Firewalls

© 2013 jan.celis@kdg.be

Inhoud

1 Wat is een firewall?

The best way to make a fire with two sticks
is to make sure one of them is a match
— Will Rogers

Veel configuraties met routers en computers worden tegenwoordig als *firewall* bestempeld.
Wat we hier meestal mee bedoelen is een soort hindernis, die ervoor zorgt dat er geen ongewenst verkeer naar het interne netwerk kan komen. In het ideale geval kan een ongewenste internetreiziger niet door de firewall én er niet in slagen om enige informatie te ontdekken die we wensen geheim te houden.
Het woord 'firewall' is afkomstig van de autoindustrie. Een firewall bevindt zich tussen de passagiersruimte en de motorkap. Wanneer de motor vuur zou vatten, beschermt de firewall de chauffeur en de passagiers tegen de vlammen.

1.1 Firewall technieken

Kenmerkend voor een firewall is dat deze gebruik maakt van een combinatie van volgende tools en technieken:

a) Controle op het doorgeven van netwerkinformatie (of doelbewust het doorgeven van misleidende informatie). We kunnen het moeilijk maken om uit te zoeken welke computers er aan de andere kant van de firewall hangen, of om deze computers te bereiken.

b) Toegangslijsten (ACL's):
We kunnen net als bij de gastheer van een exclusief feestje, een lijst opstellen van uitgenodigde netwerkgebruikers. Bij routers kunnen we bepalen welk verkeer we toelaten afhankelijk van de herkomst, het doel en de soort service.
We kunnen er bij onze server voor zorgen dat we bepaalde netwerkservices afschermen met zogenaamde *wrapper* programma's. Deze wrappers nemen de plaats in van daemons zoals FTPD en telnetd. Wanneer er een aanvraag tot een verbinding binnenkomt, wordt de wrapper aangeroepen. Eén keer dat deze vaststelt dat de aanvraag van een "goedgekeurde" bron komt, zal deze de geschikte daemon opstarten. De daemon zal dan de verbinding afhandelen.

c) Proxies. Voor een nauwsluitende firewall kunnen we proxy servers oprichten. Proxy servers kunnen verkeer onderscheppen, nakijken en doorgeven aan een doelbewust beperkte verzameling van toepassingen of netwerkdiensten.

Netwerk-niveau proxies werken op een laag niveau. Ze loggen en geven pakketten door zonder inzicht in de informatie die wordt uitgewisseld.

Toepassings-niveau proxies ontrafelen de pakketten en geven aanvragen door. Deze proxy servers verstaan meestal de vorm van de informatie die doorgezonden wordt. Ze kunnen ook meer verstaanbare loggings bijhouden. Uiteraard is er een beperking aan de verstaanbaarheid van applicaties.

Verschillende van deze firewalls hangen af van een identify-and-trust relatie. We identificeren de herkomst van het verkeer en beslissen dan of het al-dan-niet mag doorgaan. Het is natuurlijk belangrijk om de firewall zo te configureren dat enkel de betrouwbare uitwisselingen mogen doorgaan. Een meer subtiel risico komt met het eerste gedeelte van de identify-and-trust relatie, de identificatie. Ik kan bijvoorbeeld zeggen “Netwerk X mag een conversatie van type Y aangaan met mijn computer Z.” Maar hoe kan ik er zeker van zijn dat ik echt met netwerk X spreek? Misschien is X wel mijn belastingscontroleur. Vragen zoals deze maken dat de configuratie van firewalls in een mistig grijs gebied blijft. We moeten niet enkel selecteren uit onder de toegelaten vrienden, we moeten ze ook nog eens wantrouwen.
Je netwerk beveiligen tegen de duistere krachten uit de omgeving is het hoofddoel van een firewall.

Je kan ook een firewall richten naar de andere kant, naar je eigen mensen.
Een firewall kan de soort Internet diensten beperken die de mensen van binnen je bedrijf kunnen gebruiken. Er kan evengoed een log van hun activiteiten bijgehouden worden.
Uiteraard zijn er een aantal dingen die niet door een firewall kunnen gebeuren. Een firewall kan bijvoorbeeld niet verhinderen dat een onguur figuur in jou bedrijf het Geheime Drank Receptje naar buiten stuurt. Het kan ook geen onderscheid maken tussen een gewoon databestand en een bestand besmet met een virus. Firewalls werken op een lager niveau en zullen meestal bepaald netwerkverkeer doorlaten of tegenhouden afhankelijk van de herkomst, het doel of het type, niet de inhoud. Wanneer we veronderstellen dat we een aantal betrouwbare computers en diensten kunnen kiezen, dan moeten we hopen dat de eigenaars of gebruikers ervan ook betrouwbaar zijn.

Opmerking:
Er bestaan ook enkele programma’s of toevoegingen, die wel naar de inhoud van de mail kunnen gaan kijken, en de mail eventueel kunnen tegenhouden.

Besluit:
Een firewall is geen vorm van veiligheidsbeleid. Het is slechts een tool om een machine of een netwerk te beveiligen. Je kan een firewall gebruiken om bepaalde aspecten van beveiliging op te leggen. Er zijn wel enkele belangrijke dingen die men zich moet afvragen eer men over de instellingen van de firewall begint de discussiëren. Zorg ervoor dat je eerst het algemene veiligheidsbeleid van je bedrijf vastlegt, eer je met de configuratie begint.

1.1 Configuraties

Stel je een straat voor waarin allemaal identieke kleine wagens staan geparkeerd. In de wagens heeft men achteloos de sleutels laten rondslingeren tussen enkele kranten, blikjes cola en een paraplu.

Op de hoek van deze straat staat de enige uitzondering op deze wagentjes. Een rode, splinternieuwe Maserati. Gesloten, een extra anti-diefstalslot op het stuurwiel en binnenin een aantal grommende Rottweilers met bebloede tanden. Wat is de meest begeerde wagen om te stelen? Dit hangt af van de dief natuurlijk. De meeste van

ons zullen het ermee eens zijn dat een kleine wagen niet origineel is en een beetje saai om mee rond te toeren. De sportwagen daarentegen is wel aanlokkelijk om mee te pronken.

De moraal van deze straat is dat anonimiteit en saaiheid meer bescherming bieden, dan iets wat opvalt. Wanneer ik je niet opmerk of niet verwacht dat ik iets interessants kan vinden bij jou, dan zal ik ook geen moeite doen om bij je locatie binnen te vallen. Aan de andere kant is een slecht beschermde, saaie locatie een goede plaats om aanvallen te beginnen naar andere locaties. Daarenboven is een zwak beschermde locatie ideaal om je sporen uit te wissen.
Om even terug te komen op de straat. Misschien is een kleine, saaie wagen die niet opvalt wel ideaal om een misdaad mee te plegen, en neem je daarom de kleine wagen.
Waarschijnlijk zal jou locatie niet de diamant van het Internet zijn, maar een minimale bescherming is noodzakelijk om enkele verderfelijke figuren af te wimpelen.
Een “goede” bescherming kan een irriterende factor zijn voor gebruikers, zonder het gebruik van Internet te “zwaar” te maken, en dit kan verwezenlijkt worden zonder een fulltime firewall beheerder. Je moet zelf trachten af te wegen in welke mate of hoe zwaar je gaat beveiligen, maar het is aangeraden om minstens een minimale vorm van bescherming te gebruiken.

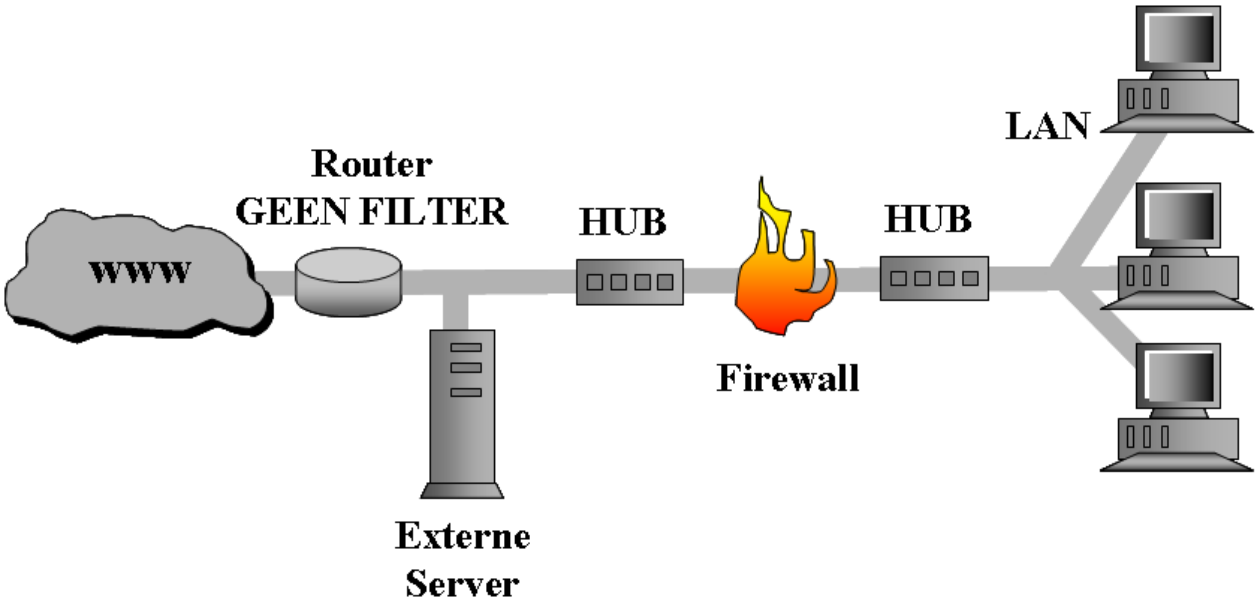
1.1 Firewall Architecturen

Er zijn een hoop mogelijkheden voor het gebruik van een firewall.
De eerste computer-firewall was een gewone (non routing) Unix host met verbindingen naar twee verschillende netwerken. De ene netwerk kaart stond in verbinding met het Internet en de andere naar de interne LAN. Om het Internet te bereiken moest je inloggen in de firewall (Unix) server. Je gebruikt dan de voorzieningen in het systeem om toegang tot Internet te krijgen. Je kon bijvoorbeeld X-windows gebruiken om met de Netscape browser de weergave op je werkstation te zien. De browser draait op de firewall en heeft dus toegang tot beide netwerken.
Dit systeem met een dubbele netwerkverbinding is leuk als je alle gebruikers kan VERTROUWEN.

Niemand kon een bestand rechtreeks downloaden naar een werkstation. Een bestand moet eerst gedownload worden naar de firewall, en kan dan gedownload worden van de server naar het werkstation.

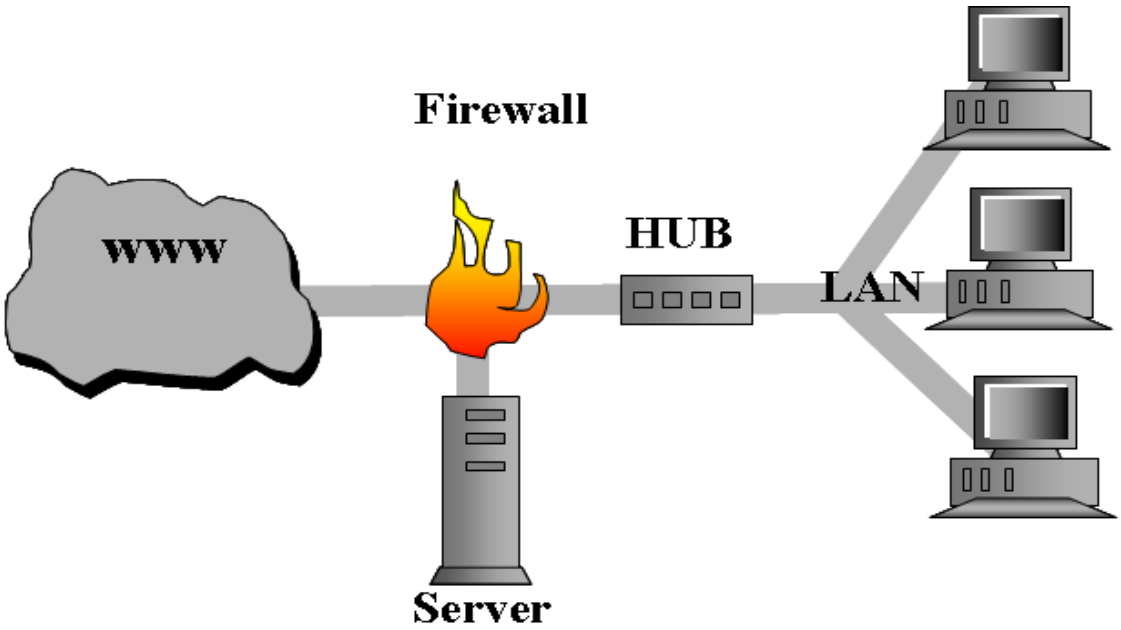
Opmerking:

99% van alle inbraken begint met het innemen van een gewone account van het systeem dat men wil aanvallen. Daarom is dit type firewall alles behalve aan te raden.
Wanneer je met het Internet verbonden bent via een router, dan kan je de router direct aansluiten aan je firewall. Of je kan door een extra switch ervoor zorgen dat je eigen servers buiten je firewall blijven.
Je zou enkele zware filters kunnen toepassen in je router. Het is echter ook mogelijk dat de router van jouw ISP is, en dat je zelf dus geen instellingen kan maken. Je kan ook je ISP vragen om enkele dingen te filteren. Over het algemeen zijn ISP’s echter niet zo happig op routeraanpassingen.



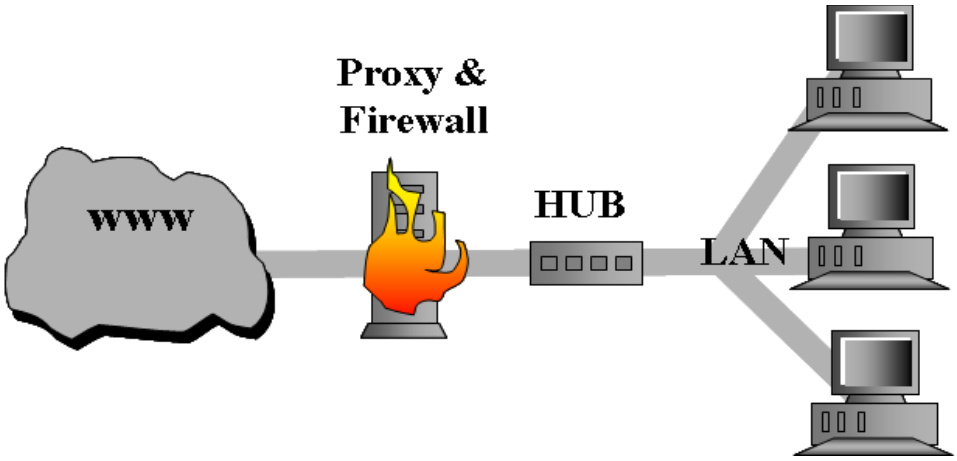
Figuur: Voor de firewall een hub met een externe server voor Internet diensten.

Het kan dat je gebruik maakt van een 'dialup' verbinding zoals een ISDN of ADSL lijn. Je kan dan gebruik maken van een firewall met twee netwerkverbindingen. Eén naar je server en één naar je hub. Dit geeft je nog steeds de controle over je Internet diensten en deze blijven gescheiden van je interne netwerk.



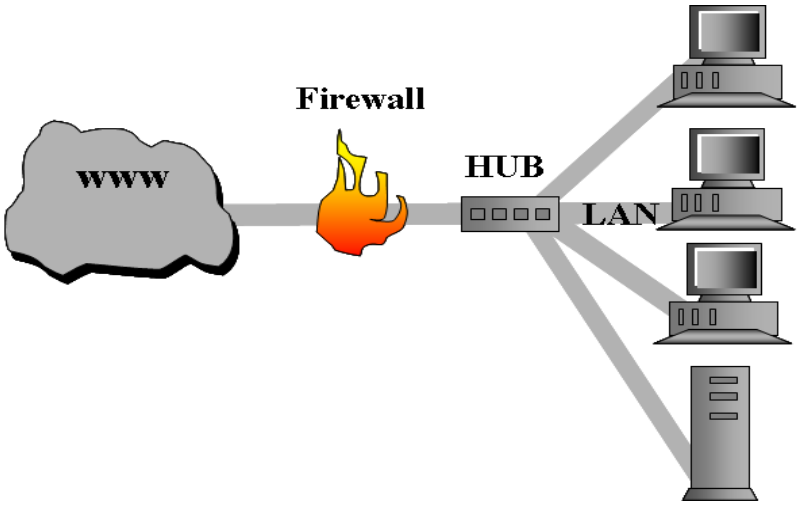
Figuur: Firewall met dubbele netwerkkaart met server voor Internet diensten.

Wanneer je zelf geen Internet diensten aanbiedt, maar als je in het oog wilt houden waar je gebruikers naartoe gaan, kan je best gebruik maken van een proxy server. Deze kan geïntegreerd worden in de firewall.



Figuur: Firewall met proxy server geïntegreerd.

Je kan de proxy server ook binnen de LAN installeren. In dit geval moet de firewall regels hebben die de doorgang van de communicatie met de proxy server verzekeren. Op deze manier kunnen de gebruikers zich in verbinding stellen met de proxy server om van Internet gebruik te maken.



Figuur: Firewall met proxy server binnen het Lokale Netwerk

Het is niet moeilijk om de controle over de communicatie op je netwerk te verliezen. Zorg dat je elke verbinding nakijkt. Er is enkel een achteloze gebruiker met een modem nodig om heel je beveiliging naar de maan te helpen.

2 Een minimale firewall

Anarchy may not be the best form of government,
but it's better than no government at all.

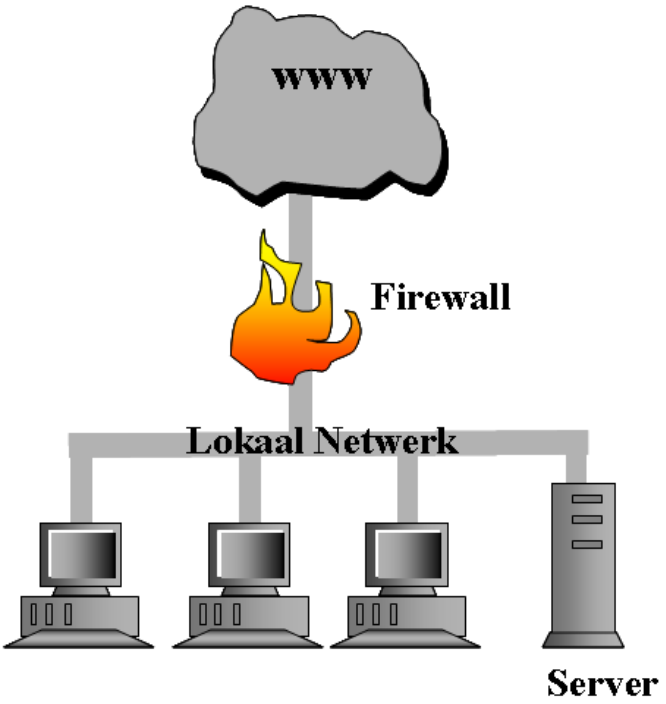
Een minimale firewall is opgebouwd uit volgende componenten:

Router toegangslijsten of Access Control Lists (Regels over de communicatie)
Application wrappers (Beveiliging van programma's)
Self-policing (Beveiliging server)

Voor ons voorbeeld, gaan we er van uit dat we 1 enkele server hebben die dienst doet als mail server, webserver, DNS server, enzovoort ...

We willen dat:

1. Mensen van binnen ons bedrijf op Internet geraken vanaf hun desktop computer.
2. Mensen van binnen ons bedrijf onze server kunnen bereiken.
3. Mensen vanop het Internet enkel de server kunnen bereiken, en niet een andere computer van ons intern netwerk.



Het gevaar schuilt natuurlijk in de details hier, maar we hebben een beknopte weergave van waar we naartoe willen. We beginnen met het instellen van de router/firewall.

1.1 Router toegangslijsten

Elk IP pakket dat door een Internet verbinding, router en netwerk passeert, heeft een header. Hierin staat iets over zijn herkomst, bestemming en dikwijls ook doeleinde. Niet IP verkeer heeft uiteraard ook headers, maar we beperken ons hier tot IP.
Figuurlijk gesproken, kan je even in de kabel van het netwerk kruipen en de headers bekijken van de pakketten die voorbij komen gevlogen. Je zal zien dat er een grote variëteit aan pakkettypes bestaat, die allen behoren tot verschillende soorten IP transporten. Zo kan je een onderscheid maken tussen TCP, UDP en ICMP pakketten. Het aantal en de bestemming zijn afhankelijk van de toepassingen die op het netwerk draaien. Sommige onder hen zullen we direct herkennen omdat ze van bekende applicaties komen zoals Telnet of FTP. Anderen behoren dikwijls tot applicaties waar we niet aan gedacht hadden, of die we niet kennen. Voor ons doeleinde is het verschil tussen beiden wel belangrijk. We moeten immers filters gaan toepassen op de communicatie. We starten met het TCP verkeer.

1.1.1 TCP/IP (Transmission Control Protocol)

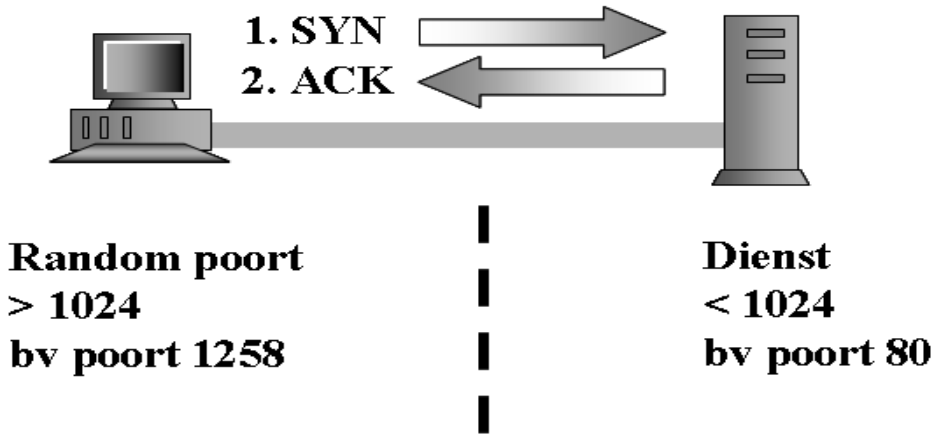
Mensen gebruiken dikwijls de term “TCP/IP” voor de hele boterham transportprotocollen (TCP, UDP, ...). Wat men eigenlijk met deze term bedoelt, is gewoon “IP” (Internet Protocol). TCP is slechts één van de mogelijke IP transport protocollen. Voor de meeste mensen doet het verschil er niet toe. Deze mensen vinden een computer wel een leuke schrijfmachine. Voor ons heeft dit echter wel belang, omdat de eigenschappen van elk transport belangrijk zijn.
TCP is een betrouwbaar, *verbindingsgericht* protocol. Dit wil zeggen dat voor toepassingen die via TCP communiceren, de data uitwisseling gebeurt alsof het loopt over een speciaal toegekende lijn. Voor de toepassing lijkt het alsof de gegevens ongeschonden en in volgorde aankomen. In werkelijkheid gaan pakketten verloren en kunnen ze in de verkeerde volgorde aankomen. De netwerk software moet er maar voor zorgen dat alles juist geschikt wordt, voor alles wordt doorgegeven aan de het programma.

TCP/IP conversaties beginnen met een uitwisseling van elementaire informatie. De inbellende computer opent een verbinding op een lokale TCP poort met een nummer hoger dan 1024.
De bestemming is dikwijls een *welbekende dienst* op een poortnummer lager dan 1024. Hier verwacht de aanvrager een warm onthaal door een netwerktoepassing van de bestemmingscomputer.

Bijvoorbeeld:

> SMTP mail conversaties worden via poort 25 afgehandeld . Wanneer ik jou computer zou contacteren om mail te krijgen, verwacht ik dat ik mij kan verbinden met poort 25 om daar met de mail daemon te spreken.
> Telnet luistert bijvoorbeeld naar poort 23.

De combinatie van de hoge poortnummer, vaste poortnummer, bron -en doeladres levert een unieke tuppel op waarmee we een welbepaalde conversatie vastleggen.

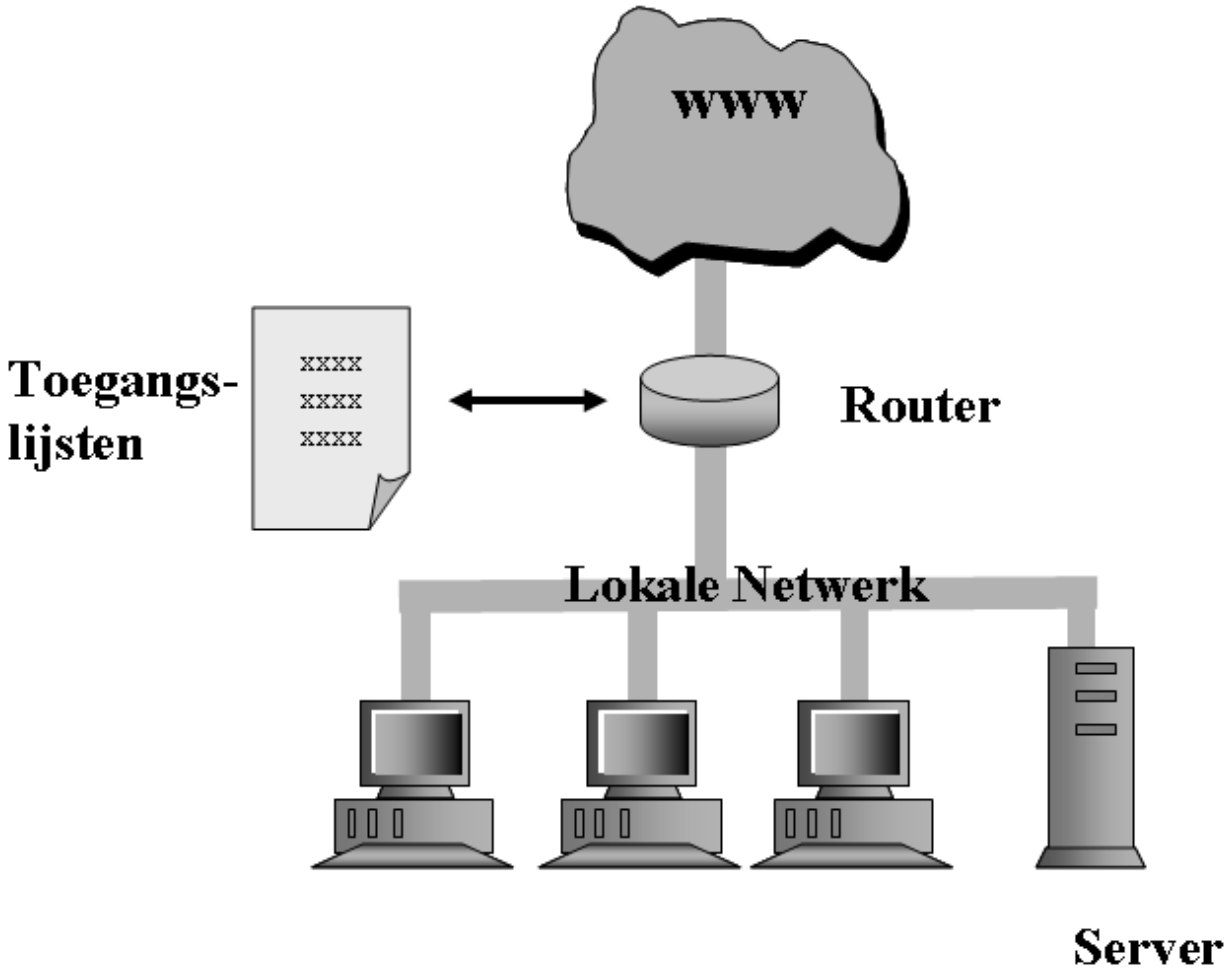


Figuur: Vereenvoudigd begin van een TCP conversatie

1. Om te beginnen communiceren zendt de beller een SYN (synchronize) pakket. Het SYN pakket bevat:
   Het doeladres, de doelpoort, het bronadres en de bronpoort
   Een volgnummer dat gebruikt wordt om bij te houden in welke volgorde datapakketten moeten behandeld worden.

1. De ontvanger antwoord op het SYN pakket met een ACK (eigenlijk een SYN-ACK) pakket, en de verbinding is tot stand gekomen. Data pakketten worden doorgegeven en bevestigd ("ACKed") totdat uiteindelijk de verbinding wordt stopgezet door één of beide partijen. Vanuit het standpunt van een router is er rijkelijk veel informatie voorzien om een TCP sessie te verzorgen.

   Stel dat ik een bepaalde dienst vanuit de buitenwereld wil verhinderen (bvb Telnet). Ik kan de router zodanig instellen dat deze alle binnenkomende pakketten met doelpoort 23 laat vallen. Een andere mogelijkheid is om selectief bepaalde SYN pakketten te laten vallen. Dit verhindert dat de initiële TCP verbinding tot stand kan komen. Ik zou ook iets minder strikt kunnen zijn en bepaald verkeer naar welbepaalde computers toch toelaten. Ik kan bijvoorbeeld voor alle computers in mijn netwerk de verbinding naar poort 23 weigeren, behalve voor één machine, die ik doelbewust wil openstellen voor het publiek.



Figuur: Router met toegangslijsten

De regels die beschrijven hoe pakketten moeten worden gefilterd, worden de r*outer toegangslijsten* genoemd of *Access Control Lists* (ACL's). Deze lijsten worden in de router geprogrammeerd door de netwerkbeheerder. Typisch worden regels in toegangslijsten één voor één aangemaakt. Daarna worden ze allemaal tegelijk toegepast op de router interface. Routers kunnen meerdere netwerkinterfaces hebben, en dus ook meerdere toegangslijsten.
Hier zou het mooi zijn dat ik een aantal regels gaf en rustig verder kon gaan. Probleem is dat de inhoud van regels en de manier van filteren verschillen bij diverse routermerken. Zo zijn er merken die enkel in één richting het verkeer kunnen filteren. Anderen kunnen beide richtingen definiëren.
Sommigen kijken naar SYN pakketten, anderen zijn eenvoudiger doordat ze bron -en doeladres bekijken. Eender wat voor soort instellingen we kunnen maken, de manier om de toegangslijst op te stellen blijft hetzelfde. We beschouwen alle soorten verkeer dat we willen toelaten tot ons netwerk.

Prot	Toegelaten	Bronp	Doelp	VAN adres	NAAR adres
!ip	IN,OUT	sourceroute	-	0.0.0.0/0	0.0.0.0/0
!any	IN	elke poort	elke poort	200.2.2.0/24	200.2.2.0/24
tcp	SYN,IN	elke poort	80 (httpd)	0.0.0.0/0	200.2.2.3/32
tcp	SYN,IN	elke poort	23 (telnet)	198.8.8.0/24	200.2.2.3/32
tcp	SYN,IN	elke poort	23 (telnet)	197.7.7.3/32	200.2.2.0/24
tcp	SYN,IN	elke poort	25 (smtp)	0.0.0.0/0	200.2.2.3/32
tcp	SYN,IN	elke poort	119 (nntp)	199.9.9.3/32	200.2.2.3/32
tcp	SYN,IN	hoge nummer	21 (ftp)	0.0.0.0/0	200.2.2.3/32
tcp	OUT	20(ftpdata)	hoge nummer	200.2.2.3/32	0.0.0.0/0
tcp	SYN,OUT	elke poort	elke poort	200.2.2.0/24	0.0.0.0/0
tcp	IN	elke poort	1025-65535	0.0.0.0/0	200.2.2.0/24

Tabel : Een verzameling toegangsregels.

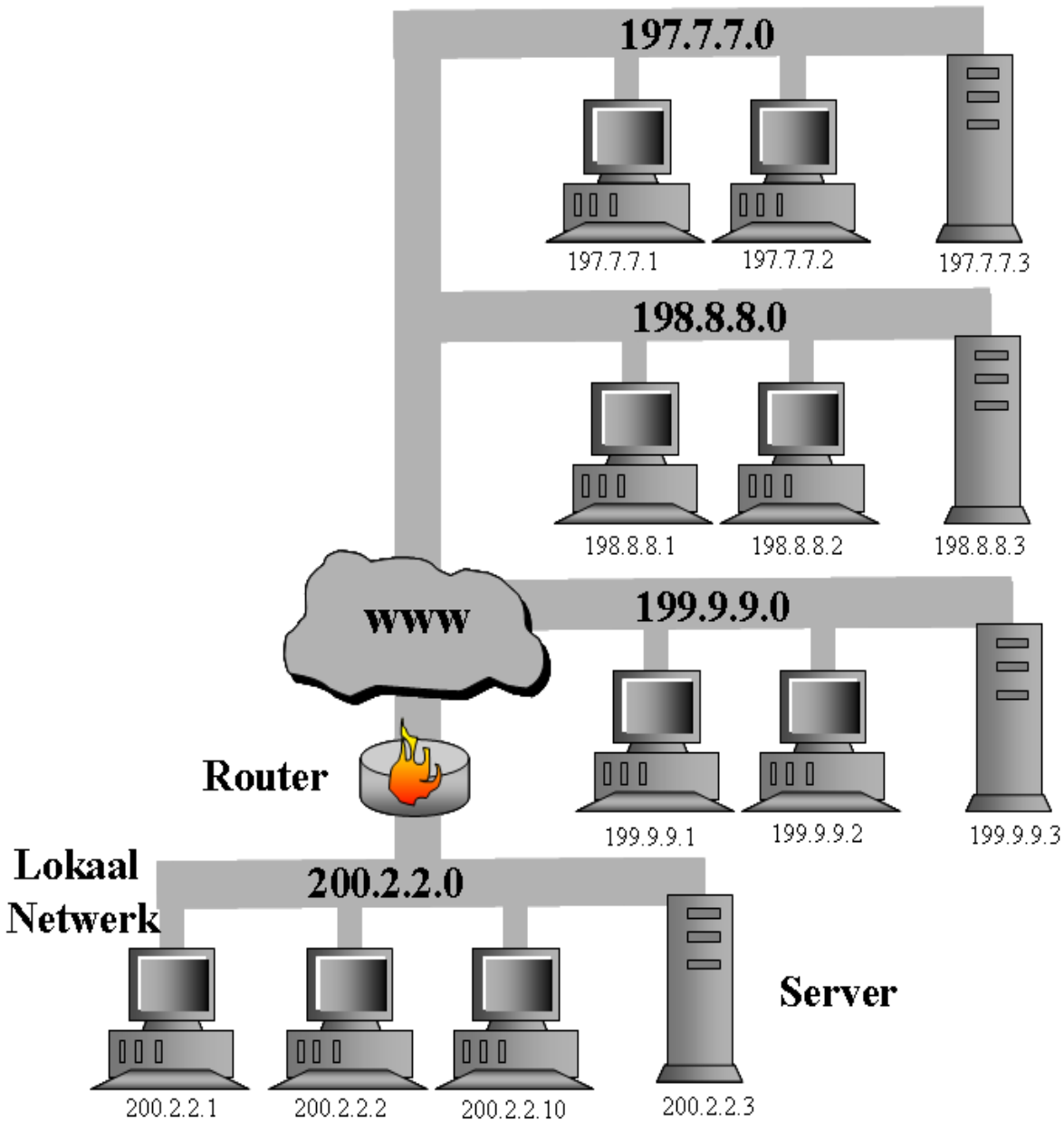
De tabel maakt gebruik van enkele voorbeeldadressen. Het is best mogelijk dat bij jou firewall een andere indeling moet gebruikt worden. De VAN en NAAR kolommen bevatten een adres met erachter het aantal bits waar moet naar gekeken worden. Zo is 200.2.2.3/32 één welbepaalde computer omdat alle 32 bits meetellen. Bij 200.2.2.0/24 betekent dit dat de 24 eerste bits meetellen (dus alle computers op een C klasse netwerk). Wanneer er enkel nullen staan (0.0.0.0/0) betekent dit dat geen enkele bit moet bekeken worden en dus alle adressen automatisch aan de voorwaarde voldoen.

!ip	IN,OUT	sourceroute	-	0.0.0.0/0	0.0.0.0/0

Aanvaardt geen (! uitroepteken is niet) IP verkeer waarbij source routing geactiveerd is. (zie hierna)

!any	IN	elke poort	elke poort	200.2.2.0/24	200.2.2.0/24

Aanvaardt geen verkeer dat van mijn netwerk komt en naar mijn netwerk moet.
Deze regel is een beetje raar. Hoe kan er nu ooit verkeer van en naar mijn netwerk voorkomen aan twee kanten van een firewall? De mogelijkheid bestaat dat iemand doelbewust valse bronadressen doorstuurt, om te doen alsof de pakketten vanuit het eigen netwerk komen. Dit heet men **IP spoofing** Hierdoor kunnen bedriegers gebruik maken van bepaalde diensten die normaal enkel door mensen van binnen het bedrijf mogen gebruikt worden. Deze bekende veiligheidslek kan je vermijden door na te gaan of je geen conversaties met jezelf aangaat.



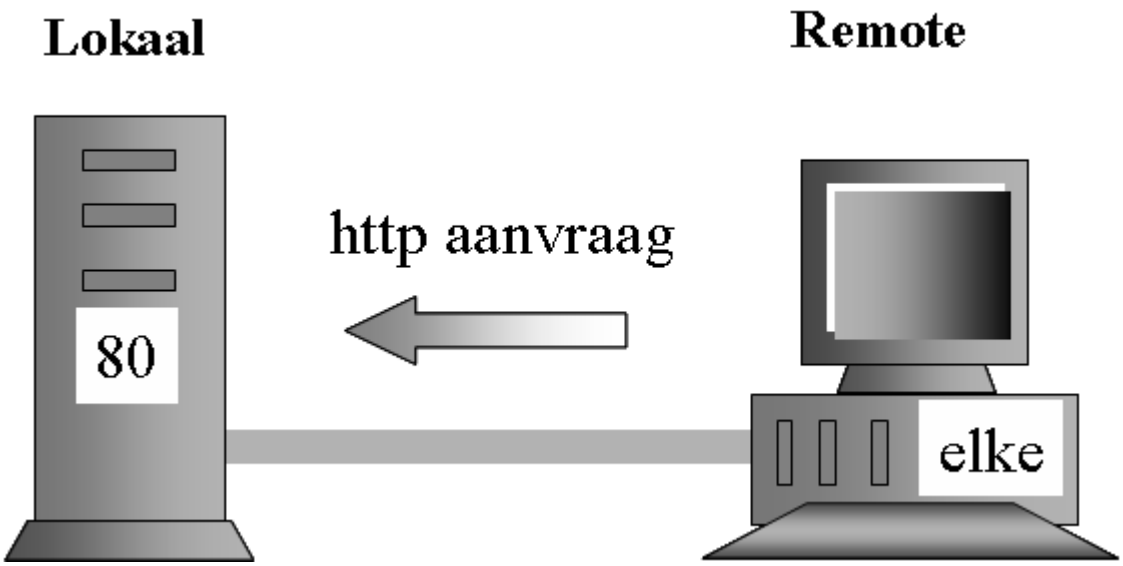
Figuur: Voorbeeldnetwerk waarop we onze toegangslijsten toepassen

REGELS VOOR TELNET EN HTTP

```
tcp   SYN,IN      elke poort  80 (httpd)  0.0.0.0/0         200.2.2.3/32
tcp   SYN,IN      elke poort  23 (telnet) 198.8.8.0/24       200.2.2.3/32
tcp   SYN,IN      elke poort  23 (telnet) 197.7.7.3/32       200.2.2.0/24
```

Deze regels laten verkeer toe vanuit het netwerk (IN) voor de TCP dienst http. Door SYN pakketten toe te laten kunnen er ook conversaties opgestart worden vanuit Internet met deze diensten. Zoals eerder vermeld, wordt deze conversatie enkel vanop 200.2.2.3 toegelaten (de /32 duidt erop dat alle bits er toe doen).

Dus deze regels zorgen er in het kort voor dat TCP verkeer toegelaten is op poort 80, van eender waar, naar één bepaalde host 200.2.2.3. Elke host van het klasse C netwerk 198.8.8.0 mag een telnet sessie beginnen met host 200.2.2.3.
De tweede regel geeft aan dat één bepaalde computer 197.7.7.3 een telnet sessie mag openen met eender welke computer op netwerk 200.2.2.0.



Figuur: externe aanvraag van een http-dienst

REGELS VOOR SMTP en NNTP

```
tcp   SYN,IN      elke poort  25  (smtp)  0.0.0.0/0         200.2.2.3/32
tcp   SYN,IN      elke poort  119 (nntp)  199.9.9.3/32      200.2.2.3/32
```

De SMTP en NNTP regels zijn analoog aan deze van telnet en http. We beperken deze diensten door een beperking op te leggen aan de computers binnen het bedrijf die er van gebruik maken.

REGELS VOOR ACTIEVE FTP

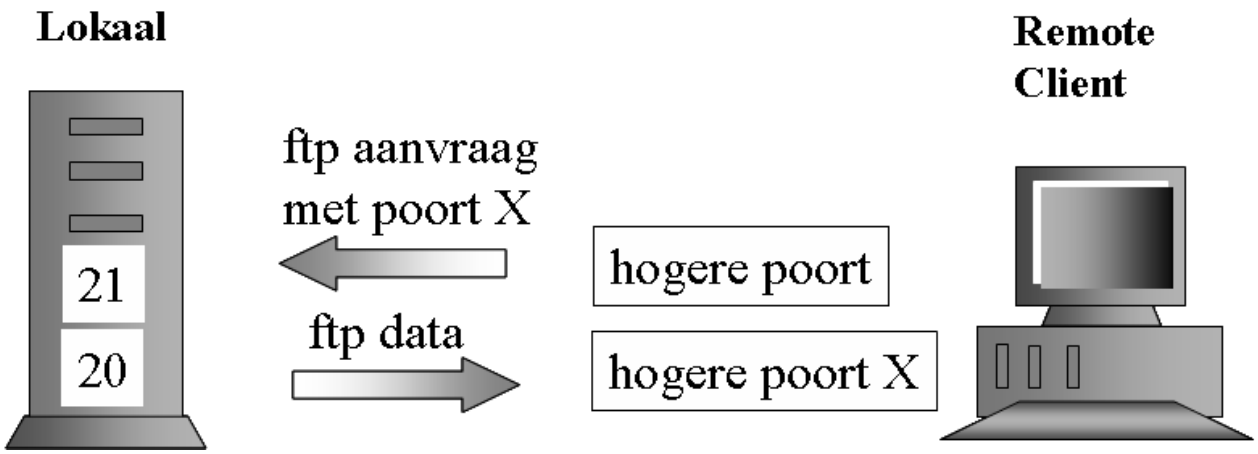
FTP heeft wat extra uitleg nodig bij de regels. FTP bestaat in werkelijkheid uit twee verschillende connecties. Een controle verbinding en een data verbinding. De FTP gebruiker zal eerst een poort met een hoge nummer (>1024) gebruiken en daarna de afgelegen host contacteren op poort 21. Wanneer er data getransfereerd wordt, zal de afgelegen computer via poort 20, de lokale computer op zijn poort met hoge nummer contacteren om de data te verzenden.
* De eerste verbinding (met poort 21) is voor de controle verbinding (inloggen, transfer opties, ...)
* De tweede (met poort 20) zal de data overzenden. Dit heet men **actieve FTP**. De client moet hierbij toelaten dat er een verbinding vanaf poort 20 vanuit de server gemaakt wordt.

FTP heeft 2 soorten regels. Regels voor binnenkomende en uitgaande FTP.

REGELS VOOR BINNENKOMENDE FTP (NAAR ONZE FTP SERVER) :

```
tcp   SYN,IN      hoge nummer 21 (ftp)    0.0.0.0/0          200.2.2.3/32
tcp   OUT         20(ftpdata) hoge nummer 200.2.2.3/32       0.0.0.0/0
```

Binnenkomende data geeft niet zoveel problemen. We stellen gewoon dat alle binnenkomende conversaties mogelijk zijn enkel op onze server op poort 21. Enkel de server kan zijn poort 20 gebruiken om data terug te sturen op een poort met hogere nummer van de remote host.

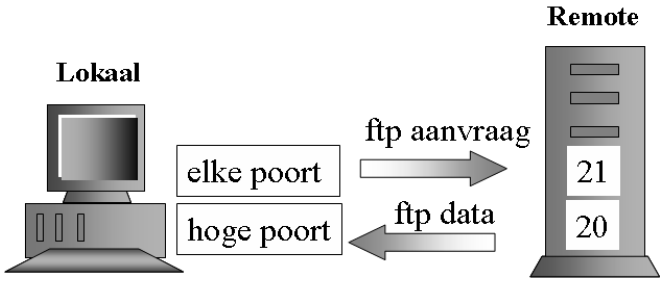


Figuur: Externe computer vraagt een FTP dienst aan

REGELS VOOR UITGAANDE FTP (VAN ONZE CLIENTS) :

```
tcp   SYN,OUT     hoge nummer  21 (ftp)   200.2.2.0/24      0.0.0.0/0
tcp   IN          20(ftpdata) hoge nummer 0.0.0.0/0         200.2.2.0/24
```

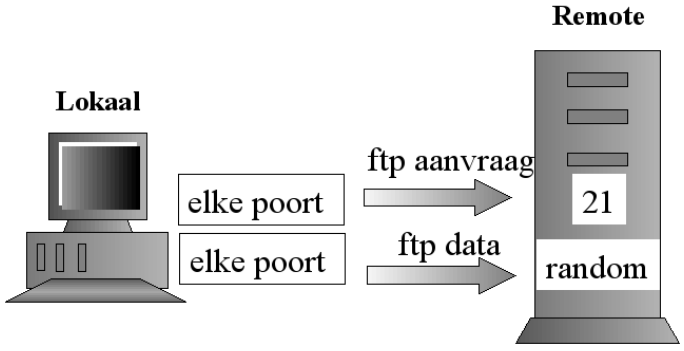
Een lastiger veiligheidsprobleem is een uitgaande FTP verbinding. Onze lokale computers toelaten om een poort met hoge nummer op een afgelegen computer te bereiken is geen probleem. De data die moet terugkeren vormt wel een probleem. We moeten een gat in de firewall voorzien, dat toelaat om een poort met hoge nummer op ons netwerk te bereiken. We moeten voorzien dat de conversatie kan worden gestart (SYN toelaten). Een verderfelijk persoon van buiten ons bedrijf kan echter wel een connectie op poort 20 openen, zonder een FTP sessie te doen. Voor de tweede (data) verbinding gebruikt de Linux firewall iptables de benaming "**RELATED**" om dit probleem op te lossen. Er moet dan altijd eerst een controleverbinding geopend zijn, alvorens er een connectie mag gemaakt worden.



Figuur: Lokale computer vraagt een FTP dienst aan

REGELS VOOR PASSIEVE FTP

Passieve FTP is veiliger. Hierbij opent de client de connectie naar poort 21 van de FTP server. De server zal na een PASV van de client zijn random poort doorsturen waarop de client moet connecteren.
Passieve FTP kan je met een client afdwingen door in de FTP client het commando (of de GUI optie) PASV in te stellen.
Er moet dus niet meer worden toegelaten dat een FTP server van buiten je netwerk een connectie naar een client opent (zoals bij Actieve FTP).



Figuur: Lokale computer vraagt een Passieve FTP dienst

ALGEMENE REGELS

tcp	SYN,OUT	elke poort	elke poort	200.2.2.0/24	0.0.0.0/0
tcp	IN	elke poort	1025-65535	0.0.0.0/0	200.2.2.0/24

De eerste regel laat toe dat eender welke computer van ons netwerk een conversatie opstart met elke afgelegen computer (als deze dat toelaat). Er zijn dus geen beperkingen op uitgaand verkeer. Wat men hier bijvoorbeeld wel kan toevoegen is een verbod op het gebruik van irc voor heel het bedrijf, of beperkingen op het gebruik van websites van de concurrentie of met een hoog xxx gehalte.
De tweede regel hebben we nodig opdat gebruikers een conversatie in stand kunnen houden met computers op het Internet. Even bij wijze van herhaling: Om een TCP verbinding te maken neemt de computer een hoge poortnummer (1025-65535) en treedt over Internet in verbinding met een bekende lage poortnummer van een afgelegen computer. We moeten dus voorzien dat over diezelfde hoge-nummer-poort data kan terugkomen. Dit wel met de beperking dat dit niet in het begin van een conversatie (met SYN) gebeurt, want dat zou betekenen dat iemand zonder meer een verbinding kan maken met elke poort met hoge nummer.

1.1.1 UDP/IP

UDP/IP (User Datagram Protocol) conversaties verschillen op meerdere manieren van TCP/IP. Vanuit het standpunt van de toepassing is UDP onbetrouwbaar. De netwerk software kijkt niet na of er pakketten verloren gaan of in de verkeerde volgorde aankomen. De problemen met volgorde en verlies van UDP gegevens moeten dus door de toepassing zelf worden opgelost . Verder zijn het opstarten van een verbinding, het verzenden en bevestigen (typische eigenschappen van TCP) niet aanwezig. De toepassing moet dus zelf de communicatie opzetten en in stand houden. Dit lijkt allemaal angstaanjagend moeilijk, maar eigenlijk valt dit goed mee. De netwerkverbindingen zijn lichter. Er is geen synchronisatie nodig en geen bevestiging van de pakketten (ACK). Voor verscheidene informatie-uitwisselingen kan UDP het verkeer ontlasten en fileproblemen vermijden.

Jammer genoeg maken de eigenschappen die UDP zo efficiënt maken, het ons onnoemelijk moeilijk om de communicatie te beveiligen. Hoe kunnen we bijvoorbeeld weten of een UDP pakket een antwoord is op een aanvraag, de start van een conversatie of ongewenste data? Conversaties hebben geen vastgelegd begin of einde, dus is het nogal moeilijk om te zeggen “wie wat van wie” nodig had. Verder gebeuren UDP aanvragen en antwoorden dikwijls op dezelfde poorten, of op eender welke poort, wat onze opdeling bij TCP van lage en hoge poorten hier betekenisloos maakt. UDP uitwisselingen zijn relatief eenvoudig te kraken door kwaadaardige en criminele krankzinnigen. Er is geen DATA-ACK cyclus en de netwerk software houdt helemaal niet bij in welke volgorde er dingen gebeuren. Valse pakketten kunnen veel eenvoudiger ingespoten worden in de conversatiestroom.

UDP filteren is niet volledig onmogelijk. De uitdaging is dat je niet enkel een fractie van het verkeer moet volgen, maar daadwerkelijk volledige conversaties moet volgen.
Ik kan bijvoorbeeld een aanvraag zien vanuit mijn netwerk naar een remote host op poort 53 (DNS). Logischerwijze kan ik dan wel iets later een antwoord verwachten van deze remote host op poort 53. Een pakket dat zomaar uit het niets verrijst kunnen we wel verwerpen. Dit probleem hebben we vooral bij het gebruik van toegangslijsten. Routers met toegangslijsten zijn eigenlijk vrij dom van nature uit. Ze onthouden niet het verkeer dat ze ooit zijn tegengekomen. Een gevolg

hiervan is dat we onmogelijk een router kunnen wijsmaken dat ze UDP verkeer dat we zelf hebben aangevraagd, moeten doorlaten. We beslissen hier dus om het meeste UDP verkeer te weigeren, en enkel poort 53 voor DNS aanvragen bij onze nameserver openhouden.
Sommige firewalls kunnen wel de conversatie bijhouden en dus beter UDP filteren. Zie verder.

!ip	IN,OUT	sourceroute -	0.0.0.0/0	0.0.0.0/0

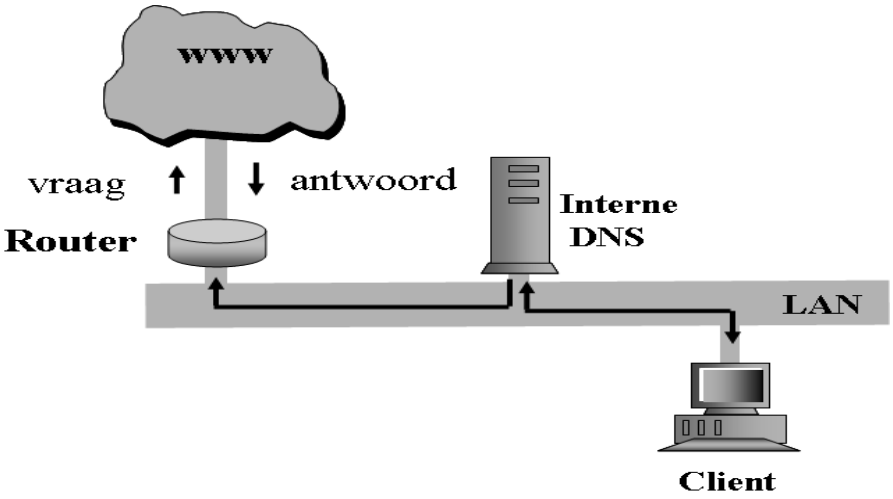
Prot	**Toegelaten**	**Bronp**	**Doelp**	**Van adres**	**Naar adres**
udp	IN	53 (domein)	53 (domein)	0.0.0.0/0	200.2.2.3/32
udp	OUT	53 (domein)	53 (domein)	200.2.2.0/24	0.0.0.0/0
tcp	SYN,IN	elke poort	53 (domein)	199.100.105.2/32	200.2.2.3/32

In bovenstaande regels worden DNS aanvragen naar de nameserver 200.2.2.3 toegelaten via UDP en de rest wordt geweigerd.

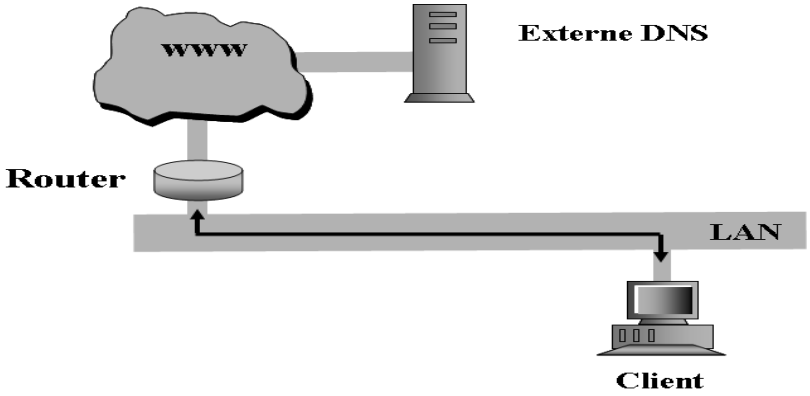
Opmerking: Je ziet dat ik een kleine opening in de firewall gelaten heb op poort 53 met TCP. Er is normaal absoluut geen verband tussen de poortnummers onder TCP en UDP. In dit geval zal de nameserver echter luisteren naar zowel TCP als UDP aanvragen op poort 53. UDP op poort 53 dient voor DNS opzoekingen. TCP op poort 53 dient voor zone-transfers, update informatie die een secundaire nameserver haalt van een primaire nameserver (hier 199.100.105.2).

In verband met DNS opzoekingen is er nog een tweede mogelijkheid. Wanneer uw LAN machines een query moeten doen, moeten zij een externe nameserver op het Internet raadplegen (zoals bij een provider). UDP conversaties hiervoor gebeuren op nummers groter dan 1023. Wanneer we al deze poorten open moeten laten voor onze UDP filter, laten we een gat open in de firewall waar je ongemerkt met een airbus kan binnenvliegen. Hier kan een firewall die verbindingen onthoudt ook weer een oplossing bieden.

Een derde (uitermate veilige) mogelijkheid is dat je een minimale interne nameserver hebt die alle aanvragen doorstuurt naar een externe nameserver.



Figuur: Client zendt queries naar interne nameserver

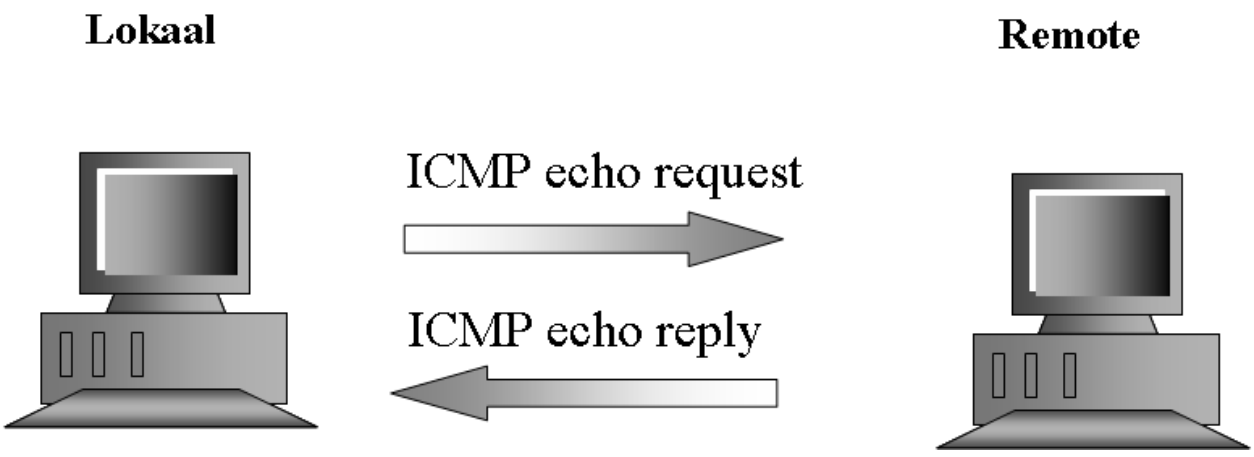


Figuur: Client zendt queries naar externe nameserver (gevaarlijker)

1.1.2 ICMP

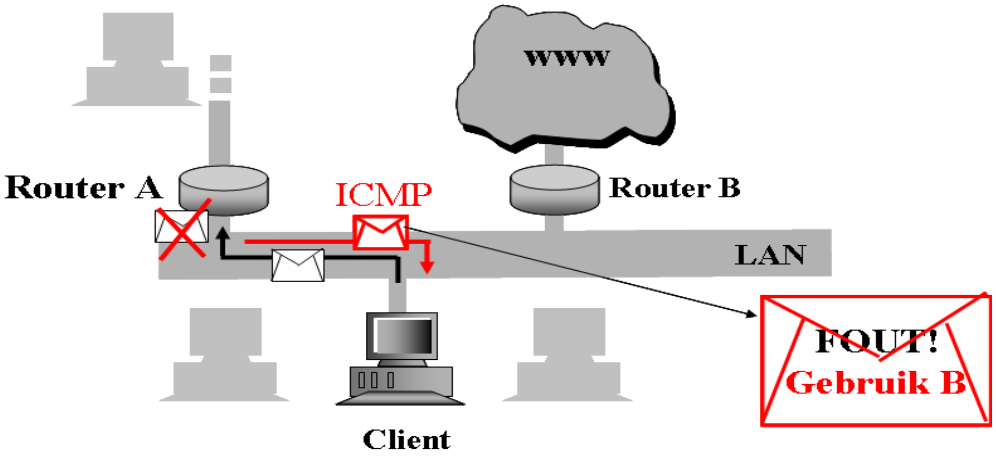
Een derde IP protocol waar we rekening mee dienen te houden is ICMP (Internet Control Message Protocol) . Zoals UDP en TCP draait ICMP ook op het hoogste niveau van IP. ICMP is, in tegenstelling tot de andere protocols van de familie, een noodzakelijk deel van IP. IP werkt niet zonder ICMP. Het doel van ICMP is het uitwisselen van statusberichten tussen gateways en hosts. Een ICMP bericht kan je iets meer vertellen over de bereikbaarheid van een host, de verkeerstoestand of de voorkeurroute tussen twee punten A en B. Omdat ICMP berichten het verkeer vlotter laten verlopen in een netwerk, staan we ICMP toe om tot bij het lokale netwerk of tot aan de firewall te geraken.

Er bestaan een handvol ICMP pakketten, elk verantwoordelijk voor een bepaald soort berichten. Je kent waarschijnlijk wel het commando ping. Ping is gebaseerd op 2 ICMP berichten, namelijk *echo (request)* en *echo reply*. De meeste van deze berichten zijn tamelijk onschadelijk op een netwerk. Het enige potentieel gevaar van ping is dat iemand je netwerk probeert af te scannen (probe) op zoek naar hosts. Er is één ICMP bericht dat een groter veiligheidsprobleem stelt, en dat we bij voorkeur uitfilteren, namelijk een **redirect**.



Figuur: Sturen van een ping: ICMP echo request en ICMP echo reply

Redirect stelt betere routes voor aan een host. Stel dat er 2 routers zijn die jou netwerk met andere netwerken verbinden. Router A leidt bijvoorbeeld naar een andere afdeling binnen je bedrijf, router B leidt naar Internet. Hosts hebben er geen benul van welke router ze nu net dienen te gebruiken om op Internet te geraken. Het zou zelfs kunnen dat pakketten ook via het andere netwerk van je bedrijf tot op internet geraken (dus via router A). Moest je bij router A gaan aankloppen om even op Internet te gaan dan zal router A met een *redirect* bericht vertellen aan je computer dat je **beter Router B** gebruikt.



Figuur: ICMP redirect

Redirect kan ook misbruikt worden in een LAN. Ik kan bijvoorbeeld een vals ICMP bericht verspreiden dat alle verkeer naar mijn computer leidt. Wanneer ik dan ook nog eens doe alsof ik de computer of het netwerk was, waarmee je in contact wou treden (met masquerading), dan kan ik al jou data gaan misbruiken. Omwille van dit potentiële misbruikgevaar zal een redirect bericht door de meeste routers geweigerd worden.

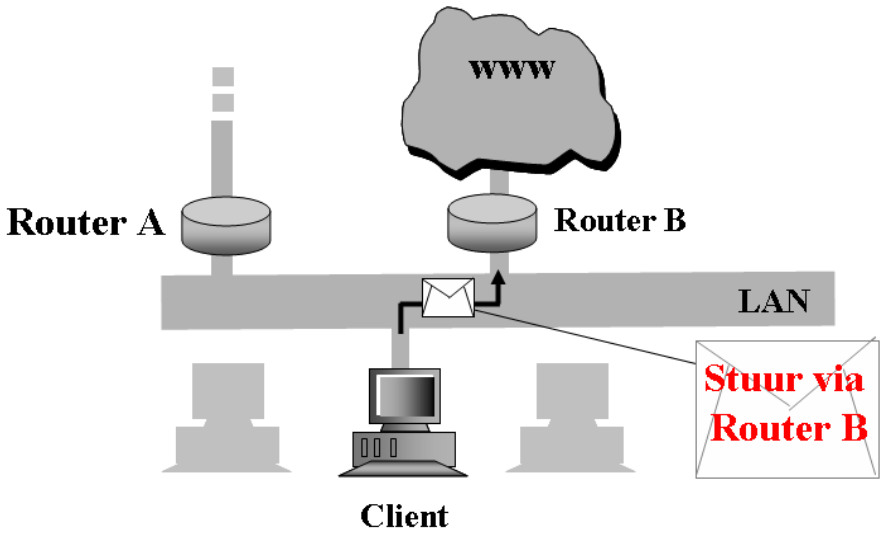
Prot Toegelaten	type	Van adres	Naar adres
!icmp IN, OUT	redirect	0.0.0.0/0	0.0.0.0/0
icmp IN, OUT	alles	0.0.0.0/0	0.0.0.0/0

Type	Naam	Referentie
0	Echo Reply	[RFC792]
3	Destination Unreachable	[RFC792]
4	Source Quench	[RFC792]
5	Redirect	[RFC792]
8	Echo	[RFC792]
9	Router Advertisement	[RFC1256]
10	Router Selection	[RFC1256]
11	Time Exceeded	[RFC792]
12	Parameter Problem	[RFC792]
13	Timestamp	[RFC792]
14	Timestamp Reply	[RFC792]
15	Information Request	[RFC792]
16	Information Reply	[RFC792]
17	Address Mask Request	[RFC950]
18	Address Mask Reply	[RFC950]
30	Traceroute	[RFC1393]
31	Datagram Conversion Error	[RFC1475]

Tabel: Enkele ICMP types

1.2 Source routing

Er is ook een andere voorziening bij alle soorten IP verkeer (TCP, UDP, ICMP enz...) dat een gevaar voor de veiligheid kan opleveren. Zoals bij *ICMP redirect* voorziet **source routing** een mechanisme om het verkeer bij te sturen. Het werkt als volgt: In een IP pakket header kan men een speciale instructie steken om het verkeer van plaats A naar plaats B te laten lopen. Je kan de routers onderweg zeggen dat ze alles vergeten wat ze 'denken' te weten over de optimale weg voor een pakket. In plaats daarvan wordt het pakket gestuurd volgens de **route** die aangegeven is in de **header**. Het antwoord op een aanvraag via source routing gebeurt ook via dezelfde vastgelegde weg.



Figuur: Source routing

In een betrouwbaar netwerk is source routing een handig tool om je pakketten volgens een bepaalde baan te doen verlopen Dit kan bijvoorbeeld om te debuggen, om bottle-necks te vermijden of om 'geheim' verkeer niet langs alle afdelingen te laten lopen.
Gevaar van source routing is dat iemand het kan misbruiken om één van jou computers na te bootsen. Stel dat jou router toegangslijst enkel een Telnet sessie toelaat met één bepaalde computer ergens aan de andere kant van de wereldbol. Wanneer een onguur individu het adres van die computer kent, kunnen er problemen ontstaan. Met source routing is er een potentiële kans dat iemand hetzelfde netwerkadres als die computer gebruikt en door je router geraakt. Source routing kan gecontroleerd worden door in je router aan te geven dat hij alle pakketten met de source routing vlag enabled weigert.

1.1 Besluit minimale firewall

Wanneer we alles even samen beschouwen hebben we de volgende zaken ingesteld in onze router:

TCP verkeer: Toegelaten in enkele voorzichtig aangebrachte gaten.
UDP verkeer: Volledig of bijna volledig verboden.
ICMP verkeer: Redirect uitschakelen
Source routing verbieden

Ander verkeer waar je eventueel op moet letten:

X11 verkeer (UNIX):	Poorten 6000/TCP tot en met 6100/TCP
Openlook verkeer (Sun):	Vanaf poort 2000/TCP
Citrix ICA client:	Poort 1494/TCP
Terminal Server (RDP):	Poort 3389/TCP
OWA (webmail):	Poort 80/TCP
OWA (SSL):	Poort 443/TCP

Zoals eerder vermeld, kunnen de mogelijke instellingen verschillen van router tot router. Het is bijvoorbeeld mogelijk dat je router geen onderscheid kan maken tussen SYN pakketten en andere pakketten. Dan zal je dus het verkeer moeten filteren enkel met bron -en doeladres en poortnummer.

Nadelen aan deze minimale firewall:

De netwerk topologie is zichtbaar voor de hele wereld. Omdat er directe uitwisselingen zijn toegelaten tussen interne hosts en hosts op het Internet zullen de interne computers gedeeltelijk zichtbaar blijven voor de buitenwereld.
Wanneer er medeplichtigheid tussen interne en externe personen op het Internet, dan is het mogelijk om tunneling toe te passen. Een vorm van tunneling is wanneer je netwerkverkeer inpakt in een andere stroom netwerkverkeer. Het ingepakte verkeer wordt toegelaten door de firewall en aan de andere kant terug uitgepakt.
Eén keer dat de server is ingenomen, kan de kraker op heel je netwerk. De moeilijkheid stijgt naarmate je meer diensten zoals WWW, FTP, SSH, ... aanbiedt. Deze diensten worden wel eens meer gekraakt.

1 Bescherming van de server

If God is dead, who will save the Queen?

Een draaispil van het hele veiligheidsgebeuren is de server. Wanneer iemand er in slaagt om deze over te nemen, heeft men toegang tot het hele netwerk. Om te beginnen moet je het aantal administrator accounts beperken tot een minimum. Verder moeten we er voor zorgen dat we op de hoogte blijven van alle mogelijke bugs en gaten die er in de bestaande programma's of besturingssystemen zijn.

1.1 Application Wrappers

Toepassingen op de server kunnen ook met een soort toegangslijsten werken. Net zoals bij routers kan je op de server het gebruik van toepassingen beperken afhankelijk van de herkomst van de aanvraag. *Wrappers* komen in de plaats van de standaard daemons. Wrappers zijn kleine programma's die nakijken of iemand een bepaald programma mag gebruiken en daarna het echte programma opstarten.
Een wrapper voor de telnet daemon zal opgestart worden door inetd wanneer een aanvraag binnenkomt op poort 23/TCP.

Een alternatieve methode is dat we inetd en de wrapper vervangen door één programma dat alles uitvoert (zowel controle als het eigenlijke programma). Zulke programma's bestaan reeds. Een voorbeeld van TCP Wrappers vind je op ftp://cert.org/pub/tools/tcp_wrapper.
Je zou je kunnen afvragen waarom we wrappers nodig hebben. Is onze minimale firewall dan niet voldoende? Het antwoord hierop is dat een firewall nooit voldoende kan zijn. Wanneer men erin zou slagen om door de firewall te geraken, dan worden de toepassingen extra beschermd en gecontroleerd door de wrappers. Het is alsof men een kasteel bestormt, na 3 dagen door de muur geraakt, en dan merkt dat de volgende muur klaarstaat.

Wrappers zijn als een UV poeder dat je uitstrooit voor je huis en waaraan je kan zien wie en waar elke indringer gelopen heeft. Je kan zien waarvan de bezoekers kwamen, wanneer ze dit deden en welke dienst ze gebruikt hebben. Wrappers zijn ook gespecialiseerd in het herkennen van patronen zoals wederkerende verbindingspogingen. Wrappers kunnen ook als trigger gebruikt worden. Je kan bijvoorbeeld een mailtje laten sturen naar jou, wanneer iemand probeert om het systeem plat te leggen. (Deze mail mogelijkheid zorgt er wel voor dat de aanvaller vanop afstand je machine kan platleggen).

1.2 Systeemcontrole

Er bestaan verschillende audit tools of controleprogramma's om na te kijken of je systeem wel veilig is ingesteld.
Wanneer we een UNIX systeem nakijken op veiligheid, dienen we te letten op volgende zaken:

setuid beperken tot een kleine afgelijnde verzameling van toepassingen (setuid geeft personen de toestemming om tijdelijk even een programma van een andere persoon uit te voeren).
Configuratiebestanden van het systeem zijn enkel schrijfbaar door de root, en soms niet leesbaar voor anderen.
Scripts en home directories zijn enkel writable door hun eigenaars.
Paswoorden zijn veilig gekozen, best werken ze met een schaduw bestand.
Anonieme FTP is veilig ingesteld.
Er is niet geknoeid met systeem executables.
Systeem directories worden beheerd door een geschikte administrator (meestal root).

Deze en nog meer taken worden door audit tools uitgevoerd. Meest bekende audit tools zijn de TAMU Tiger Scripts, COPS, Tripwire en SATAN. Een korte uiteenzetting van hun functies:

TAMU

De Texas A&M University Tiger Scripts zijn te vinden op ftp://net.tamu.edu/ en bekijken de meeste van bovenstaande controles.
Extra functies:

paswoorden van gebruikers nakijken (probeert deze samen met crack te kraken)
digitale handtekening op de systeem executables

COPS

Het Computer Oracle Password and Security systeem kan je downloaden in de directory ftp://cert.org/tools/ en heeft ongeveer dezelfde eigenschappen als TAMU.
Extra functies:

Bevat het Kuang expert systeem, dat kan uitzoeken of gebruikers direct of indirect root toegang kunnen krijgen.

Tripwire

Tripwire is gespecialiseerd in digitale handtekeningen van systeem bestanden. Digitale handtekeningen zijn kunnen CRC controlesommen zijn (de eenvoudigste vorm) of afgeleid met een hash functie (zoals MD5). Het gevaar bestaat namelijk dat het bestand met digitale handtekeningen wordt gekraakt en dat men een nieuw (vals) bestand in de plaats zet. Hierdoor zijn de veranderingen aan bestanden niet meer op te sporen.
Tripwire vind je op ftp:// ftp.cs.purdue.edu/pub/spaf/COAST/Tripwire.

SATAN

Satan is één van de bekendste en meest verregaande groep van tools. Satan kan van op afstand netwerken afscannen op gaten in de veiligheid. Satan kijkt naar gaten bij sendmail, FTP, NFS, X servers en remote shells.
Satan kan je downloaden op ftp://ftp.win.tue.nl/pub/security/.

1.1 CERT

Een maand na het uitbreken van ‘ de Worm’ bleek dat er een instantie moest ontstaan die waakte over de veiligheid en programmafouten op het Internet. DARPA richtte het Computer Emergency Response Team op (CERT) en gaf deze een waakhond functie.
Het probleem bij Internet is dat enorm veel programma’s en instellingen over het gehele Internet identiek zijn. Een fout ontdekt op één bepaalde plaats is onmiddellijk een bedreiging voor het hele netwerk. CERT probeert fouten te ontdekken, de internet gemeenschap te waarschuwen en eventueel oplossingen voorstellen. Je kan je abonneren op de veiligheidsadviezen van CERT door een *subscribe* te zenden naar cert@cert.org.
Uiteraard hebben deze adviezen ook een schaduwkant. Eens dat je de Internet gemeenschap waarschuwt voor een gat in de veiligheid, zien krakers ook meteen waar ze hun aandacht op kunnen richten. In juli 1999 bleek dat CERT achterstond in verband met informatiedoorspeling.
Neem een kijkje op: http://cert.belnet.be. De Belgische CERT heet(te?) PISA (Providing Information about Internet Security Aspects)

Op 2 november 1988 schreef Robert Morris, Jr., een student in Computer Wetenschappen van Cornell, een experimenteel, zelf voortplantend, zelf verspreidend programma dat men een ‘worm’ noemt en zette dit op het Internet. Morris ontdekte snel dat het programma zich veel sneller verspreidde en veel sneller andere computers infecteerde dan verwacht. Er zat een fout in zijn programma. Wanneer Morris realiseerde wat er aan het gebeuren was, contacteerde hij een vriend op Harvard om een oplossing hiervoor te vinden. Ze stuurden vanuit Harvard een anoniem bericht over het internet, waarin stond hoe programmeurs de worm konden vernietigen en hoe ze zichzelf konden beschermen tegen reïnfectie door de worm.
Spijtig genoeg waren er op het netwerk zelf al problemen en geraakte dit bericht niet ter plaatse. Pas nadien, toen alles opklaarde, kwam het bericht overal aan. Computers werden besmet op vele plaatsen, onder andere universiteiten, militaire netwerken en medische research bedrijven. De gemiddelde kost en het verlies dat bedrijven door de worm leden varieerde van 200 dollar tot meer dan 53000 dollar.
Het programma maakte gebruik van een mankement in de DEBUG instructie van het Unix programma sendmail (Dit programma draait op een systeem en wacht op andere systemen om e-mail door te geven) en een mankement in de finger daemon fingerd, die aanvragen van het programma finger moet behandelen. Mensen van de Universiteit van Californië in Berkeley en MIT hadden kopies van het worm-programma en waren dit aan het disassemblen (zoeken hoe de source code van een programma eruit ziet)

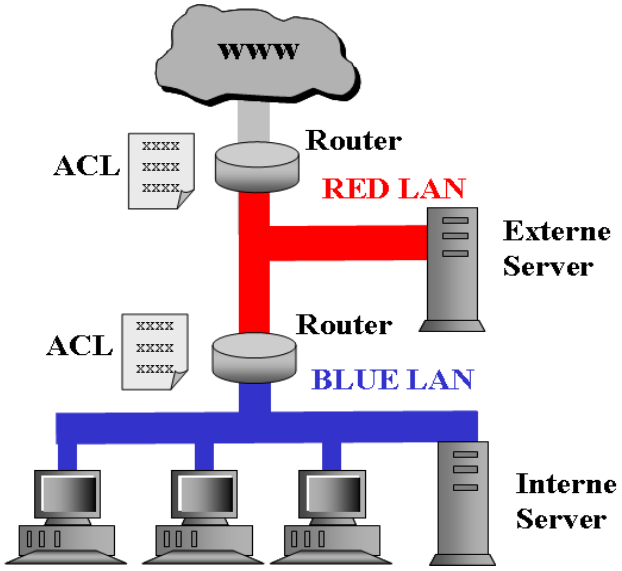
Teams van programmeurs werkten non-stop om ten minste een tijdelijke oplossing te vinden tegen de verspreiding van de worm. Na ongeveer 12 uur vond het team van Berkeley maatregelen die de verspreiding van het virus vertraagden.
Een andere methode werd in Purdue gevonden en overal gepubliceerd. De informatie geraakte desondanks niet zo snel als men wilde ter bestemming, omdat de meeste sites zich volledig van het netwerk hadden afgesloten (uit angst om de worm te krijgen of om hem te verspreiden).
Na enkele dagen keerde het computerleven naar zijn normale gangetje en iedereen wilde wel weten wie dit veroorzaakt had.
In ‘The New York Times’ werd Morris bestempeld als de auteur (hoewel dit nog niet officieel bewezen was, wezen alle bewijzen in de richting van Morris).
Robert T. Morris werd veroordeeld voor het schenden van de ‘computer Fraud and Abuse Act (Title 18)’, en moest drie jaar in hechtenis gaan, 400 uren gemeenschapsdienst doen, een boete van 10050 dollar en alle extra kosten betalen. Zijn aanklacht van december 1990, werd 3 maanden later ingetrokken

2 Verbetering van de firewall

Paradise is exactly like where you are right now ...
only much, much better.
— Laurie Anderson

2.1 Red LAN en Blue LAN

Het gevaar van het vorige type firewall is dat als iemand er in slaagt om toegang te verkrijgen tot de server, hij het hele netwerk kan bereiken.
Een verbetering op de eenvoudige firewall is het gebruik van een tweede router die leidt naar je interne netwerk.
We werken met twee LAN's. Traditioneel wordt de eerste LAN de Red LAN en de tweede LANde Blue LAN genoemd.



Figuur: Red LAN Blue LAN

Voordelen:	Dubbele bescherming
	Snelheidsverdeling: Red LAN: traag (WAN)
	Blue LAN: snel
Nadelen:	Aankoop van een extra router

Er bestaan ook three-interface firewall routers zoals de Livingston IRX Firewall Router en de Morning Star Firewall Router. Deze bevatten eigenlijk intern 2 routers en voorzien dus het gebruik van een Red en Blue LAN.

2.2 Stateful filtering

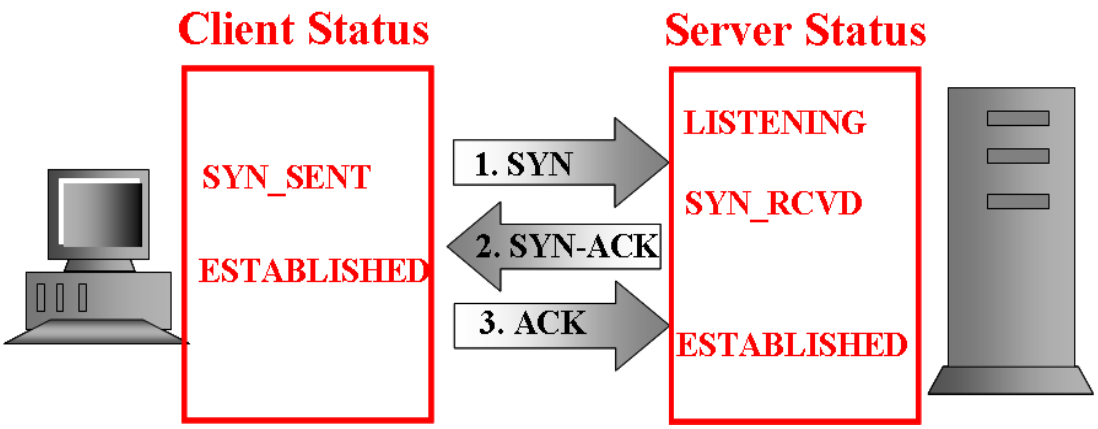
Stateful filtering is het filteren van verkeer op basis van informatie van een bepaalde verbinding. De informatie is protocolinformatie van OSI laag één tot vier. Het filteren gebeurt dus niet op basis van protocols in de applicatielaag.
Elementen die gebruikt worden voor het filteren van verkeer zijn IP adressen en poortnummers, en voor TCP ook volgordenummers(sequence), bevestigingsnummers (ACK) en vlaggen (flags).

Stateful inspection gebruikt voor het filteren alle informatie van stateful filtering én de informatie van de commando's die gebruikt worden op applicatieniveau. Al deze informatie samen zorgt ervoor dat één bepaalde verbinding van één bepaalde gebruiker redelijk zeker kan worden vastgelegd. De extra applicatielaag informatie zorgt ervoor dat lastige protocollen zoals FTP (bestandsoverdracht) en H.323 (videoconferencing, Voice over IP) om veilig en zonder problemen door de firewall te geraken.

Beide filter methodes gebruiken een *statustabel* om alle informatie van één bepaalde verbinding vast te leggen. Deze informatie wordt bijgehouden totdat een verbinding stopt (TCP FIN of TCP RST) of na een bepaalde time-out (gebruikt bij TCP, UDP en ICMP).

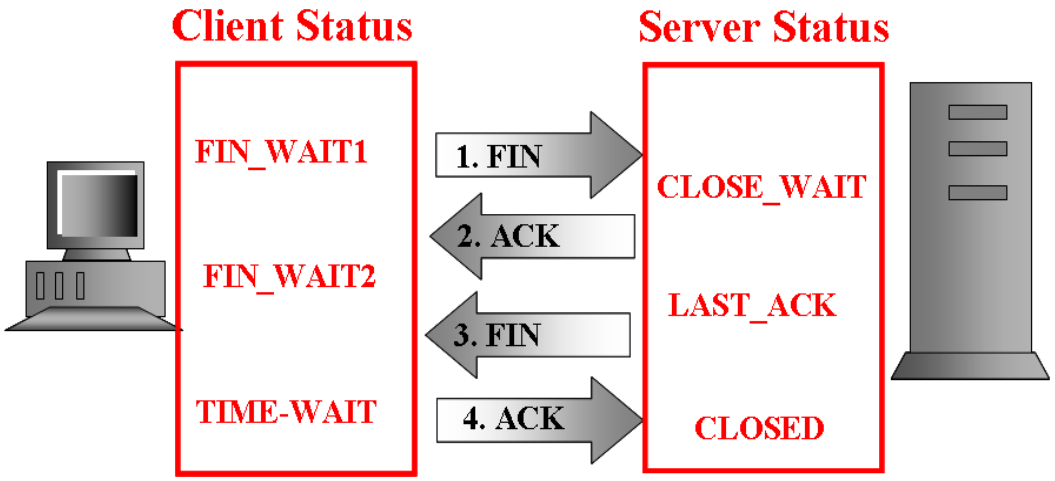
2.2.1 TCP status opstarten van een verbinding

Bij het opstarten van een verbinding stuurt de client een SYN pakket (zijn status SYN_SENT houdt bij dat hij dit verstuurd heeft). De LISTENING server antwoordt met een bevestiging van de SYN, zijn status wordt SYN_RCVD. De client stuurt een ACK pakket (en blijft met ACK flag sturen) zolang de verbinding duurt.

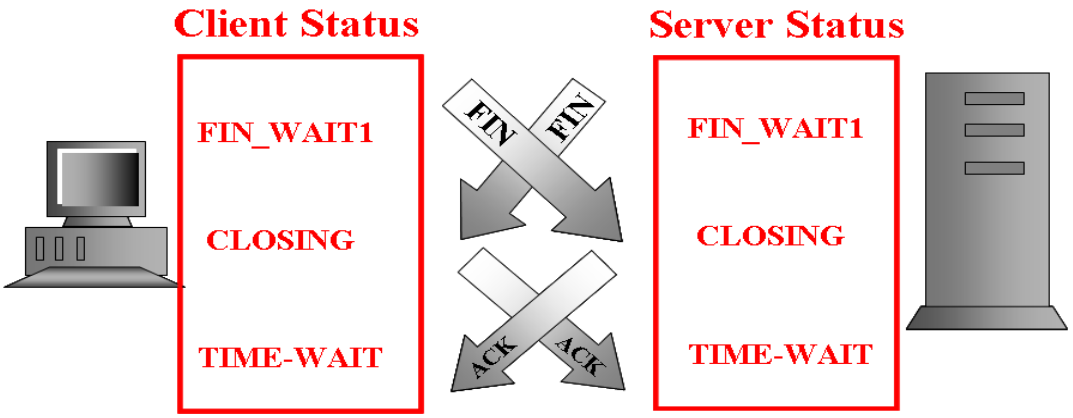


2.2.2 TCP status afsluiten van een verbinding

Bij het 'graceful' afsluiten van een verbinding stuurt de client een FIN pakket. De server bevestigd het FIN pakket met een ACK. Tenslotte stuurt de server een FIN bericht om zelf de verbinding te stoppen. Het FIN bericht van de server wordt bevestigd door de client met een ACK. Aangezien er geen bevestiging meer is van de ACK, wacht de client nog een bepaalde tijd alvorens af te sluiten.



Alternatieve en minder standaard manier is een gelijktijdige close:



3 Heavy duty firewalls

Q: Why do ducks have flat feet?
A: To stamp out forest fires.
Q: Why do elephants have flat feet?
A: To stamp out flaming ducks.

3.1 Proxies

Proxy betekent eigenlijk: een handeling uitvoeren voor iemand. Even een vergelijking.
Stel dat jij de manager bent van U2. Overal waar de groep komt, wordt deze bejaagd door fans en hopen persfotografen.
U2 heeft bij jou thuis een geheime studio waar ze hun muziek schrijven en opnemen. Om hun werk goed te doen moeten ze liefst zo ver mogelijk van alle fans en pers kunnen wegblijven. Als manager moet je er voor zorgen dat niemand iets in de weg kan leggen van de groep.
Het probleem: De groep heeft constant dingen nodig zoals gitaarsnaren, eten en cassettes. Omdat de groep niet even in de stad kan gaan wandelen, moet jij voor hen gaan inkopen. Vanuit het standpunt van de bewoners van de stad ben jij een zeer verdacht persoontje. Wat kan één persoon nu doen met 15 E snaren en 36 paar drumstokjes?
Op dat moment ben jij aan het handelen zoals een proxy voor U2. Je noteert hun bestellingen en voert deze uit. Doordat jij de boodschappen doet zal niemand in de stad ooit te weten komen dat jij voor U2 een proxy bent. Jij bent dus de firewall.
In overeenstemming met vorige situatie heeft een proxy firewall volgende kwaliteiten:
1. Alle transacties lijken vanaf één host te komen.
2. De details over het interne netwerk zijn niet zichtbaar.
3. Er is geen andere manier om naar de interne computers te geraken.

3.1.1 Proxy servers op netwerkniveau

Een firewall op netwerkniveau is meestal een afgeschermde router of een speciale computer die de adressen van de pakketten bekijkt om te bepalen of het pakket door moet worden gestuurd of geweigerd. De firewall bekijkt de header van de IP pakketten, leest het afzender -en ontvangstadres en bepaalt hiermee of het pakket mag doorgaan.
Een firewall kan zo bijvoorbeeld alle pakketten van en naar een concurrent of van en naar xxxrated adressen weigeren. Deze techniek heet men blacklisting, omdat er een soort zwarte lijst wordt opgemaakt van de te weigeren adressen. De meeste routersoftware voorziet dat je een volledige site of netwerk kan afschermen en bijvoorbeeld niet één bepaalde computer op dat netwerk. De pakketten zelf kunnen verschillende soorten informatie bevatten zoals e-mail, Telnet, FTP. Je kan deze diensten ook selecteren voor een bepaald netwerk. Zo kan je bijvoorbeeld wel het raadplegen van het Web toelaten voor een bepaald netwerk, maar het gebruik van FTP verbieden.
Of je kan internet-gebruikers wel toestaan om gegevens te downloaden, maar uploaden verbieden.

Gegevens die gebruikt kunnen worden bij firewalls op netwerkniveau:
- Bron- en doeladres van een pakket
- Sessieprotocol (bvb. TCP, UDP of ICMP)
- Bron -en doelpoorten voor een gewenste dienst

Deze firewall werkt snel en is meestal begrijpelijk voor de gebruikers (tenzij ze een geblokkeerde handeling willen uitvoeren).
Voorbeeld van een netwerkniveau proxy server is socks (ftp://ftp.inoc.dl.nec.com)

1.1.1 Proxy servers op toepassingsniveau

Wanneer je een firewall op toepassingsniveau gebruikt, dan is je lokale netwerk niet rechtstreeks verbonden met Internet. Het gegevensverkeer op je lokale netwerk staat los van het gegevensverkeer op het andere netwerk. De proxy server stuurt goedgekeurde pakketten door van het ene naar het andere netwerk.
Proxy servers op toepassingsniveau kunnen een breder gamma van netwerkverkeer verwerken, zoals een FTP sessie, een HTTP aanvraag of een telnet login. Clients geven de aanvraag door aan een bastion host. De proxy servers pakken de aanvraag opnieuw in, contacteren de bijhorende internetbronnen en geven het resultaat terug aan het interne netwerk.
Vragen aan proxy servers moeten anders gesteld worden dan wanneer men op de gewone manier op Internet vragen stelt. Gelukkig ondersteunen de meeste browsers zoals Safari, Opera, Firefox en Internet Explorer het gebruik van proxy servers. Hierenboven bestaan er ook enkele toepassingsniveau proxies die ook normale vragen kunnen behandelen.

1.1.2 Interne Netwerk Adressen en Proxy Firewalls

Wanneer je een proxy firewall gebruikt, doet het er meestal niet toe welke adressen je intern toekent aan je computers. Er ontstaat echter wel een probleem wanneer je op een bepaalde dag via Internet in verbinding wil komen met het ‘echte’ netwerkadres. Alle verkeer naar het echte netwerkadres zal dan door je proxy worden teruggestuurd naar je eigen netwerk.
Om dit probleem op te lossen werden er gereserveerde adressen voorzien voor lokale netwerken.

Gereserveerde netwerkadressen bestaan er voor elke klasse van netwerken (A, B en C). Ruwweg zijn dit volgende adressen:
10.0.0.0 klasse A
172.16.0.0 klasse B
192.168.1.0 klasse C

2 Veiligheidsnormen

The goal of science is to build better mousetraps.
The goal of nature is to build better mice.

Het Amerikaanse ministerie van defensie heeft een inleiding geschreven op netwerkbeveiliging. Deze informatie is gebundeld in 'The Orange Book'. Europa heeft hiervan een afgeleide versie. Verdere boeken zijn 'The Yellow Book' dat de minimum vereisten definïeert voor een veilige computer en 'The Green Book' dat wachtwoordbeheer behandelt.
Deze boeken zijn terug te vinden op http://nsi.org/
Het 'Orange book' bevat 4 grote beveiligingsklassen.

2.1 Beveiligingsniveau klasse D

Laagste norm. Een systeem dat hieraan voldoet biedt geen beveiliging aan bestanden of gebruikers. Dit wordt het vaakst toegekend aan besturingssystemen zoals Windows 95 of aan een onbeschermd netwerk

2.2 Beveiligingsniveau klasse C

Klasse C voorziet algemeen bescherming in de vorm van geheimhouding (wat niet weet, wat niet deert) en de mogelijkheid om activiteiten te loggen
Beveiligingsklasse C1
Een systeem van klasse C1 bereikt geheimhouding door gebruikers en gegevens te scheiden. Alle gebruikers kunnen hun privé gegevens beschermen tegen andere gebruikers. Alle gebruikers werken op eenzelfde veiligheidsniveau. Een systeem zoal Novell Netware (vanaf versie 3.11 voldoet hieraan).
Minimale eisen voor C1:
1. gedefiniëerde en beheerde toegang van gebruikers tot bestanden en diensten
2. identificatie gebruikers vooraleer ze toegang hebben tot gegevens
Beveiligingsklasse C2 Handelingen van gebruikers kunnen worden nagekeken wanneer ze op het netwerk werken.
Minimale eisen zijn deze van C1 plus de volgende eisen:
1. identificatie van een handeling met een specifieke gebruiker (wie doet wat)
2. het netwerk kan bijhouden welke gebruiker welke bestanden gebruikt

2.3 Beveiligingsniveau klasse B

Alle systemen van klasse B bevatten een verplichte bescherming. Elke systeemtoegang heeft een bepaald niveau van beveiliging. Een gebruiker kan onmogelijk bestanden opslaan zonder een beveiligingsniveau's toe te kennen.

Beveiligingsklasse B1
Moeten voldoen aan C2 en het systeem moet veiligheidsniveau's ondersteunen. Eisen:
1. Systeem moet een etiket met een beveiligingsniveau ondersteunen voor elk bestand, communicatiekanaal en I/O apparaat.
2. Deze etiketten worden telkens gebruikt om de toegang te bepalen voor een gebruiker.
3. Een I/O apparaat kan verschillende etiketten hebben, het etiket van de gegevens bepaalt uiteindelijk het niveau.
4. Op alle uitvoer voor de gebruikers (monitor, papier,...) moet het niveau van veiligheid vermeld staan.
5. Ongeoorloofde toegang moet zo mogelijk vastgelegd worden

Beveiligingsklasse B2
Biedt een bescherming tegen aanvallen. Alles van B1 plus:
1. Deelt de gebruiker mee als een beveiligingsniveau wijzigt tijdens het gebruik.
2. Ondersteunt functies voor operators en beheerders.

Beveiligingsklasse B3
Systeem moet eenvoudig blijven, zodat het analyseerbaar blijft.
1. Bij een bestand is er een lijst met gebruikers die toegang hebben én gebruikers die geen toegang hebben
2. Toegang tot het systeem wordt bepaald door externe én interne veiligheidscontrole. Het systeem meldt ontzegde toegangen
3. Het systeem voorziet speciale functies voor een veiligheidsbeheerder. (bv aparte audit trail)

2.4 Beveiligingsniveau klasse A

Het hoogste niveau van beveiliging is identiek aan B3. Alleen moeten de ontwerpers het systeem analyseren aan de hand van formele ontwerpspecificaties. Hierbij moet de ontwerper ook het systeem voldoende verifiëren om er zeker van te zijn dat zijn systeem voldoet.

2.5 Veiligheidsnormen en firewalls

Er zijn verschillende redenen waarom het beveiligingsniveau van het systeem essentieel is voor de beveiliging van de firewall.

1. Controle van de gebeurtenissen in het besturingssysteem voor besturingssystemen van klasse C2 of hoger:
Bepalen welke schade een indringer aanrichtte, bepalen of hij bezig is met een aanval,...
Dit wordt vastgelegd in een *audit* trail.

2. Verplicht toegangsbeheer:
Een toegangsbeheer van klasse B1 of hoger kan gebruikt worden om zich te beschermen tegen Trojaanse Paarden. Door alleen bepaalde gebruikers toegang te verlenen tot systeembestanden kan je de infiltratie van Trojaanse Paarden vermijden. Gebruikers van FTP en daemon processen kan je op een laag toegangsniveau laten draaien, waardoor ze geen belangrijke bestanden kunnen wijzigen.

3 Praktijkvoorbeeld: Firewall met iptables

If the facts don't fit the theory, change the facts.
-- Albert Einstein

Als praktisch voorbeeld zullen we de besproken configuratie met onze server met behulp van iptables uitwerken. Iptables is een pakketfilter op netwerkniveau en is in tegenstelling tot de redelijk overprijsde commerciële firewalls gratis als software geïntegreerd in een Linux systeem. Let erop dat het hier gaat over een beveiliging voor het klasse C netwerk 200.2.2.0. Op dit netwerk draait onze server 200.2.2.3. Deze is ftp, www, snmp, nntp, telnet en nameserver voor alle of voor een beperkt aantal individuen op het internet.

3.1 Iptables regels op Linux

Bij de meeste Linux distributies is iptables de standaard ingebouwde firewall. Met volgend commando zie je als root gebruiker de ingestelde iptables regels:

```
root@linux # iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Voorgaande zie je wanneer er geen regels ingesteld zijn. Je ziet de standaard ingebouwde gegevensstromen:
INPUTVoor binnenkomende gegevens
FORWARD Voor het doorsturen van pakketten
OUTPUT Voor buitengaande gegevens

Let ook op de benaming 'source' en 'destination'-adres. Het feit of iets nu source is of destination, is afhankelijk van de stroom die we bekijken. Het adres van onze server is bijvoorbeeld een 'destination adres' voor mensen die vanuit het Internet onze server willen bereiken. Dit gebeurt via
de 'INPUT' stroom.
Hetzelfde adres is een 'source adres' wanneer bijvoorbeeld iemand van de server naar buiten wil. Dit configureren we via de 'OUTPUT' stroom.

Eigen benamingen:
Het interne netwerk noemen we "GOOD".

Het internet noemen we "BAD"
De verbinding naar buiten heten we "GOOD-BAD".
De verbinding naar binnen heten we "BAD-GOOD".

Interfaces:
eth0:netwerkkaart naar de big bad internetworld

3.2 Opties iptables

-N	NEW	Aanmaken van een nieuwe chain (maximum 8 karakters voor de naam)
-A	APPEND	Toevoegen van een nieuwe chain
-I	INSERT	Toevoegen van een nieuwe regel Mag ook met regel nr. bv. -I INPUT 7 voegt de 7de regel toe aan INPUT
-F	FLUSH	Alle regels verwijderen

Protocol:
-p protocol zoals tcp, udp, icmp, ip

TCP uitbreidingen:
Bij het gebruik van het TCP protocol -p tcp kan je extra opties meegegeven:

--sport	Source port(s). Eén poort bv 80 of een range bv 1024:65536 Veruit het meest sportieve dat je met Linux kan doen :-)
--dport	Destination port(s). Eén poort bv 80 of een range bv 21:1023
--syn	Kijkt enkel syn pakketten na
--tcp-flags	SYN,ACK,FIN,RST,URG,PSH of NONE of ALL Meerdere vlaggen kan je met een komma aangeven bv SYN,ACK

UDP uitbreidingen:
Bij het gebruik van het UDP protocol -p udp kan je extra opties meegegeven:

--sport	Source port(s). Eén poort bv 80 of een range bv 1024:65536
--dport	Destination port(s). Eén poort bv 80 of een range bv 21:1023

ICMP uitbreidingen:
Bij het gebruik van het ICMP protocol -p icmp kan je volgende optie meegegeven:

--icmp-type	Soort ICMP bericht zoals echo-request, echo-reply,... Mag vooraf gaan door een negatie (!) Volledige lijst van types krijg je met iptables -p icmp --help

Interface:
-i interface Geeft de inkomende interface aan waarop de chain betrekking heeft

Adressen:

-s	source-address Bron-adres
-d	doel-address Doel-adres

Protocol:

-p	protocol zoals tcp, udp, icmp, ip

Acties:

-j	jump Springt uit de regels van de chain na het uitvoeren
ACCEPT	Pakket aanvaarden (en verdere regels negeren).
DROP	Pakket weigeren (en op de grond gooien).
REJECT	Pakket weigeren en vriendelijk zeggen dat het geweigerd werd.
LOG	Loggen en verder gaan met volgende regels

Loggen:

--log-prefix	Toevoegen van tekst voor de log tussen dubbele quotes
--log-level	Niveau van loggen naar de syslog. 7 is een goede waarde.
-l log	Loggen wanneer aan de regel voldaan wordt

Match extensies:

-m state --status	Schakelt het gebruik van verbindingsstatus aan Statussen waar je mee wil matchen zijn: (altijd eerst -m state gebruiken!)

NEW	Een nieuwe verbinding
RELATED	Nieuwe verbinding die te maken heeft met een eerder toegelaten verbinding
ESTABLISHED	Een eerder opgestarte verbinding (SYN is al gebeurd)
INVALID	Onmogelijk om de status vast te stellen

-m mac --mac-source	Matched MAC adressen bv -m mac --mac-source 00:60:08:91:CC:B7
-m limit --limit *nr*	Limiteert het aantal matches (bv om te grote logs te voorkomen) Default limit is 3/hour De nummer (*nr*) = Het gemiddeld aantal per seconde Dit mag ook met units: bv 5/day, 5/hour, 5/minute, 5/s
-m limit --limit-burst	Geeft een maximum aantal, vooraleer limit ingeschakeld wordt Default limit-burst is 5/s

Algemeen:

!	Negatie bv. "! --syn" alles behalve syn pakketten

Voor meer opties en mogelijkheden van iptables (zoals pakketten tellen, masquerading) zie de IPTABLES-HOWTO op http://www.linuxguruz.com/iptables/howto/ .
Volgende regels zijn verre van volledig en u kan mij niet verantwoordelijk stellen voor de rechtstreekse of onrechtstreekse gevolgen van het gebruik. Als de haren van uw hond uitvallen dan heeft uw hond pech, als uw haren uitvallen, dan bent u kaal.
Mogelijke fouten/verbeteringen kan je melden op jan.celis@kdg.be

3.3 De iptables regels bij het voorbeeld

Source-routing kan ook gedisabled worden in de kernel van de meeste UNIX systemen (BSD, Solaris, Linux,...).Voor onze regels maken we twee nieuwe chains aan. Deze voegen we toe aan de standaard INPUT en OUTPUT chains. De standaard forward chain wordt voor de duidelijkheid niet aangepast.

1. Chain "GOOD-BAD" voor buitengaande pakketten toevoegen aan "OUTPUT"

```
iptables -N GOOD-BAD
iptables -A OUTPUT -j GOOD-BAD
```

2. Chain "BAD-GOOD" voor binnenkomende pakketten toevoegen aan "INPUT"

```
iptables -N BAD-GOOD
iptables -A INPUT -j BAD-GOOD
```

3. Geen lokale ipnummers vanuit internet toelaten (IP spoofing)
Uiteraard worden deze personen gelogd !

```
iptables -A BAD-GOOD -s 200.2.2.0/24 -l -j DROP
iptables -A BAD-GOOD -s 127.0.0.0/8  -l -j DROP
```

4. TCP en IP regels

```
iptables -A BAD-GOOD -p tcp                            -d 200.2.2.3/32 --dport 80       -j ACCEPT
iptables -A BAD-GOOD -p tcp -s 198.8.8.0/24            -d 200.2.2.3/32 --dport 23       -j ACCEPT
iptables -A BAD-GOOD -p tcp -s 197.7.7.3/32             -d 200.2.2.0/24 --dport 23       -j
ACCEPT
iptables -A BAD-GOOD -p tcp -s 0.0.0.0/0               -d 200.2.2.3/32 --dport 25       -j ACCEPT
iptables -A BAD-GOOD -p tcp -s 0.0.0.0/0               -d 200.2.2.3/32 --dport 119      -j ACCEPT
iptables -A BAD-GOOD -p tcp -s 0.0.0.0/0               -d 200.2.2.3/32 --dport 21       -j ACCEPT
iptables -A GOOD-BAD -p tcp -s 200.2.2.3/32 --sport 20 -d 0.0.0.0/0  --dport 1025:65535 -j ACCEPT
iptables -A GOOD-BAD -p tcp -s 200.2.2.0/24            -d 0.0.0.0/0                     -j ACCEPT
iptables -A BAD-GOOD -p tcp ! --syn -s 0.0.0.0/0       -d 200.2.2.0/24--dport1025:65535 -j ACCEPT
```

5. DNS regels (met UDP)

```
iptables -A BAD-GOOD -p UDP       -s 0.0.0.0/0    --sport 53 -d 200.2.2.3/32 --dport 53 -j ACCEPT
iptables -A GOOD-BAD -p UDP       -s 200.2.2.0/24 --sport 53 -d 0.0.0.0/0    --dport 53 -j ACCEPT
iptables -A BAD-GOOD -p TCP --syn -s 199.100.105.2/32         -d 200.2.2.3/32 --dport 53 -j
ACCEPT
```

6. ICMP regels

```
iptables -A BAD-GOOD -p ICMP      -s 0.0.0.0/0 -d 0.0.0.0/0 --icmp-type redirect -l     -j DROP
iptables -A BAD-GOOD -p ICMP      -s 0.0.0.0/0 -d 0.0.0.0/0                             -j ACCEPT
iptables -A GOOD-BAD -p ICMP      -s 0.0.0.0/0                                          -j ACCEPT
```

7. De rest weigeren en loggen

```
iptables -A BAD-GOOD -l -j REJECT
iptables -A GOOD-BAD -l -j REJECT
```

8. Geen source routing flags

```
iptables -A input -m ipv4options --ssrr -j DROP
iptables -A input -m ipv4options --lsrr -j DROP
```

3.4 Ander gebruik van iptables

Tegen Syn-flood:

```
# iptables -A FORWARD -p tcp --syn -m limit --limit 1/s -j ACCEPT
```

Tegen port scanners:

```
# iptables -A FORWARD -p tcp --tcp-flags SYN,ACK,FIN,RST -m limit --limit 1/s -j ACCEPT
```

Tegen ping of death:

```
# iptables -A FORWARD -p icmp --icmp-type echo-request -m limit --limit 1/s -j ACCEPT
```

Tegen brute force attacks op ssh:

```
# iptables -A INPUT -i eth0 -p tcp --dport 22 -m state --state NEW -m recent --set --name SSH
# iptables -A INPUT -i eth0 -p tcp --dport 22 -m state --state NEW -m recent --update --seconds
60 --hitcount 8 --rttl --name SSH -j DROP
```

Binnenkomende SSH connecties worden beperkt tot 8 per minuut.

Port Forwarding:
Port forwarding betekent dat je gateway op internet verkeer naar een bepaalde poort kan doorsturen naar bv een interne webserver.
Dit zijn de iptables regels die je moet toepassen om een port forward te doen van de gateway 200.2.2.2 poort 8888 naar de interne webserver 192.168.1.1 poort 80 .

```
# iptables -t nat -A PREROUTING -p tcp -i eth0 -d xxx.xxx.xxx.xxx --dport 8888 -j DNAT --to
192.168.1.1:80
# iptables -A FORWARD -p tcp -i eth0 -d 192.168.1.1 --dport 80 -j ACCEPT
```

3.5 Stateful filter met iptables

Dankzij *connection tracking* kan iptables gebruikt worden als stateful filter (Een welbepaalde verbinding wordt gelogd en alle pakketten worden nagekeken of ze wel tot deze verbinding horen).
Bij het gebruik van connection tracking maakt iptables in de *state table* een rij aan voor de verbinding. In deze status tabel staat volgende informatie:

- Het protocol gebruikt voor de verbinding
- Het bron en doeladres, de bron en doelpoort
- Een lijst met het vorige omgekeerd (om het antwoord vast te leggen)
- De levenstijd van deze verbinding (time-out waarna deze wordt verwijderd)
- De TCP status van de verbinding (enkel bij TCP)

De connection-tracking status van de verbinding

Voorbeeld van een statustabel rij bij iptables:

```
TCP 6 93 SYN_SENT src=192.168.1.34 dst=172.16.2.23  sport=1054 dport=21 [UNREPLIED]
                 src=172.16.2.23  dst=192.168.1.34 sport=21   dport=1054 use=1
```

TCP Protocolnaam
6 Protocolnummer (6 voor TCP)
93 Levenstijd van deze lijn. Hierna verdwijnt de lijn uit de statustabel.
SYN_SENT TCP status van deze verbinding
src, dst Bron en doeladres
sport, dport Bron en doelpoort
UNREPLIED Er is nog geen antwoord gekomen op het starten van de verbinding
src,dst Bron en doel werden omgekeerd (dit staat klaar voor het antwoord)

Nadat de verbinding is opgestart (ESTABLISHED) wordt de tabel aangepast:

```
TCP 6 41294 ESTABLISHED src=192.168.1.34 dst=172.16.2.23  sport=1054 dport=21
                        src=172.16.2.23  dst=192.168.1.34 sport=21   dport=1054 [ASSURED] use=1
```

De [UNREPLIED] markering werd dus verwijderd.na het antwoord op de SYN. Bij een established verbinding komt er [ASSURED] in de plaats. De verbinding is opgestart, dus de timeout waarde werd zeer hoog gelegd. (41294).

We bekijken de OUTPUT regel voor een FTP client (binnen ons bedrijf) die een FTP verbinding opent naar een externe FTP server.

```
# iptables -A OUTPUT -p tcp -m state --state NEW,ESTABLISHED -j ACCEPT
```

Deze OUTPUT regel geeft aan dat nieuwe en vastgelegde, uitgaande TCP verbindingen zijn toegelaten.

NEW zorgt ervoor dat de firewall ALLE pakketten met een SYN flag worden toegelaten. Hierdoor mogen er dus nieuwe verbindingen worden opgestart. Per verbinding komt er een extra lijn in de statustabel.
ESTABLISHED zorgt ervoor dat verkeer dat reeds opgestart is, toegelaten wordt.
De -m optie van iptables zorgt ervoor dat de juiste module wordt gebruikt bij iptables (in dit geval dus de standaard *state* module).
We bekijken nu het verkeer dat terugkomt. Dit is dus voor onze firewall inkomend verkeer. Het verkeer komt dus van de externe FTP server naar onze interne client.

```
# iptables -A INPUT -p tcp -m state --state ESTABLISHED -j ACCEPT
```

De verbinding naar de server is reeds aangevraagd en opgestart door de client. Er moet dus niet NEW toegestaan worden, maar enkel ESTABLISHED. Dit verkeer werd in de status tabel vastgelegd in de tweede regel (waarbij src en dst werden gewisseld).

Voor UDP kan er pseudo-stateful tracking gebruikt worden. Echt stateful kan niet aangezien er geen opstart status en stop status bestaat zoals bij TCP. Dit geeft volgende regels voor UDP:

```
# iptables -A OUTPUT -p udp -m state --state NEW,ESTABLISHED -j ACCEPT
# iptables -A INPUT  -p udp -m state --state ESTABLISHED     -j ACCEPT
```

ICMP regels zijn bijna hetzelfde:

```
# iptables -A OUTPUT -p icmp -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
# iptables -A INPUT  -p icmp -m state --state ESTABLISHED,RELATED     -j ACCEPT
```

Het grote verschil met voorgaande regels van UDP en TCP is de RELATED optie.

Met de RELATED optie laat iptables verkeer toe dat iets of wat te maken heeft met andere ESTABLISHED verbindingen. Hiervoor is de statustabel dus handig.
Voorbeelden waarbij de ICMP berichten related zijn:

Een ICMP error bericht dat teruggestuurd wordt van een UDP of TCP verbinding, die al in de status tabel staat.

Het starten van een inkomend FTP data kanaal op poort 20, nadat het opstarten van de FTP verbinding al is gebeurd op poort 21.

Met de NEW optie voor UITGAAND verkeer, kan een aanvraage van een ICMP programma zoals as ping naar buiten gaan, de ESTABLISHED optie voor binnenkomend verkeer zal zorgen dat het antwoord op een ping terug naar binnen kan. Aangezien er geen NEW optie is bij INKOMEND verkeer, kan er geen PING naar binnen gaan.

De regels voor RELATED verkeer worden gedefinieerd in connection-tracking modules. Het afhandelen van FTP verkeer kan bijvoorbeeld met de ip_conntrack_ftp module. Voor nieuwe diensten en applicaties zoals VoIP worden er gewoon extra modules toegevoegd.
Om bijvoorbeeld de ftp-module toe te voegen gebruik je volgend commando als root:

```
# modprobe ip_conntrack_ftp
```

Voor FTP voegen we extra regels toe voor het verkeer van poort 20 met de optie RELATED. RELATED geeft aan dat er in de statustabel ergens een aanvraag op poort 21 moet zijn voor dezelfde client en server. De volledige FTP regels worden dus:

```
# iptables -A OUTPUT -p tcp --dport 21 -m state --state NEW,ESTABLISHED       -j ACCEPT
# iptables -A INPUT  -p tcp --sport 21 -m state --state ESTABLISHED           -j ACCEPT
# iptables -A INPUT  -p tcp --sport 20 -m state --state ESTABLISHED,RELATED   -j ACCEPT
# iptables -A OUTPUT -p tcp --dport 20 -m state --state ESTABLISHED           -j ACCEPT
```

1 Praktijkvoorbeeld: Firewall met cisco router

Marge, I agree with you - in theory. In theory, Communism works. In theory.
-- Homer Simpson

Als praktisch voorbeeld zullen we de besproken configuratie met onze server met behulp van een Cisco router en access-lists uitwerken. Dit is slechts een theoretische voorbeeldconfiguratie. De syntax van de gebruikte commando's is uitvoerbaar. Uit eerdere praktische tests is gebleken dat Cisco routers het niet zo nauw nemen met source ports. Mogelijke fouten of verbeteringen mag je altijd melden op jan.celis@kdg.be . U kan mij niet verantwoordelijk stellen voor de rechtstreekse of onrechtstreekse gevolgen van het gebruik. Als uw kat niet meer wil eten na het toepassen van de regels, probeer dan eens Surimi sticks van den Aldi, als je zelf niet meer wil eten na het toepassen van de regels, dan verhonger je.

1.1 ACL regels op Cisco IOS

Access Control Lists worden op een Cisco router sequentieel toegepast. Van zodra een bepaalde regel matched, spring je uit de regels (al-dan-niet met een permit of deny).
Met het commando 'show ip access-lists' kan je firewall regels bekijken.

Aanmaken van een named access-list:
Router(config)# ip access-list extended INKOMEND
Router(config-ext-nacl)# Hier geef je de regels in voor deze access-list

Toepassen van een named access-list:
Hiervoor ga je naar de interface en definieer je of je regels inkomend (in) of uitgaand (out) werken.

```
Router(config)# ip access-list extended INKOMEND
Router(config-ext-nacl)#  permit tcp any any eq 80
Router(config-ext-nacl)#  deny ip any any
Router(config-ext-nacl)#  exit
Router(config)# interface FastEthernet 0/0
Router(config-if)# ip access-group INKOMEND in
```

Deze regels zorgen er dus voor dat iedereen op mijn webserver kan (permit tcp any any eq 80) en dat er geen andere connecties mogelijk zijn. (deny ip any any)
Ze zijn toegepast op inkomend verkeer op FastEthernet 0/0

Algemene vorm van een uitgebreide access-list met een naam:
permit/deny protocol BRONadres wildcard DOELadres wildcard
Onthoud dat je altijd weergeeft VAN waar het verkeer komt en NAAR waar het verkeer gaat.
Opgelet! Bron en doel hangen af van het feit of je inkomend (in) of uitgaand (out) verkeer hebt.

Protocol:
tcp, udp, icmp en ip

ip omvat zowel tcp, udp als icmp daar het enkel filtert op ip adres

Adressen:
Een adres bestaat uit een IP adres of Netwerkadres met een wildcardmask.
Een wildcard mask zegt welke bits moeten overeenkomen en welke bits moeten genegeerd worden.
Een “0” bit betekent dat deze bit moet nagekeken worden
Een “1” bit betekent dat deze bit mag genegeerd worden (is dus EENder wat!)
Eenvoudig bekeken is de wildcardmask het omgekeerde van de subnetmask.

Voorbeelden:
- Het hele netwerk 192.168.1.0 schrijf je als 192.168.1.0 0.0.0.255
- De computer 192.168.1.1 schrijf je als 192.168.1.1 0.0.0.0 of host 192.168.1.1
- Iedereen schrijf je als 0.0.0.0 255.255.255.255 of any

Zoals je ziet mag je **host** gebruiken voor één bepaalde computer en **any** als je iedereen gebruikt

TCP en UDP uitbreidingen:
Bij het gebruik van tcp of udp als protocol kan je na elk adres/wildcard weergeven voor welke poorten dit bestemd is:
permit/deny tcp BRONadres wildcard BRONpoort DOELadres wildcard DOELpoort

TCP/UDP poort optie	Betekenis
eq 80	Enkel poort 80
gt 1023	Poortnummers groter dan 1023
lt 1024	Poortnummers kleiner dan 1024
neq 80	Poortnummers niet gelijk aan 80 (dus alles behalve 80)
range 81 1023	Poortnummer van 81 tot 1023
syn	Opstarten van een verbinding (enkel TCP)
established	Reeds opgestarte verbinding (syn is voorbij) (enkel TCP)

Volgend voorbeeld laat clients uit het netwerk 192.168.1.0 toe op de webserver 192.168.1.1

```
Router(config)# ip access-list extended INKOMEND
Router(config-ext-nacl)#  permit tcp 192.168.1.0 0.0.0.255 gt 1023 host 192.168.1.1 eq 80
```

ICMP uitbreidingen:
Bij het gebruik van icmp als protocol kan je na elk adres/wildcard weergeven voor welk type icmp pakketten dit geldt:

ICMP optie	Betekenis
nr (0-255)	Nummer van ICMP type (zie bij ICMP)
echo	Echo request (ping)
echo-reply	Antwoord op ping (pong)
host-unreachable	Host niet bereikbaar
port-unreachable	Poort niet bereikbaar

ttl-exceeded	Time To Live is verstreken (pakket is zwervend)

Loggen:
Achter elke regel kan je toevoegen of het matchen van een regel moet vastgelegd worden in een log door het woord **log** toe te voegen.

```
Router(config)# ip access-list extended INKOMEND
Router(config-ext-nacl)#  deny tcp any host 192.168.1.1 eq 22 log
```

Logt elke poging om aan te melden via SSH (poort 22)

```
%SEC-6-IPACCESSLOGP: list INKOMEND denied tcp 10.0.0.1(2341) -> 192.168.1.1(22), 1 packet
```

Opmerking: wanneer er niet expliciet op een bepaalt poortnummer gelogd wordt (dus enkel op IP adres), geef je best de optie gt 0 mee, anders worden poorten niet gelogd. (Dit is geen bug maar een efficiënt feature, naar poorten kijken kost processortijd!). Voorbeeld:

```
deny tcp any host 192.168.1.1 gt 0 log
```

1.1 De Cisco access-lists bij het voorbeeld

1. "GOOD-BAD" named access-list aanmaken

```
Router(config)#ip access-list extended GOOD-BAD
Router(config-ext-nacl)#
```

2. "BAD-GOOD" named access-list aanmaken

```
Router(config)#ip access-list extended BAD-GOOD
Router(config-ext-nacl)#
```

3. Geen lokale ipnummers vanuit internet toelaten (IP spoofing)

```
Router(config)#ip access-list extended BAD-GOOD
Router(config-ext-nacl)# deny ip 200.2.2.0 0.0.0.255       any
Router(config-ext-nacl)# deny ip 127.0.0.0 0.255.255.255   any
```

4. TCP en IP regels

```
Router(config)#ip access-list extended BAD-GOOD
Router(config-ext-nacl)#permit tcp any                     host 200.2.2.3        eq 80
Router(config-ext-nacl)#permit tcp 198.8.8.0 0.0.0.255     host 200.2.2.3        eq 23
Router(config-ext-nacl)#permit tcp host 197.7.7.3          200.2.2.0 0.0.0.255   eq 23
Router(config-ext-nacl)#permit tcp any                     host 200.2.2.3        eq 25
Router(config-ext-nacl)#permit tcp any                     host 200.2.2.3        eq 119
Router(config-ext-nacl)#permit tcp any                     host 200.2.2.3        eq 21
Router(config-ext-nacl)#permit tcp any                     host 200.2.2.0        gt 1023
established

Router(config)#ip access-list extended GOOD-BAD
Router(config-ext-nacl)#permit tcp host 200.2.2.3 eq 20    any                   gt 1023
Router(config-ext-nacl)#permit tcp 200.2.2.0 0.0.0.255     any
```

5. DNS regels (met UDP)

```
Router(config)#ip access-list extended BAD-GOOD
Router(config-ext-nacl)#permit udp any                     host 200.2.2.3        eq 53
Router(config-ext-nacl)#permit tcp host 199.100.105.2      host 200.2.2.3        eq 53 syn

Router(config)#ip access-list extended GOOD-BAD
Router(config-ext-nacl)#permit udp 200.2.2.0 0.0.0.255 eq 53 any                 eq 53
```

6. ICMP regels

```
Router(config)#ip access-list extended BAD-GOOD
```

```
Router(config-ext-nacl)#deny icmp any                    any                5
(opm: 5 is redirect)

Router(config-ext-nacl)#permit icmp any any

Router(config)#ip access-list extended GOOD-BAD
Router(config-ext-nacl)#permit icmp any any
```

7. De rest expliciet weigeren (dit gebeurt ook default)

```
Router(config)#ip access-list extended GOOD-BAD
Router(config-ext-nacl)#deny ip any any
Router(config-ext-nacl)#exit
Router(config)#ip access-list extended BAD-GOOD
Router(config-ext-nacl)#deny ip any any
```

8. Source routing uitschakelen

```
Router# configure terminal
Router(config)#no ip source-route
```

9. **TOEPASSEN van de REGELS!**

GOOD-BAD access-list toepassen op FastEthernet 0/0 uitgaand

```
Router(config)#int FastEthernet 0/0
Router(config-if)#ip access-group GOOD-BAD out
```

BAD-GOOD access-list toepassen op FastEthernet 0/0 inkomend

```
Router(config)#int FastEthernet 0/0
Router(config-if)#ip access-group BAD-GOOD in
```

1.2 Stateful Filter met Cisco Firewall

Je kan een Cisco router instellen, zodat deze stateful filtering van het verkeer doet.
Je doet dit beperkt al door gebruik te maken van ESTABLISHED of SYN bij het definiëren van firewall regels. Uitgebreider kontroleren op sessies en vervalste pakketten kan met wat bij Cisco '*reflexive access lists'* wordt genoemd.

Voorbeeldinstelling met reflexive access lists:

```
interface FastEthernet 0/0
 ip access-group INKOMEND in
 ip access-group UITGAAND out
!
ip access-list extended UITGAAND
 permit tcp any any reflect TCPVERKEER
!

ip access-list extended INKOMEND
 permit icmp any any
 deny udp any any
 evaluate TCPVERKEER
```

2 Poort nummers

If God had intended man to program,
We'd be born with serial I/O ports.

2.1 Well-known poortnummers

Dit is een selectie van vastgelegde poortnummers (aan de kant van de server). Deze nummers zijn vastgelegd in RFC's (Request For Comment).

Protocol	Afkorting	Beschrijving	Poort
File Transfer Protocol	FTP-DATA	File transfer tussen computers (data gedeelte)	20
File Transfer Protocol	FTP	File transfer tussen computers (controleverbinding)	21
TELNET protocol	TELNET	Remote terminal access	23
Simple Mail Transfer Protocol	SMTP	E-mail transfer	25
Domain Name Protocol	DOMAIN	Definieert nameserver	53
Gopher	GOPHER	Informatie zoek systeem opgebouwd als bestandssysteem	70
Finger Protocol	FINGER	Geeft informatie over een specifieke gebruiker	79
HyperText Transmission	HTTP (WWW)	Informatieuitwisseling tussen webserver en browser	80
Post Office Protocol	POP	Protocol voor verdeling van mail	110
Network News Transfer Protocol	NNTP	Protocol voor verdeling van nieuws berichten	119

2.2 Poortnummers trojans

Deze poorten worden door allerlei programma's niet meer voor "normale" netwerkactiviteiten gebruikt, maar kunnen een systeem beschadigen, een gedeelte van de controle overnemen of systeembronnen verbruiken / misbruiken.

Poort	Protocol	Naam van Trojan
21	TCP	Blade Runner, Doly Trojan, Fore, Invisible FTP,WebEx, WinCrash
23	TCP	Tiny Telnet Server
25	TCP	Antigen, Email Password Sender, Haebu Coceda, Shtrilitz Stealth, Terminator, WinPC, WinSpy, Kuang
31	TCP	Hackers Paradise
80	TCP	Executor
456	TCP	Hackers Paradise
555	TCP	Ini-Killer, Phase Zero, Stealth Spy
666	TCP	Satanz Backdoor

1001	TCP	Silencer, WebEx
1011	TCP	Doly Trojan
1095, 1097, 1098, 1099	TCP	Rat
1170	TCP	Psyber Stream Server, Voice
1234	TCP	Ultors Trojan
1243, 6711,6776	TCP	Sub 7
1245	TCP	VooDoo Doll
1349	UDP	Back Ofrice DLL
1492	TCP	FTP99CMP
1600	TCP	Shivka-Burka
1807	TCP	SpySender
1981	TCP	Shockrave
1999	TCP	BackDoor 1.00-1.03
2001	TCP	Trojan Cow
2023	TCP	Ripper
2115	TCP	BUGS
2140, 3150	TCP,UDP	Deep Throat
2140, 3150	TCP	The Invasor
2801	TCP	Phineas Phucker
3024, 5742	TCP	WinCrash
3129	TCP	Masters Paradise
3700, 9872, 9873, 9874, 9875, 10067,10167	TCP	Doom
4092	TCP	WinCrash
4567	TCP	File Nail 1
4590	TCP	ICQTrojan
5000	TCP	Bubbel
5000, 5001	TCP	Sockets de Troie
5321	TCP	Firehotcker
5400, 5401	TCP	Blade Runner
7306, 7307,7308	TCP	NetMonitor
12223	TCP	Hack´99 KeyLogger

2.3 Handige programma's voor firewall fine tuning

2.3.1 tcpdump

```
$ tcpdump
13:01:12.629691 192.168.1.187.1046 > sniffer.telnet: P 3083102:3083103(1)
ack 2962977797 win 7920 (DF)
13:01:12.629691 sniffer.telnet > 192.168.1.187.1046:P(1)ack 1 win 32667(DF)
13:01:12.789691 192.168.1.187.1046 > sniffer.telnet: . ack 2 win 7919 (DF)
13:01:13.599691 arp who-has 128.8.10.90 tell HEBEDU01
13:01:19.549691 sniffer.1627 > HEBEDU01.domain: 44664+ (42)
13:01:19.599691 arp who-has 192.33.4.12 tell HEBEDU01
13:01:33.239691 sniffer.1629 > HEBEDU01.domain: 44665+ (42)
13:01:36.599691 HEBEDU01.domain > sniffer.1624: 61844 ServFail 0/0/0 (30)
13:01:36.599691 sniffer > HEBEDU01: icmp: sniffer udp port 1624 unreachable
13:01:39.719691 NTAS1.1029 > 192.168.1.255.41508: udp 188
13:01:39.719691 NTAS1.netbios-dgm > 192.168.1.255.netbios-dgm: udp 358
13:01:39.719691 0:a0:24:a9:20:68 > 3:0:0:0:0:1 sap f0 ui/C len=320
2c00 ffef 0800 0000 0000 0000 494e 464f
2020 2020 2020 2020 2020 2000 4e54 4153
3120 2020 2020 2020 2020 2000 ff53 4d42
2500 00
13:01:39.719691 NTAS1.netbios-ns > 192.168.1.255.netbios-ns: udp 50
13:01:39.719691 0:a0:24:a9:20:68 > 3:0:0:0:0:1 sap f0 ui/C len=320
```

```
2c00 ffef 0800 0000 0000 0000 4443 5052
4143 2020 2020 2020 2020 2000 4e54 4153
3120 2020 2020 2020 2020 2000 ff53 4d42
2500 00
13:01:40.199691 arp who-has sniffer tell NTAS1
13:01:40.199691 arp reply sniffer is-at 0:a0:24:93:8e:55
13:01:40.199691 NTAS1.netbios-ns > sniffer.netbios-ns: udp 62
13:01:40.459691 NTAS1.netbios-ns > 192.168.1.255.netbios-ns: udp 50
13:01:40.599691 HEBEDU01.domain > sniffer.1625: 44664 ServFail 0/0/0 (42)
13:01:40.599691 sniffer > HEBEDU01: icmp: sniffer udp port 1625 unreachable
13:12:43.399691 192.168.1.187.netbios-ns > 192.168.1.255.netbios-ns: udp 68
13:12:54.929691 sniffer.netbios-ns > 192.168.1.255.netbios-ns: udp 68
13:12:55.019691 sniffer.netbios-dgm > 192.168.1.255.netbios-dgm: udp 221
```

2.3.2 nmap

Het tooltje nmap is een eenvoudige poortscanner. Het gokt ook, aan de hand van openstaande poorten en kleine protocolafwijkingen, welk besturingssysteem er draait.

```
$ nmap -sT -O 192.168.1.1
Starting nmap V. 2.2-BETA4 by Fyodor
Interesting ports on NTAS1 (192.168.1.1):
Port State Protocol Service
111 open tcp sunrpc
135 open tcp loc-srv
139 open tcp netbios-ssn
799 open tcp unknown
852 open tcp unknown
854 open tcp unknown
857 open tcp unknown
1521 open tcp ncube-lm
1526 open tcp pdap-np
12345 open tcp NetBus
TCP Sequence Prediction: Class=trivial time dependency
Difficulty=2 (Trivial joke)
Remote operating system guess: Windows NT4 / Win95 / Win98
```

2.3.3 nemesis

Met het tooltje nemesis kan je custom-pakketten genereren. Typische aanpassingen die je doet zijn aangepaste poortnummers, aangepaste IP adressen, aangepaste MAC adressen. Door extra logging bij de firewall of sniffing aan de andere kant van de firewall, kan je kijken of er toch nog speciale pakketten door je firewall kunnen sluipen.
Nemesis kan ARP, DNS, ETHERNET, ICMP, IGMP, IP, OSPF, RIP, TCP en UDP maken en injecteren.

Een goede firewall kan dynamisch poorten openen en sluiten voor een TCP verbinding (dit heet ook wel stateful inspection). Nemesis kan je gebruiken om te zien hoe stateful je firewall regels zijn ingesteld. Deze test is belangrijk om packet spoofing tegen te gaan.

Voorbeeld: We testen een regel die NetBIOS verkeer (TCP poort 139) toelaat tussen twee computers 10.10.10.1 en 192.168.1.1.

Bij het opstarten van de verbinding stuurt 192.168.1.1 een TCP SYN pakket vanuit poort 1266 naar 10.10.10.1 op poort 139. Hier zie je een gedeelte van de tcpdump:

```
19:34:48.663980 192.168.1.1.1266 > 10.10.10.1.139: S 847815674:847815674(0)
 win 16384 <mss1460,nop,nop,sackOK> (DF)
19:34:48.664567 10.10.10.1.139 > 192.168.1.1.1266: S 4141875831:4141875831(0)
 ack 847815675 win 17520 <mss 1460,nop,nop,sackOK> (DF)
19:34:48.665586 192.168.1.1.1266 > 10.10.10.1.139: . ack 1 win 17520 (DF)
```

De firewall laat hier de specifieke combinatie IP adressen en poortnummers toe om de verbinding te starten. De volgende TCP pakketten zullen een ACK (acknowledge) vlag hebben totdat de connectie wordt afgesloten door een FIN (finish) of RST (reset) vlag. Op dat moment kan je de firewall met 'nemesis tcp' testen.

Deze test kijkt na of de firewall een extra pakket toelaat met een FIN of RST vlag (zo kunnen we een bestaande verbinding afsluiten ;-):

```
# nemesis tcp -S 192.168.1.1 -D 10.10.10.1 -fF -x 1266 -y 139
# nemesis tcp -S 192.168.1.1 -D 10.10.10.1 -fR -x 1266 -y 139
```

We draaien tcpdump uiteraard aan de andere kant van de firewall (op het 10.x netwerk) om te kijken of onze pakketten door de regels kunnen geraken.
Normaalgezien moeten, bij een correct ingestelde 'stateful firewall', deze pakketten geblokkeerd worden aangezien de TCP sequence nummers niet kloppen (nemesis geeft random opvolgingsnummers). Als dit verkeer wordt toegelaten, is een DoS attack op 10.10.10.1 niet uit te sluiten. Door constant RST pakketten te sturen, wordt dan elke verbinding onmogelijk gemaakt.

Verder zie je hoe de firewall met ACK pakketten omgaat. Sommige hacker achterpoortjes werken volledig op ACK pakketten (Kijk bv op http://www.ntsecurity.nu/toolbox/ackcmd/ voor het tooltje AckCmd of zoek stcpshell op).

```
# nemesis tcp -S 192.168.1.1 -D 10.10.10.1 -fA -x 1266 -y 139
```

Hetzelfde kan je voor UDP verbindingen doen. Gezien de onzekere natuur van UDP zullen de meeste firewalls een tijdslimiet opleggen aan UDP verbindingen. Met 'nemesis udp' test je gelijkaardig als met tcp, alleen op UDP poort 135 (ook NetBIOS verkeer). Maak hiervoor een verbinding met bv netcat tussen 192.168.1.1 en 10.10.10.1. Voer dan volgend commando uit om een 5 minuten (300 s) time-out te testen.

```
# sleep 300; nemesis udp -S 192.168.1.1 -D 10.10.10.1 -x 1266 -y 135
```

Wanneer je tcpdump draait aan de andere kant van de firewall, en je nemesis UDP verkeer geraakt er door, dan is de time-out van de firewall meer dan 5 minuten.

Je kan ook testen hoe een firewall reageert op ICMP tunneling programma's zoals het tooltje *Loki*. Een firewall mag bijvoorbeeld nooit antwoorden op een ICMP reply, wanneer de ICMP request (bv door een ping) niet van een intern adres komt.

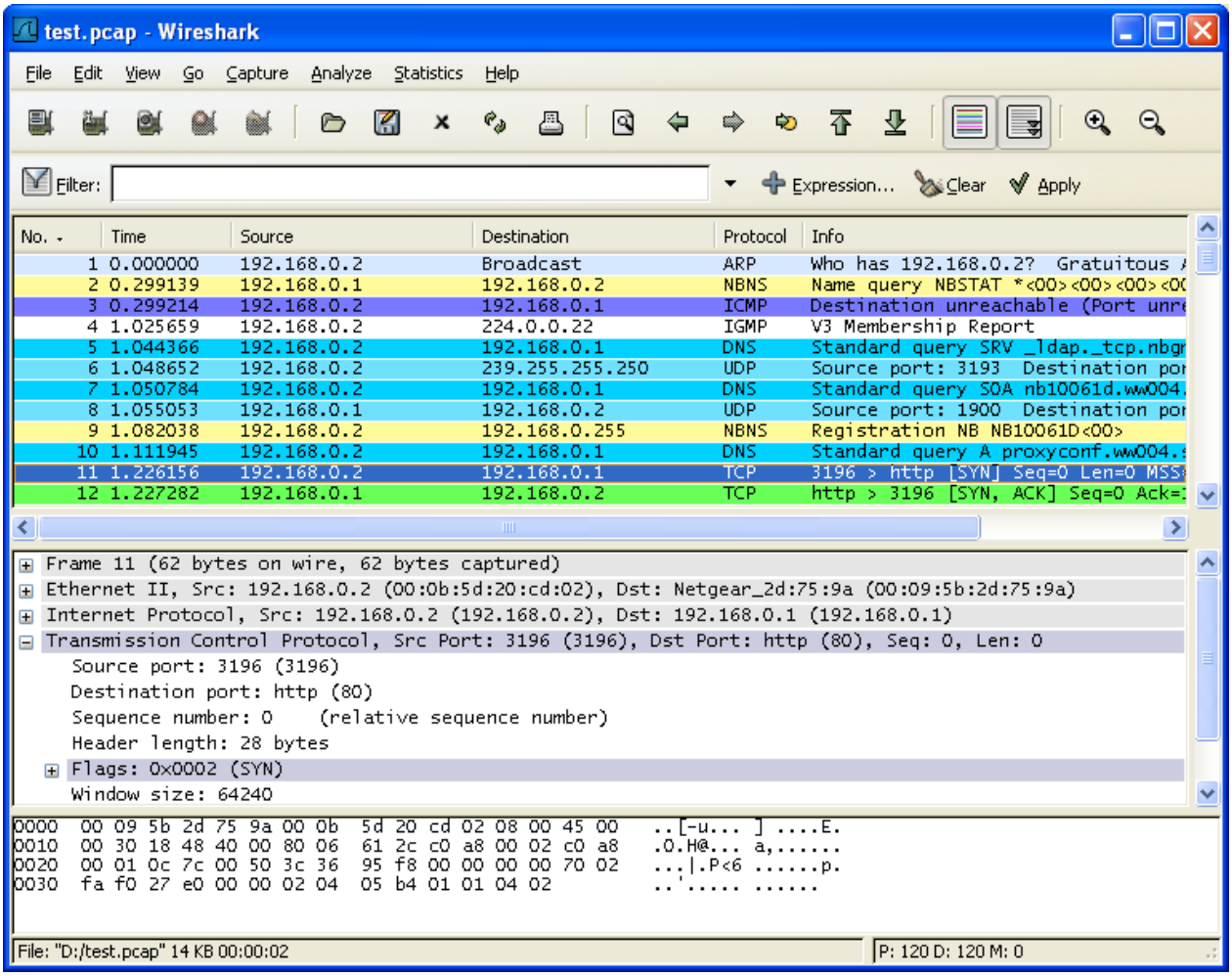
```
# nemesis icmp -S 192.168.1.1 -D 10.10.10.1 -i 0 -c 0
```

Je kan alle mogelijke (255) ICMP types testen (hoewel er maar rond de 40 echt gebruikt worden) om te kijken hoe de firewall hier mee omgaat. Best dat het pakket geblokkeerd wordt, en hopelijk krijgt het geen toegang tot het 10.x netwerk.

```
#!/bin/sh
# Genereren van icmp pakketten van alle mogelijke en onmogelijke types
# De firewall mag dit niet doorlaten
TYPE=0
while [ $TYPE -le 255 ] ; do
    nemesis icmp -S 192.168.1.1 -D 10.10.10.1 -i $TYPE -c 0
    TYPE=`expr $TYPE + 1 `
done
```

2.3.4 Wireshark

Wireshark is een sniffer voor Linux en Windows systemen, die een gebruiksvriendelijkere interface heeft dan tcpdump.



3 Woordverklaring

When all other means of communication fail, try words.

Authentificatie (of authenticatie):
Proces dat de identiteit vaststelt van een gebruiker die toegang wil krijgen tot een systeem

Autorisatie:
Proces dat na de authenticatie (zie hiervoor) de toegangsrechten tot een applicatie nakijkt.

Bastion Host:
Een systeem dat speciaal versterkt is om aanvallen te weerstaan. Wordt op een netwerk geïnstalleerd op de meest potentieel gevaarlijke plaats. Bastion hosts zijn dikwijls onderdelen van een firewall.

Cryptografische Controlesom:
Een eenrichtingsfunctie die wordt toegepast op een bestand om een unieke "fingerprint" er van te maken. Later wordt de controlesom vergeleken om te zien of niemand het systeem probeerde te wijzigen. Meestal toegepast op Unix machines (CRC, MD5 controlesommen)

Daemon:
Een systeemproces dat op de achtergrond van een systeem draait. De naam eindigt op een "d".

Firewall:
Een systeem of een combinatie die regels oplegt tussen 2 of meer netwerken.

Insider Attack:
Een aanval die zijn oorsprong vindt in een beschermd netwerk.

IP Spoofing:
Een aanval waarbij een systeem onrechtmatig de identiteit aanneemt van een ander systeem door het overnemen van het IP adres.

IP Splicing / Hijacking:
Een aanval waarbij een actieve, vastgelegde sessie wordt onderschept en geparasiteerd wordt door de aanvaller. IP Splicing kan voorkomen nadat er authentificatie heeft plaatsgevonden. De aanvaller neemt dan de rol over van een andere ingelogde gebruiker. Bescherming hiertegen kan door encryptie in de sessie -of netwerklaag.

Loggen:
Het vastleggen van informatie over gebeurtenissen die plaatsgrijpen in een firewall of netwerk.

Log Retention:
Hoe lang logs van audit programma's worden bijgehouden.

Netwerkniveau Firewall:
Een firewall die het verkeer (de pakketten) enkel op netwerkniveau nakijkt.

Policy of Beleid:
Regels van een organisatie, die handelen over het aanvaardbaar gebuik van computer bronnen, veiligheid en bedieningsprocedures.

Proxy:
Een software hulp die voor een gebruiker werkt. Een typische proxy zal een verbinding van een gebruiker toestaan, eventueel extra authentificatie doen en een verbinding aangaan met een afgelegen host computer.

Session Stealing:
Zie IP Splicing.

Trojaans Paard:
Een software entiteit die er uit ziet alsof het iets ‘normaal’ doet, maar die eigenlijk een gat maakt in de veiligheid of een aanvalsprogramma bevat.

Tunneling Router:
Een router of een systeem dat het verkeer geëncrypteerd over een niet betrouwbaar netwerk kan zenden. Daarna kunnen de gegevens terug ontcijferd worden.

Social Engineering:
Een aanval gebaseerd op het misleiden van gebruikers of administrators. Typisch gebeurt dit door contact via de telefoon.

Toegangslijsten:
Regels voor pakketfilters (meestal routers) die stellen welke pakketten door mogen of geblokkeerd moeten worden.

Toepassingsniveau Firewall:
Een firewall systeem waarbij diensten worden voorzien die een volledige TCP sessie kunnen verzorgen. Toepassingsniveau firewalls zullen dikwijls het verkeer her-adresseren zodat het lijkt alsof uitgaand verkeer van de firewall komt in plaats van de interne host

Virus:
Zichzelf verdubbelende code die zich kan vasthechten aan een programma of data file.Virussen kunnen (of kunnen geen) aanvalsprogramma’s of trojaanse paarden bevatten.

Worm:
Een onafhankelijk programma, dat zichzelf copieert bij uitvoering naar een andere host. Het bekende “Internet Virus" van 1988 was geen virus maar een worm.

4 Geraadpleegde werken

4.1 Boeken

DOWD, K., Getting connected, The Internet at 56K and Up,
O'Reilly & Associates Inc., 1998, 400 p.
KLANDER, L., Hacker proof,
Sybex, 1998, 898 p.
BOWDEN, T. en BAUER B., The SuSE firewall and masquerading setup version 2.0,
SuSE, 1999, 21 p.
HELD G., Understanding data communications, 6th ed.,
New Riders, 1999, 619 p.

4.2 Internet

BELNET CERT
http://cert.belnet.be/,
geraadpleegd op 20 september 2007

CONCEPT OF STATE
http://www.itwizard.info/technology/linux/statefull_firewall/Concept_of_State.htm
geraadpleegd op 25 september 2007

FILE TRANSFER PROTOCOL (FTP)
http://www.ietf.org/rfc/rfc0959.txt
geraadpleegd op 29 september 2008

FIREWALL-FRIENDLY FTP
http://www.ietf.org/rfc/rfc1579.txt
geraadpleegd op 29 september 2008

IPTABLES HOWTO
http://www.linuxguruz.com/iptables/howto/
geraadpleegd op 25 september 2007

NT SECURITY
http://ntsecurity.nu/toolbox,
geraadpleegd op 25 september 2007

STATEFUL FILTERING AND STATEFUL INSPECTION
http://www.informit.com/articles/article.asp?p=373431&seqNum=3, geraadpleegd op 25 september 2007

SUN, FIREWALL AND PROXY SERVER HOWTO
http://sunsite.unc.edu/LDP/HOWTO/Firewall-HOWTO.html,
geraadpleegd op 20 januari 1999

THE ORANGE BOOK, http://nsi.org/Computer/govt.html,
geraadpleegd op 2 september 1999