Push-Backup mit Duply

Mit duply/duplicity kann man per GPG-verschlüsselte Backups z.B. per FTPS auf einen Backupserver erstellen. Duply agiert als Skript zur einfacheren Verwendung von duplicity.

Installation

duply/duplicity

Installation

- ***emerge app-backup/duply***

Zusätzliche Programme

Je nachdem, welches Backend verwendet werden soll werden folgende Pakete benötigt (nicht vollständige Liste):

- FTPS: net-ftp/lftp [gnutls/ssl]
- FTP: net-ftp/ncftp

Konfiguration

GPG-Schlüssel zum Signieren und Verschlüsseln

Hinweis: Wenn die Passworteingabe aufgrund mangelnder Rechte an der Konsole nicht funktioniert (command get_passphrase failed: Operation cancelled), dann kann man mittels

- ***chmod o+rw $(tty)***

sich kurzzeitig die Rechte für die Konsole verschaffen. Nach Generierung der Schlüssen muss dies mit

- ***chmod o-rw $(tty)***

unbedingt rückgängig gemacht werden!

Schlüsselpaar zum Signieren erstellen

- ***gpg --gen-key***
  - Your selection? ***3*** (DSA, sign only)
  - What keysize do you want? (2048) ***2048***

- Key is valid for? (0) ***0***
- Is this correct? (y/N) ***y***
- Real name: ***server1.example.org*** (Server-Name)
- Email address: ***hostmaster@example.org*** (E-Mail-Adresse)
- Comment: ***Backup-GPG-Sign-Key***
- Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? ***O***
- ***2x ausgewürfeltes Passwort***

Ausgabe

```
gpg (GnuPG) 2.0.17; Copyright (C) 2011 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Please select what kind of key you want:
   (1) RSA and RSA (default)
   (2) DSA and Elgamal
   (3) DSA (sign only)
   (4) RSA (sign only)
Your selection? 3
DSA keys may be between 1024 and 3072 bits long.
What keysize do you want? (2048) 2048
Requested keysize is 2048 bits
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0) 0
Key does not expire at all
Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.

Real name: server1.example.org
Email address: hostmaster@example.org
Comment: Backup-GPG-Sign-Key
You selected this USER-ID:
    "server1.example.org (Backup-GPG-Sign-Key) <hostmaster@example.org>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
You need a Passphrase to protect your secret key.

+-----------------------------------------------------+

| Enter passphrase                                    |
|                                                     |
|                                                     |
```

```
| Passphrase _________________________________________ |
|                                                      |
|           <OK>                    <Cancel>           |
+------------------------------------------------------+

We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: WARNING: some OpenPGP programs can't handle a DSA key with this digest
size
gpg: key 11ED50F4 marked as ultimately trusted
public and secret key created and signed.

gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:   2  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 2u
pub   2048D/11ED50F4 2012-03-26
      Key fingerprint = E8BD B9BB E8B0 63A5 DC54  1CDB A9FA 125E 11ED 50F4
uid                  server1.example.org (Backup-GPG-Sign-Key)
<hostmaster@example.org>

Note that this key cannot be used for encryption.  You may want to use
the command "--edit-key" to generate a subkey for this purpose.
```

Die benötigte Key-ID lautet **11ED50F4**.

Schlüsselpaar zum Signieren exportieren

- ***gpg --armor --export-secret-key 11ED50F4 >> signkey.sec***
- ***gpg --armor --export 11ED50F4 >> signkey.pub***

Schlüsselpaar zum Verschlüsseln erstellen

- ***gpg --gen-key***
  - Your selection? ***1***
  - What keysize do you want? (2048) ***2048***
  - Key is valid for? (0) ***0***
  - Is this correct? (y/N) ***y***
  - Real name: ***server1.example.org*** (Server-Name)
  - Email address: ***hostmaster@example.org*** (E-Mail-Adresse)
  - Comment: ***Backup-GPG-Key***
  - Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? ***O***
  - ***2x ausgewürfeltes Passwort***

Ausgabe

```
gpg (GnuPG) 2.0.17; Copyright (C) 2011 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: directory `/root/.gnupg' created
gpg: new configuration file `/root/.gnupg/gpg.conf' created
gpg: WARNING: options in `/root/.gnupg/gpg.conf' are not yet active during
this run
gpg: keyring `/root/.gnupg/secring.gpg' created
gpg: keyring `/root/.gnupg/pubring.gpg' created
Please select what kind of key you want:
   (1) RSA and RSA (default)
   (2) DSA and Elgamal
   (3) DSA (sign only)
   (4) RSA (sign only)
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048) 2048
Requested keysize is 2048 bits
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0) 0
Key does not expire at all
Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.

Real name: server1.example.org
Email address: hostmaster@example.org
Comment: Backup-GPG-Key
You selected this USER-ID:
    "server1.example.org (Backup-GPG-Key) <hostmaster@example.org>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
You need a Passphrase to protect your secret key.

+-----------------------------------------------------+

| Enter passphrase                                    |
|                                                     |
|                                                     |
| Passphrase _____________________________________    |
|                                                     |
|           <OK>                    <Cancel>          |
+-----------------------------------------------------+

We need to generate a lot of random bytes. It is a good idea to perform
```

```
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.

gpg: /root/.gnupg/trustdb.gpg: trustdb created
gpg: key 34FE3252 marked as ultimately trusted
public and secret key created and signed.

gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
pub   2048R/34FE3252 2012-03-26
      Key fingerprint = 70BC C7E5 1F27 0AC4 0CF5  AA6D 6993 013A 34FE 3252
uid                  server1.example.org (Backup-GPG-Key)
<hostmaster@example.org>
sub   2048R/87A0373D 2012-03-26
```

Die benötigte Key-ID lautet **34FE3252**.

Schlüsselpaar zum Verschlüsseln exportieren und privaten Schlüssel löschen

- ***gpg --armor --export-secret-key 34FE3252 >> encryptkey.sec***
- ***gpg --armor --export 34FE3252 >> encryptkey.pub***
- ***gpg --delete-secret-key 34FE3252***
  - Delete this key from the keyring? (y/N) ***y***
  - This is a secret key! - really delete? (y/N) ***y***

Ausgabe

```
gpg (GnuPG) 2.0.17; Copyright (C) 2011 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.


sec  2048R/34FE3252 2012-03-26 server1.example.org (Backup-GPG-Key)
<hostmaster@example.org>

Delete this key from the keyring? (y/N) y
This is a secret key! - really delete? (y/N) y
```

WICHTIG: Das entfernen des privaten Schlüssels muss vor der ersten Benutzung von duply/duplicity vorgenommen werden, da dieser sonst exportiert wird. Das asymmetrisch verschlüsselte inkrementelle Backup funktioniert auch ohne privaten Schlüssel, da duplicity die Meta-Daten unverschlüsselt im Verzeichnis ~/.cache vorhält. Wenn dieses Verzeichnis entfernt wird wird der private Schlüssel benötigt um neue Backups (Inkrementell/Voll) erstellen oder Status-Informationen ermitteln zu können! Der Signier-Schlüssel wird ausschließlich zum Signieren verwendet - prinzipbedingt ist es notwendig, den privaten Signierschlüssel im Schlüsselbund des Servers zu belassen.

Privaten "GPG-Schlüssel zum Verschlüsseln" bei Bedarf importieren

- ***gpg --import <keyfile>***

Ausgabe

```
gpg: key 34FE3252: secret key imported
gpg: key 34FE3252: "server1.example.org (Backup-GPG-Key)
<hostmaster@example.org>" not changed
gpg: Total number processed: 1
gpg:              unchanged: 1
gpg:       secret keys read: 1
gpg:   secret keys imported: 1
```

HINWEIS: Nach Nutzung des privaten Schlüssels sollte dieser wieder gelöscht werden (siehe oben). Zusätzlich muss die Datei .duply/server1-backup/gpgkey.34FE3252.sec.asc gelöscht und der Wert GPG_PW in der Datei .duply/server1-backup/conf entfernt werden!

Duply-Konfiguration

- Initiale Konfiguration erstellen
  - ***duply server1-backup create*** (Backup-Bezeichnung als 2. Parameter, hier: server1-backup)

Ausgabe

```
Congratulations. You just created the profile 'server1-backup'.
The initial config file has been created as
'/root/.duply/server1-backup/conf'.
You should now adjust this config file to your needs.

IMPORTANT:
  Copy the _whole_ profile folder after the first backup to a safe place.
  It contains everything needed to restore your backups. You will need
  it if you have to restore the backup from another system (e.g. after a
  system crash). Keep access to these files restricted as they contain
  _all_ informations (gpg data, ftp data) to access and modify your backups.

  Repeat this step after _all_ configuration changes. Some configuration
  options are crucial for restoration.
)
```

Das Passwort für den privaten Schlüssel (GPG_PW) ist nur nötig, wenn das Backup zurückgespielt werden muss (Restore) oder das lokale Cache-Verzeichnis verloren gegangen ist. Der private Schlüssel muss zuvor wieder importiert werden.

.duply/server1-backup/conf

```
#GPG_KEY='_KEY_ID_'
GPG_PW=''

GPG_KEYS_ENC='34FE3252'
GPG_KEY_SIGN='11ED50F4'
..
GPG_PW_SIGN='idXuXEMddi9xAPepktqEJ7hsbad-NG_g'

...
TARGET='sftp://benutzer@backupserver1.example.org/server1/'
TARGET_PW='<Passwort des SFTP-Benutzers>'
...
SOURCE='/'
...
MAX_AGE=4W
...
MAX_FULL_BACKUPS=2
...
MAX_FULLBKP_AGE=2W
DUPL_PARAMS="$DUPL_PARAMS --full-if-older-than $MAX_FULLBKP_AGE "
```

Zu sicherende Dateien (+) und auszulassende Dateien (-) festlegen

.duply/server1-backupserver1/exclude

```
+ /etc
+ /home
+ /opt/xen/_kernel
+ /opt/xen/*/*.cfg
- /root/.cache
+ /root
+ /usr/local
+ /var/lib/portage
+ /var/log
- **
```

Inbetriebnahme

Verwendung SFTP/SSH-Backend

Bei Verwendung des SFTP/SSH-Backends ist als Root-Nutzer erstmals eine Verbindung zum Backup-Server herzustellen, damit der Fingerprint des Backup-Servers in der known_hosts-Datei gespeichert wird.

- ***ssh backupserver1.example.org***
  - Are you sure you want to continue connecting (yes/no)? ***yes***

- Ctrl+C drücken (Abbruch)

Ausgabe

```
The authenticity of host 'backupserver1.example.org (10.10.123.234)' can't
be established.
RSA key fingerprint is 3d:7b:6f:99:5f:68:53:21:73:15:f9:2e:6b:3a:9f:e3.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'backupserver1.example.org,10.10.123.234' (RSA)
to the list of known hosts.
root@backupserver1.example.org's password:
```

Erstes Full-Backup

- ***duply server1-backupserver1 backup***

Ausgabe

```
Start duply v1.5.5.4, time is 2012-03-26 19:38:54.
Using profile '/root/.duply/server1-backupserver1'.
Using installed duplicity version 0.6.17, python 3.2.2, gpg 2.0.17 (Home:
~/.gnupg), awk 'GNU Awk 3.1.8', bash '4.2.20(1)-release (x86_64-pc-linux-
gnu)'.
Using configured key '11ED50F4' as signing key.
Test - Encrypt to 34FE3252 & Sign with 11ED50F4 (OK)
Test - Decrypt (DISABLED) - No matching secret key or GPG_PW not set.
Test - Compare (DISABLED) - Nothing to compare.
Cleanup - Delete '/tmp/duply.16242.1332783534_*'(OK)

--- Start running command PRE at 19:38:54.676 ---
Skipping n/a script '/root/.duply/server1-backupserver1/pre'.
--- Finished state OK at 19:38:54.695 - Runtime 00:00:00.018 ---

--- Start running command BKP at 19:38:54.712 ---
Reading globbing filelist /root/.duply/server1-backupserver1/exclude
Local and Remote metadata are synchronized, no sync needed.
Last full backup date: none
Last full backup is too old, forcing full backup
--------------[ Backup Statistics ]--------------
StartTime 1332783537.25 (Mon Mar 26 19:38:57 2012)
EndTime 1332783538.27 (Mon Mar 26 19:38:58 2012)
ElapsedTime 1.02 (1.02 seconds)
SourceFiles 969
SourceFileSize 15959132 (15.2 MB)
NewFiles 969
NewFileSize 15959132 (15.2 MB)
DeletedFiles 0
```

```
ChangedFiles 0
ChangedFileSize 0 (0 bytes)
ChangedDeltaSize 0 (0 bytes)
DeltaEntries 969
RawDeltaSize 15427734 (14.7 MB)
TotalDestinationSizeChange 9914630 (9.46 MB)
Errors 0
-------------------------------------------------

--- Finished state OK at 19:39:03.120 - Runtime 00:00:08.408 ---

--- Start running command POST at 19:39:03.138 ---
Skipping n/a script '/root/.duply/server1-backupserver1/post'.
--- Finished state OK at 19:39:03.157 - Runtime 00:00:00.019 ---
```

Backup prüfen

- ***duply server1-backupserver1 status***

Ausgabe

```
server1 ~ # duply server1-backupserver1 status
Start duply v1.5.5.4, time is 2012-03-26 19:39:57.
Using profile '/root/.duply/server1-backupserver1'.
Using installed duplicity version 0.6.17, python 3.2.2, gpg 2.0.17 (Home:
~/.gnupg), awk 'GNU Awk 3.1.8', bash '4.2.20(1)-release (x86_64-pc-linux-
gnu)'.
Using configured key '11ED50F4' as signing key.
Test - Encrypt to 34FE3252 & Sign with 11ED50F4 (OK)
Test - Decrypt (DISABLED) - No matching secret key or GPG_PW not set.
Test - Compare (DISABLED) - Nothing to compare.
Cleanup - Delete '/tmp/duply.16758.1332783597_*'(OK)

--- Start running command STATUS at 19:39:57.979 ---
Local and Remote metadata are synchronized, no sync needed.
Last full backup date: Mon Mar 26 19:38:54 2012
Collection Status
-----------------
Connecting with backend: SSHBackend
Archive dir: /root/.cache/duplicity/duply_server1-backupserver1

Found 0 secondary backup chains.

Found primary backup chain with matching signature chain:
-------------------------
Chain start time: Mon Mar 26 19:38:54 2012
Chain end time: Mon Mar 26 19:38:54 2012
Number of contained backup sets: 1
```

```
Total number of contained volumes: 1
 Type of backup set:                            Time:      Num volumes:
                Full         Mon Mar 26 19:38:54 2012                 1
-------------------------
No orphaned or incomplete backup sets found.
--- Finished state OK at 19:40:00.572 - Runtime 00:00:02.592 ---
```

Cronjob anlegen

Um das Backup regelmäßig laufen zu lassen müssen Cronjobs angelegt werden. Der erste prüft wöchtentlich auf veraltete Backups und löscht diese, der zweite führt eine tägliche Sicherung durch. Eine Unterscheidung zwischen Voll- und Inkrementell-Backup ist nicht nötig, da duplicity dank Einstellungen selbst darauf achtet.

/etc/crontab

```
# Backup (Provider)
0 6    * * 1   root    HOME=/root && duply server1-backupserver1
cleanup_purge_purge-full --extra-clean --force
30 6   * * *   root    HOME=/root && duply server1-backupserver1 backup
```