



Italian OpenBSD Users Group

Cluster di Firewall con OpenBSD

Ver. 0.1

13 Febbraio 2005

by

fabioFVZ

fabio@opengeeks.it

www.opengeeks.it

Introduzione

Dopo l'uscita della release 3.5 di OpenBSD costruire un Cluster di Firewall e' diventato molto semplice e alla portata di tutti. Questo grazie al team di OpenBSD che ha introdotto due nuovi protocolli di rete chiamati: CARP - Common Address Redundancy Protocol e PFSYNC.

Il Cluster, in questo caso, viene inteso come ad "Alta Disponibilita'".

Per esemplificare facciamo un'esempio: collegando 2 Firewall assieme (vedi fig.1), in caso di rottura del Firewall principale, tutto il traffico passera' sul secondario, in maniera veloce e del tutto trasparente, mantenendo tutte le connessioni che erano in atto in quel momento.

Il Protocollo CARP

Il protocollo CARP deriva dal protocollo VRRP di Cisco, migliorandolo in molti aspetti. Le differenze piu'interessanti sono quattro, vediamole:

- CARP, a differenza di VRRP, e' esente da Brevetti;
- Ha la possibilita' di avere un singolo IP Virtuale con piu' MAC address virtuali (uno per ogni host.) Questa funzione viene definita "Arp Balance"
- Utilizza la crittografia SHA1-HMAC all'interno dei pacchetti CARP di Advertisement rendendo molto difficile la creazione di pacchetti AD-HOC.
- Supporta sia IPV4 che IPV6 ed e' registrato con il numero 112 IP Protocol

CARP lavora a layer 2 e 3 dello stack OSI virtualizzando sia il MAC address che l'IP address appartenenti allo stesso gruppo CARP

Vediamo ora in dettaglio l'header di un pacchetto CARP

Dal file ip_carp.h

```
/*
 * The CARP header layout is as follows:
 *
 *      0              1              2              3
 *      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 *      +-----+-----+-----+-----+-----+-----+-----+-----+
 *      |Version| Type  | VirtualHostID |   AdvSkew   |   Auth Len   |
 *      +-----+-----+-----+-----+-----+-----+-----+-----+
 *      |   Reserved   |   AdvBase   |           Checksum           |
 *      +-----+-----+-----+-----+-----+-----+-----+-----+
 *      |                                     Counter (1)                                     |
 *      +-----+-----+-----+-----+-----+-----+-----+-----+
 *      |                                     Counter (2)                                     |
 *      +-----+-----+-----+-----+-----+-----+-----+-----+
 *      |                                     SHA-1 HMAC (1)                                     |
 *      +-----+-----+-----+-----+-----+-----+-----+-----+
 *      |                                     SHA-1 HMAC (2)                                     |
 *      +-----+-----+-----+-----+-----+-----+-----+-----+
 *      |                                     SHA-1 HMAC (3)                                     |
 *      +-----+-----+-----+-----+-----+-----+-----+-----+
 *      |                                     SHA-1 HMAC (4)                                     |
 *      +-----+-----+-----+-----+-----+-----+-----+-----+
 *      |                                     SHA-1 HMAC (5)                                     |
 *      +-----+-----+-----+-----+-----+-----+-----+-----+
 *
 */
```

Come si puo' vedere l'header e' di 36 byte, vediamo in dettaglio:

Version:	Versione del Protocollo CARP
Type:	Tipo CARP_ADVERTISEMENT settato a 0x01
VirtualHostID:	ID utilizzato per i riconoscere i gruppi
ADVSkew:	Ritardo invio pacchetti di ADV
Auth Len:	Lunghezza dell'autenticazione
Reserved:	Riservato ad usi futuri
AdvBase:	Tempo di invio pacchetti
Checksum:	Controllo integrita dei pacchetti
Counter 1 e 2:	Semplice Contatore
SHA1-MAC:	Codice Crittografato. La lunghezza della chiave e' di 20 bytes

Il protocollo PFSYNC

PFSYNC e' il protocollo (IP Protocol 240) fondamentale per gestire il nostro Firewall. E' necessario solamente se il Firewall viene configurato per utilizzare lo Stateful Inspection o utilizziamo il Firewall anche come NAT. In questo caso penso ci interessi mantenere in piedi le connessioni in atto durante il "Fault" del Firewall Master.

Come si puo' intuire in tutti e due i casi (Stateful Inspection o NAT) serve utilizzare la "Tabella di Stato" la quale tiene traccia delle connessioni in corso. Il protocollo PFSYNC serve proprio a tenere sincronizzate le "Tabelle di Stato" del/i Firewall di Backup.

Per questioni di sicurezza si consiglia l'uso di una rete dedicata, collegata con un apposito cavo cross (nel caso di 2 Firewall). Questo per evitare sia la perdita di pacchetti che l'invio di pacchetti fasulli inviati ad-hoc che potrebbero andare a cambiare le "Tabelle di Stato".

Anche PFSYNC come CARP utilizza messaggi inviati in broadcast, ma, a differenza di CARP, PFSYNC per adesso non usa la crittografia per l'invio dei messaggi.

Costruiamo il nostro Firewall

Partiamo da un esempio di una piccola rete aziendale come in figura 1.

Come sistema d'esempio ho utilizzato 2 Soekris net4801 con una CPU Geode (586) a 266Mhz e 3 Schede di rete della National Semiconductor viste da OpenBSD come sis0/2.

L'ADSL entra in un router il quale ha una rete 10.0.0.x nella sua parte LAN. Il suo numero IP e' 10.0.0.1. Ricordatevi di impostare una static route nel Router, in questo caso gli indichiamo di instradare tutta la rete 192.168.1.0/24 verso il gateway 10.0.0.2 (l'ip Virtuale).

Gli utenti lato LAN devono settare il proprio Gateway su 192.168.1.1 (il virtuale lato LAN).

Vediamo la configurazione del Master

```
/etc/hostname.sis0
inet 10.0.0.200 255.255.255.0 NONE NONE

/etc/hostname.sis1
inet 192.168.1.200 255.255.255.0 NONE NONE

/etc/hostname.sis2
inet 192.168.254.200 255.255.255.0 NONE NONE

/etc/hostname.carp0
inet 10.0.0.1 255.255.255.0 10.0.0.255 vhid 1 pass 12345

/etc/hostname.carp1
inet 192.168.1.1 255.255.255.0 192.168.1.255 vhid 2 pass 5678

/etc/hostname.pfsync0
up syncif sis2
```

Vediamo la configurazione del Backup

```
/etc/hostname.sis0
inet 10.0.0.201 255.255.255.0 NONE NONE

/etc/hostname.sis1
inet 192.168.1.201 255.255.255.0 NONE NONE

/etc/hostname.sis2
inet 192.168.254.201 255.255.255.0 NONE NONE

/etc/hostname.carp0
inet 10.0.0.1 255.255.255.0 10.0.0.255 vhid 1 pass 12345

/etc/hostname.carp1
inet 192.168.1.1 255.255.255.0 192.168.1.255 vhid 2 pass 5678

/etc/hostname.pfsync0
up syncdev (syncif sis2
```

Come vedete la configurazione e' praticamente la stessa. In questo caso all'avvio, il primo server che trasmette l'Advertisement diventa Master ed l'altro lo Slave. Se si vuole imporre una macchina come Slave (quando il Master funziona) dovete aggiungere il parametro advskew con un valore che puo' variare da 1 a 255 nel server di Backup:

```

/etc/hostname.carp0
inet 10.0.0.1 255.255.255.0 10.0.0.255 vhid 1 advskew 100 pass 12345

/etc/hostname.carp1
inet 192.168.1.1 255.255.255.0 192.168.1.255 vhid 2 advskew 100 pass 5678

```

Inoltre in tutti e due i server deve essere impostato il seguente valore nel kernel.

```
sysctl -w net.inet.carp.preempt=1
```

Conviene inserire la riga nel vostro file /etc/sysctl.conf

```
net.inet.carp.preempt=1
```

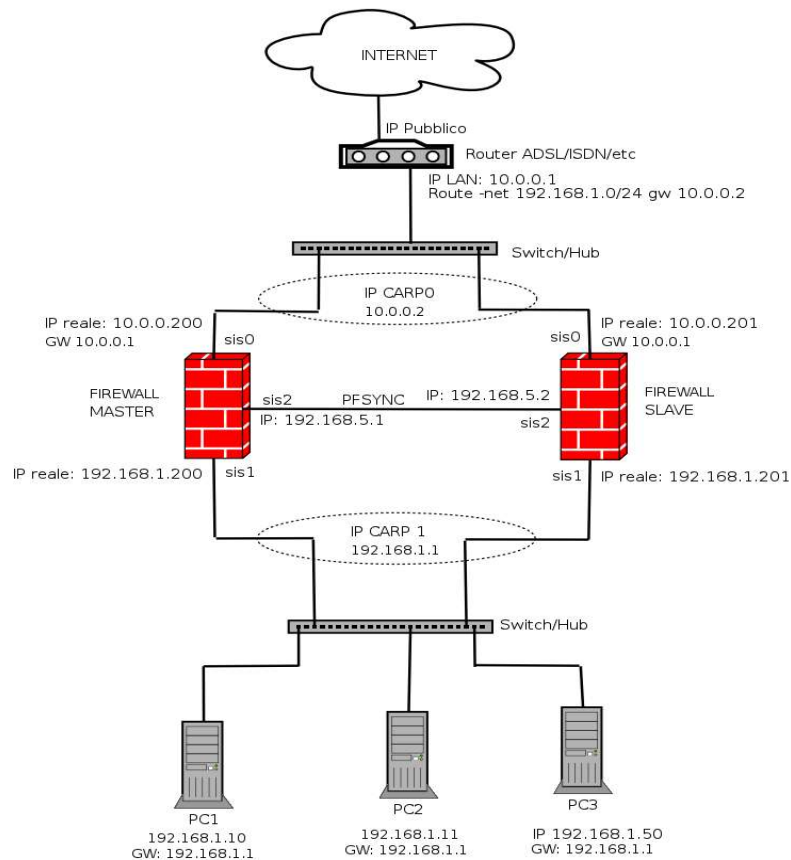
Per capire il valore advskew dobbiamo comprendere questa formula:

$$\text{advbase} + (\text{advskew} / 255) = \text{Tempo in secondi tra 2 Advertise}$$

Advbase di default e' settato a 1 secondo e advskew = 0 che significa avviene una trasmissione ogni secondo.

Se il Backup non "sente" per 3 volte di questi intervalli il pacchetto di Advertise, il backup passa a Master ed invia lui i pacchetti.

ESEMPIO DI RETE CON CLUSTER FIREWALL



Esempio fig. 1

Come funziona?

Nel normale funzionamento il Master invia i suoi messaggi di CARP per informare che è funzionante. I messaggi vengono inviati ogni secondo (se non avete specificato un valore di advskew). Tutto il traffico passa attraverso il Master il quale oltre ad aggiornare la propria tabella di stato, invia dei pacchetti sull'interfaccia di PFSYNC di aggiornamento. I backup prendono questi pacchetti ed aggiornano la propria tabella di stato.

Quando avviene la rottura il Backup non riceve più i pacchetti di advertise (circa 3 secondi per una configurazione standard) il Backup passa in Master e si mette ad inviare pacchetti CARP ed a tenere aggiornata la tabella di stato. Vengono inoltre inviati i segnali di PFSYNC perché potrebbero esserci più Firewall di Backup e devono essere aggiornati.

Appena il Master "Originale" ritorna, invia un pacchetto PFSYNC con il messaggio di BULK_UPDATE. Questo fa inviare dal "Attuale Master" una serie di pacchetti PFSYNC per risincronizzare la tabella. Alla fine della sincronizzazione (inviando un pacchetto PFSYNC con valore END_BULK_UPDATE, il master Originale ricomincia ad reinviare pacchetti CARP che fanno tornare a Backup, l'attuale Master.

Esempio di messaggi CARP

```
# tcpdump -v -n -i sis0
tcpdump: listening on sis0
18:48:54.540660 carp 10.0.0.200 > 224.0.0.18: CARPv2-advertise 36: vhid=1
adibase=1 advskew=0 (DF) [tos 0x10] (ttl 255,)
18:48:55.550642 carp 10.0.0.200 > 224.0.0.18: CARPv2-advertise 36: vhid=1
adibase=1 advskew=0 (DF) [tos 0x10] (ttl 255,)
```

Ulteriori controlli

Come avete visto il funzionamento è molto semplice e la configurazione ancora di più. Per rendere più raffinato il nostro Firewall è utile che aggiungiamo alcune funzioni molto utili.

Ricordatevi di aggiungere alle vostre regole di PF il passaggio dei pacchetti CARP e PFSYNC come nell'esempio qui sotto:

```
# Riferendoci all'esempio di fig. 1

pass quick on { sis2 } proto pfsync
pass on { sis0 sis1 } proto carp keep state
```

Ricordatevi che il file delle regole di PF deve essere sincronizzato su tutti i server. Un sistema è quello di utilizzare il programma rsync per copiare i file e risincronizzare le regole ad ogni modifica. Questo può essere fatto sia dal Master verso i Backup (push) o viceversa.

Inoltre è utile sapere quando avviene un FAULT del Master. In questo caso andremo a creare uno semplice script che monitorizzi lo stato in cui si trova il server di "riserva" che normalmente è in stato di "Backup".

Per sapere lo stato di CARP basta utilizzare ifconfig carpX come nell'esempio seguente:

```
# ifconfig carp0
carp0: flags=41<UP,RUNNING> mtu 1500
      carp: MASTER vhid 1 adibase 1 advskew 0
      inet 192.168.1.1 netmask 0xffffffff0
```

Come si può vedere nella seconda riga si trova lo stato di CARP che in questa situazione è settato in Master.

Lo script deve monitorizzare lo stato del Backup ed inviare una mail nel caso lo stato cambi in MASTER. E' inutile avere il Fault Tolerance se poi non veniamo avvertiti che qualcosa non sta funzionando.

Conclusione

Come avete visto creare un Cluster di Firewall e' molto semplice e con una spesa molto contenuta.

Visto il notevole risparmio monetario rispetto l' acquisto di sistemi di ben "nota marca", potreste aiutare il team di OpenBSD acquistando il cofanetto di CD originale presso il sito:

www.openbsd.org o dal grande WIM www.kd85.com

Inviatemi pure suggerimenti e/o correzioni.

Riferimenti

Sito OpenBSD: <http://www.openbsd.org>

Carp su OpenBSD: <http://www.openbsd.org/lyrics.html#35>

Guida in Inglese: <http://www.countersiege.com/doc/pfsync-carp/>
man: carp(4) – pfsync(4) – pf(4) – pfctl(8) – ifconfig(8)