Firewall opsætning – Smoothwall (Linux)

Udarbejdet af: René Prangsgaard &

Jonas Jensen

Fag: Informationsteknologi

Lærer: Benny Dyhr Thomsen

Afleveret den. 13-11-2003

Titelblad

Rapportens titel:	Firewall opsætning – Smoothwall (Linux)
Uddannelsesinstitution:	Erhvervsakademi Midtjylland
Uddannelse:	IT –og elektronik Teknolog
Klasse:	ITE 2-1, 3 semester
Gruppemedlemmer:	René Prangsgaard & Jonas Kruse Jensen
Lærer:	Benny Dyhr Thomsen
Fag:	Informations Teknologi
Projekt udleveret den:	30-10-03, uge 44
Projekt afleveret:	13-11-03, uge 46

Indholdsfortegnelse

Indledning/projektbeskrivelse

Rapporten omhandler opsætning af en firewall. Kravet til firewallen er opstillet i en case, som er blevet udleveret. Se bilag 1. Formålet med rapporten er at omsætte firewall teori til et praktisk projekt.

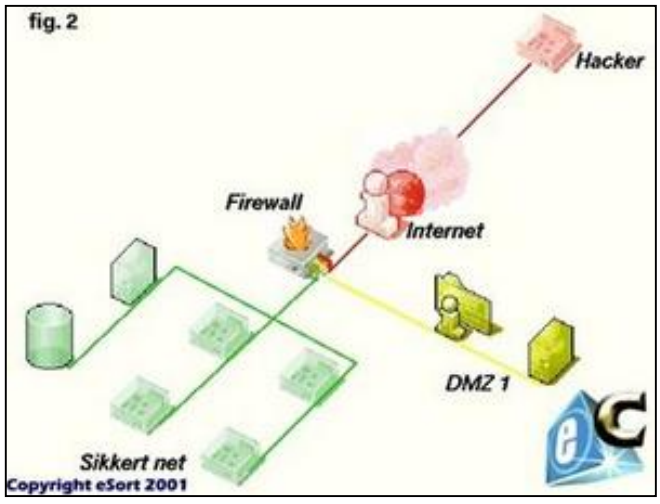
Til at løse opgaven har vi valgt at arbejde med ”open source” software, baseret på Red Hat (Linux), hvilket er Smoothwall 2.0 beta 7. Denne løsning er gratis og kan hentes på web-adressen www.smoothwall.org. Smoothwall henvender sig til mindre eller mellemstore virksomheder, som gerne vil have en billig og brugervenlig firewall løsning.

Rapporten indeholder en detaljeret beskrivelse af, hvad en firewall er og hvordan den fungere. Rapporten beskriver teorien bag en firewall. Hvilke muligheder og funktionaliteter, en firewall indeholder i dag. Hvordan SmoothWall installeres og konfigureres. FTP problematikker i forbindelse med firewall’s. Test samt sikkerhedsvurdering af Smoothwall, og det opstillede netværk.

Kap 1 - Firewall Teori

Hvad er en firewall ?

En firewall er et værktøj, som er i stand til at analysere og blokere trafik mellem netværk, ud fra et regelsæt. En firewall's funktion består typisk af at tillade bruger nogle/alle services på Internet. Derimod er hensigten at spærre for adgangen den modsatte vej, så uvedkommende ikke får adgang til de lokale ressourcer eller de kun får adgang til udvalgte services specificeres i firewallen. Tyveri af fortroligt materiale kan være en årsag til at beskytte sig, men der er lige så ofte tale om hærværk eller decideret misbrug af servere/services på ubeskyttede netværk. Selve firewallen var oprindeligt blot en router/gateway, der tillod alt udgående trafik, men blokerede al indkommende trafik eller kun tillod eks. mail og web trafik, til interne servere på et eller flere isoleret område kaldet DMZ - forklares senere. I dag er firewall's meget mere avancerede. De kan separere netværkstrafik efter ønsket mønstre og adfærd. Hvert segment (netværk/DMZ) skal konfigureres individuelt i forhold til behov/sikkerhed. Se illustration af et netværk herunder, hvor en firewall er blevet indsat i et netværk:



De 3 elementer i en firewall

Mange tror at en firewall, bare er en firewall. Men en firewall kan faktisk deles op i 3 elementer. De 3 elementer består i analyse af netværkstrafikken. Til dette formål bruges ”Pakke filtrering”, ”Application level gateway” og ”Circuit gateways”. De vil herunder blive beskrevet.

1. Pakke filtrering:

Pakke filtrering er en primitiv firewall funktion, og de fleste standard routere har denne egenskab. Funktionen har det formål at kontrollere "hvilken afsender adresse, der må routes til modtagerens adresse" og "hvilken modtager adresse, der må modtages fra afsenderens adresse" blandet med port regler. Administratorens arbejde ligger her i at definere om A, må sende til B, og om B kun må modtage fra C, og hvilke porte trafikken må forgå på ifølge det definerede regelsæt. Alle pakker der ikke opfylder reglerne, bliver droppet. Men man må ikke udelukkende, sætte sin lid til pakke filtrering. Hvad med brugerne inde på det interne netværk? En stor undersøgelse i USA viser, at det sjovt nok er herfra de fleste "angreb" sker. Afskedigede medarbejdere der lige vil rydde lidt ekstra op på serverne, og endnu værre scenarier kan man forestille sig, hvis der er tale om IT administrative medarbejdere, som selv laver nogle konfigurationer.

Spørgsmålet er så, ”om man ikke skal benytte pakke filtrering?” Svaret er jo, men man skal ikke kun bruge denne teknologi. I dag er man nød til, så vidt det er muligt og økonomien er til det, at få alle 3 elementer med. Og så er det stadigvæk kun "indgangen" til netværket der er beskyttet, ikke sikret. Der er generelt tale om, at har man en forbindelse ud i verden, så der er altid en potentielt risiko for misbrug.

2. Application level gateway:

Denne teknologi er en langt mere avanceret teknologi end pakke-filtrering. Firewallen kender udfra forskellige regelsæt/informationer, diverse applikationer og deres kommunikations mønstre, og kan være "apdaptive", altså lære nye programmers adfærd efterhånden som de tages i brug. (i stil med ZoneAlarm produktet, som er populært blandt private PC brugere). Hver gang et program prøver at tage kontakt, med en web-applikation, vil ZoneAlarm komme med en meddelelse til brugeren om at computeren nu prøver og tage kontakt til en web-applikationen og

man kan så vælge at ”Allow” trafikken eller ”Blocke” for kommunikationen. Et eksempel kan være, at en computeren forsøger, at skabe forbindelse til en maskine i Rusland uden man har bedt den om det. Dette kunne indikerer at det er en ”trojansk hest” på klientens computer. Bruges Application level gateway sammen med packet filtering, har man en (ret) sikker løsning hvor truslen (næsten) kun kan komme indefra eller fra et "trusted" segment eller lignende.

3. Circuit Gateways:

Firewallen fungerer her, udover som sikker gateway, også som service proxy (latinsk for stedfortræder). Typisk har man i denne konstellation en DMZ (demilitarized zone), hvor der er placeret servere med diverse Internet services. Et eksempel kan være en web server der hoster virksomhedens hjemmeside. Firewallen er sat til at lytte på port 80 (standard http port). Hver gang firewallen modtager trafik på port 80, laves der en "port forwarding" ind til web serveren. Det vil sige, at den trafik der kommer ind på port 80 vil blive sendt videre til web-serverens IP adresse. Denne behøver naturligvis ikke benytte port 80 når bare firewall og server ved hvilken port de kommunikerer indbyrdes på.

Firewall opsætnings muligheder

De fleste organisationer på nettet har et vist behov for at servicere WWW, DNS, FTP, e-mail og lignende tjenester. Når man har dette behov kan firewallen opsættes på følgende måder:

1. at have serverne på det interne net, og tillade kommunikation med dem gennem firewallen,
2. at køre alle tjenester på selve firewallen,
3. at have serverne på ydersiden af firewall, så de sidder direkte på Internettet,
4. at have serverne placeret på et eller flere isolerede netværk.

(1) har det oplagte problem at hvis der er en sikkerhedsfejl i firewallen, kan hackeren skaffe sig adgang til serveren, og derfra til resten af organisationens maskiner.

(2) giver næsten samme problematik som (1).

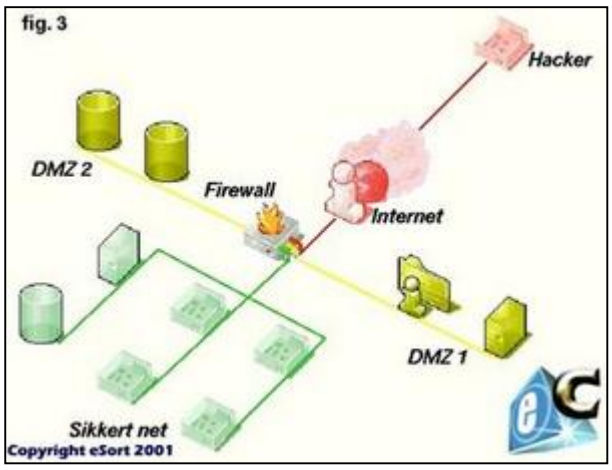
(3) kræver at maskinerne på ydersiden har stort set samme sikkerhedsniveau som selve firewallen, undtagen på de åbne services.

(4) er netop den løsning man betegner DMZ. Det separate LAN vil i firewallen være åbent for de nødvendige tjenester, men hvis en hacker skulle skaffe sig adgang til en server i DMZ'en, vil han ikke have skaffet sig direkte adgang til organisationens interne data/det interne net.

DMZ betegnes også som PSN (Protected Services Network), perimeter network og lignende, men idéen er den samme. Det er bare en anden betegnelse for DMZ. Se evt. billedet i afsnittet ”DMZ – DeMilitarized Zone” for bedre forståelse af hvor DMZ er placeret i forhold til det interne net.

DMZ – DeMilitarized Zone

DMZ er en forkortelse af DeMilitarized Zone, ingenmandsland. Betegnelsen dækker normalt over det landområde der ligger mellem to landes grænser. Indenfor netværk bruges begrebet i firewall sammenhæng. DMZ er et separat net isoleret fra det private netværk, hvor man placere de servere, som omverdenen (Internettet) tager kontakt med eks. FTP, Web osv. Under ingen omstændigheder kan DMZ-zonens servere få kontakt med det sikre/interne net, med mindre dette er konfigureret i firewallen. Billedet herunder illustrere 2 DMZ zoner:



IDS – Intrusion Detection System

IDS er en forkortelse af "Intrusion Detection System". Det er en avanceret overvågning af alle definerede elementer, således at man ikke først opdager et angreb når det er foregået, men derimod bliver gjort opmærksom på det når det begynder, ved at opsætte regelsæt. Netværks-baserede IDS (kaldet NIDS) virker ofte ved at genkende elementer af netværkspakker sendt af programmer der kan udnytte sikkerhedsfejl fra en anden maskine på Internettet.

Tro ikke at man ser alt på dit netværk blot fordi man har et IDS. Tro heller ikke, at alt hvad man ser i dit IDS er et indbrudsforsøg. En brug af NIDS kunne være at lave grafer for at se hvor mange der havde forsøgt, at bryde ind i dit system f.eks xxx-angreb.

Værts-baserede IDS (på engelsk kaldet Host-based IDS) kører på en maskine, og kan gøre opmærksom på ændringer i udvalgte filer i systemet. Dette sker ved at analysere kritiske filer og omforme denne analyse til resultatet af et matematisk udtryk (et såkaldt message digest). Dette resultat bliver så sammenlignet med en tidligere værdi, som er opbevaret sikkert. Enhver afvigelse tyder på at filen er blevet ændret og bør, indtil en anden forklaring er nået, betragtes som et vellykket indbrud. Det første som sker efter et indbrud er at den indtrængende opretter en konto til sig selv, eller ændrer et af de aktive programmer til at give den indtrængende hacker adgang til systemet, når hackeren senere vender tilbage. Værts-baserede IDS giver et godt indblik i ens system, og nogle mener at de er langt vigtigere end firewalls og NIDS.

Kap. 2 - SmoothWall 2.0 beta 7

Som løsningsforslag til casen der er blevet opstillet i vores projekt, har vi valgt at arbejde med open source softwaret SmoothWall 2.0 beta 7. SmoothWall blev udviklet af Lawrence Manning, Richard Morrell, Jon Fautley and Tom Ellis i 2000. Den første version var – version 0.9 som blev udgivet af SourceForge open source collaboration site i August 2000. Ideen var at udvikle et firewall operativsystem baseret på Linux, og at udvikle noget alternativ til de utrolige dyre hardware routere der var på dette tidspunkt.

Da den første version blev frigivet, viste det sig at projektet voksede meget hurtigt. Snart blev hundrede af kopier downloadet hver måned, og det viste sig snart at folk over hele verdenen benyttede softwaren. Folkene bag SmoothWall var forbavsede over hvor hurtigt SmoothWall var blevet populært, og de begyndte derfor at videreudvikle produktet. Der blev bl.a implementeret ISDN og ADSL support, Web proxy mm. Smoothwall er i dag en af de mest populære open source firewall's.

Målet med Smoothwall er i dag ligesom da de først startede med at udvikle den følgende:

- Protect the local network from outside attack, whilst interfering as little as possible with user activities
- Be simple enough to be installed by home users with no previous knowledge of Linux required
- Support a wide variety of network cards, modems and other hardware
- Work with many different connection methods and ISPs across the world
- Increase ease of use, management and configuration by use of tools such as web access interface

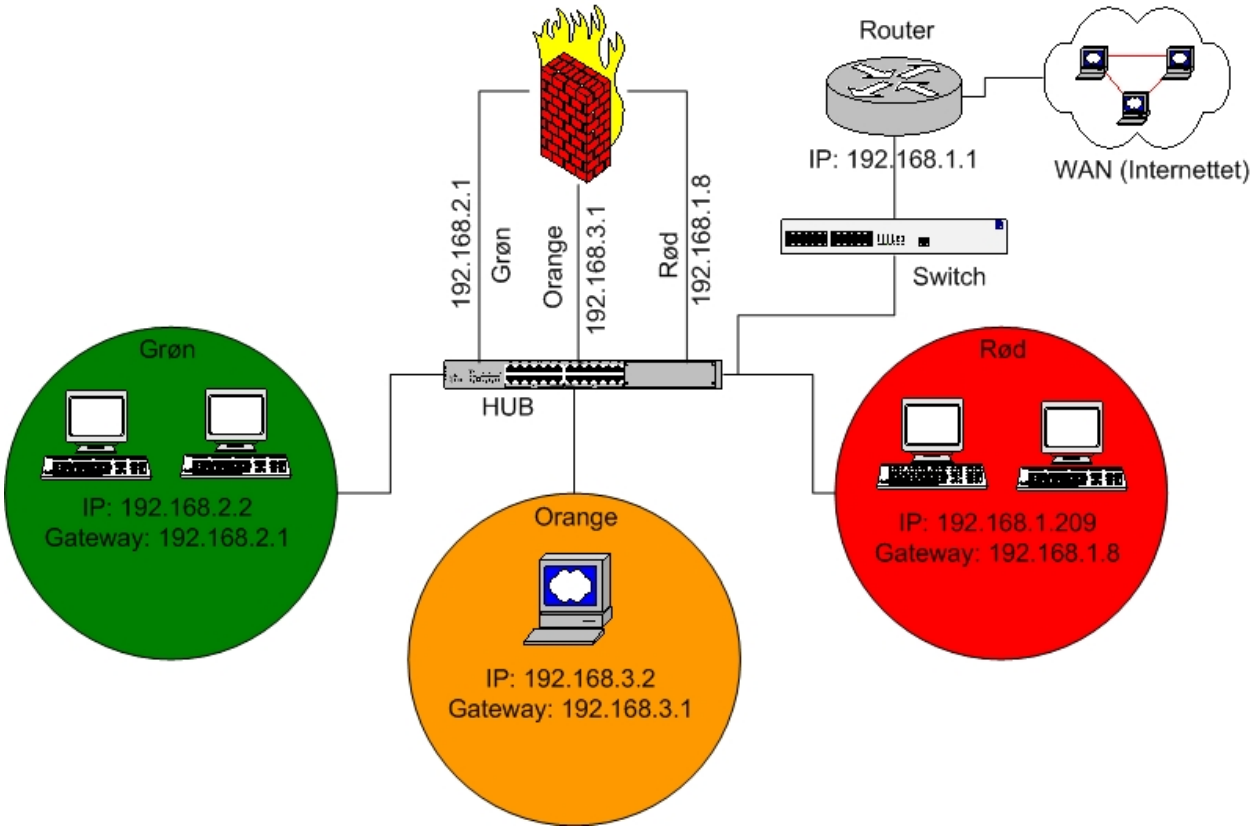
SmoothWall installation

En PC forsynes med 3 netværkskort, som repræsentere hver deres net. SmoothWall installeres fra en CD-ROM ISO image der er hentet fra http://www.smoothwall.org/. Softwaren er SmoothWall Express 2.0 beta7. Bemærk at installationen vil slette alle data på PC'en der anvendes som firewall, og PC'en kun skal anvendes som firewall og ikke andet. Når filerne på CD-ROM'en er installeret på PC'en bliver man præsenteret for SmoothWall menuen. Opsætningsrækkefølgen gennemgås herunder:

1. Først skal netværkskortet til *GREEN* net opsættes, hvilket er det private netværk. I menuen vælges probe for at få SmoothWall'en til automatisk at finde netkortet. Herefter opsættes netværksindstillingerne til *GREEN* net. I dette tilfælde tildeles netkortets (*GREEN)* IP adresse 192.168.2.1 og en subnetmaske på 255.255.255.0.
2. Keyboard sættes til dk
3. Hoastname sættes til smoothwall, som også er standard navnet.
4. I menuen Web proxy tastes ok.
5. ISDN og ADSL disables, hvis dette ikke er direkte forbundet til SmoothWall'en.
6. I *Network configuration menu* vælges *Network configuration type*. I denne menu vælges *GREEN + ORANGE + RED*, idet der anvendes 3 netkort.
7. *Drivers and card assigments* vælges. *ORANGE* og *RED* opsættes med Probe for at få SmoothWall'en til automatisk at finde netkortene.
8. *Address settings* vælges og *ORANGE* opsættes i dette tilfælde med IP adresse 192.168.3.1 og en subnetmaske på 255.255.255.0. *ORANGE* er DMZ (Demilitarised Zone). *RED* opsættes i dette tilfælde med 192.168.1.8 (WAN IP), og en subnetmaske på 255.255.255.0, som er netkortet der har direkte adgang til et usikkert net eks. Internettet. Det skal dog bemærkes at 192.168.1.8 er valgt, idet denne IP adresse tilhører IP serien der er koblet til default gateway (192.168.1.1), som er routeren der i dette tilfælde har adgang til Internettet.
9. *DNS and Gateway settings* vælges. Her indtastes IP adresserne på *Primary DNS* og *Secondary DNS*. Hvis man har installeret en DNS server på LAN'et kan denne anvendes ellers kan man bruge DNS IP adresserne ISP (Internet Service Provider) angiver. *Default Gateway* er gateway (port) til de netværk den forbinder. I dette tilfælde er *Default Gateway* routeren (192.168.1.1) der forbinder LAN'et til Internettet.. Til sidst indtastes password på admin, root og setup user. Bemærk at man altid kan reconfigure indstillingerne med kommandoen *setup*.

Kap. 3 - SmoothWall netværksopsætning (zoner)

Som nævnt i afsnittet ”SmoothWall installation” er SmoothWall installeret med 3 netkort. Hvert netkort repræsentere hver deres net (zone). Det private netværk (*GREEN)* subnet 192.168.2.0, DMZ (*ORANGE*) subnet 192.168.3.0, og forbindelse til Internettet (*RED)* subnet 192.168.1.0. Hver net er tilkoblet en Hub, som billedet herunder illustrere:

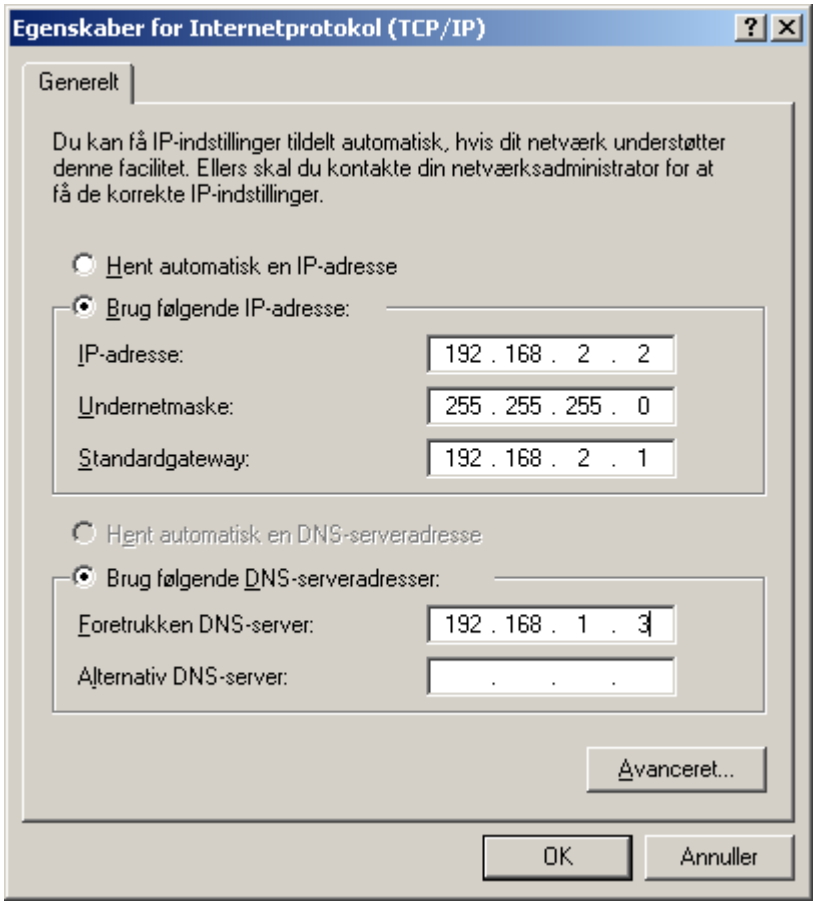


En Hub er valgt til afprøvning af SmoothWall, idet pakker sendt fra et netkort på eks. subnet 192.168.2.0 vil blive sendt ud til alle netkort på dette subnet, uanset om pakken kun er tiltænkt en modtager. Dette gør en Hub til en idel enhed til overvågning af al netværkstrafik modsat en Switch. Normalt vil man ikke koble hver zone til en Hub, men i stedet koble en switch imellem hver zone og firewallen. Zonerne er som nævnt opdelt i 3, og kan sammenlignes med et trafiklys. (*GREEN)* betragtes som værende ”*trusted”*, og her placere man virksomhedens klienter og LAN servere m.m. *ORANGE* er DMZ (demilitarized zone), og her placere man eks. Web, FTP servere osv. Omverdenen har altså kontakt til enkelte specificerede services på serverne placeret på

DMZ. (*RED)* har kontakten til et ”*unsafe*” netværk, som eks. Internettet. Denne zone er ofte kendt som ”external” eller ”untrusted” netværk.

Netværksopsætning på det private netværk

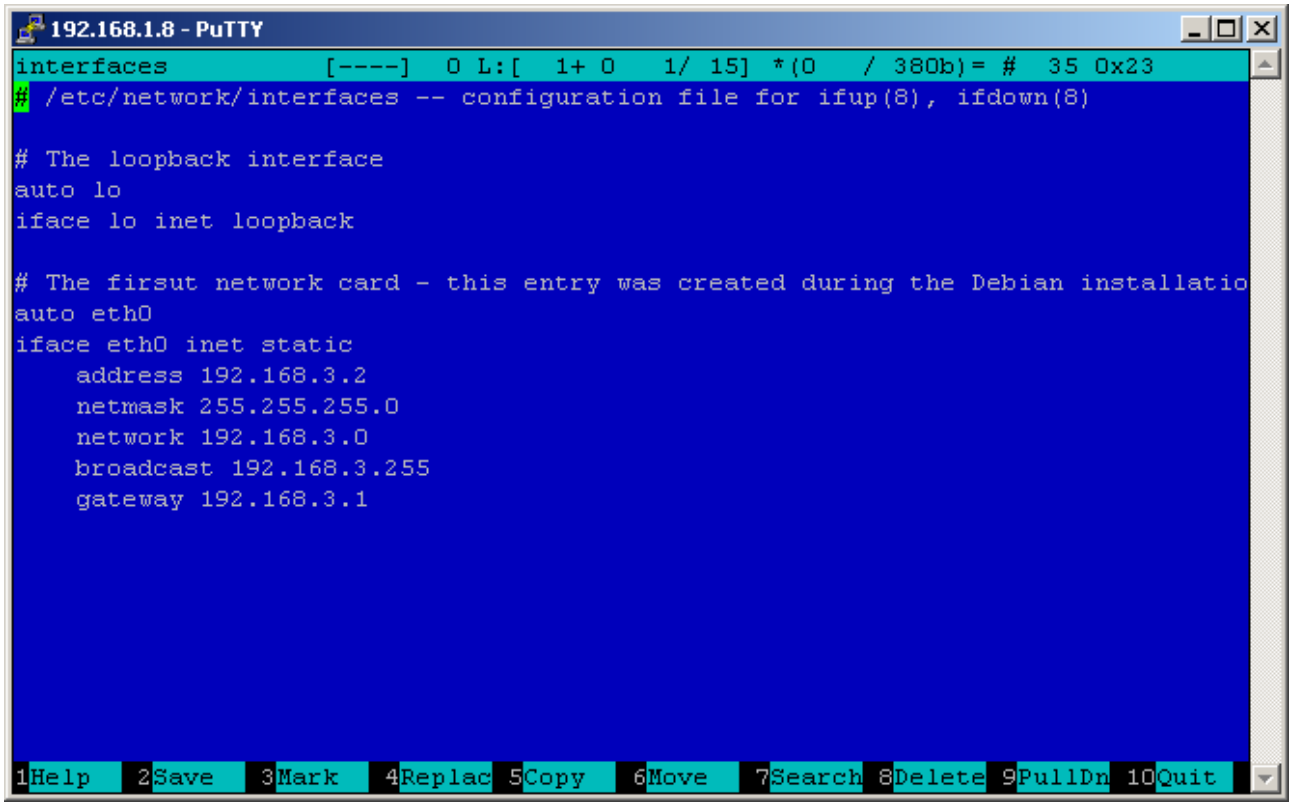
Det private netværk er den sikre zone (*GREEN)*, og der hvor klienterne er tilsluttet. Subnet er valgt til 192.168.2.0, og med en undernetmaske på 255.255.255.0. Den første klient på nettet har fået tildelt IP adresse 192.168.2.2, idet netværkskortet i SmoothWall’en der er forbundet til det private netværk har fået tildelt IP adressen 192.168.2.1. Netværkskortet (*GREEN*) i SmoothWall’en er dermed også gateway til de netværk den forbinder. Opsætningen af klienterne foregår i Netværks- og opkaldsforbindelser → LAN-forbindelse → Egenskaber → Internetprotokol (TCP/IP) → Egenskaber. Billedet herunder viser opsætningen:



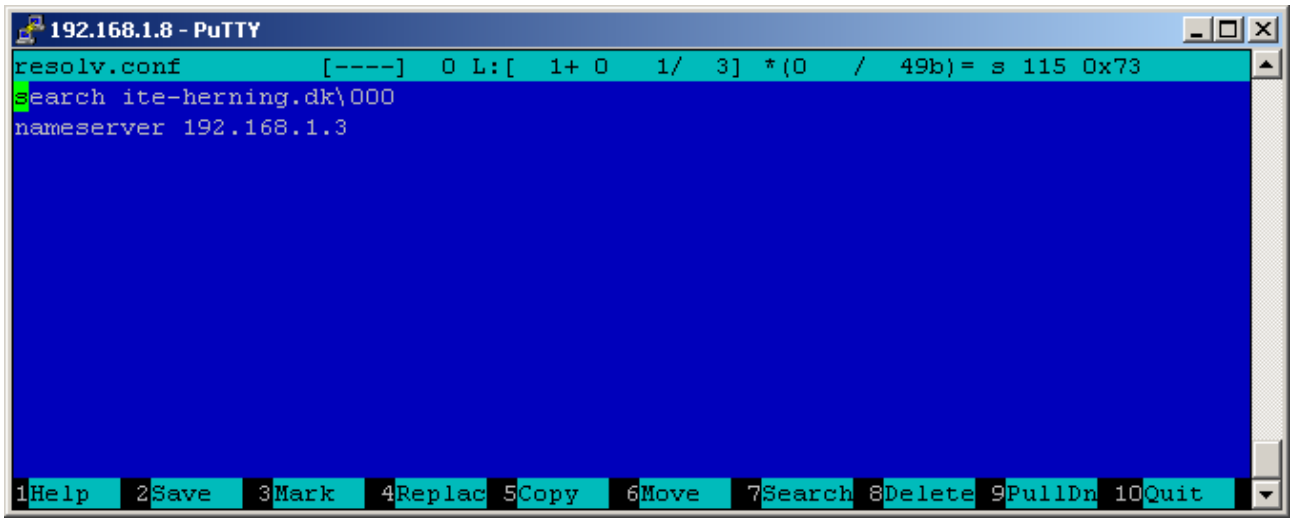
IP-indstillingerne er indtastet manuelt, idet DHCP ikke er aktiveret i SmoothWall'en og der er ikke opsat en DHCP server på LAN'et. DNS Serveren er 192.168.1.3, hvilket er en ekstern server SmoothWall'en har forbindelse til via *RED* netværkskortet.

Netværksopsætning på DMZ

DMZ (Demilitarised Zone) er der hvor servere, som omverden også har forbindelse til befinder sig. I DMZ er der installeret en PC med operativsystemet Linux (Debian). På PC'en er der installeret en WEB og FTP server. Subnet er valgt til 192.168.3.0, og med en undernetmaske på 255.255.255.0. Serveren på nettet har fået tildelt IP adresse 192.168.3.2, idet netværkskortet i SmoothWall'en der er forbundet til DMZ (*ORANGE)* har fået tildelt IP adressen 192.168.3.1. Netværkskortet (ORANGE) i SmoothWall'en er dermed også gateway til de netværk den forbinder. I Debian foretages IP indstillingerne i filen /etc/network/ interfaces. Filen interfaces åbnes med en editor eks. mcedit. Følgende kommando indtastes for at få adgang til filen: *mcedit interfaces.* Billedet herunder viser opsætningen:



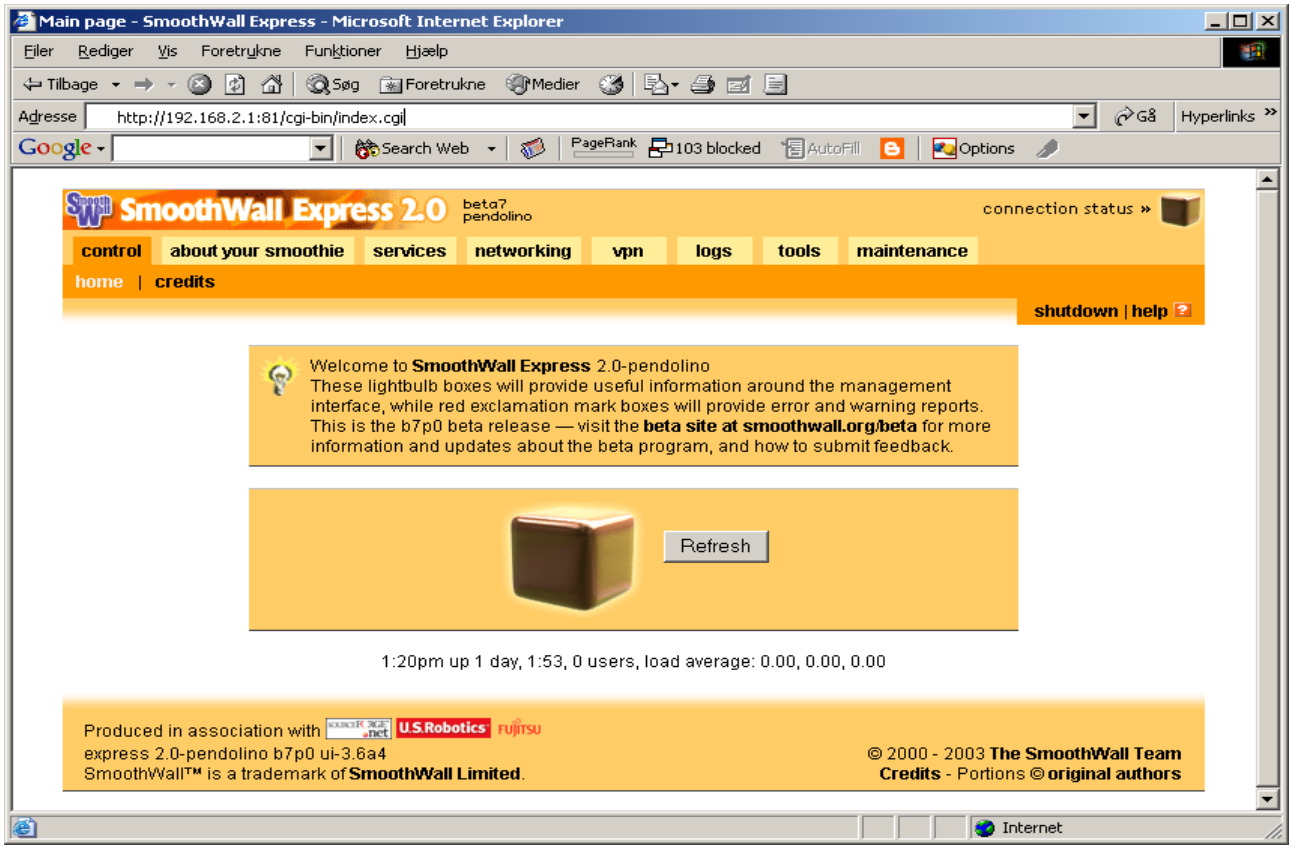
IP-indstillingerne er indtastet manuelt, idet DHCP ikke er aktiveret i SmoothWall'en og der er ikke opsat en DHCP server på LAN'et. DNS indstillingerne opsættes i /etc/ resolv.conf. Filen åbnes med mcedit. Billedet herunder viser DNS indstillingerne:



DNS er aktiveret så det er muligt at opdatere serveren via *apt-get install* der henter pakker fra Internettet. DNS Serveren er 192.168.1.3, hvilket er en ekstern server SmoothWall'en har forbindelse til via *RED* netværkskortet.

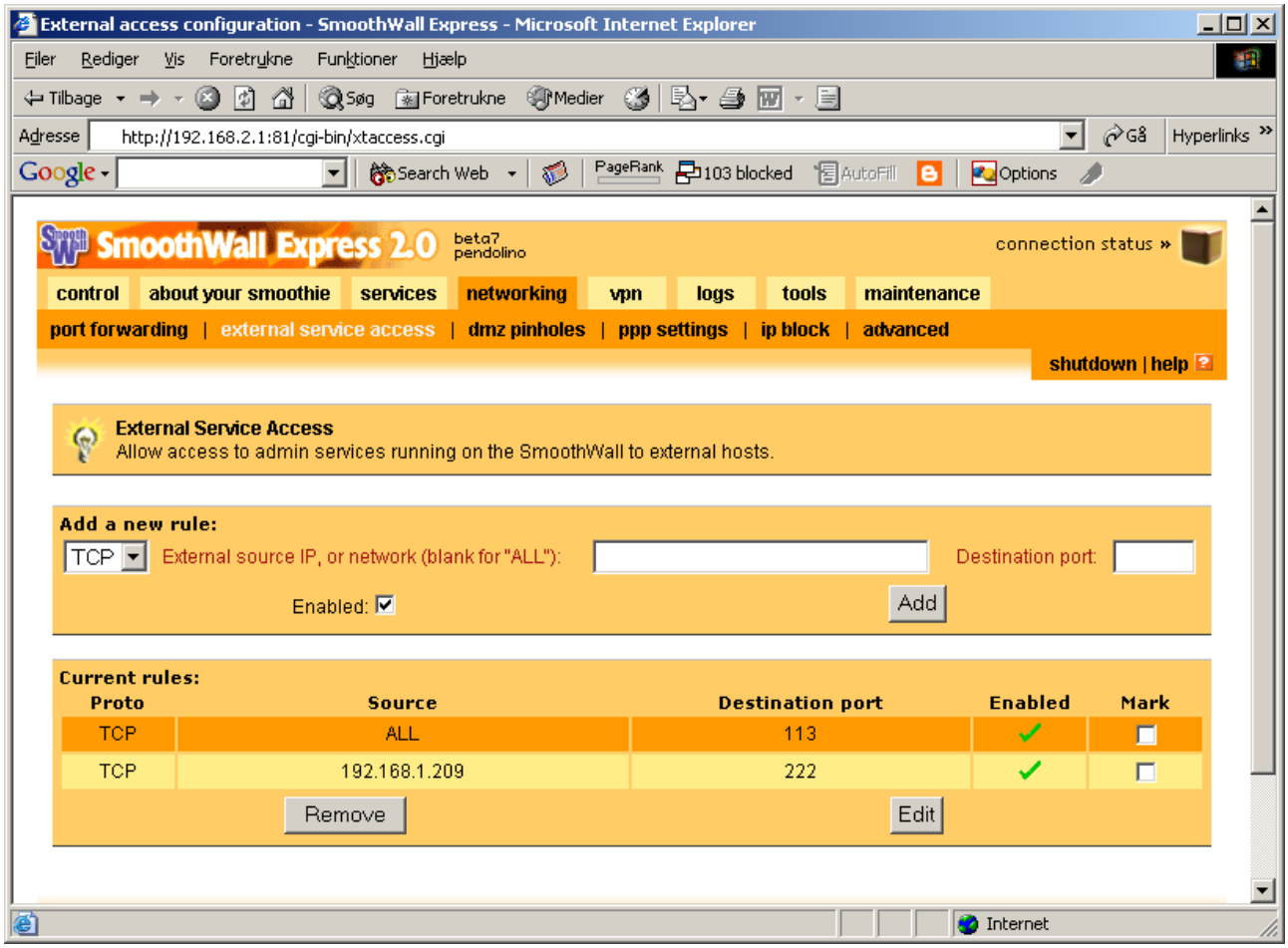
Kap. 4 - Konfiguration af SmoothWall

I forbindelse med firewall opgaven er der blevet udfærdiget en case, som firewallen skal konfigureres efter. Dette afsnit vil derfor tage udgangspunkt i hvordan casen er blevet løst på SmoothWall'en. Casen kan ses på bilag 1. Konfigurationen af SmoothWall foregår via et webinterfaces. Adgangen til webinterfaces sker fra en PC forbundet til det private netværk (*GREEN)*, idet trafik udefra som standard er blokeret. En browser på PC'en fra det private netværk åbnes, og IP adressen på SmoothWall'ens (*GREEN)* netværkskort indtastes efterfulgt af port nummer 81. I dette tilfælde indtastes http://192.168.2.1:81 og en login menu præsenteres. Til at logge ind anvendes admin brugeren, og passwordet man oprettede under installationen af SmoothWall. Når passwordet er blevet godkendt præsenteres man for SmoothWall'ens webinterfaces:

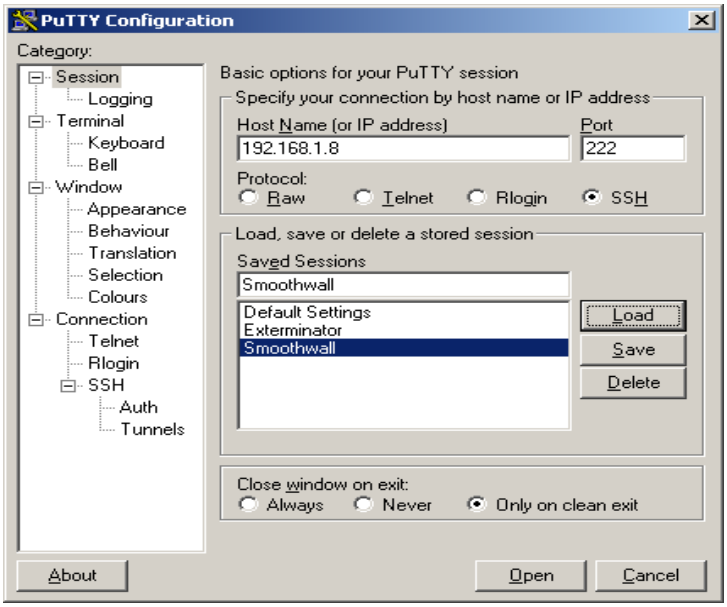


Ifølge SmoothWall's userguide er det muligt at logge på webinterfaces via en secure forbindelse. Den secure forbindelse anvender port 445, men det lykkedes aldrig at få det til at virke.

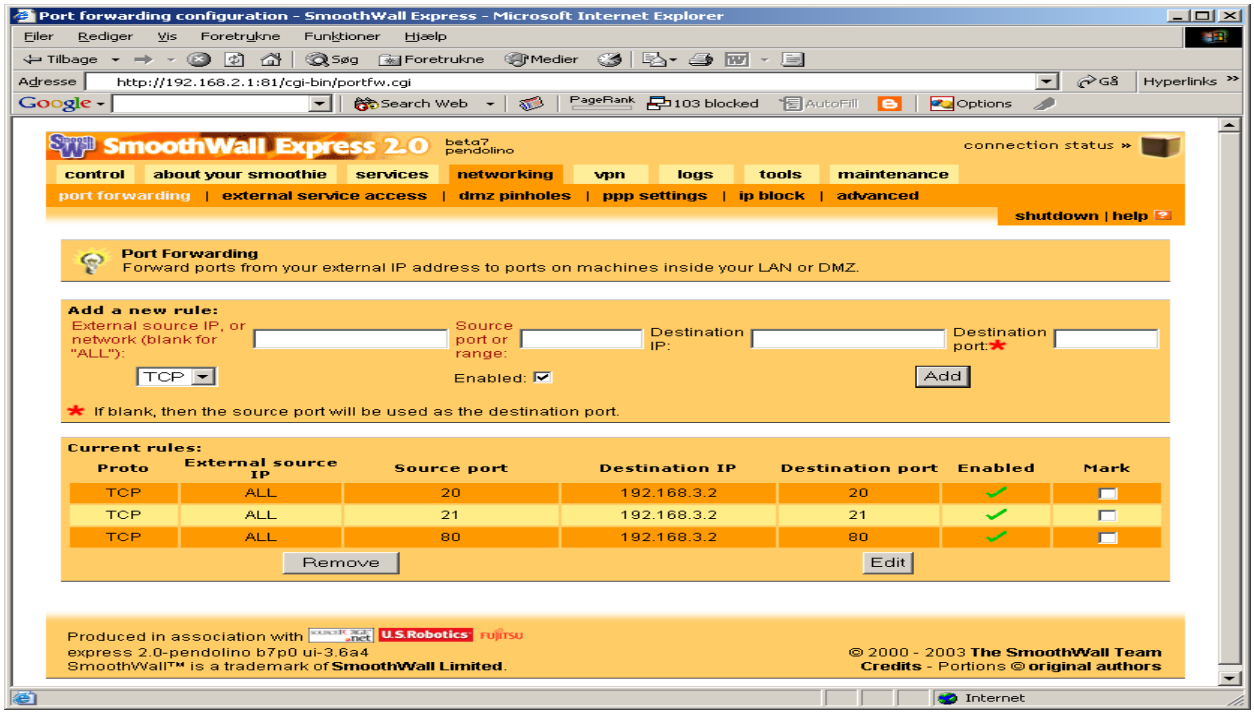
Det første der skal opsættes ifølge casen er adgang udefra til Firewallen. Administratoren skal have adgang til Firewallen fra hans hjemme computer via SSH (Secure Shell). I SmoothWall'en aktiveres SSH ved at vælges *services* → *remote access* → *SSH* → *Save* og ved at tilføje IP adressen som skal have adgang og DST port som SmoothWall lytter efter i *networking* → *external service access.* I dette tilfælde anvendes IP adressen 192.168.1.209, som er en ekstern IP der skal have adgang til SmoothWall. Det skal dog bemærkes at SmoothWall lytter efter port 222 ved SSH. Billedet herunder illustrere opsætningen:



Nu kan man afprøve SSH med programmet PuTTY. Eksemplet herunder illustere opsætningen af PuTTY:



Ifølge casen skal der også være adgang til FTP & Web Serveren i DMZ (*ORANGE)* netværket udefra. Dette opsættes i *networking → port forwarding*. I menuen gives adgang til *ALL*, *External source IP* og *Destination IP* er adressen hvor FTP og Web serveren er installeret. I dette tilfælde er det IP adresse 192.168.3.2 i DMZ (*ORANGE)*. Portene der skal *forwardes* til *Destination IP* er henholdsvis port 20, 21 (FTP) og port 80 (Web, http). Eksemplet herunder illustrere opsætningen:

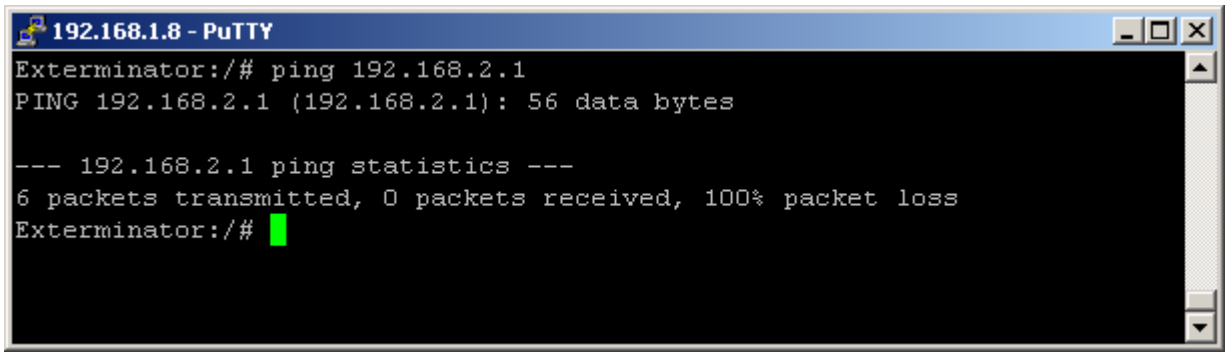


Det private netværks (*GREEN)* skal også have adgang til (DST) porte ud. Alle disse porte kan ses på bilag 1. I SmoothWall kan man dog kun konfigurerer, hvilke porte der skal være åben indadtil, men ikke ud af til.

Administratoren skal også have adgang via SSH (222) til SmoothWall fra administratorens IP adresse på det privat netværket (*GREEN)*. Som standard har alle på det private netværk (*GREEN)* adgang til SSH, idet det ikke er muligt at definere i SmoothWall, hvilke IP adresser der har adgang til forskellige services (porte) på det private netværk. Det eneste alternativ er at lave IP block, men så får IP adressen ikke adgang til nogle af servicerne der går igennem SmoothWall'en.

Det private netværk (*GREEN)* skal også have adgang til de services der kører på serveren DMZ (ORANGE). Dette er FTP (20, 21) og Web (80). Dette gøres som tidligere beskrevet ved at forwarde portene til serveren på (DMZ) i *networking → port forwarding*.

Ifølge casen skal DMZ (*ORANGE)* ikke have adgang til det private netværk (*GREEN)*. Det er som standard opsat i SmoothWall'en at DMZ ikke har adgang til det private netværk. Med ICMP (Ping) kan man teste om det virker som det skal. Følgende eksempler viser om DMZ kan kontakte det private netværk:



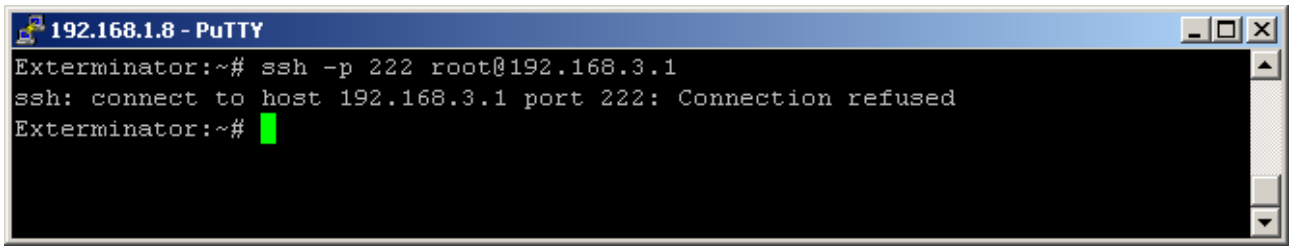
Hvis man vil have adgang til det private netværk fra DMZ skal det opsættes i SmoothWall. Det opsættes i *networking → dmz pinholes*.

Der skal hellere ikke være adgang til det private netværk udefra. Dette er også som standard opsat i SmoothWall, at der ikke er adgang til det private netværk. Hvis man vil have adgang til

det private netværk udefra skal det opsættes i *networking* → *external service access*. I menuen indtastes hvilken IP der skal have adgang og *DST* porten. Porten der indtastes i *external service access* skal selvfølgelig forwardes til serveren på det private netværk, som lytter på den valgte port.

DMZ (*ORANGE)* skal have adgang til ICMP, DNS og Web, FTP, SSH (DST) porte ud. I SmoothWall kan man dog kun konfigurerer, hvilke porte der skal være åben ind af til, men ikke ud af til.

DMZ (*ORANGE)* skal også have adgang via SSH (222) til SmoothWall'en. Det er dog ikke muligt at lave denne opsætning, idet SmoothWall ikke tillader dette. Herunder ses et forsøg på at kontakte SmoothWall fra DMZ via SSH (222):



SmoothWall'en skal have fuld adgang til det private netværk (*GREEN)* og DMZ (*ORANGE).* Dette er opsat som standard i SmoothWall.

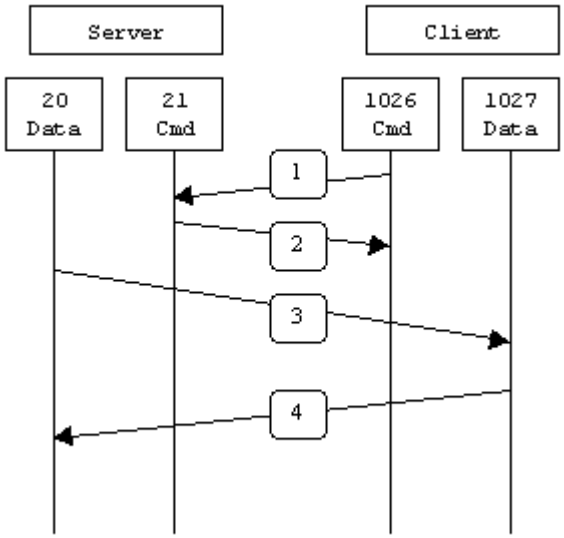
Kap. 5 - FTP problematik i forbindelse med Firewalls

FTP

File Transfer Protocol (FTP) anvendes til at overføre filer mellem computere. FTP er en applikationsorienteret protokol der er placeret i ISO lag 5, 6 og 7. FTP er udelukkende TCP orienteret, og fungere efter client/server princippet, hvor en client foretager en forespørgsel (Request) hos en server, og serveren svarer tilbage (Response) til clienten. Clienterne er initiativtager til transaktionerne, mens serveren hele tiden lytter efter forespørgsler. FTP kan overføre filer mellem to computere der har hver deres filsystem. Derudover tilbyder FTP autentifikation af bruger ID og password. Der anvendes kun et lavt sikkerhedsniveau i FTP, da password sendes i klar tekst. FTP anvender 2 porte, en *data* port og en *command* port (også kendt som kontrol porten). Traditionelt set er port 21 til *command* porten og port 20 til data porten. Dette er dog ikke altid tilfældet, idet dette afhænger af hvilken *mode* FTP er opsat til, og derfor er data ikke altid til port 20. FTP kan opsættes til at fungere i enten *Active mode* eller *Passive mode*. Forskellen beskrives herunder.

Active FTP

I *Active mode* sender klienterne en random *SRC* port (port1 > 1024) til FTP serverens *DST command* port 21. Klienten begynder at lytte på port1 *command* og port2 *data* (eks. port 1026 og 1027) og sender FTP kommandoen *PORT* (port2) til FTP serveren. FTP serveren vil derefter forbinde til clientens specifikke *data* port (port 2) fra dens lokale *data* port 20. Det primære problem med *active mode* er klient siden. Klienten laver ikke en egentlig forbindelse til data porten på serveren, idet klienten kun fortæller serveren hvilken port den lytter på og serveren forbinder til den specifikke port på klienten. Fra klientens firewall side ser det ud til at være en udefrakommende forbindelse til en intern klient, hvilket normalt er blokeret. En illustration af *Active mode* kan ses herunder:



1: Klienten kontakter Serverens *command* port 21 (*DST)* med *command* port 1026 (SRC*)*. Klienten begynder at lytte på port 1026 og 1027, og sender kommandoen *PORT* 1027.

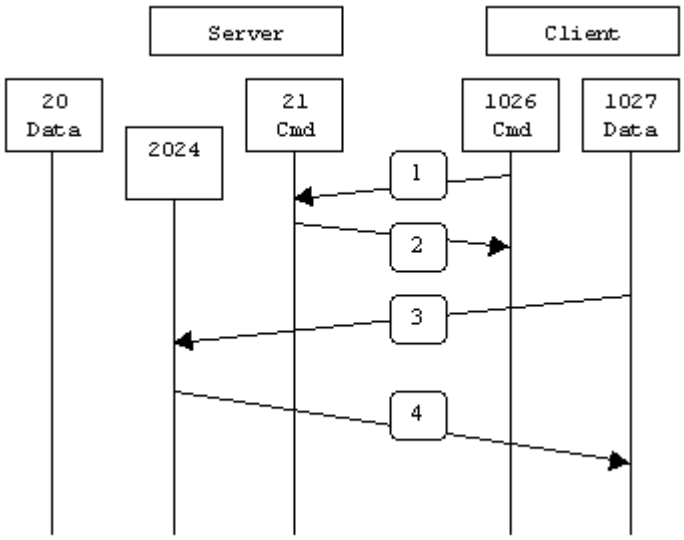
2: Serveren sender en *ACK* tilbage til klientens *command* port 1026.

3: Serveren starter en forbindelse på dens locale *data* port 20 til *data* port 1027 hos klienten.

4: Til sidst sender klienten en ACK fra *data* port 1027 til *data* port 20 hos Serveren.

Passive FTP

For at løse problemet med at serveren starter forbindelsen til klienten er der blevet udviklet en anderledes metode til FTP forbindelse. Denne metode er kendt som *passive mode* eller *PASV*. I *passive mode* er det klienten der starter begge forbindelser til serveren, og derved løser problemet med at firewalls blokere den indkomne *data* port forbindelse til klienten fra serveren. Når en FTP forbindelse oprettes er det klienten som åbner 2 random *SRC* porte (port1 og port2 > 1024). Den første port kontakter serveren på *DST* port 21, men i stedet for at udstede *PORT* kommandoen og tillade serveren, at forbinde tilbage til klientens data port vil klienten udstede en *PASV* kommando. Resultatet af det er at serven åbner en random (P > 1024) og sender *PORT P* kommando tilbage til klienten. Klienten starter nu forbindelsen fra port 2 til port P på serven til at overføre data. En illustration af *Passive mode* kan ses herunder:



1: Klienten kontakter Serverens *command* port 21 (*DST)* med *command* port 1026 (SRC*)* og udsteder *PASV* kommandoen.

2: Serveren svarer med kommandoen *PORT* 2024, som fortæller klienten at serveren lytter efter data på port 2024.

3: Klienten starter nu data forbindelsen fra dens *data* port 1027 til serverens *data* port 2024.

4.: Til sidst sender serveren en *ACK* fra dens *data* port 2024 til klientens *data* port 1027.

Imens *Passive Mode* løser mange af problemerne fra klientens side, så dukker der en række problemer op på server siden. Heldigvis tillader mange FTP daemons, at specificere en række porte, som FTP serveren kan anvende. Det andet problem involverer FTP klienter der ikke supporterer *passive mode*. Et eksempel er det medfølgende kommando prompt FTP program i Solaris. Nogle browsere understøtter også kun *passive mode* via ftp://URLs. Dette kan enten være godt eller dårligt afhængig af hvordan serverne og firewalls er konfigureret.

Installation af Proftpd

Proftpd er et FTP server program til Linux. Programmet til Linux kan installeres på flere forskellige måder, men langt det nemmeste er med apt-get pakkesystemet i Debian. Pakken installeres med følgende kommando:

apt-get install proftpd

Under konfigurationen af proftpd kan man vælge om man vil køre programmet, som *standalone* eller *inetd*.

Hvis man har valgt at køre med *inetd* skal proftpd genstartes med følgende kommando:

/etc/init.d/inetd restart

Hvis man har valgt at køre med standalone skal proftpd genstartes med følgende kommando:

/etc/init.d/proftpd restart

Forskellen på standalone og inetd er, at ved standalone ligger *demon* (programmet) hele tiden i hukommelsen, hvorimod ved inetd bliver *demon* først startet når der laves en forespørgsel. Når *standalone* anvendes er det mere ressource krævende, men til gengæld er performancen bedre end inetd.

Konfigurationen af proftpd kan altid reconfigures med følgende kommando:

dpkg-reconfigure proftpd

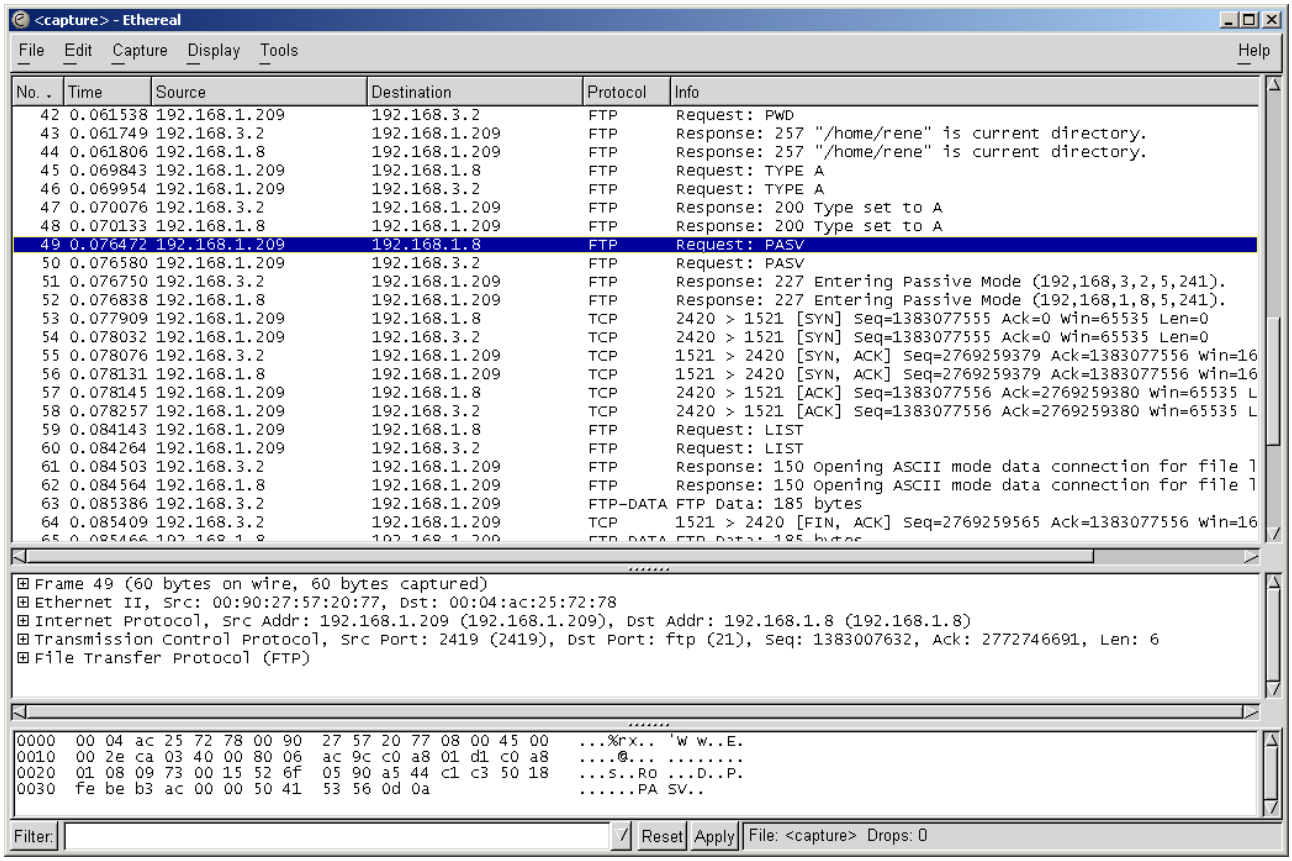
Følgende kommando viser hvilken verison af proftpd man har installeret:

apt-cache show proftpd

Bemærk at der skal oprettes bruger med kommandoen *adduser* før man kan logge på FTP serveren. Når brugeren er tilføjet bliver der oprettet en mappe i /home. Oprettede bruger kan ses i filen /etc/passwd.

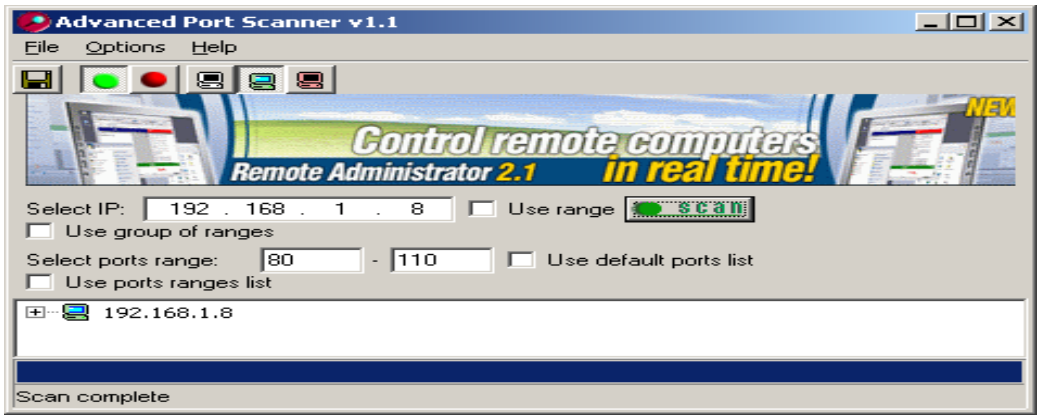
Med programmet Ethereal kan man undersøge, hvilken mode FTP serveren og FTP klienten er opsat til. Her skal man være opmærksom på at der sendes en PASV kommando fra FTP klienten til FTP Serveren, hvis *passive mode* anvendes. Klient FTP programmet FlashFXP til Windows kan anvendes til at lave en *passiv* forbindelse til FTP serveren. Det kan enables i *Option → Preferences → Proxy / Firewall / Ident→ Use Passive Mode*. Et eksempel herunder illustere at der bliver sendt en *PASV* kommando fra FTP klienten (192.168.1.209) igennem SmoothWall

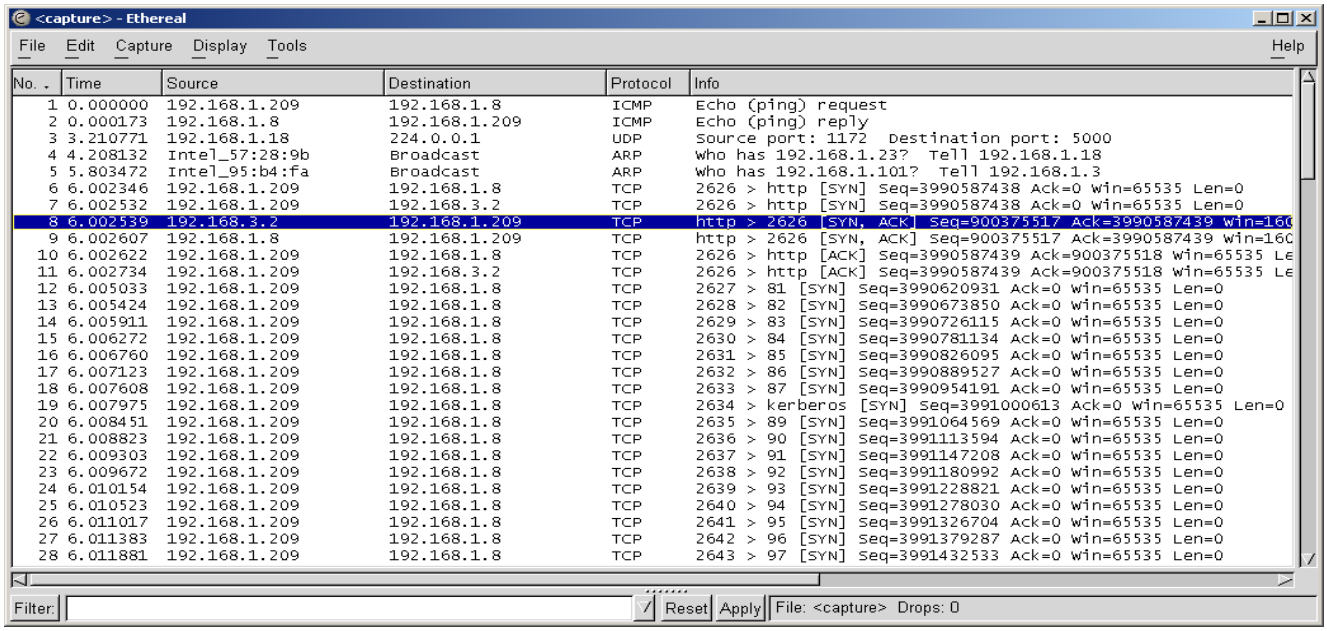
(192.168.1.8), og til FTP Serveren (192.168.3.2). Desuden kan man se at klienten (192.168.1.209) anvender en *data SRC port* 2420 og sender den til *DST port* 1521 hos FTP serveren (192.168.3.2). Port 1521 hos FTP serveren er random generet, idet serveren anvender *passive mode*.



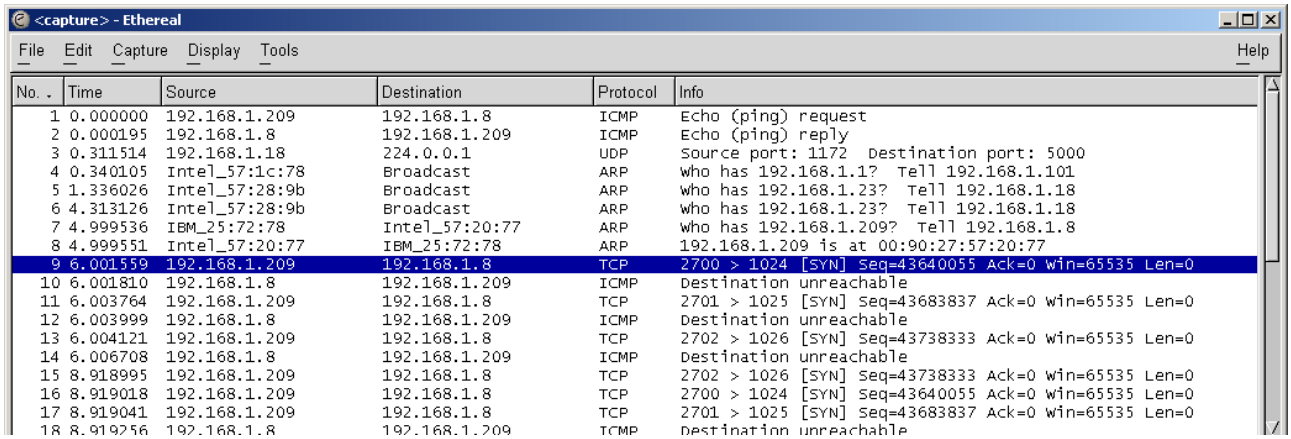
Det mest besynderlige ved det hele er, at man skulle tro at portene der bliver random generet af serveren, bør være manuelt forwardet til serveren i SmoothWall. Dette er dog ikke tilfældet, idet SmoothWall automatisk forwarder de random generede porte, som ovennævnte eksempel viser, hvis der laves en *passive mode* FTP forbindelse. En anden smart feature er, at FTP server programmet proftpd selv registrerer om det er en *aktive* eller *passive* forbindelse der oprettes.

Med programmet Advanced Port Scanner kan man kontrollere om der er andre porte der bliver forwarded videre, end dem specificeret i SmoothWall's *port forwarding*. En port range fra 80 til 110 testes af med Advanced Port Scanner. I SmoothWall er det kun port 80 (Web) der er forwarded i dette tilfælde. Ethereal startes og derefter startes port scanningen fra 80 til 110 som billederne herunder illustere:





Det ses tydeligt at kun port 80 (Web) bliver forwarded videre til Web serveren (192.168.3.2). Porte over 1024 kan også afprøves:



Resultatet er det samme, og det konkluderes at det kun er ved *passive* FTP, at der automatisk bliver forwarded til FTP serveren.

Kap. 6 – Sikkerhedsvurdering

Som administrator på en eller flere firewall(s) skal man være opmærksom på, at man aldrig bliver færdig. Administratorer der opsætter alle regler og overholder alle anbefalinger for dernæst at læne sig tilbage uden at kigge mere på løsningen, vil få en grum overraskelse. Det er et spørgsmål om tid. Som ansvarsbevidst administrator på sikkerheden, sidder man på nåle hele tiden. Der findes til stadighed "exploits" i diverse applikationer, servere og endda firewall produkter. Disse huller skal lukkes, helst inden de opdages. Det er altså vigtigt at vide hvilke applikationer / servere som potentielt kan indeholde brister og følge med på alt hvad der kan kompromittere disse. Ud over det skal administratoren hele tiden være bevidst omkring det der sker på netværket. En god løsning kunne her være at analyserer trafikken på netværket via protokol sniffere. Dette kræver dog noget speciel hardware. Men man kan også bruge en hub i stedet, med det ville være med til at sløve netværket kraftigt. Men heller ikke her kan man være 100 % sikker på at netværket er sikkert.

Men der er faktisk også produkter/løsninger der specielt er baseret på at finde huller i dine netværk. Man kan købe sig til sådanne produkter eller få nogle til at lave en sikkerhedsanalyse af netværket. De tager ud i virksomheden og tester netværket og efterlader efterfølgende en rapport om de sikkerheds brister man har på netværket. Men igen kan man ikke være 100 % sikker på at netværket er uden sikkerhedshuller. På bilag 2 kan man læse en artikel ”Om sårbarhedsanalyse er godt nok” og nedenunder er der givet et eksempel på hvordan en hacker kan infiltrer et netværk, selv om man har sat en firewall op, samt ”Andre angreb på dit netværk”!

Hvordan hackere kan infiltrere det interne netværk via en klient der åbner for en port ud igennem firewallen ?

Det er blevet installeret en firewall i virksomheden, som blokerer al indkommende trafik. Hvis en hacker skulle prøve at scanne portene vil hackeren ikke finde åbne porte. Men problemet er, at hvis en medarbejder på det interne net, får lagt en ”trojansk hest” ind på sin computer uden at være bevidst om det. Den ”trojanske hest” vil eks. automatisk forbinde computeren igennem en port til en server på Internettet. Via den SRC port medarbejderens har åbnet ud igennem firewall kan en hacker opkoble sig til medarbejderens computer og infiltrer det interne netværk. Når

hackeren først er inde kan hackeren ødelægge eller få adgang til sårbare informationer. Hackeren kan også sprede vira på netværket.

Vira og orme på netværket ?

Nu har vi været omkring de mest almindelige angreb på netværks, såsom ”Trojanske heste”, åbne porte. Men her stopper sikkerhedsrisicis ikke. En anden sikkerheds brist på netværket kan være ”pop3”. Hvis man har en mail server på DMZ-zonen og klienterne henter deres mail via mail-serveren, kan netværket blive infiltreret denne vej. F.eks. hvis der findes en vira i en af klienterne e-mail. Hvis en personen åbner mailen vil vira’en blive afviklet og den vil derefter sprede sig til de andre klienter/servere på netværket. Denne risiko kan dog afgrænses, ved at man har en virus scanne installeret på både serveren og klienterne. Fejl i programmer og operativsystemer kan også medføre sikkerhedshuller. Dette var bl.a. skyld i ”Blaster-ormen” fremkomst. En brist i Windows operativsystemet gjorde det muligt for ormen at tvinge computeren til at genstarte, efter et givent antal sekunder.

Test af Smoothwall

Efter opsætningen af Smoothwall’en lavede vi en test på forskellige kriterier vedrørende firewallen. Nogle af de ting vi testede var bl.a. adgangen til og fra åbne porte (http 80), forwarding af trafik, port scan på de forskellige net, adgang til de forskellige net, analyse af netværkstrafik over FTP osv. Testen viste, at når vi forwardede trafik fra port 80 til Web-serveren, gik det kun til web-serveren, som det skulle. Ved at lave en port scan på firewallen fik vi bekræftet at kun port 80 var åben og det var også korrekt. Adgangen fra Internet til det lokale net var spæret hvilken det også skulle være, men computerne på det lokale net kunne uden problemer få kontakt til diverse servere på Internettet. Fra det lokale net kunne man også uden problemer få fat i DMZ zonen’s servere, mens omvendt var der ingen kontakt (fra DMZ til det lokale net). Ved test af FTP var der en specielt problem, nemlig når en klient fra Internettet forespørger på en FTP forbindelse inde på DMZ-zonen, skulle dette kun, kunne foregå i passiv mode, men fordi Smoothwall er konfigureret til at skelne mellem dette, virkede det både i Aktiv og Passiv mode. Testen viser at firewallen er blevet konfigureret rigtig og fungere efter den CASE der er blevet stillet. Vi havde dog et problem vedrørende Smoothwall og det var, at når man connecte til web-interfacet, kan man gøre det via en sikker port, nemlig port 445. Men dette kunne vi ikke få til at virke, så vi har kun lavet web-interfaces via port 81.

Konklusion

Vi kan konkludere, at vi har fået løst opgaven der blev stillet, og vi har fået en bedre forståelse for teorien bag firewalls. Samtidigt er teorien vi har gennemgået blevet omsat til et praktisk projekt, idet vi har opnået praktisk erfaring i forbindelse med opsætning af SmoothWall. Hvordan en firewall fungere, og opsættes i et netværk. FTP problematikker i forbindelse med firewalls. Hvilke sikkerheds politikker der skal opsættes for at have et relativt sikkert netværk, samt hvordan nettene kan vedligeholdes sikkerhedsmæssigt.

SmoothWall er en god løsning til private og mindre/mellemstore virksomheder, som har fokus på sikkerheden. Softwaren er brugervenlig og nem at konfigurere via webinterfacet, men der mangler dog nogle mere avancerede features i forhold til kontrol af DST porte ud af firewallen m.m. Taget i betragtning af, at SmoothWall er gratis og brugervenlig, er den værd at tage med i sin overvejelse, ved implementering af en firewall.

Som vi tidligere har nævnt kan man aldrig opnå et 100 % sikkert netværk, hvis netværket er tilkoblet Internettet. Men man kan dog komme tæt på en rimelig sikker løsning, dette har vi været inde på i sikkerhedsvurderingen. På baggrund af det, kan vi konkludere, at der er to måder en firewall kan opsættes på (sikkerhedsmæssig). Den første er at man helt lukker for trafikken ind –og ud af firewallen, foruden port 80 (http), som giver klienterne rettighed til at surfe på nettet. Den anden løsning er, at lukke for alt indad gående trafik, men derimod tillade alt udad gående trafik. Dette giver klienterne fri adgang til alle portene udadtil, mens alle indadtil er alle lukkede (på nær de porte der er forwardet til serverne på DMZ).

Litteraturliste

WEB-adresser:

http://www.smoothwall.org/ - Smoothwall´s hjemmeside (Guider til opsætning og konfiguration).

http://www.bignosebird.com/notebook/proftpd.shtml - Opsætning af proftpd i Linux.

http://slacksite.com/other/ftp.html - Aktiv & Passiv mode i FTP.

http://a.area51.dk/sikkerhed/ordforklaring - Firewall teori.

http://cvs.linux.bog.dk/alle/bog/sikkerhed-virkemaader.html - Firewall virkemåder.

http://www.debianguiden.dk - Guide til debian

Bøger:

Data Kommunikation 4. udgave af Stig Jensen og Arne Gjelstrup – ISBN 87-571-1956-2

Bilag

Bilag 1 – Udleveret CASE

Bilag 2 – Sårbarhedsanalyse er godt – men ikke nok

Bilag 1 – Udleveret CASE

Case

Plot:

Denne case simulerer de forskellige rettigheder i en given virksomhed. Virksomheden skal indeholde et privat netværk med de ansattes PC'ere og derudover en DMZ med FTP og WEB-server. Brugerne bag Firewallen skal have mulighed for at anvende diverse applikationer på Internettet. Med udgangspunkt i ovenstående er følgende rettigheder udarbejdet:

NB: Vi går ud fra at Administratoren har en PC på det private netværk.

Rettigheder:

Udefra til Firewallen:

SSH fra administratorens hjemme IP.

Udefra og ind DMZ Netværket:

FTP'en & Web'en skal kunne ses udefra.

Private Netværket til ud:

Brugerne skal have tilladelse til følgende porte ud til Internettet.
MSNM (1863), Skype (33033), ICQ(TCP 5190), Web (80), DNS (53 udp), ICMP, FTP (20,21), HTTPS (443) , POP3 (110) , SMTP (25) , Samba (137,138,139), IRC (6667, 6668,6669), SSH (22)

Private Netværket til Firewall:

SSH (22) fra Administratorens IP.

Private til DMZ Netværket:

De skal have tilladelse til de services som der kører på serveren:
FTP (20,21)
Web (80)

DMZ til Private: Ingenting.

Udefra til Private: Ingenting.

Fra DMZ og ud:

ICMP, DNS, web, FTP, SSH, HTTPS.

Fra DMZ til Firewall:

SSH.

Fra Firewallen til det private netværk og DMZ:

Fuld adgang.

Bilag 2 – Sårbarhedsanalyse er godt – men ikke nok

Herlev, 09.01.2001

Sårbarhedsanalyse er godt - men ikke nok

En test af 8 af de mest populære produkter til sårbarhedsanalyse / sikkerhedsscanning af systemer på et netværk har vist, at det kan give en falsk tryghed at basere sin sikkerhedsvurdering 100% herpå.

I testen blev hvert enkelt produkt sat til at analysere / scanne et system med 17 forskellige kendte sårbarheder, og ingen af produkterne formåede at lokalisere og neutralisere alle 17. Så uden at blive specifik om de enkelte produkter kan vi anbefale at kigge på det vedhæftede analyseresultat og konkludere: Det kan være risikabelt at stole på, at en sårbarhedsanalyse alene giver et korrekt billede af netværkets sikkerhedsstatus.

Den stadig større, hurtigere og billigere adgang til Internettet medfører en risiko. Du kan blive udsat for hack, crak, spoof, spam, denial of service, scanning, trojanske heste, virus, snifning og flere andre ikke så rare begreber. Men dette er ikke en grund til ikke at benytte de mange muligheder, som Internettet giver. Netværket kan jo beskyttes. Det kan beskyttes mod alle former for angreb, og uanset hvilken risiko man bliver konfronteret med, findes der produkter og forholdsregler, der gør det muligt stadigt at benytte Internettet på en sikker måde. En firewall vil typisk være en god løsning, men det kan være risikabelt at stole på, at firewallen er nok. Analyse af logfilerne fra firewallen er et godt supplement, ligesom såkaldte Intrusion Detection systemer er at anbefale.

I dag findes der dog en stor risiko, som kommer indefra. Talrige undersøgelser har vist, at de fleste brud på sikkerheden i en virksomhed sker indefra. Men også her hjælper ovennævnte produkter. Ved at anskue IT sikkerhed fra flere vinkler er der større chance for, at alle sårbarheder på et netværk bliver fundet.

Vulnerability Scanners Detection Result

Kilde: SANS NewsBites Vol. 3 Num. 01