Fedora 12

Beveiligings gids

Een gids voor het beveiligen van Fedora Linux

Johnray Fuller

John Ha

David O'Brien

Scott Radvan

Eric Christensen

Fedora 12 Beveiligings gids
Een gids voor het beveiligen van Fedora Linux
Uitgave 1.1

Auteur	Johnray Fuller	jrfuller@redhat.com
Auteur	John Ha	jha@redhat.com
Auteur	David O'Brien	daobrien@redhat.com
Auteur	Scott Radvan	sradvan@redhat.com
Auteur	Eric Christensen	sparks@fedoraproject.org

The Linux Security Guide is designed to assist users of Linux in learning the processes and practices of securing workstations and servers against local and remote intrusion, exploitation, and malicious activity.Focused on Fedora Linux but detailing concepts and techniques valid for all Linux systems, The Linux Security Guide details the planning and the tools involved in creating a secured computing environment for the data center, workplace, and home.With proper administrative knowledge, vigilance, and tools, systems running Linux can be both fully functional and secured from most common intrusion and exploit methods.

Voorwoord

1. Document Conventie

Dit handboek hanteert verscheidene conventies om bepaalde woorden of zinsdelen te benadrukken en aandacht te vestigen op specifieke delen van informatie.

In PDF en papieren edities gebruikt dit handboek *Liberation Fonts set*[1] lettertypen. Het Liberation lettertype wordt ook gebruikt in HTML-edities indien dit lettertype op uw computer geïnstalleerd is. Indien dat niet het geval is, worden alternatieve, gelijkwaardige lettertypen gebruikt. Noot: bij Red Hat Enterprise Linux 5 en later wordt de Liberation Font set standaard meegeleverd.

1.1. Typografische Conventies

Vier typografische conventies worden gebruikt om aandacht te vestigen op specifieke woorden en zinsdelen. Deze conventies -en de omstandigheden waaronder zij gebruikt worden- luiden als volgt:

`Mono-spaced Bold`

Wordt gebruikt om systeem input, waaronder shell commando's, bestandsnamen en paden aan te geven. Wordt ook gebruikt bij toetsaanduiding of toetsencombinaties. Voorbeeld:

> To see the contents of the file **`my_next_bestselling_novel`** in your current working directory, enter the **`cat my_next_bestselling_novel`** command at the shell prompt and press **`Enter`** to execute the command.

The above includes a file name, a shell command and a key cap, all presented in Mono-spaced Bold and all distinguishable thanks to context.

Key-combinations can be distinguished from key caps by the hyphen connecting each part of a key-combination. For example:

> Press **`Enter`** to execute the command.
>
> Press **`Ctrl`**+**`Alt`**+**`F1`** to switch to the first virtual terminal. Press **`Ctrl`**+**`Alt`**+**`F7`** to return to your X-Windows session.

The first sentence highlights the particular key cap to press. The second highlights two sets of three key caps, each set pressed simultaneously.

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in **`Mono-spaced Bold`**. For example:

> File-related classes include **`filesystem`** for file systems, **`file`** for files, and **`dir`** for directories. Each class has its own associated set of permissions.

Proportional Bold

This denotes words or phrases encountered on a system, including application names; dialogue box text; labelled buttons; check-box and radio button labels; menu titles and sub-menu titles. For example:

[1] https://fedorahosted.org/liberation-fonts/

Choose **System > Preferences > Mouse** from the main menu bar to launch **Mouse Preferences**. In the **Buttons** tab, click the **Left-handed mouse** check box and click **Close** to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).

To insert a special character into a **gedit** file, choose **Applications > Accessories > Character Map** from the main menu bar. Next, choose **Search > Find…** from the **Character Map** menu bar, type the name of the character in the **Search** field and click **Next**. The character you sought will be highlighted in the **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then click the **Copy** button. Now switch back to your document and choose **Edit > Paste** from the **gedit** menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in Proportional Bold and all distinguishable by context.

Note the **>** shorthand used to indicate traversal through a menu and its sub-menus. This is to avoid the difficult-to-follow 'Select **Mouse** from the **Preferences** sub-menu in the **System** menu of the main menu bar' approach.

Mono-spaced Bold Italic or ***Proportional Bold Italic***

Whether Mono-spaced Bold or Proportional Bold, the addition of Italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

To connect to a remote machine using ssh, type **ssh *username*@*domain.name*** at a shell prompt. If the remote machine is **example.com** and your username on that machine is john, type **ssh john@example.com**.

The **mount -o remount *file-system*** command remounts the named file system. For example, to remount the **/home** file system, the command is **mount -o remount /home**.

To see the version of a currently installed package, use the **rpm -q *package*** command. It will return a result as follows: ***package-version-release***.

Note the words in bold italics above — username, domain.name, file-system, package, version and release. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

When the Apache HTTP Server accepts requests, it dispatches child processes or threads to handle them. This group of child processes or threads is known as a *server-pool*. Under Apache HTTP Server 2.0, the responsibility for creating and maintaining these server-pools has been abstracted to a group of modules called *Multi-Processing Modules* (*MPMs*). Unlike other modules, only one module from the MPM group can be loaded by the Apache HTTP Server.

1.2. Pull-quote Conventies

Twee, normaal gesproken uit meerdere regels bestaande, datatypes worden visueel van de omringende tekst gescheiden.

Tekst gezonden naar een terminal wordt getoond in `Mono-spaced Roman` en als volgt gepresenteerd:

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

Opsommingen van broncode worden ook vertoond in `Mono-spaced Roman` maar worden alsvolgt gepresenteerd en benadrukt:

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;

public class ExClient
{
   public static void main(String args[])
       throws Exception
   {
      InitialContext iniCtx = new InitialContext();
      Object         ref    = iniCtx.lookup("EchoBean");
      EchoHome       home   = (EchoHome) ref;
      Echo           echo   = home.create();

      System.out.println("Created Echo");

      System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
   }

}
```

1.3. Noten en waarschuwingen

Tenslotte gebruiken we drie visuele stijlen om aandacht te vestigen op informatie die anders misschien over het hoofd zou worden gezien.

Noot

Een noot is een tip of handigheidje of een alternatieve benadering voor de taak die uitgevoerd gaat worden. Het negeren van een noot zou geen ernstige gevolgen moeten hebben, maar het leven kan een stuk makkelijker worden indien de noot gevolgd wordt.

Belangrijk

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring Important boxes won't cause data loss but may cause irritation and frustration.

Waarschuwing

Een waarschuwing dient niet genegeerd te worden. Waarschuwingen negeren zal ongetwijfeld leiden tot data- en haarverlies.

2. We hebben terugkoppeling nodig!

Meer informatie over het Linux Security Guide project kan gevonden worden op *https://fedorahosted.org/securityguide*

Om terugkoppeling te geven over de Beveiligings gids, dien je een foutrapport in op *https://bugzilla.redhat.com/enter_bug.cgi?component=security-guide&product=Fedora%20Documentation*. Selecteer het juiste onderdeel in het uitklap menu.

Beveiligings overzicht

Door het toenemende vertrouwen op krachtige computers in een netwerk om te helpen bedrijven draaiende te houden en contact te houden met onze persoonlijke informatie, zijn complete bedrijven ontstaan rond de praktijk van netwerk en computer beveiliging. Ondernemingen hebben de kennis en vaardigheden van beveiligings deskundigen ingeroepen om systemen correct te controleren en om aangepaste oplossingen te maken voor de werk vereisten van de organisatie. Omdat de meeste organisaties in toenemende mate dynamisch van natuur zijn, met werknemers die de IT hulpbronnen van de onderneming lokaal en op afstand benaderen, is de behoefte aan beveiligde computeromgevingen sterk toegenomen.

Unfortunately, most organizations (as well as individual users) regard security as an afterthought, a process that is overlooked in favor of increased power, productivity, and budgetary concerns. Proper security implementation is often enacted postmortem — *after* an unauthorized intrusion has already occurred. Security experts agree that taking the correct measures prior to connecting a site to an untrusted network, such as the Internet, is an effective means of thwarting most attempts at intrusion.

1.1. Inleiding tot beveiliging

1.1.1. Wat is computer beveiliging

Computer beveiliging is een algemene term die een breed gebied van computer en informatie verwerking afdekt. Ondernemingen die afhangen van computer systemen en netwerken om hun dagelijkse zakentransacties uit te voeren en toegang hebben tot cruciale informatie, zien hun data als een belangrijk onderdeel van hun totale bezit. Verscheidene uitdrukkingen en metrieken zijn binnengedrongen in onze zakelijke woordenschat, zoals totale kosten van eigendom (total cost of ownership - TCO), en kwaliteit van diensten (quality of service - QoS). Met gebruik van deze metrieken kunnen ondernemingen aspecten bereken zoals data integriteit en hoge-beschikbaarheid als onderdeel van hun planning en procesbeheerskosten. In sommige industrieën, zoals elektronische handel, kan het de beschikbaarheid en betrouwbaarheid van data het verschil zijn tussen succes en mislukken.

1.1.1.1. Hoe is computer beveiliging ontstaan?

Informatie beveiliging is door de jaren heen gegroeid door het toenemende vertrouwen van publieke netwerken om hun persoonlijke, financielen en andere vertrouwelijke informatie niet te openbaren. Er zijn talrijke voorbeelden zoals de Mitnick [1]en de Vladimir Levin [2] zaken die organisaties in alle industrieën hebben aangezet om de manier waarop ze informatie behandelen te overdenken, maar ook de overbrenging en onthulling van data. De populariteit van het Internet was een van de belangrijkste ontwikkelingen die aanzette tot een toenemende inspanning in data beveiliging.

Een steeds groeiend aantal mensen gebruiken hun persoonlijke computer om toegang te krijgen tot hulpbronnen die het Internet heeft te bieden. Van onderzoeken en vinden van informatie tot elektronische mail en commerciële transacties, wordt het Internet gezien als een van de belangrijkste ontwikkelingen van de 20ste eeuw.

Het Internet en zijn eerdere protocollen zijn echter ontwikkeld als een op *vertrouwen-gebaseerd* systeem. Dat betekent dat het Internet Protocol niet ontworpen is om zelf veilig te zijn. Er zijn geen

[1] http://law.jrank.org/pages/3791/Kevin-Mitnick-Case-1999.html
[2] http://www.livinginternet.com/i/ia_hackers_levin.htm

goedgekeurde beveiligings standaarden ingebouwd in de TCP/IP communicatie stack, waardoor het open is voor kwaadwillige gebruikers en processen over het netwerk. Moderne ontwikkelingen hebben het Internet veiliger gemaakt, maar er zijn nog steeds verscheidene incidenten die landelijke aandacht krijgen en ons op het feit attenderen dat niets geheel veilig is.

1.1.1.2. Beveiliging in deze tijd

In februari 2000 werd een Distributed Denial of Service (DDoS) aanval uitgevoerd op verscheidene van de meest gebruikte sites op het Internet. De aanval maakte yahoo.com, cnn.com, amazon.com, fbi.gov, en verscheidene andere sites volkomen onbereikbaar voor gewone gebruikers, omdat het de routers urenlang bezig hield met ICMP pakket verkeer, ook wel ping flood genoemd. De aanval werd uitgevoerd door onbekende aanvallers met gebruik van speciaal gemaakte, breed verkrijgbare programma's die kwetsbare netwerkservers opzochten en daarop cliënt toepassingen installeerden (trojans genaamd), waarna de aanvallen van alle geïnfecteerde servers de slachtoffer sites overspoelden en ze onbereikbaar maakten. Veel mensen geven de schuld aan fundamentele problemen in de manier waarop routers en de gebruikte protocollen zijn ingesteld om alle binnenkomende data te accepteren, onafhankelijk waar heen of voor welk doel de pakketten verzonden worden.

In 2007 resulteerde een data inbreuk, die een algemeen bekende zwakte van het Wired Equivalent Privacy (WEP) draadloze versleutelings protocol misbruikte, in de diefstal van meer dan 45 miljoen creditkaart nummers van een globaal werkende financiële instelling.[3]

In een ander voorval werden de rekening gegevens van meer dan 2.2 miljoen patiënten, die bewaard werden op een backup tape, gestolen van de voor zitting uit de auto van een koerier.[4]

Er wordt geschat dat op dit moment 1.4 miljard mensen over de gehele wereld het Internet gebruiken of hebben gebruikt.[5] Te gelijkertijd:

- Elke dag worden er ongeveer 225 belangrijke incidenten met betrekking tot beveilingsinbreuken gerapporteerd aan de CERT Coordination Center in Carnegie Mellon.[6]
- In 2003 ging het aantal aan CERT gerapporteerde incidenten naar 137.529 vergeleken met 82.094 in 2002 en 52.658 in 2001.[7]
- De wereldwijde economische impact van de drie gevaarlijkste Internet virussen van de laatste drie jaar wordt geschat op 13.2 miljard $.[8]

Een aantal punten uit een globaal onderzoek van zakelijke en technologische leidinggevers "The Global State of Information Security"[9], uitgevoerd door *CIO Magazine*, zijn:

- Slechts 43% van de ondervraagden bewaken de naleving van beveiligingsvoorschriften door gebruikers
- Slechts 22% houdt bij welke externe bedrijven hun data gebruiken
- De oorzaak van bijna de helft van de beveiligings incidenten is omschreven als "Onbekend"
- 44% van de ondervraagden is van plan de beveiligingsuitgaven in het volgend jaar te verhogen
- 59% hebben en informatie beveiligings strategie.

[3] http://www.theregister.co.uk/2007/05/04/txj_nonfeasance/
[4] http://www.healthcareitnews.com/story.cms?id=9408
[5] http://www.internetworldstats.com/stats.htm
[9] http://www.csoonline.com/article/454939/The_Global_State_of_Information_Security_

Deze resultaten onderstrepen de werkelijkheid dat computerbeveiliging een meetbaar en verdedigbaar onderdeel is van IT budgetten. Organisaties die data integriteit en hoge beschikbaarheid vereisen, gebruiken de vaardigheden van beheerders, ontwikkelaars, en ingenieurs om zeker te zijn van ononderbroken betrouwbaarheid van hun systemen, diensten, en informatie. Het slachtoffer worden van kwaadwillige gebruikers, processen, of gecoördineerde aanvallen is een directe bedreiging van het succes van de organisatie.

Helaas is systeem en netwerk beveiliging een moeilijk vak, wat een uitgebreide kennis vereist van de manier waarop een organisatie zijn informatie beschouwt, gebruikt, manipuleert en overbrengt. Het begrijpen van de manier waarop de organisatie (en de mensen die de organisatie vormen) zaken doet is van groot belang voor het maken van een juist beveiligingsplan.

1.1.1.3. Het standaardiseren van beveiliging

Bedrijven in elke industrie tak vertrouwen op voorschriften en regels die gemaakt zijn door standaardisatie organisaties zoals de American Medical Association (AMA) of de Institute of Electrical and Electronics Engineers (IEEE). Hetzelfde ideaal geldt voor informatie beveiliging. Veel beveiligings consulenten en bedrijven omarmen het standaard beveiligings model bekend als CIA, of *Confidentiality, Integrity, and Availability* (Vertrouwelijkheid, Integriteit, en Beschikbaarheid). Dit drie-lagen model is een algemeen geaccepteerd onderdeel om de risico's van gevoelige informatie te onderzoeken en een beveiligings tactiek uit te stippelen.. Het volgende beschrijft het CIA model in meer detail:

- Confidentiality (Vertrouwelijkheid) — Gevoelige informatie moet alleen beschikbaar zijn voor een beperkt, gedefinieerd aantal mensen. Onbevoegde overdracht en gebruik van informatie moet beperkt worden. Bijvoorbeeld, vertrouwelijkheid van informatie verzekert dat de persoonlijke en financiële informatie van een klant niet verkregen kan worden door een onbevoegd persoon voor kwaadwillige doeleinden zoals identiteit diefstal of creditkaart fraude.

- Integrity (Integriteit) — Informatie moet niet veranderd worden op een manier waardoor het incompleet of incorrect wordt. Onbevoegde gebruikers moeten beperkt worden in de mogelijkheden om gevoelige informatie te veranderen of te vernietigen.

- Availability (Beschikbaarheid) — Informatie moet toegankelijk zijn voor bevoegde gebruikers op ieder moment dat dit nodig is. Beschikbaarheid is een garantie dat informatie verkregen kan worden met een afgesproken frequentie en snelheid. Dit wordt vaak gemeten in percentages en wordt formeel afgesproken in Service Level Agreements (SLAs) (Service Niveau Overeenkomsten) gebruikt door netwerk dienst aanbieders en hun zakelijke klanten.

1.1.2. SELinux

Fedora includes an enhancement to the Linux kernel called SELinux, which implements a Mandatory Access Control (MAC) architecture that provides a fine-grained level of control over files, processes, users and applications in the system. Detailed discussion of SELinux is beyond the scope of this document; however, for more information on SELinux and its use in Fedora, refer to the Fedora SELinux User Guide available at *http://docs.fedoraproject.org/selinux-user-guide/*. For more information on configuring and running services in Fedora that are protected by SELinux, refer to the SELinux Managing Confined Services Guide available at *http://docs.fedoraproject.org/selinux-managing-confined-services-guide*[10]. Other available resources for SELinux are listed in *Hoofdstuk 7, Referenties*.

[10] http://docs.fedoraproject.org/selinux-managing-confined-services-guide/

1.1.3. Beveiligings controles

computer beveiliging wordt vaak verdeeld in drie aparte hoofdgroepen, gewoonlijk aangegeven als *controles*:

- Fysiek
- Technisch
- Administratief

Deze drie brede categorieën definiëren de belangrijkste doelen van een juiste beveiligings implementatie. Binnen deze controles zijn sub-categorieën die de controles verder verfijnen en de implementatie aangeven.

1.1.3.1. Fysieke controles

Fysieke controles is de implementatie van beveiligings maatregelen in een gedefinieerde structuur voor het afschrikken of verhinderen van onbevoegde toegang tot gevoelig materiaal. Voorbeelden van fysieke controles zijn:

- Gesloten-circuit bewakings camera's
- Bewegings of thermische alarm systemen
- Bewakers
- Badges met foto
- Afgesloten en vergrendelde stalen deuren
- Biometriek (zoals vingerafdrukken, stem, gezichts, iris, handschrift, en andere automatische methoden gebruikt om individuen te herkennen)

1.1.3.2. Technische controles

Technische controles gebruiken technologie als basis voor het controleren van de toegang tot en het gebruik van gevoelige data door een fysieke structuur en via een netwerk. Technische controles reiken ver en bevatten technologieën als:

- Versleuteling
- Smart cards
- Netwerk authenticatie
- Toegangs controle lijsten (ACL's)
- Bestandsintegriteit onderzoek software

1.1.3.3. Administratieve controles

Administratieve controles definiëren de menselijke factoren van beveiliging. Ze omvatten alle niveaus van personeel binnen een organisatie en bepalen welke gebruikers toegang hebben tot welke hulpbronnen en informatie met behulp van middelen als:

- Onderwijs en bewustzijn

- Voorbereid zijn op ongevallen en herstel plannen
- Personeel aanwerven en scheiden strategieën
- Personeelsregistratie en verantwoording

1.1.4. Conclusie

Nu je iets hebt geleerd over de oorsprong, redenen, en aspecten van beveiliging, zul je het gemakkelijker vinden om de juiste actie koers te bepalen met betrekking tot Fedora. Het is belangrijk om te weten welke factoren en condities beveiliging bepalen om een juiste strategie te bedenken en te implementeren. Met deze informatie in gedachte, kan het proces geformaliseerd worden en wordt het pad duidelijker als je dieper in de specifieke aspecten van het beveiligings proces duikt.

1.2. Kwetsbaarheid beoordeling

Met voldoende tijd, hulpbronnen, en motivatie, kan een cracker in bijna elk systeem inbreken. Uiteindelijk kunnen alle beveiligings procedures en technologiën die nu beschikbaar zijn, niet garanderen dat elk systeem volledig beveiligd is tegen indringing. Routers helpen de gateways naar het Internet te beveiligen. Firewalls helpen om de randen van het netwerk te beveiligen. Virtual Private Networks geven data veilig door met een versleutelde stroom. Indringings detectie systemen waarschuwen je voor kwaadwillige activiteiten. Het succes van al deze technologiën hangt echter af van een aantal variabelen, zoals:

- De deskundigheid van de werknemers die verantwoordelijk zijn voor het instellen, bewaken, en onderhouden van de technologiën.
- De mogelijkheid om services en kernels snel en efficiënt te corrigeren en te vernieuwen.
- De mogelijkheid van de verantwoordelijken om constant waakzaam te zijn over het netwerk.

Door de dynamische toestand van data systemen en technologiën, kan het beveiligen van zakelijke hulpbronnen behoorlijk ingewikkeld zijn. Door deze complexiteit is het vaak moeilijk om deskundigen te vinden voor al je systemen. Terwijl het mogelijk is om personeel te hebben die kennis heeft van vele gebieden van informatie beveiliging, is het moeilijk personeel vast te houden die deskundig is in meer dan een paar onderwerpsgebieden. Dit komt voornamelijk door de vereiste van constante aandacht en scherpte voor ieder onderwerpsgebied van informatie beveiliging. Informatie beveiliging staat niet stil.

1.2.1. Denk als de vijand

Veronderstel dat je een zakelijk netwerk beheert. Zo'n netwerk bestaat gewoonlijk uit operating systemen, toepassingen, servers, netwerk bewaking, firewalls, indringings detectie systemen, en meer. Stel je nu voor om te proberen voor ieder van deze actueel te blijven. Gegeven de complexiteit van de huidige software en netwerk omgevingen, zijn uitbuitingen en bugs een zekerheid. Actueel te blijven met correcties en vernieuwingen voor een geheel netwerk zal een intimiderende taak blijken te zijn in een grote organisatie met heterogene systemen.

Combineer de kennis vereisten met de taak om actueel te blijven, en het is duidelijk dat vijandige incidenten zullen optreden, dat systemen verstoord worden, data corrupt raakt, en service onderbroken wordt.

Om de beveiligings technologiën te ondersteunen en te helpen met het beschermen van systemen, netwerken, en data, moet je denken als een cracker en de beveiliging van je systemen meten door

het zoeken naar zwaktes. Preventief kwestbaarheids onderzoek van je eigen systemen en netwerk hulpbronnen kan potentiële problemen laten zien voordat een cracker deze uit kan buiten.

Een kwetsbaarheidsonderzoek is een intern onderzoek van je netwerk en systeem beveiliging; waarvan de resultaten de vertrouwelijkheid, integriteit, en beschikbaarheid van je systeem aangeven (zoals uitgelegd in *Paragraaf 1.1.1.3, “Het standaardiseren van beveiliging”*). Gewoonlijk begint een kwetsbaarheidsonderzoek met een verkennings fase, waarin belangrijke gegevens over de doel systemen en hulpbronnen worden verzameld. Deze fase leidt naar de systeem gereedheids fase, waarin het doel voornamelijk bekeken wordt voor alle bekende kwetsbaarheden. De gereedheids fase bereikt zijn hoogtepunt in de rapporteer fase, waarin de bevindingen ingedeeld worden in categorieën van hoge, gemiddelde, en lage risico's; en methodes voor het verbeteren van de beveiliging (of het verlichten van het risico van kwetsbaarheid) van het doel worden besproken.

Als je een kwetsbaarheidsonderzoek van je huis zou uitvoeren, zal je waarschijnlijk willen controleren of elke deur naar je huis dicht en afgesloten is. Je zult ook elk venster controleren om er zeker van te zijn dat deze geheel dicht zijn en correct afgesloten. Dit zelfde concept is van toepassing op systemen, netwerken, en elektronische data. Kwaadwillige gebruikers zijn de dieven en vandalen van je data. Richt je op hun gereedschappen, mentaliteit, en motivatie, en je kunt dan snel reageren op hun acties.

1.2.2. Het definiëren van onderzoeken en testen

Kwetbaarheidsonderzoeken kunnen verdeeld worden in twee types:*van buiten naar binnen kijken* en *van binnen rondkijken*.

When performing an outside looking in vulnerability assessment, you are attempting to compromise your systems from the outside. Being external to your company provides you with the cracker's viewpoint. You see what a cracker sees — publicly-routable IP addresses, systems on your *DMZ*, external interfaces of your firewall, and more. DMZ stands for "demilitarized zone", which corresponds to a computer or small subnetwork that sits between a trusted internal network, such as a corporate private LAN, and an untrusted external network, such as the public Internet. Typically, the DMZ contains devices accessible to Internet traffic, such as Web (HTTP) servers, FTP servers, SMTP (e-mail) servers and DNS servers.

Als je een van binnen rondkijken kwetsbaarheidsonderzoek uitvoert, heb je het voordeel dat je intern bent en je status is verhoogd naar vertrouwd. Dit is het gezichtspunt dat jij en je collega's hebben zodra je ingelogd bent op je systeem. Je ziet printservers, bestandsservers, databases en andere hulpbronnen.

Er is een opvallend verschil tussen de twee types van kwetsbaarheidsonderzoeken. Als je intern bij je bedrijf bent heb je meer rechten dan een buitenstaander. Vandaag de dag wordt beveiliging in de meeste organisaties nog steeds ingesteld om indringers buiten te houden. Erg weinig wordt gedaan om het binnenste van de organisatie te beveiligen (zoals afdelingsfirewalls, toegangscontrole op gebruikers niveau, authenticatie procedures voor interne hulpbronnen, enzovoort). Gewoonlijk zijn er veel meer hulpbronnen als je binnen rondkijkt omdat de meeste systemen intern het bedrijf zijn. Zodra je jezelf buiten het bedrijf plaatst, krijg je onmiddellijk een onvertrouwde status. De systemen en hulpbronnen die voor je beschikbaar zijn als je extern bent is gewoonlijk erg beperkt.

Overweeg het verschil tussen kwetsbaarheidsonderzoek en *indringings testen*. Beschouw een kwetsbaarheidsonderzoek als de eerste stap van een indringingstest. De informatie verkregen van het onderzoek wordt gebruikt voor het testen. Terwijl het onderzoek wordt uitgevoerd om gaten en potentiële kwestbaarheden te onderzoeken, probeert de indringingstest de resultaten echt te gebruiken.

Het onderzoeken van de netwerk infrastructuur is een dynamisch proces. Beveiliging, zowel informatie als fysiek, is dynamisch. Het uitvoeren van een onderzoek geeft een overzicht die verkeerde plussen en verkeerde minnen kan opleveren.

Beveiligingsbeheerders zijn niet beter dan de gereedschappen die ze gebruiken en de kennis die ze hebben. Neem een van de op dit moment beschikbare beveiligingsgereedschappen, en het is zo goed als zeker dat er een paar verkeerde plussen zijn. Of dit komt door een programma fout of een gebruikers fout, het resultaat is hetzelfde. Het gereedschap kan kwestbaarheden vinden die in werkelijkheid niet bestaan (verkeerde plussen); of, nog erger, het gereedschap kan kwetsbaarheden niet vinden die werkelijk bestaan (verkeerde minnen).

Nu het verschil tussen een kwetsbaarheidsonderzoek en een indringingstest is gedefinieerd, nemen we de resultaten van het onderzoek en bekijken we deze zorgvuldig voordat we een indringingstest uitvoeren als onderdeel van je nieuwe beste praktijken aanpak.

Waarschuwing

Het proberen van het uitbuiten van kwetsbaarheden op productie hulpbronnen kan een averechts effect hebben op de productiviteit en efficiëntie van je systemen en netwerk.

De volgende lijst bekijkt een aantal voordelen van het uitvoeren van kwetsbaarheidsonderzoeken.

- Veroorzaakt een pro-actieve focus op informatie beveiliging
- Vindt potentiële uitbuitingen voordat crackers deze vinden
- Resulteert in systemen die bij de tijd gehouden worden en gecorrigeerd worden.
- Bevordert groei en helpt in het ontwikkelen van vaardigheden van het personeel
- Vermindert financiële verliezen en negatieve publiciteit

1.2.2.1. Het vaststellen van een methodologie

Om te helpen bij het kiezen van gereedschappen voor een kwetsbaarheidsonderzoek, is het nuttig om een kwetsbaarheidsonderzoek methodologie vast te stellen. Helaas is er op dit moment geen voor-gedefinieerde of door de industrie goedgekeurde methodologie; echter gezond verstand en beste praktijken kunnen als voldoende gids dienen.

Wat is het doel? Kijken we naar een server, of kijken we naar ons gehele netwerk en alles binnen het netwerk? Zijn we extern of intern van het bedrijf? De antwoorden op deze vragen zijn belangrijk omdat ze niet alleen helpen te bepalen welke gereedschappen we moeten kiezen, maar ook de manier waarop ze gebruikt moeten worden.

Om meer te weten te komen over het vaststellen van een methodologie, refereer je naar de volgende websites:

- *http://www.isecom.org/osstmm/ The Open Source Security Testing Methodology Manual* (OSSTMM)
- *http://www.owasp.org/ The Open Web Application Security Project*

1.2.3. Het evalueren van de gereedschappen

Een onderzoek kan beginnen met het gebruik van een informatie verzamel gereedschap. Als het gehele netwerk onderzocht wordt, maak dan eerst een plattegrond om de host te vinden die er in draaien. Zodra die gevonden zijn, bekijk dan iedere host individueel. Concentreren op deze hosts vereist een andere set gereedschappen. Te weten welke gereedschappen gebruikt moeten worden kan de beslissende stap zijn in het vinden van kwetsbaarheden.

Net als in elk onderdeel van het dagelijkse leven, zijn er veel verschillende gereedschappen die dezelfde taak uitvoeren. Dit concept is ook van toepassing op het uitvoeren van kwetbaarheidsonderzoeken. Er zijn gereedschappen specifiek voor operating systemen, toepassingen, en zelfs netwerken (afhankelijk van het gebruikte protocol). Sommige gereedschappen zijn gratis, andere niet. Sommige gereedschappen zijn intuïtief en gemakkelijk te gebruiken, terwijl andere cryptisch en slecht gedocumenteerd zijn maar met eigenschappen die andere niet hebben.

Het vinden van de juiste gereedschappen kan een intimiderende taak zijn en op het eind telt ervaring. Indien mogelijk maak je een test omgeving en probeer je zoveel gereedschappen als je kunt, en je noteert van elk de sterktes en zwaktes. Bekijk het README bestand of de manual pagina voor het gereedschap. Kijk bovendien op het Internet voor meer informatie, zoals artikelen, stap-voor-stap gidsen, en zelfs mailing lijsten specifiek voor een gereedschap.

De hier beneden besproken gereedschappen is slechts een klein aantal van de beschikbare gereedschappen.

1.2.3.1. Hosts scannen met Nmap

Nmap is een populair gereedschap dat onderdeel is van Fedora en gebruikt kan worden voor het bepalen van de opbouw van een netwerk. Nmap is al vele jaren beschikbaar en is waarschijnlijk het meest gebruikte gereedschap voor het verzamelen van informatie. Een uitstekende manual pagina is toegevoegd die een gedetaileerde beschrijving van zijn opties en gebruik geeft. Beheerders kunnen Nmap in een netwerk gebruiken om de host systemen te vinden en open poorten op die systemen.

Nmap is een goede eerste stap in een kwetsbaarheidsonderzoek. Je kunt alle hosts in je netwerk in kaart brengen en zelfs een optie meegeven dat Nmap de mogelijkheid geeft om het operatings systeem te bepalen wat op een bepaalde host draait. Nmap is een goede ondergrond voor het vaststellen van een tactiek voor het gebruiken van beveiligde services en het stoppen van ongebruikte services.

1.2.3.1.1. Nmap gebruiken

Nmap kan uitgevoerd worden vanaf een shell prompt door het intypen van het **nmap** commando gevolgd door de hostnaam of IP adres van de machine die bekeken moet worden.

```
nmap foo.example.com
```

The results of a basic scan (which could take up to a few minutes, depending on where the host is located and other network conditions) should look similar to the following:

```
Starting Nmap 4.68 ( http://nmap.org )
Interesting ports on foo.example.com:
Not shown: 1710 filtered ports
PORT    STATE  SERVICE
```

```
22/tcp  open   ssh
53/tcp  open   domain
70/tcp  closed gopher
80/tcp  open   http
113/tcp closed auth
```

Nmap test de meest gebruikte netwerk communicatie poorten voor het luisteren naar of wachten op services. Deze kennis kan nuttig zijn voor een beheerder die onnodige of ongebruikte services wil afsluiten.

Voor meer informatie over het gebruik van Nmap refereer je naar de officiële Nmap pagina op:

http://www.insecure.org/

1.2.3.2. Nessus

Nessus is een beveiligings scanner voor alle services. De plug-in architectuur van Nessus staat gebruikers toe het aan te passen voor hun systemen en netwerken. Zoals met elke scanner is Nessus niet beter dan de handtekeningen database waar het op steunt. Gelukkig wordt Nessus regelmatig vernieuwd en biedt het volledige rapportering, en real-time kwetsbaarheid zoeken. Denk er aan dat er verkeerde plussen en verkeerde minnen kunnen zijn, zelfs bij een gereedschap zo krachtig en regelmatig vernieuwd als Nessus.

Opmerking

The Nessus client and server software is included in Fedora repositories but requires a subscription to use. It has been included in this document as a reference to users who may be interested in using this popular application.

Voor meer informatie over Nessus refereer je naar de officiële web pagina op:

http://www.nessus.org/

1.2.3.3. Nikto

Nikto een zeer goede common gateway interface (CGI) script scanner. Nikto controleert niet alleen voor CGI kwetsbaarheden maar doet dit op een ontwijkende manier zodat het ontsnapt aan indringings detectie systemen. Het heeft een grondige documentatie die je zorgvuldig moet bekijken voordat je het programma draait. Als je Web servers hebt die CGI scripts aanbieden, kan Nikto een uitstekende hulpbron zijn om de beveiliging van deze servers te controleren.

Meer informatie over Nikto kan gevonden worden op:

http://www.cirt.net/code/nikto.shtml

1.2.3.4. VLAD de scanner

VLAD is een kwetsbaarheids scanner, ontwikkelt door het RAZOR team van Bindview, Inc., welke controleert voor de SANS Top Tien lijst van algemene beveiligings problemen (SNMP problemen, bestandsdeling problemen, enz.). Hoewel het niet zo volledig is als Nessus, is VLAD de moeite van het bekijken waard.

Opmerking

VLAD is geen onderdeel van Fedora en wordt niet ondersteund. Het wordt in dit document genoemd als een referentie voor gebruikers die er interesse in hebben om deze populaire toepassing te gebruiken.

Meer informatie over VLAD kan gevonden worden op de RAZOR team website op:

http://www.bindview.com/Support/Razor/Utilities/

1.2.3.5. Anticiperen op je toekomstige behoeftes

Afhankelijk van je doel en hulpmiddelen, zijn er veel gereedschappen beschikbaar. Er zijn gereedschappen voor draadloze netwerken, Novell netwerken, Windows systemen, Linux systemen, een nog veel meer. Een ander essentieel onderdeel van het uitvoeren van onderzoeken kan zijn het bekijken van fysieke beveiliging, personeel doorlichten, of voice/PBX netwerk beoordeling. Nieuwe concepten, zoals *war walking*, wat het scannen van de omtrek van de fysieke structuur van je onderneming inhoudt voor draadloze netwerk kwetsbaarheden, zijn opkomende concepten die je kunt bekijken en, indien nodig, onderdeel maken van je onderzoek. Verbeelding en blootstelling zijn de enigste beperkingen voor het plannen en uitvoeren van kwetsbaarheidsonderzoeken.

1.3. Aanvallers en kwetsbaarheden

Om een goede beveiligings strategie te plannen en te implementeren, moet je bewust zijn van een paar van de problemen die vastberaden en gemotiveerde hackers uitbuiten om systemen in gevaar te brengen. Voordat we echter in detail gaan over deze problemen, moeten we de terminologie definiëren voor het identificeren van een aanvaller.

1.3.1. Een kleine geschiedenis van hackers

De huidige betekenis van de term *hacker* heeft een oorsprong die teruggaat naar 1960-er jaren en de Massachusetts Institute of Technology (MIT) Tech Model Railroad Club,welke treinstellen ontwierp op een grote schaal en met complexe details. Hacker was de naam die gebruikt werd voor leden van de club die een slimme truc of workaround voor een probleem ontdekten.

Sindsdien beschrijft de term hacker iedereen van computer fanaten tot begaafde programmeurs. Een gemeenschappelijke eigenschap van de meeste hackers is de bereidheid om tot in detail te ontdekken hoe computer systemen en netwerken werken met weinig of geen motivatie van buitenaf. Open source software ontwikkelaars beschouwen zichzelf en hun collega's vaak als hackers, en gebruiken het woord als teken van respect.

Gewoonlijk volgen hackers een vorm van de *hacker ethiek* die voorschrijft dat de zoektocht naar informatie en expertise essentieel is, en dat het delen van deze kennis met de gemeenschap een plicht is. Gedurende deze zoektocht naar kennis, genieten sommige hackers van de academische uitdaging van het omzeilen van beveiligings maatregelen in computer systemen. Daarom gebruikt de pers vaak de term hacker voor het beschrijven van mensen die ongeoorloofd toegang krijgen tot systemen en netwerken met gewetenloze, kwaadwillige, of criminele voornemens. De meer nauwkeurige term voor dit type computer hacker is *cracker* — een term die midden 1980-er jaren door hackers is bedacht om het verschil tussen de twee soorten aan te geven.

1.3.1.1. Grijsschalen

Binnen de gemeenschap van individuen die kwetsbaarheden in systemen en netwerken vinden en benutten zijn verschillende aparte groepen. Deze groepen worden vaak beschreven door de kleurschakering van de hoed die ze "dragen" als ze hun beveiligingsonderzoek uitvoeren en deze kleurschakering geeft hun bedoeling aan.

De *witte hoed hacker* is iemand die netwerken en systemen test om hun prestaties te onderzoeken en te bepalen hoe kwetsbaar ze zijn voor indringing. Gewoonlijk kraken witte hoed hackers hun eigen systemen of de systemen van een opdrachtgever die ze specifiek heeft aangenomen om de beveiliging te onderzoeken. Academische onderzoekers en professionele beveiligings consulenten zijn twee voorbeelden van witte hoed hackers.

Een *zwarte hoed hacker* komt overeen met een cracker. In het algemeen zijn crackers minder gericht op programmeren en de academische kant van het inbreken in systemen. Zij steunen vaak op beschikbare crack programma's en buiten bekende zwakheden in systemen uit om gevoelige informatie te ontdekken voor persoonlijke winst of om schade aan te richten op het doel systeem of netwerk.

De *grijze hoed hacker*, daarin tegen, heeft in de meeste situaties de vaardigheden en bedoeling van een witte hoed hacker, maar gebruikt deze kennis soms voor minder nobele doeleinden. Een grijze hoed hacker kan beschouwd worden als een witte hoed hacker die soms een zwarte hoed draagt om zijn doel te bereiken.

Grijze hoed hackers onderschrijven gewoonlijk een andere vorm van de hackers ethiek, die zegt dat het geoorloofd is om in te breken in systemen zolang de hacker geen diefstal pleegt of vertrouwlijke gegevens schendt. Sommigen zullen echter zeggen dat het inbreken in systemen op zich al niet ethisch is.

Onafhankelijk van de bedoeling van de indringer is het belangrijk om de zwaktes te kennen die een cracker waarschijnlijk zal proberen uit te buiten. De rest van dit hoofdstuk concentreert zich op de zaken.

1.3.2. Bedreigingen voor netwerk beveiliging

Slechte praktijken tijdens het instellen van de volgende aspecten van een netwerk kunnen het risico van een aanval vergroten.

1.3.2.1. Onveilige architecturen

Een verkeerd ingesteld netwerk is de eerste ingang voor niet bevoegde gebruikers. Om een op vertrouwen gebaseerd, open lokaal netwerk kwetsbaar te laten voor het in grote mate onveilige Internet is zoiets als de deur op een kier laten in een misdadige omgeving — gedurende een willekeurige tijds periode gebeurt er niets, maar *uiteindelijk* benut iemand de mogelijkheid.

1.3.2.1.1. Broadcast netwerken

System administrators often fail to realize the importance of networking hardware in their security schemes. Simple hardware such as hubs and routers rely on the broadcast or non-switched principle; that is, whenever a node transmits data across the network to a recipient node, the hub or router sends a broadcast of the data packets until the recipient node receives and processes the data. This method is the most vulnerable to address resolution protocol (*ARP*) or media access control (*MAC*) address spoofing by both outside intruders and unauthorized users on local hosts.

1.3.2.1.2. Gecentraliseerde servers

Een andere potentiële netwerk valkuil is het gebruik van gecentraliseerde computers. Een vaak voorkomende kosten besparing in veel bedrijven is om alle services te bundelen op een krachtige machine. Dit is handig omdat het eenvoudiger te beheren is en behoorlijk goedkoper is dan een configuratie met meerdere servers. Een gecentraliseerde server introduceert echter een enkel punt van falen. Als de centrale server in gevaar is gebracht, kan het netwerk compleet nutteloos zijn of erger, vatbaar zijn voor data manipulatie of diefstal. In deze situaties wordt een centrale server een open deur die toegang toestaat tot het gehele netwerk.

1.3.3. Bedreigingen voor server beveiliging

Server beveiliging is even belangrijk als netwerk beveiliging omdat servers vaak een groot deel van de vitale informatie van een organisatie bevatten. Als een server in gevaar is gebracht, komt zijn gehele inhoud misschien beschikbaar voor de cracker die het naar goeddunken kan stelen of manipuleren. De volgende paragrafen bespreken sommige van de hoofdzaken.

1.3.3.1. Ongebruikte services en open poorten

Een volledige installatie van Fedora bevat meer dan 1000 toepassings en bibliotheek pakketten. De meeste serverbeheerders zullen er echter niet voor kiezen om ieder pakket in de distributie te installeren, maar zullen er de voorkeur aangeven om een installatie van basis pakketten uit te voeren, plus een aantal server toepassingen.

Het komt onder systeembeheerders vaak voor om het operating systeem te installeren zonder aandacht te schenken aan welke programma's er in feite geïnstalleerd worden. Dit kan een probleem zijn omdat onnodige services geïnstalleerd kunnen worden, ingesteld met standaard instellingen, en mogelijk aangezet. Dit veroorzaakt dat onnodige services, zoals Telnet, DHCP, of DNS, op een server of werkstation draaien zonder dat de beheerder zich dit realiseert, wat op zijn beurt kan leiden tot ongewenst verkeer naar de server, of zelfs, een potentiele weg in het systeem voor crackers. Refereer naar *Paragraaf 2.2, “Server beveiliging”* voor informatie over het sluiten van poorten en het uitzetten van niet gebruikte services.

1.3.3.2. Niet gecorrigeerde services

De meeste server toepassingen die onderdeel zijn van een standaard installatie zijn solide, goed geteste stukken software. Omdat ze al vele jaren in productie omgevingen gebruikt worden, is hun code grondig verbeterd en veel van de fouten zijn gevonden en gerepareerd.

Er is echter niet zo iets als perfecte software en er is altijd ruimte voor verdere verbeteringen. Bovendien is nieuwere software meestal niet zo grondig getest als je zou verwachten, om dat het pas kort in productie omgevingen aanwezig is of omdat het misschien niet zo populair is als andere server software.

Ontwikkelaars en systeembeheerders vinden vaak fouten in server toepassingen die uit te buiten zijn en publiceren de informatie op bug volgen en beveiligings gerelateerde websites zoals de Bugtraq mailing lijst (*http://www.securityfocus.com*) of de Computer Emergency Response Team (CERT) website (*http://www.cert.org*). Hoewel deze mechanismes een effectieve manier zijn om de gemeenschap te waarschuwen voor beveiligings kwetsbaarheden, is het de taak van de systeembeheerders om hun systeem snel te corrigeren. Dit is in het bijzonder van belang omdat crackers toegang hebben tot dezelfde kwetsbaarheids volg systemen en zij zullen de informatie gebruiken om niet gecorrigeerde systemen te kraken wanneer ze dat kunnen. Goed systeembeheer vereist oplettendheid, constant bugs volgen, en juiste systeem onderhoud om een veiliger computer omgeving te waarborgen.

Refereer naar *Paragraaf 1.5, “Beveiligings vernieuwingen”* voor meer informatie over het bij de tijd houden van een systeem.

1.3.3.3. Onoplettend beheer

Beheerders die nalaten hun systemen te corrigeren zijn een van de grootste bedreigingen van server beveiliging. Volgens het *SysAdmin, Audit, Network, Security Institute* (*SANS*), is de belangrijkste oorzaak van computer beveiligings kwetsbaarheid "het aanstellen van niet getrainde mensen voor het onderhouden van beveiliging en het niet aanbieden van training noch de tijd om het mogelijk te maken de taak uit te voeren."[11] Dit is zowel van toepassing op onervaren beheerders als voor overmoedige of niet gemotiveerde beheerders.

Sommige beheerders laten na hun servers en werkstations te corrigeren, terwijl anderen nalaten om de log boodschappen van de systeem kernel of het netwerk verkeer te bekijken. Een andere veel voorkomende fout is het onveranderd laten van standaard wachtwoorden of sleutels voor services. Bijvoorbeeld, sommige databases hebben standaard beheerswachtwoorden omdat de database ontwikkelaars aannemen dat de systeembeheerder deze wachtwoorden onmiddellijk na de installatie zal veranderen. Als een database beheerder nalaat om dit wachtwoord te veranderen, kan zelfs een onervaren cracker het algemeen bekende standaard wachtwoord gebruiken om beheersrechten te krijgen voor de database. Dit zijn slechts een paar voorbeelden van het veroorzaken van in gevaar gebrachte servers door onoplettend beheer.

1.3.3.4. Inherent onveilige services

Zelfs de meest waakzame organisatie kan het slachtoffer worden van kwetsbaarheden als de netwerk services die ze kiezen inherent onveilig zijn. Bijvoorbeeld, er zijn veel services ontwikkeld met de aanname dat ze gebruikt worden in betrouwbare netwerken; deze aanname vervalt echter zodra de service beschikbaar komt via het Internet — welke zelf inherent onbetrouwbaar is.

Een categorie van onveilige netwerk sevices zijn degene die niet-versleutelde gebruikersnamen en wachtwoorden vereisen voor authenticatie. Telnet en FTP zijn zulke services. Als pakketsnuffel software het verkeer tussen de gebruiker op afstand en zo'n service volgt, kunnen gebruikersnamen en wachtwoorden eenvoudig onderschept worden.

Zulke services kunnen inherent ook gemakkelijker prooi worden voor wat de beveiligings industrie een *de-man-in-het-midden* aanval noemt. In dit type aanval leidt een cracker het netwerkverkeer om door het misleiden van een gekraakte naamserver op het netwerk om naar zijn machine te wijzen in plaats van de bedoelde server. Zodra iemand een sessie op afstand opent naar de server, gedraagt de machine van de aanvaller zich als een onzichtbaar kanaal en zit rustig tussen de service op afstand en de argeloze gebruiker informatie te verzamelen. Op deze manier kan een cracker beheerswachtwoorden en ruwe data verzamelen zonder dat de server of de gebruiker dit realiseert.

Een andere categorie van niet veilige services zijn netwerk bestandssystemen en informatie services zoals NFS of NIS, die expliciet ontwikkeld zijn voor LAN gebruik, maar helaas zijn uitgebreid om WAN's te omvatten (voor gebruikers op afstand). NFS heeft standaard geen authenticatie of beveiligings mechanismes ingesteld om te voorkomen dat een cracker het NFS deel aankoppelt en alles kan bereiken wat zich daar bevindt. NIS heeft ook vitale informatie die aan iedere computer op het netwerk bekend moet zijn, zoals wachtwoorden en bestandsrechten, dit in een leesbare tekst ASCII of DBM (van ASCII afgeleid) database. Een cracker die toegang krijgt tot deze database heeft dan toegang tot elk gebruikersaccount op een netwerk, inclusief het account van de beheerder.

[11] http://www.sans.org/resources/errors.php

Standaard wordt Fedora vrijgegeven met al deze services uitgezet. Echter, omdat beheerders vaak gedwongen worden om deze services te gebruiken, is een voorzichtige instelling kritisch. Refereer naar *Paragraaf 2.2, “Server beveiliging”* voor meer informatie over het instellen van services op een veilige manier.

1.3.4. Bedreigingen voor werkstations en beveiliging van thuis PC's

Werkstations en thuis PC's zijn misschien niet even vatbaar voor een aanval als netwerken of servers, maar omdat ze vaak gevoelige data bevatten, zoals credit kaart informatie, zijn ze het doelwit van systeem crackers. Werkstations kunnen ook besmet zijn zonder dat de gebruiker dat weet en gebruikt door aanvallers als een "slaaf" machine in gecoördineerde aanvallen. Als gebruikers daarom de kwetsbaarheden van een werkstation kennen, kunnen ze zich de moeite besparen om het operating systeem opnieuw te moeten installeren, of erger, te moeten herstellen van data diefstal.

1.3.4.1. Slechte wachtwoorden

Slechte wachtwoorden is een van de eenvoudigste manieren om toegang te krijgen tot een systeem. Voor meer informatie over het vermijden van valkuilen bij het aanmaken van een wachtwoord, refereer je naar *Paragraaf 2.1.3, “Wachtwoord beveiliging”*.

1.3.4.2. Kwetsbare cliënt toepassingen

Als een beheerder een volledig veilige en gecorrigeerde server heeft, betekent dat niet dat gebruikers op afstand veilig zijn wanneer ze er toegang naar krijgen. Bijvoorbeeld, als de server Telnet of FTP services aanbiedt over een publiek netwerk, kan een aanvaller de leesbare tekst gebruikersnamen en wachtwoorden bemachtigen als ze passeren over het netwerk, en daarna de account informatie gebruiken voor toegang naar het werkstation van de gebruiker op afstand.

Zelfs als veilige protocols gebruikt worden, zoals SSH, kan een gebruiker op afstand kwetsbaar zijn voor bepaalde aanvallen als ze hun cliënt toepassingen niet vernieuwen. Bijvoorbeeld, v.1 SSH cliënten zijn kwetsbaar voor een X-doorsturen aanval van kwaadwillige SSH servers. Zodra er verbonden is met de server, kan de aanvaller op zijn gemak alle toetsaanslagen en muisklikken die de gebruiker maakt onderscheppen over het netwerk. Dit probleem is gerepareerd in het v.2 SSH protocol, maar het is de taak van de gebruiker om bij te houden welke toepassingen zulke kwetsbaarheden hebben en ze te vernieuwen zoals vereist.

Paragraaf 2.1, “Werkstation beveiliging” bespreekt in meer detail welke stappen beheerders en thuisgebruikers moeten nemen om de kwetsbaarheid van hun computer werkstations te beperken.

1.4. Veel voorkomende uitbuitingen en aanvallen

Tabel 1.1, “Veel voorkomende uitbuitingen” geeft details van de meest voorkomende uitbuitingen en ingangspunten gebruikt door indringers om toegang te krijgen tot hulpbronnen in het netwerk van een organisatie. De sleutel voor deze veel voorkomende uitbuitingen is de uitleg hoe ze uitgevoerd worden en hoe beheerders hun netwerk kunnen behoeden voor zulke aanvallen.

Uitbuiting	Beschrijving	Opmerkingen
Lege of standaard wachtwoorden	Leaving administrative passwords blank or using a default password set by the product vendor. This is most common in hardware such as routers and firewalls, though some services	Vaak voorkomend in netwerk hardware zoals routers, firewalls, VPN's, en aan het netwerk gekoppelde opslag (NAS) apparaten.

Uitbuiting	Beschrijving	Opmerkingen
	that run on Linux can contain default administrator passwords (though Fedora 12 does not ship with them).	Veel voorkomend in legacy operating systemen, in het bijzonder diegene die services bundelen (zoals UNIX en Windows). Beheerders maken soms gebruikersaccount met veel rechten onder tijdsdruk aan en laten het wachtwoord leeg, daarmee maken ze een perfecte ingang voor kwaadwillige gebruikers die dat account ontdekken.
Standaard gedeelde sleutels	Beveiligde services bevatten soms standaard beveiligingssleutels voor ontwikkelings of evaluatie test doeleinden. Als deze sleutels onveranderd blijven en ze worden in een productie omgeving op het Internet geplaatst, hebben *alle* gebruikers met dezelfde standaard sleutels toegang tot die gedeelde sleutel hulpbron, en alle gevoelige informatie die het bevat.	Veel voorkomend in draadloze toegangs punten en voor-ingestelde beveiligde server apparaten.
IP adres voor de gek houden (spoofing)	Een machine op afstand doet zich voor als een node in jouw lokale netwerk, vindt kwetsbaarheden van je servers, en installeert een achterdeur programma of een paard van Troje om controle te krijgen over je netwerk hulpbronnen.	Spoofing is erg moeilijk omdat het inhoudt dat de aanvaller TCP/IP volgorde nummers moet voorspellen om een verbinding naar de doelsystemen te coördineren, maar verschillende gereedschappen zijn beschikbaar die crackers helpen een dergelijke kwetsbaarheid uit te voeren. Hangt af van de op het doel systeem draaiende services (zoals **`rsh`**, **`telnet`**, FTP en andere) die *op-broncode-gebaseerde* authenticatie technieken gebruiken, wat niet aanbevolen wordt in vergelijk met PKI of andere vormen van versleutelde authenticatie zoals gebruikt in **`ssh`** of SSL/TLS.
Afluisteren	Het verzamelen van data dat tussen twee actieve nodes op een netwerk uitgewisseld wordt door het afluisteren van de verbinding tussen de twee nodes.	Dit aanvals type werkt meestal met leesbare tekst verbindings protocollen, zoals Telnet, FTP, en HTTP overdracht. Een aanvaller op afstand moet toegang hebben tot een in gevaar gebracht systeem in een LAN om een dergelijke aanval uit te voeren; gewoonlijk heeft de cracker een actieve aanval (zoals IP spoofing of de-man-in-het-midden) gebruikt om

Uitbuiting	Beschrijving	Opmerkingen
		een systeem in het LAN in gevaar te brengen. Preventieve maatregelen zijn services met versleutelde sleutel uitwisseling, eenmalige wachtwoorden, of versleutelde authenticatie om wachtwoord snuffelen te voorkomen; sterke versleuteling gedurende de overdracht is ook aanbevolen.
Service kwetsbaarheden	Een aanvaller vindt een zwakte of kijkgat in een service die op het Internet draait; met deze kwetsbaarheid compromitteert de aanvaller het gehele systeem en alle data die het bevat, en kan mogelijk andere systemen in het netwerk in gevaar brengen.	Op HTTP-gebaseerde services zoals CGI zijn kwetsbaar voor het uitvoeren van commando's op afstand en zelfs voor interactieve shell toegang. Zelfs als de HTTP service draait als een gebruiker zonder rechten zoals "nobody", kan informatie zoals configuratie bestanden en netwerk plattegronden gelezen worden, of de aanvaller kan een dienstweigerings aanval starten die de systeem hulpbronnen laat opdrogen en het onbereikbaar maakt voor andere gebruikers. Services kunnen soms kwetsbaarheden hebben die niet opgemerkt worden tijdens de ontwikkeling en het testen; deze kwetsbaarheden (zoals *buffer overloop*, waarbij aanvallers een service laten crashen door het gebruik van willekeurige waardes die het geheugen buffer van een toepassing vullen, geven de aanvaller een interactieve commando prompt van waaruit ze willekeurige commando's kunnen uitvoeren) kunnen de hele beheers controle aan de aanvaller geven. Beheerders moeten er zeker van zijn dat services niet als de root gebruiker draaien, en moeten waakzaam blijven voor correcties of vernieuwingen voor toepassingen van leveranciers of beveiligings organisaties zoals CERT en CVE.
Toepassings kwetsbaarheden	Aanvallers vinden fouten in bureaublad en werkstation toepassingen (zoals email cliënten) en voeren willekeurige code uit, brengen paarden van	Werkstations en bureaubladen zijn gevoeliger voor uitbuiting omdat werknemers niet de kennis of ervaring hebben om in gevaar brengen te

Uitbuiting	Beschrijving	Opmerkingen
	Troje aan voor toekomstig in gevaar brengen, of laten systemen chrashen. Verdere uitbuiting kan plaatsvinden als het in gevaar gebrachte werkstation beheersrechten heeft op de rest van het netwerk.	voorkomen of te ontdekken; het is noodzakelijk om mensen te informeren over de risico's die ze nemen als ze ongeoorloofde software installeren of ongevraagde bijlages van emails openen. Beschermingen kunnen aangebracht worden zodat email cliënt software niet automatisch bijlages opent of uitvoert. Daarnaast kan de automatische vernieuwing van werkstation software met Red Hat Network of andere systeembeheerdiensten, de taak van multi-seat beveiligings opstellingen verlichten.
Service weigerings (Denial of service - DoS) aanvallen	Een aanvaller of een groep van aanvallers coördineren tegen het netwerk of de server hulpbronnen van een organisatie door het ongevraagd sturen van pakketten naar de doel host (of server, router, of werkstation). Dit veroorzaakt dat de hulpbronnen niet beschikbaar zijn voor rechtmatige gebruikers.	De meest vermelde DoS aanval in de VS vond plaats in 2000. Verscheidene druk bezochte commerciële en regerings sites werden onbereikbaar gemaakt door een gecoördineerde ping overspoelings aanval door het gebruik van verscheidene in gevaar gebrachte systemen met hoge bandbreedte verbindingen die optraden als *zombies*, of doorsturende uitzend nodes. Bron pakketten zijn meestal vervalst (en ook opnieuw uitgezonden), wat het onderzoek naar de echte herkomst van de aanval moeilijk maakt. Vooruitgang in toegangs filtering (IETF rfc2267) met gebruik van **`iptables`** en netwerk indringings detectie systemen zoals **`snort`** helpen beheerders met het opsporen en voorkomen van gespreide DoS aanvallen.

Tabel 1.1. Veel voorkomende uitbuitingen

1.5. Beveiligings vernieuwingen

Als beveiligings kwetsbaarheden ontdekt worden, moet de betreffende software vernieuwd worden om elk potentieel beveiligings risico te beperken. Als de software onderdeel is van een pakket in de Fedora distributie die op dat moment ondersteund wordt, heeft Fedora zich verplicht om zo spoedig mogelijk vernieuwde pakketten vrij te geven die de kwetsbaarheid repareren. Vaak gaan aankondigingen van een bepaalde beveiligings uitbuiting gepaard met een correctie (of bron code die het probleem repareert). Deze correctie wordt dan toegepast in het Fedora pakket, getest, en vrijgegeven als een errata vernieuwing. Als de aankondiging echter geen correctie bevat, werkt een

ontwikkelaar eerst samen met de onderhouder van de software om het probleem op te lossen. Zodra het probleem opgelost is, wordt het pakket getest en vrijgegeven als een errata vernieuwing.

Als een errata vernieuwing wordt vrijgegeven voor software die op je systeem gebruikt wordt, wordt het ten sterkste aanbevolen dat je de betreffende pakketten zo spoedig mogelijk vernieuwt om de tijdsduur dat je systeem potentieel kwetsbaar is te minimaliseren.

1.5.1. Pakketten vernieuwen

Als je pakketten op een systeem vernieuwt, is het belangrijk om de vernieuwing te downloaden van een vertrouwde bron. Een aanvaller kan een pakket eenvoudig opnieuw bouwen met hetzelfde versie nummer als het pakket dat verondersteld wordt om het probleem op te lossen, maar met een andere beveiligings uitbuiting en dit vrijgeven op het Internet. Als dit gebeurt, wordt de uitbuiting niet ontdekt met veiligheids maatregelen zoals het vergelijken van bestanden met de originele RPM. Het is dus erg belangrijk om RPM's alleen te downloaden van vertrouwde bronnen, zoals van Fedora, en de ondertekening van het pakket te controleren op zijn integriteit.

Opmerking

Fedora bevat een handig paneel icoon dat zichtbare waarschuwingen laat zien als er een vernieuwing is voor een Fedora systeem.

1.5.2. Ondertekende pakketten verifiëren

Alle Fedora pakketten zijn ondertekend met de Fedora *GPG* sleutel. GPG staat voor GNU Privacy Guard, of GnuPG, een vrij software pakket voor het verzekeren van authenticiteit van verspreide bestanden. Bijvoorbeeld, een privé sleutel (geheime sleutel) sluit het pakket af terwijl een publieke sleutel het pakket opent en verifieert. Als de publieke sleutel die door Fedora geleverd wordt tijdens de RPM verificatie niet past met de geheime sleutel, kan het pakket veranderd zijn en kan daarom niet vertrouwd worden.

Het RPM programma in Fedora probeert automatisch de GPG ondertekening van een RPM pakket te verifiëren voordat het geïnstalleerd wordt. Als de Fedora GPG sleutel niet geïnstalleerd is, doe dat dan van een veilige, statische locatie, zoals een Fedora installatie CD-ROM of DVD.

Aannemende dat de schijf zich bevindt in **`/mnt/cdrom`**, gebruik je het volgende commando om het te importeren in de *sleutelring* (een database van vertrouwde sleutels op het systeem):

```
rpm --import /mnt/cdrom/RPM-GPG-KEY
```

Om een lijst te laten zien van alle sleutels die geïnstalleerd zijn voor RPM verificatie, voer je het volgende commando uit:

```
rpm -qa gpg-pubkey*
```

De output zal op het volgende lijken:

```
gpg-pubkey-db42a60e-37ea5438
```

Om details van een specifieke sleutel te laten zien, voer je het **`rpm -qi`** commando uit gevolgd door de output van het vorige commando, zoals in dit voorbeeld:

```
rpm -qi gpg-pubkey-db42a60e-37ea5438
```

Het is uiterst belangrijk dat je de ondertekening van de RPM bestanden verifieert voor het installeren om er zeker van te zijn dat ze niet veranderd zijn ten opzichte van de originele bron van de pakketten. Om alle gedownloade pakketten tegelijk te verifiëren voer je het volgende commando uit:

```
rpm -K /tmp/updates/*.rpm
```

Het commando geeft voor ieder pakket `gpg OK` terug als de GPG sleutel met succes geverifieerd is. Als dat niet het geval is, wees er dan zeker van dat je de juiste Fedora publieke sleutel gebruikt en controleer ook de bron van de inhoud. Pakketten die niet voldoen aan de GPG verificatie moeten niet geïnstalleerd worden, omdat ze door derden veranderd kunnen zijn.

Na het verifiëren van de GPG sleutel en het downloaden van alle pakketten die behoren bij het errata rapport, installeer je de pakketten als root op een shell prompt.

1.5.3. Ondertekende pakketten installeren

Het installeren van de meeste pakketten kan veilig gedaan worden (behalve voor kernel pakketten) met het volgende commando:

```
rpm -Uvh /tmp/updates/*.rpm
```

Voor kernel pakketten gebruik je het volgende commando:

```
rpm -ivh /tmp/updates/<kernel-pakket>
```

Vervang <*kernel-pakket*> in het vorige voorbeeld met de naam van de kernel RPM.

Zodra de machine veilig opnieuw opgestart is met de nieuwe kernel, kan de oude kernel verwijderd worden met het volgende commando:

```
rpm -e <oude-kernel-pakket>
```

Vervang <*oude-kernel-pakket*> in het vorige voorbeeld met de naam van de oude kernel RPM.

Opmerking

Het is niet vereist dat de oude kernel verwijderd wordt. De standaard boot loader, GRUB, staat toe dat meerdere kernels geïnstalleerd worden, waarna een gekozen kan worden in een menu tijdens het opstarten.

Belangrijk

Voordat je een beveiligings errata installeert, wees er dan zeker van om de speciale instructies te lezen die zich bevinden in het errata rapport en voer deze dan uit. Refereer naar *Paragraaf 1.5.4, "De veranderingen toepassen"* voor algemene instructies over het toepassen van de veranderingen gemaakt door een errata vernieuwing.

1.5.4. De veranderingen toepassen

Na het downloaden en installeren van beveiligings errata en vernieuwingen, is het belangrijk om het gebruik van de oudere software te stoppen en te beginnen met het gebruiken van de nieuwe software. Hoe dit gedaan wordt hangt af van het type software dat vernieuwd is. De volgende lijst somt de algemene categorieën van software op en geeft instructies voor het gebruik van de vernieuwde versies na een pakket vernieuwing.

Opmerking

In het algemeen is het systeem opnieuw opstarten de veiligste manier om te verzekeren dat de laatste versie van een software pakket wordt gebruikt; deze optie is echter niet altijd vereist, of beschikbaar voor de systeembeheerder.

Toepassingen
: Gebruikers-ruimte toepassingen zijn alle programma's die opgestart worden door een systeem gebruiker. Gewoonlijk worden deze toepassingen alleen gebruikt als een gebruiker, een script, of een automatische taak programma ze opstart en ze blijven niet voortduren voor lange tijdsperiodes.

  Zodra zo'n gebruikers-ruimte toepassing vernieuwd is, stop je alle instances van de toepassing op het systeem en je start het programma opnieuw op om de vernieuwde versie te gebruiken.

Kernel
: De kernel is het kern software onderdeel van het Fedora operating systeem. Het beheert toegang tot het geheugen, de processor, en randapparaten en het plant alle taken.

  Door zijn centrale rol, kan de kernel niet opnieuw gestart worden zonder ook de computer te stoppen. Daarom kan een vernieuwde versie van de kernel pas gebruikt worden als het systeem opnieuw opgestart wordt.

Gedeelde bibliotheken
: Gedeelde bibliotheken zijn stukken code, zoals **`glibc`**, welke gebruikt worden door een aantal toepassingen en services. Toepassingen die een gedeelde bibliotheek gebruiken laden de gedeelde code als de toepassing opgestart wordt, dus alle toepassingen die de vernieuwde bibliotheek gebruiken moeten gestopt en opnieuw opgestart worden.

  Om te bepalen welke draaiende toepassingen verbonden zijn met een bepaalde bibliotheek, gebruik je het **`lsof`** commando zoals in het volgende voorbeeld:

```
lsof /lib/libwrap.so*
```

  Dit commando geeft een lijst terug van alle draaiende programma's die TCP wrappers gebruiken voor host toegangscontrole. Daarom moet elk programma in de lijst gestopt en opnieuw opgestart worden als het **`tcp_wrappers`** pakket vernieuwd wordt.

SysV services
: SysV services zijn blijvende server programma's die opgestart worden tijdens het boot proces. Voorbeelden van SysV services zijn **`sshd`**, **`vsftpd`**, en **`xinetd`**.

  Omdat deze programma's gewoonlijk blijvend in het geheugen zijn zolang de computer aanstaat, moet iedere vernieuwde SysV service gestopt en opnieuw opgestart worden nadat het pakket

is vernieuwd. Dit kan gedaan worden met het **Service Configuratie** gereedschap of door in te loggen in een root shell prompt en het **/sbin/service** commando uit te voeren zoals in het volgende voorbeeld:

```
/sbin/service <service-naam> restart
```

In het vorige voorbeeld vervang je <*service-naam*> met de naam van de service, zoals **sshd**.

xinetd services

Services die gecontroleerd worden door de **xinetd** super service draaien alleen als er een actieve verbinding is. Voorbeelden van services die gecontroleerd worden door **xinetd** zijn Telnet, IMAP, en POP3.

Omdat nieuwe instances van deze services opgestart worden door **xinetd** iedere keer als een nieuw verzoek wordt ontvangen, worden verbindingen die optreden na de vernieuwing afgehandeld door de vernieuwde software. Als er echter actieve verbindingen zijn op het moment dat de door **xinetd** gecontroleerde service vernieuwd wordt, blijven die werken met de oude versie van de software.

Om oudere instances van een bepaalde service die door **xinetd** gecontroleerd wordt te stoppen, vernieuw je het pakket voor de service en daarna stop je alle processen die op dat moment draaien. Om te bepalen of het proces draait gebruik je het **ps** commando en je gebruikt daarna het **kill** of **killall** commando om de huidige instances van de service te stoppen.

Bijvoorbeeld, als een beveiligings errata voor de **imap** pakketten wordt vrijgegeven, vernieuw je de pakketten, en daarna type je het volgende commando in als root in een shell prompt:

```
ps -aux | grep imap
```

Dit commando geeft alle actieve IMAP sessies terug. Individuele sessies kunnen dan gestopt worden met het volgende commando:

```
kill <PID>
```

Als hiermee het stoppen mislukt, gebruik je het volgende commando:

```
kill -9 <PID>
```

In de vorige voorbeelden vervang je <*PID*> met het proces identificatie nummer (dat je vindt in de tweede kolom van de output van het **ps** commando) voor een IMAP sessie.

Om alle actieve IMAP sessies te stoppen, voer je het volgende commando uit:

```
killall imapd
```

Je netwerk beveiligen

2.1. Werkstation beveiliging

Een Linux omgeving beveiligen begint met het werkstation. Of het gaat om het vergrendelen van een persoonlijke machine of het beveiligen van een bedrijfssysteem, een gezonde beveiligingstactiek begint met de individuele computer. Een computer netwerk slechts zo veilig als zijn zwakste node.

2.1.1. Het onderzoeken van werkstation beveiliging

Als de beveiliging van een Fedora werkstation onderzocht wordt, overweeg dan het volgende:

- *BIOS en boot loader beveiliging* — Kan een onbevoegde gebruiker fysieke toegang tot de machine krijgen en opstarten in de enkele-gebruikers of reddings mode zonder een wachtwoord?
- *Wachtwoord beveiliging* — Hoe veilig zijn de gebruikersaccount wachtwoorden op de machine?
- *Beheerscontrole* — Wie heeft een account op het systeem en hoeveel beheerscontrole hebben zij?
- *Beschikbare network services* — Welke services luisteren naar verzoeken van het netwerk en moeten zij wel draaien?
- *Persoonlijke firewalls* — Welk type firewall, indien aanwezig, is nodig?
- *Beveiligings verbeter gereedschappen* — Welke gereedschappen moeten gebruikt worden om te communiceren tussen de werkstations en welke moeten gemeden worden?

2.1.2. BIOS en boot loader beveiliging

Wachtwoord bescherming voor de BIOS (of het equivalent daarvan) en de boot loader kan beletten dat onbevoegde gebruikers, die fysieke toegang tot systemen hebben, opstarten met verwijderbare media of root rechten krijgen met de enkele-gebruiker mode. De beveiligingsmaatregelen die je moet nemen om te beschermen tegen zulke aanvallen hangt zowel af van de gevoeligheid van de informatie op het werkstation als de locatie van de machine.

Bijvoorbeeld, als een machine gebruikt wordt op een beurs en geen gevoelige informatie bevat, dan hoeft het niet kritisch te zijn om zulke aanvallen te voorkomen. Als echter de laptop van een werknemer met privé, niet-versleutelde SSH sleutels voor het bedrijfsnetwerk onbeheerd achter wordt gelaten op dezelfde beurs, kan dat leiden tot een belangrijke beveiligings inbreuk met gevolgen voor het gehele bedrijf.

Als het werkstation zich op een locatie bevindt waarnaar alleen bevoegde of vertrouwde mensen toegang hebben, dan hoeft echter het beveiligen van de BIOS of de boot loader niet nodig zijn.

2.1.2.1. BIOS wachtwoorden

De twee belangrijkste redenen voor wachtwoord bescherming van de BIOS van een computer zijn[1]:

1. *Het voorkomen van het veranderen van de BIOS instelling* — Als een indringer toegang heeft tot de BIOS, dan kunnen ze deze instellen om op te starten van een CD-ROM of diskette. Dit maakt

[1] Omdat systeem BIOS'en afhangen van de leverancier, kunnen sommige geen van beide types wachtwoordbescherming ondersteunen, terwijl andere maar een van de twee types hebben.

het voor hen mogelijke om op te starten in de reddings mode of de enkele-gebruikers mode, wat op zijn beurt toestaat om willekeurige processen op het systeem op te starten of om gevoelige data te kopiëren.

2. *Systeem opstarten beletten* — Sommige BIOS'en staan wachtwoord bescherming toe voor het opstart proces. Als dit aangezet is, moet de aanvaller een wachtwoord opgeven voordat de BIOS de boot loader opstart.

Om dat de methode voor het instellen van een BIOS wachtwoord afhangt van de computer fabrikant, raadpleeg je de handleiding van de computer voor specifieke instructies.

Als je het BIOS wachtwoord vergeet, kan het terug gezet worden of met jumpers op het moederbord of door het los koppelen van de CMOS batterij. Hierdoor is het een goede praktijk om de computer kast af te sluiten als dit mogelijk is. Raadpleeg echter de handleiding van de computer of het moederbord voordat je probeert om de CMOS batterij los te koppelen.

2.1.2.1.1. Het beveiligen van een niet-x86 platform

Andere architecturen gebruiken andere programma's om taken op een laag niveau uit te voeren die ruwweg overeenkomen met die van de BIOS van x86 systemen. Bijvoorbeeld, Intel® Itanium™ computers gebruiken de *Extensible Firmware Interface* (*EFI*) shell.

Voor instructies over wachtwoordbescherming van BIOS-achtige programma's op andere architecturen, refereer je naar de instructies van de fabrikant.

2.1.2.2. Boot loader wachtwoorden

De belangrijkste redenen voor wachtwoordbescherming van een Linux boot loader zijn:

1. *Het voorkomen van toegang tot de enkele-gebruikers mode* — Als aanvallers de machine kunnen opstarten in de enkele-gebruikers mode, worden ze automatisch ingelogd als root zonder dat hen het root wachtwoord gevraagd wordt.

2. *Het voorkomen van toegang tot de GRUB console* — Als de machine GRUB als boot loader gebruikt, kan een aanvaller de GRUB bewerker interface gebruiken om zijn instelling te veranderen of om informatie te krijgen met behulp van het **`cat`** commando.

3. *Het voorkomen van toegang tot onveilige operating systemen* — Als het een systeem is met meerdere operating systemen, kan een aanvaller tijdens het opstarten een operating systeem kiezen (bijvoorbeeld, DOS), welke toegangscontrole en bestandsrechten negeert.

Fedora wordt geleverd met de GRUB boot loader op het x86 platform. Voor een uitgebreide beschrijving van GRUB refereer je naar de Fedora installatie gids.

2.1.2.2.1. Wachtwoordbescherming voor GRUB

Je kunt GRUB instellen om de eerste twee problemen getoond in *Paragraaf 2.1.2.2, “Boot loader wachtwoorden”* op te lossen door het toevoegen van een password commando in zijn configuratie bestand. Om dit te doen, kies je eerst een sterk wachtwoord, je opent een shell, je logt in als root, en dan type je het volgende commando:

```
/sbin/grub-md5-crypt
```

Als er om gevraagd wordt type je het GRUB wachtwoord in en je drukt op **Enter**. Dit geeft een MD5 hash van het wachtwoord terug.

Vervolgens bewerk je het GRUB configuratie bestand **/boot/grub/grub.conf**. Open het bestand en onder de **timeout** regel in de hoofd sectie van het document, voeg je de volgende regel toe:

```
password --md5 <wachtwoord-hash>
```

Vervang <*wachtwoord-hash*> met de waarde die teruggegeven is door **/sbin/grub-md5-crypt**[2].

De volgende keer dat het systeem opstart, voorkomt het GRUB menu toegang tot de bewerker of commando interface zonder eerst op **p** te duwen gevolg door het GRUB wachtwoord.

Helaas belet deze oplossing een aanvaller niet om op te starten in een onveilig operating systeem in een omgeving met meerdere operating systemen. Hiervoor moet een ander deel van het **/boot/grub/grub.conf** bestand bewerkt worden.

Zoek naar de `title` regel van het operating systeem dat je wilt beveiligen, en voeg een regel toe met het **lock** commando direct onder deze.

Voor een DOS systeem, moet de strofe beginnen overeenkomstig het volgende:

```
title DOS lock
```

Waarschuwing

Een `password` regel moet aanwezig zijn in de hoofd sectie van het **/boot/grub/grub.conf** bestand om deze methode goed te laten werken. Anders kan een aanvaller toegang krijgen tot de GRUB bewerker interface en de lock regel verwijderen.

Om een ander wachtwoord te maken voor een bepaalde kernel of operating systeem, voeg je een **lock** regel toe aan de strofe, gevolgd door een password regel.

Elke strofe die beschermd wordt met een uniek wachtwoord moet beginnen met regels die lijken op het volgende voorbeeld:

```
title DOS lock password --md5 <wachtwoord-hash>
```

2.1.3. Wachtwoord beveiliging

Wachtwoorden is de belangrijkste methode die Fedora gebruik om de identiteit van een gebruiker te controleren. Daarom is wachtwoord beveiliging zo belangrijk voor de bescherming van de gebruiker, het werkstation, en het netwerk.

Voor beveiligings doeleinden, stelt het installatie programma het systeem in om *Message-Digest Algorithm* (*MD5*) en schaduw wachtwoorden te gebruiken. Het wordt ten sterkste afgeraden om deze instellingen te veranderen.

Als MD5 wachtwoorden worden uitgezet tijdens de installatie, wordt het oudere *Data Encryption Standard* (*DES*) formaat gebruikt. Dit formaat beperkt wachtwoorden tot acht alfanumerieke karakters

[2] GRUB accepteert ook niet-versleutelde wachtwoorden, maar het wordt aanbevolen om een MD5 hash te gebruiken voor extra veiligheid.

(leestekens en andere speciale karakters zijn uitgesloten), en biedt een bescheiden 56-bit niveau van versleuteling.

Als schaduw wachtwoorden wordt uitgezet tijdens de installatie, worden alle wachtwoorden bewaard als een one-way hash in het **/etc/passwd** bestand wat leesbaar is voor iedereen, wat het systeem gevoelig maakt voor offline wachtwoord kraak aanvallen. Als een indringer toegang tot de machine kan krijgen als een gewone gebruiker, kan hij het **/etc/passwd** bestand naar zijn eigen machine kopiëren en daarna er een aantal wachtwoord kraak programma's op los laten. Als er een onveilig wachtwoord in het bestand is, is het slechts een kwestie van tijd voordat de wachtwoord kraker het ontdekt.

Schaduw wachtwoorden sluiten dit type aanval uit door de wachtwoord hashes te bewaren in het bestand **/etc/shadow**, welke alleen leesbaar is door de root gebruiker.

Dit forceert een potentiële aanvaller om wachtwoordkraken op afstand te proberen door in te loggen op een netwerk service van de machine zoals SSH of FTP. Dit soort brute kracht aanval is veel langzamer en laat een duidelijk spoor achter omdat honderden mislukte login pogingen naar systeembestanden geschreven worden. Natuurlijk, als de cracker de aanval midden in de nacht begint op een systeem met zwakke wachtwoorden, kan de cracker toegang hebben gekregen voor de ochtend en de logbestanden bewerkt hebben om zijn sporen te verbergen.

Naast de formaat en opslagplaats overwegingen is er die over de inhoud. Het aller belangrijkste wat een gebruiker kan doen om zijn account tegen een wachtwoordkraak aanval te beschermen is het maken van een sterk wachtwoord.

2.1.3.1. Sterke wachtwoorden maken

Als je een veilig wachtwoord wilt maken, is het een goede idee om de volgende richtlijnen te volgen:

- *Gebruik niet alleen woorden of getallen* — Gebruik nooit alleen getallen of woorden in een wachtwoord.

  Een paar onveilige wachtwoorden zijn de volgende:

  - 8675309
  - jan
  - hackmij

- *Gebruik geen herkenbare woorden* — Woorden zoals gewone namen, woordenboek woorden, of zelfs uitdrukkingen uit televisie shows of boeken moeten vermeden worden, zelfs als ze omgeven zijn met getallen.

  Een paar onveilige wachtwoorden zijn de volgende:

  - john1
  - DS-9
  - mentat123

- *Gebruik geen woorden uit vreemde talen* — Wachtwoord kraak programma's raadplegen vaak woordenlijsten die woordenboeken van vele vreemde talen omvatten. Steunen op vreemde talen voor veilige wachtwoorden is niet veilig.

Een paar onveilige wachtwoorden zijn de volgende:

- cheguevara
- bienvenido1
- 1dumbKopf

- *Gebruik geen hacker terminologie* — Als je denkt dat je bij de elite hoort omdat je hacker terminologie — ook wel l337 (LEET) spraak genoemd — in je wachtwoord gebruikt, denk dan even na. Veel woordenlijsten bevatten LEET spraak.

  Een paar onveilige wachtwoorden zijn de volgende:

  - H4X0R
  - 1337

- *Gebruik geen persoonlijke informatie* — Vermijdt het gebruik van enige persoonlijke informatie in je wachtwoorden. Als de aanvaller je identiteit kent, wordt de taak van het ontdekken van je wachtwoorden gemakkelijker. Het volgende is een lijst van het type informatie die je moet vermijden als je een wachtwoord maakt:

  Een paar onveilige wachtwoorden zijn de volgende:

  - Je naam
  - De namen van je huisdieren
  - De namen van familieleden
  - Geboorte data
  - Je telefoonnummer of postcode

- *Keer geen herkenbare woorden om* — Goede wachtwoord checkers keren gewone woorden altijd om, dus het omkeren van een slecht wachtwoord maakt het niet veiliger.

  Een paar onveilige wachtwoorden zijn de volgende:

  - R0X4H
  - naj
  - 9-DS

- *Schrijf je wachtwoord niet op* — Bewaar een wachtwoord nooit op papier. Het is veel veiliger om het te onthouden.

- *Gebruik niet hetzelfde wachtwoord voor alle machines* — Het is belangrijk om aparte wachtwoorden te maken voor iedere machine. Op deze manier zijn niet alle machines in gevaar als een systeem in gevaar is gebracht.

De volgende richtlijnen zullen je helpen om een sterk wachtwoord te maken:

- *Maak het wachtwoord ten minste acht karakters lang* — Hoe langer het wachtwoord is, des te beter. Als MD5 wachtwoorden gebruikt worden, moet het 15 karakters zijn of langer. Met DES wachtwoorden gebruik je de maximale lengte (acht karakters).
- *Vermeng hoofd en kleine letters* — Fedora is gevoelig voor hoofd en kleine letters, dus vermeng deze om de kracht van je wachtwoord te versterken.
- *Vermeng letter en cijfers* — Cijfers toevoegen aan wachtwoorden kunnen de wachtwoord sterkte verbeteren, zeker als ze in het midden toegevoegd worden (niet alleen aan het begin en het eind).
- *Voeg niet-alfanumerieke karakters toe* — Speciale karakters zoals &, $, en > kunnen de kracht van een wachtwoord sterk verbeteren (dit is niet mogelijk bij het gebruik van DES wachtwoorden).
- *Neem een wachtwoord dat je kunt onthouden* — Het beste wachtwoord van de wereld helpt je weinig als het niet kunt onthouden; gebruik letterwoorden of andere geheugensteuntjes om je te helpen wachtwoorden te onthouden.

Met al deze regels, kan het moeilijk lijken om een wachtwoord te maken dat aan alle criteria voor goede wachtwoorden voldoet terwijl de problemen van slechte vermeden worden. Gelukkig zijn er enkele stappen die je kunt nemen om een eenvoudig te onthouden, veilig wachtwoord kunt maken.

2.1.3.1.1. Een methode om veilige wachtwoorden te maken

Er zijn vele methodes die men gebruikt om veilige wachtwoorden te maken. Een van de meer populaire methodes gebruikt letterwoorden. Bijvoorbeeld:

- Bedenk een eenvoudig te onthouden zin, zoals:

  "over de rivier en door de wouden, gaan we naar grootmoeder's huis "

- Vervolgens, zet je het om een in letterwoord (inclusief de leestekens).

  `odreddw,gwngh.`

- Voeg complexiteit toe door het invullen van nummers en symbolen voor letters in het letterwoord. Bijvoorbeeld, vul **`7`** in voor **`d`** en het `@` symbool voor **`w`**:

  `o7re77@,g@ngh.`

- Voeg nog meer complexiteit toe door minsters een letter te vervangen door een hoofdletter, zoals **`H`**.

  `o7re77@,g@ngH.`

- *Tenslotte, gebruik nooit het voorbeeld wachtwoord hierboven voor een systeem*.

Terwijl het maken van veilige wachtwoorden noodzakelijk is, is het ook belangrijk ze goed te beheren, zeker voor systeembeheerders in grotere organisaties. De volgende paragraaf beschrijft goede praktijken voor het maken en beheren van gebruikerswachtwoorden in een organisatie.

2.1.3.2. Het maken van gebruikerswachtwoorden in een organisatie

Als een organisatie een groot aantal gebruikers heeft, hebben de systeembeheerders twee basis opties beschikbaar om het gebruik van goede wachtwoorden af te dwingen. Ze kunnen wachtwoorden maken voor de gebruiker, of ze kunnen gebruikers hun eigen wachtwoord laten maken, waarbij ze controleren of de wachtwoorden van acceptabele kwaliteit zijn.

Het maken van wachtwoorden voor de gebruikers verzekert dat de wachtwoorden goed zijn, maar het wordt een ontmoedigende taak als de organisatie groeit. Het vergroot ook het risico dat gebruikers hun wachtwoord opschrijven.

Om deze reden geven de meeste beheerders er de voorkeur aan dat gebruikers hun eigen wachtwoord maken, maar ze controleren dat de wachtwoorden goed zijn en, in enkele gevallen, dwingen hun gebruikers om hun wachtwoord periodiek te veranderen door middel van wachtwoord veroudering.

2.1.3.2.1. Sterke wachtwoorden afdwingen

Om het netwerk voor indringing te beschermen is het een goed idee dat systeembeheerders controleren of de wachtwoorden die in de organisatie gebruikt worden sterk zijn. Als gebruikers gevraagd worden wachtwoorden te maken of te veranderen, kunnen ze de commandoregel toepassing **`passwd`** gebruiken, die bewust is van *Pluggable Authentication Manager* (*PAM*) en daarom controleert of het wachtwoord te kort is of op een andere manier eenvoudig te kraken. Deze controle wordt gedaan door de **`pam_cracklib.so`** PAM module. Om dat PAM aan te passen is, is het mogelijk om meer wachtwoord integriteit checkers toe te voegen, zoals **`pam_passwdqc`** (beschikbaar op *http://www.openwall.com/passwdqc/*) of een nieuwe module te schrijven. Voor een lijst van beschikbare PAM modules, refereer je naar *http://www.kernel.org/pub/linux/libs/pam/modules.html*. Voor meer informatie over PAM, refereer je naar *Paragraaf 2.4, “Pluggable Authentication Modules (PAM)”*.

De wachtwoord check die uitgevoerd wordt tijdens het maken ontdekt slechte wachtwoorden niet even effectief als het draaien van een wachtwoord kraak programma voor de wachtwoorden.

Vele wachtwoord kraak programma's zijn beschikbaar die in Fedora draaien, hoewel er geen meegeleverd wordt met het operating systeem. Hieronder is een korte lijst van de bekendere wachtwoord kraak programma's:

- ***John The Ripper*** — Een snel en flexibel wachtwoord kraak programma. Het staat het gebruik van meerdere woordenlijsten toe en is in staat brute-kracht wachtwoord kraken uit te voeren. Het is beschikbaar op *http://www.openwall.com/john/*.

- ***Crack*** — Perhaps the most well known password cracking software, **Crack** is also very fast, though not as easy to use as **John The Ripper**. It can be found online at *http://www.crypticide.com/alecm/security/crack/c50-faq.html*.

- ***Slurpie*** — **Slurpie** komt overeen met **John The Ripper** en **Crack**, maar het is ontworpen om tegelijkertijd op meerdere computers te draaien, waarmee het een verspreide wachtwoord kraak aanval maakt. Het kan gevonden worden tezamen met een aantal andere verspreide aanval beveiligings evaluatie gereedschappen op *http://www.ussrback.com/distributed.htm*.

Waarschuwing

Zorg ervoor dat je altijd schriftelijke toestemming hebt voordat je wachtwoorden gaat kraken in een organisatie.

2.1.3.2.2. Wachtzinnen

Passphrases and passwords are the cornerstone to security in most of today's systems. Unfortunately, techniques such as biometrics and two-factor authentication have not yet become mainstream in many systems. If passwords are going to be used to secure a system, then the use of passphrases should

be considered. Passphrases are longer than passwords and provide better protection than a password even when implemented with non-standard characters such as numbers and symbols.

2.1.3.2.3. Wachtwoord verloop

Wachtwoord verloop is een andere techniek die gebruikt wordt door systeembeheerders voor de verdediging tegen slechte wachtwoorden in een organisatie. Wachtwoord verloop betekent dat na een vastgestelde periode (gewoonlijk 90 dagen), de gebruiker gevraagd wordt om een nieuw wachtwoord te maken. De theorie hier achter is dat als een gebruiker gedwongen wordt om zijn wachtwoord periodiek te veranderen, een gekraakt wachtwoord slechts tijdelijk voor een indringer bruikbaar is. Het nadeel van wachtwoord verloop is dat gebruikers meer de neiging hebben om hun wachtwoord op te schrijven.

Er zijn twee belangrijke programma's voor het opgeven van wachtwoord verloop in Fedora: het **chage** commando of de grafische **Gebruikersbeheerder** (**system-config-users**) toepassing.

De `-M` optie van het **chage** commando specificeert het maximale aantal dagen dat een wachtwoord geldig is. Bijvoorbeeld, om het wachtwoord van een gebruiker in te stellen om na 90 dagen te verlopen, gebruik je het volgende commando:

```
chage -M 90 <gebruikersnaam>
```

In het bovenstaande voorbeeld, vevang je *<gebruikersnaam>* met de naam van de gebruiker. Om wachtwoord verloop uit te zetten, wordt gewoonlijk een waarde van **99999** na de `-M` optie ingevuld (dit komt overeen met 273 jaren).

Je kunt het **chage** commando ook in interactieve mode gebruiken om meerdere wachtwoord verlopen en andere account details te veranderen. Gebruik het volgende commando om in de interactieve mode te komen:

```
chage <gebruikersnaam>
```

Het volgende is een voorbeeld interactieve sessie met gebruik van dit commando:

```
[root@myServer ~]# chage davido
Changing the aging information for davido
Enter the new value, or press ENTER for the default
Minimum Password Age [0]: 10
Maximum Password Age [99999]: 90
Last Password Change (YYYY-MM-DD) [2006-08-18]:
Password Expiration Warning [7]:
Password Inactive [-1]:
Account Expiration Date (YYYY-MM-DD) [1969-12-31]:
[root@myServer ~]#
```

Refereer naar de manual pagina voor meer informatie over de beschikbare opties.

Je kunt ook de grafische **Gebruikersbeheerder** toepassing gebruiken om wachtwoord verloop tactiek te maken. Merk op: je hebt beheersrechten nodig om deze procedure uit te voeren.

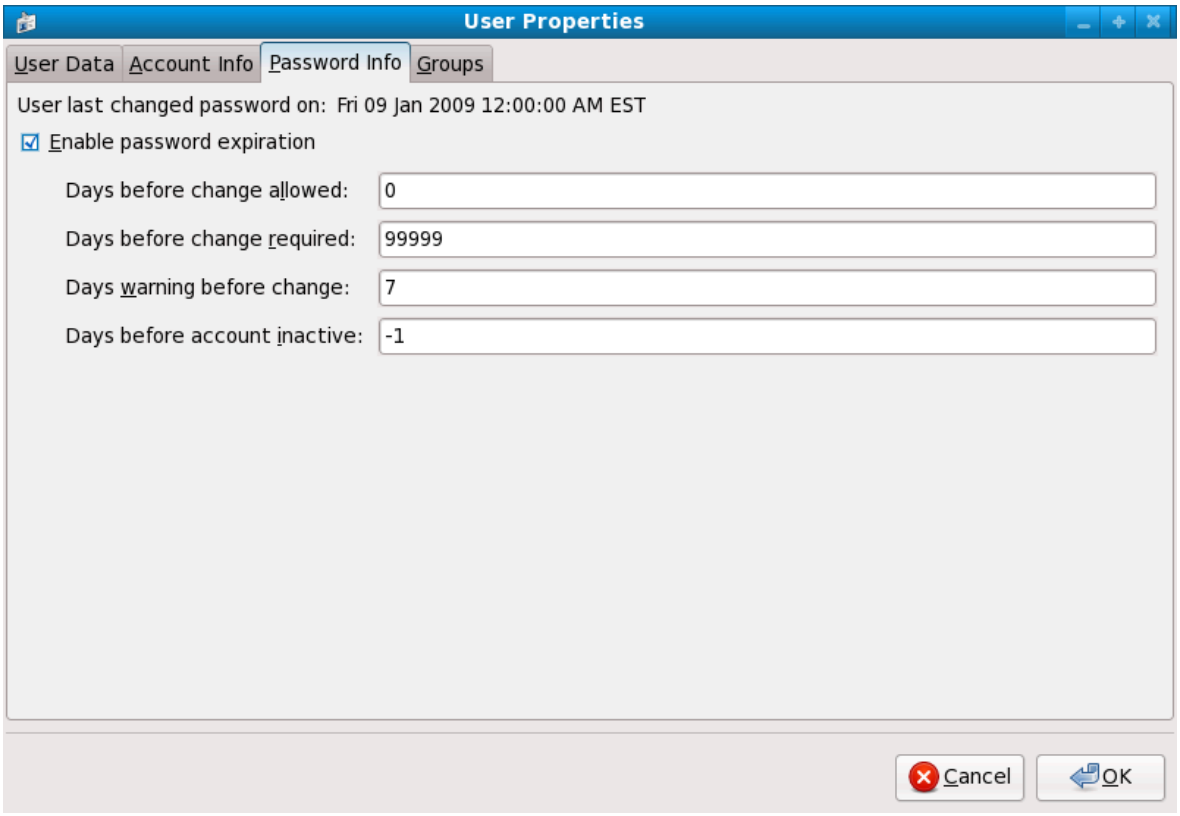
1. Klik op het **Systeem** menu op het Paneel, ga naar **Beheer** en klik dan op **Gebruikers en groepen** om de Gebruikersbeheerder te openen. Als alternatief type je het commando **system-config-users** op de shell prompt.

2. Klik op de **Gebruikers** tab en selecteer de gewenste gebruiker in de lijst van gebruikers.

3. Klik op **Eigenschappen** op de gereedsschapbalk om de Gebruikerseigenschappen dialoog te openen (of kies **Eigenschappen** in het **Bestand** menu).

4. Klik op de **Wachtwoordinformatie** tab, en selecteer het aanvinkhokje voor **Wachtwoordverloop aanzetten**.

5. Vul de vereiste waarde in het **Dagen voordat verandering is vereist** veld in, en klik op **OK**.



Figuur 2.1. Wachtwoordverloop opties opgeven

2.1.4. Administratieve controles

Als je een thuis computer beheert, moet de gebruiker sommige taken uitvoeren als de root gebruiker of door het verkrijgen van effectieve root rechten met een *setuid* programma, zoals **sudo**of met **su**. Een setuid programma is een programma dat werkt met de gebruikers ID (*UID*) van de eigenaar van het programma in plaats van de gebruiker van het programma. Zulke programma's worden aangegeven met een s in de eigenaars sectie van een lange formaat listing, zoals in het volgende voorbeeld:

```
-rwsr-xr-x 1 root root 47324 May 1 08:09 /bin/su
```

Opmerking

De s kan in hoofd letter of kleine letter zijn. Als het in hoofdletter verschijnt, betekent het dat de achterliggende toestemmings bit niet gezet is.

Voor de systeembeheerders van een organisatie, moet echter een keuze gemaakt worden over hoe veel beheerstoegang gebruikers binnen de organisatie moeten hebben voor hun machine. Met een PAM module genaamd **pam_console.so**, worden sommige activiteiten die normaal alleen voor de root gebruiker gereserveerd zijn, zoals opnieuw opstarten en het aankoppelen van verwijderbare media, toegestaan aan de eerste gebruiker die inlogt op de fysieke console (refereer naar *Paragraaf 2.4, "Pluggable Authentication Modules (PAM)"* voor meer informatie over de **pam_console.so** module). Echter, andere belangrijke systeembeheer taken, zoals het veranderen van netwerk instellingen, een nieuwe muis configureren, of netwerkapparaten aankoppelen, zijn niet mogelijk zonder beheersrechten. Het resultaat is dat systeembeheerders moeten beslissen hoeveel toegang de gebruikers op hun netwerk moeten krijgen.

2.1.4.1. Root toegang toestaan

Als de gebruikers binnen een organisatie vertrouwd zijn en computer kennis hebben, dat kan het toestaan van root toegang aan hen geen probleem zijn. Root toegang toestaan aan gebruikers betekent dat onbelangrijke activiteiten, zoals het toevoegen van apparaten of het instellen van netwerk interfaces, door de individuele gebruikers worden afgehandeld, waardoor systeembeheerders meer tijd hebben voor netwerk beveiliging en andere belangrijke zaken.

Aan de andere kant, leidt het geven van root toegang aan individuele gebruikers tot de volgende problemen:

- *Verkeerde machine instelling* — Gebruikers met root toegang kunnen hun machine verkeerd instellen en vereisen hulp om het probleem op te lossen. Nog erger, ze kunnen beveiligings gaten maken zonder dat ze het weten.
- *Het draaien van onveilige services* — Gebruikers met root toegang kunnen onveilige servers op hun machine draaien, zoals FTP of Telnet, waarmee potentieel hun gebruikersnamen en wachtwoorden in gevaar zijn. Deze services verzenden deze informatie in leesbare tekst over het netwerk.
- *Het uitvoeren van email bijlages als root* — Hoewel het zelden voorkomt, zijn er email virussen die effect hebben op Linux. De enigste keer dat ze echter een gevaar zijn, is als ze als de root gebruiker draaien.

2.1.4.2. Root toegang niet toestaan

Als een beheerder het om deze of andere redenen niet fijn vindt dat gebruikers als root inloggen, moet het root wachtwoord geheim gehouden worden, en moet toegang tot runlevel 1 of enkele-gebruikers mode verboden worden met boot loader wachtwoord bescherming (refereer naar *Paragraaf 2.1.2.2, "Boot loader wachtwoorden"* voor meer informatie hierover).

Tabel 2.1, "Methodes voor het onmogelijk maken van het root account" beschrijft manieren waarop een beheerder kan verzekeren dat inloggen als root niet toegestaan wordt:

Methode	Beschrijving	Effecten	Het geen gevolgen voor
Het veranderen van de root shell	Bewerk het **/etc/passwd** bestand en verander de shell van **/bin/bash** naar **/sbin/nologin**.	Voorkomt toegang naar de root shell en logt alle pogingen om toegang te krijgen. De volgende programma's worden belet om root toegang te krijgen: · **login**	Programma's die geen shell nodig hebben, zoals FTP cliënten, mail cliënten, en vele setuid programma's. De volgende programma's wordt root toegang *niet* belet: · **sudo**

Methode	Beschrijving	Effecten	Het geen gevolgen voor
		· **gdm** · **kdm** · **xdm** · **su** · **ssh** · **scp** · **sftp**	· FTP cliënten · Email cliënten
Root toegang onmogelijk maken via elk console apparaat (tty).	Een leeg **/etc/securetty** bestand voorkomt root login op elk apparaat dat aangesloten is op de computer.	Voorkomt toegang tot het root account via de console of het netwerk. De volgende programma's krijgen geen toegang tot het root account: · **login** · **gdm** · **kdm** · **xdm** · Andere netwerk services die een tty openen	Programma's die niet inloggen als root, maar beheerstaken uitvoeren met setuid of andere mechanismes. De volgende programma's wordt root toegang *niet* belet: · **su** · **sudo** · **ssh** · **scp** · **sftp**
Root SSH login niet toestaan	Bewerk het **/etc/ssh/sshd_config** bestand en zet de **PermitRootLogin** parameter op **no**.	Belet root toegang via de OpenSSH suite van gereedschappen. De volgende programma's krijgen geen toegang tot het root account: · **ssh** · **scp** · **sftp**	Dit belet alleen root toegang naar de OpenSSH suite van gereedschappen.
Gebruik PAM om root toegang naar services te beperken.	Bewerk het bestand voor de doel service in de **/etc/pam.d/** map. Wees er zeker van dat **pam_listfile.so** nodig is voor authenticatie.[1]	Belet root toegang naar netwerk services die bewust zijn van PAM. De volgende services worden belet om root toegang te krijgen: · FTP cliënten · Email cliënten · **login** · **gdm** · **kdm** · **xdm** · **ssh** · **scp** · **sftp** · Alle PAM services die bewust zijn van PAM	Programma's en services die niet bewust zijn van PAM.

[1] Refereer naar *Paragraaf 2.1.4.2.4, "Root onmogelijk maken met PAM"* voor details.

Tabel 2.1. Methodes voor het onmogelijk maken van het root account

2.1.4.2.1. De root shell onmogelijk maken

Om te voorkomen dat gebruikers rechtstreeks inloggen als root, kan de systeembeheerder de shell van het root account instellen als **/sbin/nologin** in het **/etc/passwd** bestand. Dit belet toegang tot het root account met commando's die een shell nodig hebben, zoals de **su** en **ssh** commando's.

Belangrijk

Programma's die geen toegang naar de shell nodig hebben, zoals email cliënten of het **sudo** commando, hebben nog steeds toegang naar het shell account.

2.1.4.2.2. Als root inloggen onmogelijk maken

Om de toegang naar het root account verder te beperken, kunnen beheerders het inloggen als root op de console onmogelijk maken door het **/etc/securetty** bestand te bewerken. Dit bestand bevat een lijst van alle apparaten waarop de root gebruiker in kan loggen. Als dit bestand niet bestaat, kan de root gebruiker inloggen op elk communicatie apparaat op het systeem, of dit nu een console is of een ruwe netwerk interface. Dit is gevaarlijk, omdat een gebruiker op zijn machine kan inloggen als root met Telnet, die het wachtwoord als leesbare tekst over het netwerk stuurt. Standaard staat het **/etc/securetty** bestand van Fedora de root gebruiker alleen toe om in te loggen op de console die fysiek aangekoppeld is aan de machine. Om te voorkomen dat root inlogt, verwijder je de inhoud van dit bestand door het intypen van het volgende commando:

```
echo > /etc/securetty
```

Waarschuwing

Een leeg **/etc/securetty** bestand voorkomt *niet* dat de root gebruiker inlogt op afstand met gebruik van de OpenSSH suite van gereedschappen omdat de console niet geopend wordt tot na de authenticatie.

2.1.4.2.3. Root SSH inloggen onmogelijk maken

Als root inloggen met het SSH protocol is standaard in Fedora uitgezet; als deze optie echter is aangezet, kan het weer uitgezet worden door het configuratie bestand van de SSH daemon (**/etc/ssh/sshd_config**) te bewerken. Verander de regel die er uitziet als:

```
PermitRootLogin yes
```

naar het volgende:

```
PermitRootLogin no
```

Om deze verandering effect te laten hebben moet de SSH daemon opnieuw opgestart worden. Dit kan gedaan worden met het volgende commando:

```
kill -HUP `cat /var/run/sshd.pid`
```

2.1.4.2.4. Root onmogelijk maken met PAM

PAM biedt door de **/lib/security/pam_listfile.so** module, een hoge flexibiliteit in het verbieden van specifieke accounts. De beheerder kan deze module gebruiken om te refereren naar een lijst van gebruikers die niet in mogen loggen. Hieronder is een voorbeeld van het gebruik van deze module voor de **vsftpd** FTP server in het **/etc/pam.d/vsftpd** PAM configuratie bestand (de \ karakter op het eind van de eerste regel in het volgende voorbeeld is *niet* nodig als het commando in een regel past):

```
auth required /lib/security/pam_listfile.so item=user \
sense=deny file=/etc/vsftpd.ftpusers onerr=succeed
```

Dit instrueert PAM om het **/etc/vsftpd.ftpusers** bestand te raadplegen en toegang naar de service te verbieden voor elke gebruiker in de lijst. De beheerder kan de naam van dit bestand veranderen, en kan aparte lijsten voor elke service hebben of een centrale lijst gebruiken voor het verbieden van toegang naar meerdere services.

Als de beheerder toegang wil verbieden naar meerdere services, kan een soortgelijke regel toegevoegd worden aan de PAM configuratie bestanden, zoals **/etc/pam.d/pop** en **/etc/pam.d/imap** voor mail cliënten, of **/etc/pam.d/ssh** voor SSH cliënten.

Voor meer informatie over PAM, refereer je naar *Paragraaf 2.4, “Pluggable Authentication Modules (PAM)”*.

2.1.4.3. Root toegang beperken

In plaats van toegang naar de root gebruiker helemaal te verbieden, kan de beheerder alleen toegang toe willen staan met setuid programma's, zoals **su** of **sudo**.

2.1.4.3.1. Het su commando

Als een gebruiker het **su** commando uitvoert, wordt deze gevraagd naar het root wachtwoord en, na authenticatie, krijgt deze een root shell prompt.

Zodra de gebruiker is ingelogd met het **su** commando, *is* de gebruiker de root gebruiker en heeft absolute beheerstoegang tot het systeem[3]. Bovendien ,zodra een gebruiker root wordt, is het voor hen mogelijk om het **su** commando te gebruiken om te veranderen naar elke andere gebruiker op het systeem zonder naar een wachtwoord gevraagd te worden.

Omdat dit commando zo krachtig is, zullen beheerders in een organisatie beperkingen willen opleggen voor wie toegang heeft tot dit commando.

Een van de eenvoudigste manieren om dit te doen is om gebruikers toe te voegen aan een speciale beheersgroep met de naam *wheel*. Om dit te doen, type je het volgende commando als root:

```
usermod -G wheel <gebruikersnaam>
```

In het vorige commando vervang je *<gebruikersnaam>* met de gebruikersnaam die je wilt toevoegen aan de **wheel** groep.

Je kunt ook **Gebruikersbeheerder** gebruiken om een groepslidmaatschap te veranderen. Merk op: je hebt beheersrechten nodig om dit uit te voeren.

[3] Deze toegang valt nog steeds onder de beperkingen opgelegd door SELinux, als deze is aangezet.

1. Klik op het **Systeem** menu op het Paneel, ga naar **Beheer** en klik dan op **Gebruikers en groepen** om de Gebruikersbeheerder te openen. Als alternatief type je het commando **`system-config-users`** op de shell prompt.

2. Klik op de **Gebruikers** tab en selecteer de gewenste gebruiker in de lijst van gebruikers.

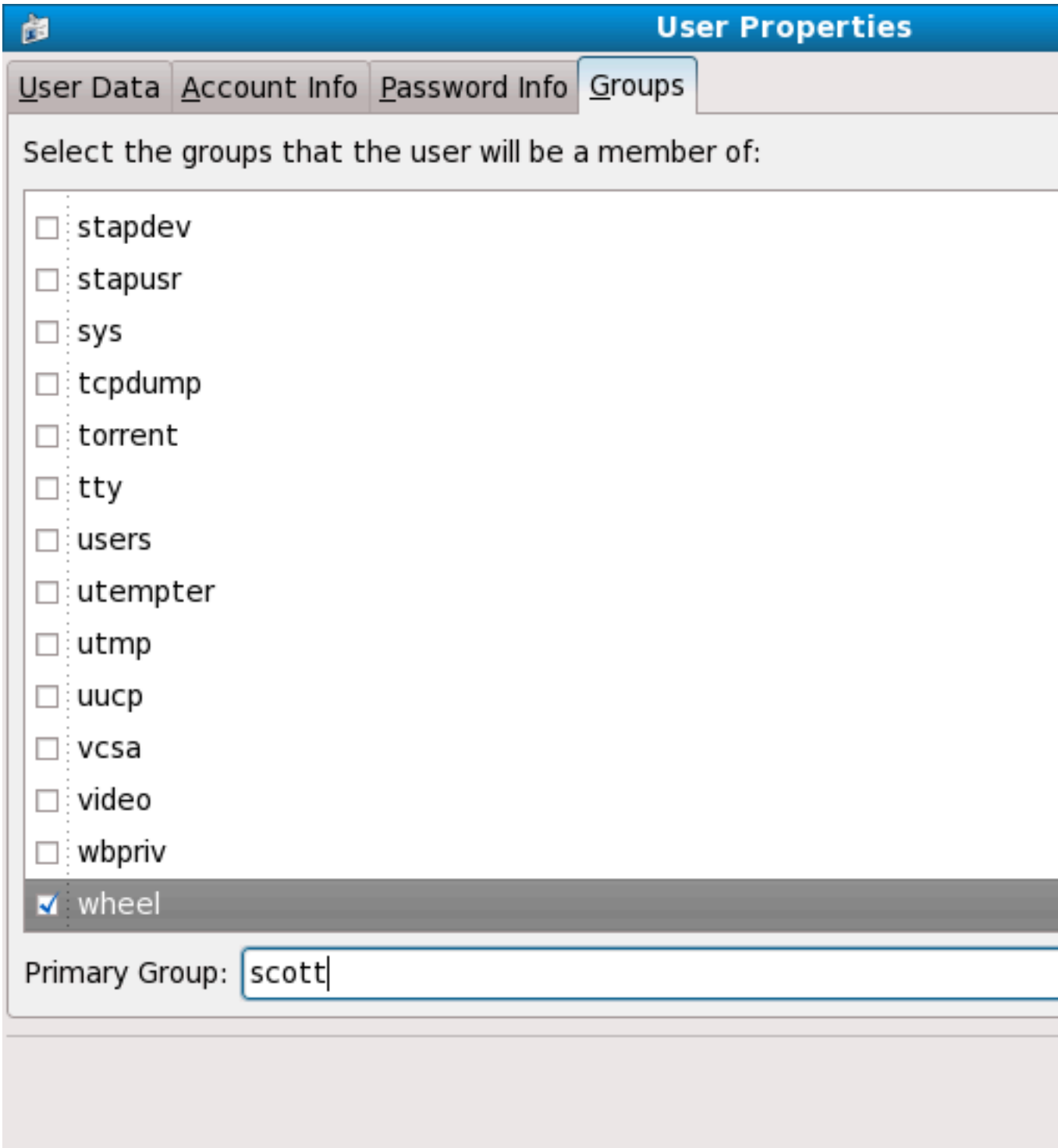
3. Klik op **Eigenschappen** op de gereedsschapbalk om de Gebruikerseigenschappen dialoog te openen (of kies **Eigenschappen** in het **Bestand** menu).

4. Klik op de **Groepen** tab, selecteer het aanvinkhokje voor de wheel groep, en klik dan op **OK**. Refereer naar *Figuur 2.2, “Gebruikers toevoegen aan de "wheel" groep.”*.

5. Open het PAM configuratiebestand voor **`su`** (**`/etc/pam.d/su`**) in een tekstbewerker en verwijder de commentaar **#** in de volgende regel:

   ```
   auth  required /lib/security/$ISA/pam_wheel.so use_uid
   ```

   Deze verandering betekent dat alleen leden van de beheersgroep `wheel` dit programma kunnen gebruiken.



Figuur 2.2. Gebruikers toevoegen aan de "wheel" groep.

Opmerking

De root gebruiker is standaard lid van de `wheel` groep.

2.1.4.3.2. Het sudo commando

Het **sudo** commando biedt een andere benadering voor het geven van beheerstoegang aan gebruikers. Als vertrouwde gebruikers een beheerscommando vooraf laten gaan door **sudo**, worden ze naar*hun eigen* wachtwoord gevraagd. Als ze authentiek verklaard zijn en aangenomen dat het commando toegestaan is, wordt daarna het beheerscommando uitgevoerd alsof ze de root gebruiker zijn.

Het basis formaat van het **sudo** commando is als volgt:

```
sudo <commando>
```

In het voorbeeld hierboven, moet <*commando*> vervangen worden door een commando dat normaal gereserveerd is voor de root gebruiker, zoals **mount**.

Belangrijk

Gebruikers van het **sudo** commando moeten er aan denken om uit te loggen voordat ze van hun machine weglopen, omdat gebruikers van sudo het commando opnieuw kunnen gebruiken binnen een periode van vijf minuten zonder dat er naar hun wachtwoord gevraagd wordt. Deze instelling kan veranderd worden met het configuratiebestand, **/etc/sudoers**.

Het **sudo** commando staat een hoge mate van flexibiliteit toe. Bijvoorbeeld, alleen gebruikers die vermeld zijn in het **/etc/sudoers** configuratiebestand hebben het recht om het **sudo** commando te gebruiken en het commando wordt uitgevoerd in de shell *van de gebruiker*, niet in een root shell. Dit betekent dat de root shell helemaal uitgezet kan worden, zoals getoond is in *Paragraaf 2.1.4.2.1, “De root shell onmogelijk maken”*.

Het **sudo** commando biedt ook een uitgebreid controle spoor. Elke succesvolle authenticatie wordt vastgelegd in het bestand **/var/log/messages** en het opgegeven commando wordt tezamen met de gebruikersnaam van de opdrachtgever weggeschreven naar het bestand **/var/log/secure**.

Een ander voordeel van het **sudo** commando is dat een beheerder verschillende gebruikers toegang kan geven tot specifieke commando's gebaseerd op hun behoefte.

Beheerder die het **sudo** configuratiebestand, **/etc/sudoers**, willen bewerken, moeten het **visudo** commando gebruiken.

Om iemand volledige beheersrechten te geven, type je **visudo** in en je voegt een regel die lijkt op de volgende toe in de gebruikersrechten specificatie sectie:

```
jan ALL=(ALL) ALL
```

Dit voorbeeld zegt dat de gebruiker, `jan`, **sudo** kan gebruiken vanaf elke host en elk commando kan uitvoeren.

Het voorbeeld hier beneden laat de fijnkorreligheid zien die mogelijk is met het instellen van **sudo**:

```
%users localhost=/sbin/shutdown -h now
```

Dit voorbeeld zegt dat elek gebruiker het commando **/sbin/shutdown -h now** mag opgeven zolang het opgegeven wordt vanaf de console.

De manual pagina voor **`sudoers`** heeft een gedetailleerde lijst van opties voor dit bestand.

2.1.5. Beschikbare netwerk services

Hoewel gebruikers toegang tot beheers controle een belangrijke zaak is voor systeembeheerders in een organisatie, is het bewaken van actieve netwerk services van levensbelang voor iedereen die een Linux systeem beheert en bedient.

Veel services in Fedora gedragen zich als netwerk servers. Als een netwerk service op een machine draait, dan luistert een server toepassing (een *daemon* genaamd) maar verbindingen op een of meer netwerk poorten. Ieder van deze servers moet behandeld worden als een potentiële toegang voor een aanval.

2.1.5.1. Risico's voor services

Netwerk services kunnen vele risico's vormen voor Linux systemen. Hieronder staat een lijst van de hoofdzaken:

- *Service weigerings aanvallen (DoS)* — Door het overspoelen van een service met verzoeken, kan een service weigerings aanval een systeem onbruikbaar maken als het probeert om ieder verzoek te verwerken.
- *Distributed Denial of Service Attack (DDoS)* — A type of DoS attack which uses multiple compromised machines (often numbering in the thousands or more) to direct a co-ordinated attack on a service, flooding it with requests and making it unusable.
- *Script kwetsbaarheids aanval* — Als een server scripts gebruikt om de acties aan de server kant uit te voeren, zoals Web servers gewoonlijk doen, kan een cracker aanvallen met onjuist geschreven scripts. Deze script kwetsbaarheids aanvallen kunnen leiden tot een buffer overloop conditie en staan de aanvaller toe om bestanden op het systeem te veranderen.
- *Buffer overloop aanvallen* — Services die verbinden met de poorten genummerd van 0 tot en met 1023 moeten draaien als een beheersgebruiker. Als de toepassing een benutbare buffer overloop heeft, kan een aanvaller toegang krijgen tot het systeem als de gebruiker die de daemon draait. Omdat benutbare buffer overlopen bestaan, gebruiken crackers automatische gereedschappen om systemen met kwetsbaarheden te identificeren, en zodra ze toegang hebben, kunnen ze geautomatiseerde rootkits gebruiken om hun toegang naar het systeem te handhaven.

Opmerking

De dreiging van buffer overloop kwetsbaarheden wordt in Fedora verlicht door *ExecShield*, een uitvoerbaar geheugen segmentatie en beschermings technologie ondersteund door x86-compatibele een- en multi-processor kernels. ExecShield vermindert het risico van buffer overloop door het verdelen van virtueel geheugen in uitvoerbare en niet-uitvoerbare segmenten. Elk programma dat probeert draaien buiten een uitvoerbaar segment (zoals kwaadwillige code ingebracht met een buffer overloop uitbuiting) veroorzaakt een segmentatie fout en stopt.

ExecShield bevat ook ondersteuning voor *No eXecute* (NX) technologie op AMD 64 platforms en *eXecute Disable* (XD) technologie op Itanium en Intel® 64 systemen. Deze technologie werkt samen met ExecShield om kwaadwillige code te beletten om te draaien in het uitvoerbare deel van virtueel geheugen met een verdeling van 4KB

uitvoerbare code, wat het risico van een aanval van onopgemerkte buffer overloop aanvallen verlaagd.

Belangrijk

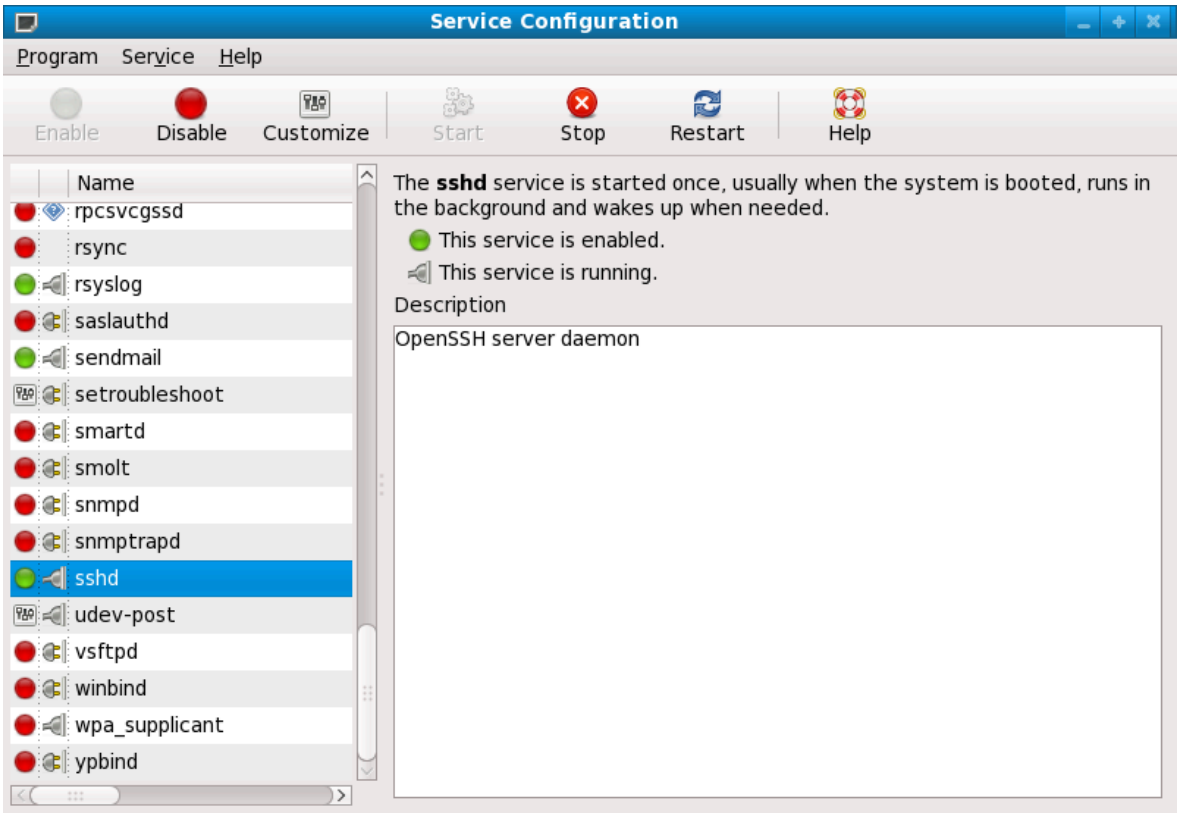
Om de blootstelling aan aanvallen over het Internet te beperken, moeten alle ongebruikte services uitgezet worden.

2.1.5.2. Het identificeren en instellen van services

Om beveiliging te verbeteren, worden de meeste netwerk services geïnstalleerd met Fedora standaard uitgezet. Er zijn echter een paar belangrijke uitzonderingen:

- **`cupsd`** — De standaard printer server in Fedora.
- **`lpd`** — Een alternatieve print server.
- **`xinetd`** — Een super server die verbindingen controleert van een reeks van ondergeschikte servers, zoals **`gssftp`** en **`telnet`**.
- **`sendmail`** — Het Sendmail *Mail Transport Agent* (MTA) is standaard aangezet, maar luistert alleen naar verbindingen van de localhost.
- **`sshd`** — De OpenSSH server, wat een veilige vervanging is voor Telnet.

Om te bepalen of deze services moeten blijven draaien, is het het beste om je gezonde verstand te gebruiken en voorzichtig te blijven. Bijvoorbeeld, als er geen printer beschikbaar is, laat dan **`cupsd`** niet draaien. Hetzelfde geldt voor **`portmap`**. Als je geen NFSv3 volumes aankoppelt of NIS (de **`ypbind`** service) gebruikt, dan moet **`portmap`** uitgezet worden.



Figuur 2.3. **Service configuratie gereedschap**

Als je niet zeker bent van het doel van een bepaalde service, laat het **Service configuratie** gereedschap een omschrijving zien met extra informatie zoals getoond in *Figuur 2.3, "**Service configuratie gereedschap**"*.

Het controleren van welke services beschikbaar zijn om te starten tijdens het opstarten van de computer is een kant van het verhaal. Je moet ook controleren welke poorten open zijn en luisteren. Refereer naar *Paragraaf 2.2.8, "Het verifiëren van welke poorten luisteren"* voor meer informatie.

2.1.5.3. Onveilige services

Potentieel is elke netwerk service onveilig. Daarom is het uitzetten van ongebruikte services zo belangrijk. Uitbuitingen voor services worden routinematig openbaar gemaakt en gecorrigeerd, wat het erg belangrijk maakt om regelmatig pakketten te vernieuwen die behoren bij een netwerk service. Refereer naar *Paragraaf 1.5, "Beveiligings vernieuwingen"* voor meer informatie.

Sommige netwerk protocollen zijn inherent onveiliger dan andere. Zij omvatten alle services die:

- *Gebruikersnamen en wachtwoorden onversleuteld over een netwerk versturen* — Vele oudere protocollen, zoals Telnet en FTP, versleutelen de authenticatie sessie niet en moeten wanneer mogelijk vermeden worden.
- *Gevoelige data onversleuteld over een netwerk versturen* — Vele protocollen verzenden data onversleuteld over het netwerk. Deze protocollen omvatten Telnet, FTP, HTTP, en SMTP. Veel netwerk bestandssystemen, zoals NFS en SMB, versturen ook informatie onversleuteld over het netwerk. Het is de verantwoordelijkheid van de gebruiker om bij het gebruik van deze protocollen het type van de verzonden data te beperken.

Geheugen dump services op afstand, zoals **netdump**, verzenden de inhoud van het geheugen onversleuteld over het netwerk. Geheugen dumps kunnen wachtwoorden bevatten of, nog erger, database ingangen en andere gevoelige informatie.

Andere services zoals **finger** en **rwhod** onthullen informatie over gebruikers van het systeem.

Voorbeelden van inherent onveilige services zijn **rlogin**, **rsh**, **telnet**, en **vsftpd**.

Alle login en shell programma's op afstand (**rlogin**, **rsh**, en **telnet**) moeten vermeden worden ten gunste van SSH. Refereer naar *Paragraaf 2.1.7, "Communicatie gereedschappen met verbeterde beveiliging"* voor meer informatie over **sshd**.

FTP is niet even inherent gevaarlijk voor de beveiliging van het systeem als shells op afstand, maar FTP servers moeten zorgvuldig ingesteld en bewaakt worden om problemen te vermijden. Refreer naar *Paragraaf 2.2.6, "FTP beveiligen"* voor meer informatie over het beveiligen van FTP servers.

Services die zorgvuldig ingesteld moeten worden en zich achter een firewall moeten bevinden omvatten:

- **finger**
- **authd** (deze werd **identd** genoemd in vorige Fedora vrijgaves.)
- **netdump**
- **netdump-server**
- **nfs**
- **rwhod**
- **sendmail**
- **smb** (Samba)
- **yppasswdd**
- **ypserv**
- **ypxfrd**

Meer informatie over het beveiligen van netwerk services is beschikbaar in *Paragraaf 2.2, "Server beveiliging"*.

De volgende paragraaf bespreekt gereedschappen om een eenvoudige firewall in te stellen

2.1.6. Persoonlijke firewalls

Als de *noodzakelijke* netwerk services zijn ingesteld, is het belangrijk om een firewall te maken.

Belangrijk

Je moet de noodzakelijke services instellen en een firewall maken *voordat* je verbindt met het Internet of elk ander netwerk.

Firewalls beletten netwerkpakketten toegang tot de netwerk interface van het systeem. Als een verzoek komt voor een poort die geblokkeerd is door een firewall, wordt het verzoek genegeerd. Als een service luistert naar een van deze geblokkeerde poorten, ontvang het deze pakketten niet en is effectief uitgezet. Daarom moet zorg worden betracht bij het instellen van een firewall om toegang te blokkeren naar poorten die niet in gebruik zijn, terwijl poorten in gebruik door ingestelde services niet geblokkeerd moeten worden.

Voor de meeste gebruikers is het beste gereedschap voor het instellen van een eenvoudige firewall het grafische firewall instel gereedschap dat met Fedora meegeleverd wordt: het **Firewall Configuration Tool** (**`system-config-firewall`**). Dit gereedschap maakt ruime **`iptables`** regels voor een algemene firewall door het gebruik van een controle paneel interface.

Refereer naar *Paragraaf 2.8.2, “Basis firewall instelling”* voor meer informatie over het gebruik van deze toepassing en zijn beschikbare opties.

Voor gevorderde gebruikers en server beheerders, is het handmatig instellen van een firewall met **`iptables`** waarschijnlijk een betere optie. Refereer naar *Paragraaf 2.8, “Firewalls”* voor meer informatie. Refereer naar *Paragraaf 2.9, “IPTables”* voor een uitgebreide gids voor het **`iptables`** commando.

2.1.7. Communicatie gereedschappen met verbeterde beveiliging

Met de de groei van de grootte en populariteit van het Internet is ook de bedreiging van communicatie onderschepping gegroeid. In de loop der jaren zijn gereedschappen ontwikkeld om communicatie die over het netwerk verstuurd wordt te versleutelen.

Fedora wordt geleverd met twee basis gereedschappen die versleutel algoritme's van hoog niveau en gebaseerd op publieke sleutels gebruiken om informatie te beschermen als het over het netwerk gaat.

- *OpenSSH* — Een vrije implementatie van het SSH protocol voor versleutelde netwerk communicatie.
- *Gnu Privacy Guard (GPG)* — Een vrije implementatie van de PGP (Pretty Good Privacy) versleutelings toepassing voor het versleutelen van data.

OpenSSH een een veiliger manier om toegang te krijgen naar een machine op afstand en vervangt oudere, onversleutelde services zoals **`telnet`** en **`rsh`**. OpenSSH bevat een netwerk service **`sshd`** genaamd en drie commandoregel cliënt toepassingen:

- **`ssh`** — Een veilige console toegang cliënt op afstand.
- **`scp`** — Een veilig kopiëer commando op afstand.
- **`sftp`** — Een veilige pseudo-ftp cliënt die interactieve bestand overdracht sessies toestaat.

Refereer naar *Paragraaf 3.6, “Beveiligde shell”* voor meer informatie over OpenSSH.

Belangrijk

Hoewel de **`sshd`** service inherent veilig is, *moet* de service up-to-date worden gehouden om beveiligings bedreigingen te voorkomen. Refereer naar *Paragraaf 1.5, “Beveiligings vernieuwingen”* voor meer informatie.

GPG is een manier om privé email communicatie te verzekeren. Het kan gebruikt worden zowel voor het emailen van gevoelige data over publieke netwerken, als het beschermen van gevoelige data op harde schijven.

2.2. Server beveiliging

Als een systeem wordt gebruikt als een server op een publiek netwerk, wordt het een doel voor aanvallen. Het versterken van het systeem en afsluiten van services is daarom van groot belang voor de systeembeheerder.

Voordat we ingaan op specifieke zaken, bekijken we eerste de volgende algemene aanwijzingen voor het verbeteren van de beveiliging van een server:

- Houd alle services bij de tijd om ze te beschermen tegen de nieuwste bedreigingen.
- Gebruik waar mogelijk veilige protocollen.
- Lever slechts een type netwerk service per machine waar mogelijk.
- Bewaak alle servers zorgvuldig voor verdachte activiteit.

2.2.1. Het beveiligen van services met TCP wrappers en xinetd

TCP wrappers bieden toegangscontrole voor een groot aantal services. De meeste moderne netwerk services, zoals SSH, Telnet, en FTP, gebruiken TCP wrappers, die de wacht houden tussen een binnenkomend verzoek en de gevraagde service.

De voordelen geboden door TCP wrappers worden versterkt als ze samengaan met **`xinetd`**, een super service die extra toegangs, logging, verbindings, omleidings, en hulpbron gebruiks controle biedt.

Opmerking

Het is een goed idee om iptables firewall regels te gebruiken samen met TCP wrappers en **`xinetd`** om redundantie te maken voor service toegangscontrole. Refereer naar *Paragraaf 2.8, “Firewalls”* meer informatie over het maken van firewalls met iptables commando's.

De volgende paragrafen veronderstellen een basis kennis van ieder onderwerp en richten zich op specifieke beveiligings opties.

2.2.1.1. Het verbeteren van beveiliging met TCP wrappers

TCP wrappers kunnen veel meer dan het weigeren van toegang tot services. Deze paragraaf laat zien hoe ze gebruikt kunnen worden om verbindings banners te sturen, te waarschuwen voor aanvallen van bepaalde hosts, en de logging functionaliteit te verbeteren. Refereer naar de **`hosts_options`** manual pagina voor informatie over de TCP wrapper functionaliteit en controle taal.

2.2.1.1.1. TCP wrappers en verbindings banners

Het laten zien van een geschikte banner als gebruikers verbinden met een service is een goede manier om potentiële aanvallers te laten weten dat de systeembeheerder waakzaam is. Je kunt

ook bepalen welke informatie over het systeem aan gebruikers gepresenteerd wordt. Om een TCP wrapper banner te maken voor een service, gebruik je de `banner` optie.

Dit voorbeeld maakt een banner voor **`vsftpd`**. Om te beginnen maak je een banner bestand. Het kan zich overal op het systeem bevinden, maar het moet dezelfde naam hebben als de daemon. Voor dit voorbeeld, wordt het bestand **`/etc/banners/vsftpd`** genoemd en bevat de volgende regel:

```
220-Hello, %c
220-All activity on ftp.example.com is logged.
220-Inappropriate use will result in your access privileges being removed.
```

Het **`%c`** symbool levert cliënt informatie, zoals de gebruikersnaam en hostnaam, of de gebruikersnaam en IP adres om de verbinding nog bedreigender te maken.

Om deze banner te laten zien voor binnenkomende verbindingen, voeg je de volgende regel toe aan het **`/etc/hosts.allow`** bestand:

```
vsftpd : ALL : banners /etc/banners/
```

2.2.1.1.2. TCP wrappers en aanval waarschuwingen

Als een bepaalde host of netwerk is ontdekt die de server aanvalt, kunnen TCP wrappers worden gebruikt om de beheerder te waarschuwen voor volgende aanvallen van die host of dat netwerk met het gebruik van de **`spawn`** instructie.

In dit voorbeeld nemen we aan dat een cracker van het 206.182.68.0/24 netwerk is ontdekt die probeerde de server aan te vallen. Plaats de volgende regel in het **`/etc/hosts.deny`** bestand om alle verbindingspogingen van dat netwerk te verbieden, en log de pogingen in een speciaal bestand:

```
ALL : 206.182.68.0 : spawn /bin/ 'date' %c %d >> /var/log/intruder_alert
```

Het **`%d`** symbool levert de naam van de service waarnaar de aanvaller toegang probeert te krijgen.

Om de verbinding toe te staan en het te loggen, plaats je de **`spawn`** instructie in het **`/etc/hosts.allow`** bestand.

Opmerking

Omdat de **`spawn`** instructie elk shell commando uitvoert, is het een goed idee om een speciaal script te maken om de beheerder te waarschuwen of een aantal commando's uit te voeren in het geval dat een bepaalde cliënt probeert met de server te verbinden.

2.2.1.1.3. TCP wrappers en verbeterde logging

Als bepaalde types verbindingen van meer belang zijn dan andere, kan het log niveau verhoogd worden voor die service door de **`severity`** optie te gebruiken.

Voor dit voorbeeld nemen we aan dat iedereen die probeert met poort 23 (Telnet) te verbinden op een FTP server een cracker is. Om dit aan te geven, plaats je een **`emerg`** vlag in de log bestanden in plaats van de standaard vlag, **`info`**, en je verbiedt de verbinding.

Om dit te doen plaats je de volgende regel in **/etc/hosts.deny**:

```
in.telnetd : ALL : severity emerg
```

Dit gebruikt de standaard **authpriv** logging faciliteit, maar verhoogt de prioriteit van de standaard waarde **info** naar **emerg**, welke de log boodschappen rechtstreeks naar de console stuurt.

2.2.1.2. Beveiliging verbeteren met xinetd

Deze paragraaf richt zich op het gebruik van **xinetd** om een valluik service in te stellen en het te gebruiken om hulpbron niveau's te controleren die beschikbaar zijn voor een gegeven **xinetd** service. Het instellen van hulpbron limieten kan helpen *Weigering van service* (DoS) aanvallen af te slaan. Refereer naar de manual pagina's voor **xinetd** en **xinetd.conf** voor een lijst van beschikbare opties.

2.2.1.2.1. Het instellen van een valluik

Een belangrijke eigenschap van **xinetd** is de mogelijkheid om hosts toe te voegen aan een globale **no_access** lijst. Hosts op die lijst wordt het verboden om latere verbindingen te maken naar services die beheerd worden door **xinetd** voor een specifieke periode of totdat **xinetd** opnieuw is gestart. Je kunt dit doen door de **SENSOR** atribuut te gebruiken. Dit is een eenvoudige manier om hosts te blokkeren die proberen om de poorten van de server te scannen.

De eerste stap in het instellen van een **SENSOR** is een service te kiezen die je niet van plan bent te gebruiken. Voor dit voorbeeld wordt Telnet gebruikt.

Bewerk het bestand **/etc/xinetd.d/telnet** en verander de `flags` regel in:

```
flags           = SENSOR
```

Voeg de volgende regel toe:

```
deny_time       = 30
```

Dit verbiedt verdere verbindings pogingen naar die poort voor die host gedurende 30 minuten. Andere mogelijke waarden voor de **deny_time** attribuut zijn FOREVER, welke het verbod laat duren totdat **xinetd** opnieuw is opgestart, en NEVER, die de verbinding toestaat en het logt.

De laatste regel moet tenslotte zijn:

```
disable         = no
```

Dit zet de valluik zelf aan.

Terwijl het gebruik van SENSOR een goede manier is om verbindingen van ongewenste hosts te detecteren en te stoppen, heeft het twee nadelen:

- Het werkt niet tegen heimelijke scans.
- Een aanvaller die weet dat een SENSOR draait kan een Weigering van service aanval opzetten tegen bepaalde hosts door hun IP adressen te vervalsen en te verbinden met de verboden poort.

2.2.1.2.2. De hulpbronnen van de server controleren

Een andere belangrijke eigenschap van **xinetd** is de mogelijkheid om hulpbron limieten in te stellen voor de services die het controleert.

Dit gebeurt met de volgende instructies:

- `cps = <number_of_connections> <wait_period>` — Beperkt de snelheid van binnenkomende verbindingen. Deze instructie heeft twee argumenten:
  - `<number_of_connections>` — Het aantal verbindingen per seconde die behandeld worden. Als de snelheid van de binnenkomende verbindingen hoger wordt, wordt de service tijdelijk uitgezet. De standaard waarde is vijftig (50).
  - `<wait_period>` — Het aantal seconden die gewacht wordt totdat de service weer aangezet wordt nadat het is uitgezet. Het standaard interval is tien (10) seconden.
- `instances = <number_of_connections>` — Specificeert het totale aantal verbindingen die toegestaan zijn voor een service. Deze instructie accepteert een geheel getal of **UNLIMITED**.
- `per_source = <number_of_connections>` — Specificeert het aantal verbindingen toegestaan naar een service voor iedere host. Deze instructie accepteert een geheel getal of **UNLIMITED**.
- `rlimit_as = <number[K|M]>` — Specificeert de hoeveelheid geheugen adres ruimte die de service kan bezetten in kilobytes of megabytes. Deze instructie accepteert een geheel getal of **UNLIMITED**.
- `rlimit_cpu = <number_of_seconds>` — Specificeert de tijd in seconden die een service magen vragen van de CPU. Deze instructie accepteert een geheel getal of **UNLIMITED**.

Het gebruiken van deze instructies kan helpen voorkomen dat een enkel **xinetd** service het systeem overspoelt, resulterende in een weigering van service.

2.2.2. Portmap beveiligen

De **portmap** service is een dynamische poort toekennings daemon voor RPC services zoals NIS en NFS. Het heeft zwakke authenticatie mechanismes en heeft de mogelijkheid om een brede reeks poorten toe te kennen aan de services die het controleert. Om deze redenen is het moeilijk te beveiligen.

Opmerking

Portmap beveiligen heeft alleen effect voor NFSv2 en NFSv3 implementaties, omdat NFSv4 het niet langer nodig heeft. Als je van plan bent om een NFSv2 of NFSv3 server te maken, is **portmap** vereist, en is de volgende paragraaf van toepassing.

Als je RPC services draait, volg dan deze basis regels.

2.2.2.1. Bescherm portmap met TCP wrappers

Het is belangrijk om TCP wrappers te gebruiken om te beperken welke netwerken of hosts toegang hebben tot de **portmap** service omdat het geen ingebouwde vorm van authenticatie bevat.

Gebruik verder *alleen* IP adressen om toegang tot de server te beperken. Vermijd het gebruik van hostnamen, omdat ze vervalst kunnen worden door DNS vergiftiging en andere methodes.

2.2.2.2. Bescherm portmap met iptables

Om verdere toegang tot de **portmap** service te beperken, is het een goed idee om iptables regels toe te voegen aan de server en de toegang te beperken tot specifieke netwerken.

Hieronder zijn twee voorbeeld iptables commando's. De eerste staat TCP verbindingen toe naar poort 111 (gebruikt door de **portmap** service) vanaf het 192.168.0.0/24 netwerk. De tweede staat TCP verbindingen toe naar dezelfde poort vanaf de localhost. Dit is nodig voor de **sgi_fam** service gebruikt door **Nautilus**. Alle andere pakketten worden genegeerd.

```
iptables -A INPUT -p tcp -s! 192.168.0.0/24 --dport 111 -j DROP
iptables -A INPUT -p tcp -s 127.0.0.1  --dport 111 -j ACCEPT
```

Om op dezelfde manier UDP verkeer te beperken, gebruik je het volgende commando.

```
iptables -A INPUT -p udp -s! 192.168.0.0/24  --dport 111 -j DROP
```

Opmerking

Refereer naar *Paragraaf 2.8, “Firewalls”* voor meer informatie over het maken van firewalls met iptables commando's.

2.2.3. Het beveiligen van NIS

Het *Network Information Service* (NIS) is een RPC service, **ypserv** genaamd, welke wordt gebruikt in combinatie met **portmap** en andere gerelateerde services om overzichten van gebruikersnamen, wachtwoorden en andere gevoelige informatie te verspreiden naar elke computer die beweert zich binnen het domein te bevinden.

Een NIS server wordt opgebouwd met verscheidene toepassingen. Ze omvatten de volgende:

- **/usr/sbin/rpc.yppasswdd** — Ook de **yppasswdd** service genoemd, deze daemon staat gebruikers toe om hun NIS wachtwoorden te veranderen.
- **/usr/sbin/rpc.ypxfrd** — Ook de **ypxfrd** service genoemd, deze daemon is verantwoordelijk voor NIS map verplaatsingen over het netwerk.
- **/usr/sbin/yppush** — Deze toepassing geeft veranderde NIS databases door aan meerdere NIS servers.
- **/usr/sbin/ypserv** — Dit is de NIS server daemon.

NIS is een beetje onveilig naar de hedendaagse standaarden. Het heeft geen host authenticatie mechanisme en verstuurt alle informatie onversleuteld over het netwerk, inclusief wachtwoord hashes. Daarom moet bijzondere zorg in acht genomen worden bij het opzetten van een netwerk dat NIS gebruikt. Dit wordt verder ingewikkeld gemaakt door het feit dat de standaard instelling van NIS inherent onveilig is.

Het wordt aanbevolen dat iedereen die van plan is een NIS server te maken, eerst de **portmap** service beveiligt zoals aangegeven in *Paragraaf 2.2.2, “Portmap beveiligen”*, en daarna de volgende zaken beschouwt, zoals netwerk planning.

2.2.3.1. Plan het netwerk zorgvuldig

Omdat NIS gevoelige informatie onversleuteld over het netwerk verstuurt, is het belangrijk dat de service achter een firewall en op een opgedeeld en veilig netwerk draait. Telkens wanneer NIS informatie over een onveilig netwerk wordt verstuurd, loopt het een risico om onderschept te worden. Zorgvuldig netwerk ontwerp kan ernstige beveiligings schendingen helpen voorkomen

2.2.3.2. Gebruik een NIS domeinnaan en hostnaam die lijkt op een wachtwoord

Elke machine binnen een NIS domein kan commando's gebruiken om informatie te achterhalen van de server zonder authenticatie, mits de gebruiker de DNS hostnaam van de server en de NIS domeinnaam weet.

Bijvoorbeeld, als iemand of een laptop computer met het netwerk verbindt of van buiten in het netwerk inbreekt (en erin slaagt om een intern IP adres te vervalsen), laat het volgende commando de **/etc/passwd** map zien:

```
ypcat -d <NIS_domein> -h <DNS_hostnaam> passwd
```

Als deze aanvaller de root gebruiker is, kan deze het **/etc/shadow** bestand verkrijgen met het volgende commando:

```
ypcat -d <NIS_domein> -h <DNS_hostnaam> shadow
```

Opmerking

Als Kerberos wordt gebruikt, wordt het **/etc/shadow** bestand niet bewaard in een NIS map.

Om toegang tot NIS mappen moeilijker te maken voor een aanvaller, maak je een DNS hostnaam bestaande uit willekeurige karakters, zoals **o7hfawtgmhwg.domain.com**. Maak op dezelfde manier een *andere* willekeurige NIS domeinnaam. Dit maakt het veel moeilijker voor een aanvaller om toegang te krijgen tot de NIS server.

2.2.3.3. Bewerk het /var/yp/securenets bestand

Als het **/var/yp/securenets** bestand leeg is of niet bestaat (zoals het geval is na een standaard installatie), luistert NIS naar alle netwerken. Een van de eerste dingen om te doen is om netmasker/netwerk paren in het bestand te plaatsen zodat **ypserv** alleen verzoeken van het juiste netwerk beantwoordt.

Hieronder staat een voorbeeld regel in een **/var/yp/securenets** bestand:

```
255.255.255.0     192.168.0.0
```

Waarschuwing

Start een NIS server nooit op voordat het **/var/yp/securenets** bestand aangemaakt is.

Deze techniek geeft geen bescherming tegen een IP adres vervalsings aanval, maar beperkt ten minste de netwerken die bediend worden door de NIS server.

2.2.3.4. Ken statische poorten toe en gebruik iptables regels

Alle servers gerelateerd aan NIS kunnen specifieke poorten toegewezen krijgen behalve voor **rpc.yppasswdd** — de daemon die gebruikers toestaat hun login wachtwoord te veranderen. Poorten toekennen aan de andere twee NIS server daemons, **rpc.ypxfrd** en **ypserv**, staat toe om firewall regels te maken om de NIS server daemons verder te beschermen tegen indringers.

Om dit te doen, voeg je de volgende rgels toe aan **/etc/sysconfig/network**:

```
YPSERV_ARGS="-p 834" YPXFRD_ARGS="-p 835"
```

De volgende iptables regels kunnen dan gebruikt worden om te forceren naar welk netwerk de server luistert op deze poorten:

```
iptables -A INPUT -p ALL -s! 192.168.0.0/24  --dport 834 -j DROP
iptables -A INPUT -p ALL -s! 192.168.0.0/24  --dport 835 -j DROP
```

Dit betekent dat de server alleen verbindingen naar poort 834 en 835 toestaat als het verzoek van het 192.168.0.0/24 netwerk komt, ongeacht het protocol.

Opmerking

Refereer naar *Paragraaf 2.8, “Firewalls”* voor meer informatie over het maken van firewalls met iptables commando's.

2.2.3.5. Gebruik Kerberos authenticatie

Een van de problemen te overwegen als NIS gebruikt wordt voor authenticatie is dat telkens als een gebruiker inlogt op een machine, een wachtwoord hash van de **/etc/shadow** map verstuurd wordt over het netwerk. Als een indringer toegang krijgt tot een NIS domein en het netwerkverkeer besnuffelt, kan deze gebruikersnamen en wachtwoord hashes verzamelen. Met genoeg tijd, kan een wachtwoord kraakprogramma zwakke wachtwoorden raden, en een aanvaller kan toegang krijgen tot een geldig account op het netwerk.

Omdat Kerberos geheime-sleutel versleuteling gebruikt, worden nooit wachtwoord hashes over het netwerk verstuurd, wat het systeem veel veiliger maakt. Refereer naar *Paragraaf 2.6, “Kerberos”* voor meer informatie over Kerberos.

2.2.4. NFS beveiligen

Belangrijk

De versie van NFS in Fedora, NFSv4, heeft de **portmap** service niet langer nodig zoals aangegeven is in *Paragraaf 2.2.2, “Portmap beveiligen”*. NFS verkeer gebruikt nu TCP in alle versies, in plaats van UDP, en vereist dit als NFSv4 gebruikt wordt. NFSv4 bevat nu Kerberos gebruiker en groep authenticatie, als onderdeel van de **RPCSEC_GSS** kernel

module. Informatie over **portmap** wordt nog steeds gegeven, omdat Fedora NFSv2 en NFSv3 ondersteunt, welke beide **portmap** nog gebruiken.

2.2.4.1. Plan het netwerk zorgvuldig

Nu NFSv4 de mogelijkheid heeft om alle informatie met gebruik van Kerberos versleuteld over een netwerk te versturen, is het belangrijk dat de service correct ingesteld wordt als het achter een firewall of in een verdeeld netwerk is. NFSv2 en NFSv3 geven data nog steeds onveilig door, en dit moet in gedachte worden gehouden. Zorgvuldig netwerk ontwerp in al deze opzichten kunnen beveiligings inbreuken helpen voorkomen.

2.2.4.2. Let op syntax fouten

De NFS server bepaalt welke bestandssystemen geëxporteerd worden en welke hosts deze mappen moeten exporteren door het raadplegen van het **/etc/exports** bestand. Let er op om geen onnodige spaties toe te voegen bij het bewerken van dit bestand.

Bijvoorbeeld, de volgende regel in het **/etc/exports** bestand deelt de map **/tmp/nfs/** met de host **bob.example.com** met lees/schrijf rechten.

```
/tmp/nfs/     bob.example.com(rw)
```

De volgende regel in het **/etc/exports** bestand, echter, deelt dezelfde map met de host `bob.example.com` met alleen-lezen rechten en deelt het met de *wereld* met lees/schrijf rechten dankzij de enkele spatie achter de hostnaam.

```
/tmp/nfs/     bob.example.com (rw)
```

Het is een goede praktijk om alle ingestelde NFS delingen te controleren met gebruik van het **showmount** commando om te verifiëren wat er gedeeld wordt:

```
showmount -e <hostnaam>
```

2.2.4.3. Gebruik de `no_root_squash` optie niet

Standaard veranderen NFS delingen de root gebruiker naar de **nfsnobody** gebruiker, een gebruikersaccount zonder rechten. Dit verandert de eigenaar van alle door root aangemaakte bestanden naar **nfsnobody**, wat verhindert dat programma's binnen gehaald worden waarbij de setuid bit gezet is.

Als **no_root_squash** wordt gebruikt, zijn root gebruikers op afstand in staat om elk bestand in het gedeelde bestandssysteem te veranderen en toepassingen besmet met een trojaan achter te laten die andere gebruikers onopzettelijk uitvoeren.

2.2.4.4. NFS firewall instelling

De voor NFS gebruikte poorten worden dynamisch toegekend door rpcbind, wat problemen kan veroorzaken bij het maken van firewall regels. Om dit proces te vereenvoudigen gebruik je het */etc/sysconfig/nfs* bestand om op te geven welke poorten gebruikt moeten worden:

- **MOUNTD_PORT** — TCP en UDP poort voor mountd (rpc.mountd)

- **STATD_PORT** — TCP en UDP poort voor status (rpc.statd)
- **LOCKD_TCPPORT** — TCP poort voor nlockmgr (rpc.lockd)
- **LOCKD_UDPPORT** — UDP poort nlockmgr (rpc.lockd)

De opgegeven poortnummers moeten niet door een andere service gebruikt worden. Stel je firewall in om de opgegeven poortnummers toe te laten, en ook TCP en UDP poort 2049 (NFS).

Voer het **rpcinfo -p** commando uit op de NFS server om te zien welke poorten en RPC programgramma's gebruikt worden.

2.2.5. De Apache HTTP server beveiligen

De Apache HTTP server is een van de meest stabiele en veilige services die meegeleverd worden met Fedora. Een groot aantal opties en technieken zijn beschikbaar om de Apache HTTP server te beveiligen — te veel om ze hier diepgaand te behandelen. De volgende paragraaf legt in het kort goede praktijken uit voor het draaien van de Apache HTTP server.

Controleer altijd dat alle scripts die op het systeem draaien werken zoals bedoeld is *voordat* je ze in gebruik neemt. Verzeker je er ook van dat alleen de root gebruiker schrijf rechten heeft in alle mappen die scripts of CGI's bevatten. Om dit te doen, voer je de volgende commando's uit als de root gebruiker:

1. ```
   chown root <map_naam>
   ```
2. ```
   chmod 755 <map_naam>
   ```

Systeembeheerders moeten opletten als de volgende configuratie opties gebruikt worden (ingesteld in **/etc/httpd/conf/httpd.conf**):

`FollowSymLinks`
: Deze instructie is standaard aangezet, dus wees voorzichtig met het maken van symbolische links naar de document root van de Web server. Bijvoorbeeld, het is een slecht idee om een symbolische link naar `/` te maken.

`Indexes`
: Deze instructie is standaard aangezet, maar kan dit kan ongewenst zijn. Om te voorkomen dat bezoekers door bestanden op de server bladeren, verwijder je deze instructie.

`UserDir`
: De `UserDir` instructie is standaard uitgezet omdat dit de aanwezigheid van een gebruikersaccount op het systeem kan bevestigen. Om het bladeren door gebruikersmappen op de server toe te staan, gebruik je de volgende instructies:

  ```
  UserDir enabled
  UserDir disabled root
  ```

  Deze instructies activeren het bladeren door gebruikersmappen voor alle gebruikersmappen anders dan **/root/**. Om gebruikers toe te voegen aan de lijst uitgezette account, voeg je een door spaties gescheiden lijst van gebruikers toe aan de `UserDir disabled` regel.

Belangrijk

Verwijder de `IncludesNoExec` instructie niet. Standaard kan de *Server-Side Includes* (SSI) module geen commando's uitvoeren. Het wordt aanbevolen dat je deze instelling niet verandert behalve als het absoluut noodzakelijk is, om dat het in potentie een aanvaller in staat stelt om commando's op het systeem uit te voeren.

2.2.6. FTP beveiligen

Het *File Transfer Protocol* (FTP) is een ouder TCP protocol ontworpen voor het overbrengen van bestanden over een netwerk. Omdat alle transacties met de server, inclusief gebruikers authenticatie, onversleuteld zijn, wordt het beschouwd als een onveilig protocol en moet het zorgvuldig ingesteld worden.

Fedora levert drie FTP servers.

- **`gssftpd`** — Een ftp daemon gebaseerd op **`xinetd`** en bewust van Kerberos, die geen authenticatie informatie over het netwerk verstuurt.
- **Red Hat Content Accelerator** (**`tux`**) — Een kernel-ruimte Web server met FTP mogelijkheden.
- **`vsftpd`** — Een op zich staande, op veiligheid gerichte uitvoering van de FTP service.

De volgende beveiligingsrichtlijnen gelden voor het instellen van de **`vsftpd`** FTP service.

2.2.6.1. FTP begroetings koptekst

Voordat een gebruikersnaam en wachtwoord verstuurd worden, krijgen alle gebruikers een begroetings koptekst. Standaard bevat deze koptekst versie informatie die nuttig is voor crackers om te proberen zwaktes in een systeem te identificeren.

Om de begroetings koptekst voor **`vsftpd`** te veranderen, voeg je de volgende instructie toe aan het **`/etc/vsftpd/vsftpd.conf`** bestand:

```
ftpd_banner=<vul_hier_begroeting_in>
```

Vervang *<vul_hier_begroeting_in>* in bovenstaande instructie met de tekst van de begroetingsboodschap.

Voor koptekksten met meerdere regels, is het het beste om een koptekst bestand te gebruiken. Om het beheer van meerdere koptekksten te vereenvoudigen, plaats je alle koptekksten in een nieuwe map met de naam **`/etc/banners/`**. Het koptekst bestand voor FTP verbindingen is in dit voorbeeld **`/etc/banners/ftp.msg`**. Hieronder is een voorbeeld hoe zo'n bestand er uit kan zien:

```
######### # Hallo, alle activiteit op ftp.example.com wordt gelogged.
 #########
```

Opmerking

Het is niet nodig om elke regel van het bestand te beginnen met **`220`** zoals opgegeven is in *Paragraaf 2.2.1.1.1, “TCP wrappers en verbindings banners”*.

Om deze begroetings koptekst te koppelen aan **vsftpd**, voeg je de volgende instructie toe aan het **/etc/vsftpd/vsftpd.conf** bestand:

```
banner_file=/etc/banners/ftp.msg
```

Het is ook mogelijk om extra kopteksten te sturen naar binnenkomende verbindingen die TCP wrappers gebruiken zoals beschreven in *Paragraaf 2.2.1.1.1, “TCP wrappers en verbindings banners”*.

2.2.6.2. Anonieme toegang

De aanwezigheid van de **/var/ftp/** map activeert het anonymous account.

De eenvoudigste manier om deze map te maken is het installeren van het **vsftpd** pakket. Dit pakket zet een mapstructuur op voor anonieme gebruikers en stelt de rechten voor mappen in op alleen-lezen voor anonieme gebruikers.

Standaard kan de anonieme gebruiker in geen enkele map schrijven.

Waarschuwing

Als anonieme toegang naar een FTP server ingesteld wordt, let er dan op waar gevoelige informatie wordt bewaard.

2.2.6.2.1. Anonieme upload

Om anonieme gebruikers toe te staan om bestanden te uploaden, wordt het aanbevolen dat een alleen-schrijven map aangemaakt wordt binnen **/var/ftp/pub/**.

Om dit te doen voer je het volgende commando uit:

```
mkdir /var/ftp/pub/upload
```

Vervolgens verander je de rechten zodat anonieme gebruikers de inhoud van de map niet kunnen bekijken:

```
chmod 730 /var/ftp/pub/upload
```

Een lang formaat inhoudslijst van de map moet moet er zo uit zien:

```
drwx-wx---    2 root     ftp          4096 Feb 13 20:05 upload
```

Waarschuwing

Beheerders die anonieme gebruikers toestaan om in mappen te lezen en te schrijven ontdekken vaak dat hun servers een repository van gestolen software worden.

Bovendien voeg je voor **vsftpd** de volgende regel toe aan het **/etc/vsftpd/vsftpd.conf** bestand:

```
anon_upload_enable=YES
```

2.2.6.3. Gebruikersaccounts

Omdat FTP voor authenticatie onversleutelde gebruikersnamen en wachtwoorden verstuurd over onveilige netwerken, is het een goed idee om systeem gebruikers toegang tot de server vanaf hun gebruikersaccount te verbieden.

Om alle gebruikersaccount in **vsftpd** uit te zetten, voeg je de volgende instructie toe aan **/etc/vsftpd/vsftpd.conf**:

```
local_enable=NO
```

2.2.6.3.1. Gebruikersaccounts beperken

Voor het uitzetten van FTP toegang voor specifieke accounts of specifieke groepen van accounts, zoals de root gebruiker en gebruikers met **sudo** rechten, is dit het eenvoudigst te doen met een PAM lijst zoals beschreven in *Paragraaf 2.1.4.2.4, “Root onmogelijk maken met PAM”*. Het PAM configuratie bestand voor **vsftpd** is **/etc/pam.d/vsftpd**.

Het is ook mogelijk om gebruikersaccounts direct binnen elke service uit te zetten.

Om een specifiek gebruikersaccount in **vsftpd** uit te zetten, voeg je de gebruikersnaam toe aan **/etc/vsftpd.ftpusers**

2.2.6.4. Gebruik TCP wrappers om toegang te controleren

Gebruik TCP wrappers om toegang tot elke FTP daemon te controleren zoals beschreven in *Paragraaf 2.2.1.1, “Het verbeteren van beveiliging met TCP wrappers”*.

2.2.7. Sendmail beveiligen

Sendmail is een Mail Transfer Agent (MTA) die het Simple Mail Transfer Protocol (SMTP) gebruikt om elektronische boodschappen te bezorgen naar andere MTA's en naar email cliënten of afleveringsagenten. Hoewel vele MTA's hun verkeer kunnen versleutelen, doen de meeste dit niet, dus het versturen van email over alle publieke netwerken wordt als een inherent onveilige manier van communicatie beschouwd.

Het wordt aanbevolen dat iedereen die van plan is een Sendmail server te maken aandacht geeft aan de volgende punten.

2.2.7.1. Het beperken van een service weigerings aanval

Door de aard van email, kan een vastbesloten aanvaller de server gemakkelijk met mail overspoelen en zo een service weigering veroorzaken. Door limieten in te stellen voor de volgende instructies in **/etc/mail/sendmail.mc**, wordt de effectiviteit van zulke aanvallen beperkt.

- **confCONNECTION_RATE_THROTTLE** — Het aantal verbindingen die de server per seconde kan ontvangen. Standaard heeft Sendmail geen limiet op het aantal verbindingen. Als een limiet ingesteld is en bereikt wordt, worden verdere verbindingen vertraagd.
- **confMAX_DAEMON_CHILDREN** — Het maximale aantal child processen die de server kan opstarten. Standaard kent Sendmail geen limiet toe aan het aantal child processen. Als een limiet ingesteld is en bereikt wordt, worden verdere verbindingen vertraagd.
- **confMIN_FREE_BLOCKS** — Het minimale aantal vrije blokken die beschikbaar moet zijn voor de server om mail te accepteren. De standaard is 100 blokken.

- **confMAX_HEADERS_LENGTH** — De maximaal aanvaardbare grootte (in bytes) van een bericht koptekst.
- **confMAX_MESSAGE_SIZE** — De maximaal aanvaardbare grootte (in bytes) voor een enkel bericht.

2.2.7.2. NFS en Sendmail

Plaats de mail spool map, **/var/spool/mail/**, nooit in een NFS gedeelde volume.

Omdat NFSv2 and NFSv3 geen controle onderhouden van gebruikers en groeps ID's, kunnen twee of meer gebruikers dezelfde UID hebben en kunnen elkaars mail ontvangen en lezen.

Opmerking

Met het gebruik van Kerberos door NFSv4 is dit hier niet het geval, omdat de **SECRPC_GSS** kernel module geen op UID gebaseerde authenticatie gebruikt. Het wordt echter nog steeds als een goede praktijk beschouwd on de mail spool map *niet* op een NFS gedeelde volume te plaatsen.

2.2.7.3. Alleen-mail gebruikers

Om uitbuitingen door lokale gebruikers van de Sendmail server te helpen voorkomen, is het het beste dat mail gebruikers alleen toegang hebben tot de Sendmail server door het gebruik van een email programma. Shell account moeten op de mail server niet toegelaten worden en alle gebruikers in het **/etc/passwd** bestand moeten ingesteld worden met **/sbin/nologin** (met de root gebruiker als mogelijke uitzondering).

2.2.8. Het verifiëren van welke poorten luisteren

Na het instellen van de netwerk services, is het belangrijk om aandacht te geven naar welke poorten van de interfaces van het netwerk van het systeem feitelijk geluisterd wordt.

Er zijn twee basis benaderingen voor het opsommen van de poorten die luisteren op het netwerk. De minst betrouwbare benadering is het bevragen van de netwerk status stack met gebruik van commando's zoals **netstat -an** of **lsof -i**. Deze methode is minder betrouwbaar omdat deze programma's niet met de machine verbinden vanaf het netwerk, maar in plaats daarvan controleren wat er op de server draait. Om deze reden zijn deze toepassingen vaak doelen voor aanvallers. Crackers proberen hun sporen te verbergen als ze ongeoorloofde poorten openen door het vervangen van **netstat** en **lsof** met hun eigen, veranderde versies.

Een meer betrouwbare manier om te controleren welke poorten op het netwerk luisteren is om een poort scanner te gebruiken zoals **nmap**.

Het volgende commando uitgevoerd vanaf de console bepaalt welke poorten luisteren naar TCP verbindingen op het netwerk:

```
nmap -sT -O localhost
```

De output van dit commando is als volgt:

```
Starting Nmap 4.68 ( http://nmap.org ) at 2009-03-06 12:08 EST
Interesting ports on localhost.localdomain (127.0.0.1):
```

```
Not shown: 1711 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
111/tcp   open  rpcbind
113/tcp   open  auth
631/tcp   open  ipp
834/tcp   open  unknown
2601/tcp  open  zebra
32774/tcp open  sometimes-rpc11
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.17 - 2.6.24
Uptime: 4.122 days (since Mon Mar  2 09:12:31 2009)
Network Distance: 0 hops
OS detection performed. Please report any incorrect results at http://
nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.420 seconds
```

Deze output laat zien dat het systeem **portmap** draait door de aanwezigheid van de `sunrpc` service. Er is echter ook een geheimzinnige service op poort 834. Om te controleren of de poort verbonden is met de officiële lijst van bekende services, type je:

```
cat /etc/services | grep 834
```

Dit commando geeft geen output terug. Dit geeft aan dat hoewel de poort zich in een gereserveerde reeks bevindt (0 tot en met 1024) en root toegang nodig heeft om geopend te worden, het niet verbonden is met een bekende service.

Vervolgens zoeken we naar informatie met het gebruik van **netstat** of **lsof** Om poort 834 te controleren met **netstat**, gebruik je het volgende commando:

```
netstat -anp | grep 834
```

Het commando geeft de volgende output terug:

```
tcp   0    0 0.0.0.0:834    0.0.0.0:*   LISTEN   653/ypbind
```

De aanwezigheid van de open poort in **netstat** is geruststellend omdat een cacker die een poort op een gekraakt systeem stiekem opent het niet zal toestaan dat dit ontdekt wordt met dit commando. De `[p]` optie laat ook het proces ID (PID) zien van de service die de poort opende. In dit geval behoort de open poort toe aan **ypbind** (NIS), wat een RPC service is die in combinatie met de **portmap** service afgehandeld wordt.

Het **lsof** commando laat soortgelijke informatie zien als **netstat** omdat het ook in staat is om open poorten te verbinden met services:

```
lsof -i | grep 834
```

Het relevante gedeelte van de output van dit commando is:

```
ypbind      653        0    7u  IPv4       1319                TCP *:834
 (LISTEN)
ypbind      655        0    7u  IPv4       1319                TCP *:834
 (LISTEN)
ypbind      656        0    7u  IPv4       1319                TCP *:834
 (LISTEN)
ypbind      657        0    7u  IPv4       1319                TCP *:834
 (LISTEN)
```

Deze gereedschappen laten veel zien over de status van services die op een machine draaien. Deze gereedschappen zijn flexibel en bieden een schat van informatie over netwerk services en instellingen. Refereer naar de manual pagina's voor **`lsof`**, **`netstat`**, **`nmap`**, en **`services`** voor meer informatie.

2.3. Eenmalig inschrijven (Single sign-on - SSO)

2.3.1. Inleiding

De Fedora SSO functionaliteit reduceert het aantal keren dat Fedora bureaublad gebruikers hun wachtwoord moeten opgeven. Verscheidene belangrijke toepassingen gebruiken hetzelfde onderliggende authenticatie en autorisatie mechanisme zodat gebruikers op Fedora in kunnen loggen met het log-in scherm, en daarna hun wachtwoord niet opnieuw hoeven op te geven. Deze toepassingen worden hieronder beschreven.

Bovendien kunnen gebruikers inloggen op hun machine zelfs als er geen netwerk is (*offline mode*) of als de netwerkverbinding onbetrouwbaar is, bijvoorbeeld draadloze toegang. In het laatste geval zullen services elegant afnemen.

2.3.1.1. Ondersteunde toepassingen

De volgende toepassingen worden op dit moment ondersteund door het verenigde log-in systeem in Fedora:

- Login
- Screensaver
- Firefox en Thunderbird

2.3.1.2. Ondersteunde authenticatie mechanismes

Fedora ondersteunt op dit moment de volgende authenticatie mechanismes:

- Kerberos naam/wachtwoord login
- Smart card/PIN login

2.3.1.3. Ondersteunde Smart Cards

Fedora is getest met de Cyberflex e-gate card en lezer, maar elke kaart die voldoet aan zowel Java card 2.1.1 als Global Platform 2.0.1 specificaties moet correct werken, evenals elke lezer die ondersteund wordt door PCSC-lite.

Fedora is ook getest met Common Access Cards (CAC). De ondersteunde lezer voor CAC is de SCM SCR 331 USB Reader.

Sinds Fedora 5.2 worden Gemalto smart cards (Cyberflex Access 64k v2, standaard met DER SHA1 waarde ingesteld als in PKCSI v2.1) ondersteund. Deze smart cards zijn nu lezer compatibel met Chip/Smart Card Interface Devices (CCID).

2.3.1.4. Voordelen van Fedora eenmalig inschrijven

Op dit moment bestaan er vele beveiligings mechanismes die een groot aantal protocollen en bewaarplaatsen voor legitimatie gebruiken. Voorbeelden zijn SSL, SSH, IPsec, en Kerberos. Fedora SSO heeft als doel om deze systemen te verenigen om de hierboven getoonde vereisten te ondersteunen. Dit betekent niet dat Kerberos vervangen wordt door X.509v3 certificaten, maar meer ze te fuseren om de last te verlichten van zowel systeem gebruikers als de beheerders die deze beheren.

Om dit doel te bereiken heeft Fedora:

- Een enkele, gedeelde instantiatie van de NSS crypto bibliotheken voor ieder operating systeem.
- Het Certificate System's Enterprise Security cliënt (ESC) met het basis operating systeem. De ESC toepassing bewaakt smart card transacties. Als het ontdekt dat een gebruiker een smart card heeft gebruikt die ontworpen was om gebruikt te worden met het Fedora Certificate System server product, laat het een gebruikers interface zien met instructies om die smart card in dienst te nemen.
- Kerberos en NSS verenigd zodat gebruikers die inloggen op het operating systeem met gebruik van een smart card ook een Kerberos legitimatie krijgen (die hen toestaat om in te loggen op de bestand server, enz.)

2.3.2. Beginnen met je nieuwe Smart Card

Voordat je jouw smart card kunt gebruiken om in te loggen op je systeem en voordeel te hebben van de verbeterde beveiligings opties die deze technologie biedt, moet je eerst een paar basis installatie en configuratie stappen uitvoeren. Deze zijn hieronder beschreven.

Opmerking

Deze paragraaf biedt een hoog-niveau overzicht om te beginnen met je smart card. Meer gedetailleerde informatie is beschikbaar in de Red Hat Certificate System Enterprise Security cliënt Guide.

1. Log in met je Kerberos naam en wachtwoord
2. Wees er zeker van dat je het **`nss-tools`** pakket geladen hebt
3. Download en installeer de root certificaten voor jouw onderneming. Gebruik het volgende commando in de root CA certificaat te installeren:

   ```
   certutil -A -d /etc/pki/nssdb -n "root ca cert" -t "CT,C,C" -i ./
   ca_cert_in_base64_format.crt
   ```

4. Controleer dat de volgende RPM's op je systeem geïnstalleerd zijn: esc, pam_pkcs11, coolkey, ifd-egate, ccid, gdm, authconfig, en authconfig-gtk.

5. Zet Smart Card login ondersteuning aan

    a. In de GNOME menu balk selecteer je Systeem->Beheer->Authenticatie

    b. Type het root wachtwoord van de machine in, indien nodig.

    c. In de Authenticatie configureren dialoog, klik je op de **Authenticatie** tab.

    d. Selecteer het **SmartCard-ondersteuning aanzetten** vakje.

    e. Klik op de **SmartCard configureren...** knop om de SmartCard-instellingen dialoog te laten zien, specificeer de gewenste instellingen:

    - **SmartCard vereist voor inloggen** — Deselecteer dit vakje. Nadat je succesvol ingelogd hebt met je smart card kun je deze optie selecteren om te voorkomen dat andere gebruikers inloggen zonder smart card.
    - **Card verwijderen actie** — Dit bepaalt wat er gebeurt als je de smart card verwijdert nadat je ingelogd hebt. De beschikbare opties zijn:
        - **Vergrendelen** — Het verwijderen de smart card blokkeert het X scherm
        - **Negeren** — Het verwijderen van de smart card heeft geen effect.

6. Als je het Online Certificate Status Protocol (OCSP) moet aanzetten, open je het **`/etc/pam_pkcs11/pam_pkcs11.conf`** bestand, en je zoekt de volgende regel:

    ```
    enable_ocsp = false;
    ```

    Verander deze waarde naar true, als volgt:

    ```
    enable_ocsp = true;
    ```

7. Neem je smart card in dienst

8. Als je een CAC card gebruikt, moet je ook de volgende stappen uitvoeren:

    a. Verander naar de root account en maak een bestand aan met de naam **`/etc/pam_pkcs11/cn_map`**.

    b. Voeg de volgende regel toe aan het **`cn_map`** bestand:

    `MY.CAC_CN.123454` -> *`mijnloginid`*

    waarin *`MY.CAC_CN.123454`*de Common Name op jouw CAC is en *`mijnloginid`* je UNIX login ID is.

9. Log uit

2.3.2.1. Foutzoeken

Als je problemen hebt om je smart card werkend te krijgen, probeer je het volgende commando om de bron van het probleem te lokaliseren:

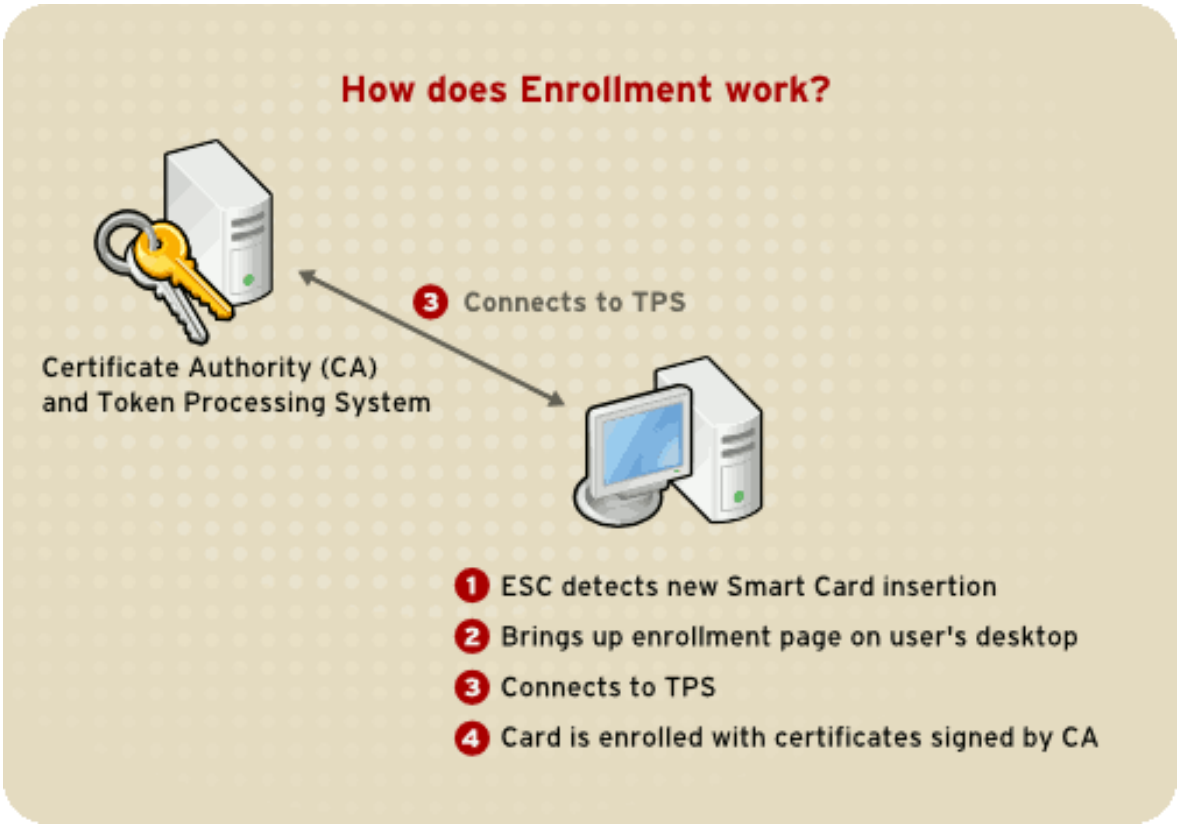
```
pklogin_finder debug
```

Als je het **`pklogin_finder`** gereedschap in de debug mode draait terwijl een in dienst genomen smart card ingebracht is, probeert het informatie te geven over de geldigheid van de certificaten, en als dat lukt probeert het een login ID te koppelen aan de certificaten op de card.

2.3.3. Hoe werkt het in gebruik nemen van een Smart Card

Smart cards zijn *in gebruik genomen* als ze een juist certificaat hebben ontvangen die is ondertekend door een geldige Certificaat Authoriteit (CA). Dit houdt verschillende stappen in, hieronder beschreven:

1. De gebruiker stopt zijn smart card in de smart card lezer van zijn werkstation. Deze actie wordt herkend door de Enterprise Security cliënt (ESC).
2. De in gebruik name pagina wordt getoond op het bureaublad van de gebruiker. De gebruiker vult de vereiste details in en het systeem van de gebruiker verbindt daarna met het Token Processing System (TPS) en de CA.
3. De TPS neemt de smart card in gebruik met een certificaat getekend door de CA.



Figuur 2.4. Hoe werkt het in gebruik nemen van een Smart Card

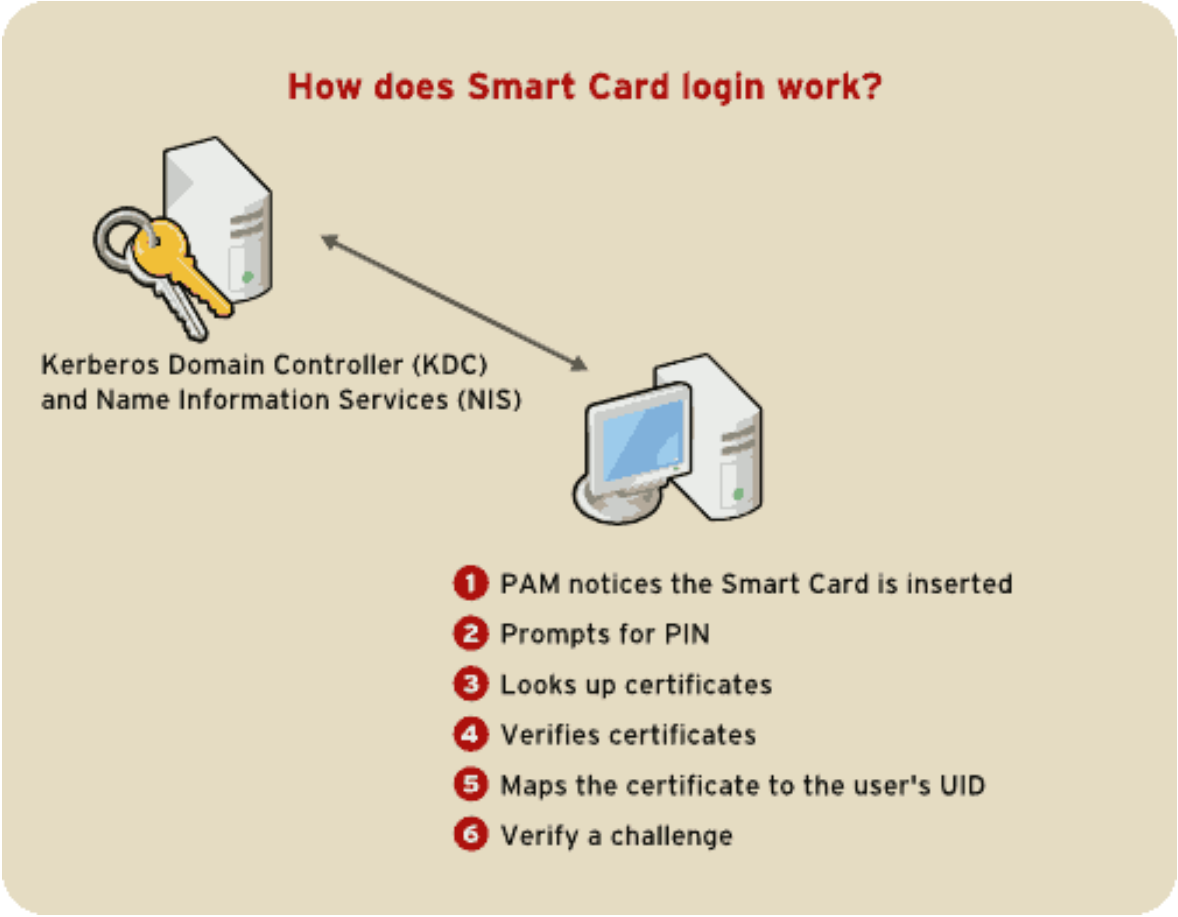
2.3.4. Hoe werkt inloggen met een Smart Card

Deze paragraaf geeft een kort overzicht van het inloggen met gebruik van een smart card.

1. als de gebruiker zijn smart card in de smart card lezer stopt, dan wordt deze actie herkend door de PAM faciliteit, die de gebruiker naar zijn PIN vraagt.

2. Het systeem zoekt dan de huidige certificaten van de gebruiker op en verifieert de geldigheid hiervan. Het certificaat wordt daarna afgebeeld op de UID van de gebruiker.

3. Dit wordt bekrachtigd door de KDC en inloggen wordt toegestaan.



Figuur 2.5. Hoe werkt inloggen met een Smart Card

Opmerking

Je kunt niet inloggen met een smart card die niet in gebruik genomen is, zelfs als deze al geformatteerd is. Je moet inloggen met een geformatteerde, in gebruik genomen card, of geen smart card gebruiken, voordat je een nieuwe card in gebruik kunt nemen.

Refereer naar *Paragraaf 2.6, “Kerberos”* en *Paragraaf 2.4, “Pluggable Authentication Modules (PAM)”* voor meer informatie over Kerberos en PAM.

2.3.5. Het instellen van Firefox om Kerberos te gebruiken voor SSO

Je kunt Firefox instellen om Kerberos te gebruiken voor eenmalig inschrijven. Om dit correct te laten werken, moet je jouw web browser instellen om jouw Kerberos legitimatie naar de juiste KDC te sturen. De volgende paragraaf beschrijft de configuratie veranderingen en andere vereisten om dit te bereiken.

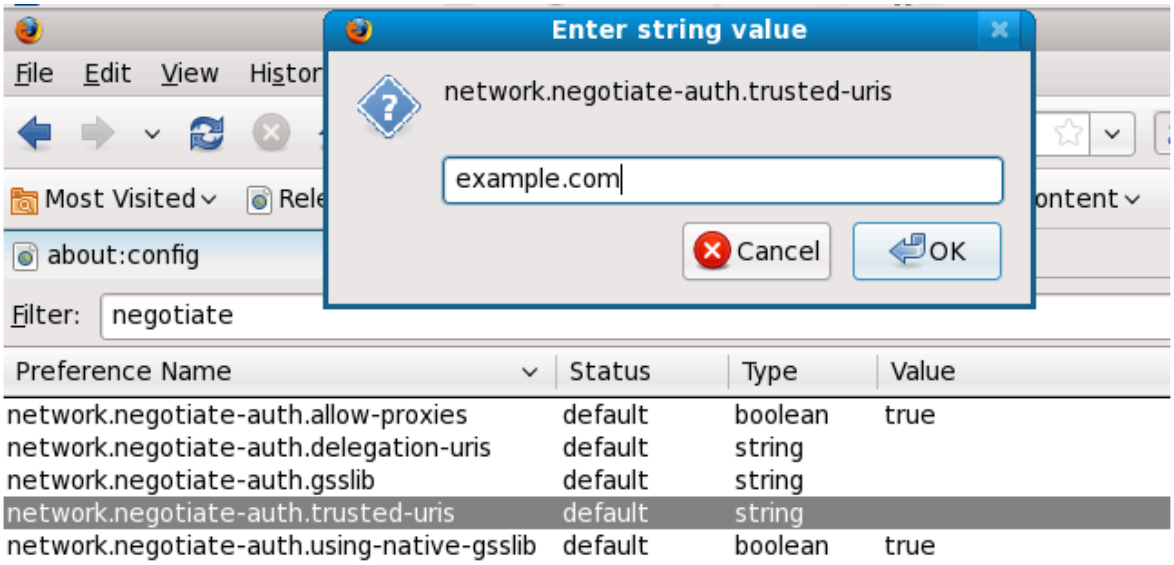
1. In de adresbalk van Firefox type je **`about:config`** om de lijst van de huidige configuratie opties te laten zien.

2. In het **Filter** veld type je **`negotiate`** om de lijst van opties te beperken.
3. Dubbel-klik op de *network.negotiate-auth.trusted-uris* regel om de *Voer stringwaarde in* dialoog op te roepen.
4. Vul de naam van het domein in voor welke je wilt authenticeren, bijvoorbeeld, *`.example.com`*.
5. Herhaal het bovenstaande voor de *network.negotiate-auth.delegation-uris* regel en gebruik hetzelfde domein.

Opmerking

Je kunt deze waarde leeg laten omdat het Kerberos kaartjes doorgeven toestaat, wat niet vereist is.

Als je deze twee configuratie opties niet ziet, is je Firefox versie misschien te oud om Negotiate authenticatie te ondersteunen, en moet je een upgrade overwegen.



Figuur 2.6. Het instellen van Firefox voor SSO met Kerberos

Je moet er nu zeker van zijn dat je Kerberos kaartjes hebt. Op de commando regel type je **`kinit`** om Kerberos kaartjes op te halen. Om de lijst van beschikbare kaartjes te laten zien, type je **`klist`**. Het volgende geeft een voorbeeld output van deze commando's:

```
[user@host ~] $ kinit
Password for user@EXAMPLE.COM:

[user@host ~] $ klist
Ticket cache: FILE:/tmp/krb5cc_10920
Default principal: user@EXAMPLE.COM

Valid starting     Expires            Service principal
10/26/06 23:47:54  10/27/06 09:47:54  krbtgt/USER.COM@USER.COM
        renew until 10/26/06 23:47:54
```

```
Kerberos 4 ticket cache: /tmp/tkt10920
klist: You have no tickets cached
```

2.3.5.1. Foutzoeken

Als je de configuratie stappen hierboven hebt opgevolgd en Negotiate authenticatie werkt niet, kun je uitgebreide logging van het authenticatie proces aanzetten. Dit kan je helpen om de oorzaak van het probleem te vinden. Om uitgebreide logging aan te zetten, gebruik je de volgende procedure:

1. Sluit alle instantiaties van Firefox.
2. Open een commando shell, en type de volgende commando's:

   ```
   export NSPR_LOG_MODULES=negotiateauth:5
   export NSPR_LOG_FILE=/tmp/moz.log
   ```

3. Start Firefox opnieuw op *in die shell*, en bezoek de website waar authenticatie eerder niet lukte. Informatie zal opgeslagen worden in **`/tmp/moz.log`**, en kan je misschien een idee geven over het probleem. Bijvoorbeeld:

   ```
   -1208550944[90039d0]: entering nsNegotiateAuth::GetNextToken()
   -1208550944[90039d0]: gss_init_sec_context() failed: Miscellaneous
    failure
   No credentials cache found
   ```

   Dit geeft aan de je geen Kerberos kaartjes hebt, en je moet **`kinit`** uitvoeren.

Als je **`kinit`** succesvol op je machine kunt uitvoeren maar je kunt geen authenticatie uitvoeren, zul je misschien zoiets als het volgende in het log bestand zien:

```
-1208994096[8d683d8]: entering nsAuthGSSAPI::GetNextToken()
-1208994096[8d683d8]: gss_init_sec_context() failed: Miscellaneous failure
Server not found in Kerberos database
```

Dit geeft in het algemeen een Kerberos configuratie probleem aan. Wees er zeker van dat je de correcte regels in het [domain_realm] gedeelte van het **`/etc/krb5.conf`** bestand hebt. Bijvoorbeeld:

```
.example.com = EXAMPLE.COM
example.com = EXAMPLE.COM
```

Als er niets in de log verschijnt dan is het mogelijk dat je achter een proxy zit, en dat de proxy de HTTP kopteksten verwijdert die nodig zijn voor Negotiate authenticatie. Als noodoplossing, kun je proberen om met de server te verbinden met gebruik van HTTPS, welke het verzoek om onveranderd door te geven zal toestaan. Ga dan verder met het foutzoeken met het logbestand, zoals hierboven aangegeven.

2.4. Pluggable Authentication Modules (PAM)

Programma's die gebruikers toegang geven tot een systeem gebruiken *authenticatie* om elkaars identiteit te controleren (dat betekent, vast te stellen dat een gebruiker is wie hij zegt te zijn).

Vroeger had ieder programma zijn eigen manier voor de authenticatie van gebruikers. In Fedora zijn veel programma's ingesteld om een centraal authenticatie mechanisme te gebruiken, *Pluggable Authentication Modules* (PAM) genaamd

PAM gebruikt een inplugbare, modulaire architectuur, welke de systeembeheerder veel flexibiliteit toestaat in het instellen van authenticatie tactieken voor het systeem.

In de meeste gevallen is het standaard PAM configuratie bestand voldoende voor een toepassing die zich bewust is van PAM. Soms is het echter nodig om een PAM configuratie bestand te bewerken. Omdat een verkeerde instelling van PAM de systeem beveiliging kan aantasten, is het belangrijk om de structuur van deze bestanden te begrijpen voordat je veranderingen maakt. Refereer naar *Paragraaf 2.4.3, "PAM configuratie bestand formaat"* voor meer informatie.

2.4.1. Voordelen van PAM

PAM biedt de volgende voordelen:

- een gemeenschappelijk authenticatie systeem dat door een groot aantal toepassingen gebruikt kan worden.
- belangrijke flexibiliteit en controle over authenticatie voor zowel systeembeheerders als toepassingsontwikkelaars.
- een enkele, volledig gedocumenteerde bibliotheek die ontwikkelaars toestaat om programma's te schrijven zonder dat ze hun eigen authenticatie schema hoeven te maken.

2.4.2. PAM configuratie bestanden

De **`/etc/pam.d/`** map bevat de PAM configuratie bestanden voor elke toepassing die zich bewust is van PAM. In eerdere versies van PAM werd het **`/etc/pam.conf`** bestand gebruikt, maar dit bestand is nu verouderd en wordt alleen gebruikt als de **`/etc/pam.d/`** map niet bestaat.

2.4.2.1. PAM service bestanden

Iedere toepassing of *service* die zich bewust is van PAM heeft een bestand in de **`/etc/pam.d/`** map. Ieder bestand in deze map heeft dezelfde naam als de service waarvoor het de toegang controleert.

Het programma dat zich bewust is van PAM is verantwoordelijk voor het bepalen van de servicenaam en het installeren van zijn eigen PAM configuratie bestand in de **`/etc/pam.d/`** map. Bijvoorbeeld, het **`login`** programma definieert zijn service naam als **`login`** en installeert het **`/etc/pam.d/login`** PAM configuratie bestand.

2.4.3. PAM configuratie bestand formaat

Elk PAM configuratie bestand bevat een groep van instructies die als volgt geformatteerd zijn:

```
<module interface>  <controle vlag>   <module naam>   <module argumenten>
```

Ieder van deze onderdelen worden in de volgende paragrafen uitgelegd.

2.4.3.1. Module interface

Op dit moment zijn er vier types PAM module interfaces beschikbaar. Elke van deze komt overeen met een verschillend aspect van het authenticatie proces:

- **auth** — Deze module geeft authenticatie voor gebruik. Bijvoorbeeld, het verzoekt en verifieert de geldigheid van een wachtwoord. Modules met deze interface kunnen ook legitimatie instellen, zoals lidmaatschap van een groep of Kerberos kaartjes.
- **account** — Deze module interface verifieert of toegang is toegestaan. Bijvoorbeeld, het kan controleren of een gebruikersaccount verlopen is en of het een gebruiker toegestaan is op een bepaalde tijd van de dag in te loggen.
- **password** — Deze module interface wordt gebruikt voor het veranderen van wachtwoorden van gebruikers.
- **session** — Deze module interface configureert en beheert sessies. Modules met deze interface kunnen ook extra taken uitvoeren die nodig zijn om toegang toe te staan, zoals het aankoppelen van de persoonlijke map van de gebruiker en het beschikbaar maken van de mailbox van de gebruiker.

Opmerking

Een individuele module kan elke of alle module interfaces hebben. Bijvoorbeeld, **pam_unix.so** biedt alle vier module interfaces.

In een PAM configuratie bestand, is de module interface gedefinieerd in het eerste veld. Bijvoorbeeld, een typische regel in een configuratie kan er als volgt uitzien:

```
auth        required        pam_unix.so
```

Dit instrueert PAM om de **auth** interface van de **pam_unix.so** module te gebruiken.

2.4.3.1.1. Module interfaces stapelen

Module interface instructies kunnen *gestapeld* worden, of op elkaar geplaatst, zodat meerdere modules tezamen gebruikt worden voor een doel. Als de controle vlag van een module de "sufficient" of "requisite" waarde gebruikt (refereer naar *Paragraaf 2.4.3.2, “Controle vlag”* voor meer informatie over deze vlaggen), dan is de volgorde waarin de modules opgesomd worden belangrijk voor het authenticatie proces.

Stapelen maakt het eenvoudig voor een beheerder om te vereisen dat specifieke condities bestaan voordat een gebruiker authenticatie wordt toegestaan. Bijvoorbeeld, het **reboot** commando gebruikt normaal meerdere gestapelde modules, zoals te zien is in zijn PAM configuratie bestand:

```
[root@MyServer ~]# cat /etc/pam.d/reboot
#%PAM-1.0
auth        sufficient        pam_rootok.so
auth        required        pam_console.so
#auth        include                system-auth
account        required        pam_permit.so
```

- De eerste regel is commentaar en wordt niet verwerkt.
- **auth sufficient pam_rootok.so** — Deze regel gebruikt de **pam_rootok.so** module om te controleren of de huidige gebruiker root is, door het verifiëren dat het UID 0 is. Als deze test succes

heeft, worden geen andere modules geraadpleegd en het commando wordt uitgevoerd. Als de test faalt, dan wordt de volgende module geraadpleegd.

- **`auth required pam_console.so`** — Deze regel gebruikt de **`pam_console.so`** module om te proberen de gebruiker te bekrachtigen. Als deze gebruiker al ingelogd is op de console, controleert **`pam_console.so`** of er een bestand is in de **`/etc/security/console.apps/`** map met dezelfde naam als de service naam (reboot). Als zo'n bestand bestaat, slaagt authenticatie en wordt de controle doorgegeven aan de volgende module.
- **`#auth include system-auth`** — Deze regel is commentaar en wordt niet verwerkt.
- **`account required pam_permit.so`** — Deze regel gebruikt de **`pam_permit.so`** module om de root gebruiker of iedereen die ingelogd is op de console toe te staan om het systeem te rebooten.

2.4.3.2. Controle vlag

Alle PAM modules genereren een succes of een faal resultaat als ze aangeroepen worden. Controle vlaggen vertellen PAM wat met dit resultaat te doen. Modules kunnen op een bepaalde manier gestapeld worden, en de controle vlaggen bepalen hoe belangrijk het succes of falen van een bepaalde module is voor het totale doel voor de authenticatie van de gebruiker voor de service.

Er zijn vier voorgedefinieerde controle vlaggen:

- **`required`** — Het resultaat van de module moet succes zijn om de authenticatie te vervolgen. Als de test op dit punt faalt, wordt dit pas bericht aan de gebruiker totdat de resultaten van alle module testen die naar die interface refereren compleet. zijn.
- **`requisite`** — Het module resultaat moet succes zijn om de authenticatie te vervolgen. Als de test echter faalt op dit punt, wordt de gebruiker hiervan op de hoogte gebracht met een boodschap welke de eerste gefaalde **`required`** *of* **`requisite`** module test aangeeft.
- **`sufficient`** — Het module resultaat wordt genegeerd als het faalt. Als echter het resultaat van een module met de vlag **`sufficient`** succes is *en* er hebben geen voorgaande modules met de vlag **`required`** gefaald, dan zijn er geen andere resultaten nodig en krijgt de gebruiker authenticatie voor de service.
- **`optional`** — Het module resultaat wordt genegeerd. Een module met de vlag **`optional`** wordt alleen nodig voor succesvolle authenticatie als geen andere module naar de interface refereert.

Belangrijk

De volgorde waarin **`required`** modules worden aangeroepen is niet kritisch. Alleen de **`sufficient`** en **`requisite`** controle vlaggen maken dat de volgorde belangrijk wordt.

Een nieuwere controle vlag syntax die een meer nauwkeurige controle toestaat is nu beschikbaar voor PAM

De **`pam.d`** manual pagina en de PAM documentatie, welke zich bevindt in de **`/usr/share/doc/pam-<versie-nummer>/`** map, waarin *<versie-nummer>* het versie nummer van PAM op jouw systeem is, beschrijft deze nieuwere syntax in detail.

2.4.3.3. Module naam

De module naam geeft PAM de naam van de inplugbare module die de opgegeven module interface bevat. In oudere versies van Fedora werd het volledige pad naar de module opgegeven in het PAM configuratie bestand. Echter sinds de komst van *multilib* systemen, die 64-bit PAM modules in de **/lib64/security/** map bewaren, wordt de map naam weggelaten omdat de toepassing gekoppeld is aan de juiste versie van **libpam**, welke de locatie van de juiste versie van de module kan bepalen.

2.4.3.4. Module argementen

PAM gebruikt *argumenten* om voor sommige modules informatie door te geven aan een inplugbare module tijdens de authenticatie

Bijvoorbeeld, de **pam_userdb.so** module gebruikt informatie die bewaard wordt in een Berkeley DB bestand voor authenticatie van de gebruiker. Berkeley DB is is een open bron database systeem ingebouwd in vele toepassingen. De module neemt een **db** argument zodat Berkeley DB weet welke database gebruikt moet worden voor de gevraagde service.

Het volgende is een typische **pam_userdb.so** regel in een PAM configuratie. De <*pad-naar-bestand*> is het volledige pad naar het Berkeley DB database bestand:

```
auth        required        pam_userdb.so db=<pad-naar-bestand>
```

Ongeldige argumenten worden *in het algemeen* genegeerd en beïnvloeden het succes of falen van een PAM module niet. Sommige modules, echter, kunnen falen op ongeldige argumenten. De meeste modules rapporteren fouten in het **/var/log/secure** bestand.

2.4.4. Voorbeeld PAM configuratie bestanden

Het volgende is een voorbeeld PAM toepassings configuratie bestand:

```
#%PAM-1.0
auth               required  pam_securetty.so
auth               required  pam_unix.so nullok
auth               required  pam_nologin.so
account               required  pam_unix.so
password        required  pam_cracklib.so retry=3
password        required  pam_unix.so shadow nullok use_authtok
session        required  pam_unix.so
```

- De eerste regel is commentaar, aangegeven door het hash teken (#) op het begin van de regel.
- Regel twee tot en met vier stapelen drie modules voor login authenticatie.

  auth required pam_securetty.so — Deze module verzekert dat *als* de gebruiker probeert in te loggen als root, de tty waarop de gebruiker ingelogd is vermeld wordt in het **/etc/securetty** bestand, *als* dat bestand bestaat.

  Als de tty niet in het bestand staat, zal elke poging om in te loggen als root falen met een `Login incorrect` boodschap.

  auth required pam_unix.so nullok — Deze module vraagt de gebruiker om een wachtwoord en controleert dan het wachtwoord met gebruik van informatie die bewaard wordt in **/etc/passwd** en, als deze bestaat, **/etc/shadow**.

  - Het argument **nullok** instrueert de **pam_unix.so** module om een leeg wachtwoord toe te staan.

- **auth required pam_nologin.so** — Dit is de laatse authenticatie stap. Het controleert of het **/etc/nologin** bestand bestaat. Als het bestaat en de gebruiker is geen root, dan faalt authenticatie.

Opmerking

In dit voorbeeld worden alle drie **auth** modules gecontroleerd, zelfs als de eerste **auth** module faalt. Dit voorkomt dat de gebruiker weet in welke fase de authenticatie faalde. Deze kennis kan in handen van een aanvaller het gemakkelijker maken om te bepalen hoe het systeem gekraakt moet worden.

- **account required pam_unix.so** — Deze module voert de nodige account verificatie uit. Bijvoorbeeld, als schaduw wachtwoorden worden gebruikt, controleert de account interface van de **pam_unix.so** module of het account verlopen is en of de gebruiker het wachtwoord niet veranderd heeft binnen de toegestane grace periode.

- **password required pam_cracklib.so retry=3** — Als een wachtwoord verlopen is, vraagt de wachtwoord component van de **pam_cracklib.so** module om een nieuw wachtwoord. Het test daarna het nieuwe wachtwoord om te zien of het eenvoudig bepaald kan worden door een kraak programma gebaseerd op een woordenboek.

  - Het argument **retry=3** specificeert dat als de test de eerste keer faalt, de gebruiker nog twee kansen heeft om een sterk wachtwoord te maken.

- **password required pam_unix.so shadow nullok use_authtok** — Deze regel specificeert dat als het programma het wachtwoord van de gebruiker verandert, moet het de **password** interface van de **pam_unix.so** module gebruiken om dit te doen.

  - Het argument **shadow** instrueert de module om schaduw wachtwoorden aan te maken als het wachtwoord van een gebruiker vernieuwd wordt.

  - Het argument **nullok** instrueert de module om de gebruiker toe te staan zijn wachtwoord te veranderen *van* een leeg wachtwoord, anders wordt een leeg wachtwoord behandeld als een account afsluiting.

  - Het laatste argument op deze regel, **use_authtok**, geeft een goed voorbeeld van het belang van volgorde bij het stapelen van PAM modules. Dit argument instrueert de module om de gebruiker niet om een nieuw wachtwoord te vragen. In plaats daarvan accepteert het elk wachtwoord dat door een vorige wachtwoord module is opgenomen. Op deze manier, moeten alle nieuwe wachtwoorden de **pam_cracklib.so** test voor veilige wachtwoorden passeren voordat ze geaccepteerd worden.

- **session required pam_unix.so** — De laatste regel instrueert de sessie interface van de **pam_unix.so** module om de sessie te beheren. Deze module logt de gebruikersnaam en het service type naar **/var/log/secure** aan het begin en het einde van iedere sessie. Deze module kan uitgebreid worden door het te stapelen met andere sessie modules voor extra functionaliteit.

2.4.5. PAM modules aanmaken

Je kunt ten alle tijde nieuwe PAM modules aanmaken of toevoegen voor het gebruik door toepassingen die zich bewust zijn van PAM.

Bijvoorbeeld, een ontwikkelaar kan een eenmalige-wachtwoord aanmaak methode maken en een PAM module schrijven om het te ondersteunen. Programma's die zich bewust zijn van PAM kunnen deze nieuwe module en wachtwoord methode onmiddellijk gebruiken zonder dat het opnieuw gecompileerd moet worden of anderszins veranderd.

Dit staat ontwikkelaars en systeembeheerders toe om authenticatie methodes voor verschillende programma's te mix-and-matchen, en ook te testen, zonder opnieuw te compileren.

Documentatie over het schrijven van modules is toegevoegd aan de **`/usr/share/doc/pam-<versie-numner>/`** map, waarin *<versie-nummer>* het versie nummer van PAM op je systeem is.

2.4.6. PAM en administratieve legitimatie opslag

Een aantal grafische administratieve gereedschappen in Fedora bieden gebruikers voor vijf minuten verhoogde rechten aan met gebruik van de **`pam_timestamp.so`** module. Het is belangrijk om te begrijpen hoe dit mechanisme werkt, omdat een gebruiker die wegloopt van een terminal terwijl **`pam_timestamp.so`** effectief is de machine open laat voor manipulatie voor iedereen met fysieke toegang tot de console.

In het PAM tijdsstempel schema, vraagt de grafische administratieve toepassing als het het opgestart wordt de gebruiker naar het root wachtwoord. Als de gebruiker gemachtigd is, maakt de **`pam_timestamp.so`** module een tijdsstempel bestand aan. Standaard wordt dit aangemaakt in de **`/var/run/sudo/`** map. Als het tijdsstempel bestand al bestaat, vragen grafische administratieve programma's niet naar een wachtwoord. In plaats daarvan ververst de **`pam_timestamp.so`** module het tijdsstempel bestand, en reserveert een extra vijf minuten van ongehinderde administratieve toegang.

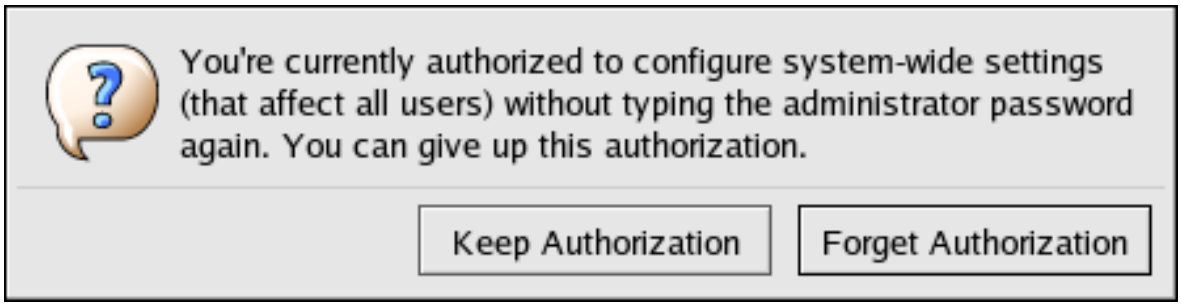
Je kunt de actuele status van het tijdsstamp bestand verifieren door het **`/var/run/sudo/<gebruiker>`** bestand te bekijken. Voor het bureaublad, is het relevante bestand **`unknown:root`**. Als het aanwezig is en zijn tijdsstempel minder dan vijf minuten oud, zijn de legitimaties geldig.

Het bestaan van het tijdstempel bestand wordt aangegeven door een authenticatie icoon, welke verschijnt in het mededelingengebied van het paneel.

Figuur 2.7. De authenticatie icoon

2.4.6.1. Het tijdstempel bestand verwijderen

Voordat je een console verlaat waar een PAM tijdstempel actief is, wordt het aanbevolen dat het tijdstempel bestand vernietigd wordt. Om dit te doen in een grafische omgeving, klik je op de authenticatie icoon op het paneel. Dit laat een dialoog scherm verschijnen. Klik op de **Authorisatie vergeten** knop om de actieve tijdsstempel bestand te vernietigen.



Figuur 2.8. Authenticatie opzeggen dialoog

Je moet op het volgende verdacht zijn met betrekking tot het tijdsstempel bestand:

- Als je op afstand ingelogd bent op het systeem met **ssh**, gebruik je het **/sbin/pam_timestamp_check -k root** commando om het tijdsstempel bestand te vernietigen.
- Je moet het **/sbin/pam_timestamp_check -k root** commando van dezelfde terminal uitvoeren als waarvan de toepassing met extra rechten hebt opgestart.
- Je moet ingelogd zijn als de gebruiker de die **pam_timestamp.so** module origineel aanriep om het **/sbin/pam_timestamp_check -k** commando te gebruiken. Log niet in als de root gebruiker om dit commando te gebruiken.
- Als je de legitimaties op het bureaublad wilt vernietigen (zonder de **Authorisatie vergeten** actie op het icoon te gebruiken), gebruik je het volgende commando:

  ```
  /sbin/pam_timestamp_check -k root </dev/null >/dev/null 2>/dev/null
  ```

  Als je dit commando niet gebruikt zal alleen de legitimaties (als ze er zijn) verwijderen van de pty waarop de het commando uitvoert.

Refereer naar de **pam_timestamp_check** manual pagina voor meer informatie over het vernietigen van het tijdsstempel bestand met gebruik van **pam_timestamp_check**.

2.4.6.2. Algemene pam_timestamp instructies

De **pam_timestamp.so** module accepteert verscheidene instructies. De volgende zijn de twee meest gebruikte opties:

- **timestamp_timeout** — Specificeert de periode (in seconden) waarvoor het tijdsstempel geldig is. De standaard waarde is 300 (vijf minuten).
- **timestampdir** — Specificeert de map waarin het tijdsstempel bestand bewaard wordt. De standaard waarde is **/var/run/sudo/**.

Refereer naar *Paragraaf 2.8.9.1, “geïnstalleerde firewall documentatie”* voor meer informatie over het controleren van de **pam_timestamp.so** module.

2.4.7. PAM en apparaat eigendom

In Fedora kan de eerste gebruiker die inlogt op de fysieke console van de machine bepaalde apparaten manipuleren en bepaalde taken uitvoeren die normaal voorbehouden zijn aan de root gebruiker. Dit wordt gecontroleerd door een PAM module met de naam **pam_console.so**.

2.4.7.1. Apparaat eigendom

Als een gebruiker inlogt op een Fedora systeem, wordt de **pam_console.so** module aangeroepen door **login** of de grafische login programma's, **gdm**, **kdm**, en **xdm**. Als deze gebruiker de eerste gebruiker is die inlogt op de fysieke console — waarnaar gerefereerd wordt als de *console gebruiker* — geeft de module de gebruiker het eigendom van een aantal apparaten die normaal eigendom zijn van root. De console gebruiker heeft deze apparaten in eigendom totdat de laatste sessie van die gebruiker beëindigd wordt. Nadat deze gebruiker uitgelogd is, vervalt het eigendom van de apparaten terug aan de root gebruiker.

De betroffen apparaten zijn, onder andere, geluidskaarten, schijfstations, en CD-ROM stations.

Deze voorziening staat een lokale gebruiker toe om deze apparaten te manipuleren zonder root toegang, wat normale taken eenvoudiger maakt voor de console gebruiker.

Je kunt de lijst van apparaten die gecontroleerd worden door **pam_console.so** veranderen door de volgende bestanden te bewerken:

- **/etc/security/console.perms**
- **/etc/security/console.perms.d/50-default.perms**

Je kunt de permissies veranderen van andere apparaten dan die vermeld zijn in de bovengenoemde bestanden, of de opgegeven standaarden veranderen. In plaats van het veranderen van het **50-default.perms** bestand, moet je een nieuw bestand aanmaken (bijvoorbeeld, ***xx*-name.perms**) en de vereiste veranderingen aanbrengen. De naam van het nieuwe standaard bestand moet beginnen met een nummer groter dan 50 (bijvoorbeeld, **51-default.perms**). Dit zal de standaarden in het **50-default.perms** bestand overschrijven.

Waarschuwing

Als het **gdm**, **kdm**, of **xdm**display beheerder configuratie bestand veranderd is om gebruikers op afstand toe te staan om en te loggen *en* de host is ingesteld om te draaien in runlevel 5, wordt het aanbevolen om de **<console>** en **<xconsole>** instructies in **/etc/security/console.perms** te veranderen naar de volgende waarden:

```
<console>=tty[0-9][0-9]* vc/[0-9][0-9]* :0\.[0-9] :0
<xconsole>=:0\.[0-9] :0
```

Dit voorkomt dat gebruikers op afstand toegang krijgen tot apparaten en beperkte toepassingen op de machine.

Als het **gdm**, **kdm**, of **xdm** display beheerder configuratie bestand is veranderd om gebruikers op afstand toe te staan in te loggen *en* de host is ingesteld op een runlevel voor meerdere gebruikers anders dan 5, wordt het aanbevolen om de **<xconsole>** instructie helemaal te verwijderen en de **<console>** instructie te veranderen naar de volgende waarde:

```
<console>=tty[0-9][0-9]* vc/[0-9][0-9]*
```

2.4.7.2. Toepassing toegang

De console gebruiker heeft ook toegang tot bepaalde programma's ingesteld voor gebruik in de **/etc/security/console.apps/** map.

Deze map bevat configuratie bestanden die de console gebruiker toestaan om bepaalde toepassingen in **/sbin** en **/usr/sbin** uit te voeren.

Deze configuratie bestanden hebben dezelfde naam als de toepassingen die ze instellen.

Een belangrijke groep van toepassingen waarnaar de console gebruiker toegang heeft zijn drie programma's die het systeem uitzetten of opnieuw opstarten.

- **/sbin/halt**
- **/sbin/reboot**
- **/sbin/poweroff**

Omdat dit toepassingen zijn die bewust zijn van PAM, roepen zij de **pam_console.so** module aan als een vereiste voor gebruik.

Refereer naar *Paragraaf 2.8.9.1, “geïnstalleerde firewall documentatie”* voor meer informatie.

2.4.8. Extra hulpbronnen

De volgende hulpbronnen leggen meer methodes uit voor het gebruik en instellen van PAM. Naast deze hulpbronnen, lees je ook de PAM configuratie bestanden op het systeem om beter te begrijpen hoe ze opgebouwd zijn.

2.4.8.1. Geïnstalleerde PAM documentatie

- Aan PAM gerelateerde manual pagina's — Verscheidene manual pagina's bestaan voor verschillende toepassingen en configuratie bestanden betrokken bij PAM. De volgende lijst laat een aantal van de meer belangrijke manual pagina's zien.

  Configuratie bestanden

  - **pam** — Goede inleidings informatie over PAM, inclusief de structuur en doel van de PAM configuratie bestanden.

    Merk op dat deze manual pagina zowel **/etc/pam.conf** als individuele configuratie bestanden in de **/etc/pam.d/** map bespreekt. Standaard gebruikt Fedora de individuele configuratie bestanden in de **/etc/pam.d/** map en negeert **/etc/pam.conf** zelfs als deze bestaat.

  - **pam_console** — Beschrijft het doel van de **pam_console.so** module. Het beschrijft ook de juiste syntax voor een regel in een PAM configuratie bestand.

  - **console.apps** — Beschrijft het formaat en de opties beschikbaar in het **/etc/security/console.apps** configuratie bestand, welke definieert welke toepassingen toegekend door PAM toegankelijk zijn voor de console gebruiker.

  - **console.perms** — Beschrijft het formaat en de opties beschikbaar in het **/etc/security/console.perms** configuratie bestand, welke permissies toegekend door PAM aan de console gebruiker beschrijft.

  - **pam_timestamp** — Beschrijft de **pam_timestamp.so** module.

- **/usr/share/doc/pam-<*versie-nummer*>** — Bevat een *System Administrators' Guide*, een *Module Writers' Manual*, en de *Application Developers' Manual*, en ook een kopie van de PAM standaard, DCE-RFC 86.0, waarin <*versie-nummer*> het versie nummer van PAM is.

- **/usr/share/doc/pam-<*versie-nummer*>/txts/README.pam_timestamp** — Bevat informatie over de **pam_timestamp.so** PAM module, waarin <*versie-nummer*> het versie nummer van PAM is.

2.4.8.2. Nuttige PAM websites

- *http://www.kernel.org/pub/linux/libs/pam/* — De hoofd distributie website voor het Linux-PAM project, deze bevat informatie over verscheidene PAM modules, een FAQ, en extra PAM documentatie.

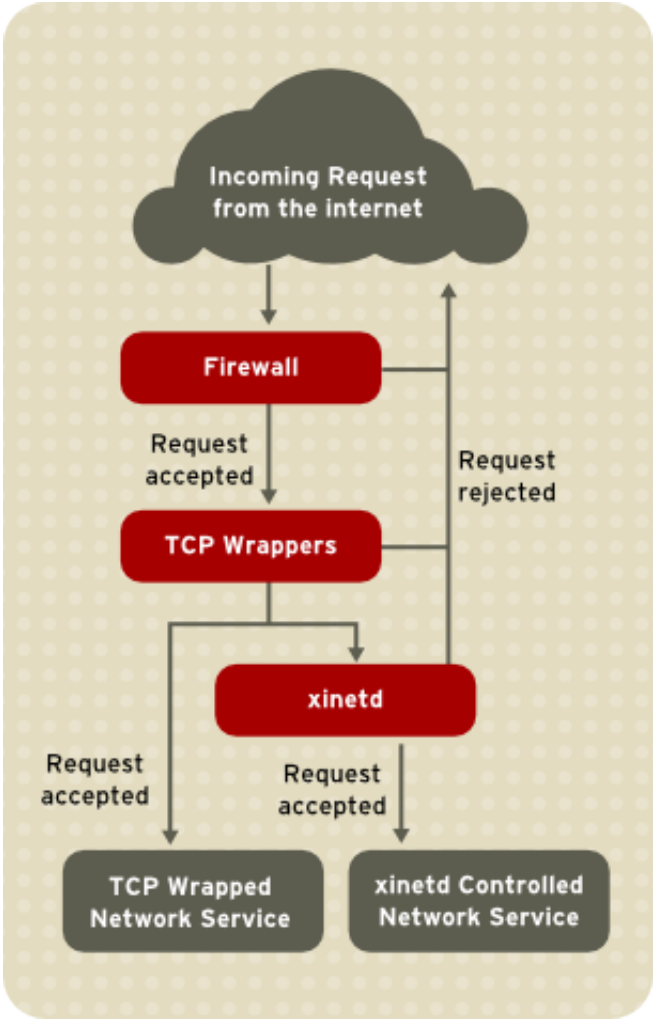
Opmerking

De documentatie op bovengenoemde website is voor de laatste vrijgegeven upstream versie van PAM en hoeft niet 100% correct te zijn voor de PAM versie in Fedora.

2.5. TCP wrappers en xinetd

Het controleren van toegang tot netwerk services is een van de belangrijkste beveiligings taken voor de beheerder van een server. Fedora biedt verscheidene gereedschappen voor dit doel. Bijvoorbeeld, een op **iptables** gebaseerde firewall filtert netwerkpakketten uit die niet welkom zijn in de netwerk stack van de kernel. Voor netwerk services die het gebruiken, voegen *TCP Wrappers* een extra beschermingslaag toe door het definiëren van welke hosts wel of geen verbinding kunnen maken met "*wrapped*" netwerk services. Een van deze wrapped netwerk services is de `xinetd` *super server*. Deze server wordt een super server genoemd omdat het de verbindingen controleert van een subset van netwerk services en toegangs controle verder verbetert.

Figuur 2.9, “Toegangs controle voor netwerk services” is een basis afbeelding die laat zien hoe deze gereedschappen samenwerken om netwerk services te beschermen.



Figuur 2.9. Toegangs controle voor netwerk services

Dit hoofdstuk richt zich op de rol van TCP wrappers en `xinetd` bij het controleren van toegang tot netwerk services en bespreekt hoe deze gereedschappen gebruikt kunnen worden om zowel logging als gebruiks beheer te verbeteren. Refereer naar *Paragraaf 2.9, "IPTables"* voor informatie over het gebruik van firewalls met **`iptables`**.

2.5.1. TCP wrappers

Het TCP wrappers pakket (**`tcp_wrappers`**) wordt standaard geïnstalleerd en biedt op host gebaseerde toegangscontrole voor netwerk services. Het belangrijkste onderdeel van het pakket is de **`/usr/lib/libwrap.a`** bibliotheek. In het algemeen is een service gewrapt met TCP een service die gecompileerd is met de **`libwrap.a`** bibliotheek.

Als er geprobeerd wordt een verbinding te maken met een service die gewrapt is met TCP, refereert de service eerst naar de toegangsbestanden van de host (**`/etc/hosts.allow`** en **`/etc/hosts.deny`**) om te bepalen of het de cliënt wel of niet toegestaan is om verbinding te maken. In de meeste gevallen gebruikt het daarna de syslog daemon (`syslogd`) om de naam van de aanvragende cliënt en de gewenste service naar **`/var/log/secure`** of **`/var/log/messages`** te schrijven.

Als het een cliënt toegestaan is om te verbinden, geeft TCP wrappers de controle van de verbinding over aan de gevraagde service en neemt verder geen deel aan de communicatie tussen de cliënt en de server.

Naast toegangscontrole en logging, kan TCP wrappers commando's uitvoeren interactief met de cliënt voor het verbieden of het overgeven van de controle over de verbinding aan de gevraagde service.

Omdat TCP wrappers een waardevolle bijdrage geeft aan het arsenaal van gereedschappen voor elke server beheerder, zijn de meeste services van Fedora gekoppeld aan de **`libwrap.a`** bibliotheek. Sommige van die toepassingen zijn `/usr/sbin/sshd`, **`/usr/sbin/sendmail`**, en `/usr/sbin/xinetd`.

Opmerking

Om te bepalen of een netwerk service binair gekoppeld is aan **`libwrap.a`**, type je het volgende commando in als de root gebruiker:

```
ldd <binaire-naam> | grep libwrap
```

Vervang *<binaire-naam>* met de naam van het netwerk service binaire bestand.

Als het commando direct naar de prompt terug komt zonder output, dan is de netwerk service *niet* gekoppeld aan **`libwrap.a`**.

Het volgende voorbeeld geeft aan dat `/usr/sbin/sshd` gekoppeld is met **`libwrap.a`**:

```
[root@myServer ~]# ldd /usr/sbin/sshd | grep libwrap
        libwrap.so.0 => /lib/libwrap.so.0 (0x00655000)
[root@myServer ~]#
```

2.5.1.1. Voordelen van TCP wrappers

TCP wrappers bieden de volgende voordelen boven andere netwerk service controle technieken:

- *Transparant voor zowel de cliënt als de gewrapte netwerk service* — Zowel de verbindende cliënt als de gewrapte netwerk service hebben niet in de gaten dat TCP wrappers gebruikt wordt. Geldige gebruikers worden gelogd en verbonden met de gevraagde service terwijl verbindingen van verbannen cliënten falen.
- *Centraal beheer van meerdere protocollen* — TCP wrappers werkt onafhankelijk van de netwerk service die ze beschermen, wat toestaat dat veel server toepassingen gezamenlijk een aantal toegangs controle configuratie bestanden delen, wat het beheer eenvoudiger maakt.

2.5.2. Configuratie bestanden voor TCP wrappers

Om te bepalen of het een cliënt toegestaan is met een service te verbinden, refereren TCP wrappers naar de volgende twee bestanden, gewoonlijk aangegeven als *hosts toegangs* bestanden:

- **`/etc/hosts.allow`**
- **`/etc/hosts.deny`**

Als een service die gewrapt is met TCP een verzoek van een cliënt ontvangt, voert het de volgende twee stappen uit:

1. *Het refereert naar* ***/etc/hosts.allow****.* — De service die gewrapt is met TCP gaat regel voor regel door het **/etc/hosts.allow** bestand en past de eerste regel toe die voor die service opgegeven is. Als het een overeenkomende regel vindt, dan wordt de verbinding toegestaan. Indien niet, dan gaat het verder met de volgende.

2. *Het refereert naar* ***/etc/hosts.deny****.* — De service die gewrapt is met TCP gaat regel voor regel door het **/etc/hosts.deny** bestand. Als het een overeenkomende regel vindt, dan wordt de verbinding verboden. Indien niet, dan wordt de toegang naar de service toegestaan.

Het volgende zijn belangrijke punten om te overwegen wanneer TCP wrappers gebruikt moeten worden om netwerk service te beschermen:

- Omdat de toegangs regels in **hosts.allow** eerst toegepast worden, hebben ze voorrang boven regels opgegeven in **hosts.deny**. Als daarom toegang tot een service toegestaan wordt in **hosts.allow**, zal een regel die toegang weigert naar dezelfde service in **hosts.deny** genegeerd worden.
- De regels in ieder bestand worden van boven naar beneden gelezen en de eerste overeenkomende regel voor een gegeven service is de enigste die toegepast wordt. De volgorde van de regels is uiterst belangrijk.
- Als in beide bestanden geen regels voor de service gevonden worden, of een of beide bestanden bestaan niet, dan wordt toegang tot de service verleend.
- Services met TCP wrappers slaan de regels van de hosts toegangs bestanden niet op, dus elke verandering in **hosts.allow** of **hosts.deny** heeft onmiddellijk effect, zonder het opnieuw moeten starten van de netwerk services.

Waarschuwing

Als de laatste regel van een hosts toegangs bestand niet een nieuwe regel karakter is (gemaakt door te duwen op de **Enter** toets), zal de laatste regel van dat bestand falen en er wordt een fout gelogd in **/var/log/messages** of **/var/log/secure**. Dit is ook het geval voor een regel die meerdere lijnen bevat zonder gebruik te maken van de backslash karakter. Het volgende voorbeeld is het relevante deel van een log boodschap voor het falen van een regel voor een van deze fouten.

```
warning: /etc/hosts.allow, line 20: missing newline or line too
 long
```

2.5.2.1. Tpegangs regels formatteren

Het formaat voor zowel **/etc/hosts.allow** als **/etc/hosts.deny** is identiek. Elke regel moet op zijn eigen lijn zijn. Lege lijnen of lijnen die beginnen met een hash (#) worden genegeerd.

Elke regel gebruikt het volgende basis formaat om toegang tot netwerk service te controleren:

```
<daemon lijst>: <cliënt lijst> [: <optie>: <optie>: ...]
```

- *<daemon lijst>* — Een door komma's gescheiden lijst van proces namen (*niet* service namen) of de ALL wildcard. De daemon lijst accepteert ook operatoren (refereer naar *Paragraaf 2.5.2.1.4, "Operatoren"*) om grotere flexibiliteit toe te staan.

- *<cliënt lijst>* — Een door komma's gescheiden lijst van hostnamen, host IP adressen, speciale patronen, of wildcards welke de hosts identificeren voor wie de regel geldt. De cliënt lijst accepteert ook operatoren getoond in *Paragraaf 2.5.2.1.4, “Operatoren”* om grotere flexibiliteit toe te staan.
- *<optie>* — Een optionele actie of een door dubbelepunten gescheiden lijst van acties die uitgevoerd worden als de regel op gang komt.

Opmerking

Meer informatie over de speciale terminologie hierboven kan elders in deze gids gevonden worden:

- *Paragraaf 2.5.2.1.1, “Wildcards”*
- *Paragraaf 2.5.2.1.2, “Patronen”*
- *Paragraaf 2.5.2.2.4, “Uitbreidingen”*
- *Paragraaf 2.5.2.2, “Optie velden”*

Het volgende is een basis voorbeeld hosts toegangs regel:

```
vsftpd : .example.com
```

Deze regel instrueert de TCP wrapper om te kijken naar verbindingen naar de ftp daemon (`vsftpd`) vanaf elke host in het `example.com` domein. Als deze regel verschijnt in **`hosts.allow`** wordt de verbinding toegestaan. Als deze regel verschijnt in **`hosts.deny`** wordt de verbinding verboden.

De volgende voorbeeld hosts toegangs regel is complexer en gebruikt twee optie velden:

```
sshd : .example.com  \ : spawn /bin/echo `/bin/date` access denied>>/var/
log/sshd.log \ : deny
```

Merk op dat ieder optie veld voorafgegaan wordt door de backslash (\). Het gebruik van de backslash voorkomt het falen van de regel door zijn lengte.

Deze voorbeeld regel zegt dat als er een verbinding naar de SSH daemon (`sshd`) wordt geprobeert vanaf een host in het `example.com` domein, het **`echo`** commando wordt uitgevoerd om de poging toe te voegen aan een speciaal logbestand, en daarna de verbinding te verbieden. Omdat de optionele **`deny`** instructie wordt gebruikt, zal deze regel toegang verbieden zelfs als het verschijnt in het **`hosts.allow`** bestand. Refereer naar *Paragraaf 2.5.2.2, “Optie velden”* voor een meer uitgebreide beschrijving van de beschikbare opties.

2.5.2.1.1. Wildcards

Wildcards staan TCP wrappers toe om groepen van daemons of hosts eenvoudiger te kunnen vergelijken. Ze worden meestal gebruikt in het cliënt lijst veld van toegangs regels.

De volgende wildcards zijn beschikbaar:

- `ALL` — Komt overeen met alles. Kan gebruikt worden voor zowel de daemon lijst als de cliënt lijst.

- `LOCAL` — Komt overeen met elke host die geen punt (.) bevat, zoals localhost.
- `KNOWN` — Komt overeen met elke host waarvan de hostnaam en hostadres bekend zijn of waar de gebruiker bekend is.
- `UNKNOWN` — Komt overeen met elke host waarvan de hostnaam of hostadres onbekend zijn of waar de gebruiker onbekend is.
- `PARANOID` — Komt overeen met elke host waarvan de hostnaam niet overeenkomt met het host adres.

Belangrijk

De `KNOWN`, `UNKNOWN`, en `PARANOID` wildcards moeten zorgvuldig gebruikt worden, omdat ze bouwen op een werkende DNS server voor hun juiste werking. Elke onderbreking van naam resolutie kan legitieme gebruikers verhinderen om toegang te krijgen tot een service.

2.5.2.1.2. Patronen

Patronen kunnen gebruikt worden in het cliënt veld van toegangs regels om nauwkeurige groepen van cliënt hosts op te geven.

Het volgende is een lijst van algemene patronen voor het cliënt veld:

- *Hostnaam beginnend met een punt (.)* — Het plaatsen van een punt aan het begin van een hostnaam komt overeen met alle hosts die de getoonde onderdelen van de naam gemeen hebben. Het volgende voorbeeld is van toepassing voor iedere host binnen het `example.com` domein:

```
ALL : .example.com
```

- *IP adres dat eindigt met een punt (.)* — Het plaatsen van een punt aan het einde van een IP adres komt overeen met alle hosts die de eerste numerieke groepen van een IP adres gemeen hebben. Het volgende voorbeeld is van toepassing voor elke host binnen het `192.168.x.x` netwerk:

```
ALL : 192.168.
```

- *IP adres/netmasker paar* — Netmasker uitdrukkingen kunnen ook gebruikt worden als een patroon om toegang naar een specifieke groep IP adressen te controleren. Het volgende voorbeeld is van toepassing voor iedere host met een adres reeks van `192.168.0.0` tot en met `192.168.1.255`:

```
ALL : 192.168.0.0/255.255.254.0
```

Belangrijk

Als je werkt in de IP4v adres ruimte dan wordt de adres/prefix lengte (*prefixlen*) paar declaraties (CIDR notation) niet ondersteund. Alleen IPv6 regels gebruiken dit formaat.

- *[IPv6 adres]/prefixlen paar* — [net]/prefixlen paren kunnen ook gebruikt worden om toegang tot een bepaalde groepen van IPv6 adressen te controleren. Het volgende voorbeeld zal

van toepassing zijn voor elke host met een adres reeks van `3ffe:505:2:1::` tot en met `3ffe:505:2:1:ffff:ffff:ffff:ffff`:

```
ALL : [3ffe:505:2:1::]/64
```

- *De asterisk (*)* — Asteriks kunnen gebruikt worden om overeen te komen met hele groepen van hostnamen of IP adressen, zolang ze niet worden vermengd in een cliënt lijst met andere patroontypes. Het volgende voorbeeld is van toepassing op elke host binnen het `example.com` domein:

```
ALL : *.example.com
```

- *De slash (/)* — Als een cliënt lijst begint met een slash, dan wordt het behandeld als een bestandsnaam. Dit is nuttig als het nodig is dat regels grote aantallen hosts opgeven. Het volgende voorbeeld refereert TCP wrappers naar het **`/etc/telnet.hosts`** bestand voor alle Telnet verbindingen:

```
in.telnetd : /etc/telnet.hosts
```

Andere, minder vaak gebruikte, patronen worden ook geaccepteerd door TCP wrappers. Refereer naar de **`hosts_access`** man 5 pagina voor meer informatie.

Waarschuwing

Wees erg voorzichtig met het gebruik van hostnamen en domeinnamen. Aanvallers kunnen een groot aantal trucjes gebruiken om nauwkeurige naam resolutie te omzeilen. Bovendien zal een onderbreking van de DNS sevice gemachtigde gebruikers beletten om netwerk services te gebruiken. Het is daarom het beste om, waar mogelijk, IP adressen te gebruiken.

2.5.2.1.3. Portmap en TCP wrappers

De implementatie van TCP wrappers in **`portmap`** ondersteunt geen host opzoeken, wat betekent dat **`portmap`** geen hostnamen kan gebruiken om hosts te identificeren. Als gevolg hiervan moeten toegangs controle regels voor portmap in **`hosts.allow`** of **`hosts.deny`** IP adressen gebruiken, of het sleutelwoord `ALL` voor het specificeren van hosts.

Veranderingen in **`portmap`** toegangs controle regels hebben misschien niet meteen effect. Het kan nodig zijn om de **`portmap`** service opnieuw op te moeten starten.

Veel gebruikt services, zoals NIS en NFS, hangen af van **`portmap`**, dus wees bedacht op deze beperkingen.

2.5.2.1.4. Operatoren

Op dit moment accepteren toegangs controle regels een operator, `EXCEPT`. Het kan gebruikt worden in zowel de daemon lijst als de cliënt lijst van een regel.

De `EXCEPT` operator staat specifieke uitzonderingen toe voor brede overeenkomsten binnen dezelfde regel.

In het volgende voorbeeld uit een **`hosts.allow`** bestand, wordt het aan alle `example.com` hosts toegestaan om naar alle services te verbinden behalve `cracker.example.com`:

```
ALL: .example.com EXCEPT cracker.example.com
```

In een ander voorbeeld uit een **`hosts.allow`** bestand, kunnen cliënten uit het `192.168.0.`*x* netwerk alle services gebruiken behalve FTP:

```
ALL EXCEPT vsftpd: 192.168.0.
```

Opmerking

Organisatorisch is het vaak eenvoudiger om het gebruik van EXCEPT operatoren te vermijden. Dat staat beheerders toe om snel de betreffende bestanden door te nemen om te zien welke host wel of geen toegang hebben tot services, zonder rekening hoeven te houden met de EXCEPT operatoren.

2.5.2.2. Optie velden

Naast de basis regels die toegang toestaan en verbieden ondersteunt de Fedora implementatie van TCO wrappersw ook uitbreidingen in toe toegangs controle taal met behulp van *optie velden*. Door optie velden in host toegangs regels te gebruiken, kunnen systeembeheerders een groot aantal taken uitvoeren, zoals het veranderen van het log gedrag, de toegangscontrole versterken, en shell commando's opstarten.

2.5.2.2.1. Logging

Optie velden laten beheerders op eenvoudige manier de log mogelijkheden en het prioriteits niveau van een regel veranderen door de `severity` instructie te gebruiken.

In het volgende voorbeeld worden verbindingen naar de SSH daemon van elke host in het `example.com` domein gelogd naar de standaard `authpriv` `syslog` voorziening (omdat er geen voorzienings waarde is opgegeven) met een prioriteit van `emerg`:

```
sshd : .example.com : severity emerg
```

Het is ook mogelijk om een voorziening op te geven met gebruik van de `severity` optie. Het volgende voorbeeld logt elke SSH verbindings poging van het `example.com` domein naar de `local0` voorziening met een prioriteit van `alert`:

```
sshd : .example.com : severity local0.alert
```

Opmerking

In de praktijk werkt dit voorbeeld niet totdat de syslog daemon (`syslogd`) is ingesteld om te loggen naar de **`local0`** voorziening. Refereer naar de **`syslog.conf`** manual pagina voor informatie over het instellen van aangepaste log voorzieningen.

2.5.2.2.2. Toegangs controle

Optie velden staan beheerders ook toe om hosts expliciet toe te staan of te verbieden in een enkele regel door het toevoegen van de `allow` of `deny` instructie als de laatste optie.

Bijvoorbeeld, de volgende twee regels laten SSH verbindingen toe van `cliënt-1.example.com`, maar verbieden verbindingen van `cliënt-2.example.com`:

```
sshd : cliënt-1.example.com : allow
sshd : cliënt-2.example.com : deny
```

Door het toestaan van toegangs contole op per-regel basis, staat het optie veld beheerders toe om alle toegangs regels te verzamelen in een enkel bestand: of **`hosts.allow`** of **`hosts.deny`**. Sommige beheerders vinden dit een eenvoudiger manier voor het organiseren van toegangs regels.

2.5.2.2.3. Shell commando's

Optie velden staan toegangs regels toe om shell commando's op te starten met behulp van de volgende twee instructies:

- **`spawn`** — Start een shell commando op als een child proces. Deze instructie kan taken uitvoeren zoals het gebruiken van **`/usr/sbin/safe_finger`** om meer informatie te krijgen over de cliënt die een verbinding vraagt of het maken van speciale log bestanden met gebruik van het **`echo`** commando.

  In het volgende voorbeeld worden cliënten die proberen toegang te krijgen tot de Telnet services vanaf het `example.com` domein stilletjes gelogd naar een speciaal bestand:

  ```
  in.telnetd : .example.com \
          : spawn /bin/echo `/bin/date` from %h>>/var/log/telnet.log \
          : allow
  ```

- **`twist`** — Vervangt de gevraagde service met het opgegeven commando. Deze instructie wordt vaak gebruikt om vallen op te zetten voor indringers (ook "honing pot" genoemd). Het kan ook gebruikt worden om boodschappen naar de verbindende cliënt te sturen. De **`twist`** instructie moet aan het einde van de regel lijn gebruikt worden.

  In het volgende voorbeeld, krijgen cliënten die proberen toegang te krijgen naar FTP services vanaf het `example.com` domein een boodschap met gebruik van het **`echo`** commando:

  ```
  vsftpd : .example.com \
          : twist /bin/echo "421 This domain has been black-listed. Access
   denied!"
  ```

Voor meer informatie over shell commando opties, refereer je naar de **`hosts_options`** manual pagina.

2.5.2.2.4. Uitbreidingen

Uitbreidingen die tezamen met de **`spawn`** en **`twist`** instructies gebruikt worden, bieden informatie over de cliënt, server en de betrokken processen.

De volgende lijst laat de ondersteunde uitbreidingen zien:

- %a — Geeft het IP adres van de cliënt terug.
- %A — Geeft het IP adres van de server terug.
- %c — Geeft verschillende informatie over de cliënt terug, zoals de gebruikersnaam en hostnaam, of de gebruikersnaam en het IP adres.
- %d — Geeft de daemon proces naam terug.
- %h — Geeft de hostnaam van de cliënt terug (of IP adres, als de hostnaam niet beschikbaar is).
- %H — Geeft de hostnaam van de server terug (of IP adres, als de hostnaam niet beschikbaar is).
- %n — Geeft de hostnaam van de cliënt terug. Als deze niet beschikbaar is, wordt `unknown` afgedrukt. Als de hostnaam en het hostadres van de cliënt niet overeenkomen, wordt `paranoid` afgedrukt.
- %N — Geeft de hostnaam van de server terug. Als deze niet beschikbaar is, wordt Als wordt `unknown` afgedrukt. Als de hostnaam en het hostadres van de server niet overeenkomen, wordt `paranoid` afgedrukt.
- %p — Geeft het proces ID van de daemon terug.
- %s — Geeft verscheidene soorten server informatie terug, zoals het daemon proces, en de host of IP adres van de server.
- %u — Geeft de gebruikersnaam van de cliënt terug. Als deze niet beschikbaar is, wordt `unknown` afgedrukt.

De volgende voorbeeld regel gebruikt een uitbreiding tezamen met het **spawn** commando om de cliënt host te identificeren in een speciaal log bestand.

Als verbindingen naar de SSH daemon (`sshd`) worden geprobeerd vanaf een host in het `example.com` domein, wordt het **echo** commando uitgevoerd om de poging te loggen, inclusief de hostnaam van de cliënt (door de %h uitbreiding te gebruiken) naar een speciaal bestand:

```
sshd : .example.com  \
        : spawn /bin/echo `/bin/date` access denied to %h>>/var/log/
sshd.log \
        : deny
```

Vergelijkbaar kunnen uitbreidingen gebruikt worden om boodschappen naar de cliënt persoonlijk te maken. In het volgende voorbeeld, worden cliënten die proberen toegang te krijgen tot FTP services vanaf het `example.com` domein verteld dat ze verbannen zijn van de server:

```
vsftpd : .example.com \
: twist /bin/echo "421 %h has been banned from this server!"
```

Voor een volledige uitleg over de beschikbare uitbreidingen, en extra toegangs controle opties, refereer je naar sectie 5 van de manual pagina's voor **hosts_access** (**man 5 hosts_access**) en de manual pagina voor **hosts_options**.

Refereer naar *Paragraaf 2.5.5, “Extra hulpbronnen”* voor meer informatie over TCP wrappers.

2.5.3. xinetd

Het `xinetd` daemon is een met TCP gewrapte *super service* welke toegang controleert naar een subset van populaire netwerk services, inclusief FTP, IMAP, en Telnet. Het biedt ook service specifieke configuratie opties voor toegangs controle, verbeterde logging, verbinding, omleiding, en hulpbron gebruiks controle.

Als een cliënt probeert te verbinden met een netwerk service die gecontroleerd wordt door `xinetd`, ontvangt de super service het verzoek en controleert of er TCP wrapper toegangs regels zijn.

als toegang toegestaan wordt, verifieert `xinetd` dat de verbinding toegestaan is onder de de eigen toegangs regels van die service. Het controleert ook of de service meer hulpbronnen toegewezen kan krijgen en of dit niet verboden wordt door een van de aangemaakte regels.

Als aan al deze voorwaarden voldaan wordt (dat betekent, toegang naar de service is toegestaan; de service heeft zijn hulpbronnen limiet niet bereikt; en de service verbreekt geen van de opgegeven regels), dan start `xinetd` een instantiering van de gevraagde service en geeft de controle over de verbinding hieraan over. Nadat de verbinding gelegd is, neemt `xinetd` verder geen deel aan de communicatie tussen de cliënt en de server.

2.5.4. xinetd configuratie bestanden

De cofiguratie bestanden voor `xinetd` zijn de volgende:

- **`/etc/xinetd.conf`** — Het globale `xinetd` configuratie bestand.
- **`/etc/xinetd.d/`** — De map die alle service specifieke bestanden bevat.

2.5.4.1. Het /etc/xinetd.conf bestand

Het **`/etc/xinetd.conf`** bestand bevat algemene configuratie instellingen die effect hebben voor iedere service onder de controle van `xinetd`. Het wordt gelezen als de `xinetd` service wordt opgestart, dus om veranderingen in de configuratie effect te laten krijgen, moet je de `xinetd` service opnieuw opstarten. Het volgende is een voorbeeld **`/etc/xinetd.conf`** bestand:

```
defaults
{
        instances               = 60
        log_type                = SYSLOG        authpriv
        log_on_success          = HOST PID
        log_on_failure          = HOST
        cps                     = 25 30
}
includedir /etc/xinetd.d
```

Deze regels controleren de volgende aspecten van `xinetd`:

- `instances` — Specificeert het maximum aantal gelijktijdige verzoeken dat `xinetd` kan verwerken.
- `log_type` — Stelt `xinetd` in om de **`authpriv`** log voorziening te gebruiken, welke log ingangen naar het **`/var/log/secure`** bestand schrijft. Toevoegen van een instructie zoals `FILE /var/log/xinetdlog` zal een aangepast log bestand met de naam **`xinetdlog`** aan maken in de **`/var/log/`** map.

- `log_on_success` — Stelt `xinetd` in om succesvolle verbindingspogingen te loggen. Standaard worden het IP adres van de host op afstand en het proces ID van de server die het verzoek verwerkt opgeschreven.
- `log_on_failure` — Stelt `xinetd` in om gefaalde verbindingspogingen of verboden verbindingen te loggen
- `cps` — Stelt `xinetd` in om niet meer dan 25 verbindingen per seconde toe te staan aan elke service. Als deze limiet wordt overschreden, wacht de service gedurende 30 seconden.
- `includedir` **`/etc/xinetd.d/`** — voegt options toe die zijn opgegeven in service specifieke configuratie bestanden in de **`/etc/xinetd.d/`** map. Refereer naar *Paragraaf 2.5.4.2, "De /etc/xinetd.d/ map"* voor meer informatie.

Opmerking

Vaak worden zowel de `log_on_success` als de `log_on_failure` instellingen in **`/etc/xinetd.conf`** verder aangepast in de service specifieke configuratie bestanden. Daarom kan er meer informatie verschijnen in het log bestand van een gegeven service dan wordt aangegeven in het **`/etc/xinetd.conf`** bestand. Refereer naar *Paragraaf 2.5.4.3.1, "Logging opties"* voor meer informatie.

2.5.4.2. De /etc/xinetd.d/ map

De **`/etc/xinetd.d/`** map bevat de configuratie bestanden voor elke service die beheerd wordt door `xinetd` en de namen van de bestanden komen overeen met die van de service. Zoals met **`xinetd.conf`**, wordt deze map gelezen als de `xinetd` service wordt opgestart. Om veranderingen effect te laten hebben, moet de beheerder de `xinetd` service opnieuw opstarten.

Het formaat van de bestanden in de **`/etc/xinetd.d/`** map gebruikt dezelfde conventie als **`/etc/xinetd.conf`**. De belangrijkste reden dat de instelling voor elke service in een apart bestand wordt bewaard is om aanpassingen eenvoudiger te maken zonder effect te hebben op andere services.

Om te begrijpen hoe de structuur van deze bestanden is, beschouw je het **`/etc/xinetd.d/krb5-telnet`** bestand:

```
service telnet
{
        flags           = REUSE
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/kerberos/sbin/telnetd
        log_on_failure  += USERID
        disable         = yes
}
```

Deze regels controleren verschillende aspecten van de **`telnet`** service:

- `service` — Specificeert de service naam, gewoonlijk een van degene opgegeven in het **`/etc/services`** bestand.

- `flags` — Stelt een aantal attributen in voor de verbinding. `REUSE` instrueert `xinetd` om de socket voor een Telnet verbinding te hergebruiken.

Opmerking

De `REUSE` vlag is verouderd. Alle service gebruiken nu de `REUSE` vlag impliciet.

- `socket_type` — Stel de netwerk socket type in naar `stream`.
- `wait` — Specificeert of de service enkel-threaded (`yes`) of multi-threaded (`no`) is.
- `user` — Specificeert onder welk gebruikers ID het proces draait.
- `server` — Specificeert welk binaire programma uitgevoerd moet worden.
- `log_on_failure` — Specificeert logging parameters voor `log_on_failure` bovenop die als ingesteld zijn in **`xinetd.conf`**.
- `disable` — Specificeert of de service uitgezet (`yes`) of aangezet (`no`) is.

Refereer naar de **`xinetd.conf`** manual pagina voor meer informatie over deze opties en hun gebruik.

2.5.4.3. xinetd configuratie bestanden veranderen

Een aantal instructies is beschikbaar voor service de beschermd worden door `xinetd`. Deze paragraaf laat een aantal veel gebruikte opties zien.

2.5.4.3.1. Logging opties

De volgende opties zijn beschikbaar voor zowel **`/etc/xinetd.conf`** als de service specifieke bestanden in de **`/etc/xinetd.d/`** map.

De volgende lijst laat een aantal vaak gebruikte logging opties zien:

- `ATTEMPT` — Logt het feit dat een gefaalde poging is gedaan (`log_on_failure`).
- `DURATION` — Logt de tijdsduur dat de service is gebruikt door een systeem op afstand (`log_on_success`).
- `EXIT` — Logt de exit status of terminal signaal van de service (`log_on_success`).
- `HOST` — Logt het IP adres van de host op afstand (`log_on_failure` en `log_on_success`).
- `PID` — Logt het proces ID van de server die het verzoek ontvangt (`log_on_success`).
- `USERID` — Logt de gebruiker op afstand met de methode gedefinieerd in RFC 1413 voor alle multi-threaded services (`log_on_failure` en `log_on_success`).

Voor een complete lijst van alle loggings opties, refereer je naar de **`xinetd.conf`** manual pagina.

2.5.4.3.2. Toegangs controle opties

Gebruikers van `xinetd` services kunnen kiezen om de TCP wrapper host toegangsregels te gebruiken, toegangs controle te bieden met de `xinetd` configuratie bestanden, of een combinatie van beide. Refereer naar *Paragraaf 2.5.2, "Configuratie bestanden voor TCP wrappers"* voor meer informatie over TCP wrapper host toegangs controle bestanden.

Deze paragraaf bespreekt het gebruik van `xinetd` om toegang tot de services te controleren.

Opmerking

In tegenstelling tot TCP wrappers, hebben veranderingen in toegangs controle pas effect als de `xinetd` beheerder de `xinetd` service opnieuw opstart.

En ook in tegenstelling tot TCP wrappers, heeft toegangscontrole met `xinetd`alleen effect voor services die gecontroleerd worden door `xinetd`.

De `xinetd` hosts toegangs controle verschilt van de methode die gebruikt wordt met TCP wrappers. Terwijl TCP wrappers alle toegangs instellingen in twee bestanden plaatst, **`/etc/hosts.allow`** en **`/etc/hosts.deny`**, wordt de toegangs controle van `xinetd` gevonden in het configuratie bestand van elke service in de **`/etc/xinetd.d/`** map.

De volgende host toegangs opties worden ondersteund door `xinetd`:

- `only_from` — Staat alleen de opgegeven hosts het gebruik van de service toe.
- `no_access` — Staat de opgegeven hosts het gebruik van de service niet toe.
- `access_times` — Specificeert het tijdsslot waarbinnen een bepaalde service gebruikt mag worden. Het tijdsslot moet in 24-uur notatie opgegeven worden, UU:MM-UU:MM.

De `only_from` en`no_access` opties kunnen een lijst van IP adressen of hostnamen gebruiken, of kunnen een heel netwerk specificeren. Net als TCP wrappers, kan het combineren van `xinetd` toegangs controle met verbeterde loggings instelling de beveiliging verbeteren door het blokkeren van verzoeken van verbannen hosts terwijl iedere verbindings poging uitvoerig opgeslagen wordt.

Bijvoorbeeld, het volgende **`/etc/xinetd.d/telnet`** bestand kan gebruikt worden om Telnet toegang vanaf een bepaalde netwerk groep te blikkeren en het tijdsslot te beperken waarbinnen zelfs goedgekeurde gebruikers in kunnen loggen:

```
service telnet
{
        disable         = no
        flags           = REUSE
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/kerberos/sbin/telnetd
        log_on_failure  += USERID
        no_access       = 172.16.45.0/24
        log_on_success  += PID HOST EXIT
        access_times    = 09:45-16:15
}
```

Als in dit voorbeeld een cliënt systeem van het `10.0.1.0/24` netwerk, zoals `10.0.1.2`, toegang tot de Telnet service probeert te krijgen, ontvang het de volgende boodschap:

```
Connection closed by foreign host.
```

Bovendien worden hun inlog pogingen als volgt in **`/var/log/messages`** gelogd:

```
Sep  7 14:58:33 localhost xinetd[5285]: FAIL: telnet address
 from=172.16.45.107
Sep  7 14:58:33 localhost xinetd[5283]: START: telnet pid=5285
 from=172.16.45.107
Sep  7 14:58:33 localhost xinetd[5283]: EXIT: telnet status=0 pid=5285
 duration=0(sec)
```

Als TCP wrappers tesamen met `xinetd` toegangs controle gebruikt wordt, is het belangrijk om de relatie tussen de twee toegangs controle mechanismes te begrijpen.

Het volgende is de volgorde van gebeurtenissen doorlopen door `xinetd` als een cliënt een verbinding aanvraagt:

1. De `xinetd` daemon krijgt toegang tot de TCP wrappers host toegangs regels met gebruik van een **`libwrap.a`** bibliotheek aanroep. Als een weigerings regel overeenkomt met de cliënt, wordt de verbinding verbroken. Als een toestemmings regel overeenkomt met de cliënt, wordt de verbinding doorgegeven aan `xinetd`.

2. De `xinetd` daemon controleert zijn eigen toegangs controle regels zowel voor de `xinetd` service als de gevraagde service. Als een weigerings regel overeen komt met de cliënt, wordt de verbinding verbroken. Anders start `xinetd` een instantiatie van de gevraagde service op en geeft de controle over de verbinding door aan die service.

Belangrijk

Let op bij het gebruik van TCP wrappers tesamen met `xinetd` toegangs regels. Een verkeerde instelling kan ongewenste effecten hebben.

2.5.4.3.3. Verbindings en omleidings opties

De service configuratie bestanden voor `xinetd` ondersteunen het verbinden van de service aan een IP adres en het omleiden van binnenkomende verzoeken voor die service naar een ander IP adres, hostnaam of poort.

Verbinden wordt gecontroleerd door de `bind` optie in de service specifieke configuratie bestanden en verbindt de service met het IP adres van het systeem. Als dit ingesteld is, staat de `bind` optie alleen verzoeken naar het juiste IP adres toe voor toegang naar de service. Je kunt deze methode gebruiken om verschillende services te verbinden met verschillende netwerk interfaces gebaseerd op de vereisten.

Dit is in het bijzonder nuttig voor systemen met meerdere netwerk adapters of met meerdere IP adressen. Op zo'n systeem, kunnen onveilige services (bijvoorbeeld, Telnet) ingesteld worden om alleen te luisteren naar de interface die verbonden is met een privé netwerk en niet naar de interface die verbonden is met het Internet.

De `redirect` optie accepteert een IP adres of hostnaam gevolgd door een poort nummer. Het stelt de service in om elk verzoek voor deze service om te leiden naar de opgegeven host en poort nummer. Deze eigenschap kan gebruikt worden om naar een ander poort nummer op hetzelfde systeem te wijzen, om het verzoek om te leiden naar een ander IP adres op dezelfde machine, om het verzoek te door te geven aan een totaal ander systeem en poort nummer, of elke combinatie van deze opties. Een gebruiker die verbindt met een bepaalde service op een systeem kan daarom omgeleid worden van een ander systeem zonder onderbreking.

De `xinetd` daemon is in staat om deze omleiding uit te voeren door het maken van een proces dat blijft bestaan tijdens de duur van de verbinding tussen de vragende cliënt machine en de host die de service in feite biedt, en dat de data overdracht tussen de twee systemen verzorgt

De voordelen van de `bind` and `redirect` opties zijn het duidelijkst zichtbaar als ze tesamen worden gebruikt. Door een service te verbinden met een bepaald IP adres op een systeem en dan de verzoeken voor deze service om te leiden naar een tweede machine die alleen de eerste machine kan zien, kan een intern systeem worden gebruikt om services aan te bieden voor een totaal ander netwerk. Deze opties kunnen alternatief gebruikt worden om de zichtbaarheid te beperken van een bepaalde service op een multi-homed machine naar een bekend IP adres, en ook elk verzoek voor die service om te leiden naar een andere machine die speciaal voor dat doel is ingesteld.

Bijvoorbeeld, overweeg een systeem dat wordt gebruikt als een firewall met deze instelling voor zijn Telnet service:

```
service telnet
{
        socket_type               = stream
        wait                       = no
        server                        = /usr/kerberos/sbin/telnetd
        log_on_success                += DURATION USERID
        log_on_failure                += USERID
        bind                    = 123.123.123.123
        redirect                = 10.0.1.13 23
}
```

De `bind` and `redirect` opties in dit bestand verzekeren dat de Telnet service op de machine is verbonden met het externe IP adres (`123.123.123.123`), het adres dat het Internet ziet. Bovendien wordt elk verzoek voor de Telnet service die gestuurd wordt naar `123.123.123.123` omgeleid met een tweede netwerk adapter naar een intern IP adres (`10.0.1.13`) dat alleen toegang heeft to de firewall en interne systemen. De firewall bestuurt dan communicatie tussen de twee systemen, en het verbindende systeem denkt dat het verbonden is met `123.123.123.123` terwijl het in werkelijkheid met een ander systeem verbonden is.

Deze eigenschap is in het bijzonder nuttig voor gebruikers met breedband verbindingen en slechts een vast IP adres. Als Network Address Translation (NAT) gebruikt wordt, zijn de systemen achter de gateway machine, welke alleen interne IP adressen gebruiken, niet beschikbaar buiten het gateway systeem. Als echter bepaalde services gecontroleerd door `xinetd` ingesteld zijn met de `bind` en `redirect` opties, kan de gateway machine als een proxy functioneren tussen de systemen buiten en een bepaalde interne machine ingesteld om de service te verlenen. Bovendien zijn de verschillende `xinetd` toegangs controle en loggings opties ook beschikbaar voor extra bescherming.

2.5.4.3.4. Hulpbronnen beheer opties

De `xinetd` daemon kan een basis niveau van bescherming bieden voor weigering van dienst (DoS) aanvallen. De instructies van de volgende lijst kunnen helpen om de effectiviteit van zo'n aanval te beperken:

- `per_source` — Definieert het maximum aantal instantiaties van een service per bron IP adres. Het accepteert alleen gehele getallen als argument en kan gebruikt worden in zowel **`xinetd.conf`** als de configuratie bestanden specifiek voor een service in de **`xinetd.d/`** map.
- `cps` — Definieert het maximum aantal verbindingen per seconde. Deze instructie heeft twee gehele getallen als argument gescheiden door spaties. Het eerste argument is het maximum aantal verbindingen per seconde toegestaan voor de service. Het tweede argument is het aantal secondes dat `xinetd` moet wachten voor dat de service weer beschikbaar komt. Het accepteert alleen gehele getallen als argument en kan gebruikt worden in zowel **`xinetd.conf`** als de configuratie bestanden specifiek voor een service in de **`xinetd.d/`** map.
- `max_load` — Definieert het CPU gebruik of gemiddelde load drempel voor een service. Het accepteert een drijvendekommagetal als argument.

  De gemiddelde load is een ruwe maatstaf voor het aantal processen dat op een gegeven moment actief is. Zie de **`uptime`**, **`who`**, en **`procinfo`** commando's voor meer informatie over gemiddelde load.

Er zijn meer hulpbron beheer opties beschikbaar voor `xinetd`. Refereer naar de **`xinetd.conf`** manual pagina voor meer informatie.

2.5.5. Extra hulpbronnen

Meer informatie over TCP wrappers en `xinetd` is beschikbaar in de systeem documentatie en op het Internet.

2.5.5.1. Geïnstalleerde TCP wrapper documentatie

De documentatie op je systeem is een goede plek om te beginnen voor het zoeken naar extra configuratie opties voor TCP wrappers, `xinetd`, en toegangs controle.

- **`/usr/share/doc/tcp_wrappers-<versie>/`** — Deze map bevat een **`README`** bestand dat uitlegt hoe TCP wrappers werken en bespreekt de verschillende hostnaam en hostadres voor de gek houden risico's.
- **`/usr/share/doc/xinetd-<versie>/`** — Deze map bevat een **`README`** bestand dat de verschillende aspecten van toegangs controle bespreekt en een **`sample.conf`** bestand met verschillende ideetjes voor het veranderen van service specifieke configuratie bestanden in de **`/etc/xinetd.d/`** map.
- TCP Wrappers en `xinetd` gerelateerde manual pagina's — Er bestaan een aantal manual pagina's voor de verschillende toepassingen en configuratie bestonden betrokken bij TCP wrappers en `xinetd`. De volgende zijn de belangrijkste:

  Server toepassingen
  - **`man xinetd`** — De manual pagina voor `xinetd`.

Configuratie bestanden

- **`man 5 hosts_access`** — De manual pagina voor de TCP wrapper hosts toegangscontrole bestanden.
- **`man hosts_options`** — De manual pagina voor de optie velden van TCP wrappers.
- **`man xinetd.conf`** — De manual pagina die de configuratie opties van `xinetd` laat zien.

2.5.5.2. Nuttige TCP wrapper websites

- *http://www.xinetd.org/*[4] — De thuisbasis van `xinetd`, welke voorbeeld configuratie bestanden bevat, een volledige lijst van eigenschappen, en een informatieve FAQ.
- *http://www.docstoc.com/docs/2133633/An-Unofficial-Xinetd-Tutorial* — Een gedegen handleiding die verschillende manieren bespreekt om standaard `xinetd` configuratie bestanden te optimaliseren voor het bereiken van specifieke beveiligings doelen .

2.5.5.3. Gerelateerde boeken

- *Hacking Linux Exposed* door Brian Hatch, James Lee, en George Kurtz; Osbourne/McGraw-Hill — Een uitstekende beveiligings hulpbron met informatie over TCP wrappers en `xinetd`.

2.6. Kerberos

Systeem beveiliging en integriteit binnen een netwerk kan onhandelbaar zijn. Het kan verschillende beheerders bezighouden om alleen maar in de gaten te houden welke services op een netwerk draaien en hoe deze services gebruikt worden.

Verder kan de authenticatie van gebruikers voor netwerk services gevaarlijk blijken als de methode gebruikt in het protocol inherent onveilig is, zoals aangetoond wordt door het versturen van onversleutelde wachtwoorden over een netwerk met FTP en Telnet protocollen.

Kerberos is een manier om de noodzaak voor protocollen die onveilige authenticatie methodes toestaan te elimineren, en op die manier de netwerk beveiliging verbeteren.

2.6.1. Wat is Kerberos?

Kerberos is een netwerk authenticatie protocol gemaakt door MIT, wat symmetrische sleutel cryptografie[5] gebruikt om gebruikers te authenticeren voor netwerk services, wat betekent dat wachtwoorden nooit echt over het netwerk verstuurd worden.

Als gevolg hiervan zullen gebruikers die authenticeren voor netwerk services met gebruik van Kerberos, ongeoorloofde gebruikers die proberen wachtwoorden te verzamelen door het bekijken van het netwerk verkeer effectief dwarsbomen.

2.6.1.1. Voordelen van Kerberos

De meeste conventionele netwerk services gebruiken een op wachtwoorden gebaseerd authenticatie schema. Zulke schema's vereisen dat een gebruiker zich authentiseert bij een bepaalde service door

[5] Een systeem waarbij zowel de cliënt als de server een gemeenschappelijke sleutel delen die gebruikt wordt voor het versleutelen en ontsleutelen van netwerk communicatie.

een gebruikersnaam en wachtwoord op te geven. Helaas gebeurt het versturen van deze authenticatie informatie voor veel services onversleuteld. Om zo'n schema veilig te laten zijn, moet het netwerk ontoegankelijk voor buitenstaanders zijn, en moeten alle computers en gebruikers op het netwerk vertrouwd en betrouwbaar zijn.

Zelfs als dit het geval is, kan een netwerk dat verbonden is met het Internet niet langer als veilig verondersteld worden. Elke aanvaller die toegang krijgt tot het netwerk kan een eenvoudig pakket analyse programma, ook bekend als pakket snuffelaar, gebruiken om gebruikersnamen en wachtwoorden te onderscheppen, en gebruikersaccounts en de integriteit van de gehele beveiligings infrastructuur in gevaar brengen.

Het belangrijkste ontwerp doel van Kerberos is om het versturen van onversleutelde wachtwoorden over het netwerk te elimineren. Als het juist gebruikt wordt, zal Kerberos de bedreiging, die pakket snuffelaars anders op een netwerk zijn, effectief te elimineren.

2.6.1.2. Nadelen van Kerberos

Hoewel Kerberos een algemene en ernstige beveiligings bedreiging verwijdert, kan het om een aantal redenen moeilijk te implementeren zijn:

- Het verhuizen van gebruikers wachtwoorden van een standaard UNIX wachtwoorden database, zoals **`/etc/passwd`** of **`/etc/shadow`**, naar een Kerberos wachtwoord database kan vervelend zijn, omdat er geen geautomatiseerd mechanisme is om deze taak uit te voeren. Refereer naar Vraag 2.23 in de online Kerberos FAQ:

  http://www.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html[6]

- Kerberos is slechts gedeeltelijk compatibel met het Pluggable Authentication Modules (PAM) systeem die door de meeste Fedora servers gebruikt wordt. Refereer naar *Paragraaf 2.6.4, “Kerberos en PAM”* voor meer informatie over dit probleem.

- Kerberos neemt aan dat elke gebruiker vertrouwd is, maar dat deze een onvertrouwde host op een onvertrouwd netwerk gebruikt. Zijn belangrijkste doel is om te voorkomen dat onversleutelde wachtwoorden over dat netwerk verstuurd worden. Als echter iemand anders dan de werkelijke gebruiker toegang heeft tot de host die kaartjes voor authenticatie uitgeeft — het *sleutel distributie centrum* (*KDC*) genaamd — is het hele Kerberos authenticatie systeem in gevaar.

- Om een toepassing Kerberos te laten gebruiken, moet zijn broncode veranderd worden om de juiste aanroepen naar de Kerberos bibliotheken te maken. Toepassingen die op deze manier veranderd zijn worden *bewust van Kerberos*, of *kerberized* genoemd. Voor sommige toepassingen kan dit een heel probleem zijn door de grootte van de toepassing of zijn ontwerp. Voor andere niet-compatibele toepassingen moeten veranderingen gemaakt worden in de manier waarop de server en cliënt communiceren. Dit kan nogal wat werk zijn. Gesloten bron toepassingen die standaard geen Kerberos ondersteuning hebben zijn vaak de meest problematische.

- Kerberos is een alles of niets oplossing. Als Kerberos op het netwerk gebruikt wordt, is elk onversleuteld wachtwoord verstuurd naar een service die zich niet bewust is van Kerberos een gevaar. Dus het netwerk heeft geen voordeel van het gebruik van Kerberos. Om een netwerk met Kerberos te beveiligen, moet men of zich van Kerberos bewuste versies van *alle* cliënt/server toepassingen die wachtwoorden onversleuteld versturen gebruiken, of *geen enkele* van zulke cliënt/server toepassingen gebruiken.

2.6.2. Kerberos terminologie

Kerberos heeft zijn eigen terminologie om verschillende aspecten van de service te definiëren. Voordat je leert hoe Kerberos werkt, is het belangrijk om de volgende vaktermen te leren.

authenticatie server (AS)
: Een server die kaartjes uitgeeft voor een gewenste service die op hun beurt gegeven worden aan gebruikers voor toegang tot de service. De AS beantwoordt verzoeken van cliënten die geen legitimatie hebben of niet hebben verstuurd met een verzoek. Het wordt gewoonlijk gebruikt om toegang te krijgen tot de kaartjes uitgevende server (TGS) service door het uitgeven van een kaartjes-gerechtigde kaartje (TGT). De AS draait gewoonlijk op dezelfde host als het sleutel verdelings centrum (KDC).

ciphertext
: Versleutelde data

cliënt
: Een eenheid op het netwerk (een gebruiker, een host, of een toepassing) die een kaartje van Kerberos kan ontvangen.

legitimatiebewijs
: Een tijdelijke set van elektronische legitimatie die de identiteit van een gebruiker voor een bepaalde service verifieert. Ook een kaartje genoemd.

legitimatie opslag of kaartjes bestand
: Een bestand dat de sleutels bevat voor het versleutelen van de communicatie tussen een gebruiker en verscheidene netwerk services. Kerberos 5 ondersteunt een raamwerk voor het gebruik van andere opslag types, zoals gedeeld geheugen, maar bestanden worden meer uitgebreid ondersteund.

crypt hash
: Een eenrichtings hash gebruikt om gebruikers te authenticeren. Deze zijn veiliger dan het gebruik van niet-versleutelde data, maar ze zijn relatief eenvoudig te ontsleutelen voor een ervaren cracker.

GSS-API
: De Generic Security Service Application Program Interface (gedefinieerd in RFC-2743 uitgegeven door The Internet Engineering Task Force) is een set functies die beveiligings services bieden. Deze API wordt gebruikt door cliënten en services om elkaar te authenticeren zonder dat een van de programma's specifieke kennis heeft van het onderliggende mechanisme. Als een netwerk service (zoals cyrus-IMAP) GSS-API gebruikt, kan het Kerberos gebruiken voor authenticatie.

hash
: Ook bekent als een *hash waarde*. Een waarde gemaakt door op opgeven van een reeks karakters aan een *hash functie*. Deze waarden worden gewoonlijk gebruikt om de verzekeren van de verstuurde data niet gemanipuleerd is.

hash functie
: Een manier om een digitale "vingerafdruk" van input data te maken. Deze functie, her-rangschikken, transponeren of veranderen data op een andere manier om een *hash waarde* te maken.

sleutel
: Data gebruikt om andere data te versleutelen of ontsleutelen. Versleutelde data kan niet ontsleuteld worden zonder de juiste sleutel of met uitzonderlijk goed geluk door de cracker.

sleutel distributie centrum (KDC)
: Een service die Kerberos kaartjes uitgeeft, en die gewoonlijk op dezelfde host draait als de kaartjes-verlenende server (TGS)

sleuteltab (of sleutel tabel)
: Een bestand dat een niet-versleutelde lijst van hoofdrolspelers en hun sleutels bevat. Servers halen de sleutel die ze nodig hebben van de sleuteltab bestanden in plaats van **`kinit`** te gebruiken. Het standaard sleuteltab bestand is **`/etc/krb5.keytab`**. De KDC beheer server, **`/usr/kerberos/sbin/kadmind`**, is de enigste service die een ander bestand gebruikt (het gebruikt **`/var/kerberos/krb5kdc/kadm5.keytab`**).

kinit
: Het **`kinit`** commando laat een hoofdrolspeler die al ingelogd is het initiële kaartjes-verlenende kaartje (TGT) verkrijgen en opslaan.

hoofdrolspeler (of hoofdrolspeler naam)
: De hoofdrolspeler is de unieke naam van een gebruiker of service die toegestaan is om Kerberos te gebruiken voor authenticatie. Een hoofdrolspeler volgt de vorm `root[/instance]@REALM`. Voor een typische gebruiker, is de root gelijk aan hun login ID. De `instance` is optioneel. Als de hoofdrolspeler een instance heeft, is het gescheiden van de root met een slash ("/"). Een lege string ("") wordt als een geldige instance beschouwd (welke verschilt van de standaard `NULL` instance), maar het gebruiken kan verwarrend zijn. Alle hoofdrolspelers in een gebied hebben hun eigen sleutel, welke voor gebruikers is afgeleid van een wachtwoord of willekeurig is ingesteld voor services.

gebied
: Een netwerk dat Kerberos gebruikt, bestaande uit een of meer servers, KDC's genaamd, en een potentieel grote groep cliënten.

service
: Een programma waarnaar over het netwerk toegang wordt verkregen.

kaartje
: Een tijdelijke set van elektronische legitimatie die de identiteit van een cliënt voor een bepaalde service verifieert. Ook legitimatiebewijs genaamd.

kaartje-verlenende server (TGS)
: Een server die kaartjes uitgeeft voor een gewenste service welke op zijn beurt gebruikers toegang geeft tot de service. De TGS draait gewoonlijk op dezelfde host als de KDC.

kaartjes-verlenende kaartje (TGT)
: Een speciaal kaartje dat de cliënt toestaat om extra kaartjes te verkrijgen zonder ze aan te hoeven vragen bij de KDC.

onversleuteld wachtwoord
: Een leesbare tekst wachtwoord

2.6.3. Hoe werkt Kerberos

Kerberos verschilt van de gebruikersnaam/wachtwoord authenticatie methodes. In plaats van het authenticeren van elke gebruiker voor elke netwerk service, gebruikt Kerberos symmetrische versleuteling en een vertrouwde derde partij (een KDC), om gebruikers te authenticeren voor een aantal netwerk services. Als een gebruiker authenticeert naar de KDC, stuurt de KDC een kaartje specifiek voor die sessie terug naar de machine van de gebruiker, en elke service bewust van Kerberos kijkt naar het kaartje op de machine van de gebruiker in plaats van het vragen aan de gebruiker om authentiek verklaard te worden met gebruik van een wachtwoord.

Als een gebruiker op een netwerk dat zich bewust is van Kerberos inlogt op zijn werkstation, wordt zijn hoofdrolspeler opgestuurd naar de KDC als onderdeel van een verzoek voor een TGT van de Authenticatie server. Dit verzoek kan verstuurd worden het login programma zodat het transparant is voor de gebruiker, of het kan verzonden worden door het **`kinit`** programma nadat de gebruiker heeft ingelogd.

De KDC controleert daarna voor de hoofdrolspeler in zijn database. Als de hoofdrolspeler wordt gevonden, maakt de KDC een TGT, die versleuteld wordt met de sleutel van de gebruiker en teruggestuurd naar die gebruiker.

Het login of **`kinit`** programma op de cliënt ontsleuteld daarna de TGT met gebruik van de sleutel van de gebruiker, welke berekend wordt van het wachtwoord van de gebruiker. De sleutel van de gebruiker wordt alleen op de cliënt machine gebruikt en wordt *niet* over het netwerk verstuurd.

De TGT wordt ingesteld om te verlopen na een zekere tijdsperiode (gewoonlijk tien tot vierentwintig uur) en wordt bewaard in de legitimatie opslag van de cliënt machine. Een verloop tijd wordt ingesteld zodat een in gevaar gebrachte TGT slechts korte tijd door een aanvaller gebruikt kan worden. Nadat de TGT is uitgegeven, hoeft de gebruiker zijn wachtwoord niet meer op te geven totdat de TGT verloopt of totdat de gebruiker uitlogt en opnieuw inlogt.

Iedere keer dat een gebruiker toegang nodig heeft tot een netwerk service, gebruikt de cliënt software de TGT om een nieuw kaartje voor die specifieke service aan de TGS. Het service kaartje wordt daarna gebruikt om de gebruiker transparant voor authentiek te verklaren voor die service.

Waarschuwing

Het Kerberos systeem kan in gevaar worden gebracht als een gebruiker op het netwerk authenticieert bij een service die niet bewust is van Kerberos door een wachtwoord in leesbare tekst te versturen. Het gebruik van services niet bewust van Kerberos wordt ten sterkste ontraden. Zulke services zijn Telnet en FTP. Het gebruik van andere versleutelde protocollen, zoals SSH of SSL beveiligde services heeft de voorkeur, maar dit is niet ideaal.

Dit is slechts een algemeen overzicht over de werking van Kerberos authenticatie. Refereer naar *Paragraaf 2.6.10, “Extra hulpbronnen”* voor meer detail informatie.

Opmerking

Kerberos is afhankelijk van de juiste werking van de volgende netwerk services.

- Bij benadering tijdssynchronisatie tussen de machines op het netwerk.

  Een tijdssynchronisatie programma moet ingesteld worden voor het netwerk, zoals **`ntpd`**. Refereer naar **`/usr/share/doc/ntp-<versie-nummer>/index.html`**

voor meer informatie over het instellen van Network Time Protocol servers (waarin *<versie-nummer>* het versie nummer is van het **ntp** pakket geïnstalleerd op je systeem).

- Domain Name Service (DNS).

  Je moet er zeker van zijn dat de DNS regels en hosts in het netwerk juist ingesteld zijn. Refereer naar *Kerberos V5 System Administrator's Guide* in **/usr/share/doc/krb5-server-*<versie-nummer>*** voor meer informatie (waarin *<versie-nummer>* het versie nummer is van het **krb5-server** pakket geïnstalleerd op jouw systeem.

2.6.4. Kerberos en PAM

Services die zich bewust zijn van Kerberos maken op dit moment geen gebruik van Pluggable Authentication Modules (PAM) — deze services mijden PAM helemaal. Toepassingen die echter PAM gebruiken kunnen gebruik maken van Kerberos voor authenticatie als de **pam_krb5** module (geleverd in het **pam_krb5** pakket) geïnstalleerd is. Het **pam_krb5** pakket bevat voorbeeld configuratie bestanden die services zoals **login** en **gdm** toestaan om gebruikers te authenticeren en ook om initiële legitimatie te verkrijgen met gebruik van hun wachtwoord. Als toegang tot netwerk services altijd wordt uitgevoerd met service die zich bewust zijn van Kerberos of services die GSS-API gebruiken zoals IMAP, kan het netwerk als redelijk veilig beschouwd worden.

Belangrijk

Beheerders moet er op letten om gebruikers niet toe te staan om authenticatie uit te voeren voor de meeste netwerk services met gebruik van Kerberos wachtwoorden. Vele protocollen versleutelen hun wachtwoorden niet voordat ze over het netwerk verzonden worden, wat de voordelen van het Kerberos systeem vernietigd. Bijvoorbeeld, gebruikers moeten niet toegestaan worden om authenticatie uit te voeren voor Telnet met hetzelfde wachtwoord dat ze voor Kerberos gebruiken.

2.6.5. Het instellen van een Kerberos 5 server

Als Kerberos ingesteld wordt, installeer dan de KDC eerst. Als het nodig is om slaaf servers in te stellen, installeer dan de meester eerst.

Om de eerste Kerberos KDC in te stellen, volg je deze stappen op:

1. Verzeker je ervan dat tijdssynchronisatie en DNS correct werken op alle cliënt en server machines voor het instellen van Kerberos. Geef in het bijzonder aandacht aan de tijdssynchronisatie tussen de Kerberos server en zijn cliënten. Als het tijdsverschil tussen de server en zijn cliënten groter is dan vijf minuten (dit is instelbaar in Kerberos 5) kunnen Kerberos cliënten geen authenticatie krijgen van de server. Deze tijdssynchronisatie is nodig om een aanvaller te beletten om een oud Kerberos kaartje te gebruiken om zich de vermommen als een geldige gebruiker.

   Het wordt aanbevolen om een Network Time Protocol (NTP) compatibel cliënt/netwerk in te stellen zelfs als Kerberos niet wordt gebruikt. Fedora bevat het **ntp** pakket voor dit doel. Refereer naar **/usr/share/doc/ntp-*<versie-nummer>*/index.html** (waarin *<versie-nummer>* het versie nummer is van het **ntp** pakket geïnstalleerd op je systeem) voor details over het

instellen van Network Time Protocol servers, en *http://www.ntp.org* voor meer informatie over NTP.

2. Installeer de **krb5-libs**, **krb5-server**, en **krb5-workstation** pakketten op de specifieke computer die de KDC draait. Deze machine moet goed beveiligd zijn — indien mogelijk moet het geen andere services draaien dan de KDC.

3. Bewerk de **/etc/krb5.conf** en **/var/kerberos/krb5kdc/kdc.conf** configuratie bestanden om de gebiedsnaam en de domein naar gebied afbeelding weer te geven. Een eenvoudig gebied kan gemaakt worden door het vervangen van instantiaties van *EXAMPLE.COM* en *example.com* met de correcte domein naam — let er op om de hoofd en kleine letter namen in het juiste formaat te houden — en door het veranderen van de KDC van *kerberos.example.com* naar de naam van de Kerberos server. Bij conventie zijn alle gebiedsnamen in hoofdletters en alle DNS hostnamen en domeinnamen in kleine letters. Voor volledige details over het formaat van deze configuratie bestanden, refereer je naar hun respectievelijke manual pagina's.

4. Maak de database met gebruik van het **kdb5_util** programma vanaf een shell prompt:

```
/usr/kerberos/sbin/kdb5_util create -s
```

Het **create** commando maakt de database dat de sleutels voor het Kerberos gebied bewaart. De **-s** schakelaar forceert het aanmaken van een *stash* bestand waarin de meester server sleutel bewaard wordt. Als er geen stash bestand aanwezig is om de sleutel van te lezen, vraagt de Kerberos server (**krb5kdc**) de gebruiker naar het meester server wachtwoord (welke gebruikt kan worden om de sleutel te genereren) iedere keer als het opstart.

5. Bewerk het **/var/kerberos/krb5kdc/kadm5.acl** bestand. Dit bestand wordt gebruikt door **kadmind** om te bepalen welke hoofrolspelers beheerstoegang hebben tot de Kerberos database en hun niveau van toegang. De meeste organisaties kunnen volstaan met een enkele regel:

```
*/admin@EXAMPLE.COM  *
```

De meeste gebruikers worden in de database gerepresenteerd door een enkele hoofdrolspeler (met een *NULL*, of lege, instance, zoals *joe@EXAMPLE.COM*). In deze configuratie zijn gebruikers met een tweede hoofdrolspeler met een instance van *admin* (bijvoorbeeld, *joe/admin@EXAMPLE.COM*) in staat om volledige macht over de Kerberos database van het gebied uit te voeren.

Nadat **kadmind** is opgestart op de server, kan elke gebruiker toegang krijgen tot zijn services door het uitvoeren van **kadmin** op een van de cliënten of servers in het gebied. Echter alleen gebruikers opgegeven in het **kadm5.acl** bestand kunnen de database op enige manier veranderen, behalve voor het veranderen van hun eigen wachtwoorden.

Opmerking

Het **kadmin** programma communiceert met de **kadmind** server over het netwerk, en gebruikt Kerberos om authenticatie af te handelen. Als gevolg hiervan moet de eerste hoofdrolspeler al bestaan voordat het met de server kan verbinden over het netwerk om het te beheren. Maak de eerste hoofdrolspeler met het **kadmin.local**

commando, welke specifiek is ontworpen om gebruikt te worden op dezelfde host als de KDC en Kerberos niet gebruikt voor authenticatie.

Type het volgende **kadmin.local** commando op de KDC termial om de eerste hoofdrolspeler aan te maken:

```
/usr/kerberos/sbin/kadmin.local -q "addprinc username/admin"
```

6. Start Kerberos met de volgende commando's:

```
/sbin/service krb5kdc start
/sbin/service kadmin start
/sbin/service krb524 start
```

7. Voeg hoofdrolspelers toe voor de gebruikers met het **addprinc** commando in **kadmin**. **kadmin** en **kadmin.local** zijn commandoregel interfaces voor de KDC. Daarom zijn vele commando's — zoals **addprinc** — beschikbaar na het opstarten van het **kadmin** programma. Refereer naar de **kadmin** manual pagina voor meer informatie.

8. Verifieer dat de KDC kaartjes uitgeeft. Voer eerst **kinit** uit om een kaartje te krijgen en het te bewaren in het legitimatie opslag bestand. Vervolgens gebruik je **klist** om de lijst van legitimaties in de opslag te zien en gebruik je **kdestroy** om de opslag en de legitimatie die het bevat te vernietigen.

Opmerking

Standaard probeert **kinit** authenticatie uit te voeren met dezelfde systeem login gebruikersnaam (niet de Kerberos server). Als die gebruikersnaam niet overeenkomt met een hoofdrolspeler in de Kerberos database, geeft **kinit** een fout boodschap. Als dat gebeurt, geeft je **kinit** de naam van de juiste hoofdrolspeler mee als argument op de commandoregel (**kinit *<hoofdrolspeler>***).

Zodra deze stappen klaar zijn, moet de Kerberos server klaar zijn en draaien.

2.6.6. Het instellen van een Kerberos 5 cliënt

Het instellen van een Kerberos 5 cliënt is eenvoudiger dan het instellen van de server. Als een minimum installeer je de cliënt pakketten en geef je iedere cliënt een geldig **krb5.conf** configuratie bestand. Terwijl **ssh** en **slogin** de voorkeur methodes zijn om in te loggen op cliënt systemen, zijn kerberized versies van **rsh** en **rlogin** nog steeds beschikbaar, hoewel het gebruik hiervan vereist dat nog een aantal extra configuratie veranderingen gemaakt worden.

1. Wees er zeker van dat tijdssynchronisatie aanwezig is tussen de Kerberos cliënt en de KDC. Refereer naar *Paragraaf 2.6.5, “Het instellen van een Kerberos 5 server”* voor meer informatie. Verifieer bovendien dat DNS correct werkt op de Kerberos cliënt voordat je de Kerberos cliënt programma's instelt.

2. Installeer de **krb5-libs** en **krb5-workstation** pakketten op alle cliënt machines. Maak een geldig **/etc/krb5.conf** bestand voor iedere cliënt (gewoonlijk kan dit hetzelfde **krb5.conf** bestand zijn gebruikt door de KDC).

3. Voordat een werkstation in het gebied Kerberos kan gebruiken voor authenticatie van gebruikers die verbinden met **ssh** of kerberized **rsh** of **rlogin**, moet het zijn eigen hoofdrolspeler hebben in de Kerberos database. De **sshd**, **kshd**, en **klogind** server programma's hebben allemaal toegang nodig tot de sleutels voor de *host* service van de hoofdrolspeler. Bovendien moet het werkstation, om de kerberized **rsh** en **rlogin** services te gebruiken, het **xinetd** pakket geïnstalleerd hebben.

   Met gebruik van **kadmin** voeg je een host hoofdrolspeler toe voor het werkstation op de KDC. De instance is in dit geval de hostnaam van het werkstation. Gebruik de **-randkey** optie voor het **addprinc** commando van de **kadmin** om de hoofdrolspeler aan te maken en ken het een willekeurige sleutel toe:

   ```
   addprinc -randkey host/blah.example.com
   ```

   Nu de hoofdrolspeler is aangemaakt, kunnen sleutels bepaald worden voor het werkstation door het uitvoeren van **kadmin** *op het workstation zelf*, en het gebruik van het **ktadd** commando binnen **kadmin**:

   ```
   ktadd -k /etc/krb5.keytab host/blah.example.com
   ```

4. Om andere kerberized netwerk services re gebruiken, moeten ze eerst gestart worden. Hieronder is een lijst van enkele algemene kerberized services en instructies over het aanzetten:

   - **ssh** — OpenSSH gebruikt GSS-API voor authenticatie van gebruikers voor servers als de configuratie bestanden van zowel de cliënt als de server beide `GSSAPIAuthentication` aangezet hebben. Als de cliënt ook `GSSAPIDelegateCredentials` aangezet heeft, wordt de legitimatie van de gebruiker beschikbaar gemaakt op het systeem op afstand.
   - **rsh** en **rlogin** — Om kerberized versies van **rsh** en **rlogin** te gebruiken, zet je **klogin**, **eklogin**, en **kshell** aan.
   - Telnet — Om kerberized Telnet te gebruiken, moet **krb5-telnet** aangezet worden.
   - FTP — Om FTP toegang te geven, bepaal je een sleutel voor de hoofdrolspeler met een root `ftp`. Wees er zeker van om de instance in te stellen naar de volledig gekwalificeerde hostnaam van de FTP server en zet dan **gssftp** aan.
   - IMAP — Om een kerberized IMAP server te gebruiken, gebruikt het **cyrus-imap** pakket Kerberos 5 als ook het **cyrus-sasl-gssapi** pakket geïnstalleerd is. Het **cyrus-sasl-gssapi** pakket bevat de Cyrus SASL plugins welke GSS-API authenticatie ondersteunen. Cyrus IMAP moet correct werken met Kerberos zo lang de **cyrus** gebruiker in staat is om de juiste sleutel in **/etc/krb5.keytab** te vinden, en de root voor de hoofdrolspeler in ingesteld naar **imap** (aangemaakt met **kadmin**).

     Een alternatief voor **cyrus-imap** kan gevonden worden in het **dovecot** pakket, welke ook toegevoegd is aan Fedora. Dit pakket bevat een IMAP server die, op dit moment GSS-API en Kerberos niet ondersteund.
   - CVS — Om een kerberized CVS server te gebruiken, gebruikt **gserver** een hoofdrolspeler met een root `cvs` en is anders identiek aan de **pserver**.

2.6.7. Domein naar gebied afbeelding

Als een cliënt probeert om toegang te krijgen tot een service die op een bepaalde server draait, weet het de naam van de service (*host*) en de naam van de server (*foo.example.com*), maar omdat er meer dan een gebieden in je netwerk kunnen zijn, moet het raden naar de naam van het gebied waarin de server zich bevindt.

Standaard wordt aangenomen dat de naam van het gebied de DNS naam van de server is in hoofdletters.

```
foo.example.org → EXAMPLE.ORG
        foo.example.com → EXAMPLE.COM
        foo.hq.example.com → HQ.EXAMPLE.COM
```

In sommige configuraties zal dit voldoende zijn, maar in andere zal de gebiedsnaam die zo afgeleid wordt, de naam van een niet bestaand gebied zijn. In die gevallen moet de afbeelding van de DNS domeinnaam van de server naar de naam van zijn gebied opgegeven worden in de *domain_realm* sectie van het **krb5.conf** bestand op het cliënt systeem. Bijvoorbeeld:

```
[domain_realm]
.example.com = EXAMPLE.COM
example.com = EXAMPLE.COM
```

De bovenstaande instelling geeft twee afbeeldingen op. De eerste afbeelding specificeert dat elk systeem in het "example.com" DNS domein onderdeel is van het *EXAMPLE.COM* gebied. De tweede specificeert dat een systeem met de exacte naam "example.com" ook onderdeel van het gebied is. (Het verschil tussen een domein en een specifieke host wordt aangegeven door de aanwezigheid of ontbreken van de eerste ".".) De afbeelding kan ook direct in DNS opgeslagen worden.

2.6.8. Instellen van secundaire KDC's

Om een aantal redenen kun je er voor kiezen om meerdere KDC's te draaien in een bepaald gebied. In dit scenario bewaart een KDC (de *meester KDC*) een schrijfbare kopie van de gebieds database en draait **kadmind** (het is ook de *admin server* voor jouw gebied), en een of meer KDC's (slaaf KDC's) bewaren alleen-lezen kopieën van de database en draaien **kpropd**.

De meester-slaaf overdrachtsprocedure houdt in dat de meester KDC zijn database dumpt in een tijdelijk dump bestand en dat bestand dan overbrengt naar iedere slaaf., welke daarna zijn vorig ontvangen alleen-lezen kopieën van de database overschrijft met de inhoud van het dump bestand.

Om een slaaf KDC in te srellen, wees er dan eerst zeker van om de **krb5.conf** en **kdc.conf** bestanden van de meester KDC te kopiëren naar de slaaf KDC.

Start **kadmin.local** op in een root shell op de meester KDC en gebruik zijn **add_principal** commando om een nieuwe regel aan te maken voor de *host* service van de meester KDC, en gebruik daarna zijn **ktadd** commando om gelijktijdig een willekeurige sleutel voor de service in te stellen en die sleutel op te slaan in het standaard keytab bestand van de meester. Deze sleutel zal door het **kprop** commando gebruikt worden voor authenticatie naar de slaaf servers. Je hoeft dit maar een keer te doen, onafhankelijk van hoeveel slaaf servers je installeert.

```
# kadmin.local -r EXAMPLE.COM
```

```
Authenticating as principal root/admin@EXAMPLE.COM with password.

kadmin: add_principal -randkey host/masterkdc.example.com

Principal "host/host/masterkdc.example.com@EXAMPLE.COM" created.

kadmin: ktadd host/masterkdc.example.com

Entry for principal host/masterkdc.example.com with kvno 3, encryption
 type Triple DES cbc mode with HMAC/sha1 added to keytab WRFILE:/etc/
krb5.keytab.

Entry for principal host/masterkdc.example.com with kvno 3, encryption type
 ArcFour with HMAC/md5 added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/masterkdc.example.com with kvno 3, encryption type
 DES with HMAC/sha1 added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/masterkdc.example.com with kvno 3, encryption type
 DES cbc mode with RSA-MD5 added to keytab WRFILE:/etc/krb5.keytab.

kadmin: quit
```

Start **kadmin** op in een root shell op de slaaf KDC en gebruik zijn **add_principal** commando om een nieuwe regel aan te maken voor de *host* service van de slaaf KDC, en gebruik daarna zijn **ktadd** commando om gelijktijdig een willekeurige sleutel voor de service in te stellen en die sleutel op te slaan in het standaard keytab bestand van de slaaf. Deze sleutel wordt door de **kpropd** srviece gebruikt voor authenticatie van cliënten.

```
# kadmin -p jimbo/admin@EXAMPLE.COM -r EXAMPLE.COM

Authenticating as principal jimbo/admin@EXAMPLE.COM with password.

Password for jimbo/admin@EXAMPLE.COM:

kadmin: add_principal -randkey host/slavekdc.example.com

Principal "host/slavekdc.example.com@EXAMPLE.COM" created.

kadmin: ktadd host/slavekdc.example.com@EXAMPLE.COM

Entry for principal host/slavekdc.example.com with kvno 3, encryption
 type Triple DES cbc mode with HMAC/sha1 added to keytab WRFILE:/etc/
krb5.keytab.

Entry for principal host/slavekdc.example.com with kvno 3, encryption type
 ArcFour with HMAC/md5 added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/slavekdc.example.com with kvno 3, encryption type
 DES with HMAC/sha1 added to keytab WRFILE:/etc/krb5.keytab.
```

```
Entry for principal host/slavekdc.example.com with kvno 3, encryption type
 DES cbc mode with RSA-MD5 added to keytab WRFILE:/etc/krb5.keytab.

kadmin: quit
```

Met zijn service sleutel, kan de slaaf KDC elke cliënt authenticeren die er naar verbindt. Het is duidelijk dat het ze niet allemaal toegestaan kan zijn om de **kprop** service van de slaaf te voorzien met een nieuwe gebiedsdatabase. Om toegang te beperken, zal de **kprop** service op de slaaf KDC alleen vernieuwingen accepteren van cliënten waarvan de hoofdrolspeler namen opgegeven zijn in **/var/kerberos/krb5kdc/kpropd.acl**. Voeg de naam van de host service van de master KDC toe aan dat bestand.

```
# echo host/masterkdc.example.com@EXAMPLE.COM > /var/kerberos/krb5kdc/
kpropd.acl
```

Zodra de slaaf KDC een kopie van de database heeft verkregen, zal het ook de meester sleutel nodig hebben die gebruikt is om het te versleutelen. Als de meester sleutel van de database van jouw KDC wordt bewaard in een *stash* bestand op de meester KCD (gewoonlijk met de naam **/var/kerberos/krb5kdc/.k5.REALM**), kopieer je het naar de slaaf KDC met een beschikbare veilige methode, of je maakt een dummy database en identiek stash bestand op de slaaf KDC door **kdb5_util create -s** uit te voeren (de dummy database zal overschreven worden door de eerste succesvolle database overdracht) en hetzelfde wachtwoord op te geven.

Verzeker je ervan dat de firewall van de slaaf KDC de meester KDC toestaat om er toegang tot te hebben met gebruik van TCP op poort 754 (*krb5_prop*), en start de **kprop** service. Controleer daarna opnieuw of de **kadmin** service *uitgezet* is.

Voer nu een handmatige database overdrachts test uit door het dumpen van de gebiedsdatabase, op de meester KDC, naar het standaard data bestand welke het **kprop** commando zal lezen (**/var/kerberos/krb5kdc/slave_datatrans**), en gebruik daarna het **kprop** commando om zijn inhoud over te brengen naar de slaaf KDC.

```
# /usr/kerberos/sbin/kdb5_util dump /var/kerberos/krb5kdc/
slave_datatrans# kprop slavekdc.example.com
```

Met gebruik van **kinit** verifieer je dat een cliënt systeem waarvan **krb5.conf** alleen de slaaf KDC laat zien in zijn lijst van KDC's voor jouw gebied, nu correct in staat is om initiële legitimatie te verkrijgen van de slaaf KDC.

Als dat gebeurd is, maak je een script welke de gebiedsdatabase dumpt en het **kprop** commando uitvoert om de database naar iedere slaaf KDC over te brengen, en stel de **cron** service in om het script periodiek uit te voeren.

2.6.9. Cross gebieds authenticatie instellen

Cross-gebieds authenticatie is de term die gebruikt wordt voor het beschrijven van situaties waarin cliënten (gewoonlijk gebruikers) van een gebied Kerberos gebruiken voor authenticatie van services (gewoonlijk server processen die draaien op een bepaald server systeem) die behoort tot een ander gebied dan hun eigen.

In het eenvoudigste geval, om voor een cliënt in een gebied met de naam **A.EXAMPLE.COM** toegang te krijgen tot een service in het **B.EXAMPLE.COM** gebied, moeten beide gebieden een sleutel delen voor een hoofdrolspeler van de naam **krbtgt/B.EXAMPLE.COM@A.EXAMPLE.COM**, en beide sleutels moeten hetzelfde sleutel versie nummer ermee verbonden hebben.

Om dit te bereiken, kies je een zeer sterk wachtwoord of wachtzin, en je maakt een regel voor de hoofdrolspeler voor beide gebieden met gebruik van kadmin.

```
# kadmin -r A.EXAMPLE.COM       kadmin: add_principal krbtgt/
B.EXAMPLE.COM@A.EXAMPLE.COM       Enter password for principal
 "krbtgt/B.EXAMPLE.COM@A.EXAMPLE.COM":       Re-enter password for
 principal "krbtgt/B.EXAMPLE.COM@A.EXAMPLE.COM":       Principal
 "krbtgt/B.EXAMPLE.COM@A.EXAMPLE.COM" created.       quit
  # kadmin -r B.EXAMPLE.COM       kadmin: add_principal krbtgt/
B.EXAMPLE.COM@A.EXAMPLE.COM       Enter password for principal "krbtgt/
B.EXAMPLE.COM@A.EXAMPLE.COM":       Re-enter password for principal
 "krbtgt/B.EXAMPLE.COM@A.EXAMPLE.COM":       Principal "krbtgt/
B.EXAMPLE.COM@A.EXAMPLE.COM" created.       quit
```

Gebruik het **get_principal** commando om te verifiëren dat beide regels overeenkomende sleutel versie nummers (**kvno** waarden) en versleuteling types hebben

Het dumpen van de database werk niet

Beheerders die bewust zijn van beveiliging proberen misschien de **-randkey** optie van het **add_principal** commando te gebruiken om een willekeurige sleutel toe te kennen in plaats van een wachtwoord, dumpen daarna de nieuwe regel uit de database van het eerste gebied, en importeren het in het tweede. Dit zal niet werken behalve als de meester sleutels voor de gebieds databases identiek zijn, omdat de sleutels in een database dump zelf versleuteld zijn met de meester sleutel.

cliënten in het **A.EXAMPLE.COM** gebied kunnen nu authentificeren voor services in het **B.EXAMPLE.COM** gebied. Anders gezegd, het **B.EXAMPLE.COM** gebied *vertrouwt* nu het **A.EXAMPLE.COM** gebied, of nog eenvoudiger, **B.EXAMPLE.COM** *vertrouwt* nu **A.EXAMPLE.COM**.

Dit brengt ons naar een belangrijk punt: cross-gebieds vertrouwen is standaard in een richting. De KDC voor het **B.EXAMPLE.COM** gebied kan de cliënten van **A.EXAMPLE.COM** vertrouwen voor authenticatie voor services in het **B.EXAMPLE.COM** gebied, maar dit heeft geen effect of cliënten in het **B.EXAMPLE.COM** gebied wel of niet vertrouwd worden voor authenticatie voor services in het **A.EXAMPLE.COM** gebied. Om vertrouwen in de andere richting te krijgen, moeten beide gebieden sleutels delen voor de **krbtgt/A.EXAMPLE.COM@B.EXAMPLE.COM** service (let op de omgekeerde volgorde van de twee gebieden vergeleken met het vorige voorbeeld).

Als directe vertrouwens relaties de enigste methode was voor het aanbieden van vertrouwen tussen gebieden, dan zouden netwerken die meerdere gebieden bevatten erg moeilijk in te stellen zijn. Gelukkig is cross-gebieds vetrouwen transitief. Als cliënten van **A.EXAMPLE.COM** authenticatie kunnen verkrijgen voor services in **B.EXAMPLE.COM**, en cliënten van **B.EXAMPLE.COM** authenticatie kunnen verkrijgen voor services in**C.EXAMPLE.COM**, dan kunnen cliënten in **A.EXAMPLE.COM** ook authenticatie verkrijgen voor services in **C.EXAMPLE.COM**, *zelfs als* ***C.EXAMPLE.COM*** ***A.EXAMPLE.COM*** *niet direct vertrouwt*. Dit betekent dat op een netwerk met meerdere gebieden die elkaar moeten vertrouwen, het maken van goede keuzes over de op te zetten vetrouwens relaties, de hoeveelheid werk aanzienlijk kan reduceren.

Nu zit je met de meer conventionele problemen: het systeem van de cliënt moet zodanig ingesteld worden, dat het correct het gebied kan bepalen waartoe een bepaalde service hoort en het moet in staat zijn om te bepalen hoe legitimatie voor services in dat gebied te verkrijgen zijn.

Eerste dingen eerst: de naam van de hoofdrolspeler voor een service geboden van af een specifieke server in een gegeven gebied ziet er als volgt uit:

```
service/server.example.com@EXAMPLE.COM
```

In dit voorbeeld is *service* gewoonlijk of de naam van het protocol dat gebruikt wordt (andere veel voorkomende waarden zijn *ldap*, *imap*, *cvs*, en *HTTP*) of *host*, *server.example.com* is de volledig gekwalifiseerde domein naan van het systeem die de service draait, en **`EXAMPLE.COM`** is de naam van het gebied.

Om te bepalen tot welk gebied de service behoort, zullen cliënten vaak DNS moeten raadplegen, of de **`domain_realm`** sectie van **`/etc/krb5.conf`** om of een hostnaam (*server.example.com*) of een DNS domein naam (*.example.com*) te koppelen aan de naam van het gebied (*EXAMPLE.COM*).

Nadat bepaald is tot welk gebied een service behoort, moet een cliënt bepalen welke set van gebieden het moet benaderen, en in welke volgorde dit moet gebeuren om legitimatie te krijgen om authenticatie te vragen aan de service.

Dit kan op twee manieren gedaan worden.

De standaard methode, die geen expliciete instelling nodig heeft, is om gebieden namen te geven binnen een gedeelde hierarchie. Als voorbeeld, beschouw gebieden met de namen **`A.EXAMPLE.COM`**, **`B.EXAMPLE.COM`**, en **`EXAMPLE.COM`**. Als een cliënt in het **`A.EXAMPLE.COM`** gebied probeert authenticatie te krijgen voor een service in **`B.EXAMPLE.COM`**, zal het standaard eerst proberen legitimatie voor het **`EXAMPLE.COM`** gebied te krijgen, en deze legitimatie gebruiken om legitimatie te krijgen voor gebruik in het **`B.EXAMPLE.COM`** gebied.

De cliënt behandelt de gebiedsnaam in dit scenario zoals men een DNS naam behandelt. Het stript herhaaldelijk onderdelen van zijn eigen gebiedsnaam af om de namen van gebieden tr genereren die "boven" zijn in de hiërarchie totdat het een punt bereikt die ook "boven" is voor het gebied van de service. Op dat punt begint het met toevoegen van onderdelen van de gebiedsnaam van de server totdat het gebied van de server bereikt wordt. Elk gebied die in dit proces betrokken is is een nieuwe "hop".

Bijvoorbeeld, met gebruik van legitimatie in **`A.EXAMPLE.COM`**, om dan authenticatie te krijgen voor een service in **`B.EXAMPLE.COM`** **`A.EXAMPLE.COM`** → **`EXAMPLE.COM`** → **`B.EXAMPLE.COM`**

- **`A.EXAMPLE.COM`** en**`EXAMPLE.COM`** delen een sleutel voor **`krbtgt/EXAMPLE.COM@A.EXAMPLE.COM`**
- **`EXAMPLE.COM`** en **`B.EXAMPLE.COM`** delen een sleutel voor **`krbtgt/B.EXAMPLE.COM@EXAMPLE.COM`**

Een ander voorbeeld, met gebruik van legitimatie in **`SITE1.SALES.EXAMPLE.COM`**, om dat authenticatie te krijgen voor een service in **`EVERYWHERE.EXAMPLE.COM`** **`SITE1.SALES.EXAMPLE.COM`** → **`SALES.EXAMPLE.COM`** → **`EXAMPLE.COM`** → **`EVERYWHERE.EXAMPLE.COM`**

- **`SITE1.SALES.EXAMPLE.COM`** en **`SALES.EXAMPLE.COM`** delen een sleutel voor **`krbtgt/SALES.EXAMPLE.COM@SITE1.SALES.EXAMPLE.COM`**

- **SALES.EXAMPLE.COM** en **EXAMPLE.COM** delen een sleutel voor **krbtgt/EXAMPLE.COM@SALES.EXAMPLE.COM**
- **EXAMPLE.COM** rn **EVERYWHERE.EXAMPLE.COM** delen een sleutel voor **krbtgt/EVERYWHERE.EXAMPLE.COM@EXAMPLE.COM**

Een ander voorbeeld, deze keer met gebiedsnamen waarvan de namen geen gemeenschappelijke achtervoegsel hebben (**DEVEL.EXAMPLE.COM** en **PROD.EXAMPLE.ORG DEVEL.EXAMPLE.COM → EXAMPLE.COM → COM → ORG → EXAMPLE.ORG → PROD.EXAMPLE.ORG**

- **DEVEL.EXAMPLE.COM** en **EXAMPLE.COM** delen een sleutel voor **krbtgt/EXAMPLE.COM@DEVEL.EXAMPLE.COM**
- **EXAMPLE.COM** en **COM** delen een sleutel voor **krbtgt/COM@EXAMPLE.COM**
- **COM** en **ORG** delen een sleutel voor **krbtgt/ORG@COM**
- **ORG** en **EXAMPLE.ORG** delen een sleutel voor **krbtgt/EXAMPLE.ORG@ORG**
- **EXAMPLE.ORG** en **PROD.EXAMPLE.ORG** delen een sleutel voor **krbtgt/PROD.EXAMPLE.ORG@EXAMPLE.ORG**

De meer ingewikkelde, maar ook meer flexibele, methode bevat het instellen van de **capaths** sectie van **/etc/krb5.conf**, zodat cliënten die legitimatie hebben voor een gebied in staat zijn om op te zoeken welk gebied de volgende is in de keten die zal leiden tot het punt waarop het authenticatie kan aanvragen aan de servers.

Het formaat van de **capaths** sectie is relatief ongecompliceerd: iedere regel in de sectie wordt genoemd naar een gebied waarin een cliënt kan bestaan. Binnen die subsectie, wordt een lijst gegeven van tussengelegen gebieden waarvan de cliënt legitimatie moet verkrijgen met als waarde de sleutel die overeenkomt met het gebied waarin een service kan bestaan. Als er geen tussenliggende gebieden zijn, wordt de waarde "." gebruikt.

Hier is een voorbeeld:

```
[capaths]
        A.EXAMPLE.COM = {
        B.EXAMPLE.COM = .
        C.EXAMPLE.COM = B.EXAMPLE.COM
        D.EXAMPLE.COM = B.EXAMPLE.COM
        D.EXAMPLE.COM = C.EXAMPLE.COM
        }
```

In dit voorbeeld kunnen cliënten in het **A.EXAMPLE.COM** gebied cross-gebied legitimatie verkrijgen voor **B.EXAMPLE.COM** rechtstreeks van de **A.EXAMPLE.COM** KDC.

Als die cliënten contact willen maken met een service in het **C.EXAMPLE.COM** gebied, moeten ze eerst de benodigde legitimaties verkrijgen van het **B.EXAMPLE.COM** gebied (dit vereist dat **krbtgt/B.EXAMPLE.COM@A.EXAMPLE.COM** bestaat), en dan **die** legitimatie gebruiken om legitimaties te verkrijgen voor gebruik in het **C.EXAMPLE.COM** gebied (met gebruik van **krbtgt/C.EXAMPLE.COM@B.EXAMPLE.COM**).

Als die cliënten contact willen maken met een service in het **D.EXAMPLE.COM** gebied, moeten ze eerst de benodigde legitimatie verkrijgen van het **B.EXAMPLE.COM** gebied, en dan legitimaties van

het **C.EXAMPLE.COM** gebied voordat ze tenslotte legitmaties voor gebruik met het **D.EXAMPLE.COM** gebied kunnen krijgen.

Opmerking

Zonder een capath regel die anders aangeeft, neemt Kerberos aan dat cross-gebieds vetrouwens relaties een hiërarchie vormen.

cliënten in het **A.EXAMPLE.COM** gebied kunnen rechtstreeks cross-gebieds legitimatie verkrijgen van **B.EXAMPLE.COM**. Zonder de "." die dit aangeeft, zou de cliënt anders proberen het hiërarchische pad te gebruiken, in dit geval:

```
A.EXAMPLE.COM → EXAMPLE.COM → B.EXAMPLE.COM
```

2.6.10. Extra hulpbronnen

Voor meer informatie over Kerberos, refereer je naar de volgende hulpbronnen.

2.6.10.1. Geïnstalleerde Kerberos documentatie

- De *Kerberos V5 Installation Guide* en de *Kerberos V5 System Administrator's Guide* in PostScript en HTML formaten. Deze kunnen gevonden worden in de **/usr/share/doc/krb5-server-<*versie-nummer*>/** map (waarin <*versie-nummer*> het versie nummer is van het **krb5-server** pakket geïnstalleerd op je systeem).

- De *Kerberos V5 UNIX User's Guide* in PostScript en HTML formaten. Deze kunnen gevonden worden in de **/usr/share/doc/krb5-workstation-<*versie-nummer*>/** map (waarin <*versie-nummer*> het versie nummer is van het **krb5-workstation** pakket geïnstalleerd op je systeem).

- Kerberos manual pagina's — Er zijn een aantal manual . voor de verschillende toepassingen en configuratie bestanden betrokken bij een Kerberos implementatie. Het volgende is een lijst van een paar van de meest interessante manual pagina's

  cliënt toepassingen
  - **man kerberos** — Een inleiding voor het Kerberos systeem welke beschrijft hoe legitimaties werken en biedt aanbevelingen voor het verkrijgen en vernietigen van Kerberos kaartjes. Onderin refereert de manual pagina naar een aantal gerelateerde manual pagina's.
  - **man kinit** — Beschrijf hoe je dit commando kunt gebruiken om een kaartjes-verlenend kaartje kunt verkrijgen en opslaan.
  - **man kdestroy** — Beschrijft hoe je dit commando kunt gebruiken om Kerberos legitimaties te vernietigen.
  - **man klist** — Beschrijft hoe je dit commando kunt gebruiken om de opgeslagen Kerberos legitimaties te tonen.

  Beheers toepassingen
  - **man kadmin** — Beschrijft hoe je dit commando kunt gebruiken om de Kerberos V5 database te beheren.

- **`man kdb5_util`** — Beschrijft hoe je dit commando kunt gebruiken voor het aanmaken en uitvoeren van beheersfuncties op laag niveau voor de Kerberos V5 database.

Server toepassingen

- **`man krb5kdc`** — Beschrijft beschikbare commandoregel opties voor de Kerberos V5 KDC.
- **`man kadmind`** — Beschrijft beschikbare commandoregel opties voor de Kerberos V5 beheersserver.

Configuratie bestanden

- **`man krb5.conf`** — Beschrijft het formaat en de opties beschikbaar in het configuratie bestand voor de Kerberos V5 bibliotheek.
- **`man kdc.conf`** — Beschrijft het formaat en de opties beschikbaar in het configuratie bestand voor de Kerberos V5 AS en KDC.

2.6.10.2. Nuttige Kerberos websites

- *http://web.mit.edu/kerberos/www/* — *Kerberos: The Network Authentication Protocol* webpagina van MIT.
- *http://www.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html* — De Kerberos vaak gestelde vragen (FAQ).
- *ftp://athena-dist.mit.edu/pub/kerberos/doc/usenix.PS* — De PostScript versie van *Kerberos: An Authentication Service for Open Network Systems* door Jennifer G. Steiner, Clifford Neuman, and Jeffrey I. Schiller. Dit document is het originele artikel dat Kerberos beschrijft.
- *http://web.mit.edu/kerberos/www/dialogue.html* — *Designing an Authentication System: a Dialogue in Four Scenes* origineel door Bill Bryant in 1988, veranderd door Theodore Ts'o in 1997. Dit document is een gesprek tussen twee ontwikkelaars die het denkproces van het maken van een authenticatie systeem lijkend op Kerberos doormaken. De gesprekstoon van de discussie maakt dit een goed startpunt voor iedereen die geheel onbekend is met Kerberos.
- *http://www.ornl.gov/~jar/HowToKerb.html* — *How to Kerberize your site* is een goede referentie voor het kerberizing van een netwerk.
- *http://www.networkcomputing.com/netdesign/kerb1.html* — *Kerberos Network Design Manual* is een gedegen overzicht van het Kerberos systeem.

2.7. Virtuele privé netwerken (VPN's)

Bedrijven met verscheidene satelliet kantoren verbinden vaak met elkaar met specifieke lijnen voor effectiviteit en bescherming van gevoelige data tijdens de overdracht. Bijvoorbeeld, veel bedrijven gebruiken frame relay of *Asynchronous Transfer Mode* (ATM) lijnen voor een eindpunt-naar-eindpunt oplossing om een kantoor met andere te verbinden. Dit kan een duur voorstel zijn, in het bijzonder voor bedrijven van gemiddelde grootte (SMB's) die uit willen breiden zonder de hoge kosten te betalen die behoren bij specifieke digitale circuits voor grote ondernemingen.

Om aan deze behoeft te voldoen, werden *Virtuele privé netwerken* (VPN's) ontwikkeld. Door het opvolgen van dezelfde functionele principes als specifieke circuits, staan VPN's beveiligde digitale communicatie toe tussen twee partijen (of netwerken), waarmee een *Wide Area Network* (WAN) gemaakt wordt van bestaande *Local Area Networks* (LAN's). Waarin het verschilt van frame relay of

ATM is het transport medium. VPN's verzenden over IP met gebruik van datagrams als de transport laag, wat het een veilig kanaal door het Internet maakt naar een bedoelde bestemming. De meeste vrije VPN implementaties bevatten open standaard versleutelings methoden om de data tijdens de overdracht verder te maskeren.

Sommige bedrijven gebruiken hardware VPN oplossingen om de beveiliging te verbeteren, terwijl andere software gebruiken of op protocollen gebaseerde implementaties. Verschillende leveranciers bieden hardware VPN oplossingen, zoals Cisco, Nortel, IBM, en Checkpoint. Er is een vrije op software gebaseerde VPN oplossing voor Fedora met de naam FreeS/Wan die een standaard *Internet Protocol Security* (IPsec) implementatie gebruikt. Deze VPN oplossingen , onafhankelijk of ze op hardware of software gebaseerd zijn, werken als speciale routers die bestaan tussen de IP verbinding van een kantoor naar een ander.

2.7.1. Hoe werk een VPN?

Als een pakket wordt verzonden van een cliënt, stuurt deze het door de VPN router of gateway, welke een *Authenticatie header* (AH) toevoegt voor de route en authenticatie. De data wordt dan versleuteld en, tenslotte, ingesloten binnen een *Encapsulating Security Payload* (ESP). Deze laatste vertegenwoordigt de versleuteling en afhandelings instructies.

De ontvangende VPN router stript de kop informatie, ontsleutelt de data, en verstuurt het naar de bedoelde bestemming (een werkstation of een andere node op een netwerk). Door gebruik van een netwerk-naar-netwerk verbinding ontvangt de ontvangende node op het lokale netwerk de pakketten al ontsleuteld en klaar om verwerkt te worden. Het versleuteling/ontsleuteling proces in een netwerk-naar-netwerk VPN verbinding is transparant voor een lokale node.

Met zo'n verbeterd niveau van beveiliging, moet een aanvaller niet alleen een pakket onderscheppen, maar het pakket ook ontsleutelen. Indringers die een man-in-het-midden aanval uitvoeren tussen een server en de cliënt moeten ook toegang hebben tot tenminste een van de privé sleutels voor authenticatie van de sessies. Omdat zij verscheidene lagen van authenticatie en versleuteling gebruiken, zijn VPN's een veilig en effectief middel om te verbinden tussen meerdere node op afstand om te werken als een verenigd intranet.

2.7.2. VPN's and Fedora

Fedora biedt verschillende opties wat betreft de implementatie van software oplossingen voor een beveiligde verbinding naar een WAN. *Internet Protocol Security* (IPsec) is de ondersteunde VPN implementatie voor Fedora, en voldoet in voldoende mate aan de gebruikersbehoeften van bedrijven met buitenkantoren of gebruikers op afstand.

2.7.3. IPsec

Fedora ondersteunt IPsec voor het verbinden van hosts op afstand naar netwerken met gebruik van een beveiligde tunnel op een gemeenschappelijk dragers netwerk zoals het Internet. IPsec kan geïmplementeerd worden door gebruik van een host-naar-host (een computer werkstation naar een ander) of netwerk-naar-netwerk (een LAN/WAN naar een ander) configuratie

De IPsec implementatie in Fedora gebruikt *Internet Key Exchange* (*IKE*), een protocol geïmplementeerd door de Internet Engineering Task Force (IETF), gebruikt voor wederzijdse authenticatie en beveiligde samenwerking tussen verbindende systemen.

2.7.4. Een IPsec verbinding maken

Een IPsec verbinding is opgedeeld in twee logische fases. In fase 1 initialiseert een IPsec node de verbinding met de node of het netwerk op afstand. De node of het netwerk op afstand controleert de legitimaties van de aanvragende node en beide partijen onderhandelen over de authenticatie methode voor de verbinding.

Op Fedora systemen gebruikt een IPsec verbinding de *pre-shared key* methode van IPsec node authenticatie. In een pre-shared key IPsec verbinding, moeten beide hosts dezelfde sleutel gebruiken om verder te gaan naar fase 2 van de IPsec verbinding.

In fase 2 van de IPsec verbinding wordt de *Security Association* (SA) aangemaakt tussen de IPsec nodes. Deze fase maakt een SA database aan met configuratie informatie, zoals de versleutelings methode, geheime sessie sleutel uitwisselings parameters, en zo voort. Deze fase beheert de actuele IPsec verbinding tussen de nodes op afstand en netwerken.

De Fedora implementatie van IPsec gebruikt IKE voor het delen van sleutels tussen de hosts over het Internet. De **racoon** keying daemon handelt de IKE sleutel distributie en uitwisseling af. Refereer naar de **racoon** manual pagina voor meer informatie over deze daemon.

2.7.5. IPsec installatie

Het implementeren van IPsec vereist dat het **ipsec-tools** RPM pakket geïnstalleerd wordt op alle IPsec hosts (als een host-naar-host configuratie gebruikt wordt) of routers (als een netwerk-naar-netwerk configuratie gebruikt wordt). Het RPM pakket bevat essentiele bibliotheken, daemons, en configuratie bestanden voor het instellen van de IPsec verbinding, inclusief:

- **/sbin/setkey** — manipuleert het sleutel beheer en beveiligings attributen van IPsec in de kernel. Dit programma wordt gecontroleerd door de **racoon** sleutel beheers daemon. Refereer naar de **setkey**(8) manual pagina voor meer informatie.
- **/usr/sbin/racoon** — de IKU sleutelbeheers daemon, gebruikt om voor het beheren en controleren van beveiligings samenwerking en sleuteldeling tussen systemen verbonden met IPsec.
- **/etc/racoon/racoon.conf** — het **racoon** daemon configuratie bestand gebruikt om verschillende aspecten van de IPsec verbinding in te stellen, inclusief authenticatie methodes en versleutelings algorithmes gebruikt in de verbinding. Refereer naar de **racoon.conf**(5) manual pagina voor een complete opsomming van de beschikbare instructies.

Om IPsec in te stellen op Fedora, kun je het **Netwerkconfiguratie** gereedschap gebruiken, of de netwerk en IPsec configuratie bestanden handmatig bewerken.

- Om twee netwerk-verbonden hosts met IPsec te verbinden, refereer je naar *Paragraaf 2.7.6, “IPsec host-naar-host configuratie”*.
- Om een LAN/WAN naar een ander te verbinden met IPsec, refereer je naar *Paragraaf 2.7.7, “IPsec netwerk-naar-netwerk configuratie”*.

2.7.6. IPsec host-naar-host configuratie

IPsec kan ingesteld worden om een bureaublad of werkstation (host) te verbinden met een andere door het gebruiken van een host-naar-host verbinding. Dit type verbinding gebruikt het netwerk waarmee iedere host verbonden is om een beveiligde tunnel tussen beide hosts te maken. De vereisten voor een host-naar-host verbinding zijn minimaal, evenals de instelling van IPsec op iedere

host. De hosts hebben alleen een specifieke verbinding naar een draag netwerk nodig (zoals het Internet) en Fedora om de IPsec verbinding te maken.

2.7.6.1. Host-naar-host verbinding

Een host-naar-host IPsec verbinding is een versleutelde verbinding tussen twee systemen, welke beide IPsec draaien met dezelfde authenticatie sleutel. Als de IPsec verbinding actief is, wordt alle netwerkverkeer tussen de twee hosts versleuteld.

Om een host-naar-host IPsec verbinding in te stellen, gebruik je de volgende stappen op iedere host:

Opmerking

Je moet de volgende procedures uitvoeren op de actuele machine die je gaat instellen. Vermijd om te proberen IPsec verbindingen op afstand in te stellen en aan te maken.

1. In een commando shell, type je **`system-config-network`** om het **Netwerkconfiguratie** gereedschap op te starten.
2. In de **IPsec** tab klik je op **Nieuw** om de IPsec - instellingen helper op te starten.
3. Klik op **Vooruit** om de instelling van een host-naar-host IPsec verbinding te starten.
4. Vul een unieke alias naam in voor de verbinding, bijvoorbeeld **`ipsec0`**. Indien vereist, selecteer je het aanvinkhokje om de verbinding automatisch te activeren bij het opstarten van de computer. Klik op **Vooruit** om verder te gaan.
5. Selecteert **Host -naar-host encryptie** als het verbindingstype, en klik op **Vooruit**.
6. Selecteer het type versleuteling om te gebruiken: handmatig of automatisch.

   Als je handmatige encryptie kiest, moet later in het proces een encryptie sleutel opgegeven worden. Als je automatische encryptie kiest, beheert de **`racoon`** daemon de encryptie sleutel. Het **`ipsec-tools`** pakket moet geïnstalleerd zijn als je automatische encryptie wilt gebruiken.

   Klik op **Vooruit** om verder te gaan.
7. Vul het IP adres van de host op afstand in.

   Om het IP adres van de host op afstand te bepalen, gebruik je het volgende commando *op de host op afstand*

   ```
   [root@myServer ~] # /sbin/ifconfig <device>
   ```

   waarin <*device*> het Ethernet apparaat is dat je wilt gebruiken voor de VPN verbinding.

   Als je slechts een Ethernet kaart in het systeem hebt, is de apparaatnaam gewoonlijk eth0. Het volgende voorbeeld laat de relevante informatie van dit commando zien (merk op dat dit slechts een voorbeeld output is):

   ```
   eth0      Link encap:Ethernet  HWaddr 00:0C:6E:E8:98:1D
             inet addr:172.16.44.192  Bcast:172.16.45.255
    Mask:255.255.254.0
   ```

Het IP adres is het nummer dat volgt op de `inet addr:` label.

Opmerking

For host-to-host connections, both hosts should have a public, routable address. Alternatively, both hosts can have a private, non-routable address (for example, from the 10.x.x.x or 192.168.x.x ranges) as long as they are on the same LAN.

Als de hosts op verschillende LAN's zijn, of een heeft een publiek adres terwijl de ander een privé adres heeft, refereer je naar *Paragraaf 2.7.7, "IPsec netwerk-naar-netwerk configuratie"*.

Klik op **Vooruit** om verder te gaan.

8. Als handmatige encryptie was geselecteerd in stap *6*, geeft je de te gebruiken encryptie sleutel op, of klik op **Genereren** om er een te maken

   a. Geef een authenticatiesleutel op of klik op **Genereren** om er een te maken. Het kan elke combinatie van cijfers en letters zijn.

   b. Klik op **Vooruit** om verder te gaan.

9. Verifieer de informatie in de **IPsec — Samenvatting** pagina en klik dan op **Toepassen**.

10. Klik op **Bestand** => **Opslaan** om de configuratie op te slaan.

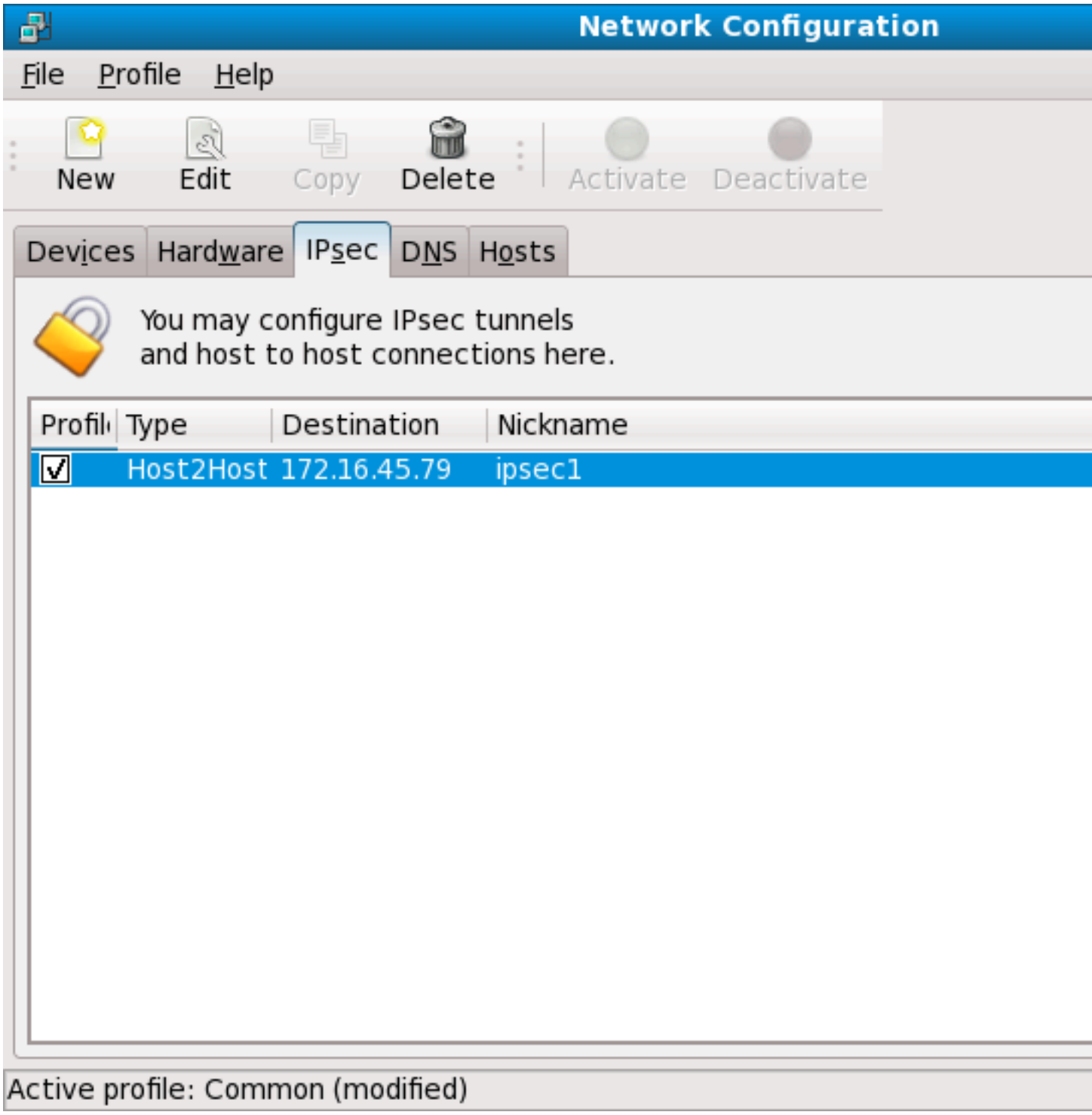
    Je moet misschien het netwerk opnieuw opstarten om de veranderingen effect te laten hebben. Om het netwerk opnieuw op te starten, gebruik je het volgende commando:

    ```
    [root@myServer ~]# service network restart
    ```

11. Selecteer de IPsec verbinding in de lijst en klik op de **Activeren** knop.

12. Herhaal de gehele procedure voor de andere host. Het is essentieel dat dezelfde sleutel van step *8* wordt gebruikt io de andere hosts. Anders zal IPsec niet werken.

Na het instellen van de IPsec verbinding, verschijnt het in de IPsec lijst zoals getoond in *Figuur 2.10, "IPsec verbinding"*.



Figuur 2.10. IPsec verbinding

De volgende bestanden worden aangemaakt als de IPsec verbinding is ingesteld:

- **`/etc/sysconfig/network-scripts/ifcfg-<nickname>`**
- **`/etc/sysconfig/network-scripts/keys-<nickname>`**
- **`/etc/racoon/<remote-ip>.conf`**

- **/etc/racoon/psk.txt**

Als automatische encryptie is geselecteerd, wordt **/etc/racoon/racoon.conf** ook aangemaakt

Als de interface actief is, wordt **/etc/racoon/racoon.conf** veranderd om **<*remote-ip*>.conf** te bevatten.

2.7.6.2. Handmatige IPsec host-naar-host configuratie

De eerste stap voor het maken van een verbinding is het verzamelen van systeem en netwerk informatie voor ieder werkstation. Voor een host-naar-host verbinding heb je het volgende nodig:

- Het IP adres van elke host
- Een unieke naam, bijvoorbeeld, `ipsec1`. Deze wordt gebruikt om de IPsec verbinding te identificeren en te onderscheiden van andere apparaten of verbindingen.
- Een vast encryptiesleutel, of een automatisch aangemaakte door **racoon**.
- Een pre-gedeelde authenticatie sleutel die gebruikt wordt tijdens de initiële fase van de verbinding en om encryptiesleutels uit te wisselen tijdens de sessie.

Bijvoorbeeld, veronderstel dat Werkstation A en Werkstation B met elkaar willen verbinden met een IPsec tunnel. Ze willen verbinden met gebruik van een pre-gedeelde sleutel met de waarde `Key_Value01`, en de gebruikers zijn erover eens om **racoon** te gebruiken voor het automatisch genereren en delen van een authenticatie sleutel tussen iedere host. Beide host gebruikers besluiten om hun verbinding `ipsec1` te noemen.

Opmerking

Je moet een PSK kiezen die een mengeling gebruikt van hoofdletters, kleine letters, cijfers en leestekens. Een gemakkelijk raadbare PSK is een beveiligings risico.

Het is niet nodig om dezelfde verbindingsnaam re gebruiken voor iedere host. Je moet een naam kiezen die handig is en betekenis heeft voor je installatie.

Het volgende is het IPsec configuratie bestand voor Werkstation A voor een host-naar-host IPsec verbinding met Werkstation B. De unieke naam om de verbinding te identificeren is in dit voorbeeld *ipsec1*,. dus het resulterende bestand wordt **/etc/sysconfig/network-scripts/ifcfg-ipsec1** genoemd.

```
DST=X.X.X.XTYPE=IPSEC
ONBOOT=no
IKE_METHOD=PSK
```

Voor Werkstation A, is *X.X.X.X* het IP adres van Werkstation B. Voor Werkstsation B, is *X.X.X.X* het IP adres van Werkstation A. Deze verbinding is niet ingesteld om op te starten tijdens het opstarten van de computer (`ONBOOT=no`) en het gebruikt een pre-gedeelde sleutel methode voor authenticatie (`IKE_METHOD=PSK`).

Het volgende is de inhoud van het pre-gedeelde sleutel bestand (met de naam **/etc/sysconfig/network-scripts/keys-ipsec1**) dat beide werkstations nodig hebben voor authenticatie van

elkaar. De inhoud van dit bestand moet op beide werkstations identiek zijn, en alleen de root gebruiker moet in staat zijn dit bestand te lezen of te schrijven.

```
IKE_PSK=Key_Value01
```

Belangrijk

Om het **keys-ipsec1** bestand zodanig te veranderen dat alleen de root gebruiker dit bestand kan lezen of veranderen, voer je het volgende commando uit na het aanmaken van het bestand:

```
[root@myServer ~] # chmod 600 /etc/sysconfig/network-scripts/keys-ipsec1
```

Om op elk gewenst moment de authenticatie sleutel te veranderen, bewerk je het **keys-ipsec1** bestand op beide werkstations.*Beide authenticatie sleutels moeten identiek zijn voor een juiste verbinding*.

Het volgende voorbeeld laat de specifieke instelling voor de fase 1 verbinding naar de host op afstand zien. Het bestand wordt ***X.X.X.X*.conf** genoemd, waarin *X.X.X.X* het IP adres van de IPsec host op afstand is. Merk op dat dit bestand automatisch aangemaakt wordt als de IPsec tunnel actief is en moet niet direct bewerkt worden.

```
remote X.X.X.X{
         exchange_mode aggressive, main;
         my_identifier address;
         proposal {
                 encryption_algorithm 3des;
                hash_algorithm sha1;
                authentication_method pre_shared_key;
                dh_group 2 ;
        }
}
```

Het standaard fase 1 configuratie bestand wordt aangemaakt als een IPsec verbinding wordt opgezet, bevat de volgende instructies gebruikt door de Fedora implementatie van IPsec

remote *X.X.X.X*
: Specificeert dat de volgende regels van dit configuratie bestand alleen van toepassing zijn voor de node op afstand geïdentificeerd met het *X.X.X.X* IP adres.

exchange_mode aggressive
: De standaard configuratie voor IPsec in Fedora gebruikt een agressieve authenticatie methode, wat de verbindings overhead verlaagt terwijl het de instelling van verscheidene IPsec verbindingen met meerdere hosts toestaat.

my_identifier address
: Specificeert de identificatie methode te gebruiken voor de authenticatie van nodes. Fedora gebruikt alleen IP adressen om nodes te identificeren.

encryption_algorithm 3des
: Specificeert de encryptie code gebruikt tijdens de authenticatie. Standaard wordt *Triple Data Encryption Standard* (3DES) gebruikt.

hash_algorithm sha1;
: Specificeert het hash algorithme gebruikt tijdens de fase 1 onderhandeling tussen nodes. Standaard wordt Secure Hash Algorithm versie 1 gebruikt.

authentication_method pre_shared_key
: Specificeert de authenticatie methode die gebruikt wordt tijdens de node onderhandeling. Standaard gebruikt Fedora pre-gedeelde sleutels voor authenticatie.

dh_group 2
: Specificeert het Diffie-Hellman groep getal voor het instellen van dynamisch aangemaakte sessie sleutels. Standaard wordt modp1024 (groep 2) gebruikt.

2.7.6.2.1. Het racoon configuratie bestand

De **`/etc/racoon/racoon.conf`** bestanden moeten identiek zijn op alle IPsec nodes *behalve* voor de **`include "/etc/racoon/X.X.X.X.conf"`** instructie. Deze instructie (en het bestand waarnaar het verwijst) wordt aangemaakt als de IPsec tunnel actief wordt. Voor Werkstation A is de *`X.X.X.X`* in de **`include`** instructie het IP adres van Werkstation B. Het omgekeerde geldt voor Werkstation B. Het volgende toont een typisch **`racoon.conf`** bestand als de IPsec verbinding actief is.

```
# Racoon IKE daemon configuration file.
# See 'man racoon.conf' for a description of the format and entries.

path include "/etc/racoon";
path pre_shared_key "/etc/racoon/psk.txt";
path certificate "/etc/racoon/certs";

sainfo anonymous
{
        pfs_group 2;
        lifetime time 1 hour ;
        encryption_algorithm 3des, blowfish 448, rijndael ;
        authentication_algorithm hmac_sha1, hmac_md5 ;
        compression_algorithm deflate ;
}
include "/etc/racoon/X.X.X.X.conf";
```

Dit standaard **`racoon.conf`** bestand bevat gedefinieerde paden voor IPsec configuratie, pre-gedeelde sleutel bestanden, en certificaten. De velden in `sainfo anonymous` beschrijven een fase 2 SA tussen IPsec nodes — het karakter van de IPsec verbinding (inclusief de ondersteunde encryptie algorithmes die gebruikt worden) en de methode voor het uitwisselen van sleutels. De volgende lijst definieert de velden van fase 2:

sainfo anonymous
: Geeft aan dat SA anoniem kan initialiseren met iedere gelijke mits de IPsec legitimaties overeen komen.

pfs_group 2
: Definieert het Diffie-Hellman sleutel uitwisselings protocol, welke de methode bepaalt waarmee de IPsec nodes een gemeenschappelijke tijdelijke sessie sleutel vaststellen voor de tweede fase van de IPsec verbinding. Standaard gebruikt de Fedora implementatie van IPsec groep 2 (of `modp1024`) van de Diffie-Hellman cryptografische sleutel uitwisselings groepen. Groep 2 gebruikt een 1024-bit modulair machtverheffen dat aanvallers belet om vorige IPsec overdrachten te ontsleutelen zelfs als de privé sleutel in gevaar is gebracht.

lifetime time 1 hour
: Deze parameter specificeert de levensduur van een SA en kan opgegeven worden met tijd of data bytes. De standaard Fedora implementatie van IPsec specificeert een uur levensduur.

encryption_algorithm 3des, blowfish 448, rijndael
: Specificeert de ondersteunde encryptoe codes voor fase 2. Fedora onderstent 3DES, 448-bit Blowfish, en Rijndael (de code gebruikt in de *Advanced Encryption Standard*, of AES).

authentication_algorithm hmac_sha1, hmac_md5
: Laat de ondersteunde hash algorithmes voor authenticatie zien. Ondersteunde modes zijn sha1 en md5 hash boodschap authenticatie codes (HMAC).

compression_algorithm deflate
: Definieert het Deflate compressie algorithme voor IP Payload Compression (IPCOMP) ondersteuning, die een potentieel sneller transport van IP datagrams over langzame verbindingen toestaat.

Om de verbinding te starten gebruik je het volgende commando op iedere host:

```
[root@myServer ~]# /sbin/ifup <nickname>
```

waarin <nickname> de naam is die je opgegeven hebt voor de IPsec verbinding.

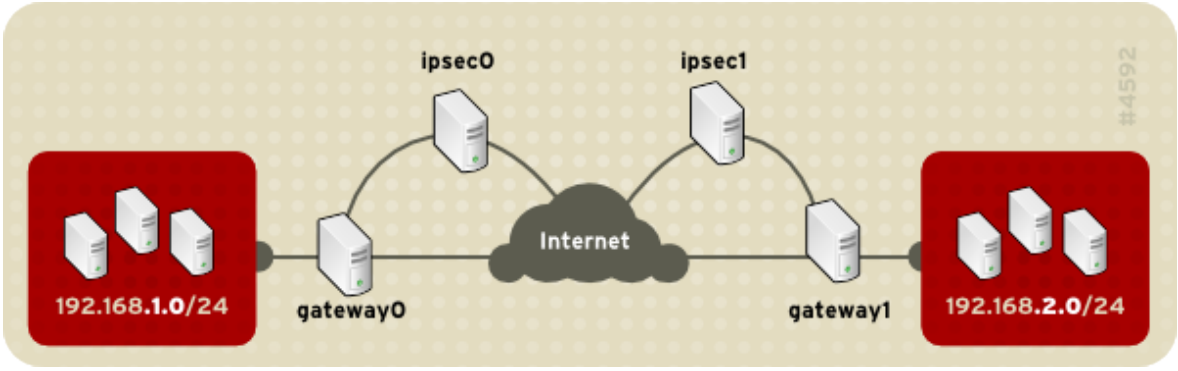
Om de IPsec verbinding te testen, voer je het **`tcpdump`** commando uit om de netwerkpakketten te zien die tussen de hosts uitgewisseld wordt en verifieer dat ze versleuteld zijn met IPsec. Het pakket moet een AH koptekst bevatten en moet getoond worden als ESP pakketten. ESP betekent dat het versleuteld is. Bijvoorbeeld:

```
[root@myServer ~]# tcpdump -n -i eth0 host <targetSystem>

IP 172.16.45.107 > 172.16.44.192: AH(spi=0x0954ccb6,seq=0xbb):
 ESP(spi=0x0c9f2164,seq=0xbb)
```

2.7.7. IPsec netwerk-naar-netwerk configuratie

IPsec kan ook ingesteld worden om een geheel netwerk (zoals een LAN of WAN) te verbinden met een netwerk op afstand met gebruik van een netwerk-naar-netwerk verbinding. Een netwerk-naar-netwerk verbinding vereist het instellen van IPsec routers aan iedere kant van de verbindende netwerken voor het transparant bewerken en doorgeven van informatie van een node op een LAN naar een node op een LAN op afstand. *Figuur 2.11, “Een netwerk-naar-netwerk IPsec verbinding met tunnel”* laat een netwerk-naar-netwerk IPsec verbinding met tunnel zien.



Figuur 2.11. Een netwerk-naar-netwerk IPsec verbinding met tunnel

Deze illustratie laat twee aparte LAN's zien gescheiden door het Internet. Deze LAN's gebruiken IPsec routers voor authenticatie en initiatie van een verbinding met gebruik van een beveiligde tunnel door het Internet. Pakketten die in de overdracht onderschept worden zullen brute-kracht ontsleuteling vereisen om de code te kraken die de pakketten tussen deze LAN's beschermd. Het proces van communicatie van een node in de 192.168.1.0/24 IP reeks naar een andere in de 92.168.2.0/24 reeks is geheel transparant voor de nodes omdat het verwerken, versleutelen/ontsleutelen, en omleiden van de IPsec pakketten geheel afgehandeld wordt door de IPsec router.

De informatie nodig voor een netwerk-naar-netwerk verbinding omvat:

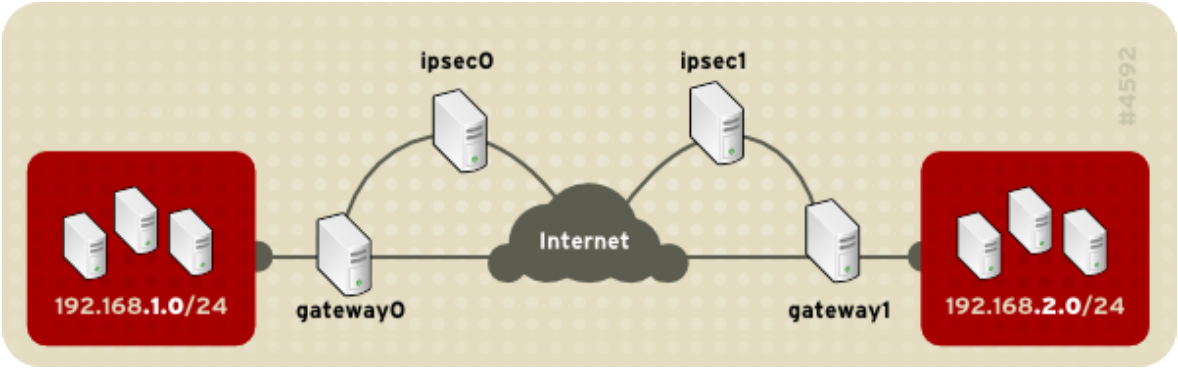
- De extern toegankelijke IP adressen van de specifieke IPsec routers
- De netwerk adres reeksen van de LAN/WAN bedient door de IPsec routers (zoals 192.168.1.0/24 of 10.0.1.0/24)
- Het IP adres van de gateway apparaten die de data van de netwerk nodes omleiden naar het Internet
- Een unieke naam, bijvoorbeeld, `ipsec1`. Deze wordt gebruikt om de IPsec verbinding te identificeren en te onderscheiden van andere apparaten of verbindingen.
- Een vaste encryptie sleutel of een automatisch aangemaakte door **`racoon`**
- Een pre-gedeelde authenticatie sleutel die gebruikt wordt tijdens de initiële fase van de verbinding en om encryptiesleutels uit te wisselen tijdens de sessie.

2.7.7.1. Netwerk-naar-netwerk (VPN) verbinding

Een netwerk-naar-netwerk IPsec verbinding gebruikt twee IPsec routers, een voor ieder netwerk, waarmee het netwerkverkeer voor de privé subnetten omgeleid wordt.

Bijvoorbeeld, zoals getoond in *Figuur 2.12, “Netwerk-naar-netwerk IPsec”*, als het 192.168.1.0/24 privé netwerk netwerkverkeer verstuurt naar het 192.168.2.0/24 privé netwerk, gaan de pakketten door gateway0, naar ipsec0, door het Internet, naar ipsec1, naar gateway1, en naar het 192.168.2.0/24 subnet.

IPsec routers vereisen publiek adresseerbare IP adressen en een tweede Ethernet apparaat verbonden met hun respectievelijke privé netwerken. Verkeer gaat alleen door een IPsec router als het bedoeld is voor een andere IPsec router waarmee het een versleutelde verbinding heeft.



Figuur 2.12. Netwerk-naar-netwerk IPsec

Alternatieve netwerk configuratie opties zijn een firewall tussen elke IP router en het Internet, en een intranet firewall tussen elke IPsec router en subnet gateway. De IPsec router en de gateway voor het subnet kunnen een systeem zijn met twee Ethernet apparaten: een met een publiek IP adres dat werkt als de IPsec router; en een met een privé IP adres dat werkt als de gateway voor het privé subnet. Iedere IPsec router kan de gateway gebruiken voor zijn privé netwerk of een publieke gateway om de pakketten naar de andere IPsec router te sturen.

Gebruik de volgende procedure om een netwerk-naar-netwerk IPsec verbinding in te stellen:

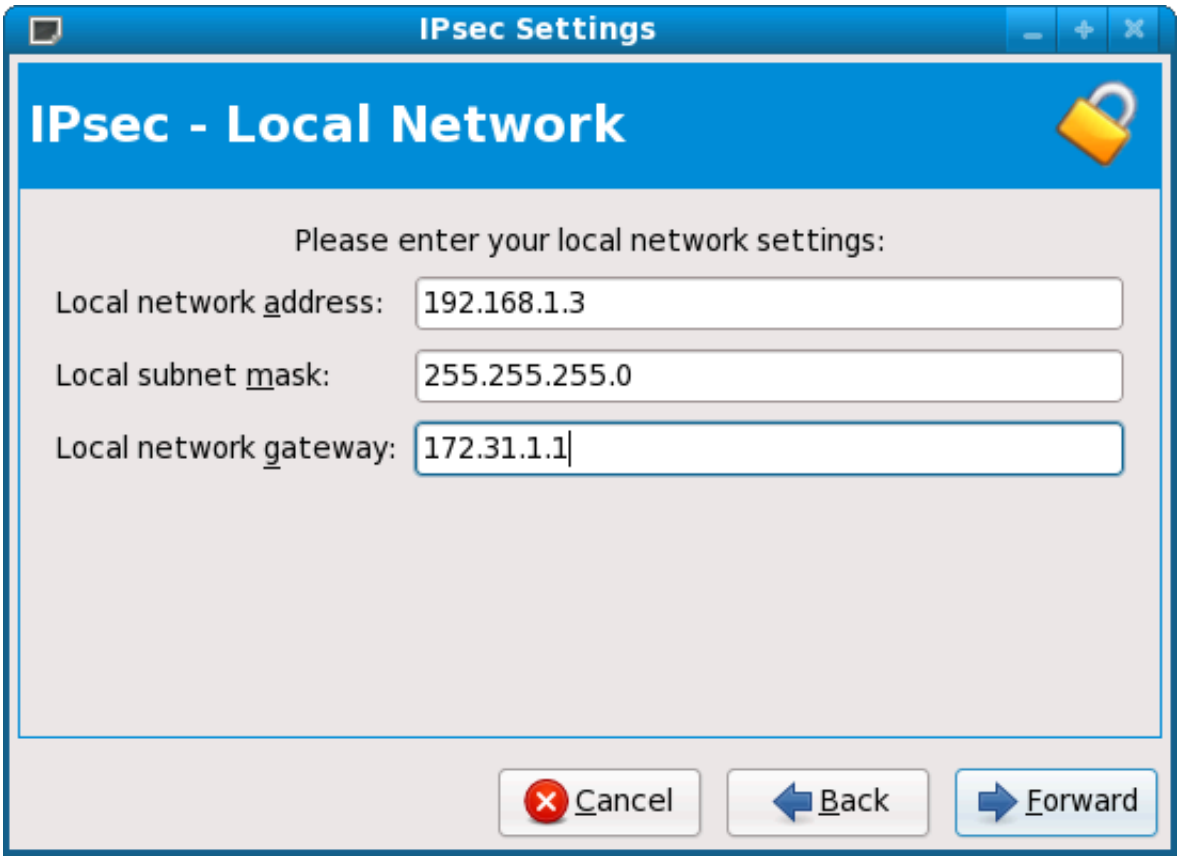
1. In een commando shell, type je **`system-config-network`** om het **Netwerkconfiguratie** gereedschap op te starten.
2. In de **IPsec** tab klik je op **Nieuw** om de IPsec - instellingen helper op te starten.
3. Klik op **Vooruit** om de netwerk-naar-netwerk IPsec verbinding in te stellen.
4. Vul een unieke naam in voor de verbinding, bijvoorbeeld, **`ipsec0`**. Indien nodig selecteer je het aanvinkhokje om de verbinding automatisch op te starten als de computer opgestart wordt. Klik op **Vooruit** om verder te gaan.
5. Selecteer **Netwerk-naar-netwerk encryptie (VPN)** als het connectie type, en klik dan op **Vooruit**.
6. Selecteer het type versleuteling om te gebruiken: handmatig of automatisch.

   Als je handmatige encryptie kiest, moet later in het proces een encryptie sleutel opgegeven worden. Als je automatische encryptie kiest, beheert de **`racoon`** daemon de encryptie sleutel. Het **`ipsec-tools`** pakket moet geïnstalleerd zijn als je automatische encryptie wilt gebruiken.

   Klik op **Vooruit** om verder te gaan.
7. Op de **Lokaal netwerk** pagina vul je de volgende informatie in:

   - **Lokaal netwerkadres** — Het IP adres van het apparaat op de IPsec router verbonden met het privé netwerk.
   - **Lokaal subnetmasker** — Het subnetmasker van het lokale netwerk IP adres.
   - **Lokaal netwerk gateway** — De gateway voor het privé subnet.

   Klik op **Vooruit** om verder te gaan.



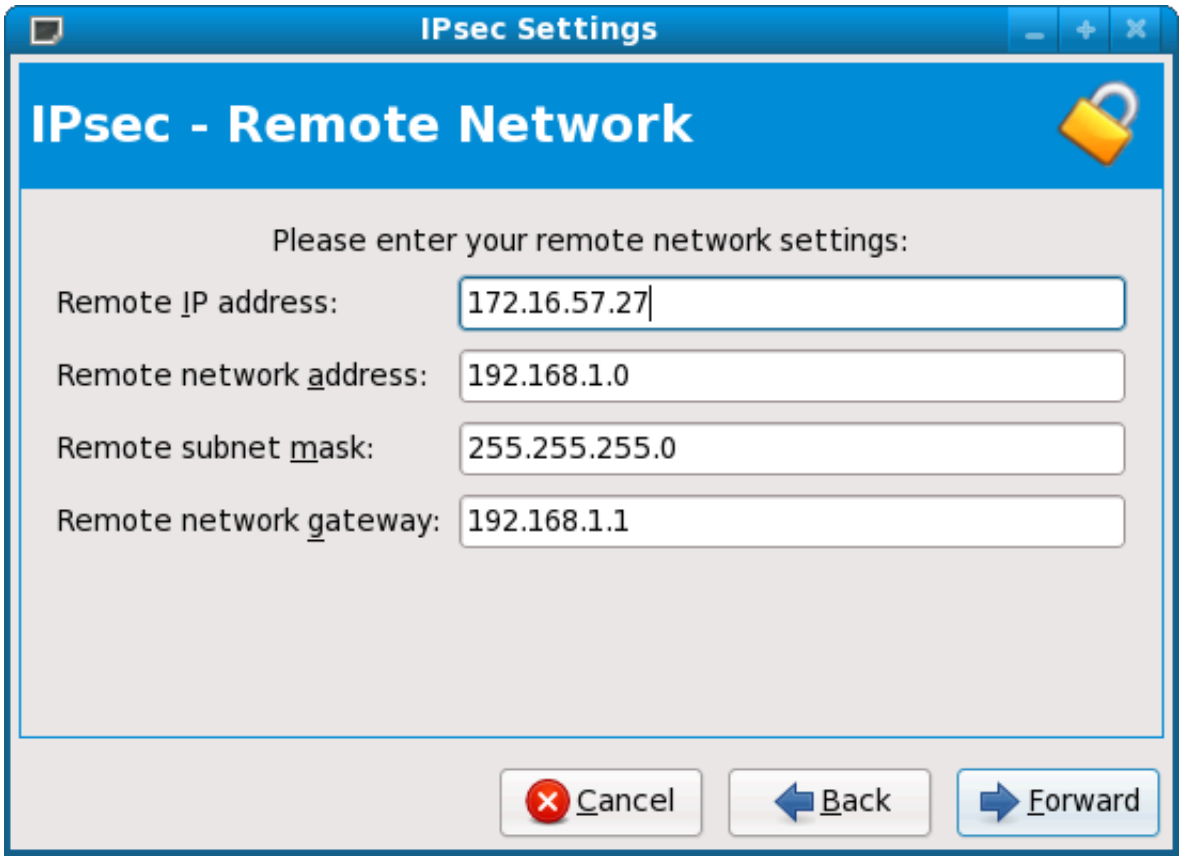
Figuur 2.13. Lokaal netwerk information

8. Op de **Netwerk op afstand** pagina vul je de volgende informatie in:

   - **IP-adres op afstand** — Het publiek adresseerbare IP adres van de IPsec router voor het *andere* privé netwerk. In ons voorbeeld, voor ipsec0, vul je het publiek adresseerbare IP adres van ipsec1 in, en omgekeerd.
   - **Netwerkadres op afstand** — Het netwerk adres van het privé subnet achter de *andere* IPsec router. In ons voorbeeld, vul je **`192.168.1.0`** in voor het instellen van ipsec1, en vul je **`192.168.2.0`** in voor het instellen van ipsec0.
   - **Subnetmasker op afstand** — Het subnet masker van het IP adres op afstand.
   - **Netwerk gateway op afstand** — Het IP adres van de gateway voor het netwerk adres op afstand.
   - Als handmatige encryptie was geselecteerd in stap *6*, specificeer je de te gebruiken encryptie sleutel of je klikt op **Genereren** om een aan te maken.

     Specificeer een authenticatie sleutel of klik op **Genereren** om een aan te maken. Deze sleutel kan een willekeurige combinatie kan cijfers en letters zijn.

   Klik op **Vooruit** om verder te gaan.



Figuur 2.14. Netwerk op afstand information

9. Verifieer de informatie in de **IPsec — Samenvatting** pagina en klik dan op **Toepassen**.
10. Selecteer **Bestand** => **Opslaan** om de instelling op te slaan.
11. Selecteer de IPsec verbinding in de lijst en klik dan op **Activeren** om de verbinding te activeren.
12. Zet IP forwarding aan:
    a. Bewerk **`/etc/sysctl.conf`** en zet `net.ipv4.ip_forward` op **`1`**.
    b. Gebruik het volgende commando om de verandering aan te zetten:

```
[root@myServer ~]# /sbin/sysctl -p /etc/sysctl.conf
```

Het netwerk script om de IPsec verbinding te activeren maakt automatisch netwerkroutes om pakketten te versturen door de IPsec router indien nodig.

2.7.7.2. Handmatige IPsec netwerk-naar-netwerk configuratie

Veronderstel dat LAN A (lana.example.com) en LAN B (lanb.example.com) met elkaar verbinden willen via een IPsec tunnel. Het netwerk adres voor LAN A is in de 192.168.1.0/24 reeks, terwijl LAN B de 192.168.2.0/24 reeks gebruikt. Het gateway IP adres is 192.168.1.254 voor LAN A en 192.168.2.254 voor LAN B. De IPsec routers zijn apart van iedere LAN gateway en gebruiken twee netwerkapparaten: eth0 is toegekend an een extern toegankelijk statisch IP adres die toegang geeft

tot het Internet, terwijl eth1 werkt als een routing punt voor het bewerken en verzenden van LAN pakketten van een netwerk node naar netwerk nodes op afstand.

De IPsec verbinding tussen elk netwerk gebruikt een pre-gedeelde sleutel met de waarde `r3dh4tl1nux`, en de beheerders van A en B komen overeen om **racoon** te gebruiken voor het automatisch genereren en delen van authenticatie sleutels tussen elke IPsec router. De beheerder van LAN A besluit de IPsec verbinding `ipsec0` te noemen, terwijl de beheerder van LAN B de IPsec verbinding `ipsec1` noemt.

Het volgende voorbeeld laat de inhoud van het **ifcfg** bestand voor een netwerk-naar-netwerk IPsec verbinding voor LAN A zien. De unieke naam om de verbinding in dit voorbeeld te identificeren is *ipsec0*, dus het resulterende bestand wordt **/etc/sysconfig/network-scripts/ifcfg-ipsec0** genoemd.

```
TYPE=IPSEC
ONBOOT=yes
IKE_METHOD=PSK
SRCGW=192.168.1.254
DSTGW=192.168.2.254
SRCNET=192.168.1.0/24
DSTNET=192.168.2.0/24
DST=X.X.X.X
```

De volgende lijst beschrijft de inhoud van dit bestand:

TYPE=IPSEC
: Specificeert het type verbinding.

ONBOOT=yes
: Specificeert dat de verbinding opgezet moet worden bij het opstarten van de computer.

IKE_METHOD=PSK
: Specificeert dat de verbinding de pre-gedeelde sleutel methode voor authenticatie gebruikt.

SRCGW=192.168.1.254
: Het IP adres voor de bron gateway. Voor LAN A, is dit de LAN A gateway, en voor LAN B, de LAN B gateway.

DSTGW=192.168.2.254
: Het IP adres voor de bestemmings gateway. Voor LAN A, is dit de LAN B gateway, en voor LAN B, de LAN A gateway.

SRCNET=192.168.1.0/24
: Specificeert het bron netwerk voor de IPsec verbinding, welke in dit voorbeeld de netwerk reeks voor LAN A is.

DSTNET=192.168.2.0/24
: Specificeert het bestemmings netwerk voor de IPsec verbinding, welke in dit voorbeeld de netwerk reeks voor LAN B is

DST=X.X.X.X
: Het extern toegankelijke IP adres van LAN B.

Het volgende voorbeeld is de inhoud van het pre-gedeelde sleutel bestand met de naam **`/etc/sysconfig/network-scripts/keys-ipsecX`** (waarin *X* 0 is voor LAN A en 1 voor LAN B) dat beide netwerken gebruiken voor authenticatie van elkaar. De inhoud van dit bestand moet identiek zijn en alleen de root gebruiker moet in staat zijn dit bestand te lezen of te schrijven.

```
IKE_PSK=r3dh4tl1nux
```

Belangrijk

Om het **`keys-ipsecX`** bestand te veranderen zodat alleen de root gebruiker dit bestand kan lezen of schrijven, gebruik je het volgende commando na het aanmaken van het bestand:

```
chmod 600 /etc/sysconfig/network-scripts/keys-ipsec1
```

Om op een willekeurig moment de authenticatie sleutel te veranderen, bewerk je het **`keys-ipsecX`** bestand op beide IPsec routers. *Beide sleutels moeten identiek zijn voor een juiste verbinding*.

Het volgende voorbeeld is de inhoud van het **`/etc/racoon/racoon.conf`** configuratie bestand voor de IPsec verbinding. Merk op dat de `include` regel onder in het bestand automatisch gegenereerd wordt en alleen verschijnt als de IPsec tunnel actief is.

```
# Racoon IKE daemon configuration file.
# See 'man racoon.conf' for a description of the format and entries.
path include "/etc/racoon";
path pre_shared_key "/etc/racoon/psk.txt";
path certificate "/etc/racoon/certs";

sainfo anonymous
{
        pfs_group 2;
        lifetime time 1 hour ;
        encryption_algorithm 3des, blowfish 448, rijndael ;
        authentication_algorithm hmac_sha1, hmac_md5 ;
        compression_algorithm deflate ;
}
include "/etc/racoon/X.X.X.X.conf"
```

Het volgende is de specifieke configuratie voor de verbinding naar het netwerk op afstand. Het bestand wordt ***`X.X.X.X`*`.conf`** genoemd (waarin *`X.X.X.X`* het IP adres is van de IPsec router op afstand). Merk op dat dit bestand automatisch gegenereerd wordt als de IPsec tunnel actief is en moet niet direct bewerkt worden.

```
remote X.X.X.X{
        exchange_mode aggressive, main;
        my_identifier address;
        proposal {
                encryption_algorithm 3des;
                hash_algorithm sha1;
```

```
                authentication_method pre_shared_key;
                dh_group 2 ;
        }
}
```

Voor het opstarten van de IPsec verbinding, moet IP forwarding aangezet worden in de kernel. Om IP forwarding aan te zetten:

1. Bewerk **`/etc/sysctl.conf`** en zet `net.ipv4.ip_forward` op **`1`**.

2. Gebruik het volgende commando om de verandering aan te zetten:

```
[root@myServer ~] # sysctl -p /etc/sysctl.conf
```

Om de IPsec verbinding te starten gebruik je het volgende commando op elke router:

```
[root@myServer ~] # /sbin/ifup ipsec0
```

De verbindingen zijn geactiveerd en zowel LAN A als LAN B zijn in staat om met elkaar te communiceren. De routes worden automatisch aangemaakt via het initialisatie script aangeroepen door het uitvoeren van **`ifup`** op de IPsec verbinding. Om een lijst van routes voor het netwerk te zien, gebruik je het volgende commando:

```
[root@myServer ~] # /sbin/ip route list
```

Om de IPsec verbinding te testen, voer je het **`tcpdump`** commando uit op het extern-routable apparaat (eth0 in dit geval) om de netwerkpakketten te zien die verstuurd worden tussen de kosts (of netwerken), en verifieer dat ze versleuteld zijn met IPsec. Bijvoorbeeld, om de IPsec verbinding van LAN A te controleren, gebruik je het volgende commando:

```
[root@myServer ~] # tcpdump -n -i eth0 host lana.example.com
```

Het pakket moet een AH koptekst bevatten en moet getoond worden als ESP pakketten. ESP betekent dat het versleuteld is. Bijvoorbeeld (back slashes (\) geven het vervolgen van een regel aan):

```
12:24:26.155529 lanb.example.com > lana.example.com:
 AH(spi=0x021c9834,seq=0x358): \
        lanb.example.com > lana.example.com: ESP(spi=0x00c887ad,seq=0x358)
 (DF) \
        (ipip-proto-4)
```

2.7.8. Het starten en stoppen van een IPsec verbinding

Als de IPsec verbinding niet ingesteld is om op te starten tijdens het opstarten van de computer, kun je het besturen vanaf de commandoregel.

Om de verbinding te starten, gebruik je het volgende commando voor elke host voor host-naar-host IPsec, of iedere IPsec router voor netwerk-naar-netwerk IPsec:

```
[root@myServer ~] # /sbin/ifup <nickname>
```

waarin <*nickname*> de naam is die je eerder hebt ingesteld, zoals `ipsec0`.

Om de verbinding te stoppen, gebruik je het volgende commando:

```
[root@myServer ~] # /sbin/ifdown <nickname>
```

2.8. Firewalls

Informatie beveiliging wordt meestal beschouwd als een proces en niet als een product. Standaard beveiligings implementaties gebruiken gewoonlijk echter een specifiek mechanisme om de toegangsrechten te controleren en netwerkhulpbronnen te beperken tot gebruikers die gemachtigd, identificeerbaar en traceerbaar zijn. Fedora bevat verscheidene gereedschappen om beheerders en beveiligings ingenieurs te helpen met toegangscontrole zaken op netwerk niveau.

Firewalls zijn een van de kernonderdelen van een netwerkbeveiligings implementatie. Verscheidene leveranciers brengen firewall oplossingen op de markt die zich richten op alle niveau's van de markt: van thuisgebruikers die een PC beschermen tot datacentrum oplossingen die vitale bedrijfsinformatie beschermen. Firewalls kunnen op zichzelf staande hardware oplossingen zijn, zoals firewall apparaten van Cisco, Nokia, en Sonicwall. Leveranciers zoals Checkpoint, McAfee, en Symantec hebben ook eigendomsmatige software firewall oplossingen ontwikkeld voor thuis en zakelijke markten.

Naast het verschil tussen hardware en software firewalls, zijn er ook verschillen in de manier waarop firewalls werken die de ene oplossing onderscheidt van een andere. *Tabel 2.2, “Firewall types”* geeft details van drie veel voorkomende types van firewalls en hoe ze werken:

Methode	Beschrijving	Voordelen	Nadelen
NAT	*Network Address Translation* (NAT) plaatst privé IP subnetwerken achter een of een kleine aantal publieke IP adressen en vermommen alle verzoeken alsof ze afkomstig zijn van een bron in plaats van een groter aantal. De Linux kernel heeft ingebouwde NAT functionaliteit door middel van het Netfilter kernel subsysteem.	· Kan transparant ingesteld worden voor machines op een LAN · Bescherming van vele machines en services achter een of meer externe IP adressen vereenvoudigt beheers verplichtingen · Beperking van gebruikerstoegang tot en vanaf het LAN kan ingesteld worden door het openen en sluiten van poorten op de NAT firewall/gateway	· Kan kwaadwillige activiteit niet voorkomen zodra gebruikers verbonden zijn met een service buiten de firewall
Pakket filter	Een pakketfiltering firewall leest elk datapakket die door een LAN passeert. Het kan pakketten lezen en verwerken door de koptekst informatie en filtert de pakketten gebaseerd op een stelsel van programmeerbare	· Op maat in te stellen met het **`iptables`** front-end programma · Vereist geen aanpassingen aan de kant van de cliënt, omdat alle netwerk activiteit gefilterd wordt op het router niveau in plaats van het toepassings niveau	· Kan geen pakketten filteren op inhoud zoals proxy firewalls · Verwerken pakketten op de protocol laag, maar kunnen pakketten niet filteren op een toepassings laag · Complexe netwerk architecturen kunnen

Methode	Beschrijving	Voordelen	Nadelen
	regels geimplementeerd door de firewallbeheerder. De Linux kernel heeft ingebouwde pakketfiltering functionaliteit door middel van het Netfilter kernel subsysteem.	· Omdat pakketten niet via een proxy verzonden worden, is de netwerk snelheid hoger door de directe verbinding tussen de cliënt en de host op afstand	het opstellen van pakketfilterings regels moeilijk maken, in het bijzonder als het gekoppeld wordt met *IP masquerading* of lokale subnetten en DMZ netwerken
Proxy	Proxy firewalls filteren alle verzoeken van een bepaald protocol of type van LAN cliënten naar een proxy machine, welke dan deze verzoeken op het Internet plaatst namens de lokale cliënt. Een proxy machine werkt als een buffer tussen kwaadwillige gebruikers op afstand en de interne netwerk cliënt machines.	· Geeft beheerders controle over welke toepassingen en protocols buiten het LAN kunnen werken · Sommige netwerk servers kunnen vaak benaderde data lokaal opslaan in plaats van de Internet verbinding te gebruiken om het te verzoeken. Dit helpt om het bandbreedte verbruik te verkleinen. · Proxy services kunnen nauwlettend gevolg en gelogd worden, wat een betere controle toestaat van het gebruik van de hulpbronnen op het netwerk	· Proxies zijn vaak toepassings-specifiek (HTTP, Telnet, enz.), of beperkt tot een protocol (de meeste proxies werken alleen met services verbonden via TCP) · Toepassingsservices kunnen niet achter een proxy draaien, dus je toepassingsservers moeten een andere vorm van netwerkbeveiliging gebruiken · Proxies kunnen een netwek flessenhals worden, omdat alle verzoeken en verzendingen door een bron gaan in plaats van direct van een cliënt naar een service op afstand

Tabel 2.2. Firewall types

2.8.1. Netfilter en IPTables

De Linux kernel biedt een krachtig netwerk subsysteem met de naam *Netfilter*. Het Netfilter subsysteem biedt met-toestand of zonder-toestand pakket filtering te samen met NAT en IP vermommings services. Netfilter heeft ook de mogelijkheid om IP koptekst informatie te mangelen voor geavanceerde routings en verbindings toestand beheer. Netfilter wordt gecontroleerd met gebruik van het **`iptables`** gereedschap.

2.8.1.1. IPTables overzicht

The power and flexibility of Netfilter is implemented using the **`iptables`** administration tool, a command line tool similar in syntax to its predecessor, **`ipchains`**, which Netfilter/iptables replaced in the Linux kernel 2.4 and above.

`iptables` uses the Netfilter subsystem to enhance network connection, inspection, and processing. **`iptables`** features advanced logging, pre- and post-routing actions, network address translation, and port forwarding, all in one command line interface.

Deze paragraaf geeft een overzicht van **`iptables`**. Voor meer gedetaileerde informatie refereer je naar *Paragraaf 2.9, “IPTables”*.

2.8.2. Basis firewall instelling

Net zoals een firewall in een gebouw probeert te voorkomen dat het vuur zich verspreidt, probeert een computer firewall te voorkomen dat kwaadwillige software zich verspreidt naar je computer. Het helpt ook om niet gemachtigde gebruikers toegang tot je computer te beletten.

In een standaard Fedora installatie is er een firewall tussen jouw computer of netwerk en alle niet vertrouwde netwerken, bijvoorbeeld het Internet. Het bepaalt tot welke services op jouw computer gebruikers op afstand toegang kunnen hebben. Een juist ingestelde firewall kan de beveiliging van je systeem sterk verbeteren. Het wordt aanbevolen dat je een firewall instelt voor elk Fedora systeem met een Internet verbinding.

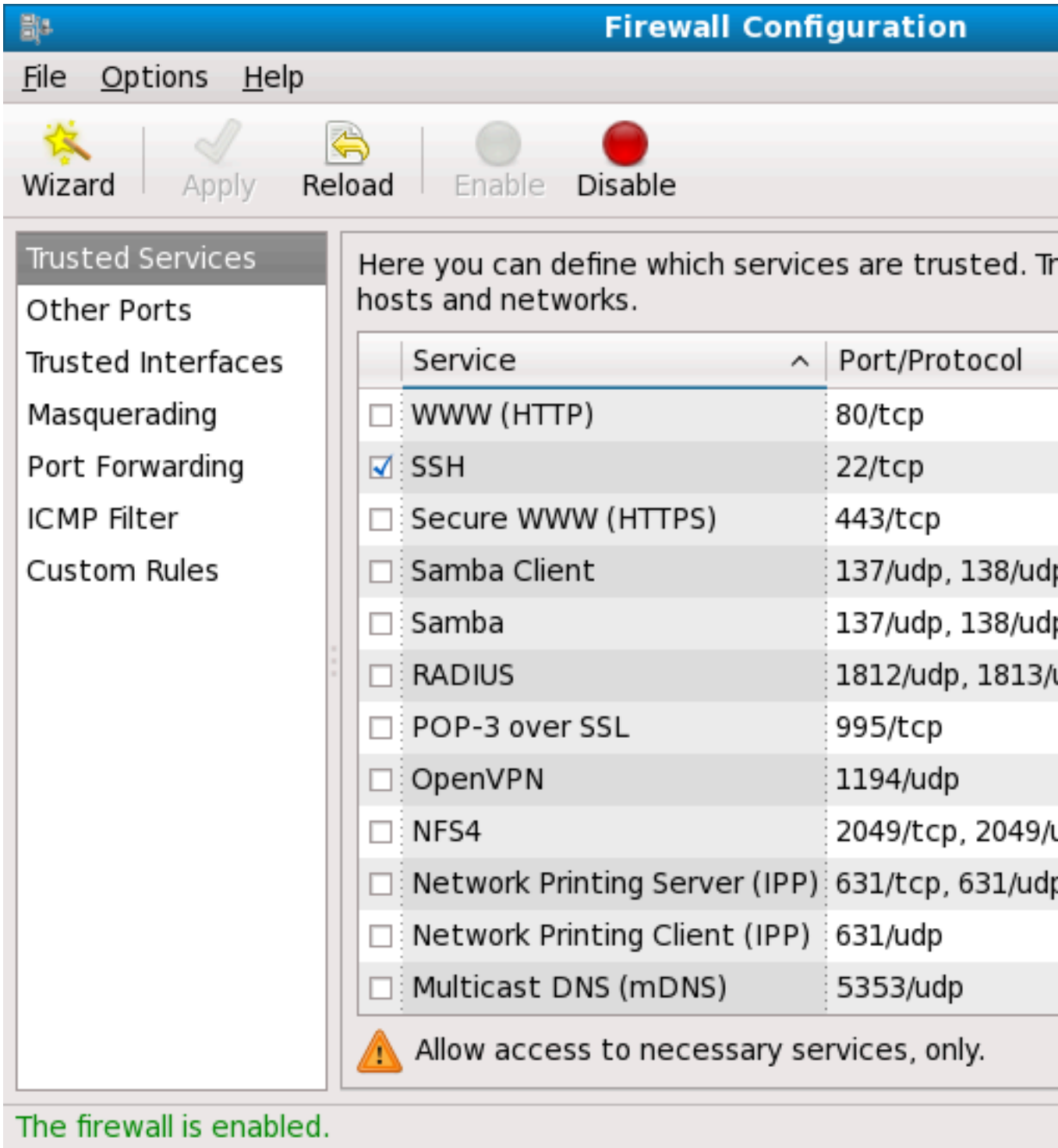
2.8.2.1. Firewallconfiguratie

Op het **Firewallconfiguratie** scherm van de Fedora installatie, heb je de mogelijkheid gekregen om een basis firewall aan te zetten evenals het toestaan van specifieke apparaten, binnenkomende servies, en poorten.

Na de installatie, kun je deze instelling veranderen met het **Firewall Configuration Tool** gereedschap.

Om deze toepassing te starten gebruik je het volgende commando:

```
[root@myServer ~] # system-config-firewall
```



Figuur 2.15. **Firewallconfiguratie**

Opmerking

De **Firewall Configuration Tool** toepassing stelt een basis firewall in. Als het systeem meer complexe regels nodig heeft, refereer je naar *Paragraaf 2.9, "IPTables"* voor details over het instellen van specifieke **`iptables`** regels.

2.8.2.2. Het aan- en uitzetten van de firewall

Selecteer een van de volgende opties voor de firewall:

- **Uitzetten** — De firewall uitzetten geeft complete toegang tot je systeem en er vindt geen beveiligingscontrole plaats. Dit moet alleen geselecteerd worden als je op een vertrouwd netwerk draait (niet het Internet) of als je een aangepaste firewall moet instellen met gebruik van het iptables commandoregel gereedschap.

  Waarschuwing

  Firewall configuraties en alle handmatige firewall regels worden opgeslagen in het **`/etc/sysconfig/iptables`** bestand. Als je kiest voor **Uitzetten** en op **OK** klikt, zullen deze configuraties en firewall regels verloren gaan.

- **Aanzetten** — Deze optie stelt je systeem in om binnenkomende verbindingen niet toe te staan als ze geen antwoord zijn op uitgaande verzoeken, zoals DNS antwoorden of DHCP verzoeken. Als toegang tot services die op deze machine draaien nodig is, kun je er voor kiezen om specifieke services door de firewall toe te laten.

  Als je jouw systeem met het Internet verbindt, maar je bent niet van plan een server te draaien, is dit de veiligste keuze.

2.8.2.3. Vertrouwde services

Het aanzetten van opties in de **Vertrouwde diensten** lijst, laat de opgegeven service door in de firewall.

WWW (HTTP)

Het HTTP protocol wordt gebruikt door Apache (en door andere Web servers) om web pagina's beschikbaar te maken. Als je van plan bent om je Web server publiek beschikbaar te maken, selecteer je dit aanvinkhokje. Deze optie is niet nodig om lokaal pagina's te bekijken of voor het ontwikkelen van web pagina's. Deze service vereist dat het **`httpd`** pakket geïnstalleerd is.

Het aanzetten van **WWW (HTTP)** zal geen poort openen voor HTTPS, de SSL versie van HTTP. Als deze service vereist is, selecteer je **Secure WWW (HTTPS)**.

FTP

Het FTP protocol wordt gebruikt voor het overbrengen van bestanden tussen machines op een netwerk. Als je van plan bent op je FTP server publiek beschikbaar te maken, selecteer je dit aanvinkhokje. Deze service vereist dat het **`vsftpd`** pakket geïnstalleerd is.

SSH

Secure Shell (SSH) is een verzameling gereedschappen voor het inloggen en uitvoeren van commando's op een machine op afstand. Om toegang op afstand naar de machine toe te staan, selecteer je dit aanvinkhokje. Deze service vereist dat het **`openssh-server`** pakket geïnstalleerd is.

Telnet

Telnet is een protocol voor het inloggen op machines op afstand. Telnet communiceert onversleuteld en biedt geen bescherming tegen netwerk snuffelen. Het toestaan van binnenkomende Telnet toegang wordt niet aanbevolen. Om toegang op afstand naar de machine

met Telnet toe te staan, selecteer je dit aanvinkhokje. Deze service vereist dat het **telnet-server** pakket geïnstalleerd is.

Mail (SMTP)
SMTP is een protocol die hosts op afstand toestaat om rechtstreeks met je machine te verbinden om mail af te leveren. Je hoeft deze service niet aan te zetten als je jouw mail ophaalt van de server van jouw ISP met gebruik van POP3 of IMAP, of als je een gereedschap zoals **fetchmail** gebruikt. Om het afleveren van mail op je machine toe te staan, selecteer je dit aanvinkhokje. Merk op dat een onjuist ingestelde SMTP server machines op afstand kan toestaan om jouw server te gebruiken voor het versturen van spam.

NFS4
Het Network File System (NFS) is een bestandsdelings protocol veel gebruikt op *NIX systemen. Versie 4 van dit protocol is veiliger dan zijn voorgangers. Als je bestanden of mappen op je systeem wilt delen met andere netwerkgebruikers, selecteer je dit aanvinkhokje.

Samba
Samba is een implementatie van het eigendomsmatige SMB netwerk protocol van Microsoft. Als je bestanden, mappen, of lokaal verbonden printers moet delen met Microsoft Windows machines, selecteer je dit aanvinkhokje.

2.8.2.4. Andere poorten

Het **Firewall Configuration Tool** gereedschap heeft een **Andere poorten** sectie voor het opgeven van specifieke IP poorten als vertrouwd voor **iptables**. Bijvoorbeeld, om IRC en Internet printing protocol (IPP) toe te staan door de firewall, voeg je de volgende poorten toe in de **Andere poorten** sectie:

```
194:tcp,631:tcp
```

2.8.2.5. De instellingen opslaan

Klik op **OK** om de veranderingen op te slaan en de firewall aan of uit te zetten. Als **Aanzetten** was geselecteerd, worden de geselecteerde opties vertaald naar **iptables** commando's en weggeschreven naar het **/etc/sysconfig/iptables** bestand. De **iptables** service wordt ook gestart zodat de firewall onmiddellijk na het opslaan van de geselecteerde opties geactiveerd wordt. Als **Uitzetten** was geselecteerd, wordt het **/etc/sysconfig/iptables** bestand verwijderd en de **iptables** service wordt onmiddellijk gestopt.

De geselecteerde opties worden ook naar het **/etc/sysconfig/system-config-securitylevel** bestand geschreven zodat de instellingen hersteld kunnen worden als de toepassing de volgende keer opstart. Bewerk dit bestand niet handmatig.

Hoewel de firewall onmiddellijk geactiveerd wordt, wordt de **iptables** service niet ingesteld om automatisch op de starten tijden het opstarten van de machine. Refereer naar *Paragraaf 2.8.2.6, “Het activeren van de IPTables service”* voor meer informatie.

2.8.2.6. Het activeren van de IPTables service

De firewall regels zijn alleen actief als de **iptables** service draait. Om de service handmatig te starten, gebruik je het volgende commando:

```
[root@myServer ~] # service iptables restart
```

Om er zeker van te zijn dat **iptables** start als het systeem opgestart wordt, gebruik je het volgende commando:

```
[root@myServer ~] # chkconfig --level 345 iptables on
```

2.8.3. IPTables gebruiken

De eerste stap voor het gebruik van **iptables** is om de **iptables** service te starten. Gebruik het volgende commando om de **iptables** service te starten:

```
[root@myServer ~] # service iptables start
```

Opmerking

De **ip6tables** service kan uitgezet worden als je alleen de **iptables** service gebruikt. Als je de **ip6tables** service uitzet, denk er dan aan om ook het IPv6 netwerk uit te zetten. Laat nooit een netwerk apparaat actief zonder een bijbehorende firewall.

Om te forceren dat **iptables** standaard opstart als het systeem opgestart wordt, gebruik je het volgende commando:

```
[root@myServer ~] # chkconfig --level 345 iptables on
```

Dit forceert dat **iptables** start als het systeem opgestart wordt in runlevel 3, 4, of 5.

2.8.3.1. IPTables commando syntax

Het volgende voorbeeld **iptables** commando laat de basis commando syntax zien:

```
[root@myServer ~ ] # iptables -A <keten> -j <doel>
```

De `-A` optie specificeert dat de regel toegevoegd wordt aan <*keten*>. Elke keten bestaat uit een of meer *regels*, en staat daarom ook bekend als een *ketenstelsel*.

De drie ingebouwde ketens zijn INPUT, OUTPUT, en FORWARD. Deze ketens zijn permanent en kunnen niet verwijderd worden. De keten specificeert het punt waarop een pakket gemanipuleerd wordt.

De `-j` <*doel*> optie specificeert het doel van de regel; d.w.z. wat te doen als het pakket overeenkomt met de regel. Voorbeelden van ingebouwde doelen zijn ACCEPT, DROP, en REJECT.

Refereer naar de **iptables** manual pagina voor meer informatie over beschikbare, ketens, opties, en doelen.

2.8.3.2. Basis firewall tactieken

Het vaststellen van basis firewall tactieken maakt een ondergrond voor het bouwen van meer gedetaileerde, door de gebruiker gedefinieerde regels.

Elke **iptables** keten bestaat uit een standaard tactiek, en geen of een aantal regels die samenwerken met de standaard tactiek om het gehele regelstelsel voor de firewall te definiëren.

De standaard tactiek voor een regel kan DROP of ACCEPT zijn. Beheerders met beveiliging in gedachten implementeren gewoonlijk een standaard tactiek van DROP, en staan alleen specifieke pakketten toe op een van-geval-tot-geval basis. Bijvoorbeeld, de volgende tactiek blokkeert alle ingaande en uitgaande pakketten op een netwerk gateway:

```
[root@myServer ~ ] # iptables -P INPUT DROP
[root@myServer ~ ] # iptables -P OUTPUT DROP
```

Het wordt ook aanbevolen om alle *forwarded packets* — netwerk verkeer dat geleid moet worden van de firewall naar zijn bestemmings node — ook te verbieden, om interne cliënten te beperken voor onbedoelde blootstelling aan het Internet. Om dit te doen gebruik je de volgende regel:

```
[root@myServer ~ ] # iptables -P FORWARD DROP
```

Als je de standaard tactiek voor iedere keten hebt vastgesteld, kun je andere regels maken en opslaan voor je specifieke netwerk en beveiligings vereisten.

De volgende paragrafen beschrijven hoe je iptables regels opslaat en geven een aantal regels aan die je misschien wilt gebruiken voor het bouwen van jouw iptables firewall.

2.8.3.3. Opslaan en terugzetten van IPTables regels

Veranderingen in **iptables** zijn tijdelijk, als het systeem opnieuw opgestart wordt of als de **iptables** service opnieuw opgestart wordt, worden de regels automatisch verwijderd en de standaarden worden terug gezet. Om de regels te bewaren zodat ze geladen worden als de **iptables** service gestart wordt, gebruik je het volgende commando:

```
[root@myServer ~ ] # service iptables save
```

De regels worden opgeslagen in het bestand **/etc/sysconfig/iptables** en worden iedere keer toegepast als de service gestart wordt of als de machine opnieuw opgestart wordt.

2.8.4. Algemene IPTables filtering

Een van de belangrijkste aspecten van netwerkbeveiliging is het voorkomen dat aanvallers op afstand toegang krijgen tot een LAN. De integriteit van een LAN moet beschermd worden tegen kwaadwillige gebruikers door het gebruik van strenge firewall regels.

Met een standaard tactiek om alle binnenkomende, uitgaande, een doorgestuurde pakketten te blokkeren, is het echter onmogelijk voor de firewall/gateway en interne LAN gebruikers om met elkaar of met externe hulpbronnen te communiceren.

Om gebruikers netwerk gerelateerde functies toe te staan en netwerk toepassingen te gebruiken, moeten beheerders bepaalde poorten openen voor communicatie.

Bijvoorbeeld, om toegang toe te staan tot poort 80 *op de firewall*, voeg je de volgende regel toe:

```
[root@myServer ~ ] # iptables -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
```

Dit staat gebruikers toe om websites te bezoeken die communiceren met gebruik van de standaard poort 80. Om toegang toe te staan voor beveiligde websites (bijvoorbeeld, https://www.example.com/), moet je als volgt ook toegang geven tot poort 443:

```
[root@myServer ~ ] # iptables -A INPUT -p tcp -m tcp --dport 443 -j ACCEPT
```

Belangrijk

Bij het maken van een **`iptables`** regelstelsel is de volgorde belangrijk.

Als een regel specificeert dat alle pakketten van het 192.168.100.0/24 subnet gedropd moeten worden, en dit wordt gevolgd door een regel die pakketten toestaat van 192.168.100.13 (welke binnen het gedropte subnet ligt), wordt de tweede regel genegeerd.

De regel die pakketten toestaat van 192.168.100.13 moet voor de regel komen die de rest van het subnet dropped.

Om een regel in te voegen op een specifieke locatie in een bestaande keten, gebruik je de `-I` optie. Bijvoorbeeld:

```
[root@myServer ~ ] # iptables -I INPUT 1 -i lo -p all -j ACCEPT
```

Deze regel wordt ingevoegd als de eerste regel in de INPUT keten om local loopback apparaat verkeer toe te staan.

Het kan voorkomen dat je toegang op afstand naar het LAN nodig hebt. Beveiligde services, bijvoorbeeld SSH, kunnen gebruikt worden voor versleutelde verbinding op afstand naar LAN services.

Beheerders met hulpbronnen gebaseerd op PPP (zoals modems of bulk ISP accounts), kunnen dial-up toegang gebruiken om firewall beperkingen veilig te omzeilen. Omdat het directe verbindingen zijn, bevinden modem verbindingen zich gewoonlijk achter een firewall/gateway.

Voor gebruikers op afstand met breedband verbindingen, kunnen echter speciale gevallen gemaakt worden. Je kunt **`iptables`** instellen om verbindingen van SSH cliënten op afstand te accepteren. Bijvoorbeeld, de volgende regels staan SSH toegang op afstaand toe:

```
[root@myServer ~ ] # iptables -A INPUT -p tcp --dport 22 -j ACCEPT
[root@myServer ~ ] # iptables -A OUTPUT -p tcp --sport 22 -j ACCEPT
```

Deze regels staan binnenkomende en uitgaande toegang toe voor een individueel systeem, zoals een enkele PC direct verbonden met het Internet of een firewall/gateway. Sta echter nodes achter de firewall/gateway toegang tot deze services niet toe. Om LAN toegang naar deze services toe te staan, kun je *Network Address Translation* (NAT) gebruiken met **`iptables`** filterings regels.

2.8.5. FORWARD en NAT regels

De meeste ISP's bieden slechts een beperkt aantal publiek bereikbare IP adressen aan de organisaties die ze bedienen.

Beheerders moeten daarom een alternatieve manier vinden om toegang tot Internet services te delen zonder aan iedere node op het LAN een publiek IP adres te geven.

Rand routers (zoals firewalls) kunnen binnenkomende verzendingen van het Internet ontvangen en de pakketten door sturen naar de bedoelde LAN node. Tegelijkertijd kunnen firewalls/gateways ook uitgaande verzoeken doorsturen van een LAN node naar de Internet service op afstand.

Dit doorsturen van netwerk verkeer kan soms gevaarlijk worden, zeker met de beschikbaarheid van moderne kraak gereedschappen die zich voor kunnen doen als *interne* IP adressen, en die de machine van de aanvaller op afstand laten opereren als een node op je LAN.

Om dit te voorkomen, biedt **`iptables`** routing en forwarding tactieken die geïmplementeerd kunnen worden om abnormaal gebruik van netwerk hulpbronnen te voorkomen.

De `FORWARD` keten laat een beheerder controleren welke pakketten binnen een LAN doorgestuurd kunnen worden. Bijvoorbeeld, om doorsturen voor het gehele LAN toe te staan (aannemende dat aan de firewall/gateway een intern IP adres op eth1 is toegekend), gebruik je de volgende regels:

```
[root@myServer ~ ] # iptables -A FORWARD -i eth1 -j ACCEPT
[root@myServer ~ ] # iptables -A FORWARD -o eth1 -j ACCEPT
```

Deze regel geeft systemen achter de firewall/gateway toegang tot het interne netwerk. De gateway stuurt pakketten van een LAN node naar zijn bedoelde bestemmings node, door alle pakketten door zijn **`eth1`** apparaat te sturen.

Opmerking

Standaard zet de IPv4 tactiek in Fedora kernels ondersteuning voor IP forwarding uit. Dit belet machines die Fedora draaien om te werken als specifieke rand routers. Om IP forwarding aan te zetten, gebruik je het volgende commando:

```
[root@myServer ~ ] # sysctl -w net.ipv4.ip_forward=1
```

De instellingsverandering is alleen geldig voor de huidige sessie; het verdwijnt na een systeem herstart of een netwerk service herstart. Om IP forwarding permanent aan te zetten, bewerk je het **`/etc/sysctl.conf`** bestand als volgt:

Zoek de volgende regel:

```
net.ipv4.ip_forward = 0
```

Verander de regel als volgt:

```
net.ipv4.ip_forward = 1
```

Gebruik het volgende commando om de verandering in het **`sysctl.conf`** bestand te activeren:

```
[root@myServer ~ ] # sysctl -p /etc/sysctl.conf
```

2.8.5.1. Postrouting en IP masquerade

Het accepteren van doorgestuurde pakketten via het interne IP apparaat van de firewall laat LAN nodes met elkaar communiceren; ze kunnen echter nog steeds niet extern met het Internet communiceren.

Om LAN nodes met privé IP adressen te laten communiceren met externe publieke netwerken, configureer je de firewall voor *IP masquerading*, welke verzoeken van LAN nodes met het IP adres van het externe apparaat van de firewall (in dit geval, eth0) maskeert:

```
[root@myServer ~ ] # iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Deze regel gebruikt NAT pakket matching tabel (`-t nat`) en specificeert de ingebouwde POSTROUTING keten voor NAT (`-A POSTROUTING`) op het externe netwerkapparaat van de firewall (`-o eth0`).

POSTROUTING staat toe dat pakketten veranderd worden zodra ze het externe apparaat van de firewall verlaten.

Het `-j MASQUERADE` doel is opgegeven om het privé IP adres van een node te maskeren met het externe IP adres van de firewall/gateway.

2.8.5.2. Prerouting

Als je een server in je interne netwerk hebt die je extern beschikbaar wilt maken, kun je het `-j DNAT` doel van de PREROUTING keten in NAT gebruiken om een bestemmings IP adres en poort te specificeren waarnaar binnenkomende pakketten die een verbinding verzoeken naar je interne service geforward kunnen worden.

Bijvoorbeeld, als je binnenkomende HTTP verzoeken wilt forwarden naar je specifieke Apache HTTP server op 172.31.0.23, gebruik je het volgende commando:

```
[root@myServer ~ ] # iptables -t nat -A PREROUTING -i eth0 -p tcp --dport
 80 -j DNAT --to 172.31.0.23:80
```

Deze regel specificeert dat de nat tabel de ingebouwde PREROUTING gebruikt om binnenkomende HTTP verzoeken uitsluitend door te sturen naar het opgegeven bestemmings IP adres 172.31.0.23.

Opmerking

Als je een standaard DROP tactiek in je FORWARD keten hebt, moet je een regel toevoegen om alle binnenkomende HTTP verzoeken te forwarden zodat bestemmings NAT routing mogelijk is. Om dit te doen gebruik je het volgende commando:

```
[root@myServer ~ ] # iptables -A FORWARD -i eth0 -p tcp --dport 80
 -d 172.31.0.23 -j ACCEPT
```

Deze regel forward alle binnenkomende HTTP verzoeken van de firewall naar de bedoelde bestemming; de Apache HTTP server achter de firewall.

2.8.5.3. DMZ's en IPTables

Je kunt **`iptables`** regels maken om verkeer naar een bepaalde machine, zoals een specifieke HTTP of FTP server, te sturen in een *demilitarized zone* (DMZ). Een DMZ is een speciaal lokaal subnetwerk bestemd om services op een publieke drager (zoals het Internet) aan te bieden.

Bijvoorbeeld, om een regel te maken om binnenkomende HTTP verzoeken door te sturen naar een specifieke HTTP server op 10.0.4.2 (buiten de 192.168.1.0/24 reeks van het LAN), gebruikt NAT de `PREROUTING` tabel om de pakketten door te sturen naar de juiste bestemming:

```
[root@myServer ~ ] # iptables -t nat -A PREROUTING -i eth0 -p tcp --dport
 80 -j DNAT --to-destination 10.0.4.2:80
```

Met dit commando, worden alle HTTP verbindingen naar poort 80 van buiten het LAN doorgestuurd naar de HTTP server op een netwerk los van de rest van het interne netwerk. Deze vorm van netwerkverdeling kan veiliger zijn dat het toestaan van HTTP verbindingen naar een machine in het netwerk.

Als de HTTP server is ingesteld om veilige verbindingen te accepteren, dan moet poort 443 ook doorgestuurd orden.

2.8.6. Kwaadwillige software en bedrogen IP adressen

Uitgebreidere regels kunnen gemaakt worden om toegang te controleren naar specifieke subnetten, of zelfs specifieke nodes, binnen een LAN. Je kunt ook bepaalde verdachte toepassingen of programma's zoals, trojans, wormen, en andere cliënt/server virussen beperken om contact op te nemen met hun server.

Bijvoorbeeld, sommige trojans scannen netwerken voor services op poorten van 31337 tot 31340 (de *elite* poorten genoemd in krakers terminologie).

Omdat er geen legitieme services zijn die communiceren via deze niet-standaard poorten, kan het blokkeren hiervan de kans effectief verkleinen dat potentieel besmette nodes op je netwerk onafhankelijk communiceren met hun meester servers op afstand.

De volgende regels verbieden alle TCP verkeer dat probeert poort 31337 te gebruiken:

```
[root@myServer ~ ] # iptables -A OUTPUT -o eth0 -p tcp --dport 31337 --
sport 31337 -j DROP
[root@myServer ~ ] # iptables -A FORWARD -o eth0 -p tcp --dport 31337 --
sport 31337 -j DROP
```

Je kunt ook buiten verbindingen blokkeren die proberen zich voor te doen als een adres uit privé IP adres reeksen om je LAN te infiltreren.

Bijvoorbeeld, als je LAN de 192.168.1.0/24 reeks gebruikt, kun je een regel maken die het netwerkapparaat die met het Internet verbonden is (bijvoorbeeld, eth0) instrueert om alle pakketten naar dat apparaat te weigeren met een adres die in jouw LAN IP reeks ligt.

Omdat het aanbevolen is om alle forwarded pakketten als een standaard tactiek te weigeren, wordt elk ander vermomd IP adres naar het apparaat dat extern aangesloten is (eth0) automatisch geweigerd.

```
[root@myServer ~ ] # iptables -A FORWARD -s 192.168.1.0/24 -i eth0 -j DROP
```

Opmerking

Er is een verschil tussen de `DROP` en `REJECT` doelen in samenhang met *toegevoegde* regels.

Het `REJECT` doel verbiedt toegang en stuurt een `connection refused` fout terug naar de gebruikers die proberen met de service te verbinden. Het `DROP` doel, zoals de naam al aangeeft, verwijdert het pakket zonder enige waarschuwing.

Beheerders kunnen hun eigen goeddunken gebruiken bij het gebruik van deze doelen. Om echter verwarring bij gebruikers en herhaalde verbindings pogingen te vermijden, wordt het `REJECT` doel aanbevolen.

2.8.7. IPTables en verbindingen volgen

Je kunt verbindingen naar services inspecteren en beperken gebaseerd op hun *verbindingstoestand*. Een module binnen **`iptables`** gebruikt een methode, *verbindingen volgen* genaamd, om informatie op te slaan over binnenkomende verbindingen. Je kunt toegang toestaan of weigeren op basis van de volgende verbindingstoestanden:

- `NEW` — Een pakket verzoekt een nieuwe verbinding, zoals een HTTP verzoek.
- `ESTABLISHED` — Een pakket dat onderdeel is van een bestaande verbinding.
- `RELATED` — Een pakket dat een nieuwe verbinding vraagt maar onderdeel is van een bestaande verbinding. Bijvoorbeeld, FTP gebruikt poort 21 om een verbinding op te zetten, maar de data overdracht gebeurt op een andere poort (gewoonlijk poort 20).
- `INVALID` — Een pakket dat van geen enkele verbinding in de verbindingen volgen tabel onderdeel is.

Je kunt de met-toestand functionaliteit van **`iptables`** verbindingen volgen gebruiken bij elk netwerkprotocol, zelfs als het protocol zelf zonder-toestand is (zoals UDP). Het volgende voorbeeld laat een regel zien die verbindingen volgen gebruikt om alleen pakketten door te sturen die behoren bij een bestaande verbinding:

```
[root@myServer ~ ] # iptables -A FORWARD -m state --state
 ESTABLISHED,RELATED -j ACCEPT
```

2.8.8. IPv6

De introductie van het volgende generatie Internet protocol met de naam IPv6, gaat boven de 32-bit adres limiet van IPv4 (of IP). IPv6 ondersteunt 128-bit adressen, en vervoersnetwerken die bewust zijn van IPv6 kunnen daarom een groter aantal adressen aan dan IPv4.

Fedora supports IPv6 firewall rules using the Netfilter 6 subsystem and the **`ip6tables`** command. In Fedora 12, both IPv4 and IPv6 services are enabled by default.

De **`ip6tables`** commando syntax is in alle aspecten identiek aan **`iptables`**, behalve dat het 128-bit adressen ondersteunt. Bijvoorbeeld, je gebruikt het volgende commando om SSH verbindingen aan te zetten op een netwerk server bewust van IPv6:

```
[root@myServer ~ ] # ip6tables -A INPUT -i eth0 -p tcp -s
 3ffe:ffff:100::1/128 --dport 22 -j ACCEPT
```

Voor meer informatie over IPv6 netwerken, refereer je naar de IPv6 informatie pagina op *http://www.ipv6.org/*.

2.8.9. Extra hulpbronnen

Er zijn verscheidene aspecten van firewalls en het Linux Netfilter subsysteem die in dit hoofdstuk niet behandeld konden worden. Voor meer informatie refereer je naar de volgende hulpbronnen.

2.8.9.1. geïnstalleerde firewall documentatie

- Refereer naar *Paragraaf 2.9, "IPTables"* voor meer details over het **iptables** commando, inclusief definities voor vele commando opties.
- De **iptables** manual pagina bevat een korte samenvatting van de verschillende opties.

2.8.9.2. Nuttige firewall websites

- *http://www.netfilter.org/* — De officiële thuispagina van het Netfilter en **iptables** project.
- *http://www.tldp.org/* — De Linux Documentation Project bevat verscheidene nuttige handleidingen met betrekking tot het maken en beheren van firewalls.
- *http://www.iana.org/assignments/port-numbers* — De officiële lijst van geregistreerde en algemene service poorten zoals toegekend door de Internet Assigned Numbers Authority.

2.8.9.3. Gerelateerde documentatie

- *Red Hat Linux Firewalls*, door Bill McCarty; Red Hat Press — een uitgebreide referentie voor het bouwen netwerk en server firewalls met gebruiken van open bron bron pakket filtering technologie zoals Netfilter en **iptables**. Het bevat onderwerpen die het analyseren van firewall logs behandelen, het ontwikkelen van firewall regels, en het aanpassen van je firewall met gebruik van meerdere grafische gereedschappen.
- *Linux Firewalls*, door Robert Ziegler; New Riders Press — bevat een schat van informatie over het bouwen vn firewalls met gebruik van zowel **ipchains** van kernel 2.2 als Netfilter en **iptables**. Extra beveiligings onderwerpen zoals problemen over toegang op afstand en indringings detectie systemen worden ook behandeld.

2.9. IPTables

Onderdeel van Fedora zijn geavanceerde gereedschappen voor netwerk *pakket filtering* — het proces van het controleren van netwerkpakketten tijdens het binnenkomen, doorlopen, en verlaten van de netwerkstack in de kernel. Kernel versies voor 2.4 vertrouwden op **ipchains** voor pakket filtering en gebruikten lijsten met regels toe toegepast werden voor pakketten in iedere stap van het filter proces. De 2.4 kernel introduceerde **iptables** (ook *netfilter* genaamd), welke overeenkomt met **ipchains** maar die het bereik en de controle beschikbaar voor het filteren van netwerk pakketten verder uitbreidt.

This chapter focuses on packet filtering basics, explains various options available with **iptables** commands, and explains how filtering rules can be preserved between system reboots.

Refereer naar *Paragraaf 2.9.6, “Extra hulpbronnen”* voor instructies over het maken van **iptables** regels en het instellen van een firewall gebaseerd op deze regels.

Belangrijk

Het standaard firewall mechanisme in 2.4 en latere kernels is **iptables**, maar **iptables** kan niet gebruikt worden als **ipchains** al draait. Als **ipchains** aanwezig is tijdens het opstarten, geeft de kernel een fout en wordt **iptables** niet opgestart.

De functionaliteit van **ipchains** wordt niet getroffen door deze fouten.

2.9.1. Pakket filtering

De Linux kernel gebruikt **Netfilter** om pakketten te filteren en staat toe dat sommige van hen ontvangen of doorgelaten worden, terwijl gestopt worden. Deze functionaliteit is ingebouwd in de Linux kernel, en heeft drie ingebouwde *tabellen* of *regel lijsten*, namelijk de volgende:

- `filter` — De standaard tabel voor het afhandelen van netwerk pakketten.
- `nat` — Wordt gebruikt om pakketten te veranderen die een nieuwe verbinding maken en wordt gebruikt voor *Network Address Translation* (*NAT*).
- `mangle` — Wordt gebruikt voor specifieke soorten pakket verandering.

Elke tabel heeft een groep van ingebouwde *ketens*, welke overeenkomen met de acties die **netfilter** uitvoert op de pakketten.

De ingebouwde ketens voor de `filter` tabel zijn de volgende:

- *INPUT* — Wordt toegepast op netwerk pakketten die bestemd zijn voor de host.
- *OUTPUT* — Wordt toegepast op lokaal aangemaakte pakketten.
- *FORWARD* — Wordt toegepast op netwerk pakketten die door de host geleid worden.

De ingebouwde ketens voor de `nat` tabel zijn de volgende:

- *PREROUTING* — Verandert netwerk pakketten als ze binnenkomen.
- *OUTPUT* — Verandert lokaal aangemaakte pakketten voordat ze verzonden worden.
- *POSTROUTING* — Verandert netwerk pakketten voordat ze verzonden worden.

De ingebuwde ketens voor de `mangle` tabel zijn de volgende:

- *INPUT* — Verandert netwerk pakketten die bestemd zijn voor de host.
- *OUTPUT* — Verandert lokaal aangemaakte pakketten voordat ze verzonden worden.
- *FORWARD* — Verandert netwerk pakketten die door de host geleid worden.
- *PREROUTING* — Verandert binnenkomende netwerk pakketten voordat ze verder geleid worden.
- *POSTROUTING* — Verandert netwerk pakketten voordat ze verzonden worden.

Elk netwerkpakket dat wordt ontvangen of verstuurd door een Linux systeem wordt onderworpen aan tenminste een tabel. Een pakket kan echter onderworpen zijn aan meerdere regels binnen elke tabel voordat het aan het einde van de keten verschijnt. De structuur en het doel van deze regels kan anders zijn, maar gewoonlijk proberen ze een pakket te identificeren als het komt van of gaat naar een bepaald IP adres, of een verzameling van adressen, bij het gebruik van een bepaalde protocol of netwerk service.

Opmerking

Standaard worden firewall regels opgeslagen in de **`/etc/sysconfig/iptables`** of **`/etc/sysconfig/ip6tables`** bestanden.

De **`iptables`** service start voor elke andere aan DNS gerelateerde service als een Linux systeem wordt opgestart. Cit betekent dat firewall regels alleen naar numerieke IP adressen (bijvoorbeeld, 192.168.0.1) kunnen refereren. Domein namen (bijvoorbeeld, host.example.com) veroorzaken in zulke regels fouten.

Als een pakket overeenkomt met een bepaalde regel in een van de tabellen wordt, onafhankelijk van hun bestemming, een *doel* of actie op hen uitgevoerd. Als de regel een **`ACCEPT`** doel voor een overeenkomend pakket specificeert, slaat het pakket de rest van de regel controles over en wordt het toegestaan om te vervolgen naar zijn bestemming. Als een regel een **`DROP`** doel specificeert, wordt dat pakket toegang tot het systeem geweigerd en er wordt niets terug gestuurd naar de host die het pakket verzond. Als de regel een **`QUEUE`** doel specificeert, wordt het pakket doorgegeven naar de gebruikersruimte. Als de regel het optionele **`REJECT`** doel specificeerde, wordt het pakket geweigerd, maar een fout pakket wordt terug gestuurd naar de verzender van het pakket.

Elke keten heeft een standaard tactiek voor **`ACCEPT`**, **`DROP`**, **`REJECT`**, of **`QUEUE`**. Als geen van deze regels inb de keten van toepassing is op het pakket, dan wordt het pakket behandeld overeenkomstig de standaard tactiek.

Het **`iptables`** commando configureert deze tabellen, en zet zo nodig ook nieuwe tabellen op.

2.9.2. Commando opties voor IPTables

Regels voor het filteren van pakketten worden gemaakt met behulp van het **`iptables`** commando. De volgende aspecten van het pakket worden het meest als criteria:

- *Pakket type* — Specificeert het type van de pakketten die het commando filtert.
- *Pakket bron/bestemming* — Specificeert welke pakketten door het commando gefilterd worden gebaseerd op de bron of de bestemming van het pakket
- *Doel* — specificeert welke actie wordt ondernomen voor pakketten die overeenkomen met de vorige criteria.

Refereer naar *Paragraaf 2.9.2.4, “IPTables overeenkomst opties”* en *Paragraaf 2.9.2.5, “Doel opties”* voor meer informatie over specifieke opties die deze aspecten van een pakket aangaan.

De opties die gebruikt worden bij specifieke **`iptables`** regels moeten logisch gegroepeerd worden, gebaseerd op het doel en de condities van de gehele regel, wil de regel geldig zijn.

2.9.2.1. Structuur van de IPTables commando opties

Vele **`iptables`** commando's hebben de volgende structuur:

```
iptables [-t <tabel-naam>] <commando> <keten-naam>
\ <parameter-1> <optie-1> \ <parameter-n> <optie-n>
```

<*tabel-naam*> — Specificeert op welke tabel de regel betrekking heeft. Als het weggelaten wordt, wordt de `filter` tabel gebruikt.

<*commando*> — Specificeert de uit te voeren actie, zoals het toevoegen of verwijderen van een regel.

<*keten-naam*> — Specificeert de keten die veranderd, gemaakt, of verwijderd moet worden.

<*parameter*>-<*optie*> paren — Parameters en bijbehorende opties die aangeven hoe het pakket verwerkt moet worden als het overeenkomt met de regel.

De lengte en complexiteit van een **iptables** commando kan behoorlijk veranderen, dit afhankelijk van zijn doel.

Bijvoorbeeld, een commando om een regel van een keten te verwijderen kan erg kort zijn:

iptables -D <*keten-naam*> <*regel-nummer*>

In tegenstelling hiermee kan een commando dat een regel toevoegt welke pakketten filtert van een bepaald subnet met gebruik van verscheidene specifieke parameters en opties kan behoorlijk lang zijn. Bij het maken van **iptables** commando's is het belangrijk om te onthouden dat sommige parameters en opties extra parameters en opties nodig hebben om een geldige regel te maken. Dit kan een explosie veroorzaken, als de extra parameters weer nieuwe parameters nodig hebben. Slechts nadat aan elke parameter en optie die een ander paar opties nodig heeft voldaan is, is de regel geldig.

Type **iptables -h** om een uitgebreide lijst te bekijken van alle **iptables** commando structuren.

2.9.2.2. Commando opties

Commando opties instrueren **iptables** om een bepaalde actie uit te voeren. Slechts een commando optie is toegestaan per **iptables** commando. Met uitzondering van het help commando, worden alle commando's in hoofdletters geschreven.

De **iptables** commando's zijn de volgende:

- `-A` — Voeg de regel toe aan het einde van de gespecificeerde keten. In tegenstelling tot de `-I` optie hieronder beschreven, accepteert het geen geheel getal als argument. Het voegt de regel altijd toe aan het einde van de keten.
- `-C` — Controleert een bepaalde regel voordat het deze toevoegt aan de keten door de gebruiker opgegeven. Dit commando kan je helpen om complexe **iptables** regels te maken omdat het vraagt om extra parameters en opties.
- `-D <integer> | <regel>` — Verwijdert een regel in een bepaalde keten volgens nummer (zoals 5 voor de vijfde regel in een keten), of volgens regel specificatie. De regel specificatie moet exact overeenkomen met een bestaande regel.
- `-E` — Verandert de naam van een keten opgegeven door een gebruiker. Een keten opgegeven door een gebruiker is een keten anders dan de standaard, voor-gedefinieerde ketens. (Refereer naar de `-N` optie hier beneden voor informatie over het maken van ketens gedefinieerd door een gebruiker). Dit is een cosmetische verandering en heeft geen effect op de structuur.

Opmerking

Als je probeert een van de standaard ketens van naam te veranderen, rapporteert het systeem een `Match not found` fout. Je kunt de standaard ketens geen andere naam geven.

- `-F` — Verschoont de geselecteerde keten, wat effectief elke regel in de keten verwijdert. Als geen keten is opgegeven, verschoont dit commando elke regel van elke keten.
- `-h` — Geeft een lijst van commando structuren, en ook een korte samenvatting van commando parameters en opties.
- `-I [<integer>]` — Voeg in regel toe in de opgegeven keten op de plek aangegeven door het gehele getal opgegeven door de gebruiker. Als geen argument wordt opgegeven, wordt de regel bovenin de keten toegevoegd.

Belangrijk

Zoals eerder is aangegeven, bepaalt de volgorde van de regels in een keten welke regels voor welke pakketten gelden. Dit is belangrijk om te onthouden bij het toevoegen van regels met zowel de `-A` als de `-I` optie.

Dit is in het bijzonder belangrijk als regels toegevoegd worden met `-I` met een integer argument. Als je een bestaand nummer opgeeft bij het toevoegen van een regel aan een keten, voegt **`iptables`** de nieuwe regel toe *voor* (of boven) de bestaande regel.

- `-L` — Geeft een lijst van alle regels in de keten die na het commando gespecificeerd zijn. Om alle regels in de keten in de standaard `filter` tabel te laten zien, hoef je geen keten of tabel op te geven. Anders moet de volgende syntax gebruikt worden om de regels te laten zien van een specifieke keten in een specifieke tabel:

```
iptables -L <keten-naam> -t <tabel-naam>
```

Extra opties voor de `-L` commando optie, welke regel nummers geven en meer uitgebreide regel beschrijving toestaan, worden beschreven in *Paragraaf 2.9.2.6, “Lijst opties”*.

- `-N` — Maakt een nieuwe keten aan met de opgegeven naam. De keten naam moet uniek zijn, anders wordt een foutboodschap getoond.
- `-P` — Stelt de standaard tactiek in voor de opgegeven keten, zodat pakketten die door een gehele keten gaan zonder dat ze overeenkomen met een regel, naar het opgegeven doel gestuurd worden, zoals ACCEPT of DROP.
- `-R` — Vervangt een regel in de opgegeven keten. Het nummer van de regel moet opgegeven worden na de naam van de keten. De eerste regel in de keten komt overeen met regel nummer een.
- `-X` — Verwijdert een door de gebruiker opgegeven keten. Je kunt geen ingebouwde keten verwijderen.
- `-Z` — Zet de byte en pakket tellers in alle ketens voor een tabel op nul.

2.9.2.3. IPTables parameter opties

Bepaalde **iptables** commando's, zoals diegene die gebruikt worden voor het toevoegen, aansluiten, verwijderen, invoegen, of vervangen van regels in een bepaalde keten, vereisen verschillende parameters om een pakket filterings regel te maken.

- `-c` — Zet de tellers voor een bepaalde regel op nul. Deze parameter accepteert de PKTS en BYTES opties op gewenste teller op te geven.
- `-d` — Stelt de bestemmings hostnaam, IP adres, of netwerk in van een pakket dat overeenkomt met de regel. Voor het overeenkomen met een netwerk, worden de volgende IP adres/netmasker formaten ondersteund:
  - *N.N.N.N/M.M.M.M* — Waarin *N.N.N.N* de IP adres reeks is en *M.M.M.M* het netmasker.
  - *N.N.N.N/M* — Waarin*N.N.N.N* de IP adres reeks is en *M* het bitmasker.
- `-f` — Laat deze regel alleen werken op gefragmenteerde pakketten.

  Je kunt de uitroepteken (`!`) optie na deze parameter gebruiken om op te geven dat alleen ongefragmenteerde pakketten als overeenkomend worden beschouwd.

- `-i` — Stelt de binnenkomende interface in, zoals `eth0` of `ppp0`. Met **iptables** mag deze optionele parameter alleen gebruikt worden in de INPUT en FORWARD ketens als het met de `filter` tabel gebruikt wordt en de PREROUTING keten met de `nat` en `mangle` tabellen.

  Deze parameter ondersteunt ook de volgende speciale opties:

  - Uitroepteken (`!`) — Keert de instructie om, wat betekent dat voor alle opgegeven interfaces deze regel niet geldt.
  - Plus (+) — Een wildcard karakter gebruikt om overeen te komen met alle interfaces die overeenkomen met de opgegeven reeks karakters. Bijvoorbeeld, de parameter `-i eth+` is van toepassing voor alle Ethernet interfaces maar sluiten alle andere interfaces uit, zoals `ppp0`.

  Als de `-i` parameter wordt gebruikt maar er is geen interface opgegeven, dan geldt de regel voor elke interface.

- `-j` — Springt naar het opgegeven doel als een pakket overeenkomt met een bepaalde regel.

  De standaard doelen zijn `ACCEPT`, `DROP`, `QUEUE`, en `RETURN`.

  Uitgebreide opties zijn ook beschikbaar met behulp van modules die standaard geladen worden met het **iptables** RPM pakket van Fedora. Geldige doelen in deze modules zijn onder andere `LOG`,

`MARK`, en `REJECT`. Refereer naar de **`iptables`** manual pagina voor meer informatie over deze en andere doelen.

Deze optie kan ook gebruikt worden om een pakket dat overeenkomt met een bepaalde regel door te sturen naar een door de gebruiker gedefinieerde keten buiten de huidige keten zodat andere regels op het pakket toegepast kunnen worden.

Als geen doel is opgegeven, gaat het pakket voorbij de regel zonder dat er actie ondernomen wordt. De teller voor deze regel wordt echter met een verhoogd.

- `-o` — Stelt de uitgaande netwerk interface in voor een regel. Deze optie is alleen geldig voor OUTPUT en FORWARD ketens in de `filter` tabel, en de POSTROUTING keten in de `nat` en `mangle` tabellen. Deze parameter accepteert dezelfde opties als de binnenkomende netwerk interface parameter (`-i`).
- `-p <protocol>` — Stelt het IP protocol in waarvoor de regel geldt. Dit kan `icmp`, `tcp`, `udp`, of `all` zijn, of het kan een numerieke waarde zijn die een van deze of een ander protocol representeert. Je kunt ook elk protocol gebruiken dat opgegeven is in het **`/etc/protocols`** bestand.

  Het "`all`" protocol betekent dat de regel geldt voor elk ondersteund protocol. Als in deze regel geen protocol wordt opgegeven, dan wordt "`all`" de standaard.
- `-s` — Stelt de bron in voor een bepaald pakket met gebruik van dezelfde syntax als de bestemmings (`-d`) parameter.

2.9.2.4. IPTables overeenkomst opties

Verscheidene netwerk protocollen bieden gespecialiseerde overkomst opties welke ingesteld kunnen worden om overeen te komen met een bepaald pakket gebruikmakend van dat protocol. Het protocol moet echter eerst opgegeven worden in het **`iptables`** commando. Bijvoorbeeld, `-p <protocol-naam>` zet opties aan voor het opgegeven protocol. Merk op dat je ook het protocol ID kunt opgeven in plaats van de protocol naam. Refereer naar de volgende voorbeelden, die ieder hetzelfde effect hebben:

```
iptables -A INPUT -p icmp --icmp-type any -j ACCEPT
```

```
iptables -A INPUT -p 5813 --icmp-type any -j ACCEPT
```

Service definities zijn te vinden in het **`/etc/services`** bestand. Voor de leesbaarheid wordt het echter aanbevolen dat je service namen gebruikt in plaats van poortnummers.

Waarschuwing

Beveilig het **`/etc/services`** bestand om ongeoorloofde bewerking te voorkomen. Als dit bestand bewerkbaar is, kunnen krakers het gebruiken om poorten aan te zetten op je computer die je gesloten had. Om dit bestand te beveiligen, type je het volgende commando's in als root:

```
[root@myServer ~]# chown root.root /etc/services
[root@myServer ~]# chmod 0644 /etc/services
```

```
[root@myServer ~]# chattr +i /etc/services
```

Dit belet dat het bestand een andere naam krijgt, verwijderd wordt of dat er verwijzingen naar zijn.

2.9.2.4.1. TCP protocol

De volgende overeenkomst opties zijn beschikbaar voor het TCP protocol (`-p tcp`):

- `--dport` — Stelt de bestemings poort in voor het pakket.

  Om deze optie in te stellen, gebruik je een netwerk service naam (zoals www of smtp); een poortnummer; of een reeks van poortnummers.

  Om een reeks van poortnummers op te geven, scheidt je de nummer met een dubbelepunt (`:`). Bijvoorbeeld, `-p tcp --dport 3000:3200`. De grootste geaccepteerde geldige reeks is `0:65535`.

  Gebruik een uitroepteken (`!`) na de `--dport` optie om overeen te komen met alle pakketten die *geen gebruik* maken van die netwerk service of poort.

  Om de namen en bijnamen van netwerk services en de poortnummers die ze gebruiken te bekijken, refereer je naar het **`/etc/services`** bestand.

  De `--destination-port` overeenkomst optie is synoniem met `--dport`.

- `--sport` — Stelt de bron poort van het pakket in met dezelfde opties als `--dport`. De `--source-port` overeenkomst optie is synoniem met `--sport`.

- `--syn` — Is van toepassing op alle TCP pakketten ontworpen om communicatie op te zetten, gewoonlijk *SYN pakketten* genaamd. Alle pakketten die een data lading bevatten worden niet beïnvloed.

  Gebruik een uitroepteken (`!`) na de `--syn` optie om overeen te komen met alle niet-SYN pakketten.

- `--tcp-flags <geteste vlag lijst> <ingestelde vlag lijst>` — Staat TCP pakketten die specifieke bits (vlaggen) gezet hebben, om overeen te komen met een regel.

  De `--tcp-flags` overeenkomst optie accepteert twee parameters. De eerste parameter is het masker, een door komma's gescheiden lijst van vlaggen in het pakket die bekeken worden. De tweede parameter is een door komma's gescheiden lijst van vlaggen de gezet moeten zijn om met de regel overeen te komen.

  De mogelijke vlaggen zijn:

  - `ACK`
  - `FIN`
  - `PSH`
  - `RST`
  - `SYN`

- `URG`
- `ALL`
- `NONE`

Bijvoorbeeld, een **`iptables`** regel die de volgende specificatie bevat komt alleen overeen met TCP pakketten die de SYN vlag gezet hebben en de ACK en FIN vlaggen niet gezet:

`--tcp-flags ACK,FIN,SYN SYN`

Gebruik het uitroepteken (`!`) na de `--tcp-flags` om het effect van de overeenkomst optie om te keren.

- `--tcp-option` — Probeert overeen te komen met TCP specifieke opties die gezet kunnen zijn in een bepaald pakket. Deze overeenkomst optie kan ook omgedraaid worden met het uitroepteken (`!`).

2.9.2.4.2. UDP protocol

De volgende overeenkomst opties zijn beschikbaar voor het UDP protocol (`-p udp`):

- `--dport` — Specificeert de bestemmingspoort voor het UDP pakket met gebruik van de service naam, poortnummer, of reeks van poortnummers. De `--destination-port` overeenkomst optie is synoniem met `--dport`.
- `--sport` — Specificeert de bronpoort van het UDP pakket met gebruik van de service naam, poortnummer, of reeks van poortnummers. De `--source-port` overeenkomst optie is synoniem met `--sport`.

Om voor de `--dport` en `--sport` opties een reeks van poortnummers op te geven, scheidt je de twee nummers met een dubbelepunt (:). Bijvoorbeeld, `-p tcp --dport 3000:3200`. De grootste geaccepteerde geldige reeks is `0:65535`.

2.9.2.4.3. ICMP protocol

De volgende overeenkomst opties zijn beschikbaar voor het Internet Control Message Protocol (ICMP) (`-p icmp`):

- `--icmp-type` — Stelt de naam of het nummer in van het IMCP type die overeen moet komen met de regel. Een lijst van geldige IMCP namen kan verkregen worden door het **`iptables -p icmp -h`** commando in te typen.

2.9.2.4.4. Extra overeenkomst optie modules

Extra overeenkomst opties zijn beschikbaar met modules die geladen worden door het **`iptables`** commando.

Om een overeenkomst optie module te gebruiken, laad je de module bij naam met gebruik van `-m <module-naam>`, waarin `<module-naam>` de naam van de module is.

Standaard zijn vele modules beschikbaar. Je kunt ook modules maken om extra functionaliteit te bieden.

Het volgende is een gedeeltelijke lijst van de meest gebruikte modules:

- `limit` module — Brengt limieten aan voor hoeveel pakketten kunnen overeenkomen met een bepaalde regel.

  Als dit gebruikt wordt tesamen met het **LOG** doel, kan de `limit` module een vloed van overeenkomende pakketten beletten om de systeemlog te vullen met herhaalde boodschappen of het opmaken van systeem hulpbronnen.

  Refereer naar *Paragraaf 2.9.2.5, “Doel opties”* voor meer informatie over het **LOG** doel.

  De `limit` module maakt de volgende opties beschikbaar:

  - `--limit` — Stelt het maximum aantal overeenkomsten in voor een bepaalde tijdsperiode, opgegeven als een *<waarde>/<periode>* paar. Bijvoorbeeld, het gebruiken van `--limit 5/hour` staat vijf regel overeenkomsten per uur toe.

    Periodes kunnen opgegeven worden in seconden, minuten, uren, of dagen

    Als dit paar niet wordt opgegeven, wordt de standaard waarde van `3/hour` aangenomen.

  - `--limit-burst` — Stelt een limiet aan het aantal pakketten die ten alle tijde kunnen overeenkomen met een pakket.

    Deze optie wordt opgegeven als een geheel getal en moet gebruikt worden tezamen met de `--limit` optie.

    Als geen waarde wordt opgegeven, wordt de standaard waarde vijf (5) aangenomen.

- `state` module — Zet toestand overeenkomst aan.

  De `state` module maakt de volgende opties beschikbaar:

  - `--state` — Kijkt of een pakket overeenkomt met de volgende verbindingstoestanden:

    - `ESTABLISHED` — Het pakket dat overeenkomt is gerelateerd aan andere pakketten in een bestaande verbinding. Je moet deze toestand accepteren als je een verbinding tussen een cliënt en een server wilt onderhouden.

    - `INVALID` — Het pakket dat overeenkomt past niet bij een bekende verbinding.

    - `NEW` — Het pakket dat overeenkomt is of een pakket dat een nieuwe verbinding aanmaakt, of onderdeel van een tweezijdige verbinding die eerder niet gezien is. Je moet deze toestand accepteren als je nieuwe verbindingen naar een service wilt toestaan.

    - `RELATED` — Het pakket dat overeenkomt start een nieuwe verbinding die op een bepaalde manier gerelateerd is aan een bestaande verbinding. Een voorbeeld hiervan is FTP, die een verbinding gebruikt voor controle verkeer (poort 21), en een aparte verbinding voor data overdracht (poort 20).

    Deze verbindings toestanden kunnen in combinatie met elkaar gebruikt worden door ze te scheiden met komma's, zoals `-m state --state INVALID,NEW`.

- `mac` module — Zet hardware MAC adres overeenkomst aan.

  De `mac` module maakt de volgende opties beschikbaar:

- `--mac-source` — Maakt een overeenkomst met een MAC adres van het netwerk interface dat het pakket verzendt. Om een MAC adres uit te zonderen van een regel, plaats je een uitroepteken (`!`) na de `--mac-source` overeenkomst optie.

Refereer naar de **`iptables`** manual pagina voor meer overeenkomst opties die beschikbaar zijn met behulp van modules.

2.9.2.5. Doel opties

Als een pakket overeenkomt met een bepaalde regel, kan de regel het pakket naar een aantal verschillende doelen sturen welke de juiste actie bepalen. Elke keten heeft een standaard doel, welke gebruikt wordt als geen enkele regel in die keten overeenkomt met een pakket of als geen van de regels die overeenkomen met het pakket een doel opgeven.

De volgende zijn de standaard doelen:

- *<gebruikers-gedefinieerde-keten>* — Een gebruikers gedefinieerde keten binnen de tabel. De namen van gebruikers gedefinieerde ketens moeten uniek zijn. Dit doel geeft het pakket door aan de opgegeven keten.
- `ACCEPT` — Laat het pakket doorgaan naar zijn bestemming of naar een andere keten.
- `DROP` — Laat het pakket vallen zonder bericht aan de aanvrager. Het systeem dat het pakket stuurt krijgt geen bericht van het mislukken.
- `QUEUE` — Het pakket wordt doorgegeven voor afhandeling door een toepassing in de gebruikersruimte.
- `RETURN` — Stopt met het controleren van het pakket voor de regels in de huidige keten. Als een pakket met een `RETURN` doel overeenkomt met een regel in een keten die aangeroepen is van een andere keten, wordt het pakket teruggestuurd naar de eerste keten om verder te gaan met de regel controle op de plek waar het gebleven was. Als de `RETURN` regel wordt gebruikt voor een ingebouwde keten en het pakket kan kan niet verder gaan naar zijn vorige keten, dan wordt het standaard doel voor de huidige keten gebruikt.

Bovendien zijn uitbreidingen beschikbaar die toestaan dat andere doelen worden opgegeven. Deze uitbreidingen worden doel modules of overeenkomst optie modules genoemd en de meeste zijn alleen van toepassing voor specifieke tabellen en situaties. Refereer naar *Paragraaf 2.9.2.4.4, “Extra overeenkomst optie modules”* voor meer informatie over overeenkomst optie modules.

Er bestaan vele uitgebreide doel modules, waarvan de meeste alleen van toepassing zijn voor specifieke tabellen of situaties. Sommige van de meer populaire modules die standaard beschikbaar zijn in Fedora zijn:

- `LOG` — Logt alle pakketten die overeenkomen met deze regel. Omdat pakketten gelogd worden door de kernel, bepaalt het **`/etc/syslog.conf`** bestand waar deze log boodschappen geschreven worden. Standaard komen ze in het **`/var/log/messages`** bestand.

  Extra opties kunnen gebruikt worden na het `LOG` doel om op te geven hoe de logging gebeurt:

  - `--log-level` — Zet het prioriteitsniveau van een log gebeurtenis. Refereer naar de **`syslog.conf`** manual pagina voor een lijst van prioriteitsniveau's.
  - `--log-ip-options` — Legt elke optie die ingesteld is in de koptekst van een IP pakket.

- `--log-prefix` — Plaatst een karakter reeks van maximaal 29 karakters voordat de log regel wordt geschreven. Dit is nuttig voor het schrijven van syslog filters tezamen met pakket logging.

  Opmerking
  Door een probleem met deze optie, moet je een spatie laten volgen na de *`log-prefix`* waarde.

- `--log-tcp-options` — Logt elke optie die gezet is in de koptekst van en TCP pakket.
- `--log-tcp-sequence` — Schrijft het TCP volgorde nummer voor het pakket in de log.

- `REJECT` — Stuurt een fout pakket terug naar het systeem op afstand en laat het pakket vallen.

  Het `REJECT` doel accepteert `--reject-with <type>` (waarin *<type>* het weigeringstype is) wat meer detail toestaat in de informatie die teruggestuurd wordt met het fout pakket. De boodschap `port-unreachable` is het standaard fout type dat gegeven wordt als geen andere optie wordt gebruikt. Refereer naar de **`iptables`** manual pagina voor een volledige lijst van *<type>* opties.

Andere doel uitbreidingen, waaronder verscheidene die nuttig zijn voor IP vermomming met gebruik van de `nat` tabel, of met pakket verandering met gebruik van de `mangle` tabel, kunnen gevonden worden in de **`iptables`** manual pagina.

2.9.2.6. Lijst opties

Het standaard lijst commando, **`iptables -L [<keten-naam>]`**, biedt een basis overzicht van de huidige ketens van de standaard filter tabel. Extra opties bieden meer informatie:

- `-v` — Laat uitgebreide output zien, zoals het aantal pakketten en bytes die elke keten verwerkt heeft, het aantal pakketten en bytes die overeenkwamen met elke regel, en welke interface van toepassing is voor een bepaalde regel.
- `-x` — Expandeert nummers naar hun exacte waarde. Op een druk systeem, kunnen de nummers van pakketten en bytes verwerkt door een bepaalde keten of regel afgekort worden naar `Kilobytes`, `Megabytes` of `Gigabytes`. Deze optie forceert dat het volledige nummer getoond wordt.
- `-n` — Laat IP adressen en poortnummers in numeriek formaat zien, in plaats van de standaard hostnaam en netwerk service formaat.
- `--line-numbers` — Laat de regels in elke keten zien met hun numerieke volgorde in de keten. Deze optie is nuttig als je probeert een specifieke regel in een keten te verwijderen of om te bepalen waar een regel ingevoegd moet worden in een keten.
- `-t <tabel-naam>` — Specificeert een tabel naam. Als dit weggelaten wordt is de standaard de filter tabel.

2.9.3. Het opslaan van IPTables regels

Regels die gemaakt worden met het **`iptables`** commando worden opgeslagen in het geheugen. Als het systeem opnieuw opgestart wordt voordat de **`iptables`** regels opgeslagen worden, gaan

alle regels verloren. Om netfilter regels bestand te laten zijn tegen een systeem opstart, moeten ze opgeslagen worden. Om netfilter regels op te slaan, type je het volgende commando als root:

```
/sbin/service iptables save
```

Dit voert het **iptables** init script uit, welke het **/sbin/iptables-save** programma uitvoert om de huidige **iptables** configuratie naar **/etc/sysconfig/iptables** te schrijven. Het bestaande **/etc/sysconfig/iptables** bestand wordt bewaard als **/etc/sysconfig/iptables.save**.

Als het systeem de volgende keer opstart, laadt het **iptables** init script de regels opgeslagen in **/etc/sysconfig/iptables** door het **/sbin/iptables-restore** commando te gebruiken.

Terwijl het altijd een goed idee is om een nieuwe **iptables** regel te testen voordat je het naar het **/etc/sysconfig/iptables** bestand schrijft, is het mogelijk om de **iptables** regels te kopiëren in dat bestand vanaf de versie van dit bestand op een andere machine. Dit biedt een snelle manier om stelsels van **iptables** regels te verspreiden over meerdere machines.

Je kunt ook de iptables regel opslaan in een aparte bestand voor verspreiding, backup, of andere doeleinden. Om je iptables regels op te slaan, type je het volgende commando in als root:

```
[root@myServer ~]# iptables-save > <bestandsnaam>waarin <bestandsnaam> een
door jouw gedefinieerde naam voor je regelstelsel.
```

Belangrijk

Als het **/etc/sysconfig/iptables** bestand naar andere machines verspreid wordt, type je **/sbin/service iptables restart** om deze nieuwe regels effectief te maken.

Opmerking

Merk het verschil op tussen het **iptables** *commando* (**/sbin/iptables**), welke wordt gebruikt voor het manipuleren van de tabellen en regels die de **iptables** functionaliteit bevatten, en de **iptables** *service* (**/sbin/iptables service**), welke wordt gebruikt om de **iptables** service zelf aan en uit te zetten.

2.9.4. IPTables controle scripts

Er zijn twee basis methodes voor het controleren van **iptables** in Fedora:

- **Firewall Configuration Tool** (**system-config-securitylevel**) — Een grafische interface voor het maken, activeren, en opslaan van basis firewall regels. Refereer naar *Paragraaf 2.8.2, "Basis firewall instelling"* voor meer informatie.
- **/sbin/service iptables <optie>** — Wordt gebruikt om de verschillende functies van **iptables** te manipuleren met behulp van zijn initscripts. De volgende opties zijn beschikbaar:
  - **start** — Als een firewall is ingesteld (dat wil zeggen, **/etc/sysconfig/iptables** bestaat), worden alle draaiende **iptables** helemaal gestopt en daarna gestart met het **/sbin/iptables-restore** commando. Deze optie werkt alleen als de **ipchains** kernel module niet

geladen is. Om te controleren of deze module geladen is, type je het volgende commando als root:

```
[root@MyServer ~]# lsmod | grep ipchains
```

Als dit commando geen output teruggeeft, betekent het dat de module niet geladen is. Indien noodzakelijk, gebruik je het **/sbin/rmmod** commando om de module te verwijderen.

- **stop** — Als de firewall draait, worden de firewall regels in het geheugen verwijderd en alle iptables modules en hulpprogramma's worden verwijderd.

  Als de **IPTABLES_SAVE_ON_STOP** instructie in het **/etc/sysconfig/iptables-config** configuratie bestand is veranderd van zijn standaard waarde naar **yes**, worden de huidige regels opgeslagen in **/etc/sysconfig/iptables** en alle bestaande regels worden verhuisd naar het bestand **/etc/sysconfig/iptables.save**.

  Refereer naar *Paragraaf 2.9.4.1, “IPTables controle scripts configuratie bestand”* voor meer informatie over het **iptables-config** bestand.

- **restart** — Als een firewall draait, worden de firewall regels in het geheugen verwijderd, en de firewall wordt opnieuw gestart als het ingesteld is in **/etc/sysconfig/iptables**. Deze optie werkt alleen als de **ipchains** kernel module niet geladen is.

  Als de **IPTABLES_SAVE_ON_RESTART** instructie in het **/etc/sysconfig/iptables-config** configuratie bestand is veranderd van zijn standaard waarde naar **yes**, worden de huidige regels opgeslagen in **/etc/sysconfig/iptables** en alle bestaande regels verhuisd naar het bestand **/etc/sysconfig/iptables.save**.

  Refereer naar *Paragraaf 2.9.4.1, “IPTables controle scripts configuratie bestand”* voor meer informatie over het **iptables-config** bestand.

- **status** — Laat de status van de firewall zien en geeft een lijst van alle actieve regels.

  De standaard instelling voor deze optie laat IP adressen zien in elke regel. Om de domein en hostnaam informatie te tonen, bewerk je het **/etc/sysconfig/iptables-config** bestand en je verandert de waarde van **IPTABLES_STATUS_NUMERIC** naar **no**. Refereer naar *Paragraaf 2.9.4.1, “IPTables controle scripts configuratie bestand”* voor meer informatie over het **iptables-config** bestand.

- **panic** — Verwijdert alle firewall regels. De tactiek van alle ingestelde tabellen wordt op **DROP** gezet.

  Deze optie kan nuttig zijn als het bekend is dat een server in gevaar is gebracht. In plaats van de machine fysiek van het netwerk af te halen, of het systeem uit te zetten, kun je deze optie gebruiken om alle verdere netwerkverkeer te stoppen, maar de machine in een toestand te laten klaar voor analyse of andere onderzoeken.

- **save** — Slaat de firewall regels op in **/etc/sysconfig/iptables** met behulp van **iptables-save**. Refereer naar *Paragraaf 2.9.3, “Het opslaan van IPTables regels”* voor meer informatie.

Opmerking

Om dezelfde initscripts commando's te gebruiken om netfilter te controleren voor IPv6, vervang je **iptables** met **ip6tables** in de **/sbin/service** commando's getoond in deze paragraaf. Voor meer informatie over IPv6 en netfilter, refereer je naar *Paragraaf 2.9.5, "IPTables en IPv6"*.

2.9.4.1. IPTables controle scripts configuratie bestand

Het gedrag van het **iptables** initscripts wordt gecontroleerd door het **/etc/sysconfig/iptables-config** configuratie bestand. Het volgende is een lijst van de instructies in dit bestand:

- **IPTABLES_MODULES** — Specificeert een door spaties gescheiden lijst van extra **iptables** modules die geladen moeten worden als de firewall geactiveerd wordt. Ze kunnen ook verbindingsvolger en NAT hulpprogramma's bevatten.
- **IPTABLES_MODULES_UNLOAD** — Verwijdert modules bij herstarten en stoppen. Deze instructie accepteert de volgende waardes:
  - **yes** — De standaard waarde. Deze optie moet ingesteld zijn om een correcte toestand te bereiken voor het opnieuw starten en stoppen van een firewall.
  - **no** — Deze optie moet alleen gebruikt worden als er problemen zijn met het verwijderen van de netfilter modules.
- **IPTABLES_SAVE_ON_STOP** — Slaat de huidige firewall regels op in **/etc/sysconfig/iptables** als de firewall gestopt wordt. Deze instructie accepteert de volgende waardes:
  - **yes** — Slaat de bestaande regels op in **/etc/sysconfig/iptables** als de firewall gestopt wordt, en verhult de vorige versie naar het **/etc/sysconfig/iptables.save** bestand.
  - **no** — De standaard waarde. Slaat de bestaande regels niet op als de firewall gestopt wordt.
- **IPTABLES_SAVE_ON_RESTART** — Slaat de huidige firewall regels op als de firewall opnieuw gestart wordt. Deze instructie accepteert de volgende waardes:
  - **yes** — Slaat de bestaande regels op in **/etc/sysconfig/iptables** als de firewall opnieuw gestart wordt, en verhuist de vorige versie naar het **/etc/sysconfig/iptables.save** bestand.
  - **no** — De standaard waarde. Slaat de bestaande regels niet op als de firewall opnieuw opgestart wordt.
- **IPTABLES_SAVE_COUNTER** — Bewaart en herlaad alle pakket en byte tellers in alle ketens en regels. Deze instructie accepteert de volgende waardes:
  - **yes** — Bewaart de teller waardes.
  - **no** — De standaard waarde. Bewaard de teller waardes niet.
- **IPTABLES_STATUS_NUMERIC** — Laat de IP adressen in numerieke vorm zien in plaats van domein of hostnamen. Deze instructie accepteert de volgende waardes:
  - **yes** — De standaard waarde. Geeft alleen IP adressen terug binnen een status output.

- **no** — Geeft domein of hostnamen terug in een status output

2.9.5. IPTables en IPv6

Als het **iptables-ipv6** pakket geïnstalleerd is, kan netfilter in Fedora het volgende generatie IPv6 Internet protocol filteren. Het commando dat gebruikt wordt om het IPv6 netfilter te manipuleren is **ip6tables**.

De meeste instructies zijn identiek aan degene die gebruikt worden voor **iptables**, behalve dat de **nat** nog niet ondersteund wordt. Dit betekent dat het nog niet mogelijk is om IPv6 netwerk adres vertalings taken uit te voeren, zoals vermomming en poort forwarding.

Regels voor **ip6tables** worden bewaard in het **/etc/sysconfig/ip6tables** bestand. Vorige regels opgeslagen door de **ip6tables** initscripts worden opgeslagen in het **/etc/sysconfig/ip6tables.save** bestand.

Configuratie opties voor het **ip6tables** init script worden opgeslagen in **/etc/sysconfig/ip6tables-config**, en de namen voor elke instructie verschillen een klein beetje vergeleken met hun **iptables** tegenhangers.

Bijvoorbeeld, de **iptables-config** instructie **IPTABLES_MODULES**: het equivalent in het **ip6tables-config** bestand is **IP6TABLES_MODULES**.

2.9.6. Extra hulpbronnen

Refereer naar de volgende bronnen voor meer informatie over pakket filtering met **iptables**.

- *Paragraaf 2.8, “Firewalls”* — Bevat een hoofdstuk over de rol van firewalls binnen een algehele beveiligingsstrategie en strategieën voor het maken van firewalls.

2.9.6.1. Geïnstalleerde IPTables documentatie

- **man iptables** — Bevat een beschrijving van **iptables** en een uitgebreide lijst van doelen, opties, en overeenkomst uitbreidingen.

2.9.6.2. Nuttige IPTables websites

- *http://www.netfilter.org/* — De thuis pagina van het netfilter/iptables project. Bevat verschillende informatie over **iptables**, inclusief een FAQ voor specifieke problemen en verscheidene nuttige gidsen door Rusty Russell, de Linux IP firewall onderhouder. De HOWTO documenten op de site behandelen onderwerpen zoals basis netwerk concepten, kernel pakket filtering, en NAT configuraties.

- *http://www.linuxnewbie.org/nhf/Security/IPtables_Basics.html* — Een inleiding over de manier waarop pakketten door de Linux kernel bewegen, plus een inleiding voor het maken van basis **iptables** commando's.

Versleuteling

Er zijn twee hoofd types data die beschermd moeten worden: data in beweging en data in rust. Deze verschillende types data worden beschermd op een vergelijkbare manier met gebruik van een vergelijkbare technologie maar de implementatie kan geheel verschillend zijn. Geen enkele beschermings implementatie kan alle mogelijke manier van in gevaar brengen voorkomen omdat dezelfde informatie in rust en in beweging kan zijn op verschillende tijdsmomenten.

3.1. Data in rust

Data is in rust op het moment dat het opgeslagen is op een harde schijf, tape, CD, DVD, of andere media. De grootste bedreiging voor deze informatie is dat het fysiek gestolen wordt. Laptops op vliegvelden, CD's in de post, backup tapes die op de verkeerde plaats terecht komen, dit zijn allemaal voorbeelden van gebeurtenissen waar data in gevaar kan worden gebracht door diefstal. Als de data versleuteld was op de media dan zou je er niet zo druk over hoeven te maken dat de data in gevaar wordt gebracht.

3.2. Volledige schijf versleuteling

Volledige schijf of partitie versleuteling is een van de beste manieren om je data te beschermen. Niet alleen is elk bestand beschermd, maar ook de tijdelijke opslag die onderdelen van die bestanden kan bevatten is beschermd. Volledige schijf versleuteling zal al je bestanden beschermen zodat je er niet druk over hoeft maken wat je wilt beschermen en mogelijk een bestand vergeet.

Fedora 9, en later, ondersteunt ingebouwde LUKS versleuteling. LUKS zal je gehele harde schijf partities versleutelen zodat je date beschermd is als de computer uit staat. Dit zal je computer ook beschermen tegen aanvallers die proberen in enkele-gebruikers mode in te loggen op je computer of op een andere manier toegang krijgen.

Volledige schijf versleuteling zoals LUKS beschermen je data alleen als de computer uit staat. Zodra de computer aan is en LUKS heeft de schijf ontsleuteld, zijn de bestanden op die schijf beschikbaar voor iedereen die normaal toegang tot deze heeft. Om je bestanden te beschermen als je computer aan staat, gebruik je volledige schijf versleuteling in combinatie met een andere oplossing zoals versleuteling op bestands basis. Denk er ook aan je computer af te sluiten als je er vandaan bent. Een met een wachtwoord beschermde screensaver die activeert na een paar minuten van inactiviteit is een goede manier om indringers buiten te houden.

3.3. Bestand gebaseerde versleuteling

GnuPG (GPG) is een open bron versie van PGP dat je toestaat om een bestand of een email bericht te tekenen en/of te versleutelen. Dit is nuttig om de integriteit van het bericht of bestand te behouden en ook om de vertrouwelijkheid van de informatie in het bestand of email bericht te beschermen. In het geval van email, biedt GPG dubbele bescherming. Het biedt niet alleen bescherming voor data in rust, maar ook data in beweging bescherming zodra het bericht over het netwerk verstuurd wordt.

Bestand gebaseerde versleuteling is bedoeld om een bestand te beschermen nadat het je computer heeft verlaten, zoals wanneer je een CD verstuurt met de post. Sommige bestand gebaseerde versleutelings oplossingen laten restanten van het versleutelde bestand achter die een aanvaller die fysieke toegang tot je computer heeft onder sommige omstandigheden kan ontsleutelen. Om de inhoud van zulke bestanden te beschermen voor aanvallers die toegang tot je computer kunnen hebben, gebruik je bestand gebaseerde versleuteling gecombineerd met een andere oplossing zoals volledige schijf versleuteling.

3.4. Data in beweging

Data in beweging is data die verstuurd wordt over een netwerk. De grootste bedreiging voor data in beweging zijn onderschepping en verandering. Je gebruikersnaam en wachtwoord moeten nooit zonder bescherming over het netwerk verstuurd worden omdat het onderschept kan worden en gebruikt door iemand anders om zich als jou voor te doen of om toegang te krijgen tot gevoelige informatie. Andere privé informatie zoals bankrekening informatie moet ook beschermd worden als het over een netwerk verstuurd wordt. Als de netwerk sessie versleuteld was dan zou je er niet zo druk over hoeven te maken dat data in gevaar wordt gebracht als het verstuurd wordt.

Data in beweging is in het bijzonder kwetsbaar voor aanvallers omdat de aanvaller zich niet in de buurt van je computer, waarop de data bewaard wordt, hoeft te bevinden, ze hoeven zich slechts ergens langs het pad te bevinden. Versleutelde tunnels kunnen data beschermen langs het communicatie pad.

3.5. Virtuele privé netwerken

Virtuele privé netwerken (VPN) bieden versleutelde tunnels tussen computers of netwerken van computers over alle poorten. Met een aanwezige VPN wordt alle netwerkverkeer van de cliënt doorgestuurd naar de server via de versleutelde tunnel. Dit betekent dat de cliënt logischerwijs op hetzelfde netwerk is als de server waarmee het verbonden in via de VPN. VPN's komen veel voor en zijn eenvoudig te gebruiken en in te stellen.

3.6. Beveiligde shell

Beveiligde shell (SSH) is een krachtig netwerk protocol dat wordt gebruikt om te communiceren met een ander systeem over een beveiligd kanaal. De verzendingen over SSH zijn versleuteld en beschermd tegen onderschepping. Cryptografisch aanloggen kan ook gebruikt worden om een betere authenticatie methode te bieden dan de traditionele gebruikersnamen en wachtwoorden.

SSH is erg eenvoudig te activeren. Simpel door het starten van de sshd service, zal het systeem beginnen met het accepteren van verbindingen en zal het toegang tot het systeem toestaan als een juiste gebruikersnaam en wachtwoord wordt aangeboden tijdens het verbindings proces. De standaard TCP poort voor de SSH service is 22, dit kan echter veranderd worden door het veranderen van het configuratie bestand */etc/ssh/sshd_config* en het opnieuw opstarten van de service. Dit bestand bevat ook andere configuratie opties voor SSH.

Beveiligde shell (SSH) biedt ook versleutelde tunnels tussen computers maar gebruiken hierbij een enkele poort. *Poort forwarding kan gedaan worden met een SSH tunnel*[1] en verkeer zal versleuteld worden als het passeert door die tunnel, maar het gebruik van poort forwarding is niet zo instabiel als een VPN.

3.7. LUKS schijf versleuteling

Linux Unified Key Setup-on-disk-format (of LUKS) staat je toe om partities op je Linux computer te versleutelen. Dit is in het bijzonder belangrijk voor draagbare computers en verwijderbare media. LUKS staat meerdere gebruikerssleutels toe om een meestersleutel te ontsleutelen die daarna gebruikt wordt voor de bulk ontsleuteling van de partitie.

[1] http://www.redhatmagazine.com/2007/11/27/advanced-ssh-configuration-and-tunneling-we-dont-need-no-stinking-vpn-software

3.7.1. De LUKS implementatie in Fedora

Fedora 9, en later, gebruikt LUKS om bestandssysteem versleuteling uit te voeren. Standaard is de optie om het bestandssysteem te versleutelen tijdens de installatie niet aangevinkt. Als je deze optie selecteert om je harde schijf te versleutelen, zal je gevraagd worden naar een wachtzin waarnaar iedere keer als je de computer opstart naar gevraagd zal worden. De wachtzin "opent" de bulk sleutel die wordt gebruikt om je partitie te ontsleutelen. Als je ervoor kiest om de standaard partitietabel te veranderen kun je kiezen welke partities je wilt versleutelen. Dit wordt ingesteld in de instelling van partitietabel.

De standaard inplementatie van LUKS in Fedora gebruikt AES 128 met een SHA256 hash. De beschikbare codes zijn:

- AES - Advanced Encryption Standard - *FIPS PUB 197*[2]
- Twofish (A 128-bit Block Cipher)
- Serpent
- cast5 - *RFC 2144*[3]
- cast6 - *RFC 2612*[4]

3.7.2. Handmatig mappen versleutelen

Waarschuwing

Het opvolgen van deze procedure zal alle data verwijderen op de partitie die je gaat versleutelen. Je ZULT al je informatie verliezen! Wees er zeker van dat je een backup maakt van je data naar een externe opslag voordat je met deze procedure begint!

Als je een eerdere versie van Fedora gebruikt dan Fedora 9 en je wilt een partitie versleutelen, of je wilt een partitie versleutelen na de installatie van de huidige versie van Fedora, moet je de volgende instructies gebruiken. Het voorbeeld hieronder laat het versleutelen van je /home partitie zien maar elke partitie kan gebruikt worden.

De volgende procedure zal al je bestaande data wegpoetsen, wees er dus zeker van dat je een geteste backup hebt voordat je begint. Dit vereist ook dat je een aparte partitie hebt voor /home (in mijn geval is dat /dev/VG00/LV_home). De volgende stappen moeten gedaan worden als root. Als een van deze stappen mislukt betekent dat je niet verder moet gaan totdat de stap geslaagd is.

3.7.3. Stap-voor-stap instructies

1. ga naar runlevel 1: `telinit 1`
2. koppel je bestaande /home af: `umount /home`
3. als dat mislukt gebruik je `fuser` om het proces dat /home gebruikt te vinden en af te schieten: `fuser -mvk /home`
4. controleer dat /home niet langer aangekoppeld is: `cat /proc/mounts | grep home`
5. vul je partitie met willekeurige data: `dd if=/dev/urandom of=/dev/VG00/LV_home` Dit proces kan vele uren duren.

Belangrijk

Het proces is echter belangrijk om een goede bescherming te hebben tegen inbraak pogingen. Laat het gewoon een nacht draaien.

6. initialiseer je partitie: `cryptsetup --verbose --verify-passphrase luksFormat /dev/VG00/LV_home`
7. open het nieuw versleutelde apparaat: `cryptsetup luksOpen /dev/VG00/LV_home home`
8. controleer of het er is: `ls -l /dev/mapper | grep home`
9. maak een bestandssysteem: `mkfs.ext3 /dev/mapper/home`
10. koppel het aan: `mount /dev/mapper/home /home`
11. controleer of het zichtbaar is: `df -h | grep home`
12. voeg het volgende toe aan /etc/crypttab: `home /dev/VG00/LV_home none`
13. bewerk je /etc/fstab, verwijder de oude regel voor /home en voeg toe: `/dev/mapper/home / home ext3 defaults 1 2`
14. controleer je fstab regel: `mount /home`
15. herstel de standaard SELinux beveiligings context: `/sbin/restorecon -v -R /home`
16. start opnieuw op: `shutdown -r now`
17. de regel in /etc/crypttab laat je computer jouw vragen naar je `luks` wachtzin tijdens het opstarten
18. login als root en herstel je backup

3.7.4. Wat heb je zojuist bereikt

Gefeliciteerd, je hebt nu een versleutelde partitie waar al je data veilig kan rusten als de computer uit is.

3.7.5. Interessante verwijzingen

Voor extra informatie over LUKS of het versleutelen van harde schijven met Fedora bezoek je een van de volgende verwijzingen:

- *LUKS - Linux Unified Key Setup*[5]
- *HOWTO: Creating an encrypted Physical Volume (PV) using a second hard drive, pvmove, and a Fedora LiveCD*[6]

3.8. 7-Zip versleutelde archieven

7-Zip[7] is een, op meerdere platforms beschikbaar en van de volgende generatie, bestandscompressie gereedschap dat ook een sterke versleuteling (AES-256) kan gebruiken om de inhoud van het archief te beschermen. Dit is bijzonder nuttig als je data moet verplaatsen tussen meerdere computers die verschillende operating systemen draaien (b.v. Linux thuis, Windows op het werk) en je zoekt hiervoor naar een overdraagbare versleutelings oplossing.

3.8.1. 7-Zip installatie in Fedora

7-Zip is niet een standaard geïnstalleerd pakket in Fedora, maar het is beschikbaar in de software repository. Zodra het pakket geïnstalleerd is zal het samen met rest van de software op het systeem vernieuwd worden zonder dat daar speciale aandacht voor nodig is.

3.8.2. Stap-voor-stap installatie instructies

- Open een terminal: `Klik op ''Toepassingen'' -> ''Systeemgereedschap'' -> ''Terminalvenster''`
- Installeer 7-Zip met sudo toegang: `sudo yum install p7zip`
- Sluit de terminal: `exit`

3.8.3. Stap-voor-stap gebruiks instructies

Door het opvolgen van deze instructies kun je jouw "Documenten" map comprimeren en versleutelen. Je originele "Documenten" map blijft ongewijzigd. Deze techniek kan toegepast worden voor elke map of bestand waarnaar je toegang hebt op het bestandssysteem.

- Open een terminal: `Klik op ''Toepassingen'' -> ''Systeemgereedschap'' -> ''Terminalvenster''`
- Comprimeer en versleutel: (type een wachtwoord in als er om gevraagd wordt) `7za a -mhe=on -ms=on -p Documenten.7z Documenten/`

De "Documenten" map wordt nu gecomprimeerd en versleuteld. De volgende instructies zullen het versleutelde archief verplaatsen naar een nieuwe plek en het daar uitpakken.

- Maak een nieuwe map aan: `mkdir nieuweplek`
- Verplaats het versleutelde bestand: `mv Documenten.7z nieuweplek`
- Ga naar de nieuwe map: `cd nieuweplek`
- Pak het bestand uit: (type het wachtwoord in als er om gevraagd wordt) `7za x Documenten.7z`

Het archief is nu uitgepakt in de nieuwe locatie. De volgende instructies zullen alle vorige stappen opschonen en je computer weer in de vorige toestand terugbrengen.

- Ga een map omhoog: `cd ..`
- Verwijder het test archief en de test map: `rm -r nieuweplek`
- Sluit de terminal: `exit`

[7] http://www.7-zip.org/

3.8.4. Merk op

7-Zip wordt niet standaard verstuurd met Microsoft Windows of Mac OS X. Als je jouw 7-Zip bestanden op die platformen moet gebruiken, moet je de juiste versie van 7-Zip op die computers installeren. Zie de 7-Zip *download pagina*[8].

De File-roller toepassing van Gnome zal jouw .7z bestanden herkennen en proberen ze te openen, maar dit mislukt met een fout. Dit komt omdat File-roller op dit moment het uitpakken van versleutelde 7-Zip bestanden niet ondersteund. Een bug rapport ([http://bugzilla.gnome.org/show_bug.cgi?id=490732 Gnome Bug 490732]) is ingediend.

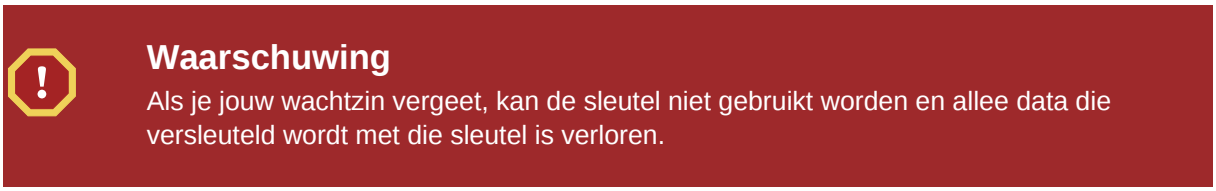
3.9. GNU Privacy Guard (GnuPG) gebruiken

GPG wordt gebruikt om jezelf te identificeren en voor authenticatie van je communicatie, zelfs voor mensen die je niet kent. GPG laat iedereen die een met GPG ondertekende email leest de echtheid hiervan verifiëren. Met andere woorden, GPG laat iemand er behoorlijk zeker van zijn dat de communicatie door jou ondertekent inderdaad van jou komt. GPG is nuttig omdat het helpt te voorkomen dat derden code veranderen of conversaties onderscheppen en de boodschap veranderen.

3.9.1. Het maken van GPG sleutels in GNOME

Installeer het Seahorse programma, dat het beheer van GPG sleutels eenvoudiger maakt. Van het hoofdmenu selecteer je `Systeem > Beheer > Software toevoegen/verwijderen` en wacht tot PackageKit opgestart is. Vul `Seahorse` in het zoektekstveld en selecteer Find. Selecteer het aanvinkhokje naast het "seahorse" pakket en selecteer "Toepassen" om de software toe te voegen. Je kunt `Seahorse` ook installeren op de commandoregel met het commando `su -c "yum install seahorse"`.

Om een sleutel te maken, selecteer je van het "Toepassingen > Hulpmiddelen" menu "Wachtwoorden en sleutels", wat de `Seahorse` toepassing opstart. In het "Bestand" menu selecteer je "New" en dan "PGP-sleutel" en klik op "Doorgaan". Vul je volledige naam, emailadres, en een optioneel commentaar wat beschrijft wie je bent (b.v.: Jan. Smit, jsmit@example.com, De Man). Klik op "Aanmaken". Een venster verschijnt die je vraagt voor een wachtzin voor de sleutel. Kies een sterke wachtzin die ook eenvoudig te onthouden is. Klik op "OK" en de sleutel wordt gemaakt.



Om je GPG sleutel ID te vinden, kijk je in de "Sleutel-id" kolom naast de nieuw aangemaakte sleutel. Als je wordt gevraagd naar je sleutel ID, moet je in de meeste gevallen "0x" voor het sleutel ID plaatsen, zoals in "0x6789ABCD". Je moet een backup van je privé sleutel maken en het op een veilige plek bewaren.

3.9.2. Het maken van GPG sleutels in KDE

Start het KGpg programma van het hoofdmenu door het selecteren van Toepassingen > Hulpmiddelen > KGPG. Als je KGpg nooit eerder gebruikt hebt, begeleidt het programma je door het

[8] http://www.7-zip.org/download.html

proces van het aanmaken van je eigen GPG sleutelpaar. Een dialoog venster verschijnt die je vraagt om een nieuwe sleutelpaar te maken. Vul je naam, emailadres, en extra commentaar in. Je kunt ook een verloopdatum voor je sleutel kiezen en de sterkte van de sleutel (het aantal bits) en algorithmes. Het volgende dialoog venster vraagt je naar je wachtzin. Daarna verschijnt je sleutel in het hoofd `KGpg` venster.

Waarschuwing

Als je jouw wachtzin vergeet, kan de sleutel niet gebruikt worden en allee data die versleuteld wordt met die sleutel is verloren.

Om je GPG sleutel ID te vinden, kijk je in de "Sleutel-id" kolom naast de nieuw aangemaakte sleutel. Als je wordt gevraagd naar je sleutel ID, moet je in de meeste gevallen "0x" voor het sleutel ID plaatsen, zoals in "0x6789ABCD". Je moet een backup van je privé sleutel maken en het op een veilige plek bewaren.

3.9.3. Het maken van GPG sleutels met de commandoregel

Gebruik het volgende shell commando: `gpg --gen-key`

Dit commando genereert een sleutelpaar dat bestaat uit een publieke en een privé sleutel. Anderen gebruiken jouw publieke sleutel voor authenticatie en/of ontsleutelen van je communicatie. Verspreid je publieke sleutel zo breed mogelijk, in het bijzonder naar mensen waarvan je weet dat ze autentieke communicatie van jou willen ontvangen, zoals een emaillijst. Het Fedora Documentation Project, bijvoorbeeld, vraagt deelnemers om een publieke GPG sleutel toe te voegen aan hun zelf-introductie.

Een serie prompts leidt je door het proces. Duw op de `Enter` toets om zo nodig een standaard waarde toe te kennen. De eerste prompt vraagt voor welke soort sleutel je de voorkeur hebt:

Selecteer het soort sluetel dat u wilt: (1) DSA en ElGamal (default) (2) DSA (alleen ondertekenen) (4) RSA (alleen ondertekenen) Uw keuze? In bijna alle gevallen is de default de juiste keuze. EEn DSA/ ElGamal sleutel laat je niet alleen communicatie ondertekenen, maar ook bestanden versleutelen.

Kies vervolgens de sleutellengte: DSA sleutelpaar krijgt 1024 bits. ELG-E sleutels moeten tussen 1024 en 4096 bits lang zijn. Welke sleutellengte wilt u? (2048). Opnieuw is de standaard voldoende voor bijna alle gebruikers, en biedt een "uitzonderlijk" sterk niveau van beveiliging.

Vervolgens kies je hoe lang de sleutel geldig moet zijn. Het is een goed idee om een verloopdatum te liezen in plaats van de standaard, die "nooit" is. Als, bijvoorbeeld, het email adres ongeldig wordt, zal een verloop datum anderen eraan herinneren om die publieke sleutel niet meer te gebruiken.

Geef aan hoe lang de sleutel geldig moet zijn. 0 = sleutel verloopt nooit d = sleutel verloopt na n dagen w = sleutel verloopt na n weken m = sleutel verloopt na n maanden y = sleutel verloopt na n jaar Is de sleutel geldig voor? (0)

Het invullen van de waarde `1y`, bijvoorbeeld, maakt de sleutel geldig voor een jaar. (Je kunt de verloop datum nog veranderen nadat de sleutel is aangemaakt als je van gedachte verandert.)

Voordat het gpgcode > programma vraagt om handteken informatie, verschijnt de volgende prompt:`Is dit correct (y/N)?`. Vul ycode > in om het proces te vervolgen.

Vul daarna je naam en emailadres is. Denk eraan dat dit proces over de authenticatie van jou gaat als een echt individu gaat. Vul daarom je echte naam in. Gebruik geen bijnamen of titel, omdat deze je identiteit vermommen of vertroebelen.

Vul je echte emailadres in voor je GPG sleutel. Als je een nep emailadres kiest, wordt het moeilijker voor anderen om je publieke sleutel te vinden. Dit maakt de authenticatie van je communicatie moeilijk. Als je deze GPG sleutel gebruikt voor [[DocsProject/SelfIntroduction| zelf-introductie]] op een emaillijst, bijvoorbeeld, vul je het emailadres in die je op die lijst gebruikt.

Gebruik het opmerkingen veld om bijnamen of andere informatie toe te voegen. (Sommigen gebruiken verschillende sleutels voor verschillende doeleinden en identificeren elke sleutel met een opmerking, zoals "Kantoor" of "Open bron projecten".)

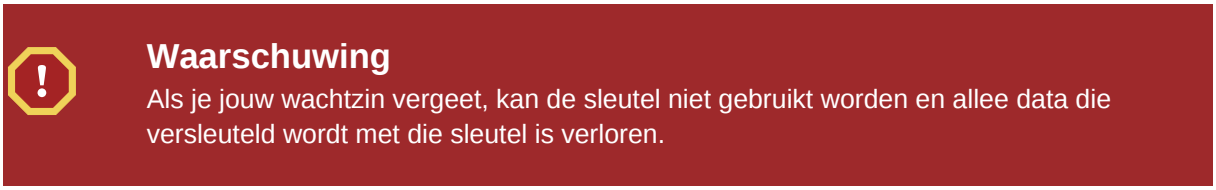
Vul de letter O in op de bevestigings prompt als alles correct is, of gebruik een van de andere opties om problemen te herstellen. Tenslotte vul je een wachtzin in voor je geheime sleutel. Het `gpg` programma vraagt je om de wachtzin twee keer in te vullen om er zeker van te zijn dat je geen typefout gemaakt hebt.

Tenslotte genereert `gpg` willekeurige data om je sleutel zo uniek mogelijk te maken. Beweeg je muis, type willekeurige karakters, of voer andere taken uit op het systeem tijdens deze stap om het proces te versnellen. Als deze stap klaar is, zijn je sleutels gereed en klaar voor gebruik:

```
pub  1024D/1B2AFA1C 2005-03-31 John Q. Doe (Fedora Docs Project)
 <jqdoe@example.com>
Vingerafdruk van de sleutel = 117C FE83 22EA B843 3E86  6486 4320 545E 1B2A
 FA1C
sub  1024g/CEA4B22E 2005-03-31 [expires: 2006-03-31]
```

De vingerafdruk van de sleutel is een verkorte "handtekening" voor je sleutel. Het laat anderen bevestigen dat ze jouw echte publieke sleutel hebben ontvangen zonder enig geknoei. Je hoeft deze vingerafdruk niet op te schrijven. Je kunt je vingerafdruk ten alle tijde bekijken met het volgende commando, waarbij je jouw emailadres invult: `gpg --fingerprint jqdoe@example.com`

Jouw "GPG sleutel-id" bestaat uit 8 hex getallen die de publieke sleutel identificeren. In het voorbeeld hierboven is de GPG sleutel-id 1B2AFA1C. Als je gevraagd wordt naar je sleutel-id moet je in de meeste gevallen "0x" vooraan de sleutel toevoegen, zoals in "0x1B2AFA1C".



3.9.4. Over publieke sleutel versleuteling

1. *Wikipedia - Public Key Cryptography*[9]
2. *HowStuffWorks - Encryption*[10]

Algemene principes van informatie beveiliging

De volgende algemene principes geven een overzicht van goede beveiligings praktijken:

- versleutel alle data die over het netwerk verstuurd wordt om de-man-in-het-midden aanvallen en afluisteren te helpen voorkomen. Het is belangrijk om authenticatie gegevens, zoals wachtwoorden, te versleutelen.
- minimaliseer de hoeveelheid geïnstalleerde software en draaiende services.
- gebruik beveiligings-verbeterde software en gereedschappen, bijvoorbeeld, Security-Enhanced Linux (SELinux) voor Mandatory Access Control (MAC), Netfilter iptables voor pakket filtering (firewall), en de GNU Privacy Guard (GnuPG) voor het versleutelen van bestanden.
- draai, indien mogelijk, elke netwerk service op een apart systeem om het risico te verkleinen dat een in gevaar gebrachte service wordt gebruikt om andere services in gevaar te brengen.
- onderhoud gebruikersaccounts: maak en forceer een sterke wachtwoorden tactiek; verwijder ongebruikte gebruikersaccounts.
- bekijk de systeem en toepassing log als routine. Standaard worden systeem logs die relevant zijn voor beveiliging naar **`/var/log/secure`** en **`/var/log/audit/audit.log`** geschreven. Merk op: het sturen van de logs naar een specifieke log server helpt om te voorkomen dat aanvallers lokale logs eenvoudig kunnen veranderen om ontdekking te voorkomen.
- log nooit in als de root gebruiker behalve als dat absoluut noodzakelijk is. Het wordt aanbevolen dat beheerders **`sudo`** gebruiken om commando's als root uit te voeren als dat nodig is. Gebruikers die **`sudo`** kunnen draaien worden opgegeven in **`/etc/sudoers`**. Gebruik het **`visudo`** programma om **`/etc/sudoers`** te bewerken.

4.1. Tips, gidsen, en gereedschappen

De *National Security Agency (NSA)*[1] uit de VS biedt versterkings gidsen en tips voor vele verschillende operating systemen om regerings organen, bedrijven, en individuelen te helpen om hun systemen te beveiligen tegen aanvallen. De volgende gidsen (in PDF formaat) bieden een leidraad voor Red Hat Enterprise Linux 5:

- *Hardening Tips for the Red Hat Enterprise Linux 5*[2]
- *Guide to the Secure Configuration of Red Hat Enterprise Linux 5*[3]

De *Defense Information Systems Agency (DISA)*[4] biedt documentatie, checklijsten, en testen om te helpen met het beveiligen van je systeem (*Information Assurance Support Environment*[5]). De *UNIX SECURITY TECHNICAL IMPLEMENTATION GUIDE*[6] (PDF) is een zeer specifieke gids voor UNIX beveiliging - een gevorderde kennis van UNIX en Linux is aanbevolen voordat je deze gids leest.

[1] http://www.nsa.gov/
[4] http://www.disa.mil/
[5] http://iase.disa.mil/index2.html
[6] http://iase.disa.mil/stigs/stig/unix-stig-v5r1.pdf

De DISA *UNIX Security Checklist Version 5, Release 1.16*[7] biedt een verzameling van documenten en checklijsten, variërend van de juiste eigenaar en mode voor systeembestanden tot code-aanpassings controle.

DISA heeft ook *UNIX SPR scripts*[8] beschikbaar gemaakt die beheerders specifieke instellingen op systemen laten controleren. Deze scripts bieden rapportlijsten in XML formaat van alle bekende kwetsbare instellingen.

[7] http://iase.disa.mil/stigs/checklist/unix_checklist_v5r1-16_20090215.ZIP
[8] http://iase.disa.mil/stigs/SRR/unix.html

Veilige installatie

Beveiliging begint met de eerste keer dat je een CD of DVD aanbrengt in je schijfstation om Fedora te installeren. Het veilig instellen van je systeem vanaf het begin maakt het gemakkelijker om later extra beveiligings instellingen te maken.

5.1. Schijfpartities

Het NSA beveelt het aanmaken van aparte partities voor /boot, /, /home, /tmp, en /var/tmp aan. De redenen hiervoor zijn verschillend en we zullen elke partitie behandelen.

/boot - Deze partitie is de eerste partitie die door het systeem gelezen wordt tijdens het opstarten. De boot loader en kernel images die gebruikt worden om je computer op te starten in Fedora worden op deze partitie bewaard. De partitie mag niet versleuteld zijn. Als deze partitie onderdeel is van / en die partitie wordt versleuteld of is op een andere manier niet beschikbaar dan zal het systeem niet in staat zijn om op te starten.

/home - Als gebruikers data (/home) wordt opgeslagen in / in plaats van een aparte partitie, kan de partitie opgevuld worden waardoor het operating systeem onstabiel wordt. Bovendien wordt het upgraden van je systeem naar de volgende versie van Fedora een stuk gemakkelijker als je jouw data in de /home partitie kan houden zodat het niet overschreven kan worden tijdens de installatie. Als de root (/) partitie corrupt wordt kan jouw data voor altijd verloren zijn. Door het gebruik van aparte partities is er een klein beetje meer bescherming tegen dataverlies. Je kunt deze partitie ook het doel laten zijn voor regelmatige backups.

/tmp and /var/tmp - Zowel de /tmp als de /var/tmp map wordt gebruikt om data op te slaan die niet voor een lange tijdsperiode beschikbaar hoeft te zijn. Als echter heel veel data een van deze mappen overspoelt kan het al je opslagruimte opmaken. Als dit gebeurt en deze mappen zijn binnen / dan kan je systeem onstabiel worden en crashen. Daarom is het verplaatsen van deze mappen naar een eigen partitie een goed idee.

5.2. LUKS partitie versleuteling gebruiken

Sinds Fedora 9 is de implementatie van *Linux Unified Key Setup-on-disk-format*[1](LUKS) versleuteling veel eenvoudiger geworden. Tijdens het installatie proces wordt een optie om de partities te versleutelen aangeboden aan de gebruiker. De gebruiker moet de benodigde wachtzin opgeven wat de sleutel zal zijn om de bulk sleutel te openen die gebruikt zal worden om de data van de partitie te beveiligen.

[1] http://fedoraproject.org/wiki/Security_Guide/9/LUKSDiskEncryption

Software onderhoud

Software onderhoud is erg belangrijk om de beveiliging van een systeem te handhaven. Het is van vitaal belang om software te patchen zodra deze beschikbaar komen om aanvallers te verhinderen om bekende problemen te gebruiken om je systeem binnen te dringen.

6.1. Installeer minimale software

Het is een goede praktijk om alleen die pakketten te installeren die je wilt gebruiken omdat ieder stukje software op je systeem een kwetsbaarheid kan bevatten. Als je installeert van de DVD media gebruik dan de mogelijkheid om precies die pakketten te selecteren die je tijdens de installatie wilt installeren. Als je ontdekt dat je nog een pakket nodig hebt, kun je het altijd later nog aan je systeem toevoegen.

6.2. Het plannen en configureren van beveiligingsvernieuwingen

Alle software bevat fouten. Vaak kunnen deze fouten resulteren in een kwetsbaarheid die je machine blootstelt aan kwaadwillige gebruikers. Niet gecorrigeerde systemen zijn een veel voorkomende oorzaak van computer indringingen. Je moet een plan hebben om beveiligings correcties tijdig aan te brengen om deze kwetsbaarheden te verwijderen zodat ze niet uitgebuit kunnen worden.

Voor thuis gebruikers, moeten beveiligings correcties zo spoedig mogelijk geïnstalleerd worden. Het instellen van automatische installatie van beveiligings vernieuwingen is een manier om er niet over na te hoeven denken, maar het heeft een klein risico dat iets een conflict met je instelling veroorzaakt of met andere software op je systeem.

Voor zakelijke of gevorderde thuis gebruikers, moeten beveiligings vernieuwingen getest worden en ingepland voor installatie. Extra controles zullen gebruikt moeten worden om het systeem te beschermen gedurende de tijd tussen de vrijgave van de correctie en zijn installatie op het systeem. Deze controles zullen afhangen van de exacte kwetsbaarheid, maar kunnen onder andere extra firewall regels, het gebruik van externe firewalls, of veranderingen in de software instellingen inhouden.

6.3. Het aanpassen van automatische vernieuwingen

Fedora is ingesteld om alle vernieuwingen op een dagelijkse basis toe te passen. Als je wilt veranderen hoe je systeem vernieuwingen installeert moet je dit doen met de '"Software-bijwerkvoorkeuren"'. Je kunt onder andere instellen, het schema, het type vernieuwingen toe te passen of je bericht geven van beschikbare vernieuwingen.

In Gnome, you can find controls for your updates at: `System -> Preferences -> Software Updates`. In KDE it is located at: `Applications -> Settings -> Software Updates`.

6.4. Installeer ondertekende pakketten van goed bekende repositories

Software pakketten worden verspreid met behulp van repositories. Alle bekende repositories ondersteunen pakket ondertekening. Pakket ondertekening gebruikt publieke sleutel technologie om te bewijzen dat het pakket dat verspreid wordt door de repositorie niet veranderd is nadat de

ondertekening was aangebracht. Dit biedt enige bescherming tegen het installeren van software die kwaadwillig veranderd kan zijn nadat het pakket is gemaakt maar voordat jij het downloadt.

Het gebruiken van te veel repositories, onvertrouwde repositories, of repositories met niet ondertekende pakketten heeft een hoger risico voor het introduceren van kwaadwillige of kwetsbare code op je systeem. Wees voorzichtig als je repositories toevoegt aan yum/software vernieuwing.

Referenties

De volgende referenties zijn verwijzingen naar extra informatie die relevant is voor SELinux en Fedora maar die buiten het kader van deze gids valt. Merk op dat door de snelle ontwikkeling van SELinux sommige onderdelen van deze informatie alleen betrekking heeft op specifieke vrijgaves van Fedora.

Boeken

SELinux by Example

Mayer, MacMillan, en Caplan

Prentice Hall, 2007

Handleidingen en hulp

Understanding and Customizing the Apache HTTP SELinux Policy

http://fedora.redhat.com/docs/selinux-apache-fc3/

Handleidingen en voordrachten van Russell Coker

http://www.coker.com.au/selinux/talks/ibmtu-2004/

Algemene SELinux tactiek schrijven HOWTO

http://www.lurking-grue.org/writingselinuxpolicyHOWTO.html

Red Hat Knowledgebase

http://kbase.redhat.com/

Algemene informatie

NSA SELinux hoofd website

http://www.nsa.gov/selinux/[1]

NSA SELinux FAQ

http://www.nsa.gov/selinux/info/faq.cfm[2]

Fedora SELinux FAQ

http://fedora.redhat.com/docs/selinux-faq-fc3/

SELinux NSA's Open Source Security Enhanced Linux

http://www.oreilly.com/catalog/selinux/

Technologie

Een overzicht van object classes en permissies

http://www.tresys.com/selinux/obj_perms_help.html

Integrating Flexible Support for Security Policies into the Linux Operating System (a history of Flask implementation in Linux)

http://www.nsa.gov/research/_files/selinux/papers/selsymp2005.pdf

[1] http://www.nsa.gov/research/selinux/index.shtml
[2] http://www.nsa.gov/research/selinux/faqs.shtml

Implementing SELinux as a Linux Security Module
http://www.nsa.gov/research/_files/publications/implementing_selinux.pdf

A Security Policy Configuration for the Security-Enhanced Linux
http://www.nsa.gov/research/_files/selinux/papers/policy/policy.shtml

Gemeenschap

Fedora SELinux gebruikers gids
http://docs.fedoraproject.org/selinux-user-guide/

Fedora SELinux Managing Confined Services Guide
http://docs.fedoraproject.org/selinux-managing-confined-services-guide/

SELinux gemeenschap pagina
http://selinux.sourceforge.net

IRC
irc.freenode.net, #selinux, #fedora-selinux, #security

Geschiedenis

Een korte geschiedenis van Flask
http://www.cs.utah.edu/flux/fluke/html/flask.html

Volledige achtergrond over Fluke
http://www.cs.utah.edu/flux/fluke/html/index.html