

DynDNS tra Client e server Linux Ubuntu (Client 12.10 e server 8.04 LTS)

Di base tutti i pacchetti che servono per mettere in marcia questo servizio sono già installati sia sul server che nel client, nelle loro configurazioni di base.

Senza entrare troppo nella teoria che sta sotto questo sistema, l'aggiornamento del DNS con dati dinamici è possibile tramite una autenticazione tra il client ed il server con l'utilizzo di chiavi, che sono create attraverso un comando specifico e che devono essere custodite e protette per evitare abusi.

Le chiavi devono essere copiate nella cartella dove è depositato il zone file. Come esempio prendo il mio dominio di "gioco" **adunanza.ch**, che voglio che punti ad una macchina virtuale ubuntu installata fresca fresca su un computer remoto, che uso poi per delle prove.

Parte server

Prima di tutto scelgo di spostare i files di descrizione del dominio dinamico (o dei domini dinamici) dalla cartella `/etc/bind/zones` alla cartella `/var/named`, perché con i domini DDNS `named`, durante la loro configurazione, scrive dei files nella cartella dove sono depositati, e non voglio che ci siano problemi e promiscuità con i domini statici (sarò un purista (-: ?)).

Per prima cosa creo la cartella `/var/named`, se questa non esiste già. I permessi sono `777`, l'utente `root.bind`. Mi assicuro di chownarla e di chmodarla se così non fosse.

Entro poi nella cartella `/var/named` e creo le chiavi con il comando:

```
dnnsec-keygen -a HMAC-MD5 -b 512 -n USER adunanza.ch.
```

Dopo un attimo i due files (`.key` e `.private`) sono generati. Prendo la chiave contenuta nel file `.key`, copio.

Se non esiste già, creo nella stessa cartella il file `keys.conf`, e aggiungo il seguente contenuto

```
key adunanza.ch. {  
    algorithm "HMAC-MD5";  
    secret "chiave incollata";  
};
```

In **chiave incollata** incollo la chiave copiata dal file precedente.

Altre chiavi per altri domini saranno poi generate come sopra descritto, ed aggiunte a questo file.

In `/etc/bind/named.conf` aggiungo in fondo la riga seguente per includere il key file:

```
// includo le chiavi per la funzione di DynDNS  
include "/var/named/keys.conf";
```

Ora in `/etc/bind/named.conf.local` aggiungo il mio dominio `adunanza.ch`:

```
// sezione domini DynDNS  
  
// adunanza.ch  
zone "adunanza.ch" {  
    type master;  
    file "/var/named/adunanza.ch.db";  
    allow-update {  
        key adunanza.ch;  
    };  
};
```

...che in pratica include un file statico di base come tutti gli altri domini, con una parte aggiornata dinamicamente (parte `allow-update` con la chiave da usare per permettere le modifiche richieste dal client). Importante ricordare che il zone file `adunanza.ch.db` non è assieme agli altri statici in `/etc/bind/zones`, ma nella cartella `/var/named`. Questo è importante perché è nella stessa cartella verrà creato il file dinamico, e se si trova altrove il file dinamico non potrà essere creato (i permessi della `/etc/bind/zones` non dovrebbero assolutamente essere così aperti!).

Il file `adunanza.ch.db` contiene la seguente configurazione:

```
; BIND data file for adunanza.ch zone
;
$ORIGIN .
$TTL 60
adunanza.ch.      IN SOA      gesero.nelweb.ch.
hostmaster.nelweb.ch. (
                    2012102002      ; Serial
                    10800           ; refresh
                    3600            ; Retry
                    604800          ; Expire
                    60 )            ; Negative Cache TTL

$TTL 3600

                    IN  NS          gesero.nelweb.ch.
                    IN  NS          ns1.cicitt.ch.
adunanza.ch.      IN  MX          10      adunanza.ch.

$ORIGIN adunanza.ch.
```

`gesero.nelweb.ch` è il nome del mio server DNS (primary), `ns1.cicitt.ch` il nome del mio secondary DNS.

Bene, ora riavviando il server (`/etc/init.d/bind9 restart`) dovremmo poter caricare il tutto. Con il comando `dig @localhost adunanza.ch ANY` dovremmo poter verificare che funziona.



Parte client

Se il server è ok, il client non è troppo complicato (basta il copia-incolla dello script...).

Seguendo un esempio trovato su internet, ho creato lo script `do-nsupdate.sh` che praticamente fa tutto.

Quindi ecco lo script:

```
#!/bin/sh

# aggiornamento del DynDNS di gesero con i dati del client locale
(C) Damon Bonesi, Studio ing. Bonesi, 18.10.2012-21.10.2012
# ultima revisione: 21.10.2012
# ultimo aggiornamento: 20.10.2012 - prima versione

# utilizzo: ./do-nsupdate.sh

# il sito web ip.nelweb.ch ritorna l'IP del router del client
FILE="myip.txt"
wget http://ip.nelweb.ch -q -O $FILE
IP=$(awk '{ print $1 }' $FILE)

# creo il nuovo file di configurazione del DDNS
echo "server gesero.nelweb.ch" > dnsupdate.txt
echo "zone adunanza.ch" >> dnsupdate.txt
echo "update delete adunanza.ch. A" >> dnsupdate.txt
echo "update delete www.adunanza.ch. A" >> dnsupdate.txt
echo "update delete imap.adunanza.ch. A" >> dnsupdate.txt
echo "update add adunanza.ch. 3600 A $IP" >> dnsupdate.txt
echo "update add www.adunanza.ch. 3600 A $IP" >> dnsupdate.txt
echo "update add imap.adunanza.ch. 3600 A $IP" >> dnsupdate.txt

# show lo posso togliere se voglio un output completamente
silenzioso
echo "show" >> dnsupdate.txt
echo "send" >> dnsupdate.txt

# invio la configurazione al DNS
nsupdate -k Kadunanza.ch.+157+49078.private -v dnsupdate.txt
```

Il comando `show` si può togliere se si mette lo script in un cron, l'IP `$IP` è l'IP attualmente configurato dinamicamente sul router di casa, ricavato dal sito `ip.nelweb.ch`, che ho creato appositamente.



Ora il file `Kadunanza.ch.+157+49078.private` lo modifico (o creo) come segue perché quello originale creato sul server non sempre va proprio bene... :

```
key adunanza.ch. {  
    algorithm HMAC-MD5;  
    secret "chiave segreta";  
};
```

La chiave segreta è quella già usata lato server.

E infine si avvia l'update dinamico:

```
./do-nsupdate
```

Ora manca di mettere nel crontab lo script, così che l'update è effettuato dinamicamente. Un aggiornamento ogni 10 minuti è più che sufficiente. Il task è meglio metterlo nel crontab principale e non in quello dell'utente, quindi un `sudo -s` prima del comando `crontab -e` e l'inserimento della riga:

```
*/10 * * * * /home/damon/do-nsupdate.sh
```

La cartella `/home/damon` deve essere cambiata per corrispondere alla cartella dove è depositato lo script.

Ah... la pagina web che mi ritorna l'IP del mio router

Beh lì sono stato pigro ed ho approfittato dei miei server con una soluzione semplice che mi ha preso 5 minuti per metterla in funzione. Di seguito lo script php della pagina web (`index.php`) che mi ritorna l'IP del mio router:

```
<?php  
$ip=$_SERVER['REMOTE_ADDR'];  
echo $ip;  
?>
```

Per chi non ha la possibilità di mettere una paginetta web sul suo server, serve un'altra soluzione, sinceramente non ho neanche pensato ad altre soluzioni, se qualcuno la ha ed ha voglia di consigliarmela può scrivermi (damon@bonesi.ch). Anche lo script penso possa essere migliorato (generare un file solo invece di due), ma son pigro e sono le tre di mattina, quindi come soluzione mi va bene così!

E alla fine non dimenticarsi di configurare il router di casa per instradare le richieste esterne sulla macchina di destinazione (il client) corretta, altrimenti il DDNS va, ma nient'altro!