Hardening the Operating System: A Step-by-Step Guide

By Chinwendu Imegwu, Deandre Brown, Devin Young, & Bangalie Koroma

What is OS Hardening?

The process of strengthening an operating system's security posture to reduce vulnerabilities and protect against cyberattacks.

- Involves configuring settings, applying patches, and implementing security measures to minimize attack surface.
- Crucial for protecting sensitive data, maintaining system integrity, and ensuring business continuity.
- Unhardened systems are highly susceptible to data breaches, malware infections, and unauthorized access.

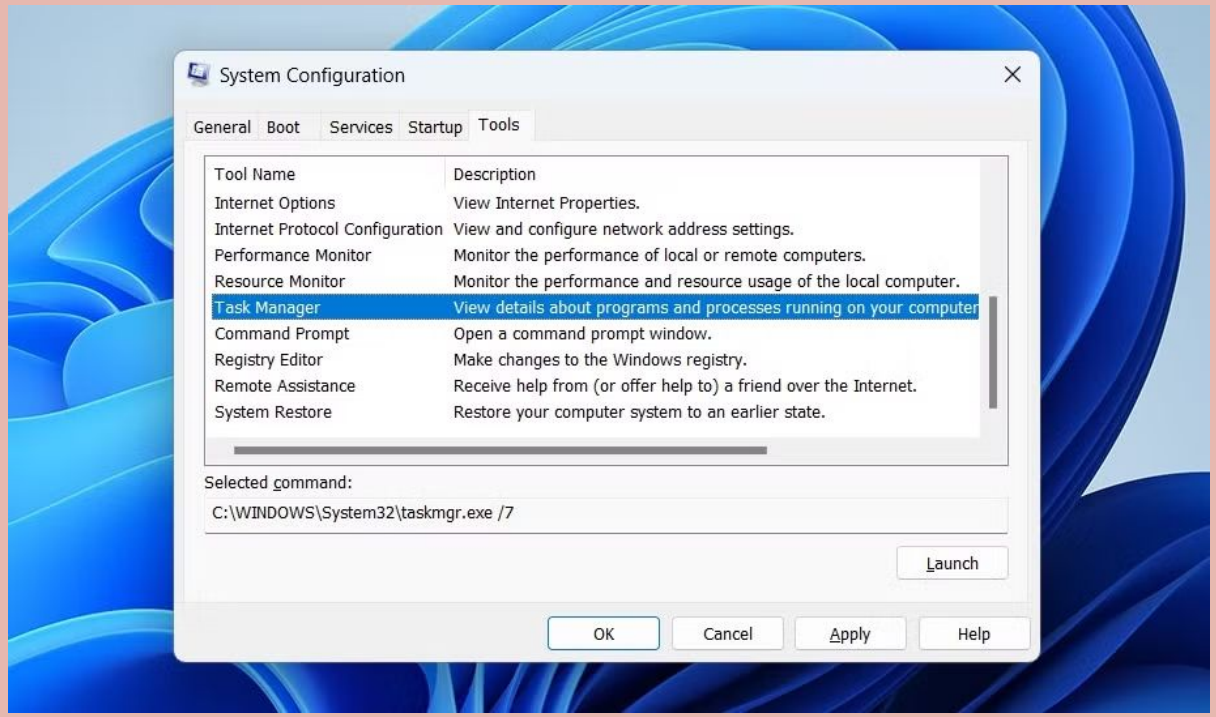
Scope and Objectives

Focus: Hardening Windows Server 2022

- Key Areas: System Configuration, User Management, Access Control, Patch Management, Network Security
- Objective: To demonstrate practical steps for hardening the chosen OS in a virtualized environment.
- Deliverable: A "How-To" guide documenting the hardening process and assessment results.

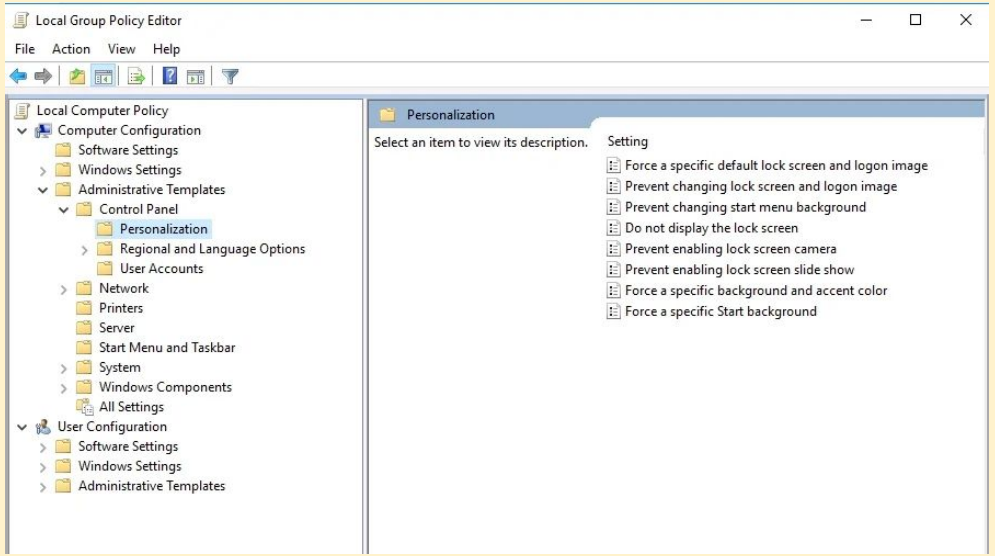
System Configuration

- Minimize Installed Software: Remove unnecessary applications and features.
- Disable Unnecessary Services: Stop and disable unused services.
- Secure BIOS/UEFI: Enable Secure Boot, set passwords, disable booting from removable media.
- Regularly Review System Logs: Monitor logs for suspicious activity.
- Implement a Baseline Configuration: Create a standardized secure template.



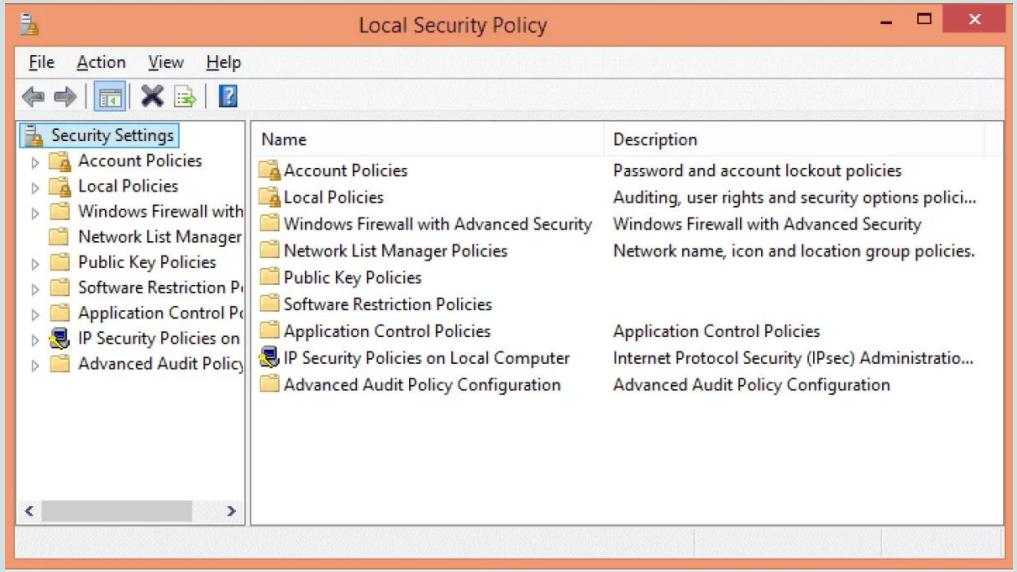
User Management

- Enforce Strong Passwords: Implement complexity requirements and regular changes.
- Principle of Least Privilege: Grant only necessary permissions.
- Disable Default/Guest Accounts: Rename or disable default administrator accounts.
- Regularly Review User Accounts: Remove inactive or unnecessary accounts.
- Implement Multi-Factor Authentication (MFA): Add an extra layer of security.



Access Control

- File System Permissions: Configure granular permissions for files and directories.
- Access Control Lists (ACLs): Implement for fine-grained access control.
- Network Access Control: Use firewalls and other tools to restrict access.
- Principle of Least Connection: Only allow necessary network connections.
- Intrusion Detection/Prevention System (IDS/IPS): Monitor network traffic.

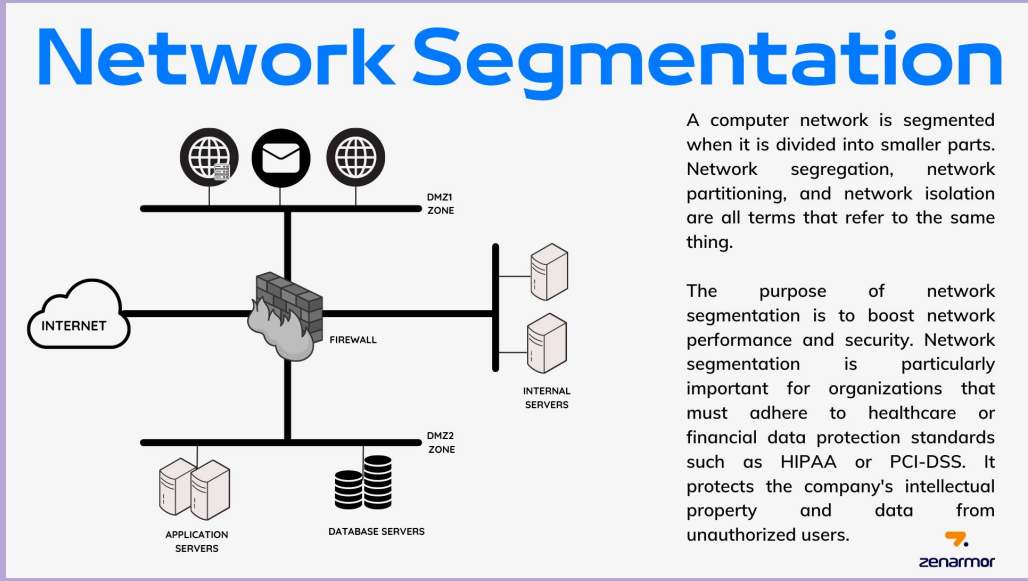


Patch Management

- Establish a Patching Schedule: Implement a regular schedule for updates.
- Automate Patching: Use automated tools to streamline the process.
- Test Patches: Test in a non-production environment before deployment.
- Stay Informed: Subscribe to security advisories and vulnerability databases.
- Vulnerability Scanning: Regularly scan for vulnerabilities.

Network Security

- Firewall Configuration: Restrict inbound and outbound traffic.
- Network Segmentation: Isolate critical systems.
- Disable Unnecessary Network Protocols: Reduce attack vectors.
- Secure Remote Access: Use SSH or VPN.
- DNS Security: Implement DNSSEC.



Configuring iptables Firewall (Example)

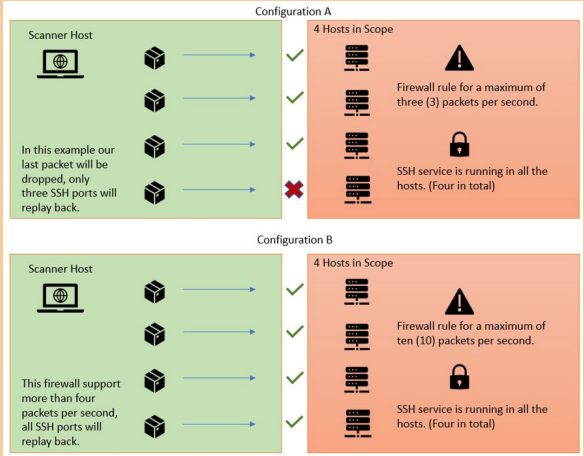
Initial State:

- Defining Rules:
- Commands used : `iptables -A INPUT -p tcp --dport 22 -j ACCEPT`, etc. (Explain each)
- Saving and Applying:
- Commands: `iptables-save > /etc/sysconfig/iptables`, etc.
- Verification: Commands: `iptables -L -nv`

```
[root@securitytrails-centos ~]# nmap -p 1-65535 localhost

Starting Nmap 6.40 ( http://nmap.org ) at 2019-12-03 19:16 UTC
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000040s latency).
Other addresses for localhost (not scanned): 127.0.0.1
rDNS record for 127.0.0.1: securitytrails-centos
Not shown: 65532 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
25/tcp  open  smtp
111/tcp open  rpcbind

Nmap done: 1 IP address (1 host up) scanned in 0.86 seconds
[root@securitytrails-centos ~]#
```



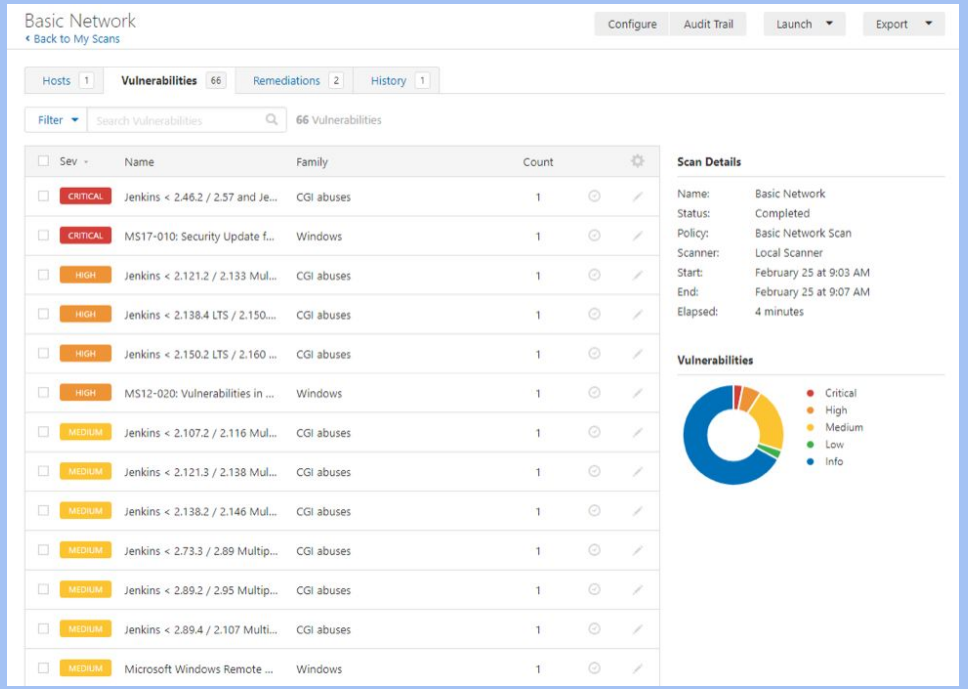
Assessment and Testing

Tools and Techniques:

- Nmap (Network Mapper): While primarily a network scanning tool, Nmap can also be used for basic OS vulnerability scanning. It can identify open ports, running services, and even attempt to determine the OS version.
- OpenVAS (Open Vulnerability Assessment System): A full-featured vulnerability scanner that performs comprehensive checks for known vulnerabilities in operating systems and applications.

Findings:

- No vulnerabilities were found during this project.



Conclusion

- OS hardening is essential for protecting against cyber threats.
- Key steps include system configuration, user management, access control, patch management, and network security.
- Regular security assessments and updates are crucial.

Areas	Core	Resources	Services	Environment
System Hardening	Boot Process Containers Frameworks **Kernel** Service Manager Virtualization	Accounting **Authentication** Cryptography Logging Network Software Storage Time	Database Mail Middleware Monitoring Printing **Shell** **Web**	Forensics Incident Response Malware Risks Security Monitoring System Integrity
Security Auditing				
Compliance				