vsftpd – Una Introduzione al Demone "Very Secure FTP"

by Mario M. Knopf
<netzmeister/at/neo5k/dot/org>

About the author:

Mario si occupa attivamente di Linux, reti di calcolatori e di tematiche legate alla sicurezza informatica. Nel tempo libero cura due pagine web:
neo5k.org e
linuxwallpapers.de.

Abstract:

Questo articolo costituisce una introduzione al demone "Very Secure FTP Daemon". Partendo da una descrizione generale del protocollo FTP e di vsftpd si passa alla procedura di installazione, alla configurazione e alle opzioni di base del demone, il tutto per fornire un sistema di test con le funzionalità di base.

Introduction

Il File Transfer Protocol (FTP) consente il trasferimento dati su internet in modo indipendente dalla piattaforma operativa ed è basato sul paradigma client/server. Nel riferimento ufficiale RFC 959[1] FTP consiste di due canali di comunicazione, uno per il trasporto dati (porta 20/TCP) e l'altro per il controllo (porta 21/TCP). Nel canale di controllo i due attori (server e client) si scambiano i comandi per la inizializzazione della sessione di trasferimento dati
Una sessione FTP si compone di 4 momenti:

- Autenticazione dell'utente
- Istituzione del canale di controllo
- Istituzione del canale dati
- Terminazione della sessione

FTP utilizza TCP (Transmission Control Protocol) per il trasporto e questo assicura che i dati arrivino al destinatario correttamente e in modo affidabile. Non c'è dunque bisogno che sia FTP a gestire l'eventuale perdita di pacchetti o a verificare la loro integrità. In altre parole TCP assicura che ogni pacchetto giunga a destinazione senza ripetizioni, integro e nell'ordine corretto.

Nella trasmissione dati si distinguono tre tipi di trasferimento a seconda che la terminazione del flusso sia marcata con un end−of−file (EOF) oppure come end−of−record (EOR).

- Stream
- Block
- Compressed

A questo si aggiungono due modalità di trasferimento:

- ASCII
- Binary

La modalità ASCII è utilizzata nel trasferimento di files di testo, quella binary (binaria) è usata per trasferire programmi e altri tipi di files. L'utente non ha bisogno di selezionare la modalità di trasferimento dal momento che oramai tutti i client FTP riconoscono il tipo di file e commutano la modalità di conseguenza.

Essendo le credenziali dell'utente, username e password, non cifrate, è estremamente importante considerare questo aspetto come una falla nella sicurezza di FTP. Questa è la ragione dalla quale hanno avuto inizio gli studi sulla sicurezza di FTP. Nell'ottobre 1997 è stata rilasciata la documentazione di riferimento RFC 2228[2] che ha definito le modifiche di sicurezza per FTP

vsftpd

vsftpd è un server per sistemi operativi unix−like, attualmente è disponibile per Linux, BSD, Solaris, HP−UX e IRIX. Offre una varietà di di opzioni che sono per lo più assenti negli altri sever FTP. Le principali sono:

- altissimi standard di sicurezza
- controllo della banda
- scalabilità
- la possibilità di creare utenti virtuali
- supporto IPnG
- performance
- possibilità di assegnare IP virtuali
- velocità

Il nome *vsftpd* sta per "very secure FTP daemon" e la sicurezza è una delle principali caratteristiche che il suo creatore, Chris Evans, ha voluto curare fin dall'inizio dello sviluppo dell'applicazione.

Un esempio eloquente è che *vsftpd* è eseguito in modalità *chroot*, cioè al programma (in questo caso *vsftpd*) è assegnata una versione della directory di root "ridotta" e diversa da quella originaria (l). In questo modo il programma non può in alcun modo accedere a risorse al di fuori di questa particolare directory a lui assegnata: esso è per così dire "confinato" (jailed). Nel caso in cui il server FTP dovesse essere compromesso l'attaccante sarebbe isolato dal resto del sistema limitando così i danni alla sola applicazione attaccata. Ulteriori informazioni sulla tecnica di *chrooting* si possono trovare in [3]. L'articolo [4] è raccomandato a coloro interessati alle specifiche soluzioni di sicurezza adottate in *vsftpd*

Questa varietà di caratteristiche peculiari − tra le quali spiccano quelle rivolte alla sicurezza − fanno di *vsftpd* un server considerevolmente più evoluto degli omologhi server FTP. WU−FTPD[5] può essere qui citato come esempio negativo a causa delle numerose falle di sicurezza rilevate nei passati due anni.

Installazione

L'installazione di *vsftpd* è semplice dal momento che i pacchetti RPM di *vsftpd* completi sono reperibili per le principali distribuzioni, addirittura in molti casi è già installato. In ogni caso i sorgenti si trovano in [6] per chi vuole effettuare una installazione manuale.

Dopo aver scompattato il tarball, vai alla directory creata ed esegui il *make*. Ecco i comandi:

neo5k@phobos> tar xzvf vsftpd−x.x.x.tar.gz
neo5k@phobos> cd vsftpd−x.x.x
neo5k@phobos> make

Prima di questo dovresti controllare se l'utente *"nobody"* e la directory *"/usr/share/empty"* esistono e se necessario crearli. Se si immagina di garantire l'accesso anonimo deve essere creato un utente *"ftp"* con la home *"/var/ftp"*. Questo si ottiene con i seguenti comandi:

neo5k@phobos> mkdir /var/ftp
neo5k@phobos> useradd −d /var/ftp ftp

Per motivi di sicurezza la directory *"/var/ftp"* non deve appartenere all'utente *"ftp"*, nè questo utente deve avere diritto di scrittura in questa directory. Con i due comandi che seguono possiamo cambiare le impostazioni relative alla proprietà e ai privilegi di scrittura di eventuali utenti su questa directory:

neo5k@phobos> chown root.root /var/ftp
neo5k@phobos> chmod og−w /var/ftp

Solo dopo che tutte questi pre−requisiti sono soddisfatti possiamo installare il demone *vsftp*:

neo5k@phobos> make install

Il manuale e il programma dovrebbero ora essere presenti nelle posizioni corrette. Se così non dovesse essere si può eseguire una copia manuale.

neo5k@phobos> cp vsftpd /usr/sbin/vsftpd
neo5k@phobos> cp vsftpd.conf.5 /usr/share/man/man5
neo5k@phobos> cp vsftpd.8 /usr/share/man/man8

Dal momento che il nostro file di configurazione esemplificativo non è stato ancora copiato − per rendere questa introduzione più semplice − dobbiamo creare manualmente la voce come segue:

neo5k@phobos> cp vsftpd.conf /etc

Configurazione

Il file di configurazione si trova in *"/etc/vsftpd.conf"*. Come accade nella maggior parte dei file di configurazione, i commenti sono segnalati da un particolare carattere, nel nostro caso:

Comment line

Una configurazione tipo potrebbe essere la seguente:

 # Accesso FTP anonimo? YES/NO
anonymous_enable=NO

 # Upload anonimo? YES/NO
anon_upload_enable=NO

 # Gli utenti anonimi possono creare directories? YES/NO
anon_mkdir_write_enable=NO

 # Permesso per gli utenti anonimi di effettuare operazioni di renaming or deleting? YES/NO
anon_other_write_enable=NO

 # Log on per gli utenti locali? YES/NO
local_enable=YES

 # Confinamento degli utenti locali alla loro home? YES/NO
chroot_local_user=YES

 # Massimo transfer rate (in bytes al secondo) per gli utenti locali. Default = 0 (unlimited)
local_max_rate=7200

 # Permesso di scrittura? YES/NO
write_enable=YES

 # Abilitazione di messaggi di notifica al cambio directory? YES/NO
dirmessage_enable=YES

 # Welcome banner (banner di benvenuto) da presentare al logon del sistema.
ftpd_banner="Welcome to neo5k's FTP service."

 # Abilitare il logging (tracciamento delle operazioni)? YES/NO
xferlog_enable=YES

 # Logging di tutte le attività FTP? YES/NO
 # Attenzione! Questo può generare grandi quantità di dati.
log_ftp_protocol=NO

 # Connessioni solo dati sulla porta 20 (ftp data). YES/NO
connect_from_port_20=YES

 # Timeout per inattività di una sessione
idle_session_timeout=600

 # Timeout di connessione dati
data_connection_timeout=120

 # Accesso via PAM (Pluggable Authentication Modules)
pam_service_name=vsftpd

 # Modalità operativa standalone? YES/NO − può essere di uno dei tipi (inetd, xinetd, Standalone)

 # In questo esempio il servizio FTP viene avviato con xinetd, il valore è dunque NO.
listen=NO

Avvio del servizio FTP

vsftpd può operare in tre modalità. Una è attraverso *inetd* o *xinetd*, la terza è standalone.

inetd

Se il servizio FTP è avviato con *inetd* editiamo il file di configurazione *"/etc/inetd.conf"* con vi, per esempio:

neo5k@phobos> vi /etc/inetd.conf

Cerchiamo le linee che riguardano il servizio FTP e rimuoviamo il commento davanti a *vsftpd*. Se non c'è tale riga la aggiungiamo. Al termine facciamo ripartire *inetd*. La voce per l'avvio del servizio con *inetd* è questa:

```
# ftp   stream   tcp   nowait   root   /usr/sbin/tcpd   in.ftpd
ftp   stream   tcp   nowait   root   /usr/sbin/tcpd   vsftpd
```

xinetd

E' preferibile avviare *vsftp* con *xinetd* perché è di concezione più recente di *inetd*. Alcune caratteristiche non presenti in *inetd* sono, per esempio, il logging delle richieste, il controllo dell'accesso, l'associazione del servizio a una specifica interfaccia di rete e altro ancora. Una eccellente introduzione a *xinetd* si trova qui [7]. Dopo le modifiche fai ripartire *xinetd*. La configurazione di *xinetd* potrebbe essere la seguente:

```
# vsftp daemon.
service ftp
{
     disable = no
     socket_type = stream
     wait = no
     user = root
     server = /usr/sbin/vsftpd
     per_source = 5
     instances = 200
     no_access = 192.168.1.3
     banner_fail = /etc/vsftpd.busy_banner
     log_on_success += PID HOST DURATION
     log_on_failure += HOST
     nice = 10
}
```

Avvio in modalità Standalone

Esiste anche la possibilità di eseguire *vsftp* in modalità standalone. Per fare questo editiamo *"/etc/vsftpd.conf"* come segue:

 # Shall the vsftp daemon run in standalone operation? YES/NO
listen=YES

Al termine il demone può essere avviato in backgroud come segue

neo5k@phobos> /usr/sbin/vsftpd &

Se il percorso è stato inserito correttamente (cioè fa parte delle variabili d'ambiente condivise dall'utente e dal sistema) il servizio può partire anche così

neo5k@phobos> vsftpd &

Un breve controllo sulla correttezza della variabile d'ambiente è il seguente:

neo5k@phobos> echo $PATH
/usr/sbin:/bin:/usr/bin:/sbin:/usr/X11R6/bin

In modalità standalone dobbiamo, naturalmente, controllare che *vsftp* non è avviato da *inetd* o da *xinetd*.

Test del servizio

Al termine di una corretta installazione e configurazione del server FTP possiamo dunque inaugurarlo col primo avvio.

```
neo5k@phobos> ftp phobos
Connected to phobos
220 "Welcome to neo5k's FTP service."
Name (phobos:neo5k): testuser
331 Please specify the password.
Password:
230 Login successful
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -l
229 Entering Extended Passive Mode
150 Here comes the directory listing
drwxr-xr-x      11  500      100         400  May 07 16:22  docs
drwxr-xr-x       9  500      100         464  Feb 01 23:05  hlds
drwxr-xr-x      39  500      100        4168  May 10 09:15  projects
226 Directory send OK.
ftp>
```

Conclusioni

Come abbiamo avuto modo di osservare *vsftp* non è difficile da installare e da configurare e offre una ricca varietà di opzioni e di caratteristiche di sicurezza.

Questa introduzione fornisce solo un breve sguardo su quanto offre *vsftpd*, dal momento che il server FTP presenta una vasta gamma di possibilità di configurazione. Chi volesse approfondire la conoscenza di *vsftpd* può visitare la home del progetto [6] e accedere alla documentazione completa.

Links

[1] ftp://ftp.rfc−editor.org/in−notes/rfc959.txt [RFC 959 − File Transfer Protocol]
[2] ftp://ftp.rfc−editor.org/in−notes/rfc2228.txt [RFC 2228 − FTP Security Extensions]
[3] linuxfocus.org: article225, January2002 [chroot]
[4] http://vsftpd.beasts.org/DESIGN [Security vsftpd]
[5] http://www.wu−ftpd.org/ [WU−FTPD]
[6] http://www.vsftpd.beasts.org/ [Home of vsftpd]
[7] linuxfocus.org: article 175, November2000 [xinetd]

Translation information:
de −−> −− : Mario M. Knopf <netzmeister/at/neo5k/dot/org>
en −−> it: Davide Lo Vetere <glitch/at/tiscali.it>

2005−01−10, generated by lfparser_pdf version 2.51