



# Linux Security and Automation

---

Learn with Bilal : <https://www.youtube.com/@bilalmazhar100>

GitHub : <https://github.com/BilalMaz/DevOps-Architect-BootCamp>



# About me

---



Hi there, my name is Bilal, and I will Welcome you to DevOps boot camp! I am thrilled to have you join us for this exciting journey of learning and discovery.

▪

In this boot camp, we will be exploring the principles and practices of DevOps, which is a set of methodologies and tools that aims to bridge the gap between software development and operations. DevOps is an increasingly important area in the field of software engineering, as it helps organizations to streamline their processes, improve their agility, and deliver better value to their customers.

▪

By the end of this boot camp, you will have gained a comprehensive understanding of DevOps and its key concepts, as well as practical skills in areas such as infrastructure automation, continuous integration and delivery, monitoring and logging, and more. You will be equipped with the knowledge and tools to apply DevOps principles in your own work and contribute to the success of your organization.

▪

I am always looking to connect with other professionals in the field, share ideas and insights, and stay up to date on the latest trends and developments. I welcome the opportunity to connect with you and explore ways in which we can collaborate and support each other.

▪

GitHub : <https://github.com/BilalMaz/DevOps-Architect-BootCamp>

LinkedIn : <https://www.linkedin.com/in/bilalmazhar-cyber-security-consultant/>



# Objective and Expectation

---

Expectation is Nothing !!

Objective is to learn 😊

Teaching Method , Prospective + Theory

Prior knowledge

Linux Basic : <https://devsecops-university.com/330-2/>

Ansible Basic : [https://devsecops-university.com/ansible\\_bilal/](https://devsecops-university.com/ansible_bilal/)

Learning and positive mind set

Self learning is key to success !!

There is no Sliver 





# Outline

---

- Security Mind Set ,
- Introduction of Linux Security ,
- User and Group Management ,
- File and Directory Permissions ,
- Firewalls and Network Security ,
- Patch Management,
- Vulnerability assessment & Hardening ,
- Logging and Monitoring ,
- Ansible Security automation



# Security Mind Set ,

---

A security mindset, often referred to as a "security-conscious mindset" or "security-aware mindset," is a way of thinking and approaching tasks, processes, and activities with a strong focus on security.

- Always look for What , How , Why ,
- Risk based approach ,
- Preventive Approach ,
- Research and learning ,
- Ownership ,
- Problem Solving ,
- Communication ,
- Learn and unlearn



# Security 101

---

**Everything starts with CIA** : Confidentiality , Integrity , Availability ,

**Other Factor** : Authentication , Authorization , Accountability ,

## Definition

- A vulnerability, in the context of cybersecurity and information technology, refers to a weakness or flaw in a system, software, hardware, network, or process that can be exploited by a threat actor to compromise the confidentiality, integrity, or availability of the system ,
- **Weaknesses:** Vulnerabilities can take various forms, such as programming errors, misconfigurations, design flaws, or even human errors. They are unintended issues that create potential security risks ,
- **Exploitation:** Threat actors, including hackers and malicious actors, attempt to exploit vulnerabilities to gain unauthorized access, steal data, disrupt services, or carry out other malicious activities.



# Security 101

---

- **Disclosure and Patching:** Once a vulnerability is discovered, it is often disclosed responsibly to the vendor or organization responsible for the affected system or software. They typically release patches or updates to mitigate the vulnerability ,
- **Risk Management:** Vulnerabilities are a critical component of risk management in cybersecurity. Organizations assess and prioritize vulnerabilities based on their potential impact and likelihood of exploitation ,
- **Common Vulnerability and Exposure (CVE):** Vulnerabilities are often assigned unique identifiers known as CVE numbers to facilitate tracking and communication about specific vulnerabilities across the industry ,
- **Security Best Practices:** Mitigating vulnerabilities involves implementing security best practices, including regular software updates, security audits, penetration testing, and adherence to secure coding standards.



# Introduction of Linux Security,

Linux is an important and widely used operating system in today's computing landscape for various reasons, including its **flexibility, reliability, security, and cost-effectiveness.**

- Public Cloud Services ,
- Containerization ,
- Serverless Computing ,
- Web Servers ,
- Supercomputing ,
- Mobile Devices ,
- IoT Devices ,
- Data Centers ,
- We ❤️ Opensource ,
- Cost Saving ,

Security Tools are available for Linux free !!



Examples of Linux operating system



Red Hat Enterprise Linux



Ubuntu



Arch Linux



Kali Linux



Linux Mint



Red Hat Linux



# Prerequisites & Lab Setup

---

- Basic of Linux
- <https://www.vulnhub.com/entry/metasploitable-1,28/>
- <https://www.kali.org/>
- <https://ubuntu.com/>
- <https://www.tenable.com/products/nessus/nessus-essentials>
- <https://www.ansible.com/>
- <https://nmap.org/>
- <https://linuxsecurity.expert/security-tools/top-100/>
- <https://www.sans.org/security-resources/?msc=main-nav>



# User and Group Management

---

**User and Group Management** refers to the processes and tools used in information technology to create, modify, and manage user accounts and groups within an operating system

- **Security Objectives for User and Group Management** ,
  - Authentication and Authorization ,
  - Access Control ,
  - Least Privilege Principle ,
  - Accountability ,
  - Password Policies ,
  - User Monitoring ,



# Secure Account Management

---

**Lab Scenario:** Enhancing User and Group Management Security ,

**Objective:** In this lab, you will practice securing user accounts, implementing strong access controls, and ensuring proper authentication and authorization within a Linux environment ,

**Lab Setup:** A Linux server or virtual machine (e.g., Ubuntu, CentOS, or another Linux distribution of your choice).

- Creating Users and Groups ,
- Implementing Password Policies ,
- User Home Directory Permissions ,
- Group Management ,
- Access Control Lists (ACLs) ,
- Sudo Configuration ,
- User Locking and Account Deactivation ,
- Audit Logging ,
- SSH Security ,



# Creating Users and Groups

---

## Creating User Accounts:

**Step 1:** Open a terminal on your Linux system ,

**Step 2:** Create a new user account named "user1" using the user add command ,

```
bilalworker@bilalworker-virtual-machine: ~  
bilalworker@bilalworker-virtual-machine:~$ sudo useradd BilalMazhar100  
[sudo] password for bilalworker:  
bilalworker@bilalworker-virtual-machine:~$
```

**Step 3:** Set a password for the new user using the passwd command.

```
bilalworker@bilalworker-virtual-machine: ~  
bilalworker@bilalworker-virtual-machine:~$ sudo passwd BilalMazhar100  
New password:  
Retype new password:  
passwd: password updated successfully  
bilalworker@bilalworker-virtual-machine:~$
```



# Creating Users and Groups

---

Step 4: Test the new user's login by switching to the user ,

```
bilalworker@bilalworker-virtual-machine: ~  
bilalworker@bilalworker-virtual-machine:~$ su - BilalMazhar100  
Password:  
su: warning: cannot change directory to /home/BilalMazhar100: No such file or directory  
$ whoami  
BilalMazhar100  
$
```

## Creating a User Group:

Step 1: Create a new group named "mygroup" using the groupadd command ,

```
bilalworker@bilalworker-virtual-machine: ~  
bilalworker@bilalworker-virtual-machine:~$ sudo groupadd devsecop-university  
bilalworker@bilalworker-virtual-machine:~$
```



# Creating Users and Groups

Step 2: Add the user "user1" to the "mygroup" group using the user mod command ,

```
bilalworker@bilalworker-virtual-machine: ~  
bilalworker@bilalworker-virtual-machine:~$ sudo usermod -aG devsecop-university BilalMazhar100  
bilalworker@bilalworker-virtual-machine:~$
```

## Viewing User and Group Information

Step : List all user accounts on the system ,

```
gdm:x:128:134:Gnome Display Manager:/var/lib/gdm3:/bin/false  
bilalworker:x:1000:1000:bilalworker,,,:/home/bilalworker:/bin/bash  
T BilalMazhar100:x:1001:1001:/:/home/BilalMazhar100:/bin/sh  
bilalworker@bilalworker-virtual-machine:~$
```

```
bilalworker@bilalworker-virtual-machine:~$ cat /etc/passwd  
root:x:0:0:root:/root:/bin/bash
```



# Creating Users and Groups

Step 2: List all groups on the system ,

```
docker:x:137:  
BilalMazhar100:x:1001:  
devsecop-university:x:1002:BilalMazhar100  
bilalworker@bilalworker-virtual-machine:~$
```

```
bilalworker@bilalworker-virtual-machine:~$ cat /etc/group  
root:x:0:
```

Step 3: View the details of the "user1" account.



bilalworker@bilalworker-virtual-machine: ~

```
bilalworker@bilalworker-virtual-machine:~$ id BilalMazhar100  
uid=1001(BilalMazhar100) gid=1001(BilalMazhar100) groups=1001(BilalMazhar100),1002(devsecop-university)  
bilalworker@bilalworker-virtual-machine:~$
```



# Implementing Password Policies ,

You can create a new user using the user add or adduser command. Here's an example using user add:

```
bilalworker@bilalworker-virtual-machine: ~  
bilalworker@bilalworker-virtual-machine:~$ sudo useradd testuser  
[sudo] password for bilalworker:  
bilalworker@bilalworker-virtual-machine:~$
```

## Set Password Complexity Rules

`sudo nano /etc/security/pwquality.conf`

```
# Skip testing the password quality for users that are not p  
# /etc/passwd file.  
# Enabled if the option is present.  
# local_users_only  
minlen = 8  
minclass = 4  
  
^G Help      ^O Write Out  ^W Where Is   ^K Cut  
^X Exit      ^R Read File  ^\ Replace    ^U Paste
```



# User Home Directory Permissions

---

**User home directory permissions** refer to the access controls and permissions set on a user's home directory in a Linux or Unix-like operating system. The user's home directory is a personal space where they store their files, configuration files, and data. Properly configuring these permissions is crucial for maintaining the security and privacy of a user's data ,

Key points about user home directory permissions:

- **Owner:** By default, the user is the owner of their home directory ,
- **Group:** Users may belong to a specific group, and group permissions can be set for the home directory ,
- **Others:** Permissions for other users not in the owner's group ,
- **Permission Types:** The permissions include read (r), write (w), and execute (x) for the owner, group, and others ,



- Group: Read and execute permissions.
- Others: No permissions (no read, write, or execute).

# User Home Directory Permissions

---

List the user's

```
bilalworker@bilalworker-virtual-machine:~$ getent passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bilalworker:x:1000:1000:bilalworker,,,:/home/bilalworker:/bin/sh
BilalMazhar100:x:1001:1001::/home/BilalMazhar100:/bin/sh
testuser:x:1002:1003::/home/testuser:/bin/sh
```

## Set Permissions

For each user (user1 and user2), set the following permissions on their respective home directories:

- Owner (user): Full read, write, and execute permissions.
- Group: Read and execute permissions.
- Others: No permissions (no read, write, or execute).



# User Home Directory Permissions

---

```
bash
```

```
sudo chmod 700 /home/user1
```

```
sudo chmod 750 /home/user2
```

**Testing Permissions:** Log in as user1 and create a file in their home directory. Ensure that you can read, write, and execute files in your own home directory. Attempt to access the home directory of user2 while logged in as user1. Verify that you receive a permission denied error.

**Group Membership Testing:** Log in as user2 and create a file in their home directory. Confirm that you can read, write, and execute files in your own home directory. Attempt to access the home directory of user1 while logged in as user2. Verify that you receive a permission denied error.



# Best practices to Secure Access Management

---

- Create individual user accounts for each person using the system. Avoid shared accounts ,
- Enforce password complexity rules, minimum length, and password expiration policies. ,
- Whenever possible, enable MFA for user accounts, especially for administrative accounts. ,
- Assign users the minimum permissions necessary to perform their tasks. Avoid granting unnecessary superuser (root) privileges ,
- Instead of sharing the root password, grant users sudo privileges for specific administrative tasks ,
- Periodically review and audit user accounts to identify and disable or remove inactive or unnecessary accounts. ,
- Set an account lockout policy to disable accounts after a defined period of inactivity. ,
- Configure account lockout policies to protect against brute-force attacks. ,
- Provide security awareness training to users and encourage secure password practices.



# Best practices to Secure Access Management

---

- Educate users about security best practices, social engineering risks, and phishing awareness ,
- Create groups based on roles or functions (e.g., "admins," "developers") to simplify permission management ,
- Assign file and directory permissions based on group membership to simplify access control ,
- Limit the number of users in a group to maintain control over permissions ,
- Establish procedures for reviewing and updating group memberships and policies on a regular basis ,
- Maintain documentation that clearly outlines group membership and associated policies for reference ,



# ACLS : Access Control List ,

---

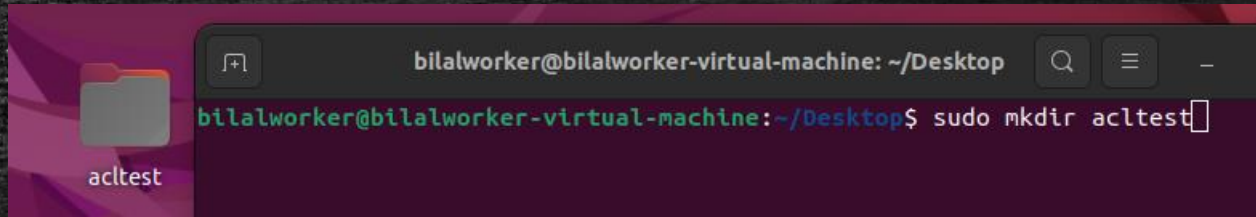
Access Control Lists (ACLs) are a set of permissions attached to files and directories on a computer's file system. ACLs provide a more fine-grained level of access control compared to traditional Unix file permissions. While traditional Unix permissions are based on the owner, group, and others, ACLs allow you to define permissions for specific users and groups beyond the owner and group, providing greater flexibility in access control.

- **Users and Groups:** ACLs can specify permissions for specific users and groups, not just the owner and group of a file ,
- **Multiple Permissions:** ACLs can define multiple permissions for different users and groups on the same file or directory ,
- **Inheritance:** ACLs can be inherited by child objects (e.g., files within a directory) or explicitly set ,
- **Default ACLs:** Some systems support default ACLs that apply to newly created objects within a directory ,
- **Common Permissions:** Common ACL permissions include read (viewing), write (modification), execute (run as a program or access a directory), and special permissions for attributes ,



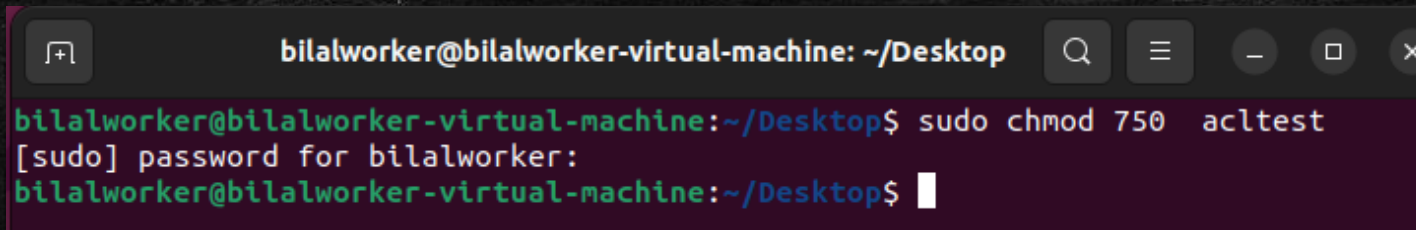
# ACLS : Access Control List ,

**Create Test Directory:** Create a directory where you can practice using ACLs. For example:

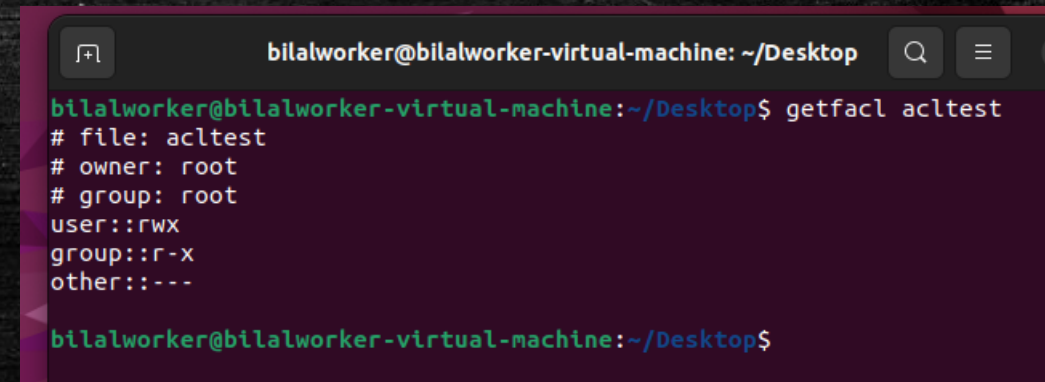
A terminal window titled 'bilalworker@bilalworker-virtual-machine: ~/Desktop' showing the command 'sudo mkdir acltest' being executed. To the left of the terminal, a file manager icon shows a folder named 'acltest'.

```
bilalworker@bilalworker-virtual-machine: ~/Desktop
bilalworker@bilalworker-virtual-machine:~/Desktop$ sudo mkdir acltest
```

Set initial permissions on the directory and create a test file:

A terminal window titled 'bilalworker@bilalworker-virtual-machine: ~/Desktop' showing the command 'sudo chmod 750 acltest' being executed. It prompts for the password for 'bilalworker' and then shows the prompt again.

```
bilalworker@bilalworker-virtual-machine:~/Desktop$ sudo chmod 750 acltest
[sudo] password for bilalworker:
bilalworker@bilalworker-virtual-machine:~/Desktop$
```

A terminal window titled 'bilalworker@bilalworker-virtual-machine: ~/Desktop' showing the command 'getfacl acltest' being executed. The output shows the file's ACL permissions.

```
bilalworker@bilalworker-virtual-machine:~/Desktop$ getfacl acltest
# file: acltest
# owner: root
# group: root
user::rwx
group::r-x
other::---

bilalworker@bilalworker-virtual-machine:~/Desktop$
```



# Best Practices for ACL's

---

- Before implementing ACLs, thoroughly understand your access control needs. Identify which users or groups require specific access permissions ,
- Maintain clear and well-documented ACLs. Document the purpose of each ACL entry, the associated user or group, and the granted permissions ,
- Before implementing ACLs in a production environment, test them in a controlled or staging environment to ensure they meet your requirements without unintended ,
- Instead of creating numerous user-specific ACLs, consider using group ACLs when multiple users require the same permissions. This simplifies management ,
- Apply ACLs with the principle of least privilege in mind. Only grant the minimum permissions necessary to complete tasks ,
- Perform security audits and reviews of ACLs periodically to identify any security weaknesses or misconfigurations ,



# Sudo Configuration ,

---

- The sudo configuration defines which users or groups are allowed to run specific commands with elevated privileges, and it helps enhance security by limiting who can perform administrative tasks on a system ,
- Key points about sudo configuration:
  - **Fine-Grained Access Control:** Sudo allows administrators to specify detailed rules for command execution, providing fine-grained control over who can do , hat on a system ,
  - **Logs and Auditing:** Sudo logs all command usage, providing an audit trail of privileged operations ,
  - **Password Authentication:** Sudo typically requires users to enter their own passwords to execute commands with elevated privilege, enhancing security ,
  - **Centralized Configuration:** Sudo's configuration is typically defined in the /etc/sudoers file, but it can include additional configuration files from the /etc/sudoers.d/ directory ,
  - **Flexibility:** Sudo configurations can be customized to meet the specific needs of an organization, allowing for a balance between security and convenience.



# Lab - Sudo Configuration ,

---

- Ensure that sudo is installed on your Linux system. Most Linux distributions come with sudo pre-installed, but you can check by running ,
- Edit the sudoers file using the visudo command, which opens the configuration file in a safe manner ,

```
# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
bilalworker ALL=(ALL:ALL) ALL
```



**bilalworker@bilalworker-virtual-machine: ~**

```
bilalworker@bilalworker-virtual-machine:~$ sudo visudo
[sudo] password for bilalworker:
```



# Best Practices for Sudo

---

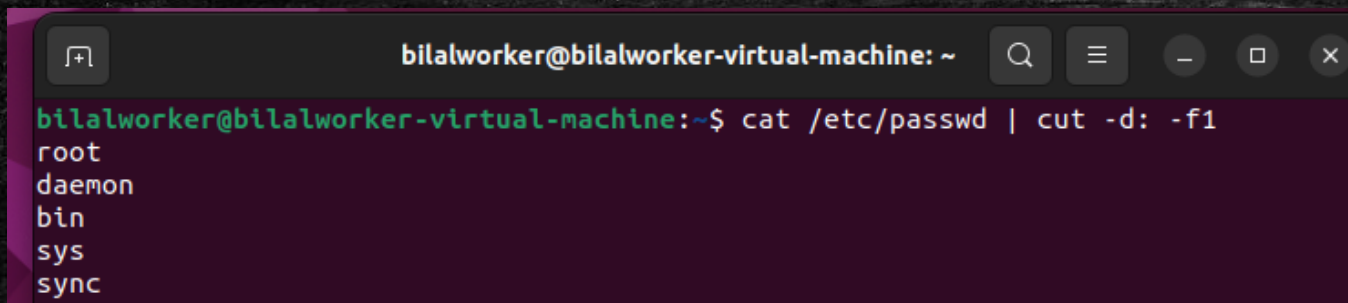
- Define specific rules in the sudoers configuration (/etc/sudoers or files in /etc/sudoers.d/) rather than granting blanket sudo access. Only allow users to execute the commands they need for their tasks ,
- Instead of individually listing users, use groups in sudo rules. This simplifies user management and allows for easier additions and removals from access lists ,
- Only grant sudo access to trusted users who need it. Avoid enabling sudo for all regular users ,
- ensure that sudo logs are enabled and correctly configured ,
- Regularly review sudo logs and configure monitoring and alerting systems to notify you of unusual or unauthorized sudo activity ,
- Never share sudo passwords or allow multiple users to use the same sudo account. Each user should have their own account and password.



# User Locking and Account Deactivation ,

- User locking and account deactivation are security measures used in Linux and Unix-like operating systems to restrict access to user accounts, particularly in situations where accounts are no longer needed or pose a security risk ,
  - **User Locking** refers to suspending a user account, often due to security concerns or temporary inactivity. When a user account is locked, the user cannot log in until it is unlocked ,
  - **Account Deactivation** refers to permanently disabling a user account. Deactivation is typically performed when an account is no longer needed or when a user leaves an organization ,

List Existing Users: Use the cat command to list the existing user accounts on your Linux system:

A terminal window with a dark background and light green text. The window title bar shows 'bilalworker@bilalworker-virtual-machine: ~' and standard window controls. The terminal shows the command 'cat /etc/passwd | cut -d: -f1' being executed, resulting in a list of usernames: root, daemon, bin, sys, and sync.

```
bilalworker@bilalworker-virtual-machine: ~  
bilalworker@bilalworker-virtual-machine:~$ cat /etc/passwd | cut -d: -f1  
root  
daemon  
bin  
sys  
sync
```



# User Locking and Account Deactivation ,

- Create a test user account for this lab (e.g., testuser) using the adduser command ,
  - Lock the User Account: Lock the testuser account using the passwd command with the -l option to disable password-based login ,

```
testuser
bilalworker@bilalworker-virtual-machine:~$ sudo passwd -l testuser
[sudo] password for bilalworker:
passwd: password expiry information changed.
bilalworker@bilalworker-virtual-machine:~$
```

- Unlock the User Account: Unlock the testuser account using the passwd command with the -u option:

```
bilalworker@bilalworker-virtual-machine: ~
bilalworker@bilalworker-virtual-machine:~$ sudo usermod -p '*' testuser
bilalworker@bilalworker-virtual-machine:~$ sudo passwd -u testuser
passwd: password expiry information changed.
bilalworker@bilalworker-virtual-machine:~$
```



# Audit Logging

---

Audit logging is the process of recording and monitoring events or activities on a computer system, network, or application to ensure security, compliance, and accountability. Audit logs provide a detailed record of what happened, when it occurred, who initiated the event, and any relevant details ,

Audit logs typically include information such as:

- User login and logout events ,
- File and directory access ,
- System configuration changes ,
- Network activity ,
- Application usage ,
- Security policy violations ,



# Lab - Audit Logging

- **Check if auditd is Installed:** Verify if the auditd package is installed on your Linux system. Most modern Linux distributions come with auditd pre-installed, but you can check by running:

```
bilalworker@bilalworker-virtual-machine: ~  
bilalworker@bilalworker-virtual-machine:~$ sudo apt-get install auditd  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  libauparse0  
Suggested packages:  
  audispd-plugins  
The following NEW packages will be installed:  
  auditd libauparse0  
0 upgraded, 2 newly installed, 0 to remove and 179 not upgraded.  
Need to get 270 kB of archives.  
After this operation, 876 kB of additional disk space will be used.  
Do you want to continue? [Y/n]
```



# Lab - Audit Logging

Start the Audit Service: If auditd is not already running, start the audit service:

```
bilalworker@bilalworker-virtual-machine:~$ sudo systemctl start auditd
bilalworker@bilalworker-virtual-machine:~$ sudo systemctl status auditd
● auditd.service - Security Auditing Service
   Loaded: loaded (/lib/systemd/system/auditd.service; enabled; vendor preset:
   Active: active (running) since Fri 2023-08-25 22:36:34 PKT; 2min 23s ago
     Docs: man:auditd(8)
           https://github.com/linux-audit/audit-documentation
   Process: 5380 ExecStart=/sbin/auditd (code=exited, status=0/SUCCESS)
   Process: 5389 ExecStartPost=/sbin/augenrules --load (code=exited, status=0/
 Main PID: 5386 (auditd)
    Tasks: 2 (limit: 4548)
   Memory: 556.0K
      CPU: 50ms
   CGroup: /system.slice/auditd.service
           └─5386 /sbin/auditd

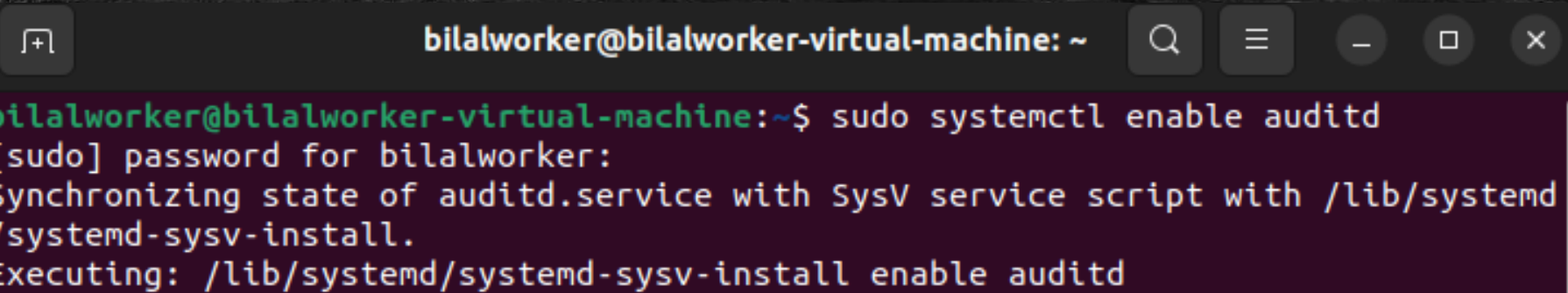
22:36:34 25 اگست bilalworker-virtual-machine augenrules[5402]: enabled 1
```



# Lab - Audit Logging

---

**Enable Auditd Service:** Enable the audit service to start automatically at boot:



```
bilalworker@bilalworker-virtual-machine: ~  
bilalworker@bilalworker-virtual-machine:~$ sudo systemctl enable auditd  
[sudo] password for bilalworker:  
Synchronizing state of auditd.service with SysV service script with /lib/systemd  
/systemd-sysv-install.  
Executing: /lib/systemd/systemd-sysv-install enable auditd
```



# Best Practices for Audit Logging

---

- **Define Clear Objectives:** Clearly define the objectives of your audit. Understand what you need to monitor, why you're monitoring it, and what you hope to achieve ,
- **Understand Regulatory Requirements:** If your organization is subject to regulatory requirements (e.g., GDPR, HIPAA, PCI DSS), ensure that your audit configurations align with these standards ,
- **Use Auditd Tools:** Familiarize yourself with auditd tools like auditctl, ausearch, and aureport to configure, search, and generate reports from audit logs ,
- **Regularly Review Logs:** Periodically review audit logs to identify unusual or suspicious activities. Set up log monitoring and alerting to receive notifications for critical events ,
- **Use Centralized Logging:** Consider sending audit logs to a centralized log management system (e.g., syslog, SIEM) for aggregation, correlation, and long-term storage.



# SSH Security ,

---

- **SSH (Secure Shell)** is a cryptographic network protocol that is used to secure communication between two networked devices. It is commonly used for remote access to servers and network devices, allowing users to log in securely and execute commands on remote systems over an insecure network. SSH is designed to provide secure authentication, encryption, and data integrity, making it a crucial component of modern network security.
- **Key aspects of SSH security include:**
  - Authentication ,
  - Encryption ,
  - Access Control ,
  - Auditing and Logging ,
  - Key Management ,
  - Host Verification



# Lab - SSH Security ,

- Change the SSH Port: Edit the SSH server configuration file to change the default SSH port (typically 22) to a non-standard port (e.g., 2222) ,

```
bash
```

```
sudo nano /etc/ssh/sshd_config
```

```
# For more details, see below on each command. Uncommented options are
# default values.

Include /etc/ssh/sshd_config.d/*.conf

Port 2222
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Es
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J J
```



# Best practices for Secure SSH ,

---

- Disable Root Login ,
- Use Strong Passwords or Key Pairs ,
- Change the Default SSH Port ,
- Implement Key-Based Authentication: ,
- Limit SSH Access ,
- Implement Two-Factor Authentication (2FA) ,
- Regularly Update SSH Software ,
- Enable Connection Timeout ,
- Limit and Monitor SSH Protocol Versions ,



# File and Directory Permissions ,

---

File and directory permissions are essential elements of Unix-like operating systems, including Linux. They determine who can access, modify, or execute files and directories. Understanding and correctly setting permissions are crucial for maintaining system security and data integrity.

**File Permissions:** File permissions define what actions can be performed on a file and who can perform them. There are three types of permissions for files:

**Read (r):** Allows users to view the contents of the file. For directories, it allows listing the files and subdirectories ,

**Write (w):** Permits users to modify the file's content, rename it, or delete it. For directories, it allows creating, deleting, and renaming files and subdirectories within it ,

**Execute (x):** Allows users to execute the file if it's a program or script. For directories, it permits access to the contents if you know the filename but doesn't allow listing files unless you have read permission.



# File and Directory Permissions ,

---

- **Owner (u):** The user who owns the file.
- **Group (g):** Users who are members of the group associated with the file.
- **Others (o):** All other users who are not the owner or in the group.

## Directory Permissions:

- Directory permissions work similarly to file permissions but with some differences:
  - **Read (r):** Allows listing the directory's contents (files and subdirectories).
  - **Write (w):** Allows creating, deleting, or renaming files and subdirectories within the directory.
  - **Execute (x):** Permits access to the directory's contents. To access files or subdirectories within a directory, you need execute permission on the directory itself.



# File and Directory Permissions ,

## ▪ Setting Permissions:

- Permissions can be set using either numeric or symbolic notation:
- Numeric Notation: Uses three octal digits (0-7) to represent the permission bits (e.g., 644 or 755).
- Symbolic Notation: Uses symbols (u, g, o, +, -, =) to modify or set permissions (e.g., u+rwx, o-w).

Notation Type	Description	Example 1	Example 2
<b>Numeric Notation</b>	Represents permissions as an octal number. Each digit corresponds to a permission category and sums the permission values.	<code>chmod 644 file.txt</code> (Owner: Read+Write, Group: Read, Others: Read)	<code>chmod 755 script.sh</code> (Owner: Read+Write+Execute, Group: Read+Execute, Others: Read+Execute)
<b>Symbolic Notation</b>	Uses symbols (u, g, o, +, -) to modify or set permissions for user categories (owner, group, others).	<code>chmod a+rx script.sh</code> (All: Read+Execute)	<code>chmod u=rw,go= private_file.txt</code> (Owner: Read+Write, Group: None, Others: None)



# Firewalls and Network Security in Linux ,

---

- Firewalls are essential components of network security in Linux and are used to control incoming and outgoing network traffic based on a set of predefined rules. They act as barriers between trusted and untrusted networks, helping to protect systems and data from unauthorized access, threats, and attacks. There are two primary types of firewalls in Linux: host-based firewalls and network-based firewalls.
  - Host-Based Firewalls ,
    - **iptables:** iptables is a powerful and versatile firewall management tool in Linux. It allows you to define rules for filtering network packets at the individual host level ,
    - **nftables:** nftables is a newer framework for packet filtering in Linux that provides more flexibility and better performance compared to iptables ,
    - **Firewalld:** firewalld is a dynamic firewall management tool that simplifies firewall configuration for administrators.



# Firewalls and Network Security in Linux ,

---

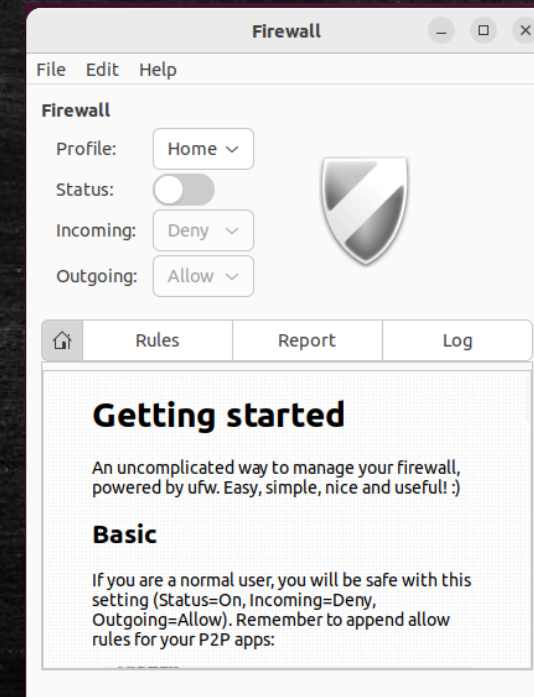
- Network-Based Firewalls:
  - **Netfilter/IPTables:** Netfilter is a framework used in Linux to handle packet filtering, network address translation (NAT), and connection tracking ,
  - **Security-Enhanced Linux (SELinux):** SELinux is a Linux security module that enforces access control policies. While not a traditional firewall ,
  - **Unified Threat Management (UTM) Appliances:** UTM appliances are all-in-one security solutions that often include firewall functionality along with other security features like intrusion detection/prevention, antivirus, and web content filtering. These appliances are typically used in enterprise environments.



# Lab - Firewalls and Network Security in Linux ,

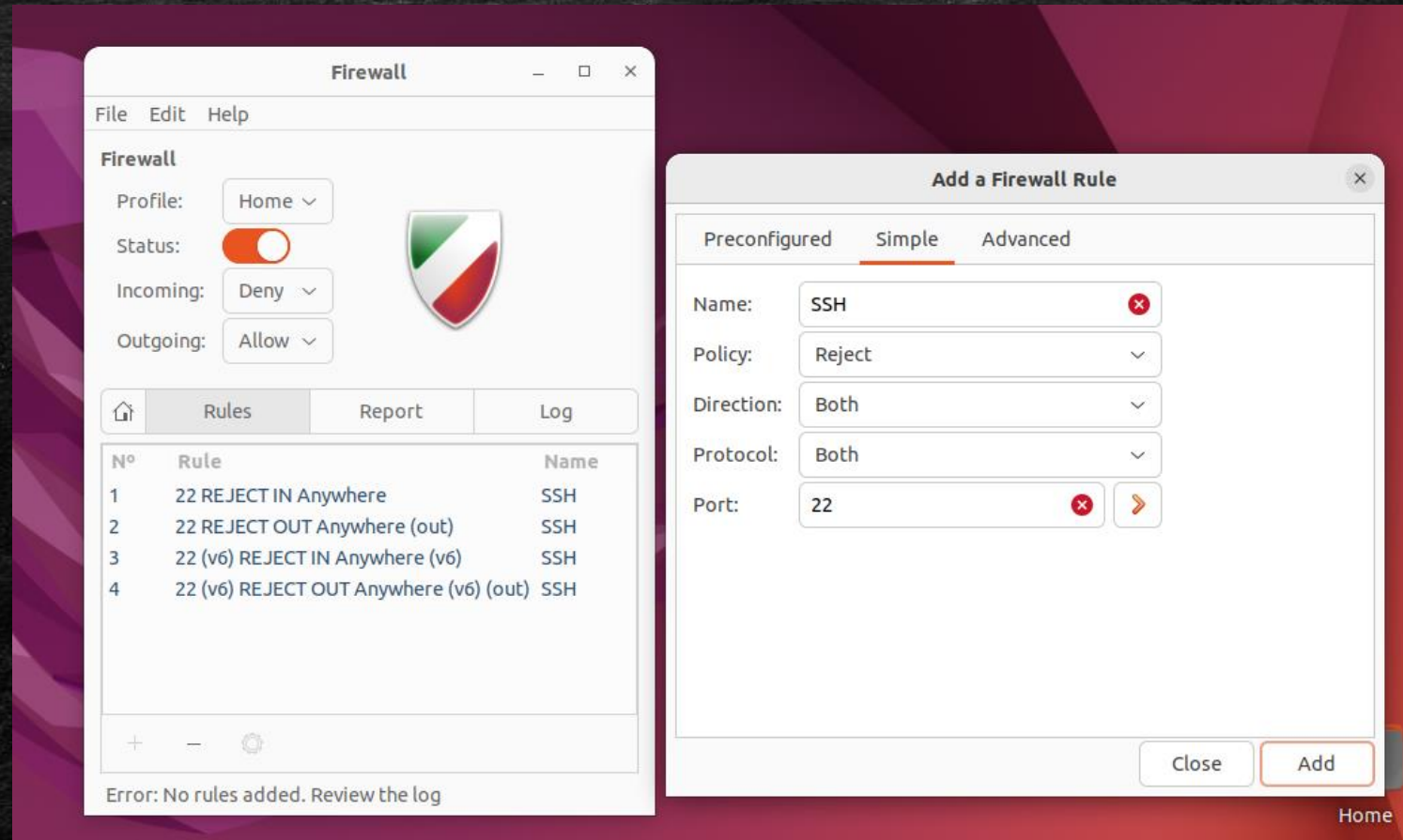
- UFW (Graphical Uncomplicated Firewall) is a user-friendly graphical interface for configuring and managing the iptables firewall in Linux. It provides an easy way for both beginners and experienced users to control incoming and outgoing network traffic on a Linux system through a graphical interface, without the need to write complex firewall rules manually.

```
bilalworker@bilalworker-virtual-machine: ~  
bilalworker@bilalworker-virtual-machine:~$ sudo apt-get install gufw  
[sudo] password for bilalworker:  
Sorry, try again.  
[sudo] password for bilalworker:  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following NEW packages will be installed:  
  gufw  
0 upgraded, 1 newly installed, 0 to remove and 179 not upgraded.  
Need to get 954 kB of archives.  
After this operation, 3,673 kB of additional disk space will be used.  
Get:1 http://pk.archive.ubuntu.com/ubuntu jammy/universe amd64 gufw all 22.04.0-0ubuntu1 [954 kB]  
Fetched 954 kB in 3s (334 kB/s)  
Selecting previously unselected package gufw.  
(Reading database ... 182957 files and directories currently installed.)  
Preparing to unpack .../gufw_22.04.0-0ubuntu1_all.deb ...  
Unpacking gufw (22.04.0-0ubuntu1) ...  
Setting up gufw (22.04.0-0ubuntu1) ...  
Processing triggers for mailcap (3.70+nmu1ubuntu1) ...  
Processing triggers for desktop-file-utils (0.26-1ubuntu3) ...  
Processing triggers for hicolor-icon-theme (0.17-2) ...  
Processing triggers for gnome-menus (3.36.0-1ubuntu3) ...
```





# Lab - Firewalls and Network Security in Linux ,





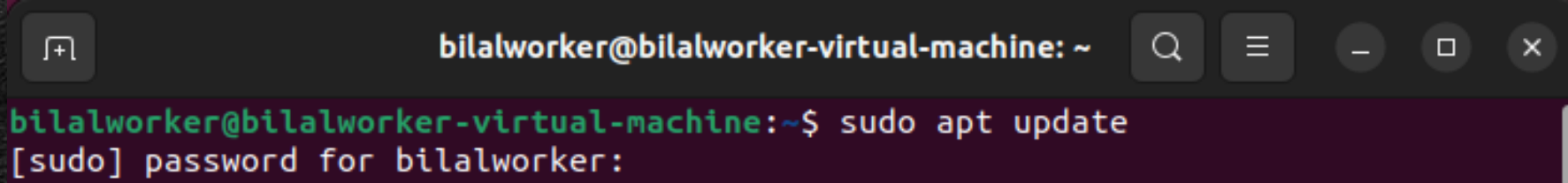
# Cheat sheet

Port Number	Description	Logic
<b>HTTP (80)</b>	Used for unencrypted web traffic.	Allow incoming traffic to port 80 for web server access.
<b>HTTPS (443)</b>	Used for encrypted web traffic (SSL/TLS).	Allow incoming traffic to port 443 for secure web server access.
<b>SSH (22)</b>	Used for secure remote shell access.	Allow incoming traffic to port 22 for SSH access but restrict it to trusted IP addresses or use key-based authentication.
<b>FTP (21)</b>	Used for File Transfer Protocol.	Allow incoming traffic to port 21 for FTP access but consider using secure alternatives like SFTP or FTPS.
<b>SMTP (25)</b>	Used for sending email	Allow outgoing traffic to port 25 for email sending from trusted mail servers.
<b>DNS (53)</b>	Used for domain name resolution	Allow outgoing traffic to port 53 for DNS queries to trusted DNS servers.

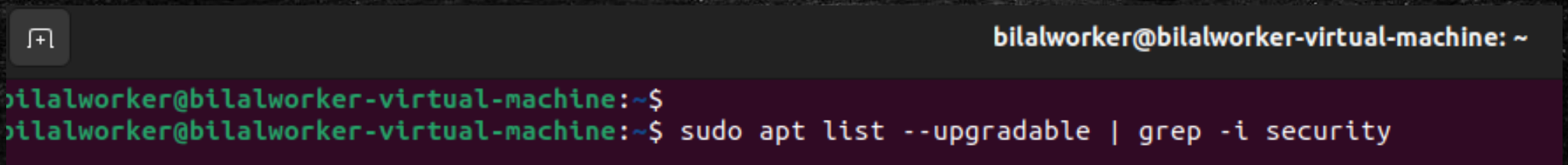


# Patch Management,

- Patch management in Linux is a critical aspect of maintaining the security and stability of your Linux systems. It involves the process of identifying, applying, and managing software updates, including security patches, bug fixes, and feature enhancements. Effective patch management helps mitigate security vulnerabilities, improve system performance, and ensure the reliability of your Linux infrastructure.



```
bilalworker@bilalworker-virtual-machine: ~  
bilalworker@bilalworker-virtual-machine:~$ sudo apt update  
[sudo] password for bilalworker:
```



```
bilalworker@bilalworker-virtual-machine: ~  
bilalworker@bilalworker-virtual-machine:~$  
bilalworker@bilalworker-virtual-machine:~$ sudo apt list --upgradable | grep -i security
```



# Vulnerability assessment & Hardening ,

---

**Vulnerability assessment** is a systematic and comprehensive process used to identify, evaluate, and prioritize vulnerabilities or weaknesses in computer systems, networks, applications, and infrastructure components. These vulnerabilities are areas where a system might be susceptible to security threats or attacks, which could potentially result in unauthorized access, data breaches, or service disruptions

- **Preparation:** Identify the scope of the assessment, including the systems, networks, and assets to be assessed. ,
- **Scanning:** Automated vulnerability scanning tools are used to scan the target systems and networks. The tools attempt to identify vulnerabilities by probing services, applications, and configurations ,
- **Analysis:** Security professionals review the results of the vulnerability scans to validate the findings ,
- **Prioritization and validate :** Vulnerabilities are prioritized based on factors such as severity, exploitability, and potential impact ,
- **Reporting:** A detailed report is generated, documenting the vulnerabilities, their risk assessments, and remediation recommendations



# Tools

---

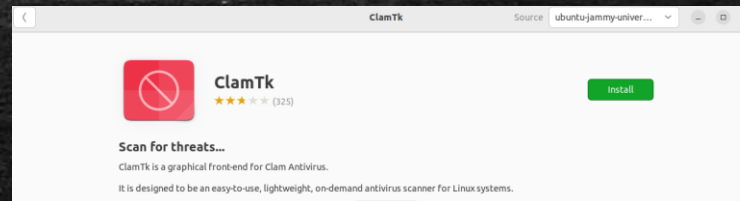
## How to find the Vulnerabilities

- OpenVAS (Open Vulnerability Assessment System) ,
- Nessus ,
- OpenSCAP ,
- Lynis ,
- OWASP ZAP (Zed Attack Proxy) ,
- OSQuery ,
- Rkhunter (Rootkit Hunter) ,
- ClamAV ,
- Namap



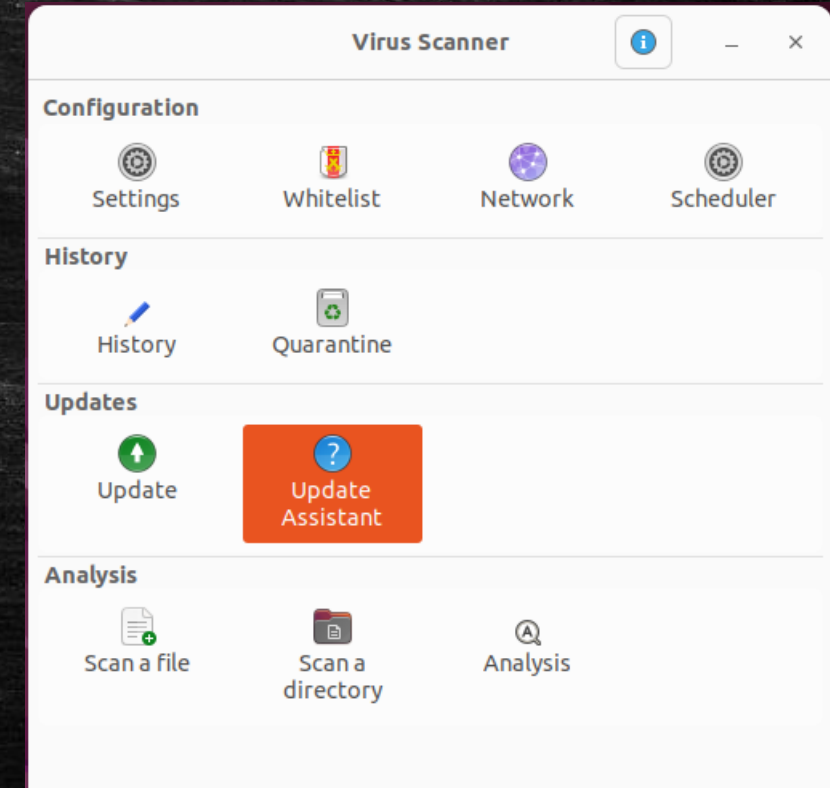
# Lab – VA

ClamAV is an open-source antivirus engine designed for detecting viruses, malware, and other malicious software on Unix-like operating systems, including Linux. It's a popular choice for scanning files, emails, and web content for potentially harmful content



```
bilalworker@bilalworker-virtual-machine: ~/Desktop
bilalworker@bilalworker-virtual-machine:~/Desktop$ ls
acltest bilal Git
bilalworker@bilalworker-virtual-machine:~/Desktop$ clamscan bil.txt
bil.txt: No such file or directory
WARNING: bil.txt: Can't access file

----- SCAN SUMMARY -----
Known viruses: 8671870
Engine version: 0.103.9
Scanned directories: 0
Scanned files: 0
Infected files: 0
Data scanned: 0.00 MB
Data read: 0.00 MB (ratio 0.00:1)
Time: 13.340 sec (0 m 13 s)
Start Date: 2023:08:26 22:12:10
End Date: 2023:08:26 22:12:23
bilalworker@bilalworker-virtual-machine:~/Desktop$
```





# Nessus

The screenshot displays the Nessus Essentials web interface. The top navigation bar includes the 'nessus Essentials' logo, 'Scans' and 'Settings' tabs, a notification bell, and a user profile for 'jmvela'. The left sidebar contains 'FOLDERS' (My Scans, All Scans, Trash) and 'RESOURCES' (Policies, Plugin Rules, Scanners). The main content area is titled 'Discovery' with a 'Back to My Scans' link. It features tabs for 'Hosts' (17), 'Vulnerabilities' (3), 'Notes' (1), and 'History' (1). A search bar and a 'Filter' dropdown are present above a table of hosts. The table lists 10 hosts, each with a checkbox, IP address, and a bar chart showing the number of vulnerabilities (3 for the first host, 2 for the others). To the right, the 'Scan Details' section shows: Name: Discovery, Status: Completed, Policy: Host Discovery, Scanner: Local Scanner, Start: Today at 3:07 PM, End: Today at 3:10 PM, and Elapsed: 3 minutes. Below this, the 'Vulnerabilities' section includes a donut chart and a legend for severity levels: Critical (red), High (orange), Medium (yellow), Low (green), and Info (blue).

Host	Vulnerabilities
192.168.86.35	3
192.168.86.250	2
192.168.86.249	2
192.168.86.248	2
192.168.86.247	2
192.168.86.246	2
192.168.86.238	2
192.168.86.32	2
192.168.86.31	2

**Scan Details**

- Name: Discovery
- Status: Completed
- Policy: Host Discovery
- Scanner: Local Scanner
- Start: Today at 3:07 PM
- End: Today at 3:10 PM
- Elapsed: 3 minutes

**Vulnerabilities**

- Critical
- High
- Medium
- Low
- Info

<https://www.tenable.com/products/nessus/nessus-essentials>



# Vulnerability assessment & Hardening ,

---

System hardening is a security process that involves configuring and strengthening a computer system, network, or software application to reduce its attack surface, minimize vulnerabilities, and enhance its overall security posture. The goal of system hardening is to make it more resistant to security threats and attacks.

- Disable Unnecessary Services ,
- Access Control ,
- Patch Management ,
- Authentication and Authorization ,
- Data Encryption ,
- Firewalls ,
- Monitoring and Logging ,
- Auditing and Compliance,



# CIS Center for Internet Security

The screenshot displays the CIS Center for Internet Security website. At the top, a navigation bar lists various categories: CLOUD PROVIDERS, DESKTOP SOFTWARE, DEVSECOPS TOOLS, MOBILE DEVICES, MULTI FUNCTION PRINT DEVICES, NETWORK DEVICES, OPERATING SYSTEMS (which is highlighted with an orange underline), and SERVER SOFTWARE. Below this, a sidebar titled 'SELECT BY CATEGORY' contains four buttons: IBM, Linux (which is highlighted in blue), Microsoft Windows, and UNIX. The main content area shows a list of operating systems, each with a downward arrow to its right: AlmaLinux OS, Amazon Linux, Azure Linux, Bottlerocket, CentOS Linux, and Debian Family Linux.

SELECT BY CATEGORY	
IBM	
Linux	
Microsoft Windows	
UNIX	

AlmaLinux OS	▼
Amazon Linux	▼
Azure Linux	▼
Bottlerocket	▼
CentOS Linux	▼
Debian Family Linux	▼

<https://www.cisecurity.org/cis-benchmarks>



# Logging and Monitoring ,

**Logging:** Logging involves recording events, actions, and system information in log files. These log files serve as a historical record of system activities and can be invaluable for troubleshooting, auditing, and security analysis. Key aspects of logging in Linux include ,

- Log Files: Linux systems generate a variety of log files located in the `/var/log/` directory. Common log files include `syslog`, `auth.log`, `messages`, and application-specific logs ,

Log level	Severity	Description
DEBUG	Lowest	Detailed information useful for debugging purposes.
INFO	Low	General operational messages about the system's status.
NOTICE	Moderate	Important events or conditions requiring attention.
WARNING	Moderate	Indications of potential issues or conditions that may need review.
ERROR	High	Errors or issues that require immediate attention but are not critical.
CRITICAL	Highest	Critical errors that typically result in system or application failure.
ALERT	Highest	Immediate action required due to a critical condition.
EMERGENCY	Highest	System is unusable, and immediate attention is needed.



# Logging and Monitoring ,

---

- **Log Rotation:** Log files can grow large over time, consuming disk space. Log rotation is the process of compressing, archiving, and deleting old log files to manage disk space effectively ,
- **Application-Specific Logs:** Many Linux applications and services, such as web servers (e.g., Apache, Nginx) and databases (e.g., MySQL, PostgreSQL), maintain their own logs in custom locations.

**Monitoring** involves real-time or near-real-time observation of system activities, performance metrics, and security events. It helps system administrators and security professionals identify issues and respond proactively. Key aspects of monitoring in Linux include:

- **Performance Monitoring:** Tools like top, htop, and vmstat provide real-time insights into CPU, memory, disk, and network usage. Tools like sar (System Activity Reporter) collect performance data over time for analysis ,
- **Resource Utilization:** Monitoring tools like Nagios, Zabbix, and Prometheus enable the tracking of resource utilization, including CPU, RAM, disk space, and network bandwidth ,
- **Security Monitoring:** Security Information and Event Management (SIEM) solutions, intrusion detection systems (IDS/IPS), and log analysis tools help identify security threats and anomalies in log data.



# Logging and Monitoring ,

---

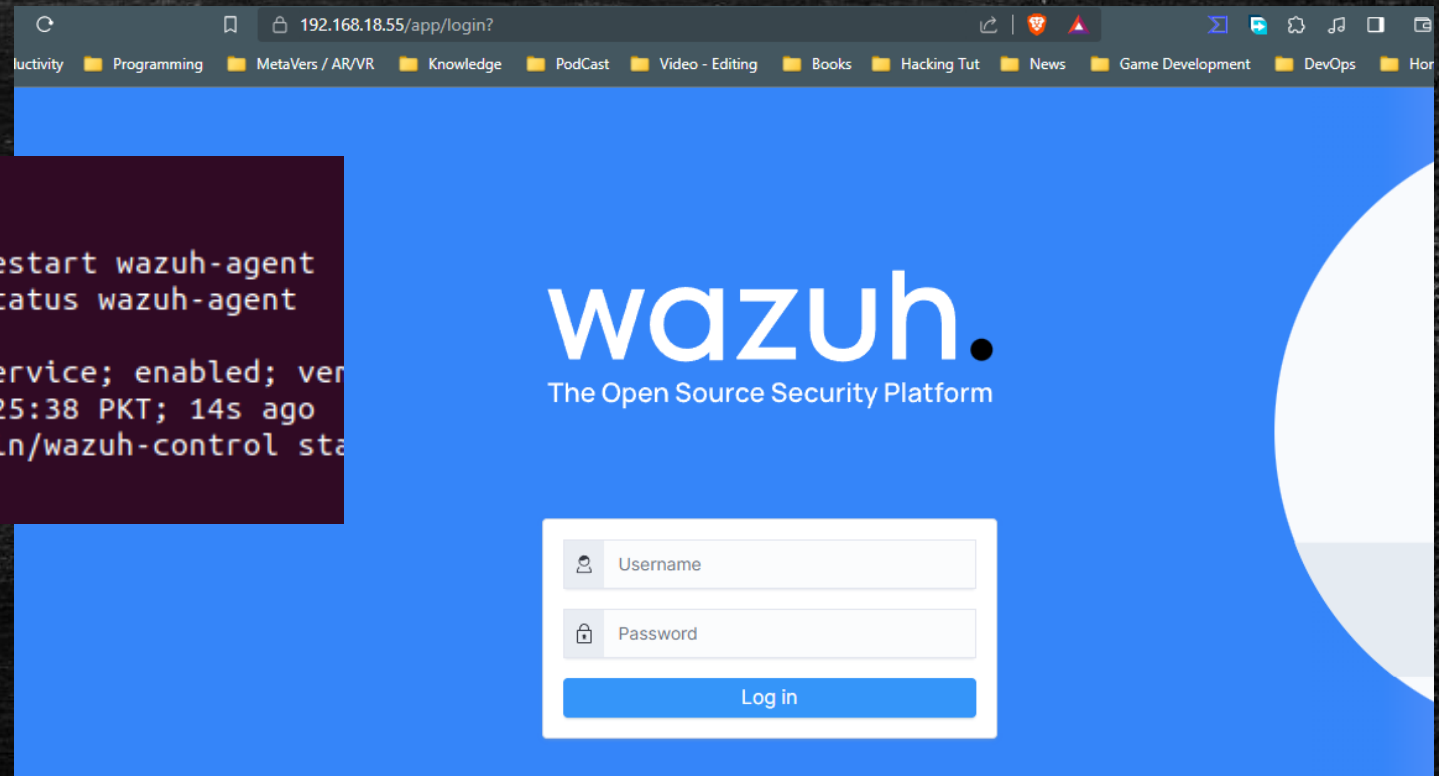
- **Network Monitoring:** Network monitoring tools like Wireshark, tcpdump, and NetFlow analyzers assist in monitoring network traffic for suspicious activities and performance issues ,
- **Alerting:** Monitoring tools often provide alerting mechanisms to notify administrators when predefined thresholds are exceeded or when specific events occur ,
- **Logging and Monitoring Integration:** Combining log data with real-time monitoring helps in contextual analysis and faster incident response. Tools like ELK Stack (Elasticsearch, Logstash, Kibana) are commonly used for log management and analysis ,



# Lab - Logging and Monitoring ,

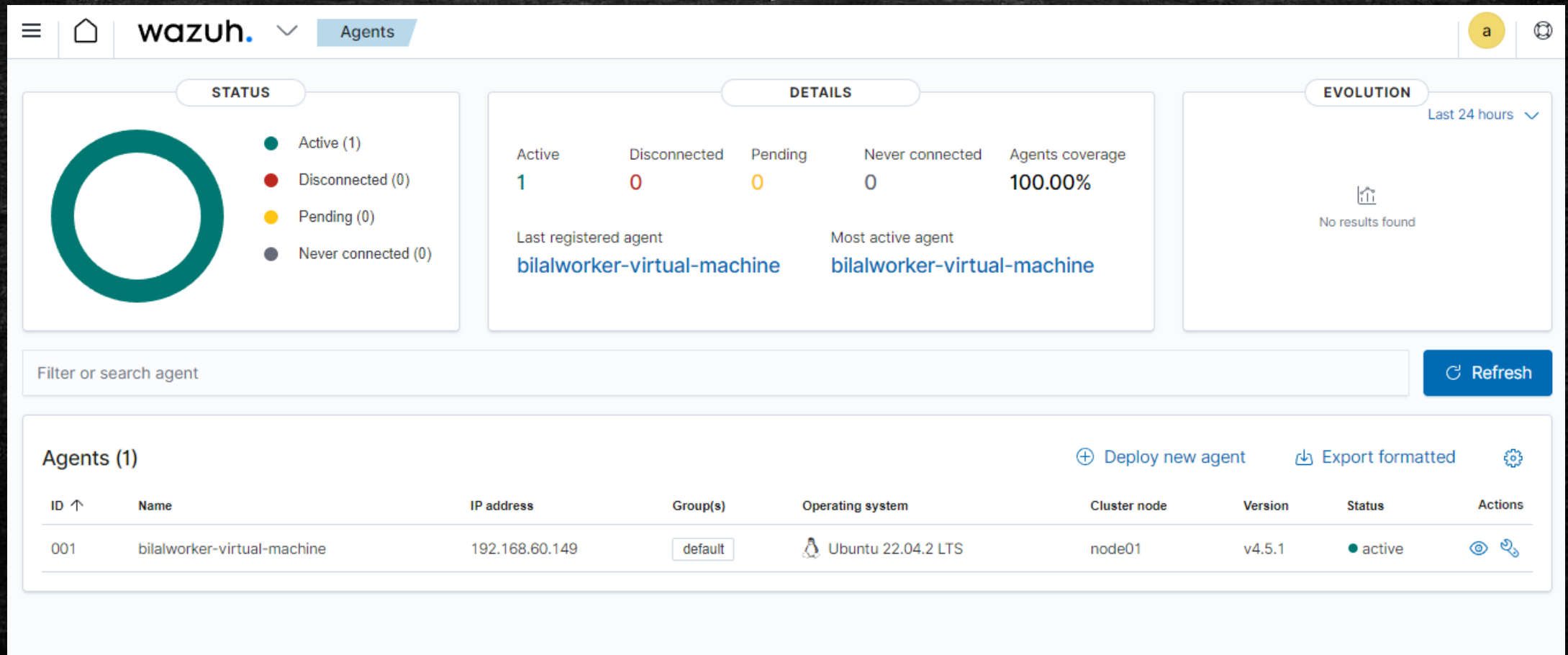
<https://documentation.wazuh.com/current/deployment-options/virtual-machine/virtual-machine.html>

```
bilalworker@bilalworker-virtual-machine:~$ systemctl restart wazuh-agent
bilalworker@bilalworker-virtual-machine:~$ systemctl status wazuh-agent
● wazuh-agent.service - Wazuh agent
   Loaded: loaded (/lib/systemd/system/wazuh-agent.service; enabled; version=4.4.2-1)
   Active: active (running) since Sun 2023-08-27 13:25:38 PKT; 14s ago
     Process: 10001 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start
    Tasks: 32 (limit: 4548)
   Memory: 70.3M
```





# Lab - Logging and Monitoring ,

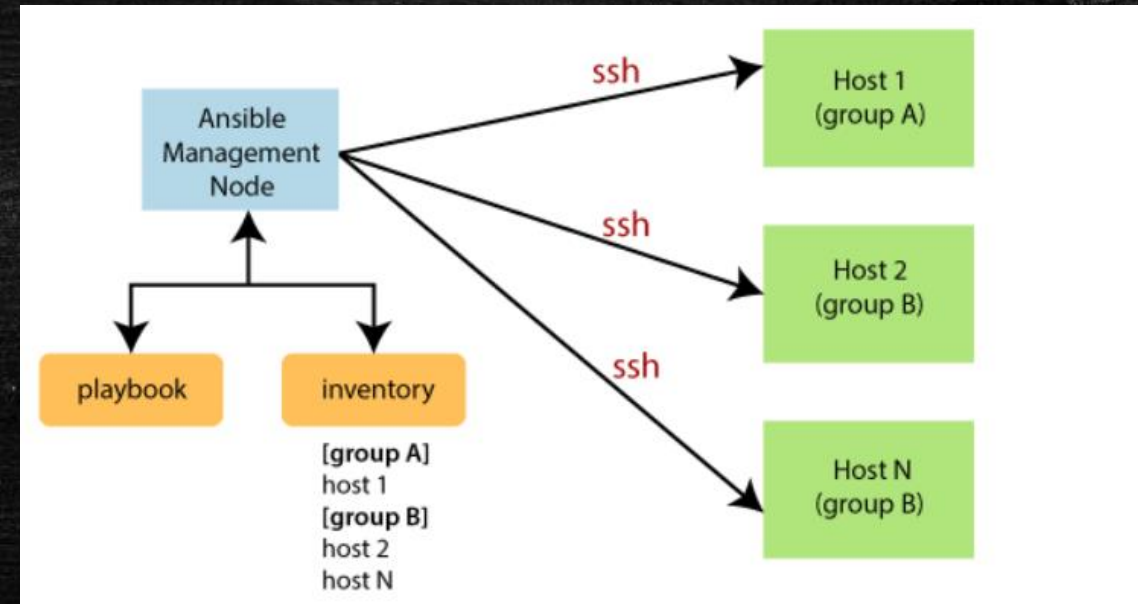




# Ansible Security automation ,

Ansible is an open-source IT engine that automates application deployment, cloud provisioning, intra service orchestration, and other IT tools.

- It can easily connect to clients using SSH-Keys, simplifying though the whole process. Client details, such as hostnames or IP addresses and SSH ports, are stored in the files, which are called inventory files. If you created an inventory file and populated it, then Ansible can use it.
- It is very Simple tool to use yet powerful enough to automate Complex IT applications and infrastructures





# Lab – Ansible Security automation

```
ansible@localhost:~/Desktop/Bilal
File Edit View Search Terminal Help
--- #target Playbook
- hosts: demo
  user: ansible
  become: yes
  connection: ssh
  gather_facts : yes

tasks:
  - name: Run Nmap scan
    command: nmap {{inventory_hostname}}
    register: nmap_output

  - name: Display Nmap output
    debug:
      var: nmap_output.stdout_lines
```

```
ansible@localhost:~/Desktop/Bilal_playbook/nmap
File Edit View Search Terminal Help
[ansible@localhost nmap]$ ansible-playbook bilal_playbook_namp.yml

PLAY [demo] *****

TASK [Gathering Facts] *****
ok: [192.168.60.144]

TASK [Run Nmap scan] *****
changed: [192.168.60.144]

TASK [Display Nmap output] *****
ok: [192.168.60.144] => {
  "nmap_output.stdout_lines": [
    "",
    "Starting Nmap 6.40 ( http://nmap.org ) at 2023-02-17 07:18 PST",
    "Nmap scan report for 192.168.60.144",
    "Host is up (0.0000030s latency).",
    "Not shown: 998 closed ports",
    "PORT      STATE SERVICE",
    "22/tcp    open  ssh",
    "111/tcp   open  rpcbind",
    "",
    "Nmap done: 1 IP address (1 host up) scanned in 1.58 seconds"
  ]
}

PLAY RECAP *****
192.168.60.144      : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```