

IPS FLEXIBLE RESPONSE

Pierpaolo Palazzoli, Brescia, Italy

Fabio Mostarda, Brescia, Italy

Claudia Ghelfi, Brescia, Italy

L'acronimo IPS fa riferimento ad un Intrusion Prevention System, ovvero ad un dispositivo per la sicurezza di rete. Tale sistema monitora il flusso di traffico in transito sulla rete per bloccare, in tempo reale, eventuali attacchi di origine dolosa o involontaria.

L'IPS può essere implementato mediante una macchina sulla quale il software Snort svolge le funzionalità di IPS, mentre Iptables si occupa di adempiere ai compiti specifici di un firewall e di creare, appoggiandosi ad appositi moduli, le code dei pacchetti da analizzare.

SNORT

Snort è per definizione un sistema di rilevamento e prevenzione delle intrusioni (IDPS) e può operare in due diverse modalità. In entrambe le modalità l'IPS risulta trasparente e quindi invisibile sulla rete.

Solo Snort offre il funzionamento in modalità Flexible Response, tutti i dispositivi analoghi infatti operano unicamente Inline.

- *Inline*. In questa modalità Snort funziona come un bridge Ethernet, ovvero, per poter monitorare il traffico in un segmento di rete, va inserito in maniera trasparente tramite due schede in bridge.

Così facendo il pacchetto attraversa indisturbato il bridge da una scheda all'altra, a meno che non rispetti le regole di drop. In tale caso lo switch si apre e blocca il passaggio del pacchetto.

- *Flexible Response*. In questa seconda modalità tutto il traffico in ingresso viene replicato dallo switch verso la "mirror port". Su tale porta viene inserito l'ips che analizza il flusso dei pacchetti e, nel caso in cui esso rispetti le regole di drop, si preoccupa di terminare la connessione corrispondente. Il termine "*flexible response*" richiama con molta probabilità una strategia difensiva messa in atto negli anni sessanta dal presidente americano John Kennedy. Con questa soluzione tattica, Kennedy decise di rispondere con mezzi proporzionali alla minaccia e non secondo i canoni della rappresaglia massiccia ad ogni azione nemica, la cosiddetta Massive Retaliation, dato che ormai gli

Stati Uniti erano stati raggiunti dall'Unione Sovietica nel campo degli armamenti.

Le migliori prestazioni come IPS sono ottenute nella modalità inline perché essa permette di bloccare anche attacchi formati da un numero limitato di piccoli pacchetti. Al contrario la modalità flexible response impiega più tempo ad identificare un'intrusione e quindi a resettare la corrispondente porta.

D'altra parte, nei casi in cui la latenza introdotta da un IPS in modalità Inline sia troppo elevata, è conveniente usare un IPS in modalità Flexible Response.

È comunque da considerare che Snort flexresp è un dispositivo passivo e quindi la sua efficacia è direttamente dipendente dall'occupazione della CPU e dai collegamenti del sistema su cui è implementato, dalla memoria disponibile, dalla situazione di I/O e dalle latenze di rete.

Snort offre due modi alternativi di operare in Flexible Response:

- FlexRespo: non 'blocca' realmente le connessioni, ma invia messaggi di errore ai mittenti dei pacchetti malevoli inducendoli a credere che la rete, piuttosto che la porta o il dispositivo non esiste oppure non è raggiungibile. Ogni utente ha la possibilità di configurare le proprie regole, che tenteranno di terminare i tentativi di collegamento.
- FlexRespo2: rappresenta la nuova versione e sostituisce le libdnet con le libevent. Flexrespo2, oltre alle funzionalità del FlexRespo originale, previene il verificarsi di loop di risposta che possono provocare il sovraccarico della CPU o più semplicemente l'interruzione del servizio. Inoltre si preoccupa di inviare risposte multiple per assicurarsi che il collegamento venga correttamente terminato. In conclusione è conveniente implementare questa seconda versione di FlexRespo, se possibile, poiché la prima risulta primitiva.

APPLICAZIONE

L'introduzione di un IPS in modalità Flexible Response è subordinata all'utilizzo di switches dotati di mirror port, una particolare porta sulla quale viene replicato tutto il traffico di rete.

La mirror port acquisisce nomenclature diverse a seconda del costruttore, ad esempio, nel caso di Cisco è detta SPAN (Switched Port ANalyzer).

Il traffico della mirror port replicato viene convogliato verso l'IPS, che lo analizza. Nel caso in cui vengano riconosciuti pacchetti malevoli, Snort può operare in due diversi modi:

- RST, con cui cancella i pacchetti pericolosi;

- ICMP, con cui comunica al mittente dell'attacco che la rete, piuttosto che l'host o la porta in questione diviene irraggiungibile.

Un IPS può ricevere ed analizzare il traffico proveniente da più mirror port generando, su un'unica porta in uscita, RST oppure ICMP.

La realizzazione di un IPS con Snort in modalità Flexible Response segue questi passi:

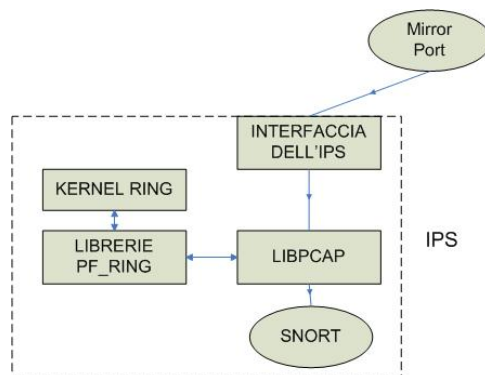
1. **Installazione dei pacchetti propedeutici:**

apt-get install vim gcc make build-essential libtool automake autoconf flex bison libpcre3-dev psmisc ethtool apache2 ntop ntpdate subversion apache2 kernel-package libncurses5-dev fakeroot wget bzip2 flex bison zlibc zlib1g-dev

2. **Implementazione del PF_RING e installazione delle librerie Libpcap relative:**

PF_RING è una tipologia di socket di rete che accresce notevolmente la velocità di cattura dei pacchetti.

Per questo motivo, se inserita nella struttura dell'IPS, come qui schematizzato, ne aumenta le prestazioni.



Innanzitutto si scarica il PF_RING, in questo caso l'ultima versione:

```
cd /usr/src
svn co https://svn.ntop.org/svn/ntop/trunk/PF_RING/ scarico ultima versione
pf_ring
```

Si verifica il kernel attualmente in uso:

```
uname -a
```

Si modifica lo script scaricato affinché generi la **patch** del kernel che si preferisce, in questo caso si è scelto di creare la patch per quello in uso:

```
cd /usr/src/PF_RING/
vi mkpatch.sh
edito kernel giusto
VERSION=${VERSION:-2}
PATCHLEVEL=${PATCHLEVEL:-6}
SUBLEVEL=${SUBLEVEL:-26}
```

Si lancia lo script che genera la patch:

```
sh ./mkpatch.sh
cd workspace/
```

```
cd linux-2.6.26-1-686-smp-PF_RING
cp /boot/config-2.6.26-1-686-bigmem .config
```

Si abilita il pfring cosicché ne venga tenuto conto nella generazione del **nuovo kernel**:

```
make menuconfig
```

Seleziona *Networking* -> *Networking Options* e ci si assicura che PF_RING socket sia abilitato.

Si esce dal menù salvando le modifiche effettuate.

Ora è possibile generare il pacchetto contenente il nuovo kernel e installarlo:

```
make-kpkg clean
fakeroot make-kpkg -initrd -revision=pfring.1.0 linux-image
cd ../
dpkg -i linux-image-2.6.26_pfring.1.0_i386.deb
```

L'utilizzo di questo nuovo kernel di default ad ogni boot va impostato nel seguente file:

```
vi /boot/grub/menu.lst
```

Creiamo il file :

```
vi /etc/modprobe.d/options
```

inserendo la riga : options ring num_slots=32740 transparent_mode=0

Dove 65535 è la memoria occupata dal buffer ring. (scegliere a seconda della disponibilità, in caso sia maggiore bisogna cambiare nel grub la vmalloc=256M esempio di 256M)

Con il riavvio della macchina si verifica che il nuovo kernel venga effettivamente caricato:

```
reboot
```

Copio il sorgente del ring nella directory linux:

```
cp /usr/src/PF_RING/workspace/linux-2.6.26-1-686-smp-PF_RING/include/
linux/ring.h /usr/include/linux/
cd /usr/src/PF_RING/userland
make
```

A questo punto si installano le librerie **Libpfring**:

```
cd lib
```

```
gcc -shared -Wl,-soname -Wl,libpfring.so.0.9.7 -o libpfring.so.0.9.7 *.o -lc
cp libpfring.a libpfring.so.0.9.7 /usr/local/lib
cp pfring.h /usr/local/include
ln -s /usr/local/lib/libpfring.so.0.9.7 /usr/local/lib/libpfring.so
ldconfig
```

```
ldconfig -v |grep pfring
```

Mediante quest'ultima istruzione si controlla che le librerie installate siano quelle corrette. La risposta deve quindi essere:
libpfring.so.0.9.7 -> libpfring.so.0.9.7

Infine, si installano le librerie **Libpcap-ring**:

```
cd /usr/src/PF_RING/userland
wget http://www.tcpdump.org/release/libpcap-0.9.7.tar.gz
tar -zxvf libpcap-0.9.7.tar.gz
cd libpcap-0.9.7
mv pcap-int.h pcap-int.h.orig
mv pcap-linux.c pcap-linux.c.orig
cp ../libpcap-0.9.7-ring/pcap* .
./configure CPPFLAGS="-I/usr/local/include" LDFLAGS="-L/usr/local/lib"
CFLAGS="-D_LARGEFILE_SOURCE -D_LARGEFILE64_SOURCE -D_FILE_OFFSET_BITS=64"
make && gcc -shared -Wl,-soname -Wl,libpcap.so.`cat VERSION` -o
libpcap.so.`cat VERSION` *.o -lc
make install && cp libpcap.so.0.9.7 /usr/local/lib
ldconfig -v |grep pcap
```

Mediante quest'ultima istruzione si controlla che le librerie installate siano quelle corrette. La risposta deve quindi essere:
libpcap.so.0.9.7 -> libpcap.so.0.9.7

3. **Installazione delle librerie Libdnet:**

Si opera come segue:

```
wget http://prdownloads.sourceforge.net/libdnet/libdnet-1.11.tar.gz?download
tar zxvf libdnet-1.11.tar.gz
cd libdnet-1.11
./configure
make
make install
ldconfig (riassetta le librerie)
```

4. **Installazione di Snort:**

Si installa Snort eseguendo:

```
wget http://www.snort.org/dl/snort-2.8.3.2.tar.gz
tar zxvf snort-2.8.3.1.tar.gz
cd snort-2.8.3.1
./configure --enable-flexresp2 --enable-memory-cleanup --enable-linux-smp-
stats --enable-pthread
vi src/Makefile
```

Modifico il Makefile in modo tale che le righe seguenti risultino:

```
LDFLAGS = -L/usr/lib -lpcap -L/usr/local/lib -ldnet -lpfring -lpcap
CPPFLAGS = -DENABLE_RESPONSE2 -I/usr/local/include -fno-strict-aliasing
```

Se tutto fin qui è andato a buon fine, si esegue:

```
make
make install
cd ..
mkdir /var/log/snort
```

5. **Definizione delle regole di Snort:**

Le regole secondo cui Snort decide di resettare le connessioni o meno, vengono inserite in `/etc/snort/rules/`, organizzate in files.

Al termine di ogni regola deve essere inserito: `resp: reset_both,icmp_all;` che si riferisce alla modalità flexible response di funzionamento di Snort.

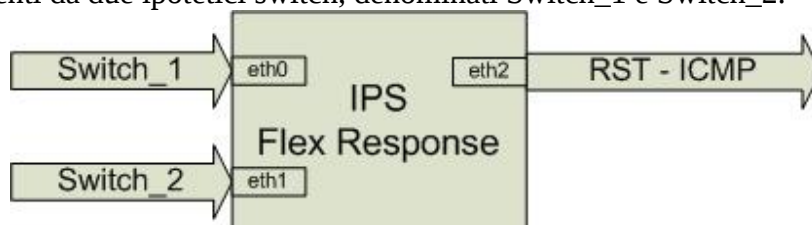
Un esempio di regola è il seguente

```
alert tcp $EXTERNAL_NET 6112 -> $HOME_NET any (msg:"ET GAMES Battle.net 'emote' message"; flow:established,from_server; content:"|FF 0F|"; depth:2; content:"|17 00 00 00|"; offset:4; depth:4; classtype: policy-violation; sid:2002152; rev:2; resp: reset_both,icmp_all;)
```

6. **Avvio di Snort:**

Infine si avvia uno Snort, con regole dedicate, relativo al traffico in arrivo da ogni mirror port.

Si consideri, per esempio, il caso in cui si abbiano due flussi in ingresso provenienti da due ipotetici switch, denominati Switch_1 e Switch_2.



In questo caso particolare Snort deve essere avviato due volte, una relativa allo Switch_1:

```
snort -A fast -i eth0 -b -d -D -c /etc/snort/snort_1.conf
```

e una relativa allo Switch_2:

```
snort -A fast -i eth1 -b -d -D -c /etc/snort/snort_2.conf
```