# LINUX / SERVICES PLAYBOOK

## CCDC Linux Defense Guide

## FIRST 15 MINUTES

Execute in order. No exceptions.

### Minute 0-5: Credential Hygiene

```
# Change root password
echo "root:$(openssl rand -base64 24)" | chpasswd

# Or set specific password
passwd root

# List all users with shell access
cat /etc/passwd | grep -v nologin | grep -v false | grep -v sync

# Change passwords for all active users
for user in $(cat /etc/passwd | grep -E '/bin/(ba)?sh' | cut -d: -f1); do
    echo "Changing password for $user"
    echo "$user:NewSecureP@ss2024!" | chpasswd
done
```

### Minute 5-10: Lock Dangerous Access

```
# Lock unnecessary accounts
passwd -l nobody
passwd -l www-data
passwd -l mail

# Check for UID 0 accounts (should only be root)
awk -F: '$3 == 0 {print $1}' /etc/passwd

# Check for empty passwords
awk -F: '$2 == "" {print $1}' /etc/shadow

# Disable direct root login via SSH
sed -i 's/PermitRootLogin yes/PermitRootLogin no/' /etc/ssh/sshd_config
systemctl restart sshd
```

### Minute 10-15: Verify Services

```
# List running services
systemctl list-units --type=service --state=running

# Check web server
curl -I http://localhost/
```

```
# Check if service responds
nc -zv localhost 80
nc -zv localhost 443
nc -zv localhost 22
```

# SSH HARDENING CHECKLIST

- ■ Disable root login
- ■ Disable password authentication
- ■ Change default port (optional)
- ■ Limit allowed users
- ■ Enable key-based auth only
- ■ Set up fail2ban or equivalent

## SSH Config Hardening

```
# Edit /etc/ssh/sshd_config
cat >> /etc/ssh/sshd_config << 'EOF'
# Security Hardening
PermitRootLogin no
PasswordAuthentication no
PubkeyAuthentication yes
PermitEmptyPasswords no
X11Forwarding no
MaxAuthTries 3
AllowUsers admin operator  # Only allow specific users
Protocol 2
EOF

# Restart SSH
systemctl restart sshd
```

## Generate SSH Keys

```
# On admin machine
ssh-keygen -t ed25519 -C "ccdc-admin"

# Copy to server
ssh-copy-id -i ~/.ssh/id_ed25519.pub user@server

# Verify key-only auth works BEFORE disabling passwords
```

# IPTABLES FIREWALL

## Quick Lockdown

```
# Backup current rules
iptables-save > /backup/iptables-$(date +%Y%m%d).rules

# Flush existing rules
iptables -F
iptables -X
```

```
# Set default policies
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT

# Allow loopback
iptables -A INPUT -i lo -j ACCEPT

# Allow established connections
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# Allow SSH (adjust port if changed)
iptables -A INPUT -p tcp --dport 22 -j ACCEPT

# Allow HTTP/HTTPS
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -p tcp --dport 443 -j ACCEPT
```

### Block Specific IP

```
# Block attacker
iptables -I INPUT -s 10.10.10.50 -j DROP

# Block range
iptables -I INPUT -s 10.10.10.0/24 -j DROP

# View rules
iptables -L -n -v
```

### UFW Alternative (Simpler)

```
# Enable UFW
ufw enable
ufw default deny incoming
ufw default allow outgoing

# Allow services
ufw allow 22/tcp
ufw allow 80/tcp
ufw allow 443/tcp

# Block IP
ufw deny from 10.10.10.50

# Status
ufw status verbose
```

## SERVICE HARDENING

### Apache/Nginx

```
# Disable server signature
# Apache: /etc/apache2/apache2.conf
ServerTokens Prod
ServerSignature Off
```

```
# Nginx: /etc/nginx/nginx.conf
server_tokens off;

# Restart
systemctl restart apache2  # or nginx
```

### MySQL/MariaDB

```
# Run security script
mysql_secure_installation

# Manual hardening
mysql -u root -p << 'EOF'
DELETE FROM mysql.user WHERE User='';
DELETE FROM mysql.user WHERE User='root' AND Host NOT IN ('localhost', '127.0.0.1', '::1');
DROP DATABASE IF EXISTS test;
DELETE FROM mysql.db WHERE Db='test' OR Db='test\\_%';
FLUSH PRIVILEGES;
EOF

# Change root password
mysqladmin -u root password 'NewSecureP@ss2024!'
```

### PostgreSQL

```
# Edit pg_hba.conf - change 'trust' to 'md5'
sed -i 's/trust/md5/g' /etc/postgresql/*/main/pg_hba.conf

# Change postgres password
sudo -u postgres psql -c "ALTER USER postgres PASSWORD 'NewSecureP@ss2024!';"

systemctl restart postgresql
```

## LOG LOCATIONS

| Service | Log Location |
|---|---|
| System | /var/log/syslog or /var/log/messages |
| Auth | /var/log/auth.log or /var/log/secure |
| Apache | /var/log/apache2/ |
| Nginx | /var/log/nginx/ |
| MySQL | /var/log/mysql/ |
| SSH | /var/log/auth.log |
| Cron | /var/log/cron |

### Quick Log Analysis

```
# Failed SSH logins
grep "Failed password" /var/log/auth.log | tail -50

# Successful logins
```

```
grep "Accepted" /var/log/auth.log | tail -20

# Sudo usage
grep "sudo" /var/log/auth.log | tail -30

# Web server errors
grep -E "error|404|500" /var/log/apache2/error.log | tail -50

# Look for web shells
grep -rE "eval|exec|system|passthru|shell_exec" /var/www/ 2>/dev/null
```

# PERSISTENCE HUNTING

## Check Cron Jobs

```
# System cron
cat /etc/crontab
ls -la /etc/cron.*/

# User crons
for user in $(cut -f1 -d: /etc/passwd); do
    crontab -u $user -l 2>/dev/null
done

# Look for suspicious cron
grep -r "wget\|curl\|nc\|bash -i" /etc/cron* /var/spool/cron/ 2>/dev/null
```

## Check Startup Scripts

```
# Systemd services
systemctl list-unit-files --type=service | grep enabled

# Init scripts
ls -la /etc/init.d/
ls -la /etc/rc.local

# Check for modifications
ls -la /etc/systemd/system/
```

## Check for Backdoors

```
# SUID binaries (privilege escalation)
find / -perm -4000 -type f 2>/dev/null

# World-writable files in sensitive locations
find /etc /usr -type f -perm -002 2>/dev/null

# Recently modified files
find / -mmin -60 -type f 2>/dev/null | grep -v proc | head -50

# Hidden files in home directories
find /home -name ".*" -type f 2>/dev/null

# Check for reverse shells in bash history
cat /home/*/.bash_history 2>/dev/null | grep -E "nc|netcat|bash -i|/dev/tcp"
```

```
# Check authorized_keys for unauthorized keys
find /home -name "authorized_keys" -exec cat {} \;
cat /root/.ssh/authorized_keys
```

### Check Running Processes

```
# All processes with full command
ps auxf

# Network connections
netstat -antup
ss -antup

# Look for suspicious processes
ps aux | grep -E "nc|netcat|perl|python|ruby|php" | grep -v grep

# Check for processes running as root
ps aux | awk '$1=="root" {print}'
```

## QUICK RECOVERY

### Service Won't Start

```
# Check status and errors
systemctl status <service>
journalctl -u <service> -n 50

# Check config syntax
nginx -t
apache2ctl configtest
named-checkconf

# Try restart
systemctl restart <service>
```

### Restore Config from Backup

```
# Always backup before changes!
cp /etc/nginx/nginx.conf /backup/nginx.conf.$(date +%Y%m%d)

# Restore
cp /backup/nginx.conf.original /etc/nginx/nginx.conf
systemctl restart nginx
```

### File Permissions Reset

```
# Web directory
chown -R www-data:www-data /var/www/html
chmod -R 755 /var/www/html
chmod -R 644 /var/www/html/*.php

# SSH
chmod 700 ~/.ssh
chmod 600 ~/.ssh/authorized_keys
chmod 600 ~/.ssh/id_*
```

```
chmod 644 ~/.ssh/*.pub
```

# CONFIG BASELINES

## Backup All Configs

```
mkdir -p /backup/$(date +%Y%m%d)
cp -rp /etc/ssh /backup/$(date +%Y%m%d)/
cp -rp /etc/nginx /backup/$(date +%Y%m%d)/ 2>/dev/null
cp -rp /etc/apache2 /backup/$(date +%Y%m%d)/ 2>/dev/null
cp -rp /etc/mysql /backup/$(date +%Y%m%d)/ 2>/dev/null
cp /etc/passwd /etc/shadow /etc/group /backup/$(date +%Y%m%d)/
iptables-save > /backup/$(date +%Y%m%d)/iptables.rules
```

# LINUX DEFENDER MANTRA

> *""Change passwords."*
>
> *"Lock SSH."*
>
> *"Block by default."*
>
> *"Then harden services.""*

*CCDC.x1000.ai - Championship Training*