Forelesning 11

Inntrengere, deteksjon og virus

Inntrengere

- **Utenforstående** (Masquerador)
  Omgår aksesskontrollmekanismer for å utnytte legitime brukerkonti
- **Misbruker** (Misfeasor)
  Gyldig bruker som aksesserer ressurser vedkommende *ikke* er autorisert for
- **Skjult** (Clandestine user)
  Inntrenger som tiltvinger seg system-rettigheter, og bruker disse for å unngå å bli oppdaget

Metoder for inntrenging

- Gjetting av passord
- Avlytting av passord
- Utnyttelse av kjente feil i aksess-kontrollmekanismer og tjenester
  - Sendmail, BIND
- Trojanske hester
  - Sub7, BackOrifice

Gjetting av passord

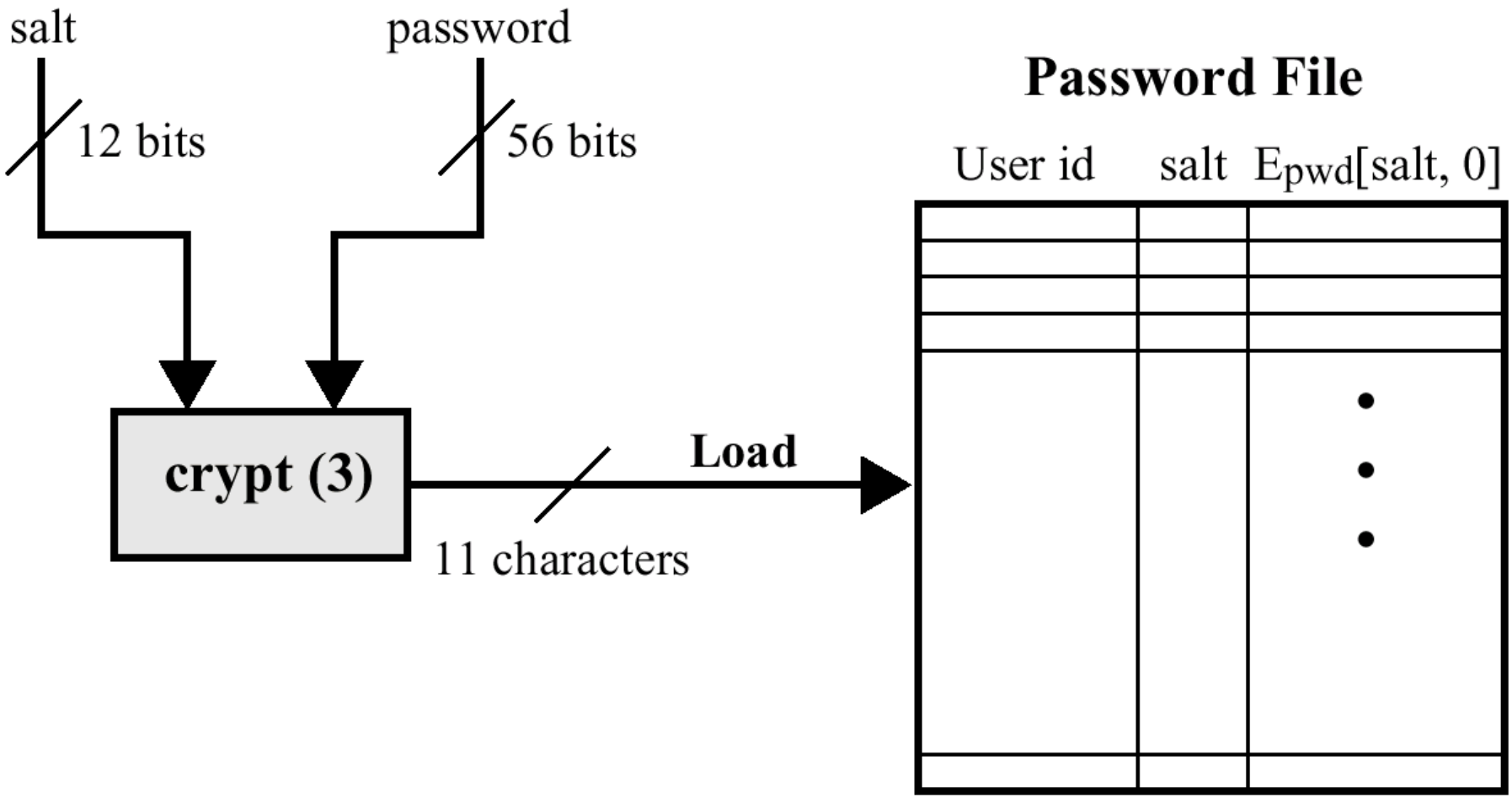
- Standardpassord
- Uttømmende søk av alle korte passord (opp til 3 tegn)
- Ordlister
- Egne lister over ofte brukte passord
- Brukerspesifikk-informasjon
- Alle gyldige registreringsnummer for bil

Beskyttelse av passordfiler

- Enveis kryptering
  Passordfilen er lesbar av alle, men inneholder bare kryptert informasjon
- Aksesskontroll
  Bare priviligerte brukere har tilgang til passordfilen

- Shadow-filer i Unix benytter begge!

Klassisk Unix passordlagring



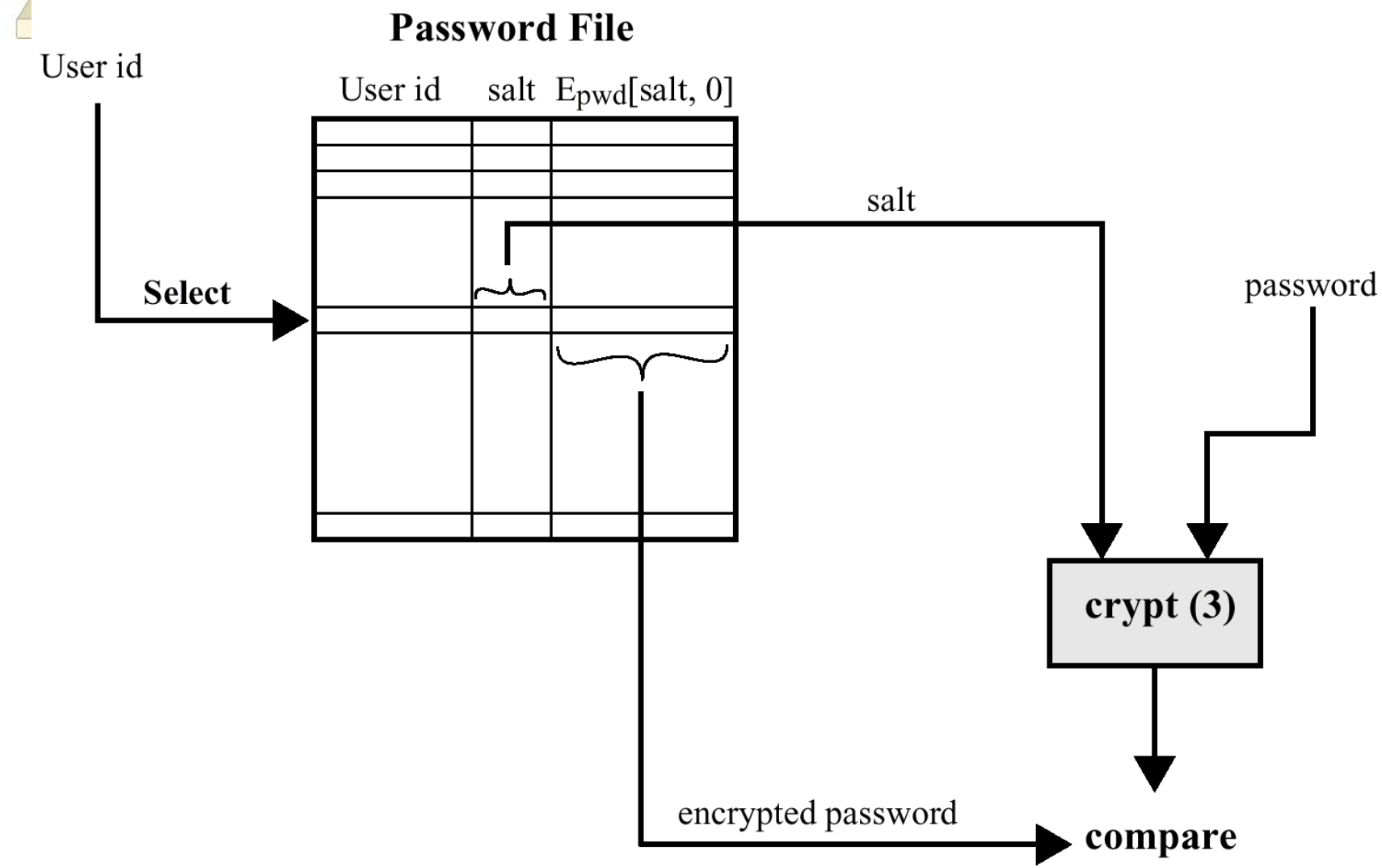
Bruker Unix DES?

- crypt(3) er DES gjentatt 25 ganger
- crypt(3) tar to parametre: passord og salt
- I første iterasjon blir saltet kryptert med passordet
- Resultatet blir så kryptert med passordet igjen, etc.

...men hvorfor ikke?

- Siden passordfila i utgangspunktet var leselig for alle, var det viktig å hindre bruk av raske hardware-implementa-sjoner av DES for uttømmende søk av passord
- Forutsetter selvfølgelig at det ikke finnes HW-implementasjoner av crypt(3)

Verifikasjon av passord



Hensikten med salt

- Like passord vil se forskjellige ut i passordfila
- Uttømmende søk i ferdigkrypterte ordlister blir ”umulig” – må ha $2^{12} = 4096$ innslag per ord

... men når alt kommer til alt

- Moderne PCer har etter hvert så stor regnekapasitet at det nesten uansett er uforsvarlig å la passordfiler være tilgjengelige
- Brukere har dessuten en lei tendens til å velge for korte/enkle passord
- De fleste Unix- og Linux-systemer bruker i dag ”shadow” passordfiler

Et utvalg UNIX passordlengder

Statistikk fra
54 maskiner ved
Purdue University
i 1992.

Lengde	Antall	Andel
1	55	0,4%
2	87	0,6%
3	212	2%
4	449	3%
5	1260	9%
6	3035	22%
7	2917	21%
8	5772	42%
Totalt	**13787**	**100%**

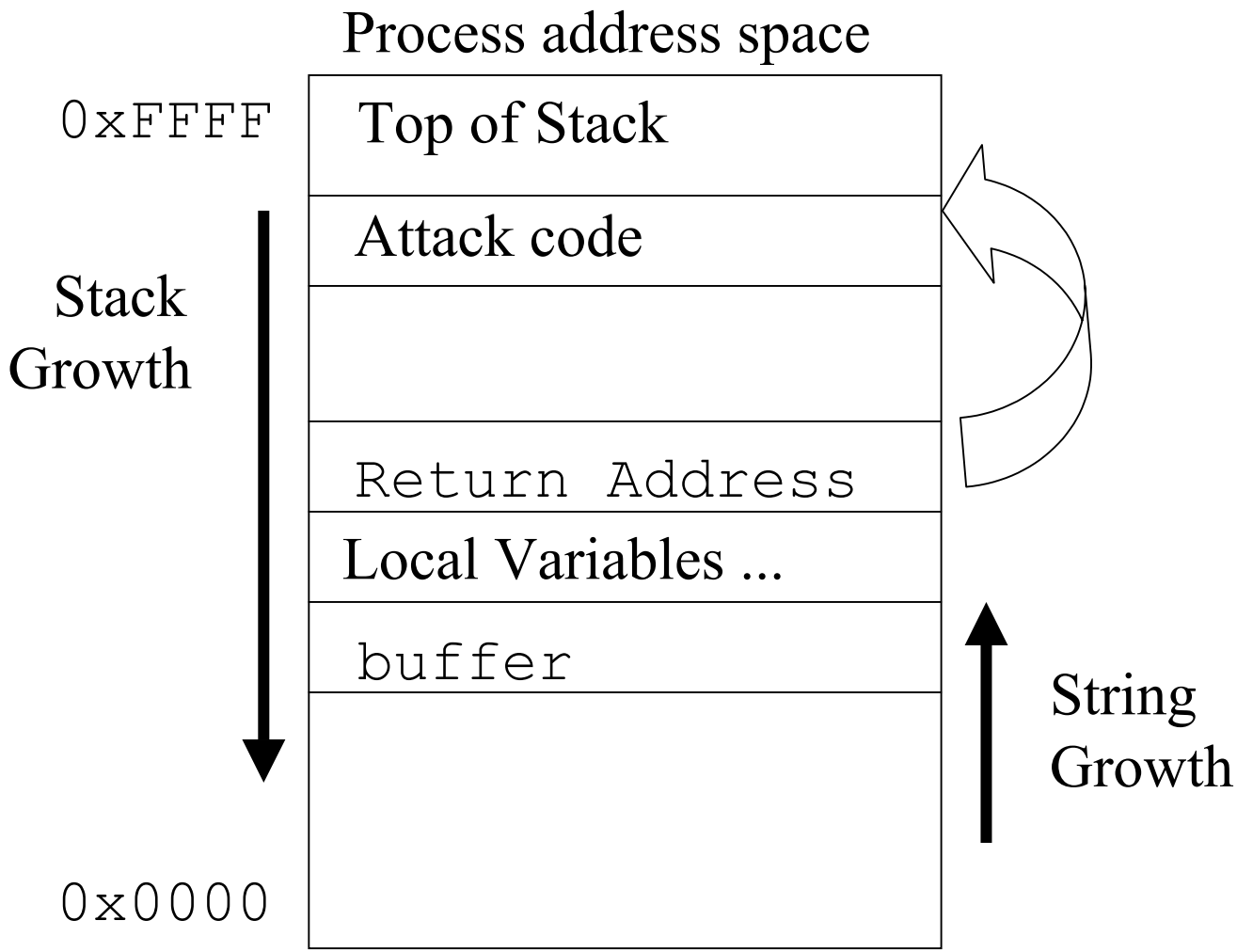
Buffer Overflow

- Mange tjenester kjører med system-privilegier, og aksepterer input fra brukere
- Hvis programmet ikke sjekker *hvor mye* data som blir lest inn (bounds checking) risikerer man at bufferet (typisk: arrayet) som skal holde inputen flyter over - man skriver til minne man egentlig ikke har allokert

Buffer Overflow forts.

- Det minste man risikerer er at tjenesten tryner (DoS)
- Med litt flaks/dyktighet kan det imidlertid være mulig å overskrive returpekeren i den aktuelle rutinen
- Har da mulighet til å kjøre valgfri kode med systemrettigheter!

Stack Smashing



Fra Cowan et al, USENIX SC '98

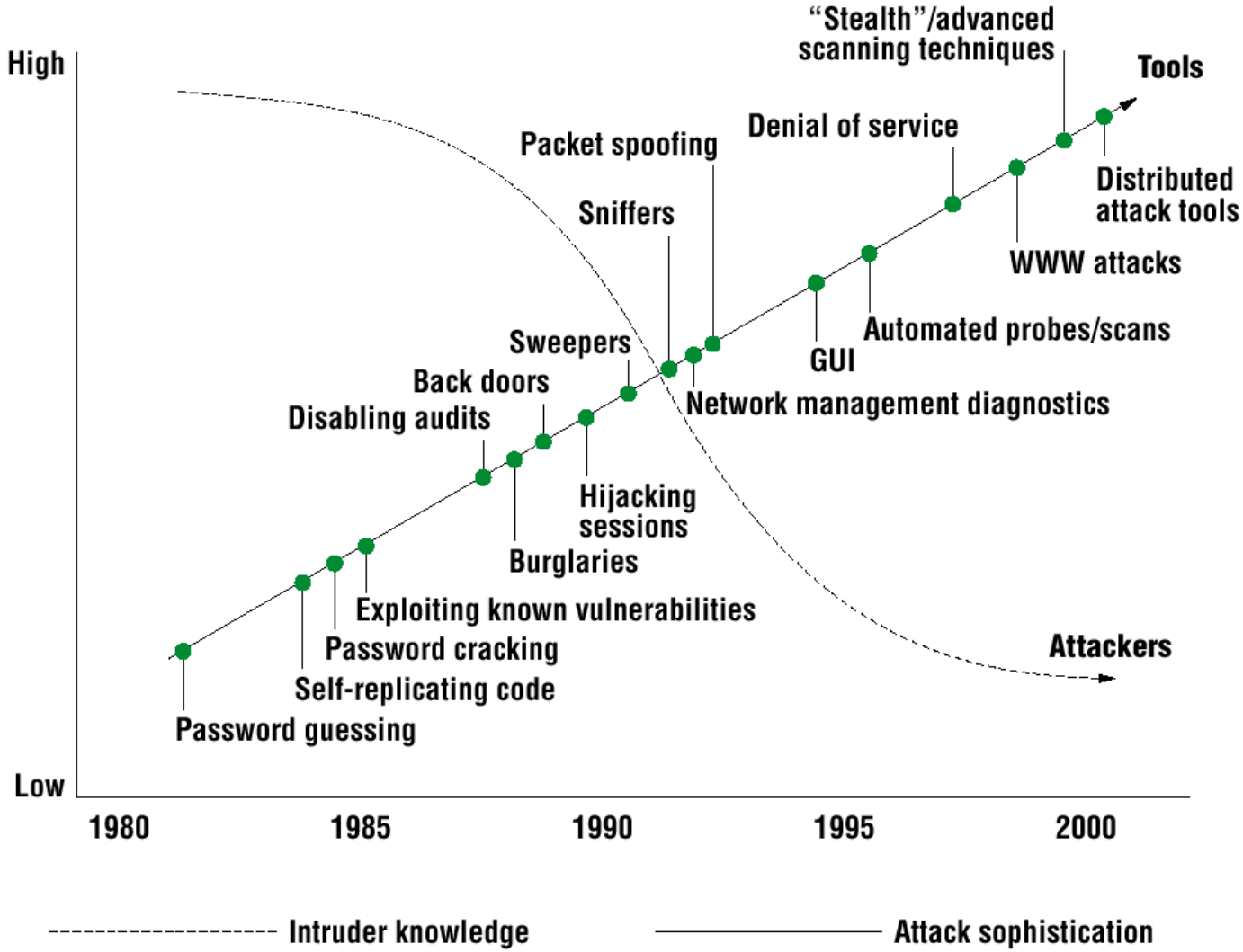
Beskyttelse mot Buffer Overflow

- Problemet er bruk av ”usikre” primitiver for lesing av data (gets, strcpy)
  - Sjekker ikke grenser for input
- En mulighet er å bruke en spesiell kompilator for å ”luke ut” tvilsomme kall
  - Ikke 100% sikker...?
- En annen mulighet er modifikasjon av operativsystemet for å hindre endring av stakken under kjøring
  - StackGuard

Hvorfor alltid på vikende front?

- Den som skal beskytte et system, må tette alle mulige sikkerhetshull og svakheter
- Den som skal angripe et system, trenger bare finne *en* svakhet eller sikkerhetshull

Angrepsterskel



Inntrengningsdeteksjon (ID)

- Hvis man ikke kan hindre inntrengere i å komme inn, er det nest beste å oppdage dem så tidlig så mulig

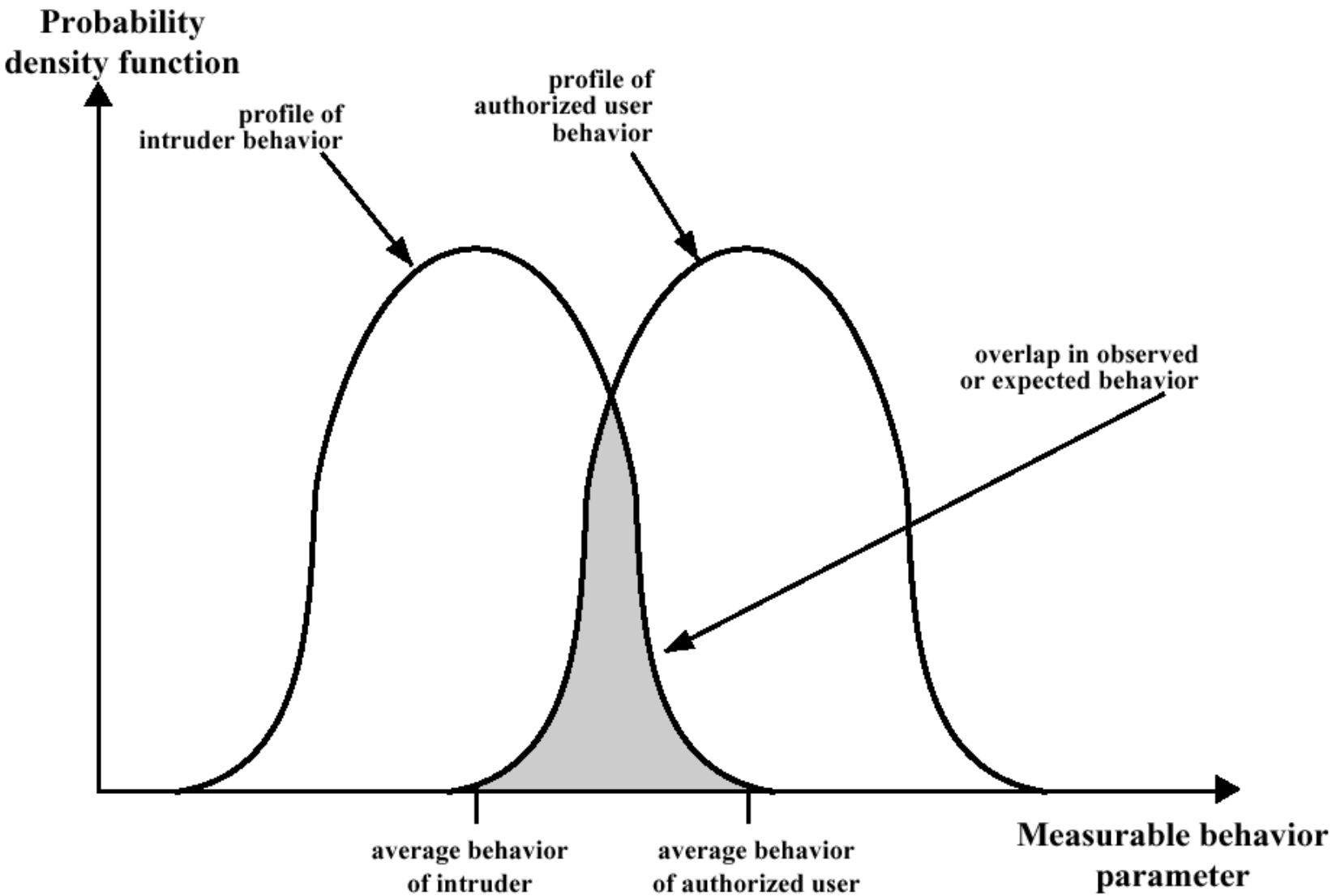
Hvorfor IDS?

- Tidlig oppdagelse kan medføre at inntrengere kan identifiseres og stoppes før de får gjort skade
- Kan ha avskrekkende effekt (bilalarm-effekten)
- Et IDS kan bidra til å bygge opp kunnskap som i neste omgang kan forhindre inntrenging

Tradisjonell ID

- Uvanlige aktiviteter for en gitt bruker
- Aktiviteter på uvanlige tidspunkt
- Forsøk på å utføre priviligerte instruksjoner
- Baserer seg på tradisjonelle ”audit”-logger

Oppførselsprofiler



Offerets perspektiv

- Hva skjedde?
- Hvem ble påvirket, og hvordan?
- Hvem er inntrengeren?
- Hvor og når oppstod inntrengningen?
- Hvordan og hvorfor skjedde inntrengningen?

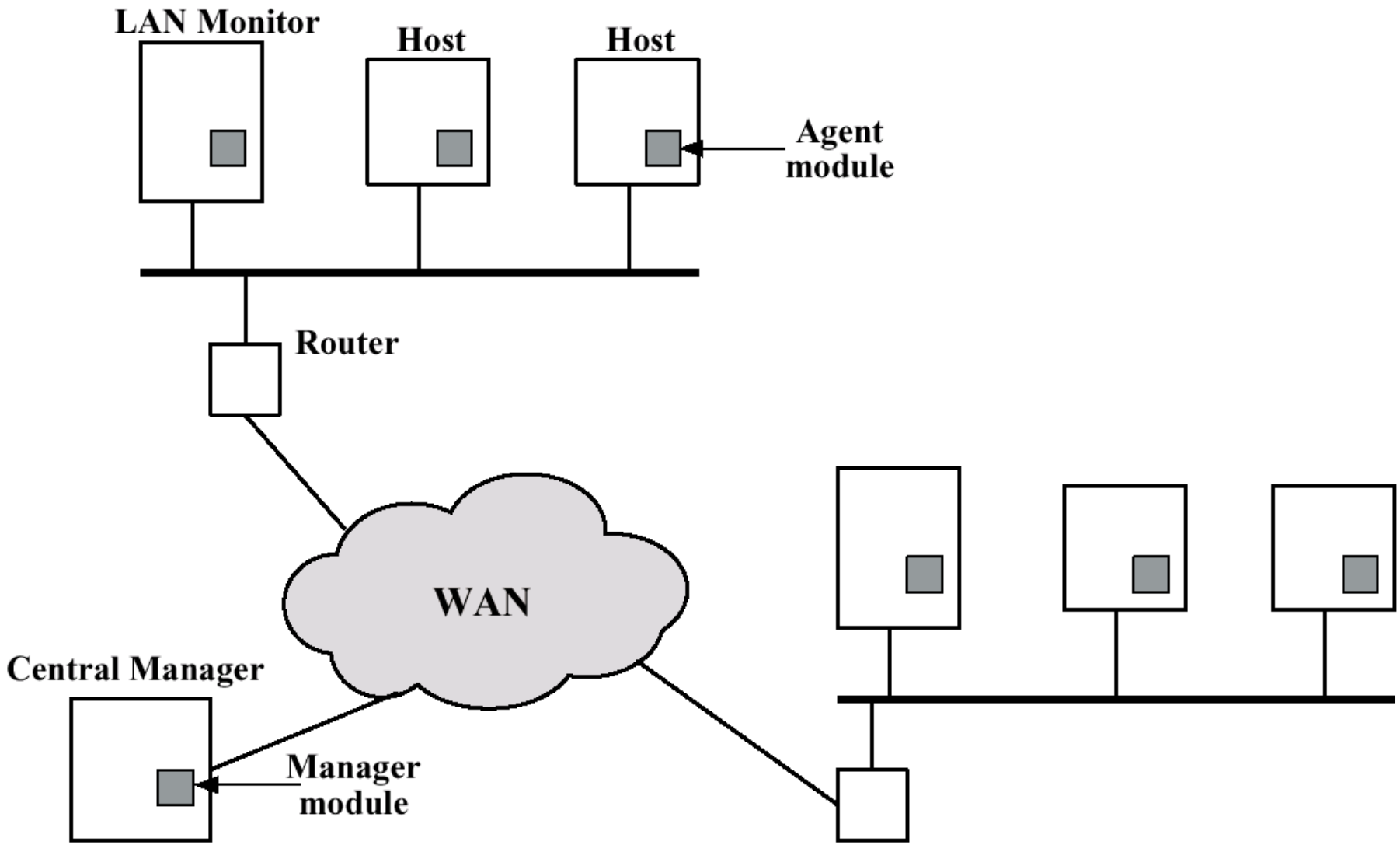
Angriperens perspektiv

- Hva skal jeg oppnå?
- Hvilke svakheter finnes på mål-systemet?
- Hvilken skade eller andre konsekvenser er sannsynlige?
- Hvilke ”exploits” eller andre angreps-verktøy er tilgjengelige?
- Hva er risikoen for at jeg blir oppdaget?

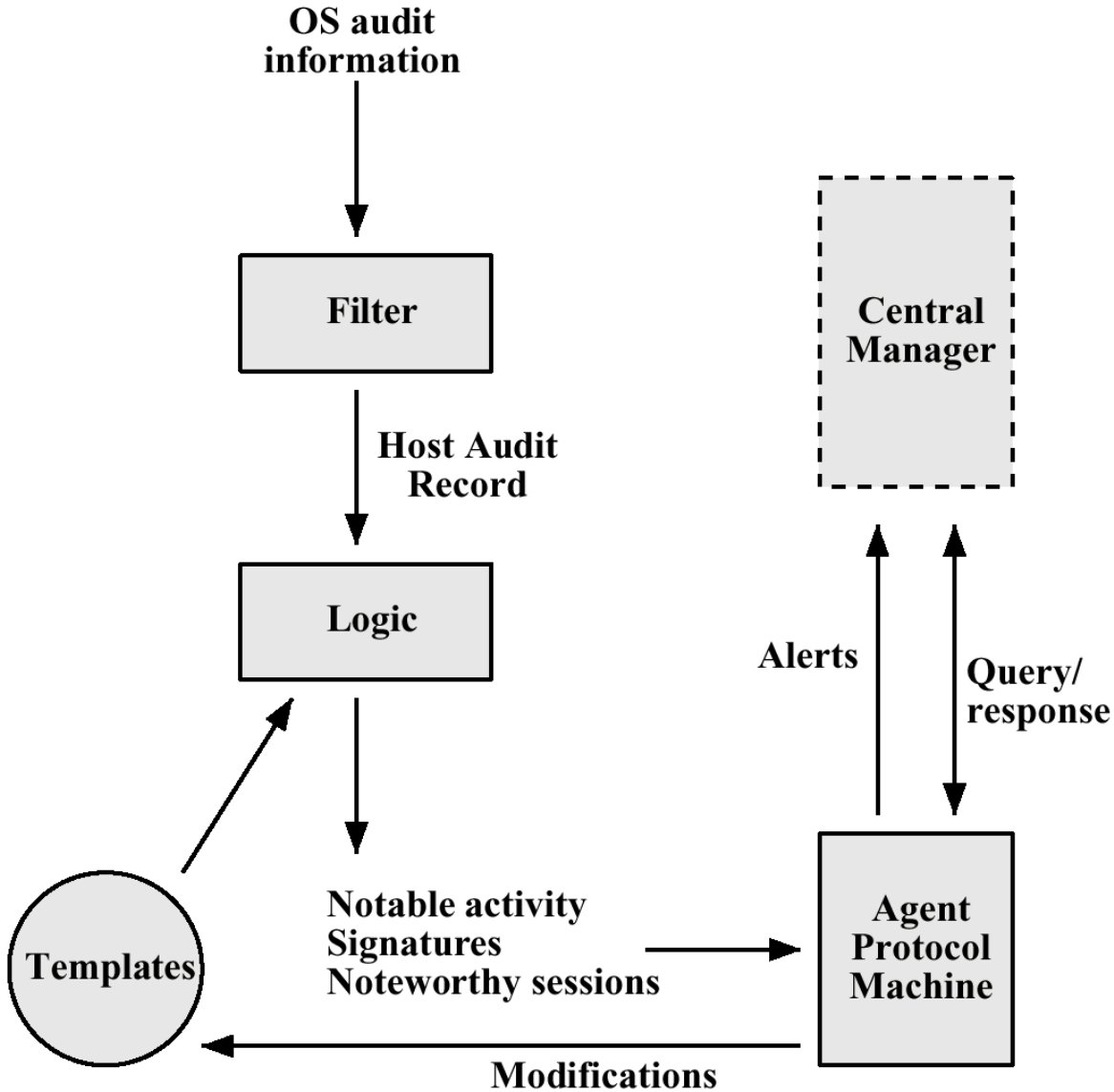
To hovedtyper IDS

- Nettverksbasert (NIDS)
  - ”Sniffer” nettverket
- Maskinbasert (Host-based IDS)
  - Ser trafikk til egen maskin
  - Kan også vurdere interne logger etc.

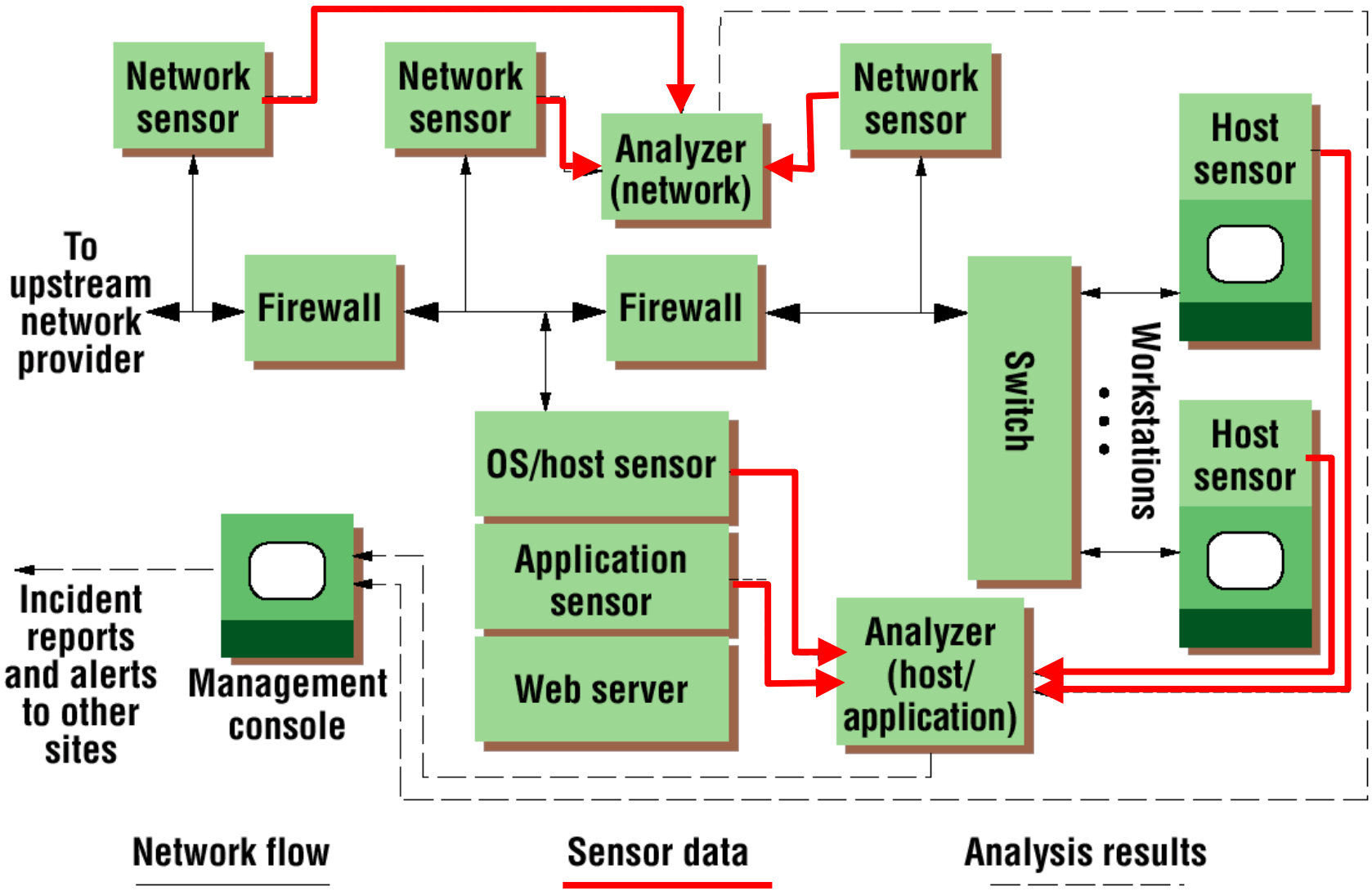
Distribuert IDS



Agent-arktitektur



Et IDS-beskyttet nettverk



Tilnærminger til ID

- ID kan ses på som signal-detekterings-problem
- Tegn på inntrengning er signalet
- Normale operasjoner er støy

Tilnærminger - hovedtyper

- Signal
  (Signaturbasert IDS)
- Støy
  (Anomalibasert IDS)

Signaturbasert IDS

- Signaturer til kjente angrep bygges opp
- Spesiell pakkeflyt
- Spesielle porter
  - 31337 – ”eleet”-porten (BO)

Anomalibasert IDS

- Må ”lære seg” hva som er ”normal” oppførsel/trafikk for et system/nettverk
- Vanskelig!
- Klassifisering av angrep vanskelig

Kommersiell IDS

- Kun signaturbaserte IDS tilgjengelige på markedet i dag
- NFR
- ISS RealSecure

- Dessuten: Tripwire (integritets-sjekk)

Faser i livsløpet til et IDS

- Evaluering og utvelgelse
- Installering
- Bruk
  - Analyse
- Vedlikehold
  - Signaturer, oppgradering

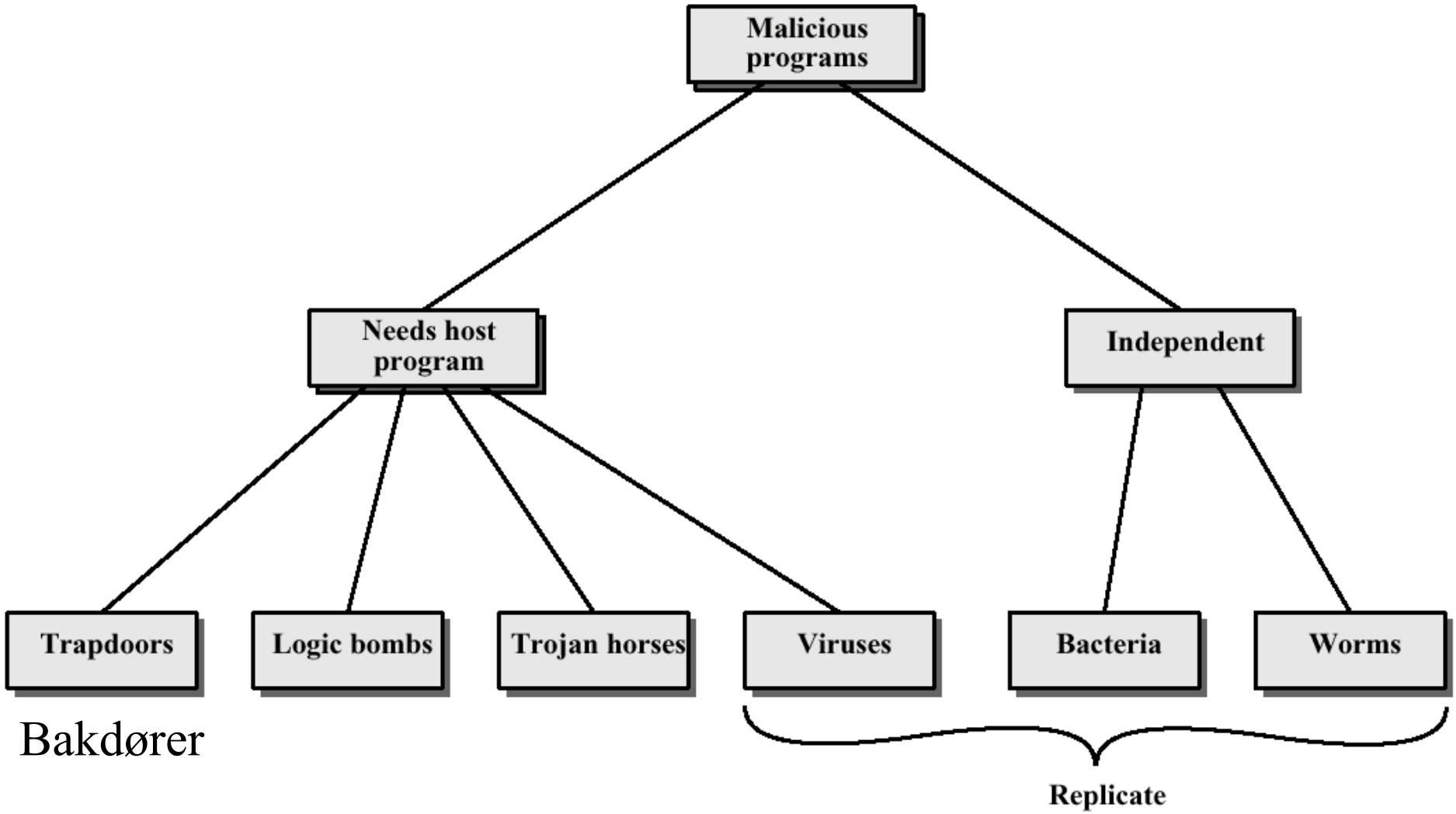
Administrative IDS-utfordringer

- Ingen inntrengningsdeteksjonssystemer i dag kan sies å være helautomatiske
- De beste av dagens IDS klarer ikke mer enn 80% av alle angrep
- Det kreves høyt kvalifisert personell for å vurdere resultater fra et IDS
- For mange vil det være bedre med ”ingenting” enn ”litt”

Virus og ondartede programmer

- Bakdører
- Logiske bomber
- Trojanske hester
- Virus
- Ormer
- Bakterier

Familietre


Malicious programs
Needs host program
Independent
Trapdoors
Logic bombs
Trojan horses
Viruses
Bacteria
Worms
Bakdører
Replicate

Hva er et virus?

- Et virus er et (lite) program som ”haiker” med andre programmer
- Et virus sprer seg til andre programmer
- Et virus har vanligvis en ”nyttelast” (payload) med mer eller mindre destruktive konsekvenser
- Et virus benytter forskjellige mekanismer for å holde seg skjult

Virus-historikk

- Tanke-eksperimenter og science fiction allerede på 70-tallet
- Første kjente PC-virus ”in the wild”: Brain, 1986
- Stammet fra Lahore, Pakistan
- Kopibeskyttelse?
- Infiserte mengder av amerikanske universiteter

Hovedtyper av virus

- Boot-sektor-virus
  - Smitter via disketter
  - Smitter når en infisert diskett glemmes igjen i PCen når den slås på
- Filvirus
  - Smitter når infisert program kjøres
- Makrovirus
  - Viruset er en makro i f.eks. et Word-dokument
  - Smitter når dokumentet åpnes

Andre egenskaper til virus

- Stealth
  - Benytter avanserte mekanismer for å skjule seg
  - Kan modifisere systemkall til å vise feil størrelse på filer, etc.
- Polymorfisk
  - Endrer seg hver gang det sprer seg for å unnslippe deteksjon
  - Vanligvis vha. kryptering med tilfeldig nøkkel

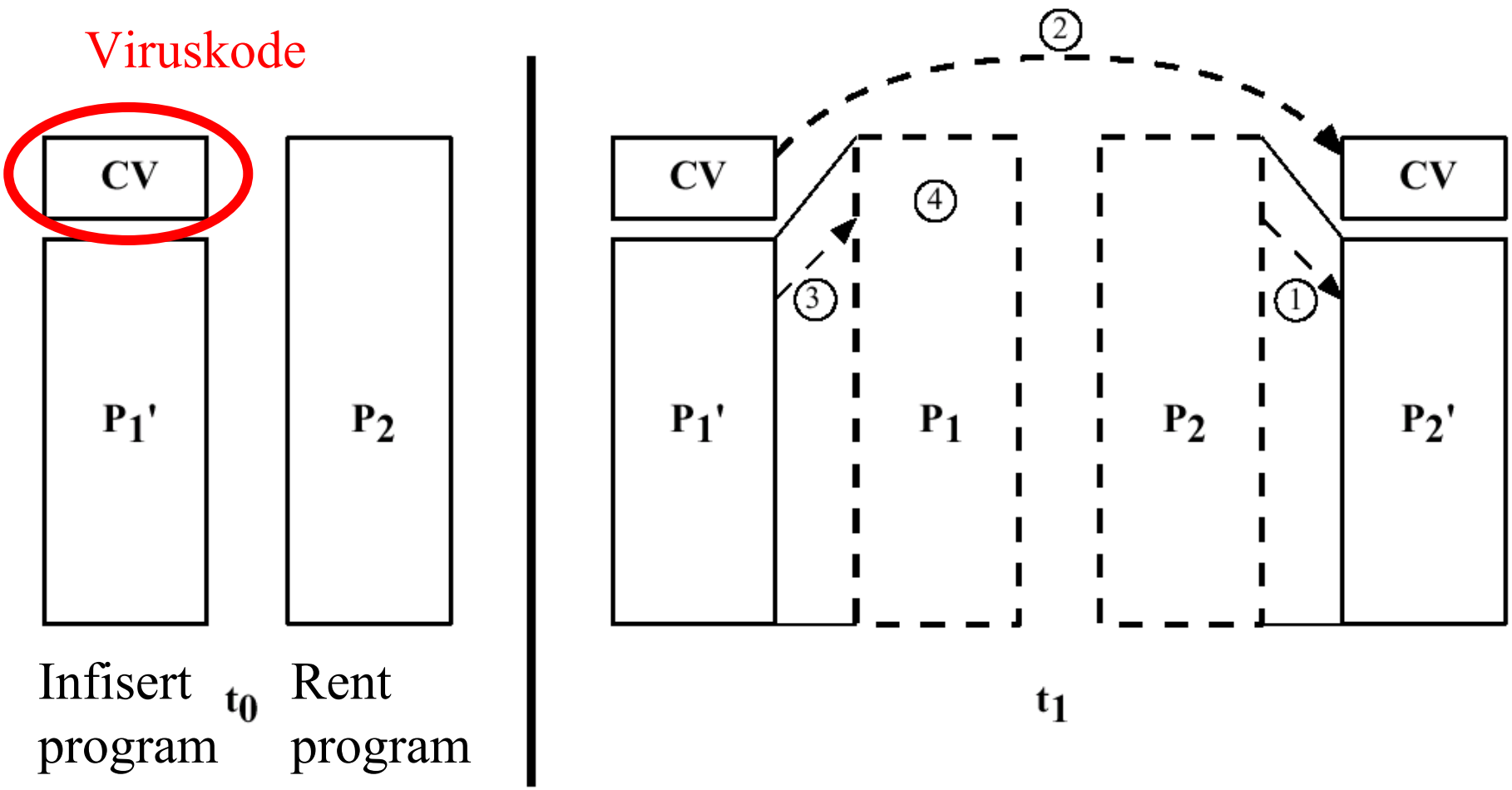
Skjuling ved komprimering

- Et program infisert med et filvirus vil i utgangspunktet øke i størrelse – dette kan oppdages
- Viruset kan skjule seg ved å *komprimere* det opprinnelige programmet slik at lengden ikke forandres

Kjøring av et komprimert offer

- Når et infisert program eksekveres, vil viruskoden typisk kjøre først
- Viruset gjør det det skal (smitter, payload, etc.), og dekomprimerer deretter det opprinnelige programmet
- Det dekomprimerte programmet får så kjøre som vanlig

Komprimeringsvirus



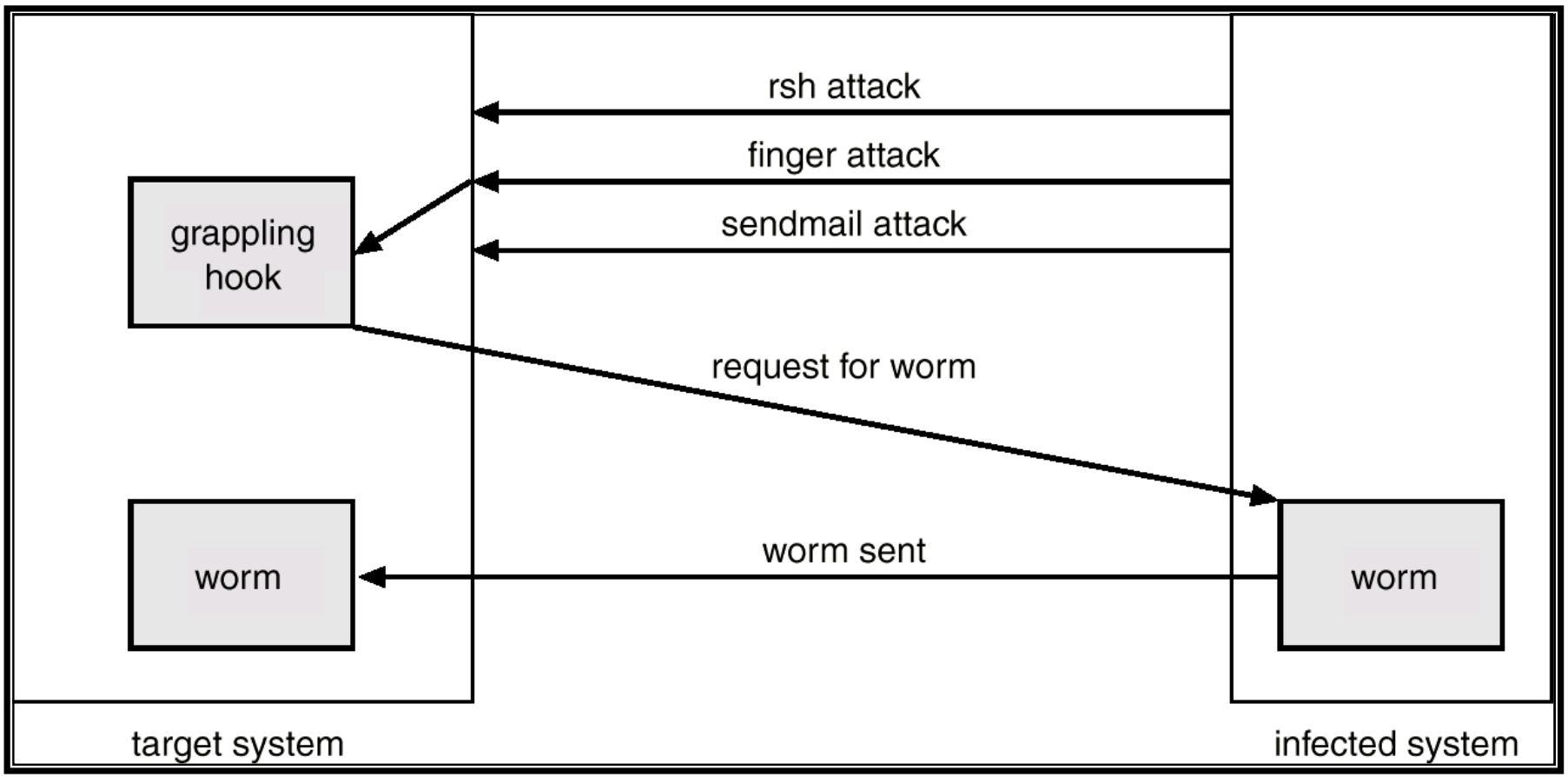
Klassiske virusteknikker

- Tidlige virus var svært HW-nære
- Utnyttet skjulte kanaler
  - Ekstra track på disketter

Ormer

- Ofte kalt virus!
  - Iloveyou
  - Kournikova
- Sprer seg tradisjonelt med mail-systemer
- Klassisk: Morris Worm (1987)
  - Buffer overflow i ”finger”
  - (de)Bug i ”Sendmail”

The Internet (Morris) Worm



Fra Silberschatz et al: "Operating Systems Concepts" 6e
©2002 John Wiley & Sons, Inc

Virus mottiltak

- Scannere
- Sjekksummere
- Anomalibasert deteksjon

Scannere

- Virus-spesifikk metode
- Baserer seg på en spesifikk signatur per kjent virus
- Må kontinuerlig oppdateres
- Kan ikke oppdage ukjente virus

Sjekksummere

- Helt analogt til sjekksum-verktøy for IDS
- Med utgangspunkt i et ”rent” system, generer sjekksum (hash) for signifikante statiske komponenter
- Brukes for å avgjøre om et system er infisert
- Kan ikke *hindre* infeksjon

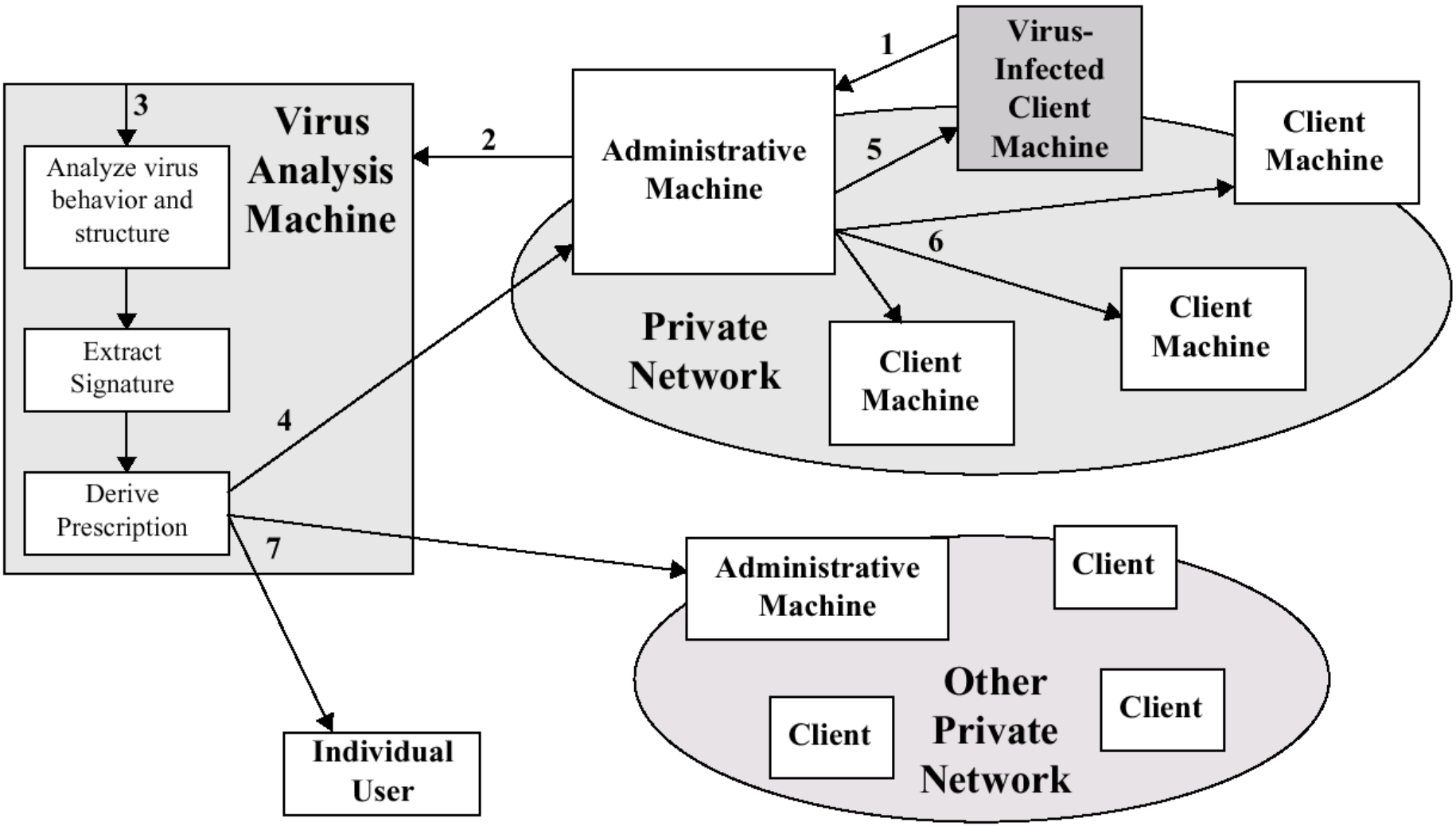
Anomalibasert antivirus

- Hardware-basert eller minne-resident
- Evaluerer operasjoner utført av pro-grammer for å vurdere om de er ”lovlige”
- ”Ulovlige” operasjoner medfører alarm
- F.eks. skriving til boot-sektor av bruker-programmer
- Falske alarmer et problem!

Nyere antivirus-teknikker

- ”Virtual machine”
  Kjører kode i et kontrollert miljø
  - Kjøre scannere mot kode i minnet
  - Evaluere handlinger utført av programmet

Et digitalt immunsystem



Dagens website

- http://computer.org/computer/sp/s1%20s&p%20supplement.lo.pdf
- Eget Security&Privacy vedlegg til IEEE Computer Magazine - last ned gratis!