BorgBackup

BorgBackup (kurz: Borg) ist ein deduplizierendes Backup-Programm, das Kompression und authentifizierte Verschlüsselung unterstützt.

Quickstart

1. Initialisierung des Repository

Bevor eine Sicherung durchgeführt werden kann, muss ein Repository initialisiert werden:

```
$ borg init --encryption=repokey /pfad/zu/backup-repo
```

–encryption, auch -e kann folgendes sein
unverschlüsselt: none, authenticated, authenticated-blake2 oder verschlüsselt: repokey *keyfile* repokey-blake2 *keyfile-blake2*

2. Sicherung von Verzeichnissen

Die Sicherung von Verzeichnissen hier ~/bilder und ~/docs in ein Archiv namens montag:

```
$ borg create /pfad/zum/backup-repo::montag ~/bilder ~/docs
```

Die Option –stats bewirkt, dass Borg Statistiken über das neu erstellte Archiv ausgibt, wie z.B. die Menge der eindeutigen Daten (nicht gemeinsam mit anderen Archiven):

```
------------------------------------------------------------------------------
--
Archive name: montag
Archive fingerprint:
bd31004d58f51ea06ff735d2e5ac49376901b21d58035f8fb05dbf866566e3c2
Time (start): Mo, 2019-11-16 18:15:11
Time (end):   Mo, 2019-11-16 18:15:34

Duration: 23.19 seconds
Number of files: 127
------------------------------------------------------------------------------
--
                       Original size      Compressed size    Deduplicated
size
This archive:                4.16 MB             4.17 MB             26.78
kB
All archives:                8.33 MB             8.34 MB              4.19
MB
```

```
                       Unique chunks         Total chunks
Chunk index:                     132                  261
------------------------------------------------------------------------------
--
```

3. Alle Archive aus dem Repository auflisten

```
$ borg list /pfad/zum/backup-repo

montag                               Mon, 2016-11-11 18:15:34
dienstag                             Tue, 2016-11-12 19:15:11
```

4. Den Inhalt des montag-Archivs auflisten

```
$ borg list /pfad/zum/backup-repo::montag

drwxr-xr-x user   group          0 Mon, 2016-02-15 18:22:30 home/user/doc
-rw-r--r-- user   group       7961 Mon, 2016-02-15 18:22:30
home/user/docs/2018-05-11_Bewerbung.odt
...
```

5. Alle Daten des montag-Archivs extrahieren

extrahiert im aktuellen Pfad

```
$ borg extract /pfad/zum/backup-repo::montag
```

---

borg init

Dieser Befehl initialisiert ein Repository. Ein Repository ist ein Dateisystemverzeichnis, das die deduplizierten Daten aus keinem (wenn noch kein backup angelegt wurde) oder mehr Archiven enthält. Die Verschlüsselung kann zur Repository-Init-Zeit aktiviert werden. Es kann später *nicht mehr geändert* werden.

Es wird empfohlen *mit* Verschlüsselung zu arbeiten. Die Verschlüsselung schützt davor, dass ein Angreifer Zugriff auf das Repository bekommt.

Borg verlässt sich auf zufällig generiertes Schlüsselmaterial und verwendet dieses für Chunking, ID-Generierung, Verschlüsselung und Authentifizierung. Das Schlüsselmaterial wird mit der von Ihnen angegebenen Passphrase verschlüsselt, bevor es auf der Festplatte gespeichert wird.

Sie müssen mit dem Schlüssel / der Passphrase vorsichtig sein:

Wenn Sie „nur Passphrase“-Sicherheit wünschen, verwenden Sie einen der Repository-Modi. Der

Schlüssel wird im Repository (in seiner „config“-Datei) gespeichert. Im oben genannten Angriffsszenario hat der Angreifer den Schlüssel (aber nicht die Passphrase).

Wenn Sie „Passphrase und den Schlüssel haben“-Sicherheit wünschen, verwenden Sie einen der Schlüsseldateimodi. Der Schlüssel wird in Ihrem Home-Verzeichnis (in .config/borg/keys) gespeichert. Im Angriffsszenario hat der Angreifer, der nur Zugriff auf Ihr Repository hat, nicht den Schlüssel (und auch nicht die Passphrase).

Erstellen Sie eine Sicherungskopie der Schlüsseldatei (Schlüsseldatei-Modus) oder Repo-Konfigurationsdatei (Repokey-Modus) und bewahren Sie sie an einem sicheren Ort auf, damit Sie den Schlüssel noch haben, falls er beschädigt wird oder verloren geht. Bewahren Sie auch die Passphrase an einem sicheren Ort auf. Das mit diesem Schlüssel verschlüsselte Backup wird Ihnen dabei natürlich nicht weiterhelfen.

Stellen Sie sicher, dass Sie eine gute Passphrase verwenden. Nicht zu kurz, nicht zu einfach. Der echte Verschlüsselungs- / Entschlüsselungsschlüssel wird mit Ihrer Passphrase verschlüsselt / gesperrt. Wenn ein Angreifer Ihren Schlüssel erhält, kann er ihn nicht entsperren und verwenden, ohne die Passphrase zu kennen.

Seien Sie vorsichtig mit Sonderzeichen oder Nicht-ASCII-Zeichen in Ihrer Passphrase:

- Borg verarbeitet die Passphrase als Unicode (und codiert sie als utf-8), sodass es selbst mit den seltsamsten Zeichen keine Probleme hat.
- ABER: das gilt nicht unbedingt für Ihre Betriebssystem-/VM-/Tastaturkonfiguration.

Verwenden Sie also besser eine lange Passphrase, die aus einfachen ASCII-Zeichen besteht, als eine, die Nicht-ASCII-Zeug oder Zeichen enthält, die auf einem anderen Tastaturlayout schwer/unmöglich eingegeben werden können.

Sie können Ihre Passphrase für bestehende Repositorys jederzeit ändern, dies hat keinen Einfluss auf den Verschlüsselungs-/Entschlüsselungsschlüssel oder andere Geheimnisse. Verschlüsselungsmodi

Sie können pro Repository aus den in der folgenden Tabelle aufgeführten Verschlüsselungsmodi wählen. Der Modus bestimmt den Verschlüsselungsalgorithmus, den Hash/MAC-Algorithmus und auch den Schlüsselspeicherort.

Beispiel: `borg init –encryption repokey …`

Hash/MAC	Nicht verschlüsselt keine Authentifizierung	Nicht verschlüsselt, aber authentifiziert	Verschlüsselt (AEAD mit AES) und authentifiziert
SHA-256	keiner	authentifiziert	Repository-Schlüsseldatei
BLAKE2b	n/A	authentifiziert-blake2	repokey-blake2 keyfile-blake2

Modi, die in der obigen Tabelle so gekennzeichnet sind, sind neu in Borg 1.1 und nicht abwärtskompatibel mit Borg 1.0.x.

Auf modernen Intel/AMD-CPUs (außer bei sehr günstigen) ist AES normalerweise hardwarebeschleunigt. BLAKE2b ist schneller als SHA256 auf Intel/AMD 64-Bit-CPUs (außer AMD Ryzen und zukünftigen CPUs mit SHA-Erweiterungen), wodurch authentifiziert-blake2 schneller ist als keine und authentifizierte .

Auf modernen ARM-CPUs bietet NEON Hardwarebeschleunigung für SHA256 und ist damit schneller

als BLAKE2b-256. NEON beschleunigt auch AES.

Hardwarebeschleunigung wird immer automatisch verwendet, wenn verfügbar.

repokey und keyfile verwenden AES-CTR-256 für die Verschlüsselung und HMAC-SHA256 für die Authentifizierung in einer Encrypt-then-MAC (EtM) Konstruktion. Der Chunk-ID-Hash ist ebenfalls HMAC-SHA256 (mit separatem Schlüssel). Diese Modi sind mit Borg 1.0.x kompatibel.

repokey-blake2 und keyfile-blake2 sind ebenfalls authentifizierte Verschlüsselungsmodi, verwenden jedoch BLAKE2b-256 anstelle von HMAC-SHA256 zur Authentifizierung. Der Chunk-ID-Hash ist ein verschlüsselter BLAKE2b-256-Hash. Diese Modi sind neu und nicht kompatibel mit Borg 1.0.x.

Der authentifizierte Modus verwendet keine Verschlüsselung, sondern authentifiziert Repository-Inhalte über denselben HMAC-SHA256-Hash wie der Repository- und Schlüsseldateimodus (er verwendet ihn als Chunk-ID-Hash). Der Schlüssel wird wie repokey gespeichert . Dieser Modus ist neu und nicht kompatibel mit Borg 1.0.x.

authenticated-blake2 ist wie authenticated , verwendet aber den verschlüsselten BLAKE2b-256-Hash aus den anderen blake2-Modi. Dieser Modus ist neu und nicht kompatibel mit Borg 1.0.x.

Der Modus none verwendet keine Verschlüsselung und keine Authentifizierung. Es verwendet SHA256 als Chunk-ID-Hash. Dieser Modus wird nicht empfohlen, Sie sollten lieber einen authentifizierten oder authentifizierten/verschlüsselten Modus verwenden. In diesem Modus treten möglicherweise Denial-of-Service-Probleme auf, wenn er auf Inhalten ausgeführt wird, die von einem Angreifer kontrolliert werden. Verwenden Sie es nur für neue Repositorys, bei denen keine Verschlüsselung erwünscht ist und wenn Kompatibilität mit 1.0.x wichtig ist. Wenn die Kompatibilität mit 1.0.x nicht wichtig ist, verwenden Sie stattdessen authenticated-blake2 oder authentifiziert . Dieser Modus ist mit Borg 1.0.x kompatibel.borg create

Modi, die in der obigen Tabelle so gekennzeichnet sind, sind neu in Borg 1.1 und nicht abwärtskompatibel mit Borg 1.0.x.

Auf modernen Intel/AMD-CPUs (außer bei sehr günstigen) ist AES normalerweise hardwarebeschleunigt. BLAKE2b ist schneller als SHA256 auf Intel/AMD 64-Bit-CPUs (außer AMD Ryzen und zukünftigen CPUs mit SHA-Erweiterungen), wodurch authentifiziert-blake2 schneller ist als keine und authentifizierte .

Auf modernen ARM-CPUs bietet NEON Hardwarebeschleunigung für SHA256 und ist damit schneller als BLAKE2b-256. NEON beschleunigt auch AES.

Hardwarebeschleunigung wird immer automatisch verwendet, wenn verfügbar.

repokey und keyfile verwenden AES-CTR-256 für die Verschlüsselung und HMAC-SHA256 für die Authentifizierung in einer Encrypt-then-MAC (EtM) Konstruktion. Der Chunk-ID-Hash ist ebenfalls HMAC-SHA256 (mit separatem Schlüssel). Diese Modi sind mit Borg 1.0.x kompatibel.

repokey-blake2 und keyfile-blake2 sind ebenfalls authentifizierte Verschlüsselungsmodi, verwenden jedoch BLAKE2b-256 anstelle von HMAC-SHA256 zur Authentifizierung. Der Chunk-ID-Hash ist ein verschlüsselter BLAKE2b-256-Hash. Diese Modi sind neu und nicht kompatibel mit Borg 1.0.x.

Der authentifizierte Modus verwendet keine Verschlüsselung, sondern authentifiziert Repository-Inhalte über denselben HMAC-SHA256-Hash wie der Repository- und Schlüsseldateimodus (er verwendet ihn als Chunk-ID-Hash). Der Schlüssel wird wie repokey gespeichert . Dieser Modus ist neu

und nicht kompatibel mit Borg 1.0.x.

authenticated-blake2 ist wie authenticated , verwendet aber den verschlüsselten BLAKE2b-256-Hash aus den anderen blake2-Modi. Dieser Modus ist neu und nicht kompatibel mit Borg 1.0.x.

Der Modus none verwendet keine Verschlüsselung und keine Authentifizierung. Es verwendet SHA256 als Chunk-ID-Hash. Dieser Modus wird nicht empfohlen, Sie sollten lieber einen authentifizierten oder authentifizierten/verschlüsselten Modus verwenden. In diesem Modus treten möglicherweise Denial-of-Service-Probleme auf, wenn er auf Inhalten ausgeführt wird, die von einem Angreifer kontrolliert werden. Verwenden Sie es nur für neue Repositorys, bei denen keine Verschlüsselung erwünscht ist und wenn Kompatibilität mit 1.0.x wichtig ist. Wenn die Kompatibilität mit 1.0.x nicht wichtig ist, verwenden Sie stattdessen authenticated-blake2 oder authentifiziert . Dieser Modus ist mit Borg 1.0.x kompatibel.borg create

Beispiele

```
# Local repository, repokey encryption, BLAKE2b (often faster, since Borg
1.1)
$ borg init --encryption=repokey-blake2 /path/to/repo

# Local repository (no encryption)
$ borg init --encryption=none /path/to/repo

# Remote repository (accesses a remote borg via ssh)
# repokey: stores the (encrypted) key into <REPO_DIR>/config
$ borg init --encryption=repokey-blake2 user@hostname:backup

# Remote repository (accesses a remote borg via ssh)
# keyfile: stores the (encrypted) key into ~/.config/borg/keys/
$ borg init --encryption=keyfile user@hostname:backup
```

Tricks

Die einzelnen Backup-repositories lassen sich mounten

```
borg mount /pfad/tum/backup-repo /media/bup/
```

Mit diesem Befehl wird ein Archiv als FUSE-Dateisystem gemountet. Dies kann nützlich sein, um ein Archiv zu durchsuchen oder einzelne Dateien wiederherzustellen. Wenn die Option –foreground nicht angegeben ist, wird der Befehl im Hintergrund ausgeführt, bis das Dateisystem umgeschaltet wird.