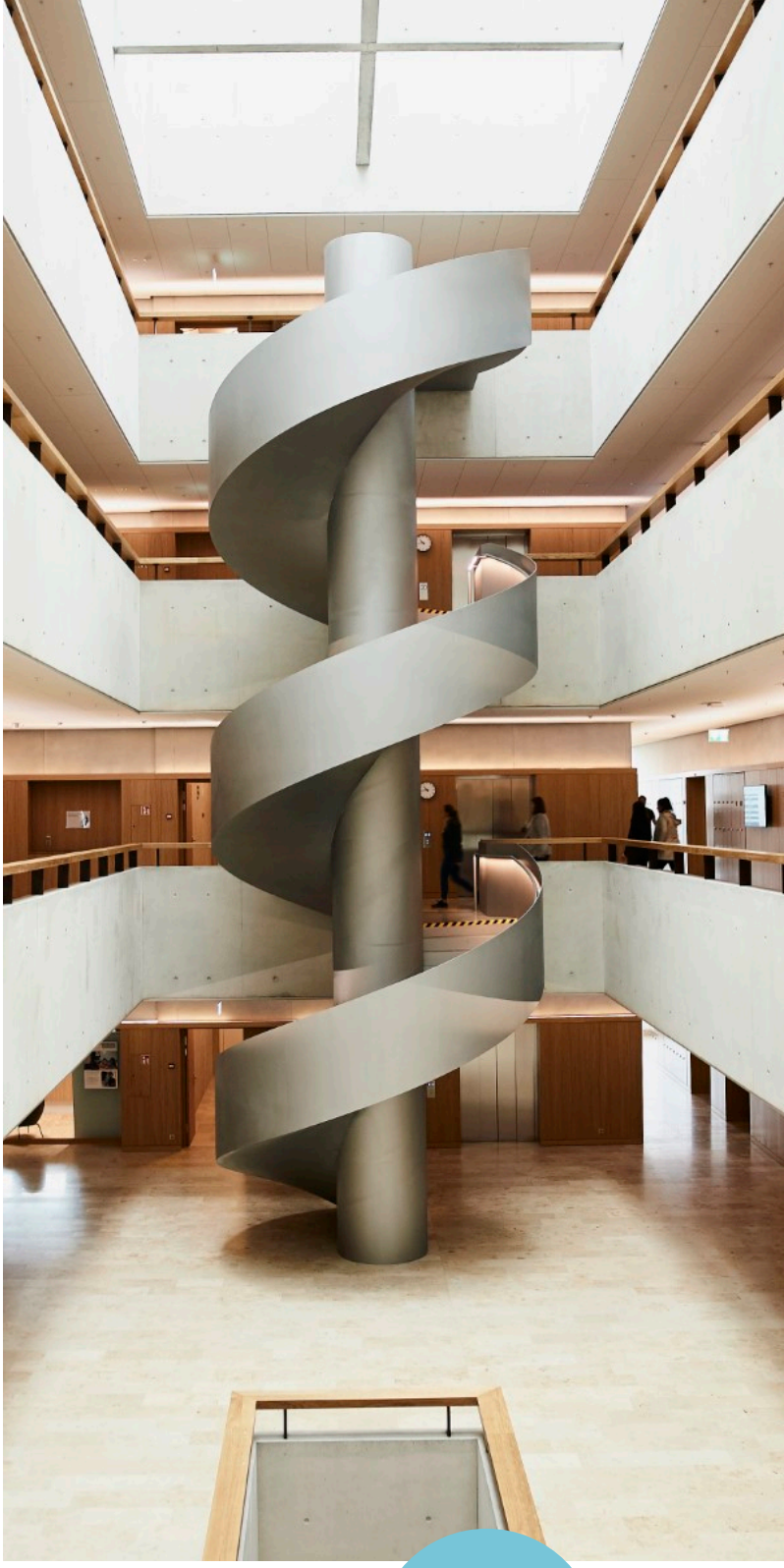
**Hochschule Luzern**
Modul I.BA_ISLAB



# Laborübung

## Access-Management Linux

Version
**1.0.17-2-g3dd8e56**

Dokument Version: 1.0.17-2-g3dd8e56#3dd8e56
Template Version: 0.0.2#4799e29
Letzte Änderung: 2023-10-11

N. Neher, T. Jösler, E. Sturzenegger, J. Hemmings, S. Renggli, R. Bolfing, J. Keiser

HSLU Hochschule Luzern

# I. Allgemeine Informationen

## Name:

Aurel Corti, David Jäggli

## Gruppe:

Gruppe 13

## Bemerkungen:

### Liste der Verfasser

| | |
|---|---|
| N. Neher | Erster Entwurf der Laborübung<br>Korrektur & Überprüfung der Laborübung<br>Finalisierung & Korrektur<br>Änderungen & Optimierung<br>Feinschliff & Korrekturen |
| T. Jösler | Feinschliff |
| E. Sturzenegger | Korrekturen |
| J. Hemmings | Korrekturen |
| S. Renggli | Korrekturen |
| R. Bolfing | Aktualisierung & Kapitel SSSD |
| J. Keiser | Erweiterung SSSD<br>Korrekturen |

### Copyright Informationen

# II. Inhaltsverzeichnis

# III. Vorwort

## Feedback

Mit Ihrer Mithilfe kann die Qualität des Versuches laufend den Bedürfnissen angepasst und verbessert werden.

Falls in diesem Versuchsablauf etwas nicht so funktioniert wie es beschrieben ist, melden Sie dies bitte direkt dem Laborpersonal oder erwähnen Sie es in Ihrem Laborbericht oder Protokoll. Behandeln Sie die zur Verfügung gestellten Geräte mit der entsprechenden Umsicht.

Bei Problemen wenden Sie sich bitte ebenfalls an das Laborpersonal.

## Legende

In den Versuchen gibt es Passagen, die mit den folgenden Boxen markiert sind. Diese sind wie folgt zu verstehen:

**Wichtig**

Dringend beachten. Was hier steht, unbedingt merken oder ausführen.

**Aufgabe III.1**

Beantworten und dokumentieren Sie die Antworten im Laborprotokoll.

**Hinweis**

Ergänzender Hinweis / Notiz / Hilfestellung.

**Information**

Weiterführende Informationen. Dies sind Informationen, die nicht zur Ausführung der Versuche benötigt werden, aber bekannt sein sollten.

**Story**

Hierbei wird die Geschichte vermittelt, die in den Versuch einleitet oder den Zweck des Versuches vorstellt.

**Zielsetzung**

Lernziele, die nach dem Bearbeiten des Kapitels erfüllt sein sollten.

**Erkenntnis**

Wichtige Erkenntnisse, die aus dem Versuch mitgenommen werden sollten.

# 1. Vorbereitung

Dieses Kapitel beschreibt die Vorbereitungsmassnahmen, die Sie vor Beginn des Laborversuches durchführen müssen.

## 1.1. Einleitung

Je grösser Unternehmen werden, umso wichtiger wird es, Identitäten und Berechtigungen von Benutzern sorgfältig zu verwalten. Grundsätzlich möchte man mit Identity & Access Management (IAM) ein effizientes Zugriffsmanagement mit dem Minimum der erforderlichen Berechtigungen umsetzen. Anders formuliert: Man möchte den Schutz von Informationen und Systemen – die Informationssicherheit, bestmöglich realisieren (Augsten Security Insider, 2010).

Dennoch stehen Unternehmen bei der Umsetzung eines IAMs immer wieder vor Herausforderungen. Sind Berechtigungskonzepte zu detailliert, steigt der Administrationsaufwand ins unermessliche. Wird zu grob zwischen Berechtigungsgruppen unterschieden, kann der Zugriffsschutz nicht gewährleistet werden und die Ressourcen sind nicht sicher. Im Windows-Umfeld nimmt hier vor allem das Microsoft Active Directory, ein Verzeichnisdienst, für die Verwaltung von Identitäten eine zentrale Rolle ein.

## 1.2. Theorie

Theorie wird im Laufe der Übung erklärt beziehungsweise von den Studenten selbst recherchiert.

## 1.3. Benötigte Mittel

Im Rahmen dieser Laborübungen werden keine expliziten Hardwareressourcen benötigt. Sämtliche Aufgaben werden auf den Laborgeräten bzw. auf virtuellen Servern und Maschinen durchgeführt. Bei Bedarf kann auch das eigene Geräte herangezogen werden.

## 1.4. Versuchsumgebung

**Wichtig**

Diese Durchführung wird auf neuer Infrastruktur durchgeführt, somit kann es zu Abweichungen kommen. Vor und während dem Bearbeiten der Laborübungen bitte Informationen des Laborpersonals beachten und den Discord-Kanal im Auge behalten. Das Laborpersonal wird mitteilen, sollte trotz der Tests etwas nicht wie in diesem Dokument beschrieben funktionieren.

Die Studierenden erhalten zu Beginn der Übung folgende Arbeitsumgebung, welche auf Basis von virtuellen Maschinen auf SWITCHengines umgesetzt ist:

### 1.4.1. Windows Server

Das Active Directory (AD) läuft unter Windows Server 2022 und ist als Domänencontroller (DC) für die Domäne «gXX.ckteck.com» eingerichtet (XX entspricht Ihrer Gruppennummer).

#### 1.4.2. Linux Client

Auf dem Linux Client ist ein Debian 11 (bullseye) als Betriebssystem installiert.

### 1.5. Bemerkungen / Rechtlicher Hinweis

Die vorliegenden Übungen werden als Partnerarbeiten geführt. Es ist daher notwendig, vorgängig Zweierteams zu bilden.

### 1.6. Zeitliche Aspekte

Die gesamte Übung verläuft über eine Semesterwoche und ist auf 3 Unterrichtslektionen ausgelegt. Je nach persönlichem Interesse oder entsprechendem Vorwissen kann dies auch über / unter die genannte Zeitangabe hinausgehen.

# 2. Access Management unter Linux

## 2.1. Linux Zugriffsrechte

Mithilfe von Zugriffsrechten (Permissions) wird geregelt, welche Benutzer und oder Gruppen den Inhalt eines Verzeichnisses lesen dürfen. Das Wissen um Zugriffsberechtigungen haben Sie schon in der vorherigen Übung, siehe Access Management Windows, erworben. Auch hier im Linux Access Management gilt, eine Berechtigung besteht immer aus zwei Komponenten:

- der Ressource, auf welche Berechtigungen vergeben werden
- die zu berechtigenden Operationen, welche für die Ressource freigegeben bzw. gesperrt werden.

Das Rechtesystem des Linux Dateisystems regelt, welche Rechte pro Datei und Verzeichnis definiert sind. Um diese Rechte zu ändern, können die Terminal-Befehle «**chmod**», «**chown**» sowie «**chgrp**» herangezogen werden. Diese Befehle werden sie im Folgenden genauer kennenlernen.

### 2.1.1. Darstellung Zugriffsrechte/Dateiberechtigungen

Unix/Linux unterscheidet 3 verschiedene Benutzerklassen, welche für Dateiberechtigungen definiert werden können:

- Eigentümer (user) - Berechtigungen des Besitzers der Datei
- Gruppe (group) – Berechtigungen der Personen, die zur berechtigten Gruppe gehören
- Sonstige (others) – Berechtigungen aller anderen Personen

Jede dieser Benutzerklasse wird wiederum in Lese-, Schreibe- und Ausführrecht unterteilt. Dadurch ergibt sich folgende Darstellung:

| - | user | group | others |
|---|---|---|---|
| Filetype | read (r) write (w) execute (x) | read (r) write (w) execute (x) | read (r) write (w) execute (x) |

Je nachdem ob eine Berechtigung für eine Datei oder ein Verzeichnis gesetzt wird, verhält sich diese Berechtigung anders.

| | Read | Write | Execute |
|---|---|---|---|
| **Einfache Datei** | Datei öffnen und Inhalt lesen | Datei modifizieren | Datei ausführen. Ist nur bei Dateien sinnvoll, die ausführbaren oder interpretierbaren Code enthalten |
| **Verzeichnis** | Verwendung des | Dateien zum | Anzeigen weitere |

| | Read | Write | Execute |
|---|---|---|---|
| | ls-Kommando, um Dateien im Verzeichnis anzuzeigen | Verzeichnis hinzufügen, umbenennen oder löschen | Informationen (ls-Kommando mit Optionen) + setzen des «current directory» mit dem Befehl cd |

Jede Datei ist sowohl einem Benutzer (einer UID) als auch einer Gruppe (GID) zugeordnet. UID und GID gehören zu den elementaren Verwaltungsinformationen von Dateien und Verzeichnissen.

Der UID (user identifier) ist eine Nummer, welche von Linux jedem Benutzer zugeordnet wird. Diese Nummer wird benutzt, um den Benutzer eindeutig zu identifizieren und um festzustellen, auf welche Ressourcen dieser zugreifen kann.

Gruppen werden in Linux über den GID (group identifier) identifiziert.

**Information**

Weitere Informationen zu UID und GID finden Sie unter folgenden Links:
*https://www.cyberciti.biz/faq/understanding-etcpasswd-file-format/*
*https://www.cyberciti.biz/faq/understanding-etcgroup-file/*

**Aufgabe 2.1**

Welche UID hat der root-User? Recherchieren Sie!

root hat die UID 0.

Wird auf eine Datei zugegriffen, werden UID und GID des zugreifenden Users/Prozesses mit jenen der Datei verglichen.

Verbinden Sie sich mit Ihrem zugeordneten Linux Client **via Remote Desktop Verbindung**. Verwenden Sie den User «labadmin». Die Login-Informationen Ihrer Gruppe wurden Ihnen zugesandt.

Öffnen Sie eine Terminal-Session. Nachdem Sie das Terminal geöffnet haben, geben Sie folgenden Befehl ein:

```
1 ls -l /etc/passwd
```

### Aufgabe 2.2

Um was für eine Datei handelt es sich bei der Datei /etc/passwd? Recherchieren Sie und dokumentieren Sie deren Zweck sowie Inhalte.

In /etc/passwd ist eine textbasierte Datenbank der User.
Inhalte:
1. Username:2. Passwort(nicht mehr):3. UID:4. GID:5. GECO: 6. Login Shell

### Aufgabe 2.3

Was bewirkt der Befehl **ls -l**? Recherchieren und erläutern Sie in kurzen Worten.

-l ist für Long Mode.
Das bedeutet, dass die Files und Directories mit mehr Angaben ausgegeben werden. Der Besitzer und die Gruppe werden ausgegeben, die Berechtigungen derer und die Bearbeitungs - Zeit und File Namen werden zu jedem File ausgegeben.

```
labadmin@islab-lc-01:/$ ls -l /etc/passwd
-rw-r--r-- 1 root root 2303 Aug 31 16:19 /etc/passwd
```

### Hinweis

Ist ein Recht nicht gesetzt, wird dies mit einem **-** gekennzeichnet. Bei dieser Darstellung der Dateiberechtigung handelt es sich um die Symbolische Notation.

Die ersten 10 Zeichen des Outputs stellen die Dateiberechtigung auf die Datei /etc/passwd dar. Das erste Zeichen des Outputs gibt an, um was für eine Datei es sich handelt.

Die folgenden **üblichen** Dateitypen werden unterschieden. Für weitere Dateitypen recherchieren Sie im Internet:

| **Dateityp** | **Symbol/Zeichen** |
|---|---|
| Einfache Datei | - |
| Verzeichnisse | d |
| Verweise | l |

### Aufgabe 2.4

Um was für einen Dateityp handelt es sich bei /etc/passwd?

Es ist eine einfache Datei.

**Aufgabe 2.5**

Wer ist der Besitzer der Datei und welche Rechte hat er?

Der besitzer ist root, er kann sie lesen und bearbeiten.

Öffnen Sie die Datei /etc/passwd und suchen Sie den Benutzer, mit dem Sie gerade eingeloggt sind.

**Aufgabe 2.6**

Wie lautet die UID & GID des Benutzers? Dokumentieren Sie!

UID : 0 , GID : 0

**Aufgabe 2.7**

Nehmen Sie an, ein neuer Mitarbeiter würde bei seinem Benutzeraccount fälschlicherweise dieselbe UID & GID wie ein gerade erst gekündigter Mitarbeiter zugeordnet bekommen. Welche gefährlichen Konsequenzen könnte dies mit sich bringen?

Dieser Benutzer wäre dann root und könnte somit das ganze System verwalten. Root hat meistens die Rechte alles, oder sicher fast alles zu bearbeiten.

Eine weitere Möglichkeit die UID & GID eines Benutzers herauszufinden, ist der Befehl «getent». Mit diesem Befehl können Sie die Benutzer einer unterstützten Userdatenbank auflisten.

```
1 getent passwd labadmin
2 getent passwd root
```

**Aufgabe 2.8**

Vergleichen Sie root und labadmin – was müsste geändert werden, damit der Benutzer labadmin auch root-Rechte hat?

sudo <cmd>
[enter labadmin passwd]
für temporäre root Rechte

Permant root Rechte: GUI von labadmin auf 0 setzen.

Erstellen Sie in einem nächsten Schritt einen neuen User mit dem Benutzernamen «Bob». Geben Sie hierzu folgenden Befehl im Terminal ein:

```
adduser bob
```

**Aufgabe 2.9**

Welche Fehlermeldung erhalten Sie? Können Sie sich erklären, wieso Sie diese erhalten?

Authentication failure
Man braucht root Rechte.

```
sudo adduser bob
```

**Aufgabe 2.10**

Wie lautet die UID & GID des neu erstellten Users? Dokumentieren Sie.

1003:1001

### 2.1.2. Ändern von Berechtigungen mit chmod

Nachdem Sie nun kennengelernt haben, welche Benutzerklassen es gibt und wie Berechtigungen dargestellt werden, werden Sie in einem nächsten Schritt die Änderung von Benutzerrechten vornehmen.

Erstellen Sie deshalb zuerst einen Ordner vom Home-Verzeichnis aus auf dem Desktop mit dem Namen «Dokumente» und überprüfen Sie die Berechtigungen des Ordners.

```
mkdir Desktop/Dokumente
ls -l Desktop
```

Folgende Ausgabe sollte nun für Sie ersichtlich sein:

```
drwxr-xr-x 2 labadmin labadmin 4096 Oct 11 14:04 Dokumente
```

Wie sie sehen, hat der Besitzer (**owner**) alle Berechtigungen, während **group** und **others** nur Lese- bzw. Ausführberechtigungen besitzen. Um dies zu testen, loggen Sie sich nun mit dem neu erstellten User «bob» ein und versuchen im Verzeichnis «Dokumente» eine neue Datei zu erstellen.

```
su bob
cd Desktop/Dokumente
nano newfile
```

**Aufgabe 2.11**

Wofür steht der su-Befehl? Recherchieren Sie!

su steht für substitute user. Er kann genutzt werden, um den Benutzer zu wechseln.

**Aufgabe 2.12**

Welche Fehlermeldung erhalten Sie, wenn Sie die mit dem Editor nano erstellte Datei speichern wollen?

Permission denied

Nachdem Sie sich davon überzeugt haben, dass Bob keine Schreibberechtigung auf diesem Verzeichnis hat, werden Sie Ihm im nächsten Schritt welche erteilen.

Hierfür wird der Befehl «**chmod**» verwendet.

**Information**

Detaillierte Information zum Befehl **chmod** erhalten Sie über das Manual, indem Sie im Terminal den Befehl `man chmod` eingeben.

Das Kommando **chmod** kann im sogenannten symbolischen Modus oder im absoluten Modus verwendet werden. Während der symbolische Modus für die Änderung von einzelnen Berechtigungen verwendet wird, wird der absolute Modus für die Neudefinition sämtlicher Berechtigungen verwendet. Beim absoluten Modus werden die Berechtigungen auf eine vierstellige Oktalzahl abgebildet:

| 1. Stelle | 2. Stelle | 3. Stelle | 4. Stelle |
|---|---|---|---|
| Besondere Modi | Owner | Group | Others |
| SUID - SGID - t-Bit | r - w - x | r - w - x | r - w - x |
| 4 - 2 - 1 | 4 - 2 - 1 | 4 - 2 - 1 | 4 - 2 - 1 |

In der vierten Zeile sind die einzelnen Wertigkeiten der Rechte-Bits abgebildet. Dabei hat die Read-Permission einen Wert von 4, die Write-Permission eine Wert von 2 und die Execute-Permission einen Wert von 1. Um jetzt eine neue Berechtigung für die einzelnen Benutzerklassen zu definieren, müssen die Wertigkeiten der Rechte-Bits für jede Klasse entsprechend addiert werden.

Beispiel: Der Datei `Testat` sollen die Berechtigungen `rw-r--r--` gegeben werden. Dies könnte dann mit folgendem Befehl erreicht werden: `chmod 644 Testat`

Wie vorher erwähnt, werden Sie nun auf dem Verzeichnis «Dokumente» allen anderen Benutzern (others) neben der Lese- und Ausführberechtigung auch die Schreibberechtigung erteilen.

**Aufgabe 2.13**

Wie lautet die gesuchte Zahl für die Definition dieser Berechtigungen?

chmod 757 Dokumente

Wechseln zu zuerst wieder mit dem Befehl `exit` zurück zu Ihrem vorherigen Benutzer **labadmin**, dem Ersteller und Eigentümer des Verzeichnisses.

**Hinweis**

Um die Berechtigungen ändern zu können, müssen Sie mit dem Besitzer bzw. dem Superuser eingeloggt sein.

Geben Sie nun folgenden Befehl für die Definition der neuen Berechtigungen ein. **Ersetzen Sie dabei die *** mit der von Ihnen gefundenen Zahl.**

```
1 chmod *** Desktop/Dokumente
```

Wenn alles geklappt hat, sollten die Dateiberechtigungen der Datei nun folgendermassen aussehen:

```
drwxr-xrwx 2 labadmin labadmin 4096 Oct 11 14:04 Dokumente
```

Weiter geht es nun mit den Besonderen Modi SUID, SGID und t-Bit.

**Aufgabe 2.14**

Recherchieren Sie, welche Bedeutungen SUID, SGID und t-Bit haben und dokumentieren Sie dies hier. Gehen Sie hier auch auf den Unterschied ein, welchen Einfluss diese für Dateien bzw. Verzeichnisse haben.

SUID und SGID sind sticky bits. Sie könen auf Verzeichnisse gesetzt werden. Je nach dem, welches gesetzt ist werden alle Dateien, mit den Berechtigungen eines bestimmten Users (z.B. root) bearbeitet/ausgeführt.

t-Bit: Die Dateien in diesem Verzeichnis können dann nur vom Owner oder Gruppe der Datei gelöscht werden.

Wechseln Sie hierzu in das root-Verzeichnis Ihres Linux-Rechners und sehen Sie sich die Berechtigungen, welche auf dem Verzeichnis /tmp zugewiesen sind, an. Im Verzeichnis /tmp werden von einigen Programmen temporäre Dateien zur Zwischenspeicherung von Laufzeitdaten angelegt.

```
1 cd /
2 ls -l
```

Folgende Ausgabe (Auszug) sollte für Sie ersichtlich sein:

```
dr-xr-xr-x  13 root root     0 Jul  8 10:41 sys
drwxrwxrwt  15 root root  4096 Sep  2 11:48 tmp
drwxr-xr-x  14 root root  4096 Jun 21 14:50 usr
drwxr-xr-x  12 root root  4096 Jul  4 16:33 var
```

Sie sehen, dass beim Verzeichnis /`tmp` bei der Benutzerklasse others an der Stelle des Rechte-Bit execute ein `t` vermerkt ist. Dieses `t` steht für ein gesetztes Sticky-Bit.

**Aufgabe 2.15**

Weshalb macht das gesetzte Sticky-Bits beim Verzeichnis /tmp Sinn? Begründen Sie anhand des Zwecks des Verzeichnisses.

/tmp ist world-writable Ordner, heisst jeder und alles kann dort Dateien bearbeiten/erstellen. Es macht jedoch Sinn das Löschen der Dateien von anderen Applikationen/User zu verweigern, weil das zu Problemen führen könnte.

Erstellen Sie eine neue Datei im Verzeichnis `Desktop`/`Dokumente` des Benutzer labadmin mit dem Namen «financedata.txt». Damit nur Sie diese Datei lesen können, vergeben Sie folgende Berechtigung:

```
1 chmod 600 financedata.txt
```

Loggen Sie sich wieder mit Bob ein und versuchen Sie die Datei zu lesen, indem Sie sie mit dem nano-Editor öffnen oder mittels Befehl `cat` im Terminal ausgeben.

```
1 nano financedata.txt
2 cat financedata.txt
```

Sie werden die Meldung erhalten, dass Sie nicht die benötigten Berechtigungen dafür haben – Gratuliere, ihre Bitcoins sind jetzt sicher :)

Sie werden jetzt an dieser Stelle einen kurzen Exkurs machen und stellen sich folgendes Szenario vor:

Bob benötigt nun doch Zugriff auf die Datei – sie wollen aber nicht die Berechtigung dafür ändern. Kurzerhand entscheiden Sie sich dafür, das Programm, mit dem die Datei gelesen bzw. bearbeitet wird, mit einem SUID-Bit anzureichern, sodass, wenn Bob die Datei mit diesem Programm öffnet, auf jeden Fall genügend Berechtigungen besitzt – hier geht es um den Editor ***nano***.

Wechseln Sie wieder zurück auf den Benutzer **labadmin** und geben Sie folgenden Befehl im Terminal ein:

```
1 ls -l /bin/nano
```

Die Berechtigungen sollten folgendermassen aussehen:

```
-rwxr-xr-x 1 root root 348816 Feb  8  2021 /bin/nano
```

Sie werden nun das SUID-Bit mit folgendem Befehl setzen:

```
1 sudo chmod u+s /bin/nano
```

**Aufgabe 2.16**

Wofür steht das u+s? Tipp: Es handelt sich hier um eine andere Schreibweise (Verändern von Berechtigungen mittels symbolischem Modus)

u+s steht für das Ergänzen eines User Sticky bits (SUID)

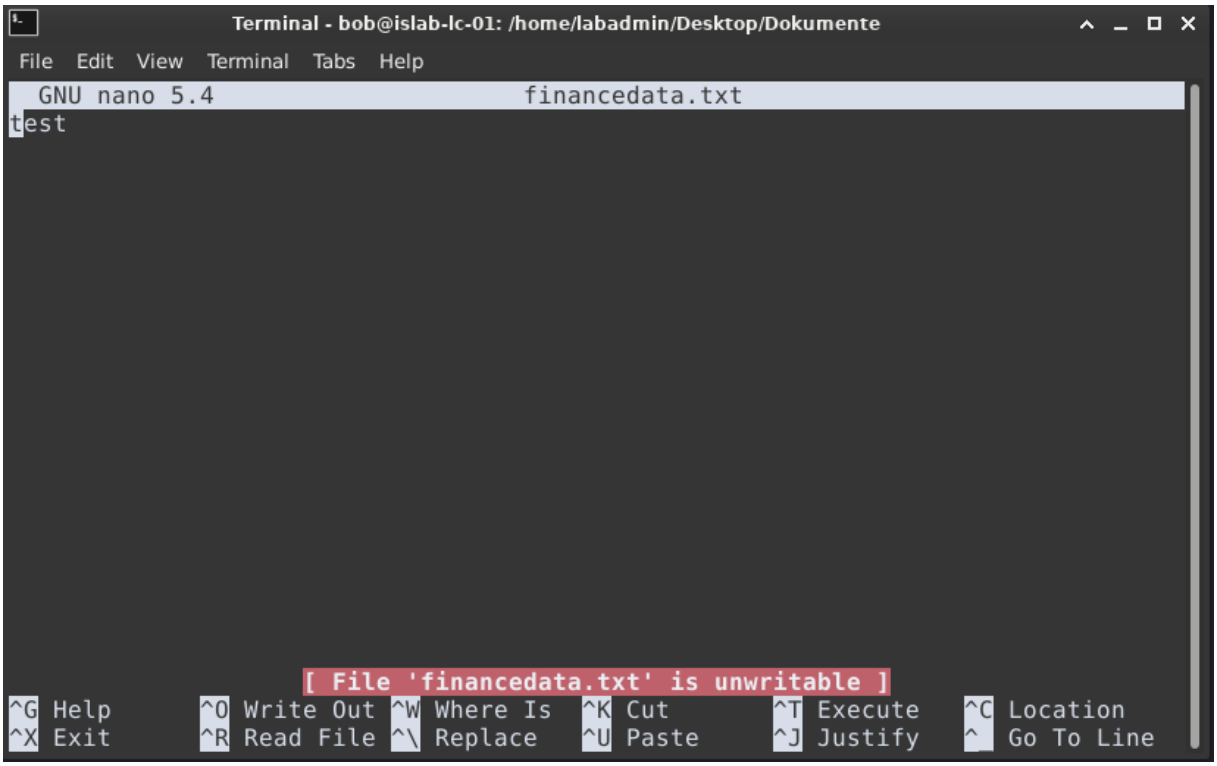
Nachdem Sie diesen Befehl ausgeführt haben, wurde das SUID-Bit gesetzt. Kontrollieren Sie die neu gesetzten Berechtigungen:

```
1 ls -l /bin/nano
```

Folgende Ausgabe sollte ersichtlich sein:

```
-rwsr-xr-x 1 root root 348816 Feb  8  2021 /bin/nano
```

Loggen Sie sich nun wieder mit dem User Bob ein und versuchen Sie, ob sie mithilfe des nano-Editors nun Zugriff auf die Finanzdaten erhalten.




Gratuliere – Sie haben nun die benötigten Berechtigungen mit dieser Datei zu arbeiten. Lassen Sie sich von der Meldung nicht täuschen. Sie haben trotzdem das Recht das File zu bearbeiten.

**HALT STOP – Sie haben nicht nur gerade die benötigten Berechtigungen erteilt um mit dieser Datei arbeiten zu können, sondern sich selbst auch noch eine Backdoor platziert.**

Durch das Setzen des SUID-Bits auf dem nano-editor stellen Sie sicher, dass das Programm mit den Rechten des Besitzers, im Falle von nano mit denen des SUPERUSERS, ausgeführt wird. Somit kann ein Hacker, sofern er sich mit einem nicht-privilegierten User anmelden kann, immer wieder auf root-Ebene agieren und beispielsweise alle Passwörter neu setzen oder öffentliche SSH-Keys über den Befehl setzen.

Weitere Informationen dazu finden Sie unter https://gist.github.com/dergachev/7916152

```
cat /etc/shadow //Permissions denied
nano /etc/shadow //SUCCESS
nano /root/.ssh/authorized_keys //SUCCESS
```

**Fazit**: *SUID-Bits zu setzen kann sehr gefährlich sein und muss mit äusserster Vorsicht und Wachsamkeit durchgeführt werden. Ein solcher Fehler kann zu verheerenden Sicherheitslücken führen.*

Machen Sie das ganze Rückgängig, indem Sie sich wieder mit dem Benutzer **labadmin** anmelden & folgendes Kommando ausführen:

```
sudo chmod u-s /bin/nano
```

### 2.1.3. Ändern des Dateieigentümers mit chown

Nachdem Sie nun wissen, wie Berechtigungen auf Dateien und Verzeichnissen verändert werden können und dass dies mit grösster Sorgfalt passieren sollte, werden Sie nun kennenlernen, wie sie den Besitzer bzw. eine berechtigte Gruppe ändern können.

**Hinweis**

Der Besitzer oder die berechtigte Gruppe kann nur vom Superuser abgeändert werden.

Wechseln sie wieder auf den Desktop Ihres Benutzers. Sie werden nun den Eigentürmer der von Ihnen erstellten Datei «financedata.txt» ändern.

```
cd /home/labadmin/Desktop/Dokumente
```

Der Besitzer eines Verzeichnisses oder einer Datei kann mit dem Befehl «**chown**» geändert werden. Dabei können die Eigentümer bzw. Gruppen entweder mit deren Namen oder UID bzw. GID angegeben werden.

Geben Sie folgenden Befehl ein, um den Besitzer der Datei auf den User «Bob» zu ändern. Somit hat dieser nachher auch genügend Rechte mit der Datei zu arbeiten, ohne gleich das ganze System kompromittieren zu müssen ;-)

```
sudo chown bob financedata.txt
```

Kontrollieren Sie, ob Bob nun der neue rechtmässige Besitzer Ihrer Bitcoins ist, indem Sie die Berechtigungen mit dem Befehl `ls -l` ausgeben.

**Aufgabe 2.17**

Kann Bob nun die Datei bearbeiten? Was fällt Ihnen noch an der Ausgabe auf?

ja, der Benutzer wurde gewechselt, die gruppe ist aber immer noch labadmin

In einem nächsten Schritt werden Sie nun noch die berechtigte Gruppe ändern. Dies können Sie entweder über chgrp (siehe nächstes Kapitel) oder auch mit chown erledigen.

```
sudo chown bob:bob financedata.txt
```

**Hinweis**

Jeder User gehört seiner eigenen Gruppe an – deshalb lassen sich für alle User auch Gruppen mit dem gleichen Namen finden. Nach Unix-Philosophie gehört jeder User zu einer oder mehreren Gruppen – Zugriffsrechte werden an Gruppen vergeben. Bei einer Standardinstallation ist die eigene Gruppe (selber Name) dabei die primäre Gruppe.

#### 2.1.4. Arbeiten mit Gruppen über chgrp

Um die Gruppenzugehörigkeit von Dateien und Verzeichnissen zu ändern, kann der Befehl «**chgrp**» herangezogen werden. Der Befehl kann nur angewendet werden, wenn der ausführende Benutzer Eigentümer der Datei oder Superuser ist. Gleichzeitig stehen auch nur jene Gruppen zur Verfügung, denen man selbst angehört.

Um die Gruppenzugehörigkeit zu ändern wird folgender Befehl verwendet:

```
chgrp gruppeX testfile
```

### 2.2. Integration eines Linux Clients in eine Windows Domäne

Nachdem Sie nun die Basics zu Unix-Berechtigungen kennengelernt haben, werden Sie nun in diesem Teil den Linux Client in die bestehende Windows Domäne aufnehmen. Heterogene Systemlandschaften sind heute grundsätzlich *State-of-the-Art*, weshalb diesem Teil eine hohe Wichtigkeit zugeschrieben werden kann.

#### 2.2.1. Früher: Integration mittels Samba

Bis anhin war die AD-Integration mittels Samba weit verbreitet. Samba ist ein Open-Source-Projekt, welches auf die Integration zwischen Windows- und Linux-Umgebungen abzielt. Samba enthält Komponenten, die Linux-Computern Zugriff auf Datei- und Druckdienste von Windows Geräten ermöglichen sowie Linux-basierte Dienste bereitstellen, die Windows NT 4.0-Domain Controllers emulieren. Mithilfe der Samba-Clientkomponenten können Linux-Computer Windows-Authentifizierungsdienste nutzen, die von Windows NT- und Active Directory Domain Controller zur Verfügung gestellt werden. Dazu wurde der Dienst Winbind verwendet. Winbind fungiert als Proxy für die Kommunikation zwischen dem Authentisierungsdienst von Linux (PAM) und Active Directory. Winbind verwendet Kerberos zur Authentifizierung am Active Directory und LDAP, um Benutzer- und Gruppeninformationen abzurufen. Winbind bietet darüber hinaus zusätzliche Dienste, z.B. die Möglichkeit der Lokalisierung von DCs sowie die Möglichkeit, Active Directory-Kennwörter mit Hilfe von RPCs zurückzusetzen.

#### 2.2.2. Heute: Integration mittels SSSD

SSSD steht für System Security Services Daemon. SSSD wird unter Linux für verschiedene zentrale IAM (Identity and Access Management) Lösungen verwendet. Unteranderem für FreeIPA, 389 Directory Server, Microsoft Active Directory, OpenLDAP und weitere Verzeichnisdienste.

SSSD beinhaltet eine Reihe von Daemons, die zur Verwaltung des Zugriffs auf Remote-Verzeichnisdienste und Authentifizierungsmechanismen verwendet werden. SSSD wurde durch das Open-Source Softwareprojekt «FreeIPA»

ins Leben gerufen. Die Entwicklung begann bereits im Jahr 2009. Die erste stabile Version (2.4.0) wurde am 12. Oktober 2020 veröffentlicht.

Heute wird SSSD als «Best Practice» Variante empfohlen, um Linux Clients in eine Windows-Domäne zu integrieren.

**Information**

Mehr Informationen über das «FreeIPA» Projekt können Sie auf deren Website nachlesen: https://www.freeipa.org/page/About

### 2.2.3. Wie funktioniert SSSD

Das lokale System, in unserem Fall der Linux Host, fungiert als SSSD Client und erlaubt uns eine Verbindung zu einem externen Backend-System, dem sogenannten Provider herzustellen. In unserem Fall ist dies das Windows Active-Directory.

SSSD nutzt dabei zwei Phasen:

1. Der SSSD Client verbindet sich zu einem Remote Provider und erhält die Identitäts- und Authentifizierungsinformationen.
2. Die erhaltenen Authentifizierungsinformationen wie Benutzer und Anmeldeinformationen werden in einem lokalen Cache gespeichert.

SSSD erstellt keine Benutzerkontos auf dem lokalen System. Für die Remote-Benutzer ist es möglich ein Home-Verzeichnis zu konfigurieren. Das Home-Verzeichnis wird nach dem Abmelden des Remote-Benutzers nicht gelöscht.

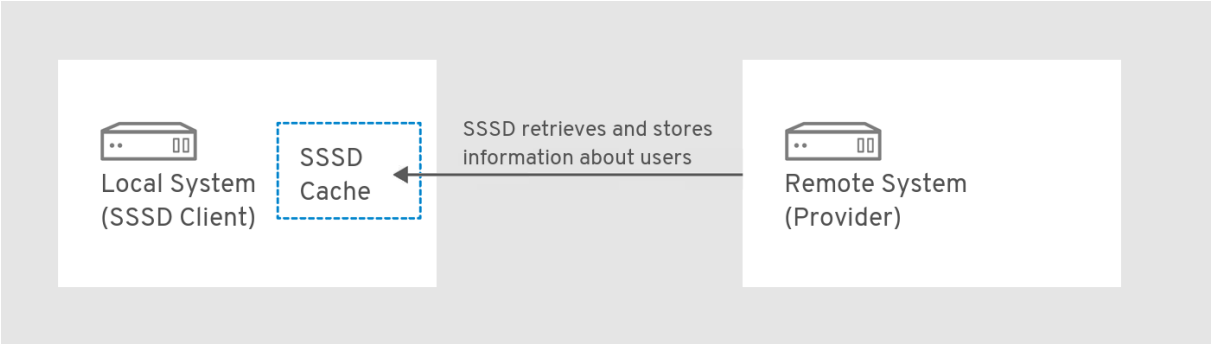



**Abbildung 1:** How SSSD works (Quelle: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/configuring_authentication_and_authorization_in_rhel/understanding-sssd-and-its-benefits_configuring-authentication-and-authorization-in-rhel )

### 2.2.4. Kontrolle DNS-Auflösung

Um den Linux-Client mit der Windows Active-Directory Umgebung zu verbinden, muss die Auflösung des Domänen-Namens korrekt funktionieren. Hierzu kontrollieren wir den primären DNS-Server auf dem Linux Client.

Nutzen Sie zur Kontrolle das integrierte Tool «nslookup».

```
1 nslookup
2 > server
```

```
labadmin@islab-lc-01:/$ nslookup
> server
Default server: 10.177.25.1
Address: 10.177.25.1#53
```

Vergleichen Sie die IP-Adresse des Eintrags «Default server», mit der IP-Adresse des Domänencontrollers Ihrer Gruppe. Die IP-Adressen sollten übereinstimmen.

Sie finden den Namen Ihres Windows Servers (Domänencontroller) in der ISLAB Hostliste.
islab-ws-**XX**.zh.switchengines.ch (**XX** = Gruppennummer)

### 2.2.5. Installation und Konfiguration von SSSD

Installieren Sie alle benötigten Pakete für die Integration ins Active-Directory. Geben Sie hierzu folgenden Befehl im Terminal ein:

```
1 sudo DEBIAN_FRONTEND=noninteractive apt-get install sssd-ad sssd-tools
    realmd adcli krb5-user
```

Damit uns das integrierte Sicherheitsfeature «Apparmor» nicht stört, müssen wir zuerst «sssd» aus Apparmor ausschliessen. Apparmor wird in einem späteren Versuch und im Verlaufe des Studiums in einem weiteren Modul tiefer behandelt.

Verschieben Sie dazu folgende Datei in den Apparmor «disabled» Ordner. Dazu können Sie folgenden Befehl ausführen:

```
1 sudo mv /etc/apparmor.d/usr.sbin.sssd /etc/apparmor.d/disable/.
```

In einem nächsten Schritt überprüfen wir, ob der Domänencontroller erreichbar ist – geben Sie hier den FQDN Ihres Domänencontrollers an:

```
1 ping FQDN
```

**Aufgabe 2.18**

Was wird unter dem FQDN verstanden? Wie lautet der FQDN Ihres Domänencontrollers?

Fully Qualified Domain Name
islab-ws-13.zh.switchengines.ch

Das installierte Paket «realmd» beinhaltet den Befehl «realm» um den Linux-Client mit der Domäne zu verbinden und erstellt die sssd Konfiguration.

Zuerst überprüfen wir jedoch nochmals die Domäne. Beachten Sie dabei, dass Sie alle **XX** im grauen Kasten mit Ihrer Gruppennummer ersetzen.

```
1 sudo realm -v discover gXX.ckteck.com
```

**Aufgabe 2.19**

Welche Auflösung wird mit diesem Befehl primär überprüft? Was wird unter type angegeben?

Besteht eine Verbindung zum Windows Server
kerberos

Standartmässig verwendet Kerberos reverse DNS-Resolution um Hostnames in Service Principal Names (SPNs) umzuwandeln. Der über die reverse DNS-Abfrage erhaltene Hostname muss einer FQDN entsprechen, die auf der AD Domäne basiert der beigetreten werden soll, in diesem Fall also islab-ws-XX.gXX.ckteck.com. Ein reverse DNS Lookup auf die IP des Domänencontrollers (z.B. mit host IP-Adresse) liefert allerdings die FQDN islab-ws-XX.zh.switchengines.ch zurück. Damit der Beitritt des Linux Clients zur Domäne funktioniert, bearbeiten Sie die Konfigurationsdatei /etc/krb5.conf.

```
1 sudo nano /etc/krb5.conf
```

Ergänzen Sie in der Sektion [libdefaults] die folgende Zeile und speichern Sie die Änderungen.

```
1 rdns = false
```

Verbinden Sie den Linux-Client mit der Domäne. Dazu verwenden Sie folgenden Befehl.
Damit wir ein Resultat sehen, verwenden wir die Option -v.

```
1 sudo realm join -U labadmin gXX.ckteck.com -v
```

Wenn Ihr Linux-Client erfolgreich in Ihre Domäne aufgenommen wurde, hat «realm» für Sie die SSSD-Konfigurationsdatei erstellt. Schauen Sie sich die Konfiguration an.

```
1 sudo cat /etc/sssd/sssd.conf
```

**Aufgabe 2.20**

Was könnten die folgenden zwei Zeilen bedeuten? Erläutern Sie.

```
krb5_store_password_if_offline = True
cache_credentials = True
```

Hinweis:
https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/identity_management_guide/kerberos-pwd-cache
https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/deployment_guide/sssd-cache-cred

Die password if offline option verhindert, dass Benutzer blockiert werden, falls sie sich einloggen wollen, falls der service offline ist.
Die zweite Option Cached die Credentials, so dass der Service nicht bei jedem Zugriff auf den Auth Server zugreifen muss. Das verhindert zu viel Traffic und ermöglicht Zugriff, falls der Auth Server für kuze Zeit offline ist.

### 2.2.6. Überprüfen der Verbindung zum Active-Directory Server

Moment der Wahrheit. Mittels «getent» können Sie die Verbindung zum Active-Directory überprüfen. Verwenden Sie den Benutzer «sysing01» der nur im Windows Active-Directory erfasst wurde.

```
1 getent passwd sysing01@gXX.ckteck.com
```

So sollte die Ausgabe in etwa aussehen:

```
sysing01@g01dev.ckteck.com:*:1698201107:1698200513:sysing01:/home/sysing01@g01dev.ckteck.com:/bin/bash
```

**Aufgabe 2.21**

Was bedeuten die zwei gelb markierten Zahlen und wie werden diese erstellt?
https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/windows_integration_guide/sssd-integration-intro

Das sind gemappte UIDs und GIDs für AD Benutzer. Diese werden vom Service errechnet, um einen Abgleich zwischen dem AD Server und dem Gerät zu ermöglichen.

Sie können sich jetzt mit dem Windows-Benutzer «sysing01» anmelden. Dazu können Sie den Befehl «login» verwenden.

```
1 sudo login sysing01@gXX.ckteck.com
```

### 2.2.7. Verwaltung von Berechtigungen für Domänen Benutzer

In der Standardkonfiguration von SSSD, wird die Zugriffskontrolle über die Domäne geregelt. Dieses Standardverhalten bewirkt, dass alle Domänen-Benutzer automatisch Zugriff auf das Linux System erhalten.

Möchten wir die Berechtigungen auf der Client-Seite einschränken, so müssen wir Regeln auf dem Client-System implementieren.

In unserem Beispiel möchten wir nur gewissen Active-Directory Gruppen Zugriff auf das Linux System erteilen (Whitelisting). Anstatt einzelnen AD-Accounts oder AD-Gruppen die Berechtigung zu entziehen (Blacklisting).

**Aufgabe 2.22**

Welche Vorteile bietet uns die Strategie, nur einzelnen AD-Gruppen den Zugriff auf das Linux System zu erlauben?

Neue Benutzer müssen so in der richtigen Gruppe sein, um Zugriff zu erhalten, und müssen nicht geblacklisted werden.
Es ist wichtig, dass nicht einfach alle Zugriff auf das System haben, da das Defense in Depth stärkt.

Melden Sie sich mit dem Benutzer «sysing01» ab. Sie können dazu den Befehl «exit» verwenden.

Deaktivieren Sie den lokalen Zugriff für alle Benutzer aus der Windows Active-Directory Domäne mittels folgenden Befehles:

```
1 sudo realm deny --all
```

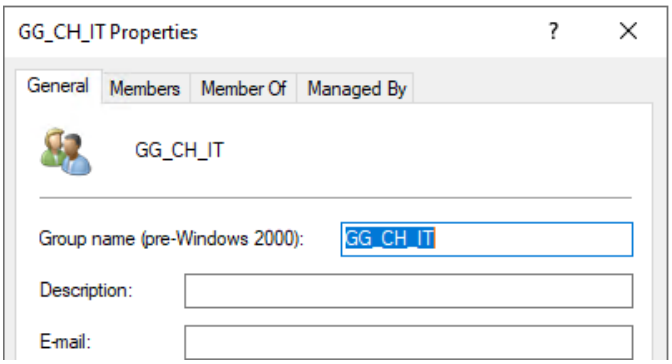
Versuchen Sie sich nochmals mittels «sysing01» anzumelden. Der Zugriff sollte jetzt nicht mehr funktionieren.

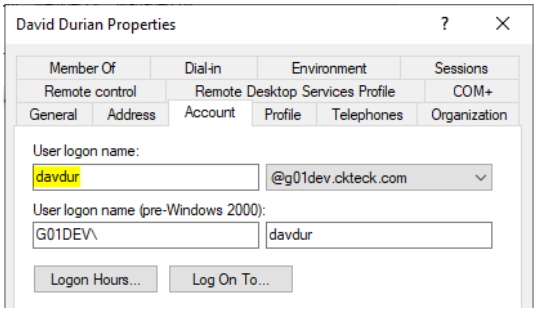
```
labadmin@islab-lc-dev-05:~/Desktop$ sudo login sysing01@g05dev.ckteck.com
Password:

Permission denied
```

Letzte Woche haben Sie im Active-Directory zusätzliche Gruppen angelegt.
(Kapitel 3.3.1 Gruppe erstellen von der ISLAB_Anleitung_Laborübung_Access_Management_Windows)

Jetzt möchten Sie nur die IT-Abteilung auf die Linux-Clients berechtigen.

Kontrollieren Sie auf Ihrem Windows Server im Active-Directory den korrekten Gruppennamen und überprüfen Sie den korrekten Loginname des Benutzers.
Existiert die Gruppe nicht, können Sie auch eine neue erstellen und einen Benutzer hinzufügen.




Erteilen Sie jetzt auf dem Linux Host der IT-Abteilung (AD-Gruppe) die Berechtigung sich am System anzumelden.

```
1 sudo realm permit -g "gXX.ckteck.com\GG_CH_IT"
```

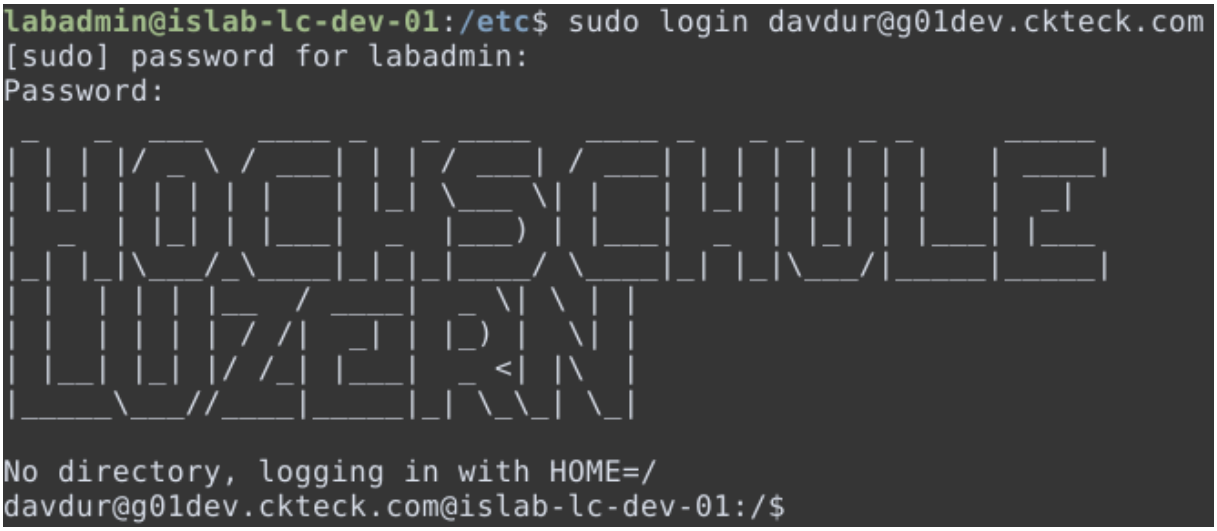
**Hinweis**

Hier erlauben wir den Zugriff für die Mitglieder einer globalen Gruppe und nicht einer Domänen lokalen Gruppe. Somit erfolgt die Vergabe der Zugriffsberechtigung nicht nach IGDLA, wie wir es in der vorangegangenen Übung gelernt haben. Der einfachheitshalber verwenden wir hier dennoch die globale Gruppe.

Jetzt sollten Sie sich mit dem Benutzer der Gruppe anmelden können.

Versuchen Sie es aus:

```
1 sudo login davdur@gXX.ckteck.com
```




Die Anmeldung als Domänen-User funktioniert nicht nur lokal über den Befehl `login`, sondern auch über das Netzwerk mit SSH.

**2.2.7.1. Home-Verzeichnis und PAM** Für den bei der Anmeldung verwendeten Domänenbenutzer existiert allerdings noch kein Home-Verzeichnis, worauf die Meldung «No directory, logging in with HOME=/» nach der Anmeldung hinweist. Standardmässig ruft SSSD das Format des Home-Verzeichnisses vom AD-Identitätsanbieter ab. Da wir im AD kein Home-Directory festgelegt haben, wird das Fallback-Verzeichnis verwendet, welches in der `[domain]`-Sektion in der Datei `/etc/sssd/sssd.conf` definiert ist.

Damit das Home-Verzeichnis erfolgreich erstellt werden kann, ist noch eine Anpassung an der PAM-Konfiguration notwendig. PAM steht für Pluggable Authentication Modules und ist eine Sammlung von Libraries, welches die Anwendungen (login, ssh, ftp, etc.) von den Mechanismen zur Benutzer-Authentisierung trennt. Meist ist ein System vom Netzwerk aus über verschiedene Zugänge erreichbar. PAM dient dazu, dass nicht jedes Zugangsprogramm die Authentisierung z.B. über Kerberos implementieren müssen. Mit PAM kann für jede Anwendung festgelegt werden, welche Sicherheitsmodule wie durchlaufen werden.

Falls noch nicht geschehen, kehren Sie zum User labadmin zurück. Bearbeiten Sie die Datei `/etc/pam.d/common-session` und fügen Sie am Ende der Datei die folgende Zeile ein.

```
1 session required pam_mkhomedir.so skel=/etc/skel/ umask=0077
```

**Aufgabe 2.23**

Wer hat welche Rechte auf dem erstellten Home-Verzeichnis?

user hat alle Rechte alle Anderen haben keine Rechte.

**Hinweis**

Nach einem Neustart des Linux Clients kann sich jeder Domänen User ebenfalls über RDP anmelden, sofern er der Gruppe `GG_CH_IT` angehört.

**2.2.7.2. Standard Domänen-Suffix** Derzeit ist für die Anmeldung mit einem Domänen-Benutzer die Angabe der Domäne als Suffix an den Benutzernamen notwendig (z.B. ddurian@gXX.ckteck.com). Damit die Anmeldung als Domänen User auch ohne die Angabe der Domäne funktioniert, wie bei lokalen Benutzern, legen Sie die default Domäne mit angepasster Gruppennummer in der `[sssd]`-Sektion der Datei `/etc/sssd/sssd.conf` fest.

```
1 default_domain_suffix = gXX.ckteck.com
```

Damit die Änderungen wirksam werden, muss SSSD neugestartet werden.

```
1 sudo systemctl restart sssd
```

Aufgabe 2.24

Wenn ein Benutzer mit identischem Anmeldenamen sowohl lokal als auch im Active Directory existiert, welcher von beiden wird bei der Anmeldung verwendet? Hinweis: Sehen Sie sich die Datei `/etc/nsswitch.conf` an, verändern Sie jedoch nichts daran.

Der vom Active Directory

### 2.2.8. Verwalten der sudo-Berechtigung

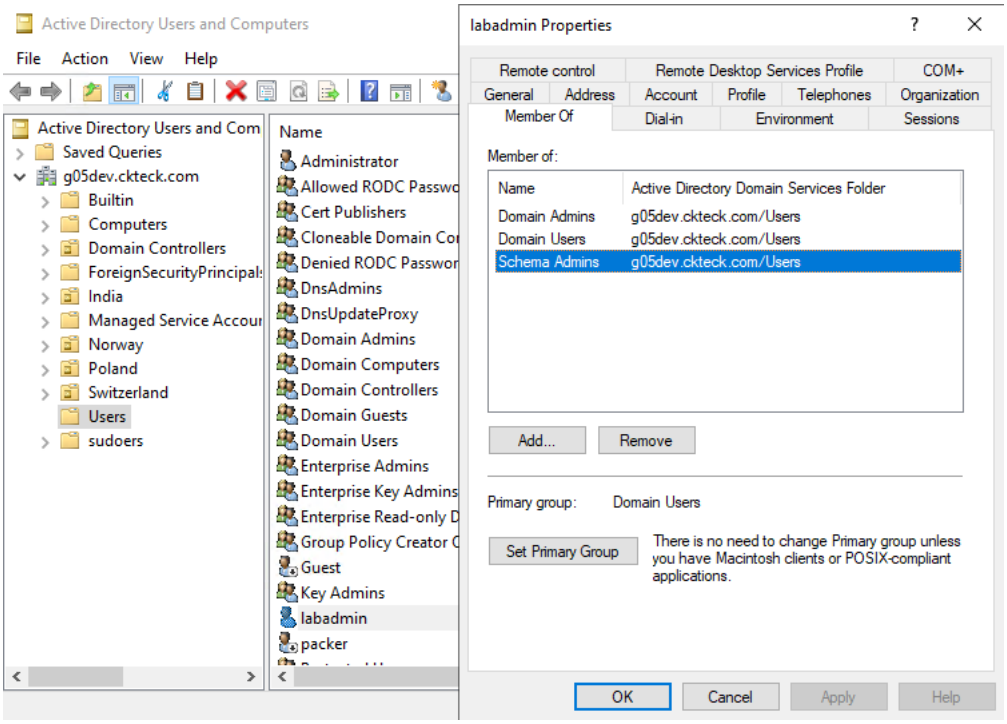
Die Datei `/etc/sudoers` legt fest, welche User und Gruppen welche Befehle als root User (oder als ein anderer Benutzer) ausführen dürfen. In der sudoers Datei können nicht nur lokal Benutzer und Gruppen sudo-Berechtigungen erhalten, sondern auch Active Directory Benutzer und Gruppen. Damit die sudoers Datei nicht auf jedem einzelnen Linux Client gepflegt werden muss, kann sudo auch zentral via Active Directory verwaltet werden. In grossen und verteilten Umgebungen muss sudo somit nicht die gesamte lokale sudoers Datei lesen, sondern führt pro Aufruf lediglich zwei bis drei Abfragen an das Active Directory durch. Ein weiterer Vorteil neben der zentralen Verwaltung dieses Vorgehens besteht darin, dass durch das Schema die korrekte Syntax für die Einträge erzwungen wird. Dies ist wichtig, da eine falsche Syntax das System zerstören kann, da es nicht mehr möglich ist erhöhte Berechtigungen zu erhalten. Werden die sudo-Berechtigungen lokal über die sudoers-Datei verwaltet, sollte die Datei aus diesem Grund nie direkt mit einem Texteditor, sondern immer über den Befehl `visudo` bearbeitet werden.

**2.2.8.1. Verwalten der sudo-Berechtigungen auf AD** Um die zusätzlichen Informationen zu den sudoern speichern zu können, muss das Schema des Active Directorys erweitert werden. Das sudo-Project stellt vorgefertigte Schemas unteranderem für Microsoft Active Directory bereit. Wechseln Sie als Benutzer labadmin zum Windows Server. Mit dem folgenden Befehl kann das Schema in einer PowerShell ins aktuelle Arbeitsverzeichnis heruntergeladen werden.

```
1 wget https://raw.githubusercontent.com/sudo-project/sudo/main/docs/schema.
    ActiveDirectory -O schema.ActiveDirectory
```

Nun muss der User labadmin der Gruppe Schema Admins hinzugefügt werden, ansonsten werden Anpassungen am Schema des ADs verweigert. Öffnen Sie erneut Active Directory Users and Computers und wählen Sie in der OU Users

(nicht Switzerland/Users) den User labadmin und öffnen Sie per Rechtsklick dessen Eigenschaften. Fügen Sie ihn im Tab “Member Of” der Gruppe Schema Admins hinzu. Damit die Änderungen wirksam werden, müssen Sie sich vom Windows Server abmelden (Sign out) und die RDP-Verbindung anschliessen erneut aufbauen.



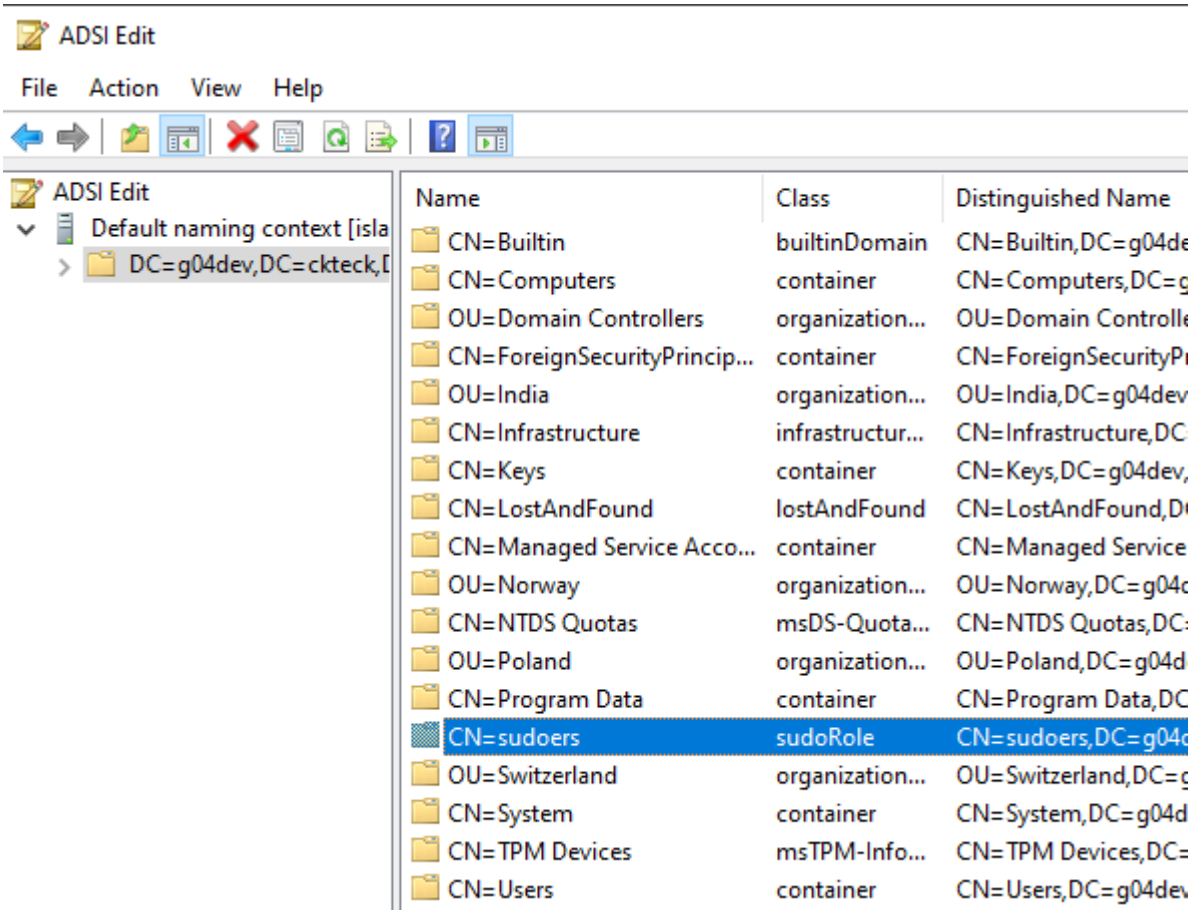

Starten Sie eine Eingabeaufforderung oder PowerShell als Administrator. Navigieren Sie in das Verzeichnis, in dem sich das heruntergeladene Schema befindet (falls Verzeichnis nicht geändert wurde: `C:\Users\labadmin`) und installieren Sie das Schema über den folgenden Befehl. Das erste X (dc=X) muss **nicht** durch Ihre Gruppennummer ersetzt werden, lediglich das zweite (dc=GXX).

```
1 ldifde -i -f schema.ActiveDirectory -c dc=X dc=GXX,dc=CKTECK,dc=COM
```

Mit dem bereits bekannten “Active Directory Users and Computers” lassen sich die neu hinzugefügten Attribute nicht bearbeiten, deshalb wird das Tool “ADSI Edit” verwendet. Es ist ebenfalls unter den Windows Administrative Tools zufinden.

Auf “ADSI Edit” wird mit Rechtsklick “Connect to …” ausgewählt. Im neuen Fenster kann ein beliebiger Name gewählt werden, die restlichen Einstellungen können Sie auf den Standard-Werten belassen.

Wählen Sie auf der linken Seite die eben erstellte Verbindung mit dem gewählten Namen und anschliessend den darunterliegenden Ordner (DC=gXX,DC=ckteck,DC=com) aus. Erstellen Sie per Rechtsklick auf diesen Ordner –> New –> Object… ein neues Objekt mit der Klasse sudoRole. Wählen sie als Value “sudoers” und schliessen die den Vorgang mit finish ab.




Der Webentwickler hat ein Ticket erstellt, in dem er sudo-Berechtigungen anfordert, da er manchmal den Webserver auf dem Linux Client neustarten muss. Im Folgenden soll der Webentwickler die notwendigen Berechtigungen nach dem Principle of least privilege erhalten. Erstellen Sie in “Active Directory Users and Computers” in der OU Switzerland/Groups eine neue Gruppe “GG_CH_Webdevs” für die Webentwickler. Fügen Sie der Gruppe die beiden Mitarbeiter hinzu, welche Mitglied der Gruppe GG_CH_IT sind.

**Hinweis**

Gemäss IGDLA müsste hier ebenfalls noch eine Domänen lokale Gruppe erstellt werden, welche die globale Gruppe als Mitglied enthält. Die Berechtigung würde dann korrekterweise der Domänen lokalen Gruppe erteilt. Auch hier ersparen wir uns der einfachheitshalber diesen Schritt.

Wechseln Sie zurück zu ADSI Edit und fügen Sie in dem Common Name (CN) sudoers per Rechtsklick ein neues Objekt hinzu, wählen Sie als Objekt sudoRole. Den Value für den Namen können Sie frei wählen, beispielsweise “Webdevs - allow webserver restart”. Jedes sudo-Rollenobjekt entspricht einer Zeile in der Datei sudoers. Für mehrere Regeln müssten also mehrere sudo-Rollenobjekte erstellt werden.

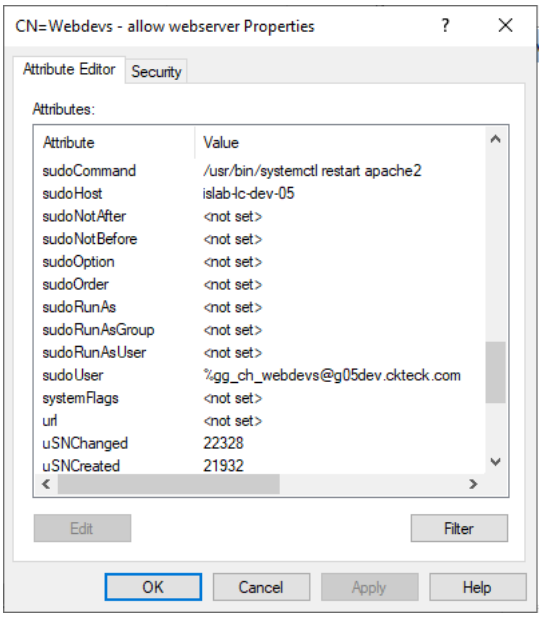
Die Eigenschaften der sudo-Regel werden nun per Rechtsklick auf das eben erstellte Objekt –> Properties bearbeitet.

Die folgenden Eigenschaften werden festgelegt:

| **Attribut** | **Wert** | **Beschreibung** |
|---|---|---|
| sudoCommand | /bin/systemctl restart apache2 | Legt fest welcher Befehl mit sudo ausgeführt werden darf. |

| Attribut | Wert | Beschreibung |
|---|---|---|
| sudoHost | islab-lc-XX | Legt fest auf welchem Host der Befehl ausgeführt werden darf (Hostname, IP-Adresse, IP-Netzwerk oder Host Netgroup). (An Gruppennummer anpassen) |
| sudoUser | %gg_ch_webdevs@gXX.ckteck.com | Es wird der User (oder die Gruppe) festgelegt der den Befehl ausführen darf. (An Gruppennummer anpassen) |

Die Eingaben werden nur übernommen, wenn sie zuerst über den Button “ADD” hinzugefügt wurden, bevor sie mit OK und Apply gespeichert werden.




Mehr Informationen und weitere Attribute finden Sie auf https://www.sudo.ws/docs/man/sudoers.ldap.man/.

Damit ist die Konfiguration auf dem Windows Server abgeschlossen. Nun muss der Linux Client so konfiguriert werden, dass er die sudoer-Regel aus dem Active Directory verwendet.

**2.2.8.2. Konfiguration des Linux-Clients** Installieren Sie das Packet libsss-sudo, welches die Kommunikation zwischen sudo und SSSD ermöglicht, um sudo-Regeln in SSSD zwischenzuspeichern.

```
1 sudo apt install libsss-sudo
```

Der Linux Client wird über den Name Service Switch (NSS) angewiesen, die sudo-Berechtigungen zusätzlich aus dem Active Directory zu lesen, nachdem die lokale sudoers Datei zum Tragen kommt. Die NSS-Konfigurationsdatei `/etc/nsswitch.conf` wird von Anwendungen verwendet, um die Quellen und Reihenfolge zu bestimmen, von denen Name-Service-Informationen bezogen werden sollen. Sehen Sie sich die Datei `nsswitch.conf` an.

```
1 cat /etc/nsswitch.conf
```

Die folgende Zeile unterhalb von netgroup veranlasst das System neben den lokalen sudo Regeln in der sudoers Datei ebenfalls die sudo Einstellungen aus dem AD zu berücksichtigen.

```
1 sudoers: files sss
```

In einer produktiven Umgebung ist es sinnvoll SSSD anzuweisen die sudo-Regeln lokal im Cache vorzuhalten. Sollte das AD einmal nicht erreichbar sein, kann so sudo weiterhin verwendet werden. Hierfür würde in der Datei `/etc/sssd/sssd.conf` unter der Sektion `[sssd]` zu den bestehenden Services `nss` und `pam` der Service `sudo` ergänzt und SSSD neu gestartet. Auf Systemen die systemd verwenden, wie unserem Debian, ist dieser Schritt optional, weil die Services über Sockets aktiviert werden.

Aktivieren und starten Sie `sssd-sudo.socket`.

```
1 sudo systemctl enable sssd-sudo.socket --now
```

Aktualisieren Sie die sudo-Regeln manuell, damit der Linux Client die eben erstellte Regel erkennt. Dieser Vorgang geschieht in vorgegebenen Zeitintervallen auch automatisch, der Befehl aktualisiert die Regeln allerdings umgehend.

```
1 sudo sss_cache -E
2 sudo sss_cache --sudo-rules
```

Melden Sie sich mit einem Mitarbeiter der Gruppe Webentwickler an (switch user - `su`) und überprüfen Sie mit `sudo -l` welche Befehle der User auf diesem Host mithilfe von sudo ausführen darf. Starten Sie den Webserver neu:

```
1 sudo systemctl restart apache2
```

**Hinweis**

Sollte die Berechtigung für den Befehl noch nicht funktionieren, melden Sie sich mit `exit` ab, starten Sie SSSD neu, wechseln Sie erneut zum Mitarbeiter und versuchen Sie es erneut.

### 2.2.9. Kerberos Authentifizierung

Der Kerberos Authentifizierungsdienst ist ein Authentifizierungsdienst, welcher auf dem Needham-Schroeder-Protokoll basiert.

Die Motivation hinter dem Authentifizierungsdienst ist zum einen die sichere und einheitliche Authentifizierung in einem Netzwerk, als auch der Schutz von Ressourcen vor unberechtigten Zugriffen. Zudem unterstützt Kerberos Single-Sign-On, sprich die einmalige Anmeldung zur Nutzung verschiedener Services.

Das Microsoft Active Directory verwendet Kerberos als Standardprotokoll für die Authentifizierung. Zentral dabei ist, dass Kerberos Tickets für die Authentifikation verwendet und der Domänencontroller als Key Distribution Center (KDC) agiert. Ein Kerberos Setup besteht aus drei Parteien, die beteiligt sind:

- dem Client – welcher sich authentisieren will
- dem Server – welcher vom Client genutzt werden will
- dem Kerberos Server – der als Authentication Server agiert

**Aufgabe 2.25**

Was wird unter KDC verstanden und weshalb ist dieses für die Funktionsweise von Kerberos zentral? Recherchieren Sie und erläutern Sie in kurzen Worten.

KDC ist der Key Distribution Controller. Er ist zentral, weil es der Service ist, der zwischen dem Server und dem Client vermittelt. Er organisiert die Schlüssel und gibt diese dem Client, so dass er sich mit seinen Berechtigungen im Server bewegen kann.

Im nächsten Schritt überprüfen Sie, ob Ihnen ein Kerberos Ticket ausgestellt wurde. Geben Sie hierzu folgenden Befehl ein währen Sie als Domänen User angemeldet sind.

```
1 klist
```

Die Ausgabe sollte in etwa so ausschauen (Abweichungen bezgl. Nummern):

```
davdur@g01dev.ckteck.com@islab-lc-dev-01:/$ klist
Ticket cache: FILE:/tmp/krb5cc_1698201112_aazQqr
Default principal: davdur@G01DEV.CKTECK.COM

Valid starting       Expires              Service principal
09/08/2022 16:08:34  09/09/2022 02:08:34  krbtgt/G01DEV.CKTECK.COM@G01DEV.CKTECK.COM
        renew until 09/09/2022 16:08:34
```

**Aufgabe 2.26**

Was wird unter den Begriffen «Default principal», «Service principal» verstanden und wofür steht das «krbtgt»? Recherchieren Sie!

default principle: lokaler Account, verantwortlich für die Verschlüsselung aller Kerberos Tickets
Service principal: repräsentiert eine client application im Kerberos Realm, zuständig für Authentifizierung und Zugriff auf den application server.
krbtgt: Standard für Verbindung auf Windows AD

Untenstehende Abbildung zeigt den Kerberos Authentifizierungsablauf auf und das Zusammenspiel mit dem KDC. Bei Interesse können weitere Informationen aus der Bildquelle entnommen werden.

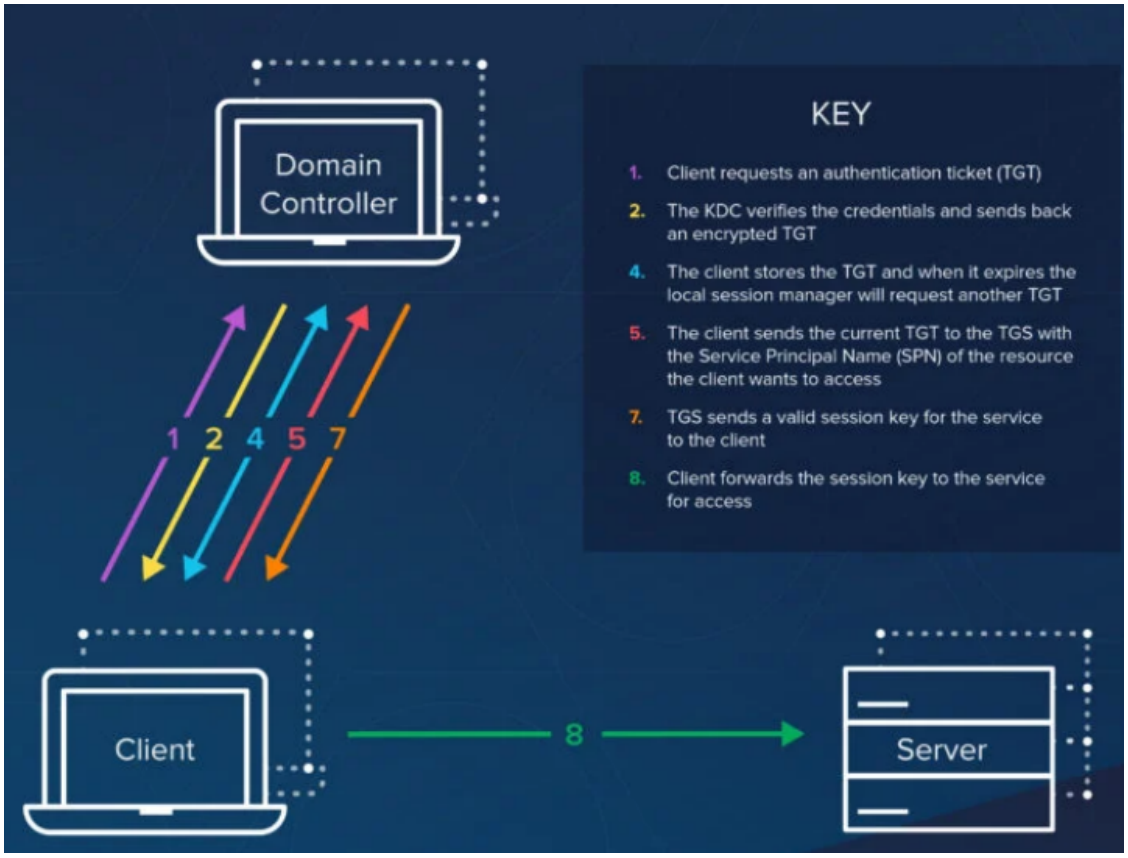
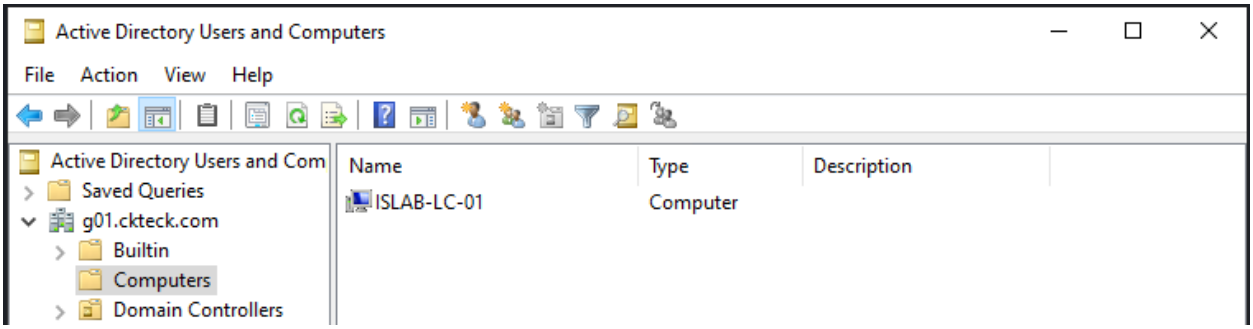



**Abbildung 2:** Simple Kerberos Diagramm (Quelle: https://www.varonis.com/de/blog/wie-funktioniert-die-kerberos-authentifizierung)

Jetzt werden Sie noch überprüfen, wo Sie den hinzugefügten Linux Client in der AD-Struktur auffinden können. Loggen Sie sich deshalb an Ihrem Windows Server (siehe letzte Woche) mit dem Benutzer **labadmin** ein & öffnen Sie «Active Directory Users and Computers».

Öffnen Sie den Container «Computers» im Wurzelverzeichnis der Domäne um zu überprüfen, ob der Linux-Rechner übernommen wurde.



Aufgabe 2.27

Weshalb wurde der hinzugefügte Computer in Container «Computers» im Wurzelverzeichnis hinzugefügt? (Tipp: letzte Woche!)

Computer wird in der Domain erkannt, weil wir uns damit angemeldet haben. Win AD schmeisst alle in den "Computers" Folder.

# 3. Betreiben eines Service unter Linux

In diesem Kapitel lernen Sie, wie ein Service unter Linux korrekt betrieben wird. Als Beispiel werden Sie ein einfaches Script als Service konfigurieren. Nach diesem Kapitel sollten Sie jedoch auch in der Lage sein, Ihre eigenen Softwareprojekte korrekt und sauber auf einem Linux Server zu betreiben.

Dabei werden wir uns zuerst kurz mit der Herkunft von Prozessen in Unix Systemen auseinandersetzen. Da dies aber sehr technisch werden würde, belassen wir es bei einer einfachen Bilderbucherklärung.

Beim Start eines Unix Systems wird der ”init”-Prozess gestartet. Jeder weitere Prozess wird vom init Prozess abgeleitet/kopiert (siehe Bild unten). Ein Prozess kann wiederum von sich selbst ableiten und weitere sogenannte Child Processes starten. Ein neuer Prozess wird mit dem Systemcall fork() abgeleitet.

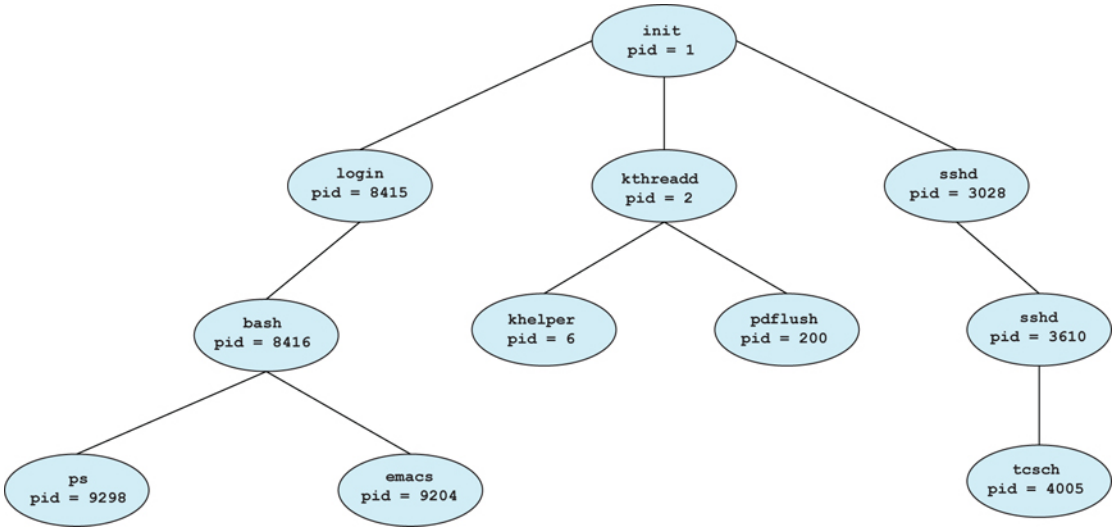



**Abbildung 3:** Unix Process creation (Quelle: https://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/3_Processes.html)

Interessierte können unter folgendem Link gerne mehr dazu lesen: https://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/3_Processes.html

## 3.1. Erkunden bestehende Applikationslandschaft

Da wir nicht wissen wie das mit den Prozessen/Applikationen und Berechtigungen genau funktioniert, wollen wir uns zuerst bestehende Prozesse/Applikationen ansehen.

Begonnen wird mit dem Apache Web Server. Dazu werden wir einen Befehl, den Sie schon aus der Linux-Einstiegsübung kennen, verwenden. Ps zeigt alle laufenden Prozesse mit mehr oder weniger Details (je nach Parameter) dazu an. Wechseln Sie dazu zum Benutzer labadmin.

```
1 ps -ef | grep apache
```

**Aufgabe 3.1**

Welche Prozesse werden angezeigt? Listen Sie zu jedem Prozess den User auf, unter welchem der Prozess ausgeführt wird.

/usr/sbin/apache2: root & www-data
selbstverständlich auch:
grep apache: labadmin

Tun Sie das gleiche für den SSSD service.

```
1 ps -ef | grep sssd
```

**Aufgabe 3.2**

Warum laufen alle SSSD Prozesse unter dem User ”Root”? Das ist doch sicherheitstechnisch nicht gut?! Was soll das?! Hinweise: https://docs.pagure.org/sssd.sssd/design_pages/not_root_sssd.html https://github.com/SSSD/sssd/issues/4898

Die Prozesse werden standardmässig alle unter Root gestartet. Man kann diese Gefahr mit einem System wie SELinux einschränken oder für SSSD einen separaten User erstellen, der nur die benötigten Rechte hat.

Wir haben also gelernt, dass gewisse Funktionen nur vom User Root durchgeführt werden können, Applikationen aber grundsätzlich mit einem Service User betrieben werden sollten. Ein Beispiel ist der Apache2 Web Server. Die Web-Anfragen werden von Prozessen des Users www-data bearbeitet, aber das Port-Binding (TCP/80 und TCP/443) muss mit dem User Root geschehen. Portnummern 1 bis 1023 können nur mit Root-Access an eine Applikation gebunden werden. Höhere Portnummern jedoch auch von Non-Root Usern.

**Hinweis**

Es ist Best Practice alle Applikationen ohne Root Rechte laufen zu lassen und an Ports grösser 1023 zu binden. Damit die Applikationen weiter über Standard-Ports erreichbar sind (zum Beispiel 80 für HTTP) wird zum Beispiel eine iptables Rule erstellt, welche den Traffic von extern 80 auf den korrekten Port > 1023 umleitet. Auch möglich sind Konstrukte mit einem Reverse Proxy Server.

Als nächstes wollen wir herausfinden, wo Applikationen unter Linux abgelegt werden. Bei Windows gibt es das allgemein bekannte ”C:\Programme”. Doch die Linux-Filestruktur ist komplett anders… Was nun?

Wir ziehen den Filesystem Hierarchy Standard der Linux Foundation zu rate. Die aktuelle Version 3.0 wurde im Jahre 2015 verabschiedet und gibt Auskunft darüber, welcher Ordner in der Linux Datenstruktur für was verwendet werden sollte.

Plötzlich hören Sie eine leise Stimme in Ihrem Kopf die Ihnen ”Kapitel 4.9” zuflüstert. Sie als Person, die schon seit Jahren im Einklang mit Bits und Bytes lebt, wundern sich nicht.

Sie blättern im Standard zum zugeflüsterten Kapitel.

https://refspecs.linuxfoundation.org/FHS_3.0/fhs-3.0.pdf

**Aufgabe 3.3**

Was für ein Ort in der Filestruktur schlägt der Linux Foundation Fileystem Hierarchy Standard vor?

/etc/opt ist für allgemein zugängliche zusätzliche Software.
/bin: allgemeine Software
/sbin: system Software
/usr/local/bin: user spezifische Anwendungen

Gut, wir reflektieren. Wir wissen nun:

- dass wir einen Service Account benötigen
- wo wir die Files des Programms ablegen sollen

## 3.2. Service User anlegen

Als erstes werden wir einen Service User und eine dazugehörige System Group anlegen. Erstellen Sie eine System Group mit folgendem Command.

```
1 sudo groupadd -r myappgroup
```

Erstellen Sie als nächstes einen Service User mit dem Namen myappuser. Diesen fügen Sie mit dem ”-g” Parameter direkt der zuvor erstellten Group hinzu.

```
1 sudo useradd -r -s /bin/false -g myappgroup myappuser
```

**Aufgabe 3.4**

Was aber bewirken die Parameter ”-r” und ”-s /bin/false”? *Welche Bedeutung hat “/bin/false”?*:
Hinweis: Die man-page des Befehls useradd oder das Internet helfen Ihnen weiter.

-r ist für Systemuser, dieser wird nicht in der User DB angezeigt
-s steht für die user shell, mit der sich der neue user einloggt.
/bin/false für dei shell, bewirkt, dass sich der User nicht einloggen kann.

Überprüfen Sie, dass der User myappuser existiert und dieser der richtigen Gruppe zugeordnet wurde.

```
1 id myappuser
```

## 3.3. Executable ablegen

In diesem Kapitel werden wir das Executable, für welches wir den Service erstellen am richtigen Ort ablegen. Wie vorhin herausgefunden, werden wir unser Executable unter ”/usr/local/bin/myapp” ablegen. **Erstellen Sie den Ordner ”/usr/local/bin/myapp”.**

Aufgrund fehlendem Executable werden wir uns eines Scripts bedienen, welches sich gleich wie eine ”normale” Applikation verhält. Erstellen Sie die folgende Datei:

```
sudo nano /usr/local/bin/myapp/myapp.sh
```

Fügen Sie folgenden Inhalt in die Datei ein:

```
#!/bin/bash
tail -f /dev/null
```

**Aufgabe 3.5**

#!/bin/bash – was macht diese Deklaration

es legt fest, bin welchem Programm die folgenden befehle ausgeführt werden sollen. hier mit der bash shell

Als nächstes Ordnen Sie alle zu Ihrem Service gehörenden Ordner und Dateien Ihrem Service User zu.

```
sudo chown -R myappuser:myappgroup /usr/local/bin/myapp
```

Kontrollieren Sie, ob Ownership und Group korrekt geändert wurde.

**Aufgabe 3.6**

Damit die Datei "myapp.sh" ausgeführt werden kann, ist ein weiterer Command notwendig. Welcher ist das und wie sieht dieser im vorliegenden Fall aus? Hinweis: Schauen Sie sich die Berechtigungen des Files mittels "ls -la" an.

sudo chmod 77* myapp.sh

Applizieren Sie obigen Command auf Ihr "Executable".

## 3.4. Service erstellen

Services werden über den System Service "systemd" verwaltet. Wikipedia sagt folgendes dazu:

Systemd (stylized as systemd) is a suite of software that provides fundamental building blocks for a Linux operating system. Among other features, it includes the systemd "**System and Service Manager**", an init system used to bootstrap the user space and to manage system processes after booting. … as of 2015, the majority of Linux distributions have adopted systemd as their default init system.

Quelle: https://en.wikipedia.org/wiki/Systemd

Systemd Service Definitionen werden unter /etc/systemd/system/ abgelegt. Erstellen Sie in diesem Ordner eine neue Datei namens myapp.service.

```
sudo nano /etc/systemd/system/myapp.service
```

In der Service Definition können verschiedenste Parameter übergeben werden. Fügen Sie folgende Konfiguration in die Datei myapp.service ein.

```
[Unit]
Description=Manage Myapp service

[Service]
WorkingDirectory=/usr/local/bin/myapp
ExecStart=/usr/local/bin/myapp/myapp.sh
User=myappuser
Type=simple
Restart=on-failure
RestartSec=10

[Install]
WantedBy=multi-user.target
```

Was genau die einzelnen Parameter bedeuten und was für weitere Konfigurationsmöglichkeiten es gibt, können Sie folgendem Link entnehmen: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/system_administrators_guide/chap-managing_services_with_systemd

### 3.5. Service starten

Um den Service überhaupt nutzen zu können, muss systemd zuerst bekannt gemacht werden, dass ein neuer Service existiert. Dazu muss folgender Command ausgeführt werden. Dabei werden alle Service-Definitionen neu eingelesen – auch unser neu hinzugefügter myapp Service.

```
sudo systemctl daemon-reload
```

Nun kann der Service wie auch alle anderen Services mit dem Command ”systemctl” verwaltet werden. Starten Sie als nächstes den Service.

```
sudo systemctl start myapp
```

Überprüfen Sie mit dem folgenden Command, ob alles funktioniert hat.

```
sudo systemctl status myapp
```

**Aufgabe 3.7**

Welche Prozess-ID wurde Ihrem Service vergeben?

5552 /bin/bash /usr/local/bin/myapp/myapp.sh
und der tail command, den er ausführt:
5559 tail -f /dev/null

Gratulation! Sie wissen nun, wie ein Service unter Linux korrekt betrieben wird!

## Notizen