

Universita' degli Studi del  
Molise

Laurea in Produzione e  
Gestione  
di Servizi Informatici  
a.a. 2005/06

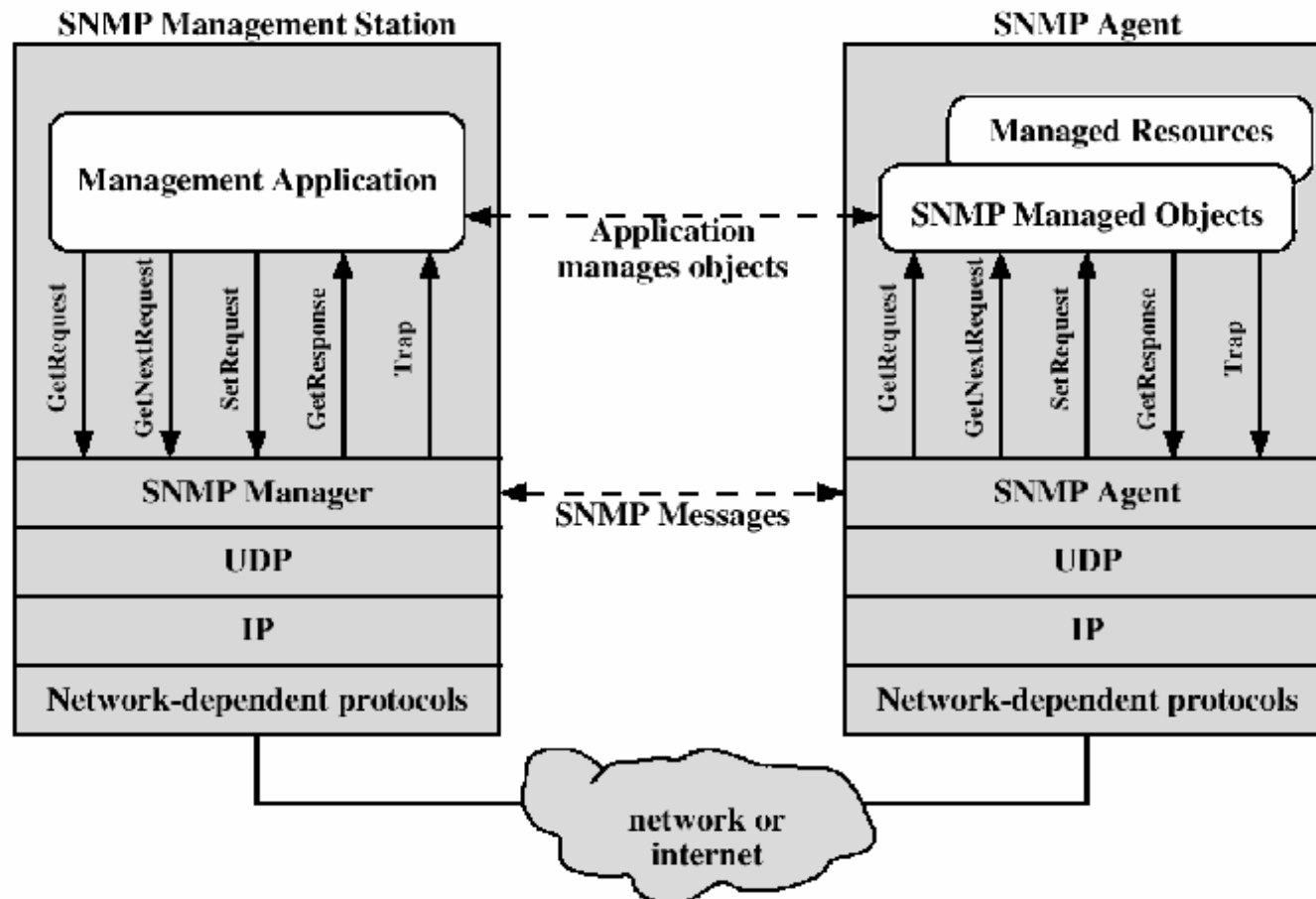
Sicurezza delle reti

*Prof. Mario Petrone*

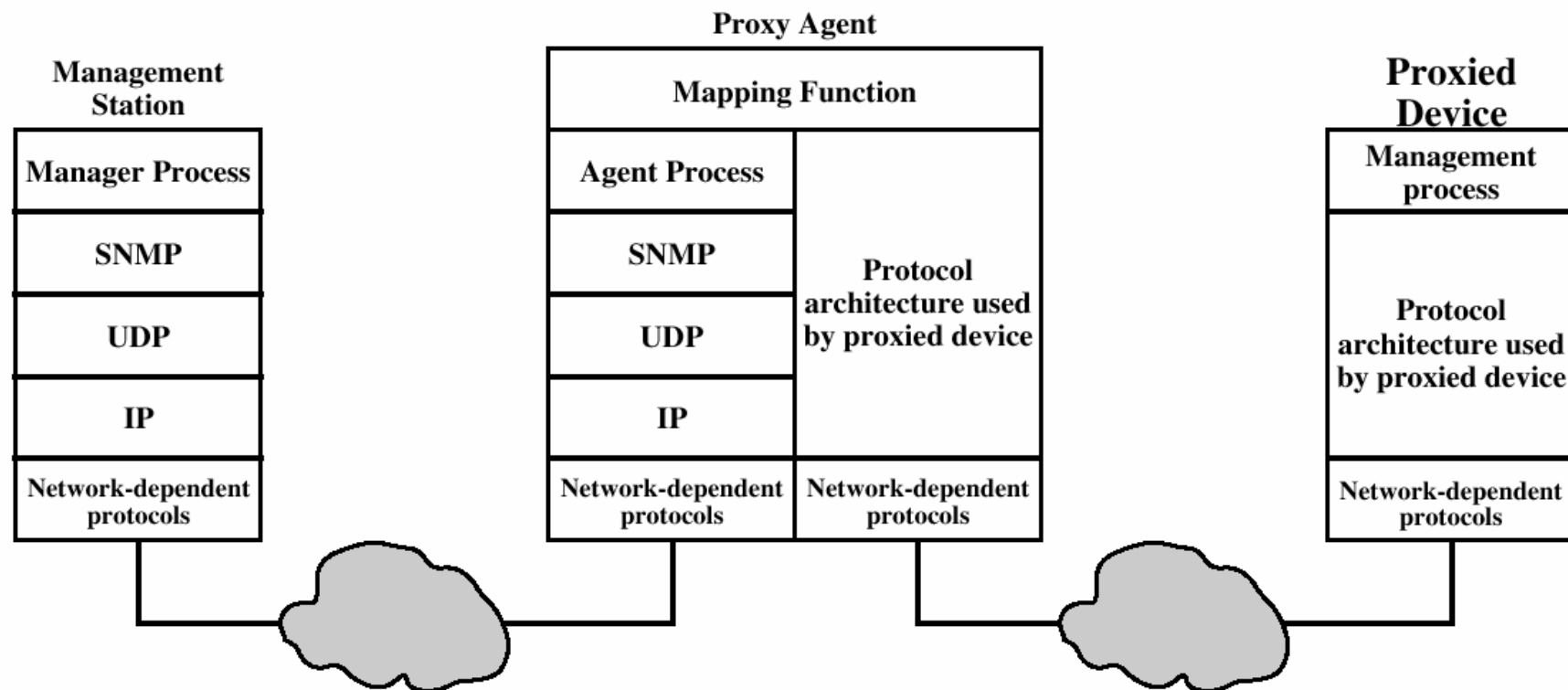
# Concetti di base di SNMP

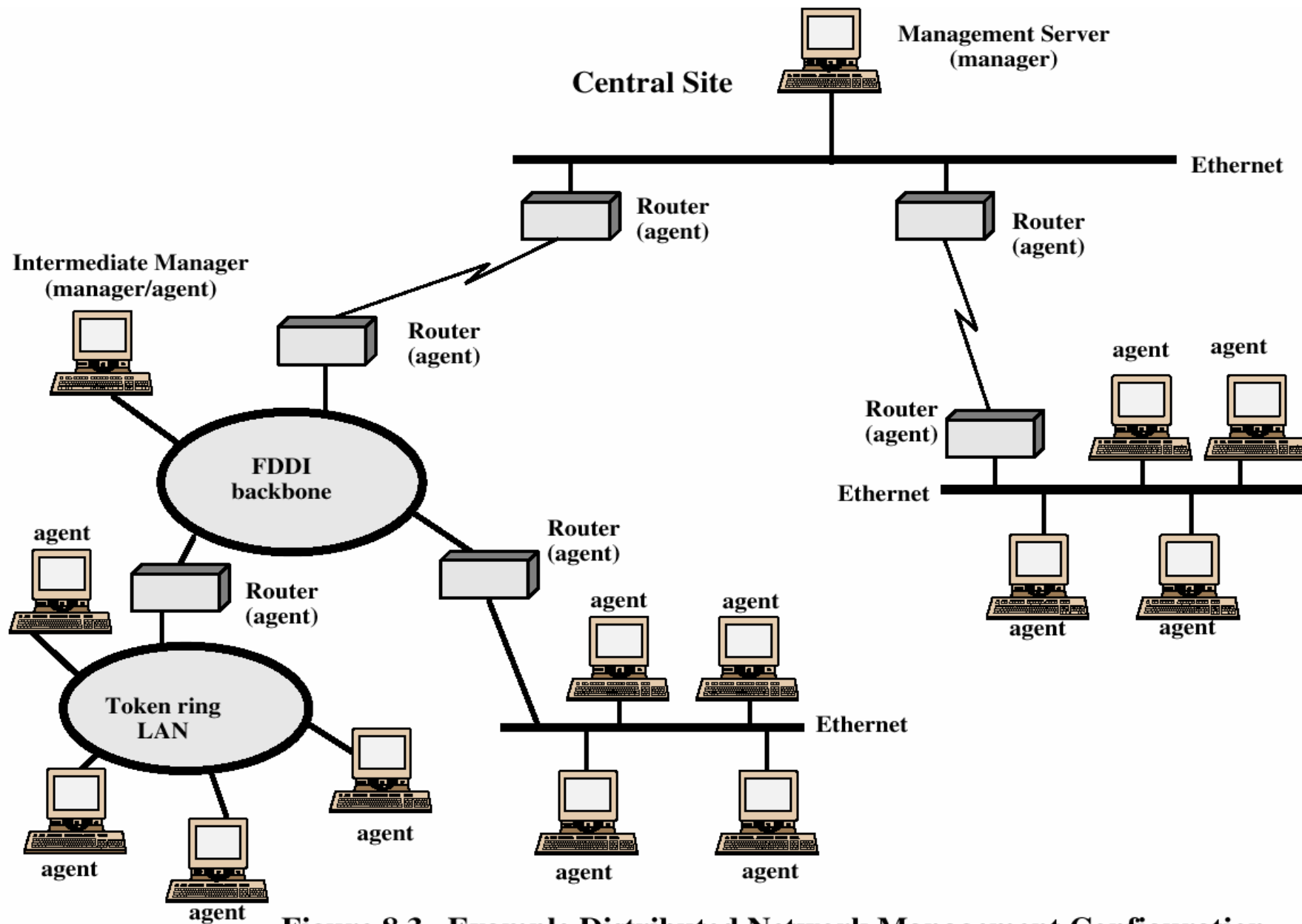
- Insieme di tool per il monitoraggio ed il controllo delle reti.
  - Singola interfaccia utente
  - Insieme minimo di apparecchiature aggiuntive, in quanto la maggior parte dei dispositivi hardware e software necessari è incorporata all'interno dei sistemi utente
- Elementi chiave di SNMP:
  - Stazione di gestione
  - Agente gestore
  - Base informativa di gestione (MI B)
  - Protocollo di gestione di rete

# Protocollo SNMP



# Uso dei Proxy in SNMP





**Figure 8.3 Example Distributed Network Management Configuration**

# SNMP v1 e v2

- Notifiche (Trap)
- SNMPv1 è "connectionless" poiché utilizza UDP
- SNMPv2 permette l'utilizzo di TCP per ottenere un servizio "reliable, connection-oriented".

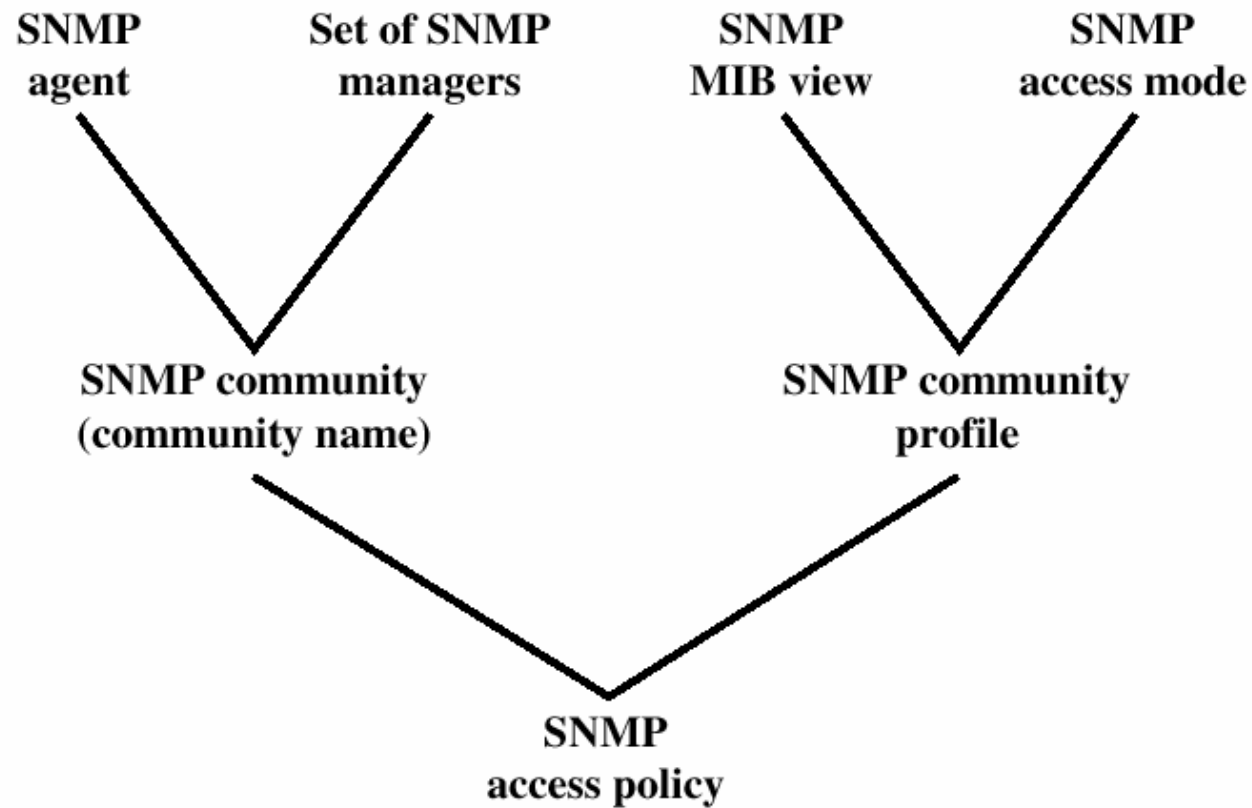
# Confronto fra SNMPv1 e SNMPv2

| SNMPv1 PDU     | SNMPv2 PDU     | Direction                                      | Description                               |
|----------------|----------------|--|---|
| GetRequest     | GetRequest     | Manager to agent                               | Request value for each listed object      |
| GetNextRequest | GetRequest     | Manager to agent                               | Request next value for each listed object |
| -----          | GetBulkRequest | Manager to agent                               | Request multiple values                   |
| SetRequest     | SetRequest     | Manager to agent                               | Set value for each listed object          |
| -----          | InformRequest  | Manager to manager                             | Transmit unsolicited information          |
| GetResponse    | Response       | Agent to manager or Manager to manager(SNMPv2) | Respond to manager request                |
| Trap           | SNMPv2-Trap    | Agent to manager                               | Transmit unsolicited information          |

# SNMPv1 – Concetto di Community

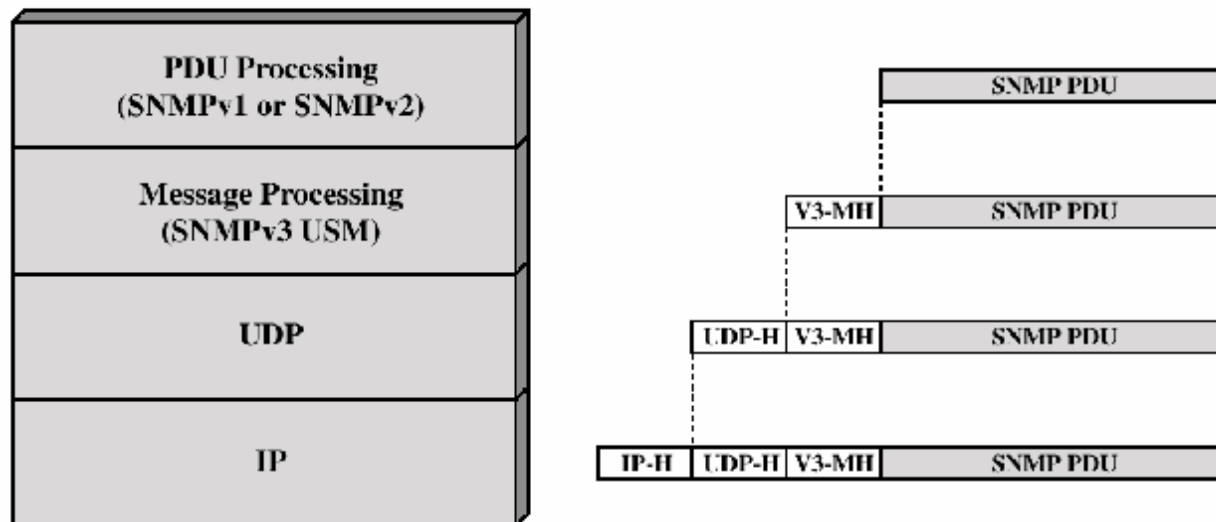
- SNMP Community – funzionalità di sicurezza basata sul concetto di comunità
- Tre aspetti:
  - Servizio di autenticazione: accessi alla MIB solo ai gestori autorizzati
  - Politica di accesso: tipologie di accesso differenti per gestore
  - Servizio Proxy: autenticazione e politiche di accesso per gli altri agenti nel proxy
- Una comunità SNMP è una relazione tra un agente ed un insieme di gestori, che definisce le caratteristiche dell'autenticazione, del controllo dell'accesso e dei proxy

# SNMPv1 Elementi di amministrazione



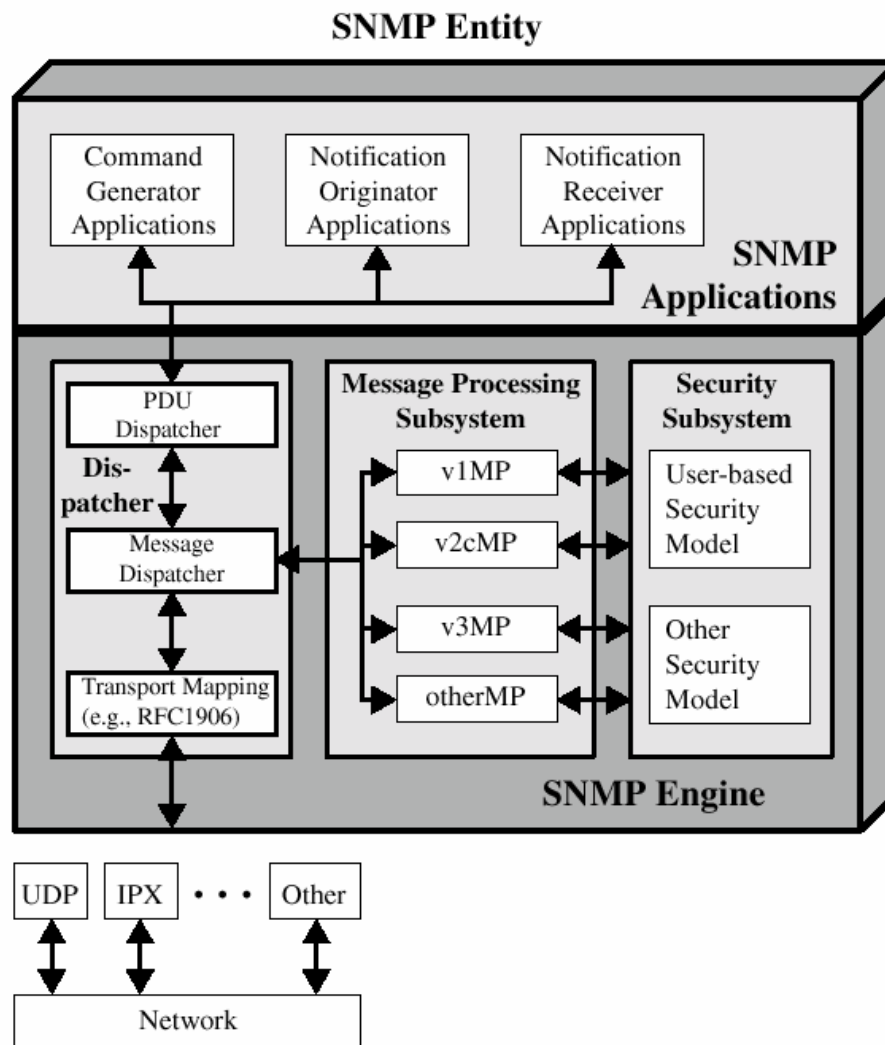
# SNMPv3

- SNMPv3 mette a disposizione funzionalità di sicurezza da usare in combinazione con SNMPv1 o v2

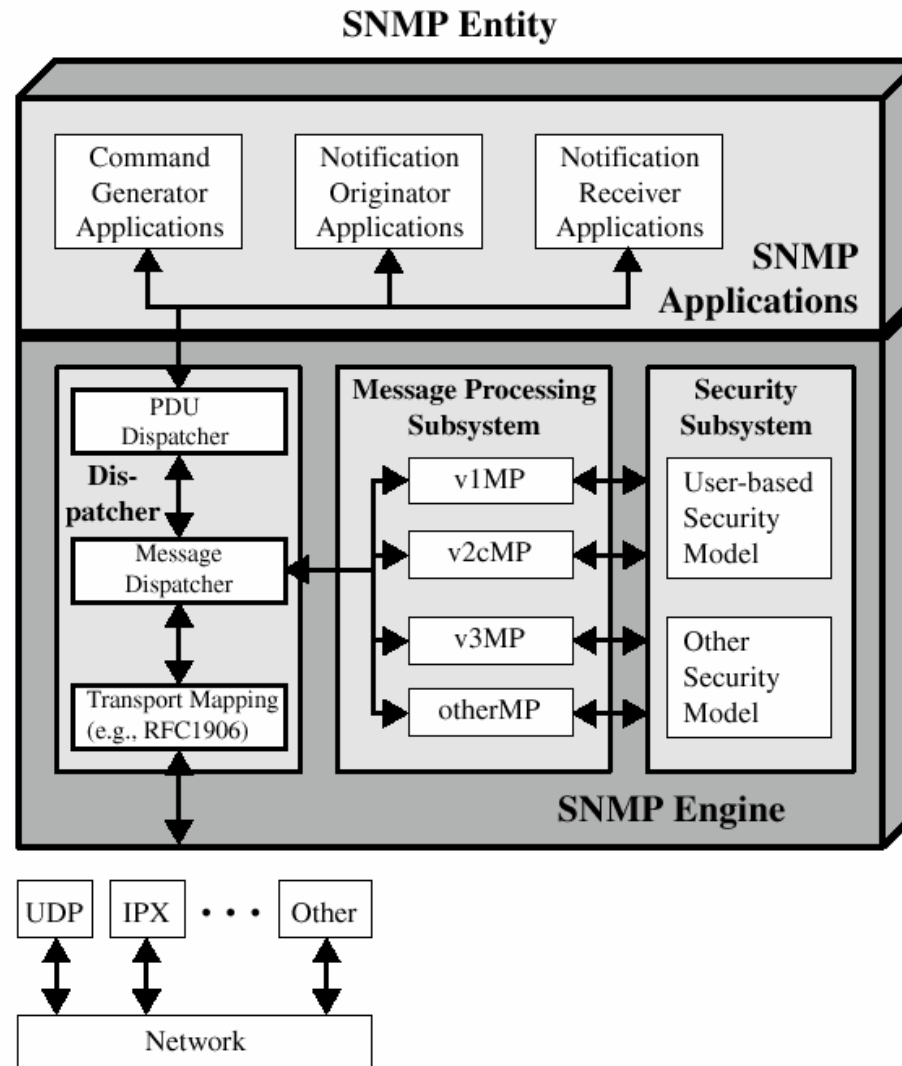


IP-H = IP header  
UDP-H = UDP header  
V3-MH = SNMPv3 message header  
PDU = Protocol data unit

# Manager SNMP



# Agente SNMP



# User Security Model (USM)

- Progettato per fornire protezione contro:
  - Modifiche delle informazioni: alterare il contenuto del messaggio
  - Masquerade: impersonare un'entità autorizzata
  - Alterazione del flusso dei messaggi: riordinare, ritardare, replicare messaggi
  - Divulagazione: catturare e decodificare messaggi
- Non garantisce sicurezza contro:
  - Denial of Service
  - Analisi del traffico

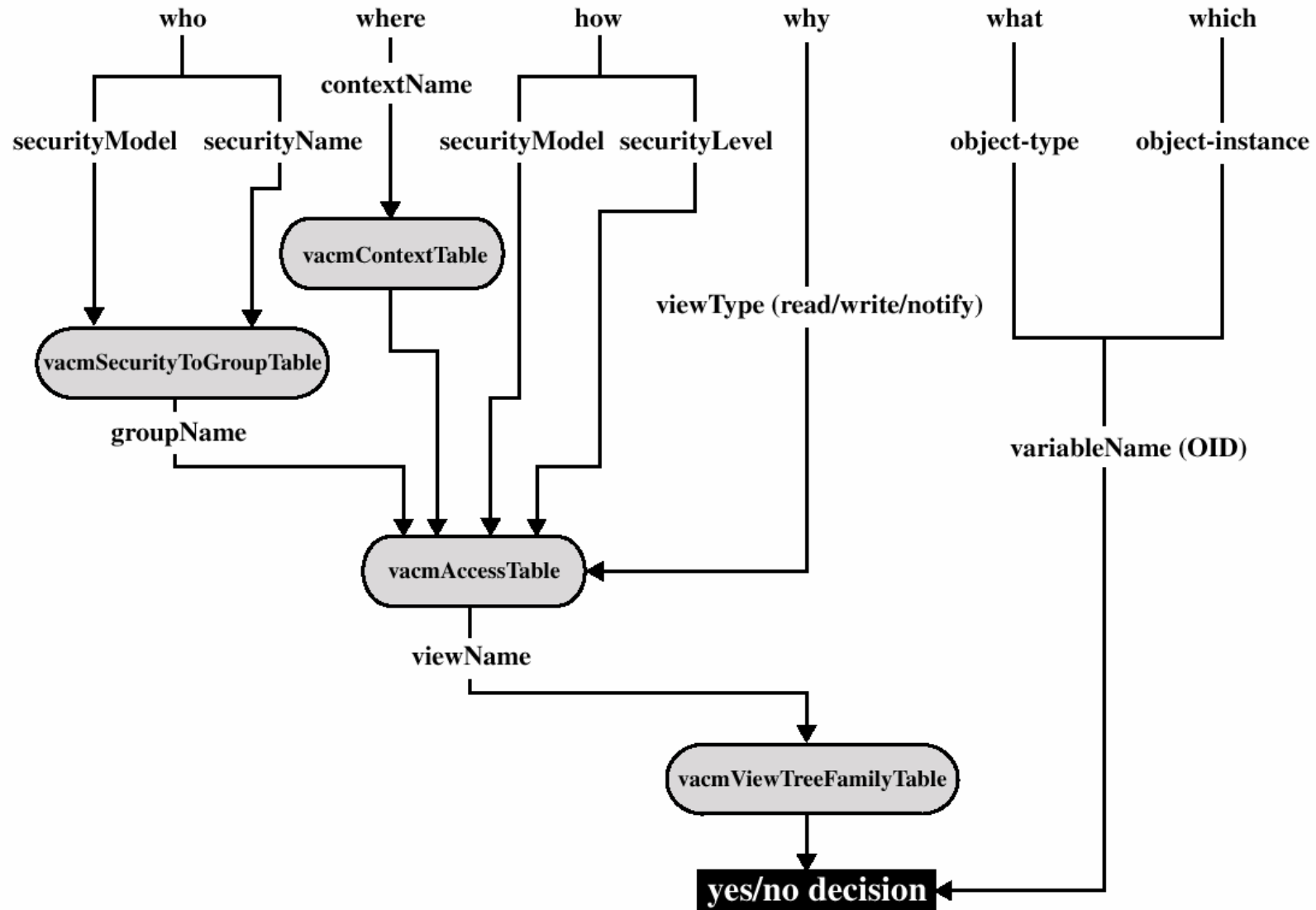
# User Security Model (USM)

- Permette Autenticazione e Cifratura
- Necessità di definire due chiavi:
  - privKey: chiave utilizzata per assicurare riservatezza
  - authKey: chiave utilizzata per l'autenticazione
- Nell'autenticazione sono utilizzati HMAC-MD5-96 e HMAC-SHA-96
- Per la cifratura è usato DES in modalità CBC

# View-Based Access Control Model (VACM)

- Definisce meccanismi per determinare quando una richiesta di accesso ad un oggetto di una MIB locale, effettuata da remoto, deve essere autorizzata
- Motivazioni:
  - Necessità di chiarire le relazioni coinvolte nell'accedere all'informazione di gestione
  - Minimizzare le risorse di memorizzazione e di elaborazione presso l'agente

# Schema logico di VACM



# Intrusioni

- Tre categorie di intrusi:
  - Masquerader: individuo che sfrutta l'account di un utente legittimo
  - Misfeasor (malintensionato): utente legittimo che accede a dati, programmi o risorse per i quali non possiede autorizzazioni di accesso
  - Clandestine: utente che si impadronisce del controllo per la supervisione del sistema e lo utilizza per eludere i meccanismi di controllo dell'accesso e di audit.

# Tipi di Intrusioni

- Accedere al sistema utilizzando semplicemente una coppia {username,password} valida, ottenuta tramite tecniche di tipo tecnologico o meno
- Utilizzo di applicazioni scritte ad-hoc per sfruttare eventuali vulnerabilità e/o creare porte di accesso "secondarie"

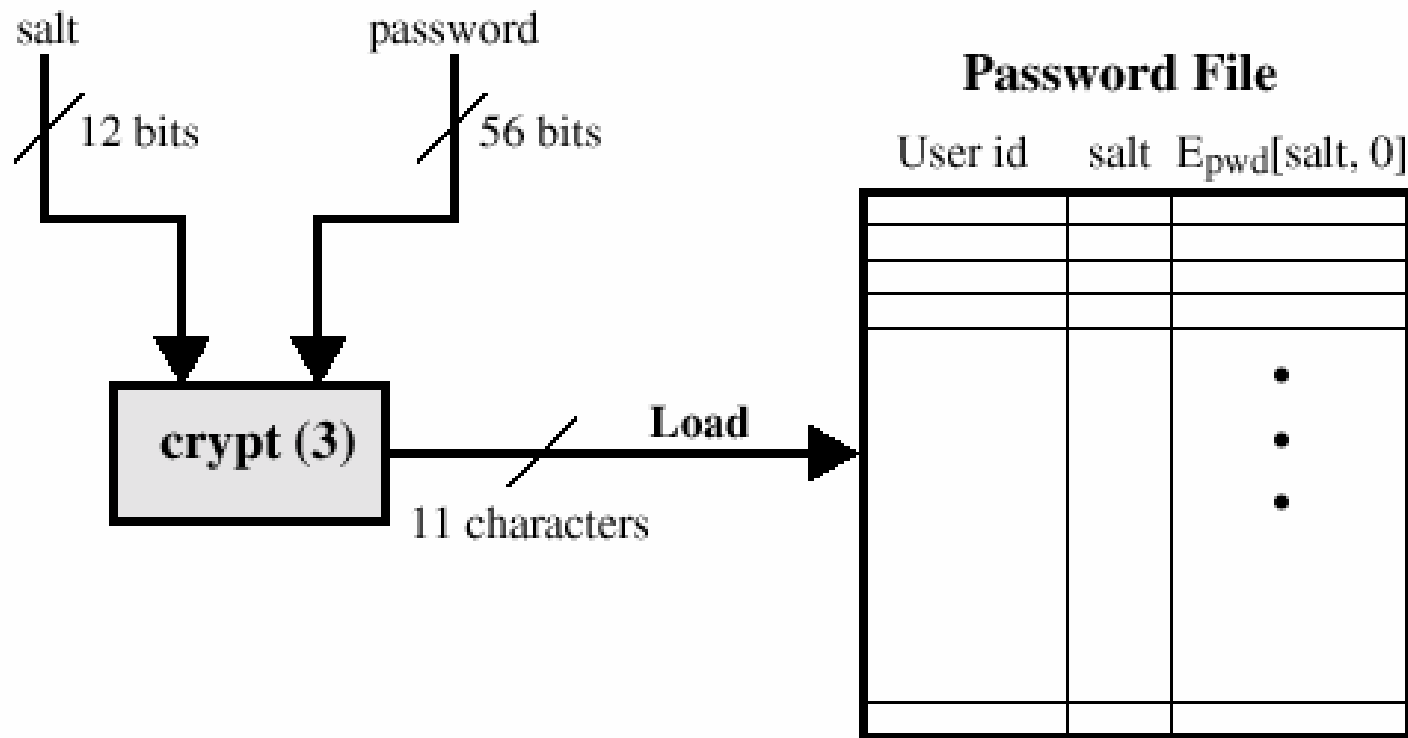
# Scoprire le password

- Il sistema mantiene un file in cui le password sono associate ai rispettivi utenti.
- Il file delle password può essere protetto:
  - Tramite l'utilizzo della cifratura unidirezionale
  - Limitando l'accesso al file delle password ad un solo account (o ad un gruppo ristretto)

# Scoprire le password

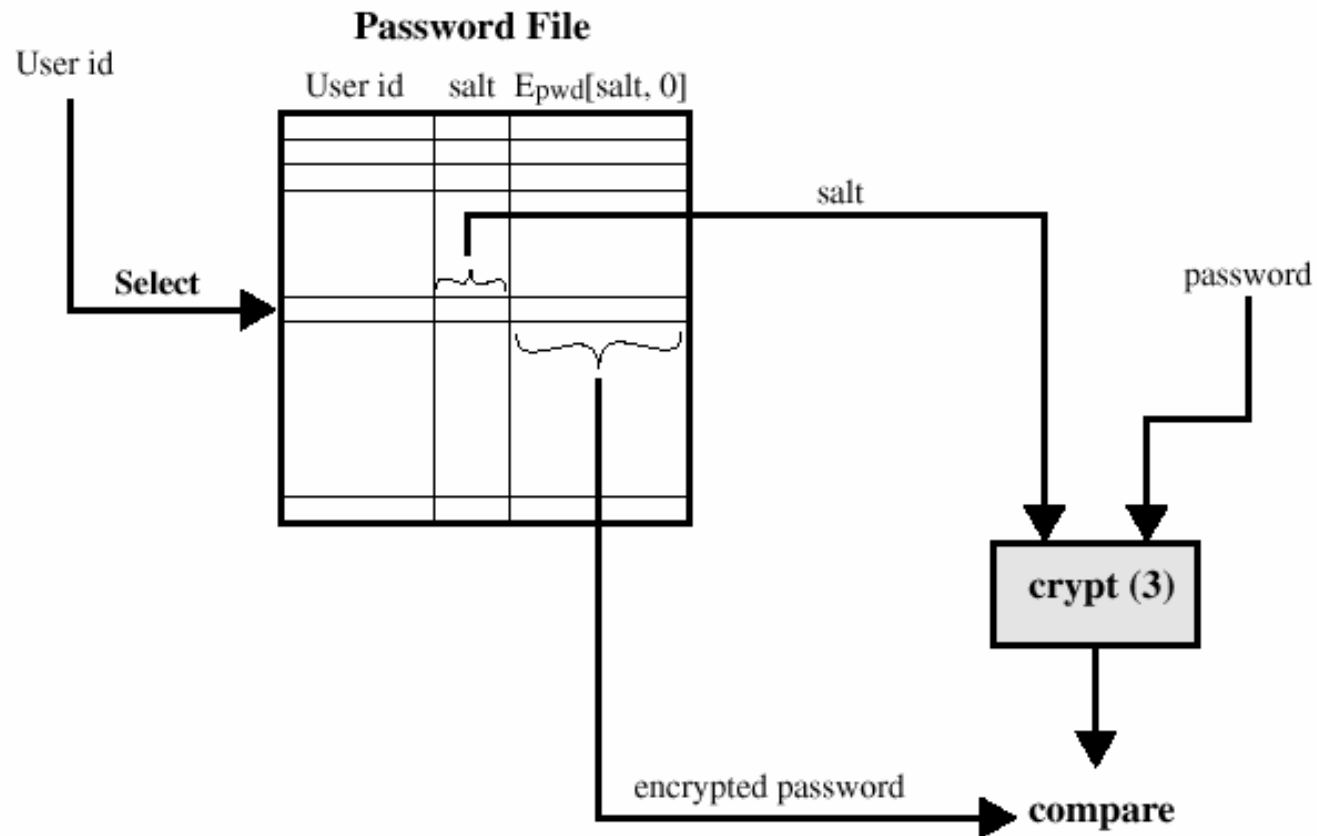
- Principali tecniche per scoprire le password:
  - Provare tutte le password di default.
  - Provare tutte le password di lunghezza limitata (da 1 a 3 caratteri)
  - Provare tutte le parole di un dizionario
  - Raccogliere informazioni sugli utenti, come ad esempio i loro nomi, quelli dei parenti prossimi, libri, hobby etc.
  - Tentare di usare il numero di telefono, il codice fiscale, etc.
  - Provare tutti i numeri di targa legittimi per la provincia
  - Usare un Trojan horse
  - Sfruttare la linea fra un utente remoto e il sistema host
- Prevenzione: Scegliere delle buone password (Ij4Gf4Se%f#)

# Password in UNIX



Inserimento di una nuova password

# Password in UNIX



Verifica della password

# File delle password

- Le password in UNIX erano memorizzate in un file pubblico accessibile dall'utente, tale file di solito è situato nella directory `etc` ed è chiamato: `passwords`
- Oggigiorno il file delle password è memorizzato in una `"shadow"` directory ed è visibile solo dall'utente `"root"`.

# "Salt"

- Il valore salt ha tre scopi:
  - Evitare password duplicate
  - Aumentare la lunghezza delle password
  - Prevenire l'uso di implementazioni hardware di DES per effettuare attacchi alle password
- Nei sistemi attuali particolari valori del salt permettono l'utilizzo di digest MD5 invece che cifratura basata su DES

# Strategie di selezione delle password

- Educazione degli utenti
- Password generate dal computer
- Controllo delle password di tipo reattivo
- Controllo delle password di tipo proattivo

# Sfruttare le vulnerabilità

- E' possibile sfruttare le vulnerabilità degli applicativi per:
  - Ottenere privilegi superiori
  - Permettere l'upload di file
  - Ottenere informazioni che permettono l'accesso al sistema
  - Eseguire codice sulla macchina vittima
- Tra le tipologie di vulnerabilità più pericolose circa la possibilità di eseguire codice sulla macchina remota vi è il Buffer Overflow

# Sfruttare le vulnerabilità

- Un buffer overflow si verifica quando la dimensione dei dati scritti in un buffer supera la lunghezza del buffer stesso
- I dati eccedenti la lunghezza del buffer possono sovrascrivere dati e/o codice. In alcuni casi è possibile che vengano sovrascritti particolari dati o registri utilizzati successivamente in operazioni di salto. Di solito in questo caso si verificano degli errori gravi e l'applicazione deve essere terminata
- E' comunque possibile che i dati da utilizzare nell'attacco di tipo buffer-overflow contengano codice macchina e indirizzi di memoria, codificati in maniera tale da rendere possibile l'esecuzione delle istruzioni sulla macchina vittima

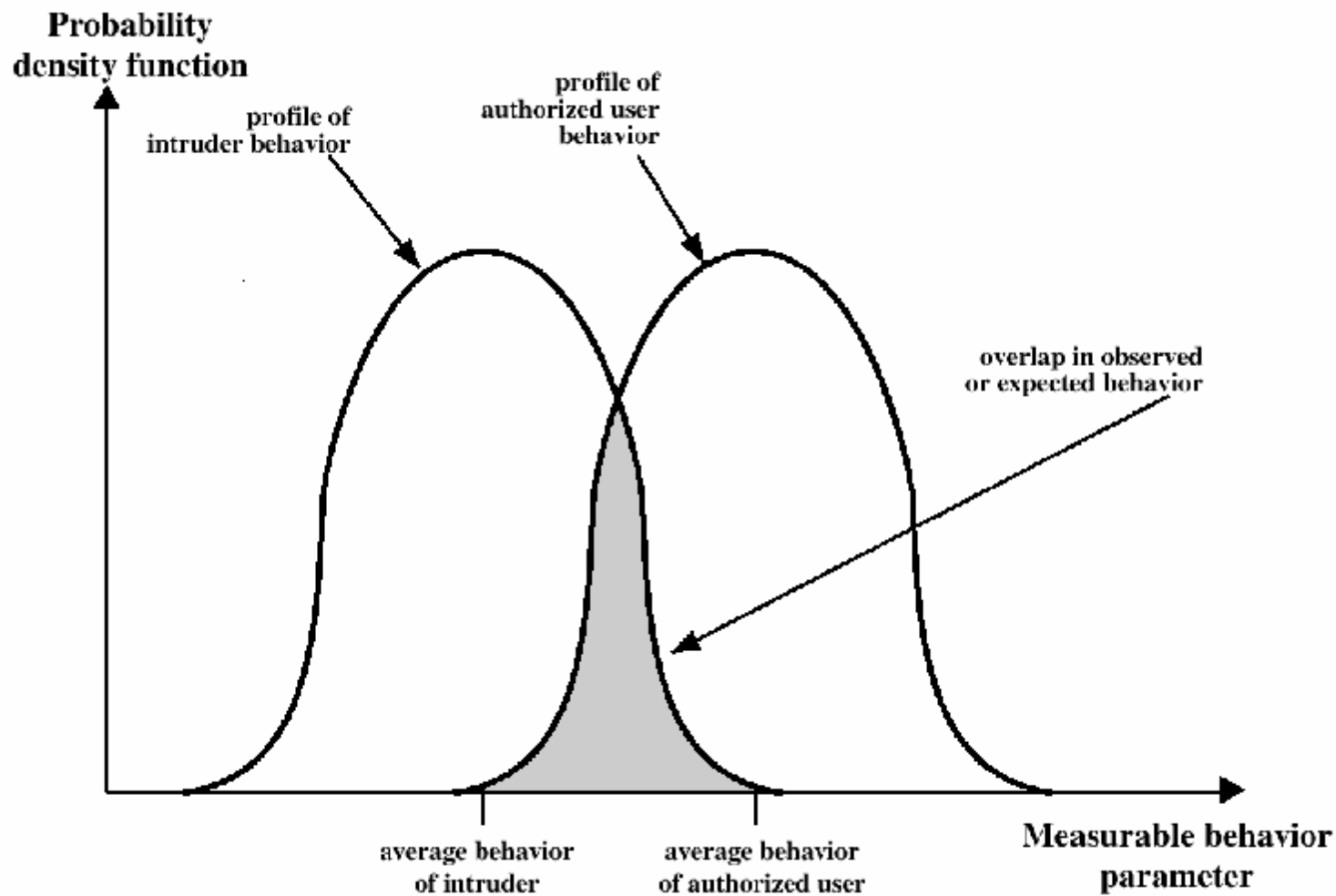
# Fasi di un attacco

1. Scansione della rete per:
  - localizzare quali macchine e relativi IP sono attivi
  - identificare quali sistemi operativi sono in esecuzione
  - scoprire quali porte TCP e UDP sono aperte
  - identificare servizi e demoni attivi (possibilmente anche la versione)
2. Eseguire "Exploit script" contro i servizi vulnerabili (utilizzando le porte aperte)
3. Avere accesso ad una shell con i privilegi di "root"
4. Installare i rootkit, in modo da crearsi una back-door e rendere invisibili le successive azioni
5. Cancellare le proprie tracce dai record di audit

# Intrusion Detection

- L'intruso può essere identificato e escluso dal sistema
- Un efficace sistema di intrusion detection può agire come deterrente e prevenire le intrusioni
- La rilevazione delle intrusioni permette di raccogliere informazioni circa le tecniche utilizzate. Tale conoscenza può quindi essere utilizzata per rendere più sicuro il sistema

# Profili di comportamento per intrusi e utenti autorizzati



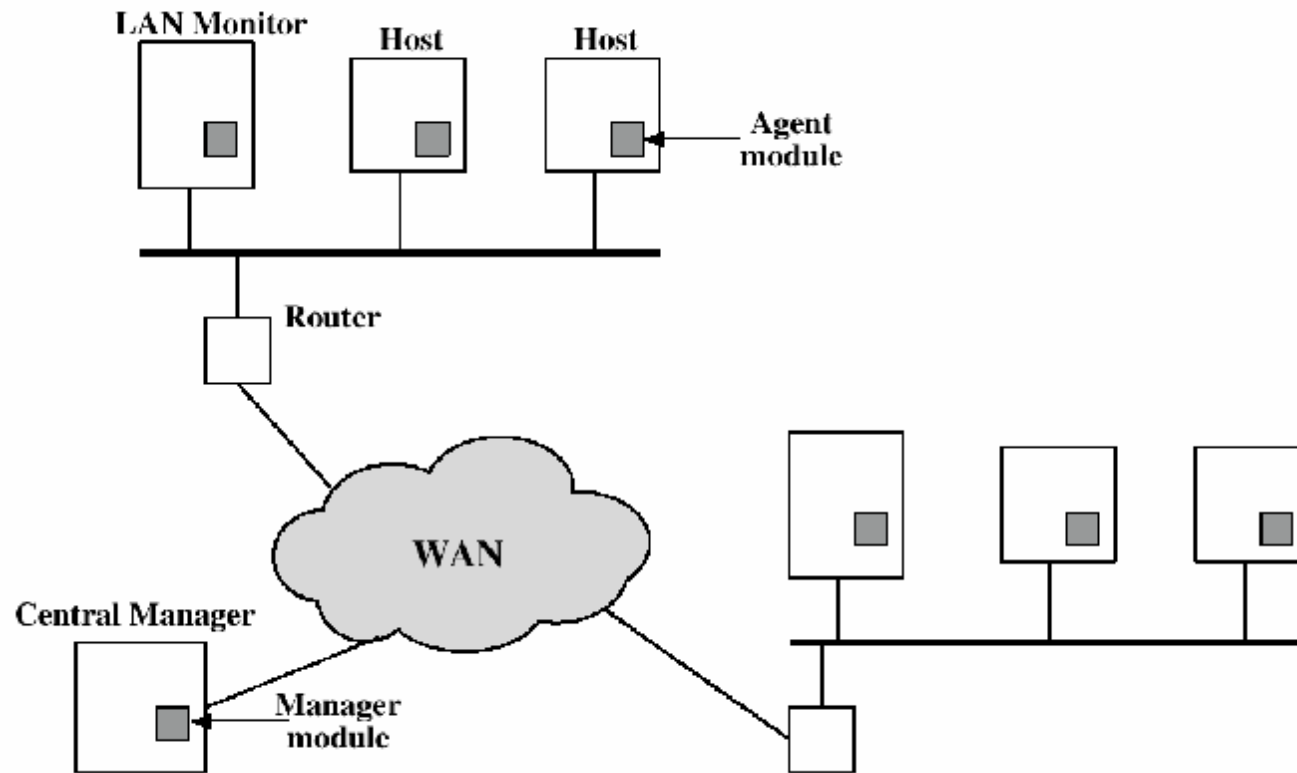
# Tecniche di Intrusion Detection

- Rilevazione statistica delle anomalie
  - Analisi delle soglie (numero delle occorrenze di un evento in un dato intervallo di tempo)
  - Basata sul profilo (caratterizzazione del comportamento degli utenti e rilevazione di deviazioni significative)
- Rilevazione basata su regole
  - Rilevazione delle anomalie (si osservano lo storico delle azioni degli utenti al fine di creare automaticamente delle regole)
  - Rilevazione delle penetrazioni (si basa su sistemi esperti, usa regole per identificare penetrazioni conosciute o sconosciute)

# Indici usati nell'Intrusion Detection

- Frequenza di Login in giorni e ore
- Frequenza di Login in postazioni diverse
- Tempo trascorso dall'ultimo login.
- Numero di "Password failures"
- Frequenza di esecuzione di applicazioni
- Fallimenti di privilegio nell'esecuzione
- Frequenza di Read, write, create e delete
- Numero di fallimenti nelle operazioni di read, write, create e delete.

# Distributed Intrusion Detection



# Distributed Intrusion Detection

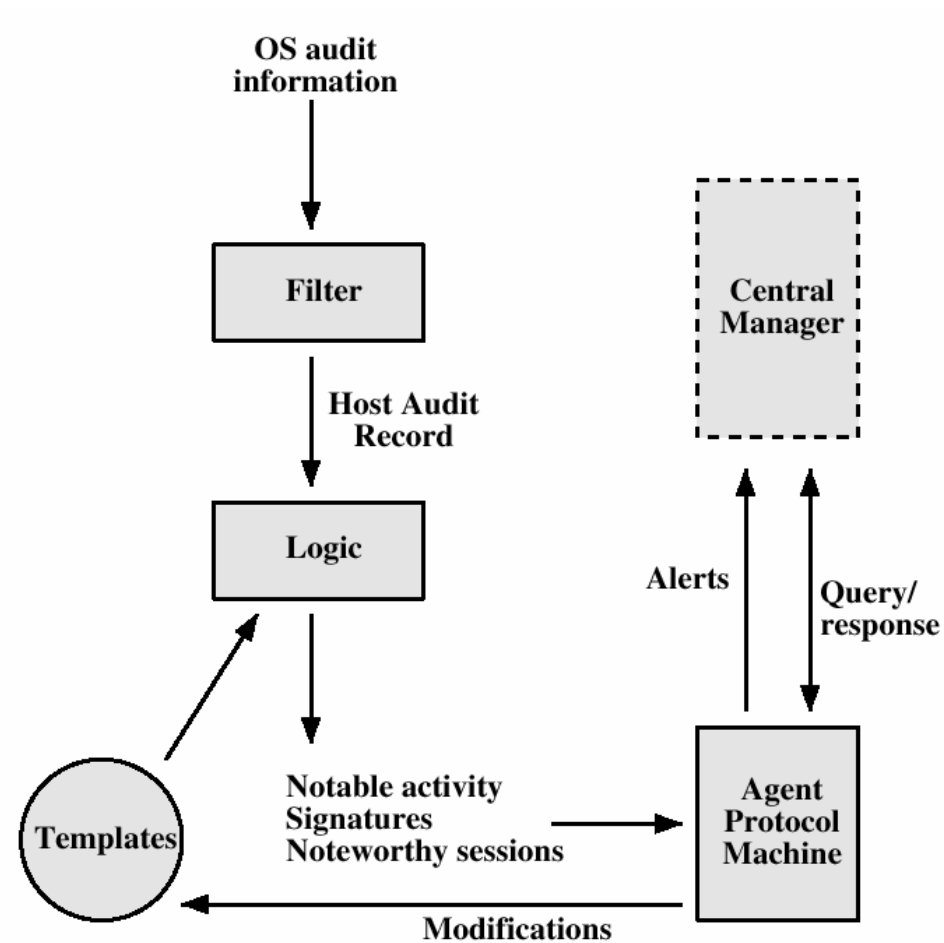


Figure 9.6 Agent Architecture

# Negazione del servizio

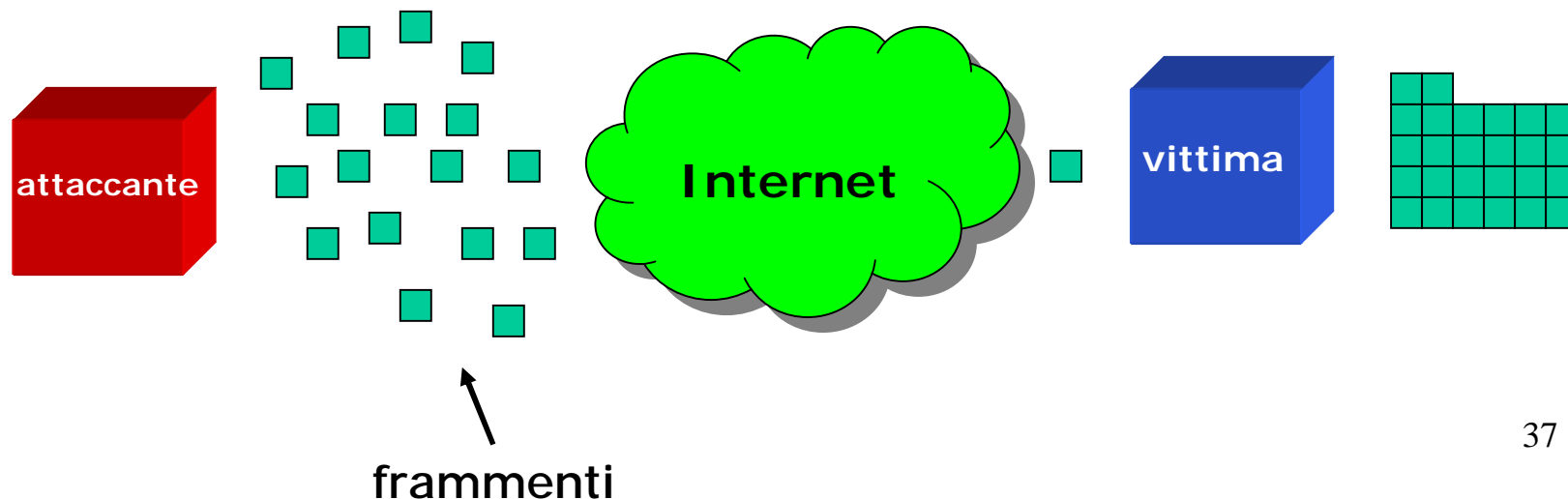
- Attacchi di tipo Denial of Service (DoS):
  - scopo: rendere inutilizzabile un servizio o una risorsa (eventualmente per sostituirsi ad essa).
  - metodo: inibire la connessione al router o al server, provocarne il crash o comunque il blocco
- Distributed DoS (DDoS)
  - attacco di tipo DoS proveniente da più sorgenti contemporaneamente
  - sfruttano molti host compromessi su reti diverse per lanciare attacchi DoS su vittima
- Sempre usate tecniche di IP spoofing
  - molto difficile rintracciare l'origine dell'attacco

# Esempio di DoS: SYN Flood

- Richiesta grande numero di connessioni da host diversi (spoofing), tramite invio pacchetti TCP SYN, senza mai inviare pacchetto di chiusura del "three-way handshake"
  - causa riempimento coda di connessione (può bloccare un router)
  - non è possibile rintracciare origine attacco (mittente falsificato)
  - non è possibile usare access-list (ip sorgente varia in modo casuale)
  - Contromisure: aumentare la dimensione della coda di connessione (SYN ACK queue) e diminuire il tempo di time-out per il "three-way handshake"

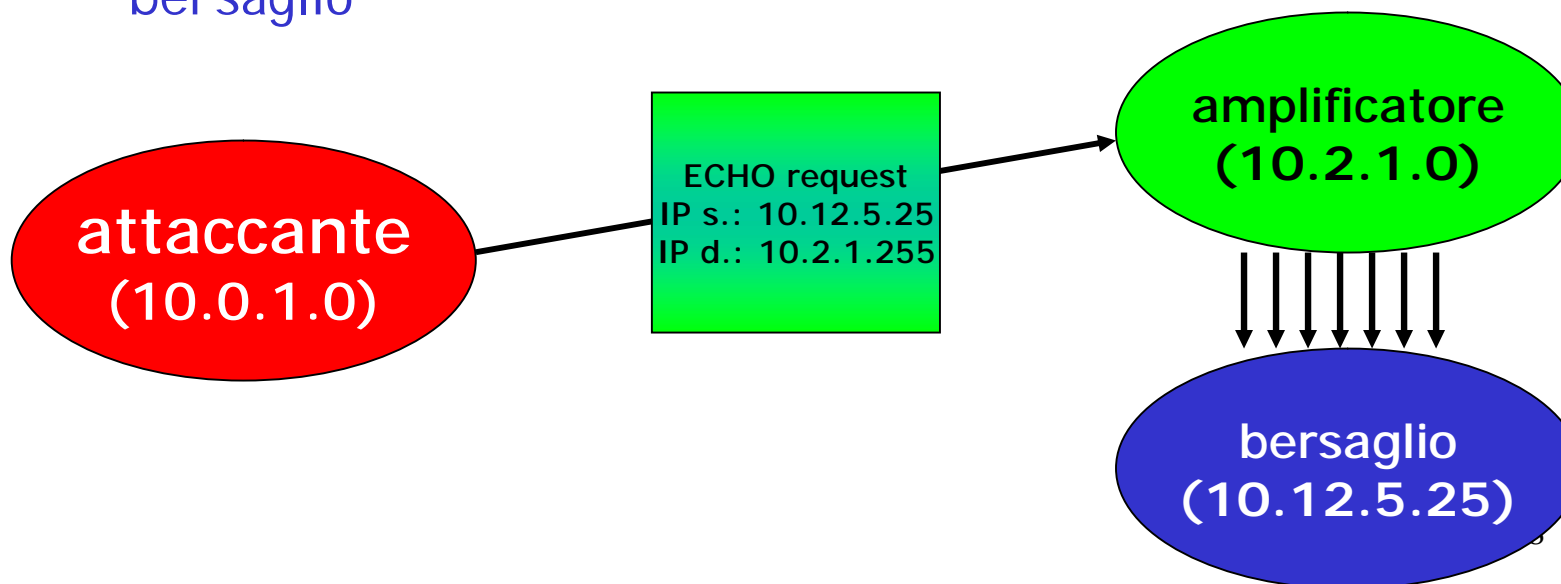
# Esempio di DoS: ping of death

- Invio pacchetti ICMP ping di dimensione maggiore della dimensione massima consentita (65535 bytes)
  - Attaccante frammenta pacchetti
  - Frammenti riassembleti da vittima
  - Ultimo frammento causa overflow



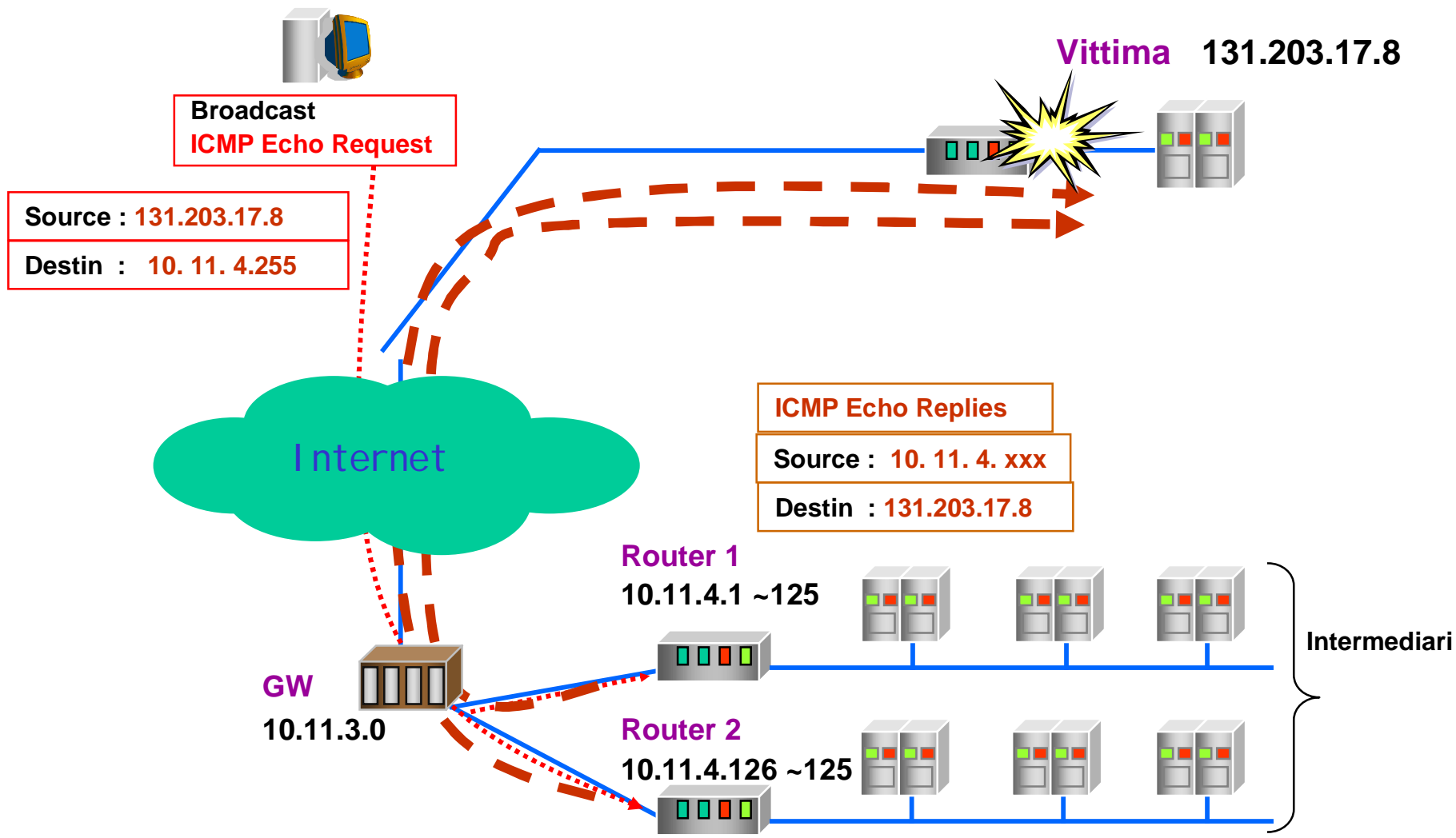
# Esempio di DoS: smurf / fraggle

- Broadcast storm
  - Invio di pacchetti ICMP echo all'indirizzo di broadcast di una rete (fraggle usa pacchetti UDP)
- Coinvolti solitamente almeno tre siti:
  - sito origine dell'attacco ("attaccante")
  - 2 siti vittime, uno intermedio ("amplificatore") ed uno "bersaglio"



# Esempio di DoS: smurf / fraggle

Aggressore 192.100.5.17



# Esempio di DoS: Teardrop

Si utilizza il meccanismo di frammentazione dell'IP

=> Il pacchetto di grandi dimensioni viene diviso in frammenti

=> Ogni frammento è identificato dall'offset al momento del riassetramento

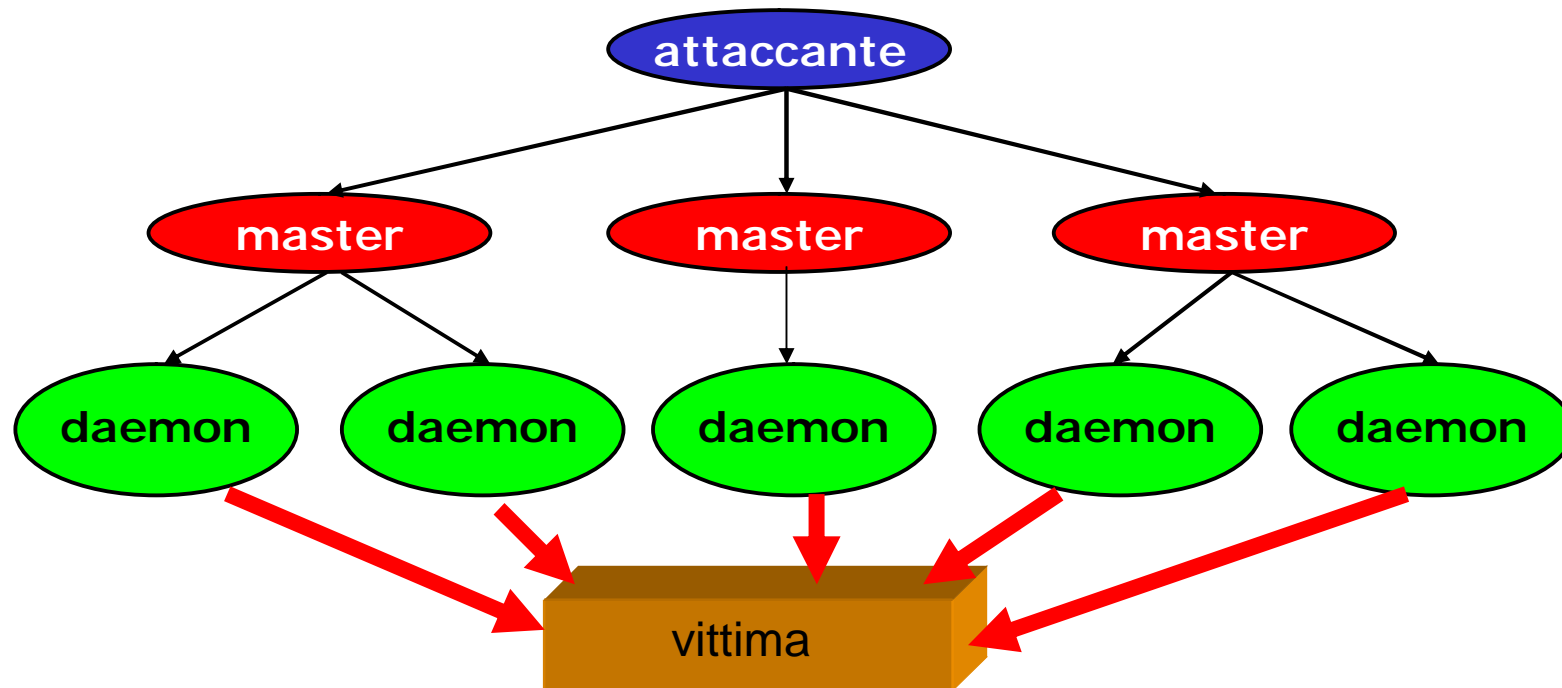
=> L'aggressore inserisce alcuni valori di offset non validi nel secondo frammento e nei successivi

=> Se il sistema operativo non è programmato per fronteggiare questo evento si verifica un crash di sistema

Di solito un semplice riavvio ripristina il sistema

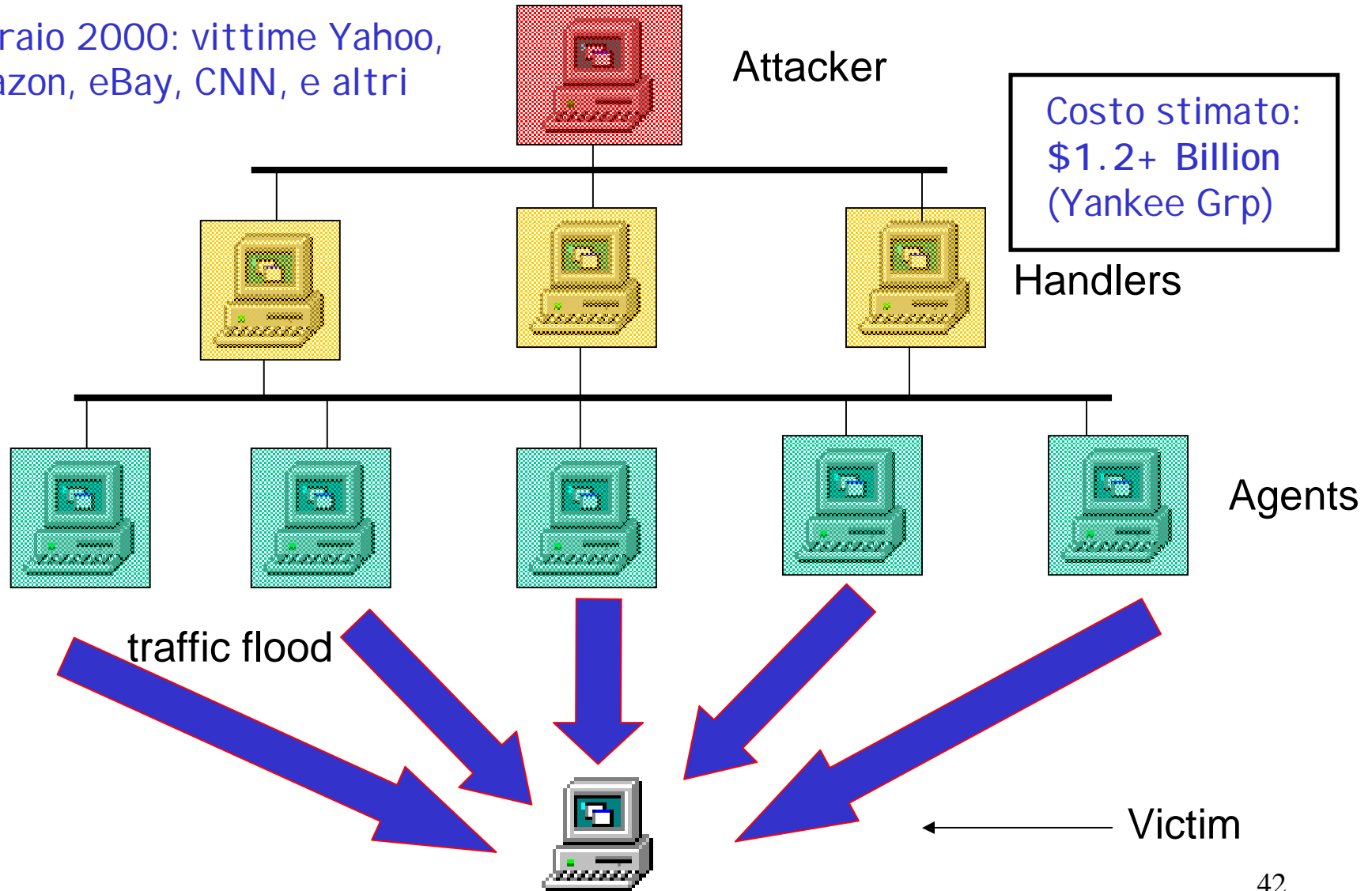
# Distributed DoS (DDoS)

- Attacchi DoS provenienti contemporaneamente da più sorgenti (anche ~ 1000)
- Struttura multi-livello  
attaccante → master → demoni → vittima



# Celebre attacco DDOS

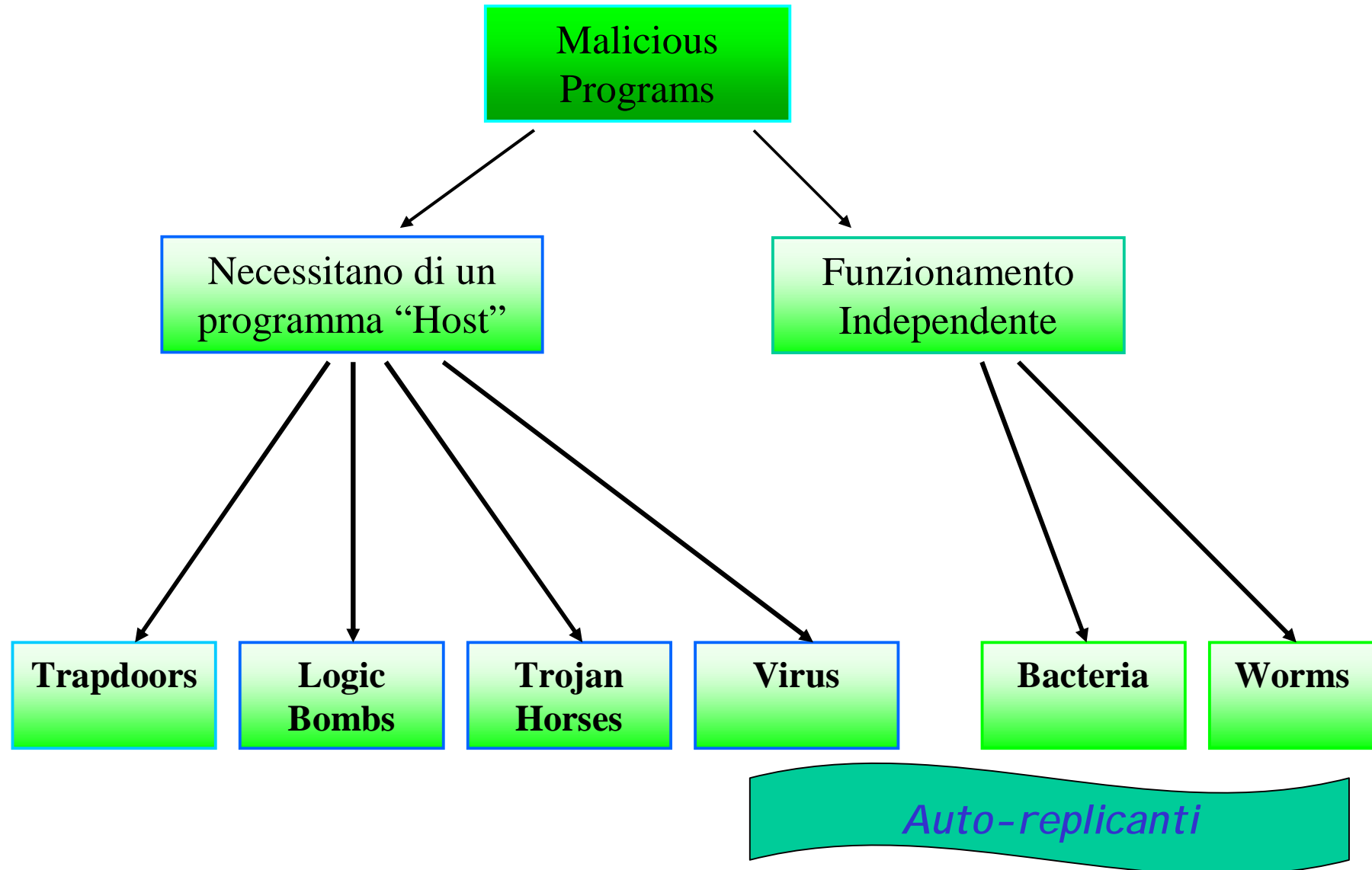
Febbraio 2000: vittime Yahoo, Amazon, eBay, CNN, e altri



# Virus e "Malicious Program"

- I "virus" o programmi simili hanno la funzione di replicarsi su un numero sempre maggiore di computer. Originariamente il metodo più usato per trasmettere "l'infezione" era il floppy disks, oggi essi utilizzano come mezzo primario la rete Internet (Worm)
- Gli altri "Malicious Program" possono essere installati manualmente su una singola macchina oppure essere contenuti in pacchetti software largamente diffusi. Essi sono difficili da rilevare prima che si attivino le loro funzioni nascoste (Trojan Horses, Trap Doors, and Logic Bombs).

# Tassonomia dei "Malicious Program"



# Definizioni

- **Virus** – è del codice contenuto un programma che ha la capacità replicarsi in altri programmi.
- **Bacteria** – semplici programmi il cui scopo è quello di replicarsi il più possibile, con conseguenze che portano alla saturazione delle risorse del computer.
- **Worm** – un programma che utilizza l'infrastruttura di rete per replicarsi (di solito viaggiano come allegati nei messaggi di posta elettronica).

# Definizioni

- Trojan Horse – applicazione che oltre a svolgere la sua funzione “principale”, contiene del codice che permette l’esecuzione funzioni non desiderate o dannose (es. inviare dati e/o password all’aggressore).
- Logic Bomb – codice “malevolo” contenuto in qualche programma legittimo che si attiva al verificarsi di determinati eventi (es. una particolare data, un giorno della settimana).
- Trap Door (or Back Door) – punto di ingresso non documentato che consente di entrare nel sistema senza passare attraverso le normali procedure di controllo dell’accesso

# Fasi di un Virus

- Dormant phase - il virus è inattivo
- Propagation phase - il virus salva la copia di se stesso in un altro programma
- Triggering phase - il virus è attivo e pronto ad effettuare le azioni per cui è stato progettato
- Execution phase - viene eseguita la funzione per cui è stato concepito

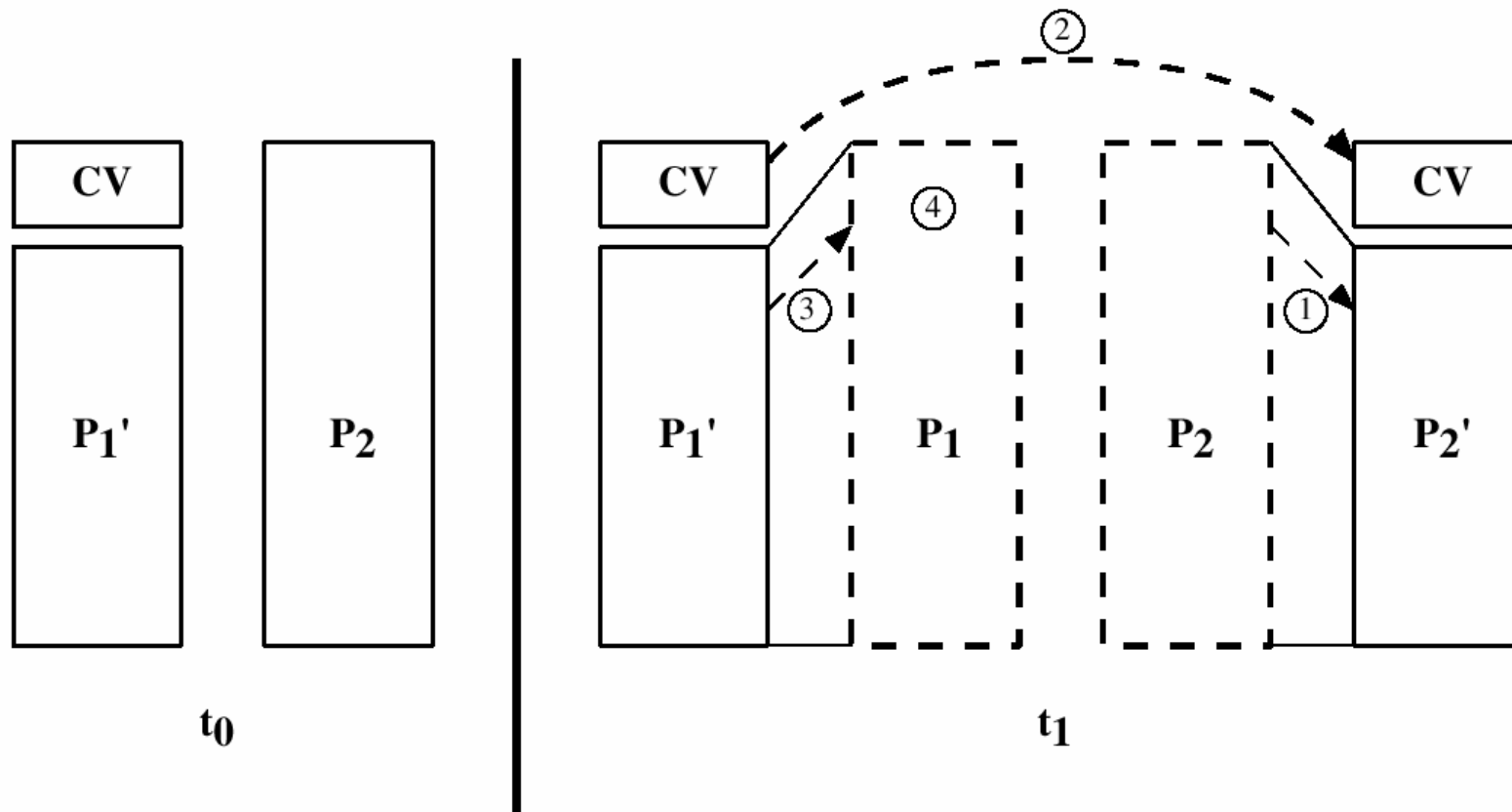
# Protezione dai Virus

- Installare un buon antivirus, attivare la funzione di scansione realtime del sistema e tenerlo costantemente aggiornato.
- Non eseguire programmi o script "macro" di cui non si è sicuri della fonte
- Evitare di utilizzare client email e/o programmi applicativi più comuni e quindi più esposti a rischi

# Struttura di un Virus

```
program V :=  
  
  { goto main;  
    1234567;  
  
    subroutine infect-executable :=  
      { loop:  
        file := get-random-executable-file;  
        if (first-line-of-file = 1234567)  
          then goto loop  
          else prepend V to file; }  
  
    subroutine do-damage :=  
      { whatever damage is to be done }  
  
    subroutine trigger-pulled :=  
      { return true if some condition holds }  
  
main:  main-program :=  
      { infect-executable;  
        if trigger-pulled then do-damage;  
        goto next; }  
  
next:  
  
}
```

# Tecniche di compressione usate nei Virus



# Tipi di Virus

- **Virus Parassiti** – I “classici” virus. Si inseriscono in un file eseguibile e si replicano il più possibile. Vengono eseguiti quando viene eseguito il programma ospitante.
- **Memory-resident Virus** – Una volta eseguiti risiedono in memoria principale ed infettano qualunque programma venga eseguito.
- **Boot Sector Virus** – infettano un record di boot di un disco e vengono eseguiti in seguito al processo di boot.
- **Stealth Virus** – progettati esplicitamente per non essere rilevati da gran parte degli antivirus.
- **Virus Polimorfi** – virus mutante che varia a ogni processo di infezione, rendendo impossibile la sua rilevazione sulla base della signature.

# Macro Virus

- Le applicazioni Microsoft Office permettono l'uso di "macro" all'interno del documento. La macro può essere eseguita durante la fase di caricamento oppure al verificarsi di un evento (es. Salvataggio del file)
- Sono indipendenti dalla piattaforma Hardware

# Tecniche Antivirus

- 1° Generazione - Scanners: ricerca all'interno dei file di una "signature" conosciuta. Controllo dei cambiamenti nella lunghezza dei file.
- 2° Generazione - Heuristic Scanners: ricerca di parti di codice comuni nei virus. Effettuano delle verifiche di integrità dei file tramite l'uso di checksum o hash.
- 3° Generazione - Activity Traps: risiedono in memoria ed identificano i virus in base alle loro azioni. (es. Scansione di un file eseguibile, scrittura in un file eseguibile, etc.).
- 4° Generazione - Full Featured: utilizzano più tecniche antivirus congiuntamente.

# Tecniche Antivirus Avanzate

- I virus che usano tecniche di cifratura polimorfica, sono difficile da identificare
- Generic Decryption (GD): i virus vengono analizzati in un ambiente emulato e "protetto", composto da:
  - Emulatore di CPU
  - Programma di scansione delle signature del virus
  - Modulo di controllo dell'emulazione
- Il problema è quanto tempo deve durare l'emulazione
  - Più tempo passa e più si ha la sicurezza che la routine di decifratura del virus sia stata eseguita
  - Purtroppo l'analisi di un virus non può richiedere troppo tempo altrimenti diventa inutilizzabile

# Tecniche Antivirus Avanzate

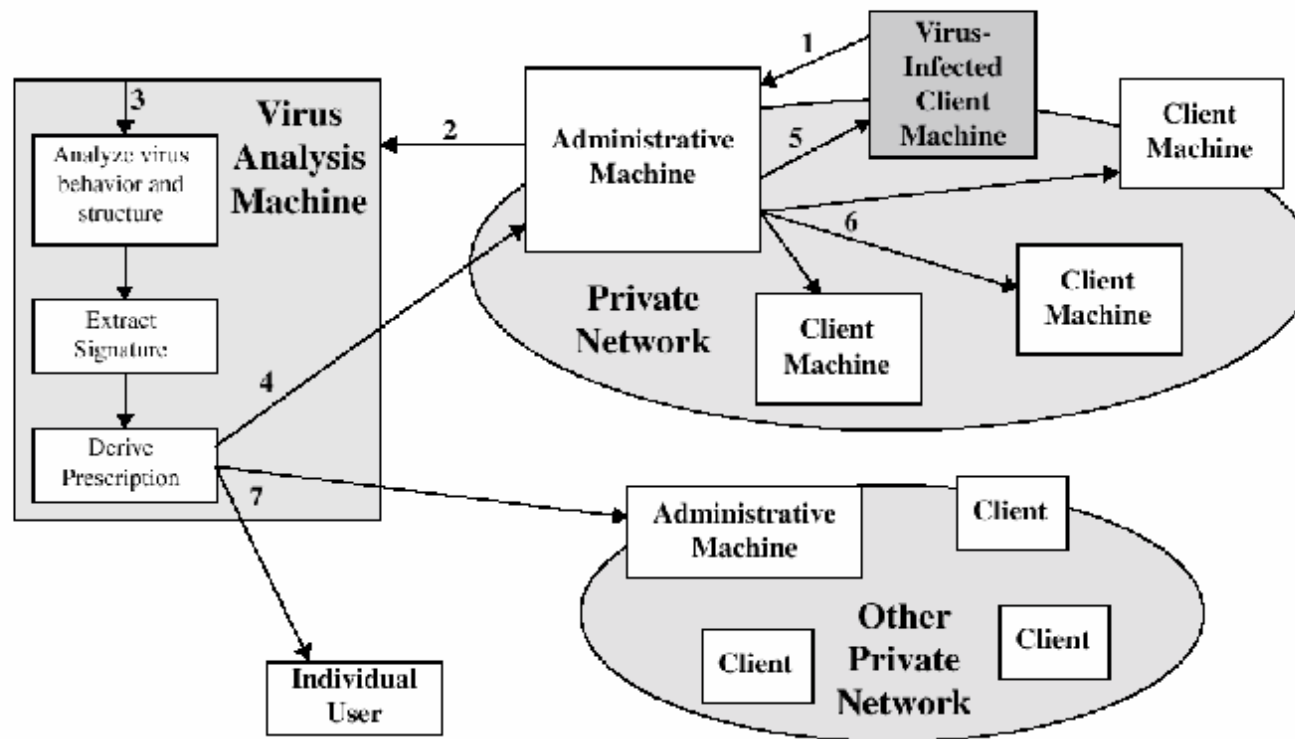


Figure 9.11 Digital Immune System