

SAMBA Konfigurieren unter Linux

Mit die nachfolgende Anleitung beschreibt die Grundkonfiguration von SAMBA, einem SMB Server der unter Linux Betrieben werden kann.

Sicherheitshinweis: SMB ist ein Protokoll, dass es nun schon sehr lange gibt und in den letzten Jahren starke Veränderungen erfahren hat. Ab der SMB Version 3 wurde Verschlüsselung eingeführt. Zuvor war es nicht möglich, den SMB Datenverkehr ausreichend sicher zu verschlüsseln. Eine Nutzung im LAN ist hierbei noch zu vertreten, aber in diesem Setup richten Sie eine SMB-Verbindung über das Internet ein. Dieses Setup darf nicht mit einer SMB Version kleiner als SMBv3.1.1 vorgenommen werden. **SMB1 und SMB2 sind nicht sicher und für den Betrieb über das Internet nicht ausgelegt.**

Die Version SMB3 ist ausreichend sicher und wird auch von Microsoft in der Azure Cloud in dieser Form bereitgestellt. **In diesem Setup wird daher die SMB1 und SMB2 Unterstützung des SAMBA Server deaktiviert. Wird dies vergessen, gehen Sie ein grosses Risiko ein, was entsprechend in der Bewertung der Arbeit sich niederschlagen wird!**

Grundkonfiguration

Sie finden die Konfigurations-Datei unter folgendem Pfad: `/etc/samba/smb.conf`

Die Direktive der SAMBA-Konfiguration sind wie beim BIND Dienst in einzelne Sektionen unterteilt. Diese sind jedoch nicht mit geschweiften Klammern umschlossen. Stattdessen beginne jede Sektion mit einem Titel in eckigen Klammern z. B. die **global**-Sektion:

```
[global]
    workgroup = SAMBA
    security = user

    passdb backend = tdbsam

    printing = cups
    printcap name = cups
    load printers = yes
    cups options = raw
```

Sektion [global]

In der globalen Sektion werden die allgemeinen Einstellungen und Sicherheitsoptionen festgelegt. Sie beginnt, wie der Namen vermuten lässt, mit der Bezeichnung **global**. Diese Bezeichnung ist reserviert und darf nur für die globale Konfiguration verwendet werden. (Es kann keine Freigabe mit diesem Namen erstellt werden). Üblicherweise müssen nur wenige Werte in dieser Sektion angepasst werden.

In der Standard-Konfiguration der **global**-Sektion fehlen ein paar wichtige Punkte, welche zwingend gesetzt werden sollten:

protocol = SMB3	Setzt die Protokoll-Version auf SMB Version 3. Damit können sich zwar ältere Clients nicht mehr verbinden, aber die Verbindung kann dafür sauber verschlüsselt werden. SMB Version 3 wird ab Windows 8 unterstützt. SMT Version 3 wird ab dem Linux Kernel 3.12 unterstützt.
lanman auth = no	Deaktiviert eine unsichere Authentifizierungsmethode
smb encrypt = required	Die Verschlüsselung des SMB Traffics wird erzwungen.
encrypt passwords = yes	Stellt sicher, dass die Passwörter verschlüsselt übertragen werden.
access based share enum = yes	Dadurch werden Freigaben ausgeblendet, wenn keine Berechtigung auf diesen Freigaben existieren. Diese Option ist standardmässig deaktiviert. Somit werden standardmässig dem Benutzer alle Freigaben auf dem SMB-Server angezeigt, auf wenn der Benutzer auf gewisse keinen Zugriff hat. Der Benutzer kann dann diese Freigaben öffnen, erhält aber einen "Zugriff verweigert" Fehler.

Dann gibt es ein paar Punkte, welche wir in der global-Sektion existieren und nicht angepasst werden müssen. Es ist aber wertvoll die Bedeutung dieser Punkte zu verstehen.

workgroup = SAMBA	Wenn der SMB-Server in einer Arbeitsgruppe verwendet wird, müsste hier die korrekte Arbeitsgruppe definiert werden. Dies ist in unserem Fall nicht nötig.
security = user	Definiert, dass die Authentifizierung des SMB-Server gegenüber (vom Client) mittels Benutzername und Passwort erfolgt. Es gibt hier weitere mögliche Optionen: • share: Es wird kein Benutzername benötigt, aber ein Passwort pro Share. • domain: Die Authentifizierung wird über einen Domänencontroller abgewickelt. Der Benutzer muss dann keine Anmeldung vornehmen. • ADS: Die Authentifizierung wird über einen Active Directory Server vorgenommen.
passdb backend = tdbsam	Diese Direktive steuert wo Samba die Passwörter und Benutzerdaten speichert. Diese speichert Samba standardmässig in einer " TDB " (Trivial Database). Die Verwendung dieses Backends erfordert keine zusätzliche Konfiguration. Dieses Backend wird für neue Installationen empfohlen, die kein LDAP benötigen. Wird die Option ldapsam verwendet, dann werden die Benutzerdaten in einem ldap Verzeichnis gespeichert.

Sektion [homes]

Über die homes-Sektion können die Heim-Verzeichnisse der Benutzer aktiviert werden. Jeder Benutzer erhält dadurch automatisch Zugriff auf das Verzeichnis mit seinem Namen.

Die Homes-Verzeichnisse liegen in Linux unter /home und werden automatisch angelegt, wenn ein neuer Benutzer mit **adduser** hinzugefügt wird. Dieser Vorgang ist im Abschnitt "Benutzer hinzufügen" genauer beschrieben.

Standardmässig sieht diese Sektion wie folgt aus:

```
[homes]
    comment = Home Directories
    valid users = %S
    browseable = No
    read only = No
    inherit acls = Yes
```

Nachfolgend die Erklärung zu den einzelnen Direktiven:

comment = Home Directories	Dies ist ein Kommentar der dieser Freigabe mitgegeben wird.
valid users = %S	Definiert, welche Benutzer auf Ihre Heim-Verzeichnisse zugreifen können. Der Parameter %S wird bei einer Anmeldung automatisch mit dem Benutzernamen ersetzt. Alternativ kann auch die folgende Option gesetzt werden: %S, %D%w%S D: Name der Domäne oder Arbeitsgruppe des aktuellen Benutzers. w: Winbind Separator. S: Benutzernamen Dies wird benötigt, wenn die Anmeldung mit dem vollen Domänen-Namen erfolgen soll. In unserem Fall ist dies aber nicht relevant.
browseable = No	Diese Direktive verhindert nicht, wie angenommen werden könnte, dass der Benutzer in seiner Freigabe browsen kann, sondern, dass die Freigabe doppelt sichtbar ist. Einmal unter /homes und einmal unter /benutzernamen
read only = No	Der SMB-Benutzer erhält Schreibrechte auf diese Freigabe Achtung: alternativ kann auch die Variante writeable = yes verwendet werden.
inherit acls = Yes	Sie kann verwendet werden, um sicherzustellen, dass, wenn Standard-Zugriffsrechte auf übergeordneten Verzeichnissen existieren, diese immer beachtet werden, wenn eine neue Datei oder ein Unterverzeichnis in diesen übergeordneten Verzeichnissen angelegt wird.

Verzeichnis für die Freigabe erstellen

Definieren Sie ein Verzeichnis, in dem Sie die Unterverzeichnisse für die Freigaben einrichten. Erstellen Sie das Verzeichnis /shared im Verzeichnis /var

```
mkdir -p /var/shared
```

Neue Freigabe einrichten

Die oben besprochene Home-Freigabe (Benutzer / Heimatverzeichnis) ermöglichen es dem Benutzer, seine Daten zentral auf dem Server zu organisieren, jedoch bietet diese keine Möglichkeit zum Datenaustausch unter mehreren Benutzern. Die Verzeichnisse sind für andere Benutzer unsichtbar und

unerreichbar. Eine Lösung hierfür sind

Eine Freigabe wird eingerichtet, indem ein weiterer Block der Konfiguration hinzugefügt wird:

```
[FREIGABE_NAME]
    path= /var/shared/VERZEICHNIS_NAME
    writeable = yes
```

Standardmässig erhalten alle Benutzer des File-Servers Zugriff, da der Zugriff nicht auf bestimmte Benutzer eingeschränkt ist. Dies kann beispielsweise verwendet werden, um eine unternehmensweite Freigabe zu erstellen.

Gruppenfreigabe

Unter valid users können eine oder mehrere berechtigte Gruppen angegeben werden.

```
[FREIGABE_NAME]
    path= /var/shared/VERZEICHNIS_NAME
    writeable = yes
    valid users = @gruppenname
```

Will man mehreren Gruppen den Zugriff erlauben, dann müssen die Gruppen kommasepariert hinterlegt werden:

```
[FREIGABE_NAME]
    path= /var/shared/VERZEICHNIS_NAME
    writeable = yes
    valid users = @gruppenname1, @gruppenname2
```

Wenn mehrere Gruppen auf eine Freigabe Zugriff haben, muss sichergestellt werden, dass wenn eine neue Datei erstellt wird, die Berechtigung von der Hauptgruppe übernommen wird. (Die Gruppe der auch das Verzeichnis auf dem File-Server gehört), dies wird mit der Direktiven **force group** erreicht. Mit create mask und Directory mask wird zudem sichergestellt, dass die Schreib und Leseberechtigung korrekt gesetzt wird beim Anlegen einer neuen Datei oder eines neuen Verzeichnisses:

```
[FREIGABE_NAME]
    path= /var/shared/VERZEICHNIS_NAME
    writeable = yes
    valid users = @gruppenname1, @gruppenname2
    force group = +gruppenname1
    create mask = 0660
    directory mask = 0770
```

Sollen nur bestimmte Gruppen oder Benutzer Schreibberechtigung auf ein Verzeichnis erhalten, dann muss das Verzeichnis auf readonly gesetzt werden. Jene Benutzer der nun Schreibberechtigung erhalten soll, muss separat in der Direktive **write list** freigeschaltet werden.

```
[FREIGABE_NAME]
    path= /var/shared/VERZEICHNIS_NAME
    writeable = no
    write list = @gruppenname1
    valid users = @gruppenname1, @gruppenname2
    force group = +gruppenname1
    create mask = 0660
    directory mask = 0770
```

Benutzerverwaltung

Samba verwendet die Berechtigungen der eingerichteten Benutzer auf dem Betriebssystem.

Nachfolgend wird beschrieben, wie die Benutzer unter Linux verwaltet werden.

Linux Benutzer hinzufügen

Als erstes muss auf dem Betriebssystem ein Benutzer angelegt werden. Dies wird mit dem Befehl **adduser** erledigt:

```
adduser BENUTZERNAME
```

Nun muss dem neuen Benutzer noch ein Samba-Passwort zugewiesen werden

```
smbpasswd -a BENUTZERNAME
```

Linux Gruppe hinzufügen

Als erstes muss auf dem Betriebssystem eine Gruppe angelegt werden. Dies wird mit dem Befehl **groupadd** erledigt:

```
groupadd BENUTZERNAME
```

Nun kann auf dem Betriebssystem ein Verzeichnis angelegt werden, in dem die Daten dieser Gruppe abgelegt werden:

```
mkdir -p /var/shared/GRUPPE_VERZEICHNIS
```

Anschliessend muss die Gruppe diesem Verzeichnis zugewiesen werden:

```
chgrp GRUPPE /var/shared/GRUPPE_VERZEICHNIS
```

Damit existiert auf dem Betriebssystem nun eine Gruppe mit einem eigenen Verzeichnis. Damit dieses Verzeichnis auch als Freigabe existiert, muss dieses nun noch als Freigabe eingerichtet werden. Dazu wird ein weitere Block eingefügt:

```
[ GRUPPE]  
path= /var/shared/GRUPPEN_VERZEICHNIS  
writeable = yes  
valid users = @GRUPPE
```

Nun kann die Datei gespeichert und der Dienst neu gestartet werden.

Linux Benutzer einer Gruppe hinzufügen:

Um einem Gruppe einen Benutzer hinzuzufügen, wird der Befehl **usermod** verwendet:

```
usermod -a -G gruppenname benutzername
```

Linux Benutzer aus einer Gruppe löschen:

Um einen Benutzer aus einer Gruppe zu löschen, wird der Befehl **gpasswd** verwendet:

```
gpasswd --delete gruppenname benutzername
```

Gruppen eines Benutzers anzeigen:

Wenn Sie wissen möchten, in welchen Gruppen ein Benutzer ist, können Sie den Befehl **groups** verwenden:

```
groups benutzername
```

Konfiguration testen

Mit dem Befehl **testparm** kann die Samba-Konfiguration getestet werden. Ungültige Direktiven und Probleme werden im Bericht ausgegeben:

```
testparm
```

Revision #15

Created 8 January 2023 20:05:26 by Admin

Updated 31 January 2023 21:56:51 by Admin