Spatti

Securing public access to the Internet

Harm Jan Blok, Marya Steenman, Carlos Groen en Wouter Borremans

10 juni 2005

Inhoudsopgave

1 Project Spatti

1.1 Achtergrond

Tegenwoordig kunnen we bijna niet meer zonder Internet verbinding, sterker nog: we zouden graag willen dat we altijd en overal een internetaansluiting tot onze beschikking hadden. Je ziet steeds meer dat bedrijven zich bezig gaan houden met het opzetten van Hotspots of soortgelijke projecten.
Het nadeel van dit soort projecten is dat ze niet vrij toegankelijk zijn. Er moet vaak op een of andere manier toegang verkregen worden, bijvoorbeeld door het opgeven van een wachtwoord. Als het verkrijgen van toegang zich zou beperken tot het opgeven van een wachtwoord dan zou dat mooi zijn, maar helaas is dit niet het geval. Veel verbindingen zijn beveiligd met wachtwoorden, encryptie, adresbeperkingen etc. Dit heeft als gevolg dat het voor veel gebruikers lastig wordt snel een verbinding op te zetten en hier gebruik van te maken.

1.2 Project omschrijving

In dit project willen wij de mogelijkheden bekijken voor het opzetten van een netwerk dat voor de gebruikers makkelijk toegankelijk is. Dit betekent weinig configuratie aan netwerk- en security instellingen.

Daarnaast moet het voor de provider van het netwerk mogelijk zijn inzicht te krijgen in, en grip te hebben op, het netwerkverkeer zodat misbruik voorkomen kan worden.

1.3 Onderzoeksvragen

Om het project te kunnen realiseren zullen een aantal vragen beantwoord moeten worden. Door in een vooronderzoek antwoord te geven op deze vragen kunnen wij een testomgeving opzetten aan de hand waarvan wij aanbevelingen kunnen doen over de haalbaarheid van het project wanneer deze publiekelijk uitgerold zou gaan worden. De vragen die wij willen beantwoorden zijn:

- Wel of geen encryptie?
- Anonieme toegang of accounting?
- Volledig gebruik of alleen bepaalde diensten?
- Welke infrastructuur?
- Wat gaan we meten?
- Hoe reguleren we het netwerk verkeer?
- Wat te doen bij misbruik?

1.4 Doelstelling van het project

De doelstelling van dit project is een gefundeerde aanbeveling doen over het opzetten van een veilige publieke internet-infrastructuur.

2 Vooronderzoek

In dit hoofdstuk worden de resultaten besproken van het gedane vooronderzoek. In dit vooronderzoek worden zo veel mogelijk de vragen uit voorgaand hoofdstuk beantwoord.

2.1 Encryptie

Om het accespoint een open karakter te geven hebben we besloten standaard geen gebruik te maken van de beschikbare wireless encryptietechnieken als WEP, WPA of 802.1x. Voor deze encryptievormen is namelijk kennis van een sleutel nodig (WEP en WPA) of een certificaat en/of een gebruikersnaam/wachtwoord combinatie (802.1x) en het instellen ervan vereist handmatige configuratie. Mocht blijken dat er toch behoefte is aan encryptie dan staan verderop in dit document een tweetal mogelijke oplossingen die naast de open infrastructuur geïmplementeerd kunnen worden.

2.2 Services

De services die we ter beschikking stellen op het netwerk zijn diegene waarvan het meest gebruikt wordt op het Internet:

- Web
- E–mail
- Filetransfer

Van deze services zullen alleen de standaard protocollen gebruikt kunnen worden en niet diegene die voor een beveiligde verbinding kunnen zorgen. De reden voor deze keuze is dat bij het gebruik van de versleutelde varianten het moeilijk wordt inzicht te krijgen in het verkeer dat gegenereerd wordt door (via) deze protocollen, daar dit een risico met zich meebrengt voor de provider van het netwerk. De protocollen die beschikbaar zijn op het netwerk zijn:

- HTTP
- SMTP
- POP3
- IMAP
- FTP

Verder hebben deze protocollen voor hun functioneren de diensten van andere services nodig: daarvoor worden DNS en DHCP ook toegelaten op het netwerk. Daarnaast kunnen vanaf het vrij toegankelijke netwerk wel verbindingen voor deze protocollen opgezet worden naar het Internet maar niet andersom, het is daarom bijvoorbeeld niet mogelijk een webserver ter beschikking te stellen via dit netwerk.

2.3 Bedreigingen

Gebruikers die gebruik maken van het vrij toegankelijke netwerk zouden via dit netwerk misbruik kunnen maken via verschillende protocollen om zo bij anderen schade toe te brengen.
In deze paragraaf worden van de bovengenoemde protocollen de mogelijke bedreigingen en misbruiken beschreven, daarnaast wordt kort een mogelijke oplossing om deze bedreigingen tegen te gaan genoemd.

2.3.1 Misbruik HTTP

Het HTTP protocol, gebruikt om alledaags Internetgebruik mogelijk te maken, wordt vaak misbruikt om andere partijen schade toe te brengen. In deze paragraaf bespreken we de meest voorkomende misbruiken van het HTTP protocol.

Misbruik door de gebruiker

- Packet Injection[1]
- Phishing[2]
- Tunneling[3]
- Virus verspreiding

Oplossing: Controleren op veel voorkomende hacks (met bijvoorbeeld een inline IDS). Verder zorgt ratelimiting of trafficshaping om restrictie te leggen aan de hoeveelheid verkeer dat mogelijk is via het HTTP protocol. Virusscannen via een transparante proxy is een mogelijkheid om virusverspreiding tegen te gaan.

Bedreiging voor de gebruikers (intern)

- Man in the middle[4]
- Phishing
- Session Hijacking[5]

Oplossing: HTTPS of encryptie met authenticatie tussen de client en de gateway/proxy.

Bedreiging voor de gebruikers (extern)

- Phishing

[1] Injecteren van foutieve pakketten in de datastroom
[2] Het imiteren van een bestaande website, die vaak veel bezoekers krijgen, om vertrouwelijke gegevens te onderscheppen.
[3] Data encapsuleren in pakketten van een ander protocol
[4] Een derde partij doet zich voor als de communicerende partij (bijvoorbeeld een server)
[5] Een (TCP) sessie wordt overgenomen door een derde partij (Hacker)

- Virus ontvangst

Oplossing: Zoals al eerder genoemd worden verbinding vanaf het Internet naar het interne net geblokkeerd. Voor HTTP betekent dit dat inkomende connecties niet mogelijk zijn en uitgaande connecties statefull zijn. Door het verkeer via een (transparante) proxy te laten lopen (er kan niet direct met het Internet gecommuniceerd worden maar alleen via de proxy) is controle op het verkeer mogelijk. Het scannen op virussen moet het verspreiden van virussen via/op dit netwerk tegengaan.

2.3.2 Misbruik SMTP

Misbruik door de gebruiker

- Spam
- Hacks uitvoeren op server
- Tunnelen
- Virusverspreiding (kan alleen unfragmented gecontroleerd worden)

Oplossing Een limiet op het aantal SMTP connecties dat opgezet kan worden. Verder virusscanning en een controle op bekende hacks via SMTP (bijvoorbeeld door gebruik van een inline IDS).

Bedreiging voor de gebruikers (intern)

- Sniffen[6]
- Sniffen van berichten
- Session hijacking

Oplossing Secure SMTP of encryptie met authenticatie tussen de client en de gateway/proxy.

Bedreiging voor de gebruikers (extern) Het is niet noodzakelijk voor de draadloze gebruiker om email via SMTP te ontvangen. We blokkeren daarom SMTP verkeer naar de gebruiker toe.

Oplossing Uitgaande SMTP verbindingen toestaan en inkomende SMTP verbindingen blokkeren.

[6] Door pakketten op het netwerk te bekijken kan informatie (zoals wachtwoorden) achterhaald worden

2.3.3 Misbruik POP3

Misbruik door de gebruiker

- Tunnelen
- Hacks uitvoeren op server

Oplossing Controleren op bekende hacks (inline IDS)

Bedreiging voor de gebruikers (intern)

- Sniffen van wachtwoorden
- Sniffen van berichten
- Session hijacking

Oplossing Secure POP of encryptie met authenticatie tussen de client en de gateway/proxy.

Bedreiging voor de gebruikers (extern)

- Virus ontvangst

Oplossing Het is niet noodzakelijk inkomende POP3 verbindingen te accepteren. Uitgaande POP3 verbindingen toestaan en inkomende POP3 verbindingen blokkeren. Daarnaast inline virusscannen of met behulp van een proxy.

2.3.4 Misbruik IMAP

Misbruik door de gebruiker

- Tunneling
- Hacks uitvoeren op server

Oplossing Check op bekende hacks (inline IDS)

Bedreiging voor de gebruikers (intern)

- Sniffen van wachtwoorden
- Sniffen van berichten
- Session hijacking

Oplossing Secure IMAP of encryptie met authenticatie tussen de client en de gateway/proxy.

Bedreiging voor de gebruikers (extern)

- Virus ontvangst

Oplossing Het is niet noodzakelijk inkomende IMAP verbindingen te accepteren. Uitgaande IMAP verbindingen toestaan en inkomende IMAP verbindingen blokkeren. Daarnaast inline virusscannen of met behulp van een proxy.

2.3.5 Misbruik FTP

Misbruik door de gebruiker

- Hacks
- Tunnelen
- Virusverspreiding

Oplossing Controleren op bekende hacks (met bijvoorbeeld een inline IDS) Rate limiting om het aantal connecties te beperken, virusscannen inline of via transparante proxy.

Bedreiging voor de gebruikers (intern)

- Man in the middle
- Session Hijacking

Oplossing Secure FTP of encryptie met authenticatie tussen de client en de gateway/proxy.

Bedreiging voor de gebruikers (extern)

- Virus ontvangst

Oplossing Het is niet noodzakelijk inkomende FTP verbindingen te accepteren. Uitgaande FTP verbindingen toestaan en inkomende FTP verbindingen blokkeren. Daarnaast inline virusscannen of met behulp van een proxy.

2.3.6 Conclusie

Hierboven zijn een aantal mogelijke (en redelijke) dreigingen genoemd waarmee de verschillende protocollen die we aan willen bieden te maken kunnen krijgen. De oplossingen die hierboven genoemd zijn kunnen de problemen oplossen, maar voor sommige oplossingen is bewust gekozen (op voorhand) deze niet te implementeren. Een voorbeeld hiervan is het gebruik van encryptie. Dit zal verschillende problemen oplossen, maar voor de provider van het netwerk brengt het weer extra risico met zich mee omdat het verkeer niet te controleren is.

2.4 Gelijksoortige projecten

Om een beeld te vormen van wat er reeds bestaat op het gebied van public accesspoints hebben we gezocht naar soortgelijke projecten en documenten over dit onderwerp. Hieronder een kleine opsomming van de meest veelbelovende.

ZoneCD

Live CD gebaseerd op Morhpix/Knoppix. Hoeft niet geinstalleerd te worden maar draait direct vanaf CD. Heeft veel configuratieopties zoals authenticatie en content filtering. Kan gebruikt worden in Open mode (een idee zoals spatti) en Closed mode (met authenticatie en limitering).

http://www.publicip.net/

Secure Wireless Access in Public Places - The CHOICE network

Het CHOICE netwerk is een onderzoeksproject van de Microsoft Research group en is gebaseerd op een werelwijd authenticatie systeem (denk bijvoorbeeld aan Microsoft Passport) om gebruikers op een veilige manier toegang te geven tot een draadloze breedband verbinding. Het CHOICE systeem kan zowel in zakelijke omgevingen worden gebruikt als in niet commerciële omgevingen. Het systeem is gebaseerd op een softwaremodule genaamd PANS (Protocol for Authorization and Negotiation of Services).

CHOICE biedt mensen de mogelijkheid tot breedband Internet toegang mits ze zich juist hebben aangemeld. In combinatie met het PANS protocol biedt CHOICE authenticatie, accounting, bedrijfsbeleid, QoS en accounting voor netwerk operators. Een CHOICE implementatie is opgebouwd uit een aantal systemen welke de bovenstaande taken uitvoeren. Door de koppeling van meerdere systemen onstaat een situatie waarbij er een relatief grote kans is dat een van de componenten faalt. Het CHOICE systeem bestaat onder andere uit:

- Global Authenticator

  Gebruikt MS Passport om gebruikers te authenticeren.

- DHCP Server

  Wijst gebruikers dynamisch een IP adres toe bij juiste authenticatie.

- PANS Authorizer

  Authenticeert de gebruiker voor het gebruik van het netwerk. (Gebruik van services etc.) Tevens zorgt de authorizer voor IP level filtering op basis van een key en token paar.

- PANS Verifyer

  Zorgt voor accounting en filtering op packet niveau.

- PANS Client

  Zorgt voor traffic tagging door middel van het key en token paar dat is toegewezen door de PANS Authorizer

http://www-cse.ucsd.edu/users/abalacha/research/talks/ICC_Presentation.pdf
http://research.microsoft.com/users/bahl/papers/pdf/icc01.pdf

3 Onderzochte technieken

3.1 Honeywall

Het controleren of een gebruiker probeert andere computersystemen op het Internet te hacken kan met behulp van een inline Intrusion Detection Systeem. Een inline Intrusion Detection Systeem kijkt niet alleen of er kwaadaardig verkeer langskomt maar houdt het kwaadaardige verkeer ook daadwerkelijk tegen. Deze functionaliteit staat ook wel bekend als honeywall.

Om het kwaadaardige verkeer te kunnen filteren moet de honeywall tussen de gebruikers en Het Internet geplaatst worden. Er zijn twee technische mogelijkheden om dit te realiseren.

Bridge De honeywall kan opgezet worden als bridge waarbij al het verkeer dat van de gebruikers komt, door de bridge wordt doorgestuurd naar het inline IDS. Het inline IDS controleert dit verkeer op bekende hacks en filtert ongewenst verkeer er tussen uit. Al het overige verkeer laat het inline IDS doorgaan naar Het Internet. Ook het terugkomende verkeer kan op deze manier gecontroleerd worden. Het voordeel van deze oplossing is dat het transparant is voor de gebruiker.

Nat Een andere mogelijkheid is om de honeywall op te zetten als NAT gateway. Ook hierbij filtert het inline IDS het ongewenste verkeer eruit en laat het overige verkeer doorgaan. Deze oplossing is niet transparant voor de gebruiker, de gebruiker moet namelijk de NAT machine als gateway instellen.

3.2 Proxy

Met een proxy wordt in dit geval een NAT-machine bedoeld die luistert op de, door de gewenste services gebruikte, poorten en hierop requests accepteert. Deze requests worden vervolgens verwerkt en eventuele benodigde informatie wordt door de proxy server opgehaald van het Internet. Er is dus geen sprake van directe communicatie tussen de clients (de applicaties op de systemen van de gebruikers) en externe servers maar al het verkeer loopt via gecontroleerde processen op de proxy machine.

Doel

Het doel van een proxyserver binnen dit project is:

- Ervoor zorgen dat alleen er alleen 'geldig' verkeer plaatsvindt (verkeer op de juiste poorten en volgens de geldende protocollen behorende bij de aangeboden service).
- De mogelijkheid bieden om het verkeer te scannen op de aanwezigheid van virussen en ongeldige content.

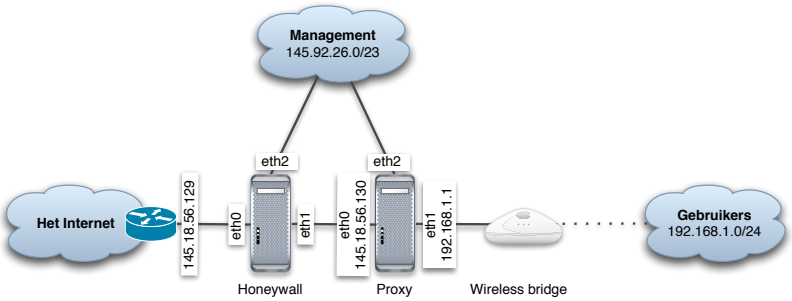
- De mogelijkheid bieden verkeer te limiteren op basis van grootte en/of aantal requests per tijdseenheid.
- Voorkomen dat de clients direct bereikbaar zijn vanaf het Internet (NAT).
- Het aanbieden van een welkomstpagina met spelregels en aanvullende informatie.

3.3 Smart Accesspoint

De huidige generatie accespoints en routers biedt diverse mogelijkheden met betrekking tot uitbreidingen. Zo zijn er voor de door ons gebruikte router, de Linksys WRT54G, diverse niet-originele firmwares verkrijgbaar. Zo is er OpenWRT welke, naast een uitgebreidere webinterface, een simpele embedded linux omgeving biedt met implementaties van IPtables en servers als DHCP en een web/ftp server. Het is mogelijk hiervoor ook andere applicaties te compileren waardoor eventueel de functionaliteit van voorgenoemde proxyserver op het accespoint zelf geïmplementeerd kan worden. Een beperking hierbij is de kleine hoeveelheid geheugen, hoewel er ook accespoint zijn die de mogelijkheid bieden USB sticks als werkgeheugen te gebruiken (de ASUS WL500G bijvoorbeeld). Hoewel we binnen het project te weinig tijd hadden om hier uitgebreid onderzoek naar te doen is het, in het licht van een alles-in-een oplossing, interessant hier in de toekomst eens naar te kijken.

4 Test infrastructuur

Om een aantal van de onderzochte technieken te testen is een test-infrastructuur opgezet. Deze test infrastructuur bestaat uit een honeywall, een proxy en een wireless bridge, zie figuur 1.



Figuur 1: Test infrastructuur.

4.1 Honeywall

4.1.1 Installatie

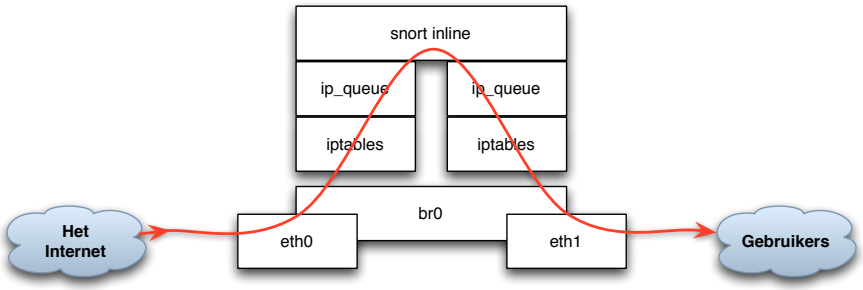
In eerste instantie hebben we er niet voor gekozen om de kant en klare honeywall CD te gebruiken, maar om zelf een honeywall samen te stellen. We wilden de honeywall zelf samenstellen omdat we het idee hadden om later nog meer diensten op de honeywall te installeren (o.a. de transparante proxy).

We hebben ervoor gekozen om als inline IDS Snort te gebruiken. Snort is het meest bekende (open source) IDS, ook is Snort één van de weinige Intrusion Detection Systemen die ook pakketen uit de datastroom kunnen filteren. Om al het binnenkomende verkeer om te leiden naar Snort hebben we gebruik gemaakt van iptables.

IPtables bestaat uit drie zogenaamde tabellen, in deze tabellen zijn een aantal chains gedefinieerd. Deze chains bevinden zich op verschillende punten op de route van een pakket door het systeem. Sommige chains ontvangen alleen pakketten die voor het systeem zelf bedoeld zijn, sommige chains ontvangen alleen pakketten die doorgestuurd worden (door bridging of routering).

We hebben de honeywall opgezet als transparante bridge, hiervoor hebben drie netwerkinterfaces gebruikt. Twee netwerkinterfaces werden opgezet als bridge, één netwerkinterface werkt gebruikt als managementinterface. In bijlage A.1 staat uitgelegd hoe je een Linux systeem als bridge opzet.

Na de installatie van Snort (zie bijlage A.3 moest het verkeer nog doorgestuurd worden naar Snort. Dit hebben we gedaan met behulp van iptables in combinatie met ip_queue. Ip_queue maakt het mogelijk om verkeer uit kernelspace (iptables) door te sturen naar userspace (Snort inline). In iptables geven we aan welke verkeer er moet worden doorgestuurd naar Snort. Hiervoor worden



Figuur 2: Verkeersflow door de transparante bridge.

standaard iptables-rules gebruikt maar dan met als 'target' QUEUE. We hebben ervoor gekozen om het verkeer naar Snort te sturen vanuit de FORWARD-chain in de 'filter'-table. Dit hebben gedaan omdat al het verkeer dat gebridged wordt (dus van de gebruikers naar Het Internet wordt doorgestuurd) hier langs komt. In bijlage A.2 staan de iptables-rules die hiervoor zorgen. In figuur 2 is te zien hoe de flow van de pakketten is.

4.1.2 Testen

Na het installeren van de honeywall hebben we de honeywall getest. Dit hebben gedaan door een (test)gebruiker een bekende hack uit te laten voeren op een webserver. Uit deze test bleek dat Snort het kwaadaardige verkeer zag maar het niet uit de datastroom filterde. We hebben meerdere dagen geprobeerd dit probleem op te lossen. We hebben onder andere de Snort rules aangepast van het type ALERT naar DROP. Ook hebben we geprobeerd of het type REJECT wel het verkeer uit de datastroom filterde, maar helaas zonder resultaat.

Tegelijkertijd met het uittesten van de honeywall bleek dat de transparante proxy niet in combinatie met een bridge werkte. De transparante proxy is toen voor de testsituatie op een aparte machine opgezet en voor de honeywall hebben we besloten om de kant en klare honeywall CD te proberen.

Ook de kant en klare honeywall kregen we niet direct werkend. Uiteindelijk is het gelukt om de honeywall CD werkend te krijgen en hebben we ook aanwijzing waar mogelijk het probleem zat bij de zelf opgezette honeywall. Bij het configureren van de honeywall CD bleek namelijk dat de Snort configuratie van essentieel belang en dan met name de instellingen voor de aangesloten netwerken. In Snort termen, het is belangrijk om goed na te denken wat je HOME_NET is en wat je EXTERNAL_NET is.

4.2 Proxy

4.2.1 Implementatie

De proxy server is globaal als volgt ingericht: Een Debian Sarge machine met een IPtables firewall en ipv4 forwarding disabled. Hierop draait een DHCP server voor het uitgeven van ip-adressen en het automatisch configureren van DNS servers op de clients. Daarnaast draait er een DNS server voor het forwarden en cachen van DNS requests en per aangeboden service een proxy applicatie die geconfigureerd is om transparant te werken. De firewall dropt in principe al het verkeer van en naar het wireless subnet tenzij dit verstuurd wordt op de, standaard door de gedefinieerde services gebruikte, poorten. Dit verkeer wordt middels NAT omgeleid naar speciale poorten waarop proxyapplicaties luisteren. Welke applicaties dit zijn, welke mogelijkheden ze bieden en welke poorten ze gebruiken wordt in de volgende paragrafen uitgelegd.

DNS: bind9 Als DNS server is gekozen voor bind9. Deze biedt in de standaard configuratie een volwaardige forwarding DNS oplossing. DNS requests vanaf de client komen binnen op de server, welke ze vervolgens forward naar de DNS server van de UVA. Hiermee wordt de geldigheid van uitgaande requests gewaarborgd en wordt voorkomen dat er (ongeldig) verkeer van de clients via de DNS poort naar buiten gaat. De bijbehorende firewall rule accepteert verkeer op TCP en UDP poort 53 vanaf het wireless subnet (192.168.1.0/24) bestemd voor de eigen server (192.168.1.1).

DHCP: dhcpd3 De DHCP server biedt de clients relatief korte leases aan (30 minuten) uit het bereik 192.168.1.150 tot 192.168.1.200. Ook wordt als DNS server standaard de eigen server ingesteld (192.168.1.1). Hiermee heeft de client voldoende informatie om zonder handmatige configuratie gebruik te maken van spatti.

De bijbehorende firewall rule accepteert verkeer op UDP poort 67 en 68 vanaf het wireless subnet (192.168.1.0/24) en 0.0.0.0 bestemd voor de eigen server (192.168.1.1) en het broadcast adres 255.255.255.255.

HTTP/FTP: squid Als HTTP/FTP proxy is gekozen voor Squid. Squid is een veelgebruikte proxy met veel configuratieopties en uitbreidingsmogelijkheden. Voor ons doeleinde hebben we Squid geconfigureerd als transparante proxy. Dat houdt in dat verkeer bestemd voor externe adressen op port 80 en port 21 door de firewall wordt geredirect naar de squid poort (poort 3128) waarna de squid proxy de pagina ophaalt en aan de client doorgeeft. Daarbij worden pagina's standaard gecached. In deze configuratie is het niet mogelijk om gebruik te maken van HTTPS, iets dat volgens onze projectspecificaties ook niet nodig is (of mag !). Met het gebruik van squid wordt voorkomen dat poort 80 wordt misbruikt voor het doorlaten van niet web- of ftp-gerelateerd verkeer. Alleen geldige http en ftp protocollen worden verwerkt. Het is echter tot op bepaalde hoogte nog steeds mogelijk om verkeer te encapsuleren in http pak-

ketten (MSN maakt hiervan bijvoorbeeld gebruik). Ook is het mogelijk om een maximale grootte aan te geven voor te downloaden bestanden. Squid biedt de mogelijkheid om te werken met ACL's (Access Control Lists). Hiermee kunnen groepen van gebruikers worden gedefinieerd die bepaalde rechten hebben. Zo kan bijvoorbeeld verkeer op basis van een IP adres (gekoppeld aan een MAC adres door het gebruik van de DHCP server) bij overmatig gebruik worden gelimiteerd. Ook kan er op basis hiervan aan nieuwe gebruikers een welkomstpagina worden getoond.

Verder kan Squid gebruikt worden in combinatie met ClamAV om te helpen virussen buiten het netwerk te houden. Voordeel van deze aanpak boven het gebruik van ClamAV in een honeywall configuratie is dat bestanden eerst compleet kunnen worden gedownload voordat ze gechecked worden op virussen: hiermee kunnen ook ingepakte bestanden worden gescanned. Squid is uit te breiden met DansGuardian. DansGuardian biedt de mogelijkheid te filteren op content op basis van bijvoorbeeld pattern matching of blacklists. Hiermee kan het binnenhalen van illegale content bemoeilijkt worden, zoals bijvoorbeeld sites met racistische uitlatingen of kinderporno.

SMTP: ProxSMTP ProxSMTP is een simpel programma dat SMTP verkeer accepteert, door een externe applicatie stuurt en (eventueel) verder stuurt. Als externe applicatie kan bijvoorbeeld ClamAV worden gebruikt voor het checken op virussen of een eigen geschreven script dat filtert op content of aantal recipients. Ook kan in een eigen script een counter worden opgenomen die maar een beperkt aantal SMTP connecties per tijdseenheid accepteert om spammen tegen te gaan.

De bijbehorende firewall rule redirect verkeer op TCP poort 25 vanaf het wireless subnet (192.168.1.0/24) bestemd voor 'the world' naar poort 10025 op de eigen server en accepteert dit.

POP: p3scan P3scan is een simpel programma dat POP3 verkeer accepteert, door een externe applicatie stuurt en (eventueel) verder stuurt. Het kan, net als ProxSMTP, het verkeer door een externe applicatie leiden en zo scannen op malicious content. Naast op worms en virussen (in combinatie met ClamAV) kan zo ook gescanned worden op HTML emails en spam. De bijbehorende firewall rule redirect verkeer op TCP poort 110 vanaf het wireless subnet (192.168.1.0/24) bestemd voor 'the world' naar poort 8110 op de eigen server en accepteert dit.

IMAP Voor IMAP kwamen we wel een aantal proxy applicaties tegen maar geen geschikte die transparant kan werken. Wellicht is er eenvoudig een zelf te schrijven maar wegens tijdgebrek hebben we hier van afgezien. Het principe is hetzelfde als de hierbovengenoemde oplossingen.

5 Encryptie

Zoals uit ons vooronderzoek is gebleken is het niet verstandig om het gebruik van versleutelde sessies tussen de gebruiker en een externe machine toe te staan, daar dit door de aanbieder niet te controleren is. Toch is het gebruik van encryptie aan te bevelen omdat het uitgangspunt wireless verkeer is dat makkelijk onderschept kan worden. Daarom hebben we een tweetal oplossingen bedacht die het verkeer tussen de gebruiker en onze machine versleutelen. Dit kan als optie worden aangeboden, waarbij het aan de gebruiker is of hij ervan gebruik wenst te maken. De (onveilige) standaardfunctionaliteit blijft bestaan waarbij de welkomstpagina gebruikt kan worden om de gebruiker op de mogelijkheid van encryptie te attenderen.

5.1 VPN

Middels VPN kan een gebruiker een beveiligde verbinding opzetten over een onveilig medium. Functionaliteit hiervoor is vaak standaard aanwezig in diverse besturingssystemen zoals Windows XP in de vorm van een VPN client. Mocht dit niet zo zijn dan kan deze eenvoudig gedownload worden via de standaard (onveilige) verbinding.

Het idee is dat de gebruiker via de standaard welkomstpagina op een beveiligde pagina komt (https) waar hij een persoonlijke gebruikersnaam en wachtwoord krijgt en het adres van onze VPN server (de proxy). Met deze gegevens kan hij vervolgens z'n VPN client configureren en een veilige verbinding opzetten. Een alternatief is deze informatie via SMS naar de gebruiker te sturen, nadat deze op de welkomstpagina z'n mobiele nummer heeft ingevuld.

5.2 802.1x

Een andere mogelijkheid voor encryptie is 802.1x. Helaas zijn er (voor zover we hebben kunnen vinden) geen consumeraccesspoints verkrijgbaar die tegelijkertijd open (dus niet versleuteld) en 802.1x verkeer ondersteunen: het is of het ene of het andere. Daarom zou er een 2e accesspoint geplaatst kunnen worden dat wel gebruik maakt van 802.1x. Middels een https pagina op het standaard accesspoint kan de gebruiker een certificaat en/of een gebruikersnaam en wachtwoord verkrijgen alsmede instructies hoe deze te configureren. Daarmee kan vervolgens verbinding worden gemaakt met het 802.1x accespoint.

6 Aanbevelingen

Uit het project is gebleken dat het goed mogelijk is om betrekkelijk veilig publiek Internet aan te bieden. Hierbij zijn nog wel een paar punten te noemen die verbeterd kunnen worden, met name op het gebied van veiligheid voor de gebruiker.

6.1 Encryptie

Wij bevelen aan te kijken naar een van de hierboven genoemde encryptiemethoden voor het versleutelen van het verkeer tussen de gebruiker en onze gateway. Daarbij is het aan de gebruiker of hij hiervan gebruik wenst te maken. Voor het eenvoudige surfen is dit wellicht niet nodig en volstaat de open (niet versleutelde) verbinding.

6.2 Authenticatie en accounting

Hoewel het buiten de scope van ons project valt, is het vrij eenvoudig om gebruikers te authenticeren middels het versturen van bijvoorbeeld een SMS met gebruikersgegevens. Hiermee is er toch een vorm van accounting mogelijk en kan aan een gebruiker meer mogelijkheden worden geboden omdat hij bij misbruik (deels) te achterhalen is.

6.3 All-in-one oplossing

Er wordt nu nog gebruik gemaakt van twee losse machines en een accesspoint. Voor commerciële toepassingen moet dit goedkoper en eenvoudiger. Daarom is het aan te bevelen de honeywall en de proxy te verenigen in één machine. Uitgaande van de Honeywall CD kan dit door de deze in de NAT configuratie te installeren en op dezelfde machine de genoemde proxyservers te implementeren. Mocht voor een geheel eigen configuratie worden gekozen dan moet de proxy machine uitgebreid worden met Inline Snort en de bijhorende configuratie. Het zou zelfs mogelijk moeten kunnen zijn om dit op het accesspoint te doen maar daarvoor is verder onderzoek nodig.

7 Conclusie

Een vrij toegankelijk netwerk is straks niet meer weg te denken uit de samenleving. Op dit moment gaan zelfs de grote telecom aanbieders in Nederland zich bezighouden met het realiseren van een soortgelijk landelijk dekkend netwerk op basis van WiFi. Dit zegt iets over de huidige vraag naar een (draadloos) vrij toegankelijk Internet in en om de werkplek. Voornamelijk particulieren en het MKB zullen niet afhankelijk willen zijn van de grote(re) aanbieders en zullen dus zelf op kleine schaal draadloos Internet willen gaan aanbieden. Hierbij is te denken aan bijvoorbeeld kroegen, lunchrooms en andere gelegenheden. Over de vraag naar een implementatie zal dus geen twijfel zijn.

Tijdens het project is gebleken dat het opzetten van een vrij toegankelijk netwerk met zo min mogelijk configuratie voor de gebruikers goed te doen is. Een vervolg project op SPATTI zou de stappen die tijdens dit onderzoek niet aan bod zijn gekomen verder kunnen uitwerken. Op een aantal punten zou, na nader onderzoek te hebben gedaan, het volgende als uitbreiding van het SPATTI project kunnen dienen

- Accounting
- Encryptie
- Authenticatie dmv 802.1x

De bovengenoemde punten van onderzoek zijn al behoorlijk goed onderzocht, maar niet in combinatie met transparantie voor de gebruikers, een doel van het SPATTI project. Naar onze mening is het SPATTI project een eenvoudige manier om beveiligd draadloos Internet te kunnen aanbieden wat goed bereikbaar is voor de gebruiker zonder daar enige vorm van complexe instelling voor te moeten doen.

A Honeywall

Voor de honeywall hebben we Debian Sarge als OS gebruikt.

A.1 Bridge

Het opzetten van de bridge gaat op de volgende manier. Eerst installeren we de benodigde tools, hiervoor maken we gebruik van apt-get.

```
apt-get install bridge-utils
```

Vervolgens maken we een nieuwe bridge aan en voegen de interfaces toe (in dit geval eth0 en eth2).

```
brctl addbr br0

brctl addif br0 eth0

brctl addif br0 eth2
```

Daarna activeren we de bridge.

```
ifconfig br0 up

ifconfig eth0 up

ifconfig eth2 up
```

A.2 IPtables

```
    #!/bin/sh

# Fysiek interface -iptables naam -omschrijving
# eth0 -br0 -internet (UvA)
# eth1 -eth1 -management
# eth2 -eth2 -accesspoint

# Een pakket niet bestemd voor de host gaat door ([] = alleen de eerste keer):
# mangle:PREROUTING - [nat:PREROUTING] - mangle:FORWARD -
# filter:FORWARD - mangle:POSTROUTING - [nat:POSTROUTING]

###
### mangle table
###

#iptables -t mangle -F

#iptables -t mangle -P PREROUTING DROP
#iptables -t mangle -P POSTROUTING DROP
```

```
###
### nat table
###

#iptables -t nat -F

#iptables -t nat -P PREROUTING DROP
#iptables -t nat -P POSTROUTING DROP



###
### filter table
###

iptables -F

iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP

# Verkeer van het accesspoint naar buiten
iptables -t filter -A FORWARD -m physdev --physdev-in eth2 --physdev-out eth0
-m state --state NEW,RELATED,ESTABLISHED -p tcp --dport http -j QUEUE

iptables -t filter -A FORWARD -m physdev --physdev-in eth2 --physdev-out eth0
-m state --state NEW,RELATED,ESTABLISHED -p tcp --dport https -j QUEUE

iptables -t filter -A FORWARD -m physdev --physdev-in eth2 --physdev-out eth0
-m state --state NEW,RELATED,ESTABLISHED -p tcp --dport smtp -j QUEUE

iptables -t filter -A FORWARD -m physdev --physdev-in eth2 --physdev-out eth0
-m state --state NEW,RELATED,ESTABLISHED -p tcp --dport pop3 -j QUEUE

iptables -t filter -A FORWARD -m physdev --physdev-in eth2 --physdev-out eth0
-m state --state NEW,RELATED,ESTABLISHED -p tcp --dport imap2 -j QUEUE

iptables -t filter -A FORWARD -m physdev --physdev-in eth2 --physdev-out eth0
-m state --state NEW,RELATED,ESTABLISHED -p tcp --dport ftp -j QUEUE

iptables -t filter -A FORWARD -m physdev --physdev-in eth2 --physdev-out eth0
-m state --state NEW,RELATED,ESTABLISHED -p tcp --dport ftp-data -j QUEUE
```

```
iptables -t filter -A FORWARD -m physdev --physdev-in eth2 --physdev-out eth0
-m state --state NEW,RELATED,ESTABLISHED -p tcp --dport ssh -d 192.168.1.1
-j ACCEPT

iptables -t filter -A FORWARD -m physdev --physdev-in eth2 --physdev-out eth0
-m state --state NEW,RELATED,ESTABLISHED -p udp --dport domain -j QUEUE

iptables -t filter -A FORWARD -m physdev --physdev-in eth2 --physdev-out eth0
-p udp --dport bootps -j ACCEPT

# Verkeer van buiten naar het accesspoint
iptables -t filter -A FORWARD -m physdev --physdev-in eth0 --physdev-out eth2
-m state --state RELATED,ESTABLISHED -p tcp --sport http -j QUEUE

iptables -t filter -A FORWARD -m physdev --physdev-in eth0 --physdev-out eth2
-m state --state RELATED,ESTABLISHED -p tcp --sport https -j QUEUE

iptables -t filter -A FORWARD -m physdev --physdev-in eth0 --physdev-out eth2
-m state --state RELATED,ESTABLISHED -p tcp --sport smtp -j QUEUE

iptables -t filter -A FORWARD -m physdev --physdev-in eth0 --physdev-out eth2
-m state --state RELATED,ESTABLISHED -p tcp --sport pop3 -j QUEUE

iptables -t filter -A FORWARD -m physdev --physdev-in eth0 --physdev-out eth2
-m state --state RELATED,ESTABLISHED -p tcp --sport imap2 -j QUEUE

iptables -t filter -A FORWARD -m physdev --physdev-in eth0 --physdev-out eth2
-m state --state RELATED,ESTABLISHED -p tcp --sport ftp -j QUEUE

iptables -t filter -A FORWARD -m physdev --physdev-in eth0 --physdev-out eth2
-m state --state RELATED,ESTABLISHED -p tcp --sport ftp-data -j QUEUE

iptables -t filter -A FORWARD -m physdev --physdev-in eth0 --physdev-out eth2
-m state --state RELATED,ESTABLISHED -p tcp --sport ssh -s 192.168.1.1
-j ACCEPT

iptables -t filter -A FORWARD -m physdev --physdev-in eth0 --physdev-out eth2
-m state --state RELATED,ESTABLISHED -p udp --sport domain -j QUEUE

iptables -t filter -A FORWARD -m physdev --physdev-in eth0 --physdev-out eth2
-p udp --sport bootps -j ACCEPT

# Verkeer op management interface
iptables -A INPUT -i eth1 -p tcp --dport ssh -j ACCEPT
iptables -A INPUT -i eth1 -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth1 -m state --state NEW,RELATED,ESTABLISHED
```

```
-j ACCEPT
```

A.3 Snort

Voordat we Snort kunnen installeren moeten er eerst een aantal andere programma's geïnstalleerd worden. Dit doen we met behulp van apt-get.

```
apt-get install iptables-dev
```

```
apt-get install libpcap0.8-dev
```

```
apt-get install libpcre3-dev
```

```
apt-get install libnet0-dev
```

Vervolgens downloaden we de source code van Snort van de website, pakken het uit en installeren het. Hierbij geven we aan dat we Snort willen gebruiken in inline mode.

```
./configure --enable-inline --with-libipq-includes=/usr/include/libipq
```

```
make
```

```
make install
```

Vanuit de installatiemap kopiëren we vervolgens een aantal bestanden.

```
cp installatiemap/etc/snort.conf /etc
```

```
cp installatiemap/etc/reference.config /etc
```

```
cp installatiemap/etc/classification.config /etc
```

```
cp installatiemap/etc/unicode.map /etc
```

De configuratie van Snort kun je eventueel nog aanpassen. Hierna kun je Snort starten door de volgende commando's uit te voeren.

```
mkdir /var/log/snort/
```

```
modprobe ip_queue
```

```
snort -Qc /etc/snort.conf -l /var/log/snort
```

B Proxy

B.1 Bind9

```
apt-get install bind9
Standaard config
```

B.2 dhcpd3

/etc/dhcp3/dhcp3.conf

```
ddns-update-style none;
authoritative;
log-facility local7;

subnet 192.168.1.0 netmask 255.255.255.0 {
        range 192.168.1.150 192.168.1.200;
        option routers 192.168.1.1;
        option domain-name-servers 192.168.1.1;
        option time-servers 192.168.1.1;
        option broadcast-address 192.168.1.255;
        default-lease-time 300;
        max-lease-time 600;
}
```

B.3 Squid

/etc/squid/squid.conf

```
# Allow only http requests from wireless subnet
acl our_networks src 192.168.1.0/24
http_access allow our_networks
http_access allow localhost

http_access deny all

# Max download filesize (5mb)
reply_body_max_size 5242880 allow all

# Setting for transparent functionality
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
```

B.4 ProxSMTP

/etc/proxsmtpd.conf

```
# The amount of time to wait for data from FilterCommand
FilterTimeout: 10

# The type of filter ('pipe' to pipe data through filter,
# or 'file' to pass a file to the filter)
#FilterType: pipe
```

```
# The maximum number of connection allowed at once.
# Be sure that clamd can also handle this many connections
MaxConnections: 64

# Amount of time (in seconds) to wait on network IO
TimeOut: 180

# A header to add to all scanned email
Header: X-Filtered: By ProxSMTP

# Keep Alives (ie: NOOP's to server)
KeepAlives: 0

# Send XCLIENT commands to receiving server
#XClient: off

# Address to listen on (defaults to all local addresses on port 10025)
Listen: 192.168.1.1:10025

# Directory for temporary files
TempDirectory: /tmp

# Enable transparent proxy support
TransparentProxy: on
```

B.5 P3scan

/etc/p3scan/p3scan.conf

```
# PID File
#  where to write a pid-file
pidfile = /var/run/p3scan/p3scan.pid

# Max Childs
#   The maximum number of connections we will handle at once. Any further
#   connections will be dropped. Keep in mind that a number of 10 also
#   means that 10 viruscanner can run at once.
maxchilds = 10

# IP Address
#   The IP Address we listen on default: 0.0.0.0 (any address)
ip = 192.168.1.1

# Port
#   The tcp port on we should listen. If you need a privileged port you
```

```
#   need to start p3scan as root (but don't set username to root,
#   that's not necessary, because first after opening the port we will
#   switch to that user).
port = 8110

# TargetIP, TargetPort
#   targetip and targetport are the ip and port to connect -
#   default is 0.0.0.0 (transparent proxy mode) and 8110 respectively
targetip = 0.0.0.0
targetport = 8110

# Virus Directory
#   The directory in which infected mails will be stored.  It is also
#   used for temporary storing. Ensure that the above specified user is
#   allowed to write into!
 virusdir = /var/spool/p3scan
 virusdir = /var/spool/p3scan

# Scanner Type
#   Select here which type of scanner you want to use.
#   At the moment you can choose between 'basic' and 'avpd'.
scannertype = basic

# Virusscanner
#   Depends on scannertype. Read the above section of that scannertype
#   you're going to use and you do not need to ask what to fill in here.
scanner = /usr/bin/clamdscan --no-summary

# Enable Spam checking
#
#  If set, will scan for Spam before scanning for a virus.
#
#  P3scan has been tested with both dspam-3.0.0-rc2 and
#  Mail::SpamAssassin v2.6.
checkspam

# Mail::SpamAssassin
#  Where to find spamc, the link to the Mail::SpamAssassin daemon spamd.
spamcheck = /usr/bin/spamc

# Rename Attachments
#
#  If renattach is installed and this option is un-commented, we
#  will execute renattach to rename dangerous attachments.
#  (See README for more information)
# renattach = /usr/local/bin/renattach
```

```
# Overwrite (disable) HTML
#
#  If a person views an HTML message, not only can the client
#  download pictures automatically, it enables someone viewing
#  the remote log file to confirm the email address is valid,
#  making it "worth" keeping/selling, etc...
#
#  p3scan comes with a separate program p3pmail that can be
#  installed for this purpose.
# overwrite = /usr/bin/p3pmail
# END of configuration
```