LUENTO 11

Käyttöjärjestelmät II

TIETOTURVA

Ch 16 [Stal 05]

Suojaus (security)

- **Salakirjoitus** App 16A [Stal 05]
- **Uhat** Ch 16 [Stal 05]
  - turvallisuusuhat
  - pahantahtoiset ohjelmat
  - tunkeutujat
- **Suojaus**
  - suojausympäristöt **Luento 11**
  - virustorjunta
  - luotettu järjestelmä
- **UNIX: suojaus (Ch 10.7 [Tane 01])** **Luento 12**
- **W2K: suojaus (Ch 11.8 [Tane 01])**

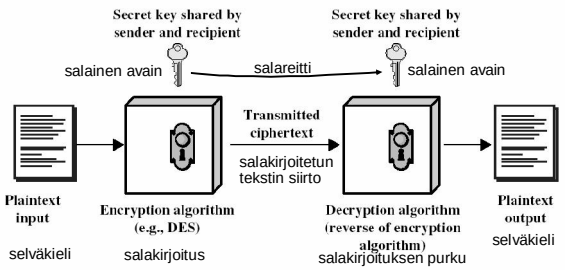
Käyttöjärjestelmät II

Salakirjoitus

Appendix 16A [Stal 05]

Perinteinen, symmetrinen salaus

Sama avain molemmilla!



(Fig 16.14 [Stal 05])

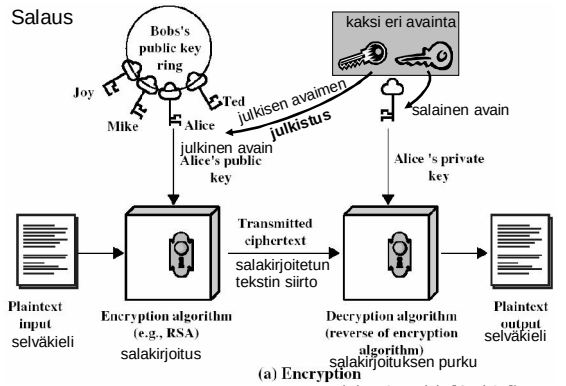
DES: Data Encryption Standard

- **Symmetrinen: sama avain molemmissa**
  - perustuu Lucifer algoritmiin (Horst Feistel, IBM), 1977
- **Avain 56 bittiä (plus 8 pariteettibittiä)**
  - kryptaus 64 bitin lohkoissa
- **Iteroi lohkolle 16 kertaa bittioperaatioita**
  - eri kierroksella alkuperäisen avaimen eri 48 bittiä avaimena
  - kierrosten välillä
    - sekoita bittien järjestystä
    - korvaa bittikuvioita toisilla
- **Pystytty murtamaan erikoislaitteistolla**
  - brute-force, muutama tunti
- **Triple DEA**
  - käyttää kolmea DES-avainta (168b + 24 pariteettibittiä)
  - kolme peräkkäistä DES'iä (encrypt-decrypt-encrypt)

AES – Advanced Encryption Standard

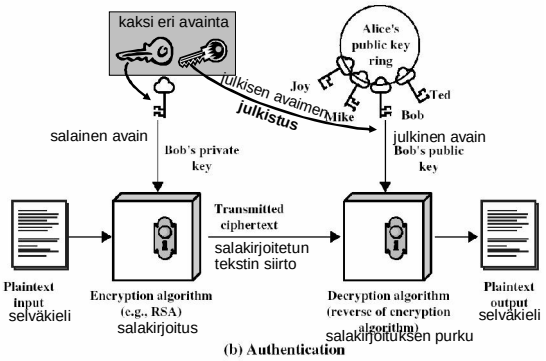
- **DES seuraaja, Rijndael lohkosalaaja** **click**
  - Joan Daemen & Vincent Rijmen (Belgia), 2000
- **eri kokoisia avaimia: 128b, 192b, 256b**
- **lohkon koko 128b**
- **eri moodeja**
  - nopeampi vai suojatumpi?
- **piirteitä**
  - "alkuluku" polynomit (irreducible polynomials)
  - polynomien kertolasku
  - alkuperäistä avainta laajennetaan ja siitä johdetaan dynaamisesti vaihtuvat avaintilat, joista johdetaan kussakin vaiheessa käytettävä avain

Julkisen avaimen salakirjoitusmenetelmä



(Fig 16.15 (a) [Stal 05])

Autentikointi julkisen avaimen menetelmällä



(Fig 16.15 (b) [Stal 05])

Julkisen avaimen salakirjoitus

- **Perustuu matemaattisiin funktioihin, ei bittitason operaatioihin**
  - Diffie Hellman 1976
  - moduloaritmetiikka, laskennallisesti erittäin vaikeaa ilman avaimia
  - perustuu hyvin pitkiin (300 numeroa?) alkulukuihin ja aikaavievään polynomiaaliseen (*siis ei NP-täydelliseen*) tekijöihinjako-ongelmaan
- **Asymmetrinen: kaksi avainta**
  - julkinen publ: kryptaa tällä
  - salainen secr: pura tällä plain = Decrypt$_{\mathbf{secr}}$ (Crypt$_{\mathbf{publ}}$ (plain))
- **Voi tehdä myös toisin päin**
  - salainen secr: kryptaa tällä
  - julkinen publ: pura tällä plain = Decrypt$_{\mathbf{publ}}$ (Crypt$_{\mathbf{secr}}$ (plain))
- **RSA-algoritmi** Lisää tietoja Tietoturvakurssilla
  - Rivest, Shamir, Adleman 1977
  - samoja avainpareja voi käyttää kummin päin vain!
  - käytetään nyt melkein kaikkialla avainten jakeluun

Käyttöjärjestelmät II

Turvallisuusuhat

Turvallisuustarpeet

Fig 16.1 [Stal 05]

- **Suojattu pääsy tietoon** protection
  - kellä pääsy mihin **tietoon muistissa**
- **Kontrolloitu järjestelmän käyttö** user authentication
  - kuka käyttää järjestelmää eli **käyttäjän tunnistus**
- **Suojattu tiedon siirto järjestelmien välillä**
  - **verkkoyhteyksien** suojaus network security
- **Suojattu tiedostojen käyttö** file security
  - kellä pääsy mihin tietoon **tiedostojärjestelmässä**

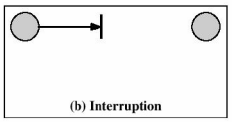
Turvallisuusvaatimuksia

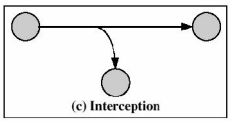
- **Luottamuksellisuus (confidentiality, secrecy)**
  - tietoa saa lukea vain ne, joilla siihen lupa
  - ei välttämättä tietoa edes tiedon olemassaolosta
- **Eheys, koskemattomuus (integrity)**
  - tietoa saa tuottaa/muuttaa vain ne, joilla siihen lupa
- **Saatavuus (availability)**
  - tieto oltava saatavilla käyttötarkoituksen mukaisesti
- **Oikeaksi todentaminen (authenticity)**
  - tiedon käyttäjä pystyttävä todentamaan siksi, joka väittää olevansa
    - kuka on? mitä tietää? mitä omistaa?

Uhkia

DoS – denial of service

- **Häirintä, pysäyttäminen, "ilkivalta" (interruption)**
  - tiedon tuhoaminen tai saatavuuden estäminen
  - esim. kovalevy tuhottu, tietoliikennelinja katkaistu, tiedostojärjestelmä kytketty toiminnasta



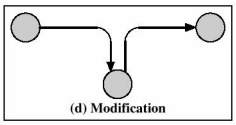


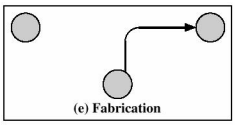
(Fig 16.2 [Stal 05])

- **Sieppaus (interception)**
  - luottamuksellisen liikenteen salakuuntelu
  - kopiointi

Uhkia

- **Muuntelu, "peukalointi" (modification)**
  - tiedon korvaaminen muutetulla tiedolla
  - esim. ohjelman toimintaa / datatiedostoa muutettu, sanomien väärentäminen





(Fig 16.2 [Stal 05])

- **Valmistus, "satuilu" (fabrication)**
  - järjestelmän tietojen muuttaminen, jotta saadaan haluttu (luvaton) toiminta
  - esim. tekaistut tietueet, tunnukset, sanomat

Suojattavaa ja uhkia

- **Laitteisto** Tbl 16.1 [Stal 05]
  - haavoittuvin osa tietokonejärjestelmää
    - saatavuus, luottamuksellisuus, eheys, oikeaksi todentaminen
  - vaikea käyttää automaattisia turvajärjestelyjä
    - lukitut konehuoneet, piilotetut kaapelistot
    - pääsynvalvonta
- **Ohjelmisto**
  - haavoitettavana saatavuus: tuhottu, muutettu
  - tietoturva ylläpitohenkilökunnan vastuulla
  - osa automatisoitavissa
    - varmuuskopiot
    - tarkistussummat

Suojattavaa ja uhkia

Tbl 16.1 [Stal 05]

- **Data**
  - haavoitettavana
    - saatavuus
    - luottamuksellisuus
    - eheys
  - ylläpito käyttäjien vastuulla
    - oltava käyttöoikeuksia
  - tärkeä tieto voi olla analysoitavissa muita tietoja yhdistelemällä, vaikkei itse tietoon pääse suoraan käsiksi

Passiiviset hyökkäykset (kuuntelu)

Fig 16.3 [Stal 05]

- **Luottamuksellisuus rikkoontuu, eheys ei rikkoudu**
- **Tietoliikenneyhteydet, -verkko**
  - salakuuntelu, tarkkailu, vuotaminen julkisuuteen (release of contents)
  - puhelut, sähköposti, tiedostojensiirto
  - salaus, salakirjoitus
    - silti analysoitavissa (traffic analysis)

Aktiiviset hyökkäykset

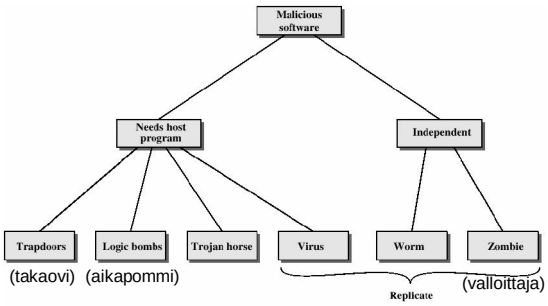
- **Tiedon eheys rikkoontuu**
- **Tietoliikenneyhteydet, -verkko**
  - lähettäjä teeskentelee olevansa joku muu (masquerade) Fig 16.4 (a) [Stal 05]
  - virheellinen toisto (replay) Fig 16.4 (b) [Stal 05]
  - viivyttäminen, muuttaminen, uudelleenjärjestely (modification of msg contents) Fig 16.4 (c) [Stal 05]
  - käytön esto (DoS = denial of service) Fig 16.4 (d) [Stal 05]
    - ylikuormitus, yhteyksien sabotointi
    - yritetään havaita ja toipua nopeasti

Käyttöjärjestelmät II

Pahantahtoiset ohjelmat

(Malicious software)

Luokittelua



(Fig 16.8 [Stal 05])

Takaovi (salaovi, trap door)

Fig 16.8 [Stal 05]

- **Ohjelmoijan / testaajan oikopolku sopivaan kohtaan koodia**
  - ko. haaraan pääsee ei-julkisella näppäilyllä
  - välttää kaikenmaailman hidastavat alustukset ja salasanat Fig 9-10 [Tane 01]
  - esim. takaa eteenpäin pääsyn, vaikka testaus muuten jumittaisi
    - laillinen käyttö, joka ”unohtunut” koodiin
- **Mukamas ”tietoturvapäivitys”, mutta sisältääkin heikennystä / takaoven lisäämisen...**
  - päivitys vain luotettavalta taholta


```
while (TRUE) {
    printf("login: ");
    get_string(name);
    disable_echoing( );
    printf("password: ");
    get_string(password);
    enable_echoing( );
    v = check_validity(name, password);
    if (v) break;
}
execute_shell(name);
```

(a)

```
while (TRUE) {
    printf("login: ");
    get_string(name);
    disable_echoing( );
    printf("password: ");
    get_string(password);
    enable_echoing( );
    v = check_validity(name, password);
    if (v || strcmp(name, "zzzzz") == 0) break;
}
execute_shell(name);
```

(b)

Fig. 9-10. (a) Normal code. (b) Code with a trap door inserted. [Tane 01]

Fig. 9-9. (a) Correct login screen. (b) Phony login screen.

Looginen pommi (aikapommi, logic bomb)

Fig 16.8 [Stal 05]

- **Ohjelmassa koodinpätkä, joka suoritetaan, kun tietyt ehdot täyttyvät**
  - joku tiedosto olemassa / puuttuu
  - tietty viikonpäivä
  - tietty käyttäjä
  - tietylle käyttäjälle ei maksettu palkkaa 2 kk:een
- **Kiristys ... vai ”konsulttipalkkio”**
  - poista pommi
  - laita uusi, parempi tilalle?

Troijan hevonen

- **Hyödyllinen (tai siltä näyttävä) ohjelma, joka ajettaessa tekee muutakin kuin leipätyötään**
  - hävittää tiedostoja
  - antaa muille oikeuksia
- **Houkuttele laillinen käyttäjä ajamaan ohjelmaa**
  - hänen oikeuksillaan pahanteko onnistuu
    - anna käyttäjälle Pahis tai käyttäjän Pahis ohjelmalle P super-user oikeudet
- **Ei näy välttämättä lähdekoodissa**
  - kääntäjää, kirjastoa peukaloitu?
  - muutos vain binäärissä?
- **Login spoofing** Fig 9-9 [Tane 01]
  - CTR-ALT-DEL toimiva lääke

Puskurin ylivuoto (buffer overflow)

- **Koodissa vakiopituinen taulukko**
- **Indeksiä tai merkkijonon pituutta ei tarkisteta**
- **Talletus muuttaa tietoa muualla**
  - esim. aliohjelmasta paluuosoite

Fig 9-11 [Tane 01]

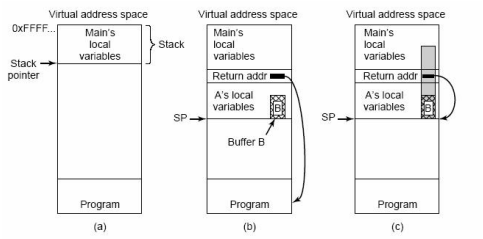


Fig. 9-11. (a) Situation when the main program is running. (b) After the procedure *A* has been called. (c) Buffer overflow shown in gray.

[Tane 01]

Virus

- **Upotettu "kohdetiedostoon" (Troijan hevonen)**
  - peli, työkalu, kuva, artikkeli
  - **dropper** – viruksen upotustyökalu
  - kohdekäyttäjä kopioi sen itselleen
- **Odottaa, kunnes kohdetiedosto aktivoidaan**
  - käynnistyy aina tai joskus (looginen pommi)
- **Saastuta kone pysyvämmin**
  - upota virus muihin tiedostoihin
- **Suorita payload**
  - harmiton viesti
  - tuhoisa toiminta (esim. tuhoa BIOS)

Viruksen elinkaari

- **Lepovaihe (dormant)**
  - se vaan olla möllöttää
  - katselee almanakkaa, tarkkailee levyn täyttöastetta...
- **Lisääntymisvaihe (propagation)**
  - kloonautuu muihin ohjelmiin ja tietyille levyalueille
- **Laukaisuvaihe (triggering)**
  - herkistyy toimimaan
  - almanakka oikealla sivulla, kopioitunut riittävän monta kertaa, tms.
- **Suoritusvaihe (execution)**
  - tekee ilkeämieliset temppunsa

Mato

- **Käyttää verkkoa levitäkseen koneesta toiseen**
  - leviää itsestään ilman käyttäjän myötävaikutusta
  - harmiton, tuhoisa tai tuottava payload
- **Sähköposti**
  - mato postittaa itseään osoitelistasta löytyville
  - mato postittaa harkittua roskapostia osoitelistasta löytyville
    - roskapostiin reagoidaan → **madon tekijä saa rahaa**
- **Etäkomentojen suorittaminen**
  - mato suorituttaa itsensä löytämissään etäkoneissa
- **Etäistuntojen hyödyntäminen**
  - mato ottaa istunnon etäkoneeseen ja käyttää normaaleja komentoja leviämiseen
- **Viisas mato ei leviä jo madolliseen koneeseen**
- **Viisas mato piiloutuu normaalinnäköiseksi (nimiseksi) prosessiksi**

Zombie valloittaa koneen

- **Asettuu uhriksi valittuihin koneisiin ja laukaisee sieltä käsin 'ikävät' toiminnot**
- **Ei laukea polun alkupään koneissa**
  - syntypaikan jäljittäminen vaikeaa
- **Kun laukeaa, monistuu eksponentiaalisesti**
  - valloittaa CPU-kapasiteetin
  - täyttää muistin
  - täyttää levytilan
- **Distributed DoS – Distributed Denial of Service**
  - zombiet pommittavat uhria esim. SYN-sanomilla
    - kolmivaiheinen kättely pulmallinen
  - saturoi web-palvelimen tuhansilta koneilta

Virustyyppejä

- **Loinen (parasitic)**
  - kun saastunut ohjelma ajetaan, tutkii levyn ja tarttuu muihin ohjelmiin
- **Muistiresidentti**
  - hengailee keskusmuistissa muistiresidentin ohjelman osana
    - ei löydy levyskannauksella
  - tarttuu kaikkiin suoritettaviin ohjelmiin
- **Käynnistyslohkovirus (boot sector)**
  - tarttuu järjestelmän käynnistyslohkoon
  - pääsee muistiin heti, kun järjestelmä käynnistetään

Virustyyppejä

- **Stealth, ”salamyhkäinen”**
  - yrittää piiloutua virustorjuntaohjelmilta
    - saastunut ohjelman ei näytä muuttuneen
    - sieppaa esim. levypyynnön ja näyttää epäilijälle alkuperäisen tiedoston
- **Polymorphinen** Fig 9-17 [Tane 01]
  - yrittää piiloutua virustorjuntaohjelmilta
    - muuttaa itseään jokaisella käynnistyskerralla
    - salakirjoittaa / purkaa itseään eri avaimin
  - muuttunut virus toiminnaltaan aiemman kaltainen, mutta bittikuviot (sormenjäljet) erilaisia
  - mutation engine sober.f click

(a)	(b)	(c)	(d)	(e)
MOV A,R1	MOV A,R1	MOV A,R1	MOV A,R1	MOV A,R1
ADD B,R1	NOP	ADD #0,R1	OR R1,R1	TST R1
ADD C,R1	ADD B,R1	ADD B,R1	ADD B,R1	ADD C,R1
SUB #4,R1	NOP	OR R1,R1	MOV R1,R5	MOV R1,R5
MOV R1,X	ADD C,R1	ADD C,R1	ADD C,R1	ADD B,R1
	NOP	SHL #0,R1	SHL R1,0	CMP R2,R5
	SUB #4,R1	SUB #4,R1	SUB #4,R1	SUB #4,R1
	NOP	JMP .+1	ADD R5,R5	JMP .+1
	MOV R1,X	MOV R1,X	MOV R1,X	MOV R1,X
			MOV R5,Y	MOV R5,Y

Fig. 9-17. Examples of a polymorphic virus.

[Tane 01]

Virustyyppejä

LoveLetter click

- **Makrovirukset**
  - MS-Word ja MS-Excel suorittavat makrokomentoja käynnistyessään (oletus)
    - automaattisen toiminnon voi kääntää pois
  - sotkevat / hävittävät dokumentteja
  - kopioituvat dokumentteihin
  - leviää helposti lähettämällä asiakirja sähköpostitse
    - "*I love you*" viidessä tunnissa maailman ympäri
    - [*"Slammer"* mato löysi lähes kaikki haavoittuvat koneet maailmalla 10 minuutissa (25.1.2003)]
  - vuosi 2001 ennätyksellisen vilkas virusvuosi
    - n. 100 tartuntaa 1000 tietokonetta kohden

F-Secure 2005: “*Vuoden toisella puoliskolla virusten määrän kasvu jatkui hälyttävällä tahdilla. Määrä nousi vuoden loppuun mennessä ennennäkemättömälle tasolle, 110.000 viruksesta 150.000 virukseen.”*

F-Secure 2005 click

Käyttöjärjestelmät II

Tunkeutujat

Tunkeutujat (intruders)

- **Kasvava ongelma**
  - vieraan tunnuksen käyttö masquerader
  - oman tunnuksen väärinkäyttö misfeasor
  - salattu käyttö clandestine user
    - hommaa root-oikeudet, piilota jäljet
- **Asiakas/palvelija ympäristö**
  - ei enää keskuskoneympäristössä
  - verkon kautta tulevat yhteydenotot
- **Krakkerit saavat oppia ja välineitä muilta**
  - se verkko...

Miten sisään yritetään?

- **Arvaa / kokeile salasanoja**
  - standarditunnuksia + oletussalasana / ei salasanaa
  - järjestelmällisesti lyhyitä salasanoja
  - käytä apuna järjestelmän sanastoa tai jotain muuta valmista "top100"-listaa
  - käytä käyttäjään liittyviä tietoja
    - puh., nimet, seinällä olevat sanat, ...
- **Käytä Troijan hevosta**
  - hyötyohjelma, joka myös kokoaa käyttäjätietoa
- **Salakuuntele verkkoa**
  - tunnus/salasana voi olla selväkielisenä

Identiteetin kalastelu (phishing)

- **Identiteettivarkaus**
- **Huijaus**
  - ei virus, ei mato
- **Uskottava väärennetty sähköposti**
  - sisältää linkin väärennetylle kotisivulle
  - käyttäjä validoi itsensä ja “päivittää” tietonsa
  - validointitietojen avulla hyökkääjällä käyttäjän identiteettitiedot, tunnukset, salasanat, jne

Phishing click

Käyttöjärjestelmät II

Suojautuminen (protection)

eli

Miten uhkia torjutaan?

Suojaustasoja

- **Ei suojausta, mutta**
  - haavoittuvat prosessit ajetaan erillään muista
- **Eristäminen**
  - kukin prosessi toimii itsenäisesti
  - ei yhteiskäyttöä tai kommunikointia muiden kanssa
- **Kaikki tai ei mitään julkiseksi**
  - omistaja antaa resurssin julkiseen jakeluun tai pitää yksityisenä
- **Rajoitettu (kiinteä) yhteiskäyttö**
  - käyttöoikeus tietyillä käyttäjillä tiettyihin resursseihin
  - KJ tarkistaa käyttöoikeuden resurssia käytettäessä
    - ainakin silloin, kun käyttö alkaa

Suojaustasoja (jatkuu)

- **Dynaaminen käyttöoikeuksien hallinta**
  - (omistaja) voi muuttaa
- **Käyttöoikeuksien/tavan rajoittaminen**
  - käyttöoikeuden lisäksi voidaan määritellä myös käyttötapa
    - esim. käyttäjä saa tilastollisia tunnuslukuja, mutta ei näe yksittäisiä arvoja
    - tilastolliset tunnusluvut saa vain jos
      - populaatio > 3?
      - populaatio > 10?
      - populaatio > 100?

Muistinsuojaus

- **Moniajojärjestelmä**
  - muistissa useiden käyttäjien prosesseja
  - saavat viitata vain hallitusti muistiin
    - eivät saa luvatta viitata toisten data-alueelle
    - eivät saa vaihtaa toisten funktioita toisiksi
- **Toteutus: virtuaalimuisti**
  - osittain laitteistolla, osittain KJ:ssa
- **Yhteiskäyttö**
  - sivu/segmentti esiintyy useassa sivu/segmenttitaulussa
  - toteutus helpompi segmentoinnissa
    - oma segmentti yhteiskäyttöalueelle

Käyttäjän tunnistus

- **Käyttöoikeus vain rekisteröidyillä käyttäjillä**
  - käyttäjätunnus ja salasana
- **Vieraille voi olla guest / visitor tunnuksia**
  - rajoitetut oikeudet
- **Rekisteröinnin jälkeen tunnus mukana käyttäjän prosessien PCB:ssä**
  - oikeuksien tarkistaminen
  - prosessien oikeudet perustuvat käyttäjän identiteettiin
    - yleensä tämä ei riitä!

Käyttöoikeudet

- **Kuka saa käyttää ja mitä?**
- **Peruslähtökohta**
  - käyttäjän tunnistus (**user**)
  - toimialue (suojausympäristö, **domain**)
    - mitä resursseja ja miten tähän suojausympäristöön kuuluva käyttäjä tai muu subjekti (**subject**, **principal**) saa käyttää
- **Pääsymatriisi**
  - rivi: toimialue (domain)
  - sarake: resurssi, objekti (object)
  - alkio: toimialueen subjektin käyttöoikeus resurssiin
    - domain on myös objekti!

Fig 16.5 (a) [Stal 05]

Fig 9-24 [Tane 01]

	Object										
Domain	File1	File2	File3	File4	File5	File6	Printer1	Plotter2	Domain1	Domain2	Domain3
1	Read	Read Write								Enter	
2			Read	Read Write Execute	Read Write		Write				
3						Read Write Execute	Write	Write			

Fig. 9-24. A protection matrix with domains as objects. [Tane 01]

Käyttöoikeudet

- **Käyttöoikeudet käyttäjän yhteydessä (mitä käytetään?)**
  - käyttäjäprofiili
  - valtakirjalistat (capability lists), väärentämättömät
- **Käyttöoikeudet kohteen yhteydessä (kuka käyttää?)**
  - kohde: data, ohjelma
  - pääsylistat (ACL, access control list)
  - yleisempi, helpompi toteuttaa
    - tieto vain yhdessä kohdassa
- **Molemmat**
  - vain pääsymatriisin ei-tyhjät alkiot
- **KJ tarkistaa oikeudet käytön yhteydessä**
  - esim. vertaa PCB:ssä olevaa uid+gid paria tiedoston attribuutteihin talletettuun uid+gid pariin

Fig 16.5 (c) [Stal 05]

Fig 16.5 (b) [Stal 05]

Käyttöoikeuspolitiikat

- **DAC – discretionary access control**
  - tiedon omistaja päättää, kuka siihen pääsee käsiksi ja miten
  - käyttäjä voi dynaamisesti muuttaa omistamiensa tietojen (tiedostojen) pääsyoikeuksia
    - vaikutus alkaa ... milloin?
  - normaali yksityiskäyttö
- **MAC – mandatory access control**
  - keskitetty politiikka, joka oletusarvoisesti määrittelee kuka pääsee käsiksi mihin tietoon ja miten
  - käyttäjä ei voi muuttaa pääsyoikeuksia
  - luokitellun tiedon käyttöympäristöt

harkinnan-varainen

poista lukuoikeus?

pakollinen

Hyvä salasana

- **Koneen generoima**
  - vaikeampi arvata
  - vaikeampi muistaa → paperille?
- **Käyttäjän valitsema**
  - hylkää liian lyhyet ja helpohkosti arvattavat
  - järjestelmä voi laajentaa, salaisella 'suolalla'
    - sama salasana ei näytä samanlaiselta kryptattuna
    - salasana käytännössä pitenee
    - brute-force hyökkäys hidastuu (suola salainen tai ainakin kaikilla erilainen)
- **Järjestelmä yrittää itse aktiivisesti arvata salasanan**
  - vaihdettava, jos osoittautui liian helpoksi
  - hakkeri voi tehdä tätä kopioimallaan passwd-tiedostolla
    - login-yritysten rajoittaminen ei hidasteena
    - "suolaus" on hyvä hidaste
    - passwd-tiedosto suojatulle muistialueelle olisi hyvä idea

Fig 16.6 [Stal 05]

Tunkeilijan huomaaminen

- **Tunkeilijaa vaikeaa estää vaikeuttamatta samalla normaalia käyttöä**
- **Tunnuksen käyttöprofiili muuttuu yllättäen**
  - aamu-uninen Arskako töissä kello 5?
  - eikö Villen pitäisi olla lomalla?
- **Tilastollinen poikkeama**
  - kerää perustietoa laillisten käyttäjien tyypillisestä kuormasta tietyn jakson ajan
  - vertaa uutta jaksoa perusjaksoon
  - mikä on normaalia? mikä poikkeavaa?
- **Mitä on automatisoitavissa?**

Tunkeilijan huomaaminen

- **Sääntöpohjainen eksperttijärjestelmä**
  - perussäännöstö normaalille käytölle
    - eri yrityksissä/kulttuureissa erilaista
  - mikä on normaalia? mikä poikkeavaa?
- **KJ tarjoaa perusvälineet**
  - kirjaa tietoa käyttäjän login-ajoista, CPU-ajasta jne.
  - loki- ja historiatiedostot
- **Omat räätälöinnit parempia**
  - tunkeilija tuntee perus-KJ:n
- **Erillinen audit-järjestelmä**
  - kerää tunkeilijan huomaamisessa tarvittavaa tietoa
- **Ansat**
  - *user* guest, *password* guest ⟶ login OK, soita poliisille
- **Kuka on tunkeilija? Kuka tuntee nykyisen lain?**

Käyttöjärjestelmät II

Virustorjunta

Virustorjunta

- **Havaitse - tunnista**
  - virustorjuntaohjelmalla
  - vertaile ohjelmien pituuksia ja tarkistussummia
  - etsi viruksen sormenjäljet
  - muistiresidentti virusskanneri huomaa, kun virus yrittää tehdä työtänsä
- **Hävitä**
  - käynnistä järjestelmä puhtaalta kirjoitussuojatulta levykkeeltä / CD:ltä (vältä käynnistyslohkovirukset)
  - aja virustorjuntaohjelma
    - ajantasainen virustietokanta – ikä max 2 tuntia?
  - joskus ohjelmia asennettava uudestaan

Generic Decryption Scanner

- **Polymorfiset virusten etsintään**
- **Tutki ohjelma ensin GD-skannerilla**
  - CPU-emulaattori
  - viruksien ”sormenjälkien” tunnistin
  - ohjausmoduuli
- **Emulaattori <u>tulkitsee ohjelmaa</u> käsky kerrallaan**
- ***”Sormenjälkitunnistus”* selaa koodin aika-ajoin**
  - jos virus löytyy, ei koodia päästetä todelliseen suoritukseen
- **Ongelma:**
  - kauanko ajettava, ennen kuin virus purettu?
  - ei saa hidastaa tarpeettomasti ohjelmien käynnistystä

Digital Immune System (IBM)

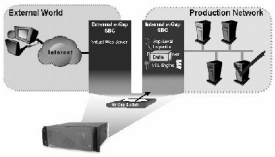
Fig 16.9 [Stal 05]

- **Kussakin koneessa ’viritelty’ virustorjunta**
  - tunnetut: normaali virustorjunta
  - uudet: etsi epäilyttäviä piirteitä (heuristiikka)
- **Lähetä epäilyttävät ohjelmat tarkemmin tutkittavaksi immuuniin koneeseen**
  - emulointi, monitorointi
  - jos virus, kirjaa sormenjäljet, kehitä lääkkeet
- **Tunnisteet ja lääkkeet automaattisesti muille koneille**
  - nopeammin kuin virus itse leviäisi

Firewalls

(palomuuri)

- **Packet-filtering firewall** (pakettifiltteri)
  - look at packet header info
  - accept/reject packet based on rules
    - accept from this domain, reject for know bad guy
- **Application-level gateway** (sovellustason yhdyskäytävä)
  - look at data in packet
  - scan for viruses, etc
- **Air-gap technology** (ilmarako)
  - create ”empty space” firewall
  - two servers, memory bank in middle
    - memory bank has no OS,
    - memory bank connected to max one server at a time
  - E.g., e-Gap Systems (Whale Communications)



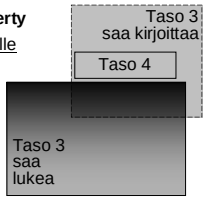
http://www.whalecommunications.com

Käyttöjärjestelmät II

Luotettu järjestelmä
(trusted system)

Multilevel Security

- **Tieto luokitellaan tärkeyden mukaan**
  - unclassified, confidential, secret, top secret
- **Käyttäjälle määritelty ’luottamustaso’**
- **No-read-up eli simple security property**
  - kukin saa nähdä vain omalle tai sen alapuoliselle tasolle luokiteltua tietoa
- **No-write-down eli *-property eli star-property**
  - tietoa saa tuottaa vain omalle tai ylemmälle tasolle
  - tietoa saa ’vuotaa’ alemmille tasoille vain, jos siihen on saatu erikseen lupa
- **Esimerkki MAC-suojauksesta**
  - Mandatory Access Control



Reference Monitor

Fig 16.10 [Stal 05]

- **Säätelee/laillistaa käyttäjien (subject) pääsyä kohteisiin (object) kumpiinkin liitettyjen attribuuttien mukaisesti**
- **Security kernel database**
  - käyttäjien pääsyoikeudet, liikkumavara
  - kohteiden käyttöoikeudet, luokittelutasot
- **Pakottaa noudattamaan säännöstöä**
  - MAC: no-read-up, no-write-down
- **Toteutus laitteistossa ja KJ:ssa**
  - pelkkä ohjelmallinen toteutus liian hidasta
  - ”viittausvalvoja”, tarkkain (reference monitor)
    - jatkuvaan seurantaan tarkoitettu laite

Reference Monitor -politiikka

- **Ominaisuudet**
  - täydellinen sääntöjen noudattaminen (**mediation**)
    - säännöt tarkistetaan jokaisella viitteellä, ei pelkästään esim. tiedostoa avattaessa
  - eristäminen (**isolation**)
    - sekä monitor että tietokanta suojattu täysin luvattomilta muuttajilta
  - verifioitavuus (**verifiability**)
    - monitorin **oikeellisuus pitää pystyä osoittamaan** matemaattisesti
- **Jos verifioitavissa, sallitaan käyttää termiä luotettu järjestelmä (trusted system)**
  - aika paljon vaadittu …

Reference Monitor

- **Audit-file**
  - tärkeät tietoturvaan liittyvät tapahtumat rekisteröidään
    - muutokset tietokantaan eli oikeuksiin
    - login ja logout -tapahtumat
    - tietoturvan rikkomisyritykset
- **Tutkittavissa jälkikäteen**
  - jos rikosta ei voi estää, niin toivottavasti …
    - se voidaan edes myöhemmin havaita ja
    - pahatekijä saadaan kiinni
      - kiinnijäämisen pelko on viisauden alku

Audit: tarkastus, arviointi, tilintarkastus

Esimerkki

- **Troijan hevonen ja normaalit käyttöoikeudet**
  - Alice vokottelee Bobin ajamaan ohjelmansa
    - lukee Bobin yksityistä tietoa
    - luo tiedoston, jonka Alice voi lukea, mutta Bob ei

Fig 16.11 (a, b) [Stal 05]

- **Troijan hevonen ja luotettu järjestelmä**
  - Tasot: arkaluonteinen (harmaa), julkinen (valkea)
  - Bob tasolla, jolla oikeus arkaluonteiseen tietoon
  - Alice tasolla julkinen
  - Kun Bob suorittaa Alicen ohjelman, se ei voi kirjoittaa Bobin tasoa alemmalle tasolle (*-property)

Fig 16.11 (c, d) [Stal 05]

Æ **suojaustaso tässä tärkeämpi kuin käyttöoikeudet !**

OpenBSD

(www.openbsd.org)

- **Unix project to create most secure UNIX**
  - try to be the #1 most secure operating system
- **Default setup**
  - very secure system
  - proactive security, integrated cryptography
  - every source file analyzed by core team
  - continuous development
  - no break in
  - have almost no external communications at all
- **Enable better, user-friendly communication**
  - not so easy, need good understanding of system
  - what if maintenance by inexperienced administrators?

Kertauskysymyksiä

- **Miten tiedostojen käyttöoikeudet tavallisimmin määritellään?**
- **Miten ja milloin oikeudet tarkistetaan?**
- **Miten valtakirjat ja pääsylistat eroavat toisistaan?**
- **Reference Monitorin idea**