Linux Säkerhets HOWTO

Kevin Fenzi, `kevin@scrye.com` & Dave Wreski, `dave@nic.com`. Svensk översättning av Tomas Carlsson `md5tc@mdstud.chalmers.se`

v0.9.11, 1 May 1998. Svensk version Augusti 1998

Detta dokument är en allmän översikt av de säkerhetsaspekter som en administratör av ett Linuxsystem stöter på. Det täcker allmän säkerhetsfilosofi och ett antal specifika exempel på hur du kan göra ditt Linuxsystem säkrare mot inkräktare. Det hänvisas även till annat säkerhetsrelaterat material och program. OBS: Detta är en betaversion av detta dokument. Förbättringar, konstruktiv kritik, tillägg och rättningar tas tacksamt emot. Var vänlig skicka din feedback till båda författarna per e-post. Se till att ha med "Linux", "security" eller "HOWTO" i ämnesraden i ditt meddelande för att undvika spammningsfilter och för att ditt meddelande skall uppmärksammas snabbt av författarna.

Innehåll

1 Introduktion

Detta dokument täcker några av de huvudsakliga säkerhetsaspekter som påverkar säkerheten i Linux. Generell filosofi och ”nätfödda” resurser diskuteras.

Ett antal andra HOWTO-dokument överlappar med dessa säkerhetsaspekter och det har hänvisats till dessa där så är fallet.

Det är INTE meningen att detta dokument skall vara ett färskt dokument om olika attacker som kan förekomma. Ett stort antal nya attacker förekommer ständigt. Detta dokument talar om var man kan leta efter sådan färsk information, och några allmäna metoder för att förebygga att sådana attacker äger rum.

1.1 Nya versioner av detta dokument

Nya versioner av detta dokument kommer periodvis att postas till diskussionsgruppen *comp.os.linux.answers*. De kommer även att finnas på de olika anonyma FTP-sajter som arkiverar sådan information, inklusive:

`ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO`

Dessutom så bör du vanligtvis kunna hitta detta dokument på Linux WorldWideWeb hemsidan via:

`http://sunsite.unc.edu/mdw/linux.html`

Till sist, den allra senaste versionen av detta dokument bör också finnas tillgängligt i olika format från:

`http://scrye.com/\~{}kevin/lsh/`

1.2 Feedback

Alla kommentarer, felrapporter, ytterligare information och kritik av alla de slag skall skickas till:

`kevin@scrye.com`

och

`dave@nic.com`

OBS: Var vänlig och skicka din feedback till _båda_ författarna. Se även till att ha med ”Linux”, ”security” eller ”HOWTO” i din ämnesrad för att undvika Kevins spammningsfilter.

1.3 Disclaimer

Inget ansvar för innehållet i detta dokument kan accepteras. Använd principer, exempel och annat innehåll på egen risk. Dessutom, detta är en tidig version med många möjligheter för oriktigheter och fel.

Ett antal av exemplen och beskrivningarna använder upplägg och systemkonfiguration från distributionen RedHat(tm). Användbarheten kan variera.

Så långt som vi vet så beskrivs endast program som under särskilda villkor kan användas eller utvärderas i personligt syfte. De flesta av programmen är tillgängliga med komplett källkod under GNU-lika villkor.

1.4 Information om kopieringsrätt

1.4.1 Svensk översättning

Denna svenska översättning skall inte betraktas som juridiskt bindande. Det är originalversionen på engelska som gäller, se nästa rubrik.

Detta dokument är kopieringsrättsskyddat (c)1998 Kevin Fenzi och Dave Wreski, och distribueras under följande villkor:

- Linux HOWTO-dokument får reproduceras och distribueras i sin helhet eller i delar, i vilket fysiskt eller elektroniskt medium som helst, så länge som denna kopieringsrätts notis inkluderas oförändrad på alla kopior. Kommersiell distribution är tillåten och uppmuntras; men, författarna vill gärna bli informerade om alla sådana distributioner.
- Alla översättningar, härledda arbeten eller förenade arbeten som införlivar något Linux HOWTO-dokument måste täckas under denna kopieringsrättsnotis. Det vill säga, du får inte producera ett härlett arbete från en HOWTO och sedan lägga till ytterligare restriktioner för dess distribution.

Undantag för dessa regler kan tillåtas under speciella förhållanden; var vänlig kontakta koordinatorn för Linux HOWTOs på adressen som ges nedan.

- Om du har några frågor var vänlig kontakta Tim Bynum, koordinatorn för Linux HOWTOs på:

`linux-howto@sunsite.unc.edu`

1.4.2 Engelsk version (denna gäller)

2 Översikt

Detta dokument försöker förklara några tillvägagångssätt och vanligt förekommande mjukvara som kan hjälpa ditt Linuxsystem att bli säkrare. Det är viktigt att diskutera några av de grundläggande principerna först, och skapa en stomme för säkerheten innan vi sätter igång.

2.1 Varför behöver vi säkerhet?

Säkerhet blir en allt viktigare aspekt i den ständigt förnyande världen av global datakommunikation, billiga Internetanslutningar och snabb mjukvaruutveckling. Säkerhet är numer ett grundläggande krav eftersom global dataöverföring är osäker till sin natur. När din data går från punkt A till punkt B på Internet, till exempel, kan det passera genom flera andra punkter på vägen. Detta ger andra användare möjligheten att snappa upp, och till och med ändra, din data. Även andra användare på ditt system kan illvilligt förändradin data till något som du inte alls hade tänkt. Otillåten access till ditt system kan fås av inkräktare, även kända som "crackers", som sedan använder avancerad kunskap för attimitera dig, stjäla information från dig eller till och med neka dig access till dina egna resurser. Om du fortfarande undrar vad det är för skillnad mellan en "Hacker" och en "Cracker" så bör du läsa Eric Raymonds dokument, "How to Become A Hacker", tillgängligt på `http://sagan.earthspace.net/\~{}esr/faqs/hacker-howto.html`. (Det finns även en svensk version tillgänglig som "inofficiell" HOWTO på samma ställe som _detta_ dokument).

2.2 Hur säkert är säkert?

Först och främst, kom ihåg att inget datorsystem kan någonsin vara "helt och hållet säkert". Allt du kan göra är att göra det svårare för någon att äventyra säkerheten på ditt system. För den normale hemanvändaren av Linux så krävs det inte mycket för att hålla en planlös cracker på avstånd. För högprofilanvändare av Linux (banker, telekommunikationsföretag, etc) så krävs mycket mer arbete.

En annan faktor at ta med i beräkningarna är att ju säkrare ditt system är desto mer störande blir säkerheten. Du måste avgöra i denna balansgång ditt system fortfarande äranvändbart och fortfarande säkert för dina syften. Till exempel, du kan kräva att alla som ringer in till ditt system använder ett ”callback” modem som ringer upp respektive användare på deras hemnummer. Detta är säkrare, men om någon inte är hemma så blir det svårt för dem att logga in. Du kan också konfigurera ditt system utan nätverksanslutningar till Internet, men detta gör det svårare att surfa på webben.

Om du har en stor till medelstor sajt så bör du ställa upp en ”Säkerhetspolicy” som säger hur mycket säkerhet som krävs av din sajt och vilken granskning som behövs för att kolla den. Du kan hitta ett välkänt exempel på en säkerhetspolicy på `http://ds.internic.net/rfc/rfc2196.txt`. Den har nyss blivit uppdaterad och innehåller ett bra ramverk för en säkerhetspolicy åt ditt företag.

2.3 Vad försöker du att skydda?

Innan du försöker säkra ditt system så bör du slå fast vilken nivå av hot du måste skydda dig emot, vilka risker du bör eller inte bör ta och hur sårbart ditt system är som resultat. Du bör analysera ditt system för att veta vad du skyddar, varför du skyddar det, vilket värde det har och vem som har ansvaret för din data och andra tillgångar.

- Risken finns att en inkräktare kan lyckas med ett försök att få tillgång till din dator. Kan en inkräktare läsa, skriva filer eller exekvera program som kan göra skada? Kan de radera kritisk data? Kan de hindra dig eller ditt företag från att göra viktigt arbete? Glöm inte att någon som får tillgång till ditt användarkonto eller system kan också imitera dig. Dessutom, att ha ett osäkert användarkonto på ditt system kan resultera i att hela ditt nätverk kan äventyras. Genom att ge en ensam användare tillåtelse att logga in genom en rhosts-fil, eller genom att tillåta användning av en osäker tjänst, som tftp, riskerar du att en inkräktare använder detta för ”få in en fot genom dörren”. När väl inkräktaren har ett användarkonto på ditt system, eller någon annans system, så kan det användas för att få access till ett annat system eller ett annat användarkonto.
- Typiska hot kommer från någon med motivation att få otillåten access till ditt nätverk eller dator. Du måste avgöra vem du kan lita på som kan ha tillgång till ditt system och vilka hot de kan införa. Det finns flera sorters inkräktare och det är användbart att komma ihåg de karakteristiken för de olika när du säkrar ditt system.
  - **Den Nyfikne** - Denna typ av inkräktare är huvudsakligen intresserad av att få reda på vilket typ av system och data du har.
  - **Den Illvillige** - Denna typ av inkräktare är antingen ute efter att sänka ditt system, eller att ändra utseendet på din webbplats, eller på något annat sätt få dig att ödsla tid och pengar på att återställa det.
  - **Högprofilinkräktaren** - Denna typ av inkräktare försöker använda ditt system för att bli populär och berömd. Han kan använda ditt högprofilsystem för att annonsera sina kunskaper.
  - **Tävlingsinkräktaren** - Denna typ av inkräktare är intresserad av vad du har för data på ditt system. Det kan vara någon som tror att du har något som som kan främja honom ekonomiskt eller på något annat sätt.
- Sårbarhet beskriver hur bra skyddad din dator är från ett annat nätverk, och potentialen för att någon skall kunna få otillåten tillgång. Vad är det som står på spel om någon bryter sig in i ditt system? Självklart är intressena hos en dynamisk PPP-hemanvändare annorlunda från dem hos ett företag som ansluter sin maskin till Internet, eller något annat stort nätverk.

Hur mycket tid skulle det ta att återfå/återskapa någon data som försvann? En första investering i tid nu kan spara tio gånger mer tid senare om du måste återskapa data som försvann. Har du kollat din backupstrategi och verifierat din data på senare tid?

2.4 Utveckla en säkerhetspolicy

Skapa en enkel, allmän policy för ditt system som dina användare kan förstå och följa. Den bör skydda den data som du skyddar, såväl som användarnas integritet. Vissa saker som man kan tänka sig att lägga till är vem som har access till systemet (Får min kompis använda mitt användarkonto?), vem får installera mjukvara på systemet, vem äger vilken data, återställning efter katastrof och korrekt användning av systemet.

En allmänt accepterad säkerhetspolicy börjar med frasen:

”Det som inte är tillåtet är förbjudet”

Detta betyder att om du inte ger en användare access till en tjänst så skall den användaren inte använda tjänsten förrän du ger access. Se till att policyn fungerar på ditt normala användarkonto. Att säga, ”Ah, jag kan inte lista ut problemen med rättigheterna här, jag gör det som root”, kan leda till säkerhetsläckor som är väldigt självklara och till och med sådana som inte har upptäckts än.

2.5 Möjligheter att säkra din sajt

Detta dokument kommer att diskutera olika möjligheter på vilka du kan säkra dina tillgångar som du har jobbat hårt för: din lokala maskin, data, användare, nätverk och till och med ditt rykte. Vad skulle hända med ditt rykte ifall en inkräktare raderade dina användares data? Eller gjorde om din webbplats? Eller publicerade ditt företags interna projektplan för nästa kvartal? Om du planerar en nätverksinstallation så finns det många faktorer som du måste ta med i beräkningarna innan du ansluter en enda maskin till ditt nätverk.

Även om du har en enda uppringd PPP-förbindelse, eller bara en liten sajt så betyder inte det att inkräktare inte är intresserade i dina system. Det är inte bara stora, högprofilsajter som är potentiella mål, många inkräktare vill bara attackera så många sajter som möjligt, oavsett storlek. Dessutom, så kan de använda ett säkerhetshål på din sajt för att få access till andra sajter som du är ansluten till.

Inkräktare har mycket tid att spilla och kan undvika att gissa hur du har blottat ditt system genom att helt enkelt testa alla möjligheter. Det finns också flera anledningar till varför en inkräktare kan vara intresserad av dina system, vilka vi diskuterar senare.

2.5.1 Datorsäkerhet

Den kanske största ansträngningen med säkerhet görs på datorbaserad säkerhet. Detta handlar vanligtvis om att se till att ditt eget system är säkert, och hoppas på att alla andra på ditt nätverk gör detsammma. Att välja bra lösenord, säkra din dators lokala nätverkstjänster, ha bra loggningsmöjligheter och uppgradera program med kända säkerhetsläckor ingår bland de saker som den lokala säkerhetsadministratören har ansvar för att göra. Även att detta är absolut nödvändigt så kan det bli en skrämmande uppgift när ditt nätverk av makiner blir större.

2.5.2 Nätverkssäkerhet

Nätverkssäkerhet är lika nödvändigt som lokal datorsäkerhet. Med ditt system, eller ett distribuerat datanätverk, Internet eller hundratals, om inte tusentals datorer på samma nätverk, så kan du inte lita till att alla de systemen är säkra. Se till att endast tillåtna användare kan använda dina nätverksresurser, bygga brandväggar, använda stark kryptering och se till att det inte finns några ”skurkaktiga” eller osäkra maskiner på ditt nätverk ingår i åtaganden för en administratör för nätverkssäkerhet.

Detta dokument kommer att diskutera några av de tekniker som kan användas för att säkra din sajt, och förhoppningsvis visa dig några av de sätt som finns för att förhindra en inkräktare från att få tillgång till det du försöker skydda.

2.5.3 Säkerhet genom mörkläggning

En typ av säkerhet som måste diskuteras är ”säkerhet genom mörkläggning”. Detta betyder att göra något som till exempel byta loginnamn från ”root” till ”toor” för att förhindra någon att bryta sig in i ditt system som root är bara en falsk känsla av säkerhet, och kommer att reslutera i väldigt otrevliga konsekvenser. Lita på att en som attackerar ditt system snabbt kommer att se igenom sådana tomma säkerhetstilltag. Bara för att du har en liten sajt eller relativt låg profil betyder inte att en inkräktare inte är intresserad av vad du har. Vi kommer att diskutera vad du skyddar i följande avsnitt.

2.6 Organisation av detta dokument

Detta dokument har delats in i ett antal avsnitt. De täcker ett flertal typer av säkerhetsaspekter. Det första, fysisk säkerhet, behandlar hur du behöver skydda din fysiska maskin från manipulation. Det andra beskriver hur du skyddar ditt system från att bli manipulerat av lokala användare. Det tredje, filer och filsystemsäkerhet, visar hur du bör sätta upp dina filsystem och rättigheter på dina filer. Nästa avsnitt, lösenordsäkerhet och kryptering, behandlar hur du kan använda kryptering för att säkra din maskin och ditt nätverk. Säkerhet i kärnan behandlar hur behandlar vilka parametrar du bör sätta eller känna till för en säkrare maskin. Nätverkssäkerhet beskriver hur du säkrar ditt Linuxsystem från nätverksattacker. Säkerhetsförberedning behandlar hur du förbereder dina maskiner innan du kopplar in dem. Nästa avsnitt behandlar vad man kan göra när man upptäcker ett intrång är på gång eller nyss har inträffat. Sedan följer länkar till andra säkerhetsresurser och slutligen lite frågor och svar och några avslutande ord.

De två huvudaspekterna att inse när du läser detta dokument är:

- Känn till ditt system. Kolla systemloggar som `/var/log/messages` och håll ett öga på ditt system, och
- Håll ditt system uppdaterat genom att se till att du har installerat de senaste versionerna av mjukvaran och har uppgraderat efter uppgifter om säkerhetsrisker. Endast detta hjälper dig att hålla ditt system betydligt mer säkert.

3 Fysisk säkerhet

Det första ”lagret” av säkerhet som du måste ta med i beräkningarna är den fysiska säkerheten för ditt datorsystem. Vem har direkt fysisk access till din maskin? Bör de ha det? Kan du skydda din maskin från deras mixtrande? Bör du det?

Hur mycket fysisk säkerhet du behöver för ditt system beror väldigt mycket på din situation och/eller budget.

Om du är en hemanvändare så behöver du antagligen inte göra så mycket (även om du kanske måste skydda din maskin från klåfingriga barn eller släktingar). Om du är i en labmiljö så behöver du nämnvärt mer, men användarna måste fortfarande kunna arbeta med maskinerna. Många av de avsnitt som följer kommer att hjälpa dig. Om du jobbar på ett kontor så måste du kanske eller kanske inte säkra din maskin när du inte är där. På vissa företag är det ett brott som kan straffas med avsked att inte säkra sin maskin när man lämnar den.

Självklara fysiska säkerhetsmetoder som lås på dörrar, kablar, låsta skåp och videoövervakning är alla bra, men ligger utanför räckvidden av detta dokument. :)

3.1 Datorlås

Många moderna pc-lådor inkluderar en låsfunktion. Vanligtvis är detta ett lås på framsidan av lådan där du kan låsa eller låsa upp med en medföljande nyckel. Dessa lås kan hjälpa till att förebygga att någon stjäl din pc, eller öppnar lådan och manipulerar/stjäler din hårdvara. De kan ibland också hindra att någon återstartar din dator på sin egen diskett eller annan hårdvara.

Dessa lås gör olika saker beroende på vad som stöds av moderkortet och hur lådan är konstruerad. På många pc's måste du förstöra lådan för att kunna öppna den. På vissa andra kan man inte koppla in nya tangentbord eller möss. Se i instruktionerna till ditt moderkort eller låda för mer information. Detta kan ofta vara en användbar egenskap, men låsen är ofta av dålig kvalitet och kan lätt forceras av en attack med dyrkar.

Vissa lådor (mest sparc och mac) har en bygel på baksidan så att om du trär en kabel igenom den så måste attackerare klippa av kabeln eller förstöra lådan för att komma in i den. Att endast sätta ett hänglås eller kombinationslås igenom dessa kan förhindra att någon stjäl din maskin.

3.2 BIOS-säkerhet

BIOS är den lägsta nivån av mjukvara som konfigurerar eller manipulerar din x86-baserade hårdvara. LILO och andra uppstartsmetoder för Linux använder BIOS för att få reda på hur de skall starta upp din Linux-maskin. Annan hårdvara som Linux körs på har liknande mjukvara (OpenFirmware på mac och nya sun, sun boot prom, osv...). Du kan använda ditt BIOS för att hindra attackerare att återstarta din maskin och manipulera ditt Linuxsystem.

Under Linux/x86 tillåter många pc BIOS att du kan använda ett "boot"-lösenord. Detta tillhandahåller inte så värst mycket säkerhet (BIOS kan nollställas, eller tas bort om någon kan öppna lådan), men det kan vara ett bra avskräckningsmedel (dvs det tar tid och lämnar spår av mixtring).

Många x86 BIOS tillåter även att du specificerar diverse andra bra säkerhetsinställningar. Titta i din manual för BIOS eller titta i det nästa gång du startar upp. Några exempel är: tillåt inte uppstart från diskett och lösenord för att få tillgång till vissa BIOS-egenskaper.

På Linux/Sparc kan ditt SPARC EEPROM ställas in för att kräva ett lösenord för uppstart. Detta kan sakta ner attackerare lite granna.

OBS: Om du har en servermaskin och du ställer in ett lösenord för uppstart så kan din maskin inte startas upp utan att någon finns där. Kom ihåg att du måste gå dit och skriva in lösenordet även om det har varit strömavbrott. ;(

3.3 Boot-laddarsäkerhet

De olika boot-laddarna som finns för Linux kan också ställas in för att kräva ett lösenord. Med lilo kan du titta på inställningarna "restricted" och "password". "password" låter dig ställa in ett lösenord för uppstart. Med "restricted" så startas maskinen upp som vanligt _om inte_ någon specificerar några parametrar till lilo-prompten (som "single").

Kom ihåg att när du ställer in alla dessa lösenord så måste du även komma ihåg dem. :) Kom också ihåg att lösenord saktar endast ner en beslutsam attackerare. Detta hindrar ingen från att starta från en diskett och montera din rootpartition. Om du använder säkerhet i samband med en boot-laddare, så kan du likaväl neka uppstart från diskett i ditt BIOS, såväl som att lösenordsskydda ditt BIOS.

Om någon har säkerhetsrelaterad information från en annan boot-laddare, så vill vi gärna höra den. (grub, silo, milo, linload, etc).

OBS: Om du har en servermaskin och du ställer in ett lösenord för uppstart så kan din maskin inte startas upp utan att någon finns där. Kom ihåg att du måste gå dit och skriva in lösenordet även om det har varit strömavbrott. ;(

3.4 xlock och vlock

Om du går iväg från din maskin då och då, så kan det vara trevligt att ”låsa” din konsoll så att ingen mixtrar eller tittar på ditt arbete. Två program som gör detta är: xlock och vlock.

Xlock låser din X-skärm. Den bör finnas i alla Linux-distributioner som stödjer X. Titta på manualbladet för det för mer valmöjligheter, men generellt kan du köra xlock från en xterm på din konsoll och det låser skärmen och kräver ditt lösenord för att låsa upp.

vlock är ett enkelt litet program som låter dig låsa vissa eller alla virtuella konsoller på din Linuxburk. Du kan låsa den du arbetar i eller allihopa. Om du bara låser en så kan andra komma och använda konsollen, de kommer bara inte kunna använda din vty förrän du låser upp den. Vlock finns i RedHat Linux, men din lycka kan variera.

När du låser din konsoll så hindrar det att någon mixtrar med ditt arbete, men det hindrar inte att någon startar om din maskin eller på annat sätt avbryter ditt arbete. Det hindrar inte heller att någon skapar problem genom att mixtra med din maskin via nätverket.

3.5 Upptäcka äventyrande av fysisk säkerhet

Den första saken att alltid kolla upp är när din maskin blev återstartad. Eftersom Linux är ett robust och stabilt operativsystem, så är de enda gånger din maskin skall återstartas när DU tar ner den för uppgradering av operativsystem, utbyte av hårdvara eller liknande. Om din maskin har återstartats utan att du har gjort det, så skall varningslampan tändas. Många av de sätt på vilka din maskins säkerhet kan äventyras kräver att inkräktaren startar om eller stänger av din maskin.

Titta efter tecken på att någon mixtrat med lådan och datorområdet. Även om många inkräktare städar undan alla spår av deras närvaro från systemloggar, så är det en bra ide att titta igenom allihop och notera alla avvikelser.

Några saker att titta efter i dina loggar:

- Korta eller inkompletta loggar.
- Loggar som innehåller konstiga tidmarkeringar.
- Loggar med inkorrekta rättigheter eller ägare.
- Meddelanden om återstart av system eller tjänster.
- Saknade loggar.
- Rader med su eller inloggning från konstiga platser.

Vi behandlar data från systemloggar senare i denna HOWTO.

4 Lokal säkerhet

Nästa sak att titta på är säkerheten i ditt system mot attacker från lokala användare. Sade vi just _lokala_ användare? Ja.

Att få tillgång till en lokal användares användarkonto är en av de första saker som en inkräktare försöker med på sin väg för att komma åt root-användarkontot. Med avsaknad av lokal säkerhet så kan de ”uppgradera” sin normala användaracceess till rootaccess genom att använda sig av en mängd buggar och dåligt inställda lokala tjänster. Om du ser till att din lokala säkerhet är snäv, så har inkräktaren ytterligare ett hinder att gå förbi.

Lokala användare kan också skapa mycket trubbel med ditt system även (särskilt) om de verkligen är de de säger att de är. Att ge användarkonton till personer som du inte känner eller inte har någon information om är en väldigt dålig ide.

4.1 Skapa nya användarkonton

Du bör se till att endast ge de minimala rättigheter till ett användarkonto som krävs för dess uppgifter. Om du vill ge din son (ålder 10) ett användarkonto så kanske du bara vill att han skall ha tillgång till en ordbehandlare och ett ritprogram, men inte kunna radera data som inte är hans.

Flera bra tumregler när du skall ge andra personer tillgång till din Linuxmaskin:

- Ge dem minimalt med rättigheter.
- Ha kontroll på när och var de loggar in ifrån, eller bör logga in ifrån.
- Se till att ta bort deras användarkonto när de inte längre behöver det.

Många lokala användarkonton som används i nedbrytning av säkerhet är sådana som inte har använts på flera månader eller år. Eftersom ingen använder dem så är de det ideala sättet att göra en attack med.

4.2 Root-säkerhet

Det mest eftersökta användarkontot på din maskin är användarkontot för ”superuser”. Detta användarkonto har auktoritet över hela maskinen, vilket även kan inkludera auktoritet över andra maskiner på nätverket. Kom ihåg att du endast skall använda root-användarkontot för väldigt korta specifika uppgifter och skall mestadels köra som en normal användare. Att köra som root hela tiden är en väldigt väldigt väldigt dålig ide.

Flera trix för att undvika att stöka till i din egen burk som root:

- När du skall göra ett komplext kommando, försök först med att köra det på ett ickedestruktivt sätt...särskilt kommandon med mycket utskrifter: dvs, du är på väg att göra en ”rm foo*.bak”, men istället gör du först ”ls foo*.bak” och ser till att du är på väg att radera de filer du har tänkt. Att använda echo istället för destruktiva kommandon kan också fungera ibland.
- Vissa personer använder ”touch /-i” på sina system. Detta gör att kommandon som ”rm -rf *” frågar om du verkligen vill ta bort alla filerna. (Det gör detta genom att ditt shell löser upp ”-i” filen först, och behandlar den som -i parametern till rm.) Detta hjälper dock inte med rm kommandon utan * i sig. ;(
- Bli endast root för att göra enstaka specifika uppgifter. Om du märker att du håller på att lista ut hur man gör en viss sak, gå tillbaka till normal användare tills du är **säker** på vad som behöver göras som root.

- Sökvägen för kommandon för root-användaren är väldigt viktig. Kommandot path, eller omgivningsvariabeln PATH, definierar var shellet skall söka efter program. Försök att minimera sökvägen för root-användaren så mycket som möjligt, och använd aldrig ".", vilket betyder "aktuell katalog" i din PATH. Dessutom så skall du aldrig ha skrivbara kataloger i sökvägen, eftersom detta kan tillåta attackerare att placera nya binärfiler i din sökväg och låter dem köra som root nästa gång du kör det kommandot.
- Använd aldrig rlogin/rsh/rexec (kallas r-programmen) sviten av verktyg som root. De är föremål för många typer av attacker, och är riktigt farliga om de körs som root. Skapa aldrig en .rhosts-fil för root.
- Filen /etc/securetty innehåller en lista på terminaler som root kan logga in från. Som standard (i RedHat Linux) är detta satt endast till de lokala virtuella konsollerna (vtys). Var väldigt försiktig med att lägga till något annat i denna filen. Du bör kunna logga in från ett annat system som en normal användare och sedan göra "su" om du behöver (förhoppningsvis via ssh eller någon annan krypterad kanal), så det finns ingen anlednig att kunna logga in direkt som root.
- Var alltid långsam och genomtänkt när du kör som root. Ditt agerande kan påverka många saker. Tänk innan du skriver!

Om du absolut måste låta någon (förhoppnigsvis en som du litar på) få "superuser" access till din maskin, så finns det några verktyg som kan hjälpa till. Sudo låter användare använda sitt lösenord för att få tillgång till ett begränsat antal kommandon som root. Med detta kan du, till exempel, låta en användare mata ut och montera flyttningsbar media på din Linuxburk, men inte ha några andra root-privilegier. Sudo loggar alla lyckade och misslyckade sudo-försök, vilket låter dig spåra vem som använde vilket kommando för att göra vad. Av denna anledningen så fungerar sudo bra även på platser där ett antal personer har root-access, men använd sudo så att du kan hålla reda på ändringar som görs.

Även om sudo kan användas för att ge specifika användare specifika privilegier för specifika uppgifter, så har det flera nackdelar. Det skall enbart användas för ett begränsat antal uppgifter, som att starta om en server, eller lägga till nya användare. Alla program som tillhandahåller att man kan hoppa ut i shellet kommer att ge användaren root-access. Detta inkluderar de flesta editorer, till exempel. Dessutom, ett så oskyldigt program som /bin/cat kan användas för att skriva över filer, vilket kan göra att root blir attackerat. Tänk på sudo som ett hjälpmedel för att ha kontroll, och förvänta dig inte att det skall ersätta root-användaren och dessutom vara säkert.

5 Säkerhet i filer och filsystem

Några minuters förberedelse och planering innann du kopplar upp ditt system kan hjälpa till att skydda ditt system och den data som finns lagrad på det.

- Det bör aldrig finnas någon anledning att användarnas hemkataloger skall tillåta att SUID/SGID-program körs ifrån dem. Använd parametern "nosuid" i /etc/fstab för partitioner som är skrivbara av andra än root. Du kanske även bör använda "nodev" och "noexec" på användarnas hempartitioner, såväl som på /var. Detta förbjuder exekvering av program, och att man skapar blockenheter, vilket aldrig skall vara nödvändigt ändå.
- Om du exporterar filsystem via NFS, se till att du konfigurerar /etc/exports med de mest restriktiva rättigheterna som går. Detta betyder att inte använda jokrar, inte tillåta root-skrivaccess, och montera "readonly" när det är möjligt.
- Konfigurera dina användares umask för filskapning till att vara så restriktiv som möjligt. Vanliga inställningar är 022, 033 och den mest restriktiva 077, och läggs i /etc/profile.

- Sätt gränser för filsystemen istället för att tillåta ”obegränsad” som är standard. Du kan kontrollera begränsningar per-användare genom att använda PAM-modulen för resursbegränsningar och /etc/pam.d/imits.conf. Till exempel, gränserna för gruppen ”users” kan se ut så här:

```
@users     hard  core    0
@users     hard  nproc   50
@users     hard  rss     5000
```

  Detta säger att det är förbjudet att skapa core-filer, begränsar antalet processer till 50, och begränsar minnesanvändningen per användare till 5Mb.

- Filerna /var/log/wtmp och /var/run/utmp innehåller logininformation om alla användare på ditt system. Dess integritet måste bibehållas eftersom de kan användas för att avgöra när och varifrån en användare (eller en potentiell inkräktare) har kommit in i ditt system. Dessa filerna bör också ha rättigheterna 644, utan att påverka normal systemoperation.

- Biten för ”icke-ändringsbar” kan användas för att förhindra att någon av misstag raderar eller skriver över en fil som måste vara skyddad. Det förhindrar även att någon skapar en symbolisk länk till den filen, vilket har varit grunden till attacker som handlat om att radera /etc/passwd eller /etc/shadow. Se manualbladet för chattr(1) för mer information biten för ”icke-ändringsbar”.

- Alla SUID och SGID filer på ditt system är potentiella säkerhetsrisker, och skall kollas noga. Eftersom dessa program ger speciella privilegier till användaren som kör dem, så är det nödvändigt att vara säker på att inga osäkra program installeras. Ett favorittrix förcrackers är att attackera SUID ”root” program, och sedan lämna ett SUID-program som bakdörr för att komma in nästa gång, även om det ursprungliga hålet är igenpluggat. Hitta alla SUID/SGID-program på ditt system, och håll reda på vad de gör, så att du är medveten om alla ändringar som tyder på en potentiell inkräktare. Använd följande kommando för att hitta alla SUID/SGID-program på ditt system:

```
root#  find / -type f \( -perm -04000 -o -perm -02000 \)
```

  Du kan ta bort SUID eller SGID rättigheterna på ett misstänkt program med chmod(1), sedan ändra tillbaka om tycker att det är absolut nödvändigt.

- Filer som är skrivbara för alla, särskilt systemfiler, kan vara en säkerhetsrisk om en cracker får access till ditt system och ändrar dem. Dessutom är det farligt med kataloger som är skrivbara för alla eftersom de låter en cracker skriva och radera filer som han vill. För att lokalisera alla filer som är skrivbara för alla, använd följande kommando:

```
root# find / -perm -2 -print
```

  och se till att du varför de filerna är skrivbara. I normal operation så är flera filer skrivbara, inklusive några från /dev och symboliska länkar.

- Filer utan ägare kan också vara en indikation att en inkräktare har haft tillgång till ditt system. Du kan hitta filer som inte har någon ägare eller inte tillhör någon grupp med kommandot:

```
root# find / -nouser -o -nogroup -print
```

- Att hitta .rhosts-filer bör vara en del av dina vardagliga uppgifter som systemadministratör, eftersom dessa filer inte skall få existera på ditt system. Kom ihåg att en cracker bara behöver ett osäkert användarkonto för att få access till hela ditt nätverk. Du kan hitta alla .rhosts-filer på ditt system med följande kommando:

```
root# find /home -name .rhosts -print
```

- Till sist, innan du ändrar rättigheter på några systemfiler, se till att du förstår vad du gör. Ändra aldrig rättigheterna på en fil för att det verkar vara ett lätt sätt att få saker och ting att fungera. Kontrollera alltid varför en fil har vissa rättigheter innan du ändrar dem.

5.1 Umask-inställningar

Kommandot umask kan användas för att bestämma/ta reda på vilka rättigheter som skapade filer skall få som standard på ditt system. Det är det oktala komplementet till filrättigheterna. Om filer skapas utan hänsysn till rättigheter så kan en användare oavsiktligt ge läs eller skrivrättigheter till någon som inte borde ha det. Typiska umask-inställningar är 022, 027 och 077 (vilket är det mest restriktiva). Normalt sätts umask i /etc/profile så det gäller alla användare på systemet. Du kan till exempel ha en rad som ser ut så här:

```
# Set the user's default umask
umask 033
```

Se till att umask för root är 077, vilket slår av läs-, skriv- och exekverarättigheterna för andra användare om det inte explicit ändras med chmod(1).

Om du använder RedHat, och använder deras sätt att skapa användare och grupper (User Private Groups), så räcker det att använda 002 som umask. Detta för att standardkonfigurationen är en användare per grupp. (Åtminstonde versionerna 1.3 och 2.0 av Debian gör också detta!(SvÖ))

5.2 Filrättigheter

Det är viktigt att du försäkrar dig om att dina systemfiler inte är öppna för slapphänt modifiering av användare och grupper som inte borde utöva sådant systemunderhåll.

UNIX delar upp accesskontroll på filer och kataloger i tre kategorier: ägare, grupp och andra. Det finns alltid exakt en användare, hur många medlemmar av gruppen som helst och alla andra.

En snabb förklaring av rättigheter i UNIX:

- **Ägarskap** - Vilken/vilka användare och grupper har kontroll över rättigheterna för noden och nodens förälder.
- **Rättigheter** - Bitar som antingen är på eller av för att tillåta vissa typer av access till noden. Rättigheter för kataloger kan ha annan betydelse än samma rättigheter för en fil.
  - **Read**
    - Att kunna titta på innehållet i en fil.
    - Att kunna läsa en katalog
  - **Write**
    - Att kunna lägga till eller ändra en fil.
    - Att kunna radera eller flytta filer i en katalog.
  - **Execute**
    - Att kunna exekvera ett binärprogram eller shellskript.
    - Att kunna söka i en katalog, tillsammans med läsrättigheter.
  - **Save Text Attribute (för kataloger)**
    Den "klibbiga" (sticky) biten har också olika betydelse när den finns på kataloger. Om den är satt på en katalog så får en användare endast radera filer som användaren äger eller filer där han har explicit skrivrättighet, även om han har skrivrättighet till katalogen. Detta är gjort för kataloger som /tmp, vilken är skrivbar för alla, men det kanske inte är önskvärt att en användare kan ta bort filer som han vill. Denna biten ses som ett 't' i en lång kataloglistning.

- **SUID Attribut (för filer)**
  Detta beskriver sätt-användar-id rättigheter för filen. När SUID är satt i ägarrättigheterna, och filen är exekverbar, så får processer som kör den tillgång till systemresurser baserat på användaren som startade processen. Detta är anledningen till många "buffer overflow"-attacker.
- **SGID Attribut (för filer)**
  Om detta är satt i grupprättigheterna så kontrollerar denna biten sätt-grupp-id statusen för en fil. Detta beter sig på samma sätt som SUID, förutom att gruppen påverkas istället. Filen måste även vara exekverbar för att det skall ha någon effekt.
- **SGID Attribut (för kataloger)**
  Om du sätter SGID biten för en katalog (med "chmod g+s katalognamn"), så kommer filer som skapas i den katalogen att ha sin grupp satt till katalogens grupp.

Du - Ägaren av filen

Grupp - Gruppen du tillhör

Alla - Vem som helst på systemet som inte är ägare eller medlem i gruppen.

Filexempel:

```
-rw-r--r--  1 kevin  users         114 Aug 28  1997 .zlogin
1st bit - katalog?               (nej)
 2nd bit - läsbar för ägaren?     (ja, för kevin)
  3rd bit - skrivbar för ägaren?   (ja, för kevin)
   4th bit - exekverbar för ägaren? (nej)
    5th bit - läsbar för gruppen?    (ja, för users)
     6th bit - skrivbar för gruppen?  (nej)
      7th bit - exekverbar för gruppen?(nej)
       8th bit - läsbar för alla?       (ja, för alla)
        9th bit - skrivbar för alla?     (nej)
         10th bit - exekverbar för alla?  (nej)
```

Följande rader är exempel på de minimala rättigheter som krävs för att kunna utföra den access som beskrivs. Du kanske vill ge mer rättigheter än vad som listas, men detta bör beskriva vad de minimala rättigheterna på filer gör:

```
-r--------  Tillåt läsacces til filen för ägaren.
--w-------  Tillåter ägaren att ändra eller radera filen.
---x------  Ägaren kan exekvera detta program, men inte shellskript
            där även läsrättigheter krävs.
---s------  Kommer att exekvera med användar-id = ägare.
-------s--  Kommer att exekvera med användar-id = grupp.
-rw------T  Ingen uppdatering av "sista modifieringstid". Vanligtvis
            för swapfiler.
---t------  Ingen effekt. (före detta "sticky bit")
```

Katalogexempel:

```
drwxr-xr-x  3 kevin  users         512 Sep 19 13:47 .public_html/
1st bit - katalog?              (ja, den kan innehålla filer)
 2nd bit - läsbar för ägare?     (ja, för kevin)
  3rd bit - skrivbar för ägare?   (ja, för kevin)
```

```
4th bit - exekverbar för ägare? (ja, för kevin)
 5th bit - läsbar för grupp?     (ja, för users)
  6th bit - skrivbar för grupp?   (nej)
   7th bit - exekverbar för grupp? (ja, för users)
    8th bit - läsbar för alla?      (ja, för alla)
     9th bit - skrivbar för alla?    (nej)
      10th bit - exekverbar för alla? (ja, för alla)
```

Följande rader är exempel på de minimala rättigheter som krävs för att kunna utföra den access som beskrivs. Du kanske vill ge mer rättigheter än vad som listas, men detta bör beskriva vad de minimala rättigheterna på kataloger gör:

```
dr--------  Innehållet kan listas men filattributen kan inte läsas.
d--x------  Man kan gå in i katalogen och den kan användas i fulla
            exekveringssökvägar.
dr-x------  Filattribut kan nu läsas av ägaren.
d-wx------  Filer kan nu skapas/raderas, även om katalogen inte är
            den aktuella.
d------x-t  Förhindrar att filer kan raderas av andra med
            skrivrättigheter. Används på /tmp
d---s--s--  Ingen effekt.
```

Filer för systemkonfiguration (vanligtvis i /etc) har oftast rättigheterna 640 (-rw-r—–) och ägs av root. Beroende på säkerhetskraven på din sajt, så kanske du vill ändra detta. Låt aldrig systemfiler vara skrivbara för en grupp eller för alla. Vissa konfigurationsfiler, som /etc/shadow, bör endast vara läsbara för root. Kataloger i /etc bör åtminstonde inte vara tillgängliga för alla.

SUID shellskript

Shellskript med SUID-biten satt är en allvarlig säkerhetsrisk, av denna anledningen så tillåter inte kerneln sådana. Oavsett hur säkert du tycker att skriptet är, så kan det attackeras för att ge en cracker ett root-shell.

5.3 Integritetskontroll med Tripwire

Ett annat bra sätt att upptäcka lokala attacker (och även från nätverket) mot ditt system, är att köra en integritetskontrollerare som Tripwire. Tripwire kör ett antal checksummor på alladina viktiga binärfiler och konfigurationsfiler och jämför dem med en databas med tidigare värden som man vet är korrekta. Vilket betyder att ändringar i filer noteras.

Det är en bra ide att installera Tripwire på en diskett, och sedan fysiskt skrivskydda disketten. Nu kan inkräktare inte mixtra med själva Tripwire eller ändra i databasen. När du väl har installerat Tripwire så är det en bra ide att köra det som en del av ditt normalasystemunderhåll för att se om något har ändrats.

Du kan till och med lägga till en post i crontab för att köra Tripwire från din diskett varje natt och sedan e-posta dig resultaten på morgonen. Något som:

```
# set mailto
MAILTO=kevin
# run tripwire
15 05 * * * root /usr/local/adm/tcheck/tripwire
```

kommer att ge dig en rapport varje morgon klockan 05:15.

Tripwire kan vara en gudagåva för att upptäcka inkräktare tidigare än du hade gjort annars. Eftersom det är många filer som ändras på ett normalt system, så måste du vara försiktig med vad som är crackeraktivitet och vad du själv gör.

5.4 Trojanska hästar

Trojanska hästar benämns efter Homeros bra litteratur. Iden är att du lägger upp ett program eller binär som verkar jättebra, och får andra personer att ladda ner det och köra det som root. Sedan, kan du äventyra deras system medans de inte är uppmärksamma. Medans de tror att binären som de laddade ner gör en sak (som den mycket väl kan göra), så äventyrar den samtidigt deras säkerhet.

Du skall vara nogrann med vilka program som du installerar på din maskin. RedHat tillhandahåller MD5-summor och PGP-signaturer för RPM-filer så du kan verifiera att du installerar rätt program. Andra distributioner har liknande metoder. Du skall aldrig köra en binärfil som du inte har källkoden till eller som inte är välkänd som root. Få attackerare är villiga att släppa källkoden till allmänheten.

Även om det kan vara komplext, se till att du hämtar källkoden för program från dess riktiga distributionssajt. Om programmet skall köra som root, se till att antingen du eller någon du litar på tittar igenom källkoden och verifierar den.

6 Lösenordssäkerhet och kryptering

En av de viktigaste säkerhetsegenskaperna som används nuförtiden är lösenord. Det är viktigt för både dig och dina användare att ha säkra, icke gissningsbara lösenord. De flesta nyare Linuxdistributioner innehåller ”passwd”-program som inte tillåter för lätta lösenord. Se till att ditt ”passwd”-program är uppdaterat och har dessa egenskaper.

Djup diskussion om kryptering är utanför räckvidden av detta dokument, men en introduktion är på sin plats. Kryptering är väldigt användbart och till och med nödvändigt nuförtiden. Det finns alla möjliga olika metoder för att kryptera data, var och en med sina egna egenskaper.

De flesta unixar (och Linux är inget undantag) använder i huvudsak en envägs krypteringsalgoritm, som heter DES (Data Encryption Standard), för att kryptera lösenorden. Det krypterade lösenordet lagras sedan i (vanligtvis) /etc/passwd eller (mer sällan) /etc/shadow. När du försöker att logga in så krypteras det du skriver in och jämförs sedan med posten i den fil som lagrar lösenorden. Om det stämmer, så måste det vara rätt lösenord, och du får tillgång till systemet. Även om DES är en tvåvägs krypteringsalgoritm (du kan kryptera och dekryptera ett meddelande, givet att du har rätt nycklar.), så är varianten som de flesta unixar använder envägs. Detta betyder att det inte skall vara möjligt att göra krypteringen baklänges för att få fram lösenord i klartext från /etc/passwd (eller /etc/shadow).

Råstyrke-attacker (brute force), såsom ”Crack” eller ”John the Ripper” (se nedan) kan ofta gissa lösenord om lösenorden inte är tillräckligt slumpartat. PAM-moduler (se nedan) låter dig använda en annan krypteringsalgoritm till dina lösenord (som MD5 eller liknande).

Du kan gå till `http://consult.cern.ch/writeup/security/security\_3.html` för mer information om hur du väljer ett bra lösenord.

6.1 PGP och Public Key kryptografi

Public key kryptografi, så som den som används för PGP, handlar om kryptografi som använder en nyckel för kryptering och en annan nyckel för dekryptering. Traditionellt så har kryptografi använt samma nyckel för kryptering och dekryptering. Denna ”privata” nyckel måste vara känd av båda parter, och på något sätt transporteras från en till en annan på ett säkert sätt.

Public key kryptering upphäver behovet att säkert transportera nyckeln som behövs för dekryptering genom att använda två skilda nycklar, en publik nyckel och en privat nyckel. Varje persons publika nyckel är tillgänglig för vem som helst och används för krypteringen, samtidigt så har varje person sin privata nyckel som han kan använda för att dekryptera meddelanden som har krypterats med den rätta publika nyckeln.

Det finns fördelar både med Public Key och Private Key kryptering, och du kan läsa om dessa skillnader i RSA Cryptography FAQ, vars adress listas i slutet av denna sektion.

PGP (Pretty Good Privacy) stöds bra i Linux. Det är känt att versionerna 2.6.2 och 5.0 fungerar bra. För en bra grund om PGP och hur man använder det, titta på PGP FAQ. `http://www.pgp.com/service/export/faq/55faq.cgi` Se till att använda den versionen som passar för ditt land, eftersom exportrestriktioner av den amerikanska regeringen gör att stark kryptering anses vara ett militärt vapen och inte får föras ut ur landet i elektronisk form.

Det finns även en steg-för-steg guide för att konfigurera PGP på Linux på: `http://mercury.chem.pitt.edu/\~{}angel/LinuxFocus/English/November1997/article7.html` Den skrevs för den internationella versionen av PGP, men anpassas lätt till den amerikanska versionen. Du kan också behöva en patch för vissa av de senaste versionerna av Linux, som finns på `ftp://sunsite.unc.edu/pub/Linux/apps/crypto`.

Mer information om kryptografi hittar du i RSA Cryptography FAQ, som finns på `http://www.rsa.com/rsalabs/newfaq/`. Här hittar du information om termer som ”Diffie-Hellman”, ”public-key cryptography”, ”Digital Certificates”, osv.

6.2 SSL, S-HTTP, HTTPS och S/MIME

Ofta frågar användare om skillnaden mellan olika säkerhets- och krypteringsprotokoll, och hur man använder dem. Även om detta inte är ett dokument om kryptering, så är det en bra ide att ge en kort förklaring av varje, och var man hittar ytterligare information.

- **SSL** - SSL, eller Secure Sockets Layer, är en krypteringsmetod som är utvecklad av Netscape för att tillhadahålla säkerhet över Internet. Den stöder flera olika krypteringsprotokoll, och tillhandahåller klient- och serverautentisering. SSL opererar på transportlagret, skapar en säker krypterad kanal av data, och kan därför kryptera data av många olika slag. Detta visar sig vanligtvis när man går till en säker sajt för att titta på ett säkert dokument med Communicator, det agerar som bas för säker kommunikation med Communicator, såväl som för många andra program från Netscape Communications som använder kryptering. Mer information kan hittas på `http://www.consensus.com/security/ssl-talk-faq.html`. Information om Netscapes andra säkerhetsimplementationer, och en bra startpunkt för dessa protokoll hittar du på `http://home.netscape.com/info/security-doc.html`.
- **S-HTTP** - S-HTTP är ett annat protokoll som tillhandahåller säkerhetstjänster över Internet. Det designades för att tillhandahålla konfidentialitet, autentisering, integritet och icke-imitering [kan inte missuppfattas som någon annan] samtidigt som det stödjer hantering av multipla nycklar och krypteringsalgoritmer via överenskommelse mellan parterna som är inblandade i transaktionen. S-HTTP begränsas av den specifika mjukvara som implementerar det, och krypterar varje meddelande individuellt. [Från RSA Cryptography FAQ, sidan 138]
- **S/MIME** - S/MIME, eller Secure Multipurpose Internet Mail Extension, är en krypteringsstandard som används för att kryptera elektronisk post, eller andra typer av meddelanden på Internet. Det är en öppen standard som är utvecklad av RSA, så det är förhoppningsvis sannolikt att vi kommer att se det i Linux snart. Mer information om S/MIME hittar du på `http://home.netscape.com/assist/security/smime/overview.html`.

6.3 Linux x-kernel IPSEC implementering

Tillsammans med CIPE, och andra former av datakryptering, så finns det också en implementering av IPSEC för Linux. IPSEC är en ansträngning av IETF för att skapa en kryptografiskt säker kommunikation på IP nätverkslagret, som även tillhandahåller autentisering, integritett, accesskontroll och konfidentialitet. Information om IPSEC och ett Internet utkast hittar du på `http://www.ietf.org/html.charters/ipsec-charter.html`. Du kan även hitta länkar till andra protokoll som handlar om nyckelhantering, och en e-postlista och arkiv för IPSEC.

Linuximplementationen, som utvecklas vid University of Arizona, använder ett objektbaserat ramverk för att implementera nätverksprotokoll som heter x-kernel, och hittas på `http://www.cs.arizona.edu/xkernel/hpcc-blue/linux.html`. Stort sett så är x-kernel en metod att skicka meddelanden på kärnnivån, vilket leder till en enklare implementering.

Som med andra former av kryptografi, så distribueras den inte med kärnan som standard, på grund av exportrestriktioner.

6.4 SSH (Secure Shell), stelnet

SSH och stelnet är program som låter dig logga in på andra system och ha en krypterad anslutning.

SSH är en svit av program som används som en säker ersättare till rlogin, rsh och rcp. Den använder Public key kryptografi för att kryptera kommunikationen mellan två datorer, såväl som för användarautentisering. Detta kan användas för att på ett säkert sätt logga in till en annan dator eller kopiera data mellan datorer, samtidigt som det förhindrar man-mitt-i-mellan attacker (sessionskapning) och DNS ”spoofing”. Den utför datakompression på dina anslutningar, och säkra X11-kommunikationer mellan datorer. Hemsidan för SSH finns på `http://www.cs.hut.fi/ssh/`

Du kan också använda SSH från din Windows-arbetsstation till din Linux SSH-server. Det finns flera fritt tillgängliga implementationer av Windowsklienter, inlkusive den på `http://guardian.htu.tuwien.ac.at/therapy/ssh/` såväl som en komersiell implementation från DataFellows, på `http://www.datafellows.com`.

SSLeay är en fri implementation av Netscapes Secure Socket Layer protokoll, inklusive applikationer, som Secure telnet, en modul för Apache, flera databaser såväl som flera algoritmer inklusive DES, IDEA och Blowfish.

Med detta bibliotek har en säker telnet-ersättare skapats som krypterar över en telnetanslutning. Till skillnad från SSH så använder stelnet SSL, Secure Sockets Layer protokollet som utvecklats av Netscape. Du kan hitta Secure telnet och Secure FTP genom att börja med SSLeay FAQ, som finns på `http://www.psy.uq.oz.au/\~{}ftp/Crypto/`

6.5 PAM - Pluggable Authentication Modules

Nyare versioner av Linuxdistributionen RedHat levereras med ett enhetligt autentiseringssätt som heter ”PAM”. PAM låter dig ”on the fly” ändra dina autentiseringsmetoder, krav och kapsla in alla lokala autentiseringsmetoder utan att kompilera om någon av dina binärer. Konfigurationen av PAM ligger utanför detta dokuments räckvidd, men ta en titt på hemsidan för PAM för mer information. `http://www.kernel.org/pub/linux/libs/pam/index.html`

Bara några saker du kan göra med PAM:

- Använda en icke-DES kryptering för lösenord. (Vilket gör dem svårare att knäcka med ”råstyrka”)
- Sätta resursbegränsningar för dina användare så de inte kan utföra attacker som bygger på att de blir nekade en tjänst (antal processer, mängd minne, etc).

- Använda skuggade lösenord (se nedan) ”on the fly”.
- Tillåta vissa användare att endast logga in på vissa tider från vissa platser.

Inom några timmar av installation och konfiguration av ditt system, så kan du förhindra många attacker redan innan de inträffar. Till exempel, använd PAM för att stänga av den systemvida användningen av .rhosts filer i användarnas hemkataloger genom att lägga till dessa rader i /etc/pam.d/login:

```
#
# Disable rsh/rlogin/rexec for users
#
login auth required pam_rhosts_auth.so no_rhosts
```

6.6 Kryptografisk IP inkapsling (CIPE)

Huvudmålet med denna mjukvara är att tillhandahålla en facilitet för säker (mot ”eavesdropping”, inklusive trafikanalys och fejkad meddelandeinsättning) subnätverksanslutning över osäkra paketnätverk som Internet.

CIPE krypterar datan på nätverkslagret. Paket som färdas mellan datorer blir krypterade. Krypteringsmotorn placeras nära drivrutinen som skickar och tar emot paket.

Detta är olikt SSH, som krypterar data vid anslutningen på socketnivå. En logisk anslutning mellan program som kör på olika datorer är krypterad.

CIPE kan användas i tunnling för att skapa ett Virtuellt Privat Nätverk. Lågnivåkryptering har fördelen att den kan göras transparent mellan de två nätverken som är ihopkopplade i VPN, utan att ändra något i applikationsmjukvaran.

Sammanfattning från dokumentationen av CIPE:

Standarden för IPSEC definierar en mängd protokoll som kan användas (bland annat) till att bygga krypterade VPN. Men IPSEC är en ganska tungviktig och komplicerad protokollmängd med många valmöjligheter. Implementationer av det kompletta protokollet är fortfarande sällan använda och vissa aspekter (såsom nyckelhantering) är fortfarande inte helt lösta. CIPE använder en enklare metod, i vilken många saker som kan bli parametriserade (som valet av vilken krypteringsalgoritm som skall användas) är ett val som görs vid kompileringen. Detta begränsar flexibiliteten, men tillåter en enkel (och därför effektiv, lätt att debugga...) implemetation.

Mer information finns på `http://www.inka.de/\~{}bigred/devel/cipe.html`

Som med andra former av kryptografi, så distribueras den inte med kärnan som standard, på grund av exportrestriktioner.

6.7 Kerberos

Kerberos är ett autentiseringssystem som är utvecklat av Athena projektet på MIT. När en användare loggar in så autentiserar Kerberos den användaren (med lösenord) och ger användaren ett sätt att bevisa sin identitet till andra servrar och datorer som finns på nätverket.

Denna autentisering används sedan av program som rlogin för att låta användaren logga in på andra datorer utan lösenord (istället för filen .rhosts). Autentiseringen används även av e-postsystemet för att garantera att e-posten levereras till rätt person, såväl som att garantera att avsändaren är den han utger sig för.

Den övergripande effekten av att installera Kerberos och de flertaliga programmen som följer med är att nästan eliminera möjligheten för användare att lura systemet att tro att de är någon annan. Tyvärr så är det väldigt irriterande att installera Kerberos, eftersom det kräver att man modifierar eller ersätter ett antal standardprogram.

Du kan hitta mer information om Kerberos på `http://www.veritas.com/common/f/97042301.htm` och källkoden finns på `http://nii.isi.edu/info/kerberos/`

[Från: Stein, Jennifer G., Clifford Neuman, and Jeffrey L. Schiller. ”Kerberos: An Authentication Service for Open Network Systems.” USENIX Conference Proceedings, Dallas, Texas, Winter 1998.]

6.8 Skuggade lösenord

Skuggade lösenord är ett sätt att hålla den krypterade lösenordsinformationen hemlig för normala användare. Normalt så sparas det krypterade lösenordet i /etc/passwd, vilken alla kan läsa. De kan då köra program som gissar lösenord mot den filen och försöka få ut lösenorden i klartext. Skuggade lösenord sparar istället informationen i filen /etc/shadow, som bara priviligerade användare kan läsa. För att kunna använda skuggade lösenord så måste du se till att alla dina program som behöver ha tillgång till lösenordsinformation stödjer det. PAM (ovan) tillåter dock att du bara pluggar in en shadow-modul utan att du behöver kompilera om de exekverbara filerna. Titta i Shadow-Password HOWTO för mer information om det behövs. Den finns på `http://sunsite.unc.edu/LDP/HOWTO/Shadow-Password-HOWTO.html`. Den är ganska gammal nu, och behövs inte för distributioner som stödjer PAM.

6.9 Crack och John the Ripper

Om ditt passwd-program av någon anledning inte tvingar användarna att ha lösenord som är svåra att gissa, så kan det vara lämpligt att köra ett program som försöker knäcka lösenorden för att försäkra dig om att användarnas lösenord är säkra.

Program som knäcker lösenord baseras på en enkel ide. De försöker med alla ord i ordlistan och sedan med variationer av dessa ord. De krypterar var och ett och kollar det mot ditt krypterade lösenord. Om det stämmer så är de inne.

Det finns ett antal program som gör detta... av vilka två av de mest kända är ”Crack” och ”John the Ripper” `http://www.false.com/security/john/index.html`. De använder mycket av din CPU-tid, men du kan avgöra om en inkräktare kan använda programmen för att komma in genom att köra dem själv först, och meddela användare med för enkla lösenord. Observera att en inkräktare först måste hitta en säkerhetsläcka för att komma åt din passwd-fil (unix /etc/passwd), men dessa är vanligare än du kanske tror.

6.10 CFS - kryptografiskt filsystem och TCFS - transparent kryptografiskt filsystem

CFS är en metod att kryptera ett helt filsystem och låta användare lagra krypterade filer på dem. Det använder en NFS-server som kör på den lokala maskinen. Rpms finns på `http://www.replay.com/redhat/` och mer information om hur det hela fungerar finns på `ftp://ftp.research.att.com/dist/mab/`

TCFS förbättrar CFS genom att det är mer integrerat med filsystemet, så att det är transparent för alla användare av filsystemet att det är krypterat. Mer information på `http://edu-gw.dia.unisa.it/tcfs/`

6.11 X11-, SVGA- och displaysäkerhet

6.11.1 X11

Det är viktigt att du säkrar din grafiska display för att hindra attackerare från att göra saker som: fånga upp dina lösenord utan att du vet om det medans du skriver dem, läsa dokument eller information som du läser på din skärm eller till och med använda en läcka för att få superuser-access. Att köra X-applikationer över ett nätverk kan också vara farligt, detta kan ”sniffare” använda för att se all din interaktion med det andra systemet.

X har ett antal mekanismer för accesskontroll. Den enklaste av dem är datorbaserad. Du kan använda xhost för att specificera vilka datorer som har tillgång till din display. Detta är inte alls särskilt säkert. Om någon har tillgång till din maskin kan de köra ”xhost +sin_maskin” och lätt få tillträde. Dessutom, om du måste tillåta access till en opålitlig maskin så kan alla personer på den maskinen få tillgång till din display.

Om man använder xdm (x display manager) för att logga in, så får man en mycket bättre accessmetod: MIT-MAGIC-COOKIE-1. En 128bitars cookie genereras och lagras i din fil .Xauthority. Om du måste låta en annan dator få tillgång till din display så kan du använda kommandot xauth och informationen i .Xauthority för att ge access till endast den anslutningen. För mer information, se Remote-X-Apps mini-howto på `http://sunsite.unc.edu/LDP/HOWTO/mini/Remote-X-Apps.html`.

Du kan även använda ssh (se ovan) för att få säkra X-anslutningar. Detta har fördelen att det är transparent för slutanvändaren och betyder att ingen okrypterad data flödar över nätverket.

Ta en titt på manualbladet för Xsecurity för mer information om säkerhet i X. Det säkra valet är att använda xdm för att logga in till din konsoll och sedan använda ssh för att köra X-applikationer från andra datorer.

6.11.2 SVGA

SVGAlib-program är vanligtvis SUID-root för att kunna ha tillgång till Linuxmaskinens videohårdvara. Detta gör dem väldigt farliga. Om de kraschar, måste du antagligen starta om din maskin för att få tillbaka en användbar konsoll. Se till att alla SVGA-program som du kör är autentiska och åtminstonde är hyfsat pålitliga. Ännu bättre, kör dem inte alls.

6.11.3 GGI (Generella Grafikgränssnittsprojektet)

Linux GGI-projekt försöker lösa flera av problemen med grafikgränssnitten i Linux. GGI kommer att flytta in en liten del av grafikkoden in i Linuxkärnan, och sedan kontrollera access till grafiksystemet. Detta betyder att GGI kan återställa din konsoll till ett stabilt läge när som helst. `http://synergy.caltech.edu/\~{}ggi/`

7 Kärnans säkerhet

Detta är en beskrivning av de konfigurationsval i kärnan som relaterar till säkerhet, och en förklaring till vad de gör, och man använder dem.

Eftersom kärnan kontrollerar hur din dator agerar på nätverket, så är det viktigt att kärnan är väldigt säker och att des säkerhet inte äventyras. För att förhindra några av de senaste nätverksattackerna så skall du försöka hålla din kärna uppdaterad. Du hittar nya kärnor på `ftp://ftp.kernel.org`.

7.1 Kompileringsval för kärnan

- IP: Drop source routed frames (CONFIG_IP_NOSR) Detta val bör vara påslaget. Källroutade ramar innehåller hela vägen till sin destination inuti paketet. Detta betyder att routrar som paketet passerar inte behöver inspektera paketet utan bara skickar det vidare. Detta kan leda till att data som utgör ett potentiellt hot tar sig in i ditt system.
- IP: Firewalling (CONFIG_IP_FIREWALL) Detta val är nödvändigt om du skall konfigurera din maskin som en brandvägg, använda IP-förklädnad (masquerading) ellr vill skydda din arbetsstation från intrång via din uppringda PPP-länk.
- IP: forwarding/gatewaying (CONFIG_IP_FORWARD) Om du slår på IP forwarding, så blir din Linuxbox i princip en router. Om din maskin är på ett nätverk så kan du skicka vidare data från ett nätverk till ett annat, och kanske motarbeta en brandvägg som installerades just för att förhindra detta från att hända. Vanliga användare med uppringd förbindelse skall slå av detta val, och andra användare skall koncentrera sig på säkerhetsriskerna av att använda detta. Brandväggar skall ha detta påslaget och använda det tillsammans med mjukvara för brandväggar.

  Du kan slå på IP-forwarding dynamiskt med följande kommando:

  ```
  root#  echo 1 > /proc/sys/net/ipv4/ip_forward
  ```

  och slå av det med:

  ```
  root#  echo 0 > /proc/sys/net/ipv4/ip_forward
  ```

  Denna filen (och många andra i /proc) ser alltid ut att ha längden noll, men det har den egentligen inte. Detta är en nyintrodiucerad egenskap i kärnan, så se till att använda en kärna med version 2.0.33 eller senare.
- IP: firewall packet logging (CONFIG_IP_FIREWALL_VERBOSE) Detta val ger dig information om paket som din brandvägg tar emot, som sändare, mottagare, port, osv.
- IP: always defragment (CONFIG_IP_ALWAYS_DEFRAG) Generellt så är detta val avslaget, men om du bygger en brandvägg eller förklädd värd, så bör du slå på det. När data sänds från en dator till en annan, så sänds det inte alltid som ett enda paket med data, utan det fragmenteras till flera delar. Problemet med detta är att portnumret lagras endast i det första fragmentet. Detta innebär att någon kan lägga till information som inte skall vara där i de andra paketen för din ansluting.
- IP: syn cookies (CONFIG_SYN_COOKIES) SYN attack är en nekande-av-tjäns-attack (DoS) som konsumerar alla resurser på din maskin och tvingar dig att starta om. Vi kan inte komma på någon anledning för att du inte skulle slå på detta valet.
- Packet Signatures (CONFIG_NCPFS_PACKET_SIGNING) Detta valet finns i kärnor i serien 2.1 och det signerar NCP-paket för att öka säkerheten. Normalt kan du lämna det avslaget, men det finns att tillgå om du behöver det.
- IP: Firewall packet netlink device (CONFIG_IP_FIREWALL_NETLINK) Detta är ett väldigt trevlig val som låter dig analysera de första 128 byten av paketen i ett användar-program, för att avgöra om du vill acceptera eller neka paketet, baserat på dess validitet.

7.2 Enheter i kärnan

Det finns några block- och teckenenheter tillgängliga i Linux som kan hjälpa dig med säkerheten.

De två enheterna /dev/random och /dev/urandom tillhandahålls av kärnan för att kunna hämta slumpartad data när som helst.

Både /dev/random och /dev/urandom bör vara tillräckligt säkra för att generera PGP-nycklar, SSH-utmaningar och andra applikationer där säker slumpartade tal behövs. Attackerare bör vara oförmögna att förutsäga nästa tal som kommer från dessa källor oavsett vilken sekvens som har varit innan. Det har lagts ned mycken möda för att talen som du får från dessa källor verkligen är slumpvisa i alla bemärkelser.

Den enda skillnaden är att /dev/random får slut på slumpvisa bytes och låter dig vänta på att fler ackumuleras. Observera att på vissa system kan den blockera en lång tid i väntan på att en ny användargenererad post skall komma in i systemet. Så du måste vara försiktig innan du använder /dev/random. (Kanske är det bästa att använda detta när du genererar känslig nyckelinformation, och du säger till användaren att slå på tangentbordet upprepade gånger tills du skriver ut "OK, det räcker".)

/dev/random ger mycket bra slumpmässighet, genererad från att mäta tider mellan avbrott osv. Den blockerar tills tillräckligt många bitar med slumvis data finns tillgänglig.

/dev/urandom är liknande, men när dess lager med bra slumpvis data börjar ta slut, så returnerar den en kryptografiskt stark hash av det som finns. Detta är inte lika säkert men det duger för de flesta applikationer.

Du kan läsa från dessa enheter med något sånt här:

```
root#  head -c 6 /dev/urandom | uuencode -
```

Detta skriver ut sex slumvisa tecken på konsollen, lämpliga för lösenordsgenerering.

Titta i /usr/src/linux/drivers/char/random.c för en beskrivning av algoritmen.

Tack till Theodore Y. Ts'o, John Lewis och andra från Linux-kärnan för att ni hjälpte mig (Dave) med detta.

8 Nätverkssäkerhet

Nätverkssäkerhet blir allt viktigare eftersom personer tillbringar allt mer tid uppkopplade. Att äventyra nätverkssäkerhet är ofta mycket lättare än att göra det fysiskt eller lokalt, och det är mycket vanligare.

Det finns ett antal bra verktyg som kan hjälpa till med nätverkssäkerhet, och fler och fler av dem följer med i Linuxdistributioner.

8.1 Paketsniffare

Ett av de vanligaste sätten att få access till fler system på ditt nätverk är att plantera en paketsniffare på en redan erövrad dator. Denna "sniffare" ligger bara och lyssnar på ethernet-porten efter saker som "Password", "Login" och "su" på paketströmmen och sedan loggar trafiken efter det. På detta sätt kan inkräktare få lösenord till system som de inte ens försöker bryta sig in i. Lösenord i klartext är väldigt sårbara för denna typ av attacker.

EXEMPEL: dator A har blivit erövrad. En attackerare installerar en sniffare. Sniffaren loggar när administratören loggar in på dator B från dator C. Sniffaren får då administratörens personliga lösenord när han loggar in på B. Sedan gör administratören en "su" för att fixa ett problem. Nu har sniffaren root-lösenordet för dator B. Senare låter administratören någon telnet:a från hans konto till dator Z på en annan sajt. Nu har attackeraren lösenord och användarnamn på dator Z.

Nuförtiden så måste attackeraren inte ens erövra ett system för att göra detta, det räcker att han tar med sig en laptop in i din byggnad och pluggar in sig på ert nätverk.

Att använda ssh eller andra metoder med krypterade lösenord kväver sådana attacker. Saker som APOP för pop-konton förhindrar också sådana attacker. (Normala pop-inloggningar är väldigt sårbara för detta, såväl som allt annat som skickar löenord i klartext över tråden.)

8.2 Systemtjänster och tcp_wrappers

Så snart som du kopplar upp ditt Linuxsystem på NÅGOT nätverk så bör du fundera på vilka tjänster som du behöver tillhandahålla. Tjänster som du inte behöver tillhandahålla bör du stänga av så att du har en sak mindre att oroa dig för och attackerare har ett ställe mindre att leta efter hål på.

Det finns ett antal sätt att stänga av tjänster i Linux. Du kan titta på din fil /etc/inetd.conf och se vilka tjänster som erbjuds av din inetd. Stäng av alla som du inte behöver genom att kommentera bort dem (en # i början av raden), och sedan skickar du en SIGHUP till inetd-processen.

Du kan även ta bort (eller kommentera bort) tjänster i filen /etc/services. Detta innebär dock att lokala klienter inte kommer att hitta tjänsten (dvs om du tar bort ftp och försöker ftp:a till en annan sajt från den datorn så kommer det att misslyckas med ett meddelande om okänd tjänst). Det är vanligtvis inte värt besväret att ta bort tjänster från /etc/services eftersom det inte skapar någon ytterligare säkerhet. Om en person på det lokala systemet vill använda ftp även att du kommenterat bort det, så kan de göra en egen klient som använder den vanliga ftp-porten och det skulle fungera bra.

Vissa av tjänsterna som du antagligen vill ha kvar är:

- ftp
- telnet
- mail, såsom pop-3 eller imap
- identd
- time

Om du vet att du inte kommer använda ett visst paket så kan du även ta bort det helt. I RedHat tar ”rpm -e” bort ett helt paket. I Debian gör ”dpkg” samma sak.

Dessutom bör du stänga av tjänsterna rsh/rlogin/rcp, inklusive login (används av rlogin), shell (används av rcp) och exec (används av rsh), så att de inte startas av inetd. Dessa protokoll är extremt osäkra och har varit orsaken till attacker tidigare.

Du bör kolla din katalog /etc/rc.d/rcN.d/, där N är ditt systems runlevel, och se om någon av servrarna som startas där inte behövs. Filerna i /etc/rc.d/rcN.d/ är egentligen symboliska länkar till katalogen /etc/rc.d/init.d/. Att ändra namn på filerna i katalogen init.d ger samma effekt som att ta bort alla symboliska länkar i /etc/rc.d/rcN.d/. Om du endast vill stänga av en tjänst i en viss runlevel, ändra namn på den filen till att börja med ett litet ‘s’ istället för ett stort ‘S’, som i S45dhcpd.

Om du har rc-filer i BSD-stil, så skall du kolla i /etc/rc* efter program som du inte behöver.

De flesta Linuxdistributioner kommer med ”tcp_wrappers” som ”buntar ihop” alla dina tcp-tjänster. En tcp_wrapper (tcpd) startas från inetd istället för den riktiga servern. tcpd kollar då upp datorn som frågar efter tjänsten och startar sedan den riktiga tjänsten eller nekar access från den datorn. tcpd låter dig specificera restriktioner för access till dina tcp-tjänster. Du bör göra en fil /etc/hosts.allow och där skriva in de datorer som behöver ha access till tjänsterna på din maskin.

Om du är en hemanvändare med uppringd anslutning så föreslår vi att du nekar alla (ALL). tcpd loggar alla misslyckade försök att komma åt dina tjänster, så detta kan ge dig tips om att du är under attack. Om du lägger till nya TCP-baserade tjänster så bör du konfigurera även dem med tcp_wrappers. Till exempel, en vanlig användare med uppringd anslutning kan förhindra att andra ansluter till systemet, och fortfarande kunna hämta e-post och göra nätverksanslutningar till Internet. För att göra detta kan du lägga till följande till din /etc/hosts.allow:

ALL: 127.

Och såklart så skall /etc/hosts.deny innehålla:

ALL: ALL

detta förhindrar anslutningar till din maskin utifrån medans du fortfarande kan göra anslutingar till Internet inifrån.

8.3 Verifiera din DNS-information

Att hålla din DNS-information om alla datorer på ditt nätverk uppdaterad kan öka säkerheten. Ifall en otillåten dator ansluts till ditt nätverk så kan du upptäcka den genom att den inte har någon DNS-post. Många tjänster kan konfigureras för att inte acceptera anslutningar från datorer utan giltliga DNS-poster.

8.4 identd

identd är ett litet program som vanligtvis körs från din inetd. Det håller reda på vilka användare som kör vilka tcp-tjänster, och rapporterar det till den som önskar det.

Många personer misuppfattar hur användbart identd är och stänger därför av det eller blockerar alla förfrågningar till det från andra sajter. identd finns inte för att hjälpa andra sajter. Det finns inget sätt att veta om den data du får från en annan sajts identd är korrekt eller inte. Det är ingen autentisering i identd-förfrågningar.

Varför behöver du köra det då? Eftersom det hjälper _dig_, och det är ytterligare ett sätt att hålla koll på systemet. Om din identd inte är attackerad så vet du att den talar om användarnamn för personer som använder tcp-tjänster för andra sajter. Om administratören på en annan sajt talar om för dig att den och den användaren på ditt system försökte hacka sig in på deras sajt, så kan du lätt agera mot den användaren. Om du inte kör identd, så måste du titta i många loggar och lista ut vem som var påloggad vid den tidpunkten, och det tar generellt mycket längre tid att leta fram användaren.

Den identd som följer med de flesta distributioner går att konfigurera mer än många tror. Du kan stänga av identd för vissa användare (de kan skapa .noident filer), du kan logga alla förfrågningar till identd (jag rekommenderar detta), du kan låta identd returnera ett uid istället för ett användarnamn eller till och med NO-USER.

8.5 SATAN, ISS och andra Nätverksscannare

Det finns ett antal olika mjukvarupaket som scannar maskiner eller nätverk baserat på portar eller tjänster. SATAN och ISS är två av de mer kända. Denna mjukvara ansluter till målmaskinen (eller alla målmaskiner på ett nätverk), på alla portar den kan, och försöker avgöra vilken tjänst som kör på den porten. Baserat på denna information så kan du få reda på om maskinen är sårbar för en viss attack på den servern.

SATAN (Security Administrators Tool for Analyzing Networks) är en portscanner med ett webbgränssnitt. Det kan konfigureras för att göra lätt, medium eller starka kontroller på en maskin eller ett nätverk av maskiner. Det är en bra ide att hämta SATAN och scanna din maskin eller ditt nätverk och fixa till problemen som den hittar. Se till att hämta SATAN från sunsite eller en annan välkänd FTP eller webbsajt. Det fanns en trojansk kopia av SATAN som distribuerades över nätverket. `http://www.trouble.org/\~{}zen/satan/satan.html`

ISS (Internet Security Scanner) är en annan portbaserad scanner. Den är snabbare än SATAN, och kan därför vara bättre för stora nätverk. Men SATAN brukar ge mer information.

Abacus-Sentry är en komersiell portscanner från www.psionic.com. Titta på deras hemsida för mer information. `http://www.psionic.com`

Upptäcka portscanning.

Det finns några verktyg som är gjorda för att varna dig om att scanning pågår med SATAN, ISS eller annan liknande mjukvara. Men med liberal användning av tcp_wrappers och om du ser till att se över dina loggfiler ofta så skall du kunna upptäcka detta själv. Även med minimala inställningar så lämnar SATAN spår i loggarna på ett normalt RedHat-system.

8.6 Sendmail, qmail och MTAs

En av de viktigaste tjänsterna du kan ha är en e-postserver. Tyvärr så är en sådan även en av de mest sårbara mot attacker, på grund av antalet uppgifter som den måste utföra och de privilegier som den normalt behöver.

Om du använder sendmail så är det väldigt viktigt att du håller det uppdaterat. Sendmail har en lång historia av säkerhetsläckor. Var alltid noga med att köra den senaste versionen. `http://www.sendmail.org`

Om du är trött på att uppgradera din version av sendmail varje vecka så kan du fundera på att byta till qmail. qmail designades med säkerhet i åtanke från början. Det är snabbt, stabilt och säkert. `http://www.qmail.org`

8.7 Nekande-av-tjänst attacker

En nekande-av-tjänst attack är när en attackerare försöker göra en resurs för upptagen för att svara på riktiga förfrågningar, eller neka tillåtna användare tillgång till din maskin.

Sådana attacker har ökat mycket de senaste åren. Några av de mest populära listas nedan. Notera att ny dyker upp hela tiden, så detta är bara några exempel. Läs säkerhetlistornaför Linux, bugtraq-listan och arkiv för mer uppdaterad information.

- **SYN Flooding** - SYN flooding är en nekande-av-tjänst attack på nätverksnivå. Det drar nytta av ett ”kryphål” i sättet som en TCP-anslutning upprättas. Nyare Linuxkärnor (2.0.30 och nyare) har flera valmöjligheter för att förhindra SYN-attacker från att neka personer access till din maskin eller dess tjänster. Se avsnittet om säkerhet i kärnan för mer information om kompileringsalternativen.
- **Pentium ”F00F” bugg** - Det upptäcktes nyligen att om en viss serie av assembler instruktioner skickades till en äkta Intel Pentiumprocessor så skulle maskinen låsas. Detta berör alla maskiner med en Pentiumprocessor (inte kloner, inte Pentium Pro eller II), oavsett vilket operativsystem den kör. Linuxkärnor från 2.0.32 och uppåt innehåller en fix för denna buggen som förhindrar att den låser din maskin. Kernel 2.0.33 har en förbättrad version av fixen och föredras framför 2.0.32. Om du kör på en Pentium så bör du uppgradera nu!
- **Ping Flooding** - Ping flooding är en enkel råstyrkeattack. Attackeraren skickar en ”flod” av ICMP-paket till din maskin. Om de gör detta från en maskin med bättre bandbredd än din, så kommer din maskin inte att kunna skicka något på nätverket. En variant av denna attacken, som kallas ”smurfing”, skickar ICMP-paket till en annan dator med _din_ maskins IP-adress som returadress, vilket låter dem flooda dig lite mer diskret. Du kan hitta information om ”smurf”-attacken på `http://www.quadrunner.com/\~{}chuegen/smurf.txt` Om du någonsin finner dig själv vara utsatt för en ping flood attack, använd ett verktyg som tcpdump för att ta reda på var paketen kommer ifrån (eller var de ser ut att komma ifrån), kontakta sedan din leverantör med denna informationen. Ping flood kan ganska lätt stoppas på routernivå eller genom att använda en brandvägg.
- **Ping o' Death** - Ping o' Death attacker är ett resultat av att inkommande ”ICMP ECHO REQUEST”-paket är större än vad datastrukturerna i kärnan som skall lagra informationen klarar av att lagra. Eftersom att skicka ett enda stort (65510 bytes) pingpaket till många system får dem att hänga sig

eller krasha, så fick detta problemet snabbt namnet Ping o' Death. Detta är åtgärdat sedan länge och man behöver inte oroa sig för detta längre.

- **/Teardrop / New Tear** - En av de senaste attackerna använder sig av en bugg som finns i IP-fragmenteringskoden på Linux- och Windowssystem. Det är fixat i version 2.0.33 av kärnan och man behöver inte välja några speciella kompileringsalternativ för att fixen skall användas. Linux är inte sårbar för attacken "newtear".

Du kan hitta koden för de flesta attacker, och djupare information om hur de fungerar på `http://www.rootshell.com` genom att använda deras sökmotor.

8.8 NFS-säkerhet (Network File System)

NFS är ett protokoll för att dela filer som används mycket. Det låter servrar som kör nfsd och mountd exportera hela filsystem till andra maskiner som har stöd för nfs i kärnan (eller någon annan form av klientstöd om det inte är Linuxmaskiner). Mountd håller reda på monterade filsystem i /etc/mtab, och kan lista dem med "showmount".

Många sajter använder NFS för att hålla hemkataloger för användare, så att oavsett vilken maskin i nätverket som de loggar in på så har de alltid tillgång till sina egna filer.

Det finns ett litet mått av säkerhet när man exporterar filsystem. Du kan låta nfsd mappa den andra root-användaren (uid=0) till användaren nobody, vilket nekar dem total access till filerna som exporteras. Men eftersom individuella användare har tillgång till sina egna (eller åtminstonde med samma uid) filer, så kan superusern på den andra maskinen logga in som eller su:a till en annan användare och ha total access till deras filer. Detta är endast ett litet hinder för en inkräktare som har möjlighet att montera dina filsystem.

Om du måste använda NFS, se till att du endast exporterar till de maskiner som du måste exportera till. Exportera aldrig hela din rootkatalog, exportera endast de kataloger som du måste exportera.

För mer information om NFS se: `NFSHOWTO`

8.9 NIS (Network Information Service) (tidigare YP).

NIS är ett sätt att distribuera information till en grupp av maskiner. NIS-mastern har informationstabeller och konverterar dem till NIS tabellfiler. Dessa tabellfiler skickas sedan över nätverket och låter NIS-klienter att erhålla användarnamn, lösenord, hemkatalog och shellinformation (all information i en standard /etc/passwd). Detta låter användare ändra sitt lösenord en gång och det får effekt på alla maskiner i NIS-domänen.

NIS är inte alls säkert. Det var aldrig meningen att det skulle vara det. Det var meningen att det skulle vara användbart och smidigt. Vem som helst som kan gissa namnet på din NIS-domän (varsomhelst på nätet) kan få en kopia av din passwd-fil, och använda crack och john the ripper för att försöka ta reda på användarnas lösenord. Det är också möjligt att knäcka NIS och göra alla möjliga elaka trix. Om du måste använda NIS, försäkra dig att du vet om alla faror.

Det finns en mycket säkrare ersättare till NIS som heter NIS+. Kolla in NIS HOWTO för mer information: `http://sunsite.unc.edu/mdw/HOWTO/NIS-HOWTO.html`

8.10 Brandväggar

Brandväggar är ett sätt att begränsa vilken information som tillåts komma in i eller komma ut ifrån ditt lokala nätverk. Vanligtvis så är brandväggsdatorn ansluten till Internet och till ditt lokala nätverk och det enda sättet att komma i kontakt med ditt lokala nätverk är genom brandväggen. På detta sätt kan brandväggen kontrollera vad som skickas fram och tillbaka mellan Internet och ditt lokala nätverk.

Det finns ett antal olika metoder att sätta upp brandväggar. Linuxmaskiner är hyfsat bra lågkostnadsalternativ. Kod för brandväggar kan kompileras in direkt i kärnan från version 2.0 och uppåt. Med användarverktyget ipfwadm kan du, medans systemet är igång, ändra vilken typ av nätverkstrafik som du vill tillåta. Du kan även logga viss typ av information.

Brandväggar är en väldigt användbar och viktig teknik för att säkra ditt nätverk. Det är viktigt att inse att bara för att du har en brandvägg så betyder det inte att du inte måste säkra maskinenrna bakom brandväggen. Detta är ett grovt misstag. Titta i den utomordentliga Firewall HOWTOn på sunsite för mer information om brandväggar och Linux. `http://sunsite.unc.edu/mdw/HOWTO/Firewall-HOWTO.html`

Mer information hittar du även i IP-Masquerade mini-howto: `http://sunsite.unc.edu/mdw/HOWTO/mini/IP-Masquerade.html`

Mer information om ipfwadm (verktyget som låter dig ändra inställningarna för din brandvägg) hittar du på dess hemsida: `http://www.xos.nl/linux/ipfwadm/`

9 Säkerhetsförberedelser (innan du kopplar upp dig)

Okej, nu har du sett över ditt system och sett till att det är så säkert som möjligt och är redo att koppla upp det. Det finns några saker som du bör göra nu för att vara förberedd ifall du verkligen blir attackerad, så att du snabbt kan stoppa inkräktaren och få igång systemet igen.

9.1 Gör en fullständig säkerhetskopia av din maskin

Diskussion om metoder för säkerhetskopiering ligger utanför räckvidden av detta dokument, men några ord som relaterar till säkerhetskopiering och säkerhet:

Om du har mindre än 650Mb data som skall lagras, så är det bra med en kopia på CD-R (eftersom det är svårt att mixtra med senare, och om det lagras bra så håller det länge). Band och andra media som kan skrivas över bör skrivskyddas så snart säkerhetskopieringen är färdig och verifierad för att förindra att någon mixtrar med den. Se till att förvara alla dina säkerhetskopior i ett säkert utrymme som inte är uppkopplat. Med en bra säkerhetskopia försäkrar du dig om att du har ett bra tillstånd att återskapa ditt system ifrån.

9.2 Välj ett bra schema för säkerhetskopiering

En cykel med sex band är lätt att underhålla. Detta innebär fyra band under veckan, ett band på jämna fredagar och ett band på udda fredagar. Gör en kopia på förändrad data varje dag (incremental backup) och en fullständig kopia varje fredag. Om du gör någon särskilt viktig ändring eller lägger till viktig data till ditt system så kan en säkerhetskopia vara bra att göra.

9.3 Säkerhetskopiera din RPM eller Debian fildatabas

I händelse av en attack så kan du använda din RPM-databas som du skulle använda tripwire, men bara om du kan vara säker på att den inte har blivit ändrad. Du bör kopiera RPM-databasen till en diskett och hålla den på ett säkert ställe hela tiden. Debian har säkerligen någonting liknande.

Speciellt, filerna /var/lib/rpm/fileindex.rpm och /var/lib/rpm/packages.rpm får antagligen inte plats på en enda diskett. Men komprimerade får de säkert plats på varsin diskett.

Om ditt system har blivit attackerat kan du använda kommandot:

```
root# rpm -Va
```

för att verifiera alla filer på ditt system. Titta på manualbladet för RPM, eftersom det finns några andra parametrar som du kan använda för att få mindre utskrifter.

Detta innebär att varje gång en ny RPM läggs till i systemet så måste RPM-databasen arkiveras om. Du måste väga fördelarna mot nackdelarna.

9.4 Håll reda på systemloggarna

Det är väldigt viktigt att ingen har mixtrat med informationen som kommer ifrån syslog. Att göra filerna i /var/log läs- och skrivbara av endast ett begränsat antal användare är en bra början.

Håll ett öga på vad som skrivs där, särskilt under 'auth'. Ett flertal misslyckade inloggningsförsök, till exempel, kan tyda på ett försök till en attack.

Var du skall leta efter dina loggfiler beror på vilken distribution du har. I en distribution som följer "Linux Filsystemstandard", som tex RedHat, så hittar du dem i /var/log.

Du kan lista ut var ditt system har loggfilerna genom att titta i filen /etc/syslog.conf. I denna filen specificeras var syslog skall logga de olika meddelandena.

Du kan också konfigurera ditt skript eller din daemon för loggrotation så att loggarna finns kvar tillräckligt länge för att du skall hinna läsa dem. Titta på paketet "logrotate" i nyare RedHat-distributioner. Andra distributioner har antagligen liknande paket.

Om någon har mixtrat med dina loggfiler så försök att lista ut när mixtrandet började, och vilka saker som de verkade mixtra med. Finns det långa tidsperioder som inte har loggats? Att kolla efter korrekta systemloggar på band med säkerhetskopior är en bra ide.

Loggfiler ändras typiskt av inkräktare som vill sopa igen sina spår, men de bör ändå sökas igenom efter konstiga händelser. Du kan till exempel upptäcka en inkräktare som försöker få tillgång till systemet, eller upptäcka ett program som försöker komma över root-användaren. Du kanske hinner se loggfilerna innan inkräktaren hinner ändra i dem.

Du bör även se till att skilja 'auth'-faciliteten från annan loggdata, inklusive försök att använda 'su', inloggningsförsök och annan data om användare.

Om möjligt så bör du konfigurera syslog att skicka en kopia av den viktigaste datan till ett säkert system. Detta hindrar en inkräktare från att sopa igen sina spår genom att radera sina login/su/ftp/osv försök. Se manualbladet för syslog.conf och titta på avsnittet om parametern '@'.

Till sist, loggfiler är till liten nytta om ingen läser dem. Ta dig tid lite då och då för att titta igenom dina loggfiler, och skaffa dig en känsla av hur de ser ut en normal dag. Att veta detta kan hjälpa dig att upptäcka ovanliga saker.

9.5 Lägg till alla nya systemuppdateringar

De flesta Linuxanvändare installerar från en CDROM. På grund av den av naturen snabba takten på säkerhetsfixar så finns det alltid nya (fixade) program att tillgå. Innan du ansluter din maskin till nätverket så är det en bra ide att kolla på webbsajten för din distribution (till exempel ftp.redhat.com) och skaffa alla uppdaterade paket sedan du fick din CDROM. Dessa paket innehåller ofta viktiga säkerhetsfixar så det är en bra ide att installera dem.

10 Att göra efter en attack

Så du har följt några av råden här (eller någon annanstans) och har upptäckt en attack? Det första du skall göra är att vara lugn. Förhastade ageranden kan göra mer skada än vad inkräktaren skulle gjort.

10.1 Säkerhetsbrott på gång

Att upptäcka ett pågående säkerhetsbrott kan vara en känslig sak. Hur du reagerar kan få allvarliga konsekvenser.

Om brottet du ser är fysiskt så är det troligt att du har sett någon som har brytit sig in i ditt hem, kontor eller labb. Du bör då kontakta lokala myndigheter. I en labbmiljö kanske du har sett någon som försökt öppna en låda eller starta om en maskin. Beroende på din auktoritet och procedurer så kanske du kan be dem att sluta eller kontakta lokal säkerhetspersonal.

Om du har upptäckt en lokal användare som försöker bryta sig igenom säkerheten, så är det första du skall göra att bekräfta att de verkligen är de du tror att de är. Kolla sajten som de loggar in ifrån. Är det den sajten som de normalt loggar in ifrån? nej? Använd då ett icke elektroniskt sätt att komma i kontakt. Till exempel, ring dem per telefon eller gå till deras kontor/hus och prata med dem. Om de erkänner att de är påloggade så kan du be dem förklara vad de sysslar med eller be dem sluta med det. Om de inte är påloggade, och inte har en aning om vad du pratar om, så är chansen stor att incidenten kräver ytterligare undersökning. Undersök sådana incidenter, och samla på dig mycket information innan du anklagar någon.

Om du upptäcker ett säkerhetsbrott via nätverket, så är det första du skall göra (om du kan) att koppla ner ditt nätverk. Om de är anslutna via ett modem, dra ur modemsladden, om de är anslutna via ethernet, dra ur ethernetsladden. Detta hindrar dem från att göra ytterligare skada, och de antar antagligen att de har råkat ut för ett nätverksproblem snarare än att de blivit upptäckta.

Om du inte kan koppla ner nätverket (om du har en välanvänd sajt eller inte har fysisk kontroll över dina maskiner), så är det näst bästa sättet att använda program som tcp_wrappers eller ipfwadm för att neka access från inkräktarens sajt.

Om du inte kan neka alla personer från inkräktarens sajt, så får det duga med att låsa användarens användarkonto. Observera att det är inte lätt att låsa ett användarkonto. Du måste tänka på .rhosts-filer, FTP-access och en dator med bakdörrar.

När du gjort något av ovanstående (kopplat ner nätverket, nekat access från deras sajt eller låst användarkontot) så måste du döda alla deras användarprocesser och logga av dem.

Du bör kontrollera din sajt noga under de följande minuterna, eftersom attackeraren kommer att försöka komma in igen. Kanske genom att använda ett annat användarnamn och/eller från en annan nätverksadress.

10.2 Säkerhetsbrott har redan inträffat

Du har antingen upptäckt ett brott som redan inträffat eller du har upptäckt ett pågående brott och (förhoppningsvis) låst ute inkräktaren från ditt system. Sen då?

10.2.1 Stänga hålet

Om du kan lista ut på vilket sätt inkräktaren lyckades ta sig in i ditt system så bör du försöka stänga det hålet. Till exempel så kanske du ser ett flertal FTP-poster precis innan användaren loggade in. Stäng då av FTP-tjänsten och kolla om du kan hitta en uppdaterad version eller om någon av e-postlistorna vet någon fix.

Kolla alla dina loggfiler och kolla alla e-postlistor och webbsajter som handlar om säkerhet för att se om det finns några nya vanligt förekommande attacker som du kan fixa. Du kan hitta säkerhetsfixar för Caldera här: `http://www.caldera.com/tech-ref/security/`. RedHat har ännu inte delat upp säkerhetsfixar från buggfixar, men deras errata för distributionen finns på `http://www.redhat.com/errata`. Det är troligt att om en distributör har släppt en säkerhetsfix så har de flesta andra gjort det också.

Om du inte låser ute inkräktaren så kommer han antagligen tillbaka. Inte bara på din maskin utan någonstans på nätverket. Om han körde en paketsniffare så är risken stor att han har tillgång till andra lokala maskiner.

10.2.2 Ta reda på hur mycket skada som har skett

Det första att ta reda på är hur mycket skada som har skett. Vad har blivit attackerat? Om du kör en integritetskontroll som tripwire så kan du göra en körning och få reda på det. Om inte så får du titta runt bland dina viktiga data.

Eftersom Linux blir lättare och lättare att installera, så kan du överväga att spara dina konfigurationsfiler och sedan radera dina diskar och sedan installera om. Efter detta kan du återskapa användarfiler och konfigurationsfiler från säkerhetskopior. På detta sätt försäkrar du dig om att du har ett rent system. Om du måste säkerhetskopiera filer från ett attackerat system så skall du vara extra försiktig med att återskapa binärfiler, då de kan vara trojanska hästar som inkräktaren har placerat där.

10.2.3 Säkerhetskopior, Säkerhetskopior, Säkerhetskopior!

Att göra regelbundna säkerhetskopior är ovärderligt ur säkerhetssynpunkt. Om ditt system blir attackerat så kan du återskapa den data du behöver från säkerhetskopior. Självklart kan viss data även vara värdefull för attackeraren, och de kommer inte bara att förstöra den utan även stjäla den så att de har sina egna kopior, men du kommer åtminstonde att ha kvar den.

Du bör kolla flera säkerhetskopior bakåt i tiden innan du återskapar en fil som någon har mixtrat med. Inkräktaren kan ha blivit förändrade för länge sedan och du kan ha gjort flera lyckade säkerhetskopior av den filen.

Självklart så måste man tänka på säkerheten kring säkerhetskopiorna också. Se till att lagra dem på ett säkert ställe. Håll reda på vem som har tillgång till dem. (Om en inkräktare kan få tag på dina säkerhetskopior så kan han få tillgång till alla dina data utan att du någonsin får reda på det.)

10.2.4 Spåra inkräktaren

Okej, nu har du låst ute inkräktaren och återskapat ditt system, men du är inte riktigt färdig än. Även om det inte är troligt att inkräktaren åker fast så bör du rapportera attacken.

Du bör rapportera attacken till administratören på den sajt som inkräktaren loggade in ifrån. Du kan ta reda på vem administratören är med kommandot ”whois” eller genom internic-databasen. Du kan skicka ett e-postmeddelande som innehåller alla nödvändiga loggposter, datum och tider. Om du har upptäckt något annat speciellt med din inkräktare så kan du nämna det också. Efter att du har skickat e-post så bör du (om du har lust) följa upp med ett telefonsamtal. Om administratören i sin tur upptäcker inkräktaren så kan han försöka kontakta administratören på nästa sajt, osv.

Duktiga crackers använder ofta flera mellanliggande system. Vissa (eller flera) som inte ens vet om att de blivit attackerade. Att försöka spåra en cracker till sin hemmasajt kan vara svårt. Att vara artig mot administratörerna som du talar med kan hjälpa mycket.

Du bör även meddela alla säkerhetsorganisationer som du är med i (CERT eller liknande).

11 Säkerhetskällor

Det finns MÅNGA bra sajter på nätet som handlar om säkerhet i UNIX i allmänhet och Linux i synnerhet. Det är väldigt viktigt att prenumerera på en (eller flera) av e-postlistorna om säkerhet och hålla sig uppdaterad på säkerhetsfixar. De flesta av dessa listor har väldigt låg omsättning på meddelande, men är väldigt informativa.

11.1 FTP-sajter

CERT är Computer Emergency Response Team. De skickar ofta ut meddelanden om nya attacker och fixar. `cert.org`

Replay har arkiv med många säkerhetsprogram. Eftersom de är utanför USA så måste de inte lyda amerikanska lagar om kryptering. `replay.com`

Matt Blaze är författaren av CFS och en utomordentlig säkerhetsförespråkare. `MattBlaze'sstuff`

tue.nl är en mycket bra säkerhetssajt i Nederländerna. `ftp.win.tue.nl`

11.2 Webb-sajter

Hacker FAQ är en FAQ om hackers: `TheHackerFAQ`

COAST-arkivet har ett stort antal säkerhetsprogram för UNIX och mycket information: `COAST`

Rootshell.com är en bra sajt för att se vilka attacker som för närvarande används av crackers: `rootshell.comexploits`

BUGTRAQ lägger ut råd i säkerhetsfrågor: `BUGTRAQarchives`

CERT, Computer Emergency Response Team, lägger ut råd om vanliga attacker på UNIX-plattformar: `CERThome`

Dan Farmer har skrivit SATAN och många andra säkerhetsverktyg, hans hemsida innehåller intressant information om säkerhet såväl som säkerhetsverktyg: `DanFarmerstrouble.org`

Linux Security WWW är en bra sajt för information om säkerhet i Linux: `LinuxSecurityWWW`

Reptile har mycket bra säkerhetsinformation om Linux på sin sajt: `ReptilesLinuxSecurityPage`

Infilsec har en ”sårbarhetsmotor” som kan tala om på vilket sätt en viss plattform är sårbar: `Infilsecvunerabilityengine`

CIAC skickar periodvis ut information om vanliga attacker: `CIACbulitins`

En bra startpunkt för Linux Pluggable Authentication Modules hittar du på `http://www.kernel.org/pub/linux/libs/pam/`.

11.3 E-postlistor

Bugtraq: För att prenumerera på bugtraq så skall du skicka e-post till listserv@netspace.org och i meddelandekroppen skriver du ”subscribe bugtraq”. (för arkiv se länken ovan)

CIAC Skicka e-post till majordomo@tholia.llnl.gov och i meddelandekroppen skriver du ”subscribe ciac-bulletin”

11.4 Böcker - Tryckt Läsmaterial

Det finns ett antal bra böcker om säkerhet. Detta avsnittet listar några av dem. Utöver särskilda böcker om säkerhet, så behandlas säkerhet i ett antal andra böcker om systemadministration.

Building Internet Firewalls Av D. Brent Chapman & Elizabeth D. Zwicky

1st Edition September 1995

ISBN: 1-56592-124-0

Practical UNIX & Internet Security, 2nd Edition Av Simson Garfinkel & Gene Spafford

2nd Edition April 1996

ISBN: 1-56592-148-8

Computer Security Basics Av Deborah Russell & G.T. Gangemi, Sr.

1st Edition July 1991

ISBN: 0-937175-71-4

Linux Network Administrator's Guide Av Olaf Kirch

1st Edition January 1995

ISBN: 1-56592-087-2

PGP: Pretty Good Privacy Av Simson Garfinkel

1st Edition December 1994

ISBN: 1-56592-098-8

Computer Crime A Crimefighter's Handbook Av David Icove, Karl Seger & William VonStorch (Consulting Editor Eugene H. Spafford)

1st Edition August 1995

ISBN: 1-56592-086-4

11.5 Terminologi

- **Brandvägg -** En komponent eller mängd av komponenter som begränsar tillgång mellan ett skyddat nätverk och Internet, eller mellan andra nätverk.
- **Paket -** Den fundamentala enheten för kommunikation över Internet.
- **Paketfiltrering -** Agerandet från en enhet som selektivt kontrollerar flödet av data till och från nätverket. Paketfilter släpper igenom eller blockerar paket, vanligtvis när de routas från ett nätverk till ett annat (oftast från Internet till ett internt nätverk och vice versa). För att åstadkomma paketfiltrering så specificerar du regler som bestämmer vilka typer av paket (de från en viss IP-adress eller port) som skall tillåtas och vilka som skall blockas.
- **Gränsnätverk -** Ett nätverk som installeras mellan ett skyddat nätverk och ett externt nätverk för att åstadkomma ytterligare ett lager av säkerhet. Kallas ibland för DMZ.
- **Proxyserver -** Ett program som tar hand om externa servrar å interna klienters vägnar. Proxyklienter pratar med proxyservrar, som skickar vidare godkända klientförfrågningar till riktiga servrar, och skickar vidare svar tillbaka till klienterna.

- **Nekande av tjänst -** En nekande av tjänst attack är när en attackerare konsumerar din maskins resurser på ett sätt som det inte var meningen att de skulle konsumeras, vilket förhindrar normal användning av dina nätverksresurser för legitima avsikter.
- **Buffer Overflow -** Ett vanligt sätt att programmera är att aldrig allokera buffertar som är ”tillräckligt stora” och inte kontrollera om det blir overflow. När sådana buffertar får overflow så kan det exekverande programmet (daemon eller suid-program) luras till att göra andra saker. Generellt så görs det genom att skriva över en funktions returadress på stacken så att den pekar till ett annat ställe.
- **IP Spoofing -** IP-spoofing är en komplex teknisk attack som utgörs av flera komponenter. Det är en säkerhetsattack som lurar datorer till att tro att du är någon som du egentligen inte är. Det finns en bra artikel om detta skriven av daemon9, route och infinity i Phrack Magazine, Utgåva fyrtioåtta, volym sju.
- **Autentisering -** Egenskapen av att veta att datan som togs emot är samma data som sändes och att sändaren är den han utger sig för att vara.
- **Non-repudiation -** Egenskapen av att en mottagare kan bevisa att sändaren av viss data verkligen skickade datan även om sändaren senare förnekar att han gjort detta.

12 Ofta frågade frågor (FAQ)

1. Är det säkrare att kompilera in drivrutiner direkt i kärnan än att göra dem till moduler? Svar: Vissa personer tycker att det är bättre att stänga av förmågan att ladda drivrutiner genom moduler, eftersom en inkräktare skulle kunna ladda en trojansk modul som kan påverka säkerheten i systemet.

   Men för att kunna ladda moduler måste du vara root. För att kunna ändra objektfiler för moduler måste du också vara root. Detta betyder att inkräktaren måste ha root-access för att kunna ladda en modul. Och om inkräktaren har root-access så finns det allvarligare saker att oroa sig för än att han skall ladda moduler.

   Moduler är till för att dynamiskt kunna ladda in stöd för vissa enheter som sällan används. På servermaskiner, eller brandväggar till exempel, så behövs detta med all sannolikhet inte. Därför verkar det vettigare att kompilera in drivrutinerna direkt i kärnan för maskiner som används som servrar. Moduler är dessutom långsammare än inkompilerat stöd.

2. Inloggning som root från andra maskiner misslyckas alltid. Svar: Se avsnittet om root-säkerhet. Detta görs med avsikt för att förhindra externa användare att logga in till din maskin som root, vilket skulle vara ett allvarligt säkerhetsproblem. Glöm inte att potentiella inkräktare har tiden på sin sida och kan köra program för att hitta ditt lösenord.

3. Hur slår jag på skuggade lösenord i mitt RedHat 4.2 eller 5.0 Linuxsystem? Svar: Skuggade lösenord är en mekanism för att lagra dina lösenord i en annan fil än den normala /etc/paswd. Detta har flera fördelar. Den första är att skuggfilen /etc/shadow endast är läsbar av root, till skillnad från /etc/passwd som måste vara läsbar av alla. En annan fördel är att som administratör kan du sätta på och stänga av användarkonton utan att alla vet statusen för andras användarkonton.

   Filen /etc/passwd används då för att lagra användarnamn och gruppnamn som används av program som ”/bin/ls” för att mappa användarid till korrekt användarnamn i en kataloglistning.

   Filen /etc/shadow innehåller endast användarnamn med respektive lösenord och kanske bokföringsinformation, som när användarkontot slutar gälla osv.

   För att slå på skuggade lösenord, kör 'pwconv' som root och /etc/shadow bör skapas och användas av applikationer. Eftersom du kör RH 4.2 eller nyare så skall PAM-modulerna anpassa sig automatiskt till att använda /etc/shadow istället.

Eftersom du är intresserad av att säkra dina lösenord så kanske du är intresserad av att generera bra lösenord från början. För detta kan du använda modulen 'pam_cracklib' som är en del av PAM. Den kör lösenord mot Crack-biblioteken för att hjälpa dig att se om de är för lätta för crack-program att gissa.

4. Hur slår jag på SSL-utökningen i Apache? Svar:

   1. Hämta SSLeay 0.8.0 eller senare från ftp://ftp.psy.uq.oz.au/pub/Crypto/SSL

   2. Kompilera, testa och installera det.

   3. Hämta källkoden för Apache 1.2.5 eller senare.

   4. Hämta Apache SSLeay-utökningen på *denna sajten* `<ftp://ftp.ox.ac.uk/pub/crypto/SSL/apache_1.2.5+ssl_1.13.tar.gz>`

   5. Packa upp det i källkodskatalogen för apache-1.2.5 och patcha Apache som det står i README.

   6. Konfigurera och kompilera det.

   Du kan även prova `ReplayAssociates` som har många färdigkompilerade paket och som är lokaliserat utanför USA.

5. Hur kan jag manipulera användarkonton och samtidigt hålla hög säkerhet? Svar: RedHat-distributionen, särskilt RH5.0, innehåller ett stort antal verktyg för att kunna ändra användarkontonas egenskaper.

   - Programmen pwconv och unpwconv kan användas för att växla mellan skuggade och vanliga lösenord.
   - Programmen pwck och grpck kan anvädas för att verifiera organisationen av passwd och group filer.
   - Programmen useradd, usermod och userdel kan användas för att lägga till, ta bort och ändra användarkonton. Programmen groupadd, groupmod och groupdel gör samma sak för grupper.
   - Grupplösenord kan skapas med gpasswd.

   Alla dessa program är "skugg-medvetna" – dvs om du använder skuggning så använder de /etc/shadow, annars gör de det inte.

   Se respektive manualblad för vidare information.

6. Hur kan jag lösenordsskydda vissa HTML-dokument med Apache? Jag slår vad om att du inte känner till `http://www.apacheweek.org`, eller?

   Du hittar information om användarautentisering på `http://www.apacheweek.com/features/userauth` såväl som andra säkerhetstips för webbservrar på `http://www.apache.org/docs/misc/security\_tips.html`

13 Slutsats

Genom att prenumerera på e-postlistorna med säkerhetslarm, och genom att hålla dig uppdaterad så kan du höja säkerheten nämnvärt i ditt system. Om du håller reda på dina loggfiler och kör program som tripwire regelbundet så kan du höja den ännu mer.

Det är inte svårt att hålla datorsäkerheten på en hyfsad nivå på en hemmadator. Det krävs mera jobb för företagsmaskiner, men Linux kan vara en säker plattform. På grund av sättet på vilket Linux utvecklas så kommer säkerhetsfixar oftast ut mycket snabbare än för kommersiella operativsystem, vilket gör Linux till en utomordentlig plattform när det är viktigt med säkerhet.

14 Tack till

Information i detta dokument har samlats ihop från många källor. Tack till följande som har bidragit antingen direkt eller indirekt:

Rob Riggs <rob@DevilsThumb.com>

S. Coffin <scoffin@netcom.com>

Viktor Przebinda <viktor@CRYSTAL.MATH.ou.edu>

Roelof Osinga <roelof@eboa.com>

Kyle Hasselbacher <kyle@carefree.quux.soltec.net>

"David S. Jackson" <dsj@dsj.net>

"Todd G. Ruskell" <ruskell@boulder.nist.gov>

Rogier Wolff <R.E.Wolff@BitWizard.nl>