Inetd e TCP Wrappers

Daniele Venzano

3 ottobre 2003

Indice

1 Introduzione

C'é una serie di servizi, nati all'alba di Unix, che sono molto comuni su tutte le distribuzioni Linux. Alcuni esempi sono: rdate, rsh, telnet, tftp, talk, ecc. Tutti questi servizi non hanno controlli di filtraggio per definire da quali computer é possibile utilizzarli, infatti non vengono mai eseguiti direttamente ma attraverso inetd che si occupa di eseguire il servizio corretto e passargli la connessione attraverso tcpd, il tcp-wrapper, che controlla che la connessione provenga da un computer autorizzato.

2 Inetd

Inetd (e il suo rivale xinetd di Red Hat e Mandrake) funziona rimanendo in ascolto su tutte le porte dei servizi da lui gestiti[1]. Quando arriva una richiesta su una di tali porte inetd manda in esecuzione il processo destinato a gestire la connessione.
Questo sistema e' adatto a servizi poco utilizzati, infatti fa risparmiare memoria perche' il processo viene eseguito solo per gestire la connessione, e poi termina liberando tutte le risorse occupate. Per contro il costo di esecuzione di un servizio e' abbastanza alto e quindi non e' conveniente utilizzare inetd per un servizio molto usato[2].

2.1 Il file di configurazione /etc/inetd.conf

Il file di configurazione di inetd é composto da voci, ognuna di una riga, che contiene le informazioni relative alla porta e al servizio da associare ad essa. Tutte le righe che cominciano con un # sono ignorate. Ogni voce é composta da diversi campi separati da spazi o tabulazioni. I campi presenti in ciascuna voce sono:

- Nome del servizio (es. pop3)
- Tipo di socket (es. stream)
- Protocollo (es. tcp)
- Attesa della fine del processo[3] (es. nowait)
- Utente (e gruppo opzionale) con cui viene eseguito il processo (es. root)
- Eseguibile del processo server
- Opzioni della riga di comando

2.1.1 Nome del servizio

Questo campo deve essere preso dal file */etc/services* e serve a inetd per capire a quale porta deve mettersi in ascolto.

2.1.2 Tipo di socket

Puo' essere *stream*, *dgram*, *raw*, *rdm* o *seqpacket* a seconda che il processo utilizzi un protocollo con connessione, senza connessione, di basso livello, a messaggi con garanzia di consegna[4], o a pacchetti sequenziali.

[1] Vedi il file /etc/services per un elenco delle porte 'well known' e dei processi ad esse associati
[2] Ad esempio un server http, che deve gestire molte connessioni di breve durata e' poco adatto ad essere gestito attraverso inetd
[3] solo per socket di tipo datagram
[4] rdm = Reliably Delivered Messages

2.1.3 Protocollo

Deve essere un protocollo tra quelli presenti nel file */etc/protocols*

2.1.4 Attesa della fine del processo

E' applicabile solo se il tipo di socket e' dgram (per tutti gli altri deve essere impostato a nowait). Dice a inetd se deve attendere che il processo termini per indicare la fine della comunicazione (e non ascoltare per nuove comunicazioni) oppure se deve considerare ogni pacchetto come una comunicazione separata che deve essere gestita da una nuova istanza del processo.

2.1.5 Utente (e gruppo opzionale)

Questo campo permette di eseguire un processo con permessi piu' deboli di quelli di root, per il gruppo viene utilizzato quello di default presente nel file */etc/passwd* a meno che non ne sia specificato uno.

2.1.6 Eseguibile del processo server

Questo e' il percorso completo dell'eseguibili del processo che deve essere lanciato per gestire la comunicazione. Alcuni servizi come time, echo vengono gestiti direttamente da inetd e in questo campo hanno la parola *internal*.

2.1.7 Opzioni della riga di comando

Qui vanno tutte le opzioni di cui il programma specificato al punto precedente ha bisogno per essere eseguito.

3 TCP Wrappers

I TCP Wrappers sono una suite composta da file di configurazione e da un programma, *tcpd*. In genere *tcpd* viene usato insieme a inetd per controllare la validitaá di una connessione prima che venga caricato il gestore vero e proprio.
Alcuni servizi[5], invece, supportano direttamente il filtraggio tramite i due file */etc/hosts.allow* e */etc/hosts.deny*. Questi servizi funzionano come demoni a sé stante e ogni volta che arriva una connessione nuova la confrontano con le regole lette dai due file di cui sopra per decidere se la connessione debba essere accettata o meno.

3.1 tcpd

tcpd viene chiamato al posto del vero gestore della connessione, passandogli sulla linea di comando l'eseguibile del programma da eseguire nel caso la connessione sia accettata.

[5] Ad esempio OpenSSH

Tcpd utilizza syslog per scrivere una riga di log sul tentativo di connessione e poi compie nell'ordine i seguenti controlli:

1. Controllo di accesso basato sui file *hosts.allow* e *hosts.deny*
2. Controllo del nome del computer remoto tramite lookup DNS
3. Come opzione a *compile time* puó anche fare dei controlli contro lo spoofing solo sulle connessioni TCP.

3.2 *hosts.allow* e *hosts.deny*

Questi file vengono interpretati nell'ordine allow → deny, e la ricerca si ferma alla prima corrispondenza, quindi hosts.allow ha la precedenza su hosts.deny.

3.2.1 Formato

I commenti iniziano con un # e continuano fino alla fine della riga.
Ogni riga contiene una regola di accesso nel formato:

```
lista_di_demoni : lista_di_client
```

lista_di_demoni é una lista nomi di processi e/o wildcards, e *lista_di_client* é una lista di nomi DNS, indirizzi IP e/o wildcards.

3.2.2 Wildcards

Le wildcards servono per definire gruppi di processi o gruppi di client:

- Una stringa che inizia con un . corrisponde a tutti gli host il cui nome finisce con quel dominio. (es. '.lugge.net' corrisponde a 'pippo.lugge.net')
- Una stringa che finisce con un punto corrisponde a tutti gli indirizzi IP che appartengono a quella sottorete (es. '192.168.200.' comprende tutti gli indirizzi da '192.168.200.0' a '192.168.200.255')
- Un'espressione del tipo n.n.n.n/m.m.m.m é interpretata come una coppia rete/maschera (es. 192.168.200.0/255.255.255.0)
- Una stringa che inizia con una / é trattato come il nome di un file che contiene una lista di nome o indirizzi di client.
- I caratteri * e ? possono essere usati, ma non insieme ad una delle regole descritte sopra.

Inoltre esistono delle parole chiave che descrivono dei gruppi predefiniti:

ALL Corrisponde sempre, puó essere utile, ad esempio, in fondo a hosts.deny per negare l'accesso a tutti i client che non appartegano a una delle categorie descritte prima.

LOCAL Corrisponde a ogni client il cui nome non contiene punti, cioé che appartiene allo stesso dominio.

UNKNOWN Corrisponde a tutti gli host i cui nomi non possono essere tradotti in indirizzi IP.

KNOWN Il contrario della regola di sopra.

PARANOID Corrisponde a tutti client i cui nomi non corrispondono all'indirizzo IP usato per la connessione.

4 Dove trovare altre informazioni

Le *man pages* dei comandi descritti qui sono sicuramente delle fonti molto ricche, inoltre anche il comando *info* può dare utili consigli.
La *man page* di hosts_access nella sezione 5 contiene anche descrizioni di funzioni aggiuntive di tcpd che sono state evitate in questa breve introduzione.

5 Copyright e altro

L'ultima versione del documento é disponibile presso:
http://teg.homeunix.org/documentation.html.