Chroot-BIND HOWTO

Scott Wunsch, `scott su wunsch.org` v1.5, 1 Dicembre 2001

Questo documento descrive come installare il nameserver BIND 9 in una gabbia creata con chroot ed eseguirlo senza i privilegi di root per aumentare la sicurezza del sistema e minimizzare i potenziali rischi di una sua compromissione. Attenzione, questo documento è stato aggiornato per trattare specificamente BIND 9; se state ancora usando BIND 8, allora dovete leggere il Chroot-BIND8 HOWTO invece di questo. Traduzione di Massimo Soricetti (max-67@TOGLIMI.libero.it) e revisione di Claudio Cattazzo (claudio@TOGLIMI.pluto.linux.it).

Indice

1 Introduzione

Questo è il Chroot-BIND HOWTO; controllate la sezione 1.3 (Dove?) per il sito principale, che contiene la copia più aggiornata disponibile. Si assume che sappiate già usare e configurare BIND (Berkeley Internet Name Domain). In caso contrario vi consiglio di leggere prima il DNS HOWTO. Si assume anche che abbiate una certa familiarità con compilazione ed installazione di software sul vostro sistema UNIX-like.

1.1 Cosa?

Questo documento descrive alcune precauzioni supplementari da prendere quando si installa BIND; spiega come configurarlo in modo che venga eseguito in una gabbia creata con chroot e che quindi non possa accedere a nessun file al di fuori del suo piccolo albero di directory. Inoltre lo configureremo per farlo girare senza i privilegi di root.

L'idea dietro al programma chroot è piuttosto semplice: quando si esegue BIND (o qualunque altro processo) in una gabbia chroot, il processo diventa incapace di vedere la parte di filesystem esterna alla sua gabbia. Per esempio, in questo documento configureremo BIND affinché giri confinato nella directory `/chroot/named`. Bene, a BIND il contenuto di questa directory apparirà come `/`, la directory radice, e non potrà accedere a nient'altro fuori da questa. Probabilmente avete già incontrato gabbie chroot prima d'ora se avete usato `ftp` per accedere ad un sistema pubblico.

Dato che il processo di chroot è molto più semplice con BIND 9 ho iniziato ad ampliare leggermente questo documento aggiungendo alcuni consigli generali su come rendere più sicura un'installazione di BIND. Ciononostante questa non è (e non vuole essere) una guida completa su come rendere sicuro BIND: fare quello che è consigliato qui non esaurisce affatto la lista delle cose da fare per avere un nameserver sicuro!

1.2 Perché?

L'idea che sta alla base dell'usare un ambiente chroot per eseguire BIND è di limitare l'accesso che un possibile intruso può guadagnare sfruttando eventuali vulnerabilità di BIND. È per la stessa ragione per cui eseguiremo BIND senza i privilegi di root.

Questo dovrebbe essere considerato solo un supplemento alle normali precauzioni di sicurezza (utilizzare la versione più recente, controllare gli accessi, ecc...), non certo come alternativa ad esse.

Se vi interessa la sicurezza dei DNS allora probabilmente vi interesserà anche qualche altro documento in proposito. Compilare BIND con *StackGuard* `http://www.immunix.org/products.html#stackguard` è con ogni probabilità una buona idea per avere un ulteriore livello di sicurezza. Usarlo è facile, né più né meno che usare il normale gcc. Inoltre, *DNScache* `http://cr.yp.to/dnscache.html` è un sostituto sicuro per BIND scritto da Dan Bernstein. Dan è anche l'autore di qmail, e DNScache sembra seguire una filosofia simile.

1.3 Dove?

La versione più recente di questo documento è sempre disponibile sul sito web degli Utenti Linux/Open Source di Regina, Sask., su `http://www.losurs.org/docs/howto/Chroot-BIND.html` .

C'è una traduzione in giapponese di questo documento, mantenuta da Nakano Takeo `nakano at apm.seikei.ac.jp`. La si può reperire su `http://www.linux.or.jp/JF/JFdocs/Chroot-BIND-HOWTO.html` .

[NdT: in italiano il documento è reperibile su `http://ildp.pluto.linux.it/HOWTO/Chroot-BIND-HOWTO.html`]

BIND è disponibile presso l' *Internet Software Consortium* `http://www.isc.org/` su `http://www.isc.org/bind.html` . Mentre scrivo, la versione corrente di BIND 9 è la 9.2.0. BIND 9 è in giro da un po' di tempo ormai e molta gente lo usa su server di produzione. Tuttavia, alcuni conservatori preferiscono restare con BIND 8: se siete fra questi allora leggete il mio Chroot-BIND8 HOWTO (disponibile nello stesso sito) per i dettagli sul chroot, ma tenete presente che con BIND 8 è molto più difficile.

Ricordate che ci sono **note** falle di sicurezza in molte delle precedenti versioni di BIND, perciò assicuratevi di avere sempre installata la versione più recente!

1.4 Come?

Ho scritto questo documento basandomi sulla mia esperienza nel configurare BIND in un ambiente chroot. Nel mio caso avevo già un'installazione di BIND preesistente da un pacchetto di una distribuzione Linux. Assumerò che la maggior parte di voi sia presumibilmente nella stessa situazione e perciò descriverò solo come spostare e modificare i file di configurazione dalla vostra installazione esistente di BIND e come rimuovere il pacchetto prima di installare quello nuovo. Non disinstallatelo ora, però: ci possono servire alcuni dei suoi file, prima.

Se invece questo non è il vostro caso, dovreste comunque essere in grado di seguire questo documento: la sola differenza sarà che dove io dico di copiare un file esistente voi dovrete crearlo da soli. A questo scopo vi potrà tornare utile il DNS HOWTO.

1.5 Liberatoria

Questa procedura ha funzionato per me, sul mio computer: il vostro caso potrebbe essere diverso. Questo è solo uno dei tanti modi di fare la stessa cosa; ce ne sono altri che portano allo stesso risultato (sebbene l'approccio generale sia lo stesso). Questo è semplicemente il primo modo che ho trovato per far funzionare la cosa e così ho scritto questo modo e non un altro.

Fino ad oggi la mia esperienza con BIND è stata la sua installazione sui server Linux. Comunque quasi tutte le istruzioni in questo HOWTO dovrebbero funzionare facilmente su altri tipi di UNIX, e cercherò di esporre le differenze di cui sono a conoscenza. Ho anche ricevuto dei suggerimenti da gente che usa altre distribuzioni ed altre piattaforme ed ho cercato di incorporarli qui dove possibile.

Se avete Linux dovete assicurarvi di avere una versione del kernel 2.4 o superiore prima di cominciare. Lo switch `-u` (per eseguire processi senza i privilegi di root) esiste solo da questa versione in poi.

2 Preparazione della gabbia

2.1 Creazione di un utente

Come detto nell'introduzione, è una pessima idea eseguire BIND con i privilegi di root. Quindi, prima di cominciare, creiamo un utente a parte per BIND. Non dovete mai usare a questo scopo un generico utente già presente, come `nobody`. Comunque, alcune distribuzioni come SuSE e Mandrake hanno iniziato a fornire un utente specifico per BIND (di solito chiamato `named`) che potete modificare invece di crearne uno da zero.

Per creare l'utente dovete aggiungere in `/etc/passwd` una riga come questa:

```
named:x:200:200:Nameserver:/chroot/named:/bin/false
```

E per il gruppo un'altra come questa in `/etc/group`:

```
named:x:200:
```

Questo crea un utente e un gruppo per BIND chiamati `named`. Accertatevi che l'UID e il GID (nel nostro esempio entrambi 200) siano unici sul vostro sistema. La shell è impostata a `/bin/false` perché questo utente non avrà mai bisogno di fare il login.

2.2 Struttura delle directory

Ora dobbiamo impostare la struttura delle directory che useremo per la gabbia chroot in cui BIND verrà eseguito. Può essere ovunque nel vostro filesystem, i più paranoici potrebbero anche metterla in una partizione separata. In seguito assumeremo di usare `/chroot/named`. Iniziamo creando questo albero di directory:

```
/chroot
  +-- named
       +-- dev
       +-- etc
       |    +-- namedb
       |         +-- slave
       +-- var
            +-- run
```

Se usate l'utilità GNU `mkdir` (come sui sistemi Linux), potete crearlo con questi comandi:

```
# mkdir -p /chroot/named
# cd /chroot/named
# mkdir -p dev etc/namedb/slave var/run
```

2.3 Sistemare i dati di BIND

Supponendo che abbiate già installato BIND in modo convenzionale e che lo stiate già utilizzando, avrete già un file `named.conf` e i file di zona. Questi file devono essere spostati (o solo copiati, per sicurezza) nella gabbia chroot, in modo che BIND possa accedervi: `named.conf` va in `/chroot/named/etc` e i file di zona possono andare in `/chroot/named/etc/namedb`. Per esempio:

```
# cp -p /etc/named.conf /chroot/named/etc/

# cp -a /var/named/* /chroot/named/etc/namedb/
```

Normalmente BIND necessiterebbe di poter scrivere nella directory `namedb`, ma nell'interesse della sicurezza non glielo permetteremo. Se il vostro nameserver fa da slave per qualche zona dovrà aggiornare quei file di zona, il che significa che dovremo metterli in un'altra directory a cui BIND potrà accedere.

```
# chown -R named:named /chroot/named/etc/namedb/slave
```

Ricordate che dovrete spostare tutti i file delle zone per cui fate da slave in questa directory ed aggiornare il vostro `named.conf`.

BIND avrà anche bisogno di scrivere i suoi pidfile e le informazioni sulle statistiche di uso nella directory `/var/run`, perciò permettiamogli di farlo:

```
# chown named:named /chroot/named/var/run
```

2.4 File per il supporto al sistema

Una volta che BIND è nella sua gabbia chroot non potrà più accedere a **nessun** file fuori da essa. Però avrà bisogno di accedere ad alcuni file fondamentali, anche se non a tanti quanti ne servivano a BIND 8.

Uno dei file che gli serviranno nella sua gabbia è il buon vecchio `/dev/null`. Attenzione che il comando esatto per creare questo device node può variare da sistema a sistema; controllate il vostro script `/dev/MAKEDEV` per sicurezza. Alcuni sistemi potrebbero richiedere anche `/dev/zero`, che può essere creato in modo simile. Si dice che la release candidata BIND 9.2.0 richieda anche `/dev/random`. Per la maggior parte dei sistemi Linux potete usare i seguenti comandi:

```
# mknod /chroot/named/dev/null c 1 3
# mknod /chroot/named/dev/random c 1 8
# chmod 666 /chroot/named/dev/{null,random}
```

Per FreeBSD 4.3 invece:

```
# mknod /chroot/named/dev/null c 2 2
# mknod /chroot/named/dev/random c 2 3
# chmod 666 /chroot/named/dev/{null,random}
```

Vi servirà anche un altro file nella directory `/etc` all'interno della gabbia. Dovete copiare `/etc/localtime` (su alcuni sistemi noto come `/usr/lib/zoneinfo/localtime`) in modo che i log di BIND riportino l'ora esatta degli eventi registrati. Lo potete fare con il seguente comando:

```
# cp /etc/localtime /chroot/named/etc/
```

2.5 Logging

A differenza di un normale avanzo di galera, BIND non può scrivere le sue registrazioni di log sui muri :-). In genere a questo scopo BIND usa `syslogd`, il demone dei log di sistema. Comunque questo tipo di logging è effettuato scrivendo le voci di registrazione nello speciale socket `/dev/log`, che però ora non può usare perché si trova fuori dalla sua gabbia. Per fortuna ci sono un paio di soluzioni a questo problema.

2.5.1 La soluzione ideale

Il modo ideale per risolvere il problema richiede una versione ragionevolmente recente di `syslogd`, che supporti lo switch `-a` introdotto da OpenBSD. Controllate la pagina di manuale `syslogd(8)` per sapere se la versione di syslogd che avete lo supporta o no.

Se sì, tutto quello che dovete fare è aggiungere lo switch `-a /chroot/named/dev/log` alla linea di comando che lancia `syslogd`. Sui sistemi che usano il SysV-init completo (ovvero la maggior parte delle distribuzioni Linux) tale riga si trova solitamente nel file `/etc/rc.d/init.d/syslog`. Per esempio, sul mio sistema Linux Red Hat ho cambiato la riga

```
daemon syslogd -m 0
```

in

```
daemon syslogd -m 0 -a /chroot/named/dev/log
```

È interessante notare come dalla Red Hat 7.2 questo processo sia anche più facile. Ora c'è un file `/etc/sysconfig/syslog` in cui definire parametri supplementari per syslogd.

I sistemi Caldera OpenLinux usano un esecutore di demoni, `ssd`, che legge la configurazione dal file `/etc/sysconfig/daemons/syslog`. In questo caso dovrete soltanto modificare la riga delle opzioni in questo modo:

```
OPTIONS_SYSLOGD="-m 0 -a /chroot/named/dev/log"
```

In modo simile, mi è stato detto che sui sistemi SuSE il posto migliore per aggiungere questa opzione è il file `/etc/rc.config`. Cambiare la riga

```
SYSLOGD_PARAMS=""
```

e metterci

```
SYSLOGD_PARAMS="-a /chroot/named/dev/log"
```

dovrebbe funzionare.

E per ultimo ma non per importanza, con FreeBSD 4.3 sembra che dobbiate soltanto modificare il file `rc.conf` e scrivere:

```
syslogd_flags="-s -l /chroot/named/dev/log"
```

Il `-s` è per motivi di sicurezza e fa parte delle impostazioni predefinite. Il `-l` è una directory locale in cui mettere un altro nodo di log.

Una volta capito come intervenire sul vostro syslogd e scritta la sua nuova configurazione riavviate `syslogd`, o con kill e riavviandolo (con i parametri supplementari), oppure usando gli script SysV-init che lo fanno per voi:

```
# /etc/rc.d/init.d/syslog stop
# /etc/rc.d/init.d/syslog start
```

Appena ripartito syslogd, dovreste vedere un file in `/chroot/named/dev` di nome `log`, che dovrebbe apparire così:

```
srw-rw-rw-   1 root     root            0 Mar 13 20:58 log
```

2.5.2 Le altre soluzioni

Se avete un `syslogd` troppo vecchio dovrete trovare un altro sistema per scrivere i vostri log. Ci sono un paio di programmi in giro, come `holelogd`, che aiutano agendo come proxy ed accettando le voci di log dal BIND in chroot e passandole al vero socket `/dev/log`.

In alternativa potete configurare BIND per fargli scrivere i log su normali file invece di farli passare attraverso syslog. Leggete la documentazione di BIND per scoprire i dettagli su come farlo se scegliete questa strada.

2.6 Restrizione dei permessi

Prima di tutto sentitevi autorizzati a restringere l'accesso all'intera directory `/chroot` al solo utente `root`. Chiaramente non tutti potrebbero volerlo fare, soprattutto se avete altri software installati in quella directory che non apprezzano la cosa.

```
# chown root /chroot
# chmod 700 /chroot
```

Potete anche tranquillamente limitare l'accesso a `/chroot/named` all'utente `named`:

```
# chown named:named /chroot/named
# chmod 700 /chroot/named
```

Per un accesso ancora più ristretto, sui sistemi Linux possiamo rendere alcuni file e directory immutabili, usando l'utilità `chattr` sui filesystem ext2:

```
# cd /chroot/named
# chattr +i etc etc/localtime var
```

In modo equivalente, su FreeBSD 4.3 potreste voler dare un'occhiata a `chflags` se volete rendere file e directory immutabili. Per esempio il comando che segue dovrebbe rendere immutabile tutto il contenuto della directory `/chroot/named/etc`:

```
# chflags schg /chroot/named/etc/*(*).
```

Sarebbe una bella cosa farlo anche per la directory `dev` ma purtroppo questo impedirebbe a `syslogd` di creare il suo socket `dev/log`. Potete impostare il bit immutabile anche per altri file o directory della gabbia chroot, come i file di zona primaria, se sapete che non non cambieranno mai.

3 Compilare ed installare il vostro nuovo BIND

3.1 La compilazione

Compilare BIND 9 per usarlo in una gabbia chroot è un compito molto più facile di quanto fosse con BIND 8. Infatti non dovete fare niente di speciale, il solito `./configure && make` dovrebbe essere sufficiente.

Ricordatevi che se volete abilitare il supporto per l'IPv6 in BIND (`-enable-ipv6`) sui sistemi Linux vi servono versioni allineate del kernel e delle glibc. Se avete un kernel 2.2 vi servono le glibc 2.1 e se avete un kernel 2.4 vi servono le glibc 2.2. BIND è piuttosto suscettibile su questo punto.

4 Installare il vostro nuovo BIND

Devo dire che se avete già un'installazione di BIND (per esempio da un RPM) dovete rimuoverla prima di installare quella nuova. Sui sistemi Red Hat di solito significa rimuovere i pacchetti `bind` e `bind-utils`, e possibilmente `bind-devel` e `caching-nameserver`, se li avete.

Potrebbe essere una buona idea salvare una copia dello script di init (per esempio `/etc/rc.d/init.d/named`) prima di disinstallare; tornerà utile in seguito.

Se state aggiornando BIND da una versione più vecchia, come BIND 8, vi conviene leggere la documentazione di migrazione nel file `doc/misc/migration` nel pacchetto sorgente di BIND. Non mi occupo di nessun argomento che riguardi la migrazione in questo documento; assumerò soltanto che stiate rimpiazzando una preesistente versione funzionante di BIND 9.

4.1 Installare i binari

Questa è la parte facile :-). Semplicemente eseguite `make install` e lasciate fare tutto a lui. Davvero, è tutto qui!

4.2 Configurare lo script di inizializzazione

Se avete già uno script di inizializzazione dalla vostra distribuzione, sarebbe bene modificare quello per eseguire i nuovi binari con le opzioni appropriate. Gli switch sono... *(rullo di tamburi prego...)*

- `-u named`, che dice a BIND di girare come utente `named` invece che come `root`.
- `-t /chroot/named`, che dice a BIND di utilizzare la gabbia chroot che abbiamo creato per lui.
- `-c /etc/named.conf`, che dice a BIND dove trovare i suoi file di configurazione all'interno della gabbia.

Quello che segue è lo script di inizializzazione che uso nel mio sistema Red Hat 6.0. Come potete vedere è quasi esattamente lo stesso file fornito da Red Hat. Non ho ancora provato i comandi `rndc` ma non vedo motivo per cui non debbano funzionare.

```
#!/bin/sh
#
# named           This shell script takes care of starting and stopping
#                 named (BIND DNS server).
#
# chkconfig: 345 55 45
# description: named (BIND) is a Domain Name Server (DNS) \
# that is used to resolve host names to IP addresses.
# probe: true

# Source function library.
. /etc/rc.d/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

[ -f /usr/local/sbin/named ] || exit 0

[ -f /chroot/named/etc/named.conf ] || exit 0

# See how we were called.
case "$1" in
  start)
        # Start daemons.
        echo -n "Starting named: "
        daemon /usr/local/sbin/named -u named -t /chroot/named -c /etc/named.conf
```

```
                echo
                touch /var/lock/subsys/named
                ;;
        stop)
                # Stop daemons.
                echo -n "Shutting down named: "
                killproc named
                rm -f /var/lock/subsys/named
                echo
                ;;
        status)
                status named
                exit $?
                ;;
        restart)
                $0 stop
                $0 start
                exit $?
                ;;
        reload)
                /usr/local/sbin/rndc reload
                exit $?
                ;;
        probe)
                # named knows how to reload intelligently; we don't want linuxconf
                # to offer to restart every time
                /usr/local/sbin/rndc reload >/dev/null 2>&1 || echo start
                exit 0
                ;;

        *)
                echo "Usage: named {start|stop|status|restart|reload}"
                exit 1
esac

exit 0
```

Come con syslogd, dalla Red Hat 7.2 anche questo lavoro è più facile. C'è un file chiamato `/etc/sysconfig/named` in cui specificare parametri supplementari per syslogd. Lo script predefinito `/etc/rc.d/init.d/named` su Red Hat 7.2 però controllerà l'esistenza di `/etc/named.conf` prima di partire. Dovrete correggere questo percorso.

Su sistemi Caldera OpenLinux dovrete soltanto modificare le variabili definite in cima e a quanto sembra lo script si occuperà del resto al vostro posto:

```
NAME=named
DAEMON=/usr/local/sbin/$NAME
OPTIONS="-t /chroot/named -u named -c /etc/named.conf"
```

E per FreeBSD 4.3 potete modificare `rc.conf` ed inserirci queste righe:

```
named_enable="YES"
named_program="chroot/named/bin/named"
named_flags="-u named -t /chroot/named -c /etc/namedb/named.conf"
```

4.3 Modifiche alla configurazione

Dovrete anche aggiungere o cambiare alcune opzioni nel vostro `named.conf` per sistemare le varie directory. In particolare dovrete aggiungere (o cambiare, se le avete già) le seguenti direttive nella sezione `options`:

```
directory "/etc/namedb";
pid-file "/var/run/named.pid";
statistics-file "/var/run/named.stats";
```

Dato che questo file è letto dal demone `named` tutti i percorsi sono naturalmente relativi alla gabbia chroot. Al momento della stesura di questo documento, BIND 9 non supporta molti dei file di statistiche e di dump che aveva prima. Probabilmente tale supporto verrà ripristinato nelle prossime versioni. Se state utilizzando una di queste nuove versioni potreste dover aggiungere voci aggiuntive per far scrivere anche questi nella directory `/var/run`.

5 Fine

5.1 Avviare BIND

Dovrebbe essere tutto a posto e dovreste essere pronti a mettere in azione il vostro nuovo BIND corazzato. Assumendo che abbiate uno script di inizializzazione di tipo SysV, potete semplicemente avviarlo così:

```
# /etc/rc.d/init.d/named start
```

Prima assicuratevi di aver interrotto qualsiasi altra vecchia versione di BIND.

5.2 Ecco fatto!

Potete andare a farvi un sonnellino ora ;-).

6 Appendice - Successivi aggiornamenti di BIND

Ora avete il vostro BIND 9.1.2 tutto in chroot e configurato a puntino come piace a voi... e poi sentite queste fastidiose voci che BIND 9.1.3 è ora disponibile e vi viene voglia di provarlo. Dovete rifare tutto questo lungo processo da capo per provare la nuova versione?

No, infatti avete solo bisogno di compilare il nuovo BIND ed installarlo sopra il vecchio. Dovete soltanto ricordarvi di interrompere la vecchia versione e far ripartire BIND, o sarà in esecuzione ancora la vecchia versione!

7 Appendice - Ringraziamenti

Vorrei ringraziare le seguenti persone per la loro assistenza nella creazione di questo HOWTO:

- Lonny Selinger <`lonny at abyss.za.org`> per aver controllato la prima versione di questo HOWTO ed essersi assicurato che non mancasse nessun passaggio.

- Chirik <chirik at CastleFur.COM>, Dwayne Litzenberger <dlitz at dlitz.net>, Phil Bambridge <phil.b at cableinet.co.uk>, Robert Cole <rcole at metrum-datatape.com>, Colin MacDonald <colinm at telus.net> ed altri per aver segnalato errori ed omissioni ed aver fornito altri utili consigli per rendere questo HOWTO ancora migliore.
- Erik Wallin <erikw at sec.se> e Brian Cervenka <brian at zerobelow.org> per aver fornito buoni suggerimenti sul modo di restringere ulteriormente la gabbia.
- Robert Dalton <support at accesswest.com> per aver suggerito un altro paio di comandi di esempio ed aver segnalato che a BIND 9.2.0 serve `/dev/random`.
- Eric McCormick <hostmaster at cybertime.net> per le informazioni su FreeBSD 4.3.
- Tan Zheng Da <tzd at pobox.com> per i dettagli sulle modifiche introdotte in Red Hat 7.2 che hanno reso le cose un po' più facili.

E alla fine ma certo non meno importante, vorrei ringraziare Nakano Takeo <nakano at apm.seikei.ac.jp> per aver tradotto Chroot-BIND HOWTO in Giapponese. Potete trovare la sua traduzione su `http://www.linux.or.jp/JF/JFdocs/Chroot-BIND-HOWTO.html` .

[NdT: La traduzione italiana è reperibile presso `http://ildp.pluto.linux.it/HOWTO/Chroot-BIND-HOWTO.html`]

8 Appendice - Politica di distribuzione