Cfengine attraverso la rete

Cfengine consente anche un utilizzo attraverso la rete, per mezzo del demone '**cfd**', che viene avviato nell'elaboratore che offre il proprio servizio.

L'utilizzo più semplice di questa possibilità di Cfengine sta nella copia di file attraverso la rete, per sincronizzare gli elaboratori clienti. Per la precisione, questo particolare è l'unica cosa che viene mostrata qui, in questo capitolo.

Configurazione e avvio del demone

Il servizio relativo al demone '**cfd**' prevede l'accesso alla porta TCP 5308, che pertanto non è privilegiata e consente l'avvio del demone anche senza i privilegi dell'utente '**root**', se questo può essere utile per qualche motivo. Nel file '`/etc/services`' dovrebbe esserci pertanto una riga simile a quella seguente:

```
cfengine        5308/tcp
```

Per funzionare, il demone '**cfd**' richiede la presenza del file '`cfd.conf`', nella directory corrente nel momento dell'avvio del demone, che ha una struttura simile a quella di '`cfengine.conf`'.

Oltre a questo file essenziale, occorre tenere presente che il demone tiene in considerazione anche il contenuto dei file '`/etc/hosts.allow`' e '`/etc/hosts.deny`', per controllare gli accessi.

Una volta predisposto il sistema di configurazione, basta avviare il demone '**cfd**', con i privilegi dell'utente '**root**', se necessario, oppure con i privilegi di un utente comune.

```
# cfd [Invio]
```

Alcune opzioni del demone '**cfd**' sono molto utili per consentire l'analisi del file di configurazione e per poter tenere sotto controllo ciò che avviene effettivamente durante la connessione. Queste opzioni sono riepilogate nella tabella u58.2.

Tabella u58.2. Elenco delle opzioni essenziali di '**cfd**'.

Opzione	Descrizione
`-h` `--help`	Elenca brevemente le opzioni disponibili.
`-d` `--debug`	Rimane in primo piano e mostra ciò che accade.
`-v` `--verbose`	Mostra informazioni dettagliate.
`-p` `--parse-only`	Si limita a scandire il file di configurazione.

Può essere interessante il controllo della configurazione attraverso l'opzione '**-p**', unita opportunamente all'opzione '**-v**'. Inoltre, per verificare le connessioni, soprattutto alla ricerca delle motivazioni per cui qualcosa non funziona come si vorrebbe, conviene utilizzare l'opzione '**-d**', sempre in combinazione con '**-v**'.

```
# cfd -d -v [Invio]
```

Configurazione essenziale di «cfd.conf»

Il file '`cfd.conf`' ha una vaga somiglianza con il file di configurazione di un cliente normale di Cfengine. In particolare, ci sono le sezioni e possono essere presenti le classi, solo che hanno va-

lore esclusivamente nei confronti dell'elaboratore in cui si trova a funzionare il demone.

Generalmente, è probabile che non si faccia uso di classi in un file '`cfg.conf`' e qui non si mostrano esempi in tal senso. La sintassi semplificata ed essenziale di questo file, viene mostrata dal modello seguente. Si tenga presente che non vengono mostrate tutte le direttive, ma solo quelle che devono essere conosciute necessariamente.

```
control:
    [domain = ( dominio )]
    [maxconnections = ( numero_massimo_di_connessioni_indipendenti )
]
    [allowconnectionsfrom = ( numero_ip [numero_ip]… )]
    [denyconnectionsfrom = ( numero_ip [numero_ip]… )]
    [allowmultipleconnectionsfrom = ( numero_ip [numero_ip]
… )]
    [logallconnections = ( true|false )]

admit:|grant:
    file_o_directory            nodi_indicati_con_caratteri_jolly
    …

deny:
    file_o_directory            nodi_indicati_con_caratteri_jolly
    …
```

Si può osservare la presenza di una sezione di controllo, simile a quella dei clienti Cfengine. Questa sezione può anche risultare vuota.

Le sezioni '**`admit`**' (o '**`grant`**') e '**`deny`**', permettono di stabilire l'accessibilità di file e di directory, a degli elaboratori identificati per nome, anche in modo parziale attraverso caratteri jolly. Si intende che la sezione '**`admit`**' o '**`grant`**' serva a elencare i file e le directory accessibili, mentre la sezione '**`deny`**' serve a escludere successivamente parte di quanto precedentemente concesso.

Nella sezione di controllo, le direttive '**`maxconnections`**', '**`allowconnectionsfrom`**', '**`denyconnectionsfrom`**' e '**`allowmultipleconnectionsfrom`**', limitano o concedono gli accessi attraverso l'indicazione di un elenco di indirizzi IP. In generale, questo può essere un mezzo ulteriore di controllo di sicurezza per gli accessi, dal momento che spesso è sufficiente l'uso delle sezioni '**`admit`**' e '**`deny`**'. In particolare, ogni cliente Cfengine che accede, ha la possibilità di aprire una sola connessione, mentre con la direttiva '**`allowmultipleconnectionsfrom`**' è possibile autorizzare un accesso multiplo agli indirizzi indicati.

L'uso delle altre direttive indicato dovrebbe essere intuitivo; inoltre, nella sezione di controllo è possibile dichiarare delle variabili, nello stesso modo della configurazione dei clienti Cfengine.

È importante ricordare che i percorsi di cui si concede l'accesso, devono essere reali, perché i collegamenti simbolici non vengono presi in considerazione. Questo tipo di errore lo si può individuare utilizzando l'opzione '**`-d`**' quando si avvia '**`cfd`**'.

A titolo di esempio viene mostrato un caso molto semplice di configurazione, in cui si concede l'accesso alle directory '`/usr/local/file_pubblici1/`' e '`/usr/local/file_pubblici2/`', creando appositamente due variabili per semplificarne l'indicazione; inoltre si concede l'accesso anche ai file '`/etc/passwd`' e '`/etc/group`'. Per la precisione, la directory '`/usr/local/file_pubblici1/`' risulta accessibile a tutti, mentre '`/usr/local/file_pubblici2/`' è accessibile solo ai domini **.brot.dg* e **.mehl.dg*; inoltre, i due file '`/etc/passwd`' e '`/etc/group`' sono accessibili esclusivamente dal dominio **.brot.dg*. Infine, per qualche motivo, si esclude l'accesso alla directory '`/usr/local/file_pubblici2/particolare/`' al dominio **.mehl.dg*. Ogni cliente può aprire una sola connessione e sono consentiti un massimo di 10 accessi simultanei.

```
control:
    pubblici1 = ( /usr/local/file_pubblici1 )
    pubblici2 = ( /usr/local/file_pubblici2 )
    maxconnections = ( 10 )

admit:
    $(pubblici1) *
    $(pubblici2) *.brot.dg *.mehl.dg
    /etc/passwd  *.brot.dg
    /etc/group   *.brot.dg

deny:
    $(pubblici2)/particolare *.mehl.dg
```

Filosofia del sistema di distribuzione di Cfengine

È il caso di osservare che, contrariamente a Rsync, il cliente Cfengine contatta il servente per ottenere qualcosa e non per inviare lì un file.

Quando la trasmissione di un file è sottoposta al confronto di un codice di controllo, è il cliente Cfengine che invia il suo codice di controllo al servente, il quale verifica la necessità o meno di trasmettere il file aggiornato.