Lokal verschlüsseltes Cloud-Backup mit Tarsnap

Anvertraut

Michael Plura

Tarsnap ist eine auf den ersten Blick ungewöhnliche Backup-Software. Sie eignet sich vor allem für überschaubare, besonders vertrauliche Daten.

Entwicklern und Administratoren stehen im Unternehmen in der Regel etablierte und mehr oder weniger komplizierte Backup-Lösungen zur Verfügung. Im kleineren Rahmen kann Borg-Backup ausreichen, notfalls lassen sich einzelne Projekte mit rsync sichern. Den Backup-Server muss man aber selbst einrichten.

Helfen soll hier – gegen geringe Bezahlung – der Backup-Dienst Tarsnap von Colin Percival, einem langjährigen Security Officer von FreeBSD. Percival entwickelte etwa die Update-Funktion von FreeBSD und scrypt, eine Schlüsselableitungsfunktion, die viele Bitcoin-Derivate nutzen. Zudem ist er Maintainer der FreeBSD/Amazon-EC2-Plattform.

Sein Tarsnap verbindet eine mit *tar* vergleichbare Funktion samt Deduplizierung und starker lokaler Verschlüsselung mit dem Speicher von Amazons S3. Als Geek-to-Geek-Produkt ist es für große Datenmengen oder unbedarfte Endanwender weniger geeignet. Tarsnap eignet sich zum Sichern überschaubarer und vertraulicher Daten.

Statt auf der GNU-Variante basiert Tarsnap auf *bsdtar* und nutzt *libarchive*, um einen Datenstrom zu erzeugen. Diesen unterteilt es in Blöcke und dedupliziert ihn. Die verbleibenden Blöcke verschlüsselt es und signiert sie genauso wie das gesamte Archiv. Anschließend sendet es diese Daten an den Tarsnap-Server in der Amazon-Cloud (EC2) und speichert sie in Amazon S3.

Die Verschlüsselung findet komplett im quelloffenen und mehrfach auditierten Client statt und verspricht hohe Sicherheit. Hackerangriffe oder Begehren staatlicher Organe in den USA oder Kanada laufen ins Leere, da die Serverinfrastruktur den Schlüssel des Benutzers nicht kennt. Beruhigend: Als bezahlter Dienst ist die Gefahr gering, dass Tarsnap wie bei kostenfreien Diensten üblich die Daten oder das Verhalten des Benutzers durch die Hintertür monetarisiert.

Verfügbar ist es für alle BSD-Betriebssysteme und Mac OS X, für GNU/Linux sowie Windows mit Cygwin oder WSL (Windows Subsystem for Linux). Pflichtbewusste Paranoiker können *tarsnap* aus den Quellen selbst übersetzen.

Nicht optimal: Prepaid-Backup

Vor dem ersten Start ist ein Benutzerkonto auf dem Tarsnap-Server einzurichten. Die anzugebende Mailadresse validiert es per Link. Das Konto muss per Kreditkarte oder PayPal mit mindestens 5 US-Dollar gefüllt sein. An dieser Stelle sollte man überprüfen, ob die Mails korrekt ankommen. Ein Kritikpunkt an Tarsnap kann dessen Prepaid-Abomodell sein: Sieben Tage vor Ablauf des Guthabens gibt es einen Hinweis per Mail. Es folgen weitere Warnungen, aber dann geht es schnell: Wiederum sieben Tage nachdem das Guthaben aufgebraucht ist, werden die Backups gelöscht.

Vor dem Backup erstellt man auf jedem zu sichernden System einen Schlüssel:

```
tarsnap-keygen --keyfile /root/tarsnap.key ⤸
  --user <Mail-Adresse> --machine <Rechnername>
```

Die Schlüsseldatei sollte man anschließend an anderer Stelle sichern, denn ohne kann weder der Nutzer noch der Betreiber des Dienstes die Backups wiederherstellen. Wer *tar* kennt, kommt mit *tarsnap* schnell zurecht. Ein Backup des Webservers *www01* mit Zeitstempel lässt sich mit dem folgenden Kommando erzeugen:

```
tarsnap -c -f www01-2019-01-31_12-00 /var/www
```

Packt man die Zeile in ein Skript, etwa *tarsnap-backup.sh*, und erzeugt den Rechnernamen samt Zeitstempel dynamisch über Variablen, etwa

```
..."$(uname -n)-$(date +%Y-%m-%d_%H-%M-%S)"...
```

lässt sich das Backup über einen Eintrag in der *crontab* automatisieren und so auf weitere Rechner verteilen:

```
12 42 * * * /root/tarsnap-backup.sh
```

Ob Tarsnap wie angegeben täglich um 12:42 Uhr ein Backup erzeugt, lässt sich mit *tarsnap --list-archives | sort* prüfen.

Wer ein Backup in das aktuelle Verzeichnis zurückspielen will, verwendet den Archivnamen und den Befehl *tarsnap -x -f <Archivname>*.

Tarsnap rechnet das Datenvolumen Byte-genau ab. Jedes GByte in der Cloud gespeicherter Daten kostet monatlich 0,25 US-Dollar. Zusätzlich kostet jedes GByte übertragener Daten ebenfalls 0,25 US-Dollar. Diese Kosten sollte man beim Backup-Plan auf keinen Fall vernachlässigen. Umso wichtiger ist es, das Datenvolumen vorab grob abzuschätzen:

```
tarsnap --dry-run --no-default-config ⤸
  --print-stats --humanize-numbers -c /var/www
```

Der relevante Teil der Ausgabe ist die *compressed size* hinter den *unique data*.

Zu *tarsnap* gibt es BSD-typisch eine ausführliche Manpage mit Beispielen. Die Dokumentation auf tarsnap.com ist zwar holprig, aber gut nachvollziehbar. Weitere Hinweise finden sich in den Archiven von tarsnap-users@tarsnap.com. Sie lassen sich leicht mit Google durchsuchen, indem man die Suchbegriffe um den Zusatz *site:mail.tarsnap.com* ergänzt. Support-Anfragen lassen sich auch per E-Mail oder Twitter direkt an Colin Percival stellen. (sun@ix.de)

Michael Plura

arbeitet in Schweden als freier Autor mit den Schwerpunkten IT-Sicherheit, Virtualisierung und freie Betriebssysteme.

Quellen

[1] Downloads unter github.com/Tarsnap
[2] Alle Dokumentationen und User-Guides unter www.tarsnap.com