Pääsynvalvonta I

- Pääsynvalvonta on ensimmäiseksi järjestettävä asia verkkoympäristössä.
- Pääsynvalvonnassa on ensiksi määriteltävä, ketkä saavat käyttöoikeuden mihinkin järjestelmän osaan. Eli ensiksi on määriteltävä pääsynvalvonnan politiikka.
- Sen jälkeen on otettava käyttöön mekanismi, turvamekanismi, jonka avulla toteutetaan valittu politiikka.
- Tämä politiikan ja sen toteutusmekanismin erottelu selventää määrittelyjä. On esimerkiksi mahdollista käyttää yleisiä politiikkakieliä määrittelemään oikeuksia, joita sitten valvotaan käyttöjärjestelmän ja sovellusohjelmistojen suomien mekanismien avulla.

Pääsynvalvonnan tyypit I

Pääsynvalvonnan tyypit voidaan jaotella kolmeen tai neljään perustyyppiin.

Määritelmä

Jos yksittäinen käyttäjä, yleensä objektin omistaja, voi asettaa objektin käyttöoikeudet, kysymyksessä on yksilöpohjainen pääsynvalvonta (engl. discretionary access control eli DAC, identity-based access control).

- Yksilöpohjaisessa pääsynvalvonnassa pääsyoikeudet perustuvat subjektin ja objektin identiteettiin. Identiteetti on tässä avainsana: objektin omistaja asettaa pääsyrajoitukset määrittelemällä, kuka saa pääsyn objektiin. Määrittely perustuu subjektien identiteettiin.
- Pääsymatriiseja (access control matrices, ACM) voidaan käyttää pitämään kirjaa, kuka saa käyttää mitäkin objektia. Muita mahdollisuuksia ovat pääsynvalvontalistat (access control lists) ja valtakirjat (capability lists).

Määritelmä

Sääntöpohjainen pääsynvalvonta (engl. mandatory access control, rule-based access control) on sellainen keskitetty järjestelmä, että objektit on luokitelty hierarkkisille tasoille objektin turvavaatimusten mukaisesti (esim. top secret, secret, confidential), subjekteille on asetettu turvatasot ja subjektilla on pääsy objektiin, jos subjekti-objekti -pari täyttää ennalta määritellyt turvallisuusehdot hierarkioiden ja turvatasojen perusteella. Siten systeemi valvoo, kuka pääsee käsiksi objekteihin, eikä yksittäinen käyttäjä voi tätä muuttaa.

Esimerkiksi käyttöjärjestelmä pakottaa noudattamaan sääntöpohjaista pääsynvalvontaa, ainakin tietyissä tilanteissa. Ei subjekti eikä objektin omistaja voi määritellä, kuka saa pääsyoikeuden.
Sääntöpohjaisella järjestelmällä estetään mm. Troijan hevosten kautta tapahtuva tietojen vuoto.

Määritelmä

Luontipohjainen pääsynvalvonta (engl. originator controlled, ORGON) perustuu objektin luojan määrityksiin.

Tässä politiikassa objektin, esimerkiksi tiedoston, luoja päättää, kenellä on pääsyoikeus tiedostoon. Tiedoston omistaja ei voi tätä muuttaa.

Lisäksi voidaan määritellä roolipohjainen pääsynvalvonta. Oikeuksia ei myönnetä yksittäiselle subjektille, vaan roolille, jonka eri subjektit voivat ottaa sallittujen sääntöjen puitteissa. Roolin kohdalla voidaan puolestaan noudattaa edellä mainittuja kolmea pääsynvalvontatyyppiä.

Monentasoinen pääsynvalvonta

Tällaiset järjestelmät ovat MAC-järjestelmän erikoistapauksia. Ideana on, että määritellään turvallisuuustavoite ja tiedonkulku järjestetään niin, että tavoite saavutetaan. Seuraavassa mainitaan muutamia tällaisia järjestelmiä nimeltä; yksityiskohdat sivuutetaan.

- *BLP- ja Biba-malli*: BLP:ssä tiedonkulku korkeammalta turvallisuustasolta alemmalle estetään. Alemman tason toimijat voivat modifioida ylemmän tason tietoja, mutta eivät voi lukea niitä. Muuttaminen aiheuttaa eheysongelmia, joita Biban malli yrittää korjata.
- *Kiinanmuuri*: Vuodelta 1989. Tavoitteena rajoittaa tiedonkulkua organisaatiossa, joka käsittelee kilpailevien yritysten tietoja.
- *Clarkin ja Wilsonin malli*: Soveltuu tilanteisiin, jossa tiedon eheys on tärkeämpi kuin luottamuksellisuus.

Pääsynvalvontamekanismit

- Pääsynvalvontamekanismeja on useita. Käyttöjärjestelmä huolehtii osittain pääsynvalvonnasta, ainakin tiedostojen suhteen.
- Salasanoilla on tärkeä asema pääsynvalvonnan toteutuksessa.
- Viime aikoina huomiota on saanut kryptografinen pääsynvalvonta, CAC. Se pyrkii samaan kuin monentasoiset mallit, mutta yleisemmin. Menetelmät perustuvat salakirjoitukseen ja erilaisiin avaimiin (ryhmäavaimet, hierarkkiset avaimet yms).

Yleistä salasanoista I

- Tavallisin todennusmekanismi perustuu salasanoihin. Tosin kriittisissä sovelluksissa voidaan käyttää biometrisiä tunnisteita salasanojen sijasta.
- Salasanat näyttävät tarjoavan hyvän suojan ulkopuolisia vastaan, joskin niiden huolimaton valinta ja käyttö saattavat aiheuttaa riskejä.
- Lisäksi ohjelmistoissa on otettava huomioon muutamia yksinkertaisia asioita. Ohjelmisto ei saa esimerkiksi kysyä käyttäjätunnusta ja ilmoittaa heti, että se on väärin. Tällöin tunkeutuja saisi tietoonsa, että tunnus ei ole oikea. Sen sijaan parempi on kysyä sekä käyttäjätunnusta että salasanaa ja ilmoittaa vasta sitten, jos jompi kumpi on virheellinen. Tällöin tunkeutuja ei saa tietoonsa, kumpi on virheellinen, tunnus vai salasana.

Yleistä salasanoista II

- Salasanojen lisäksi voidaan käyttää muitakin tekijöitä käyttäjän identifioimiseen. Voidaan esimerkiksi ottaa huomioon aika, vaikka virka-aika, jolloin sisäänkirjoittautuminen on mahdollista. Virka-ajan ulkopuolella pääsy kielletään.

Hyökkäykset salasanoja vastaan I

- Ensimmäinen yritys on kokeilla systemaattisesti erilaisia salasanoja samalla käyttäjätunnuksella. Tämä on varsin tehokas menetelmä, sillä kone kykenee arvaamaan suunnattoman määrän lyhyessä ajassa ja monet valitsevat heikkoja salasanoja (liian lyhyitä, luonnollisen kielen sanoja, liian yksinkertaisia modifikaatioita).
  Nykyään ohjelmistot torjuvat näitä hyökkäyksiä hidastamalla salasanojen tarkistusta, jos samalle tunnukselle yritetään antaa väärä salasana useaan kertaan.
- Anastetaan salasanatiedosto. Yleensä salasanat eivät ole selväkielisessä muodossa salasanatiedostossa, joten edelleen joudutaan systemaattisesti testaamaan eri vaihtoehtoja. Nykyään tähän on kuitenkin olemassa tehokkaita menetelmiä.
- Sosiaalinen urkinta on kolmas vaihtoehto. Tekeydytään joksikin auktoriteetiksi ja udellaan salasanoja "alaisilta". Myös sähköpostikyselyt ovat arkipäivää.

Salasanatiedosto ja suolaus I

- On aina mahdollista, että kaksi käyttäjää valitsee saman salasanan. Jos esimerkiksi Aapo ja Bertil molemmat valitsevat salasanan "aprilli" ja jos Bertil pääsee katselemaan salasanatiedostoa, hän huomaa, että Aapolla on sama salasana.
- Unix ratkaisee tämän ongelman laajentamalla salasanaa ns. suolalla.
- Alunalkaen suola oli 12-bittinen luku, joka muodostettiin systeemin ajasta ja prosessin tunnuksesta. Siten suola on todennäköisesti yksikäsitteinen jokaisella käyttäjällä.
- Suola liitetään käyttäjän salasanaan, kun salasana valitaan. Jos Bertilin salasana on p, niin salasanatiedostoon viedään $Hash(p||salt_B)$, eli alkuperäinen salasana lisättynä suolalla ja lopuksi lasketaan tiiviste.
- Lisäksi suola talletetaan Bertilin tunnuksen ja muutetetun salasanan yhteyteen.

Salasanatiedosto ja suolaus II

- Vanhoissa Unix-järjestelmissä suola oli 12-bittinen. Tällaiset salasanatiedostot voidaan nykyään murtaa ns. sateenkaaritaulujen avulla.
- Linuxissa suola on nykyään 48 ja Solariksessa 128 bittiä. Tällaiset pituudet estävät ennalta lasketut hyökkäykset pitkälle tulevaisuuteen.
- Windows NT/2000:teen kuuluvat LAN Manager NT LAN Manager eivät käyttäneet suolaa, mikä tekikin niistä suosittuja hyökkäyskohteita näille menetelmille.

Sateenkaaritaulukko I

- Sateenkaaritauluja käytetään laskemaan salasana-tiiviste -pareja etukäteen. Tuloksia voidaan sitten käyttää löytämään salasana, jos salasanatiedosto on saatu haltuun.
- Oletetaan, että salasanat on talletettu tiivistefunktioiden arvoina. Ts. jos p on salasana, siitä lasketaan tietty kiinteän pituinen arvo $H(p)$ tiivistefunktiolla H, joka on yksisuuntainen eli tuloksesta $H(p)$ ei voi päätellä argumenttia p. Tiivistefunktioiden yhteydessä voidaan käyttää salaisia parametreja (MAC-funktiot). Sen sijaan salausta ei enää suositella, sillä salaisen avaimen paljastuttua kaikki salasanat on helppo purkaa.
- Jos P on kaikkien mahdollisten salasanojen (äärellinen) joukko, niin hyökkääjä voisi yrittää etukäteen laskea kaikki arvot $H(p)$, $p \in P$. Salasanoja on kuitenkin niin suuri joukko, ettei tällainen etukäteislaskenta ole mahdollista.

Sateenkaaritaulukko II

- Tiivisteketjut (engl. hash chains) ovat yritys vähentää laskenta- ja tilatarvetta.
- Ideana on käyttää funktiota R, joka muodostaa tiivistearvosta satunnaisen salasanan. Tätä funktiota ja funktiota H käyttäen lasketaan ketjuja lähtien salasanasta p_1:

$$p_1 \xrightarrow{H} h_1 \xrightarrow{R} p_2 \xrightarrow{H} h_2 \xrightarrow{R} \cdots \xrightarrow{H} h_k \xrightarrow{R} p_k,$$

  missä h_i:t ovat tiivistearvoja.
- Nyt lasketaan etukäteen tällaisia ketjuja tiettyyn syvyyteen asti. Ketjuista talletetaan vain ensimmäinen ja viimeinen.
- Jos halutaan päätellä, mitä salasanaa vastaa tiiviste h, lasketaan ketjua aloittamalla R:llä:

$$h \xrightarrow{R} p'_1 \xrightarrow{H} h'_1 \xrightarrow{R} \cdots \xrightarrow{H} h'_{k'} \xrightarrow{R} p'_{k'}.$$

Sateenkaaritaulukko III

- Jos jossain vaiheessa päädytään arvoon, joka on jonkin ketjun loppuarvo, ketjusta saadaan varsinainen salasana. Eli jos $p_k = p'_{k'}$, valitaan ketju, joka päättyy p_k:hon. Lasketaan ketjun alusta kunnes päädytään tiivisteeseen h. Edeltavä salasana on silloin etsitty salasana.
- Aina ei h ole ketjussa. Tällöin lasketaan ketjua h.sta pitemmälle, kunnes seuraava loppukohta löytyy.
- Tällä menetelmällä on se huono puoli, että ketjut voivat yhtyä, jolloin havaittavien salasanojen määrä pienenee. Yhtyminen johtuu siitä, että funktio R voi aiheuttaa yhteentörmäyksiä. Ts. kahdella tiivisteellä h_1, h_2 tapahtuu $R(h_1) = R(h_2)$. Näitä on vaikea havaita automaattisesti riittävän aikaisin. Parannuksena on esitetty sateenkaaritaulukkoja.
- Tällöin otetaan käyttöön useita funktioita R, $R_1, ..., R_k$. Ketjut ovat nyt muotoa

$$p_1 \xrightarrow{H} h_1 \xrightarrow{R_1} p_2 \xrightarrow{H} h_2 \xrightarrow{R_2} \cdots \xrightarrow{H} h_k \xrightarrow{R_k} p_k,$$

Sateenkaaritaulukko IV

- Kun lähdetään laskemaan tiivistearvon h tuottavaa salasanaa, lasketaan
  - $R_k(h)$,
  - $R_{k-1}(H(R_k(h))$,
  - ...
  - $R_1(H(R_2(...(R_k(h)...)$.
- Jos jossain vaiheessa $1, ...k$ päädytään arvoon, joka on valmiiksi lasketussa taulukossa, lähdetään taulukon arvosta liikkeelle laskemaan funktioilla H ja R_i, kunnes päädytään annettuun tiivistearvoon. Tästä ketjusta saadaan salasana.
- Itse asiassa kaikista alle 20 merkin salasanoista on jo laskettu sateenkaaritaulut, joten esimerkiksi SANS suosittelee 20 merkin pituisia salasanoja.
- Suolaus kasvattaa salasanojen pituutta, joten se on toinen keino torjua tämä tekniikka. Tällöin riittää valita lyhyempi salasana.

Tunnus-salasana -lista eli salasanatiedosto I

- Kun käyttäjän oikeuksia tarkistetaan, tietokone vertaa annettua tunnusta ja salasanaa talletettuihin vastaaviin tietoihin. Siten tietokoneessa täytyy olla talletettuna tunnus-salasana -lista.
- Joissakin systeemeissä lista on tiedosto, joka sisältää taulukon. On selvää, että tiedosto täytyy suojata hyvin.
- Suojaus voidaan toteuttaa antamalla tiedoston käyttöoikeus vain käyttöjärjestelmälle.
- Toisaalta KJ:n kaikkien moduulien ei tarvitse päästä salasanatiedoston tietoihin käsiksi. Pahaksi onneksi on käyttöjärjestelmiä, joissa KJ:tä ei ole mitenkään jaettu osiin turvaominaisuuksien perusteella, vaan KJ:n kaikki moduulit pääsevät käsiksi salasanatiedostoon.
- Jos siis hyökkääjä löytää jonkin heikon kohdan käyttöjärjestelmästä, hän pääsee käsiksi kaikkiin tietoihin. Sen tähden on parempi, jos salasanoihin pääsevät käsiksi vain ne moduulit, jotka todella tarvitsevat näitä tietoja.

Tunnus-salasana -lista eli salasanatiedosto II

- Jotta salasanatiedosto ei paljastuisi esimerkiksi varmuuskopioista tai muistivedoksesta, salasanoista on tallettu vain tiivistearvot. Toisinaan salasanat talletetaan salattuina, mutta tämä ei vaikuta enää turvalliselta, koska tällöin salatut salasanat voidaan ainakin periaatteessa purkaa. Tiivistearvojen palauttaminen on vaikeampaa, koska ne on suunniteltu yksisuuntaisiksi.
- Nyt ei ole yhtä suurta pakkoa pitää salasanatiedostoa muiden käyttäjien saavuttamattomissa kuin jos se olisi salaamattomassa muodossa, mutta tietenkin se kannattaa pitää erittäin hyvin suojattuna.
- Järjestelmissä on usein kuitenkin heikkoja kohtia. Esimerkiksi Windows XP:n muistivedoksesta löytyy kaikkien käyttäjien, myös uloskirjautuneiden, salasanat selväkielisenä!

Salasanojen valinta I

Yllä kuvattujen selvittämisyritysten vaikeuttamiseksi salasanojen valinnassa suositellaan seuraavien seikkojen huomioimista:

- Käytä muitakin merkkejä kuin kirjaimia A-Ö.
- Valitse pitkiä salasanoja. Kombinatorinen räjähdys alkaa, kun pituus ylittää 5-6 merkin pituuden, mutta hyvä salasana paljon pitempi. Nykyään suositus on 8 merkkiä.
- Vältä todellisia nimiä ja sanoja.
- Valitse epätodennäköinen salasana. Kehitä jonkinlainen muistisääntö salasanan muistamiseksi.
- Vaihda salasana säännöllisesti. Tällä tavoin estetään kauan kestävät systemaattiset salasanan murtoyritykset.

Salasanojen valinta II

- Älä kirjoita salasanaa muistiin. Tämä sääntö alkaa tosin menettää pätevyyttään, sillä erilaisten tunnusten määrä alkaa olla jo niin suuri, että jonkinlaista kirjanpitoa tarvitaan. Pidä kuitenkin salasanalistat hyvässä tallessa.
- Älä kerro salasanaa kenellekään toiselle.

Tiedostot I

- Tarkastelemme vielä tiedostojen ja käyttäjätunnusten käsittelyä pääsynvalvonnan kannalta. Näissä tarkasteluissa keskitytään UNIX-käyttöjärjestelmään, koska se on jo melko yleinen palvelimissa.
- Paikalliset systeemit identifioivat objekteja antamalla niille nimen. Nimi voi olla tarkoitettu ihmisen, prosessin tai käyttöjärjestelmän ytimen käyttöön. Jokaisella nimellä voi olla eri semantiikka.
- UNIX tarjoaa neljää erilaista tiedostonimityyppiä. Inode identifioi tiedoston yksikäsitteisesti. Se sisältää informaatiota tiedostosta kuten pääsynvalvonta-asetukset ja omistajan sekä määrittelee muistilohkot, josta tiedosto löytyy.
- Prosessit lukevat tiedostoja tiedostokuvaajien (file descriptors) avulla. Kuvaajat sisältävät samaa tietoa kuin inode, mutta sellaisessa muodossa, että prosessit voivat helposti hakea sen, kirjoittaa siihen jne. Kun kuvaaja on kerran luotu, sitä ei voi enää sitoa toiseen tiedostoon.

Tiedostot II

- Prosessit (ja käyttäjät) voivat myös käyttää tiedostonimiä, jotka identifioivat tiedoston kuvaamalla sen aseman tiedostohierarkiassa. UNIXin tiedostonimet voivat olla absoluuttisia polkunimiä, jotka kuvaavat tiedoston aseman juurihakemiston suhteen. Ne voivat olla myös suhteellisia polkunimiä, jotka kuvaavat aseman työhakemiston suhteen.
- Nimien semantiikka eroaa oleellisesti. Kun prosessi tai käyttäjä käsittelee tiedostoa, käyttöjärjestelmän ydin kuvaa tiedostonimen inodeksi käyttäen iteratiivista menetelmää. Se avaa ensin ensimmäisen hakemiston inoden hakupolulla ja etsii sieltä seuraavan alihakemiston inoden. Tämä jatkuu, kunnes halutun tiedoston inode on löytynyt. Kaksi viittausta samaan tiedostoon voivatkin viitata eri tiedostoihin, jos ensimmäisen viittauksen jälkeen tiedosto on poistettu ja uusi, saman niminen tiedosto on luotu tilalle ennen toista viittausta. Tämä voi luoda ongelmia ohjelmien yhteydessä.

Tiedostot III

- Joka tapauksessa kun tietty tiedostokuvaaja on luotu, se viittaa erityiseen objektiin. Riippumatta siitä, miten tiedostoa manipuloidaan, kuvaajaan liittyvä inode pysyy järjestelmässä siihen asti, kunnes kuvaaja on suljettu.

Symboliset linkit I

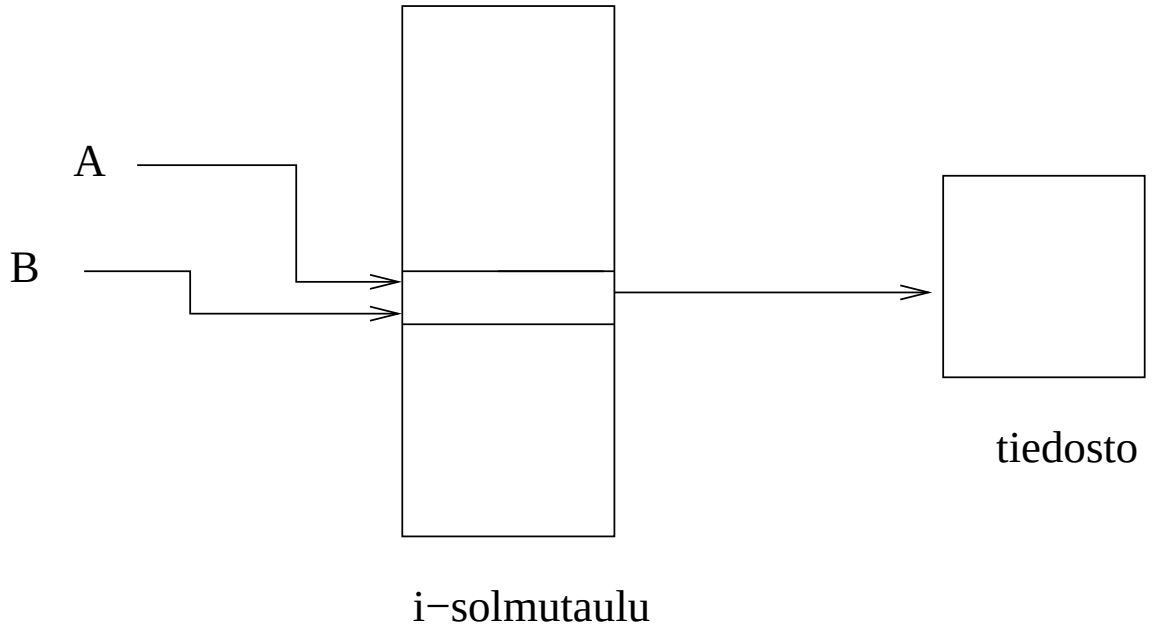
- Symbolinen linkki on erityinen tiedostotyyppi, jossa tiedosto sisältää viitteen toiseen tiedostoon tai hakemistoon absoluuttisen tai suhteellisen polkunimen muodossa. Symboliset linkit esiintyivät ensimmäisen kerran Berkeleyn Unixin versiossa 4.2 BSD. Nykyään niitä tukevat POSIX-standardi, useimmat Unixin kaltaiset KJ:t, Windows Vista ja Windows 7.
- Symbolinen linkki sisältää merkkijonon, jonka KJ tulkitsee poluksi toiseen tiedostoon tai hakemistoon. Jos symbolinen linkki tuhotaan, kohde jää ennalleen. Jos sen sijaan kohde siirretään (move), nimetään uudelleen tai tuhotaan, symbolinen linkki jää ennalleen, mutta viittaa nyt siis tyhjään.
- **Esimerkki.** POSIX:ssa ja Unixin tyyppisissä käyttöjärjestelmissä symbolinen linkki luodaan komennolla

```
ln -s target link\_name
```

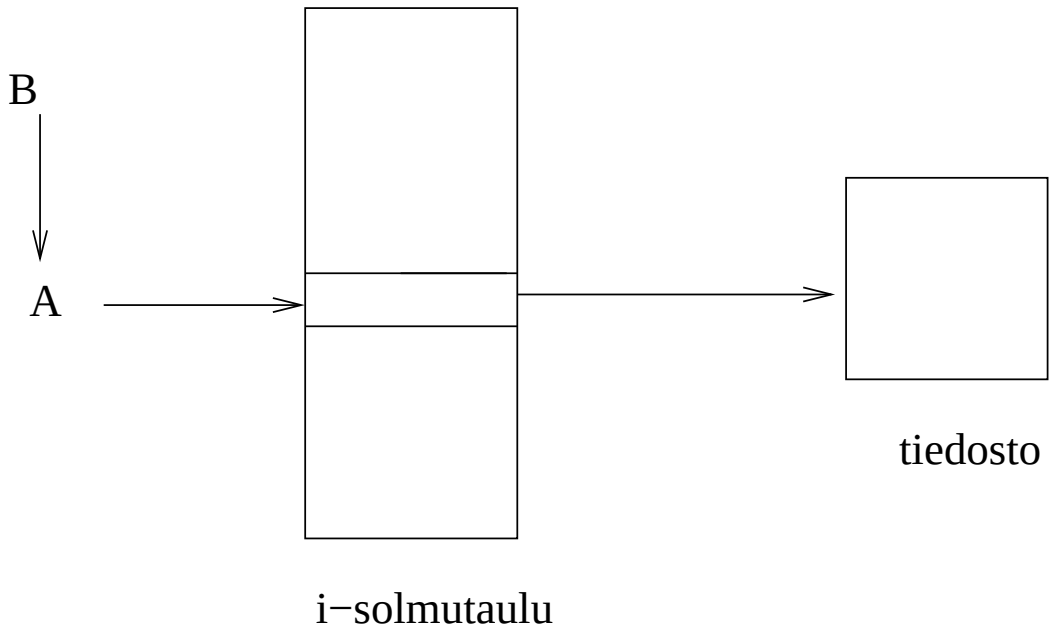
Symboliset linkit II

□

- Symbolisen linkin luomisen jälkeen se toimii kohteen aliaksena eli vaihtoehtoisena nimenä. Komennot, jotka kirjoittavat tai lukevat tiedostoja, kohdistavat toimenpiteensä kohteeseen, kun niille annetaan parametriksi symbolinen linkki. Sen sijaan rm-komento tuhoaa symbolisen linkin, ei kohdetta.
- Unixissa suoraa viittausta tiedostoon kutsutaan kovaksi linkiksi (hard link). Kuvassa 1 on kaksi nimeä, A ja B, jotka viittavat samaan tiedostoon. Nämä viittaukset ovat kovia. Symbolisen linkin rakenteen näyttää sen sijaan kuva 2.



Kuva: Kova linkki



Kuva: Symbolinen linkki

Kovia linkkejä luodaan Unixissa ln-komennolla:
$ ln A B
Voidaan tarkistaa, että molemmat viittaavat samaan i-solmuun:
$ ls -i A B
1321 A
1321 B
Symbolisten linkkien tapauksessa i-solmujen tulisi olla eri:
$ ln -s A B
$ ln -i A B
1321 A
1467 B

Käyttäjien tunnukset I

- Tarkastellaan UNIX-järjestelmää. Käyttäjän identiteettiä esittää kokonaisluku, joka on 0:n ja jonkin suuren luvun (esim. 65 535) välissä. Tätä lukua kutsutaan UID:ksi (user identification number). Lisäksi käyttäjällä voi olla login-nimi. Jokaista login-nimeä vastaa tasan yksi UID, mutta yhdellä UID:lla voi olla monta login-nimeä.
- Kun käyttöjärjestelmän ydin käsittelee käyttäjän identiteettiä, se käyttää UID:ta. Kun taas käyttäjä kirjautuu sisään koneeseen, hän käyttää login-nimeään. Yhdellä käyttäjällä voi olla monta eri identiteettiä. Tyypillisesti identiteetti vastaa jotakin toiminnallisuutta.
- UNIX-versiot käyttävät usean tyyppisiä käyttäjän identiteettejä. Koska käyttäjät käynnistävät prosesseja, nämä eri identiteetit liittyvät prosesseihin.
- Todellinen UID (real UID) on käyttäjän alkuperäinen identiteetti, kun hän kirjautuu koneelle.

Käyttäjien tunnukset II

- Efektiivinen UID (effective UID) on identiteetti, jota käytetään pääsynvalvonnassa. Esimerkiksi jos vain UID 22 voi lukea tiettyä tiedostoa ja prosessin todellinen UID on 22 ja efektiivinen UID 35, niin prosessi ei voi lukea tiedostoa. Jos taas prosessin todellinen UID olisi 35, mutta efektiivinen 22, prosessi voi lukea tiedostoa.
- Systeemiohjelmat *setuid* luovat prosesseja, joiden efektiivinen UID on sama kuin ohjelman omistajan eikä sama kuin ohjelman suorittajan. Tällöin prosessin pääsyoikeudet ovat samat kuin ohjelman omistajan eikä ohjelman suorittajan.
- Monet UNIX-versiot tarjoavat myös talletetun UID:n (saved UID). Aina kun efektiivinen UID vaihtuu, talletetun UID:n arvoksi viedään ennen vaihtoa voimassa ollut efektiivinen UID. Käyttäjä voi käyttää kaikkia kolmea UID:ta. Tämä sallii, että käyttäjälle annetaan juurioikeudet joksikin lyhyeksi ajaksi, jonka jälkeen niistä luovutaan, mutta joihin voidaan palata myöhemmin (tallennetun UID:n avulla).

Käyttäjien tunnukset III

- Traditionaalisesti todellista UID:ta käytettiin prosessin alkuperäisen UID:n jäljittämiseen. Kuitenkin juurioikeuksien haltija voi muuttaa sitä. Jotta turvattaisiin alkuperäisen todellisen UID:n seuranta, monet UNIX-järjestelmät tarjoavat vielä neljättä UID-versiota, audit- tai login-UID. Tämä UID annetaan kirjautumisen yhteydessä, eikä sitä voi vaihtaa. (Kuitenkin jotkut systeemit sallivat juuren muuttaa audit-UID:ta!)

Esimerkki. I

- Salasanatiedosto kuuluu juurioikeuksien haltijalle, joten vain hän voi periaatteessa muokata tiedostoa.
- Kuitenkin tavallinen käyttäjä voi muuttaa salasanaansa. Hän siis muokkaa salasanatiedostoa.
- Käyttäjä käynnistää prosessin, joka muuttaa salasanatiedostoa. Jos salasanatiedoston setuid-bitti on 1, niin tiedoston omistaja (siis juuri) tulee tuon prosessin efektiiviseksi omistajaksi.
- Eli salasanatiedostoa muuttavan prosessin todellinen UID on käyttäjän, mutta efektiivinen UID on juurioikeuksien haltijan. Täten käyttäjä onnistuu muuttamaan salasanatiedostoa.
- Yleensä on tietenkin riskialtista asettaa setuid-bitti juurioikeuksien haltijan mukaan, mutta tässä tapauksessa se on välttämätöntä. Salasanatiedostoa pitää sen vuoksi suojella erityisen hyvin. □

UNIX-käyttäjät kuuluvat *ryhmiin* (group). Siten jokaisella prosessilla on kaksi identiteettiä, käyttäjän identiteetti ja ryhmän identiteetti. Uudemmissa UNIX-järjestelmissä käyttäjä voi kuulua useisiin ryhmiin samalla kertaa. Kirjautuessaan sisään koneelle käyttäjä asetetaan hänelle kuuluviin ryhmiin.

Pääsyoikeudet tiedostoihin I

- UNIX:ssa on kolme tiedosto-operaatiota: lukeminen (read, r), kirjoittaminen (write, w) ja ajaminen (execution, x).
- Näistä lukeminen ja kirjoittaminen ovat selviä, mutta ajaminen on monimutkaisempi operaatio. Tavallisen tiedoston yhteydessä ajaminen tarkoittaa, että joko tiedosto ladataan keskusmuistiin ja sen sisältämät käskyt suoritetaan tai tiedoston sisältämät komentotulkin käskyt luetaan ja suoritetaan. Hakemistotiedoston kohdalla ajaminen tarkoittaa, että hakemisto voidaan käydä läpi tarvittavaa tiedostoa etsittäessä.
- Tiedoston oikeudet saadaan selville ls-komennolla. Komento `ls -l tiedosto` näyttää `tiedoston` ominaisuudet. Ensiksi näkyvät oikeudet muodossa

$$-rw-rw-r--(1)$$

Ensimmäinen viiva ilmaisee tiedoston tyypin:

Pääsyoikeudet tiedostoihin II

-
  - `-` tavallinen tiedosto
  - `d` hakemisto
  - `c` merkkitiedosto (character special file)
  - `b` lohkotiedosto (block special file)
  - `l` symbolinen linkki
  - `p` fifo-tiedosto
- Esimerkin (1) tapauksessa on siis kysymyksessä tavallinen tiedosto. Ensimmäiset kolme koodia tämän jälkeen ilmaisevat omistajan oikeudet.
- Esimerkissä omistaja voi lukea ja kirjoittaa tiedostoon, muttei ajaa tiedostoa. Toiset kolme ilmaisevat ryhmän oikeudet, jotka esimerkissä ovat samat kuin omistajan. Viimeiset kolme merkkiä ilmaisevat muiden oikeudet, jotka tässä tapauksessa rajoittuvat lukemiseen.
- Oikeuksien jälkeen komento ls -l listaa linkkien lukumäärän, käyttäjän, ryhmän, tiedoston koon, muutosten pvm:n ja tiedostonimen.

Pääsyoikeudet tiedostoihin III

- Jos halutaan nähdä hakemiston oikeudet, on käytettävä komentoa `ls -ld`.
- Tiedostojen oikeuksia voidaan muuttaa `chmod`-komennolla. Vain omistaja tai juuri voi käyttää tätä komentoa. Komennon yhteydessä tarvitaan seuraavia parametreja:

	Kuka
u	User
g	Group
o	Other
a	All

	Operaattori
-	poista oikeus
+	lisää oikeus
=	aseta oikeus

Pääsyoikeudet tiedostoihin IV

	Oikeudet
r	Read
w	Write
x	Execute
l	Set locking privilege
s	Set user or group ID mode
t	Set save text mode
u	User's current permissions
g	Group's current permissions
o	Others' current permission

Esimerkkejä I

chmod go-rw: Ryhmältä ja muilta otetaan pois tiedoston luku- ja kirjoitusoikeudet.

chmod a=rw: Kaikki voivat lukea ja kirjoittaa tiedostoon.

chmod g=u: Tiedoston käyttäjän oikeudet siirtyvät myös ryhmälle.

g+x: Ryhmälle tulee oikeus ajaa tiedosto.

Lisäksi on mahdollista käyttää rekursiivista optiota -R. Tällöin komento koskee tiedostoa ja sen alihakemistoja ja -tiedostoja.

Tiedostojen salaus I

UNIX:ssa on helppoa salata tiedoston sisältö:

crypt xyZZy325 chaptn.tex *chaptn.cry*

Komento salaa tiedoston chaptn.tex ja salattu sisältö kirjoitetaan tiedostoon chaptn.cry. Salausavain on käyttäjän valitsema merkkijono xyZZy325. Se on oleellista muistaa, jos alkuperäinen tiedosto tuhotaan! Aina ei ole turvallista kirjoittaa avainta näytölle. Komentoa voidaankin käyttää ilman, että samaan aikaan kirjoitetaan avainta. Tällöin järjestelmä kysyy avainta siten, ettei sen kirjoittaminen näytölle näy. Huomattakoon myös, ettei esimerkin avain ole riittävän pitkä kaikkiin tarkoituksiin. Lisää avainten pituuksista on seuraavissa luvuissa.

Virukset I

Tietokonevirus on ohjelmakoodia, joka pystyy kopioimaan ja levittämään itseään uusiin kohteisiin. Tietokonevirus tarvitsee leviäkseen ja aktivoituakseen aputiedoston samalla tavoin kuin biologinen virus tarvitsee avukseen isäntäsolun.
Virukset voidaan jakaa niiden tartuttamien kohteiden perusteella neljään päätyyppiin:

- tiedostovirukset,
- makrovirukset,
- komentojonovirukset ja
- käynnistyslohkovirukset.

- Virus voi kuulua samalla useampaan kuin yhteen edellä mainituista tyypeistä. Tiedostovirukset tarttuvat suorituskelpoisiin ohjelmatiedostoihin, joista tyypillisimpiä ovat Windows-käyttöjärjestelmässä .com, .pdf ja .scr-päätteiset tiedostot.
- Leviäminen tapahtuu aina kun saastunut ohjelma suoritetaan koneen muistissa. Tiedostovirukset voivat levitä kaikilla tiedonsiirtotavoilla, joilla siirretään ohjelmatiedostoja.
- Makrovirukset on ohjelmoitu toimisto-ohjelmissa käytettävien makrokielten avulla. Makrovirusten leviäminen tapahtuu dokumenttien mukana ja ne saatetaan suorittaa automaattisesti kun dokumentti avataan toimisto-ohjelmassa.
- Makrovirukset voivat levitä käyttöjärjestelmästä riippumatta, mikäli käytössä on sama toimisto-ohjelma. Toimisto-ohjelmiin on lisätty sisäänrakennettuja ominaisuuksia, joilla makrojen tahatonta suorittamista pyritään estämään.

- Komentojonovirukset on tehty käyttäen hyväksi kohdejärjestelmän tarjoamia komentikieliä. Esimerkiksi Windowsin Visual Basic Scripting on ollut suosittu tähän tarkoitukseen. Komentojonoja pystytään luomaan tavallisella tekstieditorilla.
- Käynnistyslohkovirukset tarttuvat tietovälineen, kuten kiintolevyn tai USB-muistin käynnistyslohkoon, josta tietokone etsii käynnistykseen tarvittavia tietoja. Virus voi tarttua ulkoiselta laitteelta kiintolevyn käynnistyslohkolle ja tämän jälkeen saastuttaa tietokoneessa käytettvät muut suojaamattomat muistilaitteet. Käynnistyslohkovirukset leviävät hitaasti, koska ne siirtyvät käytännössä vain muistilaitteelta toiselle. Ne ovatkin nykyään harvinaisia.

Virusten muita luokitteluja I

Viruksia voidaan luokitella myös muiden kriteerioiden mukaan:

- TSR-virus (terminate and stay resident) on sellainen, joka pysyy aktiivisena muistissa sovelluksen päättymisen jälkeen.
- Häivevirus (stealth virus) on virus, joka kätkee saastuneen tiedoston. Nämä virukset sieppaavat tiedostoon liittyvät käyttöjärjestelmäkutsut. Jos kutsun tavoitteena on saada tiedostoattribuutit, tiedoston alkuperäiset attribuutit annetaan. Jos kutsu on tiedoston lukuoperaatio, tiedosto puhdistetaan ensin ja vasta sitten luovutetaan luettavaksi. Mutta jos kutsu on tiedoston suoritus, tiedosto annetaan sellaisenaan.

Virusten muita luokitteluja II

- Salattu virus on sellainen, jonka koodista suurin osa on salakirjoitettu. Vain purkamiseen tarkoitettu koodi on selväkielisenä. Varhaiset virustentorjuntaohjelmistot olivat vaikeuksissa tällaisten virusten kanssa. Ne voidaan paljastaa rakentamalla virtuaalikone, joka suorittaa koodin. Koodi purkaa salauksen ja sen jälkeen virus voidaan tunnistaa.
- Polymorfinen virus muuttaa muotoaan joka kerta, kun se tunkeutuu toiseen ohjelmaan. Tällaisten virusten tuottamista on automatisoitu. Esimerkiksi jo 1992 oli saatavilla Mutation Engine (MtE) ja Trident Polymorphic Engine (TPE).

Täydellisen virustorjunnan mahdottomuus I

- On todistettu, että täydellinen virustorjunta on mahdotonta. Toisin sanoen ei ole olemassa algoritmia, joka löytäisi tai paljastaisi kaikki virukset.
- Sen tähden virustentorjunta perustuu tunnettujen virusten etsimiseen. Tämäkin näyttää toimivan käytännössä hyvin. Lisäksi järjestelmien omat tietoturvaominaisuudet ovat parantuneet.
- Tietokone-lehden numerossa 1-2010 yritettiin tartuttaa tahallaan syksyllä 2009 liikkuneita viruksia Windows Vista -koneeseen. Käyttäjällä eli tässä tapauksessa tartuttajalla oli vain perusoikeudet.
- Tartuttaminen osoittautui yllättävän vaikeaksi, sillä Vistan oma Defender huomasi ja esti virusten toimintaa.

Täydellisen virustorjunnan mahdottomuus II

- Ja vaikka tartunta olisikin tapahtunut, Windows olisi helppo puhdistaa: kone käynnistetään admin-tunnuksella ja käyttäjähakemiston työtiedostot kopioidaan turvaan. Sen jälkeen käyttäjätili poistetaan ja tarvittaessa perustetaan uudelleen. Kone on jälleen puhdas. Ongelmia syntyy vasta, jos virus on päässyt koneeseen admin-oikeuksilla.

Madot I

- Madot ovat haittaohjelmia, jotka kykenevät leviämään itsenäisesti ilman aputiedostoa. Madot voivat levitä nopeasti esimerkiksi sähköpostin avulla.
- Sähköpostimadot voivat olla liitetiedostoina tai osana itse viestiä. Liitetiedostoina leviävät madot vaativat yleensä aktivoituakseen, että käyttäjä avaa tiedoston. Eräät madot aktivoituvat tietyillä sähköpostiohjelmilla jo viestin esikatselutilassa. Aktivoiduttuaan sähköpostimato etsii kohdekoneesta sähköpostiosoitteita, joihin se voi lähettää itsensä edelleen.
- Verkkomadot leviävät tietokoneverkossa tarvitsematta sähköpostin tai tietokoneen käyttäjän apua. Ne käyttävät hyväkseen reaaliaikaista verkkoyhteyttä koneiden välillä. Verkkomatojen leviämiselle otollisia ovat jatkuvasti verkossa kiinni olevat, puutteellisesti ylläpidetyt palvelimet tai kotitietokoneet, joissa on madon leviämisen mahdollistavia paikaamattomia tietoturvahaavoittuvuuksia.

Madot II

- Eräs laji matoja ovat botnet-verkkoja rakentavat madot. Botnet-verkot ovat valloitetuista koneista koostuvia kokonaisuuksia, joita käytetään hajautettuihin palvelunestohyökkäyksiin, roskapostitukseen, verkkourkintaan, identitettivarkauksiin sekä lukuisiin muihin rikollisiin tarkoituksiin.
- Botnet-verkkojen rakenne on melko samanlaista kuin vertaisverkkojen. Storm-verkko on tunnetuin botnet-verkko.
- Erään tutkimuksen mukaan Storm oli vastuussa jopa yli viidenneksestä kaikesta roskapostista vuoden 2008 ensimmäisellä neljänneksellä. On myös arvoitu, että Storm koostui noin kahdesta miljoonasta koneesta vuonna 2008. Verkko katosi kokonaan syyskuussa 2008, mutta heräsi eloon uudistuneena Waledac-verkkona vuoden 2008 lopussa.
- Conficker (myös Downup, Downaup, Kido) on yksi suurimmista botnet-verkoista, joita on löydetty. Tartunnan saaneita koneita arvioitiin olevan 9-15 miljoonaa tammikuussa 2009.

Madot III

- Conficker kulkee dynaamisen linkkikirjaston (DLL) mukana, joten se ei voi asentua koneeseen ilman apua. Alun perin Conficker rupesi leviämään Windowsin eri versioissa olevan MS08-067 tietoturva-aukon kautta.
- On myös kaksi muuta tapaa, joilla se pyrkii leviämään. Ensimmäinen tapa on hyödyntää verkkoon kytkettyjen Windows-käyttöjärjestelmien avoimia levyjakoja. Toinen tapa levitä tapahtuu USB-muistitikkujen avulla. Kun muistitikku työnnetään koneen USB-porttiin, käynnistyy automaattisesti autorun-prosessi, joka lataa ja suorittaa haittakoodin.
- Confickerin käyttämä vertaisverkkoteknologia on osittain tuntematonta. Tämä johtuu siitä, että Confickerin ohjelmoijat ovat tarkoituksellisesti kirjoittaneet haittakoodista vaikeaselkoista.

Madot IV

- Conficker pyrkii myös sammuttamaan kaikki Windowsin tietoturvaominaisuudet. Esimerkiksi palautuspisteet (Windows Restore Points), Windows Update sekä sen yhteydessä hyödynnettävä tietoturvapäivitysten latausohjelma BITS (Background Intelligent Transfer Service) poistetaan käytöstä.
- Samalla Conficker lisää Windowsin rekisteriin rekisteriavaimet, jotka sallivat Confickerille tarpeellisen liikenteen palomuurin läpi. Confickerissä on myös sisäänrakennettu lista eri virustorjuntaohjelmistojen käynnistystiedostoista. Mikäli tartunnan saaneesta koneesta löytyy jokin listan ohjelmista, Conficker sammuttaa ja poistaa kyseisen ohjelman käytöstä.

Madot V

- Huolimatta siitä, että Conficker on ollut olemassa jo pitkään, ei ole keksitty tehokasta tapaa tuhota sitä keskitetysti. Myöskään viitteitä siitä, kuka Confickerin on alun perin luonut, ei ole saatu. Helmikuussa 2009 Microsoft yhteistyössä muutamien muiden teollisuuden alan yritysten kanssa ilmoitti antavansa $250 000 palkkion syyllisen jäljille johtavasta vihjeestä. Black Hat -hakkeriryhmä on vihjannut, että Confickerin rakentajat olisivat ukrainalaisia.

Troijalaiset I

- Troijalaiset ovat ohjelmia, jotka tekevät ohjelman käyttäjältä salassa jotain arvaamatonta. Troijalaiset leviävät jollain tapaa houkuttelevan tai hyödyllisen ohjelman mukana tai ovat osa itse ohjelmaa sisältäen dokumentoimattomia toimintoja. Troijalaisissa itsessään ei ole leviämismekanismia, vaan niitä käytetään tyypillisesti muun haittaohjelman haittakuormana.
- Troijalaiset voivat avata kohdekoneelle takaportin, jonka kautta luvaton tunkeutuja voi päästä murtautumaan tietokoneelle ja etähallitsemaan sitä jopa organisaation palomuurin läpi.
- Murrettua konetta voidaan käyttää roskapostitukseen, palvelunestohyökkäyksiin tai tietomurtoihin. Troijalaiset voivat myös kerätä ja lähettää verkossa eteenpäin salasanoja, sähköpostiosoitteita, näppäinpainalluksia tai tietoa kyseisestä tietokoneesta tai käyttäjän toimista, kuten vierailuista verkkosivuilla.

Vakoilu- ja mainosohjelmat I

- Jotkut ilmaiset ja jopa maksullisetkin hyöty- tai apuohjelmat sisältävät vakoilukomponentteja, jotka keräävät ja lähettävät tietoa koneen käytöstä eteenpäin.
- Kerättävät tiedot voivat olla tietoja käyttäjän vierailemista www-sivuista tai jopa verkkopalvelujen salasanoja. Ohjelmat voivat pakottaa selaimen aloitussivun omalle sivulleen siten, että aloitussivua on vaikea muuttaa takaisin halautuksi sivuksi tai ne saattavat avata lukuisia mainosikkunoita selainta käytettäessä.
- Näistä vakoiluohjelmista saatetaan kertoa ohjelmaa ladattaessa tai ohjelman lisenssisopimuksessa. Koska joillekin vakoiluohjelmille saadaan asennettaessa käyttäjän suostumus, näitä ohjelmia ei välttämättä tunnisteta virustorjunta-ohjelmalla, vaan ne vaativat tähän tarkoitukseen kehitetyn oman torjuntaohjelmansa. Osa vakoiluohjelmista asentuu työasemalle salaa käyttäjän tietämättä ja lupia kysymättä.

Vakoilu- ja mainosohjelmat II

- Nykyään yleisiä ovat myös tietoturvaohjelmat, jotka ovat joko tehottomia tai suorastaan haitallisia. Osa ohjelmista on jopa suomenkielisiä ja niitä mainostetaan hyvämaineisilla www-sivuilla. Ne väittävät löytäneensä ongelmia, joiden korjaaminen vaatii maksullisen version asentamista. Muutama "turvaohjelma jopa lataa koneelle haittaohjelmia, jotka se sitten lupaa poistaa. Älä käytä koneen siivoamiseen muita kuin turvalliseksi todettuja apuohjelmia.

Huijausviestit, ketjukirjeet, pilailuohjelmat ja sosiaalinen hyväksikäyttö I

- Huijausviestit eivät ole ohjelmia, vaan sähköpostiviestejä, jotka leviävät hyväuskoisten käyttäjien lähettäminä. Näissä viesteissä saatetaan varoittaa vaarallisesta viruksesta ja kehottaa käyttäjää lähettämään viesti eteenpäin mahdollisimman monelle.
- Huijausviestien, samoin kuin ketjukirjeiden, välittämiseen kuluu sekä lähettäjän että vastaanottajan työaikaa. Pahimmillaan huijausviestit erehdyttävät käyttäjän toimimaan virheellisesti, kuten poistamaan käyttöjärjestelmälle tärkeitä tiedostoja viruksina. Huijausviesteihin kuuluvat myös viestit, joissa pyydetään lähettään esimerkiksi pankkitunnuksia.
- Internetistä on saatavana paljon erilaisia vitsi- ja pilailuohjelmia, joilla säikytellään tavallisia käyttäjiä. Ohjelmat voivat antaa kummallisia virheilmoituksia, olla alustavinaan käyttäjän kiintolevyä tai tuhoavinaan tiedostoja.

Huijausviestit, ketjukirjeet, pilailuohjelmat ja sosiaalinen hyväksikäyttö II

- Tietokoneen hiiren tai näytön toimintaa voidaan manipuloida siten, että ne vaikuttavat olevan rikki. Nämä ohjelmat eivät yleensä ole vihamielisiä, mutta niiden mukana voi levitä vaarallisia haittaohjelmia ja ne voivat kuluttaa henkilöresursseja laitteiden tarkastukseen.
- Monet menestykselliset hyökkäykset ovat perustuneet ihmisten harhauttamiseen muutenkin kuin tietokoneen käytössä. Esimerkiksi luottamuksellisia tietoja, vaikkapa salasanoja, voi kysellä johtotason henkilöksi tekeytyvä hyökkääjä.
- Salasanoja voi urkkia hiiviskelemällä toimistoissa ja tutkimalla muistilappuja. Myös kiikarointi vastapäisestä rakennuksesta voi paljastaa salasanoja ja käyttäjätunnuksia.
- Troijalaisia voidaan myös levittää kylvämällä saastutettuja muistitikkuja organisaation parkkipaikalle ja toivomalla, että joku henkilökunnasta olisi utelias ja tutkisi tikun sisältöä.

Haittaohjelmien torjunta I

VAHTI-sarjan julkaisussa 3/2004 on laajasti tarkasteltu haittaohjelmilta suojautumista. Seuraavassa muutamia tärkeimpiä kohtia:

- Virusten torjunta vaatii käytäntöjä monella eri tapaa:
  - Organisaatiolla tulisi olla koulutusohjelma, jossa valistetaan viruksista ja muista haittaohjelmista.
  - Säännölliset tiedotteet virusongelmista ovat tarpeellisia.
  - Älä koskaan siirrä koneeseesi tiedostoja epävarmoista lähteistä, ellei virustorjunta ole ajan tasalla.
  - Testaa uudet ohjelmat tai avoimet dokumentit erillisessä koneessa ja siirrä vasta sitten tuotantokäyttöön.
  - Huolehdi, etteivät asiaankuulumattomat pääse asentamaan koneille haittaohjelmia, erityisesti Troijan hevosia.
  - Käytä käyttöjärjestelmiä, joiden sisäänkirjautumiskäytännöt ja todentamiset ovat turvallisia.
- Haittaohjelmilta suojautuminen:

Haittaohjelmien torjunta II

- Virustentorjunta on tehokkainta, kun se tapahtuu automaattisesti. Toisaalta automaattiset menetelmät eivät huomaa uusia haittaohjelmia, joten torjuntaohjelmistojen päivitysten on oltava tiheää.
- Työasemia voidaan monitoroida haittaohjelmien aiheuttamien systeemifunktioiden aktiviteetin havaitsemiseksi.
- Jos käytetään vain kaupallisia ohjelmistoja, jotka asennetaan esim. CD:ltä, vältytään perinteisiltä viruksilta.
- Kannettavan työaseman käynnistys on aina suojattava salasanalla. Se voidaan toteuttaa työasemassa olevalla nk. turvapaneelilla, kiintolevyn salakirjoitusohjelmaan kuuluvalla käynnistyssalasanalla tai BIOS-asetuksista löytyvällä koneen käynnistyssalasanalla. Vasta salasanan syöttäminen käynnistää tietokoneen käyttöjärjestelmän ja antaa käyttäjälle kirjautumisikkunan.
- Jos kannettavan kiintolevyä ei ole salakirjoitettu, on se suojattava BIO-asetuksista löytyvällä kiintolevyn suojaussalasanalla. Menettely estää levyn käytön toisessa samanlaisessa koneessa.

Haittaohjelmien torjunta III

- Työaseman käynnistys on sallittava vain kiintolevyltä. Jos tähän turva-asetukseen on jostain syystä tehtävä kevennyksiä, suositellaan käynnistysjärjestykseksi: kiintolevy, cd-rom, muut siirrettävät mediat (esim. USB).
- Kaikki BIOS-tason muutokset on suojattava muutoksilta salasanalla, jota ei saa antaa loppukäyttäjälle,
- Käytä Microsoftin Macro Virus Protection -ohjelmaa kaikissa Microsoftin sovelluksissa äläkä koskaan aja dokumentin makroja, jos niiden toiminta ei ole tiedossa.
- Päivitä käyttöjärjestelmät säännöllisesti.
- Käytä palomuureja.