SICUREZZA SU RETI

Manuale d'uso

Syslog-NG

MANUALE D'USO

Syslog-NG

Corso di Sicurezza Su Reti
Prof. Alfredo De Santis

Università degli Studi di Salerno
Facoltà di Scienze Matematiche, Fisiche e Naturali
Anno Accademico 2003/2004

Francesco Di Rienzo – fradir@tiscali.it
Gianluca Esposito – giaesp@tiscali.it
Feliciano Nigro – felnig@tiscali.it

Sommario

Capitolo

1

Introduzione a Syslog-NG

Negli ultimi anni si è visto un aumento di hackers, crackers e script kiddies dovuto ad alcuni fattori:
• la diffusione di internet;
• l'aumento della velocità di download e upload;
• costi di gestione per la connessione ridotti;
• aumento del livello medio delle conoscenze informatiche.

Quindi chi si occupa della sicurezza di una rete o dell'amministrazione dei server deve tener traccia di :
• tentate intrusioni;
• intrusioni andata a buon fine;
• modifiche fatte dall'attacker al sistema violato.

Il tracciamento dell'attività avviene tramite un analisi approfondita dei log di sistema..
I log (file che contengono informazioni sullo stato della macchina) di sistema contengono molto "rumore" - messaggi di nessuna importanza - ma contengono anche eventi importanti, che non dovrebbero andare persi nella mole dei messaggi.
Quindi bisogna essere sicuri che i log siano validi e non siano stati modificati in alcun modo. In realtà quello che si vuole è una copia che sia sempre valida dei log, anche se venissero modificati localmente, cosa che di solito un attacker fa, in caso di intrusioni andate a buon fine, con lo scopo di rimanere nascosti all'amministratore della macchina.

L'utilizzo di un log server , ovvero una macchina che ha il solo e preciso compito di raccogliere i log di altre macchine, consente di avere una copia sicuramente valida dei log delle macchine che lo utilizzano.

In tal modo, anche se l'attacker può modificare i log sulla macchina locale, ci sarà una copia non modificabile dei log sulla macchina remota, che non può essere (o meglio, non dovrebbe) essere raggiunta dall'attacker.
Inoltre un sistema sicuro di remote logging, può migliorare e semplificare notevolmente la gestione centralizzata dei log di diverse macchine e anche l'analisi di eventuali intrusioni o tentativi di intrusione.

Avendo un unico centro di raccolta dei log, diventa più semplice:
• analisi real-time dei log
• report periodico delle attività delle macchine
• archiviazione e backup dei log

Un setup di logging sicuro, che ci consente di migliorare la protezione contro attacchi esterni, dovrebbe avere le seguenti caratteristiche:

• Tutte le macchine da monitorare devono inviare i log ad una macchina remota per semplificare l'amministrazione e il controllo delle macchine e rendere vani i tentativi di un attacker di cancellare i log sulla macchina locale.

• Inviare i log in maniera crittografata, poiché, avendo una struttura centralizzata, è necessario evitare che i log possano essere intercettati, perché potrebbero fornire troppe informazioni utili ad un attacker.

• Il log server deve essere molto sicuro, poiché, dovrebbe essere accessibile solo da console e in nessun caso dovrebbe avere servizi aperti verso l'esterno.

• I log sul log server devono essere organizzati. Una buona organizzazione consente di minimizzare il tempo di controllo dei log.

• Non si deve rischiare di perdere messaggi.

• La presenza di un log server remoto dovrebbe non essere visibile ad un attacker in modo tale da poter continuare a monitorarlo fino a decisione contraria.

Per creare un secure remote log server può essere utilizzato Syslog-NG.

Syslog-NG è uno strumento di tracciamento degli eventi di sistema prodotto da Balabit (www.balabit.com). Esso fornisce una gestione centralizzata dei log di tutte le device della rete sulla quale agisce, indipendentemente dalle piattaforme presenti. Inoltre fornisce diversi caratteristiche aggiuntive, inclusi filtri basati sul contenuto dei messaggi di log, gestione personalizzabile della memorizzazione delle informazioni raccolte e diverse capacità di analisi delle stesse.

In ambito UNIX una delle aree più trascurate è la gestione del tracciamento degli eventi di sistema. Controllare giornalmente i messaggi di sistema è fondamentale per mantenere buone le condizioni di sicurezza e di salute del sistema.

Uno dei principi di progettazione di Syslog-NG è stato quello di rendere il filtraggio dei messaggi molto più potente e sottile. Syslog-NG è capace di filtrare messaggi sulla base del loro contenuto, oltre che in base alla coppia opzione/priorità. In tal modo solo il messaggio che ci interessa veramente giungerà ad una specifica destinazione. Un altro principio di progettazione è stato quello di rendere più facile il trasferimento di log tra segmenti di rete protetti da firewall: il lungo formato del nome dell'host permette di risalire facilmente alla macchina origine e alla catena di computer che ha trasportato il

messaggio, anche se questo ha attraversato parecchi computer. L'ultimo principio è stato quello di scegliere un formato per il file di configurazione chiaro e potente.

PRESENTAZIONE DEL MANUALE

Questo manuale è rivolto a tutti coloro che hanno tentato di reperire informazioni su Syslog-NG e non ci sono riusciti. Per questo tentiamo di fornire una panoramica che consenta al lettore di poter sfruttare in maniera efficace le principali funzionalità offerte da questo prodotto.

Il manuale, suddiviso in quattro capitoli, compresa l'introduzione, presenta informazioni sull'installazione, sulla configurazione e sull'utilizzo di Syslog-NG.

Il secondo capitolo copre i fondamenti dell'installazione di Syslog-NG; nel terzo capitolo si presenta la configurazione del programma ed un esempio di utilizzo; nel quarto ed ultimo capitolo si presentano le tipologie di attacco ad una rete, le vulnerabilità di una rete e la progettazione dell'infrastruttura di rete, calibrandone ogni singolo componente e, controllandone l'attività con Syslog-NG.

Nel manuale è presente anche un'appendice contenente informazioni sui principali servizi e loro livello di vulnerabilità ed informazioni sull'obbligo di manutenzione dei log.

Capitolo

2

Installazione e set-up

I passi necessari alla compilazione, installazione e configurazione di Syslog-NG: entriamo in contatto con il software attraverso esempi pratici e nozioni teoriche.

Requisiti minimi di sistema

I requisiti minimi di sistema per il funzionamento di Syslog.NG consigliati dalla casa produttrice sono:

Processore Intel o compatibile (400Mhz)
Memoria RAM almeno 256 Mb
Sistema Operativo: Linux con Kernel 2.3.2.

Per effettuare tutti i nostri test, tuttavia, abbiamo deciso di non eseguire Syslog in un ambiente dedicato, bensì emulato. Tale scelta è motivata anche dal fatto che sebbene *la procedura di compilazione e d'installazione descritta più avanti sia comune sia all'esecuzione client che server,* le risorse necessarie per effettuare tutti i nostri test (vedi capitolo 3) richiedono l'installazione di tale software su almeno quattro personal computer. Tale scelta è troppo onerosa rispetto ad un ambiente totalmente simulato che consente l'esecuzione contemporanea di ben più di quattro sistemi differenti sulla medesima macchina.

Macchina di prova

Processore AMD Athlon XP 2800+ (2.08 GHz)
RAM 256 Mb
Sistema Operativo:
SuSE Linux 8.1 – Kernel 2.4.19-4GB emulato mediante Connectix VirtualPC 5.4 (build 418) in ambiente Microsoft Windows XP Professional[1].

[1] Data la mancanza di una macchina Linux con cui testare Syslog-NG si è scelto di utilizzare un emulatore. Per creare un log server è bene eseguire un'installazione "tradizionale".

Il primo passo da compiere per installare Syslog-NG è ovviamente ottenere il pacchetto di installazione: è possibile effettuarne il download all'indirizzo web http://www.balabit.hu/en/downloads/Syslog-NG/downloads/

Nel nostro caso utilizzeremo la versione 1.4.17 ed il file sarà **Syslog-NG-1.4.17.tar.gz** (sebbene Syslog-NG sia giunto alla versione 1.9, la nostra scelta è ricaduta sulla versione 1.4.17, ultima dichiarata "stable" dalla casa produttrice stessa). Estraiamo quindi l'archivio compresso mediante il comando, ricordando che tutti i comandi descritti devono essere lanciati dall'utente root[2]:

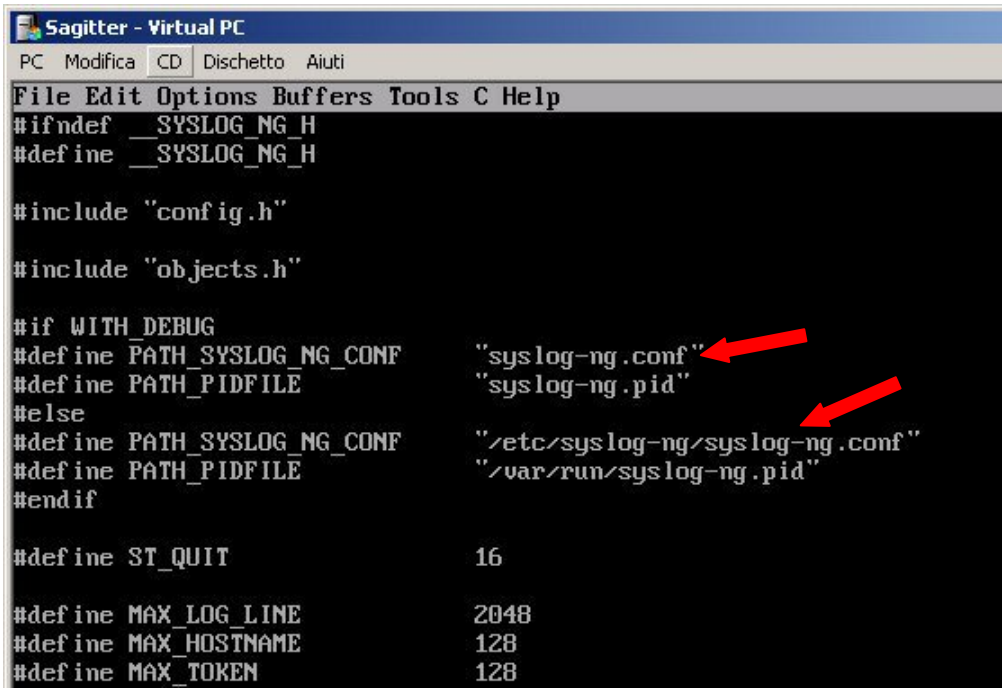
```
tar -xzvf syslog-ng-1.4.17.tar.gz
```

Editiamo il file src/syslog-ng.h con un qualunque editor di testi (VI, Emacs, etc.) modificando le linee

```
#define PATH_SYSLOG_NG_CONF "syslog-ng.conf"
#define PATH_SYSLOG_NG_CONF "/etc/syslog-ng/syslog-ng.conf"
```

in qualcosa di un poco più nascosto tipo

```
#define PATH_SYSLOG_NG_CONF "default.conf"
#define PATH_SYSLOG_NG_CONF "/etc/.conf/default.conf"
```



[2] L'utente "Root" su sistemi Unix like è l'amministratore del sistema, ovvero, colui che ha la facoltà di poter modificare le impostazioni di sistema.

Il motivo per il quale modifichiamo da sorgente il file di configurazione, è quello di renderlo invisibile ad un ps.

Prima di partire con la compilazione di syslog-ng è bene precisare che tale software necessita delle "libol", librerie aggiuntive scaricabili dallo stesso sito dal quale abbiamo prelevato syslog-ng. Una volta ottenuto l'archivio relativo a tali librerie ripetiamo l'azione di estrazione, quindi entriamo nella directory in cui abbiamo estratto le librerie e compiliamo ed installiamo eseguendo

```
./configure && make && make install
```

Se l'installazione delle librerie "libol" termina con successo possiamo passare all'installazione di syslog-ng. Dalla directory contenente l'archivio di syslog-ng estratto precedentemente lanciamo dunque il comando

```
./configure
```

Lo script denominato "*configure*" tenta automaticamente di impostare i valori corretti per le varie variabili system-dependent usate successivamente durante la compilazione del programma. Lo stesso script utilizza i medesimi valori per creare un "*Makefile*" per ogni diversa configurazione all'interno del package.

Nota: syslog-ng richiede la presenza del parser Bison e di Flex e del compilatore GNU gcc versione 2.7.2. Qualora non fossero installate lo script "configure" segnalerà tale errore impedendo di proseguire con l'installazione.

Lo script può anche creare uno o più header file ("*.h*") che contengono le definizioni system-dependent. Alla fine del processo di configurazione, viene creato uno script denominato "*config.status*", il quale può essere eseguito in futuro per ricreare la configurazione corrente.

Nel file "*config.cache*" vengono salvati i risultati dei tests per accelerare la riconfigurazione del programma di configurazione. Il file "*config.log*" contiene l'output del compilatore (utile principalmente per le operazioni di debug dello script "*configure*").

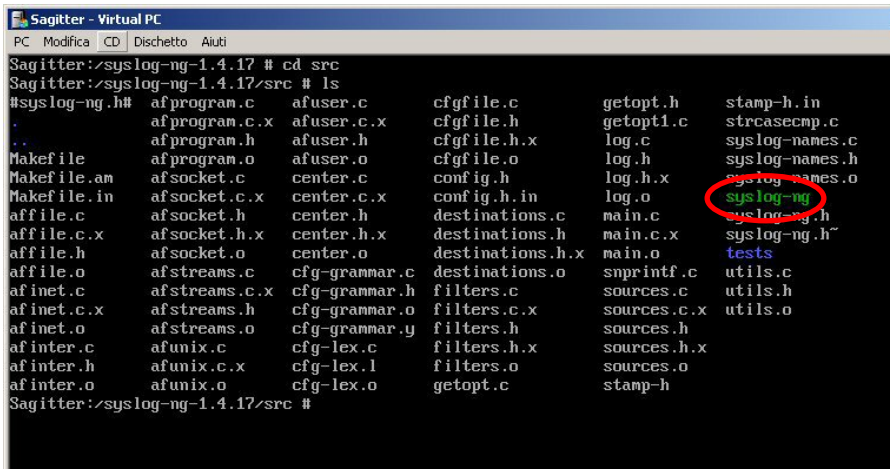
Qualora non si abbia la necessità di salvare i risultati dei tests contenuti nel file "*config.cache*" è possibile rimuovere tale file o modificarlo in base alle esigenze.

Il file "*configure.in*" è utilizzato per creare lo script "*configure*" da un programma chiamato "*autoconf*". Si ha bisogno solamente di tale file se si vuole cambiare o rigenerare lo script "*configure*" utilizzando ad esempio una versione più recente di "*autoconf*".

Il modo più semplice per compilare questo package è il seguente:

1. Digitare il comando "*cd*" dalla cartella che contiene il codice sorgente del package e quindi "*./configure*" al fine di configurare il package per il sistema di destinazione. Se si sta usando la shell "*csh*" su una vecchia versione del sistema "*System V*", si potrebbe aver bisogno di inserire la seguente riga "*sh ./configure*". Lanciando lo script "*configure*" il sistema rimarrà occupato per un pò di tempo. Durante l'esecuzione il programma visualizza alcuni messaggi che riportano quali operazioni si stanno effettuando.

2. Digitare "*make*" per compilare il package.
   Se il "make" ha avuto esito positivo potrete notare la presenza dell'eseguibile "syslog-ng" all'interno della sottocartella "src".

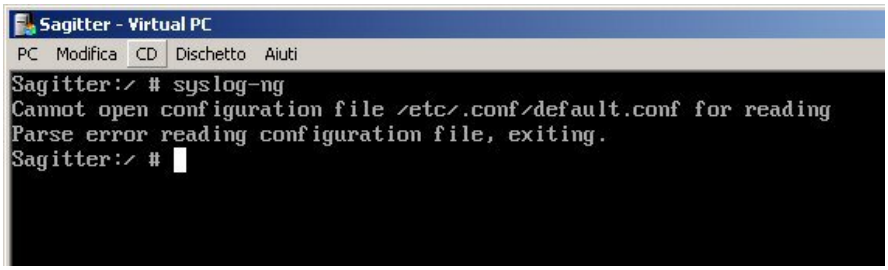


3. Opzionalmente, si digiti "*make check*" per lanciare alcuni autotest sul package.

4. Digitare "*make install*" per installare i programmi ed alcuni file di dati e la documentazione.

5. Si possono anche rimuovere i file binari del programma e i file object dalla cartella del codice sorgente digitando "*make clean*". Per rimuovere anche i files creati dallo script "*configure*" (in modo da poter compilare il package per un gruppo diverso di computers), digitare "*make distclean*". Esiste anche il comando "*make maintainer-clean*", ma è concepito principalmente per gli sviluppatori del package. Se lo si usa, si può ottenere la rigenerazione dei files creati per la distribuzione finale.

6. Provando a lanciare l'eseguibile syslog-ng possiamo avere conferma dell'avvenuta installazione. Nel nostro caso lo script di installazione non ha provveduto a creare la directory ed il file di configurazione, come prontamente segnalato dal programma stesso:



In questo caso è opportuno creare manualmente la directory utilizzando il comandi "mkdir", ed ovviamente il file di configurazione anche con un semplice "touch".

Compilatori ed Opzioni

Alcuni sistemi richiedono opzioni insolite per la compilazione o collegamenti che il programma "*configure*" non prevede. Si può dare comunque allo script "*configure*" valori di inizializzazione per particolari variabili mettendoli nella definizione dell'ambiente di sistema. Usando una shell Bourne - compatibile, si può fare tutto utilizzando la linea di comando, ad esempio:

```
CC=c89 CFLAGS=-O2 LIBS=-lposix ./configure
```

Sui sistemi che utilizzano la parola chiave "*env*", si può utilizzare la seguente sintassi:

```
env CPPFLAGS=-I/usr/local/include LDFLAGS=-s ./configure
```

Compilazione per Architetture Multiple

Si può compilare il package per più architetture nel medesimo tempo, inserendo i files object di ogni architettura nella propria cartella di compilazione. Per effettuare tale operazione è necessario usare una versione del programma "*make*" che supporti le variabili "*VPATH*", come ad esempio il "*make*" della GNU.

Se il "*make*" utilizzato non supporta "*VPATH*", è necessario compilare il package tante volte quante sono le architetture. Dopo aver compilato il package per la singola architettura, utilizzare "*make distclean*" prima di rieseguire la compilazione per un'altra architettura.

Parametri di installazione

In maniera predefinita, "*make install*" installerà i files del package nelle cartelle "*/usr/local/bin*", "*/usr/local/man*", etc. Si può specificare un altro prefisso di installazione in sostituzione di "*/usr/local*" mediante il parametro posto all'interno dello script "configure":

```
--prefix=PATH
```

Si possono specificare prefissi di installazione separati per i files di ogni singola architettura e per i files indipendenti dall'architettura. Se si da allo script "*configure*" il parametro

```
--exec-prefix=PATH
```

il package userà il PATH come il prefisso per installare i programmi e librerie. La documentazione e gli altri files dati continueranno ancora ad utilizzare il prefisso precedentemente impostato.

Ancora, se si usa una configurazione di cartelle insolita si possono dare opzioni come

```
--bindir=PATH
```

per specificare valori diversi per particolari generi di files. Digitare

```
configure -help
```

per un elenco delle cartelle che possono essere impostate e quali files contengono.

In alcune versioni, si può anche fare in modo che i programmi vengano installati con un prefisso o suffisso addizionale sui loro nomi dando a "*configure*" l'opzione:

```
--program-prefix=PREFIX
```

oppure

```
--program-suffix=SUFFIX
```

Caratteristiche opzionali

Alcuni package utilizzano l'opzione

```
--enable-FEATURE
```

dove FEATURE indica una parte opzionale del package. Altri le opzioni

```
--with-PACKAGE
```

dove PACKAGE è qualcosa come "*gnu-as*" o "*x*" (per il sistema X Window).

Per i package che usano il sistema X Window, "*configure*" di solito trova la cartella "*include*" e le librerie per la compilazione automaticamente, ma se ciò non avviene, si possono impostare in modo specifico con le seguenti opzioni:

```
--x-includes=DIR
```

e

```
--x-libraries=DIR
```

Specifiche del tipo di sistema

Ci possono essere delle caratteristiche dello script "*configure*" che non possono essere dedotte automaticamente dal sistema. Ad esempio può essere necessario determinare il tipo di host. Di solito "*configure*" riesce ad ottenere quest'informazione, ma se viene visualizzato un messaggio che riporta che non è stato possibile determinare il tipo di host allora è necessario impostare la seguente opzione:

```
--host=TYPE
```

TYPE di solito può essere il nome del sistema, come ad esempio "*sun4*", o un nome canonico a tre campi:

```
CPU-SOCIETA'-SISTEMA
```

Guardare il file "*config.sub*" per determinare i possibili valori di ogni campo. Se "*config.sub*" non è incluso nel package, non si ha bisogno di impostare manualmente il nome dell'host.

Condivisioni di Default

Se si vuole inserire valori predefiniti per lo script "*configure*" per la condivisione, si può creare uno script chiamato "*config.site*" che contiene i valori di default per le variabili come ad esempio "*CC*", "*cache_file*", e "*prefix*".

Lo script "*configure*" utilizza per effettuare tale operazione i valori contenuti in "*PREFIX/share/config.site*" se esiste, poi quelli contenuti in "*PREFIX/etc/config.site*" se esiste. Si può effettuare la medesima operazione impostando la variabile d'ambiente "*CONFIG_SITE*" ove specificare l'ubicazione degli script.

Attenzione: non tutti gli script "*configure*" cercano un sito per gli scripts.

Operazioni di controllo

"*configure*" riconosce le seguenti opzioni per controllare le operazioni dello stesso.

```
--cache-file=FILE
```

Utilizza e salva i risultati delle prove nel FILE invece di "*./config.cache*". Impostare FILE a "*/dev/null*" per disabilitare il caching, per il debug di "*configure*"

```
--help
```

Stampa un sommario delle opzioni di "*configure*", e esce.

```
--quiet
--silent
   -q
```

Fanno in modo che non si stampino messaggi che riportano quali controlli sono stati effettuati. Per eliminare questa opzione, si effettui il "*redirect*" verso "*/dev/null*" (saranno mostrati alcuni messaggi di errore).

```
--srcdir=DIR
```

Cerca il codice sorgente del package nella cartella DIR. Di solito "*configure*" può determinare automaticamente tale cartella.

```
--version
```

Stampa la versione di "*Autoconf*" utilizzata per generare lo script "*configure*".

Capitolo

3

Configurazione ed utilizzo

I pilastri di Syslog-NG: destinazioni, sorgenti e filtri. Come configurare le singole componenti e collegarle tra loro per ottenere i risultati desiderati.

I percorsi dei messaggi

In Syslog-NG il percorso di un messaggio (message path o message route) consiste di una o più sorgenti, di una o più regole di filtraggio e di una delle possibili destinazioni (pozzi). Un messaggio viene inserito in Syslog-NG in una delle sue sorgenti, e, se tale messaggio è conforme alle regole di filtraggio, viene trasmesso fuori tramite una delle uscite.

Sorgenti

Una sorgente è un insieme di driver sorgenti, che raccoglie messaggi usando un metodo predefinito. Ad esempio c'è un driver per i socket di tipo AF_UNIX, SOCK_STREAM, il quale è utilizzato dalla procedura di sistema syslog() di Linux.

Piattaforme diverse utilizzano metodi diversi per mandare messaggi al daemon che si occupa del logging, e per essere utile a tutti i sistemi operativi, Syslog-NG supporta tutti i metodi più comuni. È stato provato su Linux, BSD, ed esiste a partire dalla versione 1.1.22 di Syslog-NG anche un supporto sperimentale per Solaris.

Per dichiarare una fonte, ci sarà bisogno della dichiarazione di sorgente nel file di configurazione file con la seguente sintassi:

```
source <identificatore> { source-driver(parametri);
     source-driver(parametri); ... };
```

L'identificatore deve essere univoco per questa sorgente assegnata e ovviamente non deve essere una delle parole riservate.

Si può controllare esattamente quali driver sono usati per raccogliere i messaggi di log, anche se si dovrà conoscere come il sistema e il suo syslogd nativo comunicano.

Ogni possibile meccanismo di comunicazione ha il corrispondente driver sorgente in Syslog-NG. Ad esempio per aprire un socket Unix SOCK_DGRAM si utilizza il driver **`unix-dgram`**. Analogamente per aprire un socket Unix SOCK_STREAM – utilizzato nei sistemi Linux – è necessario il driver **`unix-stream`**.

Piattaforma	Metodo
Linux	Un socket Unix SOCK_STREAM chiamato /dev/log
BSD flavours	Un socket Unix SOCK_DGRAM chiamato /var/run/log
Solaris (ver. 2.5 o precedenti)	Un device SVR4 STREAMS chiamato /dev/log
Solaris (ver. 2.6 o successive)	In aggiunta agli STREAMS device usati nelle versioni precedenti la 2.6, usa un nuovo metodo IPC multithreaded chiamato door. Per default il door usato da syslogd è /etc/.syslog_door

Tavola 3-1. Metodi di comunicazione tra syslogd e i suoi clients

```
source src { unix-stream("/dev/log"); internal();
        udp(ip(0.0.0.0) port(514)); };
```

Esempio 3-1. Dichiarazione di sorgente su un sistema Linux-based

Ogni driver può prendere parametri, alcuni dei quali sono obbligatori, altri opzionali. I parametri richiesti sono *posizionali*, il che significa che essi devono essere specificati in un ordine ben definito. Un driver unix-stream() ha un singolo argomento richiesto, il nome del socket da ascoltare, e diversi parametri opzionali, che seguono il nome del socket. Gli argomenti opzionali possono essere specificati in qualsiasi ordine e devono avere la forma `option(value)`.

Nome	Descrizione
`internal`	Messaggi generati internamente da Syslog-NG
`unix-stream`	Apre gli unix socket specificati in modalità SOCK_STREAM, e ascolta i messaggi.
`unix-dgram`	Apre lo unix socket specificato in modalità SOCK_DGRAM, e ascolta i messaggi.
`file`	Apre i file specificati e legge messaggi.
`pipe, fifo`	Apre i named pipe specificati e legge messaggi.
`udp`	Ascolta sulla porta UDP specificata.
`tcp`	Ascolta sulla porta TCP specificata.
`sun-stream,`	Apre gli STREAMS device specificati su sistemi Solaris, e legge i

`sun-streams`	messaggi.

Tabella 3-2. Driver sorgenti disponibili in Syslog-NG

Filtri

I filtri specificano come Syslog-NG eseguirà l'instradamento dei messaggi al suo interno. Un filtro è, nella pratica, un'espressione booleana, la quale deve essere valutata vera se si vuole che il messaggio debba essere trasmesso in uscita.

Anche i filtri hanno un nome univoco, di modo che ci si possa riferire ai filtri nei propri log statements. E' possibile dichiarare un filtro utilizzando la seguente sintassi:

```
filter <identificatore> { espressione; };
```

Un'espressione potrebbe contenere parentesi, gli operatori booleani "and", "or" e "not", e ognuna delle funzioni nella Tavola 3-9.

```
filter f_lybra_deny { host("lybra") and match("deny"); };
```

Esempio 3-2. Un filter statement che trova i messaggi contenenti la parola deny proveniente dall'host "lybra"

Nelle prime versioni di Syslog-NG c'era uno speciale identificatore di filtro, "DEFAULT", il quale faceva in modo che il filtro venisse applicato a tutti i messaggi non associati ad altri filtri. Questa caratteristica è stata rimossa da Syslog-NG a partire dalle versioni 1.5.x.

Destinazioni

Una destinazione è un luogo dove si raccolgono i messaggi, ovvero dove viene inviato e memorizzato il log se le regole di filtraggio vengono soddisfatte. In modo analogo alle sorgenti, le destinazioni possono includere diversi driver che definiscono come verranno distribuiti i messaggi.

Ad esempio esiste un driver per i file, che scrive i messaggi su un determinato file, ma è anche possibile mandare messaggi su socket di tipo unix, udp e tcp.

Per dichiarare una destinazione nel file di configurazione, ci sarà bisogno di un comando di destinazione (destination statement), la cui sintassi è la seguente:

```
destination <identificatore> { destination-driver(parametri);
            destination-driver(parametri); ... };
```

Nome	**Descrizione**

`file`	Scrive i messaggi nel file specificato
`fifo, pipe`	Scrive i messaggi al pipe specificato
`unix-stream`	Invia i messaggi al socket specificato di tipo SOCK_STREAM (Linux)
`unix-dgram`	Invia i messaggi al socket specificato di tipo SOCK_DGRAM (BSD)
`udp`	Invia i messaggi all'host e alla porta UDP specificati
`tcp`	Invia i messaggi all'host e alla porta TCP specificati
`usertty`	Invia i messaggi all'utente specificato se registrato
`program`	Sblocca e lancia il programma specificato, e invia i messaggi al suo standard input.

Tavola 3-3. Destinazioni disponibili in Syslog-NG

Log paths

Abbiamo visto sorgenti, destinazioni e filtri. Per collegarli insieme c'è bisogno dei comandi di log (log statement):

```
log { source s1; source s2; ...
      filter f1; filter f2; ...
destination d1; destination d2; ... };
```

I messaggi provenienti dalle sorgenti elencate, e che soddisfano tutti i filtri impostati vengono inviati alle destinazioni stabilite.

I log statements sono trattati nell'ordine in cui appaiono nel file di configurazione. Per default tutti i log statements vengono processati, quindi un singolo messaggio di log potrebbe essere inviato alla stessa destinazione più volte, nel caso in cui la destinazione sia elencata in più log statements. Questo comportamento di default può essere cambiato tramite i parametri `flags()`.

Opzioni

Ci sono diverse opzioni che si possono specificare le quali modificano il comportamento di Syslog-NG. La sintassi generale è:

```
options { option1(parametri); option2(parametri); ... };
```

Ogni opzione può avere dei parametri, proprio come nella dichiarazione dei driver.

Flag	**Descrizione**
`final`	Questa flag indica che il processo di log termina qui. Si noti che ciò non significa necessariamente che i messaggi rilevati saranno archiviati una

	volta, poiché possono esserci log statements rilevati processati prima del corrente.
`fallback`	Questa flag rende un log statement di tipo 'fallback'. Ciò significa che solo i messaggi che non corrispondono ad alcun 'non-fallback' log statement saranno inviati.
`catchall`	Questa flag indica che la fonte del messaggio viene ignorata, e soltanto i filtri sono presi in considerazione quando i messaggi vengono rilevati.

Tavola 3-4. I flags dei log statement

Driver ed opzioni

Source drivers

I seguenti driver possono essere usati nel source statement, come descritto nel capitolo precedente.

internal()

Tutti i messaggi generati internamente "provengono" da questa speciale fonte. Se si desiderano avvertimenti, errori e avvisi da Syslog-NG, bisogna includere questa fonte in uno dei source statements.

```
internal()
```

Syslog-NG genererà un messaggio di "warning", se il driver specificato non è stato dichiarato.

```
source s_local { internal(); };
```

Esempio 3-2. Usare l'internal() driver

unix-stream() e unix-dgram()

Questi due driver si comportano similmente: essi aprono il socket AF_UNIX assegnato, e cominciano ad ascoltare su di esso i messaggi. unix-stream() è usato fondamentalmente su Linux, e utilizza SOCK_STREAM semantici (connection oriented, nessun messaggio viene perso), unix-dgram() è usato su sistemi BSD, ed utilizza SOCK_DGRAM semantici (ciò potrebbe causare la perdita di messaggi locali qualora il sistema sia sovraccaricato).

Per evitare attacchi di tipo DoS (Denail of Service) quando si utilizzano protocolli connection-oriented, il numero delle connessioni simultaneamente accettate deve essere limitato. Ciò può essere fatto usando il parametro `max-connections()`. Il valore default di questo parametro è piuttosto ristretto; è necessario aumentare questo valore nel caso di sistemi notevolmente occupati.

Sia `unix-stream` che `unix-dgram` hanno un singolo argomento richiesto, che specifica il nome del socket da creare, e diversi parametri opzionali.

```
unix-stream(filename [options]);
unix-dgram(filename [options]);
```

Nome	Tipo	Descrizione	Default
`owner()`	Testo	Imposta l'uid del socket.	root
`group()`	Testo	Imposta il gid del socket.	root
`perm()`	Numerico	Imposta la permission mask. Per numeri ottali prefissare il numero con '0', es. usare 0755 per rwxr-xr-x.	0666
`keep-alive()`	Si/No	Seleziona se mantenere le connessioni aperte quando Syslog-NG è riavviato, può essere usato solo con unix-stream().	yes
`max-connections()`	Numerico	Limita il numero di connessioni aperte simultaneamente. Può essere usato solo con unix-stream().	10

Tavola 3-5. Opzioni disponibili per unix-stream e unix-dgram

```
# Dichiarazione di sorgente in sistemi Linux
      source s_stream { unix-stream("/dev/log" max-
      connections(10)); };

# Dichiarazione di sorgente in sistemi BSD
          source s_dgram { unix-dgram("/var/run/log"); };
```

Esempio 3-3. Uso dei driver unix-stream() e unix-dgram()

`tcp()` e `udp()`

Questi driver lasciano ricevere messaggi dal network, e, come il nome dei driver suggerisce, possono usare tanto UDP quanto TCP come mezzo di trasporto.

UDP è un semplice datagram oriented protocol, che fornisce il "minor sforzo" per trasferire messaggi tra host. Potrebbe perdere messaggi, e non viene fatto alcun tentativo per ritrasmettere tali messaggi.

TCP fornisce un servizio connection-oriented, il che fondamentalmente significa un messaggio pipeline flow-controlled. In questo pipeline, ogni messaggio è autorizzato, e la ritrasmissione viene effettuata per i packets persi. Generalmente è più sicuro usare TCP, perchè le connessioni perse possono essere rilevate e nessun messaggio va perso, ma tradizionalmente il protocollo syslog utilizza UDP.

Nessuno dei driver tcp() and udp() richiede parametri posizionali. Per default essi si legano a 0.0.0.0:514, il che significa che Syslog-NG ascolterà su tutte le interface disponibili, port 514. Per limitare le connessioni accettate ad una sola interfaccia, si usi il parametro localip() come descritto in basso.

NOTA: la porta tcp 514 è riservata all'uso con rshell, quindi si deve segliere un'altra porta se si intende usare Syslog-NG e rshell contemporaneamente.

```
tcp([options]);
udp([options]);
```

Nome	Tipo	Descrizione	Default
`ip o localip`	Testo	L'indirizzo IP a cui collegarsi. Si noti che non è l'indirizzo dal quale vengono accettati i messaggi.	0.0.0.0
`port o localport`	Numerico	Il numero di porta a cui collegarsi.	514
`keep-alive`	Si/No	Disponibile solo per tcp(), stabilisce se chiudere le connessioni al ricevimento di un segnale SIGHUP.	yes
`max-connections`	Numerico	Specifica il numero massimo di connessioni simultanee.	10

Tavola 3-6. Opzioni disponibili per udp e tcp

```
source s_tcp { tcp(ip(127.0.0.1) port(1999) max-
connections(10)); };
source s_udp { udp(); };
```

Esempio 3-4. Uso dei driver udp() e tcp()

file()

```
file(filename);
```

Di solito il kernel presenta i suoi messaggi in un file speciale (/dev/kmsg su BSD, /proc/kmsg su Linux), quindi per leggere tali file speciali, ci sarà bisogno del driver file(). Si noti che non è possibile usare questo driver per seguire un file come fa la coda. Per alimentare un logfile crescente in Syslog-NG (HTTP access.log ad esempio), si usa uno script come questo:

```
#!/bin/sh tail -f logfile | logger -p local4.info
```

Esempio 3-5. script di esempio per alimentare un logfile crescente in Syslog-NG

Il file driver ha un solo parametro richiesto, che specifica il file da aprire, ed è il seguente:

Nome	Tipo	Descrizione	Default
`log_prefix`	Testo	La stringa per prepend log messages. Utile per logging messaggi kernel poiché non è prefissato da 'kernel:' per default	Stringa vuota

Tavola 3-7. Opzioni disponibili per file()

```
source s_file { file("/proc/kmsg"); };
```

Esempio 3-6. Uso del driver file()

`pipe()`
Il driver pipe apre un pipe chiamato con il nome specifico, e ascolta i mesaggi. Il driver pipe ha un solo parametro richiesto, che specifica il filename del pipe da aprire, e la seguente opzione:

Nome	Tipo	Descrizione	Default
`pad_size`	Numerico	Specifica il padding di input. Alcuni sistemi operativi (come HP-UX) riempiono tutti i messaggi per bloccare il limite. Questa opzione può essere usata per specificare la dimensione dei blocchi. (HP-UX usa 2048 bytes)	0

Tavola 3-4. Opzioni disponibili per pipe

```
pipe(filename);
```

NOTA: bisognerà creare questo pipe usando mkfifo(1).

```
source s_pipe { pipe("/dev/log"); };
```

Esempio 3-7. Uso del driver pipe()

`sun-streams()`
Solaris usa il suo STREAMS API per inviare messaggi al processo syslogd. Si dovrà compilare Syslog-NG con questo driver.

Le versioni più recenti di Solaris (2.5.1 e successive), usano un nuovo IPC in aggiunta a STREAMS, chiamato door per confermare l'invio di un messaggio. Syslog-NG supporta questo nuovo meccanismo IPC con l'opzione door().

Il driver sun-streams() ha un solo argomento richiesto, che specifica lo STREAMS device da aprire ed una singola opzione.

```
source s_stream { sun-streams("/dev/log"
  door("/etc/.syslog_door"); };
```

Esempio 3-8. Uso del driver sun-streams()

Nome	Tipo	Descrizione	Default
door	Testo	Specifica il nome di una porta da aprire, necessario su Solaris oltre 2.5.1.	Stringa vuota

Tavola 3-5. Opzioni disponibili per sun-streams

Driver di destinazione

I driver di destinazione inviano i messaggi di log verso qualche destinazione esterna a Syslog-NG, ovvero verso un file o un network socket.

file()

Il driver *file* è uno dei più importanti driver di destinazione in Syslog-NG. Esso autorizza ad inviare messaggi al file scelto, oppure come vedremo ad un gruppo di file.

Il nome della destinazione può includere macro che vengono espanse quando il messaggio è scritto, così un semplice file() driver può risultare in diversi file da creare. Le macro possono essere specificate tramite il prefisso "$" (senza le virgolette) davanti al nome della macro, proprio come in Perl/PHP.
Se il nome espanso si riferisce a una directory che non esiste, questa verrà creata automaticamente a seconda delle impostazioni del `create_dirs()` (opzione sia globale che per destinazione).

Attenzione: poichè lo stato di ogni file creato deve essere tracciato da Syslog-NG, esso consuma memoria per ogni file. Se non viene scritto nessun nuovo messaggio ad un file entro 60 secondi (controllati dall'opzione globale time_reap), viene chiuso e il suo stato è "freed".

Utilizzandolo, un attacco DoS (Denial of Service) può essere messo in atto contro il sistema nel caso in cui il numero di file di destinazione possibili sia elevato e la memoria necessaria sia maggiore di quella di cui il logserver è provvista.

La macro più ambigua, in cui le variazioni possibili sono piuttosto alte, è $PROGRAM, perciò in un ambiente poco sicuro l'uso di $PROGRAM dovrebbe essere evitato.

Nome	Descrizione
FACILITY	Il nome della facility da cui il messaggio è segnato come proveniente
PRIORITY o LEVEL	La priorità del messaggio
TAG	La priorità e la facility codificate come un numero esadecimale a 2 cifre
DATE	-
FULLDATE	-
ISODATE	-
YEAR	L'anno in cui il messaggio è stato inviato. Le macro di espansione Tempo possono anche usare il tempo specificato nel messaggio di log, ad esempio il tempo in cui il messaggio di log è stato inviato, o il tempo in cui il messaggio è stato ricevuto dal log server. Questo è controllato dall'opzione `use_time_recvd()`
MONTH	Il mese in cui è stato inviato il messaggio
DAY	Il giorno del mese in cui è stato inviato il messaggio
WEEKDAY	Le prime 3 lettere del nome del giorno della settimana in cui è stato inviato il messaggio es. "Thu" (Thursday), "Mon" (Monday), etc.
HOUR	L'ora del giorno in cui è stato inviato il messaggio.
MIN	Il minuto in cui è stato inviato il messaggio
SEC	Il secondo in cui è stato inviato il messaggio
TZOFFSET	La zona di tempo intesa come scarto di ore da GMT. Ad esempio '-0700'
TZ	L'ora locale, il nome o l'abbreviazione. Ad esempio 'PDT'
FULLHOST	-
HOST	Il nome del host da cui è partito il messaggio. Se il messaggio attraversa diversi host, e `chain_hostnames()` è attivo, viene usato il primo.
PROGRAM	Il nome del programma da cui è inviato il messaggio
MSG o MESSAGE	Contenuto del messaggio

Tavola 3-6. Macro disponibili nell'espansione di filename

Nome	Tipo	Descrizione	Default
`log_fifo_size()`	Numerico	Il numero delle entrate nell'output fifo	Usa l'impostazione globale
`fsync()`	Si/No	Forza una chiamata fsync() sulla destinazione fd dopo ogni scrittura. Nota: ciò potrebbe peggiorare seriamente le performance	
`sync_freq()`	Numerico	Il logfile è synced quando questo numero di messaggi viene scritto	Usa l'impostazione globale.
`encrypt()`	Si/No	Cripta il file risultante. NOTA: ciò non è implementato nella versione 1.3.14.	Usa l'impostazione globale
`compress()`	Si/No	Comprime il logfile risultante usando zlib. NOTA: ciò non è implementato nella versione 1.3.14.	Usa l'impostazione globale.
`owner()`	Testo	Assegna il possessore del file creato a quello specificato	root
`group()`	Testo	Assegna il gruppo di file creati a quello specificato	root
`perm()`	Numerico	La permission mask del file se è esso è creato da Syslog-NG.	0600
`dir_perm()`	Numerico	La permission mask delle directory create da Syslog-NG. Le directories di log vengono create solo se un file dopo la macroespansione si riferisce a una directory non esistente, e la creazione dir è abilitata usando create_dirs().	0600
`create_dirs()`	Si/No	Abilita la creazione di directory non esistente.	no
`template()`	Testo	Specifica un template che specifica il formato del log da usare in questo file. Le macro possibili sono le stesse dei file di destinazione	un formato conforme al default del formato del log
`template_escape()`	Si/No	Attiva i caratteri ' e " per il template. E' utile per generare dichiarazioni SQL e quotare stringhe così che quelle parti	si

		del messaggio di log non vengano interpretate come comandi per il server SQL	
`remove_if_older()`	Numerico	Se impostato ad un valore più alto di 0, prima di scrivere a un file, Syslog-NG controlla se questo file è più vecchio del tempo specificato (in secondi). Se sì, esso rimuove il file esistente e la riga da scrivere diventa la prima riga di un nuovo file con lo stesso nome. In combinazione con ad es. la macro $WEEKDAY, questo può essere usato per una semplice log rotation, nel caso in cui non tutta la storia necessiti di essere considerata.	Non rimuove mai il file esistente, ma lo sospende (= 0).

Tavola 3-7. Opzioni disponibili per file()

`pipe()`
Questo driver invia messaggi a un pipe scelto (ad esempio /dev/console).
Il driver pipe ha solo un parametro richiesto, che specifica il nome del pipe da aprire, e nessuna opzione.

```
pipe(filename);
```

NOTA: ci sarà bisogno di creare questo pipe usando mkfifo(1).

```
destination d_pipe { pipe("/dev/xconsole"); };
```

Esempio 3-8. Uso del driver pipe()

`unix-stream()` e `unix-dgram()`
Questo driver invia messaggi ad uno unix socket di tipo SOCK_STREAM o SOCK_DGRAM.
Entrambi i driver hanno un solo argomento richiesto che specifica il nome del socket a cui connettersi, e nessun argomento opzionale.

```
unix-stream(filename [options]);
unix-dgram(filename [options]);
```

`udp()` e `tcp()`
Questo driver invia messaggi ad un altro host sul locale intranet o internet che usa sia il protocollo UDP che quello TCP.

Entrambi i driver hanno un solo argomento richiesto che specifica l'indirizzo dell'host di destinazione, a cui il messaggio deve essere inviato, e diversi

parametri opzionali. Si noti che questo differisce dai source drivers, in cui l'indirizzo bind locale è implicito, e non è richiesto nessun parametro.

```
tcp(host [options]);    udp(host [options]);
```

Nome	Tipo	Descrizione	Default
`localip`	Testo	L'indirizzo IP a cui allacciarsi prima di connettersi al target.	0.0.0.0
`localport`	Numerico	Il numero di port a cui allacciarsi.	0
`port o destport`	Numerico	Il numero di port a cui connettersi.	514

Tavola 3-8. Opzioni disponibili per udp e tcp

```
destination d_tcp { tcp("10.1.2.3" port(1999);
                    localport(999)); };
```

Esempio 3-9. Uso del driver tcp()

usertty()

Questo driver scrive messaggi al terminale di un utente attualmente attivo.
Il driver usertty ha un solo argomento richiesto, che specifica lo username che dovrebbe ricevere una copia dei messaggi accoppiati, e nessun argomento opzionale.

```
usertty(username);
```

program()

Questo driver esegue una "fork" esegue il programma assegnato con gli argomenti assegnati e invia messaggi allo standard input dei processi figli.
Il driver program ha un solo parametro richiesto, che specifica un program name da iniziare, e nessuna opzione. Il programma viene eseguito con l'aiuto dello shell corrente, quindi il commando potrebbe includere entrambi i file patterns e I/O redirection, che saranno avviati.

```
program(commandtorun);
```

NOTA: il programma viene eseguito una volta allo startup, e continua ad essere attivo fino al SIGHUP o all'uscita. Il motivo è prevenire l'avviamento di un gran numero di programmi per messaggi, che implicherebbero un facile attacco DoS (Denial of Service).

```
destination d_prg { program("/bin/cat > /dev/null"); };
```

Esempio 3-10. Uso del destination driver program()

Funzioni filtro

Le seguenti funzioni possono essere usate nel filter statement, come descritto precedentemente.

Nome	Sintassi	Descrizione
`facility()`	`facility(facility[,facility])`	Unisce messaggi che hanno uno ei facility code elencati.
`level()` o `priority()`	`level(pri[,pri1..pri2[,pri3]])`	Unisce messaggi basati sulla priorità.
`program()`	`program(regexp)`	Unisce messaggi usando una regolare espressione contro il program name field dei log messages.
`host()`	`host(regexp)`	Unisce messaggi usando una regular expression per il controllo del campo "hostname" dei log messages.
`match()`	`match(regexp)`	Prova ad unire una regolare espressione al messaggio stesso.
`filter()`	-	Chiama un altro filter rule e valuta il suo valore.

Tavola 3-9. Funzioni filtro disponibili

Opzioni

Le seguenti opzioni possono essere specificate nell'options statement, come descritto precedentemente.

Nome	Valori accettati	Descrizione
`time_reopen()`	Numerico	Il tempo da aspettare prima che una connessione caduta venga ristabilita.
`time_reap()`	Numerico	Il tempo da aspettare prima che un file di destinazione venga chiuso.
`sync()`	Numerico	Il numero di righe accodate prima di essere scritte su file.
`mark()`	Numerico	Il numero di secondi tra due righe MARK. NOTA: non ancora implementato.
`stats()`	Numerico	Il numero di secondi tra due STATS.
`log_fifo_size()`	Numerico	Il numero di righe della coda di output
`chain_hostnames()`	Si/No	Attiva o disattiva il *chained hostname format*.
`keep_hostname()`	Si/No	Attiva o disattiva la riscrittura dell'hostname.
`check_hostname()`	Si/No	Attiva o disattiva qualora l'hostname contenga validi caratteri.
`bad_hostname()`	Regular expression	Un regexp che controlla la validità dei nomi di host.
`create_dirs()`	Si/No	Attiva o disattiva la creazione di directory per file di destinazione.
`owner()`	userid	-
`group()`	groupid	-
`perm()`	permission value	-
`dir_owner()`	userid	-
`dir_group()`	groupid	-
`dir_perm()`	permission value	-
`create_dirs()`	Si/No	Attiva o disattiva la creazione di directory per file di destinazione.
`use_time_recvd()`	Si/No	Usa il tempo in cui un messaggio è ricevuto invece di quello specificato

		nel messaggio.
`use_dns()`	Si/No	Attiva o disattiva l'uso di DNS. Syslog-NG si blocca sulle DNS queries, così l'attivazione di DNS può portare a un Denial di Service attack. Per prevenire DoS, è necessario proteggere il punto finale del network Syslog-NG con firewall rules, ed assicurarsi che tutti i nomi degli host che possono portare a Syslog-NG siano risolvibili.
`dns_cache()`	Si/No	Attiva o disattiva l'uso di cache DNS.
`dns_cache_size()`	Numerico	Numero di hostnames nella cache DNS.
`dns_cache_expire()`	Numerico	Numero di secondi in cui un lookup che ha avuto successo viene inserito nella cache.
`dns_cache_expire_failed()`	Numerico	Numero di secondi in cui un lookup fallito viene inserito nella cache.
`log_msg_size()`	Numerico	Lunghezza massima del messaggio in bytes.
`use_fqdn()`	Si/No	Aggiunge nomi di dominio completamente qualificati invece di un breve hostname.
`gc_idle_threshold()`	Numerico	Imposta il valore di soglia per il cestino, quando Syslog-NG è idle. La fase GC inizia quando il numero di oggetti allocati raggiunge questo valore. Default: 100.
`gc_busy_threshold()`	Numerico	Come precedente, ma usato quando Syslog-NG è occupato. Default: 3000

Tavola 3-10. Lista delle opzioni globali supportate

Performance tuning

Ci sono diverse impostazioni disponibili per poter ottimizzare il comportamento di Syslog-NG. Le impostazioni di default dovrebbero essere adeguate per un singolo server o un'installazione workstation, ma per un logserver centralizzato ricevente i log da più host tali impostazioni potrebbero non essere sufficienti.

Impostare i parametri del cestino

Syslog-NG utilizza un cestino interno, e, mentre il cestino è in funzione esso non accetta messaggi. Ciò potrebbe causare problemi se qualche protocollo non-connection oriented è in uso, come unix-dgram() o udp(). Ci sono due impostazioni che che controllano la fase di "raccolta rifiuti"

gc_idle_threshold()

Con questa opzione si può specificare la soglia del gc (garbage collector). Se il numero degli oggetti allocati raggiunge questo numero, e il sistema è idle (nessun messaggio arrivato entro 100msec), inizia una fase gc. Sin da quando il sistema è idle, presumibilmente nessun messaggio andrà perduto se il gc è avviato. Dunque questo valore dovrebbe essere basso, ma più alto di quello minimo di oggetti allocati. Il numero minimo di oggetti allocarti dipende dalla configurazione, ma si può ottenere un numero esatto specificando l'opzione -v da riga di comando.

gc_busy_threshold()

Questa soglia viene usata quando Syslog-NG è occupato ad accettare messaggi (ciò significa che in 100msec un evento di I/O è occorso). Comunque per prevenire che Syslog-NG consumi tutta la memoria, gc si dovrebbe avviare. Si imposti questo valore alto, così che i log bursts (ondate di log) non siano interrotti da gc.

Impostare la dimensione della coda di output

Syslog-NG legge sempre i suoi canali log in entrata per prevenire il blocco del daemon in uso. Ciò potrebbe causare la perdita di messaggi se la coda di output è piena. E' dunque importante impostare la dimensione della coda di output (definita in numero di messaggi), il che si può fare globalmente, o su una base per destinazione:

```
options { log_fifo_size(1000); };
```

oppure

```
destination d_messages { file("/var/log/messages");
                         log_fifo_size(1000); };
```

Dovreste impostare la dimensione fifo al numero stimato di messaggi in un message burst (ondata di messaggi).

Impostare il parametro sync

Il parametro *sync* non fa esattamente ciò che ci si aspetta. Come visto, i messaggi da inviare sono inseriti in una coda di output. Il parametro sync

specifica il numero dei messaggi fermi in questo buffer prima che qualcosa venga scritto. Si noti che esso non scrive tutti i messaggi in un singolo chunk, ma scrive ogni singolo messaggio con un unico sistema di chiamata *write()*.

File di configurazione: un esempio

```
### OPZIONI E SORGENTI ###
options { long_hostnames(off); sync(0); };

source src { unix-stream("/dev/log"); internal(); };
source kernsrc { file("/proc/kmsg"); };


### DESTINAZIONI ###

destination authlog { file("/var/log/auth.log"); };
destination syslog { file("/var/log/syslog"); };
destination cron { file("/var/log/cron.log"); };
destination daemon { file("/var/log/daemon.log"); };
destination kern { file("/var/log/kern.log"); };
destination lpr { file("/var/log/lpr.log"); };
destination user { file("/var/log/user.log"); };
destination uucp { file("/var/log/uucp.log"); };
destination ppp { file("/var/log/ppp.log"); };
destination mail { file("/var/log/mail.log"); };

destination mailinfo { file("/var/log/mail.info"); };
destination mailwarn { file("/var/log/mail.warn"); };
destination mailerr { file("/var/log/mail.err"); };
destination local0 { file("/var/log/local0.log"); };
destination local1 { file("/var/log/local1.log"); };
destination all { file("/var/log/all.log"); };

destination newscrit { file("/var/log/news/news.crit"); };
destination newserr { file("/var/log/news/news.err"); };
destination newsnotice { file("/var/log/news/news.notice");
};

destination debug { file("/var/log/debug"); };
destination messages { file("/var/log/messages"); };
destination console { usertty("root"); };
destination console_all { file("/dev/tty12"); };

### FILTRI ###

filter f_auth { facility(auth); };
filter f_authpriv { facility(auth, authpriv); };
filter f_syslog { not facility(authpriv, mail) and not
match(ppp.*LCP); };
filter f_cron { facility(cron); };
filter f_daemon { facility(daemon); };
filter f_kern { facility(kern); };
filter f_lpr { facility(lpr); };
filter f_mail { facility(mail) and not match (imapd); };
filter f_user { facility(user); };
filter f_uucp { facility(cron); };
filter f_ppp { program(ppp); };
```

```
filter f_news { facility(news); };
filter f_debug { not facility(auth, authpriv, news, mail) and
not match(ppp.*LCP); };
filter f_messages { level(info..warn)
      and not facility(auth, authpriv, mail, news); };
filter f_emergency { level(emerg); };

filter f_info { level(info); };
filter f_notice { level(notice); };
filter f_warn { level(warn); };
filter f_crit { level(crit); };
filter f_err { level(err); };


### OUTPUT ###

log { source(src); filter(f_authpriv); destination(authlog);
};
log { source(src); filter(f_syslog); destination(syslog); };
log { source(src); filter(f_cron); destination(cron); };
log { source(src); filter(f_daemon); destination(daemon); };
log { source(kernsrc); filter(f_kern); destination(kern); };
log { source(src); filter(f_lpr); destination(lpr); };
log { source(src); filter(f_mail); destination(mail); };
log { source(src); filter(f_user); destination(user); };
log { source(src); filter(f_uucp); destination(uucp); };
log { source(src); filter(f_mail); filter(f_info);
destination(mailinfo); };
log { source(src); filter(f_mail); filter(f_warn);
destination(mailwarn); };
log { source(src); filter(f_mail); filter(f_err);
destination(mailerr); };
log { source(src); filter(f_news); filter(f_crit);
destination(newscrit); };
log { source(src); filter(f_news); filter(f_err);
destination(newserr); };
log { source(src); filter(f_news); filter(f_notice);
destination(newsnotice); };
log { source(src); filter(f_debug); destination(debug); };
log { source(src); filter(f_messages); destination(messages);
};
log { source(src); filter(f_emergency); destination(console);
};
log { source(src); filter(f_ppp); destination(ppp); };
log { source(src); destination(console_all); };
log { source(src); destination(all); };
```

Capitolo

4

Scenario di utilizzo

Conoscere le tipologie di attacco e le vulnerabilità, progettare l'infrastruttura di rete, calibrarne ogni singolo componente, controllarne l'attività: Syslog-NG all'interno di una rete.

Introduzione

Domotica, Internet, e-commerce sono tutti termini ormai entrati nel linguaggio quotidiano e, contrariamente a quanto accadeva prima, quando solo le nuove generazioni riuscivano ad adattarsi ai cambiamenti sociologici, ora la rivoluzione tecnologica non ha più rispetto nemmeno per l'anzianità: essa detta le regole e l'unica strada è quella di adattarvisi, a meno di non voler essere, pian piano, espulsi dal mondo (prevalentemente) occidentale.

Qual è ora, quindi, la nuova merce di scambio che questo assetto sociale richiede? Cosa costituisce il bene più importante e, come tale, più ricercato e, quindi, degno di essere tutelato?

La risposta è quasi ovvia: in una società dell'informazione il bene primario sarà costituito dal dato.

Nella multiformità della lingua italiana, sono pochi tuttavia i termini che possano godere di tante svariate definizioni come, invece, accade per la parola "dato". Diverse sfere di influenza, da quella matematica, dove probabilmente questa parola trova il suo luogo di elezione, a quella giuridica, a quella informatica, passando anche per il linguaggio della Pubblica Amministrazione, hanno contribuito a far sorgere in ognuno di noi, l'esatta percezione di cosa sia un dato, senza però darci il privilegio di riuscire a racchiuderlo in una definizione onnicomprensiva.

La faccenda si complica ulteriormente quando, alla parola dato, vengono aggiunti degli aggettivi che, invece di migliorarne la comprensione, riescono soltanto a incanalarlo correttamente nella disciplina di appartenenza, per cui si avrà il dato informatico, il dato personale, il dato sensibile, ecc.

Del dato, però, una cosa è ben chiara: la funzione. Per la fattispecie che maggiormente ci interessa, ossia il diritto personale, possiamo affermare che la

sua funzione è sintetizzabile nel consentire di instaurare una relazione o, comunque, condurre ad un determinato soggetto o classe di soggetti con un margine di errore molto basso o inesistente.

Così come i tratti somatici e le caratteristiche fisiche di una persona sono racchiuse nella catena cromosomica, cosi i dati inerenti la personalità, i gusti e le preferenze sono racchiusi in una catena di dati che, per questa loro natura, prendono il nome, nell'accezione comune, di dati personali.

Si inserisce prepotentemente, a questo punto, il problema del processing dei dati personali, affinché, appunto, riescano a diventare inequivocabilmente la cartina tornasole di un individuo ben determinato o di una categoria di soggetti accomunabili per caratteristiche salienti.

Ormai il ricorso al mezzo informatico per l'elaborazione dei dati è quasi obbligato. La capacità intrinseca di elaborare dati, funzione primaria dell'informatica, può, infatti, risultare molto utile sia limitatamente al dato in se e per si, sia, e soprattutto, nell'ottica di quella attività che sempre più sta prendendo piede: il data mining, sempre più sfruttato per scopi prevalentemente commerciali.

Quest'ultima scoperta delle aziende consente, infatti, mediante apposite correlazioni di dati non necessariamente personali, di tracciare in maniera fedele le preferenze di un individuo e di utilizzarle come meglio si crede.

Che il dato personale rivesta questa funzione, ormai diventata fondamentale nella connected society, come viene chiamata da Negroponte la società attuale, ci viene segnalato anche da un altro mondo: il mondo del diritto.

Da quando è stata emanata la legge di tutela del trattamento dei dati personali, infatti, è stato un continuo susseguirsi di interventi legislativi volti ad integrare, attualizzare o modificare in parte la suddetta legge, oltre ad interventi che ne hanno esteso l'ambito di applicazione in settori dove prima la tutela dei dati personali non era presa assolutamente in considerazione.

Se vi è stato tanto movimento in direzione di una tutela dei dati personali da parte di un mondo storicamente refrattario ad adattarsi tempestivamente ai nuovi stimoli ed alle nuove problematiche quale è il mondo del diritto, allora vuole dire che il tanto descritto valore dietro il dato personale c'è eccome.

Inoltre, in seguito a quanto detto sopra, da una esigenza di tutelare il dato in se e per se, si è passati ad una esigenza di tutelare anche la modalità di trattazione dei dati stessi, e di sanzionare determinate correlazioni che possono violare il principio della privacy.

Si introduce, cosi, un altro termine decisamente abusato, ma forse non perfettamente compreso da tutti: la privacy.

Questo è un termine che ha conosciuto una sua evoluzione, anche e soprattutto nel mondo del diritto.

Formatosi principalmente nell'ordinamento americano, da una concezione iniziale che si può riassumere nell'espressione diritto ad essere lasciato solo si è passati, con varie evoluzioni, ad un diritto a controllare come vengono utilizzate da altri le informazioni che ci riguardano.

Capita di sentirlo usare nei modi più svariati: alcuni lo utilizzano come spauracchio, altri preferiscono dissertare sulle sue implicazioni filosofiche, ma quello che principalmente si vuole indicare è lo stretto legame che c'e tra la sempre più sentita esigenza di privacy e la sicurezza.

Vi sono aziende che arrivano a pagare dei veri e propri "cacciatori di dati", pirati informatici che vengono assoldati appositamente per violare database di dati personali legittimamente detenute da aziende ed è in questo scenario che bisogna innanzitutto proteggere i dati di cui si è in possesso e, secondariamente, esternare la propria volontà affinché, quei determinati dati, non vengano innanzitutto carpiti ed in seguito utilizzati.

Ecco, quindi, affermarsi l'esigenza di sicurezza, definita da alcuni ormai vera ricerca parossistica e paranoica, sfruttata da altri come splendido veicolo commerciale di grande successo per rifilare soluzioni all-in-one che, schiacciando il magico pulsante ON, teoricamente garantiscono protezione assoluta contro gli utilizzi impropri dei dati personali. Purtroppo, rischiando di peccare di scarsa originalità, preme ricordare che la sicurezza è un processo, e non un prodotto acquistabile sugli scaffali, e, soprattutto, che è un processo asintotico, per cui la sicurezza è praticamente irraggiungibile, ma con apposite procedure vi si può avvicinare in maniera determinante.

Per rappresentare nel miglior modo possibile lo stretto legame tra tutela dei dati personali e privacy si è, infatti, preferito disegnare uno scenario ipotetico, ma perfettamente verosimile, conforme alla realtà informatica di diverse aziende e si è voluto vedere come questo scenario poteva essere sollecitato e quali strumenti la tecnica ed il diritto ci mettono a disposizione per tutelare al meglio i dati personali.

L'attacco informatico

Dunque, sicurezza e privacy abbiamo visto essere due aspetti dello stesso problema, collegate alla gestione del "dato" informatico. Si rende, quindi, necessaria una gestione oculata del "dato", nell'ottica di una prevenzione degli utilizzi non autorizzati degli stessi, di qualsiasi natura essi siano. Per non rimanere sulla teoria spicciola, è stato selezionato un caso tra i tanti possibili che è risultato, in ultima analisi, piuttosto generale e diffuso nel panorama italiano, specie nella piccola e media impresa: si sta parlando della gestione delle informazioni in possesso di una azienda che abbia deciso di creare una "vetrina" su Internet, offrendo ai potenziali clienti un sistema di e-commerce per l'acquisto di alcuni prodotti. Le problematiche da affrontare, come si

vedrà, saranno diverse, ma avranno tutti un unico denominatore, la sicurezza degli accessi, che si traduce in una rete – sia pubblica che privata - sicura. Verranno analizzate, qua di seguito, le possibilità di prendere all'interno della rete di un eventuale attaccante, che abbia deciso di sottrarre "informazioni confidenziali" all'azienda.

Cominciamo introducendo una struttura di rete di esempio, che fungerà da scenario per i nostri attacchi. L'azienda virtuale che abbiamo creato per l'occasione si chiama Profumeria RoseNoir Srl: essa si occupa della produzione e della vendita di profumi e, per mantenersi al passo con i tempi, ha deciso di aprire un'area di e-commerce sul proprio sito web http://www.profumi-online.biz allo scopo di permettere alla propria clientela l'acquisto dei suoi prodotti anche tramite Internet. Per far ciò ha acquistato uno spazio di indirizzi IP (diciamo 1.1.1.0/29, corrispondente al range di indirizzi 1.1.1.0-7) ed ha deciso in installare presso la propria struttura tre server Linux destinati alla rete pubblica:

- Un server **mail+DNS** (avente indirizzo l.l.l.2), che eroga i seguenti servizi: **DNS** (Domain Name System) **primario** per il dominio profumi-online.biz, **SMTP** (Simple Mail Transfer Protocol), **POP-3** (Post Office Protocol v3) per la posta, e **SSH** (Secure Shcll) per l'amministrazione remota del sistema.
- Un server **web+DNS** (con indirizzo 1.1.1.3), che offre i seguenti servizi: **DNS secondario** per il dominio profumi-online.biz, **HTTP** ed **HTTPS** per il sito (front-end) web, **MySQL** per il backend, **FTP** (File Transfer Protocol) per fare upload delle pagine HTML e, per finire, **SSH** per l'amministrazione remota del sistema.
- Un **firewall** (con indirizzo pubblico 1.1.1.4 e con indirizzo privato 10.10.10.1), che maschera l'indirizzamento della classe privata 10.10.10.0/24 e permette ai client la navigazione su Internet, senza alcuna particolare restrizione: si tratta di un comune Packet Filter senza funzionalità di Application Proxy.

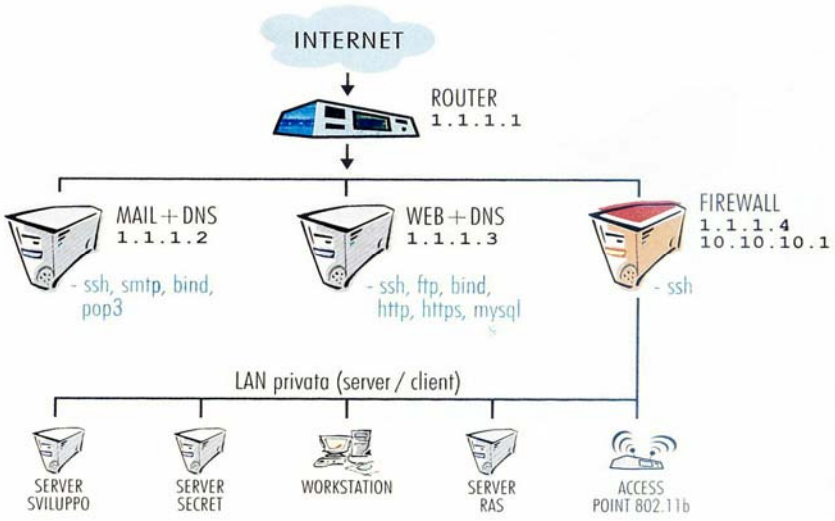


Figura 1: la rete della profumeria RoseNoir. L'azienda dispone di 4 indirizzi pubblici (da 1.1.1.1 a 1.1.1.4), assegnati a tre macchine Linux e ad un router. E' presente una LAN privata con alcuni server di lavoro, oltre ad un RAS server ed un Access Point Wireless in test.

Anche sul firewall e montato un servizio SSH per l'amministrazione remota del sistema che è, però, filtrato per connessioni non provenienti dalla classe 1.1.1.0/29. Oltre a questo, non vi sono altri servizi in ascolto.

All'indirizzo 1.1.1.1 è, inoltre, raggiungibile l'interfaccia del router Cisco che gestisce la connettività Internet verso l'ISP (Internet Service Provider) selezionato. Sulla rete privata (che, come abbiamo gia visto, ha indirizzi compresi nella classe C 10.10.10.0/24) sono attestate le postazioni di lavoro dei dipendenti (Windows-based), un server Linux per lo sviluppo del sito di e-commerce, un altro server Linux (SECRET) contenente informazioni estremamente sensibili (quali, ad esempio, i progetti dettagliati dei nuovi prodotti), un **RAS** (Remote Access Server) per l'accesso degli agenti alla rete privata da numerazione verde 800-XXXXXX ed infine un **Access Point wireless IEEE 802.11b,** attualmente configurato senza particolari misure di sicurezza ed in fase di test per una possibile implementazione futura.

Lo scenario di rete esposto e riassunto nello schema (vedi Figura 1). Prima di passare ad esporre i possibili attacchi che interessano tale struttura di rete, forniamo una breve introduzione teorica agli attacchi informatici, descrivendo le fasi da cui essi sono tipicamente costituiti.

Le fasi dell'attacco

E' possibile fornire una indicazione di massima sulle fasi che costituiscono un tipico attacco. Esse sono rappresentate all'interno dello schema seguente.

A) HIDING (Mascheramento)
Durante questa prima fase, l'attacker compie delle operazioni (a volte anche piuttosto complesse) allo scopo di "camuffarsi", nascondendo la propria reale ubicazione per scongiurare il pericolo di essere tracciato ed identificato. Tipicamente ricorre all'utilizzo di sistemi ponte (detti anche *launchpad*) precedentemente violati, al mascheramento della propria utenza telefonica (ad esempio tramite *dialout*) o ad altri simili trucchi.

B) INFORMATION GATHERING (Raccolto di Informazioni)
Affinché un attacco abbia esito positivo, è fondamentale che l'attacker raccolga il maggior numero di informazioni possibili sulla rete scelta come bersaglio. E' importante notare che in questa fase non viene eseguita alcuna reale intrusione, ma ci si limita a raccogliere le informazioni pubbliche reperibili attraverso canali standard (database del NIC, WHOis record, DNS, traceroute, ping, ecc.).

Le tecniche comunemente utilizzate allo scopo sono:

Network Surveying: si tratta di una sorta di combinazione di ricerca dei dati, accolta di informazioni importanti e controllo delle politiche di sicurezza e di information disclosure. Al termine di questa sottofase, l'attaccante è tipicamente in possesso dei nomi di dominio interessati, delle denominazioni e delle funzioni primarie dei singoli server, degli indirizzi IP di proprietà dell'azienda bersaglio, di una mappa di rete approssimativa, di indirizzi e-mail validi (tipicamente info@, sales@, webmaster@, etc.) e di informazioni specifiche riguardanti l'ISP che fornisce la connettività Internet.

Port Scanning: con il termine port scanning si intende l'attività di probing invasivo delle porte di sistema, a livello di trasporto (layer 4 dello standard di riferimento ISO/OSI). Ogni sistema che implementa il TCP/IP ha 65.536 possibili porte TCP e altrettante UDP: lo scopo di questa fase e fornire un elenco dei sistemi attivi nell'intervallo di indirizzi individuato durante l'attività di Network Surveying ed ottenere una mappa di rete ad un maggiore livello di dettaglio.

Service Identification: si tratta dell'esame attivo delle applicazioni in ascolto sulle porte individuate nel corso della sottofase precedente. A volte ad un servizio può essere associata più di una applicazione; ad esempio, nel caso di un server web, il Perl ed il PHP possono essere considerati componenti dell'applicazione che ascolta sulla porta 80/TCP (il server web Apache). I risultati ottenuti al termine di questa analisi consentono all'attaccante di individuare i possibili punti deboli dell'infrastruttura informatica scelta come bersaglio.

System Identification: con questo termine si intende il probing attivo di un sistema, alla ricerca di risposte che permettano di distinguere la tipologia di sistema operativo, la sua versione ed altre ulteriori informazioni specifiche. Una tecnica comunemente utilizzata allo scopo è quella del TCP/IP fingerprinting, realizzata ad esempio dal noto programma di port scanning nmap. Così come per quelle raccolte nel corso della sottofase di Service Identification, anche queste informazioni consentono all'attacker di ampliare la propria visione sulla rete e sulle sue esposizioni.

C) SYSTEM PENETRATION (Intrusione)

Una volta in possesso delle informazioni necessarie, è il momento per l'attacker di sferrare l'attacco vero e proprio. A seconda delle tipologie di reti, sistemi e servizi erogati, tale attacco può variare sensibilmente dal punto di vista pratico; possiamo però raggruppare le possibili insicurezze in almeno 4 aree distinte:

System security: si tratta di vulnerabilità a livello di sistema operativo e nel software di base. Tipicamente, alcuni servizi e programmi applicativi, datati oppure mal configurati, si prestano ad essere attaccati allo scopo di ottenere un accesso non autorizzato al sistema che ne fa uso (vulnerabilità remote) o di realizzare una scalata di privilegi (vulnerabilità locali). Tipici esempi di attacchi connessi a questa tipologia di insicurezza sono quelli che sfruttano l'esecuzione di codice arbitrario da remoto (tramite impiego di exploit specifici), la mancata

gestione delle eccezioni o, infine, la mancata o scorretta implementazione delle politiche di gestione delle credenziali di accesso (attacchi di tipo "password guessing" o "brute force").

Network security: in questo ambito sono comprese le vulnerabilità che minano il livello di sicurezza globale della rete. Tipicamente, i fattori che concorrono alla determinazione di questo genere di esposizioni sono la tipologia di rete e la conseguente presenza di punti di accesso (potenzialmente sprotetti o non adeguatamente protetti) alla rete privata, la tipologia e la configurazione dei sistemi per il routing ed il firewalling (servizi attivi, ACL, default password e policy del Packet Filler) e la possibilità di effettuare il monitoraggio passivo del traffico alla ricerca di informazioni riservate trasmesse in chiaro (Network Sniffing).

Application security: le insicurezze si possono manifestare anche (e soprattutto) nel software applicativo. In particolare, un gran numero di tecnologie e di implementazioni di applicazioni interne risultano vulnerabili ad attacchi mirati a reperire informazioni riservate o ad ottenere un accesso non autorizzato ai sistemi che erogano i servizi pubblici.
Alcuni esempi di vulnerabilità a livello applicativo sono l'SQL injection, il cross-site scripting, e la possibilità di effettuare session hijacking (almeno per quanto riguarda le applicazioni web).

Procedural security: infine, le stesse procedure volte a migliorare le modalità di gestione della sicurezza possono cadere vittima di falle strutturali all'interno del proprio modello di gestione fornendo, ad esempio, a chi attacca, la possibilità di sfruttare l'esistenza di relazioni di fiducia tra le singole macchine o in relazione all'elemento umano (tramite attacchi basati su social engineering ed human deception).

Per ognuna delle quattro macro-categorie di vulnerabilità descritte, l'attaccante cerca di individuare i punti deboli e colpisce ove risulta più conveniente per la riuscita dell'intrusione informatica, utilizzando i mezzi opportuni.

D) CLEANING (Pulizia)
Una volta portata a termine l'intrusione, è generalmente necessario per l'attacker procedere con la cancellazione delle tracce lasciate nei file di log, su altre regioni del file system e, se possibile, sugli eventuali dispositivi di monitoraggio delle attività di rete per il rilevamento delle intrusioni (Intrusion Detection Systems).

Fatta una breve panoramica sulla teoria di base degli attacchi informatici, possiamo ora passare a descrivere l'andamento di un ipotetico attacco alla rete pubblica (1.1.1.0/29), a partire dalla fase di Information Gathering.

Siccome l'analisi esaustiva di tutto l'iter che contraddistingue un attacco ad un sistema informatico non rientra tra gli scopi di questo documento, abbiamo scelto di tralasciare la descrizione approfondita delle fasi di Hiding e di Cleaning, per concentrarci immediatamente sui più interessanti processi di Information Gathering e System Penetration.

Il nostro attacker, verosimilmente animato dagli interessi di qualche azienda concorrente, ha deciso di attaccare la rete della Profumeria RoseNoir Sri, allo scopo di reperire informazioni utili sui clienti registrati su) sito di e-commerce ed eventualmente di accedere ad altri dati sensibili, che noi sappiamo essere contenuti sul server Linux SECRET attestato sulla rete privata. Dopo aver adeguatamente mascherato la propria reale ubicazione, l'attacker comincia a raccogliere informazioni sull'azienda e sul suo sito web istituzionale, raggiungibile all'indirizzo http://www.profumi-online.biz .

Siamo entrati nella fase di Information Gathering.

Network Surveying: come prima cosa l'attacker utilizza il proprio client DNS per risolvere l'hostname www.profumi-online.biz ad esempio tramite il comando:

```
# host www.profumi-online.biz

www.profumi-online.biz   A    1.1.1.3
```

Una volta in possesso dell'indirizzo IP 1.1.1.3, l'attacker prosegue operando una query al Database del RIPE, con un comando di questo tipo:

```
# whois -h whois.ripe.net 1.1.1.3
% This is the RIPE Whois server.
% The objects are in RPSL format.
%
% Rights restricted by copyright
% See http://www.ripe.net/ripencc
%     /pub-services/db/copyright.html

inetnum:    1.1.1.0 - 1.1.1.7
netname:    ROSENO-NET
descr:      Profumeria RoseNoir Srl
country:    IT
admin-c:    […]
tech-c:     […]
status:     ASSIGNED PA
mnt-by:     ISP-NCC
notify:     noc@isp
changed:    noc@isp 20020711
source:     RIPE

route:      1.1.0.0/19
descr:      ISP
origin:     […]
remarks:    ALLOCATED PA Space do not break up
notify:     noc@isp
mnt-by:     ISP-NCC
changed:    noc@isp 20020711
source:     RIPE

person:     Giorgio Rossi
address:    Profumeria RoseNoir Srl
address:    via Roma 00
address:    00000 Milano (MI)
address:    Italy
phone:      +39 02 XXXXXXX
fax-no:     +39 02 XXXXXXX
e-mail:     Giorgio@profumi-online.biz
nic-hdl:    […]
changed:    noc@isp
source:     RIPE

person:     Giuseppe Rossi
address:    Profumeria RoseNoir Srl
address:    via Roma 00
address:    00000 Milano (MI)
address:    Italy
phone:      +39 02 XXXXXXX
fax-no:     +39 02 XXXXXXX
e-mail:     beppe@profumi-online.biz
nic-hdl:    […]
changed:    noc@isp
source:     RIPE
```

Come si vede da questo esempio creato ad hoc, l'attacker è in grado di reperire una grande quantità di informazioni attraverso un semplice comando per l'interrogazione del Database del RIPE, l'ente preposto all'assegnazione degli indirizzi IP per l'Europa.

Oltre alla classe di proprietà dell'azienda bersaglio, sono visibili altri dati importanti, quali la sede della società, i numeri telefonici e del fax, i nomi dei titolari ed i loro indirizzi e-mail. Infine, la query al servizio WHOIS restituisce anche delle informazioni specifiche sull'ISP che fornisce la connettività Internet.

Analogamente, è possibile effettuare una query anche al WHOIS dell'autorità preposta all'assegnazione dei nomi di dominio: nel caso del TLD (Top Levei Domain,) .biz occorre rivolgersi direttamente a Network Solutions, come segue:

```
# whois -h whois.networksolutions.com profumi-online.biz
[…]
Domain Name:                  PROFUMI-ONLINE.BIZ
Domain ID:                    […]
Sponsoring Registrar:         […]
Domain Status:                ok
Registrant ID:                […]
Registrant Name:              Giorgio Rossi
Registrant Organization:      Profumeria RoseNoir Srl
Registrant Address1:          via Roma 00
Registrant City:              Milano
Registrant State/Province:    Milano
Registrant Postal Code:       00000
Registrant Country:           Italy
Registrant Country Code:      IT
Registrant Phone Numerico:    +39.02XXXXXXX
Registrant Facsimile Numerico:      +39.02XXXXXXX
Registrant Email:             g.rossi@libero.it
[…]
Name Server:                  DNS.PROFUMI-ONLINE.BIZ
Name Server:                  DNS2.PROFUMI-ONLINE.BIZ
Created by registrar:         […]
[…]
```

Anche in questo caso l'attacker ha a sua disposizione molte informazioni interessanti: in particolare, un nuovo indirizzo e-mail su libero.it ed i due server DNS registrati come autoritativi per il dominio in esame. E' giunto il momento di interrogare i name server individuati, ad esempio, tramite il seguente comando (l'output completo è riportato nel riquadro alla pagina successiva):

```
# host -t ns profumi-online.biz
```

Come accade piuttosto frequentemente, i name server autoritativi per il dominio interessato permettono a chiunque di scaricare i dati relativi alle zone

```
# host -t ns profumi-online.biz     #name server
profumi-online.biz      NS    dns.profumi-online.biz
profumi-online.biz      NS    dns2.profumi-online.biz

# host -t mx profumi-online.biz     # mail exchange
profumi-online.biz      MX    20 mail.profumi-online.biz

# host -l profumi-online.biz        # zone transfer
profumi-online.biz      NS    dns.profumi-online.biz
profumi-online.biz      NS    dns2.profumi-online.biz
profumi-online.biz      MX    20 mail.profumi-online.biz
gw.profumi-online.biz   A     1.1.1.1
mail.profumi-online.biz A     1.1.1.2
www.profumi-online.biz  A     1.1.1.3
dns.profumi-online.biz  A     1.1.1.2
dns2profumi-online.biz  A     1.1.1.3
fw.profumi-online.biz   A     1.1.1.4
```

di dominio: nel nostro caso, l'attacker è in grado di visualizzare tutti gli hostname registrati sotto il dominio profumi-online.biz.
Ciò gli consente di ampliare ulteriormente la sua visione della rete bersaglio, in particolare per quanto riguarda la funzione degli host attivi (gateway, web, mail, DNS, firewall). E' possibile tentare lo stesso tipo di query anche sulla classe di riverse 1.1.1.in-addre.arpa, allo scopo di reperire ulteriori informazioni. Infine, a volte capita che sul DNS esterno siano caricate anche le zone relative alla classe interna (sia record A che record PTR), senza l'attivazione delle opportune ACL (Access Control List): questa configurazione consente ad un attacker di ottenere dall'esterno informazioni precise sull'indirizzamento e sulle funzioni dei sistemi attestati sulla rete privata (nel nostro caso 10.10.10.0/24).

Altri tool utili per ottenere dati preziosi sulla rete bersaglio sono **traceroute**, **ping**, **hping** ed un comune browser web da utilizzare per visitare il sito istituzionale dell'azienda bersaglio.

Per finire l'attacker procede con l'indentificazione degli host attivi, attraverso un *ping sweep*:

```
# nmap -sP 1.1.1.0/29

Starting nmap V.2.54BETA30 www.insecure.org/nmap)
Host gw.profumi-online.biz (1.1.1.1) appears to be up
Host dns.profumi-online.biz (1.1.1.2) appears to be up
Host dns2.profumi-online.biz (1.1.1.3) appears to be up
Host fw.profumi-online.biz (1.1.1.4) appears to be up
```

Una volta identificate le 4 macchine raggiungibili è il momento di esaminare I servizi da esse erogati.

Port Scanning: per ognuno dei 4 sistemi pubblici individuati, l'attacker lancia un port scan alla ricerca dei servizi attivi:

```
# nmap 1.1.1.1    # gateway
Starting nmap V.2.54BETA30 www.insecure.org/nmap)
Interesting ports on gw.profumi-online.biz (1.1.1.1):
Port        State       Service
79/tcp      open        finger

# nmap -sU 1.1.1.1      # gateway
Starting nmap V.2.54BETA30 www.insecure.org/nmap)
Interesting ports on gw.profumi-online.biz (1.1.1.1):
Port        State       Service
161/tcp     open        snmp

# nmap 1.1.1.2    # mail + DNS
Starting nmap V.2.54BETA30 www.insecure.org/nmap)
Interesting ports on dns.profumi-online.biz (1.1.1.2):
Port        State       Service
22/tcp      open        ssh
25/tcp      open        smtp
53/tcp      open        domain
110/tcp     open        pop-3

# nmap 1.1.1.3    # web + DNS
Starting nmap V.2.54BETA30 www.insecure.org/nmap)
Interesting ports on dns2.profumi-online.biz (1.1.1.3):
Port        State       Service
21/tcp      open        ftp
22/tcp      open        ssh
53/tcp      open        domain
80/tcp      open        http
443/tcp     open        https
3306/tcp    open        mysql

# nmap 1.1.1.4    # firewall
Starting nmap V.2.54BETA30 www.insecure.org/nmap)
Interesting ports on fw.profumi-online.biz (1.1.1.4):
All 1549 scanned ports on fw.profumi-online.biz (1.1.1.4)
are: filtered
```

Come si nota, i due server pubblici (mail+DNS e web+DNS) offrono un certo numero di servizi pubblici, mentre il router permette solamente l'utilizzo di FINGER e SNMP. Il firewall, infine, filtra tutte le connessioni.

Service Identification: per ognuna delle porte in ascolto individuate è ora necessario identificare i servizi corrispondenti e le relative versioni, alla ricerca di un punto di accesso ai sistemi. Per fare ciò si utilizzano tipicamente dei tool multi-purpose (telnet, netcat, etc.) o specifici per un determinato servizio (dig, rpcinfo, etc.). Nel nostro caso, l'attacker è in grado di compilare una mappa dei servizi attivi, di cui conosce le vulnerabilità e le esposizioni.

Senza entrare nei dettagli implementativi (non ci interessa trattare le vulnerabilità delle specifiche implementazioni di ogni singolo protocollo),

forniamo una tabella di riferimento sulle principali esposizioni relative ai servizi individuati nel corso dell'analisi (vedi Appendice).

Infine, come abbiamo già accennato, è possibile utilizzare il meccanismo di TCP/IP fingerprinting messo a disposizione da mnap, allo scopo di identificare le versioni di sistema operativo presenti sui server :

```
# nmap -O 1.1.1.1 # gateway
[…]
Remote OS guesses:      Cisco IOS 11.3 - 12.0(11)

# nmap -O 1.1.1.2 # mail + dns
Remote operating system guess:      Linux 2.3.x - 2.4.x
Uptime 25.983 days (since Thu Nov 28 18:21:02 2002)
[…]

# nmap -O 1.1.1.3 # web + dns
Remote operating system guess:      Linux 2.3.x - 2.4.x
Uptime 25.983 days (since Thu Nov 28 18:21:09 2002)
[…]

# nmap -O 1.1.1.4 # firewall
Remote operating system guess:      Linux 2.3.x - 2.4.x
Uptime 1.452 days (since Mon Dec 23 1:00:43 2002)
[…]
```

Come si vede, il router (1.1.1.1) è con ogni probabilità un Cisco, mentre i 3 server sono dei Linux 2.4: di questi ultimi è addirittura, possibile risalire all'uptime.

In particolare l'ora di avvio del firewall può far pensare ad uno scheduling automatizzato del reboot (giornaliero o settimanale, alle 01:00 del mattino). Anche questa informazione, unitamente a quelle già raccolte, può risultare di grande utilità ad un attacker smaliziato. Lasciamo ai più fantasiosi il compito di immaginare il modo di sfruttare questo genere di configurazione. Terminata la fase di Information Gathering, passiamo ora ad analizzare le possibilità di intrusione individuate sino a questo momento.

Attacco alla rete pubblica: System Penetration

Dopo aver raccolto tutte le informazioni utili del caso, è giunto il momento per l'attacker di sferrare l'attacco vero e proprio ai sistemi selezionati come bersaglio. E' importante notare come l'attacco a sistemi protetti nella maniera corretta non sia quasi mai diretto, ma richieda spesso la correlazione di più di una vulnerabilità ai fini di guadagnare l'accesso alla rete bersaglio: per questo motivo, l'attacker può decidere di non attaccare direttamente i propri bersagli primari (il database MySQL e la rete privata), scegliendo, invece, di giungervi attraverso una via privilegiata (rappresentata, ad esempio, da un altro server attestato nella rete di proprietà della Profumeria).

A beneficio dei meno esperti, chiariamo che la compromissione dei privilegi di amministrazione di un server consente l'accesso illimitato ai dati ivi contenuti, in lettura e scrittura: una volta ottenuto questo tipo di accesso, un attacker ha la possibilità di consultare, alterare, cancellare ogni informazione conservata sul server compromesso, con ovvie implicazioni sulle attività dell'azienda vittima.

Inoltre, è solo tramite i privilegi di amministratore che è possibile creare pacchetti con indirizzi sorgenti diversi da quelli posseduti dalla PC compromesso: questo permette l'utilizzo di programmi per attacchi tipo Man-in-the-Middle e altro: il tutto per cercare di intercettare la maggior quantità di traffico di rete possibile alla ricerca di ulteriori credenziali per poter compromettere la rete privata.

A seconda dei risultati ottenuti nel corso della fase di Information Gathering, l'attacker ha la possibilità di scegliere uno o più dei seguenti approcci per portare a termine l'attacco ai server in DMZ (in particolare al database MySQL) e per ottenere l'accesso alla rete privata della propria vittima.

INFORMATION LEAK: è possibile ricavare informazioni utili sulle utenze valide presenti sul router Cisco (all'indirizzo IP 1.1.1.1) e sul server mail+DNS (1.1.1.2), sfruttando rispettivamente i servizi di FINGER e SMTP. In questo modo l'attacker può aumentare le probabilità di riuscita di un attacco di tipo brute farce o password guessing mirato al router ed al mail server, che rappresentano due ideali trampolini di lancio per sferrare l'offensiva verso i bersagli primari (il database e la rete privata).

EXPLOIT: a seconda della tipologia e della versione dei demoni in ascolto relativi ai servizi FTP, SSH, DNS, HTTP, HTTPS e MYSQL, l'attacker può essere in possesso di programmi per l'intrusione remota sui sistemi che ti utilizzano.

Se l'accesso ottenuto non è privilegiato, è inoltre possibile utilizzare in un secondo tempo exploit locali per ottenere i permessi di amministrazione (root, su sistemi Unix-like).

BRUTE FORCE: infine, è possibile lanciare un attacco a forza bruta sul meccanismo di autenticazione dei servizi FTP, SSH, POP-3 e MYSQL, allo scopo di ottenere credenziali di accesso valide per i sistemi bersaglio.

Network securìty

SNMP SPOOFING: SNMP utilizza UDP come protocollo a layer di trasporto (layer-4 del modello di riferimento ISO/OSI).

Tale protocollo è suscettibile ad attacchi del tipo spoofing o hijacking di sessione: in particolare, se si è a conoscenza delle community SNMP valide con permessi (tw) e degli indirizzi IP accettati dal servizio a livello di ACL (informazioni in larga parte note per carrier di grandi dimensioni o ottenibili

attraverso Network Sniffing), è possibile modificare remotamente la configurazione del router ed ottenere così l'accesso non autorizzato all'apparato di rete, che rappresenta un ottimo launchpad privilegiato da cui condurre i successivi attacchi.

SNIFFING: alcuni dei protocolli presenti nella rete bersaglio (FTP, SMTP, HTTP, POP-3, SNMP, MYSQL) non implementano la cifratura del traffico: qualora l'attacker riuscisse ad ottenere l'accesso privilegiato ad uno solo dei sistemi attestati sulla rete, sarebbe in grado di monitorare tutto il traffico, catturando le credenziali di accesso in transito anche verso altri sistemi. Tale situazione sarebbe ulteriormente aggravata nel caso di impiego di hub al posto di switch.
Tutto ciò va a conferma del celebre detto: "*A chain is only as strong as its weakest link*" (ovvero "*Basta un solo sistema vulnerabile per determinare la compromissione di tutta la rete*").

PUNTI DI ACCESSO: come si nota subito osservando lo schema della rete bersaglio, vi è un evidente punto di accesso alla rete privata, rappresentato dal firewall all'indirizzo 1.1.1.4. La compromissione di tale host dual-homed offre la possibilità ad un attacker esterno di accedere ai sistemi attestati sulla rete privata.

Application security

WEB APPLICATION: a seconda della tecnologia utilizzata per implementare l'applicazione di e-commerce su web (Perl, PHP, servlet Java) è possibile per l'attacker individuare delle esposizioni e delle vulnerabilità che consentano di ottenere l'accesso non autorizzato al sistema web (1.1.1.3).

Analogamente, anche la realizzazione del servizio è importante: degli eventuali bug (a livello progettuale o implementativo) possono portare alla compromissione dell'applicazione e delle informazioni sensibili da essa processate (come dati personali, numeri di carte di credito e così via) e consentire frodi su piccola e vasta scala.

Procedural security

TRUST RELATIONSHIP: l'attacker può sfruttare la presenza di relazioni di fiducia tra gli host allo scopo di ottenere l'accesso su più sistemi.

Nel nostro caso, ad esempio, potrebbe introdursi in uno dei due server (mail o web) sfruttando una vulnerabilità remota del DNS, per poi accedere alla porta 22/ TCP aperta sul firewall solo per connessioni provenienti dalla classe 1.1.1.0/29.

In questo modo otterrebbe una posizione privilegiata per sferrare un attacco al firewall, allo scopo di penetrare nella rete privata.

RICICLO DI PASSWORD: troppo spesso gli amministratori di sistema e gli utenti comuni tendono a riciclare le stesse credenziali di accesso su più sistemi. Questo comportamento ovviamente consente ad un attacker di compromettere un maggior numero di macchine con minimo sforzo, attraverso la tecnica del Password Cracking.

Abbiamo descritto alcune delle possibili strade di attacco esterne volte a compromettere la sicurezza dei sistemi attestati sulla rete bersaglio. Non si tratta, ovviamente, di una esposizione esaustiva di tutte le tipologie di attacco possibili: a seconda delle motivazioni che muovono l'attacker, le sue tecniche possono variare sensibilmente.

Nel nostro esempio, abbiamo deciso di descrivere il comportamento di un cracker avente una certa esperienza, interessato ad un certo tipo di informazioni riservate riguardanti clienti e prodotti della Profumeria: questa tipologia di attacker è chiaramente interessata ad effettuare una intrusione che passi il più possibile inosservata, pertanto essa non comporterà un danno immediato ai sistemi coinvolti in termini di disponibilità dei servizi.

Se ci fossimo trovati a descrivere la vendetta personale di un ex-dipendente, invece, l'attacco avrebbe preso una piega molto diversa: tipicamente, avremmo dovuto affrontare attacchi di tipo DoS (Denial of Service, o negazione di servizio); essi sono, infatti, generalmente più semplici da attuare e mirano a compromettere l'efficienza o la disponibilità di un servizio, una macchina o una intera rete (senza la necessità di effettuare una intrusione vera e propria sui sistemi bersaglio). Passiamo, ora, ad esaminare gli altri punti di accesso alla rete privata, per poi esporre l'andamento di un ipotetico attacco interno.

Attacco alla rete privata

Oltre al firewall (1.1.1.4), che abbiamo già esaminato, vi sono altri punti di accesso alla rete privata che risultano piuttosto evidenti dal diagramma della struttura di rete. Quello che un ipotetico attaccante deve fare nell'ottica di un attacco interno è enumerarli (come parte del processo di Information Gathering), per poi tentare l'intrusione. Nel nostro caso vi sono altri due punti di accesso alla rete privata:

Il server RAS, che accetta connessioni da rete telefonica su numerazione verde 800-XXXXXX. Tale servizio, teoricamente riservato agli agenti ed ai legittimi amministratori di sistema, può essere sfruttato da un attacker per ottenere un accesso non autorizzato alla rete bersaglio, direttamente su classe privata 10.10.10.0/24;

L'Access Point 802.11b. La tecnologia wireless è sempre più diffusa anche nelle aziende italiane: nel nostro caso, non vi è una vera e propria infrastruttura Wi-Fi, poiché l'Access Point è temporaneamente in fase di test nell'ottica di una eventuale implementazione futura. Esso è, però, poco prudentemente collegato alla rete privata e non è configurato per utilizzare particolari sistemi

di protezione (WEP, accettazione dei soli MAC Address delle schede wireless aziendali ed altri accorgimenti).

Ognuno di questi punti di accesso può rappresentare la strada ideale per un attacker esterno motivato, che voglia ottenere l'accesso diretto alla rete privata. Inoltre, è molto importante notare che un gran numero di attacchi documentati sono ad opera di insider, ovvero personale interno all'azienda con un accesso legittimo alla rete.

Abbiamo deciso di non trattare questo tipo di attacchi, ma è fondamentale che una infrastruttura di sicurezza si curi anche di questa particolare tipologia di intrusione con posizione privilegiata.

Non dimentichiamo, infatti, che l'insider, indipendentemente dalle sue motivazioni, con ogni probabilità conosce bene la struttura della rete bersaglio, sa che cosa sta cercando, ne conosce il valore e ha una idea di come utilizzarlo.

Vediamo, ora, brevemente come sia possibile scoprire e sfruttare i due ulteriori punti di accesso alla rete privata presenti nel nostro scenario di esempio.

Nel caso del server RAS su numerazione verde, la scoperta può avvenire casualmente (attraverso War Dialing su 800*) o a seguito di una operazione mirata (PBX scanning, Social Engineering, consultazione di informazioni per gli agenti presenti sul sito web istituzionale o sulla Intranet o, ancora, sniffing di messaggi e-mail in transito o compromissione di singole caselle di posta elettronica).

Una volta in possesso del numero telefonico del RAS, l'attacker è comodamente in grado di impersonare un agente, dipendentemente dalla politica di gestione delle password per l'accesso al dial-in e dalla soluzione (hardware e software) scelta per l'implementazione di tale servizio. L'attacco portato attraverso la rete wireless è, invece, tipicamente costituito da una sessione di War Driving (l'equivalente del War Dialing on the road, di cui si sente tanto parlare in questi ultimi mesi) e Wireless Leak Test per poi concludersi con la penetrazione nella rete privata e l'eventuale installazione di backdoor per il comodo accesso da Internet (nel nostro caso un ottimo posto per installare una backdoor sarebbe il firewall o, in alternativa, il server di autenticazione per il dial-in). Una volta ottenuto l'accesso alla rete privata utilizzando una qualsiasi delle tecniche appena esposte, è tipicamente molto semplice compromettere i sistemi ivi attestati.

Raramente, infatti, i client ed i server interni ricevono le cure e le attenzioni che meritano da parte degli amministratori di sistema e dei Security Manager, che li ritengono (erroneamente) protetti in modo soddisfacente dal firewall perimetrale.

Come abbiamo visto, a seconda dei casi può essere anche molto facile aggirare tale tipo di protezione. Le tecniche utilizzate per compromettere i sistemi su

rete privata seguono generalmente gli stessi principi esposti nel caso dell'attacco alla rete pubblica, con alcune differenze:

- la fase di Information Gathering è molto più veloce, proporzionalmente alla dimensione della rete privata;
- i servizi attivi sono molti di più, così come i sistemi potenzialmente vulnerabili ad attacchi noti (locali o remoti);
- gli aggiornamenti del software sono in genere meno frequenti;
- anche i dispositivi di rete (hub, switch, etc.) sono spesso scarsamente protetti;
- l'analisi passiva del traffico (Network Sniffing) è generalmente molto più redditizia in un ambiente di rete privato;
- sono più frequenti le relazioni di fiducia tra sistemi, rispetto a quanto accade su rete pubblica Internet;
- i controlli di sicurezza (nella fattispecie IDS ed altre tecnologie anti-intrusione) sono molto più rilassati.

La descrizione dettagliata delle varie possibilità di attacco interno esula dagli scopi di questo lavoro: basti sapere che, come già largamente affermato, in linea generale tale tipologia di intrusione è decisamente più semplice da portare a compimento rispetto a quella operata su rete pubblica.

Il Firewall

Spesso, discutendo della sicurezza di rete, il mantra più abusato in assoluto è probabilmente, firewall.

"E' necessario implementare un firewall", "non è consigliabile presentare la propria azienda in Internet senza la protezione di un firewall", "se vogliamo migliorare la qualità e la resa del nostro lavoro di rete abbiamo bisogno di un firewall".

Ma, in sostanza, cos'è un firewall? Innanzitutto, è necessario definire subito la palese e razionalmente ovvia mancanza di un unico strumento di sicurezza che sia in grado di risolvere ogni problema, sia esso teorico o pratico. Nessun modello implementativo, nessuna policy può garantire la massima sicurezza di ogni aspetto lavorativo in rete.

Il miglior firewall non eviterà che un dipendente copi dati riservati a mano e li spedisca via e-mail da casa propria, con comodo. Digerito questo, il firewall è forse l'elemento cardine, quanto meno in termini descrittivi, della sicurezza di rete. E' semplice definire cosa non sia un firewall.

Un firewall non è solamente un software, non è semplicemente una black-box hardware, non è, in termini implementativi, nemmeno un insieme di hardware e software.

Un firewall è un modello, una architettura, una astrazione. L'implementazione della quale richiede una amalgama di codice, hardware, policy e decisioni amministrative irrinunciabili. In termini più semplicistici, un firewall è un componente o un insieme di componenti, che controlla e restringe le comunicazioni tra una rete interna ed Internet o tra differenti reti interne (nei diagrammi classici di esempio; in realtà per estensione funziona come controllo sulla comunicazione tra due peer o entità). Esistono, nel mondo commerciale e nel mondo OpenSource, molteplici soluzioni che hanno portato a pensare che un firewall sia un software personale piuttosto che un singolo componente hardware costoso e complesso.

Spesso, la maggior parte di queste soluzioni introducono definizioni aggiuntive che fanno parte del modello del firewalling: sistema bastione, filtro di pacchetti, NAT, proxy, rete perimetrale, VPN.
Ogni definizione presenta concetti e soluzioni a problemi e richieste inerenti la sicurezza di rete; queste tecnologie non sono esclusive, ma piuttosto sono combinate in un sistema ridondante e versatile (e, spesso, inutilmente complesso).

Bastion host

Con questa definizione si intende un sistema che deve essere altamente rafforzato per risultare il maggiormente sicuro possibile in quanto è il bersaglio più in vista, sia per attacchi portati dall'esterno della propria rete, sia dall'interno.

Questo accade in quanto ospita servizi raggiungibili da Internet e, spesso, è il primo punto di contatto per gli utenti delle reti interne.

Filtro di pacchetti (Packet Filter)

II packet filter è un dispositivo in grado di controllare selettivamente la comunicazione tra due peer o tra Internet e la rete interna e viceversa, analizzando le caratteristiche dei pacchetti di dati in transito, prima di operare un instradamento degli stessi. Questa selettività è permessa da un insieme di regole che definiscono il comportamento in base a parametri specifici dei datagrammi quali, ad esempio, gli indirizzi IP e le porte.

NAT (Network Address Translation)

NAT è l'acronimo di Network Address Translation (Traduzione di Indirizzi di Rete) indica una procedura che permette di modificare gli indirizzi contenuti nei datagrammi di rete, permettendo così di associare sistemi interni sprovvisti di indirizzi IP pubblici ad un unico indirizzo pubblico o ad un set limitato di indirizzi ed, al contempo, di nascondere i reali indirizzi IP da un lato della rete

che sfrutti tale procedura. Non è propriamente una metodica di sicurezza, ma aumenta le difficoltà per un attaccante, limitando i bersagli possibili e offuscando la topologia reale di rete. Un uso possibile di questa tecnologia è la configurazione di un proxy trasparente.

Proxy

Questo è un dispositivo in grado di dialogare con server esterni per conto di client interni della propria rete. I client comunicano con il proxy che trasmette richieste valide ai server per proprio conto, per poi replicare le risposte verso i client.

Rete perimetrale (De-Militarized Zone)

Spesso definita anche DMZ, acronimo di De-Militarized Zone, Zona Smilitarizzata, è una rete inserita tra un ulteriore segmento protetto, interno, ed un'altra rete esterna, come livello ridondante di sicurezza, aumentando il processo di enumerazione e attacco degli attaccanti, al contempo riducendo i rischi per la rete interna in caso di compromissione dei punti di accesso a Internet.

VPN (Virtual Private Network)

Acronimo di Virtual Private Network, Rete Privata Virtuale, identifica una rete in cui i datagrammi di competenza di sistemi interni, passano attraverso reti esterne, pubbliche, senza che questo sia ovvio o visibile ai sistemi interni.

Le comunicazioni in transito attraverso reti esterne sono crittate, per evitare analisi passive e manipolazioni. Le VPN offrono una interessante opportunità, in termini economici, per collegare differenti reti - geograficamente disperse - attraverso Internet, garantendo autorizzazione, accesso e accounting, senza utilizzare costosi collegamenti dedicati.

Vediamo, ora, nel dettaglio le tecnologie più comuni presenti nel kernel Linux che permettono di configurare e gestire due componenti essenziali di un firewall: il filtro di pacchetti e la tecnologia NAT.

Pro e contro delle più diffuse tecnologie

Bastion host

Vantaggi:

• un singolo filtro può garantire la sicurezza di una intera rete interna;
• il filtro è un sistema molto efficiente, se confrontato, in termini computazionali e di latenza di rete, con il proxy;
• è una tecnologia ampiamente disponibile per ogni piattaforma e architettura.

Svantaggi:

• il sistema di regole di filtro è spesso difficile da ideare e testare;
• spesso il filtro penalizza le performance di un router in maniera evidente;

• i parametri associati ai processi decisionali non sono granulari: un indirizzo IP sorgente non può essere univocamente associato ad un utente specifico.

PROXY

Vantaggi:

• buona capacità di log, specifica per ogni applicazione;
• possibilità di caching dei dati;
• migliori capacità di controllo sul tipo di dato trasmesso, rispetto ad un normale filtro di pacchetti (HTTP Content-Type, virii...);
• migliori possibilità di accounting e autenticazione granulare a livello utente rispetto al normale filtro di pacchetti;
• protezione per client con implementazioni TCP/IP errate o non stabili.

Svantaggi:

• non sono sempre disponibili proxy per una applicazione specifica: è possibile usare proxy a livello di circuito, come SOCKS, perdendo alcuni dei vantaggi più importanti di un sistema proxy;
• i servizi di proxy potrebbero richiedere l'uso di più di un server da configurare e amministrare;
• l'uso dei proxy richiede, spesso, la modifica di client, applicazioni e procedure.

VPN

Vantaggi:

• migliore segretezza per le comunicazioni, rispetto all'uso della crittografia a livello applicativo: in una VPN il traffico è interamente crittato, rendendo impossibile, a livello teorico, risalire a quali sistemi interni stiano dialogando e a cosa stiano trasmettendo;
• l'uso delle VPN permette di fornire strati di sicurezza a protocolli non facilmente utilizzabili mediante filtri di pacchetti o proxy: l'esempio più comune è quello dei protocolli Microsoft basati su SMB, che, mediante le stesse porte e connessioni, identifica svariati tipi di servizi con differenti necessità in termini di sicurezza.

Svantaggi:

• la VPN sfrutta canali pubblici, come tramite della rete interna: utenti mobili attraverso la normale rete PSTN o altri sistemi connessi a Internet possono essere compromessi indipendentemente dalla VPN, ma l'effetto di tale compromissione colpirà anche la rete privata;
• le VPN estendono quindi la rete da controllare e proteggere: potrebbe essere necessario escludere la VPN dalla rete interna fisica mediante una ulteriore rete perimetrale con separato punto di accesso e controllo al segmento interno.

NAT

Vantaggi:

• aumenta il controllo su connessioni dalla rete interna verso l'esterno, in quanto i sistemi interni non dispongono di indirizzi validi al di fuori del loro segmento interno;
• permette di ridurre il traffico verso sistemi intemi, se non è presente una associazione statica degli indirizzi IP, riducendo la finestra a disposizione dell'attaccante in termini di tempo e di traffico;

• offusca la configurazione della rete interna.
Svantaggi:
• la traduzione richiede informazioni sullo stato delle connessioni che non sono sempre disponibili (datagrammi UDP, ad esempio);
• le funzioni di NAT operano sulle intestazioni dei pacchetti, ma alcuni protocolli inseriscono tali indirizzi anche nei payload, rendendo difficile e "personalizzato" il lavoro di NAT (ad es. FTP e DCC);
• non funziona con protocolli che combinino indirizzi contenuti nei payload e sistemi di protezione dell'integrità dei dati, in quanto il payload deve essere modificato per garantire il NAT;
• la correlazione dei log richiede una notevole sincronizzazione temporale e amministrativa (gli indirizzi nei log cambiano a monte e a valle del dispositivo di NAT).

Firewall e visioni architetturali

Possedere la corretta dotazione di software e hardware non è sufficiente. Fondamentale, durante la corretta messa a punto di un firewall, è senza ombra di dubbio la scelta di una architettura che ben si amalgami con la rete esistente.

Esistono sostanzialmente almeno tre diverse architetture possibili per l'allestimento di un firewall: a router di controllo, a sistema controllato, a sottorete controllata.

L'architettura a router di controllo prevede un singolo punto di accesso e controllo che separa la rete interna dal resto di Internet. Spesso, questa macchina è un router dotato di funzioni di filtro di pacchetti ma, in alcuni casi, è possibile siano presenti funzionalità di proxy a livello applicativo.

Questa architettura è poco flessibile per chiunque preveda di fornire servizi pubblici con accesso diretto da Internet, in quanto i server sono in diretto contatto con il segmento interno.

E', d'altra parte, la scelta più comune per singole utenze e piccole società che forniscono i loro servizi attraverso macchine in housing fuori sede.

L'architettura a sistema controllato prevede la presenza di un singolo sistema, opportunamente predisposto, a fare da bastion host; questo è collegato alla rete interna, non esiste una reale rete perimetrale ed il router di ingresso opera un filtro di pacchetti affinché l'unico sistema raggiungibile direttamente da Internet sia il sistema bastione.

Si provvede, poi, alla comunicazione con gli altri host della rete interna. Il precedente scenario di esempio non rappresenta questa architettura.

L'approccio a sottorete controllata, invece, introduce il concetto di rete perimetrale con due router, uno esterno ed uno interno, con la presenza di uno o più sistemi bastione raggiungibili da Internet per poter offrire servizi pubblici.

Nello scenario di esempio, abbiamo una rete perimetrale, due sistemi bastione e un terzo sistema ibrido tra un bastion host ed un router.

Manca un elemento fondamentale: un router interno vero e proprio. Questo è sostituito dal sistema ibrido con IP 1.1.1.4 associato all'interfaccia esterna che traduce, mediante NAT, gli indirizzi dei sistemi del segmento da proteggere.

Questo può essere un azzardo che avvicina il nostro scenario, concettualmente parlando, all'architettura a host controllato. Infatti, la rete interna non prevede divisioni tra sistemi di amministrazione, sviluppo, gestione e macchine dei dipendenti; esiste un notevole traffico tra il segmento interno e la rete perimetrale, in quanto i server DNS e di posta sono situati nei sistemi bastione direttamente raggiungibili da Internet.

Se la rete perimetrale fosse separata da un ulteriore router interno, la compromissione della macchina con IP 1.1.1.4 non porterebbe un attaccante all'analisi del traffico del segmento interno, ne tanto meno alla sua sovversione.

Come già notato in precedenza, quello è il bersaglio più interessante, offrendo servizi remoti ed essendo collegato alla rete interna. Vediamo, dunque, una differente architettura di firewalling applicata allo scenario in esame: l'obiettivo e quello di ottenere una migliore efficienza in termini prestazionali cercando, al tempo stesso, di separare maggiormente il traffico interno e privilegiato, da quello esterno e pubblico.

Una limitazione autoimposta è la mancanza di differenti router esterni provenienti da differenti fornitori per dividere totalmente il traffico amministrativo, quello interno degli impiegati e quello pubblico e dei clienti.

Utilizzeremo, pertanto, un solo router di ingresso, senza affrontare questioni come ridondanza default-tolerance; per motivi di semplicità imporremo alcune restrizioni alle possibilità di collegamento interno, cercando di separare, per quanto possibile, le macchine di amministrazione da quelle di ricerca e sviluppo e da quelle dei dipendenti senza modificare le configurazioni dei singoli host e senza incidere eccessivamente in termini di costi di allestimento e gestione. In questa presentazione non terremo conto di altre scelte altrettanto importanti. L'uso di entrambi i server DNS nella rete perimetrale non ha molto senso, dal momento che è facilmente configurabile un unico server su di un sistema bastione, autoritativo per il dominio ma, in realtà, in possesso delle sole informazioni sui sistemi perimetrali, mentre le informazioni relative ai sistemi interni possono essere comodamente gestite su di un server interno, maggiormente protetto, in grado di operare forwarding delle query verso il DNS perimetrale.

In questo modo è possibile nascondere utili informazioni agli attaccanti, mantenendo il controllo e la protezione del server più importante all'intemo del segmento di rete protetto.

Analogo discorso per i server di posta. Per il protocollo SMTP è utile configurare il sistema perimetrale come MTA in uscita per i client interni ed, utilizzando record MX appositi per la posta, in ingresso verso il sistema perimetrale, che processerà, nel caso, la posta verso un server interno, rifiutando, invece, ogni forma di relay.

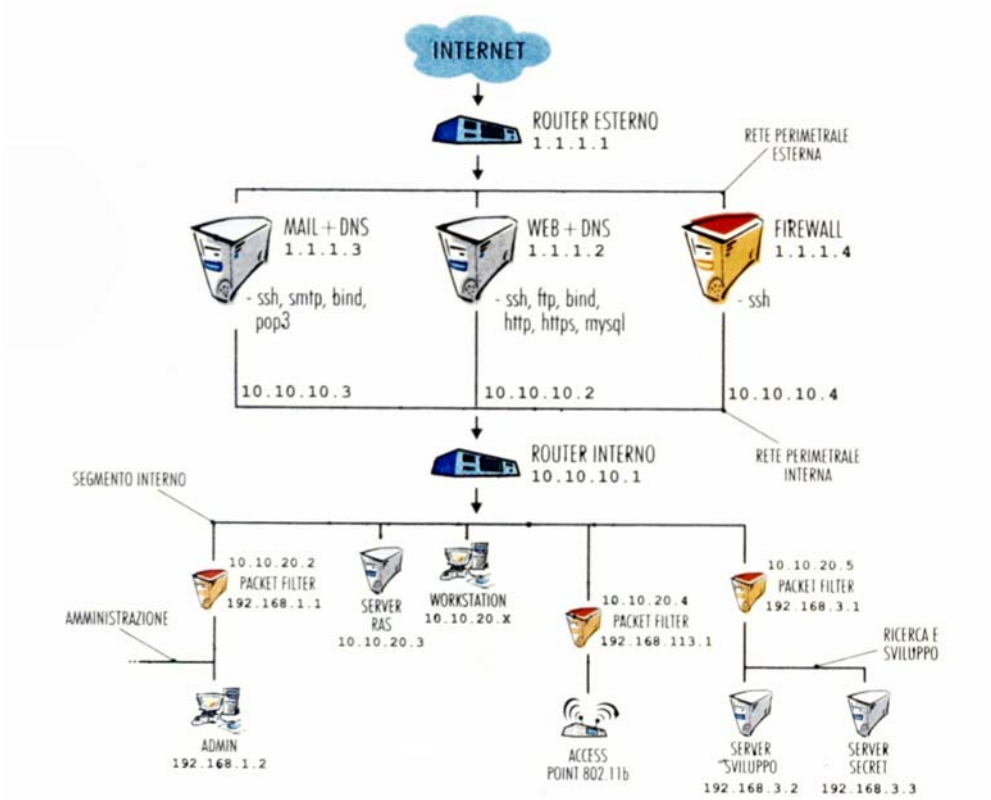


Figura 2: la rete della profumeria RoseNoir. Sono stati posati dei firewall basati su iptables a protezione delle reti relative ai vari uffici (amministrazione, ricerca e sviluppo) e dell'Access Point. Inoltre, come si può vedere, è stato inserito un secondo router tra la rete pubblica e quella privata.

Per il protocollo POP-3 (o suoi eventuali analoghi) potrebbe essere il caso impedire il trasferimento da server esterni in Intemet; potrebbe essere utile un sistema di proxy interno per permettere un più granulare controllo in fase di log; sarebbe indispensabile non utilizzare autenticazioni in chiaro. Un discorso a parte meritano il server RAS e l'Access Point (AP) per reti wireless 802.11.

Differente è il posizionamento del server RAS a seconda delle esigenze di utilizzo: per agenti in viaggio che abbiano bisogno di aggiornare database interni dalle sedi dei clienti, piuttosto che per dipendenti catalogabili come telelavoratori o lavoratori spesso fuori sede.

L'AP va considerato come surrogato wireless del segmento delle macchine dei dipendenti o come sistema di collegamento con operazioni di stoccaggio e magazzino? Il segnale dell'area ESSID è monitorabile dall'esterno della sede? Da personale non autorizzato in visita? In ogni caso potrebbe essere necessario

schermare il sistema di bridging o di NAT dal resto del segmento interno mediante layer superiori di autenticazione ed autorizzazione.

Questi esempi sono utili per capire come tutte queste considerazioni richiedano trattazioni separate rispetto a questa analisi; l'importante è capire come questi particolari siano parte integrante del disegno e gestione di un firewall e non, come spesso capita, operazioni seguenti all'installazione di un filtro di pacchetti.

La nuova architettura di rete prevede alcune differenze dallo scenario originale:
• separazione del traffico pubblico e privato mediante due reti perimetrali senza forwarding di traffico;
• un unico punto di NAT tra le due reti perimetrali per le macchine dei dipendenti;
• separazione mediante filtri di pacchetti tra il segmento interno e la seconda rete perimetrale;
• ACL per controllare il traffico dei dipendenti verso la rete perimetrale;
• separazione mediante firewall interni con router di controllo tra i segmenti di management, ricerca e sviluppo e dei dipendenti.

E' possibile utilizzare hardware generico con sistema operativo Linux per i firewall interni e per il generico NAT dei dipendenti in rete perimetrale.

Un'altra scelta in termini di costi, e gestione va fatta per il router intemo tra la seconda rete perimetrale ed il segmento interno: una buona scelta consiste in uno o più router Cisco, con i quali scegliere tra segmenti interni tisicamente separati (mediante interfacce multiple su singolo router o router interni differenti su segmenti separati) o un singolo segmento interno suddiviso in sezioni separate mediante filtri di pacchetti (schema di figura 2).

Non tratteremo gli usuali accorgimenti (sostituzione di ftp con scp, layer crittografico per lo scaricamento della posta, uso di token, OneTimePassword o altri strumenti di autenticazione); ovviamente sono tutti validi ed importanti, ma esulano dall'analisi corrente. Vediamo ora come sfruttare le robuste capacità del kernel Linux per operare all'interno di questa nuova architettura di rete.

Packet Filtering

Questo sistema permette l'instradamento di pacchetti tra sistemi intemi ed esterni, dopo un esame selettivo dell'intestazione dei datagrammi; versioni evolute del filtro di pacchetti permettono di mantenere una conoscenza delle informazioni sullo stato della connessione: questo sistema, noto come **stateful inspection** o **state tracking** permette di rendere dinamico il processo decisionale del filtro.

Ad esempio, è possibile accettare datagrammi UDP solo se contengono risposte a precedenti datagrammi UDP usciti dalla rete interna oppure segmenti TCP che facciano parte solo di connessioni in stato

ESTABLISHED. Un altro uso avanzato del filtro di pacchetti è il cosiddetto **protocol checking** che permette di operare dedsioni su datagrammi in base al contenuto dei payload, ad esempio, permettere datagrammi di risposta dal server SMTP solo se contengono messaggi validi e legali secondo lo standard SMTP.

Il kernel Linux, nella sua versione 2.4.x, contiene il supporto per una avanzata tecnologia di filtro, analisi e modifica dei pacchetti: **Netfilter**.

Questo sottosistema di rete permette di operare sia un filtro di tipo stateful, sia il NAT in ingresso e uscita con supporto per il protocol checking.
Il router esterno sarà, con molta probabilità, un dispositivo Cisco, mentre quello interno potrebbe essere una macchina generica con sistema Linux.

La scelta in ogni caso sarà dettata più che altro da questioni di costi e gestione. Quali saranno le regole dei dispositivi di filtering?

Molto probabilmente, data la tipologia della rete, le esigenze saranno differenti a seconda dei segmenti da filtrare e controllare: il traffico dovrebbe essere, dopo aver configurato il routing, differente per ogni segmento, così come saranno differenti i servizi offerti e richiesti. Probabilmente, le ACL del filtro a protezione della rete di amministrazione saranno più restrittive e, al tempo stesso, più semplici rispetto a quelle del router estemo.

Fatto spesso dimenticato, un sistema di proxy generico o un filtro di pacchetti non sono in grado, nella maggior parte
dei casi, di proteggere da problemi a livello applicativo: se una regola prevede il passaggio di traffico verso il web server, non sarà in grado di proteggere da richieste HTTP malformate o particolari, in grado di utilizzare un errore implementativo di Apache o di uno script CGI non testato a sufficienza. Il semplice posizionamento e la seguente gestione di un firewall non possono ne dovrebbero mai sostituire una corretta impalcatura applicativa.

Se il demone sshd non è stato aggiornato dopo l'uscita di un exploit pubblico, un firewall, per quanto ben strutturato, non potrà proteggere la nostra rete, se permette traffico da e per il server sshd.

Vediamo le possibili regole per il router esterno, considerando la possibilità di permettere traffico DNS, SMTP, SSH verso i due server pubblici, HTTP e HTTPS.

Disabiliteremo, rispetto al diagramma originale, il traffico FTP, utilizzando ssh per il trasferimento di file dall'esterno della rete.

Una migliore soluzione potrebbe essere quella di amministrare i bastion host dalla sola LAN amministrativa, impedendo ogni forma di traffico dall'esterno della rete.

Per semplicità, non sarà duplicata la regola in senso opposto: per esempio, il traffico HTTP richiede che in ingresso siano permessi datagrammi da porte effimere verso la porta 80 e in uscita dalla porta 80 del server verso qualunque client con porta effimera.
Le regole sono esposte mediante dicitura logica poiché la sintassi specifica varia a seconda del modello di router scelto.

Router Esterno

• BLOCCARE datagrammi in ingresso con indirizzi IP delle reti perimetrali e interne: data la normale configurazione della rete non c'è motivo per cui pacchetti con indirizzi interni arrivino all'interfaccia esterna del router;

• BLOCCARE datagrammi in uscita con indirizzi IP esterni per impedire l'invio di pacchetti con indirizzi spoofati dalla nostra rete;

• PERMETTERE datagrammi in ingresso per la porta 80/TCP e 443/TCP verso il solo web server;

• PERMETTERE datagrammi in ingresso per la porta 25/TCP verso il solo mail server;

• PERMETTERE datagrammi in uscita per la porta 25/TCP dal solo mai] server;

• PERMETTERE datagrammi UDP e TCP per la porta 53 al solo DNS server perimetrale;

• PERMETTERE datagrammi in ingresso per la porta 22/TCP ai soli server pubblici perimetrali;

• BLOCCARE ogni altro tipo di traffico.

Questa configurazione dovrebbe permettere solamente il traffico in ingresso e uscita per i soli server pubblici. Il ruolo del router esterno è quello di proteggere le reti perimetrali ed interna, filtrando tutto il traffico non voluto.

Vedremo successivamente come implementare il traffico dei dipendenti verso Intemet, qualora fosse permesso.
Il router interno, invece, deve essere in grado di filtrare il traffico verso i segmenti interni: al di fuori del traffico SMTP e DNS, non deve esserci alcun traffico in ingresso dal perimetro verso l'interno, ad eccezione del traffico di risposta per i client interni.

Router Interno

• BLOCCARE datagrammi in ingresso con indirizzi IP della rete interna;

• BLOCCARE datagrammi in uscita con indirizzi IP estemi;

• PERMETTERE datagrammi in uscita per la 80/TCP e 443/TCP verso il web server perimetrale;

• PERMETTERE datagrammi in uscita per la 25/TCP verso il mail server aziendale;

• PERMETTERE traffico UDP e TCP per la porta 53 dal DNS interno a quello perimetrale;

• PERMETTERE traffico in uscita per la 22/TCP verso i server perimetrali al solo IP 10.10.20.2 affinchè solo i client amministrativi possano raggiungere i demoni ssh;

• PERMETTERE traffico in uscita per la 110/TCP verso il solo server di posta alla sola classe 10.10.20.0/24;

• BLOCCARE ogni altro tipo di traffico.

Anche in questo caso, la configurazione minimale deve costringere il traffico alle sole comunicazioni permesse e preventivate. Come prima, tratteremo dopo il traffico dei dipendenti in uscita verso Internet. Internamente, lungo il backbone 10.10.20.0/24 abbiamo almeno tre macchine Linux configurate come filtri e gateway di tre ulteriori segmenti interni:

• la LAN amministrativa, la rete wireless e il segmento di ricerca e sviluppo, contenente dati sensibili e materiale segreto di sviluppo. Questi tre server hanno compiti simili. Le regole sono presentate con la sintassi di iptables(S), come mostrato nel riquadro 8.

• ethO è l'interfaccia estema, mentre ethl quella interna.

Gateway Amministrativo

Punto di accesso alla rete amministrativa, deve poter controllare ogni forma di accesso. Nessun flusso di dati deve essere permesso dall'esterno del segmento amministrativo, se non le legali risposte alle richieste interne. Ma deve permettere accesso verso l'esterno senza limitazioni verso ogni altro segmento della rete aziendale. Questo è l'unico nodo di accesso per il traffico normale, escludendo, quindi, eventuale traffico di management dei sensori o dei router.

Wireless Gateway

II discorso del gateway wireless è differente: quale sarà il ruolo dell'AP? Quale tipo di servizio si intende offrire ai dipendenti o agli agenti in sede? Immaginiamo che sia solo un ambiente di test, come prospettato nello scenario originale: è necessario cautelarsi dalla possibilità che un uso non autorizzato dell'AP possa provocare danni ad altre macchine sensibili.

R&D Gateway

II segmento di ricerca e sviluppo deve essere accessibile solo ad alcuni client e agli amministratori. Come permettere il traffico dei dipendenti, nel caso questo fosse contemplato? I dipendenti potrebbero volere accesso dai loro client Windows posti nel segmento interno, nella classe 10.10.20.0/24; la stessa esigenza potrebbero averla anche gli amministratori del segmento di management, anche se permettere traffico in uscita verso Internet da questo segmento è opinabile.

Non è, ovviamente, possibile instradare datagrammi con indirizzi privati oltre il router esterno: questa eventualità non porterebbe, comunque, ad alcuna risposta da parte di server esterni. Dal momento che l'azienda non ha acquistato indirizzi IP pubblici per ogni macchina della rete è necessario avvalersi del gateway 1.1.1.4.

Questo può operare mediante NAT, oppure contenere una serie di proxy per ogni servizio necessario.

In questa configurazione, l'azienda ha deciso di permettere ogni tipo di traffico TCP ai dipendenti ed ha optato per l'uso del NAT. Dovremo, quindi, aggiungere alcune regole ai router interno ed esterno.

Router Interno

• PERMETTERE traffico TCP dai client Windows a JP esterni;

• PERMETTERE traffico TCP da 10.10.20.2 a IP esterni;

• BLOCCARE traffico da 10.10.20.4 a 1.1.1.4 e IP esterni per impedire l'uso degli insicuri client wireless, senza una ulteriore forma di autorizzazione.

Router Esterno

• PERMETTERE traffico in uscita dall'IP 1.1.1.4.

Come implementare, però, il traffico in usata dalle LAN di management Wireless, dei dipendenti e da 1.1.1.4 verso Internet senza avere a disposizione IP instradabili?
Questo è un compito che il kernel Linux può svolgere mediante NAT.

Network Address Translation

Questa procedura permette di modificare gli indirizzi IP contenuti nei datagrammi in modo da mascherare, modificare, dirottare le connessioni di rete.

Gli usi per questo tipo di tecnologia sono molteplici: ad esempio, permettere la connessione ad Internet a molteplici client in presenza di un singolo indirizzo

IP pubblico; oppure utilizzare più server che rispondano ad un unico indirizzo IP, sia per limitazione logistica o per processi di load-sharing; o ancora, implementare proxy trasparenti.

Nella tecnologia NAT di netfilter esistono due tipi di NAT: Source NAT e Destination NAT. Nel primo caso, il sottosistema altera l'indirizzo IP sorgente, modificando l'origine del datagramma.

Questo avviene dopo le procedure decisionali di instradamento. Un caso particolare di Source NAT è rappresentato dal Masquerading del kernel Linux.

Nel secondo caso, invece, il sottosistema altera l'indirizzo di destinazione, modificando così il punto di arrivo del datagramma, prima di ogni processo di instradamento; esempi sono il port forwarding, load-sharing e transparent proxy.

Come già ricordato, il supporto NAT di Netfilter si avvale anche del protocol-checking. Alcuni protocolli, infatti, inseriscono indirizzi IP relativi al flusso dei dati all'intemo del payload dei pacchetti.

Come è intuibile, il modo di inserimento varia da protocollo a protocollo e affinchè il NAT sia efficace, è necessario che Netfilter conosca il funzionamento e la sintassi di questi scambi di dati, per poter modificare al volo il payload e, quindi, gli indirizzi utilizzati.

Insieme al mantenimento dello stato della connessione sarà possibile per client interni usufruire di protocolli complessi dal punto di vista della tecnologia NAT. Il kernel Linux contiene, come esempio nella distribuzione di Netfilter, il supporto per il protocollo FTP e per il pieno utilizzo del protocollo IRC DCC.

Le esigenze dei gw amministrativo e wireless sono analoghe: convertire indirizzi del segmento più interno in indirizzi del backbone. Questo è facilmente risolvibile mediante Source NAT. Non vi è particolare interesse, vista la semplice architettura, nel Destination NAT.

Gateway Amministrativo

```
-N logdrop

-A logdrop -s 10.10.20.1/24 -j LOG --log-level 4 --log-prefix
"INVALID INNER INBOUND PKT"

-A logdrop -s 10.10.10.1/24 -j LOG --log-level 4 --log-prefix
"INVALID PERIMETER INBOUND PKT"

-A logdrop -j LOG --log-level 4 --log-prefix "INVALID INBOUND
PKT "

-A logdrop -j DROP
```

```
-A INPUT -p tcp —dport 113 -j REJECT —reject-with tcp-reset
```

```
-A INPUT -i ethO -(n state —state NEW, INVALID -j logdrop

-A INPUT -i ethO -m state —state RELATED, ESTABLISHED -j
ACCEPT

Gateway Wireless

-P INPUT DROP

-A INPUT -i ethl -d $IP CLIENT WINDOWS -j ACCEPT

-A INPUT -i ethl -d 1.1.1.3 -p tcp —dport 25 -j ACCEPT

-A INPUT -i ethl -d 1.1.1.2 -p tcp -m multiport —dports
80,443 -j ACCEPT

-A INPUT -i ethO -m state —state RELATED, _ESTABLISHED -j
ACCEPT

-A OUTPUT -o ethO -d 10.10.20.2 -j DROP

-A OUTPUT -o ethO -d 10.10.20.3 -j DBOP

-A OUTPUT -o ethO -d 10.10.20.5 -j DBOP
```

```
Gateway R/D

-P INPUT DROP
-P OUTPUT DROP
-A INPUT –i ethO –s 10.10.20.2 –j ACCEPT
-A INPUT –I eth0 –s $IP_CLIENT_AUTORIZZATO –p tcp –j ACCEPT

-A OUTPUT -o ethO -m state —state RELATED, ESTABLISHED -j
ACCEPT

-A OUTPUT -o ethO -m state --state NEW, INVALID -j LOG --log-
level 4 --log-prefix "INVALID OUTBOUND PKT"
```

Le regole di alcuni gateway (iptables)

Management Gateway

```
    -t nat -A POSTROUTING -o ethO -j SNAT —to 10.10.20.2
```

Wireless Gateway

```
    -t nat -A POSTROUTING -o ethO -j SNAT —to 10.10.20.4
```

R&D Gateway

La LAN di ricerca e sviluppo, invece, non contempla nuovo traffico in uscita, ma solo la possibilità di ingresso per i client autorizzati dalle ACL del filtro di pacchetti. In questo caso, potremmo usare il DNAT utilizzando la mappatura 10.10.20.202 <> 192.168.3.2 e quella 10.10.20.203 <> 192.168.3.3:

```
-t nat -A PREROUTING -i ethO -d 10.10.20.202 -p tcp -j DNAT -
                        -to 192.168.3.2
```

```
-t nat -A PREROUTING -i ethO -d 10.10.20.203 -p tcp -j DNAT -
                        -to 192.168.3.2
```

FW/NAT 1.1.1.4

II compito della macchina 1.1.1.4 , in caso si opti per il NAT invece di un sistema di proxy, è convertire i datagrammi di client e amministratori affinchè possano essere instradati dal router esterno verso Internet.

Ovviamente, i client Windows dovranno usare 10.10.10.4 come gateway predefinito e le tabelle di instradamento dei vari dispositivi dovranno contemplare questo IP come gateway per il traffico proveniente dai client del segmento interno.

```
-P INPUT DROP
-A INPUT -i ethl -s 10.10.10.2 -j ACCEPT
-A INPUT -i ethl -S $IP_CLIENT_WINDOWS -p tcp -d ! 10.10.10.4
-j ACCEPT
-t nat -A POSTROUTING -o ethO -s 10.10.20.2 -j SNAT --to
l.JL.1.4
-t nat -A POSTROUTING -o ethO -s $IP_CLIENT_WINDOWS -p tcp -j
SNAT --to_ 1.1.1.4
```

IDS (Intrusion Detection System)

I moderni IDS sono strumenti che rilevano impronte digitali di attacchi ed eventi inusuali, noti come signature; questi sono schemi specifici che indicano solitamente attività malevole o sospette.

Esistono essenzialmente tré forme differenti di IDS: Network, Host e Stack IDS. Il primo controlla passivamente il traffico in un segmento di rete, utilizzando pacchetti di dati come fonte di informazioni. Il secondo tipo, invece, può analizzare o i log di sistema quasi in tempo reale, oppure analizzare il traffico di rete ricevuto dal sistema su cui è installato.
Il terzo modello di IDS, invece, si basa sul controllo a basso livello del traffico di rete, a livello dello stack TCP/IP, decidendo quali azioni permettere o portare avanti prima che il sistema operativo o le applicazioni abbiano accesso ai dati contenuti nei datagrammi controllati: questo tipo di IDS rappresenta, in

effetti, una via di mezzo con i filtri di pacchetto che mantengono informazioni di stato.

La soluzione OpenSource più famosa, sia per ubiquità di utilizzo che per effettiva capacità e resa di utilizzo, è rappresentata dal software Snort, liberamente prelevabile all'URL http://www.snort.org/.

Questo IDS è di tipo Network ed è, nelle parole degli autori, in grado di controllare il traffico di rete e loggare pacchetti dati in tempo reale in un ambiente basato su protocollo IR Può operare analisi del protocollo e indagini di pattern matching per individuare attacchi ed eventi di rete di varia natura. Snort utilizza un flessibile linguaggio di regole per descrivere il traffico.

A livello concettuale, la maggior parte dei Network IDS presenta elementi funzionali comuni:

• l'interfaccia di rete del sistema è posta in modalità promiscua, per catturare tutto il traffico lungo il segmento analizzato;

• possono essere utilizzati filtri specifici per modificare la tipologia e la quantità di traffico da sottoporre all'attenzione del motore di ricerca degli attacchi; questo ultimo è essenzialmente di tre tipi: a ricerca di modelli, frequenza o anomalie.

Il sistema, o dispositivo, sul quale viene fatto girare un IDS viene comunemente definito sensore. Questo è anche il termine che useremo in questa trattazione.

Snort può essere utilizzato come semplice sniffer, come legger di pacchetti di rete, oppure come vero e proprio IDS.

La maggiore complessità e, contemporaneamente, il maggiore punto debole di un IDS risiedono nel database di segnature disponibili. Se questo non è aggiornato, ogni attacco teorizzato posteriormente all'ultimo aggiornamento non sarà rilevato dal motore di ricerca.

Da questo punto di vista Snort è certamente un ottimo prodotto, allineato ai migliori prodotti commerciali e, in alcuni casi, ha mostrato migliori performance e minore incidenza di falsi negativi in risposta a specifiche tecniche di evasione dagli IDS. La rete utilizzata come scenario presenta un interessante caratteristica dal punto di vista degli IDS: la loro assenza. Dunque, il primo passo è valutare la necessità di sensori IDS all'interno della rete e, in caso affermativo, decidere la loro posizione.

Perché dotarsi di sensori IDS all'interno della propria rete? Spesso molti amministratori sono soliti aprire svariate sessioni di terminale contenenti molteplici istanze di strumenti di analisi del traffico di rete: tcpdump, arpwatch, sniff ed altri meno noti, ma altrettanto utili cugini, imperversano sugli schermi con il loro costante output.

E' vero che un lato distintivo di ogni amministratore si aforse la lettura in tempo reale dell'output di tcpdump, ma è altrettanto sicuro che di pari passo con il crescere di un segmento di rete, cresca anche la necessità di strumenti automatizzati ed indipendenti, per poi valutare correlazioni ed analisi di logi. E' oltretutto scarsa la percentuale di reti amministrate 24 ore al giorno. Come poter valutare, quindi, la presenza di attacchi o attività anomale nella rete?

Da questo punto di vista, nonostante la possibilità mai azzerabile di falsi negativi e positivi, i sistemi di Intrusion Detection sono insostituibili.

E la distribuzione ufficiale di Snort contiene tutto il necessario per un pronto utilizzo, compresi tool di supporto per la collezione e analisi dei log.

Torniamo allo scenario dell'azienda produttrice di profumi. Qual è la necessità imperante? Sicuramente essere a conoscenza di traffico non autorizzato proveniente dall'esterno nelle reti perimetrali e, ancora più importante, nella rete interna.

Una seconda necessità potrebbe essere quella di accorgersi di eventuali tentativi di attacco contro i nostri server pubblici, oppure direttamente contro i nostri nodi di difesa. Il lusso della ridondanza e della granularità suggerirebbero di controllare anche il traffico non autorizzato tra le reti amministrativa, di ricerca e dei dipendenti. E' ovvio, come questi requisiti non possano essere soddisfatti da un unico sensore.

Un sensore potrebbe essere posto nel tragitto tra il router esterno e la prima rete perimetrale per valutare il traffico diretto ai server pubblici; un secondo sensore lungo la rete perimetrale interna per osservare e analizzare il traffico diretto al router interno, sia dai server sui sistemi bastione verso il segmento interno sia traffico illegale per errate configurazioni di instradamento o ACL; un ulteriore sensore dovrebbe essere posto nel segmento interno, per valutare eventuali attacchi contro le macchine dei dipendenti o i filtri che proteggono la rete di amministrazione, quella di ricerca e sviluppo e, non ultimo, l'Access Point 802.11.

Ulteriori sensori potrebbero essere posti nella rete di amministrazione: non dovrebbe, infatti, essere presente alcun tipo di traffico che non sia correlabile alle attività dei legittimi amministratori.

L'uso di questi sensori, sia ben chiaro, non sostituisce, in alcun modo, adeguate procedure di logging di ogni applicazione o dispositivo di rete presente nei vari segmenti.

Sarà oltre modo importante, poi, sincronizzare gli orologi di ogni dispositivo e sistema, per poter operare una adeguata correlazione temporale di ogni log della rete: dove gli indirizzi sono modificati, come accade per esempio in caso di NAT, la correlazione cronologica è aiuto insostituibile all'intuito e capacità del sistemista.

Non è nostra intenzione scendere nel dettaglio delle metodiche di interfacciamento dei sensori al resto della rete, ma alcune considerazioni sono necessarie: una macchina o dispositivo che faccia da IDS deve poter soddisfare almeno due requisiti. Primo, deve poter essere in grado di visualizzare tutto il traffico del segmento esaminato. Secondo, dovrebbe essere irraggiungibile, per quanto possibile, ad eventuali tentativi dell'attaccante di manipolarlo.

Esistono fin troppe tecniche di manipolazione del traffico applicativo per sfuggire ai sensori per poter ritenere che un IDS non sarebbe un bersaglio.

Come poter quindi monitorare segmenti switchati (non sarà presa in esame la possibilità che siano utilizzati semplici hub di rete in quanto, sebbene pratica possibile ed utilizzata, presenta fin troppi problemi di sicurezza ed efficienza per valutarli tutti in questa sede)? Il sensore sarà collegato o alle Spanning Port degli switch, oppure a dispositivi noti come Ethernet TAF. La porta SPAN, acronimo di Switch Port ANalyzer, è una porta in grado di replicare il traffico TX, RX, o TX/RX di un'altra porta dello switch.

Ad esempio, il primo sensore, tra router estemo e prima rete perimetrale, potrebbe essere posto su di una porta SPAN su cui duplicare quella cui è collegato il router.

Per ovviare alla raggiungibilità dell'IDS da parte di un attaccante, potrebbe essere possibile dotare il sensore di due interfacce di rete: quella di monitoraggio non sarà dotata di alcun protocollo di rete in binding; la seconda sarà utilizzata dagli amministratori direttamente dal segmento di management.

Altra possibilità è quella di amministrare e configurare il sensore da console.

In alcuni casi, però, l'uso della porta di Spanning presenta uno svantaggio non indifferente: è in grado di monitorare una singola porta dello switch e non tutto il traffico in transito per lo switch.

Non potrebbe essere usata efficacemente per l'analisi delle reti perimetrali o di qualunque altro segmento in toto.

Certo è possibile, se lo switch lo permette, duplicare più porte o tutto il segmento sulla porta di Spanning. Ma questo potrebbe saturare facilmente, in reti ad elevato carico, la possibilità di traffico della porta stessa.

Oltretutto, la necessità di inserire più sensori per un singolo segmento sarebbe problematico in termini di disegno dell'infrastruttura di rete.

Per ovviare a questi problemi è possibile usare i TAF, dispositivi costosi ma di sicura efficacia per alcune caratteristiche: possono, a seconda dei modelli, monitorare agevolmente più porte senza impatti sull'efficienza; non alterano in alcun modo la configurazione della rete nel caso di sostituzione di dispositivi o sensori; non permettono agli attaccanti di raggiungere direttamente i sensori.

Questi dispositivi possono essere collegati a switch esterni rispetto ai segmenti da monitorare. A questi switch saranno collegati i sensori IDS.

La figura 3 mostra la disposizione dei sensori senza la descrizione delle interfacce degli stessi, per motivi di semplicità. Ora sappiamo che desideriamo sensori IDS e abbiamo una approssimativa idea di dove posizionarli.

Un concetto spesso usato parlando di reti e sicurezza è quello della ridondanza. In questo caso è inutile che tutti i sensori prestino attenzione allo stesso tipo di attività e possibili problemi.

Su hardware generico potrebbe essere complesso gestire l'analisi del traffico in un segmento a lOOMbit/s saturato.

Invece, potremmo decidere di suddividere il compito tra i differenti sensori oppure aggiungere più sensori con differenti compiti in ognuna delle locazioni scelte. Quale che sia la nostra scelta, Snort è estremamente flessibile: una rapida occhiata al file di configurazione snort.conf e all'elenco di regole presenti nella distribuzione di default nella directory rules/ permettono di suddividere i compiti agevolmente.

Ogni set di regole può inoltre essere accomodato a seconda delle esigenze: se siamo sicuri di non avere server web installati nel segmento di ricerca e sviluppo è forse inutile la ricerca di messaggi HTTP di ritorno contenenti la stringa "*Bad communication of filename*" (ma, d'altra parte, questi messaggi potrebbero portare alla scoperta i server installati senza autorizzazione); il controllo dei messaggi di notifica di backdoor come Netbus o BackOrifice è sicuramente importante nel segmento di rete dei dipendenti, vista la presenza di macchine Windows, ma è forse superfluo in quella di amministrazione, dove sono presenti solo macchine Linux; se non siamo interessati ai semplici tentativi di attacco, ma solo a quelli mirati, la ricerca di exploit contro IIS è inutile, se il nostro web server è Apache su piattaforma Linux; d'altra parte l'analisi dell'invio di shellcode, impronta palese di tentativi di exploit mediante buffer overflow, ad esempio, o il controllo sugli agenti di DDoS noti è importante in tutta la rete.

Questa è una decisione all'apparenza semplice, ma dipende da varie considerazioni: economiche, di efficienza, di eventuale privacy dei dipendenti, di policy aziendale e, non meno importanti, personali dell'amministratore della rete. Spesso differenti segmenti fanno capo a differenti amministrazioni, con differenti metodiche e approcci alla sicurezza.

Alcune semplici linee guida su cosa non tralasciare, sono le seguenti:

• regole icmp-info, icmp, experimental nei segmenti di accesso ai router;
• regole attack-responses, DNS, exploit, MySQL, POP3, smtp, snmp, TFTP, virus nelle reti perimetrali;
• regole bad-traffic, DoS, DDoS, mise, info, scan, shellcode in ogni segmento;

• regole backdoor, exploit, netbios, FTP, virus, xll nel segmento interno;
• regole chat, multimedia, p2p, porn se contemplati dalla policy per i dipendenti;
• regole web-* per il segmento che contiene web server.

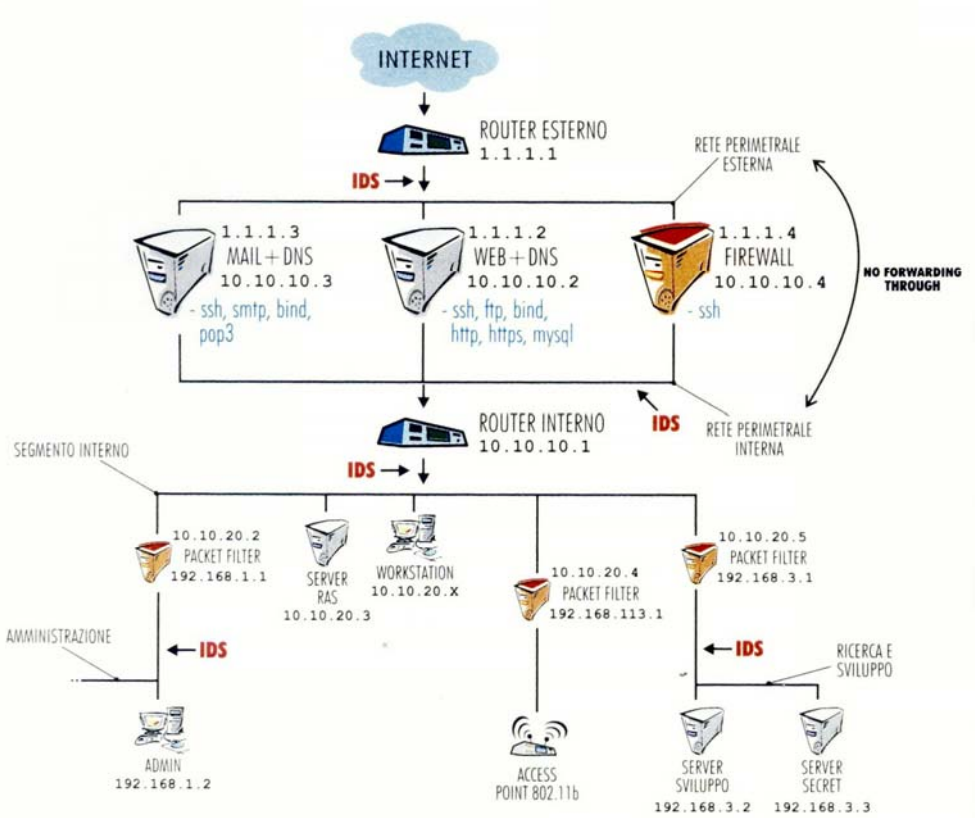


Figura 3: lo schema della rete della profumeria RoseNoir dopo la posa dei sensori IDS.

I set di regole di Snort contengono comunque istruzioni per ogni piattaforma all'interno di ognuna di queste categorie. E' quindi necessario controllare i set prima di decidere quali regole utilizzare nei file di configurazione dei sensori.

La distribuzione contiene anche il manuale utente che illustra la metodica di creazione di regole personali. Un semplice quanto banale metodo per creare un sottoinsieme di regole specifiche per sistemi Linux è il seguente:

```
adinin@sensor:~/snort-1.9.0/rules$ for f in 'ls|grep -v
Makefile'; do grep -i linux $f >> linux.rules ; done
```

Come per ogni IDS, è necessario aggiornare periodicamente il database delle regole prelevando gli aggiornamenti dall'indirizzo http://www.snort.org/ per essere sicuri che i sensori rilevino attività, tentativi e attacchi teorizzati o testati dopo l'ultimo aggiornamento.

Il passo successivo

Tutti questi passi sono incompleti, però, nel momento in cui ci si dimenticasse di politiche di hardening di ogni host e server: abbiamo tralasciato diversi punti indispensabili per la messa a punto di una rete sicura. Ad esempio, esistono molti strumenti di attacco per ambienti di rete switched che dipendenti potrebbero usare per cercare di aggirare le ACL imposte dall'amministrazione.

Questo non è compito del firewall in sé, quanto della corretta configurazione degli ambienti di lavoro condivisi, delle workstation personali e delle policy aziendali. La mancata menzione di un sistema di antivirus sui server di posta o sui file server non è una dimenticanza, quanto una scelta precisa per questa breve analisi. Oltretutto, la stesura corretta di un firewall dovrebbe essere studiata ad hoc e non generalizzata in esempi; pensiamo comunque che questo esempio possa fornire una idea di base del processo di gestione di una rete più complessa delle normali LAN casalinghe.

I log e l'Analisi Forense

La corretta collezione e gestione dei vari log è un processo indispensabile sia per mantenere un occhio vigile e conscio sulle attività della propria rete, sia per il critico e delicato compito di analisi forense.

L'eventualità che la rete non regga il colpo non va mai scartata o dimenticata, quanto piuttosto messa in cantiere come possibilità reale e, per asintoto, prima o poi certezza inevitabile, sia essa per attacchi provenienti dall'esterno o dall'interno del nostro ambiente di lavoro. Vediamo dunque come procedere.

Tutti i vari device presenti sulla rete (siano essi firewall, IDS, router o lo stesso sistema operativo) producono, in risposta a determinati eventi, una serie di informazioni che prendono genericamente il nome di log.

La gestione e l'organizzazione di tali informazioni all'interno della rete rivestirà, come vedremo, una notevole importanza, sicuramente maggiore di quello che si potrebbe erroneamente pensare.

Tradizionalmente i log sono memorizzati sui vari dischi fissi nel caso di normali computer, in specifici ambienti di raccolta nel caso di dispositivi proprietari, utilizzando svariate tecnologie, tra le quali le più diffuse syslog e trap SNMP.

Questo tipo di organizzazione delle informazioni ha però un notevole svantaggio, la frammentazione, in quanto i log sono scritti in "contenitori" diversi a seconda del device che li produce; il processo di consultazione, quindi, diventa lento e estremamente difficoltoso, dovendo attingere e correlare dati disposti in luoghi diversi ed disomogenei.

In realtà, esiste un sistema sensibilmente più efficiente per la gestione delle informazioni, che prevede una raccolta centralizzata e una organizzazione

"intelligente" delle stesse. Entriamo in questo modo nella definizione di Analisi Forense, un processo grazie al quale diviene possibile estrarre dati significativi da enormi database contenenti informazioni apparentemente eterogenee.

Un processo, si è detto. Il termine utilizzato lascia presupporre - come realmente è (o dovrebbe essere) - un comportamento "attivo" della rete; non ci si limita, quindi, ad analizzare dati di eventi accaduti, per i quali sono state lasciate tracce (per esempio analisi post-mortem di un attacco, dopo il defacement del sistema), ma si ricercano continuamente anomalie sul traffico della rete da difendere, con lo scopo di intervenire sul problema, in modo da evitarlo o, quando questo non e passibile, almeno arginarlo.

Gli elementi che caratterizzano un sistema di analisi forense sono diversi: esso si basa prima di tutto sulla raccolta centralizzata dei dati, che permette di gestire le informazioni da un unico ambiente di lavoro in modo estremamente più comodo; la raccolta dei dati da diverse tecnologie permette di sfruttare le informazioni di firewall, IDS, sistemi operativi, router ed apparati di rete in genere, in modo tale da avere una visione orizzontale di ciò che accade, non vincolata agli eventi di un unico tipo di device, offrendo la passibilità di fare una ricerca delle informazioni per severità di allarme o per tipo di evento.

Vi è un ulteriore problema inerente alla centralizzazione dei log: ogni singola tecnologia ha una sua scala di allarmi e severità differenti; per poter essere correttamente inseriti nel database, tali alert dovranno essere "normalizzati", in modo da adattarli ad una scala comune. Questa pratica ne faciliterà di gran lunga la ricerca all'interno del database, offrendo informazioni viziate da un "rumore" minore.

La correlazione dei dati è un altro elemento fondamentale dell'analisi forense: mettere in corrispondenza informazioni raccolte da diversi device, in tempi diversi, offrendo una vista logica omogenea sugli stessi permette una interpretazione dei log estremamente raffinata, focalizzando l'attenzione su eventi che, diversamente, sarebbero passati inosservati. A dimostrazione del1'importanza di questa fase prendiamo un esempio banale, ma estremamente rappresentativo: un attacker effettua uno scan di porte di un server sotto la nostra amministrazione spendendo un pacchetto ogni settimana; per un operatore che dovrà analizzare i log manualmente, sarebbe estremamente complicato, per non dire pressoché impossibile, riuscire ad immaginare che quel singolo pacchetto possa far parte di una sequenza volta al reperimento di informazione per un tentativo di intrusione.

Quando i dati sono tra loro correlati, invece, è possibile impostare una regola tale per cui, ad esempio, si può ricevere una notifica quando, se dallo stesso IP sorgente vedi un tentativo di connect a porte "strane" sempre diverse tra loro, distribuito su un lasso di tempo di un mese.

La fase di data mining è alla base di una buona infrastruttura di Information Management quale quella da noi prospettata: con questo processo si intende l'estrazione di informazioni aggregate da banche dati di grandi dimensioni

tramite applicazioni che individuano le associazioni nascoste tra le informazioni e le rendono "esplicite".

La qualità, la leggibilità e la possibilità di manipolazione, attraverso dei drill down, dei dati di un report è estremamente importante. II drill down ci consente, partendo dai dati aggregati, di entrare nel dettaglio passo per passo, mostrando ogni volta i dati che sono stati riassunti nella voce evidenziata.

Un esempio pratico permetterà di comprendere cosa permetta di fare il "drill down": l'operatore genera un report che evidenzia gli allarmi più frequenti all'interno del database nelle ultime due settimane; chiedendo un dettaglio maggiore su tali allarmi, vengono mostrati gli IP sorgenti che hanno prodotto il maggior numero di questi allarmi. Grazie a questa nuova informazione, è possibile cercare altre correlazioni all'interno del database, questa volta ricercando alcuni IP specifici. Come si capisce, un analisi di questo tipo, specialmente quando questa sia automatizzata da specifici tool, diventa estremamente efficace per la sorveglianza di una rete.

Abbiamo precedentemente evidenziato come l'analisi forense sia un processo attivo, che reagisce ad eventuali sollecitazioni dall'esterno. Quando un comportamento viene riconosciuto come "ostile" (può essere un solo evento, oppure un insieme correlato di eventi), viene attivato un sistema di notifica che, a seconda delle impostazioni dello stesso, emette un trap snmp, spedisce un SMS o una email che, di solito, attivano contromisure automatiche o manuali. Ovviamente, la possibilità di avere una real time console aiuta sicuramente a ridurre il gap che intercorre tra il momento in cui avviene la compromissione del server e il momento in cui l'infrastruttura torna completamente operativa.

Abbiamo così iniziato a delineare un profilo per un modello di infrastruttura che viene definita, secondo la terminologia corrente, Security Information Management (SIM).

Si cercherà, adesso, di presentare un potenziale modello di SIM applicabile alla topologia di rete della profumeria RoseNoir. Per una implementazione che veda il solo utilizzo di prodotti OpenSource, si prenderà in considerazione solo alcuni firewall e IDS, per i quali sono gia stati scritti appositi tool per la centralizzazione e 1'analisi dei log. Ci riferiamo in particolar modo ad **ACID**, acronimo di **Analysis Console for Intrusion Database** e a **logsnorter**, un tool in grado di inserire nel database MySQL di Snort i log generati da Netfilter.

Per semplificare il discorso, senza perdere però di generalità, consideriamo, tra tutti i dispositivi posati sulla rete, solo i seguenti device:

- IDS esterno;
- 1.1.1.4 firewall/NAT;
- IDS interno;
- 10.10.20.5 packet filter a difesa di secret/devel.

I log di questi device verranno raccolti e inseriti in un database centralizzato: questo dovrà avvenire utilizzando dei canali "sicuri" o creando dei tunnel "cifrati" laddove il traffico attraversi router e eventuali altri dispositivi che potrebbero essere soggetti ad attacco (e, quindi, a compromissione).

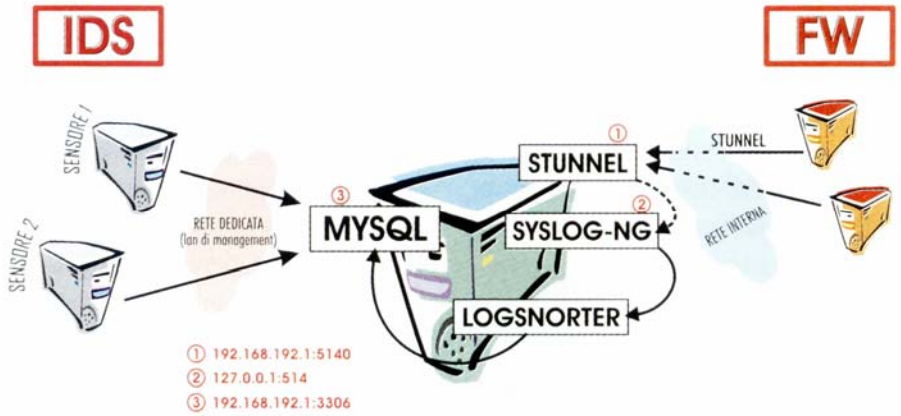


Figura 4: lo schema di raccolta dei log: gli IDS scrivono direttamente sui MySQL passando tramite la rete di management, mentre i firewall passano tramite un tunnel SSL sulla porta 5140 creato con tunnel: in ascolto nel pc di raccolta c'è Syslog-NG. Tramite logsnorter i log di syslog vengono memorizzati nell'unico ambiente di raccolta, il database snort di MySQL.

Per quanto riguarda gli IDS, Snort è in grado di loggare direttamente su database MySQL, purché lo si compili con le opzioni specifiche: si collegherà, quindi, direttamente al database centrale e invierà direttamente le informazioni raccolte. La spedizione dei log avverrà su un canale dedicato: in questi casi, infatti, conviene aggiungere agli IDS una nuova scheda di rete che si affaccerà ad un segmento di rete riservato al logging e alla gestione dell'IDS stesso.

In caso di compromissione del sensore, vedremo che i diritti concessi sul database centralizzato non saranno sufficienti, in ogni caso, per cancellare dati, ma solo per inserirne di nuovi.

I sistemi operativi dei firewall utilizzano la classica tecnologia syslog per registrare informazioni: i log saranno ridiretti verso un syslogd centralizzato per la raccolta e l'inserimento nello stesso database MySQL (oltre, eventualmente, ad essere scritti sul filesystem locale, anche se un sensore potrebbe, addirittura, essere sprovvisto di disco fisso).

syslogd ha un suo protocollo per il trasferimento dei log basato su una connessione UDP non cifrata, assolutamente insicura in questi contesti: ecco che si rende necessario, quindi, la creazione di un tunnel cifrato verso la destinazione, all'interno del quale far viaggiare i dati. Verrà, quindi, preferito a syslogd classica la sua variante syslogd-ng, in grado di effettuare connessioni TCV, che potranno viaggiare all'interno del tunnel cifrato. Si utilizzerà, quindi, stunnel per la creazione del tunnel SSL, Syslog-NG per l'invio dei log nel

tunnel ed infine il passaggio attraverso logsnorter per la memorizzazione all'interno del database, prelevando le informazioni da syslog.

Qualcuno potrà chiedersi come mai agli IDS sia stata aggiunta una scheda di rete affacciata su una rete trusted mentre per i firewall è stato preferito effettuare il tunneling cifrato: la domanda sarebbe effettivamente lecita... Aggiungere una nuova Ethernet ai firewall significherebbe complicare ulteriormente la configurazione degli stessi, in quanto andrebbero analizzati l'eventuale traffico tra tale interfaccia e quelle preesistenti, oltre ad un generale incremento del TOC (Total Cost of Ownership) del dispositivo stesso: ecco perché si preferisce, parlando di firewall, di gestire le Ethernet presenti e utilizzare un tunnel cifrato.

Uno schema del percorso seguito dai log verso il device che la console di management è mostrato in figura 5, mentre in figura 6 possiamo vedere una rappresentazione della rete e dei sensori.

Senza perderci troppo nella teoria, vediamo la configurazione relativa ai tool che abbiamo citato in precedenza.

SNORT

Come detto, per permettere a snort di scrivere direttamente nel database remoto i log, andrà ricompilato con il supporto per MySQL. Successivamente, sarà necessario inserire nel file snort.conf la seguente direttiva:

```
output database: log, mysql, user=snort
      password=snort dbname=snort
          host=192.168.192.1
```

dove log rappresenta il tipo facility (log o alert); user, password, sono username, password del database cui ci si collega; dbname, host, il nume del database e l'host che ospita il server db.

Inoltre, sarà necessario preparare il MySQL server presente su 192.168.192.1 per ricevere tali dati, creando le opportune tabelle che dovranno contenere i log.

Per fare questo, sarà necessario utilizzare il file presente nella distribuzione di snort create_mysql nel seguente modo:

```
# mysql -u root -p mysql
Enter password: [password di root di MySQL]

mysql> CREATE DATABASE SNORT
mysql> GRANT INSERT, SELECT ON SNORT.* TO SNORT@localhost;
(per logsnorter)
mysql> GRANT INSERT, SELECT ON SNORT.* TO
SNORT@192.168.192.2; (interfaccia mgmnt IDS esterno)
```

```
mysql> GRANT INSERT, SELECT ON SNORT.* TO
SNORT@192.168.192.3; (interfaccia mgmnt IDS interno)
mysql> FLUSH PRIVILEGES;
mysql> QUIT

# mysql –u root –p snort < CREATE _MYSQL
# Enter password: [password di root di MySQL]
```

A questo punto, la parte di configurazione relativa al logging di Snort è completata; passiamo quindi alla configurazione dei Syslog-NG, rispettivamente quelli remoti e quello centrale.

SYSLOG-NG

Per quanto riguarda i packet filter abbiamo visto sopra che nelle regole è presente una chain per il logging che, nello specifico, abbiamo chiamato logdrop.

Avremo, quindi, nel file di log di syslog diverse informazioni provenienti dal firewall. Anche in questo, cambieremo rispetto al syslog-ng.conf.sample di default aggiungendo le direttive di configurazione come segue:

```
source src { unix-stream("/dev/log");
  internal();
  pipe("/proc/kmsg");
};

filter f_firewall{match("IN=") and match("OUT=");};
destination firewall {tcp("127.0.0.1" port(5140));};
log{source(src); filter(f_firewall);
destination(firewall);};
```

Con la prima direttiva impostiamo un filtro tale per cui vengono individuati tutti i messaggi contenenti le stringhe "IN" ed "OUT", tipiche di netfilter; la seconda direttiva imposta una politica di destination log, tale per cui il risultato della direttiva di filtro "firewall" (quella appena definita) sia mandato all'host in questione (127.0.0.1:5140); infine, l'ultima direttiva fa in modo di rendere effettive le politiche sopra citate, ovvero applica filter e dastination insieme.

Le informazioni vengono redirezionate su 127.0.0.1:5140 perché in ascolto su quella porta ci sarà una istanza di stunnel che incapsulerà il traffico per mandarlo al Syslog-NG centralizzato. stunnel sarà lanciato sulle macchine che mandano i log con:

```
# stunnel –c –d 5140 –r 192.168.192.1:5140
```

Vediamo, quindi, il significato dei flag del comando sopra riportato:

- **-c** significa clientmode;

- **-a** significa IP: porta locale su cui mettersi in ascolto (di default IP e 127.0.0.1);
- **-r** significaIP: porta remota cui mandare i messaggi.

Sull'host remoto invece stunnel dovrà essere lanciato nel seguente modo:

```
# stunnel -p stunnel.pem -d 5140 -r 192.168.192.1:514
```

II significato dei flag, del resto abbastanza intuitivo, è il seguente

- **-p** è il flag per specificare il file .pem contenente il certificato per il secure layer, appunto in formato pem;
- **-d** significa IP:Porta locale su cui mettersi in ascolto (di default IP è 127.0.0.1);
- **-r** significa IP: porta remota cui mandare i messaggi (in ascolto c'è il Syslog-NG locale della macchina ricevente, come vedremo tra un attimo).

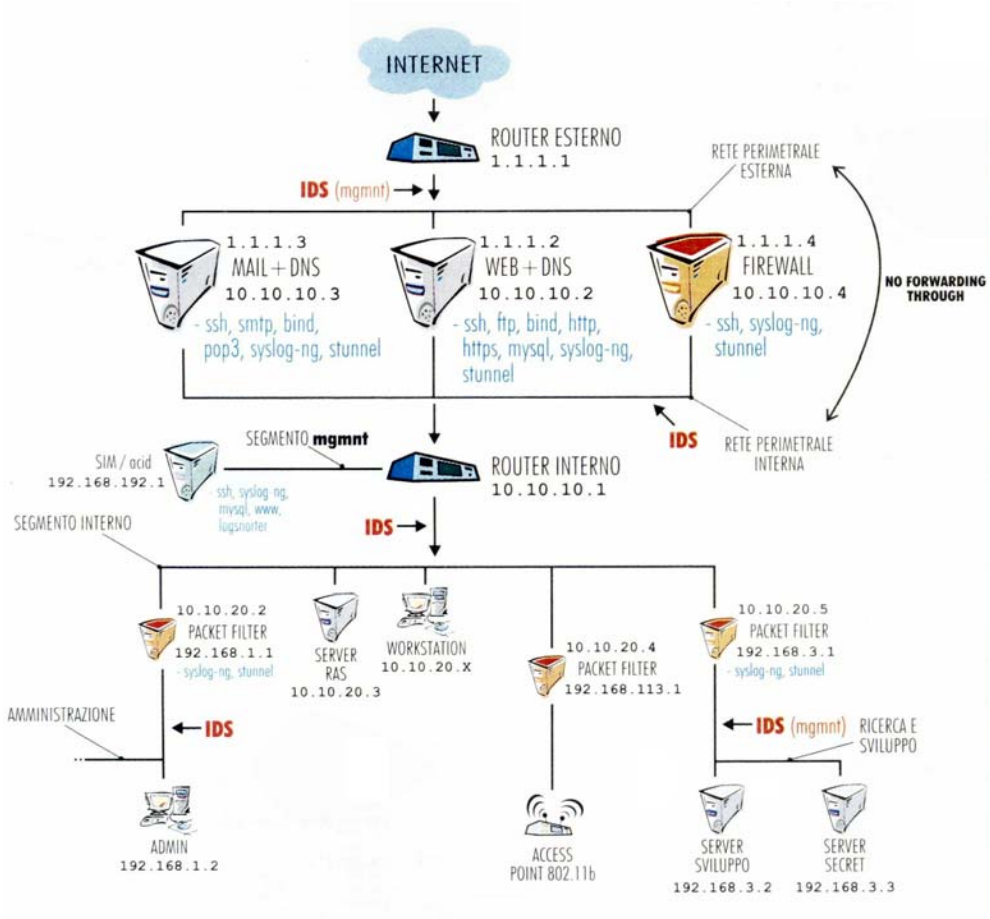


Figura 5: lo schema della rete della profumeria RoseNoir: in evidenza la rete di management.

A questo punto l'istanza stunnel descritta poco sopra riceverà le informazioni dai Syslog-NG remoti redirezionandole al Syslog-NG locale, la cui

configurazione rispetto al Syslog-NG.sample distribuito con i sorgenti cambia per la direttiva source src che diventa:

```
source src {
      unix-stream("/dev/log");
      pipe("/proc/kmsg");
      tcp(ip("127.0.0.1") port(514));
      tcp();
      internal();
};

filter f_firewall{match("IN=") and match("OUT=");};
destination firewall {file("/var/log/firewall.log"
OWNER("root") GROUP("adm") PERM(0640));};
log{source(src);
filter(f_firewall);
destination(firewall);};
```

In sostanza, abbiamo aggiunto l'ascolto sulla porta TCP ed un filtro per gestire tutte le informazioni dei firewall su un unico file.

LOGSNORTER

Manca ancora la configurazione di logsnorter, piuttosto semplice, basta una sola riga di comando:

```
# logsnorter –u snort –p snort –s 127.0.0.1 –d snort
          –T/var/log/firewall.log –L 8 -t
```

Vediamo il significato dei flag:

- **-u**, **-p**, **-s**, **-d** rappresentano il DB al quale connettersi e le relative credenziali;
- **-T/var/log/firewall.log** significa che i dati vengono importati in modalità "tailing" (tail -f) dal file Firewall .log;
- **-L 8** specifica la natura dei log, in questo caso prodotti da iptables (logsnorter può gestire i messaggi anche di altre tecnologie);
- **-t** significa che utilizza i timestamp dei messaggi di syslog piuttosto che il corrent time.

Finalmente, abbiamo tutte le informazioni necessarie all'interno di un unico database pronte per essere utilizzate tramite opportune applicazioni; sempre rimanendo in ambito OpenSource, il front-end verso questi dati che andremo ad utilizzare è ACID.

ACID

ACID è un motore di analisi, basato su PHP, studiato per analizzare eventi security-related provenienti da diverse tecnologie. E' scaricabile dal sito http://www.certorg/kb/acid/

L' installazione del programma necessita dell'installazione di alcuni pacchetti, quali ADODB, PHPlot, JPGraph, GD... nella distribuzione sono riportate tutte le informazioni e i requisiti. Una volta installato, ACID è di semplice configurazione: la prima cosa da fare è modificare le ACL sul database per l'utente snort (che gira, questa volta, da localhost):

```
# mysql -u root -p
# Enter password: [password di root di MySQL]

mysql> GRANT SELECT INSERT, UPDATE, UPDATE  ON SNORT.* TO
SNORT@localhost IDENTIFIED BY 'SNORT';
mysql> QUIT
```

In seguito passiamo alla modifica del file di configurazione acid_conf.php, eccone un esempio:

```
<?php
$ACID_VERSION = "0.9.6b22";
$DBtype = "MySQL";
$alert_dbname = "SNORT";
$alert_host = "localhost";
$alert_port = "";
$alert_user = "SNORT";
$password = "SNORT";
[snip]
```

Adesso puntando il browser su http://192.168.192.1/acid/acid_main.php riceverete un messaggio dove si dice che la tabella acid_ag (alert group) non è presente. Seguendo le istruzioni sulla pagina e seguendo, quindi, il link alla setup page sarà possibile installare via web le tabelle mancanti. A questo punto, saremo pronti a sfogliare il database delle informazioni in modo efficiente.

ACID oltre a presentarci statistiche specifiche per IP sorgente, IP destinazione, protocollo, porta sorgente, porta destinazione e allarme è anche in grado di produrre report grafici, oltre che ad avere un complesso sistema di query sul database che permette un data mining del tipo a drill down estremamente funzionale ed efficace.

L'applicazione prevede quattro tipologie di criteri di ricerca:

- Meta: sencor, alert group, signature, classification, detection time
- IP: qualsiasi header IP
- Layer-4: qualsiasi header TCP /UDP/ ICMP
- Payload: qualsiasi payload di layer-4

Possono essere fatte query composte tra i vari criteri di ricerca, inoltre in ogni criterio esistono delle chiavi di ricerca, correlate da loro operatori logici. E' anche possibile generare grafici degli allarmi partendo da query.

Verrà proposto adesso un modello di schema del database e di un "agente" software (una applicazione preposta all'analisi dei log, mediante i quali sarà possibile implementare alcune delle caratteristiche che mancano ad ACID per essere una infrastruttura di SIM a tutti gli effetti. Supponiamo di voler aggiungere un device per cui ACID non è stato pensato: si vuole analizzare e centralizzare, ad esempio, il syslog di una applicazione quale potrebbe essere il webserver Apache. Vediamo un possibile esempio di log prodotto:

```
192.168.1.55 - - [15/jan/2003:19:09:18 + 0100] "GET
/index.html http/1.0" 200 14025 "http://1.1.1.2/index.html"\
"lynx /2.8.5dev.7 libwww-FM/2.1.4 ssl –
mm/1.4.1openSSL/0.9.6b"
```

Si valutano come significative alcune informazioni estratte dalla riga di log, che quindi si vogliono inserire nel data-base. In particolare si vuole tener traccia dell'ip sorgente della richiesta http, del timestamp e dell'URL richiesto.

Come si può vedere, allo stato attuale queste informazioni non sono inseribili nel database per la struttura delle tabelle; sarà quindi necessario apportare alcune modifiche allo schema.

Il core e rappresentato dalla tabella "event", che è corrispondenza con la tabella delle signature, un insieme di tabelle che rappresentano i dati reali dell'evento (iphdr, icmphdr, udphdr, tcphdr, data) e la tabella relativa ai sensori. In questo modo un evento completo viene costruito relazionando tra loro diverse tabelle del database. Per memorizzare le informazioni relative al webserver Apache scnza stravolgere l'attuale applicazione, ogni elemento sopra indicato verrà aggiunto nel modo seguente:

ip src	→ tabella iphdr
timestamp	→ tabella event
URL richiesto	→ tabella event

Per comodità, sarebbe opportuno utilizzare la stessa scala di allarmi che usa Snort, normalizzando cosi le informazioni di Apache rispetto a quelle degli altri device: sara dunque necessario, prima di inserire il log nel database, fare una valutazione sulle informazioni da inserire.

Questo compito è assegnato all'agente software, che analizza i file di log di Apache e decide se e dove devono essere inseriti nel database.

Le difficoltà, a questo punto, potrebbero essere diverse a seconda dell'approccio seguito per la scrittura dell'agente: questo, ad esempio, potrebbe essere un piccolo demone in Perl che esamina in tailing i log di Apache o, ancora meglio, riceve i log direttamente da Apache attraverso un pipe (ad esempio, utilizzando la direttiva di configurazione di Apache: TransferLog

"|agente"). Per poter mantenere una omogeneità tra gli allarmi dei vari device, inotre, potrebbe essere conveniente utilizzare la stessa scala di Snort, utilizzando quindi le stesse regole presenti in quelle che seguono web-attacks.rules, web-cgi.rules, web-client.rules, web-coldfusion.rules, web-frontpage.rules, web-iis.rules, web-misc.rules, web-php.rules.

Prendiamo, ad esempio, la prima regola della prima serie web-attacks.rules:

```
alert tcp $EXTERNAL_NET eni -> $HTTP_SERVERS $HTTP_PORTS\
(msg: "WEB-ATTACKS PS COMMAND ATTEMPT";
flow: TO _SERVER, ESTABLISHED;
uricontent: "/bin/ps";\
nocase;
sid: 1328;
classtype: web-application-attack;
rev: 4;)
```

Questa regola scatena un alert di severità 4 quando, analizzando pacchetti HTTP, trova il pattern "/bin/ps": tale allarme è considerato un allarme del tipo web-attack.

Per uniformare a tale comportamento l'agente software, quindi, potrà utilizzare una espressione regolare che cercherà nelle informazioni ricevute il pattern /bin/ps, trovato il quale provvederà inserire tutte le informazioni nelle relative tabelle come abbiamo visto poco fa.

Sarà sufficiente una regex specifica per ogni regola di Snort e si saranno normalizzati gli allarmi. A questo punto manca l'ultima rifinitura: la possibilità di impostare degli allarmi che mettano in relazione, in maniera automatica, le varie tabelle del database degli allarmi. Un esempio teorico di regola, alla quale l'applicazione dovrà offrire una adeguata interfaccia, potrebbe essere la seguente:

Produci un allarme se entro 5 minuti, da qualsiasi dei device presenti, arrivano almeno 20 "denied connection" dallo stesso ip sorgente;
Produci un allrrnne se entro 5 minuti, da qualsiasi dei device presenti, arrivano almeno 10 signature del tipo web-attack dallo stesso ip sorgente.

Con il nuovo design, l'applicazione potrebbe necessitare di una modifica anche sostanziale, a tutto vantaggio di una infrastruttura più modulare e operativa. Ultimo punto, ma non certo per importanza, è la politica da seguire per conservazione dei log.

Che il database tenda, con il passare del tempo, a crescere di dimensione è ovvio. Serve, quindi, una politica di backup per quanto riguarda i dati storici, che devono comunque essere disponibili per eventuali richieste delle autorità giudiziarie (oltre che per verifiche tecniche).
livello operativo, si potrebbe impostare una politica di backup che effettui un tape, ad esempio, dei dati più vecchi di un mese, in modo da mantenere il db in

dimensioni accettabili e sempre con i soli dati necessari alle query sul db (se non vi sono regole di correlazione che vadano oltre il mese, ovviamente).

Appendice

Principali servizi e loro livello di vulnerabilità

FTP (21/TCP)
[Livello di vulnerabilità: ALTO]

• Il protocollo FTP non implementa la cifratura del traffico. Per questo motivo, un attacker è in grado di intercettare le sessioni, comprese le credenziali di accesso di sistema (nella forma login/password).

• Molte implementazioni del protocollo FTP hanno sofferto in passato di gravi problemi di sicurezza, alcuni dei quali consentono ad un attacker l'acquisizione remota dei privilegi di amministrazione.

• FTP consente una forma di amministrazione remota dei file: se le policy di firewalling permettono l'accesso dall'esterno a tale servizio, viene a crearsi una esposizione, dipendentemente dalla politica di gestione delle credenziali di accesso al sistema.

DNS (53/TCP/UDP)
[Livello di vulnerabilità: ALTO]

• Molte implementazioni del protocollo DNS hanno sofferto in passato di gravi problemi di sicurezza, alcuni dei quali consentono ad un attacker l'acquisizione remota dei privilegi di amministrazione.

• Se configurato in maniera errata, DNS consente ad un attacker di ottenere informazioni utili sui sistemi attestati su di una rete e sulla loro funzione (DNS AXFR).

SSH (22/TCP)
[Livello di vulnerabilità: MEDIO]

• Molte implementazioni dei protocolli SSH1 ed SSH2 (SSH, OpenSSH) hanno sofferto in passato di gravi problemi di sicurezza, alcuni dei quali consentono ad un attacker l'acquisizione remota dei privilegi di amministrazione.

• SSH consente l'amministrazione remota dei sistemi Unix: se le policy di firewalling permettono l'accesso dall'esterno a tale servizio, viene a crearsi una esposizione, dipendentemente dalla politica di gestione delle credenziali di accesso al sistema.

SMTP (25/TCP)
[Livello di vulnerabilità: MEDIO]

• Il protocollo SMTP non implementa la cifratura del traffico. Per questo motivo, un attacker è in grado di intercettare le .iasioni, quali messaggi e-mail ed eventuali allegati, potenzialmente contenenti dati sensibili.

• Se configurato in maniera errata, SMTP consente ad un attacker di individuare i nomi delle utenze valide sul sistema (attraverso i comandi VRFY ed EXPN).

• Se configurato in maniera errata, SMTP consente ad un attacker esterno l'invio anonimo di e-mail (open relay).

FINGER (79/TCP)
[Livello di vulnerabilità: BASSO]

• Il servizio FINGER consente ad un attacker di individuare i nomi delle utenze valide presenti sul sistema, allo scopo di lanciare un attacco mirato di tipo "brute force".

HTTP (80/TCP)
[Livello di vulnerabilità: ALTO]

• Il protocollo HTTP non implementa la cifratura del traffico. Per questo motivo un attacker è in grado di intercettare le sessioni, compresi eventuali dati sensibili in transito.

• Molte implementazioni del protocollo HTTP (Microsoft IIS, Apache) hanno sofferto in passato di gravi problemi di sicurezza, alcuni dei quali consentono ad un attacker di acquisire da remoto i privilegi di amministrazione.

• Le vulnerabilità dei programmi CGI e degli script presenti sul server http possono portare un attacker all'esecuzione di comandi arbitrar; ed all'acquisizione dei privilegi del server web.

POP-3 (110/TCP)
[Livello di vulnerabilità: MEDIO]

• Il protocollo POP-3 non prevede la cifratura del traffico. Per questo motivo, un attacker è in grado di intercettare le sessioni, comprese le credenziali di accesso al sistema (nella forma login/password).

SNMP (61/UDP)
[Livello di vulnerabilità: ALTO]

• il protocollo SNMP non implementa la cifratura del traffico. Per questo motivo, un attacker è in grado di intercettare le sessioni, comprese le credenziali di accesso al sistema (community SNMP).

• Il protocollo a layer di trasporto utilizzato da SNMP è UDP. Esso è soggetto ad attacchi del tipo spoofing o hijacking di sessione, che consentono ad un attacker di manipolare il traffico con relativa semplicità.

• Alcune implementazioni del protocollo SNMP hanno sofferto in passato di gravi problemi di sicurezza, alcuni dei quali consentono ad un attacker l'acquisizione remota dei privilegi di amministrazione.

• SNMP consente una forma di amministrazione remota dei sistemi: se le policy di firewalling permettono l'accesso dall'esterno a tale servizio, viene a crearsi una esposizione, dipendentemente dalla politica di gestione delle credenziali di accesso al sistema (communitv SNMP).

HTTPS(443/TCP)
[Livello di vulnerabilità: MEDIO]

• Molte implementazioni del protocollo HTTPS (Microsoft IIS, Apache-SSL) hanno sofferto in passato di gravi problemi di sicurezza, alcuni dei quali consentono ad un attacker l'acquisizione remota dei privilegi di amministrazione.

• Le vulnerabilità dei programmi CGI e degli script presenti sul server HTTPS possono portare un attacker all'esecuzione di comandi arbitrari e all'acquisizione dei privilegi del server web.

MYSQL (3306/TCP)
[Livello di vulnerabilità: MEDIO]

• I DBMS SQL non implementano generalmente la cifratura del traffico. Per questo motivo, un attacker è in grado di intercettare le sessioni, comprese le credenziali di accesso al sistema (nella forma login/password).

• Molte implementazioni di database relazionale SQL hanno sofferto in passato di gravi problemi di sicurezza, alcuni dei quali consentono ad un attacker l'acquisizione remota dei privilegi di amministrazione del database (o, in alcuni casi, dell'intero sistema).

- System Identification.

I file di log e la legge: l'obbligo di manutenzione dei log

Una delle querelle giuridiche più accese è quella che riguarda l'obbligo o meno della manutenzione dei file di log, ma è bene precisare che il termine log nel mondo del diritto potrebbe far sorridere un amministratore di sistema.

Infatti per quest'ultimo i log sono solo una categoria generale, divisibile in svariate sottocategorie (per cui avremo log del web server, log del mail server, log della macchina, ecc.), per i giuristi i log sono generalmente i file contenenti le tracce necessarie a ricostruire come è avvenuto un determinato reato commesso con il mezzo informatico.

I provvedimenti legislativi o regolamentari che parlano con un minimo di cognizione di causa dei file di log sono rarità e vanno sicuramente annoverate le circolari AIPA in tema, ad esempio, di redazione di piano di sicurezza per gli apparati della Pubblica Amministrazione.

Quando si parla di obbligo di manutenzione dei log si potrà stilare una classifica di quali log si devono assolutamente conservare e quali, invece, possono essere cestinati. Tale classifica terrà conto dei reati più comunemente commessi e della categoria professionale alla quale si appartiene. Nel caso si tratti di un ISP, infatti, è sicuramente consigliabile conservare i log di tutte le connessioni effettuate dai propri clienti. Nel caso di un webmaster, sarà bene invece conservare tutte le transazioni da e verso la pagina web che si amministra, così come è bene in generali conservare anche i log dei mail server.

Scendendo più nell'utenza domestica, non sono da sottovalutare i log della propria macchina, visto che spesso sono l'unica traccia utile per ricostruire se determinati file hanno subito delle modifiche "ad arte" o se il sistema stesso è stato infettato da qualche virus.

Non esiste comunque allo stato una norma ben specifica, visto che, anche a seguito dell'emanazione del d.lgs. 28 dicembre 2001 n° 467, che modifica la legge 675/96 sulla tutela dei dati personali, purtroppo, nessun accenno è stato fatto in relazione ai file di log ed alla manutenzione degli stessi.

Immediatamente dopo l'emanazione del d.lgs. menzionato, la Commissione Giustizia della Camera ha approvato una proposta di parere che imporrebbe al fornitore di servizi Internet la conservazione delle tracce informatiche per almeno un anno, così da facilitare eventuali indagini giudiziari per i crimini commessi a mezzo Internet.

Il problema della manutenzione dei file di log è comunque ben più sottile, dato il contrasto che potrebbe verificarsi tra l'eventuale obbligo di manutenzione e le esigenze di privacy del cittadino: tale contrasto è già sulla scrivania dei Garanti della privact dei vari paesi europei ed era già emerso durante la "Conferenza di Primavera" tenutasi ad Atene nel maggio del 2001.

In quella sede si sono ribadite le posizione espresse nella Conferenza dell'anno prima tenutasi a Stoccolma, per cui la conservazione prolungata di file atti a ricostruire fedelmente l'attività compiuta in rete da un determinato soggetto viola senza alcuna

ombra di dubbio il diritto di ogni persona all'intimità della propria sfera privata e, sempre nella stessa sede, si è esortato a regolamentare con norme ad hoc le esigenze di manutenzione prolungata dei file su menzionati che si potrebbero venire a verificare.

In Italia il quadro normativo è confuso. Il testo di partenza è costituito dal d.lgs. 13 maggio 1998 n° 171, che all'art. 4 consente di tenere traccia dei dati di fatturazione del traffico, con conseguente identificazione del chiamante, almeno finché la fattura può essere legalmente contestata.

Esiste poi un "decalogo" per la sicurezza ad opera dell'AIPA, che esorta all'individuazione di ogni singolo soggetto operante nella rete mediante appositi "activity log file", e che l'integrità di questi file di log è garantita per un periodo da concordare.

Inoltre, i provider iscritti all'AIIP garantiscono la conservazione del log file per un periodo di almeno 5 anni, pur se detto obbligo non è previsto dalla legge.

Infine la delibera 467/2000/CONS dell'AGCOM (Autorità Garante per le Comunicazioni) prevede la manutenzione dei file di log per verificare il rispetto delle misure minime di sicurezza ex DPR 318/99, previa identificazione dell'ISP nel titolare del trattamento dei dati e, quindi, soggetto alle norme previste dal DPR su menzionato, e la piena collaborazione ai fini delle indagini con le Autorità Giudiziarie.
In aggiunta a ciò e per meglio rispettare la privacy dell'utente, il file di log dovrebbe essere firmato digitalmente, disposto in maniere tale da consentire l'accesso alle informazioni relative alla sicurezza e non a quello relative alla privacy dell'utente e, infine, prevede la creazione di un protocollo per l'invio delle mail che cifri automaticamente il contenuto, così da non poterle rendere visibili ai fornitori di accesso.

Bibliografia

- **"Speciale Sicurezza"**
  di Matteo Falsetti, Marco Ivaldi, Pierluigi Perri, Daniele Verzelloni.
  Pubblicato sulla rivista mensile "Linux & C.", anno 5 numero 30
  http://www.oltrelinux.com

- **Syslog-NG Reference Manual**
  http://www.balabit.com/products/syslog_ng/reference/book1.html

- **"Syslog-NG"**
  di Balazs Scheidler (traduzione di F. Lamonica)
  Pubblicato sulla rivista Linux Gazette n. 43 – luglio 1999
  http://linux.cassino.edu/lgei/lgei9904/lgei9904_07.html

- **File di log: sicurezza ed analisi**
  di Yvette 'vodka' Agostini e Valerio 'Hypo' Verde
  http://www.sikurezza.org/webbit02/Log_analysis_1.ppt
  http://www.sikurezza.org/webbit02/Log_analysis_2.ppt

Links utili

- **SysLog-ng Official WebSite** (Inglese)
  http://www.balabit.com/products/syslog_ng/

- **SysLog-ng FAQ** (Inglese)
  http://www.campin.net/syslog-ng/faq.html

- **SysLog-ng + MySQL + PHP** (Inglese)
  http://vermeer.org/syslog/
  http://www.vermeer.org/projects/php-syslog-ng/

- **Esempi sulla configurazione di SysLog-ng** (Inglese)
  http://linux.cudeso.be/linuxdoc/syslog-ng.php

- **SysLog-ng + PostGreSQL** (Inglese)
  https://lists.balabit.hu/pipermail/syslog-ng/2002-April/003249.html

- **Central Loghost Mini-HOWTO** (Inglese)
  (Hacker Check/MySQL Support/Swatch/Stunnel)
  http://www.campin.net/newlogcheck.html

- **Secure Remote Logging with syslog-ng and stunnel HOWTO** (Inglese)
  http://venus.ece.ndsu.nodak.edu/~jezerr/linux/secure-remote-logging.html

- **Guida alla Sicurezza by Gentoo** (Italiano)
  http://www.gentoo.org/doc/it/gentoo-security.xml

Glossario

Access Point Wireless
Punto d'accesso per le reti wireless, ovvero, estensione radio della LAN.

All-In-One
Termine che sta ad indicare soluzioni multiple integrate nella medesima soluzione. Letteralmente tutto in uno.

Attacker
Solitamente colui che tenta di apportare un attacco ad un sistema.

Brute force
In crittografia e in altri campi matematici, un programma che fa eseguire al computer una funzione semplice, ma ripetuta un'infinità di volte nella speranza di trovare la soluzione a furia di tentativi. Questo sistema viene ad esempio utilizzato per scoprire la password di accesso ad un sistema.

Connection-oriented network
Una rete in cui, prima di iniziare qualsiasi trasmissione, ci si assicura che il desinatario sia pronto a ricevere, scambiando con esso una sequenza di messaggi di servizio che servono a creare una sessione di collegamento tra mittente e destinatario. Tale sessione rimarrà attiva fino al termine del trasferimento dati.

Daemon
In UNIX un programma che funziona costantemente in background sul server, pronto ad entrare in azione all'occorrenza su richiesta di altri orogrammi.

DNS (Domain Name System)
Sistema di denominazione del dominio. Un database distribuito utilizzato in ambiente TCP/IP per creare una corrispondenza tra indirizzo numerico e nome mnemonico assegnato alla macchina. E' distribuito perchè nessun computer collegato a Internet ha una situazione completa dell'intera rete, ma deve usare il particolare protocollo previsto dal DNS per comunicare con gli altri sistemi e procurarsi le informazioni.

DoS (Denail of Service)
Attacco mirato non al controllo, da parte di malintenzionati, del sistema, ma all'interruzione di sevizi.

Download
Copiare dati (di solito un'intero file elettronico) sul proprio computer, prelevandoli da una fonte primaria.

Firewall
Un meccanismo hardware e/o software che permette d'impostare restrizioni sull'accesso a uno o più computer collegati a internet, normalmente per motivi di sicurezza.

FTP (File Transfer Protocol)
Un insieme di regole per abilitare il colloquio bidirezionale tra due computer durante il trasferimento di un file. Il nome si riferisce sia al protocollo di trasmissione sia al programma che ne fa uso. Il protocollo viene usato quando si scarica un file sul proprio computer prelevandolo da un sito internet e racchiude in sè i comandi TCP/IP per eseguire login sulla rete, visuallizare l'elenco dei files esistenti e copiarli sulla stazione locale.

HTTP (HyperText Transfer Protocol)
Protocollo di trasferimento per ipertesti. Il protocollo di trasferimento usato per convogliare su Intenet le pagine del World Wide Web e concepito espressamente per consentire la creazione di collegamenti ipertestuali tra i documenti. Le quattro operazioni svolte in sequenza dal protoccollo sono connessione col server che contiene il documento, richiesta del documento, recupero della pagina interessata attraverso la risposta del server e chiusura della connessione. Questo ciclo viene ripetuto per ciascuna pagina richiesta.

ISP (Internet Service Provider)
Una socieyà che fornisce accesso a internet ad aziende e a clienti privati. Offre servizi di vario tipo, tra cui la posta elettronica, e dispone di una connessione diretta con le linee di accesso internazionali.

LAN (Local Area Network)
Una rete d'interconnessione tra diversi computer entro un'area delimitata dai muri dell'edificio o dal perimetro dello stabilimento in cui viene installata, oppure dal raggio di pochi chilometri nel caso in cui non esistano confini di riferimento precisi. Consente lo scambio diretto di dati in formato elettronico tra due o più computer.

Password Cracking
Tecnica mediante la quale è possibile recuperare parzialmente o integralmente la password di un sistema.

Performance tuning
Termine utilizzato per indicare le tecniche di ottimizzazione delle prestazioni del sistema.

POP-3 (Post Office Protocol v3)
Nasce per recuperare posta dal server senza eccessiva complessità ed è dedicato a quelle macchine che non sono constantemente collegate a Internet e hanno bisogno di uno strumento per recuperare di tanto in tanto tutti i nuovi messaggi arrivati nella casella postale.

Proxy server
Una macchina che intercetta le richieste che arrivano dai vari client a cui è abbinata, procura dai vari siti le pagine HTML corrispondenti e le replica in locale così che siano consultabili da più persone senza richiederle continuamente a internet. Può essere utilizzato anche come sistema per

abbinare una intranet locale a internet. Il proxy in questo caso, oltre a fungere da cache, agisce ache da filtro del traffico.

RAS (Remote Access Service)

Una funzione software fornita con Windows per Workgroup e Windows su tecnologia NT per accedere in remoto a un server e, attraverso di questo, alle risorse condivise sulla rete.

Real-time

Si riferisce a una modalità operativa di alcuni computer in cui le informazioni ricevute vengono elaborate immediatamente, fornendo risultati quasi istantanei.

S-HTTP (Secure-HyperText Transport Protocol)

Un'estensione del protocollo HTTP che permette la trasmissione di pagine web cifrate, consentendo quindi lo scambio su Internet di informazioni protette.

Socket Interfacca di comunicazione UNIX sviluppata dall'Università di Berkley per consentire l'agevole comunicazione tra processi e applicazioni attraverso la rete.

SSL (Secure Sockets Layer)

Un protocollo e un metodo di crittografia proposto da Netscape per proteggere le informazioni che circolano su internet. Definisce i meccanismi di trasporto delle informazioni tra un browser e un server web al fine di eseguire transazioni sicure su internet.

TCP (Trasmission Control Protocol)

Un protocollo di trasporto realizzato su specifiche richieste del Dipartimento della Difesa americano come componente della rete ARPAnet al fine di consentire l'interconnessione di sistemi tra loro differenti. E' uno standard di fatto nel mondo UNIX e costituisce l'ossatura portante di internet.

UDP (User Datagram Protocol)

Un protocollo utilizzato per lo scambio di messaggi tra applicazioni. E' di tipo connection-less perciò non grantisce la consegna del messaggio nè la corretta sequenza di pacchetti che eventualmente lo compongono nè l'unicità del pacchetto, però, è veloce. Lavora a livello di trasporto (livello 4 nel modello ISO/OSI)

VPN (Virtual Private Network)

Un servizio offerto dai gestori pubblici di telecomunicazioni che garantisce le stesse prerogative tipiche di una rete privata

condizionamento della lnea (per una migliore qualità del segnale sulla lunga distanza), verifica del livello di errore, altà velocità, trasmissione full duplex.

Worm

Può essere sinonimo di robot, ma tecnicamente il termine indica un programma replicante che si propaga sulla rete, a differenza di un robot che invece non si muove dalla sua sede e si limita a recuperare documenti attraverso la rete.