Security Testing Commands Explained

FAIL2BAN MANUAL CHECKS

Command 1: Basic Installation Check

```bash
systemctl list-unit-files | grep fail2ban
dpkg -l | grep fail2ban
```

What it does:

- `systemctl list-unit-files | grep fail2ban`: Lists all system services and filters for fail2ban
- `dpkg -l | grep fail2ban`: Lists all installed packages and filters for fail2ban

What to expect:

- **Before installation**: No output (fail2ban not found)
- **After installation**: Shows fail2ban service file and package details

Command 2: Comprehensive Status Check

```bash
echo "FAIL2BAN STATUS:"; systemctl is-active fail2ban; systemctl is-enabled fail2ban; echo "SSH
```

Breaking it down:

- `systemctl is-active fail2ban`: Checks if service is running → "active" or "inactive"
- `systemctl is-enabled fail2ban`: Checks if service starts at boot → "enabled" or "disabled"
- `cat /etc/fail2ban/jail.d/ssh.conf`: Shows SSH jail configuration file contents
- `fail2ban-client status`: Shows overall fail2ban status and active jails
- `fail2ban-client status ssh`: Shows detailed SSH jail statistics

Key differences in output:

- **Before**: All services inactive, no config files, client not responding
- **After**: Services active/enabled, config file shows protection rules, client shows jail statistics

SSH SECURITY CHECKS

Command: SSH Configuration and File Permissions

```bash
echo "SSH CONFIG:"; grep -E "^(MaxAuthTries|ClientAliveInterval|ClientAliveCountMax|LoginGraceT
```

Breaking it down:

- `grep -E "^(MaxAuthTries|...)"`: Searches for active SSH security settings (lines not starting with #)
- `ls -l /etc/passwd /etc/shadow /etc/group /etc/ssh/sshd_config`: Shows file permissions for critical system files
- `cat /etc/issue.net`: Shows login banner content

Understanding file permissions:

- `-rw-r--r--` (644): Owner can read/write, others can only read
- `-rw-r-----` (640): Owner can read/write, group can read, others no access
- `-rw-------` (600): Only owner can read/write (most secure)

Key security changes:

- **Before**: No SSH security settings visible, sshd_config is 644 (readable by all), no banner
- **After**: Security settings active, sshd_config is 600 (root only), warning banner present

NETWORK SECURITY CHECKS

Command: Network, Password, and Firewall Status

```bash
echo "NETWORK:"; sysctl net.ipv4.ip_forward net.ipv4.tcp_syncookies net.ipv4.conf.all.accept_re
```

Breaking it down:

Network Security (sysctl values):

- `net.ipv4.ip_forward`: Controls packet forwarding (router functionality)
  - 0 = disabled (secure), 1 = enabled (allows routing)
- `net.ipv4.tcp_syncookies`: SYN flood protection

    - 1 = enabled (protects against DDoS)
- `net.ipv4.conf.all.accept_redirects`: ICMP redirect acceptance
    - 0 = disabled (secure), 1 = enabled (vulnerable to routing attacks)
- `net.ipv4.icmp_echo_ignore_broadcasts`: Ping broadcast response
    - 1 = ignored (secure), 0 = responds (DDoS amplification risk)

Password Policy:

- `PASS_MIN_DAYS`: Minimum days between password changes
- `PASS_MAX_DAYS`: Maximum days before password expires
- `PASS_MIN_LEN`: Minimum password length
- `PASS_WARN_AGE`: Days of warning before password expires

Firewall Status:

- `ufw status`: Shows if firewall is active and what rules are applied
- **Before**: "Status: inactive" (no protection)
- **After**: "Status: active" with SSH access rules

Custom Files:

- `/etc/sysctl.d/99-security.conf`: Custom network security settings
- `/etc/logrotate.d/security-logs`: Log rotation configuration
- **Before**: Files don't exist
- **After**: Files present with security configurations

Security Impact Summary

What These Outputs Tell You:

FAIL2BAN:

- Shows if automated intrusion detection is working
- "Currently failed/banned" numbers indicate active threats being blocked

SSH SECURITY:

- File permission changes prevent unauthorized access to critical files
- SSH settings limit brute force attack success

- Banner warns potential attackers they're being monitored

NETWORK SECURITY:

- Network settings prevent common network-based attacks
- Password policy enforces stronger authentication
- Firewall creates network barrier
- Custom files show hardening configurations are persistent

Real-World Meaning:

1. **Before Hardening**: System has default settings optimized for functionality, not security
2. **After Hardening**: System has multiple layers of protection against common attack vectors

The commands act as a **security audit checklist -** they verify that each protection layer is actually implemented and functioning correctly, not just installed.