Windows Authentifizierung für SQL Server auf Ubuntu

Kathegorie
SQL Server

Simon
Schlaipfer

In diesem Beitrag wollen wir uns anhand eines praktischen Beispiels anschauen, wie wir einen Ubuntu 18.04 Server in eine Micosoft Active Directory Domäne hinzufügen und anschließend SQL Server für die Active Directory Authentifizierung konfigurieren können.

Wir verwenden hierfür einen frischen Ubuntu 18.04 Server und einen Windows Server 2019, der uns als Domain Controller dient und uns die Domäne madafa.local bereitstellt.

Um nun erfolgreich eine Active Directory Authentifizierung für einen SQL Server auf unserer Ubuntu Maschine zu konfigurieren, werden wir:

- den Ubuntu 18.04 der Domäne **madafa.local** hinzufügen,
- einen SQL Server auf unserer Ubuntu Maschine installieren,
- ein **Key Tab File** für die Keberos Authentifizierung erstellen,
- einen Windows Login für den SQL Server erstellen.

Den Ubuntu Server der Domäne hinzufügen

Als ersten Schritt werden wir unseren Ubuntu 18.04 Server unserer Domäne **madafa.local** hinzufügen. Hierfür verbinden wir uns zunächst mit unserem Server via SSH und setzen mit den folgenden Kommandos die Landeseinstellungen auf Deutschland:

```
export LC_ALL=de_DE.UTF-8
export LANG=de_DE.UTF-8
```

Nun aktualisieren wir die Domänen-Suchliste, indem wir das Network Interface des Ubuntu Servers bearbeiten. Um die Network Interface Konfiguration zu öffnen, bitte folgendes Kommando ausführen:

```
sudo vi /etc/network/interfaces
```


Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz

info@madafa.de
madafa.de

+49 6131 3331612

Hier fügen wir die IP Adresse unseres Domain Controllers sowie den Namen unserer Domäne hinzu:

```
auto ens160
iface ens160 inet dhcps
dns-nameservers 172.17.3.1
dns-search madafa.local
```

Um das Bearbeitete zu speichern und den Editor zu verlassen, **ESC** und anschließend zwei mal **SHIFT+Z** drücken. Anschließend müssen wir den Network Service neustarten. Hierfür das folgende Kommando ausführen:

```
sudo ifdown ens160 && sudo ifup ens160
```

Jetzt können wir testen, ob wir eine Verbindung zu unserer Domäne herstellen können. Hierfür führen wir diesen Befehl aus:

```
ping madafa.local
```

Und tatsächlich kann eine Verbindung hergestellt werden:

```
64 bytes from 172.17.1.73 (172.17.1.73): icmp_seq=9 ttl=128 time=0.627 ms
64 bytes from 172.17.1.73 (172.17.1.73): icmp_seq=10 ttl=128 time=0.578 ms
64 bytes from 172.17.1.73 (172.17.1.73): icmp_seq=11 ttl=128 time=0.573 ms
64 bytes from 172.17.1.73 (172.17.1.73): icmp_seq=12 ttl=128 time=0.584 ms
64 bytes from 172.17.1.73 (172.17.1.73): icmp_seq=13 ttl=128 time=0.534 ms
64 bytes from 172.17.1.73 (172.17.1.73): icmp_seq=14 ttl=128 time=0.540 ms
64 bytes from 172.17.1.73 (172.17.1.73): icmp_seq=15 ttl=128 time=0.510 ms
64 bytes from 172.17.1.73 (172.17.1.73): icmp_seq=16 ttl=128 time=0.578 ms
64 bytes from 172.17.1.73 (172.17.1.73): icmp_seq=17 ttl=128 time=0.504 ms
^C
--- madafa.local ping statistics ---
17 packets transmitted, 17 received, 0% packet loss, time 16027ms
```

Anschließend überprüfen wir, ob die IP Adresse unseres Domain Controllers sowie die Domäne im "**/etc/resolv.conf**" File aktualisiert wurden:

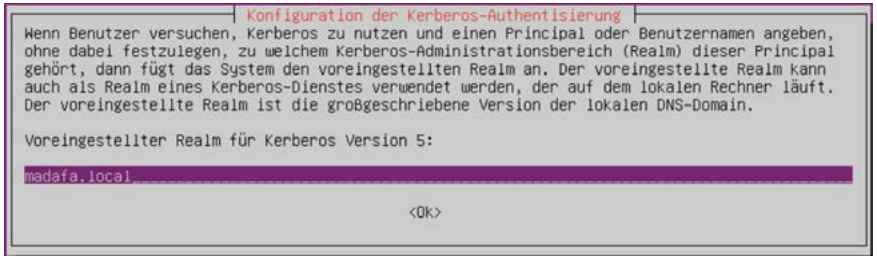
```
sudo cat /etc/resolv.conf
```

```
nameserver 127.0.0.53
options edns0
search madafa.local
```

Nachdem wir die Konfiguration abgeschlossen haben, können wir damit beginnen, den Server in die Active Directory Domäne aufzunehmen. Hierfür installieren wir **realmd, sssd, Kerberos client packages** und alle weiteren notwendigen Pakete auf unserem Ubuntu Server:

```
sudo apt-get update
sudo apt-get -y install realmd sssd sssd-tools samba-common krb5-user packagekit
samba-common-bin samba-libs adcli ntp
```

Es erscheint ein Dialog in dem wir nun unsere Domäne angeben:



Jetzt müssen wir ein realmd.conf File erstellen und konfigurieren. Wir können das File erstellen und direkt in einem Editor mit folgendem Kommando öffnen:

```
sudo vi /etc/realmd.conf
```

Das File konfigurieren wir dann wie folgt:

Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612

```
[users]
default-home = /home/%U
default-shell = /bin/bash
[active-directory]
default-client = sssd
os-name = Ubuntu Server
os-version = 16.04
[service]
automatic-install = no
[madafa.local]
fully-qualified-names = no
automatic-id-mapping = yes
user-principal = yes
manage-system = no
```

Der Server ist nun bereit in die Domäne aufgenommen zu werden. Hierfür führen wir folgendes Kommando aus:

```
sudo realm --verbose join madafa.local --user-principal=UBUNTU/
administrator@MADAFA.LOCAL
```

Nach dem Ausführen des Kommandos wird nach dem Administrator Passwort des Domain-Controllers gefragt. Nachdem wir dieses eingegeben haben, wird der Server zur Domäne hinzugefügt:

```
 * Added the entries to the keytab: RestrictedKrbHost/activedir01@MADAFA.LOCAL: FILE:/etc/krb5.keyta
b
 * /usr/sbin/update-rc.d sssd enable
 * /usr/sbin/service sssd restart
 * Successfully enrolled machine in realm
```

Da wir unseren Ubuntu Server erfolgreich in unsere Domäne aufnehmen konnten, können wir damit beginnen, einen SQL Server zu installieren und ein **Key Tab File** für eine Kerberos Authentifizierung zu erstellen.

SQL Server installieren

Um SQL Server auf Ubuntu zu installieren, müssen wir zunächst GPG Schlüssel importieren. Dies können wir mit folgendem Befehl:

```
sudo wget -qO- https://packages.microsoft.com/keys/microsoft.asc | sudo apt-key
add -
```

Nach dem Importieren der Schlüssel müssen wir das Microsoft SQL Server Ubuntu Repository registrieren:

```
sudo add-apt-repository "$(wget -qO- https://packages.microsoft.com/config/
ubuntu/18.04/mssql-server-2019.list)"
```

Jetzt können wir die SQL Server Installation mit diesen Kommandos beginnen:

```
sudo apt-get update
sudo apt-get install -y mssql-server
```

Um das Setup abzuschließen, muss die SQL Server Version und das sa Passwort festgelegt sowie die Lizenzvereinbarung akzeptiert werden. Das geht mit dem folgenden Kommando:

```
sudo /opt/mssql/bin/mssql-conf setup
```

```
Created symlink /etc/systemd/system/multi-user.target.wants/mssql-server.service → /lib/systemd/syst
em/mssql-server.service.
Das Setup wurde erfolgreich abgeschlossen. SQL Server wird jetzt gestartet.
```

Anhand dieser Nachricht können wir sehen, dass der SQL Server erfolgreich installiert wurde.

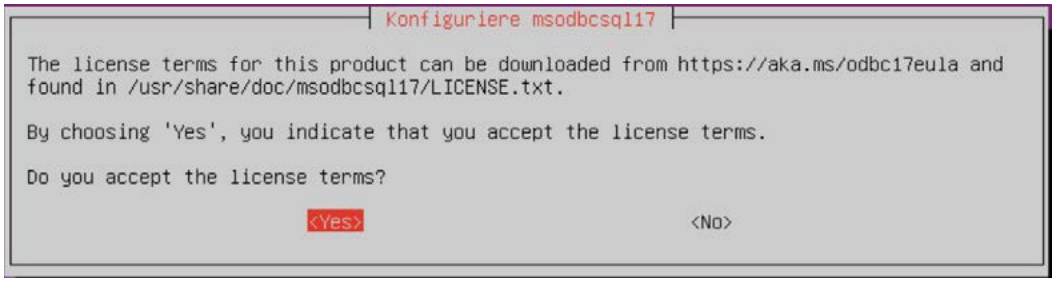
Nun installieren wir die **SQL Server Command Line Tools** und setzen die nötigen Umgebungsvariablen. Das **Command Line Tool** installiert **SQLCMD** und **BCP Utilities**. **SQLCMD** wird zum Verbinden mit der Instanz sowie für das Ausführen von Abfragen verwendet und **BCP** für das Exportieren und Importieren von SQL Server Daten.

Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612

Die **SQL Server Command Line Tools** installieren wir mit:

```
curl https://packages.microsoft.com/keys/microsoft.asc | sudo apt-key add -
curl https://packages.microsoft.com/config/ubuntu/16.04/prod.list | sudo tee /etc/
apt/sources.list.d/msprod.list
sudo apt-get update
sudo apt-get install mssql-tools unixodbc-dev
```

Zum Akzeptieren der Lizenzbedingungen öffnet sich nun folgender Dialog:



Nachdem die Installation abgeschlossen ist, setzen wir die Umgebungsvariablen, um auf SQLCMD zuzugreifen. Dies tun wir mit:

```
echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bash_profile
echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bashrc
source ~/.bashrc
```

Jetzt können wir uns mittels SQLCMD mit dem SQL Server verbinden. Hierfür führe dieses Kommando aus:

```
sqlcmd -S SERVERNAME -U sa -P 'password'
```

Hier muss das vorher konfigurierte Passwort sowie der Servername angegeben werden. Somit ist auch SQL Server auf unserem Ubuntu Server installiert und wir können mit dem dritten Schritt, dem Erstellen des **Key Tab File** für die Keberos Authentifizierung, fortfahren.

Das Key Tab File

Um ein **Key Tab File** zu erstellen, müssen wir zunächst einen Benutzer in unserem Active Directory erstellen. Hierbei achten wir darauf, dass wir die Option **Password never expires** auswählen. In diesem Beispiel nennen wir unseren neuen Active Directory Benutzer **sqltestuser**. Anschließend führen wir folgendes PowerShell Kommando auf dem Domain Controller aus:

```
setspn -A MSSQLSvc/activedir01.madafa.local:1433 sqltestuser
```

Dieses Kommando setzt den **ServicePrincipalName (SPN)**. Innerhalb des Kommandos verwenden wir den vorher erstellten User **sqltestuser** und den **FQDN** unserer Linux Maschine, auf der der SQL Server installiert ist, **activedir01**.

Nun können wir das **Key Tab File**, das für die Kerberos Authentifizierung verwendet wird, mithilfe des **ktutil** Kommandos erstellen. Hierfür führen wir dieses Kommando aus:

```
sudo kinit sqltestuser@MADAFA.LOCAL
sudo kvno MSSQLSvc/UBUNTUSQL01.madafa.local:1433
```

```
simon@activedir01:~$ sudo kvno MSSQLSvc/ActiveDir01.madafa.local:1433
MSSQLSvc/ActiveDir01.madafa.local:1433@MADAFA.LOCAL: kvno = 2
```

Der von diesem Befehl zurückgegebene kvno-Wert wird nun im folgenden **ktutil** verwendet, um das **Key Tab File** zu erstellen. In unserem Fall ist der kvno Wert **2**!

Das **Key Tab File** können wir nun mit folgenden Kommando erstellen:

```
sudo ktutil
addent -password -p MSSQLSvc/activedir01.MADAFA.LOCAL:1433@MADAFA.LOCAL -k 2 -e
aes256-cts-hmac-sha1-96
addent -password -p MSSQLSvc/activedir01.MADAFA.LOCAL:1433@MADAFA.LOCAL -k 2 -e
rc4-hmac
wkt /var/opt/mssql/secrets/mssql.keytab
exit
```


Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612

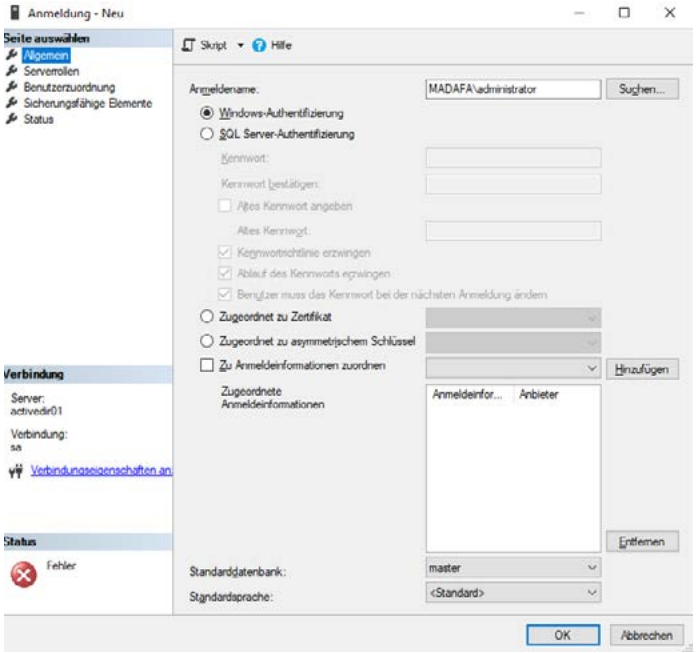
Anschließend müssen wir unser **Key Tab File** sichern und nur dem mssql Benutzer die Erlaubnis zum Lesen erteilen. Dies tun wir mit folgendem Kommando:

```
sudo chown mssql:mssql /var/opt/mssql/secrets/mssql.keytab
sudo chmod 400 /var/opt/mssql/secrets/mssql.keytab
```

Jetzt müssen wir nur noch die Konfiguration unseres SQL Servers ändern, damit wir das **Key Tab File** für die Kerberos Authentifizierung verwenden können. Nach der Konfiguration starten wir den SQL Server neu:

```
sudo /opt/mssql/bin/mssql-conf set network.kerberoskeytabfile /var/opt/mssql/
secrets/mssql.keytab
sudo systemctl restart mssql-server
```

Nun können wir mit dem letzten Schritt, dem Erstellen des Windows Login für unseren SQL Server, beginnen.



Der Windows Login

Hierfür loggen wir uns mithilfe des SQL Server Management Studio (SSMS) auf unserem SQL Server ein. SSMS führen wir hier auf einer Windows Maschine aus, die eine Verbindung zu unserem SQL Server auf dem Ubuntu Server herstellen kann. Wir erstellen anschließend einen Login mit Windows Authentifizierung.

Ein anderer Weg, den Windows Login zu erstellen, besteht darin, SQLCMD utility anzuwenden. Hierfür loggen wir uns auf unserem Ubuntu Server mit SSH ein und führen mit SQLCMD folgendes Kommando aus:

```
USE [master]
GO
CREATE LOGIN [MADAFA\administrator] FROM WINDOWS WITH DEFAULT_DATABASE=[master]
GO
ALTER SERVER ROLE [sysadmin] ADD MEMBER [MADAFA\administrator]
GO
```

Nachdem wir den Windows Login für unseren SQL Server erstellt haben, können wir auf jeder Windows Maschine, die eine Verbindung mit unserem Ubuntu Server herstellen kann, SSMS ausführen und uns mittels Windows Authentifizierung mit dem SQL Server verbinden.
Wir haben also erfolgreich eine Windows Authentifizierung für einen SQL Server auf einer Ubuntu Maschine erstellt.

Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612