

Installare e configurare BackupPC

Il programma **BackupPC** è un sistema di backup scritto in **Perl** che presenta una buona flessibilità e potenza insieme ad una discreta facilità d'uso sul lato utente grazie alla sua interfaccia web. BackupPC è utile in tutti quei casi in cui sia necessario gestire backup di realtà non troppo complesse per le quali sia sufficiente un salvataggio su disco (esterno o rimuovibile).

Qualora si debbano realizzare strategie di backup più complesse, in particolare con salvataggi su diversi supporti, può convenire utilizzare al suo posto un prodotto più sofisticato (ma anche molto più complicato da usare) come Bareos/Bacula. In questo caso lo svantaggio principale è costituito dalla complessità dell'interfaccia utente per il ripristino e la necessità di installare degli agent sulle macchine di cui si effettuano i salvataggi.

Un grande vantaggio di BackupPC infatti è che non è necessario installare nessun programma sui PC di cui si effettua il backup, in quanto tutto viene eseguito tramite l'uso di **rsync** o **tar** (via SSH) per le macchine unix e del protocollo SMB (o CIFS) per le macchine windows.

Installazione e configurazione

All'interno di **Debian** è disponibile la versione pacchettizzata di BackupPC (la 3.3.0 su **Jessie**), che può essere installata semplicemente con il comando:

```
apt-get install backuppc
```

Per poter utilizzare il supporto dei backup su unix via **rsync** oltre a quest'ultimo è comunque necessario installarsi anche i rispettivi pacchetti Perl, senza i quali il sistema non funziona; inoltre possono risultare utili altri pacchetti, in particolare programmi per la decompressione/compressione degli archivi e per l'accesso agli share Windows tramite il protocollo SMB; per cui alla fine sarà opportuno installare anche:

```
apt-get install libfile-rsync-perl libfile-rsync-perl rsync \
par2 bzip2 smbfs
```

All'installazione del pacchetto l'interfaccia web viene configurata per l'accesso autenticato generando il file **htpasswd** dentro **/etc/backuppc/** che viene referenziato direttamente dalla relativa configurazione di Apache. Questa è contenuta nel file **apache.conf** della stessa directory, a cui doveva essere creato un link simbolico in **/etc/apache2/conf.d/**, con Jessie tutto ciò non è più necessario ed il relativo link viene automaticamente creato in **/etc/apache2/conf-available/** ed abilitato su **/etc/apache2/conf-enabled/**.

L'installazione di default consente l'accesso all'interfaccia web di gestione anche in chiaro, ma il file di configurazione prevede, commentato, l'uso della direttiva **SSLRequireSSL**, che si suggerisce di abilitare sempre. Se si dispone di un sistema di gestione centralizzata degli utenti su LDAP questo può essere utilizzato usando la seguente configurazione (per i dettagli sulle direttive si veda [Apache22DavLdap](#)):

Installare e configurare BackupPC

```
Alias /backuppc /usr/share/backuppc/cgi-bin/

<Directory /usr/share/backuppc/cgi-bin/>
    AllowOverride None
    Allow from all

    SSLRequireSSL

    Options ExecCGI FollowSymlinks
    AddHandler cgi-script .cgi
    DirectoryIndex index.cgi

    AuthType basic
    AuthName "BackupPC admin"
    AuthBasicProvider ldap
    AuthzLDAPAuthoritative off
    AuthLDAPURL ldap://127.0.0.1/ou=People,dc=truelite,dc=it
    require valid-user
</Directory>
```

Il programma mantiene i dati del backup in `/var/lib/backuppc`, questo significa che si deve avere spazio sufficiente sul filesystem di `/var` per i backup. Se si desidera allocare lo spazio su una partizione separata si deve spostare la directory sunnominata nella destinazione voluta, inserendo al suo posto un link simbolico verso la nuova collocazione.

Il passo successivo è la configurazione del programma, questa è mantenuta, insieme a tutti gli altri file, in `/etc/backuppc`, il file principale è `config.pl`, che contiene la definizione di una serie di variabili Perl, con le quali viene controllato il comportamento del programma. Il file è ben commentato, e le variabili sono numerosissime, per cui ci concentreremo solo sul sottoinsieme di quelle utilizzate nei vari file di configurazione relativi alle singole macchine.

Il file `config.pl` contiene infatti i valori di default, usati in maniera generica quando non ve ne sono indicati di specifici, il sistema infatti richiede che si indichi nel file `hosts` l'elenco delle macchine di cui si vuole effettuare il backup (in genere usando il relativo hostname) e poi si inseriscano le chiavi di configurazione specifiche da applicare per ciascuna di esse in un corrispondente file `hostname.pl`. Il pacchetto Debian ad esempio inserisce un `localhost.pl` per il backup locale delle configurazioni.

Per quanto riguarda `config.pl` la direttiva probabilmente più significativa è `FullKeepCnt` che indica quanti backup completi mantenere. Il default è uno, che indica uno solo per settimana ma si può richiedere un periodo più lungo specificando un array di valori, questo ha un significato complicato in cui ogni numero successivo al primo indica il numero di backup completi da mantenere per il successivo multiplo di due settimane, ad esempio indicando:

```
$Conf{FullKeepCnt} = [4, 0, 12];
```

si richiedono quattro copie dei backup completi a cadenza settimanale, nessuna copia per i backup completi a cadenza bisettimanale e 12 copie dei backup completi a cadenza circa mensile (ad esser precisi quadrisettimanale).

Installare e configurare BackupPC

Il file `hosts` ha un formato diviso in quattro colonne, la prima colonna indica il nome della macchina, questo deve poter essere risolto direttamente (si fa riferimento o ad un hostname nel proprio dominio, o a un nome di una macchina Windows risolvibile via netbios). La seconda colonna indica se deve essere fatta o meno una ricerca netbios sul range fornito dal DHCP, ed in genere deve restare impostata a zero. Il terzo campo indica l'utente (locale) per conto del quale viene eseguito il backup (usato pure per l'accesso all'interfaccia web). Se ne possono specificare altri, sempre per username, in un elenco separato da virgole nella quarta ed ultima colonna. Un esempio di questo file potrebbe essere il seguente:

```
...
host      dhcp      user      moreUsers  # <--- do not edit this line
...
localhost 0         backuppc
client    0         backuppc
```

Utilizzando per `hosts` l'esempio precedente le configurazioni specifiche per la macchina `client` dovranno essere inserite nel file `client.pl`. Come accennato, si tratterà di modificare solo le configurazioni attinenti alla tipologia di backup da eseguire, in questo caso la prima variabile utilizzata è `XferMethod`, che indica come eseguire il backup e può assumere quattro valori; quelli più usati sono comunque `rsync` per il backup attraverso `rsync` e `tar` per il backup attraverso l'omonimo comando. Il primo è più efficiente per i trasferimenti via rete, ma quando si deve eseguire un backup di moltissimi file, comporta un grande consumo di memoria per cui spesso si rivela troppo lento e problematico per la macchina ospite.

Dopo aver indicato il metodo da usare occorre indicare di quali directory si vuole eseguire il backup, in tal caso le variabili di controllo sono due, `TarShareName` se si usa il metodo `tar` e `RsyncShareName` se si usa `rsync`, entrambe prendono come valore un array di pathnames. Se allora si intende usare il comando `tar` il file di configurazione del nostro `client.pl` sarà qualcosa del tipo:

```
$Conf{XferMethod} = 'tar';
$Conf{TarShareName} = [ '/etc', '/var', '/home', '/root' ];
$Conf{BackupFilesExclude} = [ '/var/cache', '/var/run' ];
```

Si tenga presente comunque che tutti questi file possono essere anche creati e modificati direttamente dall'interfaccia web (motivo per cui i suddetti file appartengono all'utente `backuppc` ed hanno come gruppo `www-data`). Se li si creano manualmente si abbia cura, se si vuole poterli modificare in seguito via web, di impostarne correttamente utente e gruppo proprietario.

Per poter eseguire i backup, sia che si usi il metodo `rsync` sia che si usi il metodo `tar` comunque si deve passare attraverso SSH, per questo sarà necessario installare una chiave per l'accesso alle macchine remote di cui si vuole fare il backup, questo richiede che si generi sul server una coppia di chiavi per l'utente `backuppc` (o per l'utente col quale si vuole effettuare il backup come indicato in `hosts`) per farlo si potranno usare i comandi:

```
su - backuppc
ssh-keygen
```

avendo cura di mettere una password vuota per la chiave.

Installare e configurare BackupPC

Per poter effettuare il collegamento si dovrà poi copiare sulle macchine di cui si vuole fare il backup la suddetta chiave nel file `.ssh/authorized_keys` nella home dell'utente remoto usato per il backup (in una configurazione elementare `root` per avere accesso a tutti i file). Si tenga presente che se si usa il metodo `rsync` detto programma deve essere installato anche sulla macchina di cui si vuole effettuare il backup.

Ci si ricordi inoltre di effettuare una connessione di prova verso ciascuna macchina remota, per accettare la chiave del server SSH della stessa; questa operazione dovrà essere eseguita manualmente una volta per tutte le macchine a cui ci si deve collegare, in modo da generare la voce di `.ssh/known_hosts` che le identifica come server noti, altrimenti si otterrà un fallimento dei backup per l'impossibilità di eseguire l'accettazione in modalità non interattiva.

Impostazioni per la sicurezza dell'accesso ai client

Dato che la compromissione del server con BackupPC comporterebbe anche la possibilità di ottenere la chiave privata dell'utente `backuppc`, un accesso indiscriminato di quest'ultimo ai dati dei client è da evitare. Per questo esistono diversi approcci, miranti a ridurre le possibilità di abuso, ed evitare che una compromissione del server di backup dia accesso completo anche alle macchine di cui si fanno i backup.

Il primo approccio è quello in cui si cerca di evitare l'uso di `root` come utente sulla macchina di cui si vuole effettuare il backup, quello che serve infatti è soltanto poter eseguire il comando di backup con privilegi di amministratore, per questo la procedura più corretta è creare su questa un utente non privilegiato da usare per i backup e ricorrere a `sudo` per consentire a questo l'esecuzione del comando necessario.

In questo caso si può creare anche sulla macchina di cui eseguire il backup un utente `backuppc`, poi si dovrà copiare nella `.ssh/authorized_keys` della sua home la chiave pubblica precedentemente creata, ed infine configurare `sudo`. In questo caso, posto che si intenda utilizzare per il backup il metodo `tar`, occorrerà aggiungere a `/etc/sudoers` una riga:

```
backuppc ALL=NOPASSWD: /bin/tar -c *
```

che consente solo la creazione di archivi. Occorrerà poi modificare la variabile `TarClientCmd` nella configurazione di BackupPC aggiungendo a `client.pl` qualcosa del tipo:

```
$Conf{TarClientCmd} = '$sshPath -q -x -n -l backuppc $host'
. ' env LC_ALL=C /usr/bin/sudo $starPath -c -v -f
- -C $shareName+'
. ' --totals';
```

ci si ricordi inoltre eseguire almeno una volta la prova di funzionamento del comando con qualcosa del tipo:

```
su backuppc
/usr/bin/ssh -q -x -n -l backuppc client env LC_ALL=C /usr/bin/sudo
/bin/tar -c -v -f - -C /etc .
```

Un secondo approccio, più sicuro, ma utilizzabile però soltanto con il metodo `rsync`, è quello che si appoggia allo script `rrsync` distribuito insieme al programma `rsync`, che consente di restringere l'utilizzo di quest'ultimo in modo da dare accesso in sola lettura (eventualmente ad una singola directory). Lo script è distribuito in `/usr/share/doc/rsync/scripts/rrsync.gz` con il pacchetto di `rsync`, ma la versione di Jessie

Installare e configurare BackupPC

manca dell'abilitazione di alcune opzioni necessarie per BackupPC (vedi <https://lists.samba.org/archive/rsync/2014-September/029647.html>), il patch citato è stato applicato alla versione allegata a questa pagina.

In questo caso non è necessario usare `sudo`, in quanto `rrsync` è stato scritto per essere usato come comando di accesso per SSH. Si tratta cioè di appoggiarsi ad una funzionalità presente nell'autenticazione a chiavi di SSH, che consente di specificare, in testa alla linea di `.ssh/authorized_keys` che fornisce accesso con la chiave di un utente, una serie di restrizioni (per i dettagli si consulti la pagina di manuale di `sshd`).

Fra queste restrizioni c'è la possibilità di indicare un comando che verrà utilizzato quando viene effettuato l'accesso, al posto di quello inviato nella riga di comando di `ssh` (il quale sarà passato nella variabile di ambiente `SSH_ORIGINAL_COMMAND`). In questo modo non si avrà mai un accesso alla shell, in quanto `rrsync` consente solo di eseguire `rsync` (neanche `/usr/bin/rsync`, pertanto occorrerà impostare per la macchina la variabile di configurazione `$Conf{RsyncClientPath} = 'rsync'`). Lo script inoltre prende come argomento una directory, alla quale viene ristretto l'accesso, e l'opzione facoltativa `-ro`, che blocca l'accesso in sola lettura.

Per poter utilizzare questo secondo approccio, a parte la modifica alla configurazione citata, tutto quello che serve è preporre alla chiave pubblica di `backuppc` inserita in `.ssh/authorized_keys` la seguente stringa:

```
command="nice -n 19 ionice -c 3 /usr/local/bin/rrsync -ro /",no-agent-forwarding,no-port-forwarding,no-pty,no-user-rc,no-X11-forwarding
```

(cui far seguire, separato da uno spazio, il contenuto della chiave pubblica).

In questo esempio, invece di invocare direttamente lo script `rrsync`, si è preferito un passaggio preliminare da `nice` e `ionice` per ridurre il carico dell'esecuzione del backup sulla macchina, e poi con `rrsync` si è dato accesso a qualunque file in sola lettura. Le ulteriori opzioni indicate consentono di eliminare tutte le funzionalità aggiuntive di SSH che potrebbero essere usate in caso di compromissione della chiave ma che non servono per l'esecuzione di un backup.

Si tenga presente che entrambi gli approcci non consentono a BackupPC di scrivere sulla macchina remota di cui esegue il backup, pertanto non sarà possibile utilizzare la funzionalità di ripristino direttamente sulla destinazione fornita dalla piattaforma. Nel caso la perdita di questa funzionalità comporti un aggravio amministrativo non giustificato da un rischio di compromissione ritenuto sufficientemente ridotto, nella seconda delle due ipotesi si può riabilitare la scrittura rimuovendo l'opzione `-ro` dagli argomenti di `rrsync`.



This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.