LUENTO 21

Tietoturva

Ch 16 [Stal 05]

1

Suojaus (security)

- Salakirjoitus
- Uhat
  - turvallisuusuhat
  - pahantahtoiset ohjelmat
  - tunkeutujat
- Suojaus
  - suojausympäristöt
  - virustorjunta
  - luotettu järjestelmä

App 16A [Stal 05]

Ch 16 [Stal 05]

2

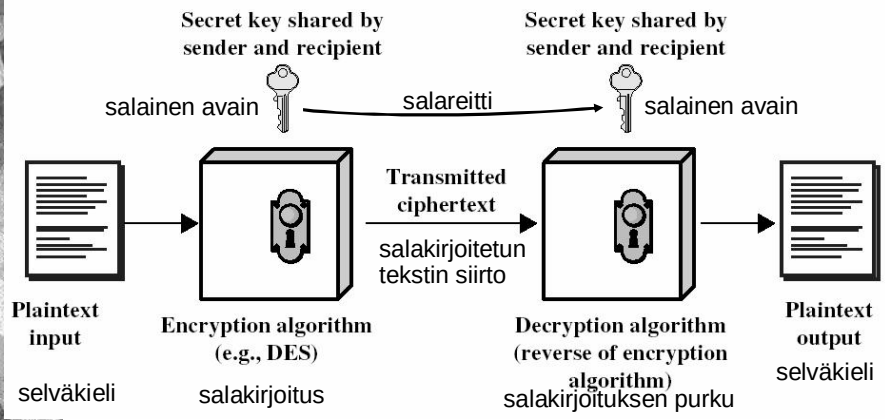
Salakirjoitus

Appendix 16A [Stal 05]

3

Perinteinen, symmetrinen salaus

Sama avain molemmilla!



(Fig 16.14 [Stal 05])

4

DES: Data Encryption Standard

- Symmetrinen: sama avain molemmissa
  - perustuu Lucifer algoritmiin (Horst Feistel, IBM), 1977
- Avain 56 bittiä (plus 8 pariteettibittiä)
  - salaus 64 bitin lohkoissa
- Iteroi lohkolle 16 kertaa bittioperaatioita
  - eri kierroksella alkuperäisen avaimen eri 48 bittiä avaimena
  - kierrosten välillä
    - sekoita bittien järjestystä ja 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)

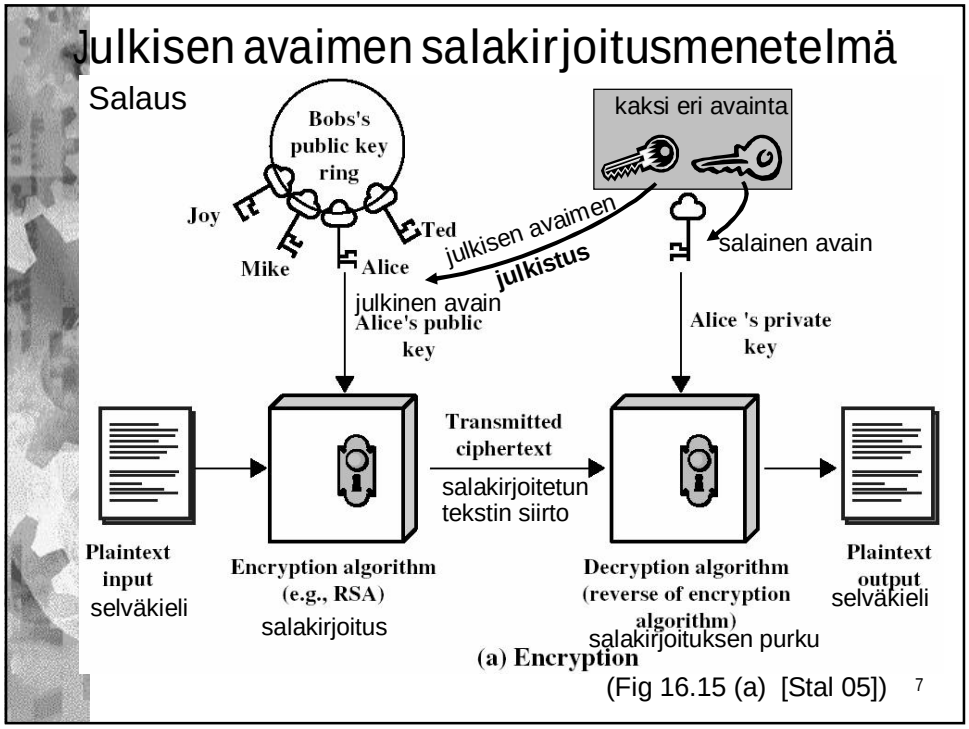
5

AES – Advanced Encryption Standard

- DES seuraaja, Rijndael lohkosalaaja
  - 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

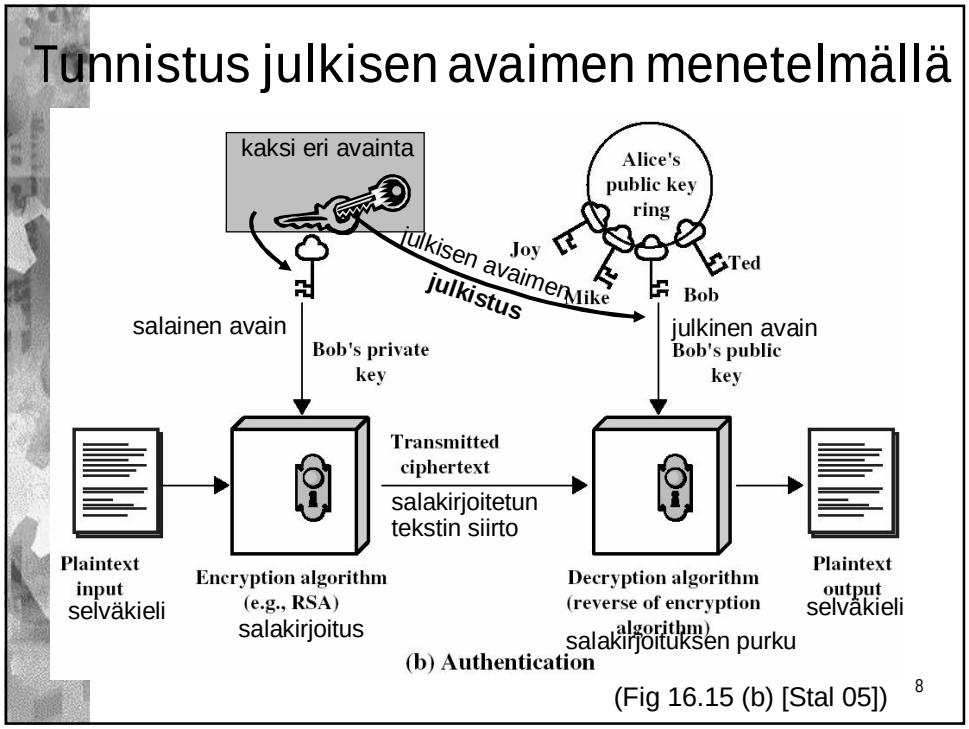
6

Julkisen avaimen salakirjoitusmenetelmä



(Fig 16.15 (a) [Stal 05]) 7

Tunnistus julkisen avaimen menetelmällä



(Fig 16.15 (b) [Stal 05]) 8

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 aikaa vievään polynomiaaliseen (*siis ei NP-täydelliseen*) tekijöihinjako-ongelmaan
- Asymmetrinen: kaksi avainta
  - julkinen publ: kryptaa tällä
  - salainen secr: pura tällä $plain = Decrypt_{secr}(Crypt_{publ} (plain))$
- Voi tehdä myös toisin päin
  - salainen secr: kryptaa tällä
  - julkinen publ: pura tällä $plain = Decrypt_{publ}(Crypt_{secr} (plain))$
- RSA-algoritmi
  - Rivest, Shamir, Adleman 1977
  - samoja avainpareja voi käyttää kummin päin vain!
  - käytetään nyt melkein kaikkialla avainten jakeluun

Lisää tietoja Tietoturvakurssilla

9

Turvallisuusuhat

10

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 tiedonsiirto järjestelmien välillä
  - verkkoyhteyksien suojaus — network security
- Suojattu tiedostojen käyttö — file security
  - kellä pääsy mihin tietoon tiedostojärjestelmässä

11

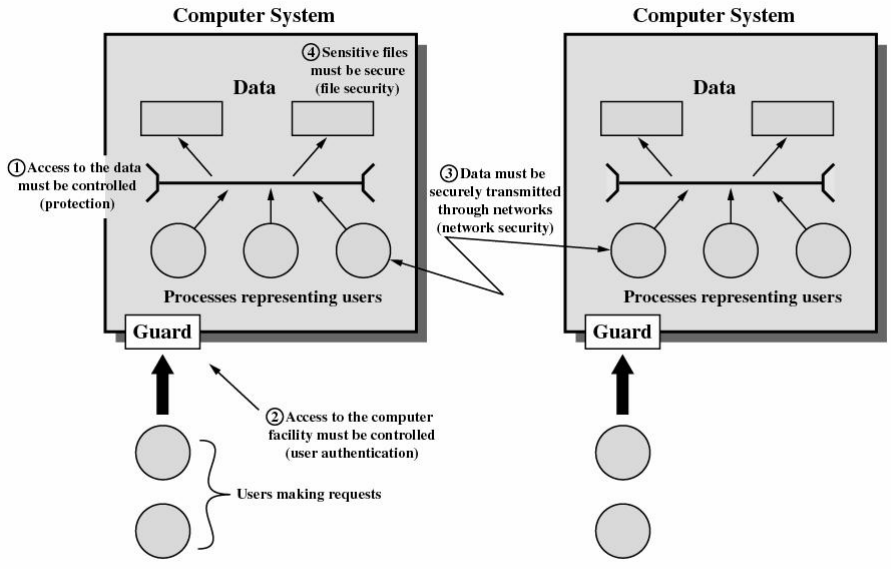


Figure 16.1 Scope of System Security [MAEK87]

Turvallisuusvaatimuksia

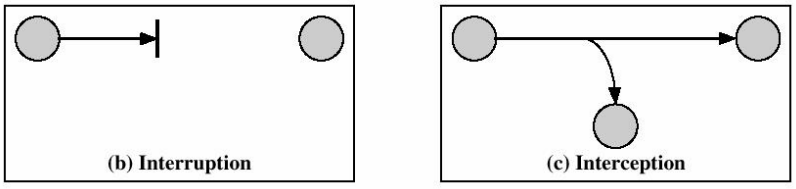
- Luottamuksellisuus (confidentiality, secrecy)
  - tietoa saa lukea vain ne, joilla siihen lupa
  - ei välttämättä tietoa edes tiedon olemassaolosta
- Eheys, koskemattomuus (integrity)
  - tietoa saa tuottaa/muuttaa vain ne, joilla siihen lupa
- Käytettävyys / Saatavuus (availability)
  - tieto oltava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

13

Uhkia

DoS – denial of service

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



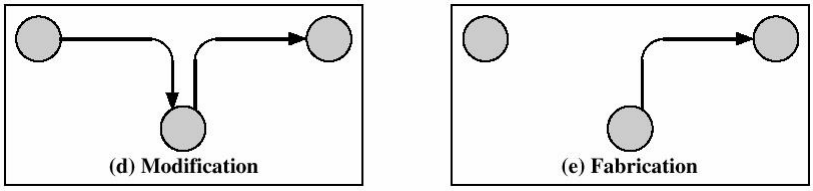
(Fig 16.2 [Stal 05])

- Sieppaus (interception)
  - luottamuksellisen liikenteen salakuuntelu
  - kopiointi

14

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

15

Suojattavaa ja uhkia

Tbl 16.1 [Stal 05]

- Laitteisto
  - 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

16

Suojattavaa ja uhkia

Tbl 16.1 [Stal 05]

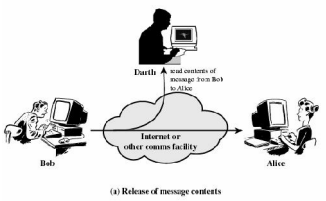
- Data
  - haavoitettavana
    - saatavuus
    - luottamuksellisuus
    - eheys
  - ylläpito käyttäjien vastuulla
    - oltava käyttöoikeuksia
  - tärkeä tieto voi olla analysoitavissa muita tietoja yhdistelemällä, vaikkei itse tietoon pääse suoraan käsiksi

17

Passiiviset hyökkäykset (kuuntelu, nuuskinta)

Fig 16.3 [Stal 05]

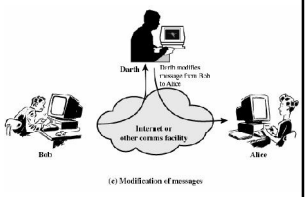
- Luottamuksellisuus rikkoontuu, eheys ei rikkoudu
- Tietoliikenneyhteydet, -verkko
  - salakuuntelu, tarkkailu, vuotaminen julkisuuteen (release of contents)
  - puhelut, sähköposti, tiedostojensiirto
  - salaus, salakirjoitus
    - silti analysoitavissa (traffic analysis)



(a) Release of message contents

18

Aktiiviset hyökkäykset



- Tiedon eheys rikkoontuu
- Tietoliikenneyhteydet, -verkko
  - lähettäjä teeskentelee olevansa joku muu (masquerade)
  - virheellinen toisto (replay)
  - viivyttäminen, muuttaminen, uudelleenjärjestely (modification of msg contents)
  - käytön esto (DoS = denial of service)
    - ylikuormitus, yhteyksien sabotointi
    - yritetään havaita ja toipua nopeasti

Fig 16.4 (a) [Stal 05]

Fig 16.4 (b) [Stal 05]

Fig 16.4 (c) [Stal 05]

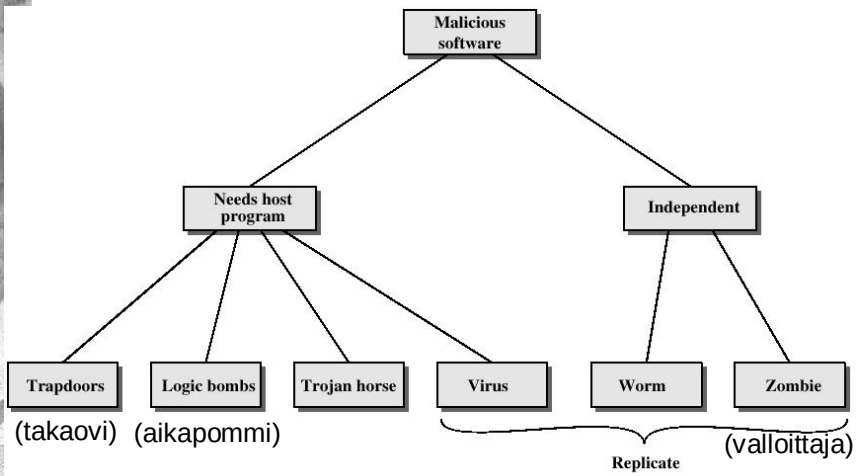
Fig 16.4 (d) [Stal 05]

19

Haittaohjelmat (Malicious software)

20

Luokittelua



(Fig 16.8 [Stal 05])

21

Takaovi (salaovi, trap door)

- Ohjelmoijan / testaajan oikopolku sopivaan kohtaan koodia
  - ko. haaraan pääsee ei-julkisella näppäilyllä
  - välttää kaikenmaailman hidastavat alustukset ja salasanat
  - esim. takaa eteenpäin pääsyn, vaikka testaus muuten jumittaisi
    - laillinen käyttö, joka ”unohtunut” koodiin

Fig 9-10 [Tane 01]

- Mukamas ”tietoturvapäivitys”, mutta sisältääkin heikennystä / takaoven lisäämisen...
  - päivitys vain luotettavalta taholta

22

```
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]

(a) (b)

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

23

Looginen pommi (aikapommi, logic bomb)

- 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?

24

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?

Fig 9-9 [Tane 01]

25

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]

- Perinteinen hyökkäysreitti
- Huonosti tehty ohjelma

26

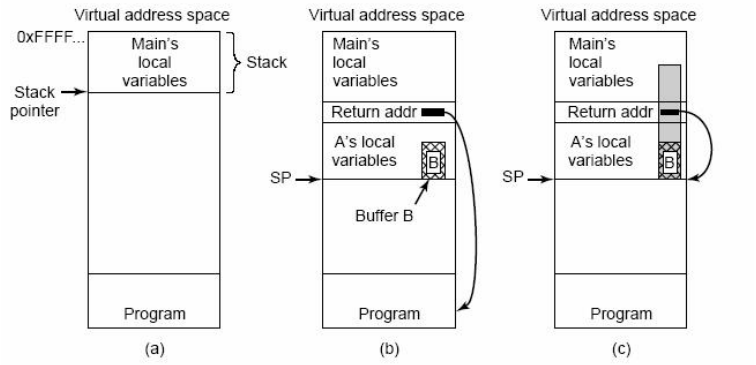


Fig. 9-11. (a) Situation when the main program is running. (b) After the procedure *A* has been called. (c) Buffer overflow shown in gray.

[Tane 01]

27

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)

28

Viruksen elinkaari

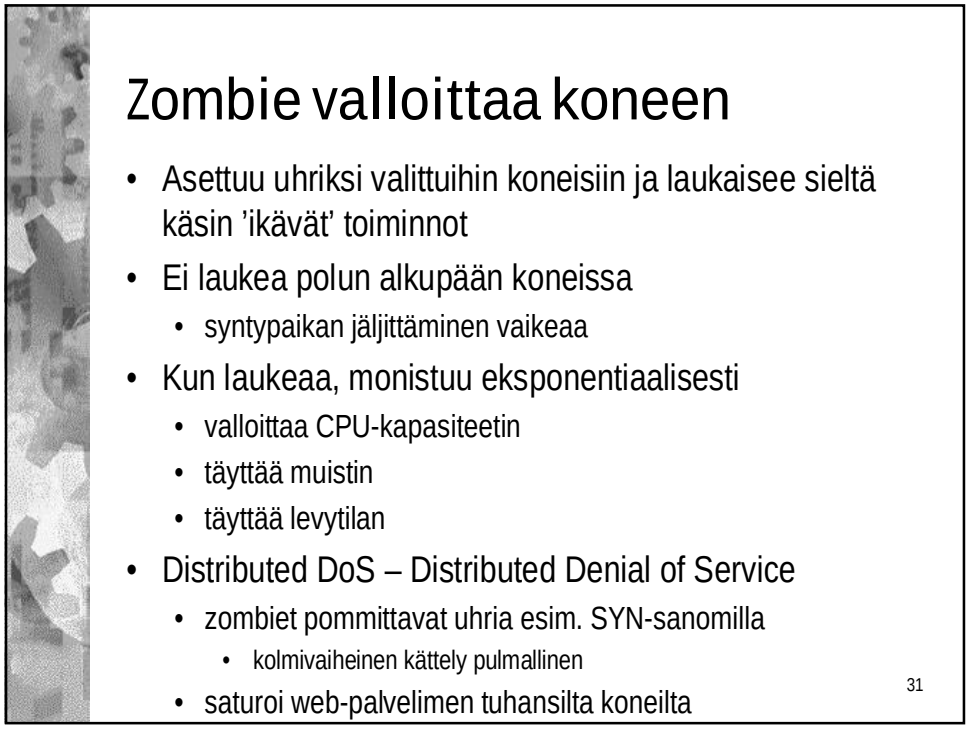
- Lepovaihe (dormant)
  - se vaan olla möllöttää
  - katselee almanakkaa, tarkkailee levyn täyttöastetta...
- Lisääntymisvaihe (propagation)
  - kloonautuu muihin ohjelmiin ja tietyille levyalueille
- Laukaisuvaihe (triggering)
  - herkistyy toimimaan
  - almanakka oikealla sivulla, kopioitunut riittävän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

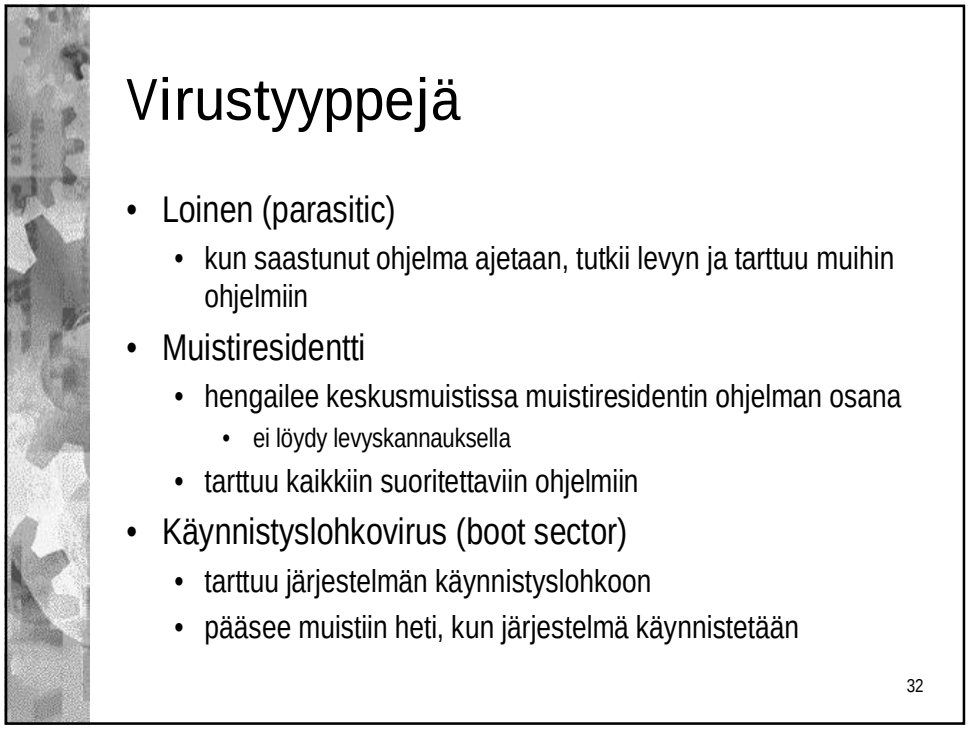
29

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

30


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
31


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
32

Virustyyppejä

- Stealth, ”salamyhkäinen”
  - yrittää piiloutua virustorjuntaohjelmilta
    - saastunut ohjelman ei näytä muuttuneen
    - sieppaa esim. levypyynnön ja näyttää epäilijälle alkuperäisen tiedoston

Fig 9-17 [Tane 01]

- Polymorfinen
  - yrittää piiloutua virustorjuntaohjelmilta
    - muuttaa itseään jokaisella käynnistyskerralla
    - salakirjoittaa / purkaa itseään eri avaimin
  - muuttunut virus toiminnaltaan aiemman kaltainen, mutta bittikuviot (sormenjäljet) erilaisia

sober.f

  - mutation engine

33

(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]

34

Virustyyppejä

LoveLetter

- 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

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.*"

35

Tunkeutujat

36

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...

37

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

38

Identiteetin kalastelu (phishing)

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

Phishing filter – Tietokalasteluntorjuntasuodatin (Microsoft IE:n termistöä)

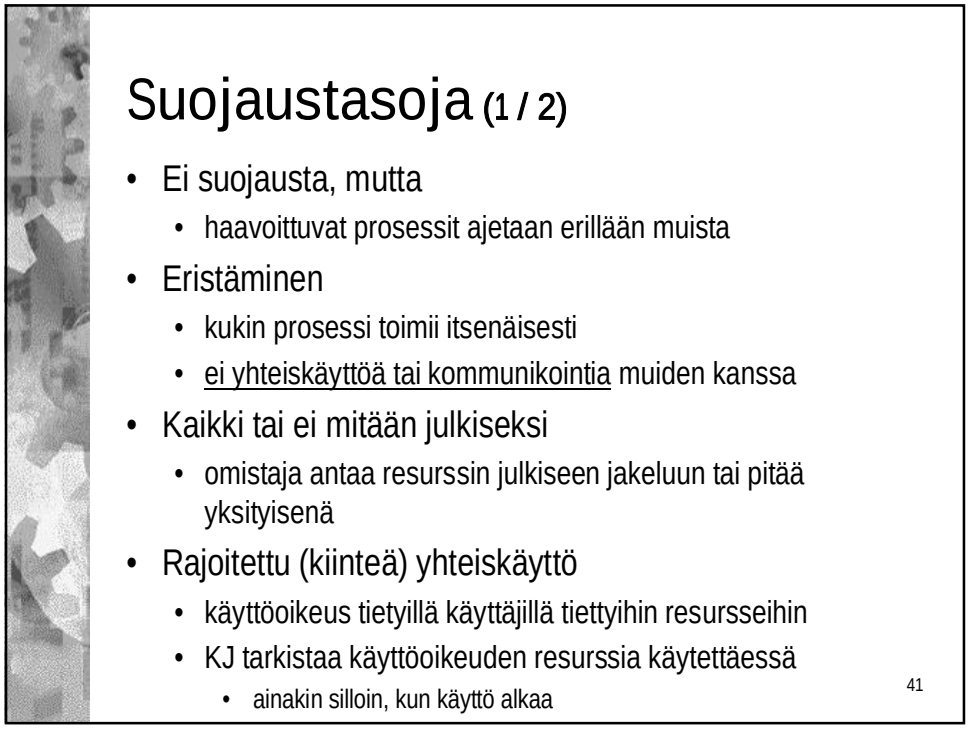
39

Suojautuminen (protection)

eli

Miten uhkia torjutaan?

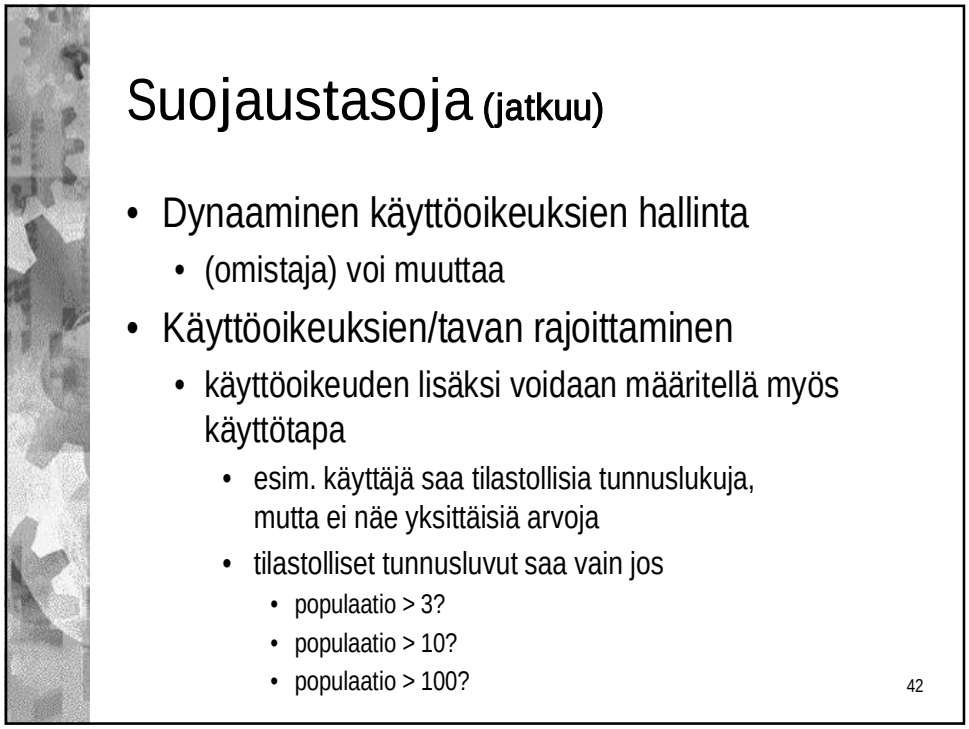
40



Suojaustasoja (1 / 2)

- 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

41



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?

42

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

43

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!

44

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 — Fig 16.5 (a) [Stal 05]
  - rivi: toimialue (domain)
  - sarake: resurssi, objekti (object)
  - alkio: toimialueen subjektin käyttöoikeus resurssiin
    - domain on myös objekti! — Fig 9-24 [Tane 01]

45

	File 1	File 2	File 3	File 4	Account 1	Account 2
User A	Own R W		Own R W		Inquiry Credit	
User B	R	Own R W	W	R	Inquiry Debit	Inquiry Credit
User C	R W	R		Own R W		Inquiry Debit

(a) Access matrix

[Stal 05]

	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			

[Tane 01]

Fig. 9-24. A protection matrix with domains as objects.

46

Käyttöoikeudet

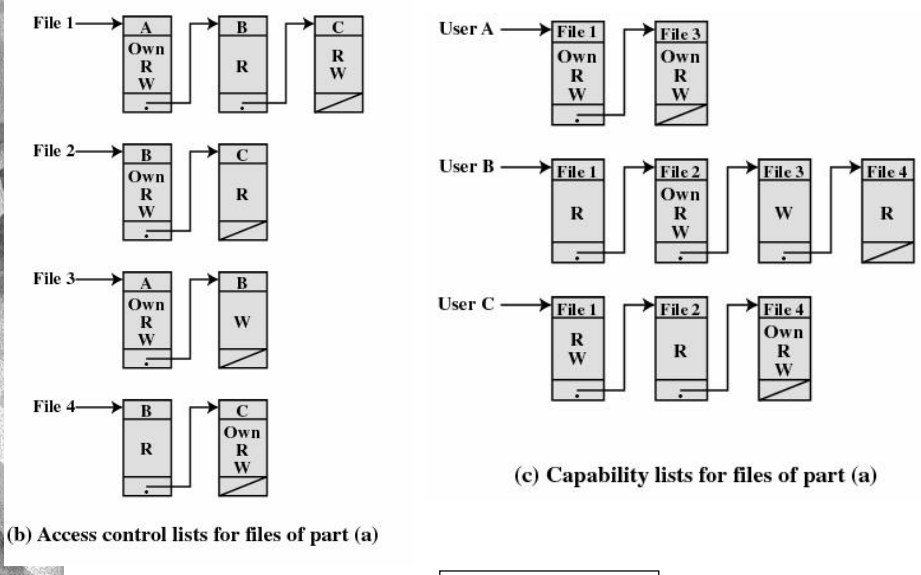
Fig 16.5 (c) [Stal 05]

- 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

Fig 16.5 (b) [Stal 05]

- 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

47



[Stal 05] Fig 16.5

48

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

49

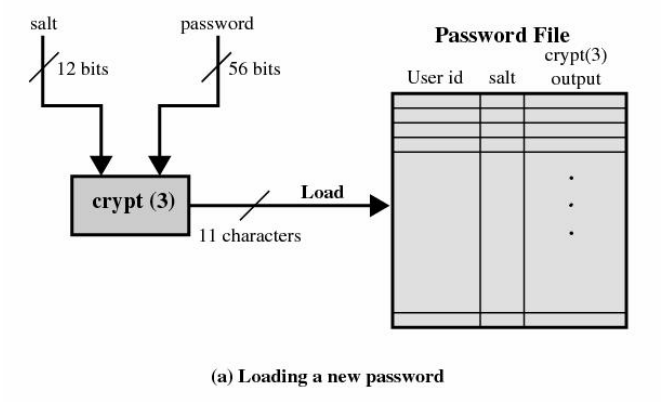
Hyvä salasana

- Koneen generoima
  - vaikeampi arvata
  - vaikeampi muistaa → paperille?
- Käyttäjän valitsema
- hylkää liian lyhyet ja helpohkosti arvattavat
  - järjestelmä voi laajentaa, salaisella 'suolalla'
    - sama salasana ei näytä aina 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]

50

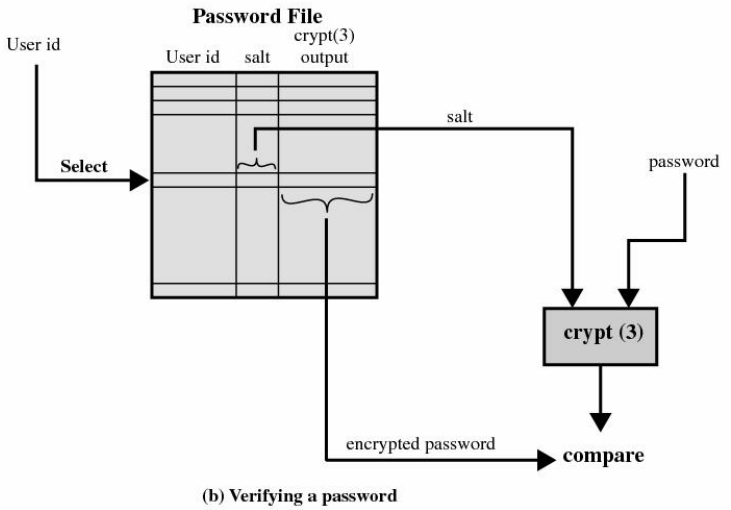
UNIXin salasanamalli



(a) Loading a new password

51

UNIXin salasanamalli



(b) Verifying a password

52

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?

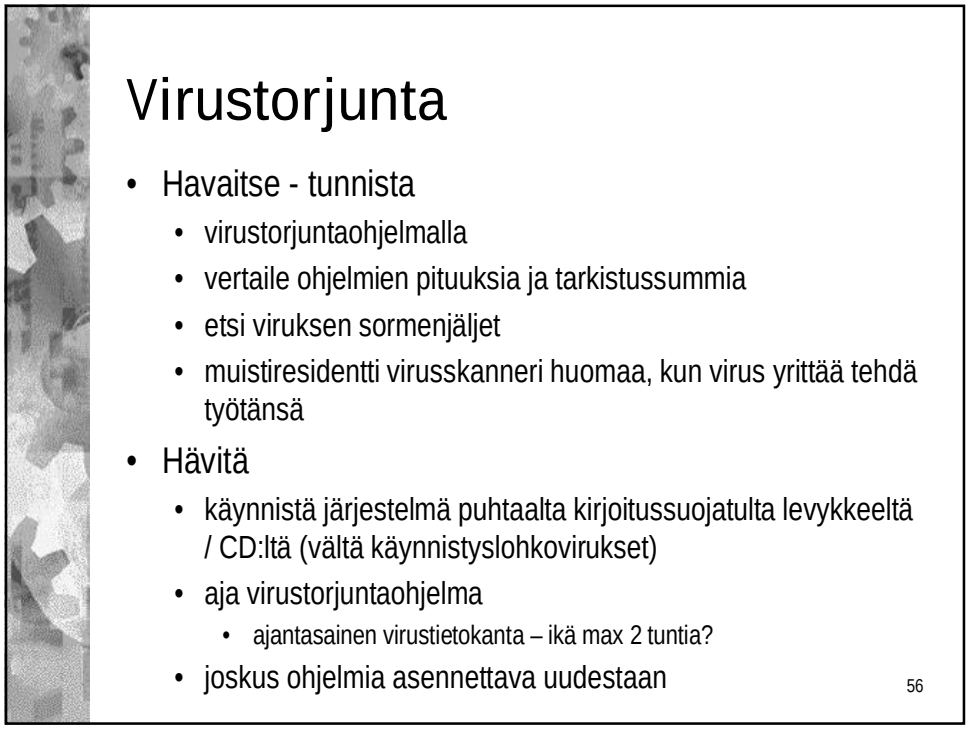
53

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?

54


Virustorjunta
55


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
56

Generic Decryption Scanner

- Polymorfisten virusten etsintään
- Tutki ohjelma ensin GD-skannerilla
  - CPU-emulaattori
  - viruksien ”sormenjälkien” tunnistin
  - ohjausmoduuli
- Emulaattori tulkitsee ohjelmaa 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

57

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

58

Digital Immune System

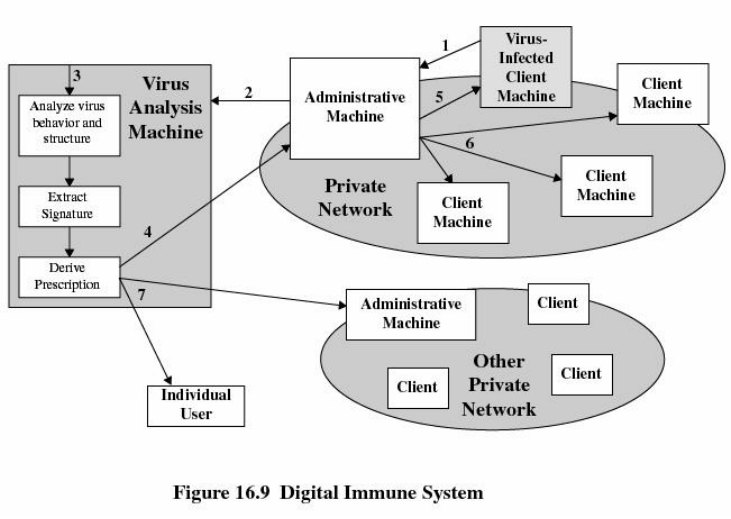


Figure 16.9 Digital Immune System

59