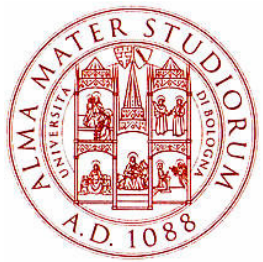
Sicurezza 2006 - 2007

Lezione 4: Attacchi, principi generali

Gabriele D'Angelo
<gda@cs.unibo.it>
http://www.cs.unibo.it/~gdangelo/



Master universitario in
Tecnologia del Software Libero
e Open Source
Università di Bologna

05/10/2007, Bologna

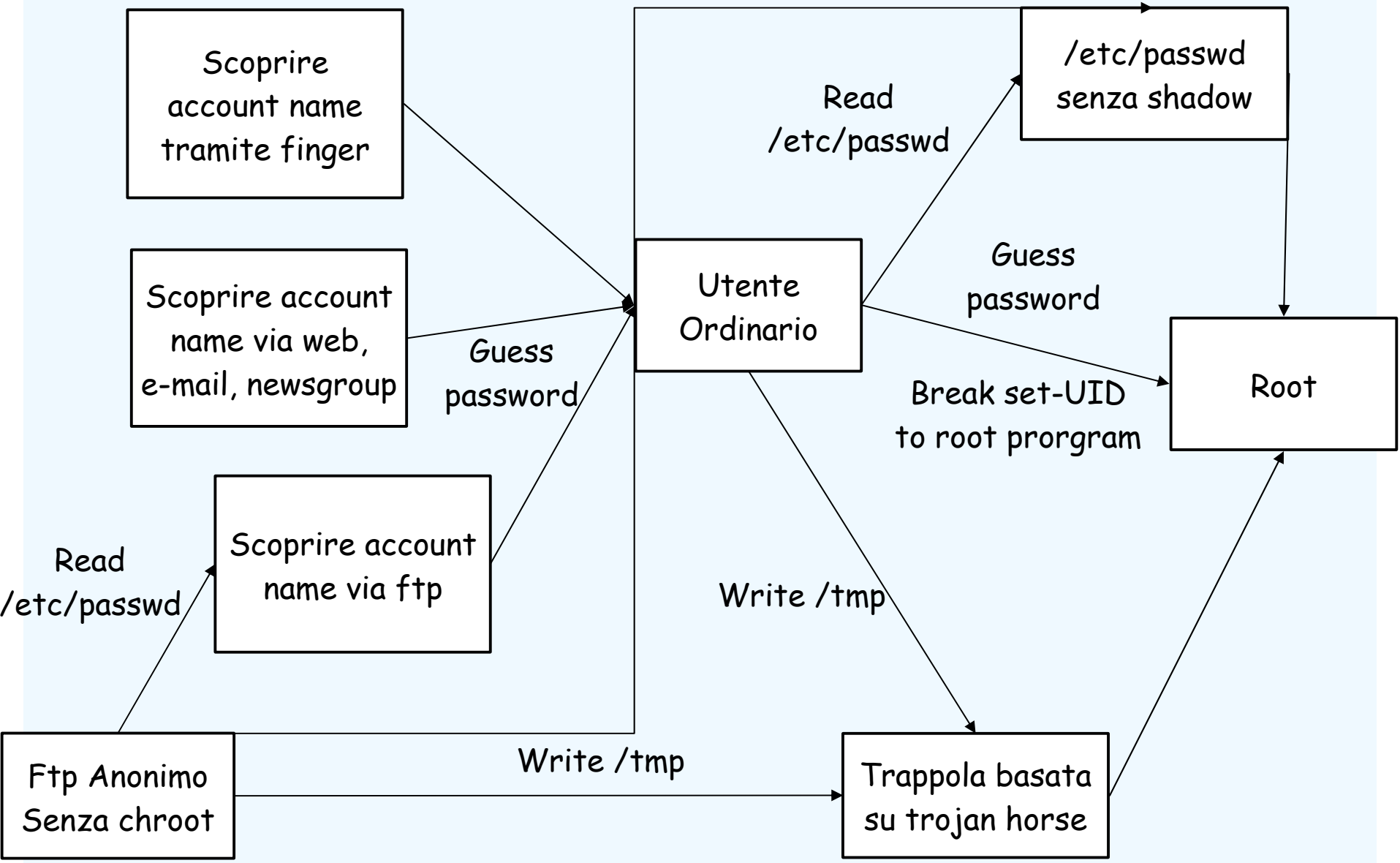
Scaletta della lezione

- Attacchi
- Percorsi di attacco
- Esempio [banale] di attacco
- Alcuni principi sui percorsi di attacco
- Metodi di difesa dagli attacchi
- Protezione dei programmi

Attacchi

- **È possibile fare una prima distinzione tra:**
  - Attacchi "**interni**"
    - derivano da software o utenti che si trovano *"all'interno" del sistema protetto*
    - possono essere la *conseguenza* di un *attacco esterno precedente*
  - Attacchi "**esterni**"
    - sono effettuati attraverso interfacce di comunicazione con l'esterno
    - esempio:
      - tramite moderni protocolli di rete (posta, web, etc.) o direttamente allo stack di comunicazione (es. TCP/IP)
      - tramite (obsoleti?) mezzi hardware (dischetti, cd, etc.)

Percorsi di attacco: un esempio pratico



Esempio [banale] di attacco

- **Individuare il sistema da attaccare**
  - tramite web, perchè si è ricevuto una mail, etc.
  - ad esempio, la ditta yahuu
- **Cercare di carpire qualche informazione**
  - tramite ping verifichiamo se esiste (e se risponde al ping, cosa tutt'altro che scontata)

```
$ ping www.yahuu.com
PING www.yahuu.com (217.12.3.11) 56(84) bytes of data.
64 bytes from www.yahuu.com (217.12.3.11): icmp_seq=1 ttl=242
   time=88.7 ms
64 bytes from www.yahuu.com (217.12.3.11): icmp_seq=2 ttl=242
   time=89.5 ms
```

Esempio [banale] di attacco

- **Cercare di carpire qualche ulteriore informazione**

- Proviamo con finger

```
$ finger root@www.yahuu.com
[www.yahuu.com]
Login: root          Name: root
Directory: /root     Shell: /bin/tsch
Last login Thu Jan  8 00:12 (CET) on tty2
```

- Ottimo! Ora sappiamo molte informazioni e che l'utente root al momento non è connesso
- NOTA: oggi è molto raro trovare finger è abilitato

Esempio [banale] di attacco

- **Il passo successivo è cercare di scoprire qualche nome utente**

- Abbiamo bisogno di un dizionario di nomi
- Possiamo utilizzare finger per scoprire se quel login esiste

```
$ finger dennis
Finger: dennis: no such user

$ finger paul
Login: paul                     Name: Paul Hughes
Directory: /home/paul           Shell: /bin/bash
Office:  789-123456
Last login Wed Jan  7 19:05 (CET) on pts/4 from ...
```

Esempio [banale] di attacco

- **Una volta scoperto il login di un utente**:
  - possiamo utilizzare un meccanismo automatico per tentare di indovinare la password
  - funziona quando la password è “banale”
- Una volta ottenuto un accesso come utente, il passo successivo è **cercare di acquisire privilegi da superutente**
- Probabilmente, indovinare la password di root non avrà successo
- Proviamo a vedere cosa contiene la variabile $PATH di root:

```
cat /root/.bash_profile
export PATH=".:/usr/local/bin:/etc:$PATH“
```

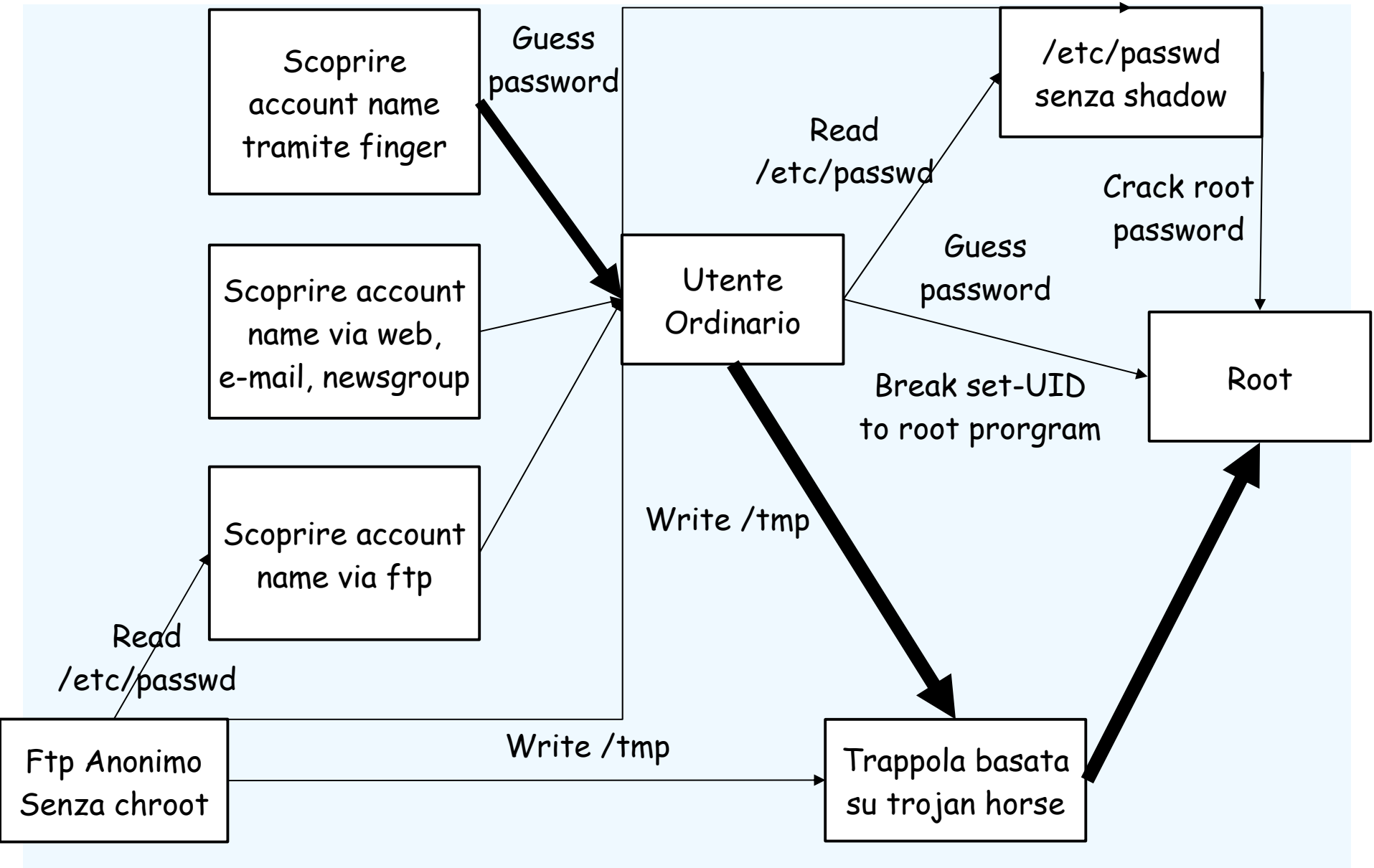
- Ottimo! Il superutente [sprovveduto] permette di eseguire qualunque programma nella directory corrente

Esempio [banale] di attacco

- **Prepariamo una trappola [estremamente semplice] utilizzando la directory /tmp**

- Si crea un file di dimensioni enormi, che in qualche modo **catturi l'attenzione del superutente** (spazio libero sotto soglia di allerta)
- Si installa (in /tmp) un insieme di trojan horse quali ad esempio: ls, vi, more, less etc.
- A causa dell'**errore** nella configurazione del path, le versioni dei programmi eseguite saranno quelle modificate (locali a /tmp) invece di quelle di sistema (che rimangono del tutto inalterate)
- Quando vengono eseguiti, questi programmi (opportunamente modificati) possono eseguire qualunque cosa l'attaccante voglia, in quanto eseguiti con i **privilegi di superutente**

Esempio [banale] di attacco

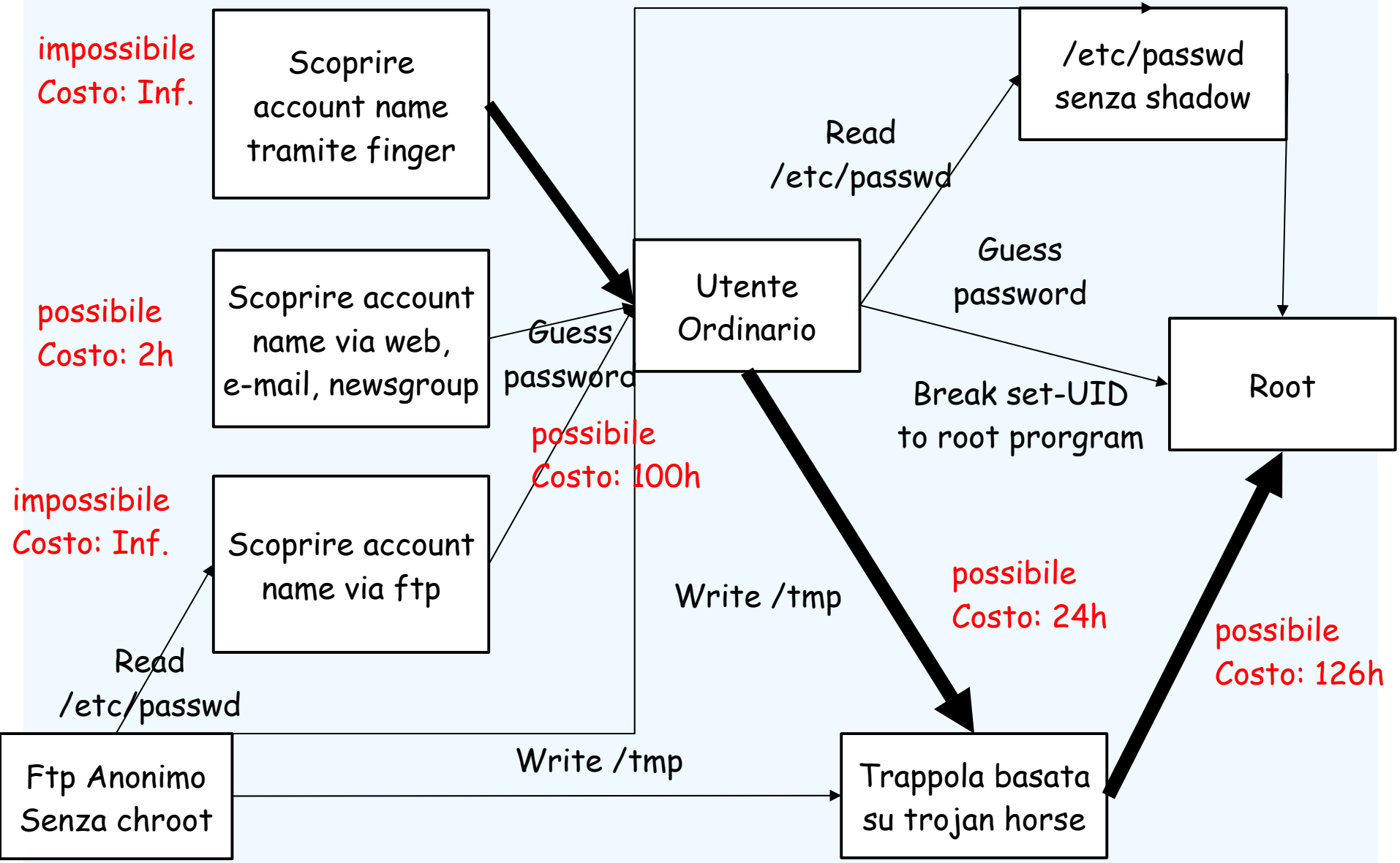


Percorsi di attacco

- **I percorsi di attacco (o attack tree) sono uno strumento molto potente e generale**
  - rappresentano gli attacchi e le contromisure su
    - strutture ad albero
    - grafi diretti aciclici
    - il nodo radice è il goal dell'attacco
  - i nodi foglia sono attacchi

- **Come si usano:**
  - si associano ai nodi foglia delle proprietà
  - si calcolano le proprietà dei nodi interni in base alla proprietà dei loro figli

Percorsi di attacco



Percorsi di attacco

- **Le domande da porsi:**
  - qual è il percorso più semplice (meno costoso) per un cracker?
  - esistono dei percorsi così semplici da rappresentare un rischio concreto ed immediato di intrusione?
  - è possibile eliminare questi percorsi, o renderli più difficili da seguire?
  - è possibile forzare (o invogliare) un cracker a seguire un percorso più complesso e costoso?
- **Ma soprattutto...**
  - sono stati trovati **tutti** i percorsi di attacco?
    - questa è indubbiamente la domanda più difficile visto che richiede una conoscenza completa del sistema da difendere
    - le proprietà associate sono realistiche o derivano da una visione distorta (es. ottimistica) del sistema?

Percorsi di attacco

- **Sono stati trovati tutti i percorsi di attacco?**
- **Un esempio emblematico:**
  - CGI-BIN per la spedizione di mail
  - alcuni di questi CGI-BIN fanno uso del comando mail
  - il comando effettivo viene composto a partire dall'input dell'utente
  - Se
    - il CGI-BIN è mal scritto e non effettua alcun controllo
    - l'attaccante sottomette una stringa del tipo "subject **;** <comando>"
  - il comando risultante sarebbe "mail -s subject ; <comando>", dando accesso [al sistema] all'attaccante (con i privilegi di esecuzione del CGI-BIN)

Alcuni principi sui percorsi di attacco

- **Principio dell'anello debole (weakest link)**
  - nessuna soluzione di sicurezza è più forte del suo anello più debole
- **Principio della protezione adeguata**
  - gli elementi di un sistema informativo devono essere protetti ad un grado coerente con il loro valore
- **Principio di efficacia**
  - i controlli devono essere utilizzati adeguatamente per essere efficaci. Devono essere efficienti, di facile impiego e appropriati

Metodi di difesa dagli attacchi

- **Come gestire un attacco**:
  - **prevenire**
    - cercare di evitarlo completamente
  - **scoraggiare**
    - ridurre le probabilità di successo / aumentare il costo
  - **deviare**
    - rendere altri sistemi più appetibili o rendere questo sistema meno attraente (eventualmente con lo scopo di creare un *honeypot*, ovvero una trappola)
  - **rilevare** (**individuare**)
    - al momento o successivamente
  - **correggere**
    - eliminare gli effetti, ripristinare la situazione precedente (come si fa a ripristinare un sistema che è stato violato?)

Alcune tecniche a disposizione

- **Crittografia**
- **Controllo sul software**
  - Controlli interni di programmi applicativi
    - e.g. gestione dei limiti
  - Controlli del sistema operativo e del sistema di rete
    - proteggere gli utenti da altri utenti
  - Programmi di controllo indipendenti
    - contro specifiche vulnerabilità, come ad esempio anti-virus
  - Controlli di sviluppo
    - implementare software seguendo ben precisi standard di qualità e tecniche di verifica
    - progettazione, codifica, testing e manutenzione

Protezione dei programmi

- **La protezione dei programmi:**
  - è il cuore della sicurezza dei sistemi informativi
- **Due domande fondamentali:**
  - come è possibile (se è possibile) impedire la presenza di difetti nei programmi
  - come è possibile proteggere le risorse da programmi contenenti difetti?
- **Quali tipologie di software:**
  - applicazioni
  - sistemi operativi
  - altri sistemi speciali: sistemi operativi trusted, database

Protezione dei programmi

- **Quali sono i possibili “difetti” del software?**
  - **Errori di programmazione**:
    - violazione dei limiti (buffer overflow)
    - errori di convalida
    - autenticazione inadeguata (che vedremo in seguito)
  - **Controllo degli accessi incompleto:**
    - codice maligno
    - virus, worm
    - cavalli di troia (trojan horse)
    - bombe logiche
    - backdoor

Ringraziamenti

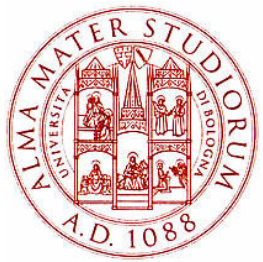
- Gran parte di questi lucidi sono direttamente tratti o “ispirati” da un lavoro precedente del Prof. Renzo Davoli e del Prof. Alberto Montresor, si ringrazia sentitamente

- In ogni caso la responsabilità di qualsiasi eventuale errore rimane integralmente dell’autore di questa lezione

Sicurezza 2006 - 2007

Lezione 4: Attacchi, principi generali

Gabriele D'Angelo
<gda@cs.unibo.it>
http://www.cs.unibo.it/~gdangelo/



Master universitario in
Tecnologia del Software Libero
e Open Source
Università di Bologna

07/09/2007, Bologna