Nätverkssäkerhet - Teori och praktik
kurs IK1355 / IK130v

2008

Robert Malmgren
Andreas Jonsson
ROMAB

Innehållsförteckning

Figurförteckning

1 Introduktion och grundläggande koncept

In privacy and computer security, real information is too hard to find. Most people don't know what's really going on, and many people who do know aren't telling.
John Gilmore & Paul Kocher[1]

Den här texten handlar om *logisk säkerhet* och *logiska hot* mot datornätverk och de tjänster som är tillgängliga där, till skillnad mot fysisk säkerhet och fysiska hot. Med logiska hot menas hot mot IT-relaterade tjänster, resurser eller funktioner. Vi kommer att beskriva tekniska aspekter av nätverkssäkerhet, men också ta upp vissa administrativa och organisatoriska frågor som relaterar till IT- och nätverkssäkerhet. Däremot lämnas de traditionella säkerhetsfrågorna, ofta rörande fysisk säkerhet i princip helt utanför texten. Det gäller till exempel skydd mot stöld, skydd mot översvämning, skydd mot förstörande sabotage.

Den allmänna bilden av nätverks- och datorsäkerhet är att man vill stänga ute någon från en dator eller från ett datornät. Därmed vill man hindra att ett säkerhetsproblem eller en IT-relaterad incident uppstår. Men kunskap om dator- och nätverkssäkerhet innefattar mycket mer än så, och det är detta som vi tar upp i den här texten. Säkerhet är ett mycket brett begrepp som kan tolkas på väldigt skilda sätt.

Nätverkssäkerhet är viktigt. Den främsta orsaken till det ökade intresset för säkerhet, och nätverkssäkerhet i synnerhet, är att allt fler tillämpningar och system, med mer och mer information, knyts ihop i allt större och mer komplexa nätverk. Att kunna skapa beständig säkerhet i denna miljö är av största vikt för att kunna erbjuda lösningar som är funktionella, trovärdiga, kvalitativa och eftertraktade. Osäkra lösningar kan leda till brott, minskat förtroende eller regelrätta katastrofer. Tyvärr är verkligheten sådan att det är få tjänster och system, både bland egenutvecklade produkter och kommersiella standardprodukter, som faktiskt är skapade med god säkerhet som ett genuint designkrav - där utvecklare och programformgivare fått de resurser som behövs. Det är med dessa bakgrundsbilder som vi kommer att titta närmare på vanliga hot, risker och misstag. Vad som är minst lika viktigt är att vi också kommer att beskriva grundläggande säkerhetsprinciper, studera lösningsförslag och skyddsåtgärder.

1.1 Vad är information?

Innan vi går närmare in på området information- och nätverkssäkerhet måste vi beskriva det centrala begreppet *information*. Information är ett ord som används ofta och i ganska skiftande sammanhang, vilket innebär att ordet fått flera definitioner, vilka ofta styrs beroende på sammanhanget. Ordet information kommer från latinets *informa´tio*, av *info´rmo*, och betyder utbilda, undervisa, samt att ge form åt något.

Definition 1: Information är en generell beteckning för det meningsfulla innehåll som överförs vid olika former av kommunikation mellan sändare och mottagare.

1 EFF - Electronic Frontier Foundation (1998). Sid xiii i Cracking DES: Secrets of Encryption Research, Wiretap Politics & Chip Design - How Federal Agencies Subvert Privacy. Sebastopol. O'Reilly & Associates Inc.

Det går också att utrycka information som innebörd i data, där data utan innebörd eller struktur snarast kan liknas med brus. Information definieras ibland som en viss mängd upplysningar, fakta, eller underrättelser. *All information kan emellertid innebära någon form av påverkan på mottagaren, t.ex. ökad kunskap. Information kan därmed även uppfattas som resultatet av ett meddelande.*

Information är en tillgång som har ett värde för, bland andra, informationens ägare och måste få ett lämpligt skydd på samma sätt som andra värdefulla tillgångar. Många tillgångar är fysiska och är lätt identifierbara och hanterbara. Information är icke-fysiskt och immateriellt, vilket innebär vissa konsekvenser. Information har, till skillnad från andra tillgångar, den särskilda egenheten att den kan bli stulen men ändå finnas kvar hos ägaren. En stulen bil saknas från den plats där den var parkerad. Ett pappersdokument som någon obehörig läst kan finnas kvar på samma skrivbord som det låg på när det lästes. En stulen fil innebär som regel att ursprungsfilen finns kvar, medan en obehörig kopia har framställts. Tack vare att originalet finns kvar är det därför lätt att missa att information stulits. *Informationsbärare* är det sätt på vilken informationen sparas eller förmedlas. Det kan vara en skriven lapp, ett fotografi, en ljudinspelning, en runsten, ett TV-program, en webbsida eller en fil.

> **Definition 2:** Datorbehandling av information, informationsbehandling, är samlingsbegreppet för alla former av insamling, lagring, distribution och bearbetning av information för få ny, eller mer användbar, information.

Många moderna företag som gjort en analys av företagets tillgångar inser att värdet på deras informationsresurser innebär att information är den enskilt dyrbaraste eller viktigaste egendomen. Kundregister, databaser med forskningsresultat, marknads- och försäljningsstatistik, konstruktionsbeskrivningar, källkod, konkurrensanalyser och affärsplaner är exempel på den information som ett företag kan anse värdefull eller särskilt skyddsvärd. I andra typer av verksamheter, som exempelvis i modern fabrik eller ett reningsverk, är det styr- och kontrollsignaler samt statusmeddelanden från automatik och maskiner som kan vara den viktiga informationen.

1.1.1 Informationssamhället

Världssamfundet påverkas av olika globala trender. Trenderna styrs av ekonomi, tekniska framsteg och människans, som individ och i grupp, drivkraft att utforska och förnya sin omvärld, sina verktyg och hjälpmedel. Trenderna påverkar hur samhället utvecklas, bland annat med utgångspunkt utifrån landets nuvarande utvecklingsnivå. Inom länder med hög industrialisering började det som kallas *informationssamhället* att göra sitt intåg under 1900-talet och ta över det som tidigare var industrisamhället. Informationssamhället karaktäriseras av ett ökat beroende av tillgång till information och ny informationsteknologi. En särskild tjänstesektor, IT-sektorn, kom att komplettera de traditionella sektorerna såsom den tillverkande industrisektorn och jordbrukssektorn.

I informationssamhället nyttjas informationsteknologin inte enbart för att utföra banktransaktioner, surfa på Internet eller framföra telefonsamtal. Informationsteknologi har blivit en hårt integrerad del i olika samhällskritiska infrastrukturer såsom bank- och finans, el- och vattenförsörjning, transport- och logistiksystem, matproduktion, etc. På en makronivå skulle dessa infrastrukturområden i dagsläget inte fungera en längre period om IT-stödet fanns tillgängligt.

En effekt av informationssamhällets framväxt är att information blir en maktresurs. Informationshantering kan nyttjas för att kontrollera, övervaka och styra delar inom samhället. En ur ett samhällsperspektiv väl avvägd och balanserad informationshantering kan leda till positiva effekter – effektivare myndighetsutövning och nya tjänster är två uppenbara exempel. Baksidan av lättillgängliga samlingar av stora datamängder kan leda till missbruk från enskilda tjänstemän eller maktutövare eller att statsapparaten kan nyttja dessa till övergrepp i enskildas eller gruppers personlig integritet. Samkörning av olika dataregister, olämpliga eller olagliga efterforskningar, publicering eller placering av insamlat material kan leda till olägenheter för personer och organisationer. Varningar att informationssamhället skall utvecklas till ett *storebrorssamhälle* har rests ända sedan George Orwell publicerade romanen 1984[2]. I samhället är det dock inte enbart statsmakten som har möjlighet att missbruka information. Företag, organisationer och enskilda kan missbruka insamla eller inköpt information, något som ibland kallats *lillebrorssamhälle*.

Viktiga egenskapar för medborgaren i informationssamhället är trygghet och säkerhet. Grundläggande trygghet består i att känna att:

- myndigheter inte missbrukar information
- myndigheter inte lämnar ut information om medborgare till främmande makt
- att myndigheter reglerar så att företag och organisationer inte missbrukar information.

1.2 Varianter av säkerhet

1.2.1 Fysisk säkerhet

Den mest konkreta formen av säkerhet, är den som är relaterad till den fysiska världen - så kallad fysisk säkerhet. Om en dörr saknar lås går den att passera utan vidare ingrepp, en enkel glasruta går lättare att forcera än ett förstärkt glas bakom ett galler. Dessa typer av resonemang är ofta enkla att förstå för alla lekmän, vilket inte alltid är fallet med IT- och nätverkssäkerhet.

Definition 3: Fysisk säkerhet är alla aspekter av säkerhet som är relaterat till fysiska objekt

Fysiskt säkerhet (eng. *physical security*) omfattar fysiskt skydd och hot mot fysiska objekt. Fysiska skydd kan vara tillträdesskydd i form av låsta dörrar eller fördröjande åtgärder i form av inbrottslarm. Det kan också vara speciella metoder för att destruera datamedia i samband med att datamedia byts ut eller informationen av någon anledning måste förstöras. Hot mot fysiska objekt inkluderar brand, skydd mot naturkatastrofer, skadegörelse, inbrott samt stöld.

1.2.2 Logisk säkerhet

Motsatsen till fysisk säkerhet är logisk säkerhet — det som inte har att göra med den fysiska världen. Logisk säkerhet handlar ofta om abstrakta begrepp som byggts upp i flera nivåer, vilket kräver kunnande om basnivåer såsom operativsystem och kommunikationsprotokoll, tillämpningar, användarbeteenden samt olika hot- och riskbilder för IT.

Definition 4: Logisk säkerhet är alla aspekter av säkerhet som är relaterat till elektroniska, digitala objekt

[2] http://www.george-orwell.org/1984/index.html

Logisk säkerhet *(eng. logical security)* brukar omfatta exempelvis tekniska skyddsmekanismer i IT-system, IT-baserade hot mot IT-system, sårbarheter i programvara samt användningsfel.

På grund av sin abstrakta karaktär och inte sällan komplexa resonemangskedjor av kunnande för att framställa ett hot, är problem inom logisk säkerhet ofta mer svårformulerade eller svårförstådda för lekmän än exempelvis fysiska hot eller säkerhetsfrågor.

1.2.3 Informationssäkerhet

Ett fundamentalt koncept inom IT- och nätverkssäkerhet är skydd av information som hanteras i IT-system och som kommuniceras via datornätverk.

> **Definition 5:** Informationssäkerhet syftar till att skydda information mot olika hot, att säkerställa verksamhetens kontinuitet och minska skador på verksamheten samt att upptäcka när hot realiserats.

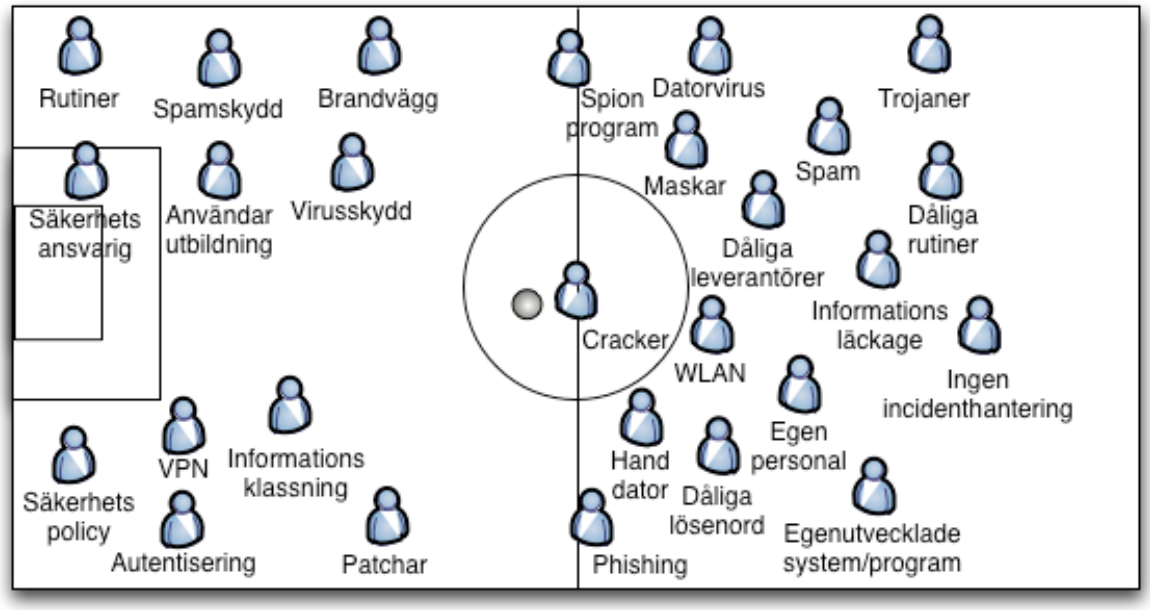
Denna beskrivning är medvetet bred och skall täcka in alla typer av omedvetna eller medvetna oönskade händelser.

Informationssäkerhet behövs för att skapa skydd för enskilda datorsystem såväl som samhällsomfattande infrastrukturer. Personalchefens stulna bärbara dator som innehåller lönelistor, underlag för den pågående lönerevisionen och personalbedömningar kan vara katastrof för ett företag. En hackad övervaknings- och reglerdator som sköter vattendistributionen hos vattenverket i en större stad kan få samhällskonsekvenser.

God informationssäkerhet uppnås genom att:

- alla som arbetar med information är medvetna om informationens värde och behovet av skydd
- man införandet av regelverk som beskriver hur man värderar och skyddar information
- resurser (ekonomi, tid, personal) läggs på framtagande och underhåll av skydd
- det sker en kontinuerlig utbildning och fortbildning
- det sker uppföljning, revision och kontroller av verksamhet, skydd och styrande dokument
- hålla en beredskap för att hantera situationer när skydd inte finns eller inte fungerar.

Att arbeta med informationssäkerhet innebär som regel en ständig ström av utmaningar, vissa är tekniska, andra är sociala eller kulturella, andra är kopplade till att arbeta på den försvarande sidan är svårare än på sidan som kan välja tid, plats och metod för sitt angrepp. Att planera och ständigt vara förberedd med ett fullgott skydd på alla upptänkliga former av angrepp är ett sisyfosarbete som de allra flesta människor inser är ett omöjligt uppdrag att genomföra till 100%.



Figur 1 Infosäkmatchen - dåliga odds att vinna i det långa loppet

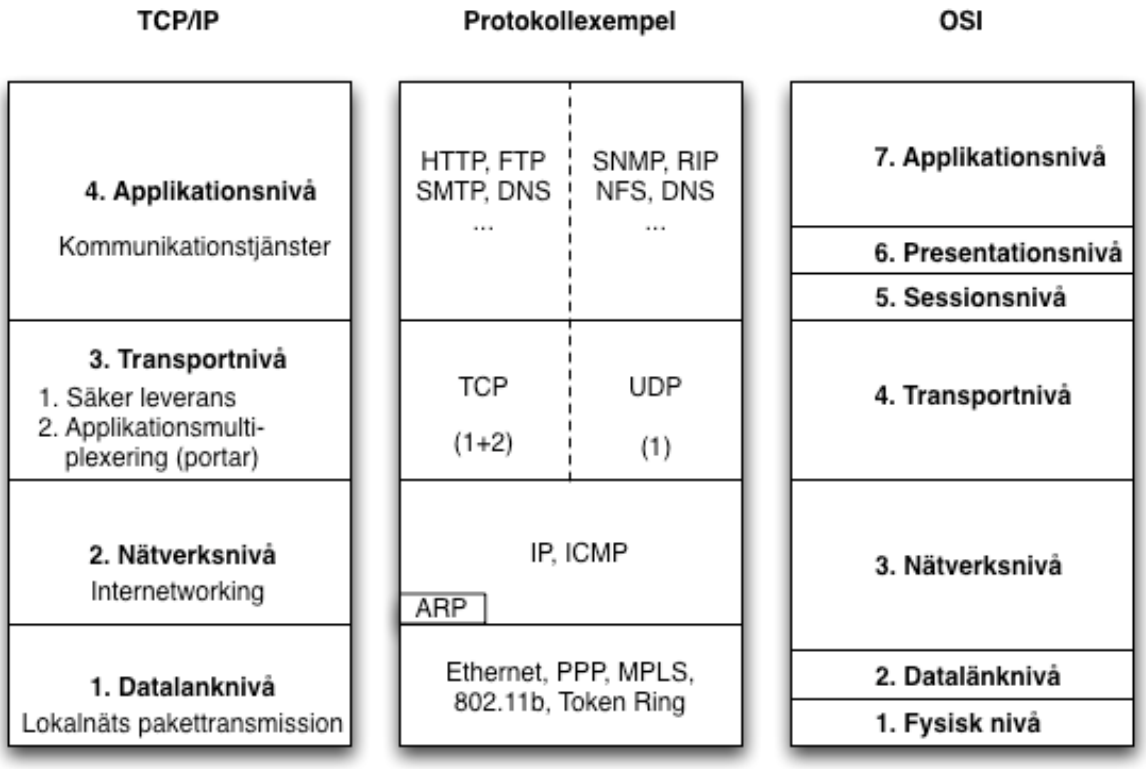
I fotbollsbilden infosäkmatchen här intill så går det att se att motståndaren inte har ett mål där vi kan få poäng, inte heller att angriparen följer reglerna och har rätt mängd personer på planen. Dessutom får de alltid avspark som en konsevens av att de är de ända som kan göra mål. Inte heller finns det någon domare på planen som kan kontrollera att de de regler som finns efterlevs.

1.2.4 Nätverkssäkerhet

Nätverkssäkerhet är centrerat runt de hot och säkerhetsmässiga skyddsfunktioner som existerar i samband med kommunikation mellan datorsystem. Vi definierar nätverkssäkerhet att inkludera:

- Kommunikationsmetoder och kommunikationsprotokoll
- nätanslutna appikationer och dess applikationsprotokoll
- kommunikationsinfrastrukturer och infrastrukturkomponenter
- information som överförs via kommunikationsmedel

I en värld som alltmer beror på tillgänglighet av information och möjlighet att skicka, ta emot och nå information via olika metoder, blir behovet av nätverkssäkerhet allt högre. En situation där stora störningar på Internet, ett utslaget SWIFT (bankernas nätverk) eller liknande centrala kommunikationsvägar, skulle snabbt leda till konsekvenser för olika tjänstesektorer eller samhället.



Figur 2 Jämförelse mellan olika protokollstackar

Nätverkssäkerhet använder metoder från IT-säkerhet och bygger i grunden på samma principiella skydds- och säkerhetsåtgärder som IT- och informationssäkerhetsområderna, fast applicerade i en nätverkad värld.

1.3 Principiella skydds- och säkerhetsåtgärder

Skydds- och säkerhetsåtgärder är de livlinor som finns för att undvika olika vanliga säkerhetsproblem. De är ofta förhindrande och förebyggande, men kan i några fall vara reaktiva och istället användas för att skapa bevis och underlag för att kunna påvisa att en viss händelse har inträffat.

Det finns anledning att gå närmare in på vissa typer av grundläggande och principiella skyddsåtgärder för att skapa en bättre förståelse för säkerhetsområdet:

- autentisering
- auktorisation
- sekretess
- riktighet
- spårbarhet
- oavvislighet
- tillgänglighet

1.3.1 Autentisering

Autentisering är metoden för att säkerställa en identitet för någon eller något när en resurs skall användas. Det vanligaste fallet är att identifiera en användare, populärt kallat inloggning,

när denne skall använda en dator. I Terminologi för informationssäkerhet definieras autentisering[3] som:

> Verifiering av uppgiven identitet eller av ett meddelandes riktighet

Vi skall dock i det här materialet använda oss av en mer generell definition av autentisering (eng. *authentication*), som inte bara involverar meddelanden, så kallad meddelandeautentisering.

> **Definition 6:** Autentisering är metoden för att knyta en identitet till ett subjekt

Den vanligaste typen av autentiseringsmekanism är lösenord - någonting man vet. Lösenord kan ses som något av en personlig kod som användaren anger för att autentisera sig mot ett system, en tjänst eller en utrustning för att bereda sig tillgång. Oavsett använd teknologi är det viktigt att man lägger tonvikt på "personlig kod", och att man inser att denna inte skall delas med eller göras åtkomlig för andra personer. Där vi säger lösenord menar vi även krypteringsnycklar, PIN-koder och liknande känslig information.

Lösenord kan i sin tur delas upp i flergångslösenord och engångslösenord, där flergångslösenord är ett ord, en mening eller en lösenordsfras (eng. *pass phrase*) som används gång på gång. Nackdelen med flergångslösenord är att de lätt kan missbrukas i händelse att lösenorden blir avlyssnade på nätverket, en lösenordsdatabas blir stulen eller en post-it-lapp sitter på en användares skärm.

Engångslösenord är ofta ett ord, en siffra eller en mening som så fort den använts blir förbrukad och inte kan användas för autentisering mera. Engångslösenord skapas ofta av lösenordsdosor eller finns listade i tabeller med förgenererade lösenord - någonting man har.

Metoder för autentisering bygger ofta på

- någonting man vet,
- någonting man har
- någonting man är

Ibland förekommer även mer esoteriska varianter, exempelvis metoden att man skall vara på en viss geografisk ort. (var man är) Denna metod används ofta i kombination med andra autentiseringsmetoder.

Förstärkt autentisering är ett begrepp som ibland används för att beskriva de lösningar som nyttjar mer avancerad teknik än enkla flergångslösenord. Ett exempel på förstärkt autentiseringslösning är att använda aktiva kort (eng. *smart cards*) för att autentisera användare vid datorinloggning.

[3] Sid 28 i SIS HB 550 *Terminologi för Informationssäkerhet*

Figur 3 Autentisering i form av ett smartkort och fingeravtrycksavläsning

En annan typ av autentiseringsmetoder som väckt intresse är biometriska autentiseringssystem, där utrustning för att mäta människors olika unika egenskaper används för identifiering – någonting man är. Fingeravtrycksläsare, ansikts- och röstigenkänning eller kontroll av ögats iris är exempel på denna typ av autentiseringssystem. Röstigenkänning kan vara en kombination av akustisk igenkänning och att den som skall autentisera sig uttalar något som bara den borde känna till, exempelvis svar på personligt unika frågor. Detta leder till en typ av biometrisk challenge-response. Det finns en del frågetecken runt de olika typerna av biometriska autentiseringsmetoder, till exempel har olika attacker mot fingeravtrycksläsare presenterats. Den naturliga följdfrågan vid sådana attacker är hur man går vidare om man väl har ett sådant systeminstallerat, man kan ju inte gärna byta fingeravtryck eller om det är möjligt att systemen kan uppgraderas så att säkerhetsproblemen elimineras. Gemensamt för de olika biometriska systemen är frågor som rör den personliga integriteten, till exempel vem som har tillgång till persondata och på vilket sätt sparade persondata kan missbrukas.

Autentisering mellan två parter kan antingen vara ensidig då enbart den ena parten autentiserar sig, eller ömsesidig autentisering då bägge parterna autentiserar sig för varandra. Enkelsidig autentisering, såsom en lösenordsbaserad direktinloggning i en dator som fysiskt är placerad framför den som logar in, fungerar bra i praktiken. En lösenordsbaserad enkelsidig autentisering vid en nätverksinloggning är teoretiskt mer tveksam då en angripare någonstans i nätverket falskeligen kan utge sig för att vara serversida och därmed stjäla autentiseringsinformationen.

Autentisering används inte bara för att identifiera *personer*. Autentisering används exempelvis ofta vid maskin-till-maskinkommunikation, mellan olika programobjekt i ett system eller mellan en dator och en kringutrustning såsom en nätverksdisk. Ett något förvånande exempel är att autentisering i ökande utsträckning används för att identifiera utrustning och reservdelar. Skrivar- och kopiatortillverkaren Xerox introducerade 1996 skrivaren N24 som kunde identifiera autenticiteten hos färgpatronen och vägrade acceptera patroner från andra tillverkare än Xerox[4]. Efter Xerox har flera tillverkare anammat samma strategi.

[4] Anderson, Ross 'Trusted Computing' and Competition Policy – Issues for Computing Professionals http://www.cl.cam.ac.uk/~rja14/Papers/tcpa-short.pdf

Eftersom lösenord är så pass vanligt förekommande skall vi nämna några ord om lösenordshantering. Inloggningsförfarandet i en dator eller vid startögonblicket för användandet av en viss webbtjänst är ett tillfälle då man autentiserar sig, ger ifrån sig uppgifter som skall bevisa för systemet eller tjänsten att man verkligen är den man utger sig för. All hantering av information som är relaterad till autentisering är en av de känsligaste delarna i ett system. Kan någon hitta lappen under tangentbordet med det uppskrivna lösenordet, avlyssna tangentbordet, spela in nätverkstrafiken eller knäcka ett sparat lösenord, så någon missbruka tjänsten eller systemet såväl som att leda spåren till en oskyldig person.

Vid användning av lösenord är det viktigt att komma ihåg att aldrig spara eller sända lösenord i klartext över nätverk. Man bör istället använda uppsäkrade protokoll eller lösningar som inte skickar klartextdata över nätet. Undvik att använda statiska (flergångs) lösenord, försök istället att använda andra lösningar, tex engångslösenord, challenge-responsprotokoll, etc. I flera system och utrustningar finns det tyvärr inget alternativ, därför bör man hantera användningen av statiska lösenorden med stor varsamhet. Använd inte heller cache- och kompletteringsfunktioner, något som fyller i hela eller resten av ordet, i applikationer och system. Dessa sparar lösenorden i temporärfiler, i registret, eller liknande som kan stjälas. Någon som temporärt får tillgång till ens dator kan därför lätt låna kontot. Skärmsläckare med lösenordskydd eller automatutloggning bör vara installerad och konfigurerad i känsliga system.

En praktiskt svår punkt, men väldigt viktig är att inte använda samma lösenord i olika system. Det är en allmänt spridd vana att återanvända lösenord i olika system eftersom det är svårt att komma ihåg stora mängder lösenord.
Det är speciellt viktigt att inte använda samma lösenord för vanliga användare och användare med hög behörighet (till exempel att systemadministratören har samma lösen för sitt personliga konto som för root- eller administratorkontot).

Det är viktigt att inte ha samma lösenord i system som är under olik administrativ kontroll (till exempel samma lösenord i hemPC-system som på arbetet). Detta är något som fått väldigt mycket uppmärksamhet i samband med några intrång som gjorts, där användarna haft samma lösenord som på olika webbplatser.

För att undvika akilleshälar måste man stänga av bakåtkompabilitet i system som sparar eller accepterar lösenord i svagare eller mer lättillgänglig form. Det främsta exemplet är bakåt-kompatibiliteten mellan Windows 2000, Windows NT och äldre Windowsvarianter. Den metod som används för att skydda lösenorden i äldre varianter av Windows går enkelt att knäcka. Motsvarande problem finns i UNIX-system där man kan ha lösenord sparade i filer läsbara för alla eller skyddade lösenordsdatabaser, sk skugglösenordsfiler. Genom att införa regelverk som beskriver hantering av lösenord och lösenordens egenskaper kan man samla olika styrkriterier, tex de som beskrivits här samt andra, för att stärka autentiseringen och hanteringen av autentiseringsinformationen. Att lösenorden måste vara starka är ett grundläggande kriterium för att skapa ett fungerande skydd.

Exempel på regler för lösenordshantering kan vara; giltighetstid, att man inte får spara lösenord i klartext, att delade konton och lösenord - så kallade gruppkonton - inte är tillåtna och liknande. Exempel på parametrar som påverkar säkerhetsnivån för lösenord är minsta tillåtna längd, krav på blandning av versaler/gemener/siffror/specialtecken, att lösenord inte skall kunna härledas ifrån kunskap om personen eller organisationen, hur ofta lösenordet måste bytas och liknande.

1.3.2 Auktorisering

Genom auktorisering (eng. *authorisation*), principen att kontrollera rättigheter till resurser, går det att begränsa och hantera personers, programs eller datorers åtkomst (eng. *access*) till resurser. eller Begränsningen kan vara inskränkningar i att använda en viss tjänst, tillämpning eller en viss typ av systemfunktion, till exempel att skriva till en fil på disk eller se en viss webbsida.

> **Definition 7:** Auktorisering är principen att tilldela eller fastställa rättigheter hos ett subjekt för åtkomst till objekt

Med hjälp av system för behörighetskontroll vill man uppnå separation av privilegier. Olika användare i ett system bör ha individuellt bestämda nivåer på sina systemprivilegier. Systemadministratörer behöver till exempel fler privilegier för att kunna läsa och skriva vilka filer som helst, medan en vanlig användare bör ha färre.

För att uppnå säkerhet i behörighetssystemet bör man anamma *principen om lägsta privilegium*. Man bör tilldela användare, program, objekt och liknande så låg behörighet och så få privilegier som möjligt. Detta för att undvika att de bereder sig tillgång till fler resurser eller mer information än de skall ha åtkomst till.

För att skydda information används ofta en mängd olika kryptografiska tekniker, exempelvis kryptering av information, signering av information samt kryptografisk säkrade tidsstämplar. Vi återkommer till dessa i krypteringsavsnittet senare i den här texten.

1.3.3 Sekretess

Skydd av information omfattar ofta att informationen skall göras oåtkomlig eller oläslig av obehöriga.

> **Definition 8:** Sekretess innebär att innehållet i ett informationsobjekt, ibland även dess existens, inte görs tillgängligt eller avslöjas för obehöriga.

Sekretess *(*eng. *confidentiality)* omfattar i IT- eller nätverkssäkerhet ofta skydd av data mot obehörig insyn under transport eller vid lagring. Ett exempel är behovet att skydda viktig data mot insyn, tex personuppgifter eller kontokortsinformation, i vägsänt från en kund till en webbserver på en e-handelsplats. Observera att insynsskydd inte per automatik innebär förändringsskydd, även om det i många fall innebär stora svårigheter att utföra förändringar på information som angriparen inte har insyn i

Det vanligaste sättet att skapa sekretess är att använda kryptering. För mer information om kryptering, se avsnitten som handlar kryptering.

Andra namn som är ganska vanliga för sekretess är konfidentialitet och insynsskydd. Sekretess kallas även för konfidentialitet eller insynsskydd.

1.3.4 Riktighet

Definition 9: Riktighet är oberörbarhet, helhet med förmåga att upprätthålla ett värde genom skydd mot oönskad förändring, påverkan eller insyn.

Tekniskt sätt brukar riktighet *(*eng. *integrity)* implementeras med hjälp av kryptografiskt starka kontrollsummor, även kallade hashvärden, elektroniska tumavtryck och message digests. Andra facktermer som används för riktighet är dataintegritet eller manipulationsskydd. Ofta förekommer enbart ordet "integrity" i engelsk facklitteratur vilket ofta leder till att man översätter det till "integritet".

I Sverige bör man vara noggrann med att skilja mellan de olika formera dataintegritet och personlig integritet då det är två helt skilda områden. Personlig integritet innebär rätten att få sin personliga egenart och inre sfär respekterad och att inte utsättas för personligen störande ingrepp. Den myndighet i Sverige som särskilt skall värna om människors personliga integritet är Datainspektionen, vilken bland annat är tillsynsmyndighet för personuppgiftslagen, PuL.

Riktighet kallas även för *dataintegritet*.

1.3.5 Spårbarhet

Skyddsmekanismer som autentisering och auktorisation bildar grundskyddet som används för att skydd av ett system, en applikation eller liknande. När skyddsmekanismerna inte fungerar, när en säkerhetsöverträdelse eller IT-incident är ett faktum, så måste man ha det näst bästa alternativet – att kunna spåra vad som inträffat, vem som gjorde det som genererade spårdatat och , när detta har inträffade.

Definition 10: Spårbarhet innebär att i efterhant entydigt kunna härleda specifika aktiviteter eller händelser till ett identifierad objekt, oftast en användare

För att skapa spårbarhet (eng. *accountability*) i ett system behövs det att applikationer skapar spårata som sparas och hanteras på ett säkert sätt. En särskild typ av spårbarhet sparas i säkerhetsloggen, vilken innefattar en sammanställning över olika säkerhetskritiska händelser i ett system.

1.3.6 Oavvislighet

Definition 11: Oavvislighet innebär att ett specifik åttagande eller en utförd handling i efterhand inte skall kunna förnekas av utföraren.

Oavvislighet (eng. *non-repudiation*) kan nyttjas för att styrka att någon:

- är avsändare eller informationsavlämnare
- har tagit emot information, exempelvis ett e-postmeddelande
- har utfört en viss handling, exempelvis autentiserat sig mot ett system, nyttjat läsåtkomst till viss information

I IT-sammanhang är oavvislighet hårt förknippat med användning av kryptografiska metoder, främst digitala signaturer.

1.3.7 Tillgänglighet

Definition 12: Tillgänglighet är möjligheten att efter behov kunna nyttja IT- och informationstillgångar i förväntad utsträckning samt inom önskad tid.

När tillgänglighet *(*eng. *availability)* avser information, snarare än system, skall definitionen även innefatta att informationen skall vara tillgänglig i ett format som användaren kan bruka.

Ett otillgängligt system kan vara helt avstängt eller ha alltför långa svarstider för att vara praktiskt användbart.

1.4 Informationssäkerhet ur ett historiskt perspektiv

Människan har i alla tider kommunicerat med andra människor. I vissa fall har kommunikationen behövts skyddas, i andra fall har anteckningar eller andra typer av informationsäbrare behövt skyddas. För att skydda informationen har människor använt de för tiden tillgängliga medlen. De som har haft råd har kunnat använda kurirpost som levererats av betrodda personer. Den grekiske historikern Herodotus (484 BC-ca. 425 BC) berättar om det klassiska grekland där en man använde sin slav som budbärare om ett annalkande angrepp från Persien[5]. Slavens huvud rakades och meddelandet tatuerarades på huvudsvålen. Efter att hår växt ut så fick slaven leverera meddelandet till mottagaren.

Med införandet av postväsendet på 1600- och 1700-talet kom behovet att skydda brev som befordrades och hanterades av okända, statsanställda personer. Officiellt så infördes regelverk med hårda straff för den som öppnade försändelser. Trots detta satte många stadsledningar i system att regelbundet öppna och kopiera försändelser till vissa personer eller till vissa länder. Franska konungen hade *Cabinet Noir*[6], i Österrike fanns ett *Geheime Kabinets-Kanzlei*[7] och i England hette motsvarigheten *Black Chamber*.

Under 1700- och 1800-talet införde många länder optiska och elektriska telegrafer. Med hjälp av telegrafen gick det att utföra automatiserad meddelandeförmedling över stora avstånd. Nackdelen för de kommunicerande parterna var att människorna som arbetade som förmedlarna måste läsa meddelandena innan de kodades och sändes. För att motverka detta införde ibland de kommunicerande parterna speciella kodade meddelanden. Detta var särskilt vanligt för diplomat meddelanden.

Telefonisystemet växte fram efter att Alexander Graham Bell uppfinner telefonen 1876. Televäxelsystemet består av manuell hantering där operatörer kopplar fram telefonsamtalet. Systemet möjliggör för operatören att avlyssna samtalet.

Under första världskriget började kryptering och kryptonanalys användas i större skala av alla inblandade i konflikten. Den första storskaliga användningen av mekaniska eller manuella system såsom det tyska ADFGVX-kryptot[8] användes regelbundet.

Information som samlas in för ett ändamål kan i någon annans händer missbrukas. I de länder som Nazityskland invaderade under andra världskriget tvingades folk att lämna från sig

[5]Sid 81 i Kahn, David (1967) *The Codebreakers: The story of secret writing*. Macmillan Co, New York.

[6]Sid 162 i Kahn:s *Codebreakers*

[7]Sid 163 i Kahn:s *Codebreakers*

[8]Newton, David E (1997). *Encyclopedia of Cryptology*. Santa Barbara: ABC-Clio Inc. ISBN 0-874-36772-7

radioapparater och radiosändare. Ofta användes de register som fanns i respektive land över radiolicensinnehavare för att spåra upp personer som inte lämnat in radioapparater.
Med datorernas intåg har behovet av informationssäkerhet ständigt ökat. Ett fåtal enkla elektromekaniska datorer började att användas under andra världskriget. Datorerna användes för militära beräkningar samt analys av fiendernas krypterade radio-, tele- och telegramtrafik. På grund av det militära bruket var det fysiska skyddet bra i form av beväpnade vakter, låsta utrymmen och begräningar i vilka som fick åtkomst till såväl utrustning som hanterad information.

Under kalla kriget som pågår under senare hälften av 1900-talet pågår en ständig tävlan mellan supermakterna USA och Sovjetunionen och dessa alierade. Bägge sidor skapar stora och mäktiga underrättelseorganisationer för att på en global nivå stjäla, avlyssna och förfalska motståndarens information. Den idag välkända amerikanska organisationen NSA, *National Security Agency*, skapas i november 1952[9], men var under större delen av 1900-talet en okänd myndighet för de flesta inom såväl utanför USA. Supermakterna skapar också säkerhetsorganisationer för att skydda egen information, spåra och utreda informationsläckor eller säkerhetsöverträdelser. Kryptoteknologi utvecklas i snabb takt och baseras på matematiska grunder. Då kryptologi anses vara en strategisk kunskap utformade många länder, däribland Sverige[10], exportrestriktioner av kryptoutrustning.

Under 1900-talet förekommer det ofta inom myndigheter och företag säkerhetsskåp där viktig material i form av pärmar med papper skulle låsas in. Pappersinformation märktes ofta med stämplar som beskrev hur hemlig informationen var.

Under 1970-talet fanns det stor- och minidatorer inom företag och myndigheter. In- och utmatning skedde ofta via hålkort som hanterades av särskilda operatörer. Interaktiva fleranvändarsystem började användas i större utsträckning. Datorsäkerhet, som ämnes- och forskningsområde, uppstod under det tidiga 1970-talet bland forskare framförallt knutna till militären i USA. Hot mot dator- och operativsystem, hot mot den datorbearbetade informationen samt datorer och operativsystems skyddsmekanismer studerades.

1980-talet medförde IBM:s nya produkt — persondatorn. Istället för dyra stor- och minidatorer som nyttjades via terminaler uppstod nu möjligheten för var och en att ha en egen dator. Stor- och minidatorer var inlåsta i datorhallar, hade operatörer som utförde säkerhetskopiering och systemadministratörer som skötte om säkerhetsinställningar och konfiguration. När datorn hamnade på skrivbordet behövde användaren själv börja tänka på dessa saker.

En annan nyhet under 1980-talet var datorvirus, program som obehörligen spreds till andra datorer och ofta hade en destruktiv påverkan på datorerna eller datorns information. Hemdatorer såsom Apple II hade drabbats av virus redan i början av 1980-talet. En av de första kända utbrotten av PC-virus är det såkallade Brain-viruset[11] kommer i omlopp 1986 och startar en våg av olika virusvarianter.

---

[9] *The Origins of NSA* http://www.nsa.gov/publications/publi00015.cfm

[10] SFS 1994:2060 (1994). *Förordning om strategiska produkter*. Stockholm. Fritzes förlag.

[11] Wells, Joe (1996): "Brief History of Computer Viruses" http://www.research.ibm.com/antivirus/timeline.htm

1983 uppmärksammades för första gånge företeelsen hacking av populärkulturen tack vare en Hollywoodfilm. Wargames[12] handlar om en pojke som bryter sig in i amerikanska militärens datorsystem och är på randen att starta tredje världskriget.

I slutet av 1980-talet och i början på 1990-talet började de flesta organisationer bygga ihop persondatorerna i lokala nätverk inom organisationens lokaler. Så småningom började de olika lokalnätverken förbindas via fjärrnät till att knyta ihop flera delar av organisationen, i exempelvis koncernnät. En obehörig användare, eller en behörig användare på upptäcksförds, någonstans kunde helt plötsligt få åtkomst till en dator någon helt annanstans via nätverket. I november 1988 släpptes ett program i form av en mask[13] ut på Arpanet, dåtidens Internet, vilket medförde att många av de anslutna datorerna blev infekterade, överlastade och smittade andra datorer. Masken var ett sofistikerat program som använde flera olika metoder för att automatiskt hacka sig in i en dator. Masken var skriven av studenten Robert Morris, son till Robert Morris Sr, som var chef på NSA.

I mitten av 1990-talet fick Internet en spridning till de flesta organisationer och personer i de industrialiserade länderna. Internet förändrade mycket av de under 1900-talet rådande skyddsstrategierna - att datorerna var säkra tack vare fysiska skydd såsom att datorn var inlåst i företagets lokaler samt att lösenord räckter som åtkomstskydd. Nätverkssäkerhet i form av brandväggar framträder som den nya tidens säkerhetsstrategi.

I slutet av 1990-talet och början av 2000-talet lyfts eller lättas exportrestriktioner på kryptoteknologi i många länder, inklusive USA[14] och Sverige[15]. Detta öppnar möjligheten att skapa säkrare datorsystem och säkrare kommunikationsprotokoll.

Persondatorn fick under slutet av 1990-talet en genomslagskraft hos de svenska hushållen i samband med hem-pc-reformen tack vare att datorer för utbildning undantogs från förmånsbeskattning. Hem-PC:n i kombination med den explosionsartade bredbandsutbyggnaden medförde att det snart stod en internetkopplad dator hemma hos varje familj. Säkerhetskopiering, uppdatering av antivirusprogram och basprogramvara, konfiguration av hembrandväggar blev snart var mans huvudvärk.

Med Internet kom även nya typer av problem såsom elektronisk skräppost och internetbedrägerier. *Spam*, ett otyg i form av oönskad skräppost, blir ett nytt ord som alla datoranvändare snabbt har lärt sig. Försök till bedrägerier på Internetauktioner, i datorspel och via förfalskade e-postuppmaningar är något som de flesta datoranvändare en eller fler gånger har stött på.

Under 1990-talet gör även informationsteknologin stor frammarsch inom vissa industritillämpningar, främst för styr- och reglerändamål. I vissa fall leder detta till olika typer av missbruk och IT-incidenter. Bland de mest kända är kärnkraftverket Davis-Besse som

[12]http://us.imdb.com/title/tt0086567/

[13]Seeley, Donn (1988): "A Tour of the Worm". http://world.std.com/~franl/worm.html

[14]Svidén, Henrik (2000) "USA häver exportrestriktioner för krypto" http://nyheter.idg.se/display.asp?ID=000113-CS9

[15]"RÅDETS FÖRORDNING (EG) nr 1334/2000 av den 22 juni 2000 om upprättande av en gemenskapsordning för kontroll av export av produkter och teknik med dubbla användningsområden" http://www.isp.se/documents/public/se/pdf/ lagar/1334_2000.pdf

drabbas av masken Slammer[16] och vattenreningsverket Maroochy Shire i Australien som en mängd gånger hackades[17] via trådlöst nätverk av Vitek Boden, en före detta entreprenör.

Med 2000-talket kom nya hot såväl som nya skydd. Massiva överlastningsattacker[18] och utpressningsförsök mot e-handelstjänster blev nya företeelser. Biometriska skydd i form av fingeravtrycksläsare och retinaskanners blev mycket omtalade, om än lite använda i verkligheten. Svenska banker som SkandiaBanken och Nordea blir utsatta för online-varianter av rån där angripare lyckas stjäla pengar.

2 Hotbeskrivningar

7The only security is the constant practice of critical thinking.
- William Graham Sumner (1840-1910)

Not everything that can be counted counts,
and not everything that counts can be counted.
- Albert Einstein

2.1 Viktiga begrepp relaterat till hot och risk

Varje uppfinning har dragits med sina egna olyckor, så också informationshantering. Mänskliga misstag, avsiktligt missbruk eller regelrätta angrepp är olika problem som drabbar alla skeden av informationshantering med varierande konsekvenser. I och med att IT i alla former alltmer påverkar allas vardagsliv ökar därmed risken att bli drabbad av de hot som finns mot IT. Detta kapitel är en genomgång av olika varianter av begreppet hot samt systematiserar olika hottyper.

Ett centralt begrepp inom informationssäkerhet är begreppet *hot*. Hot används inom riskanalys, bedömningar och analys av skydds effektivitet, diskussioner om angrepp mot IT och information och så vidare.

> **Definition 13:** Hot är oönskade medvetna eller omedvetna handlingar eller händelser, som utförs eller orsakas av människor, av människan skapade artefakter eller naturliga fenomen och som antas kunna riktas mot viktiga värden eller en aktuell tillgång.

Ett hot kan vara ett avsiktligt hot eller ett oavsiktligt hot. IT-system och information är ofta utsatta för bägge typerna av hot, varpå bägge måste hanteras i riskhanteringsprocessen.

> **Definition 14:** Ett avsiktligt hot är en möjlig aktivitet som syftar till att skada en verksamhet, ett system eller en person

Ett exempel på ett avsiktligt hot är ett angrepp på en webbserver med mål att ändra webbsidor eller att ta sig in i en databas för att ändra dess innehåll.

> **Definition 15:** Ett oavsiktligt hot existerar trots att illasinnad avsikt saknas

[16] "Slammer worm crashed Ohio nuke plant network" http://www.securityfocus.com/news/6767
[17] http://www.crime-research.org/analytics/501/

[18] Rosencrance, Linda (2000) *'Mafiaboy' to plead guilty to hacking major Web sites*. Artikel i webbversionen av tidningen COMPUTERWORLD daterad NOVEMBER 07, 2000 http://www.computerworld.com/securitytopics/security/story/0,10801,53492,00.html

Exempel på oavsiktliga hot är mänskliga misstag och naturkatastrofer. Det är svårt att beskriva en enstaka, entydlig, hotbild mot IT. Hotbilden är en samlad bedömning av vilka hot som föreligger vid en viss given tidpunkt. Inte minst på grund av att förutsättningar för hur ett hot skall realiseras är olika beroende på olika scenarion, tekniska parametrar och så vidare.

Definition 16: Hotbild är en föreställning om aktuella eller framtida risker, om en situation eller händelseutveckling där värden eller tillgångar som är viktiga för en individ eller organisation kan sättas på spel.

Definition 17: *Risk* brukar definieras som sannolikheten för att en oönskad händelse, ett hot, skall realiseras.

Ibland uttrycks detta i termer av sannolikhet, som innebär möjligheten att något inträffar, samt konsekvens, effekterna av att något inträffar:

$$Risk = Sannolikhet \times Konsekvens$$

Denna tillsynes enkla och ingenjörsmässiga formel är dock förrädisk, då uppskattning av sannolikheten att något säkerhetsproblem inträffar såväl som konsekvenserna av det inträffade kan vara extremt svårt.

Definition 18: Riskanalys är metoden att systematiskt utföra en värdering av risk mot, eller vid användning av, system eller funktioner.

Det existerar olika metoder och modeller för att försöka göra riskbedömningar och riskanalys av system, tjänster, projekt och organisationer. En i Sverige vanligt förekommande riskbedömningsmetod är den så kallade SBA-metoden, där man använder olika scenarion för att försöka beskriva de olika hoten och riskerna. Inom vissa företag och organisationer används dessa för att bygga skyddsbarriärer som är anpassade efter de mer initierade bedömningar man gjort. Tyvärr är det vanligt att konsekvensanalys eller riskanalys inte föregår ett projekt eller är inkluderat i det. Det innebär att man genomför förändringar utan att egentligen förstå hur dessa påverkar säkerheten.

Att göra realistiska och korrekta riskanalyser är extremt svårt. Det finns mängder med parametrar som påverkar slutresultatet. Har man gjort rimliga sannolikhetsbedömningar för att ett hot ska inträffa? Har man fått med alla relevanta hot? Hur lång tid håller sig bedömningarna aktuella? Är effekterna och konsekvenserna av riskerna tillräckligt utvärderade? I IT-säkerhetssammanhang är det ännu svårare att göra bra bedömningar än inom den traditionella riskanalysen. Några av orsakerna är att man har mindre erfarenhet av riskbedömningar inom IT eftersom det är en yngre bransch där man fortfarande provar vingarna, att system och nätverk – speciellt Internet- är så komplexa att få personer, om ens någon, har helhetsbilder, samt att det inom IT kan ske snabbare och större svängningar eller förändringar än vad de flesta är medvetna om. Det som är säkert och vedertaget ena dagen kan vara förkastligt eller osäkert dagen efter.

Definition 19: En **sårbarhet** eller **svaghet** är skillnaden i form av bristande förmåga att motstå hot, mellan *önskad* säkerhet och *verklig* säkerhet i ett system eller i en verksamhet, vid ett visst givet tillfälle,

Definition 20: En **katastrof** är en mycket stor olycka eller händelse, i vissa fall med omfattande materiell förödelse, efter vilken man har svårt att tänka sig ett återställande eller en fortsättning.

Sårbarheter i programvara är ofta programfel som existerat en tid, varit latenta, tills någon vid ett senare tillfälle, av misstag eller systematisk analys, upptäcker dem och på något sätt förhåller sig till denna nya kunskap. Antingen genom att kommunicera sin upptäckt till leverantören, att publicera sin upptäckt på via olika kanaler såsom exempelvis Internet, eller helt enkelt behålla kunskapet för sig själv. Mängden upptäckta, men ännu inte publikt kända säkerhetshål, är enligt flera bedömmare hög.

Hotmodellering (eng. *threat modelling*) är en metod för att systematiskt försöka hitta de hot som existerar mot ett system, en komponent eller ett användningsfall. Hotmodellering är en viktig komponent i flera utvecklingsmodeller, exempelvis Microsofts Secure Software Development Lifecycle (SDLC), för att under analys- och designfasen förebyggande lyckas införa rätt skydd, och skydd till rätt nivå.

Attackträdsmodellering är en hotmodelleringsmetod som är besläktad med felträdsmodellering. Rotvärdet i den uppochnervända trädstrukturen är det mål som skall uppnås, t.ex. ett lyckat intrång, och stammen och grenarna i trädet är moment eller förutsättningar som måste uppnås för att kunna ta sig närmare målet. Metoden utvecklades ursprungligen av Bruce Schneier[19]. Attackträdsanalys har blivit en standardmetod för att modellera hot mot kommunikationsprotokoll, industrikommunikationsprotokoll[20] och finns ofta används i de RFCer som beskriver olika IETF-utvecklade protokoll. En liknande metod för att upptäcka sårbarheter och brister är AVA, *Adversarial Vulnerability Assessment*[21], som innebär att man ikläder sig angriparens roll och försöker hitta sårbarheter och blottor.

Det finns en mängd omständigheter som kan leda till att man drabbas av angrepp. Det uppenbara är att man riskerar bli ett utvalt mål. Men vad som ofta förbises, både i det vardagliga tankesättet och i mer strukturerade genomgångar såsom riskanalyser, är att man kan bli drabbad mer eller mindre slumpmässigt.

På Internet existerar ett konstant ”bakgrundsbrus” av aktiva attacker och felkonfigurerad utrustning eller programvara, som gör att hot existerar även för som försöker hålla en låg profil. Risken att bli drabbad av någon spontan elektronisk avsökning eller att slumpvis bli utvald till IT-varianten av *drive-by shooting* är betydande.

För att hantera hot och risker måste man införa säkerhetstänkande och säkerhet i det vardagliga arbetet. Ett lågt säkerhetsmedvetande och dåligt engagemang hos personal och ansvariga kan vara den absolut största risken. Mot detta hjälper inte införande av moderna tekniska skyddsmekanismer som brandväggar och kryptoteknologi. Kontinuerlig utbildning, lättillgänglig säkerhetsinformation och lättförståeliga regler är några viktiga komponenter för att öka medvetande och engagemang. En organisations *säkerhetskultur* är det förhållningssätt och attityder som organisationen själv och dess anställda har till risker.

Definition 21: Attackyta är den del av en mjukvara som är nåbar för olika typer av angrepp. Attackytor beskriver attackerbarheten som en egenskap hos ett program eller en mjukvarukomponent

Definition 22: Attackvektor är den väg eller metod som ett angrepp nyttjar för att nå en given sårbarhet.

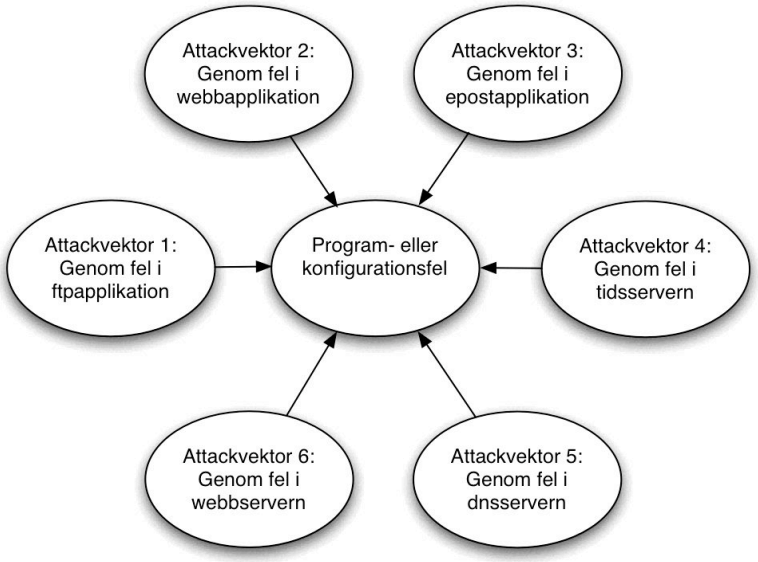
[19] Schneier, Bruce (1999). Attack Trees - Modeling security threats. Artikel i Dr. Dobbs Journal, december 1999. http://www.ddj.com/documents/s=896/ddj9912a/

[20] E.J. Byres, M. Franz & D. Miller (2004). The Use of Attack Trees in Assessing Vulnerabilities in SCADA Systems. International Infrastructure Survivability Workshop (IISW '04), IEEE,

[21] http://www.ne.anl.gov/capabilities/vat/philosophy.html

Ofta finns det mer än en metod eller ingång för någon att nyttja ett säkerhetsproblem. Detta kallas för att man nyttjar en attackvektor för att nå säkerhetsproblem. En viktig detalj att komma ihåg när det diskuteras sårbarheter i IT-system är att det kan finnas fler än en attackvektorer för att kunna missbruka ett givet säkerhetsproblem. Om själva säkerhetsproblemet finns i ett delat bibliotek, en DLL-fil, som nyttjas från flera program så kan därmed säkerhetsproblemet nyttjas genom missbruk av de olika programmen. Ett exempel är säkerhetsproblemet rapporterades i Microsoft webbserver IIS och dess tilläggstjänst WebDAV. Microsoft annonserade i mitten av mars 2003 att man hade lagat hålet i WebDAV. Problemet var att det egentliga säkerhetshålet fanns i funktionen *RtlDosPathNameToNtPathName_U* i en DLL-fil kallad *ntdll.dll* och därmed kunde säkerhetsproblemet nyttjas ifrån en mängd andra program som direkt använde ntdll. Någon kunde också missbruka RtlDosPathNameToNtPathName_U indirekt via andra funktioner som fanns i andra DLL-filer, vilka i sin tur anropade den felande funktionen.

Problemet med attackvektorer är att det är lätt hänt att en leverantör eller utvecklare stänger *en* lucka och därmed enbart eliminerar ett *symptom*. Snarare måste den som lagar ett säkerhetsproblem verkligen förstå vad som är det egentliga problemet och därmed ta bort det egentliga problemet en gång för alla.



Figur 4 Attackvektor - att hitta olika vägar att uttnyttja ett säkerhetsproblem

Skadeförebyggande åtgärder är olika införda åtgärder som påverkar sannolikheten för att en skada skall inträffa. Skadebegränsande åtgärder är åtgärder som skall påverka konsekvenserna av en skada när den väl har inträffat. Skademinimering är minimering av konsekvenser och kostnader av en skada som ibland används som synonym till skadebegränsande åtgärder. Skadeeliminimering är att helt utesluta möjligheten att skadan kan uppstå.

2.2 Hackers, Crackers och hackerkultur

Ett av de mer använda ordet inom IT- och informationssäkerhet är *hacker*, ibland kallat hackare på svenska. Ett grundläggande problem med ordet är dess tvetydighet.

Ursprunligsdefinitionen handlade om att någon var duktig på datorer. Numera nyttjas ordet hacker som ett ord för att beskriva någon som begår IT-relaterad brottslighet och hacking är liktydigt med datorintrång. För att sätta orden hack, hacker, hacking och cracker i ett nyanserat perspektiv skall vi göra en längre genomgång av nomenklaturen.
Hacker härleds ur ordet hack. Ordboken *The New Hackers Dictionary* (TNHD) definierar ordet Hack[22] som:

> **Hack**
> 1. /n./ Originally, a quick job that produces what is needed, but not well.
> 2. /n./ An incredibly good, and perhaps very time-consuming, piece of work tha produces exactly what is needed.
> 3. /vt./ To bear emotionally or physically. "I can't hack this heat!"
> 4. /vt./ To work on something (typically a program). In an immediate sense: "What are you doing?" "I'm hacking TECO." In a general (time-extended) sense: "What do you do around here?" "I hack TECO." More generally, "I hack 'foo"' is roughly equivalent to "'foo' is my major interest (or project)". "I hack solid-state physics." See Hacking X for Y.
> 5. /vt./ To pull a prank on. See sense 2 and hacker (sense 5).
> 6. /vi./ To interact with a computer in a playful and exploratory rather than goal-directed way. "Whatcha up to?" "Oh, just hacking."
> 7. /n./ Short for hacker.
> 8. See nethack.
> 9. [MIT] /v./ To explore the basements, roof ledges, and steam tunnels of a large, institutional building, to the dismay of Physical Plant workers and (since this is usually performed at educational institutions) the Campus Police. This activity has been found to be eerily similar to playing adventure games such as Dungeons and Dragons and Zork. See also vadding.

Ordet hacker[23] är i samma ordbok definierat som:

> **Hacker** /n./
> [originally, someone who makes furniture with an axe]
> 1. A person who enjoys exploring the details of programmable systems and how to stretch their capabilities, as opposed to most users, who prefer to learn only the minimum necessary.
> 2. One who programs enthusiastically (even obsessively) or who enjoys programming rather than just theorizing about programming.
> 3. A person capable of appreciating hack value.
> 4. A person who is good at programming quickly.
> 5. An expert at a particular program, or one who frequently does work using it or on it; as in 'a Unix hacker'.
> (Definitions 1 through 5 are correlated, and people who fit them congregate.)
> 6. An expert or enthusiast of any kind. One might be an astronomy hacker, for example.
> 7. One who enjoys the intellectual challenge of creatively overcoming or circumventing limitations.
> 8. [deprecated] A malicious meddler who tries to discover sensitive information by poking around. Hence 'password hacker', 'network hacker'. The correct term for this sense is cracker.
> The term 'hacker' also tends to connote membership in the global community defined by the net (see network, the and Internet address). It also implies that the person described is seen to subscribe to some version of the hacker ethic (see hacker ethic).
> It is better to be described as a hacker by others than to describe oneself that way. Hackers consider themselves something of an elite (a meritocracy based on ability), though one to which new members are gladly welcome. There is thus a certain ego satisfaction to be had in identifying yourself as a hacker (but if you claim to be one and are not, you'll quickly be labeled bogus). See also wannabee.

The New Hackers Dictionary definierar ordet Hackeretik[24] som:

> Hacker ethic /n./
> 1. The belief that information-sharing is a powerful positive good, and that it is an ethical duty of hackers to share their expertise by writing free software and facilitating access to information and to computing resources wherever possible.
> 2. The belief that system-cracking for fun and exploration is ethically OK as long as the cracker commits no theft, vandalism, or breach of confidentiality.

[22] "hack" i *The on-line hacker Jargon File*, version 4.4.7 http://www.catb.org/~esr/jargon/html/H/hack.html

[23] "hacker" i *The on-line hacker Jargon File*, version 4.4.7 http://www.catb.org/~esr/jargon/html/H/hacker.html

[24] "hacker ethic" i *The on-line hacker Jargon File*, version 4.4.7. http://www.catb.org/~esr/jargon/html/H/hacker-ethic.html

> Both of these normative ethical principles are widely, but by no means universally, accepted among hackers. Most hackers subscribe to the hacker ethic in sense 1, and many act on it by writing and giving away free software. A few go further and assert that all information should be free and any proprietary control of it is bad; this is the philosophy behind the GNU project.
> Sense 2 is more controversial: some people consider the act of cracking itself to be unethical, like breaking and entering. But the belief that 'ethical' cracking excludes destruction at least moderates the behavior of people who see themselves as 'benign' crackers (see also samurai). On this view, it may be one of the highest forms of hackerly courtesy to (a) break into a system, and then (b) explain to the sysop, preferably by email from a superuser account, exactly how it was done and how the hole can be plugged -- acting as an unpaid (and unsolicited) tiger team.
> The most reliable manifestation of either version of the hacker ethic is that almost all hackers are actively willing to share technical tricks, software, and (where possible) computing resources with other hackers. Huge cooperative networks such as Usenet, FidoNet and Internet (see Internet address) can function without central control because of this trait; they both rely on and reinforce a sense of community that may be hackerdom's most valuable intangible asset.

Någon som begår datorbrott brukar kallas för cracker. Här följer definitionen av cracker[25] i TNHD:

> **Cracker** /n./
> One who breaks security on a system. Coined ca. 1985 by hackers in defense against journalistic misuse of hacker (q.v., sense 8). An earlier attempt to establish 'worm' in this sense around 1981--82 on Usenet was largely a failure.
> Use of both these neologisms reflects a strong revulsion against the theft and vandalism perpetrated by cracking rings. While it is expected that any real hacker will have done some playful cracking and knows many of the basic techniques, anyone past larval stage is expected to have outgrown the desire to do so except for immediate, benign, practical reasons (for example, if it's necessary to get around some security in order to get some work done).
> Thus, there is far less overlap between hackerdom and crackerdom than the mundane reader misled by sensationalistic journalism might expect. Crackers tend to gather in small, tight-knit, very secretive groups that have little overlap with the huge, open poly-culture this lexicon describes; though crackers often like to describe themselves as hackers, most true hackers consider them a separate and lower form of life.
> Ethical considerations aside, hackers figure that anyone who can't imagine a more interesting way to play with their computers than breaking into someone else's has to be pretty losing. Some other reasons crackers are looked down on are discussed in the entries on cracking and phreaking. See also samurai, dark-side hacker, and hacker ethic. For a portrait of the typical teenage cracker, see warez d00dz.

Själva datorbrottet att bryta sig in i en dator kallas allmänt för cracking. Här följer definitionen av cracking[26] i TNHD:

> **Cracking** /n./
> The act of breaking into a computer system; what a cracker does. Contrary to widespread myth, this does not usually involve some mysterious leap of hackerly brilliance, but rather persistence and the dogged repetition of a handful of fairly well-known tricks that exploit common weaknesses in the security of target systems. Accordingly, most crackers are only mediocre hackers.

2.3 Informationsinsamling

Med informationsinsamling menar vi en mildare typ av attack som har som syfte att inhämta information om det tilltänkta målet. Informationsinsamling är ibland ett första steg i en mer omfattande attack. Beväpnad med viss grundinformation kan man sedan gå vidare med mer sofistikerade och skräddarsydda attacker. Exempel på informationsinsamling är:

- Portscanning
- Nätverksscanning
- Inhämtning av välkomstmeddelanden, bannerinformation,
- statusinformation, versionssträngar och liknande från aktiva nättjänster
- Inhämtande av publik information från organisationens egna webbsidor, bannertexter på nättjänster och liknande

[25] "cracker" i *The on-line hacker Jargon File*, version 4.4.7. http://www.catb.org/~esr/jargon/html/C/cracker.html

[26] "cracking" i *The on-line hacker Jargon File*, version 4.4.7. http://www.catb.org/~esr/jargon/html/C/cracking.html

- Inhämtande av publik information från andra organisationers webbsidor, databaser och liknande. Det kan handla om gula sidorna, NIC-SEs eller Internics nätadressregister eller Netcrafts register över webbservrar på Internet.

En viktig del i informationsinsamlingen kan vara att fastställa vilka tillämpningar som körs, deras version samt vilket operativsystem värddatorn har. En metod för att göra fjärrdetektering av vilket system eller operativsystem som används kallas för ”remote OS fingerprinting” och baserar sig på identifiering av vissa nätverkskarakteristika hos operativsystem. Vanliga identifierings- och urvalskriterier är hur operativsystemets TCP/IP-stack fungerar. Det kan handla om hur stacken hanterar IP-optioner, paketfragmentering, hur den skapar sina sekvensnummer för TCP (ofta kallat *ISN*, *initial sequence number*) eller hur den hanterar vissa flaggor och flaggkombinationer.

Idag finns det flera program för fjärrdetektering, en metod för att via nätverket kunna bestämma ett operativsystem. Några exempel är queso och nmap (http://www.insecure.org/nmap/) Efter att ha beväpnat sig med denna grundinformation kan en angripare göra mer specifika och målinriktade attacker. Med dessa riktade attacker uppnås ofta bättre resultat än ett mer slumpmässigt testande.

2.3.1 Port- och nätverksscanning

Med hjälp av så kallad portscanning (eng. *port scanning*) försöker man kartlägga exempelvis vilka UDP-, RPC- och TCP-portar som är tillgängliga på ett system. Det finns en lång rad av metoder och varianter för att utföra portscanning. Till de vanligaste hör följande:

- Stealthscanning (fördold scanning)
- SYN-scanning (enbart första SYN-paketet skickas)
- FIN-scanning. Genom att skicka paket med finishflaggan (FIN) kan vissa brandväggar, brandväggsregler och säkerhetsanordningar passeras.
- RST-scanning. Genom att skicka paket med resetflaggan (RST) kan vissa brandväggar, brandväggsregler och säkerhetsanordningar passeras.
- Christmas-scan. Genom att sända paket som har alla TCP-flaggor satta, så kallade julgranspaket, något som normalt aldrig skall inträffa, kan man förvirra vissa TCP/IP-implementationer.
- RPC-scanning. Speciell metod för att leta efter och upptäcka RPC- baserade tjänster.

Genom att analysera hur ett system svarar på en scanning kan programmet göra vissa bedömningar om det scannade systemet ligger bakom en brandvägg som filtrerar bort trafik eller inte. Normalt returneras ett ”*ICMP port unreachable*” när man försöker nå en port som inte är aktiv. Om det scannade systemet inte returnerar några ICMP-orienterade statusmeddelanden antar programmet att en brandvägg skyddar det scannade systemet.

De flesta scannerprogram använder olika metoder för att undvika upptäckt. En metod som används är att slumpa fram portarnas ordningsföljd istället för att systematiskt och linjärt testa port efter port.

Modernare varianter av nmap har stöd för att göra en djupare analys av vad som egentligen lyssnar på en viss port. Bara för att någon tjänst har öppnat port 80 så betyder inte det att det per automatik svarar en webbtjänst där. Det är relativt vanligt förekommande att personer installerar annorlunda tjänster, såsom SSH, på välkända portar för att dessa är öppna igenom företagets brandvägg. Nmap har numera förmågan att utföra systematiska tester och sedan analysera svaren och på så sätt komma fram till att det exempelvis är OpenSSH 3.7 som

svarar på port 80. Denna typ av detaljkunskap ger stora fördelar till någon som försöker förbereda ett angrepp. En metod för att kartlägga vilka datorer som är åtkomliga är att med hjälp av någon variant av programmet ping, som sänder ut testpaket, echo REQUEST, till en serie datorer.

En nätverksscanning är att använda olika teknologier, exempelvis ping eller uppkopplingsförsök till vanliga tjänster, för att se ifall det finns aktiva datorer på nätverket som är nåbara.

En vanlig föreställning är att i princip alla nätbaserade attacker föregås av olika aktiviteter som informationsinsamling och port- och nätverksscanningar. Detta skulle innebär att man lätt upptäcker attackerna eftersom brandväggar och andra skyddsmekanismer skulle detektera och larma på den onormala aktiviteten. Detta är en felaktig föreställning, då många attacker idag utförs med olika verktyg som utför enstaka eller fåtal attacker mot enstaka tjänster, ofta genom att utnyttja nyupptäcka hål. Detta kan passera oupptäckt genom brandväggar och IDS-system, eller döljas i det ordinarie bruset i loggarna och larmen.

2.4 Säkerhetsproblem

I diskussioner rörande nätverkssäkerhet tänker folk i allmänhet på olika attacker som sker under själva befordranstillfället, till exempel avlyssning av överförd information på nätverket. Givetvis är detta ett av de tillfällen då en attack kan ske, men det är väldigt viktigt att inse att det inte är det enda tillfället för en attack.

Om informationen, exempelvis ett kontokortsnummer, är väl skyddad vid överföringen med hjälp av kryptering, så kanske det i praktiken är omöjligt att avlyssna informationen på nätverket. Kontokortsnumret har ju ett speciellt användningstillfälle, till exempel vid en transaktion i en webbaseradtillämpning. Därefter brukar informationen om transaktionen sparas någonstans, ofta i en databas. Detta kan vara akilleshälen som angriparen letar efter. En attack mot webbserverprogramvaran, mot databashanteraren, mot operativsystemet eller liknande, kan leda till ett lyckat intrång och åtkomst till informationen – kunddata med kontokortsinformation i vårt exempel. Det är därför viktigt att ha ett helhetsperspektiv när man pratar om nätverkssäkerhet och att man försöker se hela kedjan med alla inblandade länkar. Annars gör man det alltför vanliga felet att enbart koncentrera sig på kommunikationssäkerheten, något som kan leda till ödesdigra konsekvenser.

Konsten att upptäcka och utnyttja säkerhetshål och säkerhetsproblem har historiskt sett varit förbehållet folk med djup teknisk kunskap och lite speciella tankesätt – en förkärlek att hitta fel. Detta innebar att risken inte var speciellt stor att bli utsatt för en attack. Man kanske kunde bli drabbad om man hade anställt en systemprogrammerare med alltför livlig fantasi och en kriminell läggning.

Möjligheten att missbruka system och program, att utnyttja säkerhetsproblem, har ändrats radikalt genom tillgången på färdigpaketerade attackprogram, attackskript, tillsammans med ingående beskrivningar av säkerhetshål, sårbarheter och svagheter (eng. *vulnerabilities*) och en mer allmänt spridd kunskap om säkerhetsproblem. Dessa färdigförpackade program kallas ofta sammantaget för ”exploits” eftersom de utnyttjar, exploaterar, etet säkerhetsproblem. De är väl spridda, främst via Internet, och är lättanvända verktyg eller lätthanterade beskrivningar som även gör det möjligt för okvalificerade personer att utföra mer eller mindre avancerade attacker.

Tiden mellan upptäckt, publikation och korrigerande åtgärder är kritisk för huruvida någon skall lyckas med sina angrepp. Om någon lyckas upptäcka ett säkerhetsproblem och sedan nyttja detta i tysthet har angriparen större sannolikhet att lyckas än om samme angripare istället använt sig av ett flera år gammalt och väl publicerat säkerhetsproblem.

Tidens betydelse för hur stor konsekvens ett säkerhetsproblem innebär, har gjort att det i säkerhetsdiskussioner ibland talas om nolldagarshål (eng. *zero-day exploit*) eller nära-nolldagarshål (eng. *near zero-day-exploit*) när någon pratar om angrepp som består av tidigare okända angreppssätt eller via nya, oskyddade säkerhetsproblem. Ett nolldagarshål inneär att det i princip inte finns några kända skydd eller temporära ändringar som är kända för problemet, främst på grund av att programleverantören inte haft förvarning eller kännedom om säkerhetsproblemet innan information om det offentliggjorts. Inte sällan florerar det både äkta och påstådda attackverktyg för olika jungfruattacker eller opublicerade säkerhets som enbart distribueras i färdigkompilerad form och ibland är krypterade. Det har förekommit ett antal gånger att dessa verktyg i sig innehåller gömd kod och trojanska hästar, så man gör bäst i att undvika dessa eller vara väl medveten om riskerna att testa dessa program.

Olika projekt såsom *Month of Browser Bugs*[27]*, Month of Apple Bugs*[28]*, Month of Kernel Bugs*[29]*,* med flera, har varit olika inlägg och påverkansinslag för att påtagligt och demonstrativt peka på säkerhetsbrister i olika applikationer och operativsystem. Det anses av många vara kontroversiellt att sprida information om säkerhetsproblem utan att något skydd eller motmedel finns framtaget.

Andra personer är anhängare av principen att fullständig öppenhet visavi säkerhetsproblem (eng. *full disclosure*) skall råda i diskussioner om fel, brister och säkerhetsproblem. En medelväg som fått anhängare är olika typer av såkallat *ansvarstagande informationsspridnin*g i samband diskussioner om säkerhetsproblem, där informationslämnaren givit leverantörer och andra ansvariga en viss tidsmässig respit att kunna åtgärda säkerhetsproblemen innan informationen sprids publikt.

2.4.1 Trafikanalys

> **Definition 23:** Trafikanalys (eng. *traffic analysis*) är en vetenskap som baseras på att man analyserar trafikflöden och letar efter mönster eller annan kunskap som kan användas för att införskaffa övergripande information.

Exempel på information som kan vara intressant att ta fram via trafikanalys är vem som kommunicerar med vem, alltså avsändar- och mottagaradresser, och hur mycket data som utväxlas. Det kan handla om analys av storleken på paket likväl som trafikintensitet. Tidsmässiga aspekter kan också vara viktiga, alltså när kommunikationen sker, under hur lång tid den pågår och hur lång tid det tar mellan trafik åt det ena respektive andra hållet (till exempel frågor och vändande svar). Dessutom kan relationer till andra, yttre händelser vara viktiga, om kommunikationen till exempel föregås eller följs av någonting speciellt.

Ett klassiskt exempel på trafikanalys från verkliga livet är analysen av antalet kvällsbeställningar på pizza till Pentagon i Washington. Det sägs att man, före officiella

[27] http://browserfun.blogspot.com
[28] http://projects.info-pull.com/moab
[29] http://projects.info-pull.com/mokb

kommentarer eller presskonferenser, genom att observera pizzabeställningarna kan förutse huruvida USA håller på med krigs- och krisoperationer.

Skyddet mot trafikanalys kallas för trafikflödesskydd och kan bestå av utfyllnad av data (eng. *padding*), simulerad trafik för ökad trafikintensitet eller omdirigerad trafik.

2.4.2 Avlyssning

En grundläggande risk vad det gäller all infomationsbearbetning är att information kommer i orätta händer. Ett sätt för information att läcka ut är genom avlyssning.

Definition 24: Avlyssning är otillåten eller oönskad åtkomst till information.

Definition 25: Passiv avlyssning är avlyssning som utförs utan hjälp av aktiv påverkan på själva meddelandetransporten

Definition 26: Aktiv avlyssning är avlyssning som utförs med hjälp av aktiv påverkan på själva meddelandetransporten

Vid passiv avlyssning sker ingen direkt påverkan på inblandad utrustning, såsom nätverkskomponenter, utan informationen kan inhämtas ändå på grund av det naturliga informationsflödet.

Vid aktiv avlyssning kan nätverksutrustning såsom switchar, eller så kan trafikstyrnings- och routinginformation eller liknande manipuleras för att datattrafik som skall göras tillgänglig för avlyssning skall bli tillgänglig för avlyssningsanordningen. Andra sätt att aktivt påverka trafikflödet är att manipulera ARP-tabeller, göra så kallade ICMP-redirects, manipulera routingprotokoll och liknande. Det skydd man trott växeln ger, kan lätt bli undanröjt vid en aktiv avlyssning.

I datorsammanhang kan avlyssning exempelvis vara tangentbordsinmatning[30], av information som skickas via nätverk[31] eller datapaket som skickas via USB-bussen[32]. Information kan avlyssnas även i den fysiska världen exempelvis med telefonavlyssning, utplacerade mikrofoner eller på distans via en laseravlyssnare[33].

Avlyssning kan ske storskaligt, systematiskt och med avsevärda resurser. Det har det förekommit avlyssning under alla tider. Dokumenterade fall existerar där länder, kyrkan, furstar, underrättelsetjänster, företag och brottsliga organisationer eller enskilda har använt avlyssning för att komma åt information.

Så kallade svarta kammare, allmänt kallade *cabinet noir*, speciella rum där avlyssningsverksamhet har bedrivits, har sedan medeltiden funnits i anslutning till postkontor och posthantering i alla länder där politiska eller religiösa intriger förekommit.

[30] *KeyGhost, en hårdvarubaserad tangentbordsavlyssnare.* http://www.keyghost.com/

[31] *Cain & Abel nätverksbaserad lösenordsavlyssnare.* http://www.oxid.it/cain.html

[32] *USBSnoop. Verktyg för att avlyssna USB-bussen.* http://sourceforge.net/projects/usbsnoop/

[33] *LASER-2000 Laser Room Monitoring System*. http://www.electromax.com/laser.html

Operation Shamrock[34] var en avlyssningsoperation ledd av amerikanska underrättelsetjänsten. Avlyssningsaktiviteterna inom Shamrock pågick nära nog 30 år mellan augusti 1945 och maj 1975. Den militära censurlagstiftningen och myndigheternas möjlighet att avlyssna post och telegram upphörde i samband med andra världskrigets avslut, vilket ledde till att åtkomst till ett oinskränkt informationsflödet inte längre var möjligt för underrättelsetjänsten. Genom att vädja till patriotism hos företagsledarna slöts en frivillig överenskommelse mellan telegramföretagen RCA Communication, Western Union Telegraph Company och ITT Communication och underrättelsetjänsten SSA, Signal Security Agency. En särskild rutin inrättades genom vilken telegramföretaget lät SSA så kopior eller särskild åtkomst till internationella telegram. Avlyssingen avslöjades i mitten av 1970-talet och innebar en stor rättsskandal för de inblandade.

Det amerikanska underrättelsetjänsten NSA har länge misstänkts vara inblandad i ett avlyssningssamarbete vid kodnamn ECHELON. Följande text är hämtad från en europaparlamentsrapport[35]:

> Det kan inte längre betvivlas att det finns ett globalt system för avlyssning av kommunikation, som bedrivs i samverkan mellan Förenta staterna, Förenade kungariket, Kanada, Australien och Nya Zeeland inom ramen för UKUSA-avtalet. Mot bakgrund av de indicier som föreligger och de många samstämmiga yttranden från väldiga olika personer och organisationer, bland annat från amerikanska källor, kan det antas att systemet eller delar av det åtminstone under någon tid gått under kodnamnet ECHELON.
>
> Det kan nu inte råda något tvivel om att syftet med systemet är avlyssna kommunikation från privatpersoner och företag, och inte militär kommunikation, men den analys som genomförts för betänkandet har visat att systemets tekniska möjligheter sannolikt inte alls är så stora som antagits i vissa massmedier.

Bland modernare exempel på avlyssning kan franska underrättelsetjänsten DGSE:s samarbete med Air France nämnas. Avlyssningsmikrofoner upptäcktes i förstaklasskabiner i flygplan som tillhör Air France. Avlyssningen riktades mot affärsmän.

Man kan utgå ifrån att underrättelsetjänster i dag fortsätter att utöva avlyssning av vänligt inställda länder och organisationer eller företag ifrån dessa länder. NSA har exempelvis anställda med språkkunskaper i danska, nederländska, kuwaitiska, norska och svenska.

2.4.3 Uppträdande under falsk identitet

> **Definition 27:** Förfalskning av information eller uppträdande under falsk identitet (eng. *spoofing*) är metoder för att lura system att acceptera kommunikation från en obehörig avsändare.

Förfalskning av identitet kan ske på många olika nivåer i en kommunikationsstack. Några exempel på Ppopulära metoder av spoofing är

- förfalskad hårdvaruadress (MAC-adress spoofing)
- förfalskat mottagande domännamn (DNS-spoofing),
- ARP-spoofing
- avsändande nätverksadresser (IP-spoofing)
- hyperlänkar med webbadresser (URL-spoofing) och liknande.

---

[34]sid 302-315 i Bamford, James (1983). *The Puzzle Palace: A Report on America's Most Secret Agency*. Penguin Books. ISBN 0-140-06748-5

[35]Europaparlamentet (2001) *Rapport från Europaparlamentets tillfälliga utskott om ECHELON 2001*. s. 12. http://www.europarl.eu.int/committees/echelon_home.htm

URL-spoofing är vanligt i olika bedrägerisammanhang, såsom vid phising-attacker där falska e-postutskick från banker eller liknande innehåller webbadresser som i löptext förefaller peka på en korrekt webbplats, men när länken aktiveras kopplar till en annan adress.

Förutom ren spoofing kan man även råka ut för scenarion och attacker där någon har satt upp en vilseledande webbtjänst som i så stor utsträckning skall se ut och fungera som originalwebben. Orsaken kan vara att någon vill tillskansa sig information eller pengar.

2.4.4 Återuppspelningsattack

En enkel och effektiv attack är den så kallade återuppspelningsattacken (eng. *replay attack*). I vissa protokollimplementationer och vid viss protokollformgivning kan man gå runt säkerhetsmekanismerna genom att spela in data och sedan åter spela upp dem.

Med en återuppspelningsattack kan man till exempel göra en inloggning med ett inspelat, krypterat lösenord utan att ha kunskap om klartextlösenordet,eller upprepa en transaktion flera gånger så att mer pengar överförs.

För att skydda sig mot återuppspelningsattacker kan man i protokoll bland annat använda sig av tidsstämplar, serienummer och sekvensnummer. Dessa parametrar används i kombination med att man bygger in kontroller i programvaran för att registrera vilka sekvensnummer som använts, vilka tidsomfång som är tillåtna för stunden och liknande. Sekvensnummer, tidsstämplar och liknande bör skyddas av kryptering och intregritetsskydd.

2.4.5 Buffertöverskrivning

> **Definition 28:** Buffertöverskrivningsattacker är samlingsnamnet på en mängd olika attacker som utförs genom att en kombination av data och maskinkod skrivs i och utanför en databuffert, där det injicerade programkoden sedan fås att exekveras.

Om databufferten ligger sparad i vissa delar av datorns minne, tex på stacken eller i det område som kallas för heap, kan vissa viktiga parametrar manipuleras, till exempel återhoppsadressen för en subrutin. Genom att skriva in ett förändrat värde på denna återhoppsadress kan man få datorn att anropa en programsekvens som installerats utan lov.

Buffertöverskrivningsattacker utförs bland annat mot inläsningsfunktioner och bearbetningsfunktioner i nättjänster. Genom att via nätverket sända in maskinkod och data går det därmed att fjärrexekvera och fjärrstyra ett system, till exempel en webb- eller namnserver.

En buffertöverskrivningsattack kan i teorin förefalla avancerad och kräva omfattande kunskaper i hur en kompilator och länkare genererar kod, var någonstans i objekt- och binärkod en viss rutin och dess tillhörande dataarea finns, var någonstans i minnet den kommer att laddas och liknande. I praktiken finns det emellertid mängder av detaljerade beskrivningar idag om hur en buffertöverskrivning går till, till och med massiva läroböcker[36] om hur man själv skall kunna finna de latenta buffertöverskrivningshålen i program. Det finns också mängder av exempelprogram och så kallad skelettkod för vanliga maskinvaruarkitekturer såsom X86, Alpha, SPARC och HP-PA.

[36] The Shellcode Hackers Handbook

Det finns flera sätt att skydda sig mot buffertöverskrivningsattacker. Man bör till exempel kontrollera indata och filtrera bort de data som inte är tillåtna. Bufferthanteringen bör vidare inkludera en kontroll av att det inte skrivs in mer data än vad bufferten klarar av att hantera. En annan skyddsmekanism är att se till att kompilatorer genererar speciell kod som inför särskilda kontrollsummor på stacken som kontrolleras för överskrivning innan subrutiner anropas eller returnerar data. Dessutom bör de minnesareor som används som stackutrymme märkas så att de inte är exekverbara, alltså att det inte går att köra program från dessa delar av minnet. Detta är skydd som måste byggas in i operativsystemet. Vissa operativsystem, såsom Windows Vista och OpenBSD, har dessa skydd. På engelska kallas denna typ av attacker ofta för buffer overrun, stack overrun, stack smashing eller liknande.

Det finns flera varianter av denna typ av attack. En relativt nyupptäckt variant är missbruk av formatsträngar i utmatningsfunktioner (eng. *format string attack*) i vissa programspråk, främst C. Exempelvis så sker utmaning vanligtvis via funktionen printf enligt nedanstående princip:

```
printf(”%s\n”, variabel);
```

Där %s är ett formatterings- och datadirektiv till printf-funktionen. Men ibland utelämnar en slarvig programmerare detta direktiv, med resultatet:

```
printf(variabel);
```

Angreppet innebär att man formar indatat på ett speciellt sätt så att en datadriven attack leder till att man kan lura printf att adressera minnesområden och data som normalt sett inte skulle använts.

En annan besläktad attack är integer overflows där ett heltal kan fås att tolkas annorlunda av olika programdelar. I programspråk som C och C++ så definieras heltal som positiva (eng. unsigned) heltal eller teckesatta (eng. signed) heltal. Genom att manipulera siffervärden kan man skriva utanför en definierad variabellista, få minnesallokeringsfunktioner att fungera felaktigt och andra liknande angrepp.

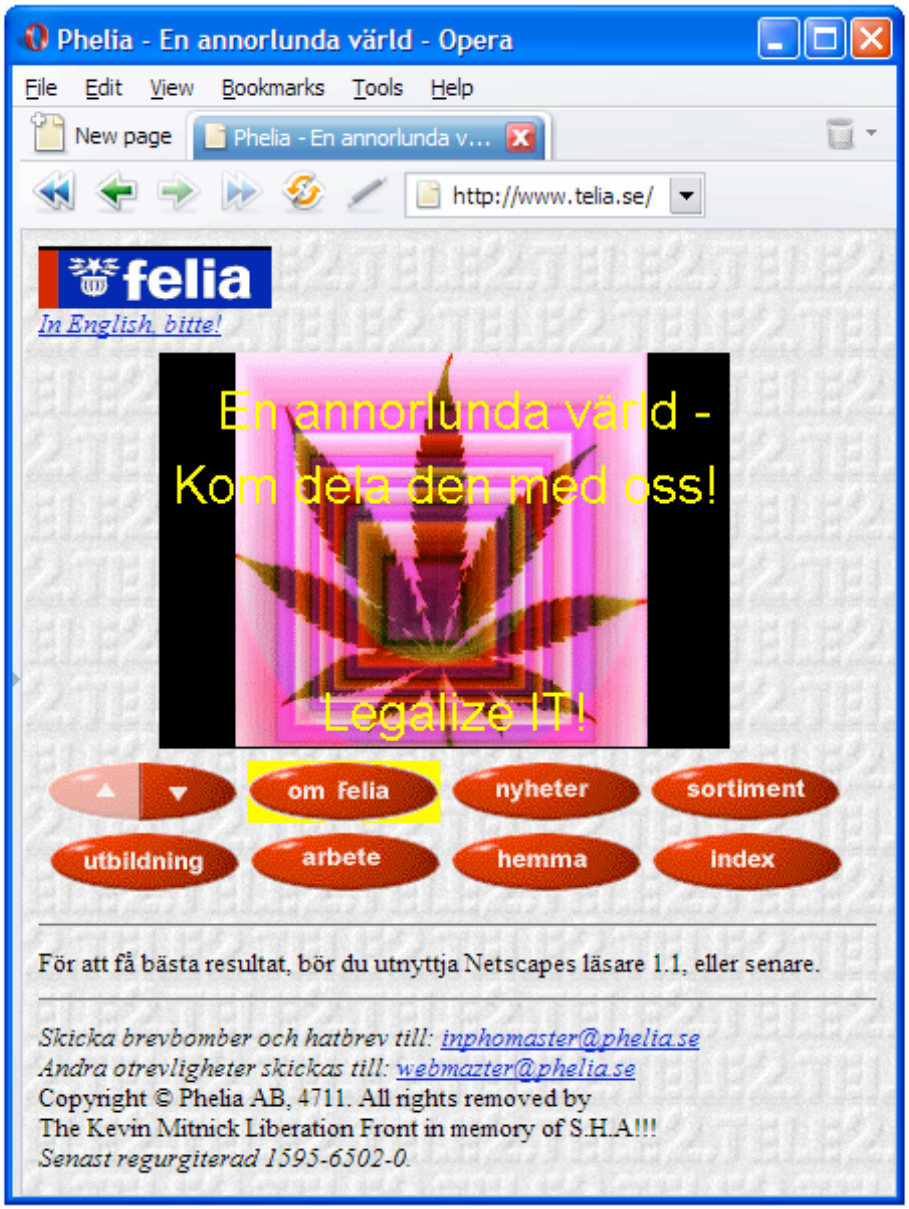
En till variant av bufferöverskrivning är så kallad heap smashing, det anmärkningsvärda med denna överskrivningsmetod är att den arbetar mot en annan del av arbetsminnet och med andra metoder, vilket gör de enkla skydd och tester som utförs mot stacksmashing inte är tillämpbara.

2.4.6 Intrång

Definition 29: Intrång är en metod för att obehörig tillgång till information eller IT-resurser

När områden som IT-säkerhet och nätverkssäkerhet är intrång en av de första konsekvenserna som folk som regel tänker på. Att någon tar sig in i ett system och där utför en eller fler gärningar.

Dataintrång kan delas upp i förstörande intrång och icke-förstörande intrång. Ett exempel på ett icke-förstörande intrång är att någon har olovligen kommit över användarnamn och lösenord för att sedan via nätverket logga in och läsa dokument som personen normalt sett inte haft tillgång till. Ett exempel på ett förstörande intrång är obehörligen förändrade webbsidor (eng. *defacements*) eller att någon raderar filer i det system som denne tagit sig in i.



Figur 5 Förändrad webbsida hos telia vid intrång

Dataintrång definieras i Brottsbalken 4:9 avsnitt c.

> 9 c § Den som i annat fall än som sägs i 8 och 9 §§ olovligen bereder sig tillgång till upptagning för automatisk databehandling eller olovligen ändrar eller utplånar eller i register för in sådan upptagning döms för dataintrång till böter eller fängelse i högst två år. Med upptagning avses härvid även uppgifter som är under befordran via elektroniskt eller annat liknande hjälpmedel för att användas för automatisk databehandling. Lag (1998:206).
>
> 10 § För försök, förberedelse eller stämpling till människorov, människohandel eller olaga frihetsberövande och för underlåtenhet att avslöja sådant brott döms till ansvar enligt vad som sägs i 23 kap. Detsamma gäller för försök eller förberedelse till olaga tvång som är grovt eller till dataintrång som om det fullbordats inte skulle ha varit att anse som ringa. Lag (2004:406).

2.4.7 Omdirigeringsattack

Omdirigeringsattacker (eng. *redirection attacks*) på nätverksnivå kan ske med hjälp av en mängd olika attacker. Det kan vara angrepp mot IP-routinginformation men det kan också vara en ARP-spoof-attack, att man skickar ICMP redirect eller en attack mot underliggande transmissionsutrustning såsom ATM-växlar, Frame Relayutrustning och liknande. Attacker direkt mot routers eller kommunikationsutrustning kan leda till att angriparen kan ändra i routingtabeller, uppsatta kretsar eller logiska portar.

2.4.8 Överlastning och resursmissbruk

Överlastningsattacker (eng. *Denial-of-Service, DoS*) hör till de lättare attackerna att utföra, samtidigt som det tillhör den svåraste klassen av attack att kunna skydda sig mot. Bland överlastningsattacker kan nämnas

- Stort antal oönskade nätverksuppkopplingar.
- Halva uppkopplingar som tar slut på köplats för ännu inte bearbetade uppkopplingar
- Stora paket som tar all tillgänglig bandbredd

Det går också att skicka fragmenterade paket som när de återskapas i målsystemet bildar onaturliga paketformationer, till exempel konstiga överlapp eller omöjligt stora paket. När paketen återskapas så uppstår problem som får system att krasha eller fungera oberäkneligt. Ett exempel på detta är *”ping-of- death*”, en metod där man använder sig av ett ping (ICMP echo request) där testpaketet gjorts större än den största tillåtna paketstorleken. Många nätverksbaserade DoS-attacker använder sig av förfalskade avsändaradresser på nätverkspaketen för att försvåra eller omöjliggöra spårning. För nuvarande pågår aktiv forskning för att hitta olika metoder för att förhindra
och spåra DoS-attacker.

En modern och mycket effektiv attackmetod är den distriberade tillgänglighetsattacken (eng *Distributed DoS attack. DDoS*). Det finns flera verktyg, till exempel TFN, Trin00 och Stacheldraht (”ståltråd” på tyska), som har använts med framgång för att utföra massiva attacker. Distribuerade DoS-verktyg består ofta av flera nivåer av program, bland annat agentprogram (även kallade slavar eller zombier) och olika typer av styrprogram.

Distribuerade tillgänglighetsförlustattacker bygger ofta på flera samtidiga, enkla DoS-attacker eller kombinationer av olika typer av DoS-attacker. Förutom den större slagkraft som uppnås med en distribuerad DoS-attack, så innebär användningen av flera nivåer också att en spårning blir svårare.

2.4.9 Mannen-i-mittenattacker

Mannen-i-mittenattacker kallas ibland också för MITM (eng. *man-in-the-middle*). Genom att sitta mitt i mellan två kommunicerande parter, antingen man är logiskt placerad där eller genom att man lyckats omdirigera trafik mellan två system via ett tredje system kan det tredje systemet utföra en mängd andra attacker såsom avlyssning, datamanipulation, mannen-i-mitten-attacker, DNS-spoofing och liknande. Ett sätt att utföra omdirigeringsattacker är att angripa vägvalsinformation som sänds via routingprotokoll, att skicka falsk ARP-information eller påverka en switch genom att skicka tillräcklig mängd påhittade ARP-svar så att switchen ändrar beteende och övergår till att fungera som en brygga på just den aktuella porten.

Det finns lättillgängliga hjälpmedel för att utföra vissa mer avancerade mannen-i-mittenattacker på nätverksnivå. Två exempelprogram är sshmitm och sslmitm.

2.4.10 Race conditions

Ett race-tillstånd (eng. *race-condition*) inträffar när ett program, eller operativsystemet självt, utför en så kallad icke-atomär serie operationer, den första en kontroll och därefter ett svar som man agerar utifrån. Med icke-atomär menas här att operationen kan fördröjas eller temporärt avbrytas genom ett avbrott (eng. *interrupt*) i systemet.

Dessa fördröjningar mellan kontroll och utförande leder till möjligheten att bakom kulisserna utföra ändringar på det kontrollerade objektet i tiden mellan kontroll och agerande. Ett exempel kan vara att symboliska länkar i filsystemet pekas om efter det att ett program kontrollerat filstatus på länken men innan det börjar läsa eller skriva data till filen som länken pekar på. Om någon ändrar den fil länken pekar på i tidsmellanrummet mellan kontroll och läs/skrivoperation kan ett program fås att agera på fel fil, till exempel en fil till vilken man normalt inte har åtkomst.

Viktiga operationer som kan drabbas av race-condition måste därför implementeras och hanteras på ett atomärt sätt. Det innebär att system- och biblioteksanrop inte kan bli temporärt avbrytna från det att kontrollen startas till det att operationen på objektet avslutats. Ett sätt att hantera detta är att stänga av avbrottshanteringen i operativsystemkärnan för vissa känsliga operationer.

Ett annat namn som ibland förekommer för race-condition är kapplöpningsföreteelse.

2.4.11 Datamanipulation

Genom datamanipulation sker otillbörlig ändring av ursprungsinformation. Ett belopp i en penningtransaktion kan till exempel ändras. I kommunikationssammanhang tänker vi ofta på manipulation av data som är under befordran. Manipulationen kan vara tillägg, utradering eller förändring av någon form av data. Det främsta skyddet mot datamanipulation är sekretessskydd i form av kryptering och starka kontrollsummor som kan påvisa ett meddelandes förväntade integritet.

2.4.12 Trunkeringsattack

En trunkeringsattack är en specialvariant av datamanipulation där man genom att radera delar av information, till exempel ett paket under befordran, lyckas påverka resultatet, till exempel minska beloppet i en transaktion.

2.4.13 Omarrangeringsattack

I en omarrangeringsattack (eng. *re-arrangement attack*) flyttas data, nätverkspaket eller liknande om så att effekten av datats användning förändras. Ett exempel på en omarrangeringsattack är en omkastning av kontonumren för uttag och insättning. Detta kan till exempel göras om kontonumren går i separata nätverkspaket eller om data sänds i oskyddade (alltså okrypterade, icke integritetsskyddade) protokoll.

Datats ordningsföljd är också en viktig parameter vid till exempel transaktioner. Om en uttag skall föregås av en insättning kan en omkastning av dessa transaktioner leda till katastrofala konsekvenser. Ett skydd mot mot omarrangeringsattacker är att varje datapaket får

sekvensnummer i kombination med kryptografiskt starka kontrollsummor som skyddar sekvensnumren från manipulation.

2.4.14 Klipp-och-klistra-attack

Man kan i vissa fall attackera krypterad information utan att ha kunskap om den egentliga klartexten. Genom att använda meddelanden som krypterats med samma nyckel, men som saknar dataintegritetskontroll (kryptografiskt starka kontrollsummor) kan man under vissa förutsättningar klippa ut krypterad information och klistra ihop till nya meddelanden.

2.4.15 Övertagande av session

En mer avancerad attack är ett *övertagande av en etablerad session*. När någon väl har skapat en nätverkkoppling, till exempel efter inloggning och autentisering, så kan någon kapa sessionen. En allvarlig konsekvens av kapningsmöjligheten är att sessioner kan övertas även om man använt sig av uppsäkrade varianter av autentisering, till exempel engångslösenord eller elektroniska signaturer.

Kapningen sker ofta på en låg nivå i vilken man sänder förfalskade paket till inblandade parter. För att lyckas med attacken påverkas TCP-stackens tillståndsmaskiner (eng. state-machine) i bägge kommunikationsändar så att man lämnar ett synkroniserat tillstånd. I ett synkroniserat tillstånd har de kommunicerande parterna samma uppfattning om sekvensnummer, antalet utestående paket och liknande. När man skapar ett desynkroniserat TCP-tillstånd mellan de bägge parterna kommer de inte längre att kunna prata med varandra på grund av att de inte längre är överens om vilka sekvensnummer, kvittensnummer och liknande som är aktuella.

Det program som används för att skapa det desynkroniserade tillståndet kan dock ha kunskap om vilka som är de nu aktuella sekvensnumren, kvittensnumren och motsvarande för alla inblandade. Därefter kan programmet sända ut förfalskade paket med parametrar som accepteras av den server det försöker kapa.

Flera tillgängliga verktyg existerar sedan många år för att automatisera kapningen. Med hjälp av dessa verktyg kan personer utan teknisk kunskap eller programmeringskunskap utföra avancerade attacker.

Skydd mot attacker av denna typ är att inte använda klartextprotokoll, utan att använda protokoll som använder sig av kryptering för att skapa sekretess (insynsskydd) och dataintegritet (manipulationsskydd). Andra namn är som förekommer på denna attack, eller varianter av den, är kapande av session, *TCP-hijacking* och *IP-splicing*.

2.4.16 Datadrivna attacker

Datadrivna attacker är attacker som är utformade så att kommandon smygs in i dataströmmen och tolkas som kommandon av mottagarsidan. Exempel på tjänster där datadrivna attacker har varit vanliga är befordran av elektronisk post. Posthanteringsprogrammet sendmail har haft flera kända säkerhetsproblem där unixkommandon kunnat skickats med i meddelandekroppen för att sedan tolkas av mottagarprogrammet. Detta har exekverat kommandona istället för hantera dem som strikt information inkluderad i e-posten.

Ett område inom datadrivna attacker som börjat få mycket uppmärksamhet är attacker där man injicerar programkod som en del i en inmatning till exempelvis en webbsida. En vanligt förekommande variant är den såkallade *SQL injection-attacken*. Genom att skicka in SQL-kod vid en inmatning, via en URL eller liknande så kan man styra databaser på serversidan. Det är tyvärr alltför vanligt att serversidan inte filtrerar bort styrtecken och programsatser utan släpper dessa vidare rakt in till databasmotorn. Med denna typ av attack så kan man utföra mängder av angrepp, det går att förbikoppla autentiseringsfunktioner, skapa informationsläckage där man får tag i valfritt data i databasen, ordna driftstörningar eller sabotera och manipulera datainnehåll.

Det finns allt från enkla SQL-injiceringsattacker till väldigt avancerade attacker som är svåra att upptäcka vid granskning och säkerhetstester. För exempel på avancerade attacker se[37] [38]

Cross-site scripting, ofta förkortat CSS eller XSS, är en annan attackmetod som svingat upp och blivit en av de mer vanliga metoderna för attacker mot webb. Denna attackmetod innebär att en användares webbläsare i smyg blir erbjuden eller påtvingad ett skadligt skript eller program från en webbtjänst. Detta program kan användas för att stjäla information, övervaka användarens förehavanden, förstöra information eller på annat sätt utföra attacker.

2.5 Att leta säkerhetshål

2.5.1 Security by obscurity

Ett vanligt förekommande problem inom säkerhetsområdet är vad man kallar ”*Security by obscurity*”. Denna term kommer av att man istället för att skapa reell säkerhet försöker skydda systemet genom att hemlighålla kunskap om system och tillämpningar. Man hoppas med andra ord att säkerheten stärks genom att man döljer något, till exempel hur en viss säkerhetsfunktion är implementerad.

[37] Anley, Chris ”Advanced SQL Injection In SQL Server Applications”
http://www.ngssoftware.com/papers/advanced_sql_injection.pdf
[38] Anley, Chris ”(more) Advanced SQL Injection In SQL Server Applications”
http://www.ngssoftware.com/papers/more_advanced_sql_injection.pdf

Figur 6 Security-by-obscurity - vad hjälper egentligen gallergrinden här?

Att enbart bygga sitt skydd på obskyritet är en fälla som vissa går i. Att inse att obskyritet som bäst kan ses som en vägbula (smygande polis) och därmed låta oss vinna lite tid från angriparen – en fördröjande skyddsåtgärd - är en viktig insikt. Därmed kan man eventuellt tänka sig att bygga lösningar som har starka, verkliga skydd men som dessutom ges ett viss extra skydd med hjälp av obskyritet. Dock är det djupt felaktigt att tro att någon inte kan klura ut hur ens skyddsmekanismer är skaffade, hur ens egenutvecklade protokoll fungerar eller liknande.

2.5.2 Källkodsgranskning

Källkodsgranskning är en metod som innebär att man gör en genomsökning av programmets ursprunglig källkod för att leta efter programmeringsfel som kan leda till sårbarheter. Genomsökandet kan vara manuellt i form av okulärbesiktning, med hjälp av olika verktyg för att automatiserat leta efter sårbarheter eller kombinationer av dessa metoder.

En metod att komma åt den kunskap som källkodstillgång innebär är att stjäla källkod. Flera uppmärksammade fall av åtkomst av källkod har nått offentligheten under årens lopp, exempelvis Kevin Mitnick:s stöld av källkoden till VAX/VMS[39], källkoden till Microsofts Windows 2000[40] som postades till P2P-nätverk och den svenska hackern Rebell:s stöld av källkoden till Cisco:s centrala styrprogramvara IOS.

2.5.3 Baklängeskonstruktion

Ett sätt att gå runt skydd uppbyggda med hemlighållna rutiner är att göra ingående observationer och försöka återskapa ursprungsfunktionen genom så kallad baklängeskonstruktion (eng. *reverse engineering*).

[39] http://all.net/journal/50/hacks.html
40 Microsoft Investigates Leak of Windows Source Code http://www.betanews.com/article/1076632515

Olika sätt att komma åt en gömd, obskyr säkerhetsfunktion är att på något sätt ta fram den gömda funktionen. Det finns mängder med sätt att göra detta, man kan dissasemblera (göra om ifrån maskinkod till läsbar assemblerkod), dekompilera (göra om till källkod), följa ett programs exekvering med hjälp av en debugger, använda nätverkspaketanalysatorer, stjäla källkod eller få tag i ej offentliga beskrivningar och dokument. Det existerar många verktyg, såväl standardprogram som specialskrivna program, som går att nyttja för att analysera program eller nätverkstrafiks hemligheter. Populära verktyg för att analysera program i Windows-miljö är SoftICE, IDA Pro och OllyDbg. Då alltmer fokus har kommit att komma på baklängeskonstruktion och dess betydelse för att leta efter säkerhetshål och sårbarheter, har på senare år metoder och verktyg förfinas med flera grader. Exempel på anpassade verktyg för säkerhetsforskning och sökande av sårbarheter är makropaket till debuggers eller specialverktyg som jämför patchar och originalfiler för att snabbare och enklare se vilka delar av koden som blivit förändrade med hjälp av patchen. Detta förenklar arbetet med att lista ut vilken sårbarhet som leverantören lagat med patchen.

2.5.4 Fuzzing

En metod som på senare år kommit starkt inom IT-säkerhetsforskning med inriktning att hitta säkerhetshål är såkallad *fuzzing* eller fuzztestning.

> **Definition 30:** Fuzzing är en testteknik som innebär att man genererar oförväntad, slumpdata, som inmatning till ett program eller en programkomponent, för att därmed upptäcka sårbarheter som beror på okontrollerad indatahantering

Fuzzing uppstod som ett begrepp när ett forskningslag i början av 90-talet använde sig av (semi)slumpmässig inmatning som en metod för att kontrollera programkvalitet på diverse välkända och sedan länge etablerade unixverktyg[41]. Resultatet av testerna var förvånande, cirka en tredjedel av alla testade program gick att krasha. Ännu mer förvånande var att resultaten var lika nedslående när testerna gjordes om i senare omgångar[42].

Efter dessa enklare testningar med slumpdata har mer avancerade testformer och verktyg utvecklats som bygger på fuzzingkonceptet. Från enkla specialverktyg för test av enstaka program, har utvecklingen numera gått mot flexibla ramverk, som exempelvis PROTOS[43] eller kommersiella verktyg som defensics[44], som går att utöka med skriptspråk, och som klarar mycket komplexa testfall.

Fuzzing som testmetod kan användas mot inmatning från tangentbord, från kommandoraden, via nätverksgränssnitt, från filinläsning, etc. Det har används för att testa nätverksprogram såsom webbservers och SNMP-tjänster, RPC-interface, nätverksstackar, operativsystemskärnor, etc.

[41] B.P. Miller, L. Fredriksen, and B. So, "An Empirical Study of the Reliability of UNIX Utilities", *Communications of the ACM* **33**, 12 (December 1990). ftp://ftp.cs.wisc.edu/paradyn/technical_papers/fuzz.pdf
[42] http://pages.cs.wisc.edu/~bart/fuzz/
[43] http://www.ee.oulu.fi/research/ouspg/protos/
[44] http://www.codenomicon.com/products/index.shtml

2.5.5 Testexekvering samt introspektion

En annan metod för att undersöka säkerhetshål är exekvering av programkod i sandlådemiljöer, under debuggerprogramvaror eller i simulerade miljöer. Inte sällan utvecklar avancerade angripare egna varianter av DLL:er, applikationsprotokollstackar och liknande för att få bättre kontroll över alla delar. Med hjälp av en instrumeterad bluetoothstack, IP-stack eller liknande går det att med skript eller olika API-anrop agera och interagera med det program som är under uttestning.

2.6 Andra typer av tekniska hot

Med tekniska hot menas här hot som innefattar, beror på eller innebär konsekvenser för tekniska lösningar, och med tekniska lösningar fokuserar vi i denna text på IT-tekniska och framförallt nätverksrelaterade hot.

2.6.1 Övertro till tekniska lösningar

Många personer har en övertro på tekniska lösningar och att dessa kommer att lösa alla säkerhetsproblem. Ett antivirusprogram tros klara av alla virusproblem, en brandvägg antas eliminera alla nätverksproblem. Självfallet kan inte tekniska system lösa alla säkerhetsproblem, utan tekniken hjälper i första hand till genom att man får verktyg och stödsystem för att bättre följa policies, riktlinjer, avtal, lagar och liknande. Man måste undvika den självbedrägliga tanken att säkerhet är något man kan kan köpa på burk, att de investerade pengarna för en pryl eller programvara innebär att man kan bocka av i kanten och gå vidare. Ofta är ett inköp eller utvecklingsprojekt bara det första lilla steget på en väldigt lång och slingrande resa.

Ett annat vanligt problem är att man inte utför ett tillräckligt förarbete genom att ställa krav och att specificera kraven. Alltför ofta står det ”säker” som en punkt på kravlistan utan närmare detaljer. Med vems definition skall man tolka det kravet? Det kan leda till att man inte vet tillräckligt mycket om de produkter som införskaffas eller byggs - och att man tror sig ha bättre lösningar än man egentligen har.

2.6.2 För hög grad av integration

För tio, tjugo år sedan var det fortfarande ovanligt att datorer var sammankopplade i nätverk. Senare har man i allt större utsträckning kopplat samman datorer i allt större och större datornätverk. Det som började som små, mer eller mindre isolerade öar av nätverk, är idag sammankopplade enheter i världsomspännande koncernnätverk. Denna ökade integration innebär att det kan bli avsevärt större konsekvenser när säkerhetsproblem uppstår. Till exempel kan nyskapade virus, maskar eller missbruk av ett säkerhetshål i ett populärt program spridas allt snabbare eller nå fler system än tidigare. Integrationen innebär ökad nåbarhet, men också nya scenarion och möjligheter för de som planerar att missbruka systemen.

Vid sidan av det som sker på nätverksnivå, så ökar även integrationen mellan olika tillämpningar (programvaror). Ett exempel på detta är Microsofts övergripande strategi, där företaget sedan lång tid tillbaka skapat kraftfullare och mer generella tillämpningar där man lagt till funktioner av typen inbyggda makron, skript- och programmeringsspråk. Därtill har man ökat tillämpningarnas anpassning för nätkoppling, något som ibland kallas ”Internet ready”. Alltför integrerade program kan också innebära risker.

2.6.3 Alltför snabb adaption av nya tekniker

Ibland införs lösningar utan normal eftertanke och utan någon som helst hot- och riskanalys. Ofta upphandlas tillämpningar och system, eller byggs inom ett projekt, utan att man som en del i den normala införandeprocessen har kontroll över de säkerhetskonsekvenser som de nyinförda systemen medför.

Det kan innebära att man skapar nya säkerhetshål eller försvagar befintliga skyddsmekanismer, till exempel öppnar nya vägar in genom brandväggar. Inte sällan har programutvecklare, projektledare, formgivare, inköpare och andra involverade en låg kunskap om säkerhet och vilka säkerhetsmässiga konsekvenser deras handlingar får. System- och nätverksadministratörer och säkerhetsansvariga får helt enkelt rätta sig efter dåliga beslut om formgivning eller programstruktur och ändra i befintliga säkerhetslösningar för att kunna infoga den nya lösningen.

2.6.4 Alltför komplexa lösningar

Ofta utvecklas, byggs eller inhandlas lösningar som är alltför komplexa för att enstaka personer skall få en ordentlig överblicksbild eller ha möjlighet till en god systemförståelse. Detta berör till exempel ofta ekonomi- och logistiksystem. Som regel innebär komplexa lösningar att det är svårt att hålla god säkerhet i lösningen. Att lösningen införlivas i redan existerande strukturer, till exempel en ytterligare extranetlösning som ansluts till ett brandväggsområde, påverkar existerande lösningars komplexitet.

En annan säkerhetskonsekvens av komplexitetsproblemet är hur stort systemet blir, räknat i antalet rader programkod – alltså hur stor källkoden är. Grundhypotesen är att alla program innehåller programfel, buggar. Till denna hypotes gör vi tillägget att ju större program, desto fler buggar. Ett schablonmässigt värde som brukar anges är ett fel per 100 rader utvecklad programkod. Ju fler kodrader, desto större sannolikhet är det för fel – och desto större risk för säkerhetsrelaterade fel. Antalet kodrader i de flesta av dagens tillämpningar uppgår till 10.000, 100.000, eller ibland flera miljoner. Som exempel uppges Windows 2000 bestå av cirka 65 miljoner kodrader.

Komplexa säkerhetslösningar är extra illavarslande då de sannolikt kommer att vara svåröverskådliga, innehålla formgivningsfel och program med buggar. Så enligt hypotesen ovan så måste även säkerhetsprogram innehålla säkerhetshål, exempel på detta ser man ofta på mailinglistor och webbar som publicerar säkerhetshål – antivirusprogram med hål, brandväggar som går att missbruka och så vidare. En grundläggande tes är därför att inte bygga onödigt komplexa brandväggsmiljöer, PKI-lösningar eller användarhanteringssystem (till exempel inloggningssystem).

2.6.5 Ej uppsäkrade system

De flesta leverantörer satsar huvuddelen av resurserna vid utveckling på att göra sina system och tillämpningar lättanvända, ge dem många funktioner, bakåtkompabilitet och liknande. Det innebär att många standardinställningar (eng. *default values*) är satta till värden som optimeras i denna riktning. Ofta är systemets utformning, installation och konfiguration gjorda för att förenkla så mycket som möjligt för användarna. Dessa inställningar –- och denna syn på systemutformning - är ofta på direkt kollisionskurs med ambitionen att skapa en säker lösning. Det kan innebära stora problem om sådana system driftsätts utan en inledande uppsäkring, ibland kallad härdning – där man väljer säkrare inställningar och tar bort onödiga eller osäkra funktioner, program och subsystem. Exempel på problem som uppstår på grund av att man använder standardinställningar i ett system är till exempel att man ofrivilligt kan få

välkända kontonamn med standardlösenord aktiva eller att nättjänster, såsom webbservrar och indexservrar som aldrig kommer att användas, ändå aktiveras automatiskt vid systemstart.

2.6.6 Lättillgängliga attackverktyg

Beskrivningar av olika säkerhetshål, hur man nyttjar dem och vilka verktyg som kan användas är idag väl spridda och lätttillgängliga på bland annat Internet. Tillgängligheten på olika verktyg för att utföra automatiserade attacker mot system har ökat markant de senaste åren i form av program, skripts och källkod. Idag finns dessa tillgängliga i stora strukturerade programarkiv med sökmotorer. Ett sådant arkiv innehåller de flesta verktyg en angripare kan önska sig för att utföra en attack. Denna tillgänglighet gör det möjligt även för teknisk okunniga personer att utföra attacker och till och med iscensätta avancerade, komplicerade eller isoteriska attacker. Dessa okunniga personer som använder attackprogram brukar med ett samlingsnamn kallas för ”*script kiddies*”. Förutom verktyg i olika form så är information alltmer lättillgänglig. Tidskrifter som ”2600 the hacker quarterly”, ”Ha9ing”, ”Phrack” eller liknande beskriver såväl attacker som sårbarheter. Webbar och e-postlistor distribuerar information om hål och i många fall även recept som beskriver hur man nyttjar hålen.

2.6.7 Formgivnings- och implementationsfel

Det är vanligt att programvara, kommunikationsprotokoll och system drabbas av formgivningsfel (eng *design errors*) i samband med utveckling och specifikation. Saker kan också ha formgivits rätt, men senare implementerats fel. Ett exempel på ett sådant formgivningsfel är så kallad ”layering violation”, alltså ogenomtänkta genvägar mellan olika skikt, säkerhetsbarriärer, protokollnivåer, tillämpningsgränssnitt (APIer) och liknande.

Ett annat vanligt misstag är att tillämpa en felaktig säkerhetsnivå i förhållande till hur tillämpningen, protokollet eller systemet ska användas. Det är också vanligt att felaktigt formgivna användargränssnitt leder till att säkerhetsfunktioner glöms bort, inte känns tillräckligt användarvänliga eller inte ens används. Det finns också många exempel på implementationsfel. Informationsläckage kan till exempel uppstå genom att känsliga data lämnas kvar i temporärfiler som inte raderas eller att programvaran inte kontrollerar huruvida en fil redan existerar när den skapar en ny fil med samma namn. Det senare kan leda till säkerhetsluckor där känslig information, såsom användardatabas eller systemkonfiguration, kan skrivas över. Ett tredje exempel på ett implementationsfel är kryptosystem där man använt sig av enkla, förutsägbara pseudoslumptal istället för kryptografiskt starka slumptal.

2.6.8 Skräppost, spam

Skräppost kallas på engelska för spam, junk mail och unsolicitated bulk email (UBE). Skräppost har på kort tid vuxit till att bli ett stort problem för merparten av de personer som använder sig av e-post. Skräppost är också allmänt känt som spam, en slags skinka på burk, efter en Monty Pythonsketch med en obstinat restaurangägare som enbart tillhandahåller rätter innehållande spam, Definitionen av skräppost innehåller oönskad elektronisk reklam, och massutskickat. Från att ha varit ett smärre irritationsmoment har spam blivit ett enormt resursslukande monster som konsumerar diskutrymme, nätkapacitet och framför allt människors tid. Användare och systemadministratörer har tvingats lägga allt mer tid på att rensa bort oönskad post, installerade uppgraderingar, upphandla tilläggsprogram och tilläggstjänster.

Problemen med spam är flerfaldiga:

- Den höga trafikvolymen som påverkar nätkapacitet och lagringsutrymme.

- Detta leder till ökade kostnader för nättjänsteleverantörer, företag och organisationer som har IT-infrastruktur samt för enskilda personer
- Spam drabbar folk urskiljningslöst. Mycket spam har innehåll som många mottagare anser är anstötande eller starkt motbjudande.
- Vissa som nyttjar spam använder hackade datorer som vidaredistributionspunkter, därmed blir intrång en (dold, men väsentlig) del av affärsmodellen för att ha billiga med kapacitetsstarka distributionskanaper.

Många som skickar spam nyttjar missbrukar teknik och människors förtroende. Spammare gömmer sin riktiga identitet, använder oetiska eller olagliga metoder för distribution och marknadsföring. Ofta tillhandahåller inte spamleverantören korrekta avsändaradresser, skickar e-post via hackade datorer eller e-postsystem som okritiskt vidarebefordrar brev (s.k. öppna reläer).

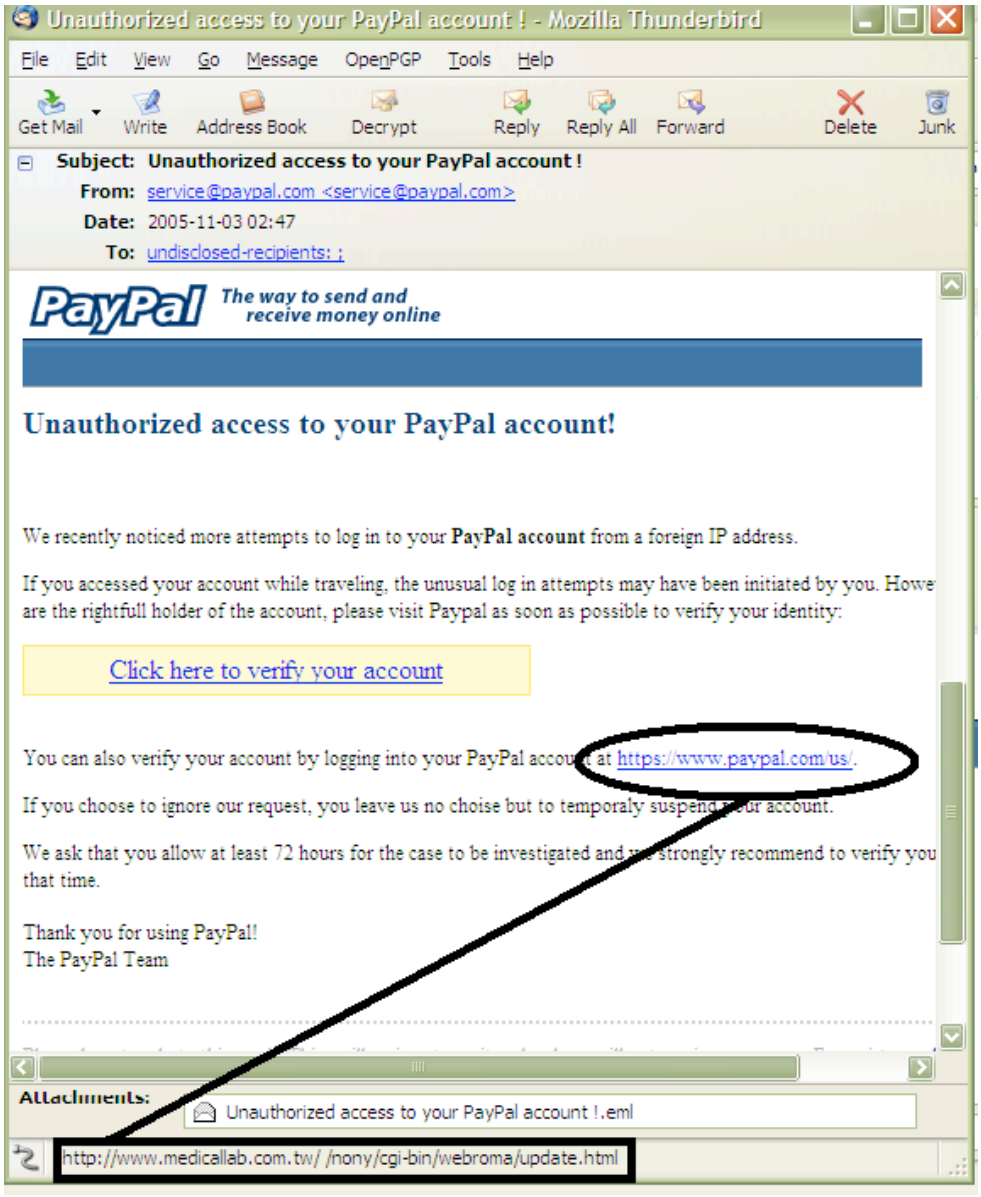
En vanlig strategi är att spammaren kapar en maskin, exempelvis med hjälp av virus eller en fjärrstyrd trojansk häst, och använder den kapade datorn som relä (proxy) för vidaresändning och leverans av skräppost. Istället för att den enskilt infekterade maskinen agerar relä och skickar ut skräpposten har spamprogramvaran numera börjat skicka e-postmeddelanden på ett sådant sätt att ISP:ernas e-postservrar nyttjas för leverans.

Spam har också en nära relation till de olika e-postbaserade bedrägerierna, såkallad phishing, som alltsom ofta förekommer på Internet, se kapitel 2.6.9.

Från och med den 1 april 2004 så gäller nya regler om obeställd elektronisk reklam via e-post i Sverige. Detta förbud som förts in i marknadsföringslagen är en följd av ett EU-direktiv om integritet och elektronisk kommunikation. Det är konsumentverket och konsumentombudsmannens roll att bedriva tillsyn av att marknadsföringslagen efterlevs.

2.6.9 Nätfiske och nigeriabrev

Phishing, bland kallat phiske eller nätfiske (eng. *phishing*) är en typ av bedrägeri som utförs via e-post och webb. Den som utför nätfiske försöker komma över värdefull, fast privat, information som kontokortsuppgifter, bankkontouppgifter, lösenord och liknande. Nätfisket utförs via förfalskad e-post, ofta professionellt utformad med företagslogotyper och auktoritivt innehåll. Mottagaren ombeds ofta att via en länk i brevet koppla upp sig till en hemsida som påstås tillhöra den organisation som bedragaren utger sig för att representera.



Figur 7 Exempel på nätfiske, notera URLerna

Vanligt i bedrägerisammanhang är att bedragaren miss- brukar kända varumärken eller namn på kända institutioner som exempelvis betalningsförmedlaren PayPal[45], Nordea[46] eller SE-Banken[47].

En vanlig variant av e-postbedrägerier är så kallade Nigeriabrev, ibland kallade 419-bedrägeri (eng. *419 fraud*) efter brottskategorin i Nigeria för denna typ av brott. Historien bakom Nigeriabrev är lång och ofta tragisk. Ursprungligen sände brottslingar i landet Nigeria

45 eBay & PayPal phishing Scams target popular institutions. http://antivirus.about.com/cs/emailscams/a/blebayscam5.htm

46 Kihlström, Stefan (2005). Kupp mot Nordeas nätbank. Artikel i Dagens Nyheter 2005-10-04. http://www.dn.se/DNet/jsp/polopoly.jsp?d=547&a=469543&previousRender-Type=6

47 **CM**
Brundin, Sverker (2005). Phishing-försök mot SEB. Artikel i Computer Sweden 2005-06-20. http://computersweden.idg.se/ArticlePages/200506/20/20050620110149_CS372/20050620110149_CS372.dbp.asp

pappersbrev till internationellt utvalda företagsledare eller personer som kunde antas ha förmögenheter. Breven kom från personer med högtidliga titlar, bar ofta färggranna logotyper och sigill samt innehöll imponerande avtryck från stämplar. I den digitala åldern ersattes de tidigare så påkostade utskicken som bara riktats mot ett fåtal med massutskick i form av e-post. Bedrägeriformen kom även att flytta utanför Nigerias men samlingsnamnet Nigeriabrev kvarstår. Typiskt för Nigeriabrev är att de innehåller ett budskap om att det är stora mängder pengar tillfälligt finns upplåsta på något bankkonto som den kontaktade medhjälparen kommer få stor del i utbyte mot en smärre administrativ avgift.

Oftast är det blufflotterier eller annan typ av bedräglig verksamhet bakom. En tumregel är att aldrig svara på sådana brev eller på annat sätt inleda dialog med avsändaren.

2.7 Organisatoriska hot och risker

Med *organisatoriska hot* menar vi hot som beror på felaktigheter inom den organisatoriska strukturen.

2.7.1 Oklara ansvarsförhållanden

Ett vanligt problem är att det inte står helt klart vem som har ansvar för alla delar i en lösning, till exempel vem som skall fatta beslut vid akuta händelser, såsom ett intrång. Ibland saknas utpekad ansvarig, ibland ligger ansvaret på flera personer eller funktioner och det inte går att ta ett beslut utan att det får internpolitiska biverkningar. För att uppnå en god organisatorisk säkerhet bör man gå igenom beslutskedjor och viktiga funktioner och kontrollera att det finns namngivna ansvariga för dessa. Det är också viktigt att de som är utpekade känner till att de är utsedda som ansvariga.

2.7.2 Personrisker

Det finns flera risker som har med personal att göra. Det kan bland annat handla om underbemanning, beroende av nyckelpersoner eller odokumenterad kunskap som bara finns hos enstaka personer i personalen.

Underbemanning har länge varit ett problem på IT-området då det varit svårt att rekrytera och behålla personal. I lägen med underbemanning hamnar man ofta i lägen där den befintliga personalen blir överarbetad och utsliten på grund av hård belastning. Underbemanning innebär ofta också att mängder med arbete, inte minst säkerhets- och kvalitetsarbete, blir eftersatt då det inte kan anses prioriterat för att hålla igång det allra nödvändigaste – systemets grunddrift.

Det är vanligt att man har enstaka tekniker som kan allt om ett system, en tillämpning eller en lösning. Om teknikern i det läget blir sjuk, åker på semester eller bestämmer sig för att sluta, så står man med ett system eller en tjänst som ingen längre förstår och inte heller kan underhålla. Nyckelpersonberoende bör avhjälpas genom att man utbildar fler personer, har god och uppdaterad dokumentation och har väl genomtänkta rutiner.

2.7.3 Organisationsförändringar

Förändringar i organisation kan också innebära förändringar i säkerhetsnivån, införande av nya sårbarheter eller ökade risker. Några exempel på olika organisationsförändringar som kan påverka säkerheten är uppköp, företagssammanslagningar (fusioner), utförsäljning av delar eller hela verksamheter eller så kallad outsourcing av bolag, bolagsenheter eller funktioner.

Sist men inte minst kan nedläggningar eller uppsägning av personal få stora konsekvenser med personer som känner sig svikna eller som har ett hämndbehär. På engelska kallas detta ofta för *disgruntled employees*.

IT ger nya möjligheter att hämnas eller ställa till besvär. Vid omvälvande förändringar, som vid till exempel företagsförsäljning, uppköp eller nedläggning kan personal, företagspartners eller kunder få minskad lojalitet till företaget, eller till och med begå illojala eller kriminella handlingar som en missnöjesyttring. Därför är det viktigt att redan i planeringen inför organisationsförändringar göra riskbedömningar och vidta förberedande säkerhetsåtgärder.

2.8 Administrativa hot och risker

2.8.1 Okunskap om system och infrastruktur

Ett allt vanligare problem är att ansvariga inte känner till hur information hanteras eller hur IT-infrastrukturen är uppbyggd. Ibland sätter de inblandade en ära i att inte ha kunskap om system och struktur. Detta innebär givetvis ett stort problem, därför att man kan fokusera fel eller skapa bristfälliga lösningar. Men det leder också till att man inte får ett övergripande säkerhetstänkande. Därmed riskerar man att lösningar som införs inte medger någon reell säkerhet.

2.8.2 Ojämna skyddsnivåer och felfokuserade lösningar

Säkerhet och skyddsnivåer i en organisation är ofta ojämt fördelade. De säkerhetslösningar som införskaffas är ofta skapade efter enkla kriterier och enligt någon buffelhjordsmentalitet, där man tänker i precis samma banor som alla andra. Det blir därmed lätt att lägga allt krut på att säkra upp huvudentrén, vilket IT-mässigt ofta motsvaras av organisationens brandvägg, medan man isjälva verket glömmer de bakdörrar och andra vägar som leder in och ut ur nätverket.

Det är minst lika viktigt att göra något åt den allmänna säkerheten på det interna nätverket, att skydda andra in- och utgångar, till exempel modem- och ISDN-pooler, hyrda förbindelser till andra företag eller hantera nya transmissionstekniker, exempelvis trådlösa nätverk.

2.8.3 Avtal, dokumentation, rutiner och policies saknas

Dokumentation är ofta en förutsättning för att en organisation ska fungera bra. Nyckelpersonberoenden kan till exempel minskas genom dokumenterade rutiner och policies. Dessutom kan man minska risken att arbetet blir godtyckligt eller slarvigt utfört.

Inte sällan blir policies och rutiner snabbt omoderna eftersom de inte underhålls eller uppdateras. Organisationer kan till exempel skapat en kontinuitetsplanering, där de dokument som beskriver planeringen är anpassade till äldre system och lösningar. Det kan vara stor- eller minidatorlösningar som kanske inte ens används längre inom organisationen.

Andra viktiga dokument som ofta saknas är avtal med leverantörer, servicebyråer, konsultföretag och liknande. Dessa saknade avtal inkluderar bland annat sekretessavtal, underhållsavtal, licenser och rättigheter samt så kallade *leveransåtaganden* (eng. *service level agreement,* SLA). Avtal med anställda och användare är en till typ av avtal som borde upprättas men som ofta saknas, exempelvis ansvarsförbindelser (eng. *Acceptable Use Policy*, AUP), distansarbetsavtal och liknande.

2.8.4 Bristande underhåll

Dåligt underhåll eller avsaknad av underhåll är ett allvarligt hot. Exempel på underhållsåtgärder som ofta saknas är installation av så kallade *patchar* och akuta säkerhetsfixar, konfigurationsändringar i väntan på programuppdateringar, säkerhetskopiering samt uppdatering av systemdokumentation så att den motsvarar det installerade systemet.

I realiteten kan man inte skapa säkerhet en gång för alla i driftsatta system. Man måste istället kontinuerligt underhålla dem med aktuella uppdateringar och säkerhetsrelaterade rättelser. Många leverantörer kallar sina uppdateringar för patchar. Ibland levereras stora samlingar som skall installeras på en gång. Dessa kallas ibland för klusterpatchar. Den kanske viktigaste spelaren, Microsoft, kallar sina uppgraderingar för *Service Packs* och akuta uppgraderingar för *Hot Fixes*.

2.8.5 Bristande uppföljning

Ett alltför vanligt fel i de flesta organisationer är bristande uppföljning. Uppföljningen är något som bör ske på alla olika nivåer. Det kan vara att utreda orsakerna till varför ett projekt lagts ner, att kontrollera efterlevnaden av avtalspunkter i ett driftavtal, att följa upp huruvida ett system installerats enligt gällande riktlinjer eller att göra periodiska, tekniska säkerhetskontroller.

Exempel på några viktiga uppföljningar är efterlevnad av säkerhetsföreskrifter, teknisk säkerhetsgenomgång av system, penetrationstester samt införande av förändringar som följd av en IT-incident eller databrott.

2.9 Kriminella hot och risker

Kriminella hot och risker är något som kan drabba alla företag, organisationer och privatpersoner. Kriminalitet kan utföras av egen personal (insiderbrott), utomstående eller en kombination av dessa. Den senare varianten där någon eller några på insidan hjälper utomstående med kunskap eller tillvägagångsätt medför en kombinerad attackstyrka är svår att skydda sig mot och är därför något man inte bör glömma. En effekt av en sådan kombinerad angriparstyrka är dessutom att eventuella spår av incidenten vara svårtydda eller manipulerade. Interna hot utförda av insiders är ett speciellt problem då de i allmänhet har större åtkomstmöjligheter, högre befogenheter och rättigheter i system och dessutom ofta har andra möjligheter att manipulera spårdata eller påverka internutredningar.

Några exempel på kriminella hot och risker är stöld av utrustning, programvara och information (data), bedrägeri, utpressning, trolöshet mot huvudman, industrispionage samt förfalskning.

Stöld av information är vanligare än man tror. Det kan vara egen personal, konsulter eller andra med legitim åtkomst till system och information som kopierar information i samband med att de avslutar en anställning eller ett uppdrag. Förutom stöld kan det i vissa fall också rubriceras som trolöshet mot huvudman. Detta är extra känsligt då någon går från ett företag till ett konkurrentföretag. Då kan det dessutom anses vara industrispionage. Utpressning är ett annat kriminellt hot. Det kan vara väldigt svårt att avstyra och hantera. Det finns flera exempel på IT-relaterad utpressning med scenarier av följande slag:

- Hot om åtkomstattacker (DoS) mot finansiella aktörer och e-handelsföretag

- Någon gör intrång i ett system och vill ha pengar eller annan motprestation för att berätta var i systemet som det finns svagheter.
- Långvarig och hemlig manipulation av data, vilket gör det svårt för den drabbade att återskapa ursprungsinformationen.

Ett område av kriminella hot som ökar på Internet är ärekränkning, förtal och olika nivåer av mobbning. På webbsidor, via e-postlistor, nyhetsgrupper och via direktadresserad e- post skickas eller publiceras information som andra anser anstötlig eller kränkande. Ibland är förtalet eller ärekränkningen en uppsåtelig handlig men ibland är moderna datorbaserade kommunikationstjänster som medium något som tyvärr frodar överilade handlingar och alltför oigenomtänkta replikskiften.

2.10 Sociala attacker

Social ingenjörskonst (eng *social engineering*) är en icketeknisk attacktyp som är så effektiv och viktig att den kräver ett extra omnämnade.

Attacken baserar sig på att angriparen utnyttjar övertalning, lurendrejeri, falsk ingivelse eller liknande metoder för att få en eller flera andra personer att lämna ut information, utföra en viss handling, eller på annat sätt agera ombud. Attacken kan till exempel utföras av antingen anställda eller utomstående med hjälp av telefon, e-post, fax och brev.

Ett exempel på en social attack är ett förfalskat e-postmeddelande ifrån systemadministratören till användare med uppmaningen att ändra lösenordet på grund av en uppdiktad säkerhetsincident. Ett annat exempel är e-postbrev som påstås komma från leverantörer som innehåller uppmaningar att göra förändringar i systemet, såsom att installera de till brevet bifogade uppdateringarna. Flera leverantörer, såsom Microsoft, skickar av dessa anledningar aldrig ut prograuppdatering via e-post utan har andra vedertagna säkra kanaler.

Sociala attacker är lätta att utföra då de ofta inriktar sig på människors godtrogenhet och servicekänsla. Av samma anledning är de svåra att skydda sig mot. Skydd mot sociala attacker är ökat säkerhetsmedvetande, kunskap om gällande säkerhetsföreskrifter och en sund kritisk granskning av oväntade händelser.

3 Skadlig kod

Avsiktligt skadlig kod (eng. *malicious code, hostile code* eller *malware*) är en fackterm och ett samlingsnamn på en mängd säkerhetsproblem som är förknippade med program som består av, eller innehåller, avsiktligt skadliga program- och programdelar. Den mest omtalade varianten kallas för datavirus eller enbart virus, och brukar i allmänhetens ögon vara liktydigt med alla varianter av avsiktligt skadlig kod.

Det är inte alltid uppenbart vad som kan innehålla dolda programfunktioner. Filer som traditionellt ansetts vara data, såsom dokument från ordbehandlare eller kalkylprogram, innehåller i dag skript- och makroprogrammeringsmöjligheter. ”Datafiler” har därför på grund av denna funktionalitet fått ökad popularitet hos dem som utvecklar virus, maskar, bakdörrar och liknande. Det finns till och med berättelser om hur ljudfiler och fontbeskrivningar lyckats manipuleras att innehålla skadlig kod. Effekten blir att program som nyttjar dessa antingen direkt läser in styrsekvenser eller med hjälp av utnyttjande av programfel tvingar programmet att få in kod som exekveras.

Inte sällan innebär avsiktligt skadlig kod till reella hot och verkliga angrepp, men det är också vanligt att media spär på risker eller konsekvenser av virushot, ofta till helt orealistiska nivåer.

Figur 8 Tidningslöpsedel i samband med maskspridning

Definition 31: Avsiktligt skadlig kod är ett samlingsnamn för all typ av programkod som avsiktligen orsakar störning, skada eller bryter mot organisationens säkerhetspolicy.

Fred Cohen anses ha myntat uttrycket datorvirus . I Cohens doktorsavhandling[48] återfinns ursprungsdefinition av virus:

> *A virus is a program that is able to infect other programs by modifying them to include a possibly evolved copy of itself.*

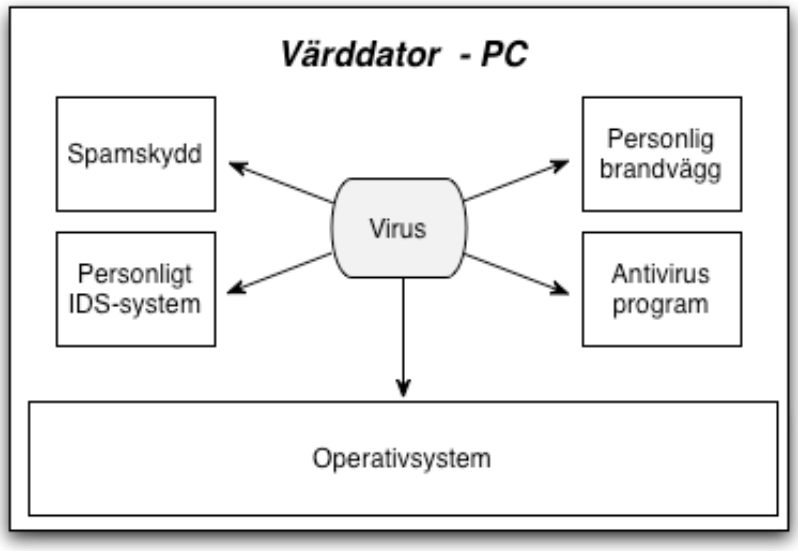
Vi kommer att nyttja en något modifierad version av Cohen:s definition:

> **Definition 32:** Ett datorvirus är ett program som rekursivt skapar explicita, möjligtvis vidareutvecklade, kopior av sig själv.

Termerna virus och datorvirus har kommit att bli mycket missbrukad och används ofta, speciellt i media, att beteckna alla former av avsiktligt skadlig kod. Datorvirus är som regel plattformsberoende, och många fall versionsberoende, vad det gäller målmiljön. Ett minnesresident virus utvecklat att infektera MS-DOS kommer inte att kunna en Linuxinstallation. Det finns plattformsoberoende virus skrivna i makrospråk, Java eller andra språk som finns i exekveringsmiljöer för flera operativsystem.

Avsiktligt skadlig kod kan finnas gömd i en mängd olika program och en användare kan drabbas vid ett antal olika tillfällen, till exempel vid medveten nedladdning av program, vid webbanvändning i samband med användning av Javaprogram, Javaskript eller ActiveX-komponenter, eller vid öppnande av epostbilagor.

Inte sällan försöker ett angripande virus sig på att avaktivera säkerhetsfunktioner såsom antivirusprogram, spamskydd, lokalt installerade brandväggar eller IDS-system. Allt för att förbättra sina odds mot upptäckt under spridnings mot nya system eller för att skydda de aktiviteter som virusets last (exempelvis trojansk häst) utför i själva datorn.



Figur 9 Virus som angriper säkerhetsfunktioner och datorns operativsystem

[48] Cohen, Fred (1987) Computer Viruses: Theory and Experiments. Computers and Security6 (1987) 22-35

En av de mer vanligt förekommande varianterna av skadlig kod är maskprogram, program som kravlar mellan datorer. Ett maskprogram är uppbyggt av flera beståndsdelar, vissa förekommer alltid såsom målväljar- och attackdelarna, medans andra såsom kommunikationsdel (för att kontrollera masken eller synkronisera mellan olika maskar som sprider sig) och lastrum med en programkomponent, exemplevis en trojansk häst, inte alltid nyttjas.

Figur 10 Principiell uppbyggnad av ett maskprogram

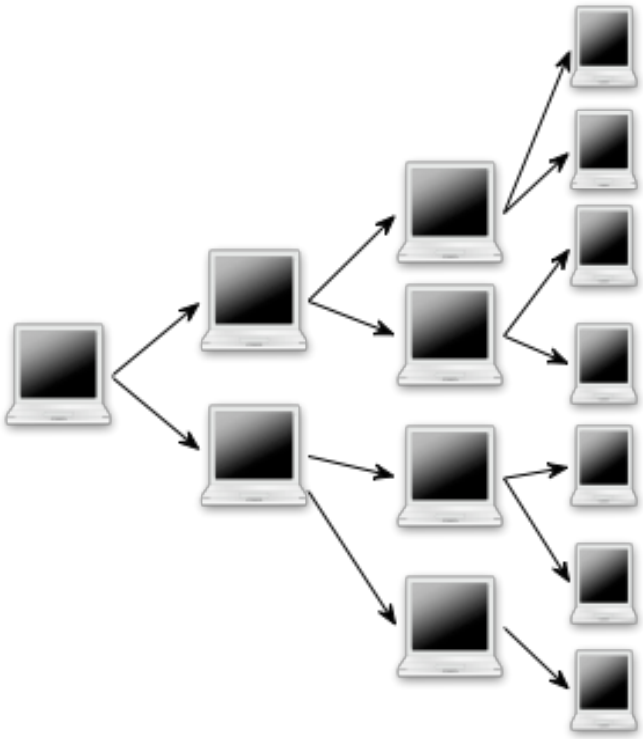
Konceptet med en mask (eng. *worm*) beskrevs första gången av science-fictionförfattaren John Brunner i novellen *The Shockwave Rider*[49]*,* skriven av i mitten av 1970-talet. Brunner kallade företeelsen för *tapeworm*. En av de första större maskangreppen på Internet var Robert Morris maskangrepp[50] som spred sig till en stor del av de vid den tiden internetanslutna datorerna.

Definition 33: En mask är självreplikerande program som sprider sig mellan datorer via ett datornätverk

Sprindningsmekanismer och spridningsprinciper för maskprogram är beroende på hur upphovsmannen av programmet har valt att masken skall agera. Om masken slumpmässigt letar efter sårbara datorer som har en eller ett begränsat antal sårbarheter, kan spridningen ta en tid på grund av att de datorer som den ursprungligt infekterade datorn, och de som kommer snart därefter, inte får träff på de adresser som slumpmässigt tas fram. Att inte få träff innbär i detta sammanhang såväl att det kanske inte är en tilldelad eller använd IP-adress likväl som att de datorer som försöks kontaktas kanske inte innehåller sårbarheter som masken försöker nyttja för sin spridning.

[49] http://vx.netlux.org/lib/mjb01.html

50 With Microscope and Tweezers: An Analysis of the Internet Virus of November 1988 http://www.morrisworm.com/worm-chronology.html



Figur 11 Spridning av en mask till andra datorer i nätverket

En trojansk häst är ett program som innehåller en dold funktion. Den dolda funktionen kan vara en fjärrstyrningsfunktion eller att en extrakopia av ett brukat lösenord sparas undan.

> **Definition 34:** En trojansk häst är ett program som förutom önskad funktion innehåller en medvetet dold funktion.

Några av de mer kända trojanska hästprogrammen är NetBus, BackOrifice och SubSeven. Dessa typer av trojanska hästar kallas ofta för RAT - Remote Access Trojan.

Ett vanligt spridningssätt för trojanska hästar är att de döljs i bilagor till elektronisk post. Genom att sända med programfiler eller dokument som innehåller makron- eller programtillägg, går det att lura mottagaren att köra program som innehåller en dold funktion. En del av den dolda funktionen kan vara en spridningsfunktion för att skicka programmet vidare, detta brukar göra trojanen till en så kallad mask.

Ett klassiskt trick med avseende på trojaner är att installera en ny modemstyrnings-programvara, en så kallad ”dialer”, som ringer ett av angriparen utsett betalnummer. Denna attack kallas populärt för *modemkapning*.

Ett program kan utföra mer funktioner än det som en normalanvändare förväntar sig att det kan göra. I vissa fall är dessa extrafunktioner bakdörrar som lagts till av en ursprungsutvecklare. Ibland är det bakdörrar som lagts till senare, exempelvis vid ett intrång i systemet.

Definition 35: En bakdörr är en odokumenterad programfunktion som kan nyttjas för att kringå ordinarie kontroller

En bakdörr (eng. *backdoor*) kan vanligtvis nyttjas för att kringå en ordinarie autentisering, andra typer av behörighetskontroller, loggning eller för att ge obegränsad åtkomst. Även legitima applikationer kan ha dolda funktioner. Logiska bomber används för att vid vissa givna stimulin aktiveras och agera.

Definition 36: En bomb är en dold och avsiktligt skadlig programfunktion i ett program eller ett programsystem som, under vissa särskilda villkor, kan aktivera en oönskad aktivitet

En logisk bomb (eng. *logical bomb*) kan aktiveras när en viss användare loggar in i ett system, när en viss mängd data har behandlas, när upphovsmannen inte varit inloggad under ett visst antal dagar eller någon annan liknande förutsättning inträffar. Den dolda funktionen kan vara att själa pengar, radera filer, manipulera information, eller någon annan kriminell eller destruktiv aktivitet.

Roger Duronio51, 60, installerade 2001 en logisk bomb på cirka 1000 av finansföretaget UBS PaineWebber 1500 datorer i hela USA. Duronio slutade på företaget i slutet av februari 2002. Den fjärde mars 2002 aktiverades det tidsbaserade programmet och ställde till skada. Duronio hade köpt värdepapper som skulle öka i värde om PaineWebbers värde sjönk, vilket gjorde att hans förfaringssätt ansågs vara bedrägeri, inte bara IT-sabotage.

En effektiv metod att stjäla information är att spela in allt som skrivs på ett tangentbord.

Definition 37: En tangentbordsavlyssnare är ett program som spelar in alla tangentbordsnedslag

Tangentbordsavlyssnare (eng. *key logger*) sparar som regel den inspelade inmatningen till en fil. Hårdvarubaserade tangentbordsavlyssnare kan anslutas till tangentbordskontakten, eller om datorn är USB-ansluten så kan den sättas på USB-bussen, eller alternativt byggas in i själva tangentbordet. En hårdvarubaserad tangentbordsavlyssnare kan som regel inte upptäckas med hjälp av säkerhetsprogramvaror.

[51] U.S. Department of Justice (2002). *Disgruntled UBS PaineWebber Employee Charged with Allegedly Unleashing "Logic Bomb" on Company Computers.* http://www.cybercrime.gov/duronioIndict.htm

Figur 12 KeyGhost, en hårdvarubaserad tangentbordsavlyssnare

Definition 38: Ett rootkit är samling program och programkod som möjliggör permanent och oupptäckbar närvaro i ett datorsystem

Rootkit kan placeras ut av en angripare efter ett lyckat intrång och används för att dölja angriparens spår. Ofta är det inloggningsinformation, sessionsinformation, processnamn, filnamn eller systemdata som skulle avslöja intrånget eller angriparens närvaro som döljs. Namnet rootkit kommer ursprungligen från UNIX och var ett samlingsbegrepp för verktygslådor med ersatta systemprogram som ps, find, netstat eller ls. Traditionellt har rootkit funnits på Linux, SunOS, Solaris och andra vanliga UNIX-dialekter, men numera finns rootkit som koncept för alla typer av operativsystem. Konceptet har tillochmed överförts till icke-operativsystemsmiljöer. Vissa säkerhetsforskare har lanserat koncept såsom databasrootkit[52].

En stor kontrovers uppstod när Sony/BMG använde rootkit-teknologi för att installera kopieringsskydd på CD-skivor med musik. Skivorna installerade dold programvara om de spelades på en PC-dator. Detta upptäcktes av en slump av utvecklaren till säkerhetsprogrammet Rootkitrevealer som publicerade upptäckten i sin blog[53] med titeln "Sony, Rootkits and Digital Rights Management Gone Too Far". Flera problem har uppstått efter Sony tilltag uppdagats. Företaget har blivit stämda i grupptalansrättegångar[54], pirater har använt rootkitet för att kringgå skyddsmekanismer i on-linespel[55] samt utvecklare av skadlig kod har skrivit trojanska hästar[56] som nyttjar mekanismen.

52 Kornbrust, Alexander "Database rootkits" http://www.red-database-security.com/wp/db_rootkits_us.pdf

53 Russinovich, Mark (2005) Sony, Rootkits and Digital Rights Management Gone Too Far. http://www.sysinternals.com/blog/2005/10/sony-rootkits-and-digital-rights.html

54 Krebs, Brian (2005) Calif. Lawsuit Targets Sony. http://blogs.washingtonpost.com/securityfix/2005/11/calif_ny_lawsui.html

55 Lemos, Robert (2005). World of Warcraft hackers using Sony BMG rootkit. http:// www.securityfocus.com/brief/34

56 Sophos Antivirus (2005) Trojan horse exploits Sony DRM copy protection vulnerability.

Speciella program har behövt användas för att hitta rootkit eller misstänkt rootkitaktivitet. Några kända exempel är Rootkitrevealer[57], chkrootkit[58], F- secure Blacklight[59] samt tripwire[60]. Rootkitrevealer och Blacklight letar efter misstänkt aktivitet i RAM-minnet och på disk medans tripwire och chkrootkit mer är inriktade på att kontrollera filers giltighet.

> **Definition 39:** Spionprogram är en samlingsterm för program som utan användarens medvetande eller medgivande visar reklam, samlar in personlig information eller ändrar inställningar i operativsystemet eller applikationsprogram

Ett spionprogram (eng. *spyware*), är en typ av program som laddas utan att användaren har vetskap om programmet eller att användaren förstått bieffekter av ett visst program. Spionprogrammen registrerar vissa typer av användaraktiviteter eller aktiveras i vissa scenarion. Dessa scenarion kan vara när man besöker vissa webbplatser och spionprogramvaran detekterar detta och föreslår alternativ, spionprogrammet gör automatiska omdirigeringar till annan plats, eller i andra fall så sänder spionprogrammet iväg data om användaren till programmets upphovsmän eller sponsorer. Användarinformation som sänds iväg är berör ofta surfvanor, namn på filer eller program användaren aktiverar, etc. Spionprogramvara används ofta i samband med reklamfinansierad programvara.

Spionprogram, till skillnad från maskar och virus, är som regel inte utvecklade för att sprida sig själva.

[57] http://www.sysinternals.com/utilities/rootkitrevealer.html
[58] http://www.chkrootkit.org/
[59] http://www.europe.f-secure.com/blacklight/
60 http://www.tripwire.com/

Figur 13 Brandväggen Zonealarm fångar spionprogrammet gator

Ofta upptäcks spionprogrammen genom att datorn går ovanligt långsamt, att datorn uppför sig konstigt eller att nya fönster, knappar eller menyer dyker upp. Ett vanligt skäl till detta är att spionprogrammet installerats i samband med att användaren godkänt ett användaravtal, end user license agreement - EULA, för ett program som har en huvudsaklig annan funktion, och därmed även godkänt installation av spionprogrammet. Genom att man läst, eller genom att inteläst eller förstått hela avtalet, har man enligt vissa länders lagstiftning godkänt spionprogrammets installation. Eftersom antivirustillverkarna inte vill ha juridiska tvister med tillverkare av spionprogram, så undviks detektion av dessa program.Tillverkare av spionprogram ogillar etiketten och försöker därför använda alternativ som adware.

Definition 40: En bot är ett fjärrstyrt program, en mjukvarurobot

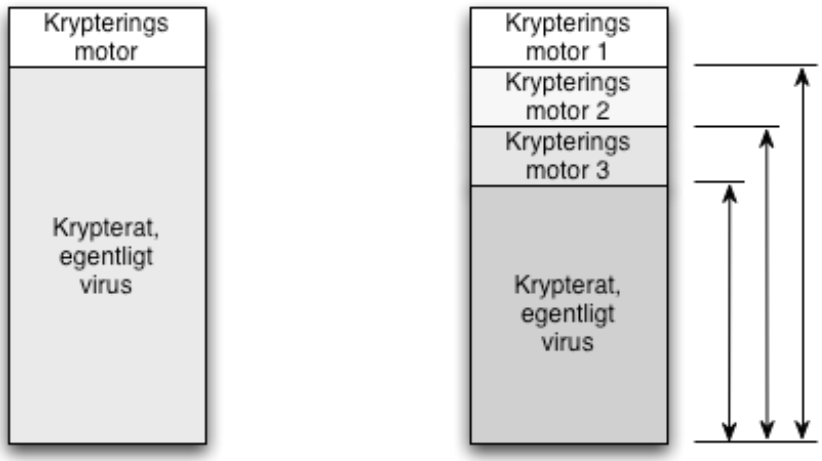
Definition 41: Botnet är en samling av botprogram som tillsammans autonomt kan kontrolleras och fjärrstyras, ofta till kriminella eller skadliga handlingar.

Ett botnet kan vara allt från små grupper av crackade datorer till massiva sammankopplingar med 1000-tals botprogram. Det finns dokumenterade botnet-installationer där 10000-tals datorer har ingått[61]. Inte sällan används botnets av kriminella som satt i system att tjäna pengar på att sälja tid på dessa botnets till intressenter.

Skadlig kod, exempelvis virus, använder alltmer avancerade funktioner för att förhindra eller försvåra upptäckt från antivirusleverantörer och de som drabbas. Smygfunktionalitet används under spridningsmomenten över nätverk.

Definition 42: Ett retrovirus är ett datorvirus som använder explicita metoder för att undgå upptäckt av antivirus eller använder olika metoder försöker slå tillbaka mot antivirusprogram.

Mutationstekniker såsom polymorft innehållsförändring av programkod och data används för att undvika mönster och signaturer som ger antivirusjägarna en känt spår. Kryptering används för att försvåra insyn. Ofta används ovanstående, och andra tekniker, i kombination, för att uppnå en fördel för angriparna.



61 Telenor takes down 'massive' botnet http://www.theregister.co.uk/2004/09/09/telenor_botnet_dismantled/

Figur 14 Virus som skyddar sig mot upptäckt med hjälp av kryptering

Förutom att använda krypteringsteknologi för att skydda viruset självt, kan kryptering användas av viruset mot den miljö som det angriper. Kryptovirologi definieras som kryptoteknologi kombinerad med skadlig kod. Ett kryptovirus är ett virus som nyttjar publik nyckelteknologi i sitt angreppssätt, exempelvis genom att föra med sig en publik nyckel och kryptera filer i ett system som angrips. Angriparen, som har tillgång till den privata nyckel, är den som kan återställa filerna till ursprungligt skick, om inte den drabbade har fungerande säkerhetskopiering. Kryptovirologi kan anses vara det ultimata high techvarianten av utpressningsmetodik.

3.1 Exempelfall

3.1.1 AIDS-disketten

År 1989 mottog omkring 20000 personer, bland annat framstående AIDS-forskare, i Europa, Afrika och USA ett kuvert med en diskett till en PC. Disketten var märkt *AIDS Information Diskette*, vilket ledde till att ett antal de addresserade personerna använde disketten. Licensen innehöll ett kryptiskt meddelande:

> In case of breach of license, PC Cyborg Corporation reserves the right to use program mechanisms to ensure termination of the use of these programs. These program mechanisms will adversely affect other program applications on microcomputers. You are hereby advised of the most serious consequences of your failure to abide by the terms of the license agreement: your conscience may haunt you for the rest of your life; you will owe compensation…

De stackars personer som hade exekverat programmet på disketten fick snabbt en hårddisk som slutade att fungera. I många fall var offren forskare inom utveckling av AIDS-medicin, vilket kunde innebära förhinder och förseningar för vidare forskning inom framtagandet av nödvändiga preparat. För att rädda informationen i datorn uppmanades användaren skicka pengar till företaget PC Cyborg Corporation på en postbox i Panama. En Dr Joseph Popp fanns senare vara skyldig till utskicket. Han häktades av engelsk polis, men ansågs för psykiskt sjuk för att kunna dömmas till straff. Senare har han, i sin frånvaro, dömts för utpressning till fängelse, av flera rättssystem, bland annat det italienska[62]. Hela historien med Popper är mycket märklig. Bara portokostnaden för utskicket beräknas ha kostat över $150000[63].

3.1.2 Concept

Concept var det första makroviruset för Microsoft Officepaketet. Concept är ett wordvirus som upptäcktes 1995 i efterdyningarna av att Microsoft släppte Windows 95. I samband med Windows 95-lanseringen släppte Microsoft en CD-ROM-skiva med namnet Microsoft Windows 95 Software Compatibility Test. Det enda problemet var att CD-skivan var smittad med Conceptviruset[13]. Eländet förvärrades av att många hade skaffat det nya operativsystemet

[62] Sid 3 i VirusBulletin (1993) *Popping Up*. Artikel om Joseph Popper. http://www.virusbtn.com/virusbulletin/archive-pdf/1993/199305.pdf

[63] 3.*858 Warning: beware the Cyborg bearing gifts!*
http://lists.village.virginia.edu/lists_archive/Humanist/v03/0852.html

men ännu inte hunnit införskaffa en ny version av anti-virusprogramvaran som fungerade med Windows 95.

3.1.3 Code Red

CodeRed var en mask som upptäcktes första gången i juli 2001. Masken hade en god spridningsförmånga och spred sig till datorer som använde IIS-webbservern och dess indexserver på Windows 2000. CodeRed hade ett antal nya, eller intressanta, egenskaper såsom:

- att den existerade enbart på nätet och i de smittade maskinernas primärminne, aldrig som en fil på disk
- spridning av masken skedde enbart mellan den första och den nittonde varje månad, angrepp mot www.whitehouse.gov skedde mellan den tjugonde och den tjugoåttonde varje månad
- att den attackerade webbservrar på port 80, en port som garanterat var öppen i brandväggar
- att den nyttjade en buffertöverskrivninsbugg i IIS indexserver[64] som hade publicerats en månad tidigare
- att den kapade exekveringstrådar (eng. *thread hijacking*) i IIS-webbservern för att sedan injicera nya exekveringstrådar som spred masken vidare
- att den innehöll ett antal buggar, bland annt att framslumpandet av nya adresser att attackera inte fungerade som avsett. Vissa av buggarna ändrades av upphovsmannen som därefter släppte rättade versioner av masken, CodeRed v2
- den utförde en samlat angrepp med överlastningsattacker mot www.whitehouse.gov

Den ursprungliga masken fanns i två versioner, ofta kallade CodeRed v1 och CodeRed v2, vilka inte skall förväxlas med den modifierade masken som kallas för CodeRed II.

3.1.4 Morrismasken

Robert Tappan Morris, Jr. numera en professor vid Massachusetts Institute of Technology, är upphovsmannen till den första Internetmasken[15]. Maskmakaren Händelsen väkte stora rubriker på grund av flera orsaker, dels på grund av själva incidenten i sig och den omfattning den fick Dels blev det rubriker när det blev klart vem som var mannen bakom masken - Robert Morris, son till Robert Morris Sr som länge hade varit en välkänd säkerhetsexpert på AT&T Bell Labs och sedan teknisk chef på NSA.

Internetmasken upptäcktes 2 november 1988. Masken, sedemera kallad Morris-masken, skapades när Morris var en student vid Cornelluniversitetet. Internetmasken var ett program som spred sig till en stor andel av det dåvarande Internet och som tog sig från dator till dator genom att utnyttja en kombination av olika säkerhetshål i sådant som e-posthanteringsprogrammet sendmail, fingertjänsten och felformgivna protokoll som berkeleys r-tjänster: rlogin, rsh, rcp med flera. Masken hade också med sig en ordlista på 432 lösenord som den använde i olika försök att sprida sig via tjänster såsom FTP. Effekten av programmet var bland annat att de flesta system blev infekterade och återinfekterade gång på gång, vilket i sin tur överbelastade systemen till sådan grad att de blev oanvändbara.

Anledningen till maskens uppkomst var enligt Morris ett försök att kartlägga Internets storlek. Masken spred sig snabbt till uppskattningsvis 6000 datorer som på grund av nätverks- och

[64] http://www.eeye.com/html/Research/Advisories/AD20010618.html

systembelastningar förlorade funktionalitet eller kraschade. Morris dömdes 1990 till tre års villkorligt fängelse, 400 timmars samhällstjänst samt böter på $10 000[65].

Förutom det rättsliga efterspelet stängdes Morris av från vidare studier vid Cornelluniversitetet. En effekt av Internetmasken och det bryska IT-säkerhetsmässiga uppvaknandet ledde till var skapandet av incidenthanteringscenters så som CERT/CC (Computer Emergency Response Team Coordination Center) och CIAC (Computer Incident Advisory Capability).

Det går att dra åtminstone par viktiga tekniska slutsatser från denna tidiga incident. Den första observationen är att masken var kapabel att sprida sig till olika operativsystem på olika hårdvara. Masken medflyttade färdigkompilerad objektkod som den nyttjade beroede på vilket operativsystem som målsystemet hade. Detta ledde till att den lyckades få en så pass bred spridning som den fick. En annan observation är att den bara spred sig i binär form och hade krypterat, visserligen med en enkel algoritm, den överförda binärerna för att försvåra analys.

3.2 Virusskydd

Virusskydd brukar normalt användas på klientdatorerna eller på filservrar. Ur ett nätsäkerhetsperspektiv kan virusskydd användas i kombination med brandväggar eller liknande utrustning för att automatiskt viruskontrollera överförd information, till exempel data överförda via FTP, HTTP eller e-post. En annan viktig aspekt är att ha antivirusprogram installerade på bastiondatorer i brandväggsområdet - för att undvika att dessa blir drabbade av virus eller angrepp från maskar. Ofta finns det tyvärr vita fläckar på nätet – maskiner som inte har virusskydd installerat. Bärbara datorer är ett sorgebarn som ofta släpar efter i uppdatering eller inte underhålls alls, vilket ofta ställer till problem när dessa flyttas mellan hemarbetsplatser och det interna nätverket. Andra oskyddade system kan vara skrivar-, databas-, applikationsservrar,

inpassagesystem och liknande som glöms bort eller man anser det vara onödigt att skydda. En annan vanligt förekommande vit fläck på virusskyddskartan är system som baserat på annat än Windows. Stora serverlösningar baserade på UNIX, Linux, VMS och liknande brukar sällan förses med något antivirusskydd.

Att ha dessa medvetna eller okända undantag är självbedrägligt. Ett relaterat problem, som kommer av att inte alla maskiner har antivirusskydd, är att dessa maskiner gör att maskar kan dröja sig kvar på internnätet och ställa till problem långt efter själva uppstädning, hanteringen av AV-uppdatering och patchning gjorts klart.

Det är viktigt att understryka att antivirusprogram i sig inte är ett fullgott skydd mot skadlig kod såsom virus, nätmaskar och liknande. Det finns många exempel där antivirusprogrammen reagerat för sent – efter infektion, eller inte alls på ett angrepp eller infektion. Många moderna maskar, virus och liknande använder många metoder för att undvika upptäckt. En metod är att komma in via ett hål i ett program, ofta via någon variant av buffertöverskrivningsattack, och sedan bara existera i minnet, vilket gör det svårt för antivirusprogrammen att upptäcka

[65] Lawrence Kestenbaum (1990) Robert Morris Sentencing. http://www.potifos.com/morris.html

infektionen. Andra metoder är att undvika upptäckt är att avaktivera antivirusprogram, personliga brandväggar och liknande.

Kompletterande skydd mot skadlig kod är att säkra upp system, applikationer och utrustning med olika metoder. Det mest uppenbara skyddet är att följa nyhetsflödet över funna säkerhetsbrister och ha snabba och väl fungerande rutiner för införande av uppdateringar, fixar, patchar och temporära säkerhetsomkonfigurationer. Andra uppsäkrande metoder är att låsa till operativsystem och applikationer mer än vad som är vanligt att man gör i dagsläget.

Några mer handgripliga skydd är att använda interna brandväggar, nyttja IDS-teknologi som ett sätt att upptäcka och eventuellt begränsa spridning och även nyttja honungsfällor på vissa dedicerade nätanslutna datorer där varje anslutning och aktivitet bevakas och hanteras. Detta ger ett mer komplett skydd än enbart antivirusprogramvara.

Ett mer övergripande typ av skydd är att ha en väl utformat nätarkitektur där man har bra kontroll över alla externa ut- och ingångar, vad och hur utrustning får anslutas till nätet, ha en strukturerad indelning i säkerhetszoner och liknande. Genom skydd på nätverksnivå kan man också förebyggande mot att bli drabbad av maskar samt att det förhindrar okontrollerat spridande om man väl blir drabbad av ett sådant problem.

4 Nätverksarkitektur

Let me quote something that I've immodestly labeled
"Bellovin's First Law of Networks":
Network interconnect,
always, and always at the edges.
Designing something on the assumptions of a closed environment,
or of an environment where the center controls the connectivity,
will fail.
-- Steve Bellovin[66]

4.1 Nätverkstopologi

Ett datorkommunikationsnätverk kan modelleras och konstrueras på olika sätt. Ett stort nät utan skydd mellan de olika delarna kan möjliggöra för en person att komma åt andra delar av nätet som denne normalt sett inte borde ha åtkomst till.

4.1.1 Säkerhetszoner

Ett sätt att logiskt komma åt prolemen med stora nätverk är att dela in det i mindre, mer hantera delar. Olika delar av nätverket kan ha olika skyddsvärde, vilket kan leda till att nätverket delas upp i olika säkerhetszoner (eng. *security zones*).

4.1.2 Virtuella nätverk

Virtuella nätverk, och främst så kallade virtuella LAN (*VLAN*) är en teknologi för att dela upp och strukturera ett nätverk. Med VLAN kan man ha gemensam trafik i stamnätet men separera enskilda LAN i de olika anslutningsportarna i en växel (eng. *switch*) eller anslutningsutrustning. VLAN används främst för möjligheten att sprida olika LAN till geografiskt spridda växlar. Med hjälp av extra fält kan man skapa virtuella LAN som kan adresseras från enstaka portar.

Många använder VLAN som en metod för att uppnå högre säkerhet genom en högre nivå av separation mellan lokalnät. Datorer och utrustning av olika känslighetsgrad placeras i skilda nät. Det är viktigt att förstå att VLAN enbart är ett logiskt skydd. Det erbjuder inte samma säkerhetsnivå som fysiskt separerade nätverk. Tyvärr har författaren sett exempel i verkligheten där man använt VLAN som en metod att minska kostnaderna för nätverksutrustning och använt samma utrustning för att skapa VLAN på olika sidor om en brandvägg. Detta leder till katastrofala följder om någon gör en felkonfiguration - eller om någon via nätverket kan missbruka konfigurationsinställningarna via SNMP, web, telnet eller liknande protokoll.

Det finns flera leverantörsberoende VLAN-varianter, men en öppen standard är IEEE 802.1q[67]. Flera av VLAN-lösningarna innehåller olika säkerhetstillägg såsom kryptering och autentisering av de kommunicerande parterna.

66 IETF mailing list, maj 2000. http://www-nrc.nokia.com/mail-archive/ietf-spatial/ msg00535.html

[67] http://standards.ieee.org/getieee802/download/802.1Q-2003.pdf

4.2 Brandväggar

I ett grundläggande infrastrukturskydd ingår att skydda det egna nätverket, och därtill ansluten utrustning, mot extern obehörig åtkomst eller datamanipulation. Brandväggar *(eng. firewall)* hör tillsammans med antivirusprogram till de idag IT-infrastrukturmässigt vanligast förekommande skydds- och säkerhetsfunktionerna.

> **Definition 43:** En brandvägg består av en eller flera nätverkskomponenter vars regelverk kontrollerar, begränsar eller övervakar trafikutbyte mellan två, eller fler, nätverk.

Namnet brandvägg har sitt ursprung i den husbyggnadsteknik, med brandskyddsförstärkta väggar, som nyttjas för att hindra och fördröja brandgenomslag mellan byggnader, eller mellan två delar inom en byggnad.

Den viktigaste funktionen som en brandvägg har är att implementera en policy för informations- och nätverkssäkerhet. En brandvägg, liksom en router, används för att koppla samman nätverk men brandväggen är, till skillnad från routern, ursprungligen utvecklad för att vara en skyddsmekanism. Den hård- och mjukvara som utgör brandväggen, tillsammans med nätverkstopologi och det regelverk som brandväggen bygger på, skall tillsammans utforma en åtkomstpolicy mellan nätverken. Åtkomstpolicyn kan vara restriktiv, med fokus på att begränsa trafikflödet, eller så kan inriktningen vara fokus på trafikgenomsläpp men med visst skydd.

Brandväggar kan placeras mellan det egna nätverket och ett externt nätverk. Därför förekommer brandväggar som ett skydd för kopplingen mot Internet hos i stort sätt alla företag och organisationer. Kopplingar mellan det egna nätverket och externa organisationer, exempelvis leverantörer eller samarbetspartners, omfattas normalt också av brandväggar. Brandväggar kan placeras i det egna nätverket för att få bättre intern kontroll av datatrafiken.

Brandväggen ger möjlighet att kontroller mängden nätverkstrafik, trafikriktningar, och till viss del, trafikinnehåll. Allt bör styras av organisationens övergripande informationssäkerhetspolicy i kombination med funktions- och kommunikationsbehov. En genomgänkt och väl definierad säkerhetspolicy är därför en förutsättning för att brandväggar, och andra säkerhetssystem, skall fungera som avsett.

Med hjälp av brandväggar kan en nätarkitektur utformas som består av olika säkerhetszoner. Med säkerhetzoner kategoriseras utrustning ansluten till nätverket, såsom servrar, och nätverket självt, till en av flera zoner. Brandväggens uppgift blir att se till reglera trafiken och att zonen inte blir fullständigt åtkomliga från andra delar av nätverket. En eller fler sådana zoner kan utgöra ett DMZ (demilitariserad zon).

> **Definition 44:** Ett DMZ är ett isolerat nätverk som konceptuellt inte ingår i någon av de andra säkerhetszonerna.

Ansluten till ett DMZ-nät är en eller fler bastionsdatorer som tillhandahåller tjänster mot en eller fler säkerhetszoner.

> **Definition 45:** En bastionsdator är ett system i ett utsatt läge som härdats att stå emot attacker.

Det vanligaste exemplet på DMZ är en Internetansluten brandvägg vars DMZ med bastions-

datorer inte tillhör det interna nätet eller står oskyddade mot Internet. På grund av det utsatta läget bör bastionsdatorer inte innehålla känslig information. Bastionsdatorerna måste underhållas (patchas) väl och övervakas extra noga då de löper en förhöjd risk. Trafik mellan bastionsdatorer och egna system i andra säkerhetszoner, framförallt på interna nätverk, måste begränsas.

För att skyddet som brandväggen erbjuder skall vara verkningsfull måste brandväggen även skydda sig själv mot olika typer av åtkomst och angrepp. En lyckad attack mot brandväggen kan leda till manipulation av brandväggsfunktionen, vilket i sin tur öppnar upp för oönskad trafik. En brandvägg måste därför vara en dedicerad funktion och utrustningen får inte användas för allmän datoranvändning. Så är fallet i de flesta brandväggslösningar, vilket också bör innebära att alla onödiga eller oanvända funktioner i systemet av säkerhetsskäl inte bör vara aktiverade.

4.2.1 Brandväggsprinciper

Brandväggar kan arbeta på olika delar av nätverksstacken. Vissa brandväggar som arbetar på nätverkslagernivån, och ibland överliggande transportlager, i TCP/IP-stacken genom att analysera lågnivå paketinformation. Andra brandväggar arbetar på applikationslagernivån och kan därför göra mer sofistikerad innehållskontroll.

Förutom de uppenbara sakerna som skall brandväggen skall filtrera så finns det ett antal tumregler som bör följas. För att skydda sig mot IP-spoofingattacker så skall man inte tillåta paket från brandväggens utsida som har avsändaradress satt till våra interna nätverksadresser. Detta kallas anti-spoofing eller spoofingskydd. I vissa brandväggar sköts detta mer automatiskt genom att man specificerar vilka nät som finns på det interna nätverket.
Likaså bör man filtrera bort adresser som uppenbart aldrig skall komma igenom brandväggen, exempelvis privata IP-adresser, RFC 1918-adresser, eller multicastadresser. I de fall man eventuellt får paket med sådana adresser, så tillvida man inte försöker använda multicasttjänster, så är det något mystiskt som pågår. Antingen är någon utrustning feluppsatt eller så är man sannolikt utsatt för en attack.

För att vara en god medborgare på Internet kan man filtrera så att brandväggen eller routrar inte släpper iväg trafik med falsk avsändaradress, något som allas egressfiltrering.

Definition 46: Egressfiltering innebär filtrering av all utgående trafik som lämnar ett visst nätverk.

Ofta utför Internetoperatören denna typ av filtrering av falska avsändaradresser från kundnät. När filtrering sker på inkommande trafik kallas detta för ingressfiltrering.

Definition 47: Ingressfiltering innebär filtrering av all ankommade trafik som når ett visst nätverk.

Internetleverantörer följer ofta RFC 2267[68], en beskrivning av hur ingressfiltering bör skötas. Det finns ändå ett incitament att själv utföra denna typ av filtrering och det är att man kan logga förekomsten av denna typ av trafik, vilket sannolikt nätvleverantören inte gör. Genom att fånga, logga och larma i det fall då utrustning på det egna nätverket försöker skicka ut paket med falska avsändaradresser så kan man anta att något är felkonfigurerat, att man haft

[68] Ferguson & Senie (1998). *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing.* http://www.ietf.org/rfc/rfc2267.txt

ett intrång där någon installerat ett program som skickar ut denna typ av trafik, eller att någon internt utför attacker. Av denna anledning så bör denna typ av larm ha hög prioritet så den kan skiljas ut ifrån övriga brandväggslarm och loggar.

4.2.2 Paketfiltrerande brandväggar

> **Definition 48:** Paketfiltrering innebär åtkomstkontroll baserat på attribut i överförda paket som sändaren eller mottagarens IP-adress, TCP-flaggor och portnummer.

Paketfiltrerande brandväggar arbetar på nätverkslagernivån. Enklare typer av brandväggar kan enbart göra statisk filtrering på enskilda paket. Avancerade brandväggar bygger upp tillståndstabeller, skapa dynamiska filterregler och kan förstå paket utifrån ett trafiksammanhang.

Paketfilter förstår oftast paketinnehåll på nätverk- och transportnivå i TCP/ IP-stacken. Det innebär att den förstår och kan filtrera på

- Protokolltyp, tex TCP, UDP och ICMP
- Avsändar- och mottagaradresser
- Avsändarens TCP- eller UDP-port
- Mottagarens TCP- eller UDP-port
- Olika flaggor och attribut i paketen
- Mönster i paketens dataarea

Ofta kombineras flera paketattribut för att skapa en regel, exempelvis TCP-paket tillåts om de är till port 80 på en viss IP-adress.

Många brandväggsprodukter har utökad funktionalitet och kan filtrera på komplexare begrepp. Domännamn eller subnät kan användas istället för enstaka IP-adresser, tjänstenamn kan användas istället för portnummer. Andra utökade funktioner inkluderar igenkännande och blockeringsmöjlighet av omöjliga flaggkombinationer, tvätt och normalisering (eng. *scrubbing*) av paket. Brandväggen kan förstå begrepp som insidesnät och andra externa nät och på så sätt erbjuda spoofingskydd , vilket blockerar inkommande trafik med falska avsändaradresser från våra interna nätverk.

Paketfiltrerande brandväggar kan i sin tur delas in i enkla paketfiltrerade och brandväggar med ett inbyggd tillstånd och möjlighet till skapande av dynamiska filtrerregler, något som ofta kallas för *stateful inspection* eller *stateful packet inspection*, SPI. Enkla paketfiltrerande brandväggar kontrollerar enstaka paket åt gången, vilket gör att de inte får en komplett bild av trafiken. Brandväggar som kan dra slutsatser från flera paket eller låta innehållet ifrån ett paket styra regelverket för andra paket leder till mer avancerade och anpassningsbara lösningar.

4.2.3 Applikationsbrandvägg

En proxybaserad brandvägg arbetar på applikationssnivån i nätverksstacken. Den har alltså kunskap om det aktuella applikationssprotokollet, till exempel FTP vid filöverföring och SMTP vid e-posthantering.

> **Definition 49:** En applikationsnivåbrandvägg arbetar på TC/IP-stackens applikationsnivå och som ofta har förståelse för applikationens protokoll och interna uppbyggnad. Därmed kan en djupare analys och kontroll utföras av överförd information och applikationsstyrdata.

En proxy terminerar normalt sett nätverkskopplet (sessionen) från klienten. Därefter tas data om hand av proxyn, som sätter upp en ny session, från proxyn, till den ursprungliga mottagaradressen.

Enklare proxyimplementationer förstår bara delar, eller inget alls, av applikationssprotokollen. Dessa implementationer är istället utförda så att stora delar av innehållet i trafiken kopieras, mer eller mindre utan kontroll och insyn, i form av enkel bytekopiering mellan de olika nätverkssessionerna. Dessa typer av proxylösningarr ger ingen högre nivå av säkerhet.

4.2.4 Interna brandväggar

Brandväggen ut mot Internet ses ofta som organisationens huvudingång är därför numera en given säkerhetsfunktion hos de flesta. Ett ibland dolt behov, och något som förutses bli ett öka användningsområde, är dock att börja införa brandväggar även i interna nätverk för att segmentera upp gigantiska interna koncernnät där alla kan komma åt datorresurser var ifrån som helst. Inom vissa organisationer har detta förekommit, men i de flesta fall är de interna nätverken okontrollerade nät där i princip all trafik och alla protokoll tillåts.

Interna brandväggar har speciella krav, inte minst ur ett tillgänglighetsperspektiv (de är oftast viktigare att ha hög tillgänglighet på än t.ex. brandväggen mot Internet), prestanda och stöd för protokoll som förekommer i internnätskommunikation. Brandväggar som används mot Internet brukar behöva stödja vissa typer av protokoll bra, såsom webbrelaterad trafik (http, https, etc), e-post och DNS, medans interna brandväggar måsta ha stöd för olika Microsoftprotokoll för nätverksdiskar, utskrifter och liknande.

4.2.5 Brandväggar och yttre anslutningar

Det är önskvärt att alla yttre nätverksanslutningar på något sätt är likvärdigt skyddade. I den fysiska världen är det viktigt att staketet är jämnhögt för att uppnå säkerhet. Att ha en tremetersgrind bara för att vid sidan om den ha 20 cm höga staket medför ingen reell säkerhet. Ett sätt att uppnå någon typ av skydd är att ansluta alla fasta anslutningar, ISDN- och modemanslutningar till olika egna brandväggar. En vanlig extrabrandvägg som kan införas inom olika organisationer och bli en ny typ av brandvägg, en så kallad *partnerbrandvägg*. Ett viktigt argument som talar för olika typer av partnerbrandväggslösningar är att de anslutningar som sker mellan företag ökar i antal. Sammankopplingarna har i många fall också skett mot samma organisationer och företag, till exempel finansinformations-leverantörer, kreditdatabaser och liknande som kommit att bli bortglömda nav i nätverken.

Modempooler är också en akilleshäl när det gäller att uppnå säkerhet i nätverk. Med oskyddade modempooler kan man omedvetet råka släppa in folk i nätet bakom brandväggen. Därför bör man ha en god strategi för huruppringda förbindelser byggs och säkras. Frågor man bör besvara är var placerar man inringningsutrustning såsom accessrouters, ISDN-och modempooler i sin nätverksinfrastruktur. Det givna svaret borde vara i anslutning till någon form av brandväggsfunktion.

Hotet mot modempoolen består framför allt ett telefonnummer kommer på avvägar, en före detta anställd eller konsult fortsätter att använda ingången, någon lånar ut utrustning eller liknande. Ett mer spektakulärt typ av hot mot uppringda anslutningar är ”wardialing”, en metod för att göra rundringningar. Angriparen letar efter modem, för att sedan använda dem som ingångar till en organisation. Det finns flera kommersiella och fria programvaror för att söka igenom telefonnummerserier efter förekomsten av modem, faxar och ISDN-anslutningar.

4.2.6 Personliga brandväggar

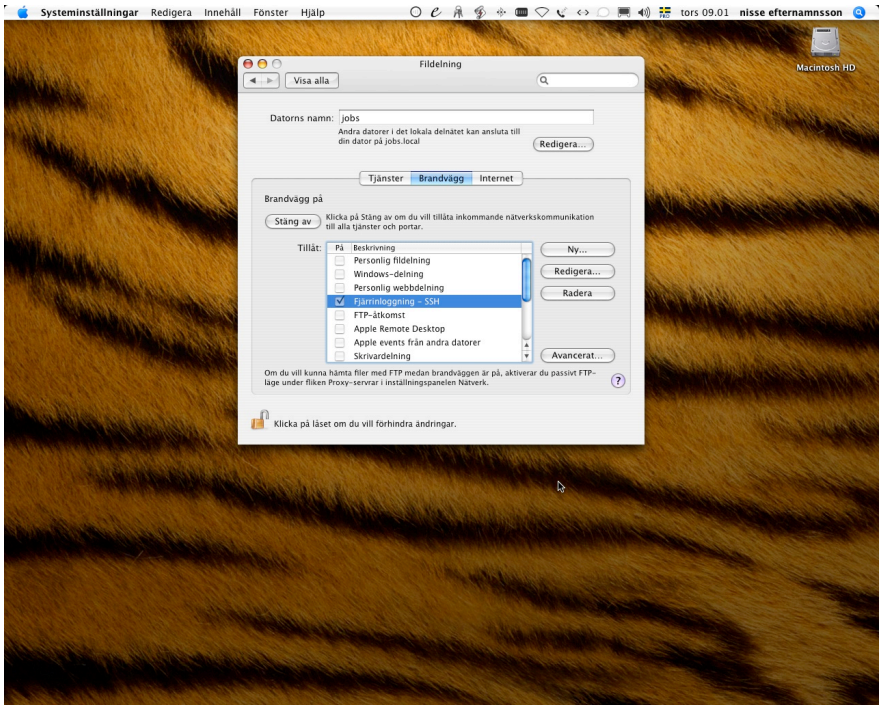
En personlig brandvägg är en programvara som installeras i datorn och som primärt skyddar mot inkommande attackförsök. De flesta personliga brandväggar som finns på marknaden idag är formgivna för hemanvändarmarknaden. Att ha en personlig brandvägg installerad på hemdatorn är något som blivit allt vanligare i takt med den ökade användningen av bredbandsanslutningar - och det faktum att datorer därmed blir kontinuerligt uppkopplade, ofta kallat ”always on”.

> **Definition 50:** En personlig brandvägg är en mjukvarubaserad brandvägg avsedd att skydda en enskild dator eller användare. Skyddet består i att hindra otillåten ankommande trafik eller obehörigt sändande, exempelvis från ett virus, ut från datorn.

Vissa av dessa programvaror är mer avancerade än andra och kan ha inbyggt stöd för exempelvis:

- att skapa krypterade tunnlar, så kallade virtuella privata nät (VPN)
- intrångsdetektering och djupare analys av attackförsök
- kontroll av program i systemet som försöker öppna utgående uppkopplingar
- andra säkerhetsfunktioner såsom filkryptering och virusskydd

I bilden här intill ser vi exempel på olika regler som används i en personlig brandvägg för att skydda en persondator.



Figur 15 Personlig brandvägg på en persondator av typen MacOS X

Personliga brandväggar ligger i en helt annan prisklass än de specialutformade brandväggar som normalt används av ett företag eller en organisation. Priset på en personlig brandvägg ligger ofta i prisklassen under 1.000 kr, medan en brandvägg för att skydda ett helt företag ofta har en prislapp som startar kring 25.000 kr. Riktigt stora brandväggssystem kan kosta runt en kvarts miljon kronor eller mer.

Flera system, såsom Windows XP, MacOS X och Linux har personliga brandväggar förinstallerade som en del i operativsystemet. Det är viktigt att den personliga brandväggen nyttjas konstant eftersom spyware, trojanska hästar, nätmaskar och liknande ständigt är ett problem för hemanvändare, bredbandsanslutna datorer eller på kontoret.

4.2.7 Vad är en brandvägg inte?

Ett av de vanliga säkerhetsproblemen med brandväggar i allmänhet, och i paketfiltrerande brandväggar i synnerhet, är att attacker sker via tillåtna tjänster som släpps igenom brandväggen. Skolboksexemplen är attacker mot webbtjänster, till exempel buggar i Microsofts IIS-tjänst (Internet Information Server).

Trots att man skaffat sig en brandvägg som skydd för nätverket så kvarstår många säkerhetsproblem. Tyvärr tror många att även dessa mer abstrakta problem automatiskt blir lösta i samband med att man skaffat sig en brandvägg. Men brandväggar kan i sig själva vara osäkra, innehålla säkerhetsluckor eller ha alltför öppna standardinställningar. Ibland är de servrar som inte bara används som brandväggar utan också som tjänsteservrar. En brandvägg kan också vara dåligt installerad, till exempel sakna härdning. Dessutom kan den vara dåligt administrerad, till exempel via klartextprotokoll, eller ha en eftersatt administration.

Det är också viktigt att ha i åtanke att brandväggar inte skyddar mot allt. Det finns många exempel på vad brandväggar inte skyddar mot:

- attacker som sker via godkända protokoll
- krypterade tunnlar genom brandväggen
- krypterade data som sänds via klartextprotokoll
- datadrivna attacker
- användares aningslöshetoch dumhet
- nät och datorkommunikation som går andra vägar än via brandväggen

4.2.8 Brandväggens framtid

Man kan fundera över brandväggens framtid. Å ena sidan ser vi situationen med en huvudsaklig brandvägg ut mot Internet där användningen blir svårare och svårare. Användare kräver att brandväggen får allt mer öppna portar, nya protokoll ställer till problem samt tunnling av protokoll, exempelvis Microsoft RPC över HTTP, holkar ut värdet av att öppna eller stänga enskilda portar.

Utökad användning av krypterade protokoll och även okrypterade tunnlingar försvårar också för brandväggen att utföra ett korrekt arbete i perimetern. Å andra sidan kan vi inte heller säga att brandväggen skall plockas bort, för vad finns idag för alternativ, även om vissa intresseorganisationer[69] lovsjunger borttagandet av det elektroniska skalskyddet som sådant.

En sannolikt utveckling som vi redan sett tendenser på är att använda ett distribuerat skydd. Istället för en huvudbrandvägg så får var och en av alla datorer och ansluten utrustning stå för sitt eget skydd. Ironiskt nog så är detta en slags tillbakagång till hur säkerhetsfilosofin var innan trenden var att all säkerhet skall skötas på nätverksnivån. Tidigt i Internets historia så stod var och en ansvarig för säkerheten på respektive ansluten dator, man hade ett hostbaserat

[69] http://www.opengroup.org/jericho/

fokus. Därefter kom den något självbedrägliga tanken att en brandvägg förenklar säkerhetsarbetet så att man inte behöver underhålla och säkra upp varje enskild enhet.

Ett sätt att uppnå detta är att införa distribuerade brandväggar som har centraliserad administration. Detta uppmärksammades i diverse forskningsprojekt i slutet på 1990-talet, främst genom Steve Bellovins banbrytande rapport ”Distributed Firewalls”[70]. Bellovin föreslog att man skulle använda IPSecteknologins möjligheter till paketfiltrering i kombination med dess hantering av elektroniska certifikat och PKI. Genom att med centralstyrda policymekanismer kontrollera filtreringen kan varje nod i nätet skydda sig mot såväl uppkopplingar och sedan kan de använda PKI för att autentisera användare och anslutande system. Speciella certifikat skulle dessutom kunna användas för att intyga att man hade en viss nivå och typ av installerad programvara. Ett anslutande system med alltför gammal eller en programvara som var i strid mot policyn kunde nekas åtkomst till tjänsten även fast den IP-mässigt redan kommit över första hindret – att ansluta till en viss port.

Vissa produkter finns idag på marknaden för denna typ av distribuerade brandväggar. Dock har det primära användningsområdet hittils varit att styra personliga brandväggar på distansarbetsplatser. Då problem med maskar och olika typer av attacker som idag slipper igenom huvudbrandväggen eller helt uppstår bakom denna så kommer behovet tvinga fram en användning av denna typ av arkitektur även inom de interna nätverken.

4.3 Adress- och portöversättning

Tillgången av IPv4-adresser är begränsad och organisationer får i allmänhet inte tillgång till IP-adresser i önskad mängd. För att hantera problemet med adresstillgången nyttjas adressöversättning i vissa brandväggar eller andra system med adressöversättningskapacitet.

> **Definition 51:** Adressöversättning är en automatisk metod att via konverternigstabell förändra adresser i IP-paket som passerar via addressöversättningsutrustningen.

*(*eng. *Network Address Translation, NAT)* För att kunna skicka eller ta emot svar behåller adressöversättaren information i en konverteringstabell om de paket ingår i kommunikationen mellan parterna.

Adressöversättning, som funktion, anses ofta vara en säkerhetshöjande åtgärd. Säkerheten består i att man internt i en organisation använder IP-adresser som inte är publikt tillgängliga. Adressöversättning används till exempel för att översätta IP-paket mellan ett internt nät som använder privata IP-adresser[71] (s.k. RFC1918-adresser) - vilka inte får förekomma ute på Internet - och en eller flera officiellt tilldelade IP-adresser och som är synliga på utsidan av adressöversättaren. Adressöversättning som säkerhetsmetod förekommer vanligt i mindre brandväggsutrustning typ bredbandsrouters eller s.k. SOHO-brandväggar för hem- och småföretagsmarkaden.

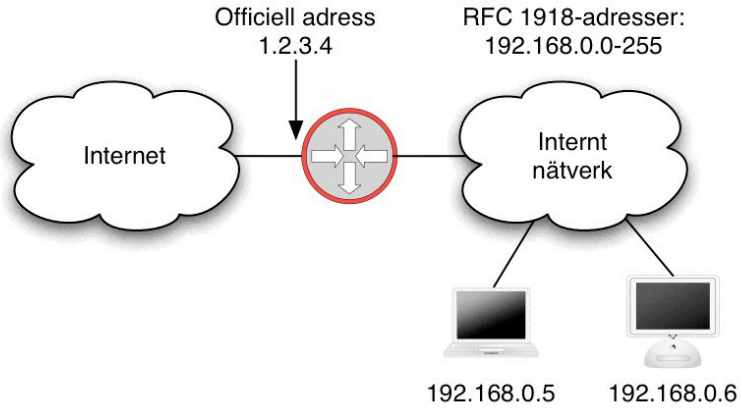
Adressöversättare består av två eller fler nätverksanslutningar. De olika nätvkersanslutningarna har egna IP-adresser. För att kunna skicka eller ta emot svar behåller adressöversättaren information i en konverteringstabell om de paket ingår i kommunikationen mellan parterna. När ett trafikflöde startar så skapas en NAT-regel innehållande

[70] Bellovin, Steven M. ”Distributed firewalls” http://www.cs.columbia.edu/~smb/papers/distfw.pdf

[71] Rekhter mfl. (1996) *RFC 1918 Address Allocation for Private Internets*. ftp://ftp.ietf.org/ rfc/rfc1918.txt

ursprungsadress, ursprungsport, översatt adress, översatt port samt viss metadata såsom tidsstämplar. När svar ankommer den översatta port och IP-adressen möjliggör NAT-regeln översättning och vidaretransport av paketet till frågans ursprungsavsändare.

Det finns olika översättningsstrategier för NAT. En metod är en många-till-många-översättning där varje privat adress på insidan översätts till motsvarande, fast unik, publik adress på utsidan. En annan metod många-till-en-översättning där privata adresser på insidan översätts till en enda publik adress, andressöversättarens egen adress, på utsidan. Se även portöversättning nedan.



Figur 16 Adressöversättare som gömmer intern RFC 1918-nät och utåt nyttjar officiell adress

Många applikationer och applikationsprotokoll fungerar inte med NAT. Applikationer som anropar dynamiskt tilldelade portnummer på adressöversättarens insidesnät fungerar inte. Protokoll som i datadelen skickar med en kopia på domännamn, IP-adresser, portnummer eller snarlik information får problem. Det vanligaste exemplet på en applikation som har problem med NAT av denna anledning är filöverföringsprotokollet FTP, som skickar med IP-adress och portnummer vid start av filöverföring. Ett annat problem med NAT är protokoll som nyttjar kryptering, tunnlar eller kombinationen krypterade tunnlar. Det vanligaste exemplet är problemet med IPSec i transportmod. Några andra exempel är:

- IKE/ISAKMP kräver att avsändaradressen är satt till 500, så att adressöversätta påverkar avsändaradressen.
- Berkley r-tjänster (rsh, rlogin och rcp) kräver en avsändaradress < 1024. Översatta NAT-portar brukar nästan alltid bli > 1024.
- Applikationer som anropar dynamiskt tilldelade portnummer på adressöversättarens insidesnät fungerar inte.

Avancerade brandväggars NAT-funktion brukar även ha specialstöd för standardprokoll, såsom FTP, så att dessa fungerar tillsammans med adressöversättningen. Egenutvecklade eller mindre spridda applikationer kan dock ha problem och måste stödjas via en egenutvecklad proxylösning.

Adressöversättning är i sig inte någon garanti för att man har ett skydd på nätverksnivå. I princip kan adressöversättning tillåta envägs- eller tvåvägsöversättning av nya uppkopplingar. I en tvåvägs, dubbelriktad, adressöversättning sker översättning både för uppkopplingar i alla trafikriktningar, vilket gör att system som anslutits bakom adressöversättaren kan bli åtkomliga. När man installerar en utrustning som använder sig av adressöversättning som ett

skydd av bakomliggande system bör man försäkra sig om att den inte adressöversätter uppkopplingsförsök via inkommande trafik. NAT-utrustning kan för vissa tjänster, såsom webbservers, ofta konfigureras att vara envägs-NAT även för externa uppkopplingar.

En funktion snarlik adressöversättnin är så kallad portöversättning, något som ofta kallas PAT (eng. *Port Address Translation*). PAT kan ses som NAT med metoden många-till-en-översättning där privata adresser på insidan översätts till en enda publik adress, andressöversättarens egen adress.

Portöversättning används till exempel för att flytta eller gömma den verkliga porten för en viss tjänst. Om en webbserver är installerad att använda port 8081 kan den via en portöversättningsfunktion ändå göras nåbar via port 80, standardporten för webbtjänsten.

Om man använder detta i kombination med adressöversättning skulle man kunna ha flera olika servrar installerade på andra portar, tex 8082 och 8083, och tilldela dessa unika publika nätadresser, alla med webbtjänster nåbara via port 80. Detta är ett scenario som ibland används istället för IP-aliases för att ha flera tjänster installerade på en och samma dator.

Ibland kallas PAT även för *IP masquerading* eller *NAT overload*.

4.4 Logisk åtkomstkontroll

Ett standardproblem i nätverkssammanhang är att kunna förhindra obehöriga från att ansluta till det egna nätverket, vilket är något man försöker hantera med hjälp av logisk åtkomstkontroll. I fallet med trådbundna nätverk så innebär det att den obehörige personen fysiskt måste finnas på plats för att ansluta sin utrustning, t.ex. i ett konferensrum eller i en mer öppen lokal såsom en reception eller en vänthall på en flygplats. Det gäller därför att skydda nätverksuttag från anslutningsmöjlighet samt nättjänster från att tillåta trafik från obehörig eller okänd utrustning. I fallet med trådbundna nätverk så finns det inte nätverksuttag på samma sätt, utan anslutningsmöjlighet till själva nätverket som sådant måste skyddas.

Enklare metoder bygger på att nätverks- eller säkerhetsadministratören lägger in spärrar mot okänd nätverksutrustning i form av listor av MAC-adresser som adderas till DHCP-servern. Detta förhindrar att okända nätverkskort tilldelas en IP-adress av DHCP-servern. En angripare kan dock lätt gå runt denna spärr genom att sätta en statisk IP-adress alternativt avlyssna nätverket och därefter stjäla en IP-adress.

En annan metod är att använda låsning av nätswitcharnas portar att enbart tillåta kända nätverksutrustning.

En modern metod för logisk åtkomstkontroll är standarden IEEE 802.1x där nätverkskomponenter såsom switchar kan integreras med mot centrala behörighetsservers och sedan genom att autentisera enskilda switchportar. 802.1x används för såväl trådbundna som trådlösa nätverk. 802.1x använder sig av en variant av EAP, Extensible Authentication Protocol, kallat EAPOL, *EAP over LANs*. EAP är ett autentiseringsramverk framtaget av IETF. EAP finns beskrivet i RFC 3748[72].

[72] Extensible Authentication Protocol (EAP) http://tools.ietf.org/html/rfc3748

En paradox är att i många organisationer så finns det en allmän förståelse för att införa visst skydd på det trådlösa nätverket, medans det för det trådbundna nätverket inte finns samma förståelse av skyddsbehovet.

Facktermer som används för logisk åtkomstkontroll på nätverket är NAC, nework access control, och NAP.

4.5 Honungsfällor

Definition 52: En honungsfälla är en dator eller en tjänst som etableras i syfte att fånga, fördröja eller analysera en angripare.

Honunsfällan (eng. *honey pot)* kan utformas på en mängd olika sätt. Den enklaste och billigaste varianten är en utrangerad dator med ett fritt operativsystem och en gratisprogramvara. Två exempel på gratisprogramvaror är Fred Cohen:s The Deception Toolkit[73], DTK, och Niels Provos *honeyd*[74]. Mer avancerade honungsfällor är kommersiella lösningar från de etablerade nätsäkerhetsföretag som Symantec,

Honungsfällor kan placeras på Internet eller på interna nätverk, beroende på användningsområde. För säkerhetsforskning eller för egen utveckling av säkerhetskunnande kan det vara en idé att placera honungsfällor på Internet. Ett projekt som aktivt arbetar med att analysera data från utplacerade honungsfällor, samt bedriva utbildningar i form av olika analystävlingar är *The honeynet project*[75].

Som vanlig säkerhetsåtgärd är det knappast lönt då den kommer skapa mer arbete än det är värt med en Internetplacerad honungsfälla. Att däremot placera honungsfällor på olika delar av det interna nätverken kan vara en idé väl värd att pröva. Genom att ha honungsfällor som olika typer av försåtsmineringar och snubbeltråd kan de hjälpa till och komplettera övriga tekniska säkerhetsfunktioner.

4.6 Intrångsdetektering

System för så kallad intrångsdetektering kallas ofta IDS (eng. *intrusion detection system*). Ibland kallas nätverksbaserade intrångsdetekteringssystem för NIDS (eng *Network IDS*). Andra varianter av IDS-system är systembaserade (eng. *host based IDS*) IDS-system och applikationsspecifika IDS-system.

En tredje typ av IDS-system är applikationsspecifika IDS-lösningar. Dessa är skräddarsydda för att kunna finna missbruk och attacker mot en viss typ av applikationer.

Intrångsdetektering bygger ofta på två huvudprinciper, regelbaserad eller avvikelsebaserad (anomali) analys. Antingen bygger IDSen en av principerna eller så är det en kombination av metoderna. Regel- eller signaturbaserade IDS-system har förprogrammerade mönster eller signaturer som den känner till. Fördelen med signaturbaserade IDS är att de ger färre falsklarm än avvikelsebaserade system. Nackdelen med regel- och signaturbaserade IDS är att de bara kan användas mot hot som de känner – har signaturer/regler - till. Förutom

[73] http://all.net/dtk/index.html
[74] http://www.honeyd.org/
[75] http://www.honeynet.org/

regelbaserade IDS-system så exister det metoder för att kontrollera beteenden och avvikkande uppträdanden (eng. anomaly based IDS). Dessa system bygger på statistiska modeller över beteende, hur nät, applikationer eller system nyttjas och i viss mån är självlärande utifrån en viss specifik situation eller användarnas egenheter. En fördel med denna typ av IDS är att de kan upptäcka nya typer av attacker för vilka det ännu inte finns signaturer eller inprogrammerade detekteringssätt och därför signaturbaserade IDS-system därmed missar.

Den ideala IDS-lösningen är en hybrid av regelbaserad och avvikelsebaserad detektering. En fördel med intrångsdetektering, speciellt på intern utrustning eller i det interna nätverket, är att metoden även kan hjälpa till mot insiderattacker, när någon internt missbrukar system.

Det finns emellertid flera problem med intrångsdetekteringssystem och de metoder systemen använder sig av. För det första kan krypterade protokoll och krypterad nätverkstrafik omöjliggöra insyn. Det kan till exempel innebära att attacker via HTTP detekteras, medan attacker via HTTPS inte detekteras. Stora nätverk, Internet i synnerhet, innehåller dessutom stora mängder störande trafik, vilket leder till onödiga larm.

IDS-system är ofta svåra att konfigurera så att de hamnar på lämplig detektionsnivå. Antingen missar de väldigt mycket, false negatives, eller också genererar de för många falsklarm, false positives. Vidare kräver ett IDS-system kontinuerlig tillsyn och ett permanent driftarbete, eftersom attacksignaturer måste uppdateras kontinuerligt.

Den viktigaste egenskapen hos ett IDS-system är hur korrekt den analyserar attacker och attackförsök. Det kan krävas mycket intrimning och konfiguration för att undvika falsklarm och få IDS-systemet att följa aktuell säkerhetspolicy. En variant som skall förebygga och eliminera attacker kallas för IPS (eng. *intrusion prevention system*), IDP, IPD eller liknande. IPS-system är en vidareutveckling av IDS-konceptet. I stället för att mer passivt skicka larm eller logga ett angrep eller en policyöverträdelse kan IPS-systemet aktivt ingripa och blockera händelsen.

5 Kryptering

It may roundly be asserted that human ingenuity cannot concoct a cipher which human ingenuity cannot resolve.
— Edgar Allan Poe

I det här kapitlet ska vi gå igenom hur kryptografi fungerar. Men innan vi börjar beskriva kryptografiska metoder och system kan det vara på plats med några definitioner.

Kryptering medger några grundläggande metoder för att skapa säkerhet. Med krypto kan man skapa insynsskydd, skydd mot datamanipulation eller bättre kontroll på vem som är utställare av information eller när denna vart utställd. Tack vare kryptering kan vi skapa många skydd, säkerhetsfunktioner och nya tjänster exempelvis säker kommunikation, förstärkt identifiering och autentisering, krypterande filsystem som skyddar bärbara datorer vid stöld, kryptering och signering av e-postmeddelande samt e-handel.

Definition 53: Kryptografi är vetemskapen om matematiska metoder och tekniker för att skydda information med avseende på sekretess, dataintegritet, autentisering med mera.

Ordet kryptografi (eng. *cryptography*) kommer från de grekiska orden "gömd" och "skrift". Ordets ursprungliga betydelse är läran om tekniker att i en kryptotext dölja information så att den med hjälp av en hemlig nyckel kan återskapas av en behörig användare.

Definition 54: Kryptoanalys är läran om analys av krypotosystem, dess kryptotexter för att utröna kryptonyckel eller klartext.

Kryptoanalys (eng. crypto analysis) används av en utomstående person eller organisation som vill skaffa sig åtkomst till den information som finns krypterad. Kryptoanalys kallas ibland även för *forcering*.

Definition 55: Kryptologi är vetenskapen om kryptografi och kryptoanalys.

Ordet kryptologi (eng. *cryptology*) är det samlingsnamn som beskriver hel området kryptografi och kryptoanalys. Ordet kommer från de grekiska orden "gömd" och "ord".

Definition 56: Kryptering är processen för att omvandla klartext till kryptotext.

Definition 57: Ett kryptosystem är en utrustning eller program som, inklusive nyckelhantering, används för kryptografisk tillämpning

Två vanliga kryptografiska tillämpningar i ett kryptosystem är kryptering och avkryptering. En kryptering utfört av ett kryptosystem kan ses om en transformation vilken har som uppgift att omvandla klartexten (eng. *plaintext*) till kryptotexten (eng. *cipher text*).

5.1 Äldre kryptosystem

5.1.1 Substitutionskrypto

Substitutionskrypto, ibland kallat utbyteschiffer, är ett kryptosystem där klartextbokstäver byts mot andra bokstäver, exempelvis från en tabell, men bibehåller position i medelande. Den enklaste varianten av substitutonskrypton är monoalfabetiska, eftersom bokstäverna ersätts från ett annat alfabet. Ett känt och enkelt exempel är den krypterinngsmetod som den romerske kejsaren Julius Caesar påstås ha använt kryptot i sin tjänsteutövning, därför kallas det för caesarrullning.

Figur 17 Caesarrullning som koncept

Mer avancerade substitutionskrypton är polyalfabetiska, där flera olika alfabet används som källa för utbytesbokstäver. Ett klassiskt exempel på ett äldre kryptosystem är det så kallade Vigenèrekryptot, ibland även känt som *le chiffre indéchiffrable* (det oknäckbara chiffret) eftersom det motstod attacker under många år. Kryptosystemet är uppkallat efter Blaise de Vigenère (f 1523 – d 1596). Systemet består av en tabell, den så kallade Vigenèretabellen som är uppbyggd av ett antal parallella alfabet och är därmed ett polyalfabetiskt substitutionskrypto. Varje alfabet är förskjutet ett steg. Ord används som nyckel. Bokstäver ur nyckelordet ger kolumnen, och i kombination med klartextbokstav i raden får man sin kryptotext.

5.1.2 Transpositionskrypto

Transpositionskrypto, ibland kallas även för omkastningskrypto, är en metod där klartextbokstävernas ordning arrangeras om jämfört med ursprungsordning. Själva kryptonyckeln är reglerna för hur bokstäverna skall flyttas om, att tex första klartextbokstaven skall bli andra bokstaven i kryptotexten och så vidare.

En egenskap av nyttjande av transpositionskrypto är att det existerar lika många tecken i klartexten som i den resulterande kryptotexten.

5.2 Modernare kryptosystem

I moderna kryptosystem nyttjas alltid en kryptonyckel som påverkar krypteringsfunktionen så att enbart den som känner till nyckeln har möjlighet att korrekt återskapa klartext från kryptotext. Likaså är moderna kryptosystem är uppbyggda enbart på matematiska fundament. De kom på allvar att börja användas runt andra världskriget. Med de modernaretiderna följde också intåget av maskinkrypton. Mekaniska och elektromekaniska maskiner började användas för att kryptera och dekryptera information. De tidiga generationerna maskinkrypton använde cylindrar, elektromekaniska rotorelement och liknande för att kunna uppnå det krypterade resultatet av det inmatade eller tänkta indatat.

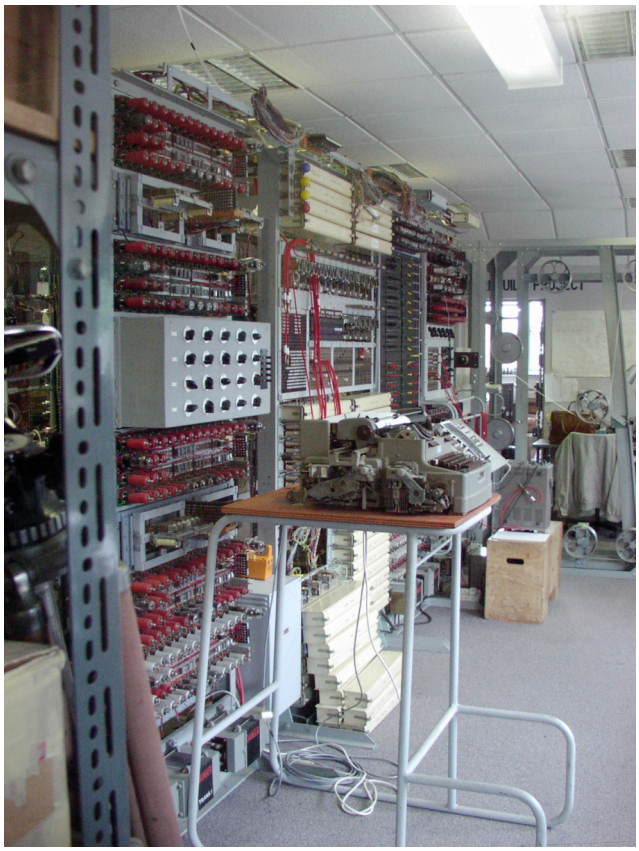
En svensk föregångare på området är Baron Fredrik Gripenstierna som påstås ha byggt den första mekaniska kryptoutrustningen. Enligt funna uppgifter, bland annat kvitton för tillverkningskostnaderna, så presenterades utrustningen redan 1786 för Konung Gustav III.

Vi kommer att diskutera moderna kryptosystem närmare när vi pratar om olika kryptoalgoritmer senare i texten.

5.2.1 Enigma och G-skrivaren

Enigma var ett kryptosystem som utvecklades av idéer från Holländaren Hugo Alexander Koch som först patenterades 1917 i Nederländerna. Den mer entrepenörsinriktade Arthur Scherbius utvecklade ett liknande maskinkrypto som patenterades 1918 i Tyskland. Scherbius köpte Koch patent och införde delar av detta i sin egen maskin under mellankrigstiden. Misslyckade försök med att nå kommersiella framgångar ersattes med att den tyska militären intresserade sig för systemet i mitten på 1920-talet. Enigma kom sedermera att bli ett standardsystem inom armén, flyget och flottan.

Polska matematikervid polska statens chifferbyrå, *Biuro Szyfrów*, med Marian Rejewski (f. 1905 d. 1980) i spetsen hade tidigt börjat analysera systemet, och lyckades sedan början av 30-talet läsa tyska meddelanden krypterade med Enigma. Som en del i sin analys och arbete med att knäcka Enigma så skapade polackerna maskinen Bomba. När Polen blev invaderat av tyska styrkor 1939 så sändes den information som polackerna skaffat sig till England. Arbetet togs där över av brittiska kryptoanalytiker vid *Government Code and Cypher School* (GC&CS) vid Bletchley Park i ett projekt som påverkades av den berömde matematikern och datorpionjären Alan Turing (f. 1912 d. 1954). som senare kunde rekonstruera Enigman och bygga en av världens första elektromekaniska datorer, Colossus och BOMBE, för att automatisera knäckandet av Enigmameddelanden. BOMBE blev klar att använda den 18e mars 1940. Med hjälp av BOMBE och mängder av manuellt arbete kunde engelsmännen följa tysk kommunikation under långa perioder. I samband med att tyskarna gjorde förändringar i rutiner, utökade enigman med extra rotorhjul och liknande, så fick engelsmännen svårt att komma tillbaka och hitta meddelanden som gick att knäcka.



Figur 18 Den återuppbyggda Colossusdatorn vid Bletchley Park

I jämförelse med alla skriverier runt Enigmasystemet och polska, engelska och amerikanska bedifter att forcera systemet så har det skrivits relativt lite om svenska insatser inom området. Känt runt detta är att Försvarets radioanstalt, som bildades under andra världskriget, bland annat lyckades forcera ett annat tyskt kryptosystem. Den tyska utrustningen Siemens T52 i olika versioner, på tyska kallad ”*Geheimschreiber*” och på svenska kallad *G-skrivaren* användes för att skicka strategisk kommunikation mellan tyska högkvarteret och olika fronter. Eftersom det var strategiska planer och uppgifter så var dessa betydligt värdefullare än den taktiska kommunikation som man använde Enigma till. Under världskriget lyckades FRA tillsammans med matematikern Arne Beurling (f. 1905 d. 1986) knäcka G-skrivarnas meddelanden. Genom tillverkning av speciella maskiner, ”appar”, så kunde man maskinellt bearbeta meddelandena. Många anser att Beurlings bedrift, som bygger på att enbart analysera 24 timmars inspelad trafik från den 25e maj 1940, är en större bedrift än någon annan kryptoanalytisk bedrift som gjordes under andra världskriget. Andra forceringar bestod ofta i att man hade kommit över kodböcker, designbeskrivningar, utrustning, klartextmeddelanden eller annan kunskap om systemets konstruktion.

5.3 Steganografi

Definition 58: Steganografi är vetenskapen att dölja kommunikation, att gömma ett meddelande - till skillnad mot kryptografi där meddelandet gjorts oläsbart

.

Det finns flera skolboksexemplet på steganografi. Det klassiska är användandet av hemlig skrift, till exempel gjord av citronvatten. Skriften framträder endast när mottagaren värmer

upp meddelandet, någon som såg pappret innan det blev uppvärmt skulle inte se att det fanns någon text. Ett annat klassiskt exempel på steganografi är akrostikon. Akrostikon är en text, oftast dikt, där radernas begynnelsebokstäver då de lästa lodrätt, bildar ett ord eller en mening.

Moderna varianter av steganografi i den digitala världen bygger på att man gömmer information i en annan informationsmängd. Bilder, ljudfiler och videofiler innehåller redundanta eller oanvända databitar som kan användas för att överlagra information. Man kan med andra ord gömma information i ett bärarobjekt, till exempel oanvända färgkodningsdata i en grafikfil eller som brus i en MP3-ljudfil. Ett annat exempel är att använda oanvända bitar i pakethuvudet hos TCP (TCP-headern) för att överlagra ett meddelande som sänds ut via nätverket.

Användningen av steganografi inom olika områden är ökande. En orsak till steganografins popularitet är att allt fler verktyg finns lättillgängliga.

I samband med terrorattackerna mot USA i september 2001 rapporterade media att terrorister använt steganografi för att utväxla bilder som innehöll hemliga attackplaner. Försök som olika steganografiforskare gjort för att hitta dessa gömda meddelanden i miljontals bilder hämtade ifrån populära webbsidor och ifrån USENET (Internet news) har inte påvisat några sådan gömd information.

Ett annat, mer tillämpbart och legitimt, användningsområde för steganografi är skapandet av digitala vattenstämplar (eng. *digital watermarking*). Vattenstämplar är en teknik som idag börjat användas som en form av kopieringsskydd eller ursprungskontroll för program och data. Genom att lägga till dold information till en fil, till exempel ett serienummer, så går det att avgöra från kopierade filer vem det är som olovligen spridit kopiorna.

En annan fackterm som flitigt används i kombination med, eller istället för, steganografi är ”information hiding”. Hela området med olika typer av dold kommunikation och dolda kanaler har växt och blivit högaktuellt i samband med IT, digitalteknik och datorkommunikation. Arbete och forskning pågår på flera ställen, främst inom amerikanska myndigheter, för att skapa olika typer av brandväggar och detekteringssystem för denna typ av nätverkstrafik med dolt informationsinnehåll.

5.4 Symmetriska kryptosystem

5.4.1 DES

Data Encryption Standard, DES, har sin ursprung i det arbete som Horst Feistel och Donald Coppersmiths grupp vid IBM utförde i början av 1970-talet. I maj 1973 gick National Bureau of Standards, NBS, ut med en allmän begäran om förslag till kandidater till en ny standardiserad federal krypteringsalgoritm. Då inga förslag mötte de grundläggande krav som ställdes gick NBS ut med en ny förfrågan i augusti 1974. Den här gången fick man in kandidater som kvalificerade sig. I juni 1977 antog NBS en modifierad version av IBM:s förslag LUCIFER till att bli en federal standard,. Standarden för DES publicerades i en så kallad U.S. Federal Information Processing Standard, FIPS, med nummer 46. Standardiseringsorganet ANSI antog DES såsom Data Encryption Algorithm, DEA, och beskrev algoritmen i standard X9.32.

DES avsågs vara en skyddsfunktion som skulle kunna användas av myndigheter i USA på känslig information som inte klassades som högt militärt känslig, för vilka andra typer av starkare kryptering skulle användas. Ett krav från NBS var att DES skulle vara effektiv att implementera i hårdvara, vilket innebar att de algoritmer som föreslogs ofta nyttjade arbetsmetoder på bitnivå, något som fick effekt på att mjukvaruimplementationer som ofta blev väldigt innefektiva.

Förlängning av godkännandet av DES som en federal standard, såkallad FIPS-standard, skedde 1983, 1988, 1993 samt 1999. Vid FIPS-förlängningen 1999 skedde en revision där man adderade trippel-DES samt ändrade status för användningen av DES att enbart godkännas för äldre, inte nyinskaffade, system.

DES använder en 56 bitars nyckel, vilket innebär algoritmen teoretisk tillhandahåller ca 7,2 * 10^{16} (72057594037927936 stycken) möjliga nycklar, tillsammans med en 64 bitars blocklängd.

Samma nyckel används både för kryptering och dekryptering, vilket innebär att DES är en symmetrisk kryptoalgoritm. Ett annat ord för detta är ”hemligt nyckelsystem” (eftersom nycklarna måste hållas hemliga). En säkerhetsmässig konsekvens av detta är att man inte bör slita för hårt på en nyckel och inte heller sprida den. Man måste använda unika nycklar för varje kommunicerande part, vilket i sin tur får som konsekvens att man får ett stort antal nycklar att administrera om man kommunicerar med många parter. Samma sak gäller om man till exempel använder DES för att kryptera filer.

DES-algoritmen består av ett antal steg där nyckel och indata omvandlas enligt vissa metoder och regler. De metoder som används i DES, enkel aritmetik och boolsk algebra, är speciellt lämpliga för effektiv implementation i maskinvara. När DES formgavs och standarden antogs var det inte många som hade programvaruimplementationer av algoritmen i åtanke. En av de grundläggande operationerna som återkommer hela tiden i såväl DES som andra kryptoalgoritmer är ”exklusivt OR” (XOR). Efter en inledande förskjutning av data, en permutation, IP, så delas indata (klartext) upp i två block om vardera 4 byte. Därefter följer sexton ”rundor” av omvandling där varje ny runda bygger på resultatet av föregående omgång - och där man dessutom hela tiden byter plats på de olika 4-bytesblocken från föregående runda. Varje runda består av en substitution i kombination med en permutation, dessa operationer brukar i abstrakta termer kallas S-boxar och P-boxar. Efter 16 rundor sker en avslutande permutation.

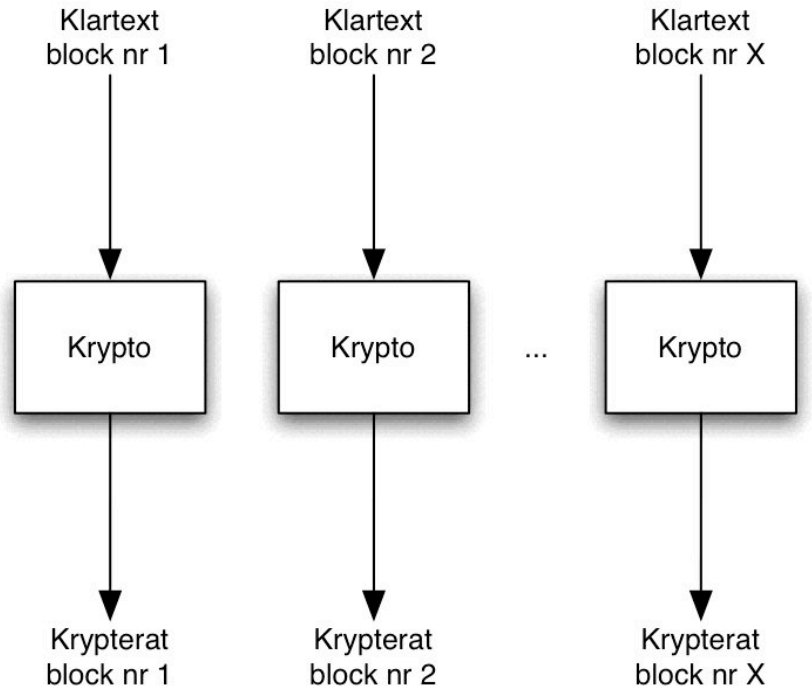
I februari 1998 meddelade webbplatsen distributed.net, som bestod av ett antal frivilliga sm donerade oanvänd datorkraft, att man lyckats knäcka ett DES-krypterat meddelande som ingår i en tävling anordnad av företaget RSA Data Security Inc. Forceringen av meddelandet tog 39 dagar att utföra. distributed.net har analyserat meddelandet med ett PC-klientprogram som installerats på 10000-tals datorer och skickar resultaten till en central server. I juli 1998 knäcker EFF's DES specialutvecklade kryptoknäckarhårdvara *Deep Crack* en DES-nyckel på 56 timmar. Vid tillfället för säkerhetsföretaget RSAs årliga IT-säkerhetskonferens i januari 1999 så används, Deep Crack tillsammans distributed.net för att knäcka en DES-nyckel på 22 timmar och 15 minuter. Med hjälp av dessa tydliga exempel lyckas kryptologerna med sin avsikt – att praktiskt påvisa att DES spelat ut sin roll som effektiv skyddsfunktion. Vid FIPS-förlängningen 1999 skedde en revision där man adderade trippel-DES samt ändrade status för användningen av DES att enbart godkännas för äldre, inte nyinskaffade, system.

5.4.2 Kryptomoder i DES

I FIPS-standard nr 81, beskrivs ett antal moder i vilka DES-algoritmen kan användas. Dessa moder har olika egenskaper både vad det gäller hantering av in- och utdata och vad gäller den kryptografiska behandlingen.

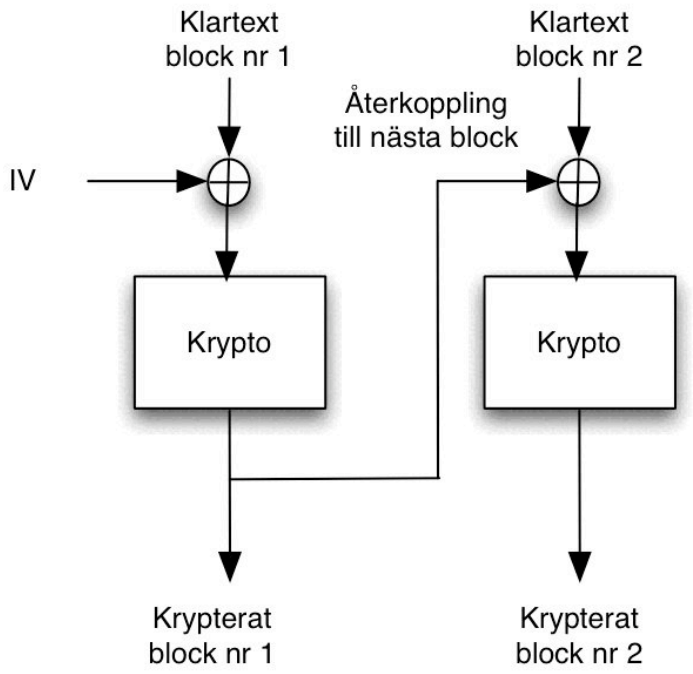
Eftersom DES i sig själv är ett så kallat blockkrypto som arbetar med data i block om 64-bitar (8 bytes) är algoritmen olämplig att använda för att kryptera till exempel interaktiv terminaltrafik när man överför enstaka tangentbordsnedtryckningar åt gången. Med hjälp av de olika DES-moderna kan man ändra vissa egenskaper hos kryptoalgoritmen. DES går till exempel att använda som ett så kallat strömkrypto istället för blockkrypto och kryptera enstaka bitar, byte eller mindre grupper än 64-bitarsblock. En intressant och önskvärd egenskap hos DES är att små förändringar i indata genererar stora förändringar i det krypterade resultatet. Den snöskredseffekt förändringar i indata ska medföra försvårar olika typer av analys av kryptotexten.

I ECB-mod så krypteras varje 64-bitarsblock separat, utan koppling till föregående eller efterkommande datablock. Nackdelen med detta är att det går att utföra en mängd angrepp i blindo, utan att ha insyn i den krypterade informationen. Mot den krypterade trafiken går att utföra återuppspelningstacker, omarrangemangsattacker, etc.



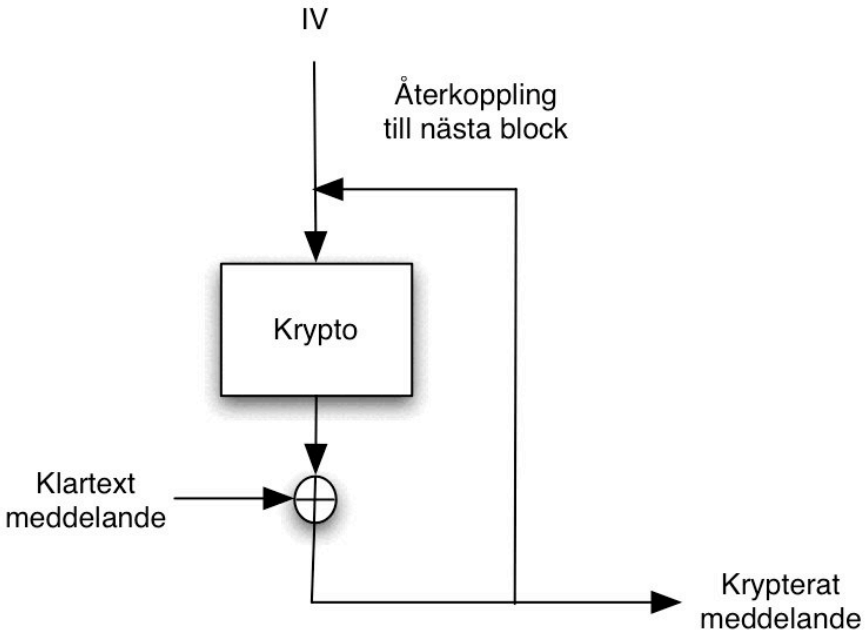
Figur 19 DES i ECB-mod

I CBC-mod, Cipher Block Chaining, så kedjas varje krypterat 64-bitarsblock ihop med föregående kryptoblock. Detta förhindrar angrepp av typen omarrangemangsattacker.



Figur 20 DES i CBC-mod

Genom att använda DES i CFB-mod så har vi från en blockkrypto lyckats skapa ett strömkrypto där det med hjälp av XOR som används för att injicera klartexten i krypteringsströmmen går att använda sig av allt från 1-bit och uppåt.



Figur 21 DES i CFB-mod

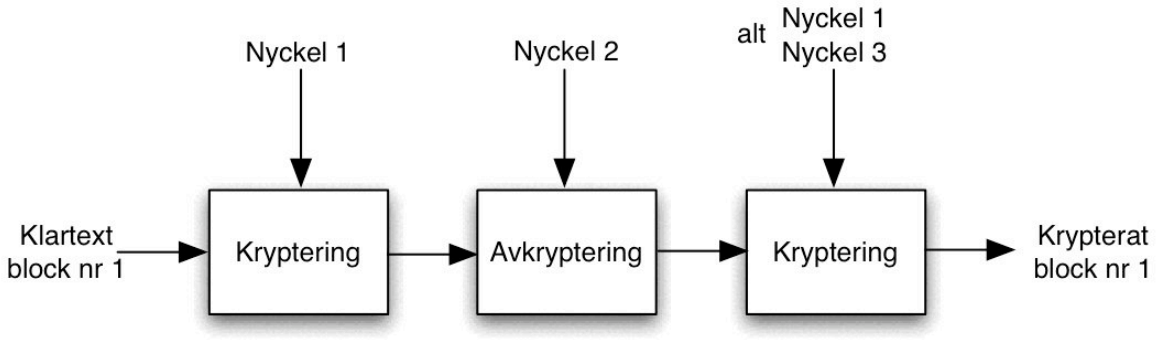
Det har skapats flera andra standarder, bland annat inom bank och finans, som bygger på DES. Förutom användningen av DES för att kryptera data så finns det standarder som beskriver användningen av DES för autentisering, kryptering av lösenord, kryptering av faxinformation och liknande.

De olika moderna som beskrivits för DES går även att använda för andra kryptoalgoritmer, t.ex. AES beskrivet nedan.

5.4.3 3DES

Trippel-DES, beskrivet i standarderna FIPS PUB 46-3 och ANSI X9.52, ofta kallat 3DES, eller TDEA enligt ANSI X9.52-nomenklatur, uppfanns av Walter Tuchman som en metod att stärka säkerheten vid användandet av DES-kryptot. 3DES skapades då DES-kryptot av många ansågs ha för svag kryptonyckellängd för att kunna skydda data under längre tidsrymder.

3DES är implementerat som ett seriellt flöde med operationerna kryptering, avkryptering samt kryptering en gång till, ofta betecknat DES-EDE (encryption, decryption, encryption).



Nyckellängden i 3DES är antingen 112 bitar eller 168 bitar, beroende på om den första DES-nyckeln, nyckel 1 i bilden ovan, återanvänds i det tredje steget eller om en ny 56-bitars nyckel används.

5.4.4 AES

Advanced Enryption Standard, AES, är den kryptostandard som togs fram för att ersätta den åldrande DES-standarden i de säkerhetssystem som behöver ett skydd som kan stå tidens test. AES är en federal amerikansk standard, vilket betyder att den gäller för amerikanska myndigheter, men påverkan av en kvalitativ amerikansk standard kommer som sådan innebär att algoritmen även inkluderas i ett stort antal allmänt tillgängliga produkter och IT-standarder vilket gör att den får internationell spridning.

I slutet av 1990-talet började National Institute of Standards, efterträdaren till NBS arbeta med att ta fram en ny symmetrisk kryptostandard som ersättare för DES. Till skillnad från när DES antogs så var det mer intresse att anmäla sina algoritmer som kandidater. AES-Standarden togs fram i en publik anbudstävling där ett stort antal respekterade grupper av kryptologer deltog med olika algoritmer såsom RC6, Mars, twofish, med flera. Efter denna mångåriga utslagsprocess där alla kandidaterna utsattes för närgången analys så blev algoritm som 2002 valdes till AES-standard ett bidrag kallat *Rijndael*, utvecklad av de belgiska kryptologerna Joan *Daemen* och Vincent *Rijmen*.

AES är ett blockkrypto som arbetar med 128-bitars datablock men har flexibilitet vad det gäller val av nyckellängder och antalet rundor som algoritmen genomlöper internt. De olika kryptonyckellängderna 128, 192 eller 256-bitar är samtidigt definierade att vara kopplade till antalet interna rundor som algoritmen nyttjar - 10, 12 eller 14 rundor.

De flesta vanliga säkerhetsprotokoll som nyttjar kryptering har blivit uppdaterade att fungera med AES, exempelvis så specificerar RFC 3268 en utökning av TLS att använda AES.

Att införa en ny krypteringsalgoritm kan stöta på mängder med problem. Trots att andra kryptoalgoritmer finns tillgängliga, och det existerar flera oberodende implementationer av AES gjord i flera programmeringsspråk, kan det finnas flera anledningar till att vilja fortsätta använda system som bygger på andra algoritmer, exempelvis DES, istället för att byta till nya kryptoalgoritmer. Den främsta invändningen är att AES ur kryptohänseende är en relativt ung och oprövad algoritm. Ett annat skäl är befintliga investeringar i maskinvara och program samt den kunskapsbas kring DES som man byggt upp inom sin organisation. Andra skäl är kompatibilitetsproblem och låsningar mot befintliga protokoll, standarder och produkter.

Specifikationen för AES som en amerikansk standard kallad FIPS-197 är allmänt tillgänglig från http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

5.5 Andra symmetriska algoritmer

En svit av krypton har utvecklats av Ron Rivest kallade RC2, RC4, RC5 och RC6. RC sägs vara en förkortning av ”Ron’s codes” eller ”Rivest Codes”. RC2 är ett blockkrypto med variabel nyckellängd och 64-bitars blockstorlek. RC4 är ett strömkrypto utvecklat 1987 av Rivest. Det har en variabel nyckellängd och jobbar med indata i byteformat. RC4 hör sannoligt till en av de mest spridda strömkryptona eftersom den förkortade 40-bitarsvarianten av algoritmen fick en särskild exportstatus. Tack vare detta så har RC4 blivit standard i många protokoll och produkter. RC5 är ett blockkrypto där man kan använda variabel blockstorlek, variabel nyckellängd och variabelt antal ronder. Blockstorleken brukar vara 32-, 64- eller 128-bitar och nyckelstorleken kan vara upp till 2048 bitar. RC6 är ett blockkrypto som RSA Labs skickade in som kandidat till den nya AES-standarden.

International Data Encryption Algorithm, IDEA, är namnet på ett blockkrypto som använder 128-bitars nyckel på ett 64-bitarsblock utvecklat av James Massey och Xuejia Lai vid Swiss Federal Institute of Technology. IDEA har fått viss begränsad användning, exempelvis i PGP. Patentfrågor och licensavgifter för användningen har gjort att implementatörer istället valt att använda andra algoritmer. IDEA är en vidareutveckling av PES, Proposed Encryption Standard, och IPES, *Improved Proposed Encryption Standard*.

Blowfish är ett blockkrypto utvecklat av Bruce Schneier som klarar variabel nyckellängd upp till 448 bitar. Algoritmen arbetar med 64-bitars block. Eftersom kryptot är patentfritt så är det använt i flera fria programvaror, såsom operativsystemet OpenBSD, säkerhetsprotokollet SSH, filkrypteringsprogram och liknande.

5.6 Asymmetriska kryptosystem

5.6.1 Diffie-Hellman-Merkle

I en artikel publicerad 1976 med titeln "New Directions in Cryptography" av Martin E. Hellman och Whitfield Diffie beskrev de bägge banbrytande idéer som revolutionerade hela kryptologin. Ett koncept som de introducerade var asymmetriska kryptosystem, även kallade öppen-nyckel-krypto (eng. *public key cryptography*) där en nyckel används för att kryptera informationen samt en annan nyckel används för att avkryptera informationen. En annan

viktig upptäckt som beskrevs i texten var ett nyckelutbytesprotokoll där två parter säkert kan utväxla kryptonycklar över ett osäkert medium.

Samtidigt med Diffie-Hellman så parallelupptäckte Xeroxforskare Ralph Merkle metoden för att utföra publik nyckelhantering.

5.6.2 RSA

RSA-kryptot, namngivet efter sina upphovsmän Ronald L. Rivest, Adi Shamir, Leonard M. Adleman, som läste om Diffe-Hellmans upptäckt och de blev som besatta av tanken på att utveckla en ny krypteringsmetod. Efter en viss tid där många metoder och algoritmer föreslogs, analyserades och prövades kom trion så småningom fann en lösing som de inte kunde finna brister med. De beskrev 1976 metoden i en numera klassisk forskarrapport med namnet ”A method for obtaining digital signatures and public-key Cryptosystems” där de vidareutvecklade Merkle/Diffie/Hellmans idéer.

År 1977 blev RSA-gruppens krypto det första publikt kända exemplet på ett asymmetriskt kryptosystem där en nyckel används för krypteringsoperationen och en annan nyckel används för avkrypteringsoperationen - systemet använder ett nyckel*par*. Krypteringsprincipen, ofta kallat publik nyckelkryptering eller öppen-nyckel-krypto, löser ett av de inneboende klassiska problemen med kryptoanvändning - hur man säkert kan distribuera nycklar mellan de parter som skall använda krypteringen.

Metoden som Rivest, Shamir och Adleman utvecklat beskrevs också 1977 i en artikel i den populära forskartidskriften Scientific American. Kolumnisten Martin Gardner erbjöd läsarna 100 dollar till den eller de personerna som skulle lyckas knäcka det RSA-krypterade meddelandet. Utmaningen, som kom att kallas RSA-129, höll i 17 år, trots att det teoretisk skulle vara säkert i många tusen år.

I ett traditionellt krypteringssystem måste nyckeln, som används för kryptering och avkryptering, skyddas väl, annars så kan obehöriga avkryptera översänd information eller falskeligen sända information som krypterats med den stulna kryptonyckeln. I ett kryptosystem som bygger på publik nyckelkryptering, så kan att nyckel som används för kryptering (*den publika nyckeln*) distribueras utan vidare insynsskydd mellan de kommunicerande parterna, ja den bör tillochmed publiceras på väl valda ställen såsom på en välkänd webbsida, för att de som vill nyttja nyckeln vet att den kommer från en legitim källa. En kompletterande nyckel, som måste hållas väl skyddad (kallad *den privata nyckeln*) används för att avkryptera meddelandet.

Skyddet som erhålls av RSA-kryptering bygger på det matematiska faktum att man idag ställs inför svårigheten att faktorisera stora tal. För att utföra RSA-kryptering skapas två stora framslumpade primtal på vilket man gör en serie aritmetiska beräkningar.

Fördelen med RSA som algoritm är att den är relativt enkel jämfört med såväl symmetriska algoritmer som andra asymmetriska algoritmer såsom elliptska kurv-algoritmer. Nackdelen med RSA är att den är beräkningsintensiv då man använder flera hundra siffror långa tal, så kallade bignum, för själva beräkningarna.

Då asymmetriska algoritmer är så beräkningskrävande brukar man implementera dem i kombination med andra mindre krävande algoritmer, ofta symmetriska algoritmer. Den

asymmetriska algoritmen används för kryptering av sessionsnycklar, temporära kryptonycklar, som sedan dekrypteras av programmet och brukas i den snabbare, bulkkryptoalgoritmen.

För sina insatser inom kryptologin tilldelades Rivest, Shamir och Adleman 2002 utmärkelsen ACM Turing Awards[76], vilket anses vara datalogins motsvarighet till Nobelpriset.

Sedan några år har patentet på RSA löpt ut vilket har fått till följd att algoritmen har fått en större spridning. RSA-algoritmen finns använd i de flesta säkra kommunikationsprodukter idag.

5.6.3 Non-secret encryption

Tre forskare James H. Ellis, Clifford Cocks och Malcom Williamson, vid den engelska underättelsetjänstens signalspanings- och kommunikationssäkerhetsbyrå Government Communications Headquarters, GCHQ, upptäckte redan under början av 1970-talet konceptet med asymmetriska kryptonycklar. Eftersom det var personal inom den militära forskningsverksamheten som gjorde upptäckten mitt under det kalla kriget mellan väst- och östblocken hemlighölls resultaten. Detta innebär att det system som Whitfield Diffie och Martin Hellman kom fram till under de nästkommande 30 åren var det som officiellt ansågs vara den första varianten av öppen-nyckel-kryptering.

I slutet av 1990-talet så offentliggjordes upptäckterna som gjorts vid GCHQ och upphovsmännen gavs slutligen viss upprättelse såsom de egentliga upptäckarna av öppen-nyckel-kryptering.

5.6.4 Kryptering med elliptiska kurvor

Det existerar andra publika nyckelalgoritmer, än RSA som är den mest kända. En intressant metod är till exempel kryptering med algoritmer som bygger på elliptiska kurvor, ofta kallat ECC-krypto (eng. *Elliptic Curve Cryptography*). En av ECC-kryptons fördel gentemot RSA-algoritmen är en kortare ECC-nyckellängd som ändå skall medge en med RSA likvärdig säkerhet. Den kortare nyckellängden innebär i sin tur mindre prestandabehov vilket gör att ECC är än mer lämplig i utrustning som inte har snabb processor, exempelvis inbyggda system, handdatorer och mobiltelefoner. Dessutom ställs det mindre krav på lagringsutrymme vilket gör dem användbara för smarta kort och lösningar med liknande inskränkningar i hårdvara.

Något som talar emot ECC är att metoden och teoriera runt ECC inte har funnits lika länge som RSA, vilket gör att man inte kunnat utsätta teoriera för lika mycket attacker och analys. Andra saker som talar emot ECC är att teoriera bakom krypteringen är betydlig svårare än de som bygger upp exempelvis RSA. Förutom att det är svårare att förstå konceptet så är det även svårare att implementera rätt.

[76] http://www.acm.org/awards/turing_citations/rivest-shamir-adleman.html

5.7 Kryptografiskt starka kontrollsummor

En grundläggande komponent i många kryptosystem och säkra kommunikationsprotokoll är kryptografiskt starka kontrollsummor. Dessa kontrollsummor skapas med hjälp av hashfunktioner med vissa särskilda säkerhetsegenskaper.

Definition 59: En hashfunktion skapar ett kondensat, utdata av en fix storlek, utifrån indata av valfri längd

Kondensatet, resultatet som hashfunktioner brukar generera, är för de flesta hashfunktioner i storleken 16-256 bitar, beroende på vald funktion. Som exempel genererar CRC-32 ett 32-bitars kondensat, MD5 ett 128-bitars kondensat och SHA-256 genererar ett 256-bitars kondensat. Ett kondensat kallas även hashvärde, hashsumma eller kontrollsumma.

Ett allmänt krav som ställs på hashfunktioner är att de skall vara datorkraftsmässigt effektiva. Detta betyder att funktionen lätt skall kunna räkna ut kontrollsumman. Men samtidigt skall det vara praktiskt omöjligt att med utgångspunkt från kontrollsumman återskapa ursprungsinformationen samt att den är *kollisionsfri*, att två olika originalmeddelanden inte genererar samma hashsumma.

Några exempel på kryptologiskt starka checksummefunktionerna är MD5, SHA och RIPEMD-160. Det har dock visat sig att de vanligt förekommande kontrollsummefunktionerna MD5[77], SHA-0 och SHA-1[78] har kryptografiska svagheter, bland annat att de inte är så kollisionsfria eller svåra att baklänges räkna fram originaldata som checksummades. Sedan den nya AES-standarden antagits så har nya varianter av SHA-2-familjen, kallade SHA-128, SHA-192 och SHA-256 mfl skapats. Då vissa frågor har börjat ställas runt den nuvarande SHA-utformningen, så har amerikanska standardiseringsorganet NIST utlyst en förfrågan om att införa en ny version, kallad SHA-3[79].

CRC-32, Cyclic Redundancy Checksum, är en hashfunktion som används för att kontrollera dataintegritet. CRC används ofta för att detektera överföringsfel i kommunikationsprotokoll eller korruption av lagrad information. Från en given CRC-summa går det lätt att räkna fram hur informationen, som användes för att skapa kontrollsumman ursprungligen, borde sett ut.

Definition 60: En kryptografiskt stark hashfunktion är vara en envägsfunktion som från ett meddelande skapar ett kondensat, där det från kondensatet är datorkraftsmässigt praktiskt omöjligt att återskapa ursprungsmeddelandet, är datorkraftsmässigt praktiskt omöjligt att leta efter ett annat meddelande som skapar samma kondensat samt vara datorkraftsmässigt omöjligt att leta efter två meddelanden som resulterar i ett kondensat.

MAC, *Message Authentication Code*, är en metod att bestämma ett meddelandes äkthet genom att man nyttjar en delad hemlig nyckel. Genom att lägga till den delade hemligheten, nyckel, till den data som skall beräkas, så kan mottagarsidan verifiera att meddeladet inte kan ha blivit ändrat under färd likväl som att den som har den hemliga nyckeln verkligen är avsändaren. I exemplet nedan är *H* den hashfunktion som används.

H(meddelande, nyckel)

[77] MD5 and MD4 Collision Generators http://www.stachliu.com/collisions.html
[78] http://people.csail.mit.edu/yiqun/sha-crypto2005-talk-distribution.pdf
[79] http://www.csrc.nist.gov/groups/ST/hash/documents/FR_Notice_Nov07.pdf

En variant av denna typ av beräkning med nyckel och kontrollsumma som används i protokoll och kommunikationssammanhang är *HMAC*. Det är en nycklad användning av en hashfunktion, där hashfunktionen nyttjas tillsammans med ett iterativ anrop till hashfunktionen självt samt att man använder XOR-funktionen och utfyllnad (padding) av vissa förbestämnda värden:

H(nyckel XOR opad, H(nyckel XOR ipad, meddelade))

Där ipad är en paddning med värdet 0x36 och opad är en paddning med värdet 0x5C. Antalet tillagda värden är relaterade till nyckellängden och på den valda hashfunktionens blocklängd. Metoden som sådan kan användas med såväl MD5, RIPEM, SHA-1 som andra ”säkra” hashfunktioner. Flera protokoll, såsom IPSec, SSL och liknande använder sig av HMAC.

SHA beskrivs närmare i FIPS PUB 180, FIPS PUB 180-2 samt RFC 3174[80]. MD5 beskrivs närmare i RFC 1321. HMAC beskrivs närmare i RFC 2104 (”Keyed-Hashing for Message Authentication”) och RFC 4634[81].

5.8 Signering

Signering är en operation där man tar en nyckel, tillsammans med en checksummefunktion och räknar ut ett värde på informationen. Normalt är signeringen en omvänd kryptering där man använder den privata nyckeln för skapa *elektroniska signaturer*. Eftersom enbart den legitime nyckelägaren känner till den *privata* nyckeln kan man uppnå ursprungskontroll, ett sätt att
försäkra sig om att meddelandet verkligen kommer ifrån den som påstås ha sänt meddelandet.

Signering är en viktig komponent för e-handel, säkra kommunikationsprotokoll, säkra system och applikationer. Det finns algoritmer framtagna som bara går att använda för signering och inte går att använda förkryptering. DSA, Digital Signature Algoritm är ett exempel på en sådan signeringsmetod.

5.9 Exempel på kryptering - elektronisk post

Två huvudsakliga standardoserade och öppna metoder för att kryptera elektronisk post, S/MIME och pgp. Vi skall titta närmare på dessa två nedan.

Andra metoder existerar, såsom Entrust proprietära lösning med företagsspecifika protokoll och format. Pretty Good Privacy, eller PGP som det mer allmänt kallas, är namnet på ett väldigt spritt krypteringsprogram. PGP finns tillgängligt till de flesta typer av populära plattformar i olika varianter alltifrån integrerade grafiska produkter till kommandoradsversioner. PGP tillkom i början av 1990-talet skapat av Phil Zimmermann för att besvarat ett upplevt behov av ökat elektronisk skydd från personer som oroade sig över den personliga intetriteten och myndigheters inblandning i kommunikation. Misstron mot myndigheter och centralgestalter återspeglas i PGPs design. Upphovsmannen har valt att nyttja en helt annan förtroendemodell än den inom PKI vanliga hierarkiska modellen. Genom att bygga en förtroendenätverk, ett s.k. web-of-trust, mellan de som nyttjar PGP kan man bygga upp förtroenderelationer mellan kommunicerande personer. Modellen fungerar även i fall där människor som inte känner varandra närmare vill kommunicera, men modellen

80 US Secure Hash Algorithm 1 (SHA1) http://tools.ietf.org/html/rfc3174
81 US Secure Hash Algorithms (SHA and HMAC-SHA) http://tools.ietf.org/html/rfc4634

undviker samtidigt den traditionella betrodda tredje parten (eng. trusted third-party) som är certifikatutgivare eller liknande som en PKI brukar innehålla.

I tidigare utgåvor använde PGP algoritmerna IDEA och RSA för kryptering. På grund av en lång rad problem med patent på dessa algoritmer så övergick man i ett senare skede till att använda DSS och ElGamal för signering och kryptering. PGP definierades i och användningsområden beskrevs i äldre RFC, nämligen 1991 (”PGP Message Exchange Formats”), RFC 2015 (”MIME Security with Pretty Good Privacy (PGP)”).

OpenPGP är en standard definierad av IETF i RFC 2440 (”OpenPGP Message Format”), RFC 3156 (”MIME Security with OpenPGP”). GPG, Gnu Privacy Guard, är en fri variant av PGP som implementerar OpenPGP. GPG finns tillgänglig från http://www.gnupg.org

Secure/Multipurpose Internet Mail Extensions, populärt förkortat som S/MIME är en standard för att skicka krypterade och signerade e-postmeddelanden. S/MIME bygger på PKCS #7 dataformat för meddelanden samt X.509v3 för certifikat. Många e-postklienter, inlusive outlook, har stöd för S/MIME-signerade och krypterade meddelanden. Såväl S/MIME och PGP skickas via MIME-kodade epostmeddelanden. S/MIME existerar i flera versioner, främst S/MIMEv2 och S/MIMEv3. Version 2 beror på 40-bitars nyckellängd och RSA-kryptering. Version 3 har utökat skydd och kan använda andra nyckellängder.

Det finns ett antal utökningar till grundprotokollet S/MIME för att erhålla signerade mottagningskvitton, säker e-postlistor.

S/MIME definieras i RFC 2630, RFC 2632 och RFC 2633

5.10 Säkerhetsaspekter i ett kryptosystem

Det finns flera kritiska säkerhetsaspekter att ta hänsyn till när man skapar ett kryptosystem. Den första är valet av krypteringsalgoritm. En väl publicerad algoritm som har klarat av många års analys och kritiskt granskande anses ofta vara ett säkrare alternativ än en ny, företagshemlig algoritm. En annan viktig säkerhetsaspekt är valet av längd på krypteringsnyckeln. Tumregeln är att längre kryptonycklar är svårare att attackera än kortare nycklar för samma eller liknande kryptoalgoritmer. Dessutom innebär hanteringen av nycklar speciella problem. Såväl nyckelgenerering som nyckeldistribution måste vara genomtänkt och väl utförd. Ofta är dessa moment systemets akilleshälar.

Dåligt genererade nycklar kan lätt forceras och svagheter i nyckeldistributionen öppnar blottor hos i övrigt säkra lösningar. Implementationsfrågorna är ytterligare en viktigt säkerhetsaspekt. Det gäller bland annat att inte få dåligt framtagna sessionsnycklar, inte glömma klartextkopior av data någonstans i minnet eller filsystemet och liknande.

Sessionsnycklar byggs ofta med hjälp av slumptal. Är dessa inte tillräckligt slumpmässiga så går det lättare att gissa sig till sessionsnyckeln – man har minskat omfånget på vilka nycklar som skall provas. Slutligen gäller det förstås också att studera hur nycklarna hanteras i datormiljön, till exempel om de lagras på hårddisk eller i datorns arbetsminne. En annan viktig säkerhetsaspekt är att inte återanvända nycklar till andra protokoll eller i andra sammanhang. Ett exempel på farlig användning är att man brukar samma kryptonyckel för att sköta autentisering, elektronisk signering och sedan använder samma nyckel för kryptering.

En lyckad attack mot det ena användningsfallet även leda till missbruk eller åtkomst i de andra fallen. En bättre lösning skulle använda olika nycklar för varje fall.
Det finns mängder av beräkningar på hur teoretiskt säkra olika typer av kryptosystem är. Detta är givetvis en gräns som kontinuerligt ändras med tekniska och vetenskapliga landvinningar. Moores lag ger oss kontinuerligt snabbare och mer kraftfulla datorer. Tekniska genombrott, som till exempel möjligheten att bygga datorer med stöd för flera samtidiga processorer, är av stor betydelse för hur säker en krypteringsmetod kan sägas vara.

5.10.1 Nyckelgenerering

För att använda ett modern krypto behövs det kryptonycklar. Dessa nycklar kan skapas manuellt på förhand, manuellt vid själva användingstillfället, automatiskt före eller vid själva användningen. Ett vanligt sätt är att kombinera ett par på förväg skapade nycklar, ett slags huvudnycklar, och sedan vid behov skapa temporära nycklar. För att uppnå bättre säkerhet brukar man därmed skapa så kallade sessionsnycklar som bara används under kortare tidsperioder, enstaka transationer, enskilda krypterade meddelanden och liknade. Detta istället för att man hela tiden använder och sliter på en och samma huvudnyckel.

Fördelen med att ha sessionsnycklar är att inte alltför mycket data blir krypterat med en och samma nyckel. Om stora datamängder krypteras med samma nyckel underlättar detta olika typer av analyser och attacker mot kryptosystemet.

5.10.2 Slump och slumpens betydelse

Förvånande nog så har slump en central betydelse i moderna kryptosystem. Kryptoalgoritmer av god kvalitet skall producera kryptotext som vid analys är svår att skilja från slumpdata – dvs det skall vara svårt att hitta strukturer i informationen. När man hittar strukturerad data så gör det mycket för att kunna attackera algoritmen eller implementationen av algoritmen. Slumpmässigt framtagna sessionsnycklar kräver att den framslumpade nyckeln är så oförutsägbar som möjligt. De slumptalsgeneratorer som normalt sätt ingår i operativsystem, i rutinbibliotek och liknande, är ofta extremt olämpliga att använda i kryptosammanhang eftersom de skapar så kallade pseudoslumptal. Dessa standardfunktioner kallas pseudo-random number generators, eller PRNG kort och gott. Just den anledningen, att det är svårt att skapa bra slumptal, har föranlett Internets standardiseringsorgan, IETF, att ta fram en speciell, informativ beskrivning som kallas ” Randomness Recommendations for Security”. Den har publicerats som RFC 1750 och ger vägledning för implementation av funktioner som bygger på enstaka slumptal. Andra klassiska beskrivningar om slumptal finns i Donald Knuths böcker om algoritmer, boken om nummeriska metoder.

I moderna operativsystem, såsom OpenBSD och Linux, finns det speciella pseudodevice, tjänster, kallade `/dev/random` eller `/dev/urandom`, från vilket applikationer och system kan begära slumptal av god kvalitet. Dessa tjänster byggar på insamlandet av många källor av dynamisk information, såsom hur många paket man läst från nätverket och statuset på en visst CPU-register, vilket gör att den sammantagna informationen blir en bra slumpkälla. Då slump har en så central betydelse så har flera tillverkare av elektronikkretsar börjat tillverka speciella komponenter för att producera bra slumptal eller brus. Vissa tillverkare av moderkort har integrerat dessa i sina system.

5.10.3 Kerckhoffs princip

Auguste Kerckhoffs, vars fulla namn är det majestätiskt klingande Jean-Guillaume-Hubert-Victor-Francois-Alexandre-Auguste Kerckhoffs von Nieuwenhof, gav 1883 ut boken ”La Cryptographie militaire”. Boken är av många ansedd som en av de främsta böckerna i ämnet. I boken har Kerckhoffs lagt fram en av de grundläggande teorierna för moderna kryptosystem, nämligen att sekretessen måste bygga på säkerheten i nyckeln, inte på sekretess om själva systemet, till exempel vald kryptoalgoritm. Systemets säkerhet skall bestå även om motståndaren får kunskap om metoder, algoritmer och annat om själva krypteringsprocessen. Obskyritet runt detta får inte vara det som bygger säkerheten. Säkerheten måste bestå om man bara håller nyckeln väl skyddad. Detta har blivit känt som *Kerckhoffs princip*.

5.10.4 Nyckelbyten

Ett vanligt antagande är att man kan undvika attacker baserade på uttömmande sökning genom att ändra sessionsnycklar regelbundet. Detta är något som inte helt och hållet stämmer. Kontinuerliga nyckelbyten, baserade på tid, antal överförda byte eller liknande kan leda till ökad säkerhet. Ett problem är dock om någon kan observera när ett nyckelutbyte sker. Denna kunskap kan underlätta försöken att knäcka kryptot. Man behöver i genomsnittbara söka igenom halva nyckelrymden för att hitta nyckeln. Nyckeln kan ligga i den första, men också i den sista procenten nyckelutrymme.

Om nyckelbytet går att observera gör man helt enkelt en omstart på sitt sökande. Om man försöker attackera ett kommunikationsprotokoll som nyttjar en krypteringsalgoritm som skulle gå att göra en uttömmade nyckelsökning på i medeltal efter 75 minuter, man hittar nyckel i genomsnitt efter halva tiden. Detta är inte ett orimligt exempel under vissa förutsättningar – korta nyckellängder eller dåligt framslumpade sessionsnycklar. Det innebär att hela nyckelrymden kan sökas igenom 150 minuter, men man har ibland tur och hittar vid vissa sökningar en nyckel som ligger i början av nyckelrymden men vid andra gånger måste man göra en längre genomsökning. Om vi har kunskap om när nyckelbyte sker och att det är exempelvis var 15:e minut så är sannolikheten att vi kan finna en nyckel under dessa 15 minuter 15 / 150 dvs 1/10. Detta innebär att vi i exemplet kan göra en ny nyckelsökning varje gång ett nyckelbyte sker och ändå ha 10 procents chans att lyckas fånga fånga rätt nyckel. Pricipen som framgår av exemplet är att man ändå måste ha en stark kryptoalgoritm och använda långa kryptonycklar även om man använder nyckelbyten eftersom metoden att byta nycklar inte är en fullgod ersättare för nyckel- och algoritmstyrka.

5.10.5 Kryptotyper

Modera kryptoalgoritmer faller i en av två kategorier - *strömkrypto* (eng. *stream cipher*) och *blockkrypto*. Ett blockkrypto arbetar med indata i form av block som ofta är 64 eller 128-bitar långa. Strömkrypton arbetar med indata av valfri längd, även fast vanliga avändningsområden är att arbeta med en byte åt gången eller ibland en bit åt gången. Generellt kan man säga att strömkrypton brukar vara snabbare än blockkrypton och de kan implementeras med mindre kod. En fara med strömkrypton är återanvändning av sessionsnycklar då det öppnar upp för attacker. Angriparen kan kombinera de två krypterade dataströmmarna och få fram ett resultat som är kombinationen (XOR) av de två klartextmeddelandena. Från detta stadium är det betydligt enklare att fortsätta angreppen än att angripa kryptotext. Om det är samma meddelande som sänts två gånger och samma nyckel använts så går det enkelt att ta fram det ursprungliga meddelandet. Ett sätt att kringå problemet med återanvändning av exakt samma nycklar är att använda ett framslumpat initieringsvärde (eg initialization vector), IV, samt

återkoppling av kryptotexten till nyckelströmmen. I fallet med IV så gäller samma sak igen, att man inte får återanvända IV-värdet tillsammans med nyckel.

Detta kan tyckas vara en självklarhet men här finns det existerande protokoll och säkerhetslösningar som är bevis på att man inte lärt sig denna läxa. Wired Equivalent Privacy, WEP, som används tillsammans med trådlösa nät bygger på RC4-kryptot som är ett strömkrypto. Där använder man en 40-bitars (eller 104-bitars) kryptonyckel tillsammans med ett 24-bitars IV som ändrar sig för varje paket som sänds. Diverse orsaker påverkar IV-beräkningen och i praktiken börjar man använda IV-värden som resulterar i att man läcker ut viss nyckelinformation efter så lite som 5000 sända paket.

5.11 Attacker mot kryptosystem

Det finns mängder med olika typer av attacker mot kryptosystem. Kryptoanalys, ibland kallat forcering på svenska, är vetenskapen om hur man knäcker kryptosystem. Det finns en rad olika attackprinciper som en kryptoanalytiker kan använda. Den ”raka” metoden går via så kallade dumma attacker, till exempel en *uttömmande sökning* (eng. *brute force*) där varje möjlig nyckel systematiskt provas tills man hittar en nyckel som kan dekryptera informationen. Ofta kräver uttömmande sökning stor datorkapacitet. Trots stora datorresurser kan sådana sökningar i teorin kräva att bearbetningen pågår oavbrutet i tusentals år.

Det finns också så kallade enkla attacker. De ”mannen-i-mitten”-attacker som vi beskrivit tidigare är ett exempel på en enkel attack, där man går in och placerar sig mellan de kommunicerande parterna.

Andra enkla, men ofta effektiva attacker kan bestå i stöld av nycklar eller kryptoutrustning eller mer klassiskt kriminella ansatser såsom utpressning, hot och mutor. Dessa metoder kallas ibland för black bag jobs. De avancerade attackerna brukar delas upp i flera distinkta grupper. För det första har vi attacker där angriparen har kännedom om såväl klartext som kryptotext (eng. *known plaintext attack*). En variant av detta är en attack under egen påverkan av valet av klartext (eng *choosen plaintext attack*). Ibland kan denna påverkan vara förvånande enkel att utföra, till exempel om man själv kan generera den nätverkstrafik som skickas i en krypterad tunnel vars nyckel man vill hitta. Dessutom har vi andra attack-kategorier som de statistiska attackerna linjär ochdifferentiell kryptoanalys respektive tidstagningsattacker (eng. *timing-attacker*). Vanligtvis så tänker gemene man på attacker mot kryptosystem som matematiska analyser där man letar efter logiska misstag i de matematiska antagade som bygger upp kryptots styrka. Differentiell kryptoanalys kan representera denna typ av attack.

Andra, ofta praktiska och förbisedda attacker, riktar sig mot implementationsdetaljer hos själva lösningen. Timing-attack är en attack där man observerar hur lång tid det tar att utföra de kryptografiska operationerna. En tidstagningsattack är en så kallad sidokanalattack (eng. side- channel attack) då den bygger på observerbar relaterad sidoinformation till själva kryptooperationen. Andra exempel på sidokanalsattacker kan bygga på observation av utrustningens, exempelvis ett smart cards, strömförbrukning eller hur utrustning beter sig under vissa speciella förhållanden, exempelvis bestrålning.

En viktig sak att känna till om attacker baserade på uttömmande sökning är att de i genomsnitt kräver att man söker igenom halva nyckelrymden för att hitta korrekt nyckel. Men den korrekta nyckeln kan i teorin ligga alldeles i början av nyckelrymden, vilket gör att man hittar nyckeln nästan omedelbart. Nyckeln kan också ligga i slutet.

I augusti 1999 lyckades en grupp personer faktorisera det som kallas RSA155, vilket motsvarar en 512-bitars RSA-nyckel. Arbetsinsatsen för detta motsvarade 7500 PC som arbetade en dag eller en PC som jobbade 20 år.

5.12 Kryptoacceleratorer och särskild hårdvara

Ett sätt att kunna bygga lösningar som klarar hög trafikvolym, många samtidiga uppkopplingar samt mycket krypterad och dekrypterad data är att använda någon typ av kryptoaccelerator.

En metod är att installera speciella kryptoacceleratorer i serverdatorn som sköter vissa eller alla typer kryptooperationer. Dessa installeras ofta i form av ett instickskort, via USB- eller SCSI-port och liknande. En annan metod är att installera en särskild utrustning på nätverket som transparent inte bara sköter krypteringen utan även förstår kommunikationsprotokollet. Denna metod, som ibland kallas inline accelerator, kan till exempel transparent omvandla SSL skyddad HTTP-trafik, dvs HTTPS, till vanlig HTTP-trafik innan den når servern. Den uppenbara fördelen är att man inte behöver göra några förändringar i mjuk- eller hårdvara i den befintliga lösningen. En annan fördel med att använda en *inline accelerator* är att man kan sätta ett IDS-system på det korta, i övrigt isolerade nätverket, mellan acceleratorn och servern.

Ibland kallas varianter av kryptoacceleratorer och krypteringshårdvara för Hardware Security Modules (HSM), Host Security Modules (HSM) eller kryptocoprocessor. En HSM skapar, sparar och hanterar kryptonycklar och utför olika kryptooperationer.

Förutom själva prestandaförbättringen kan olika kryptohårdvara erbjuda andra fördelar såsom utökat skydd. Viss kryptohårdvara skapar nycklar och förvarar dessa i eget minnesutrymme och låter aldrig dessa nycklar lämna hårdvaran, till skillnad från när man i mjukvara skapar nycklar som sedan ändå finns lagrade i vanligt RAM-minne och på disk.

Säkerheten på HSM-utrustning kan definieras av att utrustningen har certifierats mot FIPS, Federal Information Processing Standard, 140-1. Den högsta nivån är för närvarande FIPS 140-1 nivå 4, som bland annat beskriver höga krav på obehörigt åtkomstskydd och manipulation av fysiska komponenter (eng. *tamper resistance*).

En nackdel med hårdvarukryptoutrustning är att de kanske inte stödjer alla de algoritmer som applikationen eller protokollet nyttjar. Befintlig hårdvara lär till exempel inte ha stöd för AES som kommer få en utökad användning. I bästa fall kan man göra en uppgradering via leverantörens uppgraderingsprogram eller genom att ladda ner ny mikrokod. I värsta fall måste man investera i ny hårdvara.

Olika typer av attacker mot hårdvarumoduler inkluderar attacker mot själva API:et som används för att hantera hårdvaran, mot buggar i själva hårdvaran, fysiska attacker mot hårdvaran och liknande. Många hårdvarumoduler har därför inbyggda skydd mot attacker och tömmer minnesareor eller har någon form av självdestruktion vid ett angrepp.

5.13 Viktiga lärdomar för kryptoanvändning

Den kanske absolut viktigaste lärdommen för den som skall använda krypto är använd inte något som du själv uppfunnit och tror är säkert eftersom du själv inte klarar av att knäcka det. Många metoder som man kan tänka ut själv visar sig vara olika metoder som redan blivit uppfunna, i vissa fall för flera hundra år sedan, i form av olika substitutions och transpositionskrypton. Det finns attacker, hot och befintliga analysverktyg mot i stort sett alla dessa klassiska krypton.

Lika viktigt är att inte bli lurad att använda något företagsspecifikt, ”proprietär”, kryptolösning. Istället för att vi själva hittat på något som vi tror är säkert så är risken nära på 100-procentig att vi istället sätter vår tillit till någon annan som gjort precis samma sak. Det är vanligt att dessa algoritmer och produkter enbart är tillgängliga i färdigkompilerad form och att det inte går att få åtkomst till källkod eller designdokument för analys av den egentliga nivån av säkerhet. Ofta kallas denna typ av krypton för ”snake oil” och förlöljigas i olika forum, exempelvis i Bruce Schneiers månadsbrev ”Crypto-gram”. Under rubriken ”the dog house” lyfts olika leverantörer fram och deras utfästelser om att produkten ger säkerhet dissikeras och analyseras.

Lösningen för någon som vill nyttja krypton är att använda en väl etablerad algoritm som blivit implementerad av någon eller några som har stor erfarenhet inom området. Algoritmen, designkrav och specifikationer bör vara publikt tillgängliga. Ju längre tid de varit tillgängliga och desto mer som hunnit analysera eller testattackerat algoritmen, desto bättre. En annan viktig lärdom är att bara för att inte någon hittat en svaghet eller en säkerhetsbrist i en kryptoalgoritm betyder det inte att den är säker. Det enda det betyder är så enkelt som att ingen hittat något ännu. Algoritmen kan vara säker, men det är inga garantier. Andra viktiga lärdommar är att en teoretiskt bra algoritm kan utsättas för en usel implementation vilket öppnar för svagheter. Likaså kan en bra implementation användas till ett dåligt designat protokoll eller en dålig lösning, jämför med hur RC4 används i WEP, där det egentligen inte är algoritmens fel att man har blottor utan att själva protokollet nyttjar algoritmen på ett dåligt och osäkert sätt. Var hela tiden kritisk mot implementationsfrågor, det är ofta där felen finns.

Lika viktigt som att man väljer en stark och väl implementerad kryptoalgoritm är att man bygger en lösning som är lättanvänd i samband med att den nyttjar säkerhet. Om den är oanvändnarvänlig är risken stor att folk försöker sätta säkerheten ur spel, finna genvägar eller nyttja andra lösningar.

Den bästa lösningen på detta är att under själva designfasen se till att den säkra vägen, det säkra valmöjligheten och liknande alltid skall vara det mest lättillgängliga valet och det som attraherar en användare mest.

Som man enkelt ser av ovanstående diskussion så är säkerheten i kryptofrågor något helt annat än det som folk i allmänhet fokuserar på – nyckellängden.

6 PKI

Public Key Infrastructure, PKI, är ett samlingsnamn för större system för hantering av publika nycklar och uppsäkrade tjänster som byggs upp detta. Det finns mängder av företagsstandarder, ad-hoc standarder och internationella standarder inom PKI-området. Vi ska gå igenom några av dem här.

Det är viktigt att förstå vad en PKI kan och inte kan erbjuda. Tyvärr är det vanligt med en övertro på PKI-lösningar som i delar kan skyllas på många års missvisande marknadsföring och att begreppet PKI är så pass luddigt och tolkas väldigt annorlunda av olika personer.

6.1 Certifikat och certifikathantering

Ett certifikat är en bindning mellan en publik nyckel och en unik identifierare, till exempel ett namn eller ett personnummer. Man brukar skilja på mjuka certifikat, där certifikatets innehåll vanligen sparas som en datafil på en hårddisk, och andra sätt att lagra och hantera certifikat, ofta i form av aktiva kort (eng. smart card).

Aktiva kort erbjuder i allmänhet en säkrare och mer hanterbar lösning vid användning av certifikat och kryptonycklar än vad mjuka certifikat gör. Det råder emellertid en generell, teknisk övertro på aktiva kort, vilket gör att många tror att aktiva kort inte går att lura, hacka eller missbruka.

Som den kände säkerhetsgurun och Cambridgeprofessorn Ross Anderson påpekar i boken ”security engineering” så finns det mängder med publicerade metoder och attacker för att ge sig på kortbaserade lösningar. Beroende på korttyp kan man utföra elektriska attacker, mekaniska attacker och olika typer av programvaruattacker.

Ett certifikat brukar bestå av följande komponenter:

- den publika nyckeln
- en unik identifierare för innehavaren av certifikatet
- namnet på utfärdaren av certifikatet
- en elektronisk signatur från utfärdaren
- ett serienummer
- tid och datum för utfärdandet av certifikatet
- livslängd på certifikatet

Ett EG-direktiv som blivit svensk lag har inneburit skapande av en speciell typ av certifikat, kvalificerade certifikat. Kvalificerade certifikat får enbart ges ut till fysiska personer, till skillnad mot de certifikat vi beskrivit tidigare som kan ges ut till datorresurser likväl som till juridiska och fysiska personer. Kvalificerade certifikat beskrivs närmare i lagen om kvalificerade elektroniska sigaturer (2000:832). I Sverige finns dessutom såkallade medborgarcertifikat som ges ut av olika spelare inom bankväsendet, posten med flera. Medborgarcertifikatet är ett slags certifikat som är knutet till personnummret som fungerar mot flera samhällstjänster såsom kontakter med riksskatteverket, RSV.

6.2 X.509

En vanligt förekommande standard för certifikatformat heter X.509. Den är en del av ITU-T-standarden X.500. Den finns i flera versioner, men den mest förekommande idagsläget är

version tre kallad X.509v3. I bilden på nästa sida visas innehållet från ett certifikat i X.509-format. X.509s certifikatformat innehåller vissa fält för utgivare, serienummer, publik nyckel, giltighetstid, identitet, utgivarens eller utställarens signatur.

En arbetsgrupp inom IETF jobbar med att ta fram standarder och komponenter för att möjliggöra PKI-strukturer baserad på Internetstandarder. Namnet på denna arbetsgrupp är pkix, Public-Key Infrastructure (X.509), och de återanvänder viss teknik och standarder från ITU, man gör såkallade profiler, medans andra standarder, såsom OCSP är egenutvecklade protokoll och standarder. Certifikat baserade på X.509 används av protokoll såsom SSL/TLS, S/MIME och IPSec. Ett av standarden X.509 styrkor, men också samtidigt en av dess största svagheter, är möjligheten att använda utökade certifikatfält, s.k. *X.509 extensions*. Utökningarna skapar flexibilitet och möjligheter som ursprungspersonerna inte tänkt på, men orsakar samtidigt förvirring och oklarheter hur man skall hantera fälten i programvaror som inte har stöd för utökningarna. Det finns flaggor som beskriver krav på fälten huruvida de skall tolkas som kritiska eller icke-kritiska utökningar. Kritiska utökningar skall enligt standarden behandlas för att certifikatet skall godkännas medans icke-kritiska utökningar kan ignoreras. Flera implementationer och tolkningar missar detta och orsakar ovanstående oklarheter. Exempel på utökningar är KeyUsage som inskränka hur nyckeln får användas såsom för att den publika nyckeln i certifikatet är till för kryptering (keyEncipherment). En annan variabel för KeyUsage kan indikera att certifikatet enbart är till för kontroll av signaturer (digitalSignature).

6.3 Certifikatutgivare

En utgivare eller godkännare av certifikat brukar kallas för Certification (ibland även uttytt Certificate) Authority, CA. En CA är den funktion som intygar ett certifikats korrekthet, hanterar och återkallar (revokerar) certifikat. Ibland är har CA även rollen som utgivare av certifikat, skapare av kryptonyckelpar, arkivering av nycklar och återskapare av förlorade nycklar. Underförstått i detta sammanhang är att CA har kontrollerat identiteten för den person, organisation som vill ha certifikat hanterat.

En CA brukar inneha rollen som en betrodd tredje part (eng *trusted third party*), en spelare som godkänner (signerar) certifikat så att två parter som inte känner varandra ändå kan förmås att lita på varandra. Vad en betrodd tredje part gör är att intyga att en publik krypteringsnyckel eller signaturverifieringsnyckel hör ihop med en identitet, till exempel ett personnamn. Ett certifikat som signerats av en certifikatutgivare eller betrodd tredje part brukar ha en förtroendekedja (eng. trust path). Kedjan, som bör kontrolleras och verifieras av certifikatanvändare, skall innehålla betrodda parter från själva certifikatet upp till rooten.

En CA som är den högsta instansen i en PKI, en root-CA, kan i princip inte få sitt certifikat signerat av någon annan än sig själv och använder därmed ett självsignerat certifikat (eng. self-signed certificate). Eftersom man som certifikatutgivare ofta får agera en betrodd mellanhand är det viktigt att man deklarerar exakt vad certifikattjänsten innebär, vilka krav en certifikatanvändare kan ställa och vilka tjänstegarantier utfärdaren ställer. Dessa brukar beskrivas i ett antal viktiga procedurer och rutiner som styr utgivning och hantering av certifikat, främst

- Certificate Practise Statement, CPS.
- Certificate Policy, CP.

Den mest kända kommersiella certifikatutgivaren är företaget VeriSign som

har lyckats skaffa sig en särställning, närapå monopolställning, för vissa typer av certifikat. Det gäller till exempel servercertifikat för SSL-baserad http- kommunikation.

RFC 2527 beskriver vilka viktiga beståndsdelar som skall ingå i en CPS. I standarden X.509 definieras närmare vad en CP är.

6.4 Certifikathantering och katalogtjänster

För att få en publik nyckelinfrastruktur att fungera i praktiken måste man lösa problemet att distribuera certifikat mellan användare. En ofta använd metod är att inrätta centraliserade lagringsarkiv med sökmöjligheter, så kallade katalogtjänster. Antingen kan katalogentjänsten adminstreras av den egna organisationen eller i de fall då man exempelvis nyttjar en extern utställare av certifikat så kan man kanske även nyttja en extern katalogtjänst.

Ett vanligt protokoll för att söka efter och hämta certifikat och certifikatinformation är Lightweight Data Access Protocol, LDAP. Det används sällan eller aldrig direkt av användaren utan är ett protokoll som används bakom kulisserna av tillämpningar som behöver kontakta katalogservern. LDAP använder sig av TCP-port 389. LDAPS är en variant av LDAP som körs över sessions-tunnlingsprotokollet TLS (IETF-världens motsvarighet till Netscapes SSL) för att erbjuda en högre nivå av säkerhet. LDAPS använder sig normalt av TCP-port 636.

En allt vanligare katalogtjänst är att använda Microsoft AD, bland annat via LDAPS. Om man inte skall bygga en stor PKI-lösning kan man i vissa fall nyttja manuell certifikatdistribution. Detta kan även vara nödvändigt när man skall utväxla certifikat med extern part som inte är del i en specifik PKI-lösning men som ändå måste få tillgång till ett certifikat, exempelvis för att kunna utbyta uppsäkrad e-post.

6.5 Certifikatkontroll och revokering

Ett certifikats giltighet måste kontrolleras. En kontroll är att se att certifikatet ej är ändrat eller trasigt. När ett program kontrollerar ett certifikat så bedömmer man dessutom om det har en korrekt utställare eller är underskriven av en CA som bedömmaren känner till och litar på. För att göra detta så måste bedömmare kunna följa en fullständig förtroendekedja (eng. *trust path*) mellan certifikatet och utgivaren.

Förutom den information som ingår i certifikatet om livslängd och liknande så kan ett certifikat även bli spärrat, varpå en tjänst inte bör acceptera vidare användning av certifikatet. Att spärra ett certifikat kallas för revokering. För att kontrollera om ett certifikat är spärrat måste man jämföra det med en spärrlista, en så kallad certifikatrevokeringslista (eng *certificate revocation list*, CRL). Det kan antingen ske genom att en spärrlista laddas ner till datorn eller genom en så kallad online-verifiering. En CRL skapas av en CA och publiceras regelbundet.

Hur ofta en spärrlista av det här slaget skapas, om det är var 15e minut, en gång per dygn eller veckovis, har att göra med vilken policy certifikatutgivarenhar valt. När ett certifikat spärras så adderas certifikatets serienummer till spärrlistan. En nackdel med CRLer är att det förflyter tid mellan uppdateringarna. Om spärrlistorna skapas veckovis och ett certifikat spärras dagen efter det att en spärrlista skapats, så kommer de som kontrollerar certifikat att behöva vänta sex dagar innan de blir medvetna om att certifikatet är ogiltigt. CRLer finns att hämta från

särskilda CRL-distributionspunkter, ofta kallade CRL DP. Många gånger åligger det den som erbjuder en tjänst där certifikat nyttjas att göra rätt typer av spärrlistor och certifikatinnehåll.

Då CRL:er har en förmåga att växa mycket över tid och bli stora och ohanterliga har man uppfunnit delta-CRL:er där bara information som ändrats sedan föregående leverans distribueras. Efter en första bas-CRL så sänds bara förändringarna i efterkommande uppdateringar.

För att lösa vissa typer av problem med återkallande och kontroll av certifikat försöker man idag skapa mer realtidsanpassade lösningar. Det finns dock flera problem med online-verifiering. Man kan till exempel råka ut för skalbarhetsproblem, och man kan få driftsäkerhetskänsliga lösningar. Och hur skyddar man egentligen infrastrukturen mot falska frågor och förfalskade svar?

Ett protokoll för online-verifiering är *Online Certificate Status Protocol*, OCSP. Ett program måste ha en så kallad OCSP-klient för att kunna hantera detta protokoll. När ett certifikat tas emot så åligger det klienten att kontrollera certifikatets riktighet, både via OCSP-frågor och annan verifiering av certifikatets innehåll. Från en OCSP-server, en såkallad OCSP-responder, kan man få tre typer av svar på en statusfråga rörande certifikat. Det första alternativet är ”good”, vilket innebär att certifikatet vid frågetillfället inte var återkallat. Detta innebär dock inte att man får någon garanti för att certifikatets giltighetstid är ok, eller att certifikatet överhuvudtaget är utgivet av den efterfrågade CA-servern. Detta är något som frågeställaren själv får verifiera. Det andra alternativet är ”revoked”, som innebär att certifikatet är återkallat och spärrat. Det tredje alternativet, ”unknown”, innebär att den utfrågade servern inte känner till certifikatet ifråga. Ett fjärde typ av svar man kan få ifrån OCSP-servern är felmeddelanden som kan vara relaterade till att frågan är felformaterad.

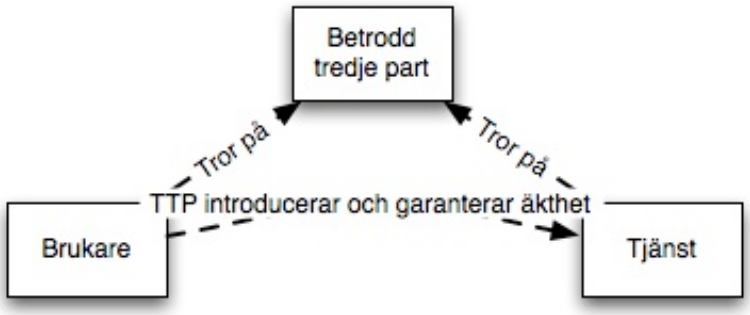
6.6 PKI-arkitekturer

En förutsättning när man skall skapa en PKI-lösning är att välja viken typ av PKI-arkitektur man skall välja att implementera. Skall vi ha en platt eller hierarkisk arkitektur? Skall våra root-servrar vara tillgängliga (on-line) eller frikopplade (off-line)? Skall lösningen vara helintern eller skall det finnas externa knytningar? Skall folk ställa ut sina egna certifikat eller skall det vara omgärdat av administrativa åtgärder? Vi skall på de efterkommande sidorna titta närmare på vilka alternativ det finns och vad de ger för olika för- och nackdelar.

6.6.1 Förtroendemodeller

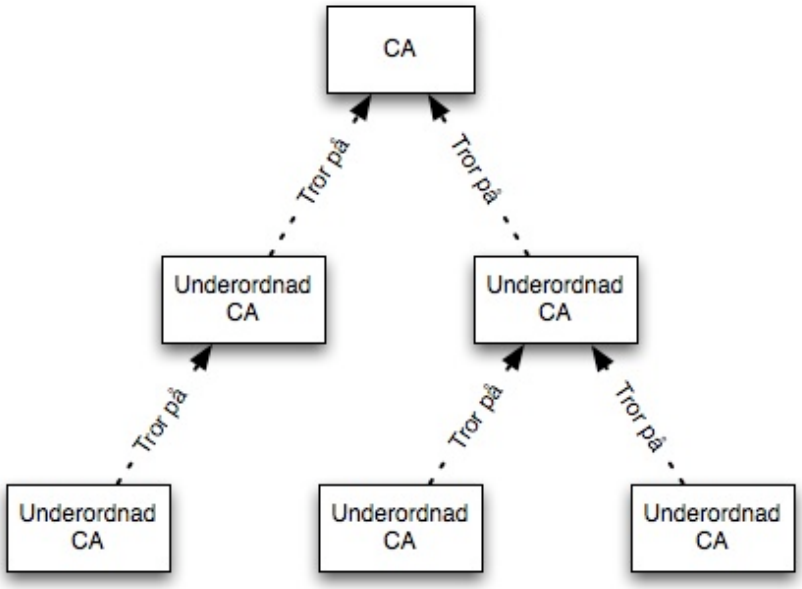
Antingen kan man skapar en PKI-lösning som enbart består av en enstaka root-CA som agerar utställare eller man bygger upp en stor PKI-lösning med flera CA, RA och andra komponenter måste man ha en förståelse för konceptet och på förhand veta vilken typ av lösning man vill skapa. Man måste tidigt i ett PKI-projekt bestämma vilken typ av förtroendemodell man skall följa.

I bilden visas ett exempel med en betrodd tredje bart som både brukaren och tjänsteleverantören litar på. Tjänsteleverantören har skaffat ett certifikat från TTPn och detta medför att brukaren litar på riktigheten att tjänsteleverantören är verkligen är den som leverantören utger sig för. Detta scenario beskriver det vanliga fallet i webbvärlden där man har ett servercertifikat men saknar klientcertifikat.



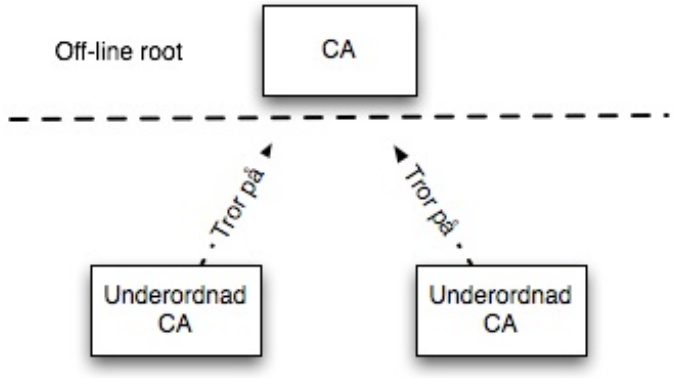
Figur 22 Scenariot är en enkel hierarkiskt struktur där användaren brukar en tjänst. Tjänsten är certifierad av en betrodd tredje part.

PKI-lösnigen består antingen av enbart en root-CA eller en hierarki av CA- tjänster, så kallade underordnade CA (eng. *subordinate CA*). Genom att iföra underordnade CA så kan man skapa såväl avlastning, redundans och en teknisk lösning som möjligen mer är anpassad för organisationens struktur. En drivkraft för att införa underordnade CA är att geografiskt eller organisatoriskt spridda enheter själva skall ha möjlighet att påverka och administrera CA-tjänsten.



Figur 23 PKI hierarki med rot-CA och flera nivåer av underordnade CA

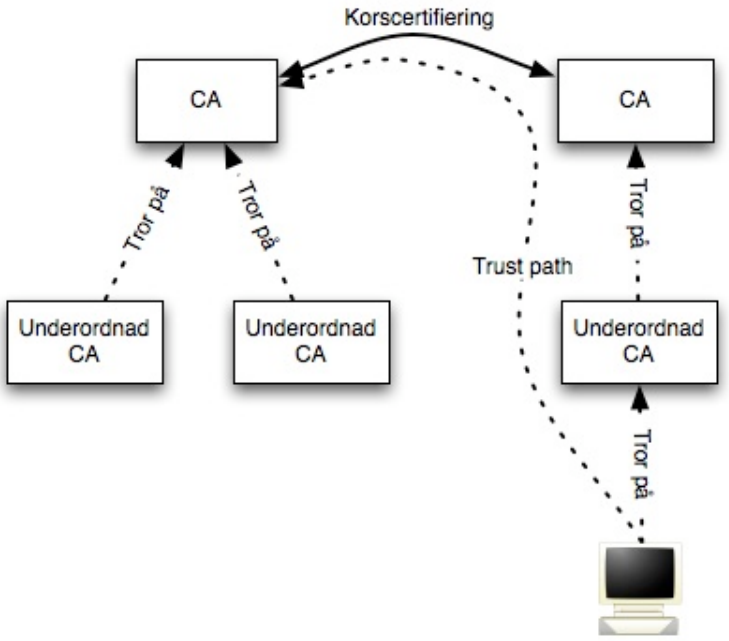
När man bygger en hierarki av CA-tjänster går det att bygga en lösning där root-CA är bortkopplad, en så kallad off-line CA. Fördelen med denna typ av konstruktion är att CAn och all information som hör ihop med den är bortkopplad från nätet vilket försvårar angrepp och missbruk.



Figur 24 I detta scenario är CA-tjänsten som sådan avkopplad ifrån nätet som en extra försiktighetsåtgärd.

Skall denna root-CA vara en organisationsintern CA eller skall man söka bygga en PKI som nyttjar en extern CA och som därmed kan fungera som en betrodd tredje part (trusted third party, TTP) vid extern kommunikation? Det beror på vilka krav och vilken säkerhetspolicy man har.

Vid skapandet av en PKI-arkitektur kommer man för eller senare ställas inför behovet att kommunicera med andra PKIer utanför den egna sfären. Hur skall man då kunna lita på de certifikat som utställs i den andra PKIn? En lösning på detta är korscertifiering (eng. cross certification). Med korscertifiering så skapar man ett särskillt förtroendeförhållande mellan de två PKI-strukturerna där respektive CA skapar en överrenskommelse att lika på certifikat och nycklar på samma sätt som om man själv var utställare. Praktiskt innebär det att man skapar korscertifikat som utväxlas mellan de olika CA-tjänsterna och de två CA signerar varandras publika nyckel.

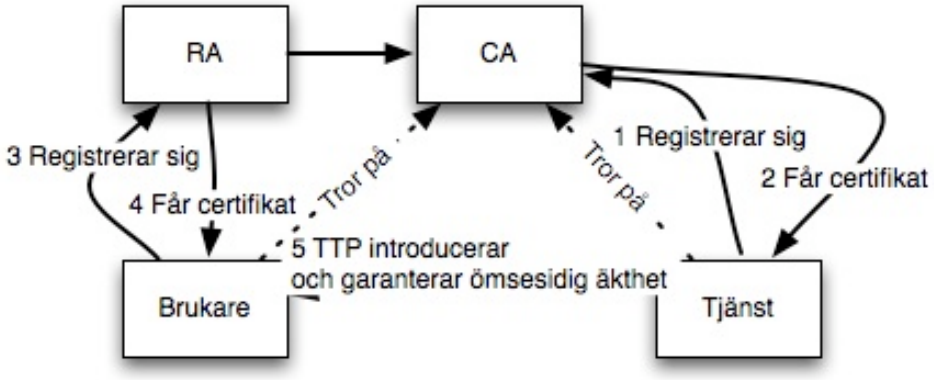


Figur 25 Scenariot beskriver ett särskillt uppsatt förtroendeförhållande, en så kallad korscertifiering.

Faran med korscertifiering är att man omedvetet kan skapa tillit till andra, okända, PKI-strukturer. Genom överförbart förtroede (transitative trust) kan jag omedvetet börja lita på andra PKI-strukturer som den jag korscertifierar mig mot också har valt att i sin tur korscertifiera sig mot. I vissa typer av PKI-teknologier kan detta begräsas med olika typer av inställningar i CA-utrustningen.

En annan viktig enhet inom en PKI är Registration Authority, RA. Det är en administrativ enhet som ansvarar för vissa delegerade uppgifter I samband med registrering av nya användare eller hantera befintliga användare i en PKI. Ett vanligt scenario när en RA används är när det är avstånd mellan CA:n och de som vill registrera sig.

I bilden visas ett exempel på en PKI-struktur där både en CA och en RA är inblandade. I första skeende så skaffar tjänsteleverantören certifikat och därefter skaffar sig brukaren ett certifikat för att sedan nyttja tjänsten. I detta exempel används ömsesidig autentisering där såväl servern som klienten kontrollerar motpartens äkthet och behörighet.



Figur 26 Detta scenario innehåller en RA, en registreringsmodul som brukaren nyttjar istället för att avändaren skall gå manuellt in till

6.7 PKI, implementation och verklighet

I många fall då man arbetar med PKI-frågor faller snabbt de tekniska frågorna in i skyggan av de alltmer dominanta politiska, organisatoriska och administrativa frågorna. Vem eller vilket bolag skall ha kontroll över den interna CA-funktionen? Vilken typ av teknisk lösning skall man välja för PKI? I princip finns det fyra huvudspår – antingen köper man kommersiella PKI-tjänster, väljer kommersiella PKI-produkter, väljer Microsofts PKI-lösning eller så bygger man en egen lösning med hjälp av open source-verktyg. Vilket som passar är lika mycket tycke och smak, vilka behov man har samt slutlige en fråga kom erfarehet, kompetens och budget.

Behovet av olika tjänster måste vara ett grundläggande krav vid PKI-införande då det styr hur PKIn skall byggas. Vad skall göras PKI-kapabelt? Skall vårt tidsredovisningssystem utökas så att tidrapporterna signeras elektroniskt? Skall kommunikationen mellan våra ekonomisystem skyddas bättre än idag med hjälp av signerade och krypterade transaktioner? I de flesta sådana fall så kräver denna typ av PKI-stöd i tjänsterna att nyutveckling görs för att göra dessa PKI-medvetna. Kommersiella PKI-tjänster och PKI-produkter har ofta licensieringsförfaranden

som baserar sig på antalet utställda certifikat, antal installerade PKI-program och liknande. Vid stora installationer kan det bli kostsamt att nyttja detta alternativ, speciellt om man har behov av flera certifikat per användare.

Microsoft har med Windows 2000 lite grann i smyg introducerat en PKI-teknologi på Windowsplattformen. Denna PKI har haft ett antal svagheter i jämförelse med flera kommersiella varianter som har haft flera års försprång att bygga upp sina tekniska plattformar och kunnande. I samband med Windows 2003 Server och XP har Microsoft vidareutvecklat sin PKI-lösning och har numera en plattform som är ett attraktivt alterativ till att såväl bygga en egen PKI som att välja en tredjepartsprodukt. En fördel som många företag har vad det gäller Microsoft är att de redan har koncernlicenser och att Microsoft inte har extra licenskostnader per utställda certifikat eller för PKI-moduler. Har man behov av mycket extern kommunikation så är det viktigt att de externa parterna lätt kan komma åt kataloger, har rätt programvara samt möjlighet att verifiera våra certifikat. I detta fall kan det vara vettigt att nyttja någon stor extern aktör, den vanligaste aktören är VeriSign, att ställa ut certifikat, och sedan nyttja program som kan importera och bruka dessa certifikat, exempelvis Netscapes eller Mozillas webb- och mailläsare.

Det är viktigt att täcka in alla bitar när man inför en PKI. Det är ofta lättare att ställa ut certifikat än att kontrollera riktighet och giltighet. Det finns flera exempel i Microsofts programvara som Internet Explorer och andra program, att man glömt eller ignorerat de viktiga stegen att kontrollera certifikatens egenskaper. Windows update struntade i att den nedladdade programvaran blev signerad med ett utgånget certifikat. När någon okänd person lyckades lura VeriSign att ställa ut och leverera ett programutgivarcertifikat utställt i Microsoft namn så blev det stora problem. Det fanns ingen funktionalitet på plats iWindows eller dess appliationer att spärra det falska certifikatet. Detta är misstag man kan lära från andra och sedan inte falla i själv när man skall införa en PKI-lösning.

Att bygga en helt egen PKI-lösning kräver utvecklarkompetens men kan leda till kostnadseffektiva och skräddarsydda lösningar. Det finns idag öppenkällkodslösningar med OpenSSL, OpenCA, OpenLDAP och andra kom- ponenter vilka kan användas för att bygga en fullvärdig PKI-lösning.

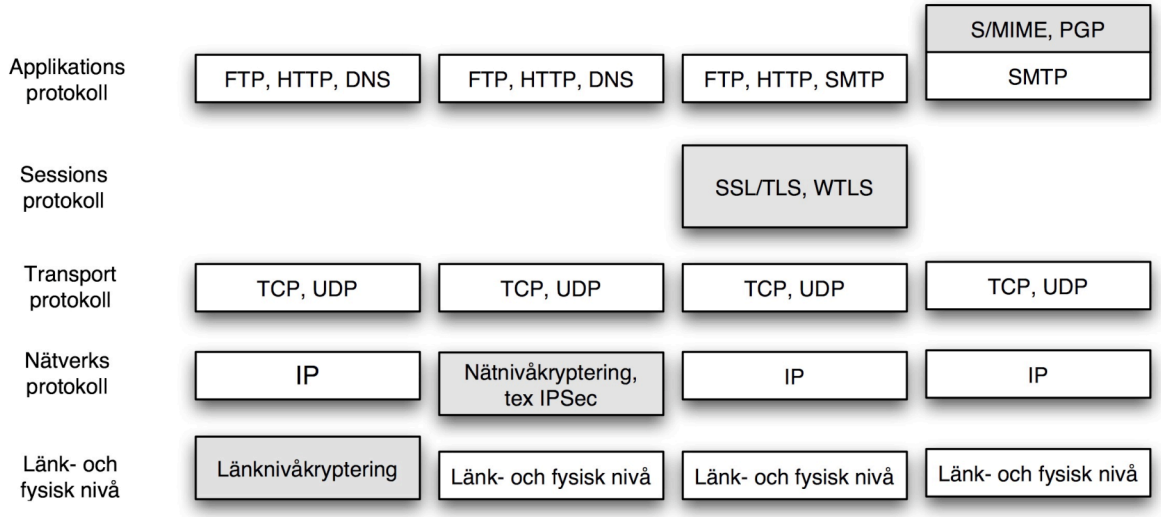
6.8 PKCS

En annan viktig samling av PKI-relaterade standarder kallas PKCS, *Public Key Crypto Standard*. PKCS är en samling företagsstandarder utgivna av RSA Data Security. PKCS har dock blivit accepterade som allmänna branschstandarder för olika bitar inom PKI. PKCS består av standarder för format på nycklar, filer och liknande samt för gränssnitt mot exempelvis smartkortsläsare.

7 Protokoll

Om man känner fienden och känner sig själv, behöver man inte frukta utgången av hundra slag.
Om man känner sig själv och inte fienden, kommer man för varje seger också att lida ett nederlag.
Om man känner varken fienden eller sig själv, kommer man att duka under i varje slag.
-- Sun Tzu, Om krigets konst

Kommunikationsprotokoll är en grundläggande komponent för att skapa ett informationsutbyte mellan olika IT-system. Äldre protokoll, skapade utan säkerhet, används dagligen i olika kommunikationssammanhang, exempelvis filöverföringsprotokollet FTP eller fjärrinloggningsprotokollet Telnet.



Bilden ovan visar schematiskt var i protokollstacken olika säkerhetsprotokoll hör hemma.

I detta avsnitt skall vi mer ingående beskriva några vanligt förekommande säkerhetsprotokoll; Kerberos och IPSec. Kortare beskrivningar finns av protokollen SASL, TLS och SSH. Kapitlet innehåller en utförlig beskrivning av protokollet DNS, som i sig inte är ett säkerhetsprotokoll, men vars nyttjande, eller missbruk, kan få stor påverkan på säkerheten i IT-systemen.

7.1 Kerberos

Med början i slutet av 70-talet och under en större del av 80-talet drevs projekt Athena vid universitetet MIT i USA. Inom universitetetsvärlden hade vid denna tidpunkt arbetsstationer och lokala nätverk börjat ersätta terminalanslutna stor- och minidatorer. Den nya datorparken bestod till stor del av en klient- och servermiljö med operativsystemet UNIX och kommunikationsstandarderna Ethernet och TCP/IP. I en öppen universitetsmiljö, med nyfikna studenter eller forskare och mängder av stora lokaler som innehöll datorer som tidvis kunde vara utan översikt, innebar denna nya typ av datormiljö säkerhetsmässiga utmaningar.

Obehöriga eller studenter skulle kunna installera falska tjänster, avlyssna nätverket, stjäla lösenord, återuppspela avlyssnad trafik (med inloggningsinformation) eller manipulera information som skickades över det delade nätverket. Ett av resultatet från Athenaprojektet är autentiseringssystemet Kerberos. Från att ha varit en speciallösning för ett enskilt universitet så har Kerberos har fått en ökad betydelse från 80-talet fram till dessa dagar, då rroblemet de underliggande koncepten var satta att lösa har visat sig finnas även på andra ställen, exempelvis inom ett företag.

Definition 61: Kerberos är ett nätverksorienterat autentiserings- och nyckeldistributionssystem med central nyckelhantering som ger uppsäkrad autentisering mellan användare och olika tjänster som användaren nyttjar.

Namnet Kerberos kommer från den grekiska mytologin och är namnet på den trehövdade vakthund som kontrollerar ingången till dödsriket, Hades.

Principen för protokollet i Kerberos är en vidareutvecklad variant av Needham-Schroeder-protokollet. En förenklad beskrivning av protokollet, där *Alice* vill kommunicera med *Bob* och TTP agerar betrodd tredje part som håller alla nycklar, visas nedan.

I Kerberos kan såväl användare som resurser och tjänster autentiseras, därför kallar man dessa med ett samlingsnamn för *principal*. Arbetet Kerberos utför är att kryptografiskt kontrollera identiteten på de olika principalerna. Som vid traditionell terminalinloggning kontrollerar systemet användaridentiteten vid uppkoppling. Vid nätverksmässig inloggning är användaren dock betjänt att kontrollera att uppkopplingen skett mot den avsedda, legitima tjänsten, inte en falsk tjänst. Därför utförs en ömsesidig, tvåvägs, autentisering.

Det finns flera typer av principaler, främst användarprincipal, tjänsteprincipal samt principaler för Kerberos autentiseringsserver. Varje principal innehåller grundinformationen namn, instans samt realm. Principalens namn är ett användarnamn eller ett namn på tjänsten. Instans kan vara roll eller attribut kopplat till namnet. Ett exempel på en principal kan därmed vara `rom.root@EXEMPEL.COM` och `rom/root@EXEMPEL.COM` för kerberos5

där principalen beskriver användaren rom:s administrativa användare, i kerberos kallad rootinstans, i realmen EXEMPEL.COM. En realm är den gruppering av system som administrativa skyddas av en kerberosserver.

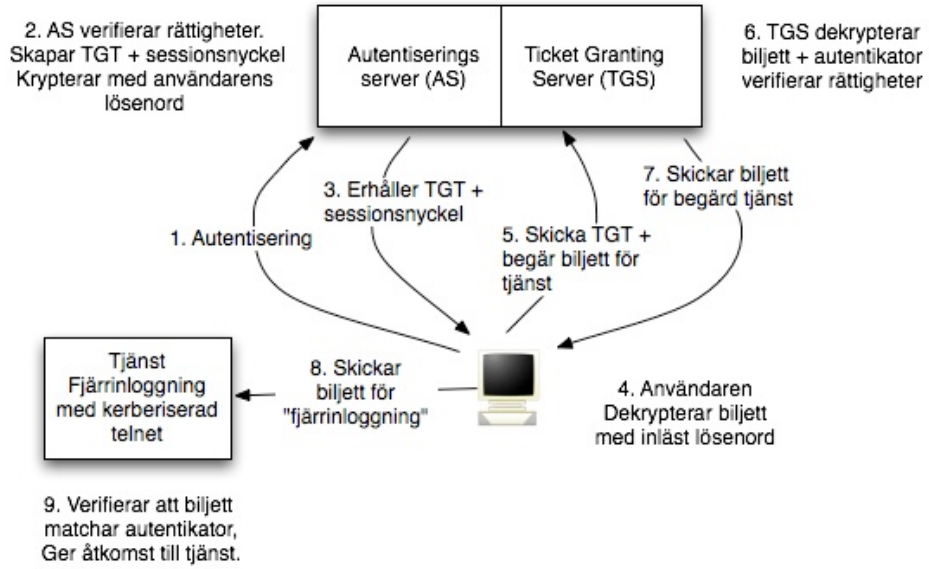
Principalen, tillsammans med användarens rättigheter, en tidsstämpel, livslängd, kontrollsumma paketeras till en *kerberosbiljett*. Tack vare att biljetten är krypterad kan rättigheter distribueras säkert mellan olika nätverksenheter eller tjänster.

Trots att Kerberos uppstod i samma tidsanda då asymmetrisk kryptering såg dagens ljus och var ett hett ämne, är principen bakom Kerberos är baserad på symmetrisk kryptering. Den ursprungliga versionen av Kerberos byggde på DES-algoritmen. För varje enskild användare, tjänst eller resurs skapas och nyttjas separata kryptonycklar.

Den centrala funktionen i Kerberos är Key Distribution Center, eller KDC som det kallas allmänt. En KDC är en databas med kryptonycklar för alla användare, resurser och tjänster. KDC agerar som betrodd tredje part och innehållet i databasen och usträckningen av användandet av en KDC bildar en Kerberosspecifik domän kallad *realm*.

KDC:n består principiellt av de två delarna autentiseringsserver, AS, och biljettserver (eng. *ticket-granting server*, TGS). Vid autentisering så skapar AS en särskild utställarbiljett kallad *ticket-granting ticket*, TGT, samt en sessionsnyckel. Dessa bägge krypteras och nyckeln som används för detta är användaren lösenord, en kopia av lösenordet som finns sparad i KDCn. Därefter avslutas användarens kontakt med AS. När användaren vill nyttja en tjänst sänds TGTn tillsammans med autentikatorn till TGS som ställer ut en ny biljett för den önskade tjänsten.

Eftersom Kerberosservern, KDC:n, säkerhetsmässigt är en vital del av systemet krävs väldiga skyddsåtgärder. KDCn behöver inte bara ett väldigt bra logiskt, IT-tekniskt, skydd utan även bra fysisk säkerhet. Notera att efter det initiella nyckelbytet mellan server och KDC krävs det oftast ingen mer kommunikation, om inte lokala tjänster på servern har behov av biljetter. Detta innebär att DMZ-system kan stå helt segmenterade.



Figur 27 Översiktsbild av Kerberoskonceptet

En större Kerberosinstallation består mängder av datorer, användare och tjänster skyddade av flera kerberosserverar indelade i olika realms. En realm kan vara ett enskilt bolag inom en större koncern eller en inskilld institution inom ett universitet. För inloggning mellan olika realm:ar utförs en korsrealmautentisering. Kontakt tas med TGS-server i den fjärran realmen kallad RTGS, remote TGS.

För att en tjänst skall kunna nyttja Kerberos måste den anpassas, en förändring kallad *kerberisering (eng. kerberization)*. Att kerberisera ett program innebär att man utför förändringar i programmets lågnivådelar som arbetar med nätverkskommunikationen. I ett kerberiserat program måste anrop ske till krypto- och kerberosanpassade kommunikationsprotokollrutiner, antingen direkt till kerberos rutinbibliotek eller via ett standardiserat säkerhetsgränssnitt kallat *GSSAPI*, Generic Security Services Application Programming Interface (RFC 4121). Kerberos i Windowsmiljön följer inte den allmäna GSSAPI-standarden. Microsoft har en egen abstraktionsmodell kallad *security support provider*, SSP, och Kerberos är åtkomligt via gränssnittet Security Service Provider Interface, SSPI, vilket semantiskt är snarlikt GSSAPI.

En fungerande, komplett kerberosmiljö består således av:

- Kerberosserver med KDC med tjänsterna AS och TGS samt databas som innehåller nycklar som delas med servertjänster, server- och klientdatorer
- Servrar med Kerberoskapabla tjänster
- Klientdatorer med kerberiserad mjukvara och klientdatorprincipaler
- Användare vars principaler finns i Kerberosdatabasen.
- En synkroniserad tidsuppfattning mellan de olika datorerna och tjänsterna.
- En dns med srv-records (DNS information om vilka hostnamn kerberosservrarna har)
- Korrekt uppsatt dns med framlänges- och baklänges namnuppslagning.

Kerberos existerar i två egentliga versioner. Kerberos version fyra är den ursprunglig MIT versionen. Version ett till tre av Kerberos var interna utvecklingsversioner vid MIT, därför räknas version fyra som originalversion. Denna finns inte standardiserad eller dokumenterad med mindre än att man nyttjar eller undersöker källkoden. Kerberos version fem (RFC 4120) innehåller utökad funktionalitet samt korrigerar vissa säkerhetsproblem som fanns i version fyra. Skillnaden mellan Kerberos 4 och 5 är:

- version 4 kunde enbart nyttja DES-kryptering.
- Version 5 använda andra symmetriska och asymmetriska kryproalgoritmer.Biljetter kan få längre livslängd, biljetter går att ställa ut i framtiden. Det finns möjlighet att använda pre-autentisering vilket innebär att andra typer av autentiseringssystem går att kombinera med kerberossystemet, exempelvis smarkortsystem eller biometri istället för lösenord.

I dag existerar flera Kerberosimplementationer. Den ursprungliga MIT-kerberos har vidare utvecklats och släppts fri i olika varianter. Microsoft har implementerat Kerberos 5 som en del av säkerhetssystemet i Windows 2000 och senare versioner, vilket gör detta till den mest spridda versionen och implementationen. KTH-KRB är en version4 implementation och Heimdal är en Kerberos5-version utvecklad vid KTH i Stockholm. Kerberos ingår, även fast det inte är aktiverat som standard, i MacOS X samt flera Linux-, BSD- och UNIX-distributioner.

I Windows är KDCn är integrerad med domänkontrollanten, och sedemera Active Directory. KDCn agerar såväl AS och TGS.

I praktiken döljs väldigt mycket av kerberos för användaren i Windows. Användaren behöver aldrig hämta ut biljetter själva utan detta görs när användaren loggar in i domänen. När användaren väljer att logga in skickas en biljett, en TGT, vilket är en sessionsnyckel krypterad med användarens symetriska nyckel (lösenord) till klienten. Om användaren slog in rätt lösenord kan biljetten dekrypteras och användaren får logga in.

När en server ansluts till domänen sker det initiella nyckelbytet mellan KDC:n och servern och DNS-servern (om man har den konfigurerad för dynamiska uppdateringar och ansluten till AD:t) automatiskt uppdaterad. Oftast agerar alla domänkontrollanter i windows KDC.

När klienten väljer att ansluta till en kerberiserad tjänst sker först en baklänges-uppslagning i DNS för att ta reda på vilken tjänstebiljett som ska hämtas. Denna är oftast i formen host/hostname.domain, tex host/dator1.romab.com. Notera att i windows är alla servicetyper i versaler. Om inloggningen inte sker direkt mot en dator (fjärrskrivbord, kommandoskal etc) utan istället går mot en specifik tjänst hämtas andra typer av biljetter, tex `FTP/dator1.romab.com`, `LDAP/dator2.romab.com` eller `CIFS/dator3.romab.com`

När allt fler program blir webbaserade tjänster, används kerberos på ett något annorlunda sätt då klienten (webbläsaren) har en del begränsningar. *SPNEGO* (Simple and Protected GSSAPI Negotiation Mechanism) har blivit standard för detta, och används främst när enklient vill autenticera mot en server men varken server eller klient vet vilka autenticeringsmetoder som stöds.

Den första implementation av SPNEGO kom i webbservern IIS5 och webbklienten internet explorer 5.01, och hade stöd för två autenticeringsmetoder, NTLM och keberos. Oftast använder man "HTTP negotiate", en utökning av http som innebär att webläsaren skickar information om authenticeringsförhandlingen i HTTP-headrar.

För att sätta upp spnego-lösning måste först en HTTP/-serviceprincipal skapas i KDCn och sedan måste webservern konfigureras för detta. I windows finns det inget behov av att göra ytterligare nyckelbyten (tex en separat keytab för webservern) utan det sker genom SSPI. I unix måste webservern konfigureras till att använda tjänstenyckeln. Klienten måste även konfigureras att använda HTTP-negotiate, vilket i internet explorer görs genom att redigera säkerhets inställningarna och i firefox genom att nå urlen about:config.

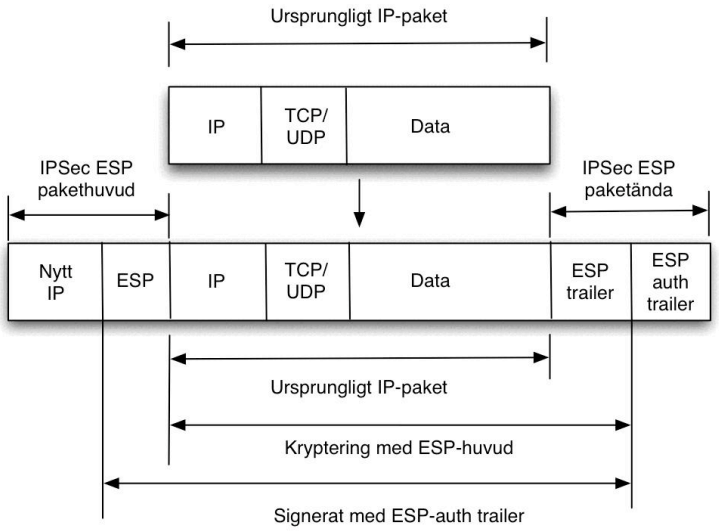
Värt att tänka på att även om det är mycket enkelt att använda spnego för autenticering mot webservrar är det oftast flerlagermodeller där webservern står framför en applikationserver, som i sin tur står framför en databasserver. Då är det oftast önskvärt att flytta autenticeringen helt till applikationslagret, eller databaslagret, men det är inte alltid det är möjligt. I det tidigare fallet p.g.a. tekniska begränsningar i applikationsservern och i det senare fallet för att applikationsservrar idag oftast har en trådpool mot databasen för att få bättre prestanda.

7.2 IPSec

IPSec är ett protokoll utvecklat av IETF för att säkra upp alla olika typer av IP-baserad datorkommunikation. IP Security, IPSec, är en standardsvit som ofta används för att bygga virtuella privata nät, VPN, över Internet.På så sätt kan man ersätta fasta förbindelser med krypterade tunnlar. VPN-tekniken har bland annat vunnit stor framgång i olika lösningar för distansarbete.

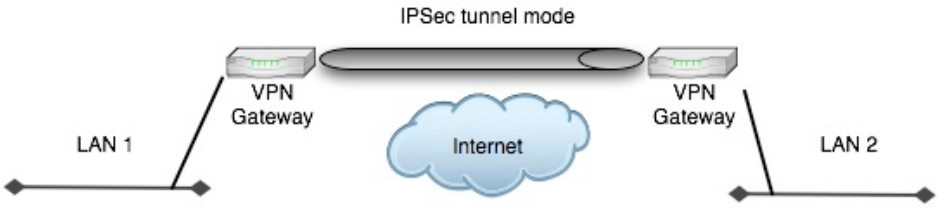
Eftersom det innebär en kostnadseffektivisering att använda Internet som bärartjänst istället för fasta kommunikationslänkar eller dyra uppringda lösningar, så är IPsec en teknik som vinner allt starkare fotfäste. IPSec togs ursprungligen fram för IP version 6, IPv6, men har även anpassats för IP version 4.

IPSec är ett skydd på nätvnivå och erbjuder därmed möjligt skydd till all överliggande kommunikation och alla typer av överliggande protokoll. Två viktiga beståndsdelar i IPSec är *Authentication Header*, AH och *Encapsulating Security Payload*, ESP. AH används för att autentisera avsändande system och för att se till att meddelandet inte kan förvanskas. I AH används ofta HMAC-MD5, HMAC-SHA1 och DES-MAC. Vanliga krypteringsfunktioner för användning av ESP-metoden i IPSec är DES, 3DES och AES. En speciell typ av krypteringsmetod kallad för ”null cipher”, som är ett krypto utan verkan, ingår som ett av standardvalen i IPSec. Det är viktigt att kontrollera inställningar och den verkliga kommunikationen så att man inte omedvetet, eller av en angripare bli nedförhandlad, att använda denna krypteringsvariant.



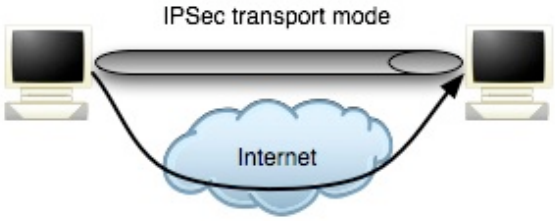
Figur 28 IPSec när ESP används

Man kan använda IPSec i två lägen som kallas *transport mode* respektive *tunnel mode.* I ”tunnel mode” binder man ihop två stycken nätverk, tex LAN.



Figur 29 IPSec uppsatt i tunnelmod mellan två nätverk

I ”transport mode” skapar en tunnel mellan två kommunicerande ändsystem (s.k. änd-till-ändsäkerhet) – oftast vanliga datorer.



Figur 30 IPSec uppsatt i trasportmod mellan två datorer

Den stora skillnaden mellan transport och tunnelmod är huruvida man lägger hela ursprungspaketet i ett tunnlat paket med ett nytt pakethuvud (IP-header) eller om man återanvänder det befintliga pakethuvudet. Änd-till-änd-säkerhet är en viktig princip då den skyddar mot olika typer av attacker som kan uppstå på vägen, tex angrepp mot mellanliggande kommunikationskomponenter som håller kryptonycklar om länkkryptering används istället.

För att etablera en IPSec-koppling så måste de kommunicerande enheterna först skapa en *Security Association*, SA, som innehåller viktig information om kopplingen såsom valda algoritmer, nycklar och giltighetstid. En SA beskriver hur en IPSec-koppling skall hanteras från fall till fall och därmed kan man skapa olika kopplingar mellan olika tjänster eller mellan olika kommunicerande utrustningar. En viktig egenskap hos en SA är att den enbart specificerar kommunikationen i ena riktningen, så för en komplett kommunikation så behövs det två stycken SA som definierar båda kommunikationsriktningarna. Tack vare att SA:n är enkelriktad så talar man oftast om SA-par för att tala om en komplett dubbelriktad session. Varje enskild SA går att bestäms utifrån typ av protokoll, AH eller ESP, IP-adressen för mottagarnoden ett unikt värde kallat SPI, Security Parameters Index, samt ett sekvensnummer för att skapa skydd mot återuppspelningsattacker. SPI-värdet är ett 32-bitars tal som även ingår i det översända pakethuvudet på alla IPSecpaket vilket gör att den mottagande enheten lätt kan hitta vilken SA som skall användas för att avtolka paketinnehållet.

Alla aktiva IPSec-kopplingar hanteras av en SAD, *Security Association Database*, vilket ofta är en lista i operativsystemets nätverkskod. En annan mer omfattande databas som används av IPSec är *Security Policy Database*, SPD. Denna databas används för all IP-trafik i alla riktningar och kan därför användas för att begränsa trafik på ett sätt som är likvärdigt med en traditionell paketfiltrerande brandvägg. SPDn används huvudsakligen för att bestämma vad som skall sändas med vanlig IP, om ens vanlig IP skall tillåtas, och vilken trafik som skall sändas med IPSec. I vissa IPSec-produkter är denna effekt inte uppenbar eller ens möjligt att nyttja för paketfiltrering, men i andra IPSec-implementationer är det lätt att nyttja. Exempel på detta är att man i Windows 2000 kan nyttja IPSec-modulen för att få bättre paketfiltrering än den ordinarie filtreringsfunktioner. Genom att nyttja funktionen så kan man därmed få ett ekonomiskt sätt att skapa brandväggsfunktioner och på så sätt bygga skydd i djupled och ha flera försvarslinjer.

En viktig aspekt av VPN-användningen är huruvida man använder statisk eller dynamisk nyckelhantering. Med statisk nyckelhantering, ofta kallad *pre-shared key*, PSK, så använder man en delad på förhand bestämd nyckel. Med dynamisk nyckelhantering så skapas unika sessionsnycklar. Dessa kan vara certifikat eller bara asymmetriska kryptonycklar. Man behöver ett till IPSec kompletterande nyckelhanteringsprotokoll för att skapa dynamisk nyckelhanering. Det vanligaste nyckeldistributionsprotokollet heter IKE, Internet Key Exchange. Det förekommer viss namnförbistring på detta område då det finns och har funnits ett antal olika protokoll, såsom Photuris, SKIP, SKEME, Oakley och ISAKMP för att utföra nyckelhantering i IPSec. Vissa av dessa, SKIP, Oakley och ISAKMP har bildat hybriden IKE och därför refereras det till IKE ibland och till komponenterna ibland beroende på vad man talar om. ISAKMP är ett ramverk för autentisering och nyckelutbyte medans Oakley är ett protokoll. Med hjälp av IKE så bestämmer de kommunicerande parterna inledningsvis vilka nycklar, algoritmer och protokoll som skall användas. IKE utför också kontroll så att man pratar med rätt motpart samt ser till att utbyta och hantera överförda kryptonycklar. Nycklar kan bytas efter viss tid eller efter en viss mängd överförd data.

ISAKMP arbetar i en av två faser. I fas ett så etableras en säker kanal mellan de två kommunicerande parterna för att sätta upp en ISAKMP SA. I fas två så förhandlar parterna om generella SA. I fas ett så finns det två arbetsmoder, *main mode* och *agressive mode*. Main mode används för att etablera en ISAKMP SA. Agressive mode kan i stort utföra samma som main mode men med färre paket tack vare att den uppkopplande sidan utför mer arbete på förhand. En nackdel med agressive mode är att den initial trafiken går i klartext vilket möjliggör för en utomstående att observera identitetera på de som förhandlar. Fördelen med

agressive mode är hastighet och det har visat sig att vissa VPN-produkter i kombination med till exempel Windows har krävt agressive mode för att inte få en time-out i uppkopplingsfasen. Quick mode är en arbetsmod som används i fas två. Den används för att förhandla nya nycklar och hantera IPSec.

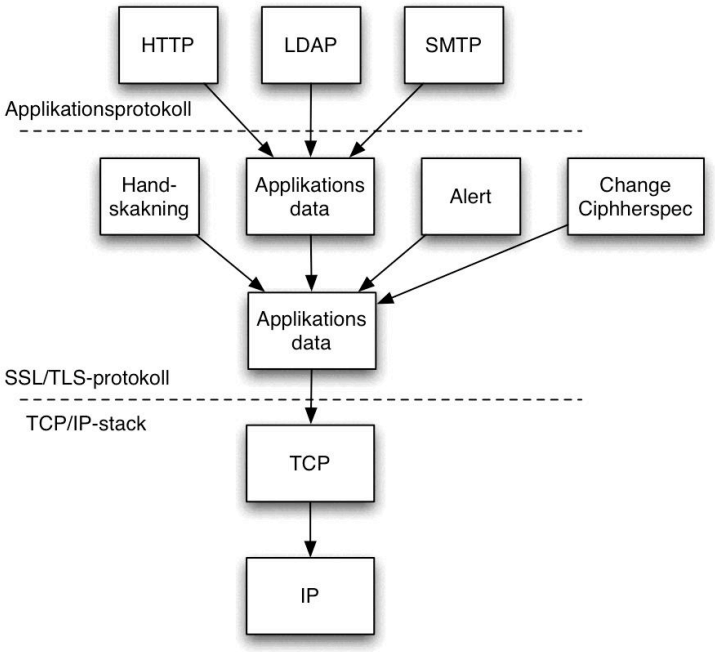
Ett problem med den nuvarande IKE versionen är att den fungerar dåligt med adress-översättning, NAT, därför har flera tillverkare egna utökningar för att få nyckelutbyte att fungera även i dessa sammahang. IETF arbetar med att ta fram en ny variant som skall klara även NAT. Med statisk nyckelhantering konfigurerar man utrustningen, till exempel två VPN-gatewaysystem, med en överrenskommen gemensam hemlighet. När man gör tekniska säkerhetskontroller av VPN-utrustning upptäcker man en av nackdelarna med med statiska lösenord – det är att administratörer tenderar att återanvända samma lösenord i flera kommunikationsutrustningar och att lösenordet sällan eller aldrig ändras - vilket leder till stora trafikvolymer som kan kryptoanalyseras, att gamla anställda fortfarande kan komma in, att servicepersonal och andra kan känna till lösenorden. ISAKMP beskrivs i RFC 2408.

7.3 Transport Layer Security – TLS

TLS, Transport Layer Security, är ett krypteringsbaserat säkerhetsprotokoll som läggs till som ett extra lager till ett applikationsprotokoll för att erbjuda en- eller tvåvägsautentisering, skydd mot avlyssning, förfalskade meddelanden samt datamanpulation. För att fungera kräver TLS ett leveranssäkert transportprotokoll, vanligtvis TCP.

SSL erbjuder ensidig eller ömsesidig autentisering av de kommunicerande parterna, till exempel användaren av klientprogrammet och en riktighetskontroll av serversystemet. Kommunikation via SSL och TLS består av ett antal delar. Den första delen är en uppkopplings- och förhandlingsfas. Denna del, SSL hanskakningsprotokollet, består i sin tur av två faser – en serverautentiseringsfas och en (valfri) klientautentisering. I samband med kommunikationsstart baserad på TLS och handskakning mellan server- och klientdel förhandlas om ett antal parametrar: Version av SSL/TLS att använda, val av kryptoalgoritmer,

TLS används som säkerhetsmekanism för att skydda ett stort antal protokoll inklusive HTTP, SMTP, IMAP, POP, FTP, XMPP, NNTP med flera.



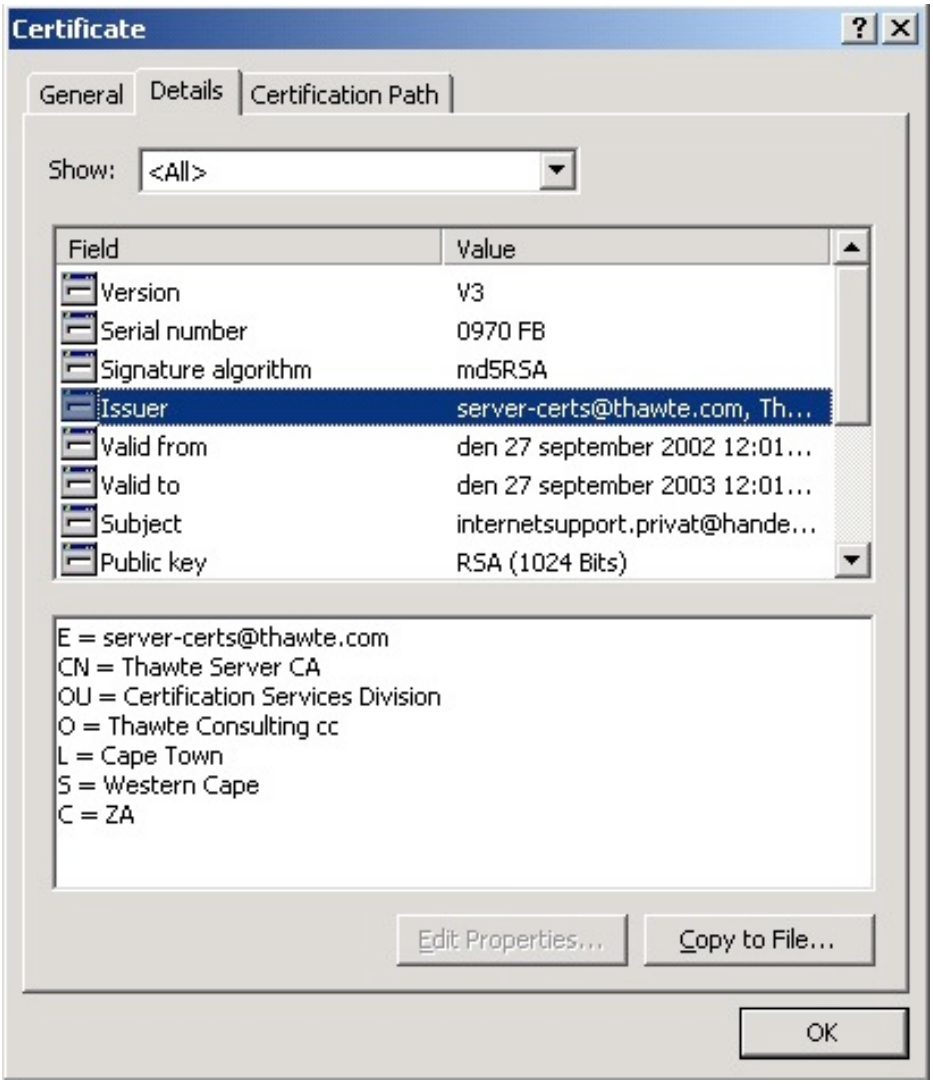
Figur 31 Översiktsbild för SSL/TLS plats i protokollstacken

En grundkomponent i SSL/TLS är ett fält (eng. *record*). I SSL/TLS använder man ett grundläggande protokoll, *record layer*, som kapslar andra protokoll och meddelanden i SSL/TLS

Grunden till TLS är SSL, Socket Layer Security, som utvecklades av företaget Netscape, ett av de första företagen som utvecklade webbserver- och webbläsarprogramvaror. Ingen version 1 släpptes publikt, utan den första allmänt tillgängliga SSL-varianten var version 2.0 som kom 1994. Då SSL utvecklats av netscapes programmerare, snarare än av kryptologer, så fann säkerhetsexperter snart ett antal säkerhetsproblem med protokollet, såväl i själva implementationen som i protokolldesignen. Trots dessa problem så anammades SSL i allmänhet brett av i de flesta webblösningar, inkl e-bankslösningar. Microsoft var en av Netscapes konkurrenter och utvecklade en alternativ säkerhetslösning kallad PCT – Private Communication Technology, som inte innehöll SSL v.2-problemen. På grund av säkerhetsbristerna och Microsofts teknikval så utvecklade Netscape en ny version, 3.0 som släpptes 1996, för vars utveckling de hyrde in säkerhetsexperter och kryptologer. Det nya protokollet är mycket förändrat i jämförelse med sin föregångare, så att i princip bara namnet är det samma.

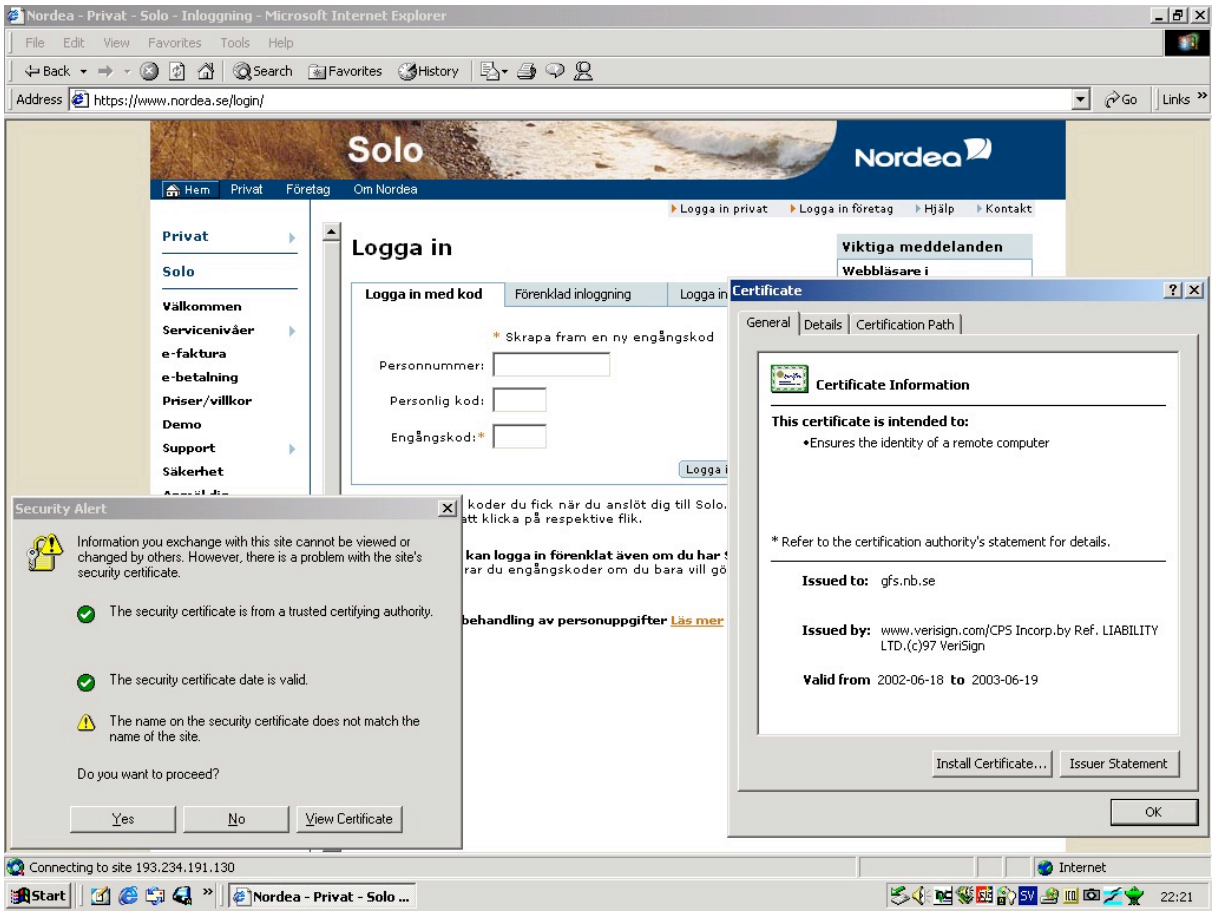
SSL version 3 kom att bilda grundstenen till TLS version 1, vilket är det standardprotokoll som IETF 1998 definierade i RFC 2246. Den nuvarande versionen, 1.1, beskrivs i RFC 4346.

En av grundstenarna i SSL/TLS är publik nyckelkryptering, med certifikat enligt x.509-standarden.För exempelvis en webbtjänst som skall skyddas av SSL/TLS krävs det att webbservern har ett servercertifikat. SSL och TLS använder en kombination av asymetriska och symetriska kryptoalgoritmer samt kryptografiskt starka checksummor. De asymetriska kryptoalgoritmerna, såsom RSA, används primärt under etableringsfasen under SSL-handskakningen. Eftersom asymetriska algoritmer är prestandakrävande så är det en anledning till att nättjänster som nyttjar SSL/TLS under hög belastning få prestandaproblem. Ett sätt att lösa det är att installera olika typer av kryptoacceleratorlösninngar.



Figur 32 Detaljer från ett X.509-baserat SSL/TLS-certifikat

Implementationer av SSL/TLS använder information från certifikaten för att kunna utföra ett antal säkerhetskontroller, exempelvis för att se ifall certifikatet är utgånget, om det finns problem med förtroendekedjan, om dator- och domännamn i certifikatet överrensstämmer med det namn som finns på datorn som besöks, etc.



Figur 33 Exempel när namn i certifikatet inte överrensstämmer med datornamn/domän

En kusin till SSL och TLS är Wireless TLS, WTLS. Det har skapats för att erbjuda säkerhet vid trådlös kommunikation. Det finns flera anledningar till att man tog fram WTLS. Dels är standarden optimerad för låg bandbredd, dels kräver SSL att man använder TCP som transporttjänst. SSL fungerar inte över UDP, något som utvecklare av trådlös kommunikation har önskat.

En vanligt förekommande implementation av SSL och TLS är OpenSSL, vilken finns tillgänglig i källkodsform ifrån www.openssl.org och som ingår i flera vanligt förekommande produkter, exempelvis webbservern Apache.

7.4 SSH

Kommunikationsprotokollet Secure Shell, SSH, skapades av Tatu Ylönen i mitten av 1990-talet och är idag ett vanligt förekommande protokoll som används för att skapa en krypteringsskyddad förbindelse mellan två system. SSH är en ersättare för de gamla, osäkra r-kommandona rlogin, rsh och rcp, vilket innebär att det är såväl en terminalemulator som program för fjärrexekvering och filkopering.. SSH är populärt bland system- och nätverksadministratörer samt avancerade användare.

SSH erbjuder bland annat säkrad autentisering via publik kryptering, säkrade sessioner med symmetriska kryptoalgoritmer samt porteftersändning (eng. port forwarding) för att krypterat tunnla valfria portar över SSH- kanalen. Porteftersändning är både en styrka och ett säkerhetsproblem hos SSH. Från en säkerhetschefs eller brandväggsadministratörs perspektiv kan denna porteftersändning vara problematisk eftersom man via SSH kan skapa tunnlar för

valfri trafik och därmed åsidosätta brandväggar och policymässiga inskränkningar. En styrka är att man lätt kan skapa tillämpningstunnlar transparent för standardtillämpningar och protokoll. Man kan till exempel lura ett e-postprogram att öppna en POP- eller IMAP-förbindelser över SSH.

Protokollet som SSH använder finns i flera versioner, version 1 och 2. Man bör undvika att använda den gamla versionen och istället konfigurera sina SSH- klienter och servrar att enbart tillåta version 2. En annan inställning man bör använda för att praktiskt säkra upp sin SSH-användning är att enbart tillåta SSH-tjänsten i kombination med nyckelstyrd inloggning istället för lösenordsbaserad inloggning.

En populär kostnadsfri variant av SSH är OpenSSH. OpenSSH har bakåtkompabilitet med andra SSH-produkter men innehåller även en mängd utökningar och förbättringar. För mer information om OpenSSH, se www.openssh.org. För windows används ofta programmen putty och winscp.

7.5 SASL

IETF har tagit fram ramverket SASL, Simple Authentication and Security Layer om erbjuder flexibela säkerhetsmekanismer såsom autentiseringsstöd och behörighetskontroll för andra internetprotokoll. Protokollet innehåller delar för att identifiera och autentisera en användare mot en server.

Flexibiliteten i SASL består i att flera olika autentiserings metoder går att nyttja. SASL har idag standardiserat stöd för fråge-svars-protokollet CRAM- MD5, Kerberos version 5, SecurID engångslösenord, Windows autentiseringsmetod NTLM, med mera. För bakåtkompabilitet erbjuds även klartextlösenord, autentiseringsvarianten PLAIN, men detta skall enbart användas när kommunikationen skyddas av andra säkerhetsmekanismer, exempelvis TLS.

Bland de protokoll som har SASL-stöd finns BEEP, LDAP, IMAP, POP och SMTP.

SASL-protokollet i sig specificeras i RFC 4422.

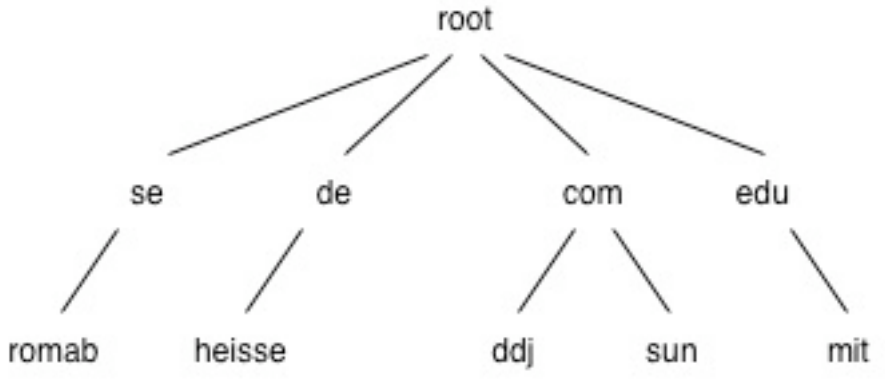
Flera implementationer av SASL existerar. Det finns ett SASL-API för både klient- och serversideapplikationer som standard i Sun:s Javaimplementation. Den mest kända fria implementationen är GNU SASL, utvecklad av svensken Simon Josefsson.

7.6 DNS

Domännamnssystemet, DNS (eng. *Domain name system*), är en grundkomponent i alla moderna TCP/IP-baserade nätverk. DNS kan ses om Internets motsvarighet till en stor telefonkataloge och har till uppgift att översätta mellan numeriska IP-aresser och teckensträngar som utgör ett för människor förståeligt representation. I Internets barndom användes en enklare metod, filen HOSTS.TXT med översättningstabeller mellan IP-adress och datornamn gick med filöverföring att hämta från en central distributionspunkt. Denna metod blev snart ohanterlig på grund av Internets tillväxt. Problemet var inte bara den ständigt växande volymen på informationen som behövde hämtas, utan också det faktum att informationen snabbt blev inaktuell. I mitten av 1980-talet utvecklades DNS, ett konceptuellt enkelt system för fråge- och svarshantering av information. Som med de flesta teknologier

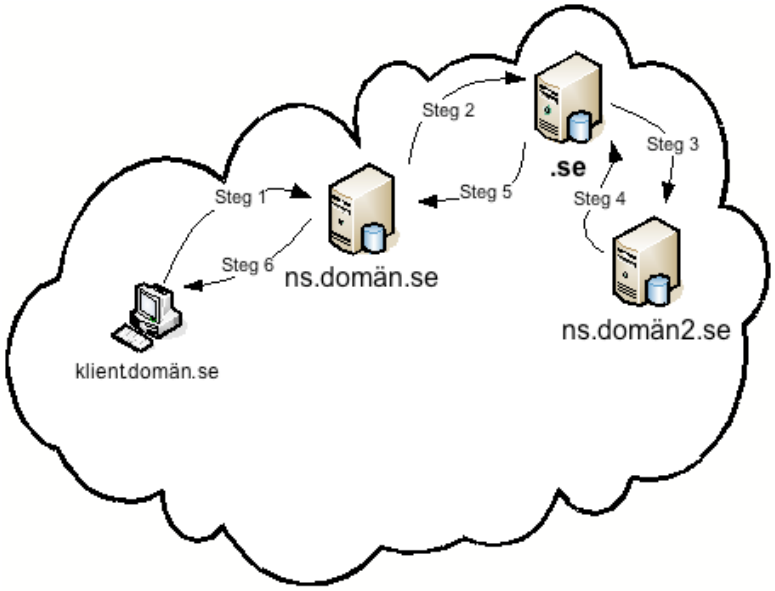
och protokoll som kom fram under den tidiga Internetutvecklingen, var fokus påfunktion och stabilitet, inte på informationssäkerhet.

På grund av Internets natur, med flera autonoma organisationer som tillsammans är sammanslutna i ett stort nät, måste DNS vara uppbyggd som en distribuerad databas där de olika organisationerna ansvarar för sitt innehåll i databasen. De olika organisationernas del av databasen hanteras av olika DNS-servrar, så kallade namnservrar. Den distribuerad lösningen har många fördelar, men också vissa nackdelar. Distributionen innebär att det måste finnas en mängd fungerande och rätt uppsatta namnservrar. Det är de olika organisationernas ansvar att ha en korrekt installation samt ha uppdaterad programvara.



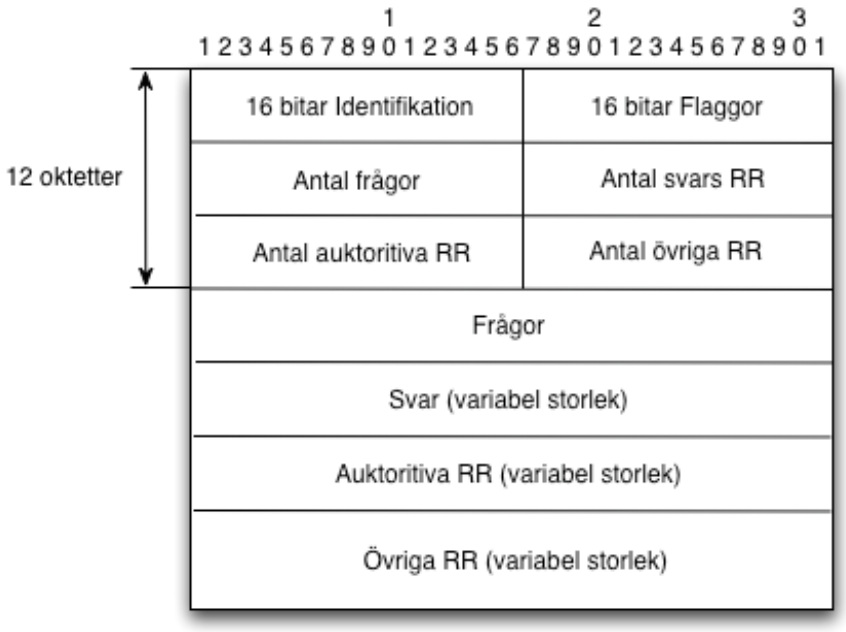
Figur 34 Den hierarkiska uppbyggnaden av DNS-systemet

DNS-protokollet är relativt simpelt, vilket är något som också har lett till dess framgång på Internet. DNS är ett fråge- och svarsprotokoll där kontrollen är distribuerad enligt en hierarkisk uppbyggnad. Varje enskild organisation eller individ som har en DNS-namnserver kontrollerar sin del av namnrymden, och kan lägga upp underdomäner eller datornamn i sin DNS. En DNS-fråga, exempelvis för att kunna hitta en IP-adress till ett domännamn, skickas till en DNS-server som antingen kan svara på frågan eller vidarebefordra den till andra DNS-servers i hierarkin.



Figur 35 DNS-fråga från klient ang domän2.se

DNS-protokollets uppbyggnad är mycket enkelt med ett antal förbestämda fält och fälttyper. På klientsidan är det ett så kallat resolverbibliotek som sköter frågeställningen till en DNS-server på mottagarsidan.



Figur 36 Generell bild av DNS-protokollets uppbyggnad

Hoten mot domännamnssystemet är av en mängd olika karaktärer. Olika tekniska hot såsom förfalskning av DNS-svar eller obehörigt övertagande av en DNS-server är några kända exempel. Administrativa, eller juridiska hot, är ofta juridiska problem i form av intrång i varumärkes- och namnrättigheter som förknippats med DNS.

7.6.1 Informationsinsamling via DNS

Informationsinsamlingsattacker är ofta inte en attack mot DNS-systemet i sig, utan snarare en metod i vilken DNS-information hämtas för nyttjande av en attack utförd i ett senare skede. Olika typer av spaning och informationsinsamling som sker via DNS, och dess tillhörande hjälpsystem, är:

- Versionskontroll av DNS-servern
- Missbruk av whois-information
- Zonöverföring
- Trafikanalys
- Informationsläckage om organisationsinterna förhållanden

Vissa mindre kända implementationsdetaljer i vanligt förekommande DNS-serverprogramvaror har gått att bruka för att via nätet hämta en DNS-servers versionsnummer och konfigurationsparametrar.

```
Rrom@www rom]$ dig -t txt -c CHAOS version.bind @ns.kth.se
; <<>> DiG 9.2.1 <<>> -t txt -c CHAOS version.bind @ns.kth.se
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 43722
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; QUESTION SECTION:
;version.bind.                  CH      TXT
;; ANSWER SECTION:
version.bind.           0       CH      TXT     "9.2.2-P1"
```



```
;; Query time: 6 msec
;; SERVER: 130.237.72.250#53(ns.kth.se)
;; WHEN: Sun Oct 30 14:38:47 2005
;; MSG SIZE  rcvd: 51
```

I exemplet ovan ser vi att DNS-servern ns.kth.se kör version 9.2.2 patchnivå 1 av DNS-programvaran bind. Modernare metoder att kontrollera programtyp och programversion inkluderar att använda program som fpdns[82], vilket utifrån fråge-svarsdialogen via fingeravtrycksmetodik kan karaktärisera vilken DNS-serverprogramvara som svarar på frågan. Genom att hämta ut information ur whois-databasen kan det i vissa fall gå att stjäla personinformation, e-postadresser, med mera. Historiskt har det gått att hämta mycket information via olika domänadministratörers whois-databaser. Allt eftersom missbruk av informationen ökat samt lagstiftning om personinformation lagt begränsningar har den publikt tillgängliga informationen minskat.

Zonöverföringar (eng. *zone transfer*), när någon hämtar ut all DNS-information om en viss domän, kan vara ett steg i informationsinsamlingen. Zonöverföring sker normalt från en primär DNS-server och de sekundära servrar som genom att spegla zonen därefter kan svara på domänfrågor. Den insamlade informationen kan användas i senare attacker.

Ett annat sätt är att nyttja specialprogramvara som automatiskt traverserar en zon, exempelvis walker[83]. Huruvida hela zoninformationen är tillgänglig eller ej beror på policymässiga frågor - anser domäninnehavaren att informationen har kommersiellt intresse eller skall vara fritt tillgänglig, likväl som tekniska frågor - hur har domänad-ministratören konfigurerat (eller inte vetat om konfigurationenför) zonen. Statistik från en större kontroll 2005 av mer än 1000,000 zoner i .com-domänen visar att mer än 40% av alla zoner tillät zonetransfer[84].

Att spärra den DNS-zonfilen som beskriver externt åtkomliga system från åtkomst för utomstående har som regel liten säkerhetshöjande effekt. Det gäller dock att tänka på vilken information som finns lagrad i DNS-datat, tex om DNS-administratören varit alltför nitisk och namngivit alla brandväggens interface för explicit och ingående.

7.6.2 Tillgänglighetsförlustattack mot DNS

En typ av attack som skulle kunna ställa till mycket problem attacker för att skapa tillgänglighetsförlust (eng. *DoS attack*). Genom att slå ut en eller fler servers eller genom att överlasta tjänsten kan potentiellt DNS-information om en viss domän störas ut. Några olika typer av tillgänglighetsförlustattacker är:

- överlasta en specifik DNS-tjänsten med alltför många frågor.
- överlasta kommunikationslänkarna som ansluter en specifik DNS-serverdatorn så att inte DNS-servern kan nås eller skicka tillbaka svar[85]. Överlastningen kan ske via DNS-frågor eller via något annat protokoll, exempelvis upprepade ICMP echo request (dvs ping).

---

[82] Roy Arends & Jakob Schlyter (2003) *fpdns - Fingerprinting DNS servers*. http://www.rfc.se/fpdns/

[83] Josefsson, Simon. *DNSSEC Walker*. http://www.josefsson.org/walker/

[84] The Measurement Factory (2005) *DNS SURVEY: JUNE 2005* http://dns.measurement-factory.com/surveys/200506.html

[85] CERT-CC (2000). *Denial of Service Attacks using Nameservers*. http://www.cert.org/incident_notes/IN-2000-04.html

- Ställa namnfrågor till flera DNS-servers men använda en falsk avsändaradress som får mottaga svar från alla servrarna[9]. DNS-svar kan vara mycket större än själva frå-gan, vilket gör att DNS-servrarna används som en tredje part som ger utväxling på attacken.
- en DNS-server kan luras att skicka frågor och svar till sig själv i en oändlig loop[86]
- en DNS-server kan fås att krasha på grund av dålig indata[87]
- en DNS-server kan fås att krasha på grund av för mycket indata[88]

Ett sätt att förstärka effekten av en tillgänglighetsförlustattack är att skala upp attacken till en massiva distribuerade attacker (eng. *DDoS attack*). DDoS-attacker kan ske mot en eller flera DNS-servers hos den som skall attackeras. Konsekvenser att DNS-tjänsten slås ut eller överlastas blir att andra tjänster som e-post eller webb slutar att fungera. Att slå ut ett e-handelsbolag eller en internetbanks DNS-tjänst är ett effektivt sätt att indirekt stoppa åtkomst till en webbtjänst.

Ett exempel på en storskalig DDoS-attack mot internetinfrastrukturen är attacken den 21:a oktober 2002 mot rootnamnservrarna[89]. Någon eller några koordinerade attacker alla tretton rootnamnservrar med olika typer av attackmetoder i ett försök att slå till mot hjärtat av Internet. Attackerna lyckades delvis i att överlasta länkar och servrar men var i realitetet inte något större hot.

7.6.3 Förfalskning av DNS-svar

Förfalskade DNS-svar (eng. *DNS-spoofing*) kan medföra att en namnfråga resulterar i ett svar som leder ett program eller en person till ett system som angriparen förfogar över. Falska DNS-svar kan nyttjas i scanarios för att förfalska adressen till en källa likväl som förfalskning av adressen till en viss destination.

Ett exempel på förfalskning av adressen till en avsändare är vid autentisering av avsändaren. Om DNS-frågan nyttjas av en nätverkstjänst som nyttjar domännamnet för autentisering, exempelvis rlogin-tjänsten, så går det att lura autentiseringsfunktionen. Destinationsförfalskning av adressen kan vara att DNS-frågan ställs av en webbläsare till en viss adress exempelvis http://www.storabanken.se. I fallet med webbåtlkomst går det att via förfalskade DNS-svar omdirigera trafiken till en falsk webbserver som kan erbjuda förvanskad information eller nyttjas för exempelvis stöld av användarnamn och lösenord.

Paul Vixie, en av huvudmännen bakom DNS-programvaran bind gjorde i mitten av 1990-talet flera förändringar[90] i bind för att motstå flera av de kända attackerna mot för- falskade DNS-svar, korruption av DNS-cache. Ungefär samtidigt påbörjades en större översyn av vilka säkerhetshöjande funktioner som DNS-systemet kunde behöva, något som så småningom ledde till DNSSec, se kapitel 7.6.9 ”Skydd i DNS” sid 120.

---

86 UNIRAS (2004). *Vulnerability Issues in Implementations of the DNS Protocol*. http://www.niscc.gov.uk/niscc/docs/al-20041130-00862.html?lang=en

87 US CERT (2005). *BIND 9.3.0 vulnerable to denial of service in validator code*. http://www.kb.cert.org/vuls/id/938617

88 US CERT (2005). *BIND 8.4.4 and 8.4.5 vulnerable to buffer overflow in q_usedns* http://www.kb.cert.org/vuls/id/327633

89 Vixie, Paul mfl (2002). *Events of 21-Oct-2002*. http://d.root-servers.org/october21.txt

90 Vixie, Paul (1995). *DNS and BIND Security Issues.* http://www.usenix.org/publications/library/proceedings/security95/full_papers/vixie.txt

7.6.4 Förvanskning av DNS-cache

Att attackera en DNS-resolver och förvanska DNS-cacheinformation (eng. *DNS cache poisioning*) tillhör en av de mer allvarligare attackerna mot DNS-systemet. Attacker mot DNS-cache har varit kända sedan länge. De forskningsresultat som Bellovin fick undanhöll han från att publicera 1990. Först flera år senare, år 1995, valde han att publicera uppgifterna[91]. Problemställningen som sådan visar ju hur svårt det ibland kan vara att hantera kunskapen om säkerhetsproblem i stora infrastrukturkomponenter – skall vi gå ut med det nu och exponera infrastrukturen eller försöka arbeta i tysthet i bakgrunden och rätta till problemet.

Ett känt exempel på storskalig förvanskning av DNS-cache utfördes 1997 av Eugene Kashpureff[92], ägaren till den alternativa registrerings AlterNIC.

7.6.5 Otillåtet övertagande av domännamn

En metod att ta över den tekniska kontrollen av ett domännamn är att påverka den *administrativa hanteringen* av en domänen. Genom en obehörig s.k. *ompekning* kan kontrollen överföras från en legitim användare till en icke-legitim användare. När väl kontrollen över domänen är överförd kan förrövaren helt kontrollera vart domännamnet skall peka, tex till en webb som används för bedrägerier eller för att sprida desinformation som tredje part uppfattar som om den kommer från den ursprunglige innehavaren.

Ett känt exempel på ett domännamnsövertagande är övertagandet av www.takedown.com[93], en webbplats för boken takedown vilken handlar om hackern Kevin Mitnick. Någon lyckades via övertalning få domännamnsoperatören Network Solutions att ändra domännamnet från takedown.com till take*n*down.com

7.6.6 Övertagande av begagnat domännamn

Ett domännamn är en handelsvara som kan vara såväl ny som begagnad. Vissa typer av problem kan uppstå då ett införskaffat domännamn tidigare har nyttjats av någon annan person eller annan organisation. Några vanliga problem med begagnade domännamn inkluderar:

- Domännamnet kan existera i loggar som sent kommer att granskas för vissa säkerhetsproblem,
- Domännamnet kan finnas upptaget på spärrlistor i brandväggar, e-posthantering (spamskydd), med mera
- E-postadresser kan finnas registrerade i gamla kundregister, etc

7.6.7 Namnnappning

Domännamnsstöld, oftast kallat namnnappning, är ett administrativt och juridiskt problem som uppstått när domännamnssystemet blivit spritt och namnrättigheter blivit en allt

[91] Bellovin, Steven M (1995). *Using the Domain Name System for System Break-in*. http://www.cs.columbia.edu/~smb/papers/dnshack.ps

[92] Department of Justice (1998) Pressmeddelande. *Eugene E. Kashpureff Pleaded Guilty to Unleashing Software on the Internet That Interrupted Service for Tens of Thousands of Internet Users Worldwide.* http://www.usdoj.gov/criminal/cybercrime/kashpurepr.htm

[93] 17 *2600 Magazine Archive*. http://www.2600.com/offthehook/1996/0296.html

intressantare fråga. Registreringar av namn som kan misstänkas tillhöra andra rättighetshavare är något som historiskt både utförts som affärsidé, som del i aktivism eller av andra anledningar.

Några historiskt kända fall av namnnappning är:

- En anställd vid teleoperatören Sprint registrerar 1994 ett domännamn i konkurrenten MCI:s namn, MCI.net[94].
- Adam Curry, en anställd videojockey på musikkanalen MTV registrerade på privat initiativ domänen mtv.com. Senare domstolsöverläggningar[95] mellan MTV och Curry avslutades med förhandlingar utanför rätten.

I Sverige registrerade företaget Control Alt Delete under mitten av 1990-talet en mängd domännamn som överrensstämde med kända personnamn, företagsnamn eller varumärken[96]. Enligt uppgift så omfattades namn som exempelvis Swedish Match, Bonniers, Folksam, Wallenberg med flera registrerades i .com-domänen av Control Alt Delete. Vissa företag som Folksam och Swedish Match valde en juridisk process om namnrättigheterna, andra företag har valt att köpa relevanta domännamn av Control Alt Delete.

För de flesta domäner gäller fri registrering av domännamn. Dock påverkar namnrättigheter vem som i slutändan har rätt till ett visst domännamn enligt principen att varumärkesinnehavare har företräde. För .se-domänen tillämpas i de flesta domännamnstvister en förenklat tvistlösning, såkallade ATF-förhandlingar[97] (Alternativt tvistlösningsförfarande), i stället för allmän domstol.

7.6.8 Fulregistreringar

Vissa organisationer registrerar medvetet domännamn som har stora likheter med andra populära eller viktiga domännamn. Likheterna brukar vanligen bestå i olika typer av felstavningar, sannolika varianter på namn eller att namnet registreras i en annan domän.

Fulregistreringar (eng. *typo squatting*) kan i ett senare skede, det egentliga attackskedet, användas för en mängd olika typer av attacker, exempelvis för att sprida skadlig kod mot de användare som kopplar upp sig mot webbservern. En annan anledning till fulregistreringar. är s.k mannen-i-mitten-attacker där det fulregistrerade domännamnet nyttjas på en webbserver. Webbtjänsten som utges vara en viss tjänst, men inloggningsuppgifter stjäls eller transaktioner förvanskas innan de når den egentligt avsedda webbtjänsten. Fulregistreringar kan ske av enstaka personer eller brottsliga organisationer som utför utför massiva mängder av fulregistreringar.

---

94 Donelan, Sean (2002) *Timeline of events with the Domain Name System*. http://www.donelan.com/dnstimeline.html

95 United States District Court, S.D. New York (1994) *MTV NETWORKS, A DIVISION OF VIACOM INTERNATIONAL, INC., Plaintiff, v. Adam CURRY, Defendant.* http://www.loundy.com/CASES/MTV_v_Curry.html

96 Nugin, Isabelle (2000). *WIPO:s tvistlösningssystem för tvister gällande domännamnsstölder*. http://www.handels.gu.se/epc/archive/00003141/01/200050.pdf

97 *Tvistlösning*. http://www.iis.se/domannamn/tvistlosning.shtml

7.6.9 Skydd i DNS

För att åtgärda de säkerhetsproblem som finns med DNS, såsom dns-spoofing, så har man inom standardiseringsorganet IETF länge arbetat med en vidareutveckling kallad DNSSec. Det erbjuder på kryptografisk väg (med hjälp av signaturer) en starkare bindning mellan IP-adresser och symboliska DNS-namn.

Modernare versioner av referensimplementationen för DNS, bind, har stöd för DNSsec, en förkortning för DNS Security Extensions. DNSSec är en variant av DNS där signerad zoninformation kan garantera att datat inte är manipulerat, IP-adressen verkligen hör ihop med domännamnet samt att servern är behörig att svara på DNS-information.

Flera nya Resource Records (RR) att använda i sina DNS-zonsdefinitioner har skapats för DNSSec såsom KEY och SIG. Varje DNS-zon har ett asymmetriskt nyckelpar med vilket man signerar zoninformationen och andra sedan kan verifiera att de är korrekt utställda.

Under 2007 blev resultatet av DNSSec-projektet hos stiftelsen för Internetinfrastruktur, den organisation som driver .SE-domänen, att .SE var den första domän på Internet där det kommersiellt var möjligt att registrera domännamn i DNSSec.

För mer information om DNSSec, se http://www.dnssec.net

För skydd mot överlastningsattacker mot DNS-servers, se kapitel 10.3 ”Redundans och nätmässiga reservvägar” sid 127.

8 Trådlösa nätverk

De vanligast förekommande trådlösa datornätverket idag är trådlös Ethernet enligt standarderna IEEE 802.11 och de olika varianterna IEEE 802.11a (54Mbit/s på 5GHz-bandet), IEEE 802.11b (11Mbit/s på 2.4GHz-bandet) samt IEEE 802.11g (54Mbit på 2.4GHz.bandet). Nya standardvarianter är under utveckling, en som är värd att omnämnas är 802.11n som skall ha en överföringskapacitet på mer än 100Mbit/s.

Denna teknik för trådlösa lokala nätverk kallas också för WLAN (wireless local area network). För att bygga WLAN-nät använder man sig dels av bärbara datorer med inbyggt stöd för trådlösa nätverk alternativt använder trådlösa nätverkskort, oftast i formatet PC-CARD (PCMCIA-kort) med inbyggd antenn, dels accesspunkter, vilket är basstationer som vidarebefordrar trafiken till och från det trådbundna nätet.

Olika siffror anges för de trådlösa nätverksprodukternas räckvidd. Ur säkerhetsperspektiv är det bäst att räkna konservativt och avrunda uppåt. Man kan räkna med att vanliga nätverkskort utan extra antenner och i en radiomiljö utan hindrande väggar har en räckvidd på cirka 100 meter. Även med fönster, väggar och liknande når de trådlösa nätverken långt.

Vissa leverantörer av trådlös utrustning tillåter att administratörer och användare kan ändra inställningar för sändareffekt, hur stark radiosignalen skall bli. Det går att argumentera för att dra ner effekten på WLAN-utrustning för att göra så att nätverket inte läcker utanför det område som man fysiskt har kontroll över. Det är dock en bedräglig tanke. Genom att använda riktantenner, ibland hembyggda såkallade cantennas[98], kan någon avlyssna trådlös trafik på ännu längre avstånd. Därför får man räkna med att nätverken är åtkomliga utanför de egna lokalerna..

8.1 Säkerheten i 802.11

Det främsta hotet mot säkerheten i trådlösa nät är avlyssning och intrång. Trådlösa nät ger inte samma säkerhet som trådbundna nät. Attacker mot de senare kräver fysisk åtkomst till nätverket, vilket i sin tur oftast kräver fysisk åtkomst till lokaler. Trådlösa nät kan lätt bli tillgängliga långt utanför en organisations fysiska lokaler - i alla tre dimensioner. I 802.11, men inte i den nyare varianten 802.11b, så kan man använda frekvenshopp enligt metoden FHSS (Frequency Hopping Spread Spectrum) mellan 79 olika kanaler. Frekvenshopp används primärt för att undvika radiostörningar men gör det svårare för någon att följa och avlyssna radiokommunikationen. I 802.11b används en annan metod som kallas DSSS (Direct Sequencing Spread Spectrum).

Wired Equivalent Privacy, WEP, är en säkerhetsmetod som ingår i 802.11-standarden. Den används för att skydda radiotrafiken från avlyssning genom att erbjuda konfidentialitet. Många leverantörer erbjuder två varianter av WEP-lösningar, dels med 40-bitars RC4-kryptering, dels med 104-bitars RC4-kryptering.

I realiteten ger WEP liten eller ingen säkerhet beroende på en mängd implikationer. Det handlar bland annat om att möjligheten att välja goda WEP-nycklar begränsas av dumma användargränssnitt som inte tillåter alla tecken. Det handlar också om lyckade attacker mot nyckelgenereringsimplementationen av RC4 och hur många tillverkare hanterar den

[98] http://en.wikipedia.org/wiki/Cantenna

initieringsvektor som används tillsammans med kryptonyckeln för att kryptera data. Publicerade analyser av nyckelhanteringen visar på stora problem. De är sårbara för så kallade FMS-attacker[99], som kommer utav en svaghet i hur WEP använder RC4 och dess nyckelskedulering. Namnet kommer från de tre forskare, Fluhrer, Mantin och Shamir, som påvisade de teoretiska bristerna. Sårbarheten som FMS-attacken påvisade innebar att det gick att göra kryptoanalys på den trådlösa trafiken. Om en angripare lyckas samla in en viss mängd krypterad trafikdata går det att härleda RC4-nyckeln utifrån insamlat material.

Ett annat säkerhetsproblem relaterat till WEP är dessutom att de flesta produkter bygger på lösningar där WEP använder ett manuellt nyckelutbyte, dvs att en administratör skall konfigurera in ett WEP-lösenord i accesspunkter och ansluten utrustning. Detta handhavandesätt gör att lösningen det ”skalar” och följaktligen inte går att använda i stora organisationer.

I viss accessutrustning finns det extra säkerhetsfunktioner, till exempel MAC-adresslåsning och brandväggsfunktioner. En annan, lite mer udda, men farlig, risk med trådlösa nät är att man kan sätta upp falska accesspunkter eller att man som en nod i ett trådlöst nät kan introducera falska tjänster, till exempel en DHCP-server. En sådan attack innebär att man kan knyta andra ovetande klienter till sin tjänst eller accesspunkt. Därmed kan man skapa mannen-i-mitten-attacker eller andra hot.

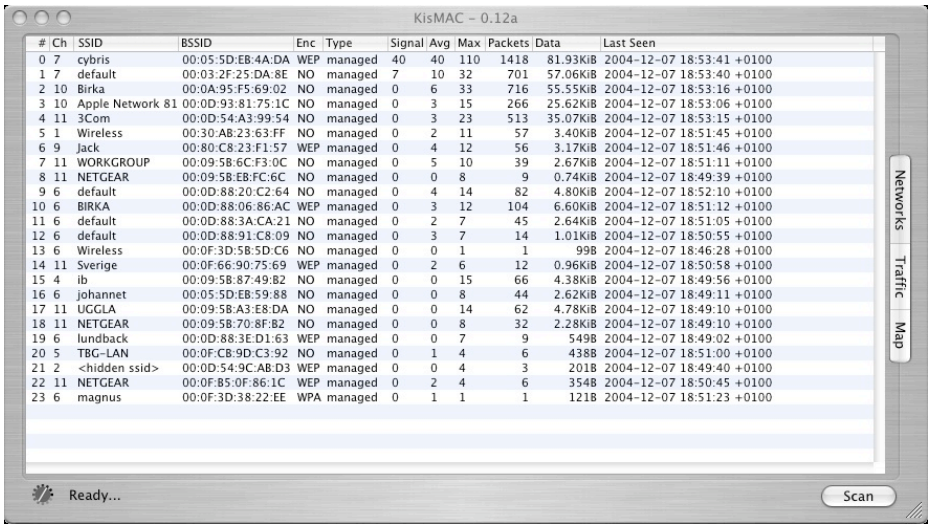
För närvarande arbetar tillverkarna på såväl helt egna lösningar som nya standarder för att skapa bättre säkerhet i de trådlösa näten bland annat TKIP, *temporal key integrity protocol*. Några nya intressanta standarder under bearbetning är IEEE 802.1x och IEEE 802.11i. Den nya versionen av 802.11-stadarden skall ha en förbättrad säkerhetslösning i form av WPA, Wi-Fi Protected Access. Varianten WPA2 innehåller bland andra förändringar stöd för krypteringsalgoritmen AES istället för RC4 samt att WPA har två varianter, en med delad hemlighet, WPA-PSK (pre-shared ket) och en enterprise mode, där man kan koppla den trådlösa autentiseringen mot centrala autentiseringsservers med hjälp av protokollet RADIUS.

8.2 Wardriving

Wardriving eller netstumbling är två olika namn på ett fenomen som innebär att någon eller några personer åker runt med en dator som är utrustad med ett trådlöst nätverkskort, och letar efter oskyddade, öppna eller knäckbara nätverk.

Bilden här intill visar programmet kismac[100]. Det är ett verktyg för att leta efter trådlösa nät enligt standarden 802.11. Näten hittades genom att gå omkring med en bärbar dator i författarens hem. Om en GPS-mottagare är ansluten till datorn kommer den att frågas om positionen för nätet som hittats.

[99] http://en.wikipedia.org/wiki/Fluhrer,_Mantin,_and_Shamir_attack
[100] http://kismac.macpirate.ch/



#	Ch	SSID	BSSID	Enc	Type	Signal	Avg	Max	Packets	Data	Last Seen
0	7	cybris	00:05:5D:EB:4A:DA	WEP	managed	40	40	110	1418	81.93KiB	2004-12-07 18:53:41 +0100
1	7	default	00:03:2F:25:DA:8E	NO	managed	7	10	32	701	57.06KiB	2004-12-07 18:53:40 +0100
2	10	Birka	00:0A:95:F5:69:02	NO	managed	0	6	33	716	55.55KiB	2004-12-07 18:53:16 +0100
3	10	Apple Network 81	00:0D:93:81:75:1C	NO	managed	0	3	15	266	25.62KiB	2004-12-07 18:53:06 +0100
4	11	3Com	00:0D:54:A3:99:54	NO	managed	0	3	23	513	35.07KiB	2004-12-07 18:53:15 +0100
5	1	Wireless	00:30:AB:23:63:FF	NO	managed	0	2	11	57	3.40KiB	2004-12-07 18:51:45 +0100
6	9	Jack	00:80:C8:23:F1:57	WEP	managed	0	4	12	56	3.17KiB	2004-12-07 18:51:46 +0100
7	11	WORKGROUP	00:09:5B:6C:F3:0C	NO	managed	0	5	10	39	2.67KiB	2004-12-07 18:51:11 +0100
8	11	NETGEAR	00:09:5B:EB:FC:6C	NO	managed	0	0	8	9	0.74KiB	2004-12-07 18:49:39 +0100
9	6	default	00:0D:88:20:C2:64	NO	managed	0	4	14	82	4.80KiB	2004-12-07 18:52:10 +0100
10	6	BIRKA	00:0D:88:06:86:AC	WEP	managed	0	3	12	104	6.60KiB	2004-12-07 18:51:12 +0100
11	6	default	00:0D:88:3A:CA:21	NO	managed	0	2	7	45	2.64KiB	2004-12-07 18:51:05 +0100
12	6	default	00:0D:88:91:C8:09	NO	managed	0	3	7	14	1.01KiB	2004-12-07 18:50:55 +0100
13	6	Wireless	00:0F:3D:5B:5D:C6	NO	managed	0	0	1	1	99B	2004-12-07 18:46:28 +0100
14	11	Sverige	00:0F:66:90:75:69	WEP	managed	0	2	6	12	0.96KiB	2004-12-07 18:50:58 +0100
15	4	ib	00:09:5B:87:49:B2	NO	managed	0	0	15	66	4.38KiB	2004-12-07 18:49:56 +0100
16	6	johannet	00:05:5D:EB:59:88	NO	managed	0	0	8	44	2.62KiB	2004-12-07 18:49:11 +0100
17	11	UGGLA	00:09:5B:A3:E8:DA	NO	managed	0	0	14	62	4.78KiB	2004-12-07 18:49:10 +0100
18	11	NETGEAR	00:09:5B:70:8F:B2	NO	managed	0	0	8	32	2.28KiB	2004-12-07 18:49:10 +0100
19	6	lundback	00:0D:88:3E:D1:63	WEP	managed	0	0	7	9	549B	2004-12-07 18:49:02 +0100
20	5	TBG-LAN	00:0F:CB:9D:C3:92	NO	managed	0	1	4	6	438B	2004-12-07 18:51:00 +0100
21	2	<hidden ssid>	00:0D:54:9C:AB:D3	WEP	managed	0	0	4	3	201B	2004-12-07 18:49:40 +0100
22	11	NETGEAR	00:0F:B5:0F:86:1C	WEP	managed	0	2	4	6	354B	2004-12-07 18:50:45 +0100
23	6	magnus	00:0F:3D:38:22:EE	WPA	managed	0	1	1	1	121B	2004-12-07 18:51:23 +0100

Figur 37 Trådlösa nät hittade med Kismac från författarens hem

På olika webbadresser såsom www.wigle.net och www.netstumbler.com går det att hitta omfattande databaser dit folk har skickat in resultat från nät de lokaliserat. Kopplat till databasen finns olika geografiska kartor över bland annat USA och Europa samt ett system som möjliggör i inzoomning på upp till 100 meter. Detta gör det lätt för någon att lokalisera ett trådlöst nät som är inlagt i databasen.

Andra program, såsom kismet[101], airsnort[102] och wepcrack[103], kan användas för att analysera den WEP-krypterade trafiken och forcera kryptot för att komma åt klartexttrafik. Airsnort bygger på en statistisk analys av en tillräckligt stor trafikmängd för att kunna forcera kryptonyckeln utifrån avlyssnad trafik. Dessa attacker har beskrivits av olika forskarteam, vartefter mer allmänt spridda verktyg har börjat dyka upp på olika ställen på Internet. Det finns varierande uppgifter på hur mycket data som behövs samlas in för att det skall gå att knäcka trafiken. Tidsangivelser från 15 minuter till 8 timmar är uppgifter som återkommer.

Det har till och med publicerats artiklar i svensk press om personer som utfört wardriving med hjälp av flygplan, så kallad warflying. Andra exempel har också beskrivnits i litteraturen och i nyhetsartiklar där folk har cyklat, åkt buss, åkt båt, etc, för att leta efter trådlösa nätverk.

[101] http://www.kismetwireless.net/
[102] http://airsnort.shmoo.com/
[103] http://wepcrack.sourceforge.net/

9 Viktiga stödfunktioner

9.1 DHCP, bootp och RARP

Dynamic Host Configuration Protocol, DHCP, är ett protokoll för att tillfälligt låna ut IP-adresser och dynamiskt konfigurera upp nätverksinställningar till datorer som ansluts till ett nät. Bootp är ett snarlikt, men mer begränsat, protokoll som kan betraktas som en föregångare till DHCP. Loggning bör vara aktiverad i DHCP-servrar så att man vid ett senare tillfälle kan kontrollera loggarna och se vilken dator som var tilldelad en viss IP-adress vid ett specifikt tillfälle. Felkonfigurerade DHCP- och bootp-servrar gör det möjligt för utomstående att ansluta till nätverket. Detta är vanliga misstag som möjliggör anonym anslutning – och som dessutom inte lämnar något spår efter vilken utrustning som anslöts eller när detta skedde. En typ av attack är införandet av falska DHCP-servers, vilket kan leda till att man kan manipulera den utrustning som ansluter till nätet och ber om DHCP-tjänsten. En vidareutvecklad variant av DHCP finns framtagen för IP version 6, kallad DHCPv6

DHCP är definierat i RFC 2131 (”Dynamic Host Configuration Protocol”), RFC 2132 (”DHCP Options and BOOTP Vendor Extensions”). Ett flertal mer DHCP-relaterade RFC:er existerar. DHCPv6 är definierat I RFC 3315 (”Dynamic Host Configuration Protocol for Ipv6”).

Reverse Address Resolution Protocol, RARP, är ett äldre protokoll för att översätta ifrån en MAC-adress till en IP-adress. RARP är i stort sett ersatt av DHCP. RARP används ofta när en nätansluten utrustning utan egen konfiguration skall ansluta till nätet och därmed söker tilldelning av IP-adress och därefter följande uppkonfiguration via protokollet BOOTP.

9.2 Logghantering

En väldigt viktig, men ofta förbisedd, säkerhetsaspekt är spårbarhet. Spårbarhet betecknar en tjänst som används för att kunna se vad som inträffat i ett system, när det inträffade och vem som utförde händelsen. Loggning eller audit är två ord som ofta betecknar skapandet av spårmeddelanden.

Protokollet syslog, som används för loggning, är vanligt i nätverkssammanhang och finns inbyggt i de flesta kommunikationsutrustningar idag. Syslog är en bastjänst som används i många nät idag för skapande, insamlande och övervakning av nättjänster och system. Syslog är UDP-baserat och servern har tilldelats port 512.

Syslog implementerades ursprungligen i Berkeley-dialekten av UNIX (BSD UNIX), men finns idag implementerad för de flesta nätverkskomponenter, såsom routrar, växlar, hubbar och liknande.Protokollet syslog finns definierat i RFC 3168.

9.3 Tidssynkronisering

Det är extremt viktigt att olika system, datorer och utrustningar har en gemensam uppfattning om tid. Flera tjänster, såsom inloggningstjänster för engångslösenord och Kerberos, kräver korrekt tid för fungera. En annan anledning till detta är att loggar som genereras i olika system

måste kunna jämföras mot varandra och detta är bara möjligt om man har god tidssynkronisering mellan systemen.

Det vanligaste protokollet för att hantera tidssynkronisering över datornätverk heter *Network Time Protocol*, NTP. NTP använder sig av avancerade algoritmer för att kompensera för överföringstider mellan fråga och inkomna svar, utröstning av avvikande tidskällor och liknande. NTP används för att bygga hierarkier av klockserverdatorer. Överst i hierarkin, på stratumnivå 1, finns det system med referensklockor som är anslutna till stabila tidskällor, till exempel atomklockor, GPS-mottagare och liknande. Andra klockservrar hämtar tidsinformation från klockservrar på en överliggande stratumnivå. Idag har statens provningsanstalt, SP, publika NTP-servrar i drift som kan användas för synkronisering. Adressen till dessa är ntp1.sp.se och ntp2.sp.se

Ett praktiskt säkerhetsproblem med NTP är att klienten ofta har en NTP-tjänst exekverande permanent. Detta leder till att andra kan ställa frågor till NTP-porten på klienten. Med dessa frågor kan man antingen få ut viss information ur NTP-konfigurationen. Ett annat större problem är i det fallet som NTP-programvaran innehåller något buffertöverskrivningsfel, då kan klienten bli angripen. En lösning på detta är att köra ett annat klientprogram som periodiskt, via cronjobb eller liknande, synkroniserar mot servern.

NTP använder sig av UDP-port 123 och protokollet som sådant finns definierat i RFC 1305. Det finns en förenklad variant av protokollet också. Det kallas för Simple Network Time Protocol, SNTP och finns definierat i RFC 2030.

9.4 Autentiseringstjänster

Några vanliga protokoll för autentisering och auktorisation av användare och utrustning är RADIUS, TACACS i dess olika varianter samt DIAMETER.

RADIUS och TACACS autenticeringstjäster skapades ursprungligen för att användas vid uppringda PPP-förbidelser och åtkomst till termialserverutrustning. När nya typer av datorkommunikationstjänster uppstått, såsom åtkomst via bredbandsanslutningar, mobil IP och trådlösa nät, så har man även använt dessa autentiseringstjänster där, även fast det ibland inte optimalt. Därför har man vidareutvecklat dessa koncept och tagit fram DIAMETER som skall vara bakåtkompatibelt med RADIUS men med mer flexibilitet och funtioner.

Användning av Remote Authentication Dial In User Service, RADIUS, utvecklades ursprunglingen en tillverkare men som öppnade upp protokollet och gjorde det tillgängligt för IETF. RADIUS-tjänsten av tre komponenter, autentiseringstjänsten, klientprotokoll och en logg- och debiteringstjänst. Protokollet baseras på UDP och definierar i egentlig mening ingen omsändning och felkontroll vid transmissionsproblem.

RADIUS definieras i RFC 2865 och i RFC 2139 för att sedan uppdateras i RFC 2868 (”RADIUS attributes for Tunnel protocol Support”) och RFC 3575 (”IANA Considerations for RADIUS”).

TACACS står för Terminal Access Controler Access Control System och utvecklades ursprungligen av Cisco och . Flera versioner av TACACS existerar, exempelvis XTACACS och TACACS+

Diameter är ett nytt protokoll som skall överbygga ett antal problem med TACACS och RADIUS samtidigt som det tillför ny funktionalitet och säkerhet. Diameter skall vara bakåtkompatibelt med RADIUS.

Protokollet TACACS definieras i RFC 1492 (”An Access Control Protocol, Sometimes Called TACACS”). Protokollet Diameter specificeras i RFC 3588 (”Diameter Base Protocol”).

10 Fysiskt skydd

Denna text är fokuserad på logisk säkerhet, inte fysisk säkerhet, så vi kommer inte att gå in på passagesystem, brandskydd i datorhallar och liknande. Några mer specifika fysiska skydd att tänka på i sammanhang med fokus på nätverkssäkerhet förtjänar dock lite uppmärksamhet. Den gamla sanningen om att ingen dator är säker om någon har fysisk åtkomst till den kan lätt parafraseras och användas även i nätverkssäkerhetssammanhang.

10.1 Fysisk åtkomst till nät

Ett ofta förbisett eller bortglömt säkerhetsproblem är möjligheten för utomstående eller obehöriga till åtkomst till nätverksuttag eller nätverksutrustning utanför skalskyddet, till exempelvis i receptioner, demoutrustning temporärt uppställda på mässor, i väntrum eller i delar av lokalerna som organisationen inte längre besitter. Dessa scenarion är hämtade från verkliga händelser och dessa och andra varianter kommer garanterat hända igen. Ofta ser man åtkomliga Ethernetuttag när man väntar i kön på banken, väntar på vårdcentralen och liknande. Därför är fysiskt skydd av nätverksutrustning och nätverksuttag något som måste beaktas.

Fysiskt skydd kan vara att ha väl etablerade och fungerande rutiner för in- och utkoppling av nätportar och nätutrustning utanför det inre skalskyddet. Det fysiska skyddet måste givetvis kombineras med logiska skydd, tex att dessa portar normalt sätt är nätmässigt logiskt avaktiverade, att DHCP-adresser inte delas ut till okända datorer och liknande, att man har certifikat eller andra metoder för att autenticera datorer innan de får ansluta till nätverket.

10.2 Avbrottsfri kraft

En annan viktig fysisk säkerhetsaspekt som tål att påpekas gång efter annan är kopplingen el, tele, data. Utan el så fungerar givetvis inte vare sig tele- eller datorsystem. Ofta är det också ett nära släktskap mellan telesystem och datorkommunikation som man också behöver ha i bakhuvudet vid riskdisussioner, nätdesign och liknande. Den mer obekanta kopplingen mellan tele- och datorsystem för att kontrollera och övervaka elproduktion och eldistribution måste också täckas in. I svåra fall kan cirkelberoenden förekomma som leder till onödigt långa driftstörningar.

Reservkraft i form av avbrottsfri kraft med batteribackuper (UPS) och t.o.m. dieselkraftverk finns ofta i anslutning till datorcentraler, datorhallar och större datorrum. Nätverksutrustning som är fysiskt utspridd i en byggnad är ofta inte ansluten till denna typ av avbrottsfri kraft, vilket leder till oönskade driftsavbrott.

Inte sällan tillför den avbrottsfria kraften en viss mängd driftstörningar, exempelvis när det uppstår fel på UPS-utrustningen,servicetekniker kopplar fel vid service av reservkraften, avgasutblåset är igensatt när man testkör dieselaggregaten, att dielsel till reservkraften blivit förstörd pgr av att den varit oanvänd för lång tid och liknande problem.

10.3 Redundans och nätmässiga reservvägar

Förutom att det behövs reservkraft för att de ordinanrie kraften kan falla bort så behövs det även reservkapacitet på andra områden såsom kabeldragning och nätutrustning. I nätverkssammanhang så designas ofta nätverk med viss redundans och reservkapacitet, men

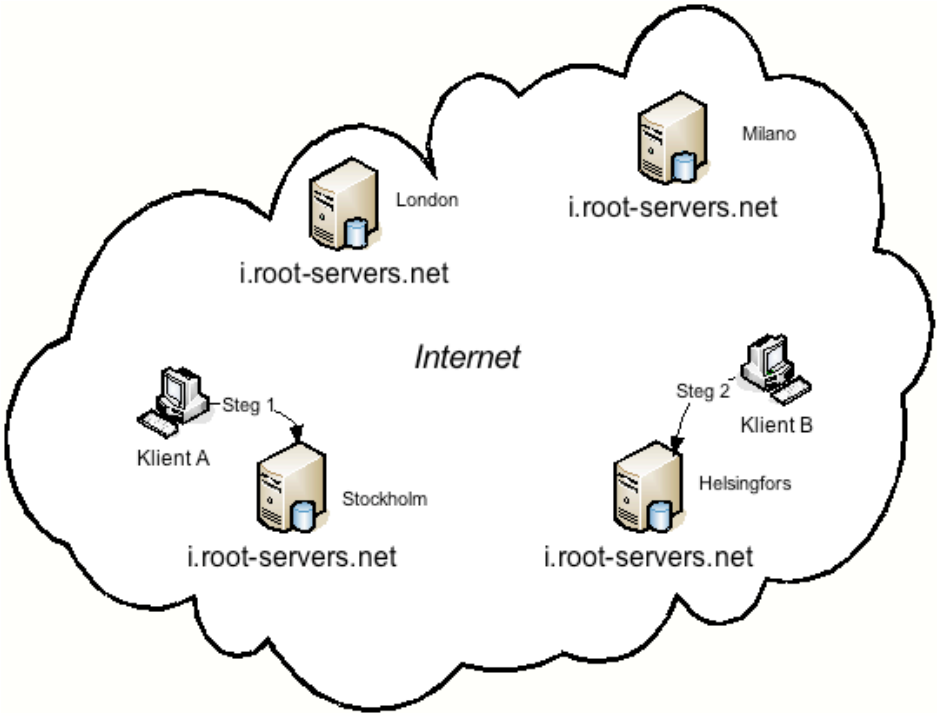
det brukar oftast enbart vara på WAN-länkar och stamnät. Ett stamnät kan vara uppbyggt i en form där man har alternativa vägar mellan olika anslutningspunkter. Ibland väljer man av kostnadsskäl att lämna vissa mindre anslutningspunkter utanför redundanslösningen, de är lövnoder, och de redundanta anslutningsvägarna finns bara hos vissa större anslutningspunter via vilka de mindre i sin tur är anslutna. Redundans på fysisk WAN-nivå byggs ibland upp med enstaka alternativanslutningar såsom beskrivits ovan, eller så kan det byggas som med en ringtopologi eller i det teoretiskt totalredundanta fallet så har man ett fully-meshed nät. Ett nät där alla kan tala med alla.

Förutom redundans på WAN-nivå så behövs det ofta, även fast det glöms bort eller förbises, redundans även på LAN-nivå. När LAN designas eller installeras behövs det skapas flera vägar genom nätverk, så att man kan utföra service och underhåll på utrustning såsom switchar och routers med bibehållen eller något reducerad funktionalitet. Detta blir alltmer viktigt då datakommunikationen används för alltfler tjänster, såsom telefoni, larm- och passagesystem, som tidigare byggde på separat kablage och egna teknologier.

Bland frågeställningarna vad det gäller reservanslutningar och reservkapacitet är om man skall välja flera anslutningar från en kommunikationsleverantör eller om man skall använda flera olika leverantörer för att på så sätt uppnå skydd mot fel i deras tjänster. Ett specialfall i det sammanhanget är valet av internetleverantör där det dessutom innebär alternativa åtkomstvägar via internet. Tänk på att olika nätoperatörer kan dela på fysiskt kablage och utrustning, såsom telestationer, optoförstärkare eller liknande så att man i vissa fall ändå är sårbar mot enkelfel som slår igenom mot flera tjänsteleverantörer.

Lika viktigt som alternativvägar i det stora nätet är det att tänka på att ha flera olika fysiska kommunikationsin- och utgångar ur byggnaden. En avgrävd kabel precis utanför huset kan ju annars i värsta fall slå ut både primär- och sekundära nätvägar.

Ett sätt att bygga in nätverksmässig redundans till olika nättjänster är att använda sig av konceptet *Anycast*, där man med hjälp av routingteknologi och nätverksaddresseringsmetoder ser till att skicka nätverkstrafik till den tjänst som är ”närmast” i nätet. Anycast kan ses som en mix av logisk och fysisk redundans. På så sätt går det att bygga upp redundanta tjänster som är åtkomliga för lokala användare. Ett angrepp eller en störning slår inte ut alla värddatorerna eller tjänsten som sådan, utan bara vissa delar av tjänstens infrastruktur drabbas. En internetbaserad tjänst som använt sig av Anycast för att skapa robusthet i tjänsteleveransen av de centrala delarna är domännamnssystemet DNS.



Figur 38 Exempel på användning av AnyCast för tjänsten DNS

I bilden ovan så används Anycast för att skapa fler uppsättningar av DNS-servern i.root-server.net som blir åtkomliga för olika användare i nätet, beroende på användarens olika routing- och adressförutsättningar. Anycasttjänsten kom att byggas ut efter en mängd uppmärksammade angrepp mot DNS centrala servrar, de så kallade rotnamnservrarna.

10.4 Strålning

Utrustning avger strålning i form av radiovågor eller elektromagnetisk strålning. Röjande signaler, *RÖS*, är ett samlingsnamn för strålning och annat som utrustning avger, till exempel akustiskt ljud. Hotet är att någon på avstånd kan uppfatta, infånga och analysera signalerna.

Dator- och kommunikationsutrustning kan drabbas av *elektromagnetisk puls*, EMP, som tillfälligt eller permanent slår ut funktioner och komponenter. EMP har man traditionellt sett bara kalkylerat med som en bieffekt av en atomvapenexplosion. Därför har man i sin IT- och nätversksäkerhet ofta kunnat överse med riskerna för denna typ av problem. På senare år har det talats mer och mer som EMP som ett vapen för utpressning eller sabotage då alternativa EMP-källor som dessutom är gjorda som mobil utrustning har skapats. Använder man radionät, såsom trådlösa LAN, är det uppenbart att man sänder ut information i etern som annars antas skickas säkert via trådbundna kanaler. Men kablar strålar också ut information och kan fungera som antenner.

Oskärmad kabel, såsom vanlig Ethernetkablage, läcker information, som skulle kunna uppfångas med särskilld utrustning. Ett intressant exempel på RÖS är elnätskommunikation, såkallad PLC - powerline communication. Det är en teknologi där man skickar datakommunikation över vanliga, oskärmade elkablar. PLC har av vissa kritiseras för att vara en teknologi som genererar såpass mycket strålning att den konkurrerar och stör ut vissa radiofrekvenser.

Skydd mot avsändande av strålning och signaler är olika typer av skärmning av kablage, utrustning och lokaler där utrustningen brukas. Skydd mot EMP är att belägga utrustning till lokaler som snarast liknar skyddsrum, bygger på principen om Faradays bur och ofta är belagda i bergrum eller liknande skyddade utrymmen.

11 Kritisk infrastruktur

Det moderna samhället har byggt upp ett ökat behov av information och kommunikation för att klara av bastjänster. Vid avbrott eller störningar som påverkar elförsörjning eller tele- och datorkommunikation så påverkas kritiska funktioner som SOS alarms nödnummer 112, trygghetslarm som installerats hos äldre, Vid teleavbrott kan. trådbunden telefoni och mobiltelefoni, likväl larmsystem och fjärrdriftssystem slås ut. Mindre kritiska funktioner såsom kreditkortskontroll kan slås ut med resultatet att folk som inte har kontanter inte genomför sina affärer.

Till kritisk infrastruktur kan vi räkna, bland annat, viktiga IT-system på nationell nivå, elförsörjning, tele- och datakommunikationer, radio- och TV-produktions- och distributionsssystem, gasnät, spårbunden trafik, färskvattensförsörjning, avlopp och fjärrvärme. Det viktiga, ur vår IT- och nätverkssäkerhetsmässiga synpunkt, är att konstatera att dessa typer av tekniska infrastrukturer i under år av utveckling har fått en ökande grad av beroende på IT och datornätverksteknik. Det innebär att sårbarheter i de IT- eller nätverkskomponenter som är inblandad i samhällets kritiska infrastruktur kan nyttjas för att påverka, tillexempel slå ut eller minska samhällsmedborgares tilltro till, dessa infrastrukturer.

Då samhället har detta starka, och som det sig verkar ständigt ökande, beroende på olika delar av kritiska samhällsinfrastrukturer, så ökar även behovet av skydd av dessa infrastrukturer. Detta skydd är till stora delar fysiskt, exempelvis genom områdesskydd i form av stängsel, tillträdesskydd, personkontroll av den personal som arbetar med infrastrukturen, etc. Det vi är intresserade av i denna text är däremot det logiska skyddet av denna infrastruktur, som i vissa delar är dålig eller saknas helt.

Skador i eller på en del av infrastrukturen kan få konsekvenser för andra delar i samhället ex. kan ett avbrott i elförsörjningen påverka informationssystemen som i sin tur leder till att andra system slutar att fungera.

Problem med delar av den kritiska samhällsinfrastrukturen kan bero på såväl oavsiktliga hot såväl som rikade hot. En höststorm kan innebära knäckta radioänkmaster eller kraftledningsstolpar. En nedfallen kraftledningsstolpe kan leda till såväl avbrott i strömförsörjning som störningar på kommunikation då många stolpar även utrustats med fiberkablage i topplinan. Ett annan vanlig anledning till avbrott i kommunikationsförsörjning är kabelavgrävning. I stockholmsförorten Kista har en serie[104] [105] uppmärksammade och omfattande kabelbränder inträffat under en relativt kort tidsperiod. Kabelbränderna har påverkat annan infrastruktur och resulterat i avbrott i såväl tele- och datorkomunikationen och persontransport i form av T-banan.

Medvetna attacker, exempelvis att slå ut möjligheten för ett inbrottslarm att sända larm till ansluten övervakningscentral, kan resultera i stora bieffekter i utslagen annan kommunikation, exempelvis data- och telekommunikation till hela det aktuella geografiska området.

[104] Fischer & Jönsson ”Elavbrottet i Kista-området mars 2001 – konsekvenser för hushåll och befolkning” http://www2.foi.se/rapp/foir1548.pdf

[105] Deverell, Edward ”Elavbrottet i Kista den 29-31 maj 2002: Organisatorisk och interorganisatorisk inlärning i kris” http://www.crismart.org/templates/Page____334.aspx

11.1 Informationsoperationer

Attacker utförda, eller understödda, av stater eller terrorister brukar kallas för informationsoperationer, tidigare ofta kallat informationskrigsföring, är åtgärder som syftar i att påverka människors beslutsfattande. Man talar i dessa sammanhang ofta om civila respektive militära informationsoperationer. Syftet kan vara att skada någons goda rykte genom att skapa ”bad-will”. Det finns flera exempel på när man använt Internet för att utföra informationsoperationer i syfte att sprida aktiekursdrivande information.

Den svenska försvarsmaktens definition av informationsoperationer är:

> *"Verksamhet för att utnyttja, förvanska eller fördröja motståndarens information samt att skydda vår information från utnyttjande, förvanskning och fördröjning."*

En annan attacktyp som kan få allvarliga konsekvenser är sabotage med hjälp av IT eller mot IT-system och IT-infrastruktur.

11.2 SCADA och processkontroll

Digitala kontrollsystem och så kallade SCADA-system (*Supervisory Control And Data Aquisition*) används för att sköta processystem, exempelvis elproduktion eller vattendistribution, som utgör viktiga delar av kritisk samhällsinfrastruktur.

De viktigaste funktionerna hos ett kontrollsystem är:
- *Styrning* (direktstyrning, börvärdesstyrning, sekvensstyrning, etc.)
- *Datainsamling* (datalagring, omvandling och skalning, tidsmärkning, rimlighetsbedömningar etc.)
- *Övervakning* (statusövervakning, trendövervakning, gränsvärdesövervakning, prestandaövervakning, händelse- och larmhantering etc.)
- *Planering/uppföljning* (Icke-realtidsfunktionalitet; planering, loggning och lagring, uppföljning och analys etc.)
- *Underhåll/förändring* (i och urtagning av drift, uppgraderingar, hantering av utvecklingsmiljöer etc.).

11.2.1 Säkerhet i digitala kontrollsystem

En av de farhågor som diskuteras bland säkerhetsspecialister, militärer, terroristforskare och andra är terrorangrepp mot kritisk infrastruktur, och då inte minst i form av IT-baserade angrepp.

SCADA-säkerhet är ett område som fortfarande är i sin linda, på grund av mängder med tekniska och organisatoriska frågor. En teknisk omsvängning från leverantörsspecifik hårdvara, mnukvara och kommunikationsprotokoll till standardplattformar och TCP/IP som kommunikationsplattform innebär ökad exponering. Samtidigt så används TCP/IP oftast som en tunneltjänst till leverantörsspecifika protokoll, för vilket det saknas stöd i de flesta brandväggar. Samma problem gäller exempelvis antivirus. Vanliga antivirusprogram går oftaste att köra på denna typ av system, på grund av den realtidsmiljö i vilken systemen används.

Trots dessa begränsningar och allmänna problem, så pågår mycket aktivitet inom området IT-säkerhet och digitala kontrollsystem, exempelvis olika typer av säkerhetsgranskningar, penetrationstester och liknande.

I amerikansk litteratur förekommer ofta terminologin *cyber security* för att definiera de IT-säkerhets- och nätverkssäkerhetsmässiga aspekterna av process- och SCADA-systemen.

12 Upphovs-, immaterialrätt, varumärken och liknande problem

Ett ständigt gissel har varit kapade domäner, domännamn som registrerats av andra än de som kan anses vara namninnehavaren eller dispyter mellan flera som anser sig ha rätt till ett visst namn. En speciell kategori problemskapare är domännamnspirater, dvs personer eller organisationer som uppsåtligen registrerar domännamn som är är välkända produkt och företagsamn, varumärken eller liknande. Avsikten med domänregistreringen är ofta att sedan sälja domännamnet till andra som önskar komma över domänen. I takt med att registrerigskraven hos de som håller domännamnsregister blivit anpassade så har möjligheterna att komma åt domännamnspiraterna ökat.

Genom marknadsföringslagar, varumärkeslagar och firmalagar kan en legitim namninnehavare angripa domännamnspirater.

Mellan varven blossar stora problem och ordkrig upp till följd utav att någon nyttjat någon annan organisation eller persons material på sin webbsida, i epost eller liknande. Internet och den utökade användningen av digitala format förenklar avsiktliga och ibland aningslöst missbruk av andras material. Det förekommer fall där domännamn som innehåller varumärken i komination med obsceniteter eller annan kritik registreras. Det är vanligt att dessa huserar webbtjänster där kritik eller liknande framförs.

Lobbyinggrupper och olika grupperingar av meningsmotståndare eller kritiker har gjort det lättare för sig att nå ut till likasinnade med hjälp av främst webb, e-post och fildelning. Kostnader för kampanjer, oavsett om det är seriösa granskningar eller enklare smutskastningsaktioner, är små med dessa
nya medel.

Ett kontinuerligt problem på Internet är piratkopiering av olika typer av upphovsrättsligt skyddat material såsom program, musik och filmer. Programmen och informationen, kallat *warez*, byter händer enkelt med hjälp av program med användarvänliga sökfunktioner. Genom peer-to-peernätverk, ofta kallade P2P-nät, såsom BitTorrent, direct connect (DC), KaZaA, freenet och liknande skaffar sig bredbandsanvändare och andra åtkomst till information och digitala produkter. De olika P2P-protokollen använder sig av olika metoder för att undvika centraliserad, och därmed lättidentiferade, servrar och kontrollpunkter. Genom att distribuera hela verksamheten så försvårar man upptäckt och motstånd mot verksamheten. Internet- och bredbandsleverantörer har gång på gång beskrivit att deras bandbreddsanvändning i hög utsträckning, ofta upp över 90%, används till olika P2P-protokoll. Eftersom det pågår en ständig kamp mellan producentföreningar och branschorganisationer å ena sidan och nätverksanvändare å andra sidan så förutspås P2P-nätverken i framtiden övergå till att använda än mer avancerade protokoll samt mer kryptering för att undvika avlyssning och insyn i verksamheten.

Organisationer som antipiratbyrån och Business Software Alliance, BSA, arbetar med att stävja problem med piratkopierad programvara. Inom musikindustrin finns RIAA, Recording Industry Association of America, och IFPI, International Federation of the Phonographic Industry. Filmbranschens organisation heter MPAA, Motion Picture Association of America. Gemensamt för dessa organisationer är att de utför ett omfattande nationellt och internationellt arbete för att motarbeta piratkopiering av programvara, musik och film.

Antipiratverksamheten omfattar informationskampanjer riktade mot företag och allmänhet, lobbyingverksamhet för förstärkt upphovsrätts- och imateriellagstiftning samt juridiska och civilrättsliga åtgärder mot upphovsrättsbrott. Tillslag görs mellan varven mot olika personer eller bolag som misstänks hålla på med piratkopiering. Tillslag som fått mycket publicitet[2] är tillslagen mot internetoperatören Bahnhof[106] och ThePirateBay[107].

[106] 2005-03-22 NyTeknik ”Razzian mot Bahnhof - rättskandal eller inte?” http://www.nyteknik.se/nyheter/it_telekom/allmant/article34986.ece
[107] Expressen 2006-05-31 ”Oskyldiga drabbade när Pirate Bay stängdes” http://www.expressen.se/1.364401

13 Sakregister

T

U

W

X

Z

Å

Ä

Ö

