# Zwei Faktoren für einen sicheren Server

Grazer Linuxtage 2021

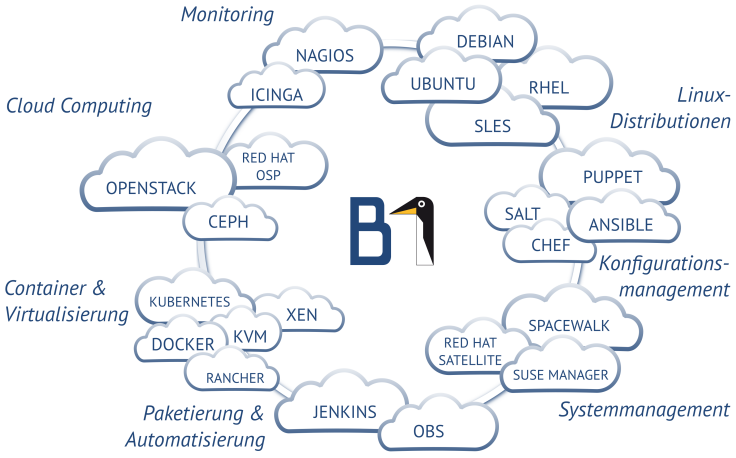
10. April 2021


Florian Winkler
Linux Consultant & Trainer
B1 Systems GmbH
winkler@b1-systems.de

# Vorstellung B1 Systems

- gegründet 2004
- Linux/Open Source-Themen
- national & international tätig
- über 140 Mitarbeiter
- unabhängig von Soft- & Hardware-Herstellern
- Leistungsangebot:
  - Beratung & Consulting
  - Support
  - Training
  - Managed Service & Betrieb
  - Lösungen & Entwicklung
- Standorte in Rockolding, Köln, Berlin & Dresden

# Schwerpunkte



# Zwei Faktoren für einen sicheren Server

# Worum geht es überhaupt?

- Wir haben einen Server, der für uns tolle Dinge tut, per SSH erreichbar ist und unter Linux läuft.

# Worum geht es überhaupt?

- Wir haben einen Server, der für uns tolle Dinge tut, per SSH erreichbar ist und unter Linux läuft.
- Wir wollen diesen Server absichern und Zwei-Faktor-Authentisierung einrichten.

# Worum geht es überhaupt?

- Wir haben einen Server, der für uns tolle Dinge tut, per SSH erreichbar ist und unter Linux läuft.
- Wir wollen diesen Server absichern und Zwei-Faktor-Authentisierung einrichten.
- Wie das geht, zeigen die nächsten 45 Minuten ...

# Warum das Ganze?

Einen Server bedrohen viele Gefahren:

# Warum das Ganze?

Einen Server bedrohen viele Gefahren:

- böse Hacker

# Warum das Ganze?

Einen Server bedrohen viele Gefahren:

- böse Hacker
- Skriptkiddies

# Warum das Ganze?

Einen Server bedrohen viele Gefahren:

- böse Hacker
- Skriptkiddies
- unsichere Dienste

# Warum das Ganze?

Einen Server bedrohen viele Gefahren:

- böse Hacker
- Skriptkiddies
- unsichere Dienste
- automatisierte Scans

# Warum das Ganze?

Einen Server bedrohen viele Gefahren:

- böse Hacker
- Skriptkiddies
- unsichere Dienste
- automatisierte Scans
- schwache Passwörter

# Warum das Ganze?

Einen Server bedrohen viele Gefahren:

- böse Hacker
- Skriptkiddies
- unsichere Dienste
- automatisierte Scans
- schwache Passwörter
- und, und, und...

# Was tun wir dagegen?

Unter anderem

# Was tun wir dagegen?

Unter anderem

- nicht benötigte Dienste abschalten, besser deinstallieren

# Was tun wir dagegen?

Unter anderem

- nicht benötigte Dienste abschalten, besser deinstallieren
- nur benötigte Pakte installieren

# Was tun wir dagegen?

Unter anderem

- nicht benötigte Dienste abschalten, besser deinstallieren
- nur benötigte Pakte installieren
- root-Login verbieten

# Was tun wir dagegen?

Unter anderem

- nicht benötigte Dienste abschalten, besser deinstallieren
- nur benötigte Pakte installieren
- root-Login verbieten
- sichere Passwörter verwenden

# Was tun wir dagegen?

Unter anderem

- nicht benötigte Dienste abschalten, besser deinstallieren
- nur benötigte Pakte installieren
- root-Login verbieten
- sichere Passwörter verwenden
- besser: nicht auf Passwörter verlassen $\rightarrow$ Zwei-Faktor-Authentisierung (2FA)

# Zwei-Faktor-Authentisierung (2FA)

Mögliche Faktoren sind

# Zwei-Faktor-Authentisierung (2FA)

Mögliche Faktoren sind

- Passwort

# Zwei-Faktor-Authentisierung (2FA)

Mögliche Faktoren sind

- Passwort
- Kryptografische Schlüssel, z. B. SSH-Keys

# Zwei-Faktor-Authentisierung (2FA)

Mögliche Faktoren sind

- Passwort
- Kryptografische Schlüssel, z. B. SSH-Keys
- Hardware-Token, z. B. Yubikey

# Zwei-Faktor-Authentisierung (2FA)

Mögliche Faktoren sind

- Passwort
- Kryptografische Schlüssel, z. B. SSH-Keys
- Hardware-Token, z. B. Yubikey
- One-Time-Passwords (OTP)

# Zwei-Faktor-Authentisierung (2FA)

Mögliche Faktoren sind

- Passwort
- Kryptografische Schlüssel, z. B. SSH-Keys
- Hardware-Token, z. B. Yubikey
- One-Time-Passwords (OTP)
- Software-Token

# Zwei-Faktor-Authentisierung (2FA)

Mögliche Faktoren sind

- Passwort
- Kryptografische Schlüssel, z. B. SSH-Keys
- Hardware-Token, z. B. Yubikey
- One-Time-Passwords (OTP)
- Software-Token
- Biometrisches Merkmal, z. B. Fingerabdruck, Face ID

# Zwei-Faktor-Authentisierung (2FA)

Mögliche Faktoren sind

- Passwort
- Kryptografische Schlüssel, z. B. SSH-Keys
- Hardware-Token, z. B. Yubikey
- One-Time-Passwords (OTP)
- Software-Token
- Biometrisches Merkmal, z. B. Fingerabdruck, Face ID
- Schlüssel (physischer Zugang)

# Zwei-Faktor-Authentisierung (2FA)

Mögliche Faktoren sind

- Passwort
- Kryptografische Schlüssel, z. B. SSH-Keys
- Hardware-Token, z. B. Yubikey
- One-Time-Passwords (OTP)
- Software-Token
- Biometrisches Merkmal, z. B. Fingerabdruck, Face ID
- Schlüssel (physischer Zugang)
- ...

# Google Authenticator

# Google Authenticator

- initiiert von Google

# Google Authenticator

- initiiert von Google
- komplett Open Source

# Google Authenticator

- initiiert von Google
- komplett Open Source
- baut keine Verbindung zu Google-Diensten auf

# Google Authenticator

- initiiert von Google
- komplett Open Source
- baut keine Verbindung zu Google-Diensten auf
- `https://github.com/google/google-authenticator`

# Google Authenticator

- initiiert von Google
- komplett Open Source
- baut keine Verbindung zu Google-Diensten auf
- `https://github.com/google/google-authenticator`
- Komponenten:

# Google Authenticator

- initiiert von Google
- komplett Open Source
- baut keine Verbindung zu Google-Diensten auf
- `https://github.com/google/google-authenticator`
- Komponenten:
  - CLI-Tool

# Google Authenticator

- initiiert von Google
- komplett Open Source
- baut keine Verbindung zu Google-Diensten auf
- `https://github.com/google/google-authenticator`
- Komponenten:
  - CLI-Tool
  - PAM-Modul

# Google Authenticator

- initiiert von Google
- komplett Open Source
- baut keine Verbindung zu Google-Diensten auf
- `https://github.com/google/google-authenticator`
- Komponenten:
  - CLI-Tool
  - PAM-Modul
  - App

# Installation

## Debian

```
$ sudo apt-get install libpam-google-authenticator
```

# Installation

## Debian

```
$ sudo apt-get install libpam-google-authenticator
```

## Ubuntu

```
$ sudo apt install libpam-google-authenticator
```

# Installation

## Debian

```
$ sudo apt-get install libpam-google-authenticator
```

## Ubuntu

```
$ sudo apt install libpam-google-authenticator
```

## OpenSUSE

```
$ sudo zypper install google-authenticator-libpam
```

# Installation

## Debian

```
$ sudo apt-get install libpam-google-authenticator
```

## Ubuntu

```
$ sudo apt install libpam-google-authenticator
```

## OpenSUSE

```
$ sudo zypper install google-authenticator-libpam
```

## Fedora

```
$ sudo dnf install -y google-authenticator
```

# App

## Google Play Store

`https://play.google.com/store/apps/details?id=com.google.android.apps.authenticator2&hl=de`

# App

## Google Play Store

`https://play.google.com/store/apps/details?id=com.google.android.apps.authenticator2&hl=de`

## Apple iTunes

`https://itunes.apple.com/de/app/google-authenticator/id388497605?mt=8`

# App

## Google Play Store

`https://play.google.com/store/apps/details?id=com.google.android.apps.authenticator2&hl=de`

## Apple iTunes

`https://itunes.apple.com/de/app/google-authenticator/id388497605?mt=8`

## FreeOTP+ (fdroid)

`https://f-droid.org/en/packages/org.liberty.android.freeotpplus`

# App

## Google Play Store

`https://play.google.com/store/apps/details?id=com.google.android.apps.authenticator2&hl=de`

## Apple iTunes

`https://itunes.apple.com/de/app/google-authenticator/id388497605?mt=8`

## FreeOTP+ (fdroid)

`https://f-droid.org/en/packages/org.liberty.android.freeotpplus`

# Google Authenticator einrichten 1/2

## CLI-Tool ausführen

```
$ google-authenticator
```

# Google Authenticator einrichten 1/2

## CLI-Tool ausführen

```
$ google-authenticator
```

## Fragen beantworten

```
Do you want authentication tokens to be time-based (y/n) y
Do you want me to update your "/home/user/.google_authenticator" file (y/n)? y
```

# Google Authenticator einrichten 1/2

## CLI-Tool ausführen

```
$ google-authenticator
```

## Fragen beantworten

```
Do you want authentication tokens to be time-based (y/n) y
Do you want me to update your "/home/user/.google_authenticator" file (y/n)? y
```

- Secret Key, Verification Code und Recovery Codes notieren und sicher aufbewahren

# Google Authenticator einrichten 1/2

## CLI-Tool ausführen

```
$ google-authenticator
```

## Fragen beantworten

```
Do you want authentication tokens to be time-based (y/n) y
Do you want me to update your "/home/user/.google_authenticator" file (y/n)? y
```

- Secret Key, Verification Code und Recovery Codes notieren und sicher aufbewahren
- QR Code mit der App einlesen

# Google Authenticator einrichten 2/2

## Weitere Fragen beantworten

```
Do you want to disallow multiple uses of the same authentication token? This restricts you to one login about every 30s,
but it increases your chances to notice or even prevent man-in-the-middle attacks (y/n) y

By default, tokens are good for 30 seconds. In order to compensate for possible time-skew between the client and the
server, we allow an extra token before and after the current time. If you experience problems with poor time synchronization,
you can increase the window from its default size of +-1min (window size of 3) to about +-4min (window size of 17 #
acceptable tokens).

Do you want to do so? (y/n) n
If the computer that you are logging into isn't hardened against brute-force login attempts, you can enable rate-limiting
for the authentication module. By default, this limits attackers to no more than 3 login attempts every 30s.
Do you want to enable rate-limiting (y/n) y
```

# PAM einrichten

# PAM einrichten

Pluggable Authentication Modules

# PAM einrichten

Pluggable Authentication Modules

- modulare Zugriffskontrolle

# PAM einrichten

Pluggable Authentication Modules

- modulare Zugriffskontrolle
- unterscheidbar nach Anwendungsfall

# PAM einrichten

Pluggable Authentication Modules

- modulare Zugriffskontrolle
- unterscheidbar nach Anwendungsfall
- zum Beispiel:

# PAM einrichten

Pluggable Authentication Modules

- modulare Zugriffskontrolle
- unterscheidbar nach Anwendungsfall
- zum Beispiel:
  - lokaler Konsolenlogin

# PAM einrichten

Pluggable Authentication Modules

- modulare Zugriffskontrolle
- unterscheidbar nach Anwendungsfall
- zum Beispiel:
  - lokaler Konsolenlogin
  - DisplayManager

# PAM einrichten

Pluggable Authentication Modules

- modulare Zugriffskontrolle
- unterscheidbar nach Anwendungsfall
- zum Beispiel:
  - lokaler Konsolenlogin
  - DisplayManager
  - ssh-Login

# Konfiguration anpassen

/etc/pam.d/sshd

```
#auth       substack     password-auth
[...]
auth requisite pam_google_authenticator.so
```

# ssh einrichten

# ssh einrichten

## ssh Key erstellen

```
$ ssh-keygen
```

# ssh einrichten

## ssh Key erstellen

```
$ ssh-keygen
```

## ssh Key auf den Server kopieren

```
$ ssh-copy-id <user>@<server>
```

# ssh einrichten

## ssh Key erstellen

```
$ ssh-keygen
```

## ssh Key auf den Server kopieren

```
$ ssh-copy-id <user>@<server>
```

# sshd einrichten

## /etc/ssh/sshd_config

```
[...]
PermitRootLogin no
[...]
ChallengeResponseAuthentication yes
[...]
PasswordAuthentication no
AuthenticationMethods publickey,keyboard-interactive
```

# sshd für Benutzer anpassen

/etc/ssh/sshd_config

```
Match User tux
ChallengeResponseAuthentication yes
PasswordAuthentication no
AuthenticationMethods publickey,keyboard-interactive

Match User ansible
ChallengeResponseAuthentication no
PasswordAuthentication no
AuthenticationMethods publickey
```

# sshd neustarten

**OpenSUSE, Fedora**

```
$ sudo systemctl restart sshd.service
```

# sshd neustarten

## OpenSUSE, Fedora

```
$ sudo systemctl restart sshd.service
```

## Debian, Ubuntu

```
$ sudo systemctl restart ssh.service
```

# ssh mit 2FA

## ssh auf den Server

```
[user@client ~]$ ssh user@example.com
Verification code:
```

# Demotime

# Vielen Dank für Ihre Aufmerksamkeit!

Bei weiteren Fragen wenden Sie sich bitte an info@b1-systems.de oder +49 (0)8457 - 931096