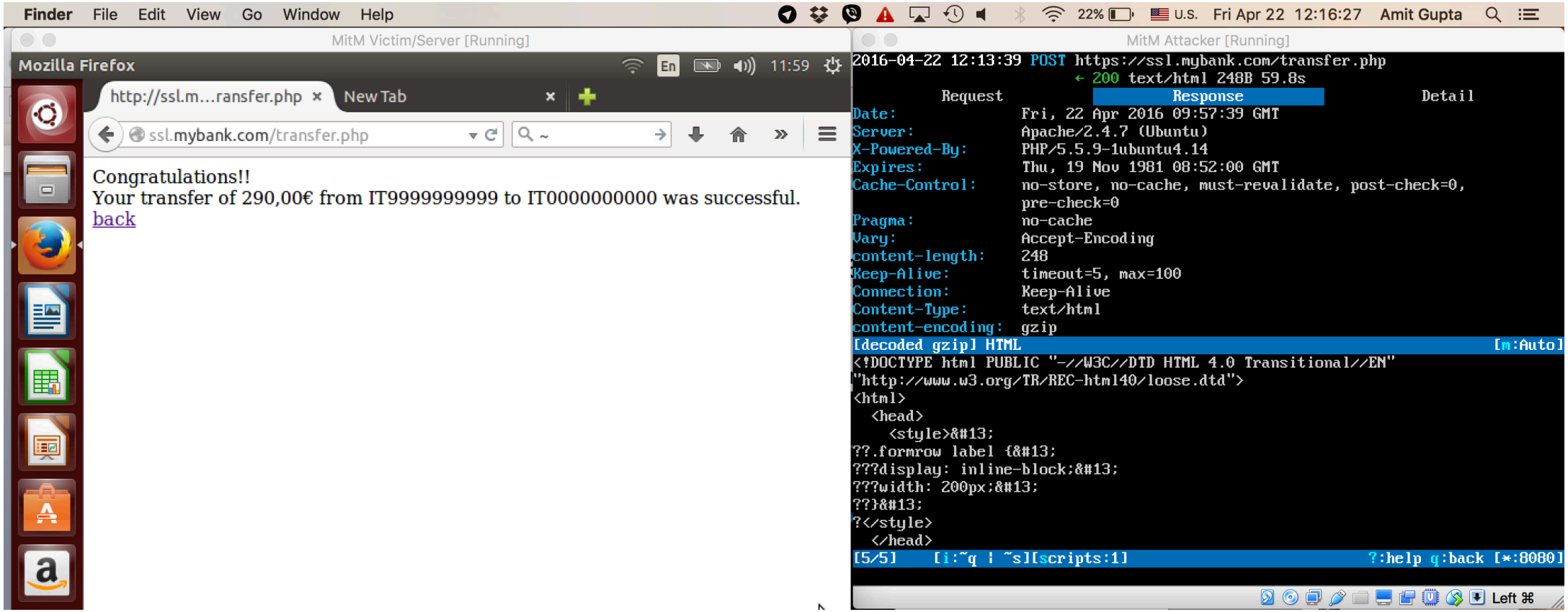
```
2016-04-19 13:20:04 POST http://ssl.mybank.com/transfer.php
                        ← 200 text/html 264B 140s
          Request          Response intercepted          Detail
Date:               Tue, 19 Apr 2016 11:22:26 GMT
Server:             Apache/2.4.7 (Ubuntu)
X-Powered-By:       PHP/5.5.9-1ubuntu4.14
Expires:            Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control:      no-store, no-cache, must-revalidate, post-check=0,
                    pre-check=0
Pragma:             no-cache
Vary:               Accept-Encoding
content-length:     264
Keep-Alive:         timeout=5, max=100
Connection:         Keep-Alive
Content-Type:       text/html
content-encoding:   gzip
[decoded gzip] HTML                                                [m:Auto]
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"
"http://www.w3.org/TR/REC-html40/loose.dtd">
<html>
  <head>
    <style>&#13;
??.formrow label {&#13;
???display: inline-block;&#13;
???width: 200px;&#13;
??}&#13;
?</style>
  </head>
[2/2]    [i:~q | ~s][scripts:1]                    ?:help q:back [*:8080]
```

HTTPS MitM – Active Attack – Step 5 (2)

- Deceive the user
  - Replace account IT5555555555 by IT0000000000
  - Replace amount by original amount (default 10)
  - Press CTRL+X to exit,
  - Save changes (`y`) to default file
  - Accept the response by pressing `a`
  - Press `q` to leave the detail view
- Press `i`, delete the current intercept filter and press Enter

```
<html>
<head>
        <style>
                .formrow label {
                        display: inline-block;
                        width: 200px;
                }
        </style>
</head>

<body>
        <div id="welcome">
                Thank You!
        </div>
        <div class="success">Your transfer of 290,00&euro; from IT9999999999 to
IT0000000000 was successful.<br /><a href="overview.php">back</a></div></body></
html>
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                                    13,13-27      All
```

HTTPS MitM – Active Attack – Step 5 (3)

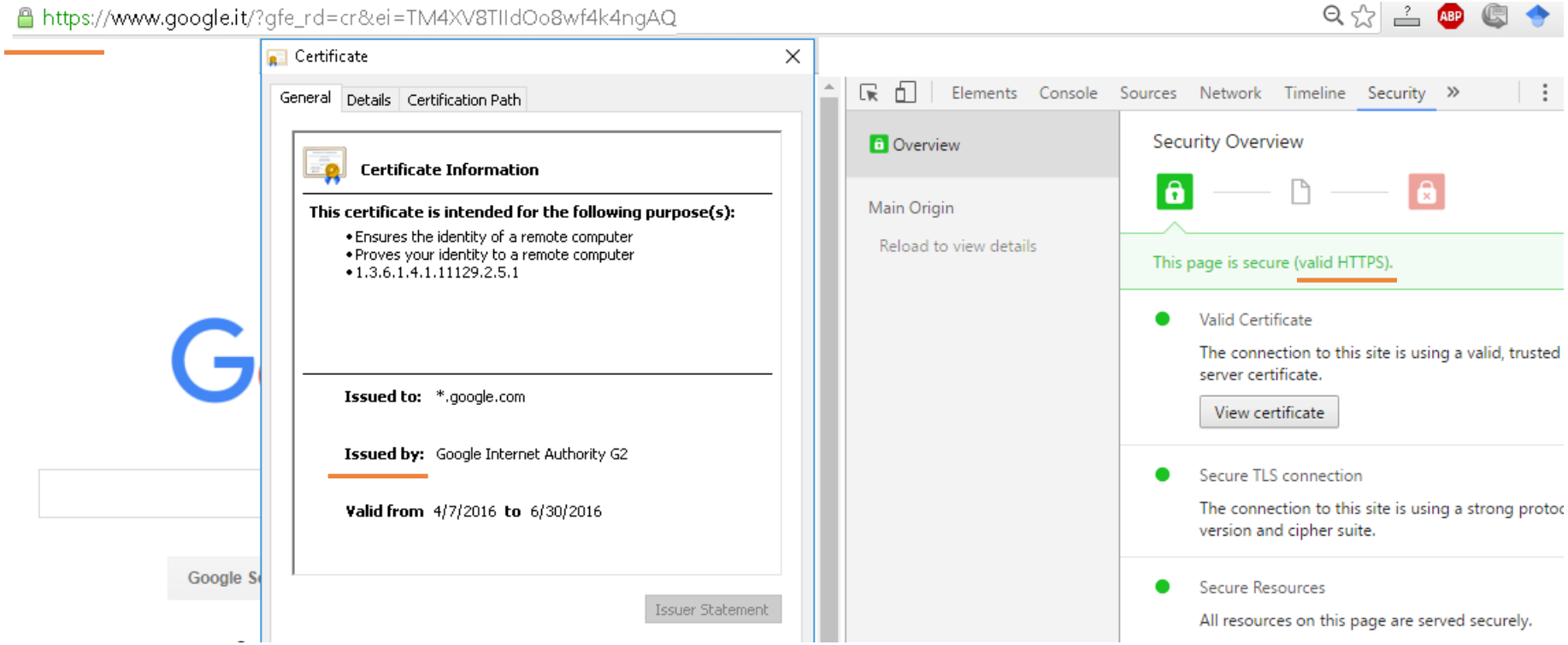


Section 3 – Phase 3

- Introduction and configuration
- HTTP MitM
- HTTPS MitM
  - Problem
  - sslstrip
  - Certificate forgery
- Defenses

HTTPS MitM – Certificate forgery (1)

- What is a signature?
  - The signature proves the authenticity of the certificate



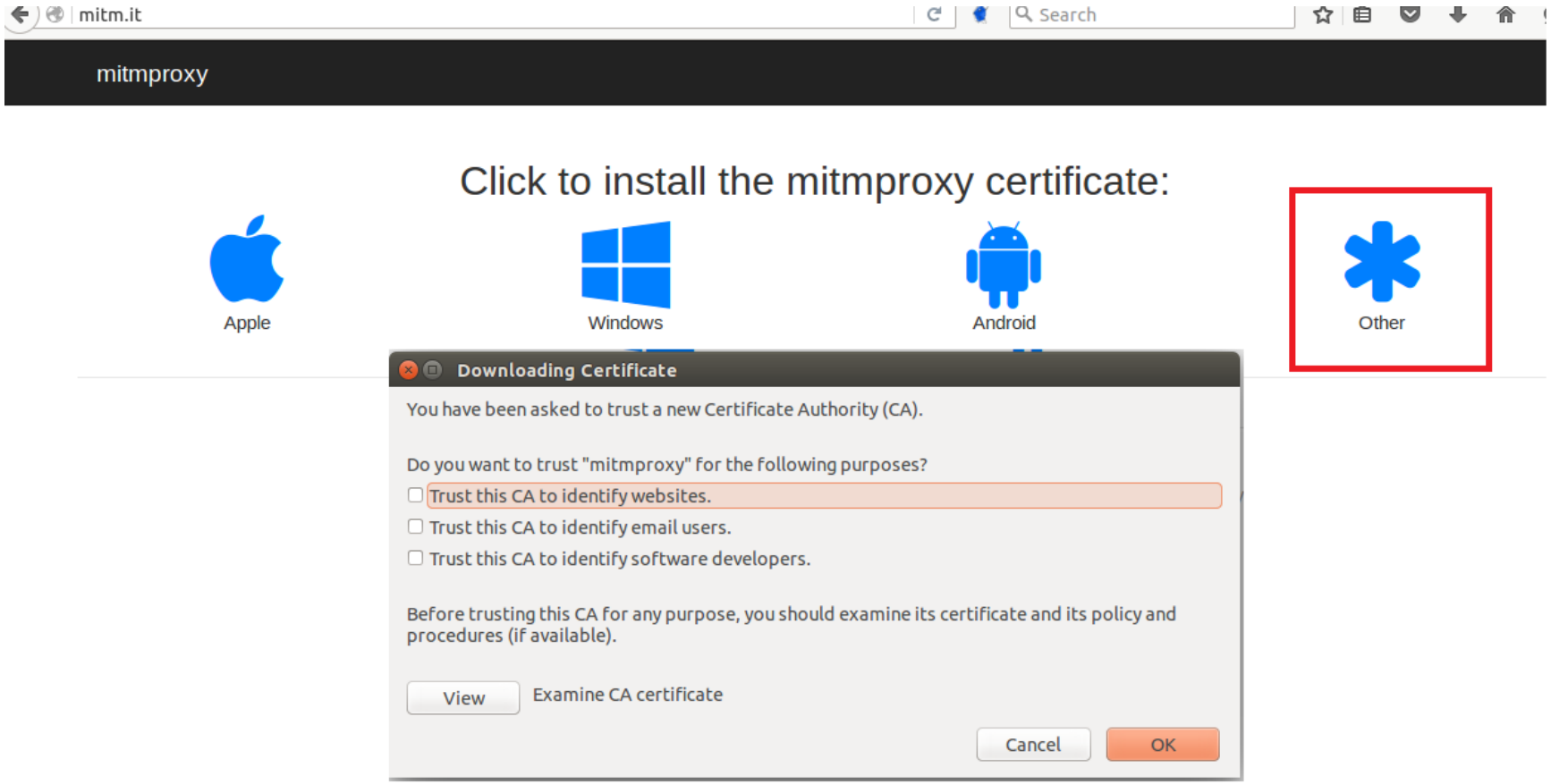
HTTPS MitM – Certificate forgery (2)

- Authentic certificate
  - If the certificate authority's signature is in your computer
- Forged certificate
  - Unknown signer -> error in your browser
- Am I secure if I don't see the error?
  - Rogue CA might be listed as trustworthy by your computer
    - Lenovo Superfish example from class
- We do the same here and install the certificate authority manually

HTTPS MitM – Certificate forgery – Step 1 (1)

- Open the victim virtual machine
- Open the browser
- Make sure the proxy is set
- Open mitm.it
- Choose “other”
- Check the first box and click ok

HTTPS MitM – Certificate forgery – Step 1 (2)



HTTPS MitM – Certificate forgery – Step 2

- Recall the earlier error message when visiting https://ssl.mybank.com while using the proxy
- Visit **https://**ssl.mybank.com again while the proxy is active

Section 4 - Defenses

- Introduction and configuration
- HTTP MitM
- HTTPS MitM
- Defenses

HTTP Strict Transport Security (HSTS)

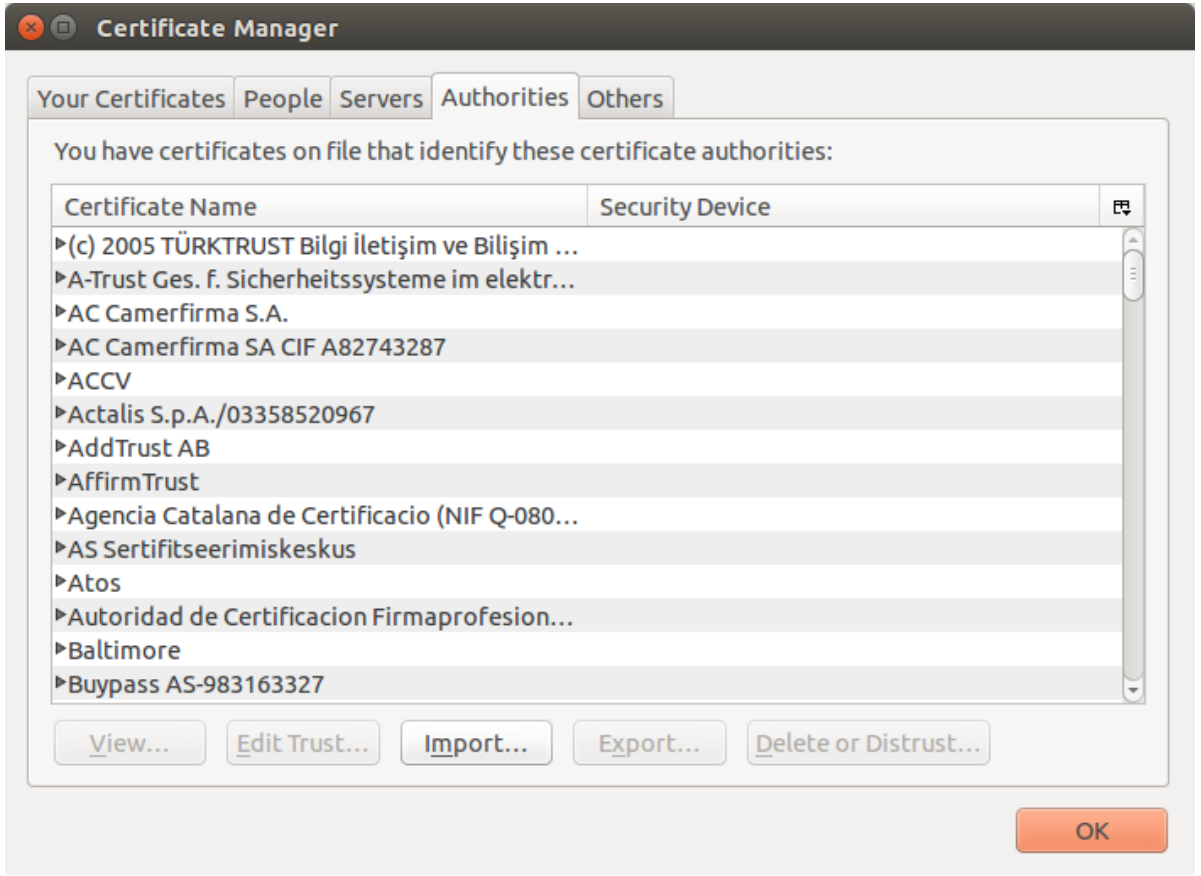
- HTTP header codified in RFC 6797 (Nov 2012)
- “TLS Supercookie”
- Based on Trust-on-First-Use model
  - User visits HTTPS website
  - Server responds with HSTS header, indicating a time period
  - Browser stores this information and will reject all non-HTTPS connections to this domain
- Browser preload possible

HTTP Public Key Pinning

- HTTP header codified in RFC 7469 (Apr 2015)
- Also called Certificate Pinning
- Based on Trust-on-First-Use model
  - User visits HTTPS website
  - Server responds with HPKP header, indicating
    - the SHA-256 hash of its public key,
    - the SHA-256 hash of a backup public key,
    - a time period
  - Browser stores this information and will reject all HTTPS connections to this domain if the presented public key does not match
- Browser preload for popular websites

…and of course

- Don’t trust unknown hotspots
- Certainly don’t trust unknown certificates
- There’s no way you’re going to trust an unknown Certification Authority
  - …right?



References

- TLS: RFC 5246 (https://tools.ietf.org/html/rfc5246)
- HSTS: RFC 6797 (https://tools.ietf.org/html/rfc6797)
- HPKP: RFC 7469 (https://tools.ietf.org/html/rfc7469)
- sslstrip: native software (https://moxie.org/software/sslstrip/) and original Blackhat talk (https://www.youtube.com/watch?v=MFol6IMbZ7Y), both by Moxie Marlinspike
- mitmproxy: https://mitmproxy.org/