

Dokumentation

LDAP-Server

hamster.panzer



Michael Herz
Andy Salewski
David Westmeyer

Inhaltsverzeichnis

1. Präambel

- 1.1 Zielsetzung der Dokumentation
- 1.2 Systemanforderungen

2. Installation der Virtuellen Umgebung und Netzwerkplan

- 2.1 Installation des Ubuntu Servers in Virtualbox mit Vagrant
- 2.2 Installation von Xubuntu-Desktop
- 2.3 Netzwerkplan

3. Einführung in LDAP

- 3.1 Grundlagen von LDAP (Lightweight Directory Access Protocol)
- 3.2 Anwendungsfälle und Architektur
- 3.3 Schlüsselbegriffe und -konzepte

4. Installation von LDAP

- 4.1 Installation der OpenLDAP-Komponenten
- 4.2 Verifizierung der LDAP-Installation

5. Konfiguration von LDAP

- 5.1 Einrichtung der Domäne hamster.panzer
- 5.2 Erstellung und Verwaltung von Benutzer- und Gruppenkonten
- 5.3 Implementierung von Zugriffskontrolllisten (ACLs)

6. Integration von SSH

- 6.1 Konfiguration des Secure Shell (SSH) für Remote-Zugriff

7. Systemhärtung und Sicherheit

- 7.1 Einrichtung von BorgBackup für systematische Backups
- 7.2 Installation und Konfiguration von ClamAV und ClamAV-Daemon
- 7.3 Implementierung von Fail2ban zum Schutz vor Brute-Force-Angriffen
- 7.4 Einsatz von RKHunter zur Rootkit-Erkennung
- 7.5 Konfiguration von Snort als Intrusion Detection System (IDS)
- 7.6 Firewall-Konfiguration mit UFW (Uncomplicated Firewall)
- 7.7 Implementierung von AppArmor zur Anwendungssicherheit
- 7.8 Konfiguration von Suricata als leistungsstarkes IDS/IPS
- 7.9 MFA (Multi-Faktor-Authentifikation) und Password Policy

8. Incident Response Management

- 8.1 Entwicklung eines Incident Response Plans (IRP)
- 8.2 Erstellung von Incident Report-Vorlagen
 - 8.2.1 Standard-Report für Sicherheitsvorfälle
 - 8.2.2 Report für Systemfehler
 - 8.2.3 Report für Datenschutzverletzungen
- 8.3 E-Mail-Vorlagen zur Kommunikation von Cyber-Incidents an Mitarbeiter

9. Bedrohungsmodellierung

- 9.1 Einführung in die STRIDE-Methode
- 9.2 Verwendung von Threat Dragon NG zur Durchführung von Bedrohungsanalysen

10. Test und Validierung

- 10.1 Testansätze und -methoden
- 10.2 Validierung der Implementierungsergebnisse

11. Glossar

- 11.1 Fachbegriffe und Abkürzungen im Kontext

12. Quellenverzeichnis

- 12.1 Relevante Literatur und technische Dokumentationen
- 12.2 Online-Ressourcen und Community-Foren

13. Appendix

- 13.1 Bedrohungsmodell
- 13.2 Benutzerhandbuch für die LDAP-Umgebung
- 13.3 Scripts und Automatisierungshilfen
- 13.4 Incident Response Plan
- 13.5 E-Mail-Vorlagen für die Incident-Kommunikation
- 13.6 Vorlagen für Incident Reports
- 13.7 Vagrantfile
- 13.8 Rechnung

1. Präambel

1.1 Zielsetzung der Dokumentation

Diese technische Dokumentation präsentiert die LDAP-Authentifizierungsstruktur der **Smooth Beans GmbH** im Rahmen unseres Abschlussprojekts. Die **LDAP-Domäne** ist ein integraler Bestandteil des Unternehmensnetzwerks und dient der effektiven Verwaltung und Authentifizierung von Benutzern und Gruppen.

Ziel dieser Dokumentation ist es, eine klare und präzise Darstellung der Implementierung, der Benutzerverwaltung sowie der Zugriffsrechte zu bieten. Hierbei werden die spezifischen Anforderungen und die technischen Aspekte der Infrastruktur berücksichtigt. Die vorliegende Arbeit legt besonderen Wert auf die Implementation einer soliden Sicherheitsstrategie, um den Anforderungen des Informationszeitalters gerecht zu werden, und das Unternehmen **Smooth Beans GmbH** und seine Infrastruktur bestmöglich gegen Bedrohungen aus dem Digitalen Raum zu schützen.

<Disclaimer> Dieses Projekt beschäftigt sich lediglich mit der Installation und Konfiguration des LDAP-Servers der Implementation von Benutzergruppen, Benutzern und der Anwendung von ACL's, sowie der Installation, Implementation und Konfiguration von Software zum Schutze des Systems nicht mit der Implementation von Physikalischen Sicherheitsmaßnahmen, diese werden als gegeben betrachtet und liegen außerhalb des Verantwortungs- und Einflussbereiches der Autoren.

1.2 Systemanforderungen

Systemanforderungen für die virtuelle Maschine in VirtualBox

Um eine optimale Funktionalität der virtuellen Maschine (VM) unter VirtualBox zu gewährleisten, sind die folgenden systemtechnischen Anforderungen zu beachten:

1. Virtuelle Maschine:

- **VirtualBox Version:** Mindestens Version 6.0 oder höher.
- **Gäste-Betriebssystem:** Ein unterstütztes Linux-Betriebssystem (in diesem Projekt Ubuntu-Bionic)
- **RAM:** Mindestens 2 GB RAM (Empfehlung: 4 GB für verbesserte Leistung).
- **CPU:** Mindestens 1 virtuelle CPU (2 oder mehr empfohlen für Multi-Threading-Anwendungen).
- **Festplattenspeicher:** Mindestens 20 GB freier Speicherplatz für die VM.

2. Guest Additions:

- **Installation:** Die Guest Additions müssen installiert werden, um die Integration zwischen Host- und Gastbetriebssystem zu verbessern.
- **Manuelles Mounten:** Das CD-ROM-Laufwerk muss manuell gemountet werden, um die Installationsdateien der Guest Additions verfügbar zu machen.

Verwendete Befehle:

```
sudo mount /dev/cdrom /media/cdrom  
cd /media/cdrom  
sudo sh VboxLinuxAdditions.run
```

3. Gemeinsame Ordner (Shared Folders):

- **Einbindung:** Shared Folders ermöglichen den einfachen Austausch von Dateien zwischen dem Host- und dem Gastbetriebssystem.
 - **Konfiguration:** Der gemeinsame Ordner muss in den VirtualBox-Einstellungen der VM eingerichtet werden.
 - **Zugriffsrechte:** Stellen Sie sicher, dass der Benutzer des Gastbetriebssystems die erforderlichen Berechtigungen für den Zugriff auf den gemeinsamen Ordner hat

Das eigentliche Erstellen des Ubuntu Server Images wurde mit **Vagrant** vorgenommen, welches auch die Netzwerkkonfiguration der Virtuellen Maschine übernahm. Dieses **Vagrantfile** ist dieser Dokumentation im Appendix angefügt. Sollte das Deutsche Tastaturlayout nicht übernommen werden, kann dies einfach über die Eingabe folgender Befehle behoben werden.

```
sudo Localectl set-keymap de  
reboot
```

2.2 Installation von Xubuntu-Desktop

Da die von uns verwendete Maschine standardmäßig nur als Terminal Version vorliegt, welche nur über Eingeschränkte Möglichkeiten insbesondere des nicht vorhanden seines einer scroll verfügt wurde eine Desktopumgebung manuell hinzugefügt, um die Nachvollziehbarkeit von Fehlermeldungen vollständig auswerten und analysieren zu können. Für diesen Zweck wurde Xubuntu gewählt.

Befehl zur Installation von Xubuntu

```
sudo apt install xubuntu-desktop -y
```

2.3 Netzwerkplan

Der in dieser Dokumentation vorgestellte **LDAP-Server** ist wie bereits erwähnt Teil des Firmen Netzwerks der **Smooth Beans GmbH** und fungiert als **Admin-Domäne**. In diesen Abschnitt wird diese Domäne nun einmal dargestellt.

Für diesen Zweck wurde von uns ein Netzwerkdiagramm angefertigt, welches die Struktur verbildlicht <Bild 1> Die In diesem Netzwerkplan aufgeführten Komponenten sind nicht Teil dieses Projektes.

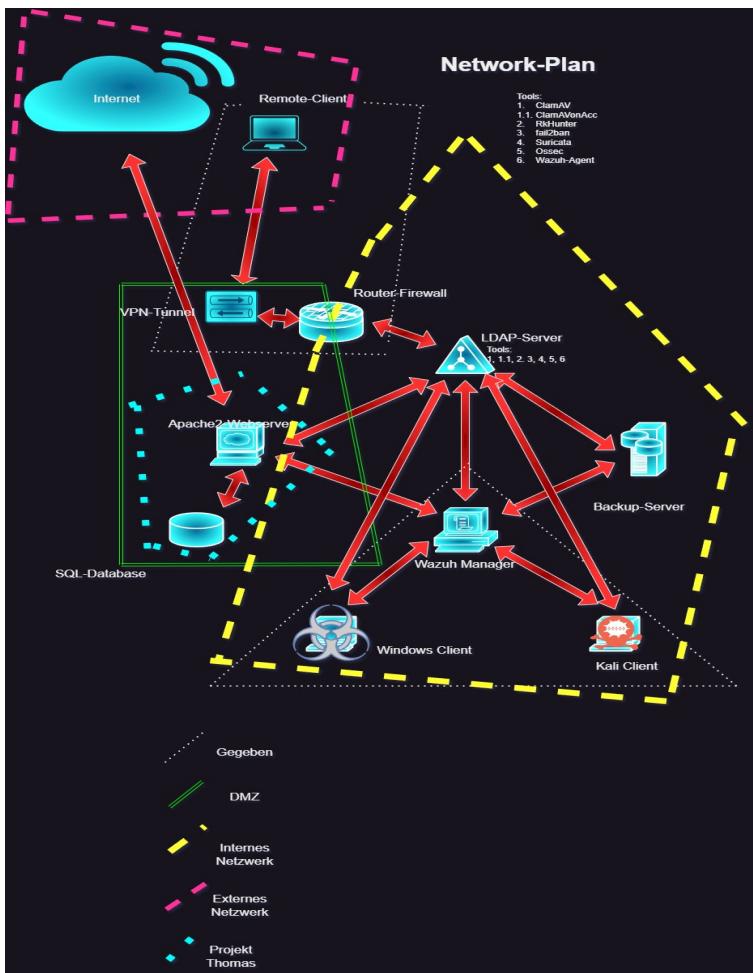


Bild 1. Die Admin Domäne „hamster.panzer“ der Smooth Beans GmbH in der Übersicht.

Für die Zukunft ist des weiteren geplant, den LDAP-Server in die **Wazuh** Überwachung einzubinden und die verbliebenen Komponenten in die Domäne zu integrieren.

3. Einführung in LDAP

3.1 Grundlagen von LDAP (Lightweight Directory Access Protocol)

LDAP (Lightweight Directory Access Protocol) ist ein offenes Protokoll zur Abfrage und Modifikation von Verzeichnisdiensten, das in einem Client-Server-Modell arbeitet. Es ermöglicht den Zugriff auf hierarchisch organisierte Daten, die als Directory

Information Tree (DIT) bezeichnet werden. Jeder Eintrag im DIT wird durch einen Distinguished Name (DN) identifiziert.

Wichtige Operationen im LDAP-Protokoll sind **Bind** (Authentifizierung), **Search** (Abfrage), **Add** (Hinzufügen von Einträgen), **Modify** (Ändern von Einträgen), **Delete** (Löschen von Einträgen) und **Unbind** (Sitzungsbeendigung). Daten werden häufig im **LDAP Data Interchange Format (LDIF)** gespeichert.

LDAP unterstützt verschiedene Authentifizierungsmethoden, einschließlich einfacher Authentifizierung und sicherer Kommunikation über SSL/TLS (LDAPS). Es wird in Unternehmensumgebungen zur zentralen Verwaltung von Benutzeridentitäten, Gruppen und Zugriffsrechten eingesetzt, insbesondere in Systemen wie Microsoft Active Directory und OpenLDAP. Vorteile von LDAP sind die Effizienz bei Suchvorgängen, hohe Skalierbarkeit und breite Unterstützung in verschiedenen Anwendungen.

3.2 Anwendungsfälle und Architektur

LDAP findet breite Anwendung in der Benutzerverwaltung, Zugriffskontrolle und der zentralen Speicherung von Informationen. Zu den typischen Anwendungsfällen gehören:

- **Zentrale Authentifizierung:** LDAP dient als Authentifizierungsdienst für Anwendungen und Systeme, wodurch eine einheitliche Anmeldung (Single Sign-On, SSO) ermöglicht wird.
- **Verzeichnisdienste:** Es wird verwendet, um Informationen über Benutzer, Gruppen und Ressourcen in Netzwerken zu speichern und zu verwalten, z. B. in Unternehmensverzeichnissen.
- **E-Mail-Verzeichnisse:** LDAP wird häufig zur Speicherung von Kontaktdaten in E-Mail-Systemen genutzt, was eine einfache Adresssuche ermöglicht.
- **Netzwerkgeräteverwaltung:** Netzwerkgeräte wie Drucker und Switches können in einem LDAP-Verzeichnis registriert und verwaltet werden.

Die Architektur von LDAP basiert auf einem Client-Server-Modell. Clients senden Anfragen an einen oder mehrere LDAP-Server, die die Daten in einem strukturierten Verzeichnis speichern. Diese Server können in einer hierarchischen Struktur angeordnet sein, um Lasten zu verteilen und Redundanz zu gewährleisten.

3.3 Schlüsselbegriffe und -konzepte

- **Distinguished Name (DN):** Ein eindeutiger Bezeichner für einen Eintrag im Verzeichnis, der die hierarchische Struktur widerspiegelt.
- **Attribute:** Merkmale oder Eigenschaften eines Verzeichniseintrags, wie Name, E-Mail oder Telefonnummer.
- **Object Class:** Definiert den Typ eines Verzeichniseintrags und die Attribute, die dieser Eintrag enthalten kann. Beispielklassen sind person oder organizationalUnit.
- **LDAP-Filter:** Eine Suchabfrage, die bestimmte Kriterien zur Identifizierung von Einträgen im Verzeichnis angibt, z. B. (uid=jdoe) für den Benutzer mit dem Benutzernamen "jdoe".
- **LDAP-Schema:** Die Definition der Objektklassen und Attribute, welche im Verzeichnis verwendet werden, einschließlich deren Typen und Regeln.
- **Replication:** Die Synchronisation von Daten zwischen mehreren LDAP-Servern, um Verfügbarkeit und Ausfallsicherheit zu gewährleisten.

- **Access Control Lists (ACLs):** Regeln, die festlegen, welche Benutzer oder Gruppen auf bestimmte Einträge oder Attribute zugreifen dürfen

4. Installation von LDAP

4.1 Installation der OpenLDAP-Komponenten

Hier ist eine Übersicht der grundlegenden Befehle zur Verwaltung von OpenLDAP. Diese Befehle sind ohne Validierung und dienen als Referenz für die Nutzung in der Installation und Konfiguration von OpenLDAP.

Installation von OpenLDAP

- Debian/Ubuntu:

```
sudo apt-get update -y
sudo apt-get install slapd ldap-utils -y
sudo systemctl enable slapd
sudo systemctl start slapd
```

4.2. Verifizierung der LDAP-Installation

Um nun zu validieren, ob die Installation erfolgreich war und um sicher zu stellen, dass der Server auch korrekt gestartet wurde wird der nachfolgende Befehl verwendet:

```
sudo systemctl status slapd
```

Die Eingabe dieses Befehls sollte die folgende Ausgabe in der Kommandozeile zurück geben. **< Bild 2 >**

```
vagrant@ubuntu-bionic:~$ sudo systemctl status slapd
● slapd.service - LSB: OpenLDAP standalone server (Lightweight Directory Access Protocol)
  Loaded: loaded (/etc/init.d/slapd; generated)
  Drop-In: /lib/systemd/system/slapd.service.d
            └─slapd-remain-after-exit.conf
  Active: active (running) since Mon 2024-09-30 11:09:33 UTC; 59s ago
    Docs: man:systemd-sysv-generator(8)
 Process: 10794 ExecStop=/etc/init.d/slapd stop (code=exited, status=0/SUCCESS)
 Process: 10800 ExecStart=/etc/init.d/slapd start (code=exited, status=0/SUCCESS)
  Tasks: 3 (limit: 2360)
 CGroup: /system.slice/slapd.service
         └─10826 /usr/sbin/slapd -h ldap:/// -g openldap -u openldap -F /etc/ldap/slapd

Sep 30 11:09:33 ubuntu-bionic systemd[1]: Starting LSB: OpenLDAP standalone server (Lightweight Directory Access Protocol)
Sep 30 11:09:33 ubuntu-bionic slapd[10800]: * Starting OpenLDAP slapd
Sep 30 11:09:33 ubuntu-bionic slapd[10822]: @(#) $OpenLDAP: slapd (Ubuntu) (May 12 2022 13:52:38) $Debian OpenLDAP Maintainers <pkg-openldap-devel@lists.alioth.debian.org>
Sep 30 11:09:33 ubuntu-bionic slapd[10826]: slapd starting
Sep 30 11:09:33 ubuntu-bionic slapd[10800]: ...done.
Sep 30 11:09:33 ubuntu-bionic systemd[1]: Started LSB: OpenLDAP standalone server (Lightweight Directory Access Protocol).
```

Bild 2. Erwartete Ausgabe. Diese Abbildung bestätigt das der slapd service korrekt gestartet wurde und Aktiv ist

5. Konfiguration von LDAP

5.1 Einrichtung der Domäne hamster.panzer

Nach der Installation und Verifizierung der korrekten Funktion des OpenLDAP-Servers wird dieser nun konfiguriert. Um diesen Vorgang möglichst einfach und anschaulich abzubilden, wurde dieser mit Screenshots festgehalten. **< Bild 3 – Bild 9 >** die jeweils zu wählende option ist diejenige, welche färblich hervorgehoben ist.

Um diesen Prozess nun zu starten und im selben Zuge die Domäne **hamster.panzer** zu erstellen wird in der Kommandozeile des Ubutu-Servers folgender Befehl eingegeben.

```
sudo dpkg-reconfigure slapd
```

Wie berits erwähnt wird dieser Prozess nun verbildlicht in der korrekten Reihenfolge dargestellt.

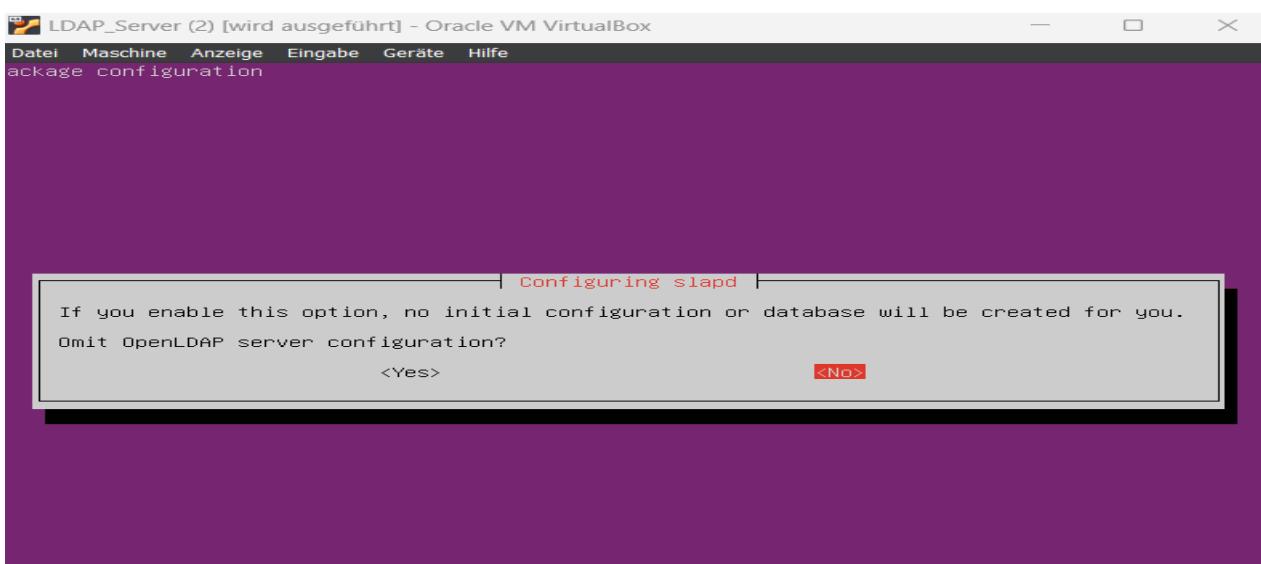


Bild 3. Starten der Konfiguration mit <NO> Bild 4. Vergeben des Domänen Namens

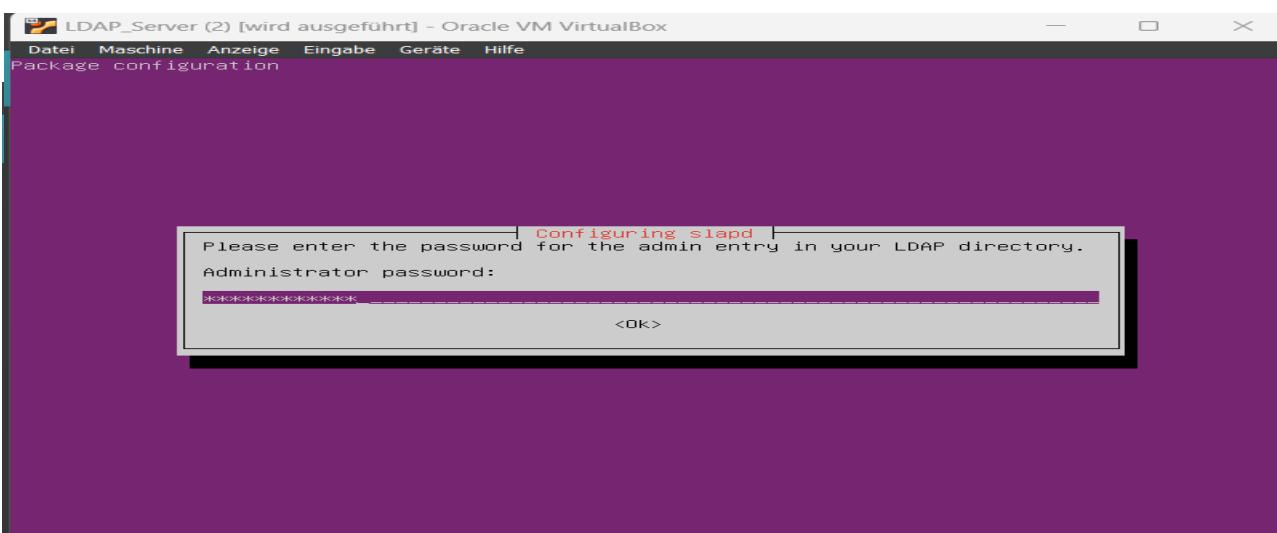


Bild 5 Vergabe des Admin Passwortes dieses muss im nächsten Schritt bestätigt werden

Implementation Ubuntu-LDAP Server, Konfiguration und Härtung

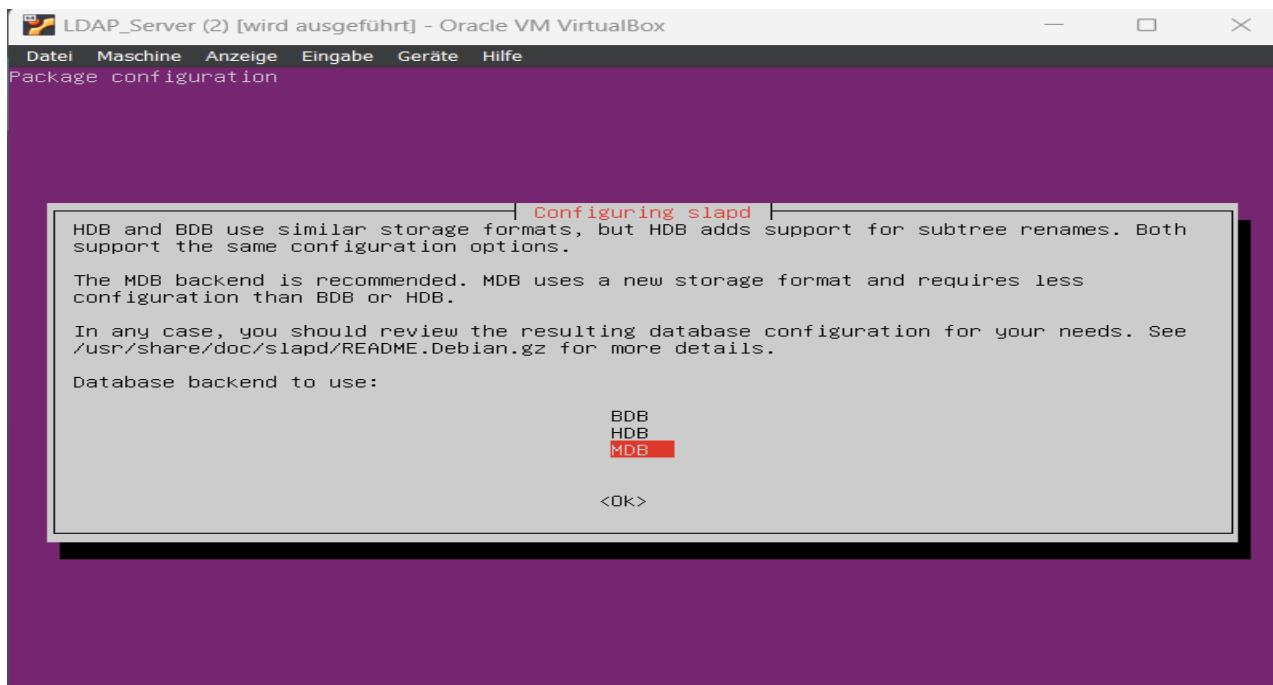


Bild 6. Database Backend festlegen

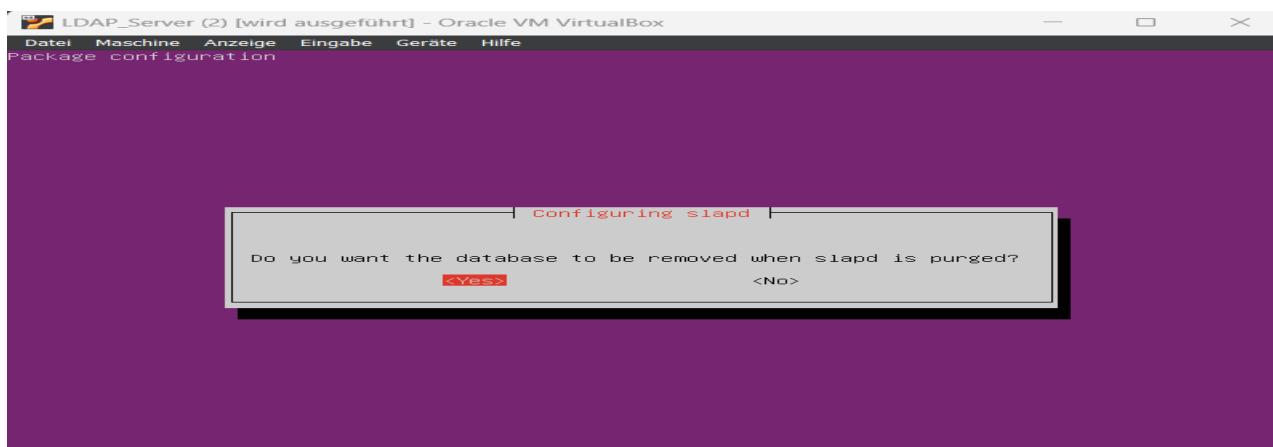


Bild 7. Database bei Purge entfernen

Nun ist die Konfiguration abgeschlossen und muss mit der Eingabe des nachstehenden Befehls überprüft werden.

```
Ldapsearch -x -b "dc=hamster,dc=panzer" -D  
ropating domainDNSZones and forestDNSZones partitions  
invalid permissions on directory '/var/lib/samba/private/msg.sock': has 0755 should be 0700  
See /var/lib/samba/private/named.conf for an example configuration include file for BIND  
and /var/lib/samba/private/named.txt for further documentation required for secure DNS updates  
Setting up sam.ldb rootDSE marking as synchronized  
Fixing provision GUIDs  
A Kerberos configuration suitable for Samba AD has been generated at /var/lib/samba/private krb5.conf  
Once the above files are installed, your Samba AD server will be ready to use  
Server Role: active directory domain controller  
Hostname: ubuntu-bionic  
NetBIOS Domain: HAMSTER ←  
DNS Domain: hamster.panzer  
DOMAIN SID: S-1-5-21-1905030480-1900917822-3011011477
```

Bild 8. Die Eingabe des Befehls ,ldapsearch -x -b "dc=hamster,dc=panzer" -D' gibt diese Ausgabe zurück und bestätigt somit die erfolgreiche Erstellung der Domäne

Um den Domänen Service nun zu erstmalig zu starten und bei einem Neustart des Systems automatisch zu starten werden nun folgende Befehle in der Kommandozeile ausgeführt, wenn dies nicht bereits wie in Kapitel 4. Installation von LDAP geschehen ist.

```
sudo systemctl enable slapd
sudo systemctl start slapd
```

Der Aktuelle Status des `slapd` services wird nun mittels des Befehls:

```
sudo systemctl status slapd
```

vallidiert und sollte die Ausgabe zurückgeben, welche bereits in <Bild 2> des Kapitels 4. Installation von LDAP abgebildet wurde. Wurde dies bereits wie in Kapitel 4.

Installation von LDAP beschrieben ausgeführt, wird der Befehl:

```
sudo systemctl restart slapd
```

verwendet und anschiesend der Status wie bereits beschrieben überprüft.

Ein ausführlicher Leitfaden zum Betrieb und zur Administration des LDAP Server wird im Appendix dieses Dokumentes angehängt. Dieser Leitfaden enthält unter anderem auch eine Sammlung aller wichtigen Befehle so wie ihrer Funktion.

5.2 Erstellung und Verwaltung von Benutzer- und Gruppenkonten

Bevor nun mit der Konfiguration der Struktur des LDAP-Servers und dem Einpflegen von Gruppen, Benutzern sowie der Zugriffsteuerung begonnen wird muss zuerst die Anmeldung von LDAP Benutzern ausdrücklich erlaubt werden, hierfür wird folgender Befehl in der Kommandozeile ausgeführt:

```
ldapmodify -x -D "cn=admin,dc=example,dc=com" -W -f /path/to/password_change.ldif
```

Dieser Befehl öffnet die Oberfläche welche in <Bild 9> dargestellt wird, hier wird der Name der Domäne in folgendem Format eingegeben und anschließend bestätigt

```
dc=hamster,dc=panzer.
```

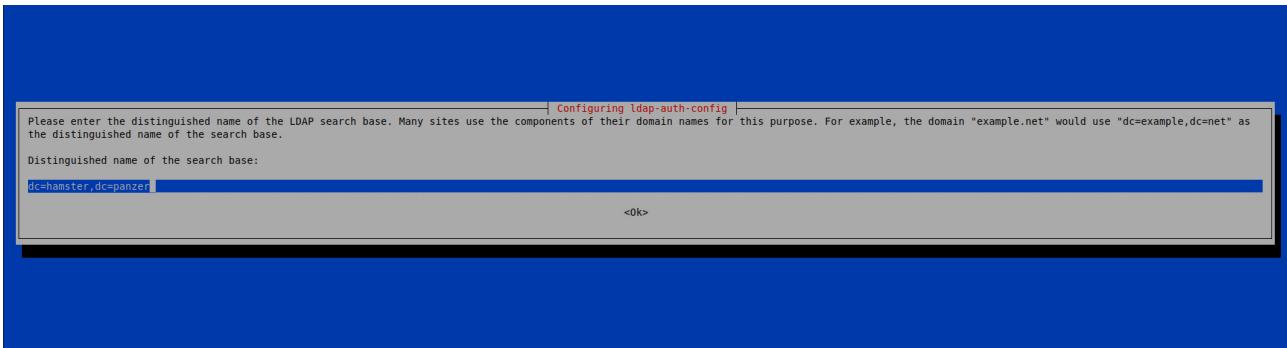


Bild 9. Erlauben des Anmelden von Benutzern auf der Domäne hamster.panzer

Bevor wir nun mit der Erstellung der Gruppen, Benutern und den spezifischen Berechtigungen welche diese erhalten sollen fortfahren, möchten ist es Notwendig die Beabsichtigte Struktur einmal als Diagramm darzustellen. <Bild 10>



Bild 10. Die vorgesehene Struktur der Gruppen wie sie auf dem LDAP-Server integriert werden sollen.

Um nun mit der Erstellung der Gruppen, Benutzern sowie den ACL's beginnen zu können müssen zunächst noch einige **ldif** Dateien erstellt und angewendet werden die Ausgangs Struktur sollte wie in <Bild 11> abgebildet aussehen wenn die nachfolgend erwähnte **base.ldif** korrekt formatiert und angewendet wurde.

```
vagrant@ubuntu-bionic:~$ ldapsearch -x -b "dc=hamster,dc=panzer" -D "cn=admin,dc=hamster,dc=panzer" -W | lolcat
Enter LDAP Password:
# extended LDIF
#
# LDAPv3
# base <dc=hamster,dc=panzer> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#
# hamster.panzer
dn: dc=hamster,dc=panzer
objectClass: top
objectClass: dcObject
objectClass: organization
o: SmoothBeans
dc: hamster

# admin, hamster.panzer
dn: cn=admin,dc=hamster,dc=panzer
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
userPassword:: e1NTSEF9TTNNeG1QVm8za0xKYTRER1Fr0UcycDZwZmQzdTFaeDc=

# search result
search: 2
result: 0 Success

# numResponses: 3
# numEntries: 2
vagrant@ubuntu-bionic:~$
```

Bild 11. Ausgangs Struktur des LDAP-Servers mit der Domäne hamster.panzer

Folgende Schritte müssen nun nach einander ausgeführt werden:

- Erstellen und Anwenden der `base.ldif` – Diese Enthält die Organizational Units und User in welchen die Gruppen und Benutzer nachfolgend angelegt werden.
- Anwenden der `base.ldif > ldapadd -x -D "cn=admin,dc=example,dc=com" -W -f base.ldif`
- Validierung der Korrekten Anwendung der `base.ldif > ldapsearch -x -b „dc=hamster,dc=panzer“ -D „cn=admin,dc=hamster,dc=panzer“ -W` Die Erwartete Ausgabe ist in **< Bild 10 >** dargestellt.
- Erstellung und Anwendung der `ou.ldif` dieser Schritt kann nötig werden, wenn bei der Anwendung der `base.ldif` Fehler auftreten und diese OU's nicht korrekt implementiert wurden. Zur Anwendung folgenden Befehl verwenden `ldapadd -x -D "cn=admin,dc=example,dc=com" -W -f ou.ldif`
- Erstellung der Gruppen/Abteilungen in der `groups.ldif` und Anwendung dieser Datei mit `ldapadd -x -D "cn=admin,dc=example,dc=com" -W -f groups.ldif`
- Verifizierung der korrekten Anwendung mit `ldapsearch -x -b "dc=hamster,dc=panzer" "(objectClass=posixGroup)"`

Nun sollte wie in **< Bild 12 >** die Ausgabe folgendermaßen aussehen:

```
root@ubuntu-bionic:~# ldapsearch -x -b "ou=groups,dc=hamster,dc=panzer" -D "cn=admin,dc=hamster,dc=panzer" -W | less
Enter LDAP Password:
# extended LDIF
#
# LDAPv3
# base <ou=groups,dc=hamster,dc=panzer> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#
# groups, hamster.panzer
dn: ou=groups,dc=hamster,dc=panzer
objectClass: organizationalUnit
ou: groups

# hr, groups, hamster.panzer
dn: cn=hr,ou=groups,dc=hamster,dc=panzer
objectClass: posixGroup
cn: hr
gidNumber: 10008

# wazuh, groups, hamster.panzer
dn: cn=wazuh,ou=groups,dc=hamster,dc=panzer
objectClass: posixGroup
cn: wazuh
gidNumber: 10009

# it_leitung, groups, hamster.panzer
dn: cn=it_leitung,ou=groups,dc=hamster,dc=panzer
objectClass: posixGroup
cn: it_leitung
gidNumber: 10002

# buchhaltung, groups, hamster.panzer
dn: cn=buchhaltung,ou=groups,dc=hamster,dc=panzer
objectClass: posixGroup
cn: buchhaltung
gidNumber: 10007

# logistik_leitung, groups, hamster.panzer
dn: cn=logistik_leitung,ou=groups,dc=hamster,dc=panzer
objectClass: posixGroup
cn: logistik_leitung
... --
```

Bild 12. Die LDAP-Domänen Struktur nach Anwendung der voranstehenden ldif Dateien.

- Die Erstellung und die Implementierung der Benutzer wurde mittels eines von uns verfassten **Python** Skripts realisiert, dieses ist wie sämtliche **ldif**, **bash** und anderen **Python** Skripte ebenfalls im Appendix anhängig.

<Disclaimer> Alle in diesem Dokument genannten Benutzernamen wurden zufällig von einer KI generiert. Jegliche Ähnlichkeiten mit realen, lebenden oder verstorbenen Personen sind rein zufällig und nicht von den Autoren beabsichtigt!

- Ausführen des **Python** Skripts mittels Eingabe von
`sudo python3 create_users.py` Generierung von Benutzer Accounts auf dem LDAP-Server. Dieses Skript erstellt nicht nur die Benutzer sondern ordnet sie auch direkt den vorgesehenen Gruppen zu was die Ersteinrichtung deutlich vereinfachte.
- Überprüfung der korrekten Implementation der Benutzer durch Eingabe des Befehls
`Ldapsearch -x -b "dc=hamster,dc=panzer" "(objectClass=posixAccount)"`

Nachfolgend in **<Bild 13>** wird die Erwartete Ausgabe verdeutlicht. Die abgebildete Ausgabe zeigt nicht nur die vorhandenen Benutzer Konten sondern auch die Gruppen in welchen sie wie vorgesehen zugeordnet wurden.

```
# thomas.falke, users, hamster.panzer
dn: uid=thomas.falke,ou=users,dc=hamster,dc=panzer
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
cn: Thomas Falke
sn: Falke
uid: thomas.falke
userPassword:: UGFzc3dvcmQxMjMh
gidNumber: 10013
homeDirectory: /home/users/thomas.falke
uidNumber: 20004

# david.thornton, users, hamster.panzer
dn: uid=david.thornton,ou=users,dc=hamster,dc=panzer
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
cn: David Thornton
sn: Thornton
uid: david.thornton
userPassword:: UGFzc3dvcmQxMjMh
gidNumber: 10014
homeDirectory: /home/users/david.thornton
uidNumber: 20005

# michael.stone, users, hamster.panzer
dn: uid=michael.stone,ou=users,dc=hamster,dc=panzer
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
cn: Michael Stone
sn: Stone
uid: michael.stone
userPassword:: UGFzc3dvcmQxMjMh
gidNumber: 10015
homeDirectory: /home/users/michael.stone
uidNumber: 20006

# search result
search: 2
result: 0 Success

# numResponses: 22
# numEntries: 21
root@ubuntu-bionic:/home/vagrant#
```

Bild 13. Ein Abriss der neuen LDAP-Server Struktur nach erfolgreichem einpflegen der Gruppen und Benutzer

5.3 Implementierung von Zugriffskontrolllisten (ACLs)

Wie schon bei der Erstellung der Gruppen setzten wir nun auch bei der Vergabe der Zugriffskontrollisten auf Automatisierung. In <Tabelle 1> wird zuvor allerdings noch eine Übersicht über die Gruppen, Benutzer sowie ihre Berechtigungen vorgestellt. Die Vergabe der Zugriffskontrollisten erfolgte nach dem „Least Privileges“ Prinzip, dies soll sicherstellen, dass alle Benutzer nur die Berechtigungen haben welche sie zur Erfüllung der ihnen übertragenen Aufgaben benötigen.

Gruppe	gidNumber	Benutzer	Rechte
Geschaeftsfuehrung	10001	Max Mustermann	Volle Zugriffsrechte auf eigene Ordner, Zugriff auf Ordner der Verwaltung, HR, Buchhaltung
IT-Leitung	10002	Klaus Müller, Sabine Schmidt	Volle Administratorrechte auf alle Systeme, Zugriff auf alle Ordner und Ressourcen, Sudo-Rechte
Logistik-Leitung	10003	Anna Becker	Lesen und Schreiben in eigenen Ordnern, Zugriff auf relevante Logistik-Daten
Marketing-Leitung	10004	Peter Klein	Lesen und Schreiben in eigenen Ordnern, Zugriff auf Marketing-Daten
Produktion-Leitung	10005	Laura Wagner	Lesen und Schreiben in eigenen Ordnern, Zugriff auf Produktions-Daten
Verwaltung-Leitung	10006	Hans Meier	Lesen und Schreiben in eigenen Ordnern, Zugriff auf Verwaltungsdaten
Buchhaltung	10007	Julia Braun	Lesen und Schreiben in eigenen Ordnern, Zugriff auf Buchhaltungsdaten
HR	10008	Lisa Schwarz	Lesen und Schreiben in eigenen Ordnern, Zugriff auf HR-Daten
Wazuh	10009	Oliver König	Lesen und Schreiben in eigenen Ordnern, Zugriff auf Sicherheitsdaten
LDAP-Administrator	10010	Admin User	Volle Administratorrechte auf alle Systeme und Ordner, Kontrolle über LDAP-Verwaltung
Web-Administrator	10011	Tom Schneider	Sudo-Rechte für Docker und Webserver, Zugriff auf Webserver-Ordner, keine root-shell
DAU	10012	Martin Kellerer	Nur An- und Abmelden, keine weiteren Berechtigungen

Tabelle 1. Gruppen und Benutzer Übersicht sowie die vorgesehenen Zugriffsberechtigungen (ACL)

Wie eingangs dieses Kapitels bereits erwähnt wurde für die Anwendung dieser ACL's ein Bash Skript verwendet der hierfür anzuwendende Befehl lautet wie folgt:

`sudo bash permissions.sh`

Die Validierung wird direkt bei Ausführung des Skripts von diesem im Terminal als Print Statement ausgegeben. <Bild 14>

```
LDAP-Berechtigungen erfolgreich angewendet.
root@ubuntu-bionic:/home/vagrant/LDAP_CONFIG/bash#
```

Bild 14 Die Berechtigungen wurden erfolgreich angewendet

Nun können sich alle Angelegten Benutzer wie vorgesehen über den **LDAP-Service** Anmelden und Autorisieren im Nachfolgenden Kapitel werden nun die Implementation eines **SMB** Dienstes (Server Message Block) sowie die Einrichtung eines **SSH** (Secure Shell Host) Dienstes behandelt, über welches die Domänen Mitglieder auf die ihnen zugeschriebenen Ressourcen welche von dem **LDAP-Service** verwaltet werden zugreifen können.

6. Integration von SSH

6.1 Konfiguration des Secure Shell (SSH) für Remote-Zugriff

Damit die Benutzer sich nun wie vorgesehen über den LDAP Verzeichnisdienst Anmelden und Authentifizieren können wird der **OpenSSH** Dienst installiert und konfiguriert.

Installation und Konfiguration von OpenSSH

- Ausführen des folgenden Befehls, um den OpenSSH-Server zu installieren:
`sudo apt-get update && sudo apt-get install -y openssh-server`
- Überprüfung des **SSH-Dienstes** Kontrolle, ob der SSH-Dienst läuft:
`sudo systemctl status ssh`
- Starten des SSH Dienstes sollte er nicht laufen.
`sudo systemctl start ssh`
- Konfiguration von SSH
`sudo vim /etc/ssh/sshd_config`
 - Root-Login deaktivieren:
`PermitRootLogin no`
 - Maximale Authentifizierungsversuche reduzieren:
`MaxAuthTries 3`
 - SSH-Protokoll-Version 2 erzwingen:
`Protocol 2`
- Neustart des **SSH-Dienstes** Nach der Konfiguration muss der SSH-Dienst neu gestartet werden:
`sudo systemctl restart ssh`
- Firewall konfigurieren (UFW)
`sudo ufw allow 22/tcp`
`sudo ufw enable`

Nun sollten die **LDAP-Domänenmitglieder** im stande sein sich remote mittels **SSH** auf dem LDAP Server einzuloggen. **< Bild 15 >**

```
[parrot@parrot]~
└─$ ssh martin.kellerer@192.168.56.100
martin.kellerer@192.168.56.100's password:
martin.kellerer@ubuntu-bionic:~$ ^C
martin.kellerer@ubuntu-bionic:~$ exit
logout
```

Bild 15. Der User Martin.Kellerer hat sich erfolgreich von einer anderen Maschine (ParrotOS) per SSH Authentifiziert und auf dem LDAP-Server eingeloggt

7. Systemhärtung und Sicherheit

7.1 Einrichtung von BorgBackup für systematische Backups

BorgBackup (kurz **Borg**) ist ein fortschrittliches, sicheres und effizientes Backup-Tool, das speziell für die Sicherung und Wiederherstellung von Daten entwickelt wurde. Es nutzt ein dedupliziertes Speicherformat, wodurch identische Daten nur einmal gespeichert werden. Dies reduziert den benötigten Speicherplatz erheblich.

Borg unterstützt sowohl lokale als auch entfernte Backups und bietet eine starke Verschlüsselung für Datensicherheit. Die integrierte Komprimierung minimiert den Speicherbedarf weiter und beschleunigt die Übertragung von Daten. Mit Borg kannst du Backups inkrementell durchführen, sodass nur Änderungen seit dem letzten Backup gespeichert werden, was die Backup-Zeiten verkürzt.

Das Tool verfügt über eine einfache und benutzerfreundliche Kommandozeilenoberfläche, die eine flexible Konfiguration ermöglicht. Außerdem können Backups automatisiert und in Skripten integriert werden, was die Verwaltung und Wartung von Backup-Jobs erleichtert. BorgBackup ist besonders beliebt in der Linux-Community und wird häufig für die Sicherung von Servern und wichtigen Daten verwendet.

Für die Installation wird der folgende Befehl wie gewohnt in der Kommandozeile eingegeben:

sudo apt install borgbackup -y

Bild 6. Database Backend festlegen
Nun kann **Borg** einfach über das Terminal gesteuert werden. Da der **Backup-Server** zur Zeit der erstellung dieser Dokumentation noch nicht in der Domäne eingebunden war, wurde auf die Konfiguration von Automatischen Backups derzeit verzichtet. Der vorgesehne Backup Plan wurde hingegen bereits definiert und wie folgt terminiert:

- Dienstags: 22:00 Uhr
- Freitags: 22:00 Uhr

```
process (default: 'ssh')
required arguments:
  <command>
  mount          mount repository
  serve          start repository server process
  init           initialize empty repository
  check          verify repository
  key            manage repository key
  change-passphrase  change repository passphrase
  create          create backup
  extract         extract archive contents
  export-tar      create tarball from archive
  diff            find differences in archive contents
  rename          rename archive
  delete          delete archive
  list             list archive or repository contents
  umount          umount repository
  info             show repository or archive information
  break-lock      break repository and cache locks
  prune           prune archives
  loads           load repository format
  recreate        re-create archives
  with-lock       run user command with lock held
  config          get and set configuration values
  debug           debugging command (not intended for normal use)
  benchmark       benchmark command
```

Bild 16> Terminal Benutzer Oberfläche con Borg

7.2 Installation und Konfiguration von ClamAV und ClamAV-Daemon

Mit **ClamAV** wurde sich für eine etablierte und renommierter Open-Source Anti Virus Lösung entschieden welche sich durch eine sehr hohe Zuverlässigkeit und nahezu tägliche Updates auszeichnet. Die Installation erfolgt auch hier wieder über die Kommandozeile, um den ClamAV On Demand Scanner durch eine Echtzeit komponente (**ClamAV-Daemon**) zu ergänzen wird diese im selben Zuge mit installiert und im Anschluss konfiguriert. Hier nun der Befehl für diese Operation:

```
sudo apt install clamav clamav-daemon
sudo freshclam (zur Aktualisierung der Signaturen der Virus Datenbank)
sudo systemctl enable clamav-daemon
sudo apt start clamav-daemon
sudo apt status clamav-daemon
```

```
vagrant@ubuntu-bionic:~$ sudo systemctl status clamav-daemon | lolcat
● clamav-daemon.service - Clam AntiVirus userspace daemon
  Loaded: loaded (/lib/systemd/system/clamav-daemon.service; enabled; vendor preset: enabled)
  Drop-In: /etc/systemd/system/clamav-daemon.service.d
            └─extend.conf
    Active: active (running) since Tue 2024-10-01 06:45:56 UTC; 2min 22s ago
      Docs: man:clamd(8)
             man:clamd.conf(5)
             https://docs.clamav.net/
   Process: 1008 ExecStartPre=/bin/chown clamav /run/clamav (code=exited, status=0/SUCCESS)
   Process: 827 ExecStartPre=/bin/mkdir -p /run/clamav (code=exited, status=0/SUCCESS)
 Main PID: 1053 (clamd)
    Tasks: 2 (limit: 2355)
   CGroup: /system.slice/clamav-daemon.service
           └─1053 /usr/sbin/clamd --foreground=true

Oct 01 06:46:17 ubuntu-bionic clamd[1053]: Tue Oct  1 06:46:17 2024 -> Portable Executable support enabled.
Oct 01 06:46:17 ubuntu-bionic clamd[1053]: Tue Oct  1 06:46:17 2024 -> ELF support enabled.
Oct 01 06:46:17 ubuntu-bionic clamd[1053]: Tue Oct  1 06:46:17 2024 -> Mail files support enabled.
Oct 01 06:46:17 ubuntu-bionic clamd[1053]: Tue Oct  1 06:46:17 2024 -> OLE2 support enabled.
Oct 01 06:46:17 ubuntu-bionic clamd[1053]: Tue Oct  1 06:46:17 2024 -> PDF support enabled.
Oct 01 06:46:17 ubuntu-bionic clamd[1053]: Tue Oct  1 06:46:17 2024 -> SWF support enabled.
Oct 01 06:46:17 ubuntu-bionic clamd[1053]: Tue Oct  1 06:46:17 2024 -> HTML support enabled.
Oct 01 06:46:17 ubuntu-bionic clamd[1053]: Tue Oct  1 06:46:17 2024 -> XMLDOCS support enabled.
Oct 01 06:46:17 ubuntu-bionic clamd[1053]: Tue Oct  1 06:46:17 2024 -> HWP3 support enabled.
Oct 01 06:46:17 ubuntu-bionic clamd[1053]: Tue Oct  1 06:46:17 2024 -> Self checking every 3600 seconds.
vagrant@ubuntu-bionic:~$
```

Die Eingabe des Status Befehls sollte nun die folgende Ausgabe zurückgeben. **Bild 17**

Bild 17. ClamAV-daemon ist Aktiv und der Echtzeitschutz des Systems und des LDAP-Servers dadurch sichergestellt, auch die Automatisierte Aktualisierung der Signatur Datenbank wird hierdurch gewährleistet.

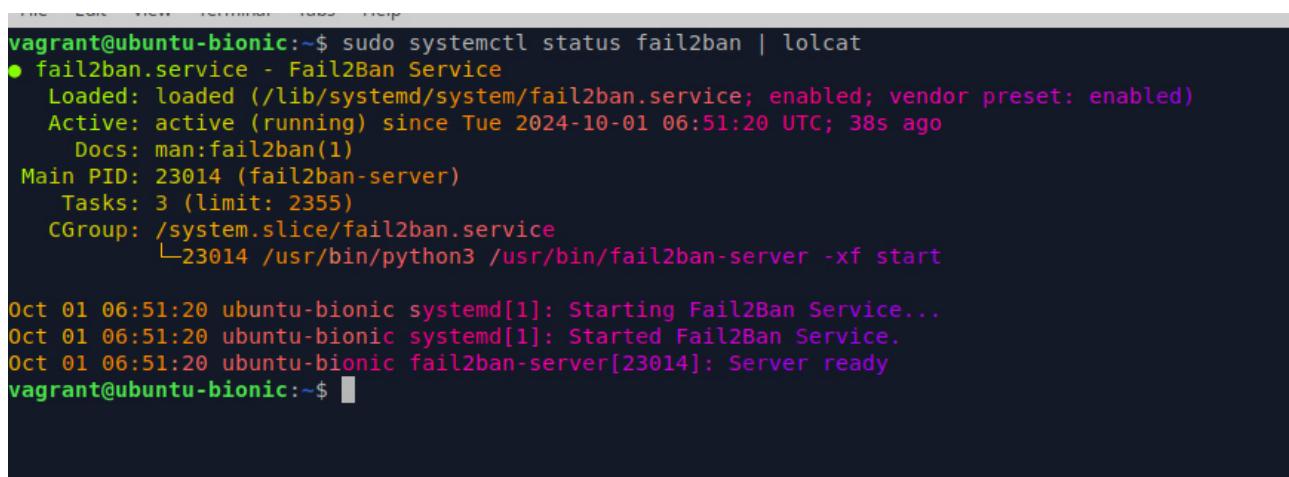
Durch den Einsatz von ClamAV als „on Demand“ Scanner und ClamAV-Daemon als „Realtime“ Monitoring Scanner welcher jedes Prozess welcher auf dem System gestartet wird bei dessen Ausführung überwacht und der permanenten Aktualisierung der Signatur Datenbank sowie der fortlaufenden Weiterentwicklung und Verbesserung dieses Produktes wird das System nun gut geschützt. Um jedoch ein Leistungsstarkes und fortschrittliches Sicherheitskonzept zu haben werden nun noch weitere Komponenten installiert und konfiguriert, um das System unter anderem auch gegen „Brute-Force“ und Root-Kits abzusichern.

7.3 Implementierung von Fail2ban zum Schutz vor Brute-Force-Angriffen

Auch **Fail2ban** ist ein Open-Source-Sicherheitstool, das dazu dient, **Brute-Force-Angriffe** auf Server zu verhindern. Es überwacht Logdateien auf verdächtige Aktivitäten, wie wiederholte fehlgeschlagene Anmeldeversuche, und **sperrt** automatisch IP-Adressen, die als Bedrohung identifiziert werden. Fail2ban ist konfigurierbar und kann für verschiedene Dienste wie

SSH, HTTP und FTP eingesetzt werden, um die Sicherheit von Systemen zu erhöhen. Durch die Reduzierung unerwünschter Zugriffsversuche verbessert es den Schutz vor unbefugtem Zugriff erheblich. Die Befehle für die Kommandozeile zur Installation und Konfiguration lauten wie folgt:

```
sudo apt install fail2ban -y  
sudo systemctl enable fail2ban  
sudo systemctl start fail2ban  
sudo systemctl status fail2ban
```



```
vagrant@ubuntu-bionic:~$ sudo systemctl status fail2ban | lolcat  
● fail2ban.service - Fail2Ban Service  
   Loaded: loaded (/lib/systemd/system/fail2ban.service; enabled; vendor preset: enabled)  
   Active: active (running) since Tue 2024-10-01 06:51:20 UTC; 38s ago  
     Docs: man:fail2ban(1)  
 Main PID: 23014 (fail2ban-server)  
    Tasks: 3 (limit: 2355)  
   CGroup: /system.slice/fail2ban.service  
         └─23014 /usr/bin/python3 /usr/bin/fail2ban-server -xf start  
  
Oct 01 06:51:20 ubuntu-bionic systemd[1]: Starting Fail2Ban Service...  
Oct 01 06:51:20 ubuntu-bionic systemd[1]: Started Fail2Ban Service.  
Oct 01 06:51:20 ubuntu-bionic fail2ban-server[23014]: Server ready  
vagrant@ubuntu-bionic:~$ █
```

Bild 18. Erwartete Ausgabe von `sudo systemctl status fail2ban`

7.4 Einsatz von RkHunter zur Rootkit-Erkennung

RKHunter (Rootkit Hunter) ist ein wichtiges Sicherheitswerkzeug für Unix-basierte Systeme, das sich auf die Aufspürung von **Rootkits** und **Malware** spezialisiert hat. Durch die Durchführung von Systemanalysen entdeckt es verdächtige Dateien und potenzielle Sicherheitsanfälligkeiten, die auf kompromittierte Systeme hinweisen könnten. RKHunter trägt dazu bei, die Sicherheit des Systems zu erhöhen, indem es Administratoren auf mögliche Bedrohungen aufmerksam macht und eine proaktive Überwachung ermöglicht.

Um RhHunter auf einem System zu installieren wird erneut die Benutzung der Kommandozeile erforderlich:

```
sudo apt install rkhunter -y
```

Zur Überprüfung des Systems auf Rootkits oder Maleware mithilfe von RkHunter wird in der Kommandozeile folgender Befehl ausgeführt:

```
sudo rkhunter -check
```

Es wird empfohlen RkHunter möglichst frühzeitig nach Installation des eigentlichen Betriebssystems zu installieren und eine Überprüfung durchzuführen um die **Falsch Positiven** Ergebnisse zu Dokumentieren und für spätere Vergleiche griffbereit zu halten.

7.5 Konfiguration von Snort als Intrusion Detection System (IDS)

Wie auch alle anderen von uns Implementierten Software-Lösungen ist auch **Snort** ein Open-Source Produkt. **Snort** ist ein Open-Source Intrusion Detection System (IDS) und

Intrusion Prevention System (IPS), das Netzwerkverkehr in Echtzeit analysiert und verdächtige Aktivitäten identifiziert. Es arbeitet paketbasiert und nutzt eine Regelbasierte Sprache, um Angriffe zu erkennen, die von bekannten Bedrohungen bis hin zu anomalem Verhalten reichen. Snort ist hochgradig konfigurierbar und kann sowohl zur Überwachung von Netzwerken als auch zur Analyse von Protokollen verwendet werden, was es zu einem unverzichtbaren Werkzeug für die Netzwerksicherheit macht.

Um Snort nun auf unserem Server zu installieren geben wir uns erneut in unseren „Happy Place“ der Linux Kommandozeile und geben dort nun folgenden Befehl ein:

```
sudo apt install snort -y
```

Im Laufe des Installation Prozesses wird der Benutzer zur Konfiguration aufgefordert, der wichtigste Schritt ist hier die Eingabe des korrekten Netzwerkinterfaces welches über den Befehl `ip a` sehr leicht zu bestimmen ist.

Abschließend zur Installation muss auch hier der Service Aktiviert, gestartet sowie sein Status überprüft werden:

```
sudo systemctl enable snort
sudo systemctl start snort
sudo systemctl status snort
```

```
vagrant@ubuntu-bionic:~$ sudo systemctl status snort | lolcat
● snort.service - LSB: Lightweight network intrusion detection system
  Loaded: loaded (/etc/init.d/snort; generated)
  Active: active (running) since Tue 2024-10-01 06:53:24 UTC; 1min 52s ago
    Docs: man:systemd-sysv-generator(8)
   Tasks: 2 (limit: 2355)
  CGroup: /system.slice/snort.service
          └─23794 /usr/sbin/snort -m 027 -D -d -l /var/log/snort -u snort -c /etc/snort/snort.conf -S HOME_NET=[192.168.0.0/16] -i enp0s8

Oct 01 06:53:24 ubuntu-bionic snort[23794]:      Preprocessor Object: SF_SSLPP Version 1.1 <Build 4>
Oct 01 06:53:24 ubuntu-bionic snort[23794]:      Preprocessor Object: SF_DCERPC2 Version 1.0 <Build 3>
Oct 01 06:53:24 ubuntu-bionic snort[23794]:      Preprocessor Object: SF_MODBUS Version 1.1 <Build 1>
Oct 01 06:53:24 ubuntu-bionic snort[23794]:      Preprocessor Object: SF_DNP3 Version 1.1 <Build 1>
Oct 01 06:53:24 ubuntu-bionic snort[23794]:      Preprocessor Object: SF_GTP Version 1.1 <Build 1>
Oct 01 06:53:24 ubuntu-bionic snort[23794]:      Preprocessor Object: SF_SMTP Version 1.1 <Build 9>
Oct 01 06:53:24 ubuntu-bionic snort[23794]:      Preprocessor Object: SF_SSH Version 1.1 <Build 3>
Oct 01 06:53:24 ubuntu-bionic snort[23794]:      Preprocessor Object: SF_SDF Version 1.1 <Build 1>
Oct 01 06:53:24 ubuntu-bionic snort[23794]:      Preprocessor Object: SF_IMAP Version 1.0 <Build 1>
Oct 01 06:53:24 ubuntu-bionic snort[23794]: Commencing packet processing (pid=23794)
vagrant@ubuntu-bionic:~$
```

Bild 19 Erwartete Ausgabe des Befehls `sudo systemctl status snort`

7.6 Firewall-Konfiguration mit UFW (Uncomplicated Firewall)

UFW (Uncomplicated Firewall) ist eine benutzerfreundliche Schnittstelle zur Verwaltung von iptables, die speziell für die einfache Konfiguration von Firewalls auf Linux-Systemen entwickelt wurde. UFW ist standardmäßig auf Ubuntu Bionic vorinstalliert, was die Inbetriebnahme erleichtert. Mit klaren und einfachen Befehlen ermöglicht UFW Administratoren, spezifische Regeln für eingehenden und ausgehenden Netzwerkverkehr festzulegen und so die Sicherheit des Systems zu erhöhen.

Für die Konfiguration der UFW wurde auch hier ein Skript von uns angefertigt, welches diese Aufgabe automatisiert. Das Skript wird mit dem Befehl

sudo bash ufw_config.sh

ausgeführt und gibt die Konfiguration als Text in der Kommandozeile zurück <Bild 19>

To	Action	From
-	-----	---
889/tcp	ALLOW IN	Anywhere
536/tcp	ALLOW IN	Anywhere
3310/tcp	ALLOW IN	Anywhere
22/tcp	ALLOW IN	Anywhere
2375/tcp	ALLOW IN	Anywhere
22/tcp (OpenSSH)	ALLOW IN	Anywhere
139	ALLOW IN	192.168.56.0/24
445	ALLOW IN	192.168.56.0/24
30/tcp	ALLOW IN	Anywhere
443/tcp	ALLOW IN	Anywhere
55000/tcp	ALLOW IN	Anywhere
889/tcp (v6)	ALLOW IN	Anywhere (v6)
536/tcp (v6)	ALLOW IN	Anywhere (v6)
3310/tcp (v6)	ALLOW IN	Anywhere (v6)
22/tcp (v6)	ALLOW IN	Anywhere (v6)
2375/tcp (v6)	ALLOW IN	Anywhere (v6)
22/tcp (OpenSSH (v6))	ALLOW IN	Anywhere (v6)
30/tcp (v6)	ALLOW IN	Anywhere (v6)
443/tcp (v6)	ALLOW IN	Anywhere (v6)
55000/tcp (v6)	ALLOW IN	Anywhere (v6)

Bild 20. Die Konfiguration der UFW auf für den LDAP-Servern

7.7 Implementierung von AppArmor zur Anwendungssicherheit

AppArmor ist ein Sicherheitsmodul für Linux, das auf der Mandatory Access Control (MAC) basiert und die Ausführung von Anwendungen durch Profile einschränkt. Es bietet eine zusätzliche Schutzschicht, indem es definiert, welche Ressourcen (wie Dateien, Netzwerkverbindungen und Systemaufrufe) eine Anwendung nutzen darf, wodurch potenzielle Schäden durch Sicherheitsverletzungen oder Malware minimiert werden. AppArmor ist standardmäßig auf Ubuntu vorinstalliert, was eine einfache Inbetriebnahme und Integration in bestehende Sicherheitsmaßnahmen ermöglicht. Durch die Erstellung und Anpassung von Profilen können Administratoren den Zugriff von Anwendungen präzise steuern und die Sicherheitsrichtlinien des Systems effektiv umsetzen.

AppArmor wurde von und mittels eines Python Skriptes entsprechend den Anforderungen unseres LDAP-Servers konfiguriert. Der Befehl lautet ***sudo python3 aa_config.py***

7.8 Konfiguration von Suricata als leistungsstarkes IDS/IPS

Den Abschluss der von unsrer Seite installierten und konfigurierten Sicherheitssystemen bildet mit Suricata ein weiteres bekanntes und weitverbreitetes Produkt. **Suricata** ist ein leistungsstarkes Open-Source-Netzwerk- und Intrusion-Detection-System (IDS), das als Sicherheitslösung für die Überwachung und Analyse des Netzwerkverkehrs eingesetzt wird. Es ermöglicht die Erkennung von Bedrohungen, Angreifern und bösartigen Aktivitäten in Echtzeit. Suricata analysiert den Netzwerkverkehr in Echtzeit und bietet Funktionen wie Protokollierung, Netzwerkanalyse, Verkehrsinspektion und Bedrohungserkennung. Dank seiner hohen Flexibilität unterstützt es sowohl die Signaturerkennung als auch die Verhaltensanalyse und kann in Kombination mit anderen Sicherheitslösungen eingesetzt werden, um eine umfassende Sicherheitsstrategie zu gewährleisten. Zur Installation dürfen wir erneut unsre heißgeliebte Linux Kommandozeile aufsuchen und voller Begeisterung und mit höchstem Elan folgenden Befehl dort eingeben:

```
sudo apt install suricata
```

Da es uns auch eine enorme Freude bereitet Prozesse zu automatisieren und wir uns nicht nur in der Linux Kommandozeile wie zuhause fühlen sondern ebenso gerne Zeit in unserem Zweit Wohnsitz dem Vim Editor verbringen, entschlossen wir uns ein weiteres Skript zur Konfiguration von suricata zu schreiben. Um dieses Bash Skript auszuführen wird der folgende Befehl im Terminal eingegeben:

```
sudo bash suricata_config.sh
```

Dieses Skript übernimmt nun nicht nur die von uns spezifizierte Konfiguration, sondern aktiviert und startet den Service auch. Das Skript endet bei erfolgreicher Anwendung mit der folgenden Ausgabe. **< Bild 21 >**

```
2024-10-02 14:11:43 (8.76 MB/s) - 'emerging-all.rules' saved [31921769/31921769]

Adding rule files to the configuration...
Starting Suricata...
Synchronizing state of suricata.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable suricata
Checking the status of Suricata...
● suricata.service - Suricata IDS/IDP daemon
   Loaded: loaded (/lib/systemd/system/suricata.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2024-10-02 14:07:57 UTC; 3min 46s ago
     Docs: man:suricata(8)
           man:suricatasc(8)
           https://redmine.openinfosecfoundation.org/projects/suricata/wiki
   Main PID: 5001 (Suricata-Main)
      Tasks: 8 (limit: 4915)
        CGroup: /system.slice/suricata.service
                 └─5001 /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml --pidfile /var/run/suricata.pid

Oct 02 14:07:57 ubuntu-bionic systemd[1]: Starting Suricata IDS/IDP daemon...
Oct 02 14:07:57 ubuntu-bionic suricata[4981]: 2/10/2024 -- 14:07:57 - <Notice> - This is Suricata version 3.2 RELEASE
Oct 02 14:07:57 ubuntu-bionic suricata[4981]: 2/10/2024 -- 14:07:57 - <Warning> - [ERRCODE: SC_ERR_SYSCALL(50)] - Failure when trying to bind to port 5000 on interface 'eth0': N
```

Bild 21. Erwartete Ausgabe, Suricata ist Aktiv.

Für die Zukunft ist geplant den **LDAP-Server** und die Domäne **hamster.panzer** durch Integration in das **Wazuh-SIEM** Zentral überwachen zu können.

Durch die Implementation und Konfiguration der von uns gewählten Komponenten ist das Unternehmen **Smooth Beans GmbH** bestens aufgestellt und der LDAP-Server kann als sehr sicher betrachtet werden. Diese Maßnahmen werden durch die in den nachfolgenden Kapiteln behandelnde Themen noch abgerundet.

7.9 MFA (Multi-Faktor-Authentifikation) und Password Policy

Die Implementierung von **MFA (Multi-Faktor-Authentifikation)** als zusätzliche Sicherheitsmaßnahme wird als gegeben angesehen, und wird Systemweit wo möglich und verfügbar von allen Angestellten angewendet, eben so wird eine Starke Passwort Policy erzwungen und die Anwendung von Passwort Managern wird strikt umgesetzt.

8. Incident Response Management

8.1 Entwicklung eines Incident Response Plans (IRP)

Präambel

Der hier vorgestellte Incident Response Plan legt jene Verfahren und Maßnahmen fest, welche nötig sind um auf sicherheitsrelevante Vorfälle effektiv reagieren zu können. Er wurde gezielt für den LDAP-Server der Smooth Beans GmbH, dessen Installation und Konfiguration Teil dieses Projektes waren, entwickelt.

Der LDAP-Server ist die Admin-Domäne mit dem Namen „hamster.panzer“ und im internen Netzwerk unter der IP-Adresse: 192.168.56.100 auf einem Ubuntu-Biotic Server zu finden. Dieser Server stellt Authentifizierungs- und Verzeichnisdienste für sämtliche Unternehmensgruppen und -benutzer bereit.

Ziel des IRP

Durch diesen Incident Response Plan wird sichergestellt, dass bei sicherheitsrelevanten Vorfällen eine zügige und effektive Reaktion eingeleitet und potentielle Schäden hierdurch minimiert und der Betrieb schnellstmöglich wiederhergestellt werden kann. Die Implementation eines IRP stellt eine essentielle Komponente des Sicherheitskonzeptes dar.

Diese Dokumentation bietet eine kurze Übersicht über die wesentlichen Inhalte des von uns erarbeiteten Incident Response Plans, welcher ebenfalls als Leitfaden im Appendix zu finden ist.

Verantwortlichkeiten und Personen

Verantwortlichkeiten	Beschreibung
LDAP-Administrator	Hauptansprechpartner für LDAP-bezogene Vorfälle, verantwortlich für Konfiguration und Zugriffsverwaltung.
IT-Leitung	Entscheidungsträger für Eskalationen und Koordination des technischen Teams.
Wazuh-Administrator	Zuständig für Monitoring und Intrusion Detection mit Wazuh
Web-Administrator	Verantwortlich für den Webserver
Geschäftsführung	Informiert über schwerwiegende Vorfälle, Entscheidungsträger bei strategischen Entscheidungen. Rechtliche Verantwortung.

Alle Mitarbeiter	Melden von potentiellen Sicherheitsrelevanten Vorfällen
-------------------------	--

<Tabelle 2> Verantwortlichkeiten und Ressourcen

<Tabelle 2> soll einen Überblick über die Verantwortlichkeiten des Personals und insbesondere des Management geben. Wir halten es für erforderlich an dieser Stelle darauf hinzuweisen, dass die Gesamtverantwortung sowohl was Innerbetriebliche aber auch rechtliche Aspekte betrifft stets bei der **Geschäftsleitung** liegt und von dieser nicht delegiert werden kann. Das Personal ist angewiesen bei Verdacht auf Sicherheitsrelevante Vorfälle unverzüglich die IT-Abteilung zu informieren, welche diese gewissenhaft zu prüfen und entsprechende Schritte einzuleiten hat.

8.2 Wichtige Dienste auf dem LDAP-Server

In <Tabelle 3> werden nun nochmals die wichtigsten Dienste zusammengefasst, welche für den sicheren Betrieb des LDAP-Servers essentiell sind. Dies umfasst alle Sicherheitsrelevanten Dienste, alle LDAP Dienste sowie das Betriebssystem und deren Funktion.

Dienst	Beschreibung
SSH	Remote-Zugriff auf den Server
SLAPD	LDAP-Dienst für Authentifizierung
AppArmor	Sicherheitsmodul zur Kontrolle von Programmzugriffen
Fail2Ban	Schutz vor Brute-Force-Angriffen und bannen von verdächtigen IP-Adressen
ClamAV/clamav-daemon	Virenschutz zur Erkennung und Beseitigung von Maleware
Snort/Suricata	Netzwerkanalyse und Intrusion Detection und Intrusion Prevention
RkHunter	Überprüfung auf Rootkits und Maleware
UFW (Uncomplicated Firewall)	Firewall zum Schutz des Systems
Ubuntu-Biotic (Ubuntu Linux)	Betriebssystem des LDAP-Servers
NTP (Network Time Protocol)	Sicherstellung der korrekten Uhrzeit/Datum auf allen Endgeräten der Domäne
OpenVPN	VPN-Service nötig damit Remote-Mitarbeiter Zugriff auf das interne Netzwerk haben
Rsyslog (Syslog)	Zentrale Protokollierung von System- und Anwendungslogs
Wazuh-Agent	Zur zentralen Analyse aller Integrierten dienste über den Wazuh Manager
Logrotate	Automatisiert das Management von Log-Dateien, um das Dateisystem vor überfüllten Logs zu schützen

<Tabelle 3> Dienste auf dem LDAP-Server im Überblick

Vorgehensweise im Falle eines Vorfalls

Erkennung und Initiale Analyse

Die Erkennung eines Vorfalls kann durch folgende Methoden erfolgen:

- Alarme von **Wazuh**, **Suricata** oder **Snort** über verdächtige Aktivitäten oder Eindringversuche.
- Meldungen von **Fail2Ban** bei wiederholten, erfolglosen Anmeldeversuchen.
- Auffälligkeiten bei Benutzeraktivitäten, welche auf unberechtigte Zugriffe hindeuten könnten, z.B. unerklärliche LDAP_Authentifizierungen oder Änderungen.
- **ClamAV** oder **RkHunter** melden erkannte Maleware oder Rootkits.
- Plötzliche änderungen in der Reaktionszeit des Systems, der Anwendungen, oder unerklärliche neue und unbekannte Prozesse, neue Software, welche bis dato nicht auf dem System vorhanden waren.

Schritte

- Information des zuständigen Personals (IT-Leitung, LDAP-Administrator oder IT-Abteilung).
- Sammeln und Auswerten der Log-Dateien zur Klärung der Ursache und des Umfangs des Vorfalls.
- Überprüfung der Netzwerkverbindungen und Zugriffe zur Identifikation verdächtiger Aktivitäten
- Bestätigung des Vorfalls oder Schließen des Vorfalls abhängig der Analyse

Eindämmung

Sobald ein Vorfall bestätigt wurde, sind folgende Schritte zur Eindämmung durchzuführen:

- **SSH-Zugriff deaktivieren** oder beschränken, falls unberechtigte Zugriffe erkannt werden.
- **UFW-Regeln anpassen**, um verdächtige IP-Adressen oder Ports zu blockieren.
- **Fail2ban-Konfiguration verschärfen**, um wiederholte Angriffe besser abzuwehren.
- Überprüfung und Sperrung von kompromittierten Benutzerkonten (LDAP-Benutzer).
- Bei Verdacht auf Malware **ClamAV** und **rkhunter** ausführen, um potenziell infizierte Dateien oder Prozesse zu isolieren.

Behebung

Nach Identifikation des Vorfalls und der Ursachen werden diese beseitigt:

Patches für Schwachstellen im Betriebssystem oder in Anwendungen einspielen.

- LDAP-Konfigurationen überprüfen und sichern, um unberechtigte Änderungen rückgängig zu machen.
- Ggf. kompromittierte Benutzerkennwörter zurücksetzen und die Zugriffsrechte überprüfen.

- Bei Malware-Befall infizierte Dateien oder Softwarekomponenten entfernen und das System bereinigen.

Wiederherstellung

Nach der Behebung werden folgende Schritte durchgeführt, um den Normalbetrieb wiederherzustellen:

- Server- und Dienstfunktionen, einschließlich LDAP, werden wieder aktiviert.
- Überwachung wird intensiviert, um sicherzustellen, dass keine weiteren verdächtigen Aktivitäten stattfinden.
- Alle Systeme und Dienste werden auf die neueste Version aktualisiert, falls dies zuvor nicht geschehen ist.

Nachverfolgung und Dokumentation

Nach jedem Vorfall ist eine gründliche **Analyse** durchgeführt:

- Erstellung eines Vorfallsberichts, der die Ursache, den Verlauf und die getroffenen Maßnahmen dokumentiert.
- Überprüfung der Protokolle und Anpassung von Sicherheitsrichtlinien, um ähnliche Vorfälle in Zukunft zu verhindern.
- Schulung des IT-Personals zu neu erkannten Risiken und Verbesserung der Reaktionsprozesse.

Kommunikation

Bei Vorfällen aller Art muss die Geschäftsführung sofort informiert werden. Externe Stellen, wie Datenschutzbehörden oder Kunden, werden bei Datenlecks umgehend benachrichtigt. Die Kommunikation erfolgt nur durch autorisierte Personen, um Informationslecks zu verhindern.

Prävention

Um zukünftige Vorfälle zu vermeiden, werden folgende Maßnahmen ergriffen:

- Regelmäßige Überprüfung von Logs und Audit-Trails.
- Regelmäßige Sicherheitsupdates für Ubuntu und alle installierten Dienste (LDAP, SSH, UFW etc.).
- Durchführung von Penetrationstests auf dem LDAP-Server, um Schwachstellen frühzeitig zu erkennen.
- Schulung der Benutzer in Bezug auf Passwortsicherheit und das Erkennen von Phishing-Angriffen.

8.2 Erstellung von Incident Report-Vorlagen

Um die Kommunikation von Sicherheitsvorfällen effizient zu gestalten wurden von uns **Incident Report** Vorlagen erstellt, diese sind ebenfalls im Appendix dieser Dokumentation zu finden, an dieser Stelle wird nur kurz darauf eingegangen, wann welcher Report zu verwenden ist.

8.2.1 Standard-Report für Sicherheitsvorfälle

Der **Standard-Report** für Sicherheitsvorfälle wird eingesetzt, wenn ein sicherheitsrelevanter Vorfall erkannt wird und eine umfassende Analyse der Ereignisse erforderlich ist. Er dient dazu, die Auswirkungen des Vorfalls zu dokumentieren, die Ursachen zu identifizieren und Maßnahmen zur Vermeidung ähnlicher Vorfälle in der Zukunft zu formulieren.

8.2.2 Report für Systemfehler

Der **Report für Systemfehler** wird eingesetzt, wenn technische Störungen oder Fehler im System auftreten, die den ordnungsgemäßen Betrieb beeinträchtigen. Dieser Report dient dazu, die Fehlerursachen zu analysieren, die Auswirkungen auf die Systemleistung zu dokumentieren und geeignete Korrekturmaßnahmen zu entwickeln.

8.2.3 Report für Datenschutzverletzungen

Der **Report für Datenschutzverletzungen** wird eingesetzt, wenn ein Vorfall festgestellt wird, der zu einer unbefugten Offenlegung oder Verarbeitung personenbezogener Daten führt. Dieser Report dient dazu, die Art und den Umfang der Verletzung zu dokumentieren, die betroffenen Personen zu informieren und Maßnahmen zur Minderung von Risiken sowie zur Einhaltung gesetzlicher Vorgaben zu ergreifen.

8.3 E-Mail-Vorlagen zur Kommunikation von Cyber-Incidents an Mitarbeiter

Die von uns bereitgestellten **E-Mail-Vorlagen zur Kommunikation von Cyber-Incidents** an Mitarbeiter dienen dazu, eine einheitliche und klare Informationsweitergabe im Falle eines Sicherheitsvorfalls sicherzustellen. Sie enthalten grundlegende Informationen über den Vorfall, die potenziellen Auswirkungen, die erforderlichen Maßnahmen für die Mitarbeiter und Ansprechpartner für Rückfragen. Durch die Verwendung solcher Vorlagen wird die Kommunikation effizienter und trägt dazu bei, Missverständnisse zu vermeiden sowie das Vertrauen der Mitarbeiter in die Sicherheitsmaßnahmen des Unternehmens zu stärken.

Bedrohungsmodellierung

9.1 Einführung in die STRIDE-Methode

STRIDE wird häufig in Kombination mit anderen Sicherheitsmethoden eingesetzt, um ein umfassendes Bild der Sicherheitslage eines Systems zu erhalten. Durch die Anwendung der STRIDE-Kategorisierung können Entwickler und Sicherheitsteams proaktiv Maßnahmen zur Risikominderung ergreifen.

Die **STRIDE-Methode** ist ein bewährtes Framework zur Bedrohungsmodellierung, das in der Software- und Systemsicherheit verwendet wird. Sie hilft, potenzielle Sicherheitsrisiken systematisch zu identifizieren, indem sie sechs Arten von Bedrohungen klassifiziert:

Spoofing: Fälschung der Identität eines Benutzers oder Systems.

- **Tampering:** Unbefugte Änderung von Daten oder Systemkomponenten.

- **Repudiation:** Das Abstreiten von Aktionen, die ein Benutzer ausgeführt hat, ohne dass dies nachgewiesen werden kann.
- **Information Disclosure:** Unbefugte Offenlegung vertraulicher Informationen.
- **Denial of Service (DoS):** Unterbrechung des Zugriffs auf ein System oder einen Dienst.
- **Elevation of Privilege:** Unbefugter Zugriff auf privilegierte Funktionen oder Daten.

9.2 Verwendung von Threat Dragon NG zur Durchführung von Bedrohungsanalysen

Die Verwendung von **Threat Dragon NG** zur Durchführung von Bedrohungsanalysen ermöglicht eine strukturierte Identifikation und Bewertung von Sicherheitsrisiken in Softwarearchitekturen. Durch die grafische Modellierung von Bedrohungen und Schwachstellen wird die Kommunikation zwischen Entwicklern und Sicherheitsexperten verbessert, was zu einer effektiveren Risikoanalyse führt. Das ausgearbeitete Bedrohungsmodell ist im Anhang der Dokumentation enthalten und bietet eine detaillierte Grundlage zur Identifikation und Minderung spezifischer Bedrohungen.

10. Test und Validierung

10.1 Testansätze und -methoden

Für die Überprüfung der Implementierung des LDAP-Servers haben wir mehrere Testansätze und -methoden angewendet. Zunächst wurde der Status des LDAP-Dienstes durch den Befehl `systemctl status slapd` überprüft, um sicherzustellen, dass der Server ordnungsgemäß läuft und keine Fehler gemeldet werden. Darüber hinaus wurde die Integrität der Benutzer- und Gruppendatenbanken durch das Abrufen relevanter Datenbankinhalte mittels LDAP-Abfragen validiert. Diese Abfragen ermöglichen uns, die korrekte Konfiguration der Gruppen und Benutzer zu überprüfen und sicherzustellen, dass die erwarteten Einträge vorhanden sind. Zusätzlich wurde getestet, ob die Authentifizierung von LDAP-Benutzern erfolgreich erfolgt, indem SSH-Anmeldeversuche von domänenunabhängigen virtuellen Maschinen durchgeführt wurden. Dieser mehrschichtige Ansatz gewährleistet eine umfassende Überprüfung der Systemkonfiguration und der Funktionalität des LDAP-Servers.

10.2 Validierung der Implementierungsergebnisse

Die Validierung der Implementierungsergebnisse erfolgte durch eine Kombination von Überprüfungen und Tests, die die Funktionalität und Sicherheit des LDAP-Systems bestätigten. Nach der Überprüfung des Dienstestatus und der Datenbankintegrität wurde die Authentifizierungsmechanik durch SSH-Anmeldetests validiert, die von virtuellen Maschinen außerhalb der Domäne durchgeführt wurden. Diese Tests bestätigten, dass Benutzer mit den korrekten Berechtigungen erfolgreich auf das System zugreifen konnten. Zudem wurde sichergestellt, dass die Authentifizierung nicht nur funktioniert, sondern auch robust gegenüber nicht autorisierten Zugriffen ist. Die durchgeführten Validierungen sind dokumentiert und bieten eine solide Grundlage für die Beurteilung der Funktionsfähigkeit und Sicherheit der implementierten LDAP-Lösung.

Das nachfolgende Bild zeigt den Eintrag der „/etc/passwd“ welche validiert, dass die Benutzer erfolgreich im System angemeldet wurden.

```

sshd:x:0:65534::/run/sshd:/usr/sbin/nologin
pollinate:x:110:1::/var/cache/pollinate:/bin/false
vagrant:x:1000:1000:,,,:/home/vagrant:/bin/bash
ubuntu:x:1001:1001:Ubuntu:/home/ubuntu:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
openldap:x:111:116:OpenLDAP Server Account,,,:/var/lib/ldap:/bin/false
kernoops:x:112:65534:Kernel Oops Tracking Daemon,,,:/usr/sbin/nologin
rtkit:x:113:118:RealtimeKit,,,:/proc:/usr/sbin/nologin
avahi-autoipd:x:114:119:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:115:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
whoopsie:x:116:121::/nonexistent:/bin/false
lightdm:x:117:122:Light Display Manager:/var/lib/lightdm:/bin/false
avahi:x:118:124:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
cups-pk-helper:x:119:125:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin
saned:x:120:127::/var/lib/saned:/usr/sbin/nologin
speech-dispatcher:x:121:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/false
colord:x:122:128:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
hplip:x:123:7:HPLIP system user,,,:/var/run/hplip:/bin/false
geoclue:x:124:129::/var/lib/geoclue:/usr/sbin/nologin
pulse:x:125:130:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
clamav:x:126:132::/var/lib/clamav:/bin/false
postfix:x:127:133::/var/spool/postfix:/usr/sbin/nologin
snort:x:128:135:Snort IDS:/var/log/snort:/usr/sbin/nologin
markus.wagner:x:20001:10012:Markus Wagner:/home/users/markus.wagner:/bin/bash
lisa.neumann:x:20002:10012:Lisa Neumann:/home/users/lisa.neumann:/bin/bash
david.zimmermann:x:20003:10012:David Zimmermann:/home/users/david.zimmermann:/bin/bash
martin.kellerer:x:10013:10012:Martin Kellerer:/home/martin.kellerer:/bin/bash

```

System-Dienste LDAP & SSH

Security Software

Angelegte Nutzer

Bild 22. Die angelegten Benutzer /etc/passwd

Zusätzlich hierzu wurden die Benutzer über die grafische Benutzeroberfläche im System angemeldet um die Funktion weiter zu verifizieren.

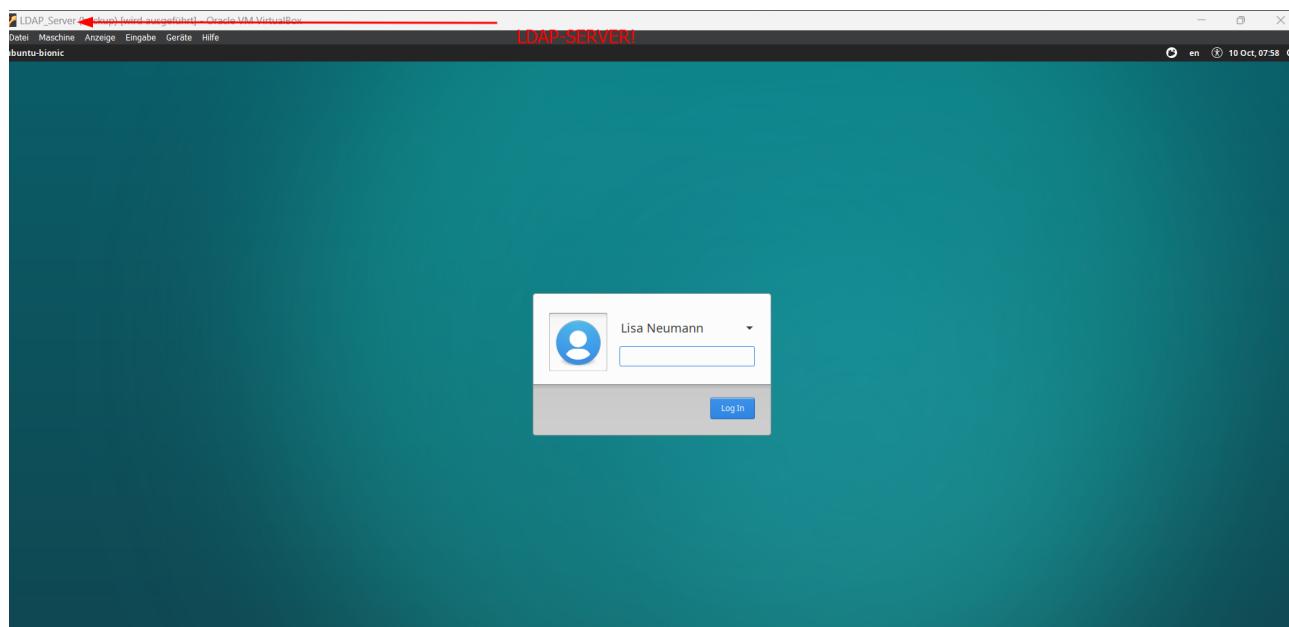
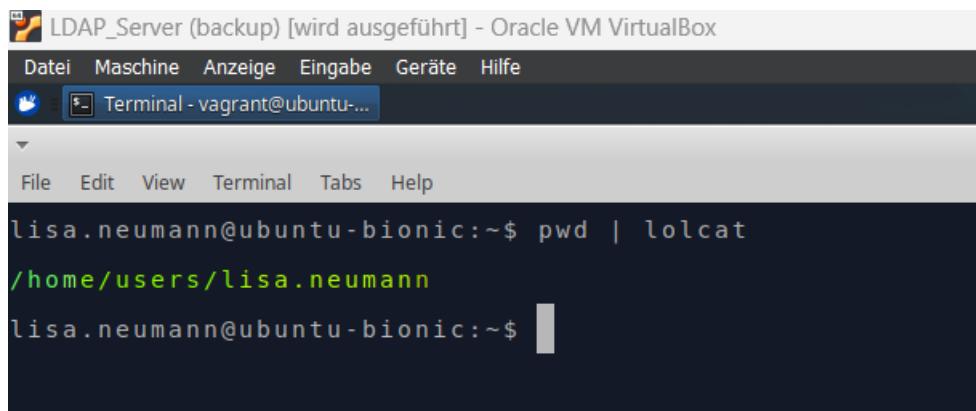


Bild 23. Login Screen der LDAP-Benutzerin „Lisa Neumann“

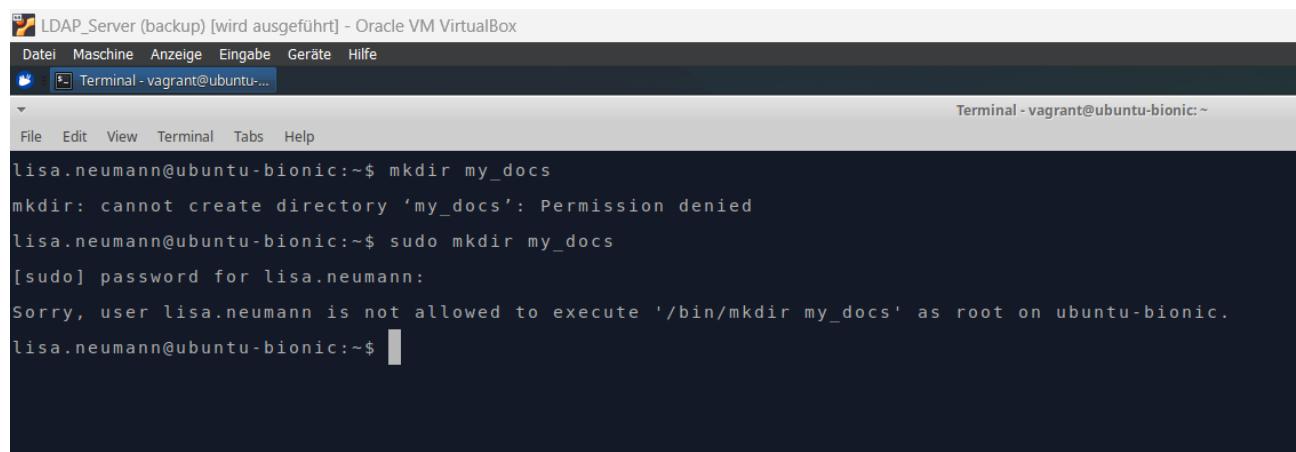
Die Bilder **Bild 24** und **Bild 25** belegen die korrekte Funktion der LDAP-Authentifizierung sowie die korrekte Anwendung der von uns erstellten und angewendeten

ACL's. <Bild 25> beweist, dass „Lisa Neumann“ sich erfolgreich auf dem LDAP-Dienst anmelden kann. <Bild 26> zeigt eindeutig das die Benutzerin „Lisa Neumann“ nicht über die nötigen Berechtigungen verfügt auf dem LDAP-Server Veränderungen wie in dem Beispiel einen Ordner anlegen vornehmen kann. Auch wird in diesem Bild nachgewiesen, dass sie nicht über sudo Rechte verfügt, wodurch die Sicherheit und Integrität des LDAP-Servers gefestigt wird.



```
lisa.neumann@ubuntu-bionic:~$ pwd | lolcat
/home/users/lisa.neumann
lisa.neumann@ubuntu-bionic:~$
```

Bild 24. Die LDAP-Benutzerin „Lisa Neumann“ ist erfolgreich eingeloggt.



```
lisa.neumann@ubuntu-bionic:~$ mkdir my_docs
mkdir: cannot create directory 'my_docs': Permission denied
lisa.neumann@ubuntu-bionic:~$ sudo mkdir my_docs
[sudo] password for lisa.neumann:
Sorry, user lisa.neumann is not allowed to execute '/bin/mkdir my_docs' as root on ubuntu-bionic.
lisa.neumann@ubuntu-bionic:~$
```

Bild 26. „Lisa Neumann“ kann keine Veränderungen auf dem LDAP-Server vornehmen und verfügt nicht über die nötigen Bechtigungen, den „sudo“ Befehl zu nutzen.

In <Bild 27> wird die Ausgabe des „auth.log“ abgebildet, was die korrekte Anwendung der Benutzer und deren Fähigkeit sich erfolgreich zu Authentifizieren weiter unterstreicht. Die korrekte Vergabe der Berechtigungen wurde weiter oben schon nachgewiesen.

```
vagrant@ubuntu-bionic:~/media/sf_Shared$ sudo tail -f /var/log/auth.log
Oct 10 13:58:53 ubuntu-bionic nscd: nss_ldap: failed to bind to LDAP server ldap://192.168.56.100: Invalid credentials
Oct 10 13:58:53 ubuntu-bionic nscd: nss_ldap: could not search LDAP server - Server is unavailable
Oct 10 13:58:53 ubuntu-bionic nscd: nss_ldap: failed to bind to LDAP server ldap://192.168.56.100: Invalid credentials
Oct 10 13:58:53 ubuntu-bionic nscd: nss_ldap: reconnecting to LDAP server...
Oct 10 13:58:53 ubuntu-bionic nscd: nss_ldap: failed to bind to LDAP server ldap://192.168.56.100: Invalid credentials
Oct 10 13:58:53 ubuntu-bionic nscd: nss_ldap: reconnecting to LDAP server (sleeping 1 seconds)...
Oct 10 13:58:54 ubuntu-bionic nscd: nss_ldap: failed to bind to LDAP server ldap://192.168.56.100: Invalid credentials
Oct 10 13:58:54 ubuntu-bionic nscd: nss_ldap: could not search LDAP server - Server is unavailable
Oct 10 13:58:54 ubuntu-bionic systemd: pam_ldap(systemd-user:account): error reading from nsldc: Connection reset by peer
Oct 10 13:58:54 ubuntu-bionic systemd: pam_unix(systemd-user:session): session opened for user martin.kellerer by (uid=0)
Oct 10 13:59:11 ubuntu-bionic sudo: vagrant : TTY=pts/3 ; PWD=/media/sf_Shared ; USER=root ; COMMAND=/usr/bin/tail -f /var/log/auth.log
Oct 10 13:59:11 ubuntu-bionic sudo: vagrant : pam_unix(sudo:session): session opened for user root by (uid=0)
Oct 10 13:59:22 ubuntu-bionic sshd[10203]: Received disconnect from 192.168.56.101 port 57228:11: disconnected by user
Oct 10 13:59:22 ubuntu-bionic sshd[10203]: Disconnected from user martin.kellerer 192.168.56.101 port 57228
Oct 10 13:59:22 ubuntu-bionic sshd[9958]: pam_unix(sshd:session): session closed for user martin.kellerer
Oct 10 13:59:22 ubuntu-bionic systemd-logind[783]: Removed session 8.
Oct 10 13:59:22 ubuntu-bionic systemd: pam_unix(systemd-user:session): session closed for user martin.kellerer
Oct 10 13:59:42 ubuntu-bionic sshd[10563]: pam_ldap(sshd:account): error reading from nsldc: Connection reset by peer
Oct 10 13:59:42 ubuntu-bionic sshd[10563]: Accepted password for unfa.ehig from 192.168.56.101 port 40128 ssh2
Oct 10 13:59:42 ubuntu-bionic sshd[10563]: pam_unix(sshd:session): session opened for user unfa.ehig by (uid=0)
Oct 10 13:59:42 ubuntu-bionic systemd-logind[783]: New session 10 of user unfa.ehig.
Oct 10 13:59:42 ubuntu-bionic systemd: pam_unix(systemd-user:session): session opened for user unfa.ehig by (uid=0)
```

Bild 27. Die Ausgabe des „auth.log“ als Nachweis der erfolgreichen Anmeldung, der Benutzer zur Validierung der erfolgreichen Erstellung, sowie der korrekten Protokollierung des Anmeldeprozesses.

11. Glossar

11.1 Fachbegriffe und Abkürzungen im Kontext

Fachbegriff/Abkürzung	Beschreibung
Active Directory (AD)	Microsoft Verzeichnisdienst zur Authentifizierung und Autorisierung von Benutzern und Computern in einem Netzwerk
Authorization	Ein Prozess, welcher bestimmt, ob ein authentifizierter Benutzer Zugriff auf bestimmte Ressourcen oder Funktionen hat.
LDAP (Lightweight Directory Access Protocol)	Ein offenes Protokoll zur Abfrage und Modifikation von Verzeichnisdiensten, wird häufig für Authentifizierung und Identitätsmanagement verwendet
SSO (Single Sign-On)	Authentifizierungsmethode, welche Benutzern ermöglicht sich einmal anzumelden um auf mehrere Anwendungen oder Systeme zuzugreifen
SSH (Secure Shell)	Netzwerkprotokoll für sicheren Fernzugriff auf Computer, verwendet Verschlüsselung und Authentifizierung
TLS (Transport Layer Security)	Kryptographisches Protokoll, einsatz für LDAP über SSL (LDAPS)
DN (Distinguished Name)	Ein eindeutiger Bezeichner für einen Eintrag in einem LDAP-Verzeichnis, welcher die Hierarchie des verzeichnisses

	widerspiegelt
RDN (Relative Distinguished Name)	Teil des DN, welcher den Eintrag innerhalb seines übergeordneten Objekts identifiziert
Bind	Der Prozess, bei dem ein Client sich mit einem LDAP-Server verbindet und sich authentifiziert.
Group	Sammlung von Benutzern in einem LDAP-Verzeichnis, die gemeinsame Berechtigungen oder Rollen haben.
Password Policy	Richtlinien, welche festlegen, wie Passwörter verwaltet werden müssen, z.B. Mindestlänge, Komplexität und Ablauf.
Replication	Prozess welcher Synchronisierung von Daten zwischen mehreren LDAP-Servern, um die Verfügbarkeit und Redundanz zu gewährleisten.
Access Control List (ACL)	Liste von Berechtigungen, die regelt, wer auf bestimmte Ressourcen im LDAP-Verzeichnis zugreifen kann.
Kerberos	Netzwerk-Authentifizierungsprotokoll, das auf geheimen Schlüsseln basiert und häufig in Verbindung mit LDAP verwendet wird.
Token	Ein kryptografisches Element, das Informationen zur Authentifizierung und Autorisierung enthält und in einigen Sicherheitsmodellen verwendet wird.
Public Key Infrastructure (PKI)	Framework zur Verwaltung von digitalen Zertifikaten und Public-Key-Verschlüsselung zur Sicherstellung sicherer Kommunikation.
Audit Log	Eine Aufzeichnung aller sicherheitsrelevanten Ereignisse und Aktionen in einem System, die zur Überprüfung und Fehlerbehebung verwendet wird.
Two-Factor Authentication (2FA)	Sicherheitsverfahren, bei dem zwei verschiedene Authentifizierungsmethoden verwendet werden, um die Identität eines Benutzers zu überprüfen.

<Tabelle 4> Glossar

12. Quellenverzeichnis

12.1 Relevante Literatur und technische Dokumentationen

- OpenLDAP Administrator's Guide
Offizielle Dokumentation zu OpenLDAP mit detaillierten Informationen zur Installation, Konfiguration und Verwaltung.
- Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map (RFC 4511)
Ein technischer Überblick über LDAP, der von der IETF bereitgestellt wird.

- **SSH - The Secure Shell: A Technical Overview**
Eine technische Einführung in SSH, die wichtige Sicherheitsaspekte und Anwendungsfälle behandelt.

- **LDAP: An Introduction**
Ein Leitfaden für die Grundlagen von LDAP, der auf die Struktur und Funktionsweise des Protokolls eingeht.
- **NIST Cybersecurity Framework**
Eine umfassende Ressource von NIST, die Richtlinien zur Verbesserung der Cybersicherheit bietet und in vielen IT-Umgebungen verwendet wird.
- **OWASP Top Ten**
Bietet Ressourcen und Informationen zu den Top Ten der Sicherheitsrisiken sowie zur Verbesserung der Anwendungssicherheit.
- **Kali Linux Documentation**
Eine umfassende Sammlung von Anleitungen und Dokumentationen zu Kali Linux, das häufig für Sicherheitstests verwendet wird.
- **Debian OpenLDAP Documentation**
Die offizielle Dokumentation für die Verwendung von OpenLDAP auf Debian-basierten Systemen.
- **The Linux Documentation Project**
Eine Sammlung von Dokumentationen und Handbüchern zu verschiedenen Linux-Themen, einschließlich Sicherheit und Netzwerkmanagement.
- **Free Software Foundation - GNU Privacy Guard (GPG)**
Informationen zu GPG, einer kostenlosen Implementierung des OpenPGP-Standards zur Datenverschlüsselung.

12.2 Online-Ressourcen und Community-Foren

- **OpenLDAP Mailing Lists**
Offizielle Mailinglisten für OpenLDAP, die eine Plattform für Diskussionen, Fragen und Informationen zu OpenLDAP bieten.
<https://www.openldap.org/lists/>
- **Stack Overflow**
Eine große Community von Entwicklern, in der Fragen zu verschiedenen Technologien, einschließlich LDAP und SSH, gestellt und beantwortet werden können.
<https://stackoverflow.com/>
- **Reddit - r/sysadmin**
Ein Subreddit für Systemadministratoren, in dem Informationen, Probleme und Lösungen im Bereich IT-Sicherheit und Netzwerkmanagement geteilt werden.
<https://www.reddit.com/r/sysadmin/>

- **Server Fault**

Eine Q&A-Website für Systemadministratoren, die Diskussionen und Lösungen zu Server- und Netzwerkproblemen bietet.

<https://serverfault.com/>

- **LDAP Support Forum**

Ein Forum speziell für LDAP-bezogene Themen, in dem Benutzer Fragen stellen und

- Lösungen diskutieren können.

<https://wwwldapguru.com/forum/>

- **Kali Linux Forums**

Ein Forum für Benutzer von Kali Linux, das Diskussionen über Sicherheitswerkzeuge, Techniken und Problemlösungen umfasst.

<https://forums.kali.org/>

- **The Linux Community**

Ein allgemeines Forum für Linux-Benutzer, das Diskussionen über alle Aspekte von Linux, einschließlich Sicherheit und Netzwerke, fördert.

<https://wwwlinux.org/forums/>

- **OpenSSH Mailing Lists**

Offizielle Mailinglisten für OpenSSH, auf denen Benutzer Fragen stellen und Informationen über SSH erhalten können.

<https://www.openssh.com/mailing.html>

- **GitHub - LDAP Projects**

Eine Sammlung von Open-Source-Projekten auf GitHub, die sich auf LDAP konzentrieren und von der Community unterstützt werden.

<https://github.com/topics/ldap>

- **NIST Cybersecurity Community**

Eine Plattform von NIST für Cybersecurity-Ressourcen, in der Benutzer Informationen und Tools für Sicherheitspraktiken finden können.

<https://csrc.nist.gov/>

13. Appendix

Die hier aufgeführten Dokumente sind im Ordner Appendix welcher im Verzeichnis dieser Dokumentation ist zu finden. Den Python und Bash Skripten liegt jeweils auch eine README.pdf als Anwendungshilfe bei.

13.1 Bedrohungsmodell

Das mit Theart-Dragon-ng erstellte Bedrohungsmodell für den LDAP-Server als PDF.

13.2 Benutzerhandbuch für die LDAP-Umgebung

PDF Dokument als Leitfaden zur Nutzung des LDAP-Servers.

13.3 Scripts und Automatisierungshilfen

- **base.ldif** Basisstruktur des LDAP
- **ou.ldif** Grundstruktur des LDAP

- **groups.ldif** Erstellt die Gruppen des LDAP
- **create_users.py** Erstellt die Benutzer des LDAP-Abfragen
- **permmissions.sh** Erstellt die ACL und wendet diese an.
- **conf_ufw.py** Konfiguriert die Firewall-Konfiguration

- **conf_suricata.sh** Konfiguriert Suricata
- **aa_config.py** Konfiguration von AppArmor

13.4 Incident Response Plan

Übersichtlicher IRP als PDF speziell für den LDAP-Server „hamster.panzer“

13.5 E-Mail-Vorlagen für die Incident-Kommunikation

Standardisierte Vorlagen zur Information der Belegschaft über Sicherheitsvorfälle.

13.6 Vorlagen für Incident Reports

Standardisierte Vorlagen zur Meldung von Vorfällen (Standard, Systemfehler und Datenschutzverletzung)

13.7 Vagrantfile

Datei welche mit Hilfe der Vagrant Software die von uns verwendete Maschine erstellt.

13.8 Rechnung

PDF Dokumentation mit konkreter Aufschlüsselung der Kosten und der Arbeitszeit.

14. Schlusswort

Die in dieser Dokumentation verwendeten, Diagramme, Bilder und Tabellen wurden von den Autoren selbst erstellt. Das Recht auf Nutzung für Schulungs- und zum Zwecke der Veröffentlichung wird hiermit ausdrücklich eingeräumt.

Sämtliche verwendete Software ist Open-Source und dauerhaft kostenlos nutzbar. Diese Tatsche ermöglichte uns es nicht nur einen sicheren sondern auch Kostengünstigen Server zu erstellen, welchen wir nun in die Hände des Auftraggebers **Smooth Beans GmbH** übergeben.

Ohne die Open-Source-Community wäre dieses Projekt nicht möglich gewesen, als kleines Dankeschön an diese wurde von unserem Team beschlossen, diese Dokumentation einschließlich sämtlicher Skripte, Dokumente und dem Vagrantfile auf GitHub öffentlich zugänglich zu machen, der Link zu diesem Repository ist
<https://github.com/n3M3Z1Z/LDAP-Project>.



Inhaltsverzeichnis

1. Präambel

- 1.1 Zielsetzung der Dokumentation
- 1.2 Systemanforderungen

2. Installation der Virtuellen Umgebung und Netzwerkplan

- 2.1 Installation des Ubuntu Servers in Virtualbox mit Vagrant
- 2.2 Installation von Xubuntu-Desktop
- 2.3 Netzwerkplan

3. Einführung in LDAP

- 3.1 Grundlagen von LDAP (Lightweight Directory Access Protocol)
- 3.2 Anwendungsfälle und Architektur
- 3.3 Schlüsselbegriffe und -konzepte

4. Installation von LDAP

- 4.1 Installation der OpenLDAP-Komponenten
- 4.2 Verifizierung der LDAP-Installation

5. Konfiguration von LDAP

- 5.1 Einrichtung der Domäne hamster.panzer
- 5.2 Erstellung und Verwaltung von Benutzer- und Gruppenkonten
- 5.3 Implementierung von Zugriffskontrolllisten (ACLs)

6. Integration von SSH

- 6.1 Konfiguration des Secure Shell (SSH) für Remote-Zugriff

7. Systemhärtung und Sicherheit

- 7.1 Einrichtung von BorgBackup für systematische Backups
- 7.2 Installation und Konfiguration von ClamAV und ClamAV-Daemon
- 7.3 Implementierung von Fail2ban zum Schutz vor Brute-Force-Angriffen
- 7.4 Einsatz von RKHunter zur Rootkit-Erkennung
- 7.5 Konfiguration von Snort als Intrusion Detection System (IDS)
- 7.6 Firewall-Konfiguration mit UFW (Uncomplicated Firewall)
- 7.7 Implementierung von AppArmor zur Anwendungssicherheit
- 7.8 Konfiguration von Suricata als leistungsstarkes IDS/IPS
- 7.9 MFA (Multi-Faktor-Authentifikation) und Password Policy

8. Incident Response Management

- 8.1 Entwicklung eines Incident Response Plans (IRP)
- 8.2 Erstellung von Incident Report-Vorlagen
 - 8.2.1 Standard-Report für Sicherheitsvorfälle
 - 8.2.2 Report für Systemfehler
 - 8.2.3 Report für Datenschutzverletzungen
- 8.3 E-Mail-Vorlagen zur Kommunikation von Cyber-Incidents an Mitarbeiter

9. Bedrohungsmodellierung

- 9.1 Einführung in die STRIDE-Methode
- 9.2 Verwendung von Threat Dragon NG zur Durchführung von Bedrohungsanalysen

10. Test und Validierung

- 10.1 Testansätze und -methoden
- 10.2 Validierung der Implementierungsergebnisse

11. Glossar

- 11.1 Fachbegriffe und Abkürzungen im Kontext

12. Quellenverzeichnis

- 12.1 Relevante Literatur und technische Dokumentationen
- 12.2 Online-Ressourcen und Community-Foren

13. Appendix

- 13.1 Bedrohungsmodell
- 13.2 Benutzerhandbuch für die LDAP-Umgebung
- 13.3 Scripts und Automatisierungshilfen
- 13.4 Incident Response Plan
- 13.5 E-Mail-Vorlagen für die Incident-Kommunikation
- 13.6 Vorlagen für Incident Reports
- 13.7 Vagrantfile
- 13.8 Rechnung

1. Präambel

1.1 Zielsetzung der Dokumentation

Diese technische Dokumentation präsentiert die LDAP-Authentifizierungsstruktur der **Smooth Beans GmbH** im Rahmen unseres Abschlussprojekts. Die **LDAP-Domäne** ist ein integraler Bestandteil des Unternehmensnetzwerks und dient der effektiven Verwaltung und Authentifizierung von Benutzern und Gruppen.

Ziel dieser Dokumentation ist es, eine klare und präzise Darstellung der Implementierung, der Benutzerverwaltung sowie der Zugriffsrechte zu bieten. Hierbei werden die spezifischen Anforderungen und die technischen Aspekte der Infrastruktur berücksichtigt. Die vorliegende Arbeit legt besonderen Wert auf die Implementation einer soliden Sicherheitsstrategie, um den Anforderungen des Informationszeitalters gerecht zu werden, und das Unternehmen **Smooth Beans GmbH** und seine Infrastruktur bestmöglich gegen Bedrohungen aus dem Digitalen Raum zu schützen.

<Disclaimer> Dieses Projekt beschäftigt sich lediglich mit der Installation und Konfiguration des LDAP-Servers der Implementation von Benutzergruppen, Benutzern und der Anwendung von ACL's, sowie der Installation, Implementation und Konfiguration von Software zum Schutze des Systems nicht mit der Implementation von Physikalischen Sicherheitsmaßnahmen, diese werden als gegeben betrachtet und liegen außerhalb des Verantwortungs- und Einflussbereiches der Autoren.

1.2 Systemanforderungen

Systemanforderungen für die virtuelle Maschine in VirtualBox

Um eine optimale Funktionalität der virtuellen Maschine (VM) unter VirtualBox zu gewährleisten, sind die folgenden systemtechnischen Anforderungen zu beachten:

1. Virtuelle Maschine:

- **VirtualBox Version:** Mindestens Version 6.0 oder höher.
- **Gäste-Betriebssystem:** Ein unterstütztes Linux-Betriebssystem (in diesem Projekt Ubuntu-Bionic)
- **RAM:** Mindestens 2 GB RAM (Empfehlung: 4 GB für verbesserte Leistung).
- **CPU:** Mindestens 1 virtuelle CPU (2 oder mehr empfohlen für Multi-Threading-Anwendungen).
- **Festplattenspeicher:** Mindestens 20 GB freier Speicherplatz für die VM.

2. Guest Additions:

- **Installation:** Die Guest Additions müssen installiert werden, um die Integration zwischen Host- und Gastbetriebssystem zu verbessern.
- **Manuelles Mounten:** Das CD-ROM-Laufwerk muss manuell gemountet werden, um die Installationsdateien der Guest Additions verfügbar zu machen.

Verwendete Befehle:

```
sudo mount /dev/cdrom /media/cdrom  
cd /media/cdrom  
sudo sh VboxLinuxAdditions.run
```

3. Gemeinsame Ordner (Shared Folders):

- **Einbindung:** Shared Folders ermöglichen den einfachen Austausch von Dateien zwischen dem Host- und dem Gastbetriebssystem.
 - **Konfiguration:** Der gemeinsame Ordner muss in den VirtualBox-Einstellungen der VM eingerichtet werden.
 - **Zugriffsrechte:** Stellen Sie sicher, dass der Benutzer des Gastbetriebssystems die erforderlichen Berechtigungen für den Zugriff auf den gemeinsamen Ordner hat

Das eigentliche Erstellen des Ubuntu Server Images wurde mit **Vagrant** vorgenommen, welches auch die Netzwerkkonfiguration der Virtuellen Maschine übernahm. Dieses **Vagrantfile** ist dieser Dokumentation im Appendix angefügt. Sollte das Deutsche Tastaturlayout nicht übernommen werden, kann dies einfach über die Eingabe folgender Befehle behoben werden.

```
sudo Localectl set-keymap de  
reboot
```

2.2 Installation von Xubuntu-Desktop

Da die von uns verwendete Maschine standardmäßig nur als Terminal Version vorliegt, welche nur über Eingeschränkte Möglichkeiten insbesondere des nicht vorhanden seines einer scroll verfügt wurde eine Desktopumgebung manuell hinzugefügt, um die Nachvollziehbarkeit von Fehlermeldungen vollständig auswerten und analysieren zu können. Für diesen Zweck wurde Xubuntu gewählt.

Befehl zur Installation von Xubuntu

```
sudo apt install xubuntu-desktop -y
```

2.3 Netzwerkplan

Der in dieser Dokumentation vorgestellte **LDAP-Server** ist wie bereits erwähnt Teil des Firmen Netzwerks der **Smooth Beans GmbH** und fungiert als **Admin-Domäne**. In diesen Abschnitt wird diese Domäne nun einmal dargestellt.

Für diesen Zweck wurde von uns ein Netzwerkdiagramm angefertigt, welches die Struktur verbildlicht <Bild 1> Die In diesem Netzwerkplan aufgeführten Komponenten sind nicht Teil dieses Projektes.

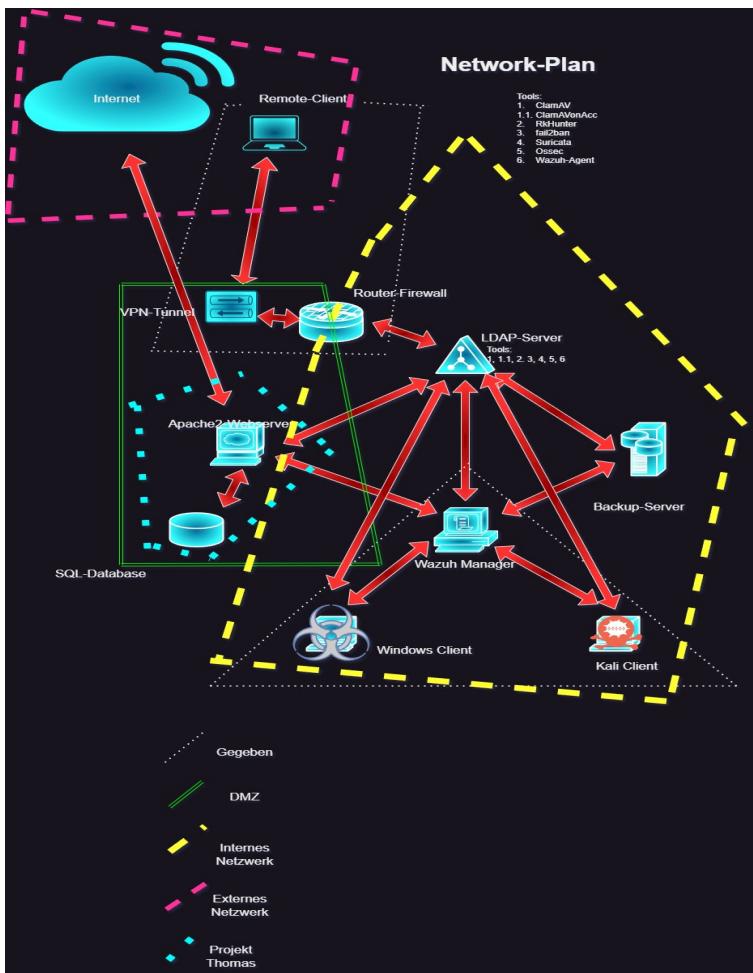


Bild 1. Die Admin Domäne „hamster.panzer“ der Smooth Beans GmbH in der Übersicht.

Für die Zukunft ist des weiteren geplant, den LDAP-Server in die **Wazuh** Überwachung einzubinden und die verbliebenen Komponenten in die Domäne zu integrieren.

3. Einführung in LDAP

3.1 Grundlagen von LDAP (Lightweight Directory Access Protocol)

LDAP (Lightweight Directory Access Protocol) ist ein offenes Protokoll zur Abfrage und Modifikation von Verzeichnisdiensten, das in einem Client-Server-Modell arbeitet. Es ermöglicht den Zugriff auf hierarchisch organisierte Daten, die als Directory

Information Tree (DIT) bezeichnet werden. Jeder Eintrag im DIT wird durch einen Distinguished Name (DN) identifiziert.

Wichtige Operationen im LDAP-Protokoll sind **Bind** (Authentifizierung), **Search** (Abfrage), **Add** (Hinzufügen von Einträgen), **Modify** (Ändern von Einträgen), **Delete** (Löschen von Einträgen) und **Unbind** (Sitzungsbeendigung). Daten werden häufig im **LDAP Data Interchange Format (LDIF)** gespeichert.

LDAP unterstützt verschiedene Authentifizierungsmethoden, einschließlich einfacher Authentifizierung und sicherer Kommunikation über SSL/TLS (LDAPS). Es wird in Unternehmensumgebungen zur zentralen Verwaltung von Benutzeridentitäten, Gruppen und Zugriffsrechten eingesetzt, insbesondere in Systemen wie Microsoft Active Directory und OpenLDAP. Vorteile von LDAP sind die Effizienz bei Suchvorgängen, hohe Skalierbarkeit und breite Unterstützung in verschiedenen Anwendungen.

3.2 Anwendungsfälle und Architektur

LDAP findet breite Anwendung in der Benutzerverwaltung, Zugriffskontrolle und der zentralen Speicherung von Informationen. Zu den typischen Anwendungsfällen gehören:

- **Zentrale Authentifizierung:** LDAP dient als Authentifizierungsdienst für Anwendungen und Systeme, wodurch eine einheitliche Anmeldung (Single Sign-On, SSO) ermöglicht wird.
- **Verzeichnisdienste:** Es wird verwendet, um Informationen über Benutzer, Gruppen und Ressourcen in Netzwerken zu speichern und zu verwalten, z. B. in Unternehmensverzeichnissen.
- **E-Mail-Verzeichnisse:** LDAP wird häufig zur Speicherung von Kontaktdaten in E-Mail-Systemen genutzt, was eine einfache Adresssuche ermöglicht.
- **Netzwerkgeräteverwaltung:** Netzwerkgeräte wie Drucker und Switches können in einem LDAP-Verzeichnis registriert und verwaltet werden.

Die Architektur von LDAP basiert auf einem Client-Server-Modell. Clients senden Anfragen an einen oder mehrere LDAP-Server, die die Daten in einem strukturierten Verzeichnis speichern. Diese Server können in einer hierarchischen Struktur angeordnet sein, um Lasten zu verteilen und Redundanz zu gewährleisten.

3.3 Schlüsselbegriffe und -konzepte

- **Distinguished Name (DN):** Ein eindeutiger Bezeichner für einen Eintrag im Verzeichnis, der die hierarchische Struktur widerspiegelt.
- **Attribute:** Merkmale oder Eigenschaften eines Verzeichniseintrags, wie Name, E-Mail oder Telefonnummer.
- **Object Class:** Definiert den Typ eines Verzeichniseintrags und die Attribute, die dieser Eintrag enthalten kann. Beispielklassen sind person oder organizationalUnit.
- **LDAP-Filter:** Eine Suchabfrage, die bestimmte Kriterien zur Identifizierung von Einträgen im Verzeichnis angibt, z. B. (uid=jdoe) für den Benutzer mit dem Benutzernamen "jdoe".
- **LDAP-Schema:** Die Definition der Objektklassen und Attribute, welche im Verzeichnis verwendet werden, einschließlich deren Typen und Regeln.
- **Replication:** Die Synchronisation von Daten zwischen mehreren LDAP-Servern, um Verfügbarkeit und Ausfallsicherheit zu gewährleisten.

- **Access Control Lists (ACLs):** Regeln, die festlegen, welche Benutzer oder Gruppen auf bestimmte Einträge oder Attribute zugreifen dürfen

4. Installation von LDAP

4.1 Installation der OpenLDAP-Komponenten

Hier ist eine Übersicht der grundlegenden Befehle zur Verwaltung von OpenLDAP. Diese Befehle sind ohne Validierung und dienen als Referenz für die Nutzung in der Installation und Konfiguration von OpenLDAP.

Installation von OpenLDAP

- Debian/Ubuntu:

```
sudo apt-get update -y
sudo apt-get install slapd ldap-utils -y
sudo systemctl enable slapd
sudo systemctl start slapd
```

4.2. Verifizierung der LDAP-Installation

Um nun zu validieren, ob die Installation erfolgreich war und um sicher zu stellen, dass der Server auch korrekt gestartet wurde wird der nachfolgende Befehl verwendet:

```
sudo systemctl status slapd
```

Die Eingabe dieses Befehls sollte die folgende Ausgabe in der Kommandozeile zurück geben. <Bild 2>

```
vagrant@ubuntu-bionic:~$ sudo systemctl status slapd
● slapd.service - LSB: OpenLDAP standalone server (Lightweight Directory Access Protocol)
  Loaded: loaded (/etc/init.d/slapd; generated)
  Drop-In: /lib/systemd/system/slapd.service.d
            └─slapd-remain-after-exit.conf
    Active: active (running) since Mon 2024-09-30 11:09:33 UTC; 59s ago
      Docs: man:systemd-sysv-generator(8)
   Process: 10794 ExecStop=/etc/init.d/slapd stop (code=exited, status=0/SUCCESS)
   Process: 10800 ExecStart=/etc/init.d/slapd start (code=exited, status=0/SUCCESS)
     Tasks: 3 (limit: 2360)
    CGroup: /system.slice/slapd.service
              └─10826 /usr/sbin/slapd -h ldap:/// -g openldap -u openldap -F /etc/ldap/slapd.d

Sep 30 11:09:33 ubuntu-bionic systemd[1]: Starting LSB: OpenLDAP standalone server (Lightweight Directory Access Protocol)
Sep 30 11:09:33 ubuntu-bionic slapd[10800]: * Starting OpenLDAP slapd
Sep 30 11:09:33 ubuntu-bionic slapd[10822]: @(#) $OpenLDAP: slapd (Ubuntu) (May 12 2022 13:52:38) $Debian OpenLDAP Maintainers <pkg-openldap-devel@lists.alioth.debian.org>
Sep 30 11:09:33 ubuntu-bionic slapd[10826]: slapd starting
Sep 30 11:09:33 ubuntu-bionic slapd[10800]: ...done.
Sep 30 11:09:33 ubuntu-bionic systemd[1]: Started LSB: OpenLDAP standalone server (Lightweight Directory Access Protocol).
```

Bild 2. Erwartete Ausgabe. Diese Abbildung bestätigt das der slapd service korrekt gestartet wurde und Aktiv ist

5. Konfiguration von LDAP

5.1 Einrichtung der Domäne hamster.panzer

Nach der Installation und Verifizierung der korrekten Funktion des OpenLDAP-Servers wird dieser nun konfiguriert. Um diesen Vorgang möglichst einfach und anschaulich abzubilden, wurde dieser mit Screenshots festgehalten. **< Bild 3 – Bild 9 >** die jeweils zu wählende option ist diejenige, welche färblich hervorgehoben ist.

Um diesen Prozess nun zu starten und im selben Zuge die Domäne **hamster.panzer** zu erstellen wird in der Kommandozeile des Ubutu-Servers folgender Befehl eingegeben.

```
sudo dpkg-reconfigure slapd
```

Wie berits erwähnt wird dieser Prozess nun verbildlicht in der korrekten Reihenfolge dargestellt.

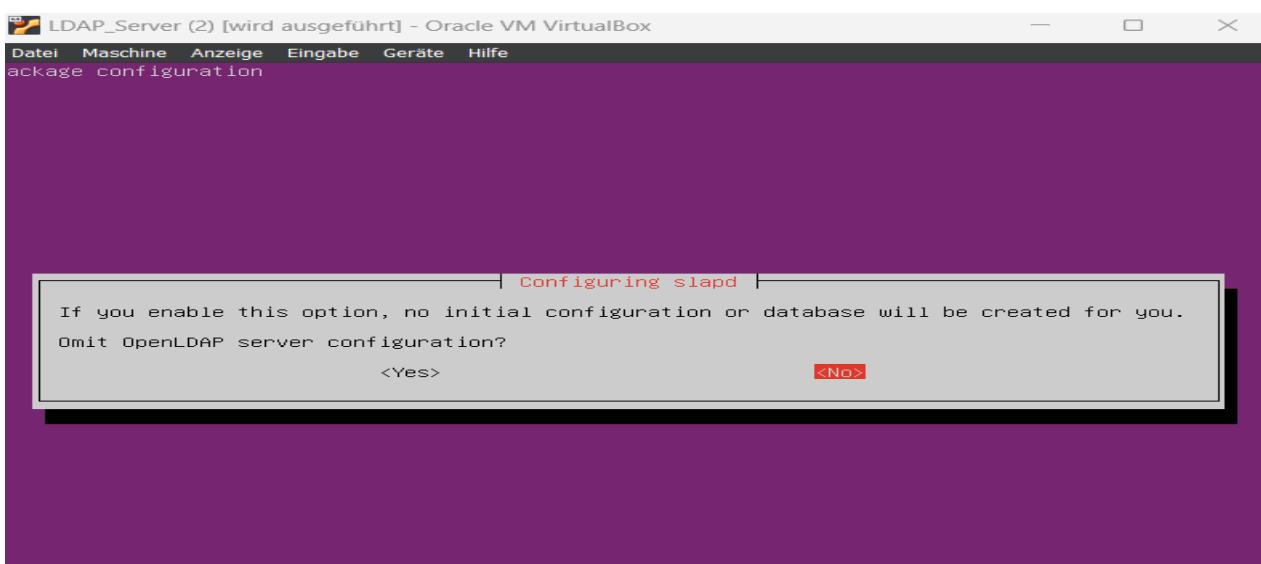


Bild 3. Starten der Konfiguration mit <NO> Bild 4. Vergeben des Domänen Namens

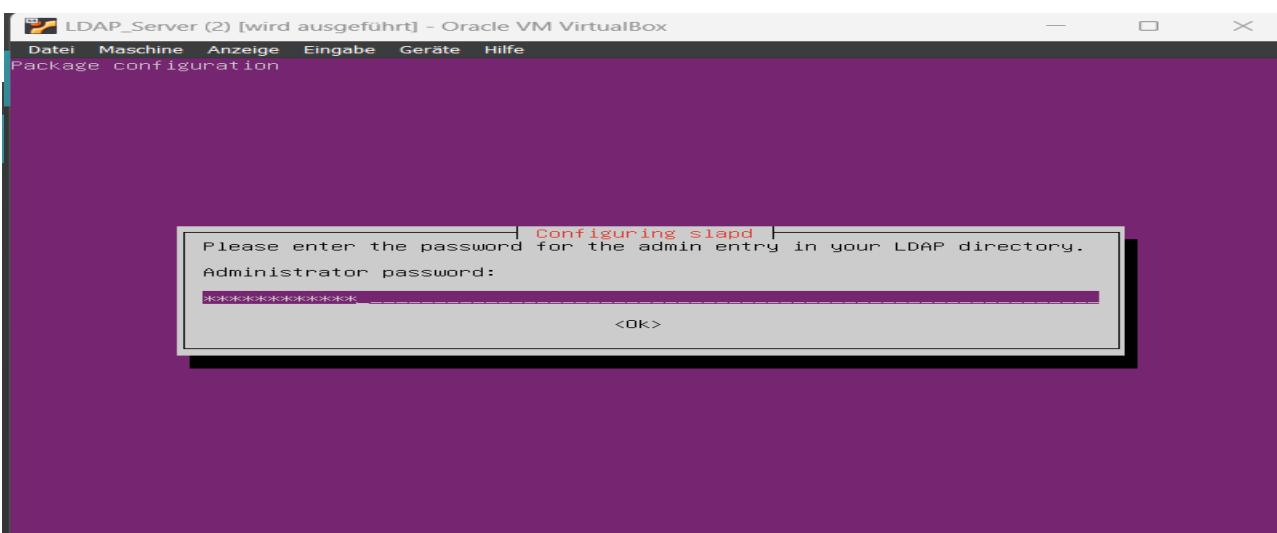


Bild 5 Vergabe des Admin Passwortes dieses muss im nächsten Schritt bestätigt werden

Implementation Ubuntu-LDAP Server, Konfiguration und Härtung

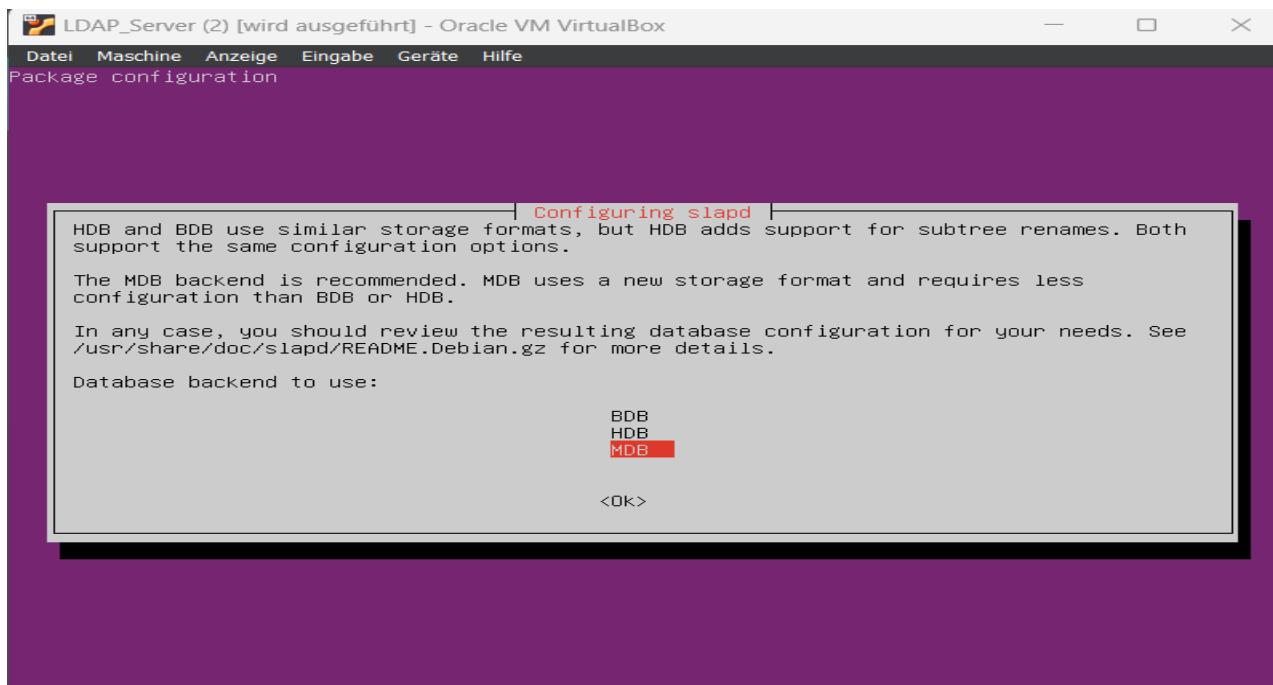


Bild 6. Database Backend festlegen

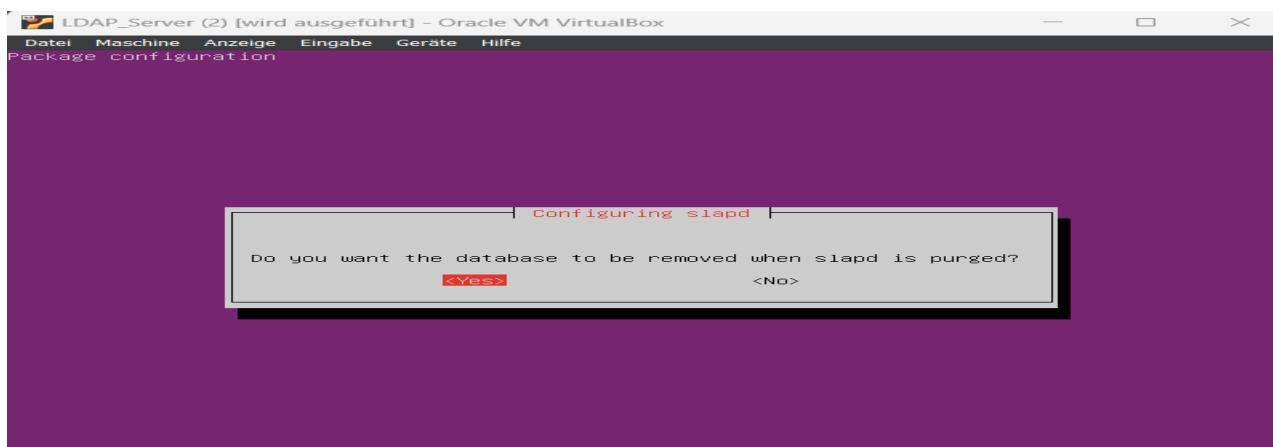


Bild 7. Database bei Purge entfernen

Nun ist die Konfiguration abgeschlossen und muss mit der Eingabe des nachstehenden Befehls überprüft werden.

```
Ldapsearch -x -b "dc=hamster,dc=panzer" -D  
ropating domainDNSZones and forestDNSZones partitions  
invalid permissions on directory '/var/lib/samba/private/msg.sock': has 0755 should be 0700  
See /var/lib/samba/private/named.conf for an example configuration include file for BIND  
and /var/lib/samba/private/named.txt for further documentation required for secure DNS updates  
Setting up sam.ldb rootDSE marking as synchronized  
Fixing provision GUIDs  
A Kerberos configuration suitable for Samba AD has been generated at /var/lib/samba/private krb5.conf  
Once the above files are installed, your Samba AD server will be ready to use  
Server Role: active directory domain controller  
Hostname: ubuntu-bionic  
NetBIOS Domain: HAMSTER ←  
DNS Domain: hamster.panzer  
DOMAIN SID: S-1-5-21-1905030480-1900917822-3011011477
```

Bild 8. Die Eingabe des Befehls ,ldapsearch -x -b "dc=hamster,dc=panzer" -D' gibt diese Ausgabe zurück und bestätigt somit die erfolgreiche Erstellung der Domäne

Um den Domänen Service nun zu erstmalig zu starten und bei einem Neustart des Systems automatisch zu starten werden nun folgende Befehle in der Kommandozeile ausgeführt, wenn dies nicht bereits wie in Kapitel 4. Installation von LDAP geschehen ist.

```
sudo systemctl enable slapd
sudo systemctl start slapd
```

Der Aktuelle Status des `slapd` services wird nun mittels des Befehls:

```
sudo systemctl status slapd
```

vallidiert und sollte die Ausgabe zurückgeben, welche bereits in <Bild 2> des Kapitels 4. Installation von LDAP abgebildet wurde. Wurde dies bereits wie in Kapitel 4.

Installation von LDAP beschrieben ausgeführt, wird der Befehl:

```
sudo systemctl restart slapd
```

verwendet und anschiesend der Status wie bereits beschrieben überprüft.

Ein ausführlicher Leitfaden zum Betrieb und zur Administration des LDAP Server wird im Appendix dieses Dokumentes angehängt. Dieser Leitfaden enthält unter anderem auch eine Sammlung aller wichtigen Befehle so wie ihrer Funktion.

5.2 Erstellung und Verwaltung von Benutzer- und Gruppenkonten

Bevor nun mit der Konfiguration der Struktur des LDAP-Servers und dem Einpflegen von Gruppen, Benutzern sowie der Zugriffsteuerung begonnen wird muss zuerst die Anmeldung von LDAP Benutzern ausdrücklich erlaubt werden, hierfür wird folgender Befehl in der Kommandozeile ausgeführt:

```
ldapmodify -x -D "cn=admin,dc=example,dc=com" -W -f /path/to/password_change.ldif
```

Dieser Befehl öffnet die Oberfläche welche in <Bild 9> dargestellt wird, hier wird der Name der Domäne in folgendem Format eingegeben und anschließend bestätigt

```
dc=hamster,dc=panzer.
```

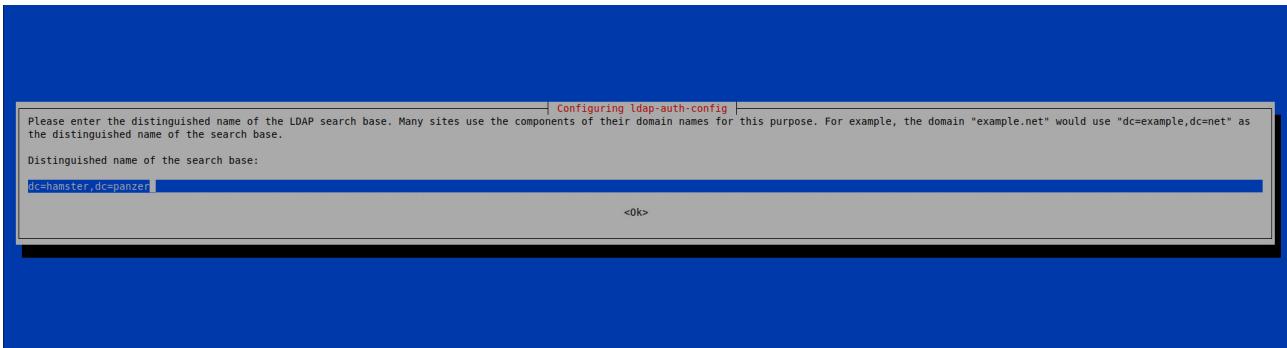


Bild 9. Erlauben des Anmelden von Benutzern auf der Domäne hamster.panzer

Bevor wir nun mit der Erstellung der Gruppen, Benutern und den spezifischen Berechtigungen welche diese erhalten sollen fortfahren, möchten ist es Notwendig die Beabsichtigte Struktur einmal als Diagramm darzustellen. <Bild 10>



Bild 10. Die vorgesehene Struktur der Gruppen wie sie auf dem LDAP-Server integriert werden sollen.

Um nun mit der Erstellung der Gruppen, Benutzern sowie den ACL's beginnen zu können müssen zunächst noch einige **ldif** Dateien erstellt und angewendet werden die Ausgangs Struktur sollte wie in <Bild 11> abgebildet aussehen wenn die nachfolgend erwähnte **base.ldif** korrekt formatiert und angewendet wurde.

```
vagrant@ubuntu-bionic:~$ ldapsearch -x -b "dc=hamster,dc=panzer" -D "cn=admin,dc=hamster,dc=panzer" -W | lolcat
Enter LDAP Password:
# extended LDIF
#
# LDAPv3
# base <dc=hamster,dc=panzer> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#
# hamster.panzer
dn: dc=hamster,dc=panzer
objectClass: top
objectClass: dcObject
objectClass: organization
o: SmoothBeans
dc: hamster

# admin, hamster.panzer
dn: cn=admin,dc=hamster,dc=panzer
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
userPassword:: e1NTSEF9TTNNeG1QVm8za0xKYTRER1Fr0UcycDZwZmQzdTFaeDc=

# search result
search: 2
result: 0 Success

# numResponses: 3
# numEntries: 2
vagrant@ubuntu-bionic:~$ 
```

Bild 11. Ausgangs Struktur des LDAP-Servers mit der Domäne hamster.panzer

Folgende Schritte müssen nun nach einander ausgeführt werden:

- Erstellen und Anwenden der `base.ldif` – Diese Enthält die Organizational Units und User in welchen die Gruppen und Benutzer nachfolgend angelegt werden.
- Anwenden der `base.ldif > ldapadd -x -D "cn=admin,dc=example,dc=com" -W -f base.ldif`
- Validierung der Korrekten Anwendung der `base.ldif > ldapsearch -x -b „dc=hamster,dc=panzer“ -D „cn=admin,dc=hamster,dc=panzer“ -W` Die Erwartete Ausgabe ist in **< Bild 10 >** dargestellt.
- Erstellung und Anwendung der `ou.ldif` dieser Schritt kann nötig werden, wenn bei der Anwendung der `base.ldif` Fehler auftreten und diese OU's nicht korrekt implementiert wurden. Zur Anwendung folgenden Befehl verwenden `ldapadd -x -D "cn=admin,dc=example,dc=com" -W -f ou.ldif`
- Erstellung der Gruppen/Abteilungen in der `groups.ldif` und Anwendung dieser Datei mit `ldapadd -x -D "cn=admin,dc=example,dc=com" -W -f groups.ldif`
- Verifizierung der korrekten Anwendung mit `ldapsearch -x -b "dc=hamster,dc=panzer" "(objectClass=posixGroup)"`

Nun sollte wie in **< Bild 12 >** die Ausgabe folgendermaßen aussehen:

```
root@ubuntu-bionic:~# ldapsearch -x -b "ou=groups,dc=hamster,dc=panzer" -D "cn=admin,dc=hamster,dc=panzer" -W | less
Enter LDAP Password:
# extended LDIF
#
# LDAPv3
# base <ou=groups,dc=hamster,dc=panzer> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#
# groups, hamster.panzer
dn: ou=groups,dc=hamster,dc=panzer
objectClass: organizationalUnit
ou: groups

# hr, groups, hamster.panzer
dn: cn=hr,ou=groups,dc=hamster,dc=panzer
objectClass: posixGroup
cn: hr
gidNumber: 10008

# wazuh, groups, hamster.panzer
dn: cn=wazuh,ou=groups,dc=hamster,dc=panzer
objectClass: posixGroup
cn: wazuh
gidNumber: 10009

# it_leitung, groups, hamster.panzer
dn: cn=it_leitung,ou=groups,dc=hamster,dc=panzer
objectClass: posixGroup
cn: it_leitung
gidNumber: 10002

# buchhaltung, groups, hamster.panzer
dn: cn=buchhaltung,ou=groups,dc=hamster,dc=panzer
objectClass: posixGroup
cn: buchhaltung
gidNumber: 10007

# logistik_leitung, groups, hamster.panzer
dn: cn=logistik_leitung,ou=groups,dc=hamster,dc=panzer
objectClass: posixGroup
cn: logistik_leitung
... --
```

Bild 12. Die LDAP-Domänen Struktur nach Anwendung der voranstehenden ldif Dateien.

- Die Erstellung und die Implementierung der Benutzer wurde mittels eines von uns verfassten **Python** Skripts realisiert, dieses ist wie sämtliche **ldif**, **bash** und anderen **Python** Skripte ebenfalls im Appendix anhängig.

<Disclaimer> Alle in diesem Dokument genannten Benutzernamen wurden zufällig von einer KI generiert. Jegliche Ähnlichkeiten mit realen, lebenden oder verstorbenen Personen sind rein zufällig und nicht von den Autoren beabsichtigt!

- Ausführen des **Python** Skripts mittels Eingabe von
`sudo python3 create_users.py` Generierung von Benutzer Accounts auf dem LDAP-Server. Dieses Skript erstellt nicht nur die Benutzer sondern ordnet sie auch direkt den vorgesehenen Gruppen zu was die Ersteinrichtung deutlich vereinfachte.
- Überprüfung der korrekten Implementation der Benutzer durch Eingabe des Befehls
`Ldapsearch -x -b "dc=hamster,dc=panzer" "(objectClass=posixAccount)"`

Nachfolgend in **<Bild 13>** wird die Erwartete Ausgabe verdeutlicht. Die abgebildete Ausgabe zeigt nicht nur die vorhandenen Benutzer Konten sondern auch die Gruppen in welchen sie wie vorgesehen zugeordnet wurden.

```
# thomas.falke, users, hamster.panzer
dn: uid=thomas.falke,ou=users,dc=hamster,dc=panzer
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
cn: Thomas Falke
sn: Falke
uid: thomas.falke
userPassword:: UGFzc3dvcmQxMjMh
gidNumber: 10013
homeDirectory: /home/users/thomas.falke
uidNumber: 20004

# david.thornton, users, hamster.panzer
dn: uid=david.thornton,ou=users,dc=hamster,dc=panzer
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
cn: David Thornton
sn: Thornton
uid: david.thornton
userPassword:: UGFzc3dvcmQxMjMh
gidNumber: 10014
homeDirectory: /home/users/david.thornton
uidNumber: 20005

# michael.stone, users, hamster.panzer
dn: uid=michael.stone,ou=users,dc=hamster,dc=panzer
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
cn: Michael Stone
sn: Stone
uid: michael.stone
userPassword:: UGFzc3dvcmQxMjMh
gidNumber: 10015
homeDirectory: /home/users/michael.stone
uidNumber: 20006

# search result
search: 2
result: 0 Success

# numResponses: 22
# numEntries: 21
root@ubuntu-bionic:/home/vagrant#
```

Bild 13. Ein Abriss der neuen LDAP-Server Struktur nach erfolgreichem einpflegen der Gruppen und Benutzer

5.3 Implementierung von Zugriffskontrolllisten (ACLs)

Wie schon bei der Erstellung der Gruppen setzten wir nun auch bei der Vergabe der Zugriffskontrollisten auf Automatisierung. In <Tabelle 1> wird zuvor allerdings noch eine Übersicht über die Gruppen, Benutzer sowie ihre Berechtigungen vorgestellt. Die Vergabe der Zugriffskontrollisten erfolgte nach dem „Least Privileges“ Prinzip, dies soll sicherstellen, dass alle Benutzer nur die Berechtigungen haben welche sie zur Erfüllung der ihnen übertragenen Aufgaben benötigen.

Gruppe	gidNumber	Benutzer	Rechte
Geschaeftsfuehrung	10001	Max Mustermann	Volle Zugriffsrechte auf eigene Ordner, Zugriff auf Ordner der Verwaltung, HR, Buchhaltung
IT-Leitung	10002	Klaus Müller, Sabine Schmidt	Volle Administratorrechte auf alle Systeme, Zugriff auf alle Ordner und Ressourcen, Sudo-Rechte
Logistik-Leitung	10003	Anna Becker	Lesen und Schreiben in eigenen Ordnern, Zugriff auf relevante Logistik-Daten
Marketing-Leitung	10004	Peter Klein	Lesen und Schreiben in eigenen Ordnern, Zugriff auf Marketing-Daten
Produktion-Leitung	10005	Laura Wagner	Lesen und Schreiben in eigenen Ordnern, Zugriff auf Produktions-Daten
Verwaltung-Leitung	10006	Hans Meier	Lesen und Schreiben in eigenen Ordnern, Zugriff auf Verwaltungsdaten
Buchhaltung	10007	Julia Braun	Lesen und Schreiben in eigenen Ordnern, Zugriff auf Buchhaltungsdaten
HR	10008	Lisa Schwarz	Lesen und Schreiben in eigenen Ordnern, Zugriff auf HR-Daten
Wazuh	10009	Oliver König	Lesen und Schreiben in eigenen Ordnern, Zugriff auf Sicherheitsdaten
LDAP-Administrator	10010	Admin User	Volle Administratorrechte auf alle Systeme und Ordner, Kontrolle über LDAP-Verwaltung
Web-Administrator	10011	Tom Schneider	Sudo-Rechte für Docker und Webserver, Zugriff auf Webserver-Ordner, keine root-shell
DAU	10012	Martin Kellerer	Nur An- und Abmelden, keine weiteren Berechtigungen

Tabelle 1. Gruppen und Benutzer Übersicht sowie die vorgesehenen Zugriffsberechtigungen (ACL)

Wie eingangs dieses Kapitels bereits erwähnt wurde für die Anwendung dieser ACL's ein Bash Skript verwendet der hierfür anzuwendende Befehl lautet wie folgt:

`sudo bash permissions.sh`

Die Validierung wird direkt bei Ausführung des Skripts von diesem im Terminal als Print Statement ausgegeben. <Bild 14>

```
LDAP-Berechtigungen erfolgreich angewendet.
root@ubuntu-bionic:/home/vagrant/LDAP_CONFIG/bash#
```

Bild 14 Die Berechtigungen wurden erfolgreich angewendet

Nun können sich alle Angelegten Benutzer wie vorgesehen über den **LDAP-Service** Anmelden und Autorisieren im Nachfolgenden Kapitel werden nun die Implementation eines **SMB** Dienstes (Server Message Block) sowie die Einrichtung eines **SSH** (Secure Shell Host) Dienstes behandelt, über welches die Domänen Mitglieder auf die ihnen zugeschriebenen Ressourcen welche von dem **LDAP-Service** verwaltet werden zugreifen können.

6. Integration von SSH

6.1 Konfiguration des Secure Shell (SSH) für Remote-Zugriff

Damit die Benutzer sich nun wie vorgesehen über den LDAP Verzeichnisdienst Anmelden und Authentifizieren können wird der **OpenSSH** Dienst installiert und konfiguriert.

Installation und Konfiguration von OpenSSH

- Ausführen des folgenden Befehls, um den OpenSSH-Server zu installieren:
`sudo apt-get update && sudo apt-get install -y openssh-server`
- Überprüfung des **SSH-Dienstes** Kontrolle, ob der SSH-Dienst läuft:
`sudo systemctl status ssh`
- Starten des SSH Dienstes sollte er nicht laufen.
`sudo systemctl start ssh`
- Konfiguration von SSH
`sudo vim /etc/ssh/sshd_config`
 - Root-Login deaktivieren:
`PermitRootLogin no`
 - Maximale Authentifizierungsversuche reduzieren:
`MaxAuthTries 3`
 - SSH-Protokoll-Version 2 erzwingen:
`Protocol 2`
- Neustart des **SSH-Dienstes** Nach der Konfiguration muss der SSH-Dienst neu gestartet werden:
`sudo systemctl restart ssh`
- Firewall konfigurieren (UFW)
`sudo ufw allow 22/tcp`
`sudo ufw enable`

Nun sollten die **LDAP-Domänenmitglieder** im stande sein sich remote mittels **SSH** auf dem LDAP Server einzuloggen. **< Bild 15 >**

```
[parrot@parrot]~
ssh martin.kellerer@192.168.56.100
martin.kellerer@192.168.56.100's password:
martin.kellerer@ubuntu-bionic:~$ ^C
martin.kellerer@ubuntu-bionic:~$ exit
logout
```

Bild 15. Der User Martin.Kellerer hat sich erfolgreich von einer anderen Maschine (ParrotOS) per SSH Authentifiziert und auf dem LDAP-Server eingeloggt

7. Systemhärtung und Sicherheit

7.1 Einrichtung von BorgBackup für systematische Backups

BorgBackup (kurz **Borg**) ist ein fortschrittliches, sicheres und effizientes Backup-Tool, das speziell für die Sicherung und Wiederherstellung von Daten entwickelt wurde. Es nutzt ein dedupliziertes Speicherformat, wodurch identische Daten nur einmal gespeichert werden. Dies reduziert den benötigten Speicherplatz erheblich.

Borg unterstützt sowohl lokale als auch entfernte Backups und bietet eine starke Verschlüsselung für Datensicherheit. Die integrierte Komprimierung minimiert den Speicherbedarf weiter und beschleunigt die Übertragung von Daten. Mit Borg kannst du Backups inkrementell durchführen, sodass nur Änderungen seit dem letzten Backup gespeichert werden, was die Backup-Zeiten verkürzt.

Das Tool verfügt über eine einfache und benutzerfreundliche Kommandozeilenoberfläche, die eine flexible Konfiguration ermöglicht. Außerdem können Backups automatisiert und in Skripten integriert werden, was die Verwaltung und Wartung von Backup-Jobs erleichtert. BorgBackup ist besonders beliebt in der Linux-Community und wird häufig für die Sicherung von Servern und wichtigen Daten verwendet.

Für die Installation wird der folgende Befehl wie gewohnt in der Kommandozeile eingegeben:

sudo apt install borgbackup -y

Bild 6. Database Backend festlegen
Nun kann **Borg** einfach über das Terminal gesteuert werden. Da der **Backup-Server** zur Zeit der erstellung dieser Dokumentation noch nicht in der Domäne eingebunden war, wurde auf die Konfiguration von Automatischen Backups derzeit verzichtet. Der vorgesehne Backup Plan wurde hingegen bereits definiert und wie folgt terminiert:

- Dienstags: 22:00 Uhr
- Freitags: 22:00 Uhr

```
process (default: 'ssh')
required arguments:
  <command>
  mount          mount repository
  serve          start repository server process
  init           initialize empty repository
  check          verify repository
  key            manage repository key
  change-passphrase  change repository passphrase
  create          create backup
  extract         extract archive contents
  export-tar      create tarball from archive
  diff            find differences in archive contents
  rename          rename archive
  delete          delete archive
  list             list archive or repository contents
  umount          umount repository
  info             show repository or archive information
  break-lock      break repository and cache locks
  prune           prune archives
  loads           load repository format
  recreate        re-create archives
  with-lock       run user command with lock held
  config          get and set configuration values
  debug           debugging command (not intended for normal use)
  benchmark       benchmark command
```

Bild 16> Terminal Benutzer Oberfläche con Borg

7.2 Installation und Konfiguration von ClamAV und ClamAV-Daemon

Mit **ClamAV** wurde sich für eine etablierte und renommierter Open-Source Anti Virus Lösung entschieden welche sich durch eine sehr hohe Zuverlässigkeit und nahezu tägliche Updates auszeichnet. Die Installation erfolgt auch hier wieder über die Kommandozeile, um den ClamAV On Demand Scanner durch eine Echtzeit komponente (**ClamAV-Daemon**) zu ergänzen wird diese im selben Zuge mit installiert und im Anschluss konfiguriert. Hier nun der Befehl für diese Operation:

```
sudo apt install clamav clamav-daemon
sudo freshclam (zur Aktualisierung der Signaturen der Virus Datenbank)
sudo systemctl enable clamav-daemon
sudo apt start clamav-daemon
sudo apt status clamav-daemon
```

```
vagrant@ubuntu-bionic:~$ sudo systemctl status clamav-daemon | lolcat
● clamav-daemon.service - Clam AntiVirus userspace daemon
  Loaded: loaded (/lib/systemd/system/clamav-daemon.service; enabled; vendor preset: enabled)
  Drop-In: /etc/systemd/system/clamav-daemon.service.d
            └─extend.conf
    Active: active (running) since Tue 2024-10-01 06:45:56 UTC; 2min 22s ago
      Docs: man:clamd(8)
             man:clamd.conf(5)
             https://docs.clamav.net/
   Process: 1008 ExecStartPre=/bin/chown clamav /run/clamav (code=exited, status=0/SUCCESS)
   Process: 827 ExecStartPre=/bin/mkdir -p /run/clamav (code=exited, status=0/SUCCESS)
 Main PID: 1053 (clamd)
    Tasks: 2 (limit: 2355)
   CGroup: /system.slice/clamav-daemon.service
           └─1053 /usr/sbin/clamd --foreground=true

Oct 01 06:46:17 ubuntu-bionic clamd[1053]: Tue Oct  1 06:46:17 2024 -> Portable Executable support enabled.
Oct 01 06:46:17 ubuntu-bionic clamd[1053]: Tue Oct  1 06:46:17 2024 -> ELF support enabled.
Oct 01 06:46:17 ubuntu-bionic clamd[1053]: Tue Oct  1 06:46:17 2024 -> Mail files support enabled.
Oct 01 06:46:17 ubuntu-bionic clamd[1053]: Tue Oct  1 06:46:17 2024 -> OLE2 support enabled.
Oct 01 06:46:17 ubuntu-bionic clamd[1053]: Tue Oct  1 06:46:17 2024 -> PDF support enabled.
Oct 01 06:46:17 ubuntu-bionic clamd[1053]: Tue Oct  1 06:46:17 2024 -> SWF support enabled.
Oct 01 06:46:17 ubuntu-bionic clamd[1053]: Tue Oct  1 06:46:17 2024 -> HTML support enabled.
Oct 01 06:46:17 ubuntu-bionic clamd[1053]: Tue Oct  1 06:46:17 2024 -> XMLDOCS support enabled.
Oct 01 06:46:17 ubuntu-bionic clamd[1053]: Tue Oct  1 06:46:17 2024 -> HWP3 support enabled.
Oct 01 06:46:17 ubuntu-bionic clamd[1053]: Tue Oct  1 06:46:17 2024 -> Self checking every 3600 seconds.
vagrant@ubuntu-bionic:~$
```

Die Eingabe des Status Befehls sollte nun die folgende Ausgabe zurückgeben. **Bild 17**

Bild 17. ClamAV-daemon ist Aktiv und der Echtzeitschutz des Systems und des LDAP-Servers dadurch sichergestellt, auch die Automatisierte Aktualisierung der Signatur Datenbank wird hierdurch gewährleistet.

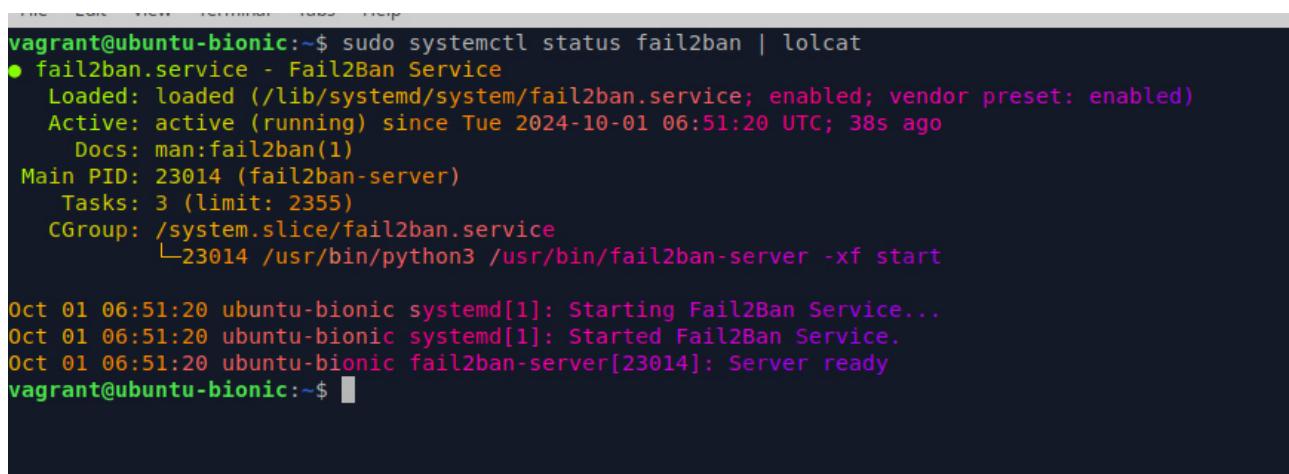
Durch den Einsatz von ClamAV als „on Demand“ Scanner und ClamAV-Daemon als „Realtime“ Monitoring Scanner welcher jedes Prozess welcher auf dem System gestartet wird bei dessen Ausführung überwacht und der permanenten Aktualisierung der Signatur Datenbank sowie der fortlaufenden Weiterentwicklung und Verbesserung dieses Produktes wird das System nun gut geschützt. Um jedoch ein Leistungsstarkes und fortschrittliches Sicherheitskonzept zu haben werden nun noch weitere Komponenten installiert und konfiguriert, um das System unter anderem auch gegen „Brute-Force“ und Root-Kits abzusichern.

7.3 Implementierung von Fail2ban zum Schutz vor Brute-Force-Angriffen

Auch **Fail2ban** ist ein Open-Source-Sicherheitstool, das dazu dient, **Brute-Force-Angriffe** auf Server zu verhindern. Es überwacht Logdateien auf verdächtige Aktivitäten, wie wiederholte fehlgeschlagene Anmeldeversuche, und **sperrt** automatisch IP-Adressen, die als Bedrohung identifiziert werden. Fail2ban ist konfigurierbar und kann für verschiedene Dienste wie

SSH, HTTP und FTP eingesetzt werden, um die Sicherheit von Systemen zu erhöhen. Durch die Reduzierung unerwünschter Zugriffsversuche verbessert es den Schutz vor unbefugtem Zugriff erheblich. Die Befehle für die Kommandozeile zur Installation und Konfiguration lauten wie folgt:

```
sudo apt install fail2ban -y  
sudo systemctl enable fail2ban  
sudo systemctl start fail2ban  
sudo systemctl status fail2ban
```



```
vagrant@ubuntu-bionic:~$ sudo systemctl status fail2ban | lolcat  
● fail2ban.service - Fail2Ban Service  
   Loaded: loaded (/lib/systemd/system/fail2ban.service; enabled; vendor preset: enabled)  
   Active: active (running) since Tue 2024-10-01 06:51:20 UTC; 38s ago  
     Docs: man:fail2ban(1)  
 Main PID: 23014 (fail2ban-server)  
    Tasks: 3 (limit: 2355)  
   CGroup: /system.slice/fail2ban.service  
         └─23014 /usr/bin/python3 /usr/bin/fail2ban-server -xf start  
  
Oct 01 06:51:20 ubuntu-bionic systemd[1]: Starting Fail2Ban Service...  
Oct 01 06:51:20 ubuntu-bionic systemd[1]: Started Fail2Ban Service.  
Oct 01 06:51:20 ubuntu-bionic fail2ban-server[23014]: Server ready  
vagrant@ubuntu-bionic:~$ █
```

Bild 18. Erwartete Ausgabe von `sudo systemctl status fail2ban`

7.4 Einsatz von RkHunter zur Rootkit-Erkennung

RKHunter (Rootkit Hunter) ist ein wichtiges Sicherheitswerkzeug für Unix-basierte Systeme, das sich auf die Aufspürung von **Rootkits** und **Malware** spezialisiert hat. Durch die Durchführung von Systemanalysen entdeckt es verdächtige Dateien und potenzielle Sicherheitsanfälligkeiten, die auf kompromittierte Systeme hinweisen könnten. RKHunter trägt dazu bei, die Sicherheit des Systems zu erhöhen, indem es Administratoren auf mögliche Bedrohungen aufmerksam macht und eine proaktive Überwachung ermöglicht.

Um RhHunter auf einem System zu installieren wird erneut die Benutzung der Kommandozeile erforderlich:

```
sudo apt install rkhunter -y
```

Zur Überprüfung des Systems auf Rootkits oder Maleware mithilfe von RkHunter wird in der Kommandozeile folgender Befehl ausgeführt:

```
sudo rkhunter -check
```

Es wird empfohlen RkHunter möglichst frühzeitig nach Installation des eigentlichen Betriebssystems zu installieren und eine Überprüfung durchzuführen um die **Falsch Positiven** Ergebnisse zu Dokumentieren und für spätere Vergleiche griffbereit zu halten.

7.5 Konfiguration von Snort als Intrusion Detection System (IDS)

Wie auch alle anderen von uns Implementierten Software-Lösungen ist auch **Snort** ein Open-Source Produkt. **Snort** ist ein Open-Source Intrusion Detection System (IDS) und

Intrusion Prevention System (IPS), das Netzwerkverkehr in Echtzeit analysiert und verdächtige Aktivitäten identifiziert. Es arbeitet paketbasiert und nutzt eine Regelbasierte Sprache, um Angriffe zu erkennen, die von bekannten Bedrohungen bis hin zu anomalem Verhalten reichen. Snort ist hochgradig konfigurierbar und kann sowohl zur Überwachung von Netzwerken als auch zur Analyse von Protokollen verwendet werden, was es zu einem unverzichtbaren Werkzeug für die Netzwerksicherheit macht.

Um Snort nun auf unserem Server zu installieren geben wir uns erneut in unseren „Happy Place“ der Linux Kommandozeile und geben dort nun folgenden Befehl ein:

```
sudo apt install snort -y
```

Im Laufe des Installation Prozesses wird der Benutzer zur Konfiguration aufgefordert, der wichtigste Schritt ist hier die Eingabe des korrekten Netzwerkinterfaces welches über den Befehl `ip a` sehr leicht zu bestimmen ist.

Abschließend zur Installation muss auch hier der Service Aktiviert, gestartet sowie sein Status überprüft werden:

```
sudo systemctl enable snort
sudo systemctl start snort
sudo systemctl status snort
```

```
vagrant@ubuntu-bionic:~$ sudo systemctl status snort | lolcat
● snort.service - LSB: Lightweight network intrusion detection system
  Loaded: loaded (/etc/init.d/snort; generated)
  Started: started
    Tasks: 2 (limit: 2355)
   CGroup: /system.slice/snort.service
           └─23794 /usr/sbin/snort -m 027 -D -d -l /var/log/snort -u snort -c /etc/snort/snort.conf -S HOME_NET=[192.168.0.0/16] -i enp0s8

Oct 01 06:53:24 ubuntu-bionic snort[23794]:      Preprocessor Object: SF_SSLPP Version 1.1 <Build 4>
Oct 01 06:53:24 ubuntu-bionic snort[23794]:      Preprocessor Object: SF_DCERPC2 Version 1.0 <Build 3>
Oct 01 06:53:24 ubuntu-bionic snort[23794]:      Preprocessor Object: SF_MODBUS Version 1.1 <Build 1>
Oct 01 06:53:24 ubuntu-bionic snort[23794]:      Preprocessor Object: SF_DNP3 Version 1.1 <Build 1>
Oct 01 06:53:24 ubuntu-bionic snort[23794]:      Preprocessor Object: SF_GTP Version 1.1 <Build 1>
Oct 01 06:53:24 ubuntu-bionic snort[23794]:      Preprocessor Object: SF_SMTP Version 1.1 <Build 9>
Oct 01 06:53:24 ubuntu-bionic snort[23794]:      Preprocessor Object: SF_SSH Version 1.1 <Build 3>
Oct 01 06:53:24 ubuntu-bionic snort[23794]:      Preprocessor Object: SF_SDF Version 1.1 <Build 1>
Oct 01 06:53:24 ubuntu-bionic snort[23794]:      Preprocessor Object: SF_IMAP Version 1.0 <Build 1>
Oct 01 06:53:24 ubuntu-bionic snort[23794]: Commencing packet processing (pid=23794)
vagrant@ubuntu-bionic:~$
```

Bild 19 Erwartete Ausgabe des Befehls `sudo systemctl status snort`

7.6 Firewall-Konfiguration mit UFW (Uncomplicated Firewall)

UFW (Uncomplicated Firewall) ist eine benutzerfreundliche Schnittstelle zur Verwaltung von iptables, die speziell für die einfache Konfiguration von Firewalls auf Linux-Systemen entwickelt wurde. UFW ist standardmäßig auf Ubuntu Bionic vorinstalliert, was die Inbetriebnahme erleichtert. Mit klaren und einfachen Befehlen ermöglicht UFW Administratoren, spezifische Regeln für eingehenden und ausgehenden Netzwerkverkehr festzulegen und so die Sicherheit des Systems zu erhöhen.

Für die Konfiguration der UFW wurde auch hier ein Skript von uns angefertigt, welches diese Aufgabe automatisiert. Das Skript wird mit dem Befehl

`sudo bash ufw_config.sh`

ausgeführt und gibt die Konfiguration als Text in der Kommandozeile zurück <Bild 19>

To	Action	From
-	-----	---
889/tcp	ALLOW IN	Anywhere
536/tcp	ALLOW IN	Anywhere
3310/tcp	ALLOW IN	Anywhere
22/tcp	ALLOW IN	Anywhere
2375/tcp	ALLOW IN	Anywhere
22/tcp (OpenSSH)	ALLOW IN	Anywhere
139	ALLOW IN	192.168.56.0/24
445	ALLOW IN	192.168.56.0/24
30/tcp	ALLOW IN	Anywhere
443/tcp	ALLOW IN	Anywhere
55000/tcp	ALLOW IN	Anywhere
889/tcp (v6)	ALLOW IN	Anywhere (v6)
536/tcp (v6)	ALLOW IN	Anywhere (v6)
3310/tcp (v6)	ALLOW IN	Anywhere (v6)
22/tcp (v6)	ALLOW IN	Anywhere (v6)
2375/tcp (v6)	ALLOW IN	Anywhere (v6)
22/tcp (OpenSSH (v6))	ALLOW IN	Anywhere (v6)
30/tcp (v6)	ALLOW IN	Anywhere (v6)
443/tcp (v6)	ALLOW IN	Anywhere (v6)
55000/tcp (v6)	ALLOW IN	Anywhere (v6)

Bild 20. Die Konfiguration der UFW auf für den LDAP-Servern

7.7 Implementierung von AppArmor zur Anwendungssicherheit

AppArmor ist ein Sicherheitsmodul für Linux, das auf der Mandatory Access Control (MAC) basiert und die Ausführung von Anwendungen durch Profile einschränkt. Es bietet eine zusätzliche Schutzschicht, indem es definiert, welche Ressourcen (wie Dateien, Netzwerkverbindungen und Systemaufrufe) eine Anwendung nutzen darf, wodurch potenzielle Schäden durch Sicherheitsverletzungen oder Malware minimiert werden. AppArmor ist standardmäßig auf Ubuntu vorinstalliert, was eine einfache Inbetriebnahme und Integration in bestehende Sicherheitsmaßnahmen ermöglicht. Durch die Erstellung und Anpassung von Profilen können Administratoren den Zugriff von Anwendungen präzise steuern und die Sicherheitsrichtlinien des Systems effektiv umsetzen.

AppArmor wurde von und mittels eines Python Skriptes entsprechend den Anforderungen unseres LDAP-Servers konfiguriert. Der Befehl lautet ***sudo python3 aa_config.py***

7.8 Konfiguration von Suricata als leistungsstarkes IDS/IPS

Den Abschluss der von unsrer Seite installierten und konfigurierten Sicherheitssystemen bildet mit Suricata ein weiteres bekanntes und weitverbreitetes Produkt. **Suricata** ist ein leistungsstarkes Open-Source-Netzwerk- und Intrusion-Detection-System (IDS), das als Sicherheitslösung für die Überwachung und Analyse des Netzwerkverkehrs eingesetzt wird. Es ermöglicht die Erkennung von Bedrohungen, Angreifern und bösartigen Aktivitäten in Echtzeit. Suricata analysiert den Netzwerkverkehr in Echtzeit und bietet Funktionen wie Protokollierung, Netzwerkanalyse, Verkehrsinspektion und Bedrohungserkennung. Dank seiner hohen Flexibilität unterstützt es sowohl die Signaturerkennung als auch die Verhaltensanalyse und kann in Kombination mit anderen Sicherheitslösungen eingesetzt werden, um eine umfassende Sicherheitsstrategie zu gewährleisten. Zur Installation dürfen wir erneut unsre heißgeliebte Linux Kommandozeile aufsuchen und voller Begeisterung und mit höchstem Elan folgenden Befehl dort eingeben:

```
sudo apt install suricata
```

Da es uns auch eine enorme Freude bereitet Prozesse zu automatisieren und wir uns nicht nur in der Linux Kommandozeile wie zuhause fühlen sondern ebenso gerne Zeit in unserem Zweit Wohnsitz dem Vim Editor verbringen, entschlossen wir uns ein weiteres Skript zur Konfiguration von suricata zu schreiben. Um dieses Bash Skript auszuführen wird der folgende Befehl im Terminal eingegeben:

```
sudo bash suricata_config.sh
```

Dieses Skript übernimmt nun nicht nur die von uns spezifizierte Konfiguration, sondern aktiviert und startet den Service auch. Das Skript endet bei erfolgreicher Anwendung mit der folgenden Ausgabe. **< Bild 21 >**

```
2024-10-02 14:11:43 (8.76 MB/s) - 'emerging-all.rules' saved [31921769/31921769]

Adding rule files to the configuration...
Starting Suricata...
Synchronizing state of suricata.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable suricata
Checking the status of Suricata...
● suricata.service - Suricata IDS/IDP daemon
   Loaded: loaded (/lib/systemd/system/suricata.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2024-10-02 14:07:57 UTC; 3min 46s ago
     Docs: man:suricata(8)
           man:suricatasc(8)
           https://redmine.openinfosecfoundation.org/projects/suricata/wiki
   Main PID: 5001 (Suricata-Main)
      Tasks: 8 (limit: 4915)
        CGroup: /system.slice/suricata.service
                 └─5001 /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml --pidfile /var/run/suricata.pid

Oct 02 14:07:57 ubuntu-bionic systemd[1]: Starting Suricata IDS/IDP daemon...
Oct 02 14:07:57 ubuntu-bionic suricata[4981]: 2/10/2024 -- 14:07:57 - <Notice> - This is Suricata version 3.2 RELEASE
Oct 02 14:07:57 ubuntu-bionic suricata[4981]: 2/10/2024 -- 14:07:57 - <Warning> - [ERRCODE: SC_ERR_SYSCALL(50)] - Failure when trying to bind to port 5000 on interface 'eth0': N
```

Bild 21. Erwartete Ausgabe, Suricata ist Aktiv.

Für die Zukunft ist geplant den **LDAP-Server** und die Domäne **hamster.panzer** durch Integration in das **Wazuh-SIEM** Zentral überwachen zu können.

Durch die Implementation und Konfiguration der von uns gewählten Komponenten ist das Unternehmen **Smooth Beans GmbH** bestens aufgestellt und der LDAP-Server kann als sehr sicher betrachtet werden. Diese Maßnahmen werden durch die in den nachfolgenden Kapiteln behandelnde Themen noch abgerundet.

7.9 MFA (Multi-Faktor-Authentifikation) und Password Policy

Die Implementierung von **MFA (Multi-Faktor-Authentifikation)** als zusätzliche Sicherheitsmaßnahme wird als gegeben angesehen, und wird Systemweit wo möglich und verfügbar von allen Angestellten angewendet, eben so wird eine Starke Passwort Policy erzwungen und die Anwendung von Passwort Managern wird strikt umgesetzt.

8. Incident Response Management

8.1 Entwicklung eines Incident Response Plans (IRP)

Präambel

Der hier vorgestellte Incident Response Plan legt jene Verfahren und Maßnahmen fest, welche nötig sind um auf sicherheitsrelevante Vorfälle effektiv reagieren zu können. Er wurde gezielt für den LDAP-Server der Smooth Beans GmbH, dessen Installation und Konfiguration Teil dieses Projektes waren, entwickelt.

Der LDAP-Server ist die Admin-Domäne mit dem Namen „hamster.panzer“ und im internen Netzwerk unter der IP-Adresse: 192.168.56.100 auf einem Ubuntu-Biotic Server zu finden. Dieser Server stellt Authentifizierungs- und Verzeichnisdienste für sämtliche Unternehmensgruppen und -benutzer bereit.

Ziel des IRP

Durch diesen Incident Response Plan wird sichergestellt, dass bei sicherheitsrelevanten Vorfällen eine zügige und effektive Reaktion eingeleitet und potentielle Schäden hierdurch minimiert und der Betrieb schnellstmöglich wiederhergestellt werden kann. Die Implementation eines IRP stellt eine essentielle Komponente des Sicherheitskonzeptes dar.

Diese Dokumentation bietet eine kurze Übersicht über die wesentlichen Inhalte des von uns erarbeiteten Incident Response Plans, welcher ebenfalls als Leitfaden im Appendix zu finden ist.

Verantwortlichkeiten und Personen

Verantwortlichkeiten	Beschreibung
LDAP-Administrator	Hauptansprechpartner für LDAP-bezogene Vorfälle, verantwortlich für Konfiguration und Zugriffsverwaltung.
IT-Leitung	Entscheidungsträger für Eskalationen und Koordination des technischen Teams.
Wazuh-Administrator	Zuständig für Monitoring und Intrusion Detection mit Wazuh
Web-Administrator	Verantwortlich für den Webserver
Geschäftsführung	Informiert über schwerwiegende Vorfälle, Entscheidungsträger bei strategischen Entscheidungen. Rechtliche Verantwortung.

Alle Mitarbeiter	Melden von potentiellen Sicherheitsrelevanten Vorfällen
-------------------------	--

<Tabelle 2> Verantwortlichkeiten und Ressourcen

<Tabelle 2> soll einen Überblick über die Verantwortlichkeiten des Personals und insbesondere des Management geben. Wir halten es für erforderlich an dieser Stelle darauf hinzuweisen, dass die Gesamtverantwortung sowohl was Innerbetriebliche aber auch rechtliche Aspekte betrifft stets bei der **Geschäftsleitung** liegt und von dieser nicht delegiert werden kann. Das Personal ist angewiesen bei Verdacht auf Sicherheitsrelevante Vorfälle unverzüglich die IT-Abteilung zu informieren, welche diese gewissenhaft zu prüfen und entsprechende Schritte einzuleiten hat.

8.2 Wichtige Dienste auf dem LDAP-Server

In <Tabelle 3> werden nun nochmals die wichtigsten Dienste zusammengefasst, welche für den sicheren Betrieb des LDAP-Servers essentiell sind. Dies umfasst alle Sicherheitsrelevanten Dienste, alle LDAP Dienste sowie das Betriebssystem und deren Funktion.

Dienst	Beschreibung
SSH	Remote-Zugriff auf den Server
SLAPD	LDAP-Dienst für Authentifizierung
AppArmor	Sicherheitsmodul zur Kontrolle von Programmzugriffen
Fail2Ban	Schutz vor Brute-Force-Angriffen und bannen von verdächtigen IP-Adressen
ClamAV/clamav-daemon	Virenschutz zur Erkennung und Beseitigung von Maleware
Snort/Suricata	Netzwerkanalyse und Intrusion Detection und Intrusion Prevention
RkHunter	Überprüfung auf Rootkits und Maleware
UFW (Uncomplicated Firewall)	Firewall zum Schutz des Systems
Ubuntu-Biotic (Ubuntu Linux)	Betriebssystem des LDAP-Servers
NTP (Network Time Protocol)	Sicherstellung der korrekten Uhrzeit/Datum auf allen Endgeräten der Domäne
OpenVPN	VPN-Service nötig damit Remote-Mitarbeiter Zugriff auf das interne Netzwerk haben
Rsyslog (Syslog)	Zentrale Protokollierung von System- und Anwendungslogs
Wazuh-Agent	Zur zentralen Analyse aller Integrierten dienste über den Wazuh Manager
Logrotate	Automatisiert das Management von Log-Dateien, um das Dateisystem vor überfüllten Logs zu schützen

<Tabelle 3> Dienste auf dem LDAP-Server im Überblick

Vorgehensweise im Falle eines Vorfalls

Erkennung und Initiale Analyse

Die Erkennung eines Vorfalls kann durch folgende Methoden erfolgen:

- Alarme von **Wazuh**, **Suricata** oder **Snort** über verdächtige Aktivitäten oder Eindringversuche.
- Meldungen von **Fail2Ban** bei wiederholten, erfolglosen Anmeldeversuchen.
- Auffälligkeiten bei Benutzeraktivitäten, welche auf unberechtigte Zugriffe hindeuten könnten, z.B. unerklärliche LDAP_Authentifizierungen oder Änderungen.
- **ClamAV** oder **RkHunter** melden erkannte Maleware oder Rootkits.
- Plötzliche änderungen in der Reaktionszeit des Systems, der Anwendungen, oder unerklärliche neue und unbekannte Prozesse, neue Software, welche bis dato nicht auf dem System vorhanden waren.

Schritte

- Information des zuständigen Personals (IT-Leitung, LDAP-Administrator oder IT-Abteilung).
- Sammeln und Auswerten der Log-Dateien zur Klärung der Ursache und des Umfangs des Vorfalls.
- Überprüfung der Netzwerkverbindungen und Zugriffe zur Identifikation verdächtiger Aktivitäten
- Bestätigung des Vorfalls oder Schließen des Vorfalls abhängig der Analyse

Eindämmung

Sobald ein Vorfall bestätigt wurde, sind folgende Schritte zur Eindämmung durchzuführen:

- **SSH-Zugriff deaktivieren** oder beschränken, falls unberechtigte Zugriffe erkannt werden.
- **UFW-Regeln anpassen**, um verdächtige IP-Adressen oder Ports zu blockieren.
- **Fail2ban-Konfiguration verschärfen**, um wiederholte Angriffe besser abzuwehren.
- Überprüfung und Sperrung von kompromittierten Benutzerkonten (LDAP-Benutzer).
- Bei Verdacht auf Malware **ClamAV** und **rkhunter** ausführen, um potenziell infizierte Dateien oder Prozesse zu isolieren.

Behebung

Nach Identifikation des Vorfalls und der Ursachen werden diese beseitigt:

Patches für Schwachstellen im Betriebssystem oder in Anwendungen einspielen.

- LDAP-Konfigurationen überprüfen und sichern, um unberechtigte Änderungen rückgängig zu machen.
- Ggf. kompromittierte Benutzerkennwörter zurücksetzen und die Zugriffsrechte überprüfen.

- Bei Malware-Befall infizierte Dateien oder Softwarekomponenten entfernen und das System bereinigen.

Wiederherstellung

Nach der Behebung werden folgende Schritte durchgeführt, um den Normalbetrieb wiederherzustellen:

- Server- und Dienstfunktionen, einschließlich LDAP, werden wieder aktiviert.
- Überwachung wird intensiviert, um sicherzustellen, dass keine weiteren verdächtigen Aktivitäten stattfinden.
- Alle Systeme und Dienste werden auf die neueste Version aktualisiert, falls dies zuvor nicht geschehen ist.

Nachverfolgung und Dokumentation

Nach jedem Vorfall ist eine gründliche **Analyse** durchgeführt:

- Erstellung eines Vorfallsberichts, der die Ursache, den Verlauf und die getroffenen Maßnahmen dokumentiert.
- Überprüfung der Protokolle und Anpassung von Sicherheitsrichtlinien, um ähnliche Vorfälle in Zukunft zu verhindern.
- Schulung des IT-Personals zu neu erkannten Risiken und Verbesserung der Reaktionsprozesse.

Kommunikation

Bei Vorfällen aller Art muss die Geschäftsführung sofort informiert werden. Externe Stellen, wie Datenschutzbehörden oder Kunden, werden bei Datenlecks umgehend benachrichtigt. Die Kommunikation erfolgt nur durch autorisierte Personen, um Informationslecks zu verhindern.

Prävention

Um zukünftige Vorfälle zu vermeiden, werden folgende Maßnahmen ergriffen:

- Regelmäßige Überprüfung von Logs und Audit-Trails.
- Regelmäßige Sicherheitsupdates für Ubuntu und alle installierten Dienste (LDAP, SSH, UFW etc.).
- Durchführung von Penetrationstests auf dem LDAP-Server, um Schwachstellen frühzeitig zu erkennen.
- Schulung der Benutzer in Bezug auf Passwortsicherheit und das Erkennen von Phishing-Angriffen.

8.2 Erstellung von Incident Report-Vorlagen

Um die Kommunikation von Sicherheitsvorfällen effizient zu gestalten wurden von uns **Incident Report** Vorlagen erstellt, diese sind ebenfalls im Appendix dieser Dokumentation zu finden, an dieser Stelle wird nur kurz darauf eingegangen, wann welcher Report zu verwenden ist.

8.2.1 Standard-Report für Sicherheitsvorfälle

Der **Standard-Report** für Sicherheitsvorfälle wird eingesetzt, wenn ein sicherheitsrelevanter Vorfall erkannt wird und eine umfassende Analyse der Ereignisse erforderlich ist. Er dient dazu, die Auswirkungen des Vorfalls zu dokumentieren, die Ursachen zu identifizieren und Maßnahmen zur Vermeidung ähnlicher Vorfälle in der Zukunft zu formulieren.

8.2.2 Report für Systemfehler

Der **Report für Systemfehler** wird eingesetzt, wenn technische Störungen oder Fehler im System auftreten, die den ordnungsgemäßen Betrieb beeinträchtigen. Dieser Report dient dazu, die Fehlerursachen zu analysieren, die Auswirkungen auf die Systemleistung zu dokumentieren und geeignete Korrekturmaßnahmen zu entwickeln.

8.2.3 Report für Datenschutzverletzungen

Der **Report für Datenschutzverletzungen** wird eingesetzt, wenn ein Vorfall festgestellt wird, der zu einer unbefugten Offenlegung oder Verarbeitung personenbezogener Daten führt. Dieser Report dient dazu, die Art und den Umfang der Verletzung zu dokumentieren, die betroffenen Personen zu informieren und Maßnahmen zur Minderung von Risiken sowie zur Einhaltung gesetzlicher Vorgaben zu ergreifen.

8.3 E-Mail-Vorlagen zur Kommunikation von Cyber-Incidents an Mitarbeiter

Die von uns bereitgestellten **E-Mail-Vorlagen zur Kommunikation von Cyber-Incidents** an Mitarbeiter dienen dazu, eine einheitliche und klare Informationsweitergabe im Falle eines Sicherheitsvorfalls sicherzustellen. Sie enthalten grundlegende Informationen über den Vorfall, die potenziellen Auswirkungen, die erforderlichen Maßnahmen für die Mitarbeiter und Ansprechpartner für Rückfragen. Durch die Verwendung solcher Vorlagen wird die Kommunikation effizienter und trägt dazu bei, Missverständnisse zu vermeiden sowie das Vertrauen der Mitarbeiter in die Sicherheitsmaßnahmen des Unternehmens zu stärken.

Bedrohungsmodellierung

9.1 Einführung in die STRIDE-Methode

STRIDE wird häufig in Kombination mit anderen Sicherheitsmethoden eingesetzt, um ein umfassendes Bild der Sicherheitslage eines Systems zu erhalten. Durch die Anwendung der STRIDE-Kategorisierung können Entwickler und Sicherheitsteams proaktiv Maßnahmen zur Risikominderung ergreifen.

Die **STRIDE-Methode** ist ein bewährtes Framework zur Bedrohungsmodellierung, das in der Software- und Systemsicherheit verwendet wird. Sie hilft, potenzielle Sicherheitsrisiken systematisch zu identifizieren, indem sie sechs Arten von Bedrohungen klassifiziert:

Spoofing: Fälschung der Identität eines Benutzers oder Systems.

- **Tampering:** Unbefugte Änderung von Daten oder Systemkomponenten.

- **Repudiation:** Das Abstreiten von Aktionen, die ein Benutzer ausgeführt hat, ohne dass dies nachgewiesen werden kann.
- **Information Disclosure:** Unbefugte Offenlegung vertraulicher Informationen.
- **Denial of Service (DoS):** Unterbrechung des Zugriffs auf ein System oder einen Dienst.
- **Elevation of Privilege:** Unbefugter Zugriff auf privilegierte Funktionen oder Daten.

9.2 Verwendung von Threat Dragon NG zur Durchführung von Bedrohungsanalysen

Die Verwendung von **Threat Dragon NG** zur Durchführung von Bedrohungsanalysen ermöglicht eine strukturierte Identifikation und Bewertung von Sicherheitsrisiken in Softwarearchitekturen. Durch die grafische Modellierung von Bedrohungen und Schwachstellen wird die Kommunikation zwischen Entwicklern und Sicherheitsexperten verbessert, was zu einer effektiveren Risikoanalyse führt. Das ausgearbeitete Bedrohungsmodell ist im Anhang der Dokumentation enthalten und bietet eine detaillierte Grundlage zur Identifikation und Minderung spezifischer Bedrohungen.

10. Test und Validierung

10.1 Testansätze und -methoden

Für die Überprüfung der Implementierung des LDAP-Servers haben wir mehrere Testansätze und -methoden angewendet. Zunächst wurde der Status des LDAP-Dienstes durch den Befehl `systemctl status slapd` überprüft, um sicherzustellen, dass der Server ordnungsgemäß läuft und keine Fehler gemeldet werden. Darüber hinaus wurde die Integrität der Benutzer- und Gruppendatenbanken durch das Abrufen relevanter Datenbankinhalte mittels LDAP-Abfragen validiert. Diese Abfragen ermöglichen uns, die korrekte Konfiguration der Gruppen und Benutzer zu überprüfen und sicherzustellen, dass die erwarteten Einträge vorhanden sind. Zusätzlich wurde getestet, ob die Authentifizierung von LDAP-Benutzern erfolgreich erfolgt, indem SSH-Anmeldeversuche von domänenunabhängigen virtuellen Maschinen durchgeführt wurden. Dieser mehrschichtige Ansatz gewährleistet eine umfassende Überprüfung der Systemkonfiguration und der Funktionalität des LDAP-Servers.

10.2 Validierung der Implementierungsergebnisse

Die Validierung der Implementierungsergebnisse erfolgte durch eine Kombination von Überprüfungen und Tests, die die Funktionalität und Sicherheit des LDAP-Systems bestätigten. Nach der Überprüfung des Dienstestatus und der Datenbankintegrität wurde die Authentifizierungsmechanik durch SSH-Anmeldetests validiert, die von virtuellen Maschinen außerhalb der Domäne durchgeführt wurden. Diese Tests bestätigten, dass Benutzer mit den korrekten Berechtigungen erfolgreich auf das System zugreifen konnten. Zudem wurde sichergestellt, dass die Authentifizierung nicht nur funktioniert, sondern auch robust gegenüber nicht autorisierten Zugriffen ist. Die durchgeführten Validierungen sind dokumentiert und bieten eine solide Grundlage für die Beurteilung der Funktionsfähigkeit und Sicherheit der implementierten LDAP-Lösung.

Das nachfolgende Bild zeigt den Eintrag der „/etc/passwd“ welche validiert, dass die Benutzer erfolgreich im System angemeldet wurden.

```

sshd:x:0:65534::/run/sshd:/usr/sbin/nologin
pollinate:x:110:1::/var/cache/pollinate:/bin/false
vagrant:x:1000:1000:,,,:/home/vagrant:/bin/bash
ubuntu:x:1001:1001:Ubuntu:/home/ubuntu:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
openldap:x:111:116:OpenLDAP Server Account,,,:/var/lib/ldap:/bin/false
kernoops:x:112:65534:Kernel Oops Tracking Daemon,,,:/usr/sbin/nologin
rtkit:x:113:118:RealtimeKit,,,:/proc:/usr/sbin/nologin
avahi-autoipd:x:114:119:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:115:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
whoopsie:x:116:121::/nonexistent:/bin/false
lightdm:x:117:122:Light Display Manager:/var/lib/lightdm:/bin/false
avahi:x:118:124:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
cups-pk-helper:x:119:125:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin
saned:x:120:127::/var/lib/saned:/usr/sbin/nologin
speech-dispatcher:x:121:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/false
colord:x:122:128:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
hplip:x:123:7:HPLIP system user,,,:/var/run/hplip:/bin/false
geoclue:x:124:129::/var/lib/geoclue:/usr/sbin/nologin
pulse:x:125:130:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
clamav:x:126:132::/var/lib/clamav:/bin/false
postfix:x:127:133::/var/spool/postfix:/usr/sbin/nologin
snort:x:128:135:Snort IDS:/var/log/snort:/usr/sbin/nologin
markus.wagner:x:20001:10012:Markus Wagner:/home/users/markus.wagner:/bin/bash
lisa.neumann:x:20002:10012:Lisa Neumann:/home/users/lisa.neumann:/bin/bash
david.zimmermann:x:20003:10012:David Zimmermann:/home/users/david.zimmermann:/bin/bash
martin.kellerer:x:10013:10012:Martin Kellerer:/home/martin.kellerer:/bin/bash

```

System-Dienste LDAP & SSH

Security Software

Angelegte Nutzer

Bild 22. Die angelegten Benutzer /etc/passwd

Zusätzlich hierzu wurden die Benutzer über die grafische Benutzeroberfläche im System angemeldet um die Funktion weiter zu verifizieren.

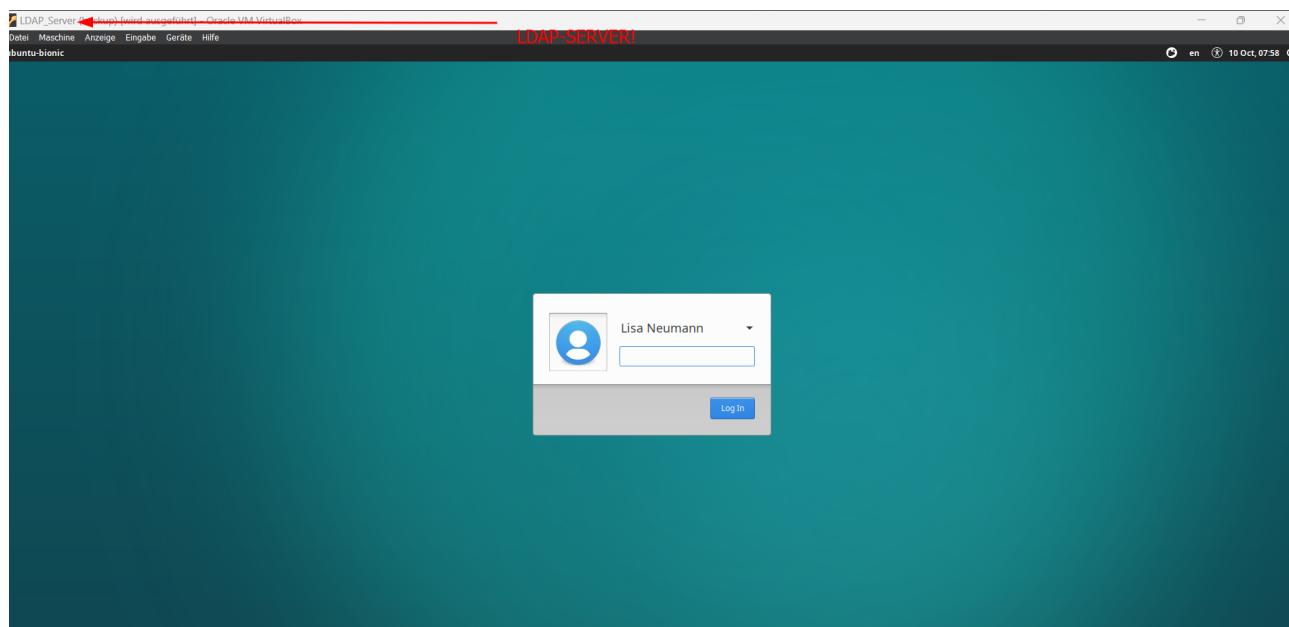


Bild 23. Login Screen der LDAP-Benutzerin „Lisa Neumann“

Die Bilder **Bild 24** und **Bild 25** belegen die korrekte Funktion der LDAP-Authentifizierung sowie die korrekte Anwendung der von uns erstellten und angewendeten

ACL's. <Bild 25> beweist, dass „Lisa Neumann“ sich erfolgreich auf dem LDAP-Dienst anmelden kann. <Bild 26> zeigt eindeutig das die Benutzerin „Lisa Neumann“ nicht über die nötigen Berechtigungen verfügt auf dem LDAP-Server Veränderungen wie in dem Beispiel einen Ordner anlegen vornehmen kann. Auch wird in diesem Bild nachgewiesen, dass sie nicht über sudo Rechte verfügt, wodurch die Sicherheit und Integrität des LDAP-Servers gefestigt wird.

```
lisa.neumann@ubuntu-bionic:~$ pwd | lolcat
/home/users/lisa.neumann
lisa.neumann@ubuntu-bionic:~$
```

Bild 24. Die LDAP-Benutzerin „Lisa Neumann“ ist erfolgreich eingeloggt.

```
lisa.neumann@ubuntu-bionic:~$ mkdir my_docs
mkdir: cannot create directory 'my_docs': Permission denied
lisa.neumann@ubuntu-bionic:~$ sudo mkdir my_docs
[sudo] password for lisa.neumann:
Sorry, user lisa.neumann is not allowed to execute '/bin/mkdir my_docs' as root on ubuntu-bionic.
lisa.neumann@ubuntu-bionic:~$
```

Bild 26. „Lisa Neumann“ kann keine Veränderungen auf dem LDAP-Server vornehmen und verfügt nicht über die nötigen Bechtigungen, den „sudo“ Befehl zu nutzen.

In <Bild 27> wird die Ausgabe des „auth.log“ abgebildet, was die korrekte Anwendung der Benutzer und deren Fähigkeit sich erfolgreich zu Authentifizieren weiter unterstreicht. Die korrekte Vergabe der Berechtigungen wurde weiter oben schon nachgewiesen.

```
vagrant@ubuntu-bionic:~/media/sf_Shared$ sudo tail -f /var/log/auth.log
Oct 10 13:58:53 ubuntu-bionic nscd: nss_ldap: failed to bind to LDAP server ldap://192.168.56.100: Invalid credentials
Oct 10 13:58:53 ubuntu-bionic nscd: nss_ldap: could not search LDAP server - Server is unavailable
Oct 10 13:58:53 ubuntu-bionic nscd: nss_ldap: failed to bind to LDAP server ldap://192.168.56.100: Invalid credentials
Oct 10 13:58:53 ubuntu-bionic nscd: nss_ldap: reconnecting to LDAP server...
Oct 10 13:58:53 ubuntu-bionic nscd: nss_ldap: failed to bind to LDAP server ldap://192.168.56.100: Invalid credentials
Oct 10 13:58:53 ubuntu-bionic nscd: nss_ldap: reconnecting to LDAP server (sleeping 1 seconds)...
Oct 10 13:58:54 ubuntu-bionic nscd: nss_ldap: failed to bind to LDAP server ldap://192.168.56.100: Invalid credentials
Oct 10 13:58:54 ubuntu-bionic nscd: nss_ldap: could not search LDAP server - Server is unavailable
Oct 10 13:58:54 ubuntu-bionic systemd: pam_ldap(systemd-user:account): error reading from nsldc: Connection reset by peer
Oct 10 13:58:54 ubuntu-bionic systemd: pam_unix(systemd-user:session): session opened for user martin.kellerer by (uid=0)
Oct 10 13:59:11 ubuntu-bionic sudo: vagrant : TTY=pts/3 ; PWD=/media/sf_Shared ; USER=root ; COMMAND=/usr/bin/tail -f /var/log/auth.log
Oct 10 13:59:11 ubuntu-bionic sudo: vagrant : pam_unix(sudo:session): session opened for user root by (uid=0)
Oct 10 13:59:22 ubuntu-bionic sshd[10203]: Received disconnect from 192.168.56.101 port 57228:11: disconnected by user
Oct 10 13:59:22 ubuntu-bionic sshd[10203]: Disconnected from user martin.kellerer 192.168.56.101 port 57228
Oct 10 13:59:22 ubuntu-bionic sshd[9958]: pam_unix(sshd:session): session closed for user martin.kellerer
Oct 10 13:59:22 ubuntu-bionic systemd-logind[783]: Removed session 8.
Oct 10 13:59:22 ubuntu-bionic systemd: pam_unix(systemd-user:session): session closed for user martin.kellerer
Oct 10 13:59:42 ubuntu-bionic sshd[10563]: pam_ldap(sshd:account): error reading from nsldc: Connection reset by peer
Oct 10 13:59:42 ubuntu-bionic sshd[10563]: Accepted password for unfa.ehig from 192.168.56.101 port 40128 ssh2
Oct 10 13:59:42 ubuntu-bionic sshd[10563]: pam_unix(sshd:session): session opened for user unfa.ehig by (uid=0)
Oct 10 13:59:42 ubuntu-bionic systemd-logind[783]: New session 10 of user unfa.ehig.
Oct 10 13:59:42 ubuntu-bionic systemd: pam_unix(systemd-user:session): session opened for user unfa.ehig by (uid=0)
```

Bild 27. Die Ausgabe des „auth.log“ als Nachweis der erfolgreichen Anmeldung, der Benutzer zur Validierung der erfolgreichen Erstellung, sowie der korrekten Protokollierung des Anmeldeprozesses.

11. Glossar

11.1 Fachbegriffe und Abkürzungen im Kontext

Fachbegriff/Abkürzung	Beschreibung
Active Directory (AD)	Microsoft Verzeichnisdienst zur Authentifizierung und Autorisierung von Benutzern und Computern in einem Netzwerk
Authorization	Ein Prozess, welcher bestimmt, ob ein authentifizierter Benutzer Zugriff auf bestimmte Ressourcen oder Funktionen hat.
LDAP (Lightweight Directory Access Protocol)	Ein offenes Protokoll zur Abfrage und Modifikation von Verzeichnisdiensten, wird häufig für Authentifizierung und Identitätsmanagement verwendet
SSO (Single Sign-On)	Authentifizierungsmethode, welche Benutzern ermöglicht sich einmal anzumelden um auf mehrere Anwendungen oder Systeme zuzugreifen
SSH (Secure Shell)	Netzwerkprotokoll für sicheren Fernzugriff auf Computer, verwendet Verschlüsselung und Authentifizierung
TLS (Transport Layer Security)	Kryptographisches Protokoll, einsatz für LDAP über SSL (LDAPS)
DN (Distinguished Name)	Ein eindeutiger Bezeichner für einen Eintrag in einem LDAP-Verzeichnis, welcher die Hierarchie des verzeichnisses

	widerspiegelt
RDN (Relative Distinguished Name)	Teil des DN, welcher den Eintrag innerhalb seines übergeordneten Objekts identifiziert
Bind	Der Prozess, bei dem ein Client sich mit einem LDAP-Server verbindet und sich authentifiziert.
Group	Sammlung von Benutzern in einem LDAP-Verzeichnis, die gemeinsame Berechtigungen oder Rollen haben.
Password Policy	Richtlinien, welche festlegen, wie Passwörter verwaltet werden müssen, z.B. Mindestlänge, Komplexität und Ablauf.
Replication	Prozess welcher Synchronisierung von Daten zwischen mehreren LDAP-Servern, um die Verfügbarkeit und Redundanz zu gewährleisten.
Access Control List (ACL)	Liste von Berechtigungen, die regelt, wer auf bestimmte Ressourcen im LDAP-Verzeichnis zugreifen kann.
Kerberos	Netzwerk-Authentifizierungsprotokoll, das auf geheimen Schlüsseln basiert und häufig in Verbindung mit LDAP verwendet wird.
Token	Ein kryptografisches Element, das Informationen zur Authentifizierung und Autorisierung enthält und in einigen Sicherheitsmodellen verwendet wird.
Public Key Infrastructure (PKI)	Framework zur Verwaltung von digitalen Zertifikaten und Public-Key-Verschlüsselung zur Sicherstellung sicherer Kommunikation.
Audit Log	Eine Aufzeichnung aller sicherheitsrelevanten Ereignisse und Aktionen in einem System, die zur Überprüfung und Fehlerbehebung verwendet wird.
Two-Factor Authentication (2FA)	Sicherheitsverfahren, bei dem zwei verschiedene Authentifizierungsmethoden verwendet werden, um die Identität eines Benutzers zu überprüfen.

<Tabelle 4> Glossar

12. Quellenverzeichnis

12.1 Relevante Literatur und technische Dokumentationen

- OpenLDAP Administrator's Guide
Offizielle Dokumentation zu OpenLDAP mit detaillierten Informationen zur Installation, Konfiguration und Verwaltung.
- Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map (RFC 4511)
Ein technischer Überblick über LDAP, der von der IETF bereitgestellt wird.

- **SSH - The Secure Shell: A Technical Overview**
Eine technische Einführung in SSH, die wichtige Sicherheitsaspekte und Anwendungsfälle behandelt.

- **LDAP: An Introduction**
Ein Leitfaden für die Grundlagen von LDAP, der auf die Struktur und Funktionsweise des Protokolls eingeht.
- **NIST Cybersecurity Framework**
Eine umfassende Ressource von NIST, die Richtlinien zur Verbesserung der Cybersicherheit bietet und in vielen IT-Umgebungen verwendet wird.
- **OWASP Top Ten**
Bietet Ressourcen und Informationen zu den Top Ten der Sicherheitsrisiken sowie zur Verbesserung der Anwendungssicherheit.
- **Kali Linux Documentation**
Eine umfassende Sammlung von Anleitungen und Dokumentationen zu Kali Linux, das häufig für Sicherheitstests verwendet wird.
- **Debian OpenLDAP Documentation**
Die offizielle Dokumentation für die Verwendung von OpenLDAP auf Debian-basierten Systemen.
- **The Linux Documentation Project**
Eine Sammlung von Dokumentationen und Handbüchern zu verschiedenen Linux-Themen, einschließlich Sicherheit und Netzwerkmanagement.
- **Free Software Foundation - GNU Privacy Guard (GPG)**
Informationen zu GPG, einer kostenlosen Implementierung des OpenPGP-Standards zur Datenverschlüsselung.

12.2 Online-Ressourcen und Community-Foren

- **OpenLDAP Mailing Lists**
Offizielle Mailinglisten für OpenLDAP, die eine Plattform für Diskussionen, Fragen und Informationen zu OpenLDAP bieten.
<https://www.openldap.org/lists/>
- **Stack Overflow**
Eine große Community von Entwicklern, in der Fragen zu verschiedenen Technologien, einschließlich LDAP und SSH, gestellt und beantwortet werden können.
<https://stackoverflow.com/>
- **Reddit - r/sysadmin**
Ein Subreddit für Systemadministratoren, in dem Informationen, Probleme und Lösungen im Bereich IT-Sicherheit und Netzwerkmanagement geteilt werden.
<https://www.reddit.com/r/sysadmin/>

- **Server Fault**

Eine Q&A-Website für Systemadministratoren, die Diskussionen und Lösungen zu Server- und Netzwerkproblemen bietet.

<https://serverfault.com/>

- **LDAP Support Forum**

Ein Forum speziell für LDAP-bezogene Themen, in dem Benutzer Fragen stellen und

- Lösungen diskutieren können.

<https://wwwldapguru.com/forum/>

- **Kali Linux Forums**

Ein Forum für Benutzer von Kali Linux, das Diskussionen über Sicherheitswerkzeuge, Techniken und Problemlösungen umfasst.

<https://forums.kali.org/>

- **The Linux Community**

Ein allgemeines Forum für Linux-Benutzer, das Diskussionen über alle Aspekte von Linux, einschließlich Sicherheit und Netzwerke, fördert.

<https://wwwlinux.org/forums/>

- **OpenSSH Mailing Lists**

Offizielle Mailinglisten für OpenSSH, auf denen Benutzer Fragen stellen und Informationen über SSH erhalten können.

<https://www.openssh.com/mailing.html>

- **GitHub - LDAP Projects**

Eine Sammlung von Open-Source-Projekten auf GitHub, die sich auf LDAP konzentrieren und von der Community unterstützt werden.

<https://github.com/topics/ldap>

- **NIST Cybersecurity Community**

Eine Plattform von NIST für Cybersecurity-Ressourcen, in der Benutzer Informationen und Tools für Sicherheitspraktiken finden können.

<https://csrc.nist.gov/>

13. Appendix

Die hier aufgeführten Dokumente sind im Ordner Appendix welcher im Verzeichnis dieser Dokumentation ist zu finden. Den Python und Bash Skripten liegt jeweils auch eine README.pdf als Anwendungshilfe bei.

13.1 Bedrohungsmodell

Das mit Theart-Dragon-ng erstellte Bedrohungsmodell für den LDAP-Server als PDF.

13.2 Benutzerhandbuch für die LDAP-Umgebung

PDF Dokument als Leitfaden zur Nutzung des LDAP-Servers.

13.3 Scripts und Automatisierungshilfen

- **base.ldif** Basisstruktur des LDAP
- **ou.ldif** Grundstruktur des LDAP

- **groups.ldif** Erstellt die Gruppen des LDAP
- **create_users.py** Erstellt die Benutzer des LDAP-Abfragen
- **permmisions.sh** Erstellt die ACL und wendet diese an.
- **conf_ufw.py** Konfiguriert die Firewall-Konfiguration

- **conf_suricata.sh** Konfiguriert Suricata
- **aa_config.py** Konfiguration von AppArmor

13.4 Incident Response Plan

Übersichtlicher IRP als PDF speziell für den LDAP-Server „hamster.panzer“

13.5 E-Mail-Vorlagen für die Incident-Kommunikation

Standardisierte Vorlagen zur Information der Belegschaft über Sicherheitsvorfälle.

13.6 Vorlagen für Incident Reports

Standardisierte Vorlagen zur Meldung von Vorfällen (Standard, Systemfehler und Datenschutzverletzung)

13.7 Vagrantfile

Datei welche mit Hilfe der Vagrant Software die von uns verwendete Maschine erstellt.

13.8 Rechnung

PDF Dokumentation mit konkreter Aufschlüsselung der Kosten und der Arbeitszeit.

14. Schlusswort

Die in dieser Dokumentation verwendeten, Diagramme, Bilder und Tabellen wurden von den Autoren selbst erstellt. Das Recht auf Nutzung für Schulungs- und zum Zwecke der Veröffentlichung wird hiermit ausdrücklich eingeräumt.

Sämtliche verwendete Software ist Open-Source und dauerhaft kostenlos nutzbar. Diese Tatsche ermöglichte uns es nicht nur einen sicheren sondern auch Kostengünstigen Server zu erstellen, welchen wir nun in die Hände des Auftraggebers **Smooth Beans GmbH** übergeben.

Ohne die Open-Source-Community wäre dieses Projekt nicht möglich gewesen, als kleines Dankeschön an diese wurde von unserem Team beschlossen, diese Dokumentation einschließlich sämtlicher Skripte, Dokumente und dem Vagrantfile auf GitHub öffentlich zugänglich zu machen, der Link zu diesem Repository ist
<https://github.com/n3M3Z1Z/LDAP-Project>.

