Top 10 Easy Cybersecurity Wins for Linux Environments

By: Michael “Sleventyeleven” Contino

Overview

- Whoami
- Go over each “Win”
  - Discuss the Idea
  - Discuss the Threat
  - Discuss the Implementation
- Other General Advice
- Questions

Whoami

- Security Engineer @ Groupon
- Run Global Vulnerability Management
- Father of two boys (6 months and 12 years old)
- High school and college cybersecurity mentor
- Holder of way too many certifications to list
- Avid CTF and Bug Bounty pursuer

Humble Moment

- I don’t claim to be any sort of Linux Ninja Master
- I simply just represent my personal experience and struggle to do better
- Not all these ideas will work or be “easy” in all environments
- An attempt has been made to make examples agnostic

#1 Disable IPV6

- Vast majority of organizations don’t route IPv6 internally
- Most commonly used routers and switches don’t fully support IPv6
- IPv6 has more inherent complexity, with more features and addresses built-in

#1 Disable IPV6: Threat

- Additional Application/Service Attack Surface
- Neighbor Discovery Denial of Service
- Possibly the Slaac attack
- Many Firewalls and IPS/IDS don’t fully support IPV6 yet

#1 Disable IPV6: Implement

```
visudo /etc/sysctl.conf

net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1

Or

sudo echo “options ipv6 disable=1” >> /etc/modprobe.d/disable-ipv6.conf
```

#2 Deny Permissions on /tmp

- No legitimate program should be executing programs in /tmp
- /tmp is only intended for temporary (avg <2 secs) write operations
- Limited to binary executables and access to shared objects (libraries)

#2 Deny Permissions on /tmp: Threat

- The first thing thing most attackers do is write tools to /tmp and try elevate access
- /tmp and/or /var/tmp are the only default world writable directories in FHS
- There's no good way to deny the ability to have text files be interpreted

#2 Deny Permissions on /tmp: Implement

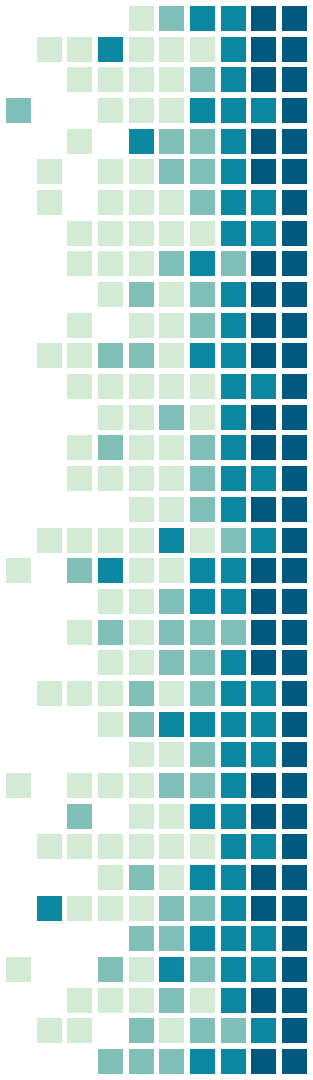
```
visudo /etc/fstab

/dev/sda5   /tmp   ext3 defaults,nosuid,nodev,noexec
1 2
```

Or with systemd

```
visudo /etc/systemd/system/tmp.mount

Options=mode=1777,rw,nosuid,nodev,noexec,relatime
```



#3 PAM Failure Lockouts

- By default PAM has no built-in way to deal with authentication failures
- Without central limitations credentials can be guessed across multiple services
- Correlating logs for login failures can be time consuming and difficult
- IP block solutions (fail2ban) are limited to authentication attempts over the network

#3 PAM Failure Lockouts: Threat

- Attackers often guess common, weak, and default credentials on targeted systems
- Once a credential is compromised, attackers will attempt to reuse them across different systems and across multiple accounts
- More sophisticated attackers may try to password guess across multiple services, applications, or systems

#3 PAM Failure Lockouts: Implement

```
visudo /etc/pam.d/password-auth

auth required pam_tally2.so file=/var/log/tallylog deny=3 even_deny_root
unlock_time=1200
account required pam_tally2.so
OR
auth required pam_tally2.so onerr=fail audit silent deny=5 unlock_time=900
auth required pam_faillock.so preauth audit silent deny=5 unlock_time=900
auth sufficient pam_unix.soauth [default=die] pam_faillock.so authfail audit
deny=5 unlock_time=900
auth sufficient pam_faillock.so authsucc audit deny=5 unlock_time=900
```

#4 ExecShield and Kernel Protection

- Today's linux kernel has a lot of the modern protections baked-in that aren't enabled in most distributions
- These protections cover base memory usage, system calls, page permissions, and much more
- Simply enabling these protections can render many publicly available exploit code ineffective

#4 ExecShield and Kernel Protection: Threat

- Attackers leverage a multitude of memory usage exploits like buffer overflows, heap sprays, and page rewrites
- Attackers also take advantage of debugging options, unprotected libraries, and system calls
- Attackers will take advantage of simplified networking implementations in order to conduct a MitM attack or simply to exfiltrate data

#4 ExecShield and Kernel Protection: Implement

```
visudo /etc/sysctl.conf

kernel.exec-shield=1
kernel.randomize_va_space=1
net.ipv4.conf.all.rp_filter=1
net.ipv4.conf.all.accept_source_route=0
net.ipv4.icmp_echo_ignore_broadcasts=1
net.ipv4.icmp_ignore_bogus_error_messages=1
net.ipv4.conf.all.log_martians = 1

https://klaver.it/linux/sysctl.conf
```

#5 Password Policies

- Credentials are rarely changed, unless a policy enforces it
- Its human nature to choose simple, easily remembered (and guessable) passwords
- Password are more likely to be shared

#5 Password Policies: Threat

- Attackers often want to compromise and leverage credentials to spread laterally through the network.
- Credentials create an easy opportunity for attackers to maintain access with a lower chance of detection.

#5 Password Policies: Implement

```
visudo /etc/pam.d/common-password

password required pam_cracklib.so try_first_pass retry=3 minlen=14 dcredit=-1 ucredit=-1
ocredit=-1 lcredit=-1

password sufficient pam_unix.so remember=5 sha512


visudo /etc/login.defs


PASS_MAX_DAYS 90
```

#6 Auditd to Watch Core Directories

- You need to watch the crown jewels
- Alerts can be generated based on logged events
- In the event the gems are accessed, a picture of true exposure is needed

#6 Auditd to Watch Core Directories: Threat

- A sophisticated attacker's end goal is going to be to access the crown jewels. This often means examining and potentially manipulating application code/directories.
- A less sophisticated attacker may instead just end up taking the service down.

#6 Auditd to Watch Core Directories: Implement

```
visudo /etc/audit/audit.rules

-w /etc/secret_directory -p rwxa -k track-secrets
```

#7 Disable Removable Devices/Filesystems

- Removal devices aren't normally needed, especially for virtualized systems
- Removal device control operates in user-space, but are allocated as system level resource
- Most filesystem types are not widely used, leaving them under-developed or unsupported entirely.

#7 Disable Removable Devices/Filesystems: Threat

- Several vulnerabilities have come about recently leveraging user-space filesystem mounts
- Removable devices can be used to exfiltrate data
- Removable devices can be leveraged as rogue or malicious devices

#7 Disable Removable Devices/Filesystems: Implement

```
sudo echo 'install usb-storage /bin/true' >> /etc/modprobe.d/disable-usb-storage.conf

sudo echo 'install jffs2 /bin/true' >> /etc/modprobe.d/disable-jffs2.conf

sudo echo 'install cramsfs /bin/true' >> /etc/modprobe.d/disable-cramsfs.conf


sudo echo "blacklist firewire-core" >> /etc/modprobe.d/firewire.conf
sudo echo "blacklist thunderbolt" >> /etc/modprobe.d/thunderbolt.conf
```

#8 Disable Unused SSH Features

- Even with no shell available, most ssh features still have limited functionality.
- The vast majority of ssh features go unused
- Features can be blanketly disabled and allowed for specific groups or users instead.

#8 Disable Unused SSH Features: Threat

- Attackers will leverage x-forwarding and port-forwarding to move around the network, even if shell or command restrictions are in place.
- A multitude of vulnerabilities have come about in recent years threatening the integrity of ssh connections based on issues with various features.

#8 Disable Unused SSH Features: Implement

```
visudo /etc/ssh/sshd_config

X11Forwarding no

IgnoreRhosts yes

HostbasedAuthentication no

ForwardAgent        no

AllowTcpForwarding no

AllowTcpForwardingForUsers
AllowTcpForwardingForGroups
```

#9 Chattr All The Things

- Protect file from deletion and manipulation
- Files can maintain non-root ownership and still be protected
- Only the superuser or a process possessing the CAP_LINUX_IMMUTABLE capability can set or clear these attributes
- Blocks log scrubbing attempts by only allowing append operations

#9 Chattr All The Things: Threat

- Stop attackers from uploading a trojan and/or web shell
- If all else fails the attacker may want to disrupt the service
- Stops attackers from scrubbing logs of their activity

#9 Chattr All The Things: Implement

```
sudo chattr -R +i webapplication

sudo chattr +a webapplication/web.log



sudo lsattr webapplication/web.log



sudo chattr -R -i webapplication

sudo chattr -a webapplication/web.log
```

#10 Restrict access to cron/at

- Normal users don't need access to cron/at
- The superuser can just use "sudo -u" if there is a true need
- Most services should be handled by sysvinit, systemd, upstart, etc
- Also makes tracking and logging these events harder

#10 Restrict access to cron/at: Threat

- Attackers often try to maintain access using built-in tools available to all users
- Attackers also know chances of detection on a standard users is lower

#10 Restrict access to cron/at: Implement

```
sudo rm /etc/at.deny

sudo echo “root” >> /etc/at.allow


sudo rm /etc/cron.deny

sudo echo “root” >> /etc/cron.allow
```

Other General Advice

- Don’t use a default $PATH that contains “.” or an empty string
- Avoid using wildcards in sudo rules
- Avoid giving sudo access to tool suites, editors, or shells
- Restrict available developer tools
- Restrict service accounts access with command restriction via the ssh authorized_keys file or sshd configuration file.
- Use passphrases on SSH keys!
- Use iptables for local ingress, egress rules, and dos protection

Questions?

Twitter:
@sleventyeleven

Site:
Hackersvanguard.com