People's Democratic Republic of Algeria
Ministry of Higher Education and
Scientific Research

Topic :
Securing CentOS

Best configuration practices, patch management, and monitoring tools specific to CentOS

This project was carried out by :

- BAHLOUL Rami
- BOUZENACHA Younes

Supervised by :

— Dr. Kamel SOUADIH

(Academic Year : 2024/2025)

Table des matières

Liste des tableaux

1 General Introduction

Operating system security is a foundational concern in modern system administration, particularly for servers in production environments. CentOS, as a robust derivative of Red Hat Enterprise Linux, has established itself as a preferred choice for professional server deployments due to its stability, reliability, and long-term support.

Despite its inherent security features, any networked operating system faces numerous threats from both external and internal sources. These include unauthorized access attempts, privilege escalation attacks, credential compromise, brute-force intrusions, data exfiltration, and service disruption. The consequences of these security breaches can be severe, ranging from data loss and service outages to compliance violations and reputational damage.

This document provides a comprehensive guide to best practices for enhancing the security of CentOS. It covers key areas such as user and password management, disabling unused services, SSH hardening, file permission management, the use of Access Control Lists (ACLs), SELinux configuration, and advanced firewall management with FirewallD. Each section aims to deliver practical recommendations that can be applied in real-world system administration contexts to ensure optimal security.

Each section offers detailed technical guidance with practical examples that system administrators can immediately apply to strengthen their security posture. The recommendations are designed to balance security requirements with operational functionality, ensuring that systems remain both protected and usable.

The primary objective is to provide administrators with a coherent and structured reference for effectively securing their CentOS systems while aligning with modern cybersecurity standards.

2 CentOs Best Configuration Practices

2.1 User & Password Policies :

In any Linux system, users and their credentials represent the first line of defense. Weak user account policies or poor password hygiene can lead to unauthorized access and compromise of the entire system. On CentOS — especially in production environments — it is critical to implement strict user and password policies to enforce system security and prevent insider or external threats.

Objectives

- Minimize the risk of unauthorized access.
- Enforce strong password standards.
- Control user privileges and prevent privilege escalation.
- Ensure account management aligns with organizational security policies.

2.1.1 Password Policy Configuration

Why Establish a Password Policy ?

A password policy outlines the rules and guidelines for creating, maintaining, and changing passwords. A well-defined policy helps to :

- Enhance Security : Strong, complex passwords reduce the risk of unauthorized access.
- Prevent Account Compromise : Regularly changing passwords helps to mitigate the impact of a potential breach.
- Promote User Awareness : Users become more conscious of security best practices.

1. **Installing Necessary Packages** : Before configuring our password policy, we have to ensure that the `pam_pwquality` module is installed. This module is used to enforce password complexity requirements. We can install it using the following command :

```
sudo yum install pam_pwquality
```

2. **Password Complexity Requirements** : To configure the password complexity, edit the `/etc/security/pwquality.conf` file :

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

   In this file, we can define our password complexity requirements. Here are some common parameters we might want to include :

   - `minlen` : Minimum length of the password.
   - `dcredit` : Maximum number of allowed digits in the password (negative values mean at least that many digits).
   - `ucredit` : Maximum number of allowed uppercase letters.
   - `lcredit` : Maximum number of allowed lowercase letters.
   - `ocredit` : Maximum number of allowed special characters.

   Here's an example configuration :

   ```
   minlen = 12
   dcredit = -1
   ucredit = -1
   lcredit = -1
   ocredit = -1
   ```

   This configuration requires a minimum password length of 12 characters, with at least one digit, one uppercase letter, one lowercase letter, and one special character.

3. **Updating PAM Configuration** : Update the PAM configuration to enforce these policies. Open the `/etc/pam.d/system-auth` file :

   ```
   sudo nano /etc/pam.d/system-auth
   ```

In this file, add the following line :

```
password requisite pam_pwquality.so retry=3
```

The `retry=3` option allows users three attempts to enter a valid password.

4. **Enforcing Password Expiration** : Edit the `login.defs` file to define password aging policies :

   ```
   sudo nano /etc/login.defs
   ```

   Look for the following parameters and set them according to your policy :

   - `PASS_MAX_DAYS` : Maximum number of days a password is valid.
   - `PASS_MIN_DAYS` : Minimum number of days between password changes.
   - `PASS_WARN_AGE` : Number of days before password expiration that the user is warned.

   Example settings :

   ```
   PASS_MAX_DAYS 90
   PASS_MIN_DAYS 1
   PASS_WARN_AGE 7
   ```

   This configuration requires users to change their passwords every 90 days, with a warning issued 7 days before expiration.

2.1.2 User Account Management

1. **Disable Unused Accounts**

   Disable or remove default/system accounts that are not needed.

   ```
   sudo usermod -L username
   ```

   Prevent non-root users from creating accounts :

   ```
   sudo chmod 750 /usr/sbin/useradd
   sudo chmod 750 /usr/sbin/groupadd
   ```

2. **Use Least Privilege**

   Users should only have the minimum access necessary to perform their tasks.

   Remove a user from the `wheel` group :

   ```
   sudo gpasswd -d username wheel
   ```

   Configure `/etc/sudoers` :

   Restrict sudo to specific users/groups :

   ```
   %wheel ALL=(ALL) ALL
   ```

   Limit the commands a user can run :

   ```
   username ALL=(root) /usr/bin/systemctl restart httpd, /usr/bin/
   ```

3. **Use Groups to Manage Permissions**

   Instead of giving users direct permissions, assign them to groups.

   Create a new group :

   ```
   sudo groupadd groupname
   ```

Add a user to the group :

```
sudo usermod -aG groupname username
```

List all users in a group :

```
getent group groupname
```

4. **Enforce Account Inactivity Policy**
   Automatically disable accounts after a period of inactivity using `chage`.
   Set an account to expire on a specific date :

   ```
   sudo chage -E YYYY-MM-DD username
   ```

   Automatically disable inactive accounts after X days :

   ```
   sudo chage -I 30 username
   ```

   Check a user's aging and inactivity settings :

   ```
   sudo chage -l username
   ```

5. **Lock Accounts After Failed Attempts (pam_tally2)**
   Edit the following files :
   `/etc/pam.d/system-auth` and `/etc/pam.d/password-auth` :

   ```
   auth    required    pam_tally2.so deny=5 unlock_time=900 onerr=
   account required    pam_tally2.so
   ```

   Check failed login attempts :

   ```
   sudo pam_tally2 --user=username
   ```

Unlock a locked account :

```
sudo pam_tally2 --user=username --reset
```

6. **Avoid Shared Accounts**
   Each user should have a unique login for auditing and accountability.

2.2 Disable Unused Services

Minimizing the number of active services on a CentOS system is a fundamental security practice. Unnecessary services can consume system resources, increase the attack surface, and potentially introduce vulnerabilities. By identifying and disabling these services, administrators can enhance system performance and security.

Objectives :

- **Reduce Attack Surface :** Limit potential entry points for attackers.
- **Optimize Resource Usage :** Free up CPU and memory by stopping unneeded services.
- **Enhance System Stability :** Prevent conflicts and reduce the likelihood of service failures.

Identifying Active Services
To list all active services :

```
systemctl list-units --type=service --state=running
```

To list all installed services and their enablement status :

```
systemctl list-unit-files --type=service
```

Disabling and Removing Unwanted Services

1. **Stop the Service**
   Immediately halt the service's operation :

   ```
   sudo systemctl stop <service-name>
   ```

2. **Disable the Service at Boot**
   Prevent the service from starting automatically on system boot :

   ```
   sudo systemctl disable <service-name>
   ```

3. **Mask the Service**
   Completely prevent the service from being started manually or automatically :

   ```
   sudo systemctl mask <service-name>
   ```

   Masking is a stronger measure than disabling and should be used when you want to ensure the service cannot be started under any circumstances.

4. **Remove the Service Package**
   If the service is not required, uninstall its package :

   ```
   sudo yum remove <package-name>
   ```

   Ensure that removing the package does not affect other system functionalities.

Cleaning Up Residual Files :

After disabling or removing services, it's prudent to clean up any residual files :

```
sudo systemctl daemon-reload
```

This command reloads the systemd manager configuration, ensuring that changes take effect.

Common Services to Evaluate

Service Name	Description	Considerations
`cups`	Printing service	Disable if the system doesn't require printing capabilities.
`bluetooth`	Bluetooth support	Disable on servers or systems without Bluetooth hardware.
`avahi-daemon`	Zeroconf networking	Often unnecessary on servers.
`postfix`	Mail transfer agent	Disable if not sending emails directly from the server.

Table 1 – Services that can often be safely disabled

2.3 SSH Hardening in CentOS

SSH (Secure Shell) is a critical service for remote system administration. However, its default configuration may expose the system to brute-force attacks and unauthorized access. SSH hardening involves implementing best practices that enhance the security of SSH access to CentOS systems.

1. **Use SSH Key Authentication Instead of Passwords**
   Using SSH key authentication provides a more secure alternative to password-based login. SSH keys use a cryptographic key pair for authentication, making it nearly impossible to brute-force access. Steps :
   - **Generate SSH key pair on the client machine : :**

     ```
     ssh-keygen -t rsa -b 4096
     ```

   - **Copy the public key to the server :**

     ```
     ssh-copy-id user@centos-server
     ```

     SSH will copy the public key to /.ssh/authorizedkeys on the server and set the correct permissions.

   - **Connect using the key :**

     ```
     ssh user@your-centos-server
     ```

2. **Disable Root SSH Login**

   Allowing root to log in directly over SSH can be dangerous. Disabling root SSH access forces users to log in with a normal user account and escalate privileges when necessary, which improves auditing and security.

   Steps :

   - **Open the SSH configuration file :**

     ```
     sudo vi /etc/ssh/sshd_config
     ```

   - **Find and change the line :**

     ```
     PermitRootLogin no
     ```

   - **Restart the SSH service :**

     ```
     sudo systemctl restart sshd
     ```

3. **Restrict Allowed Users**

   Restricting which users can log in via SSH reduces the risk of unauthorized access. This can be done using the AllowUsers or AllowGroups directives in the SSH configuration file.

   Steps :

   - **Edit the SSH configuration file :**

     ```
     sudo vi /etc/ssh/sshd_config
     ```

   - **Add specific allowed users or groups :**

     ```
     AllowUsers user1 user2
     AllowGroups sshusers
     ```

   - **Restart the SSH service :**

     ```
     sudo systemctl restart sshd
     ```

2.4 File Permissions and Ownership in CentOS

File permissions and ownership in CentOS are a foundational part of its security model. Properly configuring these attributes helps protect system files and user data from unauthorized access or modification. Misconfigured permissions are a common security weakness that can lead to privilege escalation, data leakage, or system compromise.

2.4.1 Understanding Linux File Permissions

In CentOS (and other Linux distributions), every file and directory is assigned :

- **Owner :** The user who owns the file.
- **Group :** The group that owns the file.
- **Permission Bits :** Define read, write, and execute permissions for the owner, group, and others.

Permissions are typically displayed using the `ls -l` command, and consist of 10 characters. For example :

```
-rwxr-xr-- 1 user group 1234 Apr 30 12:00 example.txt
```

Explanation :

- The first character indicates the file type :
  - `-` for regular files
  - `d` for directories
- The next 9 characters represent permissions :
  - `r` = read
  - `w` = write
  - `x` = execute

  These are grouped as :

  - First 3 : permissions for the **owner**
  - Next 3 : permissions for the **group**
  - Last 3 : permissions for **others**

2.4.2 Types of Permissions

Permission	Files	Directories
`r`	Read file contents	List directory contents
`w`	Modify file contents	Add, remove, or rename files
`x`	Execute file as a program/script	Enter the directory (use `cd`)

TABLE 2 – Effect of Permissions on Files vs. Directories

2.4.3 Changing Permissions and Ownership

Changing Permissions

`chmod` is used to modify file permissions :

- **Symbolic mode** : `chmod u+x file.txt` (adds execute permission for the user)
- **Numeric mode** : `chmod 755 file.sh` (sets permissions to `rwxr-xr-x`)

Changing Ownership

`chown` changes the file owner and/or group :

— `chown root file.txt`

— `chown user:group file.txt`

`chgrp` changes only the group :

— `chgrp admin file.txt`

Special Permission Bits

- **Setuid (s)** : Executes the file with the owner's privileges.
- **Setgid (s)** : Executes with group privileges ; on directories, new files inherit the group.
- **Sticky bit (t)** : On directories, only the file owner can delete their own files (commonly used in `/tmp`).
- **Example** : `chmod 1777 /tmp` (Sticky bit for shared directory)

Security Best Practices

- Restrict access to sensitive files : System configuration files like `/etc/passwd`, `/etc/shadow`, or `/etc/ssh/sshd_config` should have strict permissions :

— `chmod 600 /etc/shadow`

— `chown root:root /etc/shadow`

- Avoid world-writable files :

  — `find / -type f -perm -002`

- Regularly audit permissions with tools :

  — `find / -perm -4000` → find SUID files

  — `ls -la /` → check top-level directory permissions

  — `stat <file>` → detailed metadata

- Use `umask` to define default permissions for new files and directories. A commonly secure value is `umask 027`.

2.5 Access Control Lists (ACLs) in CentOS

Access Control Lists (ACLs) provide a more granular level of file system permissions than the traditional Unix permissions model. While the standard Unix permissions (owner, group, and others) define a basic level of access, ACLs allow administrators to specify fine-grained access for individual users and groups beyond just the owner and group. This enhanced flexibility is particularly useful in complex environments where multiple users need varying levels of access to a single file or directory.

2.5.1 Understanding ACLs in CentOS

In CentOS, ACLs are implemented via the `acl` package. When ACLs are enabled on a file system, you can assign permissions to specific users and groups beyond the traditional owner-group-other model. ACLs provide fine-grained permission control.

ACL Structure :

- **User ACL :** Permissions for specific users.
- **Group ACL :** Permissions for specific groups.
- **Mask ACL :** Maximum allowed permissions for users and groups.
- **Other ACL :** Permissions for everyone else.

Permission Types :

- `r` (read) : Read the contents of a file.

- `w` (write) : Modify the contents.
- `x` (execute) : Run the file if executable.
- `d` (delete) : Delete the file or directory.

2.5.2 Enabling ACLs in CentOS

- **Check if ACL is enabled :** `mount | grep acl`
- **Enable ACL temporarily :** `mount -o remount,acl /mount-point`
- **Make ACL permanent :** Edit `/etc/fstab` :

  ```
  /dev/sda1   /   ext4   defaults,acl   0   1
  ```

- Then remount : `mount -o remount /mount-point`

2.5.3 Working with ACLs

Viewing ACLs :

— `getfacl file.txt`

Setting ACLs :

— For a user : `setfacl -m u:user:r-- file.txt`

— For a group : `setfacl -m g:group:rw- file.txt`

— Default ACLs : `setfacl -d -m u:user:r-- /home/shared/`

Removing ACLs :

— Specific entry : `setfacl -x u:alice file.txt`

— All ACLs : `setfacl --remove-all file.txt`

2.5.4 Use Cases for ACLs

Case 1 : Multiple Users with Different Permissions

— `setfacl -m u:user1:rw /data/sharedfile`

— `setfacl -m u:user2:r-- /data/sharedfile`

— `setfacl -m u:user3:--- /data/sharedfile`

Case 2 : Granting Access to Multiple Groups

— `setfacl -m g:team:rw /data/project/`

— `setfacl -m g:admins:rwx /data/project/`

Case 3 : Setting Default Permissions

— `setfacl -d -m u:alice:r-- /home/shared/`

2.5.5 Security Considerations

- Limit ACL usage to avoid complexity.
- Audit regularly using `getfacl`.
- Combine ACLs with traditional Unix permissions.

Access Control Lists in CentOS provide advanced control over file access, offering administrators the ability to define precise permissions for individual users and groups. When used effectively, ACLs enhance security and flexibility in multi-user environments.

2.5.6 SELinux in CentOS

Security-Enhanced Linux (SELinux) is a mandatory access control (MAC) security mechanism implemented in the Linux kernel. Initially developed by the National Security Agency (NSA) and integrated into the Red Hat Enterprise Linux (RHEL) family, including CentOS, SELinux provides an additional layer of system security beyond the traditional discretionary access control (DAC) model. This framework is essential for hardening CentOS environments, particularly in enterprise and high-security deployments.

Core Concepts and Architecture

Mandatory Access Control
Unlike traditional Unix permissions that operate on a discretionary basis, SELinux enforces security policies that cannot be overridden by users or applications. This ensures that even if an application is compromised, its damage is constrained within predefined security boundaries.

Security Contexts
SELinux assigns security contexts to all system entities—files, processes, ports, and users. Each context consists of three components :

- **User** : The SELinux user identity.
- **Role** : The authorized roles for the user.
- **Type/Domain** : The specific type for files or domain for processes.

A typical security context appears as `user:role:type:level`, such as :

- **Files :** `system_u:object_r:httpd_sys_content_t:s0`

- **Processes :** `unconfined_u:unconfined_r:httpd_t:s0-s0:c0.c1023`

Type Enforcement

The primary security mechanism in SELinux is type enforcement, where access decisions are based on the type of the source (typically a process) and target (typically a file). Policy rules define which source types can access which target types and in what manner.

SELinux Modes SELinux can operate in three modes :

1. **Enforcing :** Security policy is enforced ; violations are denied and logged.
2. **Permissive :** Security policy is not enforced, but violations are logged.
3. **Disabled :** SELinux is completely turned off.

Implementation in CentOS

Default Configuration

CentOS implements SELinux in enforcing mode by default with the targeted policy, which focuses on protecting selected system services while allowing most user processes to run unconfined.

Policy Types :

- **Targeted :** Default policy that restricts only targeted processes.
- **Minimum :** Similar to targeted but with fewer confined domains.
- **MLS (Multi-Level Security) :** Implements Bell-La Padula mandatory access controls with security levels.

Management Tools

Command-Line Utilities :

— `sestatus` – Displays SELinux status and policy information.

— `getenforce/setenforce` – View or change current enforcement mode.

— `getsebool/setsebool` – View or set SELinux boolean values.

— `semanage` – Manages SELinux policies and mappings.

— `restorecon` – Restores default SELinux contexts.

— `chcon` – Changes SELinux context for files.

— `audit2allow` – Generates policy rules from denied operations.

— `audit2why` – Translates SELinux audit messages into explanations.

Graphical Tools :

— `system-config-selinux` – GUI for managing SELinux (must be installed).

— `setroubleshoot-server` – Provides user-friendly explanations for SELinux denials.

Best Practices for SELinux in CentOS

1. **Verify SELinux Status :**

```
sestatus
getenforce
```

2. **Configure SELinux Mode in `/etc/selinux/config` :**

```
SELINUX=enforcing
SELINUXTYPE=targeted
```

3. **Relabel the Entire Filesystem after major changes :**

```
touch /.autorelabel
reboot
```

Troubleshooting and Maintenance

1. **Identifying SELinux Denials :**

```
grep "denied" /var/log/audit/audit.log
```

2. **Using setroubleshoot for Analysis :**

```
sealert -a /var/log/audit/audit.log
```

3. **Managing File Contexts :**

```
ls -Z /path/to/file
restorecon -Rv /path/to/directory
semanage fcontext -a -t httpd_sys_content_t "/custom/web/path(/.*)?
restorecon -Rv /custom/web/path
```

4. **Managing Port Contexts :**

```
semanage port -l | grep http
semanage port -a -t http_port_t -p tcp 8080
```

5. **Managing Booleans :**

```
getsebool -a
getsebool -a | grep httpd
setsebool -P httpd_can_network_connect on
```

Performance Optimization

1. **Use Permissive Domains during troubleshooting :**

```
semanage permissive -a httpd_t
```

2. **Policy Module Development :**

```
audit2allow -a -M mymodule
semodule -i mymodule.pp
```

3. **Periodic Maintenance :**
    - Review and update boolean settings.
    - Analyze audit logs for denial patterns.
    - Update custom policies as application requirements change.

Conclusion

SELinux is a fundamental security component in CentOS that provides essential protection against a wide range of threats. When properly configured and managed, it significantly enhances system security with minimal performance impact. Although it requires an initial investment in learning and configuration, the security benefits far outweigh the administrative overhead, especially in production environments where security is paramount.

The layered approach of SELinux complements other security measures, creating a defense-in-depth strategy that is critical for modern infrastructure security. Organizations deploying CentOS should fully embrace SELinux as a core security control rather than disabling it for convenience.

2.5.7 FirewallD in CentOS : Advanced Network Security Management

FirewallD is the default firewall management tool in CentOS 7 and newer. It provides a dynamic, zone-based interface to manage firewall rules without interrupting active connections.

Core Architecture

- **D-Bus Interface :** Enables integration and programmatic control.
- **Runtime vs Permanent Configurations :** Changes can be applied live or saved persistently.
- **Zone-Based Management :** Different zones for different levels of network trust.
- **XML Config Files :** Human-readable and structured.

Zones and Use Cases

Zone	**Description**	**Typical Use Case**
`drop`	Lowest trust level ; all incoming connections are dropped	Public untrusted networks
`block`	Similar to drop, but with ICMP host prohibited responses	Public untrusted networks requiring feedback
`public`	For untrusted public networks	Default for internet-facing interfaces
`external`	For use on external networks with NAT masquerading	Router/gateway scenarios
`internal`	For internal networks when the system acts as a gateway	Protected internal networks
`dmz`	For computers in the DMZ	Demilitarized zone servers
`work`	For work environments	Corporate networks
`home`	For home environments	Home networks
`trusted`	All connections are accepted	Highly trusted networks

Table 3 – FirewallD Zones and Use Cases

Basic Commands

```
sudo systemctl start firewalld
sudo firewall-cmd --state
sudo firewall-cmd --get-active-zones
sudo firewall-cmd --zone=public --add-service=http
sudo firewall-cmd --permanent --add-service=http
sudo firewall-cmd --reload
```

Managing Ports and Services

```
# List all services
sudo firewall-cmd --get-services

# Add port
sudo firewall-cmd --zone=public --add-port=8080/tcp

# Forward port
sudo firewall-cmd --add-forward-port=port=80:proto=tcp:toport=8080
```

Rich Rules Examples

```
# Allow HTTP for local subnet
sudo firewall-cmd --add-rich-rule=’rule family="ipv4" \
source address="192.168.1.0/24" service name="http" accept’

# Limit HTTP connections
sudo firewall-cmd --add-rich-rule=’rule service name="http" \
limit value="5/m" accept’
```

Advanced Features

- **Interface Assignment :** `--change-interface`
- **Source-Based Filtering :** `--add-source`
- **Masquerading and NAT :** `--add-masquerade`
- **ICMP Filtering :** `--add-icmp-block`
- **Direct Rules :** `--direct --add-rule`

Logging and Best Practices

- Enable logging with `--set-log-denied`
- Adopt a default deny strategy
- Restrict by source IP when possible
- Regularly review and clean up obsolete rules

Conclusion :

FirewallD is a robust and modern firewall solution for CentOS. Its zone-based model enhances security by aligning firewall policy with network topology, offering fine-grained control for dynamic environments.

3 CentOS Patch Management

3.1 Patching CentOS Linux Systems : Why, How, and When

CentOS, one of the most popular distributions of the Linux operating system, is often chosen for its stability, open-source nature, and robust community support. However, as with all software, CentOS is not exempt from the necessity of regular patching. In this article, we'll delve into the importance of patching CentOS systems, the steps involved, and best practices to follow.

3.1.1 Why Patch CentOS ?

1. **Security :** Perhaps the most compelling reason, patching ensures that vulnerabilities in the system are addressed. Cyber threats evolve continuously, and software vulnerabilities can be a gateway for malicious actors.
2. **Bug Fixes :** Aside from security issues, patches can resolve functional errors or bugs in the operating system, leading to smoother and more reliable system performance.
3. **New Features :** Occasionally, updates might introduce new functionalities or improve existing ones, enhancing the user experience.
4. **Compatibility :** New software or tools might require updated system libraries or components. Regular patching ensures your system remains compatible with the latest software.

3.1.2 How to Patch CentOS ?

Patching CentOS is a straightforward process, thanks to the built-in package management tool, `yum`.

1. **Check for Updates :** Before applying patches, it's wise to check what updates are available. Use the following command :

   ```
   yum check-update
   ```

2. **Update the System :** To update all packages and their dependencies, run :

   ```
   yum update
   ```

3. **Update a Specific Package :** If you only wish to update a specific package, use this command :

   ```
   yum update [package_name]
   ```

4. **Clean Up :** After updates, it's a good practice to clean the local cache :

   ```
   yum clean all
   ```

3.1.3 When to Patch CentOS ?

1. **Regularly :** Setting a regular schedule, such as monthly or quarterly, ensures your system remains updated. However, the frequency might vary based on your organization's requirements or the criticality of the updates.
2. **After a Vulnerability Alert :** If there's news of a significant vulnerability, don't wait for your regular schedule. Patch immediately.
3. **Post-Installation :** : After a fresh install of CentOS, it's advisable to run an update to ensure you have the latest patches.

3.2 Best Practices :

1. **Backup First :** Before any significant system change, especially patching, always back up essential data.
2. **Test in a Staging Environment :** If possible, first apply the patches in a staging or test environment. This will give you a preview of any potential issues.
3. **Monitor After Patching :** : After a fresh install of CentOS, it's advisable to run an update to ensure you have the latest patches.Keep an eye on system performance and functionality after the update to catch any unforeseen issues.

3.3 conclusion

patching CentOS systems is a crucial administrative task that ensures the security, efficiency, and reliability of your server environment. With regular updates and following best practices, you can maximize the benefits of CentOS while mitigating potential risks.

4 Package Updates : Keeping Software Current

Keeping your CentOS system up to date is a key part of maintaining its security and stability. This involves setting up automatic updates, applying security patches, and performing regular vulnerability scans.

4.1 Automatic Updates

Setting up automatic updates ensures that security patches are applied without manual intervention.

- **CentOS 7** : Use `yum-cron`

  ```
  sudo yum install yum-cron -y
  sudo systemctl enable yum-cron
  sudo systemctl start yum-cron
  ```

- **CentOS 8 and later** : Use `dnf-automatic`

  ```
  sudo dnf install dnf-automatic -y
  sudo systemctl enable --now dnf-automatic.timer
  ```

4.2 Security Patches Only

To apply only security-related updates, use the following command :

```
yum update --security
```

This command helps fix known vulnerabilities without updating the entire system.

4.3 Vulnerability Scanning : Nessus vs. OpenVAS

Vulnerability scanners detect weaknesses such as outdated packages or misconfigurations. Two of the most popular tools are Nessus and OpenVAS.

4.3.1 Nessus

- **Type** : Commercial (free for personal use)
- **Pros** :
  - User-friendly interface
  - Frequently updated vulnerability database
  - Detailed and actionable reports
- **Best For** : Enterprises needing a ready-to-use and professionally supported tool.

4.3.2 OpenVAS

- **Type** : Open-source and free
- **Pros** :
  - Highly customizable
  - Integrates with other tools
  - Community-based support
- **Best For** : Security teams wanting flexibility and full control.

4.4 Summary

Nessus is recommended for organizations looking for a polished, commercial-grade experience. OpenVAS is ideal for advanced users who prefer open-source tools and full customization.

Central Patch Management : Spacewalk vs. Foreman

Both **Spacewalk** and **Foreman** are open-source centralized systems management solutions that help organizations manage software updates and patches across their infrastructure. Let us break down and compare each platform.

1. Spacewalk

Spacewalk was the upstream open-source project for Red Hat Satellite, developed by Red Hat to manage RHEL-based systems.

Key Features

- Software package management and distribution
- System inventory and monitoring
- Configuration management
- Bare-metal provisioning
- Virtual machine provisioning (via Cobbler integration)
- Errata and patch management

Pros

- Mature platform with extensive documentation
- Strong Red Hat/CentOS management capabilities
- Well-established community support
- Simple interface for package management tasks

Cons

- Limited support for non-Red Hat systems
- Development has slowed significantly
- Official support ended in May 2020 (Red Hat moved to Satellite 6)

2. Foreman

Foreman is a full lifecycle systems management tool for physical and virtual servers. With the Katello plugin, it adds advanced content and patch management features.

Key Features

- Provisioning from bare-metal to full OS installation
- Configuration management integration (Puppet, Ansible, Salt, Chef)
- Comprehensive monitoring and reporting
- Content and patch management (via Katello)
- Multi-tenancy support
- Support for multiple hypervisors

Pros

- Actively developed with regular updates
- Broader OS support (RHEL, CentOS, Debian, Ubuntu)
- Deep integration with modern DevOps tools
- Modular, flexible architecture
- Foundation of Red Hat Satellite 6

Cons

- Steeper learning curve
- More complex setup and configuration
- Requires more system resources

3. Key Differences

- **Development Status :** Spacewalk is deprecated ; Foreman is actively maintained.
- **Platform Support :** Spacewalk focuses on Red Hat-based systems ; Foreman supports a wider range of Linux distributions.

- **Architecture :** Foreman uses a modular, plugin-based design ; Spacewalk is more monolithic.
- **Configuration Management :** Foreman integrates with modern tools like Ansible and Puppet ; Spacewalk is focused on traditional package management.
- **Future Roadmap :** Foreman has an active community and roadmap ; Spacewalk has been end-of-lifed.

4. When to Choose Each

Choose Spacewalk if :

- You manage mainly RHEL/CentOS 6 or 7 systems
- You already have a working Spacewalk setup
- You prefer a simpler, lighter interface
- Your infrastructure needs are modest

Choose Foreman if :

- You want a future-proof solution
- You manage diverse Linux environments
- You need integration with DevOps tools
- You require advanced provisioning and automation
- You're setting up a new centralized patch management system

5. Conclusion

For most modern environments, **Foreman with Katello** is the recommended solution due to its active development, broader compatibility, and deeper integration with modern infrastructure and automation tools. Spacewalk remains useful for legacy systems but is no longer maintained.

5 Monitoring Tools Specific to CentOS

Monitoring the health, performance, and security of your CentOS system is crucial for ensuring reliability and preventing issues. Below are some key tools specifically suited for CentOS systems :

1. Nagios

Overview :
Nagios is a widely used open-source monitoring tool for IT infrastructure, including servers, network devices, and applications. It works well on CentOS and provides extensive monitoring capabilities.

Key Features :

- Monitors system resources (CPU, memory, disk, network).
- Customizable alerts based on thresholds.
- Supports plugins to extend monitoring functionality.

How It Works :

- Install on CentOS using the official repository or by compiling from source.
- Configure monitoring for services, hardware, and more.
- Send notifications via email or SMS based on alerts.

2. Zabbix

Overview :
Zabbix is another powerful, open-source monitoring solution that offers scalability for both small and large infrastructures.

Key Features :

- Comprehensive monitoring of servers, networks, and applications.
- Data visualization tools (graphs, charts, and maps).
- Agent-based and agentless monitoring options.

How It Works :

- Install the Zabbix server on CentOS and configure agents on monitored hosts.
- Create monitoring templates to track system performance.
- Set up triggers and actions to automate alerting and reporting.

3. Cockpit

Overview :

Cockpit is a web-based system management tool that comes pre-installed on CentOS and provides a simple interface for monitoring and managing servers.

Key Features :

- Real-time monitoring of system performance.
- User-friendly dashboard for managing system resources (CPU, RAM, disk usage).
- Integration with Docker and virtual machines.

How It Works :

- Access Cockpit via a web browser (default port 9090).
- Monitor system health and view logs, network traffic, and running services.
- Manage system services, users, and storage directly from the interface.

4. Prometheus and Grafana

Overview :

Prometheus is a popular open-source monitoring tool that collects metrics, and Grafana is often used to visualize those metrics in dashboards.

Key Features :

- Prometheus collects metrics from CentOS servers, applications, and services.
- Grafana provides a powerful dashboard for visualizing these metrics in real-time.

- Alerts can be set up in Prometheus to notify administrators when thresholds are reached.

How It Works :

- Install Prometheus on CentOS and configure it to scrape metrics from desired endpoints.
- Install Grafana to visualize the metrics and create custom dashboards.
- Set up alerting rules in Prometheus to monitor key performance indicators.

5. Netdata

Overview :

Netdata is a real-time performance monitoring tool that offers an easy-to-use interface and is lightweight on system resources.

Key Features :

- Real-time monitoring of CPU, RAM, disk, network, and services.
- Visualizes metrics with interactive dashboards.
- Alerts for performance issues or anomalies.

How It Works :

Install Netdata using the following command :

```
bash <(curl -Ss https://my-netdata.io/kickstart.sh)
```

Then access the web interface at `http://localhost:19999` to view real-time metrics.

6. Sysstat (iostat, mpstat, pidstat, sar)

Overview :

Sysstat is a collection of command-line tools that provides system performance statistics and resource utilization metrics on CentOS.

Key Features :

- `iostat` : Monitors CPU and disk I/O performance.
- `mpstat` : Displays statistics by processor.
- `pidstat` : Provides statistics on processes.

- `sar` : Collects, reports, and saves system activity information.

How It Works :

Install the sysstat package :

```
sudo yum install sysstat
```

Then use the included commands to monitor system performance over time.

Intrusion Detection Systems (IDS)

AIDE (Advanced Intrusion Detection Environment)

- Monitors file integrity by comparing current file states with a known baseline.
- Alerts on unauthorized changes.

Usage :

```
sudo aide —init
sudo aide —check
```

OSSEC

- Host-based IDS (HIDS).
- Monitors logs, detects rootkits, checks file integrity, and sends alerts.
- Can integrate with email, SIEM tools, or centralized dashboards.

Log Monitoring Tools (e.g., Logwatch, Logcheck, Logwatch + Logmatch)

Logwatch

- Summarizes and emails daily reports of system logs (e.g., SSH login attempts, sudo usage).
- Helps identify anomalies quickly.

Logcheck

- Filters log messages and alerts administrators about unusual activity.

Logmatch (from SEC)

- Pattern-matching tool for real-time log monitoring.
- Can be configured to look for specific keywords or behaviors in logs.

Why Monitoring is Important

- Helps identify system issues before they impact performance.
- Provides insights into resource usage for better capacity planning.
- Enhances security by detecting unusual activity or performance degradation.

System Auditing : Track System Events

System auditing allows administrators to monitor and record sensitive events and user activities. In CentOS, the `auditd` (Audit Daemon) provides a powerful framework to track system calls, file access, permission changes, and more.

1. Configure auditd

The `auditd` service runs in the background and logs security-relevant events based on custom rules.

- Tracks file modifications, command executions, and user actions.
- Helps in detecting policy violations, privilege escalation, or suspicious behavior.
- Logs are stored in `/var/log/audit/audit.log`.

2. Define Audit Rules

Rules are defined in the file `/etc/audit/rules.d/audit.rules` to specify what events should be captured.

Example : Monitor changes to the password file `/etc/passwd` :

```
-w /etc/passwd -p wa -k passwd_changes
```

— `-w` = watch the file

— `-p wa` = monitor write and attribute changes

— `-k` = keyword used for log filtering

3. Analyze Audit Logs

Once the rules are in place, audit logs can be analyzed with the following tools :

ausearch

`ausearch` allows filtering audit logs using keywords or event types.

```
ausearch -k passwd_changes
```

aureport

`aureport` provides summarized reports of all audit events (logins, file access, etc.).

```
aureport --file
aureport --auth
```

These tools help administrators investigate incidents and generate reports for compliance.

6 General Conclusion

Securing a CentOS server is not a one-time task, but a continuous process that combines system hardening, patch management, intrusion detection, and active monitoring.

Keeping the system up to date with regular or automatic updates helps fix known vulnerabilities. Tools like **Nessus** and **OpenVAS** assist in identifying potential threats before they can be exploited.

Intrusion Detection Systems such as **AIDE** and **OSSEC** help detect unauthorized changes to critical files. Log monitoring tools like **Logwatch** and **Logcheck** provide real-time visibility into system activity.

With **auditd**, system administrators can track sensitive operations and ensure accountability. Centralized solutions such as **Foreman** and **Spacewalk** allow consistent update management across multiple servers.

In summary, securing CentOS requires a layered approach — combining proactive patching, continuous monitoring, and audit-based controls — to protect infrastructure against evolving threats.

References

1. https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/selinux_guide/

2. https://wiki.centos.org/HowTos/SELinux

3. https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/security_guide/

4. https://www.nsa.gov/What-We-Do/Cybersecurity/SELinux/

5. https://firewalld.org/documentation/

6. https://www.openssh.com/manual.html

7. https://wiki.centos.org/Manuals/ReleaseNotes/CentOS7

8. https://csrc.nist.gov/publications/detail/sp/800-123/final

9. https://www.oreilly.com/library/view/ssh-the-secure/0596000111/

10. https://greenhost.cloud/how-to-set-password-policy-on-a-centos/

11. https://www.nostarch.com/firewalls.htm

12. https://documentation.suse.com/sles/15-SP3/html/SLES-all/cha-acls.html

13. https://sysward.com/solutions/centos-patching/