Rootkit: teoria e pratica

Michele BOLOGNA

Corso di Sicurezza dei Sistemi Informatici
Università degli Studi di Bergamo

Che cos'è un rootkit?

- Un **rootkit** è un insieme di programmi che permettono di:

  Consentire la presenza invisibile e permanente di processi e informazioni all'interno di un sistema

- Pericoloso? In base all'utilizzo che se ne fa

Etica

- Prima: dispositivi di emergenza
  - Prendere il controllo di un sistema bloccato

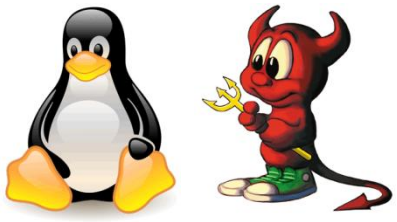
- Oggi: mantenere l'accesso ad un sistema **compromesso**
  - senza l'autorizzazione del legittimo proprietario
  - occultare la propria presenza
  - modificare (o "evadere") le politiche di sicurezza

- Considerati dalla comunità tecnica come malware e trojan horses
  - Non sono worm (non si auto-diffondono)

Dove si trovano?

- Comunità di security
  - http://packetstormsecurity.org/UNIX/penetration/rootkits/indexdate.html
  - Necessità
  - Divertimento, “sfida”
- Nei CD audio (2005)
  - Sony BMG include un rootkit in alcuni CD audio per evitare il ripping
  - Funzionava solo sui sistemi Windows (autoplay)
  - Sony ha ritirato tutti i CD in circolazione

Esempio



- Parleremo di GNU/Linux e UNIX
- Ottengo accesso ad una macchina e riesco ad elevare le mie credenziali a root
  - Exploit locale/remoto
  - Terminali lasciati incustoditi
    - Conoscenze tecniche richieste: minime (script-kiddies)
- Perché non mantenere un accesso segreto a quella macchina?

Esempio

- Per quanto una dir "…" sia difficile da individuare è comunque visibile:

```
michele@titan ~ $ ls -la
drwx------   7 michele shelluser  4096 2008-05-15 15:45 .
drwxr-xr-x 238 root    root       4096 2008-05-13 18:40 ..
drwxr-xr-x   2 michele shelluser  4096 2008-05-15 15:45 ...
```

- Posso nasconderci i miei files, …
  - Ma un amministratore può comunque rintracciarli
    - find, locate, etc.
  - Può notare i miei processi con ps
  - Vedere che sono loggato con who
- **Usiamo un rootkit!**

Funzionalità

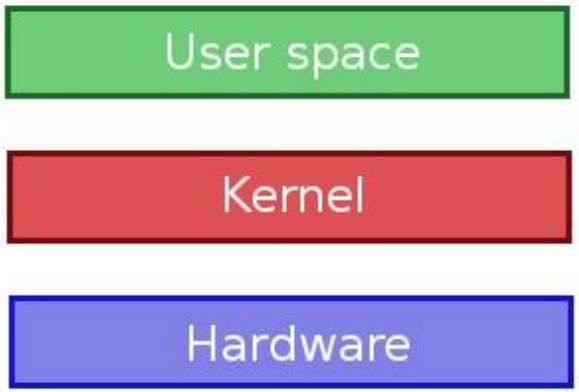
- La maggior parte dei rootkit contiene dei tool per "**diventare invisibile**":
  - Nascondere files, processi, blocchi di memoria agli occhi di **tutti** gli utenti del sistema
  - Fornire una backdoor locale/remota per l'accesso come amministratore
    - Coppia di userid/password di backup (es: utenza "toor", etc.)
    - Esempio: sshd su una porta scelta o via port knocking
- L'installazione del rootkit cancella ogni traccia
  - Tracce di avvenuta compromissione
  - Tracce delle modifiche apportate

Funzionalità /2

- La maggior parte dei rootkit contiene dei tool per:
  - Evitare il logging di alcuni servizi
  - Eliminazione di record dai log (es. user accounting, eliminazione IP)
  - Altri tool per compromettere altre macchine:
    - Sniffing, keylogging
    - Creazione di botnet (DDoS, spam, …)
    - Exploit
- L'installazione è semi-automizzata
  - Purchè il sistema sia conforme alle specifiche del rootkit

Tipi

- Distinguiamo due tipi di rootkit:
  - **User space (USR)**
    - Relativamente facili da scrivere e da identificare
    - Tipo “più vecchio”
  - **Kernel space (KSR)**
    - Complessi da progettare
    - Complessi da identificare
    - Evoluzione



USR: come funzionano?

- Idea di base: modifico il comportamento di ls
  - evitare che mostri le mie directory “nascoste”
- Modifico gli eseguibili di sistema in modo che la presenza della backdoor non sia scoperta
- Bersagli tipici:
  - `ifconfig, netstat`
  - `ls`
  - `ps`
  - `who, w, finger`
  - `login, tcpd, inetd`
  - `Ogni demone di rete`

KSR: come funzionano?

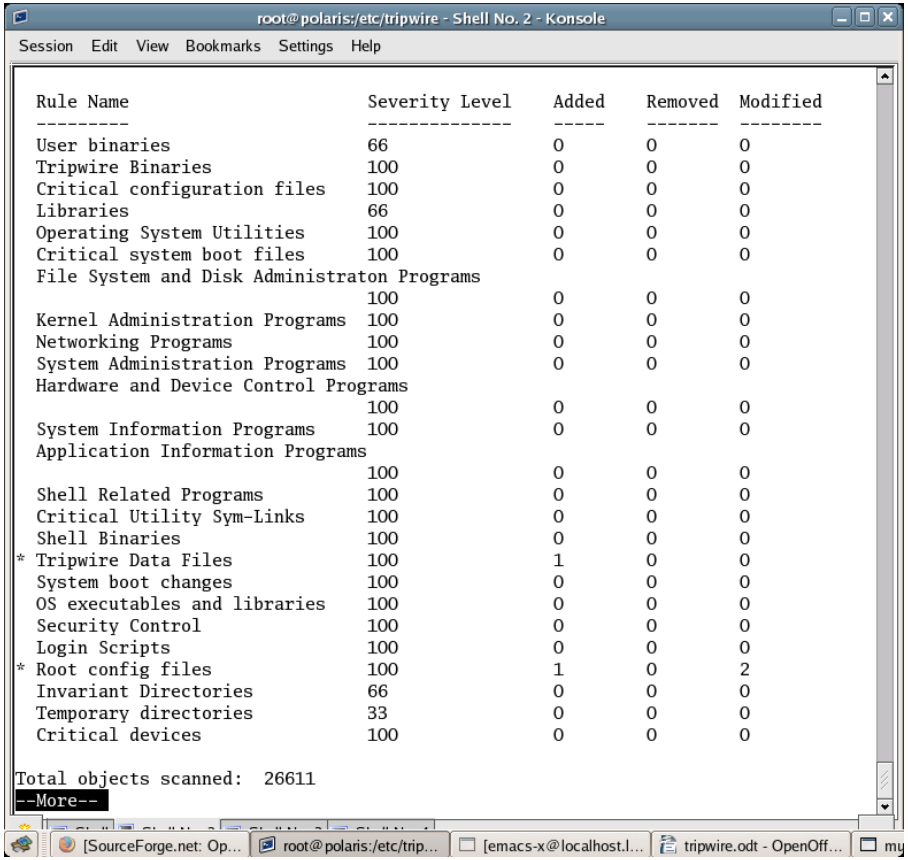
- Non sostituiscono gli eseguibili di sistema
- Sono dei moduli del kernel
  - Loadable Kernel Module (LKM)
  - Possono essere caricati a runtime (es: device driver)
- Complessi: operano allo stesso livello del kernel, quindi possono modificare o dirottare qualsiasi richiesta software

KSR: come funzionano? /2

- Aggiungono e/o modificano parti del SO
- Alterano il funzionamento dei comandi attraverso il kernel stesso
  - Accedono al kernel tramite la memoria (/dev/kmem)
  - Reindirizzano le chiamate di sistema a funzioni trojan (prima il mio codice, poi la chiamata)

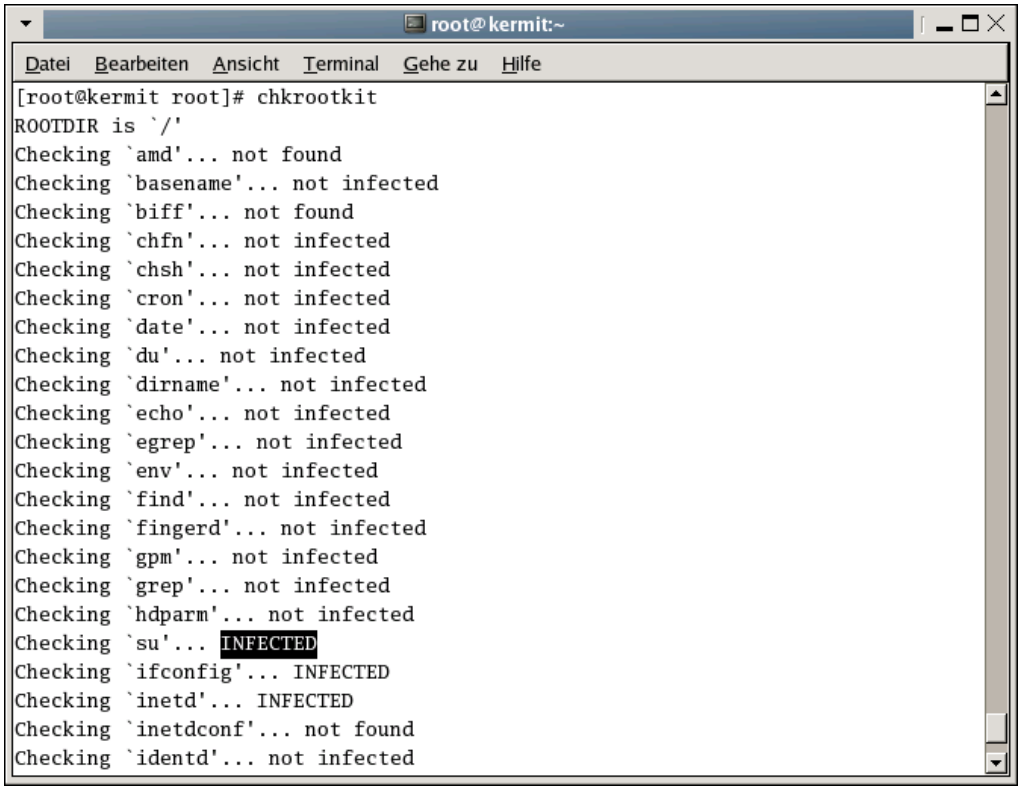
Come ci si difende? (USR)

- Fingerprinting:
  - Salvare i fingerprint dei binari di sistema su un supporto read-only
  - Confronto con hash attuali
  - Differenze?
- Processo **automatizzato** con alcuni tool (es. tripwire, integrit, etc.)



Come ci si difende? (USR/KSR)

- Ispezione: strumenti automatici che cercano **pattern** sospetti nei binari di sistema (es. chkrootkit)
  - Falsi positivi
  - Rootkit **già** noti
  - E se io scrivessi il mio rootkit?



Come ci si difende? (KSR)

- Controllare quali moduli (LKM) vengono caricati
  - LKM è il punto d'ingresso per iniettare codice nel kernel
  - Esistono moduli che non vengono visualizzati con lsmod
- Ispezionare il device /dev/kmem (usando kstat)
  - Attraverso /dev/kmem è possibile iniettare codice nel kernel a runtime
  - Controllare l'accesso in scrittura a /dev/kmem (via kernel)
    - Ad esempio utilizzando una whitelist per i programmi

Come ci si difende?

- **Processo** di sicurezza
  - Prevenzione
    - Security patch
    - …
  - Attività sospetta
    - Porte in LISTENING (netstat/lsof/nmap)
    - Segfault dei binari
    - Payload dei pacchetti in entrata/uscita
      - telnet ICMP (!)
      - Reverse shell (callback)
    - Modalità promisc della scheda di rete
    - Controllo dei log
    - Carico elevato
  - Osservare il sistema!

…

Cosa fare se si è infetti?

- Assunzione di base: se il SO è compromesso non ci si può fidare
  - meglio un OS live da CD, ad es. Knoppix forensics, Helix, etc.
- Forensics
- Ripristino
- Reinstallazione totale

Forensics

- Può essere interessante (ma varia da contesto a contesto) un'analisi forense per stabilire:
  - Come ha preso il controllo del sistema?
  - Attribuire responsabilità
  - Verificare se l'attaccante ha lasciato qualche traccia (log) utile
  - Identificare l'attaccante (provvedimenti?)
  - Evitare di commettere lo stesso errore in futuro
  - Rivedere la propria politica di sicurezza

Ripristino e reinstallazione

- Ripristino
  - Ripristinare le componenti compromesse
    - Sono state fatte copie di backup?
  - Sconsigliabile
    - Il sistema potrebbe contenere componenti compromesse non visibile: analisi?
      - Analisi che può richiedere molto tempo
- Reinstallazione
  - Nel caso sia possibile è fortemente consigliata una reinstallazione totale del sistema

Conclusioni

- Domande?

- Dove posso trovare materiale a riguardo e i riferimenti ai tool citati?
  - Queste slides (e i relativi riferimenti) sono disponibili su: http://mb.netsons.org