Linux Control Scenario – CIS Hardening Procedures

For this exercise a fresh VM of SLES 15 SP7 was created. Recommendations based on the CIS Benchmarks Guide included in the Git Repo. Assumptions are based on a Level 1 server profile. The focus was on establishing a simple baseline level of security—common across most Linux systems—since covering every CIS control in detail could easily turn this into a 100-page document.

Note: Please excuse the use of the root account in the screenshots below. While best practice is to run commands as a regular user with sudo privileges, root was used here simply to save time by avoiding repeated password prompts.

```
ryan@protera:~> cat /etc/os-release
NAME="SLES"
VERSION="15-SP7"
VERSION_ID="15.7"
PRETTY_NAME="SUSE Linux Enterprise Server 15 SP7"
ID="sles"
ID_LIKE="suse"
ANSI_COLOR="0;32"
CPE_NAME="cpe:/o:suse:sles:15:sp7"
DOCUMENTATION_URL="https://documentation.suse.com/"
ryan@protera:~>
```

1. Initial Hardening Steps

- **User & Access Control**
  Edit /etc/ssh/sshd_config file to disable root login and password login
  Make sure PubKeyAuthentication is set to yes (should be by default)
  Reload the config > systemctl reload sshd:

```
# Authentication:

#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile      .ssh/authorized_keys

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
#PermitEmptyPasswords no
```

Fresh installs allow anybody with the root password to use sudo access. Best practice is to change this in the /etc/sudoers file to remove that and add any user that needs sudo access to the wheel group.

Before:

```
## In the default (unconfigured) configuration, sudo asks for the root password.
## This allows use of an ordinary user account for administration of a freshly
## installed system.
Defaults targetpw   # ask for the password of the target user i.e. root
ALL   ALL=(ALL) ALL   # WARNING! Only use this together with 'Defaults targetpw'!

##
## Runas alias specification
##

##
## User privilege specification
##
root ALL=(ALL:ALL) ALL

## Uncomment to allow members of group wheel to execute any command
# %wheel ALL=(ALL:ALL) ALL

## Same thing without a password
# %wheel ALL=(ALL:ALL) NOPASSWD: ALL
```

After:

```
## In the default (unconfigured) configuration, sudo asks for the root password.
## This allows use of an ordinary user account for administration of a freshly
## installed system.
#Defaults targetpw   # ask for the password of the target user i.e. root

##
## Runas alias specification
##

##
## User privilege specification
##
root ALL=(ALL:ALL) ALL

## Uncomment to allow members of group wheel to execute any command
%wheel ALL=(ALL:ALL) ALL
```

Then add user to wheel group:

```
protera:~ # usermod -aG wheel ryan
protera:~ # sudo -l -U ryan
Matching Defaults entries for ryan on protera:
    always_set_home, env_reset, env_keep="LANG LC_ADDRESS LC_CTYPE
    XDG_SESSION_COOKIE", !insults, secure_path=/usr/sbin\:/usr/bin\

User ryan may run the following commands on protera:
    (ALL : ALL) ALL
```

Password Policy

Password expiration is recommended at 365 days or less and can be set in /etc/login.defs:

```
protera:~ # cat /etc/login.defs |grep PASS_
#       PASS_MAX_DAYS   Maximum number of days a password may be used.
#       PASS_MIN_DAYS   Minimum number of days allowed between password changes.
#       PASS_WARN_AGE   Number of days warning given before a password expires.
PASS_MAX_DAYS   90
PASS_MIN_DAYS   0
PASS_WARN_AGE   7
```

UMASK

Additionally, for added security, it is recommended the umask in /etc/login.defs be changed from the default 022 to 027 which would provide a good balance between security and accessibility:

```
# Default initial "umask" value used by login(1) on non-PAM enabled systems.
# Default "umask" value for pam_umask(8) on PAM enabled systems.
# UMASK is also used by useradd(8) and newusers(8) to set the mode for new
# home directories if HOME_MODE is not set.
# 022 is the default value, but 027, or even 077, could be considered
# for increased privacy. There is no One True Answer here: each sysadmin
# must make up their mind.
UMASK           027
```

- **Patching**

Refresh the repos and run latest updates:

```
protera:~ # zypper refresh && zypper update -y
Repository 'SLE-Module-Basesystem15-SP7-Updates' is up to date.
Repository 'SLE-Module-Python3-15-SP7-Updates' is up to date.
Repository 'SLE-Product-SLES15-SP7-Updates' is up to date.
Repository 'SLE-Module-Server-Applications15-SP7-Updates' is up to date.
Repository 'SLE-Module-Systems-Management-15-SP7-Updates' is up to date.
Repository 'SLE-Module-Basesystem15-SP7-Pool' is up to date.
Repository 'SLE-Module-Python3-15-SP7-Pool' is up to date.
Repository 'SLE-Product-SLES15-SP7-Pool' is up to date.
Repository 'SLE-Module-Server-Applications15-SP7-Pool' is up to date.
Repository 'SLE-Module-Systems-Management-15-SP7-Pool' is up to date.
All repositories have been refreshed.
Refreshing service 'Basesystem_Module_15_SP7_x86_64'.
Refreshing service 'Python_3_Module_15_SP7_x86_64'.
Refreshing service 'SUSE_Linux_Enterprise_Server_15_SP7_x86_64'.
Refreshing service 'Server_Applications_Module_15_SP7_x86_64'.
Refreshing service 'Systems_Management_Module_15_SP7_x86_64'.
Loading repository data...
Reading installed packages...
Nothing to do.
```

- **Services**

  Run “systemctl list-unit-files --type=service” and make sure unused services are disabled or not installed. By default, they should be but here are the biggest ones according to the CIS hardening guide:

- **Logging & Auditing**

  For logging and auditing, make sure the auditd.service, rsyslog.service, and system-journald.service are installed, active, and running:

```
protera:~ # systemctl list-units --type=service --state=running | grep -E 'rsyslog|auditd|journald'
  auditd.service           loaded active running Security Auditing Service
  rsyslog.service          loaded active running System Logging Service
  systemd-journald.service loaded active running Journal Service
protera:~ #
```

  Prohibit log files from consuming large amounts of storage space by configuring log rotation:

```
protera:/etc/logrotate.d # cat /etc/logrotate.conf
# see "man logrotate" for details
# rotate log files weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create
```

2. Verification & Compliance Checking

- **Which Linux commands or configs to check**

  Mostly the things listed in part 1:
  - sshd_config file
  - sudoers file
  - check all running services > systemctl list-units –type=service –state=running
  - configure firewall (firewalld.service should be active & running)

- **Ensuring CIS standards**

  - Security compliance and vulnerability assessments are best handled by scanning against the Security Content Automation Protocol framework. The most popular toolset for doing this is OpenSCAP which is open source and maintained by the National Institute of Standards and Technology (NIST). Packages need to be installed on the server and then can be run against different profiles to generate a human readable report. For this example, the server was run against a CIS level 1 hardened SLES 15 server profile based off the Defense Information Systems Agency Security Technical Implementation Guide (DISA STIG) best practices:

    https://documentation.suse.com/compliance/all/html/SLES-openscap/index.html

    “sudo zypper install openscap openscap-utils scap-security-guide”

    “oscap xccdf eval --profile xccdf_org.ssgproject.content_profile_cis_server_l1 --results results.xml --report report.html /usr/share/xml/scap/ssg/content/ssg-sle15-ds.xml”

```
protera:~ # oscap xccdf eval --profile xccdf_org.ssgproject.content_profile_cis_server_l1 \
> --results results.xml --report report.html /usr/share/xml/scap/ssg/content/ssg-sle15-ds.xml
WARNING: Datastream component 'scap_org.open-scap_cref_pub-projects-security-oval-suse.linux.en
terprise.15-patch.xml.bz2'. Use '--fetch-remote-resources' option to download it.
WARNING: Skipping 'https://ftp.suse.com/pub/projects/security/oval/suse.linux.enterprise.15-pat
WARNING: Skipping ./pub-projects-security-oval-suse.linux.enterprise.15-patch.xml.bz2 file whic
--- Starting Evaluation ---

Title   Disable Prelinking
Rule    xccdf_org.ssgproject.content_rule_disable_prelink
Ident   CCE-91341-8
Result  pass
```

    *See Example Report (Page 7)

- **Reducing application conflicts during deployment**

    - Communication and working closely with application teams
    - Continuous Integration process of deploying code and catching conflicts early
    - Code reviews
    - Strategic rollout strategies like blue/green

3. Maintaining Compliance Over Time

- **Keep logs in a centralized location and set up alerts**

    - Use tools such as Datadog, ELK, Splunk, or cloud provided

- **Patch management**

    - Run ‘zypper patch’ using a weekly cron job or Ansible to keep server up-to-date

- **Preserve server baseline hardening**

    - Use configuration management tools like Ansible to prevent drift
    - Run scheduled OpenSCAP scans to track compliance scores
    - Initiate OpenSCAP remediation procedures when the server is detected to be out of compliance.

4. Exceptions & Practical Trade-Offs

- **Example of why controls could be tuned or bypassed**

    - Legacy applications: Some legacy applications running on outdated systems may require older, insecure protocols like Telnet, FTP, or SMBv1. Since the application cannot be re-engineered, a full bypass of the benchmark is necessary for it to function. The risk is accepted and documented, with compensating controls added, such as network segmentation to isolate the legacy system from the rest of the network.

- **Documentation and Justification of Exceptions**

    - Create a formal record
    - Detail the deviation and attach supporting evidence
    - Track exceptions in a configuration management database (CMDB)

Evaluation Characteristics

Evaluation target	protera
Benchmark URL	#scap_org.open-scap_comp_ssg-sle15-xccdf.xml
Benchmark ID	xccdf_org.ssgproject.content_benchmark_SLE-15
Benchmark version	0.1.77
Profile ID	xccdf_org.ssgproject.content_profile_cis_server_l1
Started at	2025-10-11T11:02:55-06:00
Finished at	2025-10-11T11:03:08-06:00
Performed by	ryan
Test system	cpe:/a:redhat:openscap:1.3.6

CPE Platforms

- cpe:/o:suse:linux_enterprise_desktop:15
- cpe:/o:suse:linux_enterprise_server:15

Addresses

- IPv4 127.0.0.1
- IPv4 10.0.0.95
- IPv6 0:0:0:0:0:0:0:1
- IPv6 fd1f:85c6:a710:d379:3ca9:151b:c23c:6e5c
- IPv6 fd1f:85c6:a710:d379:be24:11ff:fee9:f4b3
- IPv6 fe80:0:0:0:be24:11ff:fee9:f4b3
- MAC 00:00:00:00:00:00
- MAC BC:24:11:E9:F4:B3

Compliance and Scoring

The target system did not satisfy the conditions of 109 rules! Please review rule results and consider applying remediation.

Rule results

168 passed | 109 failed | 10

Severity of failed rules

8 other | 3 low | 94 medium | 4 high

Score

Scoring system	Score	Maximum	Percent
urn:xccdf:scoring:default	72.233749	100.000000	72.23%