🛠️ Project: Linux Hardening Script with UFW and System Config

🔧 Phase 1: Key Hardening Components

Here's what we'll automate via Bash or Python (most use Bash unless noted):

Area	**Action**	**Command/Tool**
🔒 Firewall	Uncomplicated FW setup & rule management	sudo ufw
🔑 SSH Security	Disable root login, change port, protocol config	/etc/ssh/sshd_config
👥 User Management	Ensure strong password policy, disable guest	passwd, login.defs
🔐 File Permissions	Set sticky bits, critical file ownership	chmod, chown
📋 Logging & Auditing	Install & configure auditd	auditctl, ausearch
🛡️ Services	Disable unnecessary services (e.g., FTP, Telnet)	systemctl, chkconfig

```
linux-hardening-script/
├── scripts/
│   ├── ufw_setup.sh
│   ├── ssh_hardening.sh
│   ├── user_security.sh
│   ├── audit_config.sh
├── screenshots/
│   └── (Code & Outputs)
├── README.md
└── hardening_report_template.md
```

🔐 UFW Hardening Script

🎯 Goals:

- Deny all incoming connections by default
- Allow all outgoing traffic (can be locked down later)
- Allow SSH access (default + custom port)
- Allow specific trusted IPs
- Deny certain ports/IPs explicitly
- Enable firewall with verbose status output

🛡️ Security Notes:

- **Default deny (incoming)** ensures only whitelisted services are accessible.
- **Allowing SSH (22/2222)** gives you remote access while supporting custom ports.
- **IP-specific rules** let you control access granularity (e.g., admins only).
- **Blocking port 25 outbound** is a basic anti-spam measure.
- **IPv6 enabled** to ensure complete firewall coverage.

Check Rules: `sudo ufw status verbose`

RUN: `sudo bash ufw_setup.sh`

🔐 SSH Hardening Script

🎯 Goals:

- Disable root login
- Change default SSH port
- Enforce protocol version
- Limit authentication attempts
- Enable public key auth (optional)
- Restart SSH to apply changes

🛡️ Security Notes:

- **Changing the port (2222)** reduces automated attack noise.
- **Disabling root login** enforces privilege separation.
- **Enforcing Protocol 2** eliminates use of insecure protocol version 1.
- **Auth limits** reduce brute-force exposure.
- **Public key auth** is ideal (commented for safety until keys are in place).

✅ Next Steps:

1. Run it: `sudo bash ssh_hardening.sh`
2. Test SSH with ssh username@ip -p 2222 from another system
3. Key-based authentication can be enabled after initial SSH access is verified.

👥 User Security & Account Policy Script

🎯 Goals:

- Enforce strong password rules
- Lock inactive accounts

- Disable guest login
- List users with UID > 1000 (normal users)
- (Optional) Check for users with empty passwords

📌 Key Actions Covered:

Task	Why It Matters
Enforce password policy	Prevent easy brute force access
Disable guest login	Guests can be used to escalate privilege
Lock inactive accounts	Reduces risk from dormant accounts
List users	Helps with audit and privilege tracking
Check empty passwords	Flags dangerous misconfigs

RUN: sudo bash user_security.sh

🔐 File Permissions & Ownership Hardening Script

🎯 Goals:

- Set correct permissions for /etc/passwd, /etc/shadow, /etc/group, /etc/gshadow
- Apply sticky bit to /tmp

  (sticky bit: special permission flag: restricts deletion or renaming of files inside the directory except by the file/directory owner. This prevents tampering in shared folders like /tmp)
- Locks down /tmp and user home directories.
- Search for files with dangerous SUID/SGID/777 perms

📌 Key Actions Covered:

File	Permission	Reason
/etc/shadow	640	Restrict access to root & shadow group
/tmp	1777	Prevent users from deleting others' files
/home	750	Disallow world read of user files
find ... -0777	Audit world-writable files	

RUN: sudo bash file_permissions.sh

📋 Audit Logging Script with auditd

🎯 Goals:

- Install and start auditd
- Monitor critical files like /etc/passwd, /etc/shadow
- Log privilege escalation (e.g., use of sudo)
- Enable auditing for user logins and binary executions
- Make rules persistent across reboots

📌 Key Monitored Events:

Monitored Item	Rule	Purpose
/etc/passwd, /etc/shadow	-p wa	Detect writes/appends to user account data
sudo binary	-p x	Track execution of privilege escalation
Login logs	-p wa	Capture success/fail login attempts
/bin, /usr/bin	-p x	Log execution of binaries (watch for reverse shells, etc.)

RUN: sudo bash audit_config.sh

Monitor Logs:

```
# View all audit logs
sudo less /var/log/audit/audit.log


# Search logs with a specific key (e.g., passwd_changes)
sudo ausearch -k passwd_changes


# Summarize report of recent events
sudo aureport -x
```

📄 Security Report

Each script has a corresponding section in hardening_report_template.md for audit purposes and documentation.

🧠 Useful Log Locations

Log Source	Path	View Command
General logs	/var/log/syslog	less /var/log/syslog
SSH logs (Ubuntu)	/var/log/auth.log	sudo less /var/log/auth.log

Log Source	Path	View Command
Auditd logs	/var/log/audit/audit.log	sudo ausearch -k [keyword] or sudo aureport

---

📚 References

- CIS Ubuntu Benchmarks
- UFW Documentation
- man auditctl, man audit.rules, man ufw