OpenAnolis 龙蜥社区

Anolis OS 安全最佳实践

Anolis OS Server Security Best Practices

v1.5.1

2024-01

Contents

1 access-and-control

1.1 确保 cron 守护进程正常启用

安全等级

- Level 1

描述

cron 守护进程用于在系统上执行批处理作业。

即使操作系统目前可能没有需要运行的用户作业，也会有系统作业需要运行，其中就可能包括安全监控等重要作业，而 cron 守护进程就是用来执行这些作业的。

修复建议

目标：确保 cron 守护进程正常启用。

1. 执行以下命令来启用 cron 进程：

```
# systemctl --now enable crond
```

扫描检测

确保 cron 守护进程正常启用。

1. 执行以下命令来确定 cron 守护进程是否正常启用：

```
# systemctl is-enabled crond
enabled
```

如结果为 `enabled` ，则视为通过此项检查。

参考

1.2 确保 /etc/crontab 的权限配置正确

安全等级

- Level 1

描述

`/etc/crontab` 文件中包含了批处理作业的信息。其读写权限需严格控制，否则可能导致：系统作业信息泄露、未授权用户篡改或删除作业信息、普通用户运行特权命令、普通用户获得 root 权限等多种风险。

应对 `/etc/crontab` 文件的读写权限进行严格限制，并确保 `/etc/crontab` 文件的所有者与所属组为 `root` ，且只有所有者可以访问该文件。

修复建议

目标：正确配置 `/etc/crontab` 文件的权限和所有者。

1. 使用以下代码，配置 `/etc/crontab` 的文件权限和所有者：

```
# chown root:root /etc/crontab
# chmod og-rwx /etc/crontab
```

扫描检测

确保 `/etc/crontab` 文件的权限配置正确。

1. 执行以下命令，检查 `/etc/crontab` 文件的权限属性：

```
# stat /etc/crontab
Access: (0600/-rw-------) Uid: ( 0/ root) Gid: ( 0/ root)
```

如果输出结果中：`Uid` 与 `Gid` 均为 `0/root` ，且 `Access` 中 `group` 与 `other` 没有任何权限，则视为通过此项检查。

参考

1.3 确保 /etc/cron.hourly 的权限配置正确

安全等级

- Level 1

描述

`/etc/cron.hourly` 包含需要每小时运行的 cron 作业。此文件中的内容不能由 `crontab` 命令操作，而是由系统管理员使用文本编辑器直接修改其内容。

应将其读、写、执行权限限制为 `root` 用户，防止普通用户修改和访问此文件。否则可能导致：系统作业信息泄露、未授权用户篡改或删除作业信息、普通用户运行特权命令、普通用户获得 `root` 权限等多种风险。

修复建议

目标：正确配置 `/etc/cron.hourly` 文件的权限和所有者。

1. 使用以下代码，配置 `/etc/cron.hourly` 的文件权限和所有者：

```
# chown root:root /etc/cron.hourly
# chmod og-rwx /etc/cron.hourly
```

扫描检测

确保 `/etc/cron.hourly` 文件的权限配置正确。

1. 执行以下命令，检查 `/etc/cron.hourly` 文件的权限属性：

```
# stat /etc/cron.hourly
Access: (0700/drwx------) Uid: ( 0/ root) Gid: ( 0/ root)
```

如果输出结果中：`Uid` 与 `Gid` 均为 `0/root` ，且 `Access` 中 `group` 与 `other` 没有任何权限，则视为通过此项检查。

参考

1.4 确保 /etc/cron.daily 的权限配置正确

安全等级

- Level 1

描述

`/etc/cron.daily` 目录中包含需要每天运行的 `cron` 作业。此目录中的内容不能由 `crontab` 命令操作，而是由系统管理员使用文本编辑器直接修改其内容。

应将其读、写、执行权限限制为 `root` 用户，防止普通用户修改和访问此目录。否则可能导致：系统作业信息泄露、未授权用户篡改或删除作业信息、普通用户运行特权命令、普通用户获得 `root` 权限等多种风险。

修复建议

目标：正确配置 `/etc/cron.daily` 目录的权限和所有者。

1. 使用以下代码，配置 `/etc/cron.daily` 目录的权限和所有者：

```
# chown root:root /etc/cron.daily
# chmod og-rwx /etc/cron.daily
```

扫描检测

确保 `/etc/cron.daily` 目录的权限配置正确。

1. 执行以下命令，检查 `/etc/cron.daily` 目录的权限属性：

```
# stat /etc/cron.daily
Access: (0700/drwx------) Uid: ( 0/ root) Gid: ( 0/ root)
```

如果输出结果中：`Uid` 与 `Gid` 均为 `0/root` ，且 `Access` 中 `group` 与 `other` 没有任何权限，则视为通过此项检查。

参考

1.5 确保 /etc/cron.weekly 的权限配置正确

安全等级

- Level 1

描述

`/etc/cron.weekly` 目录中包含需要每周运行的 `cron` 作业。此目录中的内容不能由 `crontab` 命令操作，而是由系统管理员使用文本编辑器直接修改其内容。

应将其读、写、执行权限限制为 `root` 用户，防止普通用户修改和访问此目录。否则可能导致：系统作业信息泄露、未授权用户篡改或删除作业信息、普通用户运行特权命令、普通用户获得 `root` 权限等多种风险。

修复建议

目标：正确配置 `/etc/cron.weekly` 目录的权限和所有者。

1. 使用以下代码，配置 `/etc/cron.weekly` 目录的权限和所有者：

```
# chown root:root /etc/cron.weekly
# chmod og-rwx /etc/cron.weekly
```

扫描检测

确保 `/etc/cron.weekly` 目录的权限配置正确。

1. 执行以下命令，检查 `/etc/cron.weekly` 目录的权限属性：

```
# stat /etc/cron.weekly
Access: (0700/drwx------) Uid: ( 0/ root) Gid: ( 0/ root)
```

如果输出结果中：`Uid` 与 `Gid` 均为 `0/root` ，且 `Access` 中 `group` 与 `other` 没有任何权限，则视为通过此项检查。

参考

1.6 确保 /etc/cron.monthly 的权限配置正确

安全等级

- Level 1

描述

`/etc/cron.monthly` 目录中包含需要每月运行的 `cron` 作业。此目录中的内容不能由 `crontab` 命令操作，而是由系统管理员使用文本编辑器直接修改其内容。

应将其读、写、执行权限限制为 `root` 用户，防止普通用户修改和访问此目录。否则可能导致：系统作业信息泄露、未授权用户篡改或删除作业信息、普通用户运行特权命令、普通用户获得 `root` 权限等多种风险。

修复建议

目标：正确配置 `/etc/cron.monthly` 目录的权限和所有者。

1. 使用以下代码，配置 `/etc/cron.monthly` 目录的权限和所有者：

```
# chown root:root /etc/cron.monthly
# chmod og-rwx /etc/cron.monthly
```

扫描检测

确保 `/etc/cron.monthly` 目录的权限配置正确。

1. 执行以下命令，检查 `/etc/cron.monthly` 目录的权限属性：

```
# stat /etc/cron.monthly
Access: (0700/drwx------) Uid: ( 0/ root) Gid: ( 0/ root)
```

如果输出结果中：`Uid` 与 `Gid` 均为 `0/root` ，且 `Access` 中 `group` 与 `other` 没有任何权限，则视为通过此项检查。

参考

1.7 确保 /etc/cron.d 的权限配置正确

安全等级

- Level 1

描述

`/etc/cron.d` 目录中包含了需要定时执行的作业脚本文件，相比于 `cron.hourly` 、`cron.daily` 、 `cron.weekly` 等目录，此目录下的脚本支持以特定用户的身份执行，且可对执行时间进行更精细的控制。此目录中的内容不能由 `crontab` 命令操作，而是由系统管理员使用文本编辑器直接修改其内容。

应将其读、写、执行权限限制为 `root` 用户，防止普通用户修改和访问此目录。否则可能导致：系统作业信息泄露、未授权用户篡改或删除作业信息、普通用户运行特权命令、普通用户获得 `root` 权限等多种风险。

修复建议

目标：正确配置 `/etc/cron.d` 目录的权限和所有者。

1. 使用以下代码，配置 `/etc/cron.d` 目录的权限和所有者：

```
# chown root:root /etc/cron.d
# chmod og-rwx /etc/cron.d
```

扫描检测

确保 `/etc/cron.d` 目录的权限配置正确。

1. 执行以下命令，检查 `/etc/cron.d` 目录的权限属性：

```
# stat /etc/cron.d
Access: (0700/drwx------) Uid: ( 0/ root) Gid: ( 0/ root)
```

如果输出结果中：`Uid` 与 `Gid` 均为 `0/root` ，且 `Access` 中 `group` 与 `other` 没有任何权限，则视为通过此项检查。

参考

1.8 确保 at/cron 用户权限列表配置正确

安全等级

- Level 1

描述

`/etc/cron.allow` 、 `/etc/at.allow` 、 `/etc/at.deny` 、 `/etc/cron.deny` 文件都维护着一个用户列表，以允许或阻止特定的用户使用 `at` 或 `crontab` 命令。注意，即使某个用户没有被列在 `cron.allow` 中， `cron` 作业仍然可以以该用户的身份运行。 `cron.allow` 文件只控制 `crontab` 命令的使用权限，从而限制可对 `cron` 作业进行调度及修改的用户。一般情况下，只有系统管理员可以对 `cron` 作业进行调度及修改。更推荐使用 `cron.allow` 文件来进行权限控制，因为维护一个允许列表比维护一个拒绝列表更加容易。如果使用 `cron.deny` 来进行权限控制，那么将很有可能在创建一个新用户时，忘记将其添加到拒绝列表中，导致权限管理出现漏洞。

修复建议

目标：正确配置 `at/cron.allow` 文件及其权限。

1. 使用以下代码，配置 `at/cron.allow` 文件及其权限：
    - 删除 `/etc/cron.deny` 、 `/etc/at.deny`
    - 创建 `/etc/cron.allow` 、 `/etc/at.allow` 并配置其权限

```
# rm /etc/cron.deny
# rm /etc/at.deny
# touch /etc/cron.allow
# touch /etc/at.allow
# chmod og-rwx /etc/cron.allow
# chmod og-rwx /etc/at.allow
# chown root:root /etc/cron.allow
# chown root:root /etc/at.allow
```

扫描检测

确保 `at/cron` 用户权限列表配置正确

1. 检查 `/etc/cron.deny` 是否存在:

```
## [ -e /etc/cron.deny ] && stat /etc/cron.deny
Nothing should be returned
```

2. 检查 `/etc/at.deny` 是否存在:

```
## [ -e /etc/at.deny ] && stat /etc/at.deny
Nothing should be returned
```

3. 检查 `/etc/cron.allow` 文件权限，输出应为: `Uid` 与 `Gid` 均为 `0/root` ，且 `Access` 中 `group` 与 `other` 没有任何权限:

```
# stat /etc/cron.allow
Access: (0600/-rw-------) Uid: ( 0/ root) Gid: ( 0/ root)
```

4. 检查 `/etc/at.allow` 文件权限，输出应为: `Uid` 与 `Gid` 均为 `0/root` ，且 `Access` 中 `group` 与 `other` 没有任何权限:

```
# stat /etc/at.allow
Access: (0600/-rw-------) Uid: ( 0/ root) Gid: ( 0/ root)
```

如所有输出结果符合要求，则视为通过此项检查。

参考

1.9 确保 /etc/ssh/sshd_config 的权限配置正确

安全等级

- Level 1

描述

`/etc/ssh/sshd_config` 文件是 `sshd` 服务的配置文件。如其遭到破坏或修改，将影响对系统的远程管理，文件传输等。

`/etc/ssh/sshd_config` 文件权限需严格控制，防止其被非授权用户修改或删除。

修复建议

目标：正确配置 `/etc/ssh/sshd_config` 的权限及所有者。

1. 执行以下命令，配置 `/etc/ssh/sshd_config` 的权限及所有者：

```
# chown root:root /etc/ssh/sshd_config
# chmod og-rwx /etc/ssh/sshd_config
```

扫描检测

确保 `/etc/ssh/sshd_config` 的权限配置正确。

1. 执行以下命令，检查 `/etc/ssh/sshd_config` 目录的权限属性：

```
# stat /etc/ssh/sshd_config
Access: (0600/-rw-------) Uid: ( 0/ root) Gid: ( 0/ root)
```

如果输出结果中：`Uid` 与 `Gid` 均为 `0/root` ，且 `Access` 中 `group` 与 `other` 没有任何权限，则视为通过此项检查。

参考

1.10 确保 SSH 远程访问权限受到管控

安全等级

- Level 2

描述

通过对 SSH 远程访问的限制，可确保只有已授权的用户才可访问系统，对系统进行操作和管理。

对 SSH 远程访问的限制，可通过更新 `/etc/ssh/sshd_config` 配置文件，添加以下参数来实现：

- AllowUsers：
  - `AllowUsers` 参数的作用是：允许特定用户通过 `SSH` 进行远程访问。该列表由空格分隔的用户名组成（不能识别 UID）。可使用 `user@host` 的形式：仅允许某用户通过特定的主机登录，对权限进行更精细的限制。
- AllowGroups：
  - `AllowGroups` 参数的作用是：允许特定用户组通过 `SSH` 进行远程访问。该列表由空格分隔的组名组成（不能识别 GID）。
- DenyUsers：
  - `DenyUsers` 参数的作用是：禁止特定用户通过 `SSH` 进行远程访问。该列表由空格分隔的用户名组成（不能识别 UID）。可使用 `user@host` 的形式：仅禁止某用户通过特定的主机登录，对权限进行更精细的限制。
- DenyGroups：
  - `DenyGroups` 参数的作用是：禁止特定用户组通过 `SSH` 进行远程访问。该列表由空格分隔的组名组成（不能识别 GID）。

至少利用以上一个参数，对 SSH 远程访问进行限制。

修复建议

目标：更新 `/etc/ssh/sshd_config` 配置文件，对 SSH 远程访问权限进行限制。

1. 更新 `/etc/ssh/sshd_config` 配置文件，至少添加以下 4 个权限控制参数中的 1 个或更多：

```
AllowUsers <userlist>
AllowGroups <grouplist>
DenyUsers <userlist>
DenyGroups <grouplist>
```

`<userlist>` 、 `<grouplist>` 、 `<userlist>` 、 `<grouplist>` 为使用空格分隔的自定义用户名列表或组列表。

扫描检测

确保 `/etc/ssh/sshd_config` 的权限配置正确

1. 执行以下命令，并查看输出结果：

```
# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts |
 ↪  awk '{print $1}')" | grep -Pi '^\h*(allow|deny)(users|groups)\h+\H+(\h+.*)?$'
# grep -Pi '^\h*(allow|deny)(users|groups)\h+\H+(\h+.*)?$' /etc/ssh/sshd_config
```

2. 执行第 `1` 步的两条命令的输出结果至少匹配以下 4 条结果中的一条：

```
allowusers <userlist>
allowgroups <grouplist>
denyusers <userlist>
denygroups <grouplist>
```

`<userlist>` 、 `<grouplist>` 、 `<userlist>` 、 `<grouplist>` 为使用空格分隔的自定义用户名列表或组列表。

参考

1.11 确保 SSH 私钥文件的权限配置正确

安全等级

- Level 1

描述

SSH 私钥是 SSH 公钥认证中使用的两个文件之一。在这种认证方式中，只有与公钥相对应匹配的私钥才能够认证成功，所以私钥是重要身份证明。如果一个未授权用户取得了某主机的 SSH 私钥，那么他就可以将身份伪装成该主机。综上，私钥的存储和处理是非常重要的，且不应当被复制为副本。

修复建议

目标：正确配置 SSH 私钥文件的权限。

1. 执行以下命令，配置 SSH 私钥文件的权限、所有者、所属组：

```
# find /etc/ssh -xdev -type f -name 'ssh_host_*_key' -exec chmod u-x,g-wx,o-rwx {} \;
# find /etc/ssh -xdev -type f -name 'ssh_host_*_key' -exec chown root:ssh_keys {} \;
```

扫描检测

确保 SSH 私钥文件的权限配置正确

1. 执行以下命令，检查 SSH 私钥文件的权限、所有者、所属组的配置是否符合要求：

```
# find /etc/ssh -xdev -type f -name 'ssh_host_*_key' -exec stat {} \;
```

- 执行结果：

```
  File: /etc/ssh/ssh_host_ed25519_key
  Size: 399         Blocks: 8          IO Block: 4096   regular file
Device: fd01h/64769d    Inode: 787244      Links: 1
Access: (0600/-rw-------)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2022-05-26 23:37:16.614347951 +0800
```

```
Modify: 2022-05-25 16:38:17.255207117 +0800
Change: 2022-05-25 16:38:17.255207117 +0800
 Birth: 2022-05-25 16:38:17.255207117 +0800
  File: /etc/ssh/ssh_host_rsa_key
  Size: 1679 Blocks: 8 IO Block: 4096 regular file
Device: ca01h/51713d Inode: 8628138 Links: 1
Access: (0640/-rw-r-----) Uid: ( 0/ root) Gid: ( 993/ssh_keys)
Access: 2018-10-22 18:24:56.861750616 +0000
Modify: 2018-10-22 18:24:56.861750616 +0000
Change: 2018-10-22 18:24:56.873750616 +0000
 Birth: -
```

如果输出结果中:

- 如果 SSH 私钥文件的所属组（`Gid`）为 `ssh_keys`，其权限（Access）应为 `0640` 或更小，`Uid` 应为 `0/root`。
- 如果 SSH 私钥文件的所属组（`Gid`）为 `root`，其权限（Access）应为 `0600` 或更小，`Uid` 应为 `0/root`。

根据 SSH 私钥文件所属组（`Gid`）的不同，分别对其权限（Access）和 `Uid` 进行检查。如均满足条件，则视为通过此项检查。

参考

1.12 确保 SSH 公钥文件的权限配置正确

安全等级

- Level 1

描述

SSH 公钥是 SSH 公钥认证中使用的两个文件之一。在这种认证方式中，公钥是用来验证对应私钥生成的数字签名的密钥，只有与私钥相对应的公钥才能够成功认证。如果公钥文件被一个未经授权的用户修改，SSH 服务很可能会受到影响。

修复建议

目标：正确配置 SSH 公钥文件的权限。

1. 执行以下命令，配置 SSH 公钥文件的权限、所有者、所属组：

```
# find /etc/ssh -xdev -type f -name 'ssh_host_*_key.pub' -exec chmod u-x,go-wx {} \;
# find /etc/ssh -xdev -type f -name 'ssh_host_*_key.pub' -exec chown root:root {} \;
```

扫描检测

确保 SSH 公钥文件的权限配置正确

1. 执行以下命令，检查 SSH 公钥文件的权限、所有者、所属组的配置是否符合要求：

```
# find /etc/ssh -xdev -type f -name 'ssh_host_*_key.pub' -exec stat {} \;
```

2. 执行结果：

```
  File: /etc/ssh/ssh_host_ed25519_key.pub
  Size: 93          Blocks: 8          IO Block: 4096   regular file
Device: fd01h/64769d    Inode: 787245      Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2022-05-26 23:37:16.614347951 +0800
Modify: 2022-05-25 16:38:17.255207117 +0800
```

```
Change: 2022-05-25 16:38:17.255207117 +0800
 Birth: 2022-05-25 16:38:17.255207117 +0800
```

如果输出结果：`Access` 中权限为 `0644` 或更低、`group` 与 `other` 没有可执行 (`x`) 与可写 (`w`) 权限，且 `Uid` 与 `Gid` 均为 `0/root` ，则视为通过此项检查。

参考

1.13 确保 SSH 的 LogLevel 配置正确

安全等级

- Level 1

描述

SSH 配置文件：`/etc/ssh/sshd_config` 中的 `LogLevel` 参数，可定义 SSH 日志信息的冗余级别。

常用的级别有：

- INFO –> INFO 级别是默认级别，只记录 SSH 的用户登录行为。如登录记录、注销记录等。这些日志具有长期价值。
- VERBOSE –> VERBOSE 级别除了记录 SSH 的用户登录行为之外，还对所有用户登录的 SSH 密钥的密钥指纹进行了记录。
- DEBUG –> DEBUG 级别主要在开发过程中用于交互式调查的日志。这些日志主要包含对调试有用的信息，且没有长期价值。除开发调试外，不推荐使用 DEBUG 级别，因为其提供的数据太多，会对重要的安全信息造成干扰。

推荐使用 INFO 或 VERBOSE 作为生产环境中的日志级别。

修复建议

对 `/etc/ssh/sshd_config` 配置文件的 `LogLevel` 参数进行配置。

1. 编辑 `/etc/ssh/sshd_config` 配置文件，修改 `LogLevel` 参数，或添加以下代码，对 `LogLevel` 参数进行配置：

```
LogLevel VERBOSE
```

OR

```
LogLevel INFO
```

- 以上代码为二选一，可根据使用环境自行选择。

- 如 `/etc/ssh/sshd_config` 配置文件没有 `LogLevel` 参数或为：`#LogLevel INFO` （注释状态），可不进行操作，SSH 默认日志级别即为 `INFO` 级别。

扫描检测

确保 `SSH` 的 `LogLevel` 配置正确。

1. 执行以下命令，验证 `SSH` 的 `LogLevel` 配置是否正确：

```
# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts |
 ↪  awk '{print $1}')" | grep loglevel
loglevel VERBOSE or loglevel INFO
# grep -i 'loglevel' /etc/ssh/sshd_config | grep -Evi '(VERBOSE|INFO)'
Nothing should be returned
```

如果第一条命令执行后返回 `VERBOSE` 或 `INFO` 中任意一个结果，且第二条命令执行后，没有返回任何结果，则视为通过此项检查。

参考

1.14 确保 SSH 的 MaxAuthTries 设置为 4 或更小

安全等级

- Level 1

描述

SSH 配置文件：`/etc/ssh/sshd_config` 中的 MaxAuthTries 参数规定了每个会话连接所允许的最大认证尝试次数。当登录失败次数达到一半时，错误信息将被写入 syslog 文件，记录登录失败信息。

将 MaxAuthTries 参数设置为一个较低的数字，可最大限度地降低对 SSH 服务器暴力攻击的成功率。推荐设置为 4 或更小。

修复建议

对 `/etc/ssh/sshd_config` 配置文件的 `MaxAuthTries` 参数进行配置。

1. 编辑 `/etc/ssh/sshd_config` 配置文件，修改 `MaxAuthTries` 参数，或添加以下代码，对 `MaxAuthTries` 参数进行配置：

```
MaxAuthTries 4
```

- `MaxAuthTries` 参数默认为 `6` ，建议配置为 `4` 或更小。

扫描检测

确保 `SSH` 的 `MaxAuthTries` 配置正确。

1. 执行以下命令，验证 `SSH` 的 `MaxAuthTries` 配置是否正确：

```
# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts |
 ↪  awk '{print $1}')" | grep maxauthtries
maxauthtries 4
# grep -Ei '^\s*maxauthtries\s+([5-9]|[1-9][0-9]+)' /etc/ssh/sshd_config
Nothing is returned
```

如果第一条命令执行后返回 `4` 或更小的值，且第二条命令执行后，没有返回任何结果，则视为通过此项检查。

参考

1.15 确保 SSH IgnoreRhosts 参数正确配置

安全等级

- Level 1

描述

SSH 配置文件：`/etc/ssh/sshd_config` 中的 IgnoreRhosts 参数指定.rhosts 和.shosts 文件不被用于 RhostsRSAAuthentication 或 Host-basedAuthentication。

- `.rhosts` 、 `.shosts` 文件是一种控制系统间信任的关系的方法，如果一个系统信任另一个系统，则这个系统不需要密码就允许来自受信认系统的登录。这是一个老旧的配置，应当在 SSH 配置中明确禁用。

IgnoreRhosts 参数的正确设定，可强制要求 SSH 在登录时必须使用密码进行验证，提高 SSH 登录验证的安全性。

修复建议

对 `/etc/ssh/sshd_config` 配置文件的 `IgnoreRhosts` 参数进行配置。

1. 编辑 `/etc/ssh/sshd_config` 配置文件，修改 `IgnoreRhosts` 参数，或添加以下代码，对 `IgnoreRhosts` 参数进行配置：

```
IgnoreRhosts yes
```

- `IgnoreRhosts` 参数默认即为 `yes` ，如已正确配置，可不用修改。

扫描检测

确保 SSH IgnoreRhosts 参数正确配置。

1. 执行以下命令，验证 `SSH` 的 `IgnoreRhosts` 配置是否正确：

```
# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts |
 ↪  awk '{print $1}')" | grep ignorerhosts
ignorerhosts yes
```

```
# grep -Ei '^\s*ignorerhosts\s+no\b' /etc/ssh/sshd_config
Nothing is returned
```

如果第一条命令执行后返回 `yes` ，且第二条命令执行后，没有返回任何结果，则视为通过此项检查。

参考

1.16 确保 SSH HostbasedAuthentication 参数正确配置

安全等级

- Level 1

描述

SSH 配置文件：`/etc/ssh/sshd_config` 中的 `HostbasedAuthentication` 参数指定了是否允许通过受信任的主机、`.rhosts` 文件内的用户、或 `/etc/hosts.equiv` 进行身份认证。这个选项只适用于 SSH-2 版本。

即使在 `/etc/pam.conf` 配置文件中禁用了对 `.rhosts` 文件的支持，`.rhosts` 文件也是无效的，也仍然需要在 SSH 中禁用 `.rhosts` 文件，以提供额外的保护能力。

修复建议

对 `/etc/ssh/sshd_config` 配置文件的 `HostbasedAuthentication` 参数进行配置。

1. 编辑 `/etc/ssh/sshd_config` 配置文件，修改 `HostbasedAuthentication` 参数，或添加以下代码，对 `HostbasedAuthentication` 参数进行配置：

```
HostbasedAuthentication no
```

- `HostbasedAuthentication` 参数默认即为 `no`，如已正确配置，可不用修改。

扫描检测

确保 SSH HostbasedAuthentication 参数正确配置。

1. 执行以下命令，验证 `SSH` 的 `HostbasedAuthentication` 配置是否正确：

```
# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts |
 ↪  awk '{print $1}')" | grep hostbasedauthentication
hostbasedauthentication no
# grep -Ei '^\s*HostbasedAuthentication\s+yes' /etc/ssh/sshd_config
Nothing is returned
```

如果第一条命令执行后返回 `no` ，且第二条命令执行后，没有返回任何结果，则视为通过此项检查。

参考

1.17 确保禁用 root 用户通过 SSH 登录

安全等级

- Level 1

描述

SSH 配置文件：`/etc/ssh/sshd_config` 中的 `PermitRootLogin` 参数指定 `root` 用户是否可以使用 ssh 登录。

不允许 `root` 用户通过 SSH 登录：要求系统管理员使用自己的个人账户进行 SSH 登录，然后通过 `sudo` 或 `su` 提升权限到 `root` 。这样可在发生安全事件时提供清晰的审计线索。

在对此项安全建议进行配置前，应确认还有其他可用的系统管理员用户账号，否则在配置生效后，将可能导致无法进行 SSH 远程管理。

修复建议

对 `/etc/ssh/sshd_config` 配置文件的 `PermitRootLogin` 参数进行配置。

1. 编辑 `/etc/ssh/sshd_config` 配置文件，修改 `PermitRootLogin` 参数，或添加以下代码，对 `PermitRootLogin` 参数进行配置：

```
PermitRootLogin no
```

扫描检测

确保 SSH PermitRootLogin 参数正确配置。

1. 执行以下命令，验证 `SSH` 的 `PermitRootLogin` 配置是否正确：

```
# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts |
 ↪  awk '{print $1}')" | grep permitrootlogin
permitrootlogin no
# grep -Ei '^\s*PermitRootLogin\s+yes' /etc/ssh/sshd_config
Nothing is returned
```

如果第一条命令执行后返回 `no` ，且第二条命令执行后，没有返回任何结果，则视为通过此项检查。

参考

1.18 确保 SSH PermitEmptyPasswords 参数正确配置

安全等级

- Level 1

描述

SSH 配置文件：`/etc/ssh/sshd_config` 中的 `PermitEmptyPasswords` 参数指定用户是否可以使用空字符串密码进行 SSH 登录。

禁止密码为空的帐户进行远程 SSH 登录，可有效减少未授权用户登录的可能性。

修复建议

对 `/etc/ssh/sshd_config` 配置文件的 `PermitEmptyPasswords` 参数进行配置。

1. 编辑 `/etc/ssh/sshd_config` 配置文件，修改 `PermitEmptyPasswords` 参数，或添加以下代码，对 `PermitEmptyPasswords` 参数进行配置：

```
PermitEmptyPasswords no
```

- `/etc/ssh/sshd_config` 配置文件的 `PermitEmptyPasswords` 参数在未配置或注释的状态下，默认为 `PermitEmptyPasswords no`

扫描检测

确保 SSH PermitEmptyPasswords 参数正确配置。

1. 执行以下命令，验证 `SSH` 的 `PermitEmptyPasswords` 配置是否正确：

```
# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts |
 ↪  awk '{print $1}')" | grep permitemptypasswords
permitemptypasswords no
# grep -Ei '^\s*PermitEmptyPasswords\s+yes' /etc/ssh/sshd_config
Nothing is returned
```

如果第一条命令执行后返回 `no` ，且第二条命令执行后，没有返回任何结果，则视为通过此项检查。

参考

1.19 确保 SSH PermitUserEnvironment 参数正确配置

安全等级

- Level 1

描述

SSH 配置文件：`/etc/ssh/sshd_config` 中的 `PermitUserEnvironment` 参数控制是否允许用户向 SSH 守护进程设置环境变量选项。允许用户通过 SSH 守护进程设置环境变量，有可能使其绕过安全控制执行未授权的程序或脚本（例如，设置一个执行路径，让 ssh 执行木马程序）。

修复建议

对 `/etc/ssh/sshd_config` 配置文件的 `PermitUserEnvironment` 参数进行配置。

1. 编辑 `/etc/ssh/sshd_config` 配置文件，修改 `PermitUserEnvironment` 参数，或添加以下代码，对 `PermitUserEnvironment` 参数进行配置：

```
PermitUserEnvironment no
```

- `/etc/ssh/sshd_config` 配置文件的 `PermitUserEnvironment` 参数在未配置或注释的状态下，默认为 `PermitUserEnvironment no`

扫描检测

确保 SSH PermitUserEnvironment 参数正确配置。

1. 执行以下命令，验证 `SSH` 的 `PermitUserEnvironment` 配置是否正确：

```
# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts |
 ↪  awk '{print $1}')" | grep permituserenvironment
permituserenvironment no
# grep -Ei '^\s*PermitUserEnvironment\s+yes' /etc/ssh/sshd_config
Nothing is returned
```

如果第一条命令执行后返回 `no` ，且第二条命令执行后，没有返回任何结果，则视为通过此项检查。

参考

1.20 确保 SSH 配置了空闲超时时间

安全等级

- Level 1

描述

SSH 配置文件：`/etc/ssh/sshd_config` 中的 `ClientAliveInterval` 和 `ClientAliveCountMax` 两个参数控制 ssh 会话超时时间。

- `ClientAliveInterval` 参数设置了一个超时时间（以秒为单位），如果超时后没有收到来自客户端的数据，`sshd` 将通过加密通道发送一个心跳消息，要求客户端作出响应。此参数默认值为 0，表示不会向客户端发送心跳消息。
- `ClientAliveCountMax` 参数设置了在 `sshd` 没有收到客户端任何响应的情况下，重复发送心跳消息的次数。如发送次数达到阈值，sshd 将断开客户端的连接，终止会话。此参数默认值为 3。
- 客户端心跳消息是通过加密通道发送的。
- 将 `ClientAliveCountMax` 设置为 0，表示 SSH 客户端会话在到达配置的超时时间且没有对心跳消息做出响应后，不再重复发送心跳消息进行确认，直接断开其链接。

例：如 `ClientAliveInterval` 设置为 15，`ClientAliveCountMax` 设置为默认值 (3)，无响应的 SSH 客户端将在大约 45 秒 (15*3) 后被断开连接。

如果没有配置超时时间，可能会使未经授权的用户通过 ssh 会话访问到操作系统（例如，已登录的用户离开计算机，但没有及时锁定屏幕）。正确配置超时时间，可降低出现这种风险的几率。

- 建议 `ClientAliveInterval` 参数的值，设置不大于 `900` 秒 (15 分钟)。
- 建议 `ClientAliveCountMax` 参数的值，设置为 0。
- 通过以上设置，可实现自动断开不活跃时间超过 15 分钟的的 `ssh` 会话。

在某些情况下，这个设置可能会导致通过 SSH 或依赖 SSH 的长期运行的远程自动化工具或脚本意外终止。在配置这些参数时，应充分考虑到这些脚本的需求，并计算适当的 `ServerAliveInterval` 和 `ClientAliveInterval` 参数值，以确保操作的连续性。

修复建议

对 `/etc/ssh/sshd_config` 配置文件的 `ClientAliveInterval` 和 `ClientAliveCountMax` 参数进行配置。

1. 编辑 `/etc/ssh/sshd_config` 配置文件,修改 `ClientAliveInterval` 和 `ClientAliveCountMax` 参数，或添加以下代码，对 `ClientAliveInterval` 和 `ClientAliveCountMax` 参数进行配置:

```
ClientAliveInterval 900
ClientAliveCountMax 0
```

默认配置为 `clientaliveinterval 0` 、 `clientalivecountmax 3` ，需修改为建议值。

扫描检测

确保 SSH 配置了空闲超时时间。

1. 执行如下命令，验证 `ClientAliveInterval` 的值小于等于 `900` ：

```
# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts |
 ↪  awk '{print $1}')" | grep clientaliveinterval
clientaliveinterval 900
```

2. 执行如下命令，验证 `clientalivecountmax` 的值为 `0` ：

```
# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts |
 ↪  awk '{print $1}')" | grep clientalivecountmax
clientalivecountmax 0
```

3. 执行如下命令，检查是否没有返回任何结果:

```
# grep -Ei '^\s*ClientAliveInterval\s+(0|9[0-9][1-9]|[1-9][0-9][0-9][0-9]+|1[6-9]m|
 ↪  [2-9][0-9]m|[1-9][0-9][0-9]+m)\b' /etc/ssh/sshd_config
Nothing is returned
# grep -Ei '^\s*ClientAliveCountMax\s+([1-9]|[1-9][0-9]+)\b' /etc/ssh/sshd_config
Nothing is returned
```

如满足以上 `3` 条检查项目，则视为通过此项检查。

参考

1.21 确保 SSH LoginGraceTime 被设置为 60 秒或更短

安全等级

- Level 1

描述

SSH 配置文件：`/etc/ssh/sshd_config` 中的 `LoginGraceTime` 参数规定了 SSH 服务认证的等待时间。等待时间越长，未经认证的会话连接就积压越多。等待时间应被限制在一个适当的范围内，确保 SSH 服务的会话资源不被未授权用户占用。

将 `LoginGraceTime` 参数设置为较低的值，可以降低 SSH 服务器被暴力攻击成功的风险。建议设置为 `60` 秒 (1 分钟)。

修复建议

对 `/etc/ssh/sshd_config` 配置文件的 `LoginGraceTime` 参数进行配置。

1. 编辑 `/etc/ssh/sshd_config` 配置文件，修改 `LoginGraceTime` 参数，或添加以下代码，对 `LoginGraceTime` 参数进行配置：

```
LoginGraceTime 60
```

默认值为 `120` 秒，需修改为建议值。

扫描检测

确保 SSH LoginGraceTime 被设置为 60 秒或更短。

1. 执行以下命令，验证 `SSH` 的 `LoginGraceTime` 配置是否正确：

```
# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts |
 ↪  awk '{print $1}')" | grep logingracetime
logingracetime 60
```

```
# grep -Ei '^\s*LoginGraceTime\s+(0|6[1-9]|[7-9][0-9]|[1-9][0-9][0-9]+|[^1]m)' /etc/
  ↪ ssh/sshd_config
Nothing is returned
```

如果第一条命令执行后返回值 `<=60` ，且第二条命令执行后，没有返回任何结果，则视为通过此项检查。

参考

1.22 确保 SSH 已配置警告横幅

安全等级

- Level 1

描述

SSH 配置文件：`/etc/ssh/sshd_config` 中的 `Banner` 参数指定了一个文件。当用户连接到服务器，在其输入用户名之后，`Banner` 参数指定的文件中的内容将会被展示在其终端上。

横幅是用来告知或警告连接用户当前所登录设备的用途、使用政策及法律风险等，此信息会在用户正常登录之前展示。

修复建议

对 `/etc/ssh/sshd_config` 配置文件的 `Banner` 参数进行配置。

1. 编辑 `/etc/ssh/sshd_config` 配置文件，修改 `Banner` 参数，或添加以下代码，对 `Banner` 参数进行配置：

```
Banner /etc/issue.net
```

- 默认没有横幅文件，`Banner` 参数值为 `none` ，需改为建议值。
- `/etc/issue.net` 为自定义文本文件，其内容将会作为 login banner 展示：
  - 例：`Authorized uses only. All activity may be monitored and reported.`

扫描检测

确保 SSH 已配置警告横幅。

1. 执行以下命令，验证 `SSH` 的 `Banner` 配置是否正确：

```
# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts |
 ↪  awk '{print $1}')" | grep banner
banner /etc/issue.net
```

参考

1.23 确保 SSH PAM 已启用

安全等级

- Level 1

描述

PAM（Pluggable Authentication Modules）是由 Sun 提出的一种认证机制。

它通过提供一些动态链接库和一套统一的 API，将系统提供的认证服务与该服务的认证方式分开，使得系统管理员可以灵活的根据需要给不同的服务配置不同的认证方式而无需更改服务程序，同时也便于向系统中添加新的认证手段。`UsePAM` 参数如果设置为 “yes”，将启用 PAM 认证。

注意：如果启用 `UsePAM` ，将不能以非 `root` 用户的身份运行 `sshd(8)` 。

修复建议

对 `/etc/ssh/sshd_config` 配置文件的 `UsePAM` 参数进行配置。

1. 编辑 `/etc/ssh/sshd_config` 配置文件，修改 `UsePAM` 参数，或添加以下代码，对 `UsePAM` 参数进行配置：

```
UsePAM yes
```

默认值为 `yes` 。

扫描检测

确保 SSH PAM 已启用。

1. 执行以下命令，验证 `SSH` 的 `UsePAM` 配置是否正确：

```
# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts |
 ↪  awk '{print $1}')" | grep -i usepam
usepam yes
```

```
# grep -Ei '^\s*UsePAM\s+no' /etc/ssh/sshd_config
Nothing is returned
```

如果第一条命令执行后返回 `yes` ，且第二条命令执行后，没有返回任何结果，则视为通过此项检查。

参考

1.24 确保 SSH MaxStartups 参数正确配置

安全等级

- Level 1

描述

SSH 配置文件：`/etc/ssh/sshd_config` 中的 `MaxStartups` 参数指定了 SSH 守护进程的未认证连接的最大并发数量。

为了保护系统免受因大量未完成的认证连接尝试（DoS 攻击）而导致的拒绝服务或服务崩溃，需正确配置 `MaxStartups` 参数，对验证速率进行限制，防止 SSH 守护程序无法响应正常用户的访问请求。

`MaxStartups` 参数控制 SSH 等待认证的连接并发数（注意：已经成功通过认证并建立起会话的连接数，不在该参数的控制范围之内）。

该参数可以是冒号分隔的三个参数值（如：`10:30:100`），它们的含义分别如下：

- 10：等待认证阶段的最大并发连接数。若超过此数量，后面的认证请求连接将被拒绝。即：最多有 10 个人可以同时向 sshd 发起登录请求。
- 30：该参数是一个概率值（百分比的形式）。若设置了该参数，超过了上限（10）的连接，将会被随机拒绝，拒绝的比率是 30%。即：超过了 10 个连接以后，后续的认证请求连接中，每 3 个中会有一个被随机拒绝。
- 60：随着并发连接数的增加，这个拒绝连接的概率（30）也会逐步线性增加，当并发连接数达到最大值（60）后，后续的连接都将会被直接拒绝。

已通过认证、认证失败、或连接超时等，都会减少等待认证的并发连接数。

修复建议

对 `/etc/ssh/sshd_config` 配置文件的 `maxstartups` 参数进行配置。

1. 编辑 `/etc/ssh/sshd_config` 配置文件，修改 `maxstartups` 参数，或添加以下代码，对 `maxstartups` 参数进行配置：

```
maxstartups 10:30:60
```

默认值为 `10:30:100` ，需修改为建议值。

扫描检测

确保 SSH MaxStartups 参数正确配置。

1. 执行以下命令，验证 `SSH` 的 `MaxStartups` 参数配置是否正确：

```
# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts |
 ↪  awk '{print $1}')" | grep -i maxstartups
maxstartups 10:30:60
# grep -Ei '^\s*maxstartups\s+(((1[1-9]|[1-9][0-9][0-9]+):([0-9]+):([0-9]+))|
 ↪  (([0-9]+):(3[1-9]|[4-9][0-9]|[1-9][0-9][0-9]+):([0-9]+))|(([0-9]+):([0-9]+):
 ↪  (6[1-9]|[7-9][0-9]|[1-9][0-9][0-9]+)))' /etc/ssh/sshd_config
Nothing is returned
```

如果第一条命令执行后返回 `10:30:60` 或更加严格的规则，且第二条命令执行后，没有返回任何结果，则视为通过此项检查。

参考

1.25 确保 SSH MaxSessions 参数设置为 10 或以下

安全等级

- Level 1

描述

`MaxSessions` 参数指定 SSH 允许打开的最大会话数。

为了保护系统免受因大量未完成的认证连接尝试（DoS 攻击）而导致的拒绝服务或服务崩溃，需正确配置 `MaxSessions` 参数对最大会话数进行限制，防止 SSH 守护程序崩溃无法响应正常用户的访问请求。

修复建议

对 `/etc/ssh/sshd_config` 配置文件的 `MaxSessions` 参数进行配置

1. 编辑 `/etc/ssh/sshd_config` 配置文件，修改 `MaxSessions` 参数，或添加以下代码，对 `MaxSessions` 参数进行配置：

```
MaxSessions 10
```

- `/etc/ssh/sshd_config` 配置文件的 `MaxSessions` 参数在未配置或注释的状态下，默认为 `MaxSessions 10`

扫描检测

确保 SSH MaxSessions 参数设置为 10 或以下

1. 执行以下命令，验证 `SSH` 的 `MaxSessions` 配置是否正确：

```
# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts |
 ↪  awk '{print $1}')" | grep -i maxsessions
maxsessions 10
```

```
# grep -Ei '^\s*MaxSessions\s+(1[1-9]|[2-9][0-9]|[1-9][0-9][0-9]+)' /etc/ssh/
 ↪  sshd_config
Nothing is returned
```

如果第一条命令执行后返回值小于 `10` ，且第二条命令执行后，没有返回任何结果，则视为通过此项检查。

参考

1.26 确保全局加密策略不被覆盖

安全等级

- Level 1

描述

对于 openSSH 来说，可选择是否覆盖全局加密策略。

覆盖全局加密策略可能会导致系统使用不安全的密码策略，如：MAC、KexAlgoritms 和 GSSAPIKexAlgorithsm 等。

修复建议

对 `/etc/sysconfig/sshd` 配置文件的 `CRYPTO_POLICY` 参数进行配置

1. 编辑 `/etc/sysconfig/sshd` 配置文件，注释掉 `CRYPTO_POLICY` 参数：

```
# sed -ri "s/^\s*(CRYPTO_POLICY\s*=.*)$/# \1/" /etc/sysconfig/sshd
```

2. 重启 sshd 服务，使其生效：

```
# systemctl reload sshd
```

扫描检测

确保全局加密策略不被覆盖。

1. 执行以下命令，验证 `SSH` 的 `CRYPTO_POLICY` 配置是否正确：

```
# grep -i '^\s*CRYPTO_POLICY=' /etc/sysconfig/sshd
Nothing is returned
```

如无任何返回，则视为通过此项检查。

参考

1.27 确保设置了密码复杂性检查策略

安全等级

- Level 1

描述

`pam_pwquality.so` 模块用于检查密码的强度。它进行的检查包括：确保密码不是字典中的单词，有一定的长度，包含混合的字符（如字母、数字、其他）等等。以下是 `pam_pwquality.so` 选项的定义。

- `try_first_pass` - 从先前堆叠的 PAM 模块中检索密码。如果不可用，则提示用户输入密码。
- `retry=3` - 允许尝试 3 次，如都不符合规则，则发送失败信息。
- `minlen=14` - 密码必须是 14 个字符或以上。

以下任何一种方法都可以用来强制执行复杂密码：

- `minclass=4` - 为新密码提供至少四类字符。
- `dcredit=-1` - 至少包含一个数字。
- `ucredit=-1` - 至少包含一个大写的字符。
- `ocredit=-1` - 至少包含一个特殊字符。
- `lcredit=-1` - 至少提供一个小写字符。

可对以上参数进行修改，来制定符合实际环境的密码策略。

复杂密码可保护系统不被暴力入侵。

修复建议

编辑 `/etc/security/pwquality.conf` 文件，对密码复杂性检查策略进行配置。

1. 编辑 `/etc/security/pwquality.conf` 配置文件，修改或添加 `minlen` 参数，对密码长度进行规范：

```
# minlen = 14
```

`minlen` 默认值为 `8`

2. 编辑 `/etc/security/pwquality.conf` 配置文件，修改或添加 `minclass` 参数，对密码复杂度进行规范：

```
# minclass = 4
# dcredit = -1
# ucredit = -1
# ocredit = -1
# lcredit = -1
```

第 2 步的代码，可根据实际情况，任选一条或多条进行配置。

3. 执行以下命令，来更新 `system-auth` 和 `password-auth` 文件。

```
# egrep -q "^\s*password\s+requisite\s+pam_pwquality.so\s+" /etc/pam.d/system-auth &&
 ↪  sed -ri '/^\s*password\s+requisite\s+pam_pwquality.so\s+/ { /
 ↪  ^\s*password\s+requisite\s+pam_pwquality.so(\s+\S+)*(\s+try_first_pass)(\s+.*)?$/!
 ↪  s/^(\s*password\s+requisite\s+pam_pwquality.so\s+)(.*)$/\1try_first_pass \2/ }' /
 ↪  etc/pam.d/system-auth && sed -ri '/^\s*password\s+requisite\s+pam_pwquality.so\s+/
 ↪  { /^\s*password\s+requisite\s+pam_pwquality.so(\s+\S+)*(\s+retry=[0-9]+)(\s+.*)?
 ↪  $/! s/^(\s*password\s+requisite\s+pam_pwquality.so\s+)(.*)$/\1retry=3 \2/ }' /etc/
 ↪  pam.d/system-auth && sed -ri 's/(^\s*password\s+requisite\s+pam_pwquality.so(\s+
 ↪  \S+)*\s+)retry=[0-9]+(\s+.*)?$/\1retry=3\3/' /etc/pam.d/system-auth || echo
 ↪  Ensure\ password\ creation\ requirements\ are\ configured - /etc/pam.d/system-auth
 ↪  not configured.

# egrep -q "^\s*password\s+requisite\s+pam_pwquality.so\s+" /etc/pam.d/password-auth
 ↪  && sed -ri '/^\s*password\s+requisite\s+pam_pwquality.so\s+/ { /
 ↪  ^\s*password\s+requisite\s+pam_pwquality.so(\s+\S+)*(\s+try_first_pass)(\s+.*)?$/!
 ↪  s/^(\s*password\s+requisite\s+pam_pwquality.so\s+)(.*)$/\1try_first_pass \2/ }' /
 ↪  etc/pam.d/password-auth && sed -ri '/
 ↪  ^\s*password\s+requisite\s+pam_pwquality.so\s+/ { /
 ↪  ^\s*password\s+requisite\s+pam_pwquality.so(\s+\S+)*(\s+retry=[0-9]+)(\s+.*)?$/!
 ↪  s/^(\s*password\s+requisite\s+pam_pwquality.so\s+)(.*)$/\1retry=3 \2/ }' /etc/
 ↪  pam.d/password-auth && sed -ri 's/(^\s*password\s+requisite\s+pam_pwquality.so(\s+
 ↪  \S+)*\s+)retry=[0-9]+(\s+.*)?$/\1retry=3\3/' /etc/pam.d/password-auth || echo
 ↪  Ensure\ password\ creation\ requirements\ are\ configured - /etc/pam.d/password-
 ↪  auth not configured.
```

扫描检测

确保设置了密码复杂性检查策略。

1. 执行以下命令，验证密码复杂性策略配置是否正确：

```
# grep pam_pwquality.so /etc/pam.d/system-auth /etc/pam.d/password-auth
/etc/pam.d/system-auth:password    requisite     pam_pwquality.so try_first_pass
 ↪  local_users_only retry=3 authtok_type=
/etc/pam.d/password-auth:password    requisite     pam_pwquality.so try_first_pass
 ↪  local_users_only retry=3 authtok_type=
```

2. 执行以下命令，验证密码长度配置是否符合要求（`minlen>=14`）：

```
# grep ^minlen /etc/security/pwquality.conf
minlen=14
```

3. 执行以下命令，验证密码复杂度配置是否符合要求：

```
# grep ^minclass /etc/security/pwquality.conf
minclass=3
# grep -E "^\s*\Scredit\s*=" /etc/security/pwquality.conf
dcredit=-1
ocredit=-1
```

第 1 条及第 2 条检查项目输出结果应全部符合要求，第 3 项检查项输出结果应符合其中 1 条或更多。

参考

1.28 确保配置了密码验证失败超过阈值后锁定用户

安全等级

- Level 1

描述

对于连续多次登录密码验证失败的用户，应锁定其账户。

可通过修改 `system-auth` 及 `password-auth` 配置文件中以下两个参数，对上述功能进行管理。

- `deny=n` -> 密码验证的尝试次数 (n)，超过 n 次后锁定用户
- `unlock_time=n` -> 用户锁定后解锁所需的时间 (秒)

根据实际情况对密码尝试次数和解锁时间进行配置，防范密码暴力破解。

修复建议

对 `system-auth` 及 `password-auth` 配置文件的参数进行配置。

1. 运行以下两个脚本，更新 `system-auth` 及 `password-auth` 文件，添加 `deny=5` 和 `unlock_time=900` 参数。:

```
# sed -ri "/^auth.*pam_env.so$/i auth required pam_faillock.so preauth silent deny=5
 ↪  unlock_time=900\nauth required pam_faillock.so authfail deny=5 unlock_time=900" /
 ↪  etc/pam.d/password-auth
```

```
# sed -ri "/^auth.*pam_env.so$/i auth required pam_faillock.so preauth silent deny=5
 ↪  unlock_time=900\nauth required pam_faillock.so authfail deny=5 unlock_time=900" /
 ↪  etc/pam.d/system-auth
```

扫描检测

确保配置了密码验证失败超过阈值后锁定用户

1. 执行以下命令，验证 `system-auth` 及 `password-auth` 配置文件的参数是否合规:

```
# grep -E '^\s*auth\s+required\s+pam_faillock.so\s+' /etc/pam.d/password-auth /etc/
 ↪  pam.d/system-auth
/etc/pam.d/password-auth:auth required pam_faillock.so preauth silent deny=5
 ↪  unlock_time=900
/etc/pam.d/password-auth:auth required pam_faillock.so authfail deny=5 unlock_time=900
/etc/pam.d/system-auth:auth required pam_faillock.so preauth silent deny=5
 ↪  unlock_time=900
/etc/pam.d/system-auth:auth required pam_faillock.so authfail deny=5 unlock_time=900
```

输出结果中应符合：`deny<=5` 、 `unlock_time<=900` 。

参考

1.29 确保正确限制密码复用

安全等级

- Level 1

描述

应强制用户在修改密码时，不允许使用之前 n 次内曾使用过的密码，增强密码的安全性。

`/etc/security/opasswd` 文件中存储了用户的旧密码，可以通过对其进行配置来对用户修改的密码进行判断，确保其无法使用近期已经使用过的密码。

`remember=<5>` -> 保存的旧密码个数。

修复建议

对 `/etc/pam.d/system-auth` 文件配置进行修改。

1. 执行以下命令，修改 `/etc/pam.d/system-auth` 配置文件，增加 `remember` 参数，使其对密码复用进行限制：

```
##!/bin/bash
if authselect current | awk 'NR == 1 {print $3}' | grep -q custom/; then
    PTF=/etc/pam.d/"$(authselect current | awk 'NR == 1 {print $3}' | grep custom/)"/
    ↪ system-auth
else
    PTF=/etc/pam.d/system-auth
fi
if grep -Eq '^\s*password\s+(sufficient\s+pam_unix|requi(red|site)
 ↪ \s+pam_pwhistory).so\s+([^#]+\s+)*remember=\S+\s*.*$' $PTF; then
    sed -ri 's/^\s*(password\s+(requisite|sufficient)\s+(pam_pwhistory\.so|
    ↪ pam_unix\.so)\s+)(.*)(remember=\S+\s*)(.*)$/\1\4 remember=5 \6/' $PTF
else
    sed -ri 's/^\s*(password\s+(requisite|sufficient)\s+(pam_pwhistory\.so|
    ↪ pam_unix\.so)\s+)(.*)$/\1\4 remember=5/' $PTF
fi
```

扫描检测

确保正确限制密码复用

1. 执行以下命令，验证 `/etc/pam.d/system-auth` 的配置是否正确:

```
# grep -P '^\h*password\h+(requisite|sufficient)\h+(pam_pwhistory\.so|pam_unix\.so)\h+
  ↪ ([^#\n\r]+\h+)?remember=([5-9]|[1-9][0-9]+)\h*(\h+.*)?$' /etc/pam.d/system-auth
password    sufficient    pam_unix.so try_first_pass use_authtok nullok sha512 shadow
  ↪ remember=5
```

如返回结果中 `remember>=5` ，则视为通过此项检查。

参考

1.30 确保密码哈希算法为 SHA-512

安全等级

- Level 1

描述

将密码哈希算法从 `MD5` 改为 `SHA-512` （一种更强大的散列算法）。`SHA-512` 算法相比 `MD5` 更加复杂，提高了攻击者暴力破解密码的难度，可为系统提供额外的保护。

修复建议

设置密码哈希算法为 sha512。

1. 执行以下命令，修改 `password-auth` 和 `system-auth` 配置文件中 `pam_unix.so` 的哈希算法配置为 `SHA-512` ：

```
# egrep -q "^\s*password\s+sufficient\s+pam_unix.so\s+" /etc/pam.d/system-auth && sed
 ↪  -ri '/^\s*password\s+sufficient\s+pam_unix.so\s+/ { /
 ↪  ^\s*password\s+sufficient\s+pam_unix.so(\s+\S+)*(\s+sha512)(\s+.*)?$/! s/
 ↪  ^(\s*password\s+sufficient\s+pam_unix.so\s+)(.*)$/\1sha512 \2/ }' /etc/pam.d/
 ↪  system-auth

# egrep -q "^\s*password\s+sufficient\s+pam_unix.so\s+" /etc/pam.d/password-auth &&
 ↪  sed -ri '/^\s*password\s+sufficient\s+pam_unix.so\s+/ { /
 ↪  ^\s*password\s+sufficient\s+pam_unix.so(\s+\S+)*(\s+sha512)(\s+.*)?$/! s/
 ↪  ^(\s*password\s+sufficient\s+pam_unix.so\s+)(.*)$/\1sha512 \2/ }' /etc/pam.d/
 ↪  password-auth
```

扫描检测

确保密码哈希算法为 SHA-512。

1. 执行以下命令，验证 `password-auth` 和 `system-auth` 配置文件哈希算法配置是否符合要求：

```
# grep -E '^\s*password\s+sufficient\s+pam_unix.so\s+.*sha512\s*.*$' /etc/pam.d/
 ↪  password-auth /etc/pam.d/system-auth
/etc/pam.d/password-auth:password    sufficient    pam_unix.so sha512 remember=5
 ↪  try_first_pass use_authtok nullok shadow
/etc/pam.d/system-auth:password    sufficient    pam_unix.so sha512 try_first_pass
 ↪  use_authtok nullok shadow   remember=5
```

如返回结果中 `password-auth` 和 `system-auth` 配置文件均配置了 `sha512` 算法，则视为通过此项检查。

参考

1.31 确保密码过期时间不超过 365 天

安全等级

- Level 1

描述

`/etc/login.defs` 文件中的 `PASS_MAX_DAYS` 参数，指定了密码过期时间的最大值（天）。建议最大值不超过 365 天。

修复建议

修改 `/etc/login.defs` 文件中的 `PASS_MAX_DAYS` 参数。

1. 执行以下代码，修改或添加 `/etc/login.defs` 文件中的 `PASS_MAX_DAYS` 参数，使其符合安全要求。

```
PASS_MAX_DAYS 365
```

`PASS_MAX_DAYS` 默认为 `99999` ，应修改为 `365` 或更小。

2. 将密码过期策略应用于所有用户：

```
# getent passwd | cut -f1 -d ":" | xargs -n1 chage --maxdays 365
```

扫描检测

确保密码过期时间不超过 365 天。

1. 执行以下命令，验证 `/etc/login.defs` 文件中的 `PASS_MAX_DAYS` 参数是否符合要求：

```
# grep PASS_MAX_DAYS /etc/login.defs
PASS_MAX_DAYS 365
```

2. 检查用户列表，确认所有用户的密码过期策略符合要求：

```
# grep -E '^[^:]+:[^!*]' /etc/shadow | cut -d: -f1,5
<user>:<PASS_MAX_DAYS>
```

- 其中 `<user>` 为用户名，`<PASS_MAX_DAYS>` 为密码过期天数，如：`root:365` 。

如配置文件及所有用户的密码过期时间最大值均 `<=365` ，则视为通过此项检查。

参考

1.32 确保修改密码的间隔时间不少于 7 天

安全等级

- Level 1

描述

`/etc/login.defs` 文件中的 `PASS_MIN_DAYS` 参数，允许系统管理员阻止用户修改密码，直到距离用户上次修改密码的时间至少过去了 n 天。建议将 `PASS_MIN_DAYS` 参数设置为 7 天或以上。

通过限制密码修改的频率，系统管理员可以防止用户通过在短时间内频繁修改密码以刷新密码记录，从而规避密码复用限制。

修复建议

修改 `/etc/login.defs` 文件中的 `PASS_MIN_DAYS` 参数。

1. 执行以下代码，修改或添加 `/etc/login.defs` 文件中的 `PASS_MIN_DAYS` 参数，使其符合安全要求。

```
PASS_MIN_DAYS 7
```

`PASS_MIN_DAYS` 默认为 `0` ，即不限制，应修改为 `7` 或更大。

2. 将密码修改间隔策略应用于所有用户：

```
# getent passwd | cut -f1 -d ":" | xargs -n1 chage --mindays 7
```

扫描检测

确保修改密码的间隔时间不少于 7 天。

1. 执行以下命令，验证 `/etc/login.defs` 文件中的 `PASS_MIN_DAYS` 参数是否符合要求：

```
# grep ^\s*PASS_MIN_DAYS /etc/login.defs
PASS_MIN_DAYS 7
```

2. 检查用户列表，确认所有用户的密码过期策略符合要求：

```
# grep -E ^[^:]+:[^\!*] /etc/shadow | cut -d: -f1,4
<user>:<PASS_MIN_DAYS>
```

- 其中 `<user>` 为用户名，`<PASS_MIN_DAYS>` 为密码修改间隔天数，如：`root:7` 。

如配置文件及所有用户的密码修改间隔天数均 `>=7` ，则视为通过此项检查。

参考

1.33 确保密码过期警告天数大于等于 7 天

安全等级

- Level 1

描述

`/etc/login.defs` 文件中的 `PASS_WARN_AGE` 参数，用于告知用户，其密码将在 n 天内过期。

在密码即将过期前提前发出警告，让用户有时间思考和修改一个新的符合安全规定的密码。

修复建议

修改 `/etc/login.defs` 文件中的 `PASS_WARN_AGE` 参数。

1. 执行以下代码，修改或添加 `/etc/login.defs` 文件中的 `PASS_WARN_AGE` 参数，使其符合安全要求。

```
PASS_WARN_AGE 7
```

`PASS_WARN_AGE` 默认为 `7` 。

2. 将密码过期警告策略应用于所有用户：

```
# getent passwd | cut -f1 -d ":" | xargs -n1 chage --warndays 7
```

扫描检测

确保密码过期警告天数大于等于 7 天。

1. 执行以下命令，验证 `/etc/login.defs` 文件中的 `PASS_WARN_AGE` 参数是否符合要求：

```
# grep PASS_WARN_AGE /etc/login.defs
PASS_WARN_AGE   7
```

2. 检查用户列表，确认所有用户的密码过期警告策略符合要求：

```
# grep -E ^[^:]+:[^\!*] /etc/shadow | cut -d: -f1,6
<user>:<PASS_WARN_AGE>
```

- 其中 `<user>` 为用户名，`<PASS_WARN_AGE>` 为密码过期警告天数，如：`root:7` 。

如配置文件及所有用户的密码过期警告天数均 `>=7` ，则视为通过此项检查。

参考

1.34 确保不活跃用户的锁定时间为 30 天或更短

安全等级

- Level 1

描述

长时间不活跃（密码过期）的用户，应自动将其禁用。建议设置为：禁用在密码过期 30 天后仍未重新激活的用户。

修复建议

修改密码未激活锁定时间。

1. 执行以下命令，将密码未激活锁定时间设置为 30 天。

```
# useradd -D -f 30
```

默认值为 `-1` ，即不锁定，需修改为建议值。

2. 将密码未激活锁定时间策略应用于所有用户：

```
# getent passwd | cut -f1 -d ":" | xargs -n1 chage --inactive 30
```

扫描检测

确保密码未激活锁定时间是 30 天或更短。

1. 执行以下命令，密码未激活锁定时间是否符合要求：

```
# useradd -D | grep INACTIVE
INACTIVE=30
```

2. 检查用户列表，确认所有用户的密码未激活锁定时间符合要求：

```
# grep -E ^[^:]+:[^\!*] /etc/shadow | cut -d: -f1,7
<user>:<INACTIVE>
```

- 其中 `<user>` 为用户名，`<INACTIVE>` 为密码未激活锁定天数，如：`root:30` 。

如配置文件及所有用户的密码未激活锁定天数均 `<=30` ，则视为通过此项检查。

参考

1.35 确保密码修改时间被正确记录

安全等级

- Level 2

描述

所有用户的密码更改日期都应该早于当前时间。

如果一个密码修改记录文件中，某用户的密码修改日期被篡改，日期被设定在未来某个时段，那么此用户就可以绕过所有已设定的密码到期规则。

修复建议

定期检查用户的密码修改记录是否异常。

1. 执行以下命令，检查用户的密码修改记录是否异常：

```
# for usr in $(cut -d: -f1 /etc/shadow); do [[ $(chage --list $usr | grep '^Last
 ↪  password change' | cut -d: -f2) > $(date) ]] && echo "$usr :$(chage --list $usr |
 ↪  grep '^Last password change' | cut -d: -f2)"; done
```

如无任何输出结果，说明无异常，无需进行下一步操作。

如有输出结果，则需对结果中的用户进行锁定，并手动修改错误、重新修改密码。

扫描检测

确保所有用户最后一次修改密码的时间早于当前时间。

1. 执行以下命令，检查用户的密码修改记录是否异常：

```
# for usr in $(cut -d: -f1 /etc/shadow); do [[ $(chage --list $usr | grep '^Last
 ↪  password change' | cut -d: -f2) > $(date) ]] && echo "$usr :$(chage --list $usr |
 ↪  grep '^Last password change' | cut -d: -f2)"; done
```

如无任何输出结果，则视为通过此项检查。

参考

1.36 确保虚拟用户不可通过 shell 登录

安全等级

- Level 1

描述

大多数的 Linux 发行版都提供了许多用于管理应用程序的用户，此类用户不需要使用交互式 shell。

一般情况下，这些用户的密码字段会被设置为一个无效的字符串。即便如此，仍建议将 `passwd` 文件中这类用户的 `shell` 字段设置为 `nologin` ，如此可防范这些账户被用来执行恶意脚本或命令。

修复建议

修改 `passwd` 文件中虚拟用户的 `shell` 字段，并锁定虚拟用户。

1. 修改 `passwd` 文件中虚拟用户的 `shell` 字段为 `nologin` ：

```
awk -F: '($1!="root" && $1!="sync" && $1!="shutdown" && $1!="halt" && $1!~/^\+/ &&
 ↪  $3<'"$(awk '/^\s*UID_MIN/{print $2}' /etc/login.defs)"' && $7!="'"$(which
 ↪  nologin)"'" && $7!="/bin/false") {print $1}' /etc/passwd |
while read user; do
    usermod -s $(which nologin) $user
done
```

2. 锁定虚拟用户：

```
awk -F: '($1!="root" && $1!~/^\+/ && $3<'"$(awk '/^\s*UID_MIN/{print $2}' /etc/
 ↪  login.defs)"') {print $1}' /etc/passwd | xargs -I '{}' passwd -S '{}' | awk '($2!
 ↪  ="L" && $2!="LK") {print $1}' |
while read user; do
    usermod -L $user
done
```

扫描检测

确保虚拟用户不可通过 shell 登录。

1. 执行如下两条命令，确认返回结果：

```
# awk -F: '($1!="root" && $1!="sync" && $1!="shutdown" && $1!="halt" && $1!~/^\+/ &&
 ↪  $3<'"$(awk '/^\s*UID_MIN/{print $2}' /etc/login.defs)"' && $7!="'"$(which
 ↪  nologin)"'" && $7!="/bin/false") {print}' /etc/passwd
No file should be returned

##awk -F: '($1!="root" && $1!~/^\+/ && $3<'"$(awk '/^\s*UID_MIN/{print $2}' /etc/
 ↪  login.defs)"') {print $1}' /etc/passwd | xargs -I '{}' passwd -S '{}' | awk '($2!
 ↪  ="L" && $2!="LK") {print $1}'
No file should be returned
```

如以上命令，均无返回结果，则视为通过此项检查。

参考

1.37 确保用户 shell 超时时间小于等于 900 秒

安全等级

- Level 1

描述

`TMOUT` 是一个环境变量，用于指定 `shell` 的超时时间，单位为秒。

- `TMOUT=n` -> 设置 shell 超时时间为 n 秒。`TMOUT=0` 表示禁用超时时间。
- `readonly TMOUT` -> 可将 `TMOUT` 环境变量设置为只读，避免其被篡改。
- `export TMOUT` -> 对 `TMOUT` 的值进行修改。

环境变量的配置文件：

- `/etc/profile` -> 此文件内的变量为全局变量，可作用于所有用户。
- `/etc/profile.d` -> 在系统启动或用户第一次登录 shell 时，会自动运行其目录下所有 `*.sh` 文件。
- `/etc/bashrc` -> 为每个运行 bash shell 的用户执行该文件，当 bash shell 打开时，该文件被执行，其配置对所有使用 bash 的用户打开的每个 bash 都有效。当被修改后，不用重启系统只需要打开一个新的 bash 即可生效。

设置 shell 超时时间可以减少未经授权的用户通过其他已登录用户的 shell 会话进行非法操作的情况，也能及时释放被不活跃用户占用的会话资源。

修复建议

对 `TMOUT` 环境变量的值进行配置。

1. 检查

- `/etc/bashrc` 文件
- `/etc/profile` 文件
- `/etc/profile.d/` 目录下所有以 `.sh` 结尾的文件中 `TMOUT=n` 条目的值，应不超过 `900` 且不等于 `0` ：

示例：

```
readonly TMOUT=900 ; export TMOUT
```

或

```
TMOUT=900
readonly TMOUT
export TMOUT
```

以上分别为单行与多行配置，符合任意一种即可。

扫描检测

确保用户 shell 超时时间小于等于 900 秒。

1. 执行以下命令，验证 `TMOUT` 环境变量是否正确设置：

```
##!/usr/bin/env bash
CDTOS()
{
 output1="" output2=""
 [ -f /etc/bashrc ] && BRC="/etc/bashrc"
 for f in "$BRC" /etc/profile /etc/profile.d/*.sh ; do
    grep -Pq '^\s*([^#]+\s+)?TMOUT=(900|[1-8][0-9][0-9]|[1-9][0-9]|[1-9])\b' "$f" &&
     ↪  grep -Pq '^\s*([^#]+;\s*)?readonly\s+TMOUT(\s+|\s*;|\s*$|=(900|[1-8][0-9]
     ↪  [0-9]|[1-9][0-9]|[1-9]))\b' "$f" && grep -Pq '^\s*([^#]+;\s*)?
     ↪  export\s+TMOUT(\s+|\s*;|\s*$|=(900|[1-8][0-9][0-9]|[1-9][0-9]|[1-9]))\b' "$f"
     ↪  &&  output1="$f"
 done
 grep -Pq '^\s*([^#]+\s+)?TMOUT=(9[0-9][1-9]|9[1-9][0-9]|0+|[1-9]\d{3,})\b' /etc/
  ↪  profile /etc/profile.d/*.sh "$BRC" && output2=$(grep -Ps '^\s*([^#]+\s+)?
  ↪  TMOUT=(9[0-9][1-9]|9[1-9][0-9]|0+|[1-9]\d{3,})\b' /etc/profile /etc/profile.d/
  ↪  *.sh $BRC)
 if [ -n "$output1" ] && [ -z "$output2" ]; then
    echo -e "\nPASSED\n\nTMOUT is configured in: \"$output1\"\n"
 else
    [ -z "$output1" ] && echo -e "\nFAILED\n\nTMOUT is not configured\n"
```

```
    [ -n "$output2" ] && echo -e "\nFAILED\n\nTMOUT is incorrectly configured in:
    \"$output2\"\n"
 fi
}
CDTOS
```

输出:

```
PASSED


TMOUT is configured in: "/etc/profile.d/login.sh"
```

如最输出结果为 `PASSED` ，则视为通过此项检查。其中 `TMOUT is configured in` 的值为 `TMOUT` 环境变量所在文件路径。

参考

1.38 确保 root 帐号的默认组为 GID 0

安全等级

- Level 1

描述

配置 `GID 0` 为 root 用户的所属组，可防止 root 用户拥有的文件被非特权用户访问。

修复建议

配置 root 用户的所属组。1. 执行以下命令，对 root 用户的所属组进行配置：

```
usermod -g 0 root
```

扫描检测

确保 root 帐号的默认组为 `GID 0` 。

1. 执行以下命令，验证 root 帐号的默认组是否合规：

```
# grep "^root:" /etc/passwd | cut -f4 -d:
0
```

如返回结果为 `0` ，则视为通过此项检查。

参考

1.39 确保 umask 设置为 027 或更严格

安全等级

- Level 1

描述

umask 是用来指定新创建目录和文件的默认文件权限的。在 Linux 中，所有新创建的目录的默认权限是 0777（rwxrwxrwx），而所有新创建的文件的默认权限是 0666（rw-rw-rw-）。umask 通过限制（屏蔽）这些权限来修改 Linux 的默认权限。umask 不是简单的减法，而是按位处理。在生成新文件时，其默认权限会减去 umask 中设置的位。

umask 可以用八进制数字（4、2、1）或字母（r、w、x）来设置：

- 八进制数字 -> 由三位或四位数字表示，即 `umask 0027` 或 `umask 027` 。如果使用四位数的掩码，第一位数会被忽略。剩下的三位数字分别影响 user、group、other 的权限。
- 字母 -> 用逗号分隔的列表表示 user、group、other 的权限。
- 例： `umask u=rwx,g=rx,o=` 即等于八进制数字的 `umask 027` ，此设置会将新建目录的默认权限配置为 `drwxr-x---` ，将新建文件的默认权限设置为 `rw-r-----` 。

文件和目录的真正初始权限，可通过以下的计算得到：

```
文件（或目录）的初始权限 = 文件（或目录）的最大默认权限 - umask权限
```

配置 umask 值的方法：

- pam_umask 模块：
  - 在 `/etc/login.defs` 中添加或修改 `umask` 参数（使用八进制数字格式）。
- 系统 Shell 配置文件（作用于所有用户）：
  - `/etc/profile` -> 用于为用户的 shell 设置系统范围的环境变量。
  - `/etc/profile.d` -> 会在系统启动或登录 shell 时，执行该目录下所有以 `*.sh` 结尾的文件。
  - `/etc/bashrc` -> 为每一个运行 bash shell 的用户执行此文件，当 bash shell 被打开时，该文件被读取。
- 用户 shell 配置文件（作用于当前用户）：
  - `~/.bash_profile` -> 用户在登录时，会读取此文件中的配置。

– `~/.bashrc` -> 该文件存储的是专属于个人 bash shell 的信息，当登录时以及每次打开一个新的 shell 时, 执行这个文件，在这个文件里可以自定义用户专属的个人信息。

为 umask 设置一个安全的默认值可以避免新建的目录或文件具有过多的权限，被未经授权的用户读取或篡改。

修复建议

检查 `/etc/bashrc` 、 `/etc/profile` 以及 `/etc/profile.d/` 目录中所有以 `*.sh` 结尾的文件，添加或编辑所有的 `umask` 条目，使其符合安全要求： `umask 027` ， `umask u=rwx,g=rx,o=` 或更加严格的限制。

1. 在以下文件中配置 `umask` 参数

- `/etc/bashrc`
- `/etc/profile`
- `/etc/profile.d/` 目录中所有以 `*.sh` 结尾的文件：

可使用以下命令，检查 `/etc/bashrc` 、 `/etc/profile` 以及 `/etc/profile.d/` 目录中所有已配置的 `umask` 参数：

```
# grep -RPi '(^|^[^#]*)\s*umask\s+([0-7][0-7][01][0-7]\b|[0-7][0-7][0-7][0-6]\b|[0-7]
 ↪  [01][0-7]\b|[0-7][0-7][0-6]\b|(u=[rwx]{0,3},)?(g=[rwx]{0,3},)?o=[rwx]+\b|
 ↪  (u=[rwx]{1,3},)?g=[^rx]{1,3}(,o=[rwx]{0,3})?\b)' /etc/login.defs /etc/profile* /
 ↪  etc/bashrc*
/etc/login.defs:UMASK       022
/etc/profile:    umask 002
/etc/bashrc:       umask 002
```

根据以上结果，对 `umask` 参数进行修改，例如：

```
# vi /etc/profile
umask 027
# vi /etc/profile.d/set_umask.sh
umask 027
```

2. 添加或修改 `/etc/login.defs` 文件中的 `UMASK` 及 `USERGROUPS_ENAB` 参数。

```
UMASK 027
USERGROUPS_ENAB no
```

3. 编辑 `/etc/pam.d/password-auth` 及 `/etc/pam.d/system-auth` ，添加或编辑以下内容:

```
session optional pam_umask.so
```

默认的 UMASK 配置为： `UMASK 022` 。

扫描检测

确保 umask 设置为 027 或更严格。

1. 执行以下命令，验证 umask 参数是否符合以下要求：

- 创建目录的默认权限为 750 (drwxr-x—)，创建文件的默认权限为 640 (rw-r—-)，或者更严格。

执行以下命令，观察返回结果：

```
##!/bin/bash
passing=""
grep -Eiq '^\s*UMASK\s+(0[0-7][2-7]7|[0-7][2-7]7)\b' /etc/login.defs && grep -Eqi
 ↪  '^\s*USERGROUPS_ENAB\s*"?no"?\b' /etc/login.defs && grep -Eq '^\s*session\s+
 ↪  (optional|requisite|required)\s+pam_umask\.so\b' /etc/pam.d/common-session &&
 ↪  passing=true
grep -REiq '^\s*UMASK\s+\s*(0[0-7][2-7]7|[0-7][2-7]7|u=(r?|w?|x?)(r?|w?|x?)(r?|w?|
 ↪  x?),g=(r?x?|x?r?),o=)\b' /etc/profile* /etc/bashrc* && passing=true
[ "$passing" = true ] && echo "Default user umask is set"
```

如返回 `Default user umask is set` ，则视为通过。

```
# grep -RPi '(^|^[^#]*)\s*umask\s+([0-7][0-7][01][0-7]\b|[0-7][0-7][0-7][0-6]\b|[0-7]
 ↪  [01][0-7]\b|[0-7][0-7][0-6]\b|(u=[rwx]{0,3},)?(g=[rwx]{0,3},)?o=[rwx]+\b|
 ↪  (u=[rwx]{1,3},)?g=[^rx]{1,3}(,o=[rwx]{0,3})?\b)' /etc/login.defs /etc/profile* /
 ↪  etc/bashrc*
No file should be returned
```

如没有返回任何结果，则视为通过。

如以上两组命令的输出结果，均符合要求，则视为通过此项检查。

参考

1.40 确保 su 命令的使用受到限制

安全等级

- Level 1

描述

`su` 命令允许一个用户以另一个用户的身份执行命令或 shell 脚本。该命令已被 `sudo` 取代，后者可以对特权访问进行更精细的控制。默认情况下，`su` 命令可以由任何用户执行。通过添加配置或取消对 `/etc/pam.d/su` 文件中 `pam_wheel.so` 语句的注释，可以限制 `su` 命令只允许 `wheel` 组的用户执行。

限制 `su` 命令的使用，并使用 `sudo` 来代替它，可以使系统管理员更好地控制用户权限提升、执行特权命令等操作。`sudo` 有更好的审计记录机制，其可以记录通过 `sudo` 执行的每个命令，而 `su` 只能记录使用 `su` 的用户，无法记录具体执行的命令。

修复建议

限制 `su` 命令的使用权：

1. 添加以下命令至 `/etc/pam.d/su` 文件：

```
auth required pam_wheel.so use_uid
```

2. 可在 `/etc/group` 文件的 `wheel` 参数中创建一个逗号分隔的用户列表，在列表内的用户可使用 `su` 命令：
    - 例：

```
# wheel:x:<GID>:root,<user list>
wheel:x:10:root,user1,user2,user3
```

扫描检测

确保 su 命令的使用受到限制。

1. 执行以下命令，验证输出结果是否符合要求：

```
# grep pam_wheel.so /etc/pam.d/su
auth required pam_wheel.so use_uid
```

2. 执行以下命令，检查 `wheel` 用户列表中的用户是否符合安全要求。如果没有用户被列出，则只有 `root` 用户可以使用 `su` 命令：

```
# grep wheel /etc/group
wheel:x:10:
```

如以上两条检查项输出结果均符合要求，则视为通过此项检查。

参考

1.41-ssh 服务使用协议 2

安全等级

- Level 1

描述

建议 ssh 服务使用相对于旧版本 (1) 更安全的协议 2

修复建议

ssh 配置中确保 protocol 2 ，存在该选项则修改为 2，没有则添加：

1. 执行以下命令，修改或添加 ssh 配置文件中的 Protocol 配置：

```
# grep -qiP '^Protocol' /etc/ssh/sshd_config && sed -i "/^Protocol/cProtocol 2" /etc/
 ↪  ssh/sshd_config || echo -e "Protocol 2" >> /etc/ssh/sshd_config
```

2. 执行以下命令，重启 sshd 服务：

```
# systemctl restart sshd
```

扫描检测

查看 ssh 配置文件 Protocol 行内容

```
# grep -R "^Protocol" /etc/ssh/sshd_config
Protocol 2
```

如结果为 `Protocol 2` ，则视为通过此项检查。

参考

1.42 确保密码过期时间在 30 至 90 天之间

安全等级

- Level 1

描述

`/etc/login.defs` 文件中的 `PASS_MAX_DAYS` 参数，指定了密码过期时间的最大值（天）。建议最大值在 30 至 90 天之间。

修复建议

修改 `/etc/login.defs` 文件中的 `PASS_MAX_DAYS` 参数。

1. 执行以下代码，修改或添加 `/etc/login.defs` 文件中的 `PASS_MAX_DAYS` 参数，使其符合安全要求。

```
PASS_MAX_DAYS 90
```

`PASS_MAX_DAYS` 默认为 `99999` ，应修改为 `30` 至 `90` 之间。

2. 将密码过期策略应用于所有用户：

```
# getent passwd | cut -f1 -d ":" | xargs -n1 chage --maxdays 90
```

扫描检测

确保密码过期时间在 30 至 90 天之间。

1. 执行以下命令，验证 `/etc/login.defs` 文件中的 `PASS_MAX_DAYS` 参数是否符合要求：

```
# grep PASS_MAX_DAYS /etc/login.defs
PASS_MAX_DAYS 90
```

2. 检查用户列表，确认所有用户的密码过期策略符合要求：

```
# grep -E '^[^:]+:[^!*]' /etc/shadow | cut -d: -f1,5
<user>:<PASS_MAX_DAYS>
```

- 其中 `<user>` 为用户名，`<PASS_MAX_DAYS>` 为密码过期天数，如：`root:90` 。

如配置文件及所有用户的密码过期时间最大值均在 30 至 90 天之间，则视为通过此项检查。

参考

1.43 确保修改密码的间隔时间在 7 至 14 天之间

安全等级

- Level 1

描述

`/etc/login.defs` 文件中的 `PASS_MIN_DAYS` 参数，允许系统管理员阻止用户修改密码，直到距离用户上次修改密码的时间至少过去了 n 天。建议将 `PASS_MIN_DAYS` 参数设置为 7 至 14 天之间。

通过限制密码修改的频率，系统管理员可以防止用户通过在短时间内频繁修改密码以刷新密码记录，从而规避密码复用限制。

修复建议

修改 `/etc/login.defs` 文件中的 `PASS_MIN_DAYS` 参数。

1. 执行以下代码，修改或添加 `/etc/login.defs` 文件中的 `PASS_MIN_DAYS` 参数，使其符合安全要求。

```
PASS_MIN_DAYS 7
```

`PASS_MIN_DAYS` 默认为 `0` ，即不限制，应修改为 `7` 至 `14` 之间的值。

2. 将密码修改间隔策略应用于所有用户：

```
# getent passwd | cut -f1 -d ":" | xargs -n1 chage --mindays 7
```

扫描检测

确保修改密码的间隔时间在 7 至 14 天之间。

1. 执行以下命令，验证 `/etc/login.defs` 文件中的 `PASS_MIN_DAYS` 参数是否符合要求：

```
# grep ^\s*PASS_MIN_DAYS /etc/login.defs
PASS_MIN_DAYS 7
```

2. 检查用户列表，确认所有用户的密码过期策略符合要求：

```
# grep -E ^[^:]+:[^\!*] /etc/shadow | cut -d: -f1,4
<user>:<PASS_MIN_DAYS>
```

- 其中 `<user>` 为用户名，`<PASS_MIN_DAYS>` 为密码修改间隔天数，如：`root:7` 。

如配置文件及所有用户的密码修改间隔天数均在 7 至 14 天之间，则视为通过此项检查。

参考

1.44 确保正确限制密码复用

安全等级

- Level 1

描述

应强制用户在修改密码时，不允许使用之前 n 次内曾使用过的密码，增强密码的安全性。

`/etc/security/opasswd` 文件中存储了用户的旧密码，可以通过对其进行配置来对用户修改的密码进行判断，确保其无法使用近期已经使用过的密码。

`remember=<5>` -> 保存的旧密码个数。

修复建议

对 `/etc/pam.d/system-auth` 文件配置进行修改。

执行以下命令，修改 `/etc/pam.d/system-auth` 配置文件，增加 `remember` 参数，使其对密码复用进行限制：

```
##!/bin/bash
if authselect current | awk 'NR == 1 {print $3}' | grep -q custom/; then
    PTF=/etc/pam.d/"$(authselect current | awk 'NR == 1 {print $3}' | grep custom/)"/
     ↪ system-auth
else
    PTF=/etc/pam.d/system-auth
fi
if grep -Eq '^\s*password\s+(sufficient\s+pam_unix|requi(red|site)
 ↪ \s+pam_pwhistory).so\s+([^#]+\s+)*remember=\S+\s*.*$' $PTF; then
    sed -ri 's/^\s*(password\s+(requisite|sufficient)\s+(pam_pwhistory\.so|
     ↪ pam_unix\.so)\s+)(.*)(remember=\S+\s*)(.*)$/\1\4 remember=5 \6/' $PTF
else
    sed -ri 's/^\s*(password\s+(requisite|sufficient)\s+(pam_pwhistory\.so|
     ↪ pam_unix\.so)\s+)(.*)$/\1\4 remember=5/' $PTF
fi
```

扫描检测

确保正确限制密码复用

执行以下命令，验证 `/etc/pam.d/system-auth` 的配置是否正确：

```
# grep -P '^\h*password\h+(requisite|sufficient)\h+(pam_pwhistory\.so|pam_unix\.so)\h+
  ↪ ([^#\n\r]+\h+)?remember=([5-9]|[1-9][0-9]+)\h*(\h+.*)?$' /etc/pam.d/system-auth
password    sufficient    pam_unix.so try_first_pass use_authtok nullok sha512 shadow
  ↪ remember=5
```

如返回结果中 `remember` 的值在 5 至 25 之间，则视为通过此项检查。

参考

1.45 确保正确配置了密码尝试失败次数和失败后锁定时间

安全等级

- Level 1

描述

对于连续多次登录密码验证失败的用户，应锁定其账户。

可通过修改 `system-auth` 及 `password-auth` 配置文件中以下两个参数，对上述功能进行管理。

- `deny=n` -> 密码验证的尝试次数 (n)，超过 n 次后锁定用户。建议值：3-8（次）。
- `unlock_time=n` -> 用户锁定后解锁所需的时间 (秒)。建议值：600-1800（秒）。

根据实际情况对密码尝试次数和解锁时间进行配置，防范密码暴力破解。

修复建议

对 `system-auth` 及 `password-auth` 配置文件的参数进行配置。

1. 运行以下两个脚本，更新 `system-auth` 及 `password-auth` 文件，添加 `deny=5` 和 `unlock_time=900` 参数。:

```
# sed -ri "/^auth.*pam_env.so$/i auth required pam_faillock.so preauth silent deny=5
 ↪  unlock_time=900\nauth required pam_faillock.so authfail deny=5 unlock_time=900" /
 ↪  etc/pam.d/password-auth
```

```
# sed -ri "/^auth.*pam_env.so$/i auth required pam_faillock.so preauth silent deny=5
 ↪  unlock_time=900\nauth required pam_faillock.so authfail deny=5 unlock_time=900" /
 ↪  etc/pam.d/system-auth
```

扫描检测

确保配置了密码验证失败超过阈值后锁定用户

1. 执行以下命令，验证 `system-auth` 及 `password-auth` 配置文件的参数是否合规:

```
# grep -E '^\s*auth\s+required\s+pam_faillock.so\s+' /etc/pam.d/password-auth /etc/
 ↪  pam.d/system-auth
/etc/pam.d/password-auth:auth required pam_faillock.so preauth silent deny=5
 ↪  unlock_time=900
/etc/pam.d/password-auth:auth required pam_faillock.so authfail deny=5 unlock_time=900
/etc/pam.d/system-auth:auth required pam_faillock.so preauth silent deny=5
 ↪  unlock_time=900
/etc/pam.d/system-auth:auth required pam_faillock.so authfail deny=5 unlock_time=900
```

输出结果中应符合：`deny` 的值在 3-8 之间、`unlock_time` 的值在 600-1800 之间。

参考

1.46 确保用户 shell 超时时间在 600 至 1800 秒之间

安全等级

- Level 1

描述

`TMOUT` 是一个环境变量，用于指定 `shell` 的超时时间，单位为秒。

- `TMOUT=n` -> 设置 shell 超时时间为 n 秒。`TMOUT=0` 表示禁用超时时间。
- `readonly TMOUT` -> 可将 `TMOUT` 环境变量设置为只读，避免其被篡改。
- `export TMOUT` -> 对 `TMOUT` 的值进行修改。

环境变量的配置文件：

- `/etc/profile` -> 此文件内的变量为全局变量，可作用于所有用户。
- `/etc/profile.d` -> 在系统启动或用户第一次登录 shell 时，会自动运行其目录下所有 `*.sh` 文件。
- `/etc/bashrc` -> 为每个运行 bash shell 的用户执行该文件，当 bash shell 打开时，该文件被执行，其配置对所有使用 bash 的用户打开的每个 bash 都有效。当被修改后，不用重启系统只需要打开一个新的 bash 即可生效。

设置 shell 超时时间可以减少未经授权的用户通过其他已登录用户的 shell 会话进行非法操作的情况，也能及时释放被不活跃用户占用的会话资源。

修复建议

对 `TMOUT` 环境变量的值进行配置。

检查 * `/etc/bashrc` 文件 * `/etc/profile` 文件 * `/etc/profile.d/` 目录下所有以 `.sh` 结尾的文件中 `TMOUT=n` 条目的值，应在 600~1800 之间：

示例：

```
readonly TMOUT=900 ; export TMOUT
```

或

```
TMOUT=900
readonly TMOUT
export TMOUT
```

以上分别为单行与多行配置，符合任意一种即可。

扫描检测

确保用户 shell 超时时间在 600 至 1800 秒之间。

执行以下命令，验证 `TMOUT` 环境变量是否正确设置：

```
# grep -Pio "TMOUT=[0-9]+" /etc/profile /etc/bashrc /etc/profile.d/*.sh
/etc/profile:TMOUT=1800
```

如输出 `TMOUT` 的值在 600~1800 之间，则视为通过此项检查。

参考

1.47 确保 SSH 的 MaxAuthTries 设置为 3~5

安全等级

- Level 1

描述

SSH 配置文件：`/etc/ssh/sshd_config` 中的 MaxAuthTries 参数规定了每个会话连接所允许的最大认证尝试次数。当登录失败次数达到一半时，错误信息将被写入 syslog 文件，记录登录失败信息。

将 MaxAuthTries 参数设置为一个较低的数字，可最大限度地降低对 SSH 服务器暴力攻击的成功率。推荐设置为 3~5。

修复建议

对 `/etc/ssh/sshd_config` 配置文件的 `MaxAuthTries` 参数进行配置。

1. 编辑 `/etc/ssh/sshd_config` 配置文件，修改 `MaxAuthTries` 参数，或添加以下代码，对 `MaxAuthTries` 参数进行配置：

```
MaxAuthTries 4
```

- `MaxAuthTries` 参数默认为 `6` ，建议配置为 3~5。

扫描检测

确保 `SSH` 的 `MaxAuthTries` 配置正确。

1. 执行以下命令，验证 `SSH` 的 `MaxAuthTries` 配置是否正确：

```
# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts |
 ↪  awk '{print $1}')" | grep maxauthtries
maxauthtries 4
# grep -Ei '^\s*maxauthtries\s+([5-9]|[1-9][0-9]+)' /etc/ssh/sshd_config
Nothing is returned
```

```
# grep -Ei '^\s*maxauthtries\s+([0-2])' /etc/ssh/sshd_config
Nothing is returned
```

如果第一条命令执行后返回 3~5，且其余两条命令执行后，没有返回任何结果，则视为通过此项检查。

参考

1.48 对通过网络进行管理的终端进行限制

安全等级

- Level 2

描述

对于能过 xinetd 程序启动的网络服务，比如 ftp telnet，我们就可以修改/etc/hosts.allow 和/etc/hosts.deny 的配制，来许可或者拒绝哪些 IP、主机、用户可以访问。

- /etc/hosts.allow：
  - 在此文件内加入的 IP 访问请求将被允许。
- /etc/hosts.deny：
  - 在此文件内加入的 IP 访问请求将被拒绝。

修复建议

向 `/etc/hosts.allow` 、 `/etc/hosts.deny` 配置文件中添加允许及拒绝的 IP 地址。

1. 编辑 `/etc/hosts.allow` 、 `/etc/hosts.deny` 配置文件，如没有此文件，需创建：

```
echo "ALL: 0.0.0.0/0" >> /etc/hosts.allow
echo "ALL: ALL" >> /etc/hosts.deny
```

请根据实际情况，替换 `""` 内的 IP 地址。

扫描检测

执行以下命令，验证 `/etc/hosts.allow` 、 `/etc/hosts.deny` 配置文件的内容是否正确：

```
# cat /etc/hosts.allow
# cat /etc/hosts.deny
```

检查两个配置文件中的 IP 地址，是否符合实际生产环境的需求，如符合则视为通过此项检查。

参考

1.49 锁定或删除 shutdown、halt 用户

安全等级

- Level 1

描述

锁定或删除 shutdown、halt 用户，避免生产环境内的服务器等设备被非法关机。

修复建议

锁定或删除 shutdown、halt 用户。

执行以下命令，锁定 shutdown、halt 用户

```
usermod -L shutdown
usermod -L halt
```

扫描检测

执行以下命令，验证 shutdown、halt 用户是否被锁定：

```
# passwd -S shutdown | grep -E "shutdown\s+LK"
shutdown LK 2021-06-16 7 90 7 -1 (Alternate authentication scheme in use.)

# passwd -S halt | grep -E "halt\s+LK"
halt LK 2021-06-16 7 90 7 -1 (Alternate authentication scheme in use.)
```

如输出结果符合预期，则视为通过此项检查。

参考

1.50 确保 SSH X11 转发功能被禁用

安全等级

- Level 1

描述

`X11` 协议可使用户通过 `SSH` 远程运行或操作一个 `Linux` 服务器上有图形界面的程序。但如果通过 `SSH` 登录了一个启用 `X11` 转发功能的服务器，因建立了 `X11` 转发通道，将有可能遭受到其他用户的攻击。

所以，除非确实需要直接使用 `X11` 应用程序，否则均建议禁用 `X11` 转发。

`/etc/ssh/sshd_config` 配置文件中的 `X11Forwarding` 参数控制了 `SSH` 服务是否允许 `X11` 流量转发。

注意：即使禁用了 `X11` 转发，用户仍然可以自行安装其他转发工具。

修复建议

对 `/etc/ssh/sshd_config` 配置文件的 `X11Forwarding` 参数进行配置。

1. 编辑 `/etc/ssh/sshd_config` 配置文件，修改 `X11Forwarding` 参数：

```
X11Forwarding no
```

扫描检测

确保 `SSH` 的 `X11Forwarding` 配置正确。

1. 执行以下命令，验证 `SSH` 的 `X11Forwarding` 配置是否正确：

```
# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts |
 ↪  awk '{print $1}')" | grep -i x11forwarding
x11forwarding no
# grep -Ei '^\s*x11forwarding\s+yes' /etc/ssh/sshd_config
Nothing is returned
```

如果第一条命令执行后返回 `no` ，且第二条命令执行后，没有返回任何结果，则视为通过此项检查。

参考

1.51 确保 udf 文件系统的挂载被禁用

安全等级

- Level 1

描述

udf 文件系统类型是用于实现 ISO/IEC 13346 和 ECMA-167 规范的通用磁盘格式。这是一种开放的供应商文件系统类型，用于在广泛的媒体上存储数据。该文件系统类型是必需的，以支持编写 DVD 和较新的光盘格式。

删除不需要的文件系统类型的支持可以减少系统的本地攻击面。如果不需要此文件系统类型，请禁用它。

修复建议

目标：确保 udf 文件系统的挂载被禁用。

1. 执行以下命令，在 `/etc/modprobe.d/` 目录中编辑或创建一个以 `.conf` 结尾的文件，并添加配置。

```
# echo "install udf /bin/false" >> /etc/modprobe.d/udf.conf
# echo "blacklist udf" >> /etc/modprobe.d/udf.conf
```

2. 运行以下命令以卸载 udf 模块。

```
# modprobe -r udf
```

扫描检测

运行以下命令并验证输出是否符合预期。

1. 模块将如何加载。

   运行如下命令，若输出为 `install /bin/false` ，则视为通过此项检查。

```
# modprobe -n -v udf | grep "^install"
install /bin/false
```

2. 模块当前是否已加载。

   运行如下命令，若输出为空，则视为通过此项检查。

```
# lsmod | grep udf
<No  output>
```

3. 模块是否已列入黑名单。

   运行如下命令，若输出为 `/etc/modprobe.d/udf.conf:blacklist udf` ，则视为通过此项检查。

```
# grep -E "^blacklist[[:blank:]]*udf" /etc/modprobe.d/*
/etc/modprobe.d/udf.conf:blacklist     udf
```

参考

1.52 确保已禁用 cramfs 文件系统的挂载

安全等级

- Level 1

描述

cramfs 文件系统是压缩的只读 Linux 文件系统，用于嵌入式系统。cramfs 映像可以在不必先解压映像的情况下使用。

删除不需要的文件系统类型的支持可以减少系统本地攻击面。如果不需要这种文件系统类型，请禁用它。

修复建议

目标：确保 cramfs 文件系统的挂载被禁用。

1. 执行以下命令，在/etc/modprobe.d/目录中编辑或创建一个以.conf 结尾的文件，并添加配置。

```
# echo "install cramfs /bin/false" >> /etc/modprobe.d/cramfs.conf
# echo "blacklist cramfs" >> /etc/modprobe.d/cramfs.conf
```

2. 运行以下命令以卸载 cramfs 模块：

```
# modprobe -r cramfs
```

扫描检测

运行以下命令并验证输出是否符合预期。

1. 模块将如何被加载

```
# modprobe -n -v cramfs | grep "^install"
install /bin/false
```

2. 模块当前是否已加载

```
# lsmod | grep cramfs
<No output>
```

3. 模块是否被列入黑名单

```
# grep -E "^blacklist\s+cramfs" /etc/modprobe.d/*
blacklist cramfs
```

参考

1.53 确保已禁用 squashfs 文件系统的挂载

安全等级

- level 1

描述

squashfs 文件系统类型是一种压缩的只读 Linux 文件系统，常用于小型嵌入式系统。可以直接使用 squashfs 镜像而不需要解压。

删除不需要的文件系统类型的支持可以减少系统的受攻击面。如果不需要这种文件系统类型，请禁用。

由于 Snap 包使用 squashfs 作为压缩文件系统，禁用 squashfs 将导致 Snap 包不可用。Linux 发行版一般内置了 Snap 软件包。Snap 与传统的 Linux 软件包管理方法（如 APT 或 RPM）不同，后者在进行更新应用程序时需要根据每个 Linux 发行版对软件包特别适配，因此导致从开发到发布应用的时间变长。Snap 可以从任何来源获得，不依赖于应用商店，可以用于上游软件部署。

修复建议

1. 编辑或创建 `/etc/modprobe.d/squashfs.conf` 文件，并添加内容：

```
printf "install squashfs /bin/false
blacklist squashfs
" >> /etc/modprobe.d/squashfs.conf
```

2. 运行以下命令以卸载 squashfs 模块：

```
# modprobe -r squash
```

扫描检测

运行以下命令并验证输出是否如指示的那样。

1. 模块将如何被加载

```
# modprobe -n -v squashfs | grep "^install"
install /bin/false
```

2. 模块当前是否已加载

```
# lsmod | grep squashfs
<No output>
```

3. 模块是否被列入黑名单

```
# grep -E "^blacklist\s+squashfs" /etc/modprobe.d/*
/etc/modprobe.d/squashfs.conf:blacklist squashfs
```

若输出结果符合预期则视为通过检查。

参考

1.54 锁定或删除 bin、adm 用户

安全等级

- Level 1

描述

锁定 bin、adm 用户，避免特定的权限和任务被强制关闭。

修复建议

锁定或删除 bin、adm 用户。

执行以下命令，锁定 bin、adm 用户

```
usermod -L bin
usermod -L adm
```

扫描检测

执行以下命令，验证 bin、adm 用户是否被锁定:

```
# passwd -S bin | grep -E "bin\s+LK"
bin LK 2022-07-12 0 99999 7 -1 (Alternate authentication scheme in use.)

# passwd -S adm | grep -E "adm\s+LK"
adm LK 2022-02-12 0 99999 7 -1 (Alternate authentication scheme in use.)
```

如输出结果符合预期，则视为通过此项检查。

参考

2 logging-and-auditing

2.1 确保审计日志的文件权限被正确配置

安全等级

- Level 1

描述

审计信息（审计记录、审计设置、审计报告）含有系统配置、用户数据、登录信息、操作记录等敏感数据，需严格控制其读写权限，避免数据泄露或丢失风险。

通过对**审计日志**文件权限进行配置，确保其不被未授权的用户读取或修改。

修复建议

目标：确保审计日志文件的权限为 `0600` 或更低。

1. 执行以下命令来确定审计日志的路径：

```
# grep -iw log_file /etc/audit/auditd.conf
log_file = /var/log/audit/audit.log
```

2. 根据上一步得到的审计日志文件路径，执行以下命令，将其权限配置为 `0600` 或更低：

```
# chmod 0600 /var/log/audit/*
```

扫描检测

确保审计日志文件的权限为 `0600` 或更低。

1. 执行以下命令来确定审计日志的路径：

```
# grep -iw log_file /etc/audit/auditd.conf
log_file = /var/log/audit/audit.log
```

2. 根据上一步得到的审计日志文件路径，执行以下命令来检查审计日志文件的权限是否为 `0600` 或更低：

```
# stat -c "%n %a" /var/log/audit/*
/var/log/audit/audit.log 600
```

如结果为 `0600` 或更低，则视为通过此项检查。

参考

2.2 确保审计日志文件的所有者为已授权用户

安全等级

- Level 1

描述

审计信息（审计记录、审计设置、审计报告）含有系统配置、用户数据、登录信息、操作记录等敏感数据，需正确配置其所有者，避免数据泄露或丢失。

通过对审计日志文件的**所有者**进行配置，确保其只被已授权用户所拥有。

修复建议

目标：将审计日志文件的所有者配置为 `root` 用户。

1. 执行以下命令来确定审计日志的路径：

```
# grep -iw log_file /etc/audit/auditd.conf
log_file = /var/log/audit/audit.log
```

2. 根据上一步得到的审计日志文件路径，执行以下命令，将其所有者配置为 `root` 用户：

```
# chown root /var/log/audit/*
```

扫描检测

确保审计日志文件的所有者为 `root` 。

1. 执行以下命令来确定审计日志的路径：

```
# grep -iw log_file /etc/audit/auditd.conf
log_file = /var/log/audit/audit.log
```

2. 根据上一步得到的审计日志文件路径，执行以下命令来检查审计日志文件的所有者是否为 `root` ：

```
# stat -c "%n %U" /var/log/audit/*
/var/log/audit/audit.log root
```

如结果为 `root` ，则视为通过此项检查。

参考

2.3 确保审计日志文件的所属组为已授权的用户组

安全等级

- Level 1

描述

审计信息（审计记录、审计设置、审计报告）含有系统配置、用户数据、登录信息、操作记录等敏感数据，需正确配置其所属组，避免数据泄露或丢失风险。

通过对审计日志文件**所属组**进行配置，确保其只被已授权用户组所拥有。

修复建议

目标：将审计日志文件的所属组配置为 `adm` 用户组。

1. 执行以下命令来确定审计日志的路径：

```
# grep -iw ^log_file /etc/audit/auditd.conf
log_file = /var/log/audit/audit.log
```

2. 根据上一步得到的审计日志文件路径，执行以下命令，将其所属组配置为 `adm` 用户组：

```
# chown :adm /var/log/audit/
```

3. 将审计配置文件的 `log_group` 参数的值设置为 `adm` ，确保在新日志文件生成时，默认所属组为 `adm` ：

```
# sed -i '/^log_group/D' /etc/audit/auditd.conf
# sed -i /^log_file/a'log_group = adm' /etc/audit/auditd.conf
```

4. 向 `auditd` 进程发送信号，使配置文件生效：

```
# systemctl kill auditd -s SIGHUP
```

扫描检测

确保审计日志文件的所属组为 `adm` 。

1. 使用以下命令，验证 `auditd.conf` (审计配置文件) `log_file` 的值是否被正确设置为 `adm` 或 `root` ：

```
# grep -iw log_group /etc/audit/auditd.conf
log_group = adm
```

2. 使用以下命令，确定审计日志文件的路径:

```
$ sudo grep -iw log_file /etc/audit/auditd.conf
log_file = /var/log/audit/audit.log
```

3. 根据上一步得到的审计日志文件路径，执行以下命令，来验证审计日志文件的所属组是否为 `adm` 或 `root` ：

```
# stat -c "%n %G" /var/log/audit/*
/var/log/audit/audit.log root
```

参考

2.4 确保审计目录的权限小于 0750

安全等级

- Level 1

描述

审计信息（审计记录、审计设置、审计报告）含有系统配置、用户数据、登录信息、操作记录等敏感数据。需正确配置其所在目录的权限，避免非授权用户篡改或删除审计数据，破坏审计信息的准确性，影响系统管理人员对恶意活动的发现和判断。

通过对**审计目录**的权限进行配置，确保其不被非授权用户篡改或删除。

修复建议

目标：确保审计目录的权限小于 `0750`

1. 执行以下命令来确定审计目录的路径：

```
# grep -iw ^log_file /etc/audit/auditd.conf
log_file = /var/log/audit/audit.log
```

2. 根据上一步得到的审计目录路径，执行以下命令，将审计日志目录的权限配置为 `0750` 或更小：

```
# chmod -R g-w,o-rwx /var/log/audit
```

扫描检测

确保审计目录的权限小于 `0750`

1. 执行以下命令来确定审计目录的路径：

```
# grep -iw ^log_file /etc/audit/auditd.conf
log_file = /var/log/audit/audit.log
```

2. 根据上一步得到的审计目录路径，执行以下命令，检查审计目录的权限是否为 `0750` 或更小：

```
# stat -c "%n %a" /var/log/audit /var/log/audit/*
/var/log/audit 750
```

参考

2.5 确保审计配置文件的权限小于 0640

安全等级

- Level 1

描述

审计配置文件影响审计的内容、审计的事件、日志保存路径等。如不对其权限进行限制，未经授权的人员可能会篡改审计配置文件，阻止其对关键事件的审计。破坏审计信息的准确性，影响系统管理人员对恶意活动的发现和判断。

错误的审计配置也可能会导致审计日志过载，从而降低系统性能。

通过对**审计配置文件**的权限进行配置，确保其不被非授权用户篡改或删除。

修复建议

目标：确保审计配置文件的权限小于 `0640`

1. 设置 `/etc/audit/audit.rules`，`/etc/audit/rules.d/*`，`/etc/audit/auditd.conf` 文件的权限为 `0640` 或更低：

```
# chmod -R 0640 /etc/audit/audit*.{rules,conf} /etc/audit/rules.d/*
```

扫描检测

确保审计配置文件的权限小于 `0640`

1. 执行以下命令，确认 `/etc/audit/audit.rules`，`/etc/audit/rules.d/*`，`/etc/audit/auditd.conf` 文件的权限为 `0640` 或更低：

```
# ls -al /etc/audit/ /etc/audit/rules.d/
/etc/audit/:
-rw-r----- 1 root root 804 Nov 25 11:01 auditd.conf
-rw-r----- 1 root root 9128 Dec 27 09:56 audit.rules
-rw-r----- 1 root root 9373 Dec 27 09:56 audit.rules.prev
```

```
-rw-r----- 1 root root 127 Feb 7 2018 audit-stop.rules
drwxr-x--- 2 root root 4096 Dec 27 09:56 rules.d
/etc/audit/rules.d/:
-rw-r----- 1 root root 10357 Dec 27 09:56 stig.rules
```

参考

2.6 确保审计配置文件的所有者为已授权用户

安全等级

- Level 1

描述

审计配置文件影响审计的内容、审计的事件、日志保存路径等。如不对其所有者进行限制，未经授权的人员可能会篡改审计配置文件，阻止其对关键事件的审计。破坏审计信息的准确性，影响系统管理人员对恶意活动的发现和判断。

错误的审计配置也可能会导致审计日志过载，从而降低系统性能。

通过对审计配置文件的**所有者**进行配置，确保其不被非授权用户篡改或删除。

修复建议

目标：确保审计配置文件所有者为 `root`

1. 设置 `/etc/audit/audit.rules` , `/etc/audit/rules.d/*` , `/etc/audit/auditd.conf` 文件的所有者为 `root` ：

```
# chown root /etc/audit/audit*.{rules,conf} /etc/audit/rules.d/*
```

扫描检测

确保审计配置文件所有者为 `root`

1. 执行以下命令，确认 `/etc/audit/audit.rules` , `/etc/audit/rules.d/*` , `/etc/audit/auditd.conf` 文件的所有者为 `root` ：

```
# ls -al /etc/audit/ /etc/audit/rules.d/
/etc/audit/:
drwxr-x--- 3 root root 4096 Nov 25 11:02 .
drwxr-xr-x 130 root root 12288 Dec 19 13:42 ..
-rw-r----- 1 root root 804 Nov 25 11:01 auditd.conf
```

```
-rw-r----- 1 root root 9128 Dec 27 09:56 audit.rules
-rw-r----- 1 root root 9373 Dec 27 09:56 audit.rules.prev
-rw-r----- 1 root root 127 Feb 7 2018 audit-stop.rules
drwxr-x--- 2 root root 4096 Dec 27 09:56 rules.d
/etc/audit/rules.d/:
drwxr-x--- 2 root root 4096 Dec 27 09:56 .
drwxr-x--- 3 root root 4096 Nov 25 11:02 ..
-rw-r----- 1 root root 10357 Dec 27 09:56 stig.rules
```

如所有文件的所有者均为 `root` ，则视为通过此项检查。

参考

2.7 确保审计配置文件的所属组为已授权的用户组

安全等级

- Level 1

描述

审计配置文件影响审计的内容、审计的事件、日志保存路径等。如不对其所属组进行限制，未经授权的人员可能会篡改审计配置文件，阻止其对关键事件的审计。破坏审计信息的准确性，影响系统管理人员对恶意活动的发现和判断。

错误的审计配置也可能会导致审计日志过载，从而降低系统性能。

通过对审计配置文件的**所属组**进行配置，确保其不被非授权用户篡改或删除。

修复建议

目标：确保审计配置文件的所属组为 `root`

1. 设置 `/etc/audit/audit.rules` , `/etc/audit/rules.d/*` , `/etc/audit/auditd.conf` 文件的所属组为 `root` ：

```
# chown :root /etc/audit/audit*.{rules,conf} /etc/audit/rules.d/*
```

扫描检测

确保审计配置文件所属组为 `root`

1. 执行以下命令，确认 `/etc/audit/audit.rules` , `/etc/audit/rules.d/*` , `/etc/audit/auditd.conf` 文件的所属组为 `root` ：

```
# ls -al /etc/audit/ /etc/audit/rules.d/
/etc/audit/:
-rw-r----- 1 root root 804 Nov 25 11:01 auditd.conf
-rw-r----- 1 root root 9128 Dec 27 09:56 audit.rules
-rw-r----- 1 root root 9373 Dec 27 09:56 audit.rules.prev
```

```
-rw-r----- 1 root root 127 Feb 7 2018 audit-stop.rules
drwxr-x--- 2 root root 4096 Dec 27 09:56 rules.d
/etc/audit/rules.d/:
-rw-r----- 1 root root 10357 Dec 27 09:56 stig.rules
```

如所有文件的所属组均为 `root` ，则视为通过此项检查。

参考

2.8 确保审计工具的权限为 0755 或更低

安全等级

- Level 1

描述

审计工具包括但不限于：查看和操作审计信息所需的供应商或开源工具，如自定义查询和报告生成器等。因此，对审计工具的保护是非常必要的，以防未经授权的用户对审计信息进行提取或操作。

审计工具的权限必须为 `0755` 或更低。

修复建议

目标：确保审计工具的权限为 `0755` 或更低。

1. 使用以下命令将审计工具的权限设置为 `0755` 或更低：

```
# chmod 0755 /sbin/auditctl
# chmod 0755 /sbin/aureport
# chmod 0755 /sbin/ausearch
# chmod 0755 /sbin/autrace
# chmod 0755 /sbin/auditd
# chmod 0755 /sbin/augenrules
```

扫描检测

确保审计工具的权限为 `0755` 或更低。

1. 执行以下命令，检查审计工具的权限是否为 `0755` 或更低：

```
# stat -c "%n %a" /sbin/auditctl /sbin/aureport /sbin/ausearch /sbin/autrace /sbin/
 ↪  auditd /sbin/augenrules
/sbin/auditctl 755
/sbin/aureport 755
```

```
/sbin/ausearch 755
/sbin/autrace 750
/sbin/auditd 755
/sbin/augenrules 755
```

如所有审计工具的权限均为 `0755` 或更低，则视为通过此项检查。

参考

2.9 确保审计工具属于 root 用户

安全等级

- Level 1

描述

审计工具包括但不限于：查看和操作审计信息所需的供应商或开源工具，如自定义查询和报告生成器等。因此，对审计工具的保护是非常必要的，以防未经授权的用户对审计信息进行提取或操作。

审计工具的所有者应为 `root`

修复建议

目标：确保审计工具的所有者为 `root` 。

1. 使用以下命令将审计工具的所有者设置为 `root` ：

```
# chown root /sbin/auditctl
# chown root /sbin/aureport
# chown root /sbin/ausearch
# chown root /sbin/autrace
# chown root /sbin/auditd
# chown root /sbin/augenrules
```

扫描检测

确保审计工具的所有者为 `root` 。

1. 执行以下命令，检查审计工具的所有者是否为 `root` ：

```
# stat -c "%n %U" /sbin/auditctl /sbin/aureport /sbin/ausearch /sbin/autrace /sbin/
 ↪  auditd /sbin/augenrules
/sbin/auditctl root
/sbin/aureport root
```

```
/sbin/ausearch root
/sbin/autrace root
/sbin/auditd root
/sbin/augenrules root
```

如所有审计工具的所有者均为 `root` ，则视为通过此项检查。

参考

2.10 确保审计工具属于 root 用户组

安全等级

- Level 1

描述

审计工具包括但不限于：查看和操作审计信息所需的供应商或开源工具，如自定义查询和报告生成器等。因此，对审计工具的保护是非常必要的，以防未经授权的用户对审计信息进行提取或操作。

审计工具的所属组应为 `root`

修复建议

目标：确保审计工具的所属组为 `root` 。

1. 使用以下命令将审计工具的所属组设置为 `root` ：

```
# chown :root /sbin/auditctl
# chown :root /sbin/aureport
# chown :root /sbin/ausearch
# chown :root /sbin/autrace
# chown :root /sbin/auditd
# chown :root /sbin/augenrules
```

扫描检测

确保审计工具的所属组为 `root` 。

1. 执行以下命令，检查审计工具的所属组是否为 `root` ：

```
# stat -c "%n %G" /sbin/auditctl /sbin/aureport /sbin/ausearch /sbin/autrace /sbin/
 ↪  auditd /sbin/augenrules
/sbin/auditctl root
/sbin/aureport root
```

```
/sbin/ausearch root
/sbin/autrace root
/sbin/auditd root
/sbin/augenrules root
```

如所有审计工具的所属组均为 `root` ，则视为通过此项检查。

参考

2.11 确保使用加密机制来保护审计工具的完整性

安全等级

- Level 1

描述

审计工具包括但不限于：查看和操作审计信息所需的供应商或开源工具，如自定义查询和报告生成器等。因此，对审计工具的保护是非常必要的，以防未经授权的用户对审计信息进行提取或操作。

攻击者常替换审计工具或向现有工具中注入代码，从而将审计日志中的信息隐藏或删除。这种情况并不少见。

要解决此风险，审计工具必须进行加密签名，以便识别审计工具是否有被修改、操纵或替换。

修复建议

目标：使用加密机制保护审计工具的完整性

1. 向配置文件 `/etc/aide/aide.conf` 中添加或更新以下配置，对审计工具进行加密：

```
# Audit Tools
/sbin/auditctl p+i+n+u+g+s+b+acl+xattrs+sha512
/sbin/auditd p+i+n+u+g+s+b+acl+xattrs+sha512
/sbin/ausearch p+i+n+u+g+s+b+acl+xattrs+sha512
/sbin/aureport p+i+n+u+g+s+b+acl+xattrs+sha512
/sbin/autrace p+i+n+u+g+s+b+acl+xattrs+sha512
/sbin/augenrules p+i+n+u+g+s+b+acl+xattrs+sha512
```

如没有此配置文件，需新建。

扫描检测

确保审计工具有加密机制。

1. 执行以下命令，检查 `/etc/aide/aide.conf` 配置文件中，审计工具是否有加密机制：

```
# grep -E '(\/sbin\/(audit|au))' /etc/aide/aide.conf
/sbin/auditctl p+i+n+u+g+s+b+acl+xattrs+sha512
/sbin/auditd p+i+n+u+g+s+b+acl+xattrs+sha512
/sbin/ausearch p+i+n+u+g+s+b+acl+xattrs+sha512
/sbin/aureport p+i+n+u+g+s+b+acl+xattrs+sha512
/sbin/autrace p+i+n+u+g+s+b+acl+xattrs+sha512
/sbin/augenrules p+i+n+u+g+s+b+acl+xattrs+sha512
```

如所有审计工具均有加密机制，则视为通过此项检查。

参考

2.12 确保已安装 rsyslog

安全等级

- Level 1

描述

rsyslog 软件是原 syslogd 守护程序的替代品，其相比于 syslogd 做了很多改进，添加了如：面向连接（即 TCP）的日志传输、将日志记录到数据库，以及在与中央日志服务器交互中对日志数据进行加密等特性及功能。

以上多种新特性与功能，都证明了安装和配置 rsyslog 软件包的合理性及必要性。

修复建议

目标：安装 `rsyslog` 软件包。

1. 运行以下命令安装 `rsyslog` ：

```
# yum install rsyslog -y
```

扫描检测

确保已安装 rsyslog。

1. 执行以下命令，检查 rsyslog 软件是否正确安装：

```
# rpm -q rsyslog
rsyslog-<version>
```

`<version>` 为版本号，如： `rsyslog-8.2102.0-5.an8.x86_64`

参考

2.13 确保 rsyslog 服务已启用

安全等级

- Level 1

描述

rsyslog 软件是原 syslogd 守护程序的替代品，其相比于 syslogd 做了很多改进，添加了如：面向连接（即 TCP）的日志传输、将日志记录到数据库，以及在与中央日志服务器交互中对日志数据进行加密等特性及功能。

在安装了 rsyslog 软件后，应正确的激活并启用该服务。

修复建议

目标：确保 rsyslog 服务已启用。

1. 运行以下命令启用 rsyslog 服务：

```
# systemctl --now enable rsyslog
```

扫描检测

确保 rsyslog 服务已启用。

1. 执行以下命令，检查 rsyslog 服务是否启用：

```
# systemctl is-enabled rsyslog
enabled
```

如输出结果为 `enabled` ，则视为通过此项检查。

参考

2.14 确保正确配置了 rsyslog 默认文件权限

安全等级

- Level 1

描述

对日志文件进行正确有效的权限管理是尤为重要的，通过权限管理可以确保敏感数据被正确的归档和保护。

rsyslog 服务会生成新的日志文件，以下配置的作用在于控制这些新生成的日志文件权限。

修复建议

目标：正确配置 rsyslog 默认文件权限。

1. 编辑 `/etc/rsyslog.conf` 和 `/etc/rsyslog.d/*.conf` 文件，将 `$FileCreateMode` 的值设置为 `0640` 或更低：

```
# echo "\$FileCreateMode 0640" >> /etc/rsyslog.conf
# echo "\$FileCreateMode 0640" >> /etc/rsyslog.d/listen.conf
```

扫描检测

确保正确配置 rsyslog 默认文件权限。

1. 执行以下命令，验证配置文件中 `$FileCreateMode` 的值是否为 `0640` 或更低：

```
# grep ^\$FileCreateMode /etc/rsyslog.conf /etc/rsyslog.d/*.conf
/etc/rsyslog.conf:$FileCreateMode 0640
/etc/rsyslog.d/listen.conf:$FileCreateMode 0640
```

如 `$FileCreateMode` 的值为 `0640` 或更低，则视为通过此项检查。

参考

2.15 确保 rsyslog 配置了远程日志主机

安全等级

- Level 2

描述

将日志数据存储在远程主机上可以保护日志的完整性，使其免受本地攻击。

如果攻击者获得了本地系统的 root 权限，他们就可以篡改或删除存储在本地系统中的日志数据。而远程日志主机上的数据则不会受到影响。可通过远程日志主机轻松的对本地数据进行比对和恢复。

rsyslog 工具支持将其收集的日志发送到运行 syslogd(8) 的远程日志主机或接收来自远程主机的信息，从而减少管理开销。

修复建议

目标：正确配置 rsyslog 远程日志主机。

1. 编辑 `/etc/rsyslog.conf` 和 `/etc/rsyslog.d/*.conf` 文件，并添加以下代码 (其中 loghost.example.com 是您的中央日志主机的名称，根据实际情况进行修改。):

```
*.* @@loghost.example.com
```

如没有此配置文件，需新建。

扫描检测

确保正确配置 rsyslog 远程日志主机。

1. 运行以下命令，检查 `/etc/rsyslog.conf` 和 `/etc/rsyslog.d/*.conf` 文件是否正确配置了远程日志主机 (其中 loghost.example.com 是您的中央日志主机的名称，根据实际情况进行修改。):

```
# grep "^*.*[^I][^I]*@" /etc/rsyslog.conf /etc/rsyslog.d/*.conf
*.* @@loghost.example.com
```

参考

2.16 确保配置 journald 向 rsyslog 发送日志

安全等级

- Level 1

描述

来自 journald 的数据可以存储在易失性内存中，也可以在服务器上本地保存。使用 rsyslog 服务可以提供一致的日志收集和导出方式。

修复建议

目标：正确配置 journald 向 rsyslog 发送日志。

1. 编辑 `/etc/systemd/journald.conf` 文件，增加如下代码：

```
ForwardToSyslog=yes
```

扫描检测

确保配置 journald 向 rsyslog 发送日志

1. 检查 `/etc/systemd/journald.conf` ，确认日志是否被发送到 rsyslog：

```
# grep -e ^\s*ForwardToSyslog /etc/systemd/journald.conf
ForwardToSyslog=yes
```

参考

2.17 确保 journald 日志压缩功能正确启用

安全等级

- Level 1

描述

未经压缩的日志文件可能会填满硬盘，导致主机性能下降，资源不可用等问题。

journald 有大型日志文件压缩功能，应正确配置并启用该功能，以避免日志文件填满硬盘。

修复建议

目标：正确配置 journald 大型日志压缩功能。

1. 编辑 `/etc/systemd/journald.conf` 文件，增加如下代码：

```
Compress=yes
```

扫描检测

确认 journald 日志压缩功能正确配置

1. 检查 `/etc/systemd/journald.conf` 文件，确认 journald 日志压缩功能被正确配置：

```
# grep -e ^\s*Compress /etc/systemd/journald.conf
Compress=yes
```

参考

2.18 确保 journald 日志文件写入硬盘功能正确开启

安全等级

- Level 1

描述

journald 的日志数据既可以保存在内存中，也可以保存在本地硬盘中。内存中的日志在系统崩溃或重启后将会丢失。将日志数据写入硬盘，以保证数据的安全性。

修复建议

目标：正确配置 journald 日志写入硬盘功能。

1. 编辑 `/etc/systemd/journald.conf` 文件，增加如下代码：

```
Storage=persistent
```

扫描检测

确认 journald 日志写入硬盘功能正确配置

1. 检查 `/etc/systemd/journald.conf` 文件，确认 journald 日志写入硬盘功能被正确配置。

```
# grep -e ^\s*Storage /etc/systemd/journald.conf
Storage=persistent
```

参考

2.19 确保审计工具已安装

安全等级

- Level 1

描述

审计工具是 Linux 审计系统的用户空间组件。它负责将审计记录写入磁盘，使管理员能够确定是否正在发生对其系统的未经授权的访问

审计工具应该在系统上安装

修复建议

目标：确保审计工具已安装。

1. 使用以下命令安装审计工具：

```
# dnf install audit audit-libs
```

扫描检测

1. 执行以下命令，检查审计工具是否安装：

```
# rpm -q audit audit-libs
audit-<version>
audit-libs-<version>
```

为软件版本信息。如输出结果符合预期，则视为通过此项检查。

参考

2.20 确保已启用审计服务

安全等级

- Level 3

描述

审计工具包括但不限于：查看和操作审计信息所需的供应商或开源工具，如自定义查询和报告生成器等。因此，启用审计服务是非常必要的，以防未经授权的用户对审计信息进行提取或操作。

审计服务已启用

修复建议

目标：确保审计服务已启用。

1. 使用以下命令启用审计服务：

```
# systemctl --now enable auditd
```

扫描检测

1. 执行以下命令，检查审计服务是否已启用：

```
# systemctl is-enabled auditd
enabled
```

输出结果为 enabled ，那么审计服务已启用，则视为通过此项检查。

参考

2.21 确保收集用户的文件删除事件

安全等级

- Level 3

描述

对删除文件的操作进行审计记录。因 ARM 架构的部分规则与 X86 不一致，需在操作前使用 `arch` 或 `uname -m` 命令，对硬件架构进行判断，并使用相应的规则进行加固修复，否则会导致审计规则报错，无法生效：- X86 架构：

```
# arch
x86_64

# uname -m
x86_64
```

- ARM 架构：

```
# arch
aarch64

# uname -m
aarch64
```

修复建议

目标：对删除文件的操作进行审计记录。

运行以下命令，配置审计服务，确保收集用户的文件删除事件：

- X86 架构：

```
# echo -e "\n-a always,exit -F arch=b64 -S unlink -S unlinkat -S rename -S renameat -F
 ↪  auid>=1000 -F auid!=4294967295 -k delete\n-a always,exit -F arch=b32 -S unlink -S
 ↪  unlinkat -S rename -S renameat -F auid>=1000 -F auid!=4294967295 -k delete\n" >> /
 ↪  etc/audit/rules.d/audit.rules
```

- ARM 架构：

```
# echo -e "\n-a always,exit -F arch=b64 -S unlinkat -S renameat -F auid>=1000 -F auid!
 ↪  =4294967295 -k delete\n-a always,exit -F arch=b32 -S unlink -S unlinkat -S rename
 ↪  -S renameat -F auid>=1000 -F auid!=4294967295 -k delete\n" >> /etc/audit/rules.d/
 ↪  audit.rules
```

扫描检测

确保收集用户的文件删除事件。

1. 执行以下命令，检查文件删除审计收集是否正确配置：

- X86 架构：

```
# grep -P "\-a\salways\,exit\s\-F\sarch=b(32|64)\s\-S\sunlink\s\-S\sunlinkat\s\-
 ↪  S\srename\s\-S\srenameat\s\-F\sauid>=1000\s\-F\sauid!=4294967295\s-k\sdelete" /
 ↪  etc/audit/rules.d/*.rules /etc/audit/*.rules

/etc/audit/rules.d/audit.rules:-a always,exit -F arch=b64 -S unlink -S unlinkat -S
 ↪  rename -S renameat -F auid>=1000 -F auid!=4294967295 -k delete
/etc/audit/rules.d/audit.rules:-a always,exit -F arch=b32 -S unlink -S unlinkat -S
 ↪  rename -S renameat -F auid>=1000 -F auid!=4294967295 -k delete
/etc/audit/audit.rules:-a always,exit -F arch=b64 -S unlink -S unlinkat -S rename -S
 ↪  renameat -F auid>=1000 -F auid!=4294967295 -k delete
/etc/audit/audit.rules:-a always,exit -F arch=b32 -S unlink -S unlinkat -S rename -S
 ↪  renameat -F auid>=1000 -F auid!=4294967295 -k delete
```

- ARM 架构：

```
# grep -P "\-a\salways\,exit\s\-F\sarch=b(32|64)\s\-S\sunlink.*-k\sdelete" /etc/audit/
 ↪  rules.d/*.rules /etc/audit/*.rules

/etc/audit/rules.d/audit.rules:-a always,exit -F arch=b64 -S unlinkat -S renameat -F
 ↪  auid>=1000 -F auid!=4294967295 -k delete
```

```
/etc/audit/rules.d/audit.rules:-a always,exit -F arch=b32 -S unlink -S unlinkat -S
 ↪  rename -S renameat -F auid>=1000 -F auid!=4294967295 -k delete
/etc/audit/audit.rules:-a always,exit -F arch=b64 -S unlinkat -S renameat -F
 ↪  auid>=1000 -F auid!=4294967295 -k delete
/etc/audit/audit.rules:-a always,exit -F arch=b32 -S unlink -S unlinkat -S rename -S
 ↪  renameat -F auid>=1000 -F auid!=4294967295 -k delete
```

2. 执行以下命令，检查文件删除审计收集是否正确加载：

- X86 架构：

```
# auditctl -l | grep -P "^\-a\s+always,exit\s+\-F\s+arch=b64\s+\-
 ↪  S\s+rename,unlink,unlinkat,renameat\s+\-F\s+auid>=1000\s+\-F\s+auid!=-1\s+\-
 ↪  F\s+key=delete" && auditctl -l | grep -P "^\-a\s+always,exit\s+\-F\s+arch=b32\s+\-
 ↪  S\s+unlink,rename,unlinkat,renameat\s+\-F\s+auid>=1000\s+\-F\s+auid!=-1\s+\-
 ↪  F\s+key=delete"
-a always,exit -F arch=b64 -S rename,unlink,unlinkat,renameat -F auid>=1000 -F auid!
 ↪  =-1 -F key=delete
-a always,exit -F arch=b32 -S unlink,rename,unlinkat,renameat -F auid>=1000 -F auid!
 ↪  =-1 -F key=delete
```

- ARM 架构：

```
# auditctl -l | grep -P "^\-a\s+always,exit\s+\-F\s+arch=b64\s+\-
 ↪  S\s+unlinkat,renameat\s+-F\s+auid>=1000\s+-F\s+auid!=-1\s+-F\s+key=delete" &&
 ↪  auditctl -l | grep -P "^\-a\s+always,exit\s+\-F\s+arch=b32\s+\-
 ↪  S\s+unlink,rename,unlinkat,renameat\s+\-F\s+auid>=1000\s+\-F\s+auid!=-1\s+\-
 ↪  F\s+key=delete"
-a always,exit -F arch=b64 -S unlinkat,renameat -F auid>=1000 -F auid!=-1 -F
 ↪  key=delete
-a always,exit -F arch=b32 -S unlink,rename,unlinkat,renameat -F auid>=1000 -F auid!
 ↪  =-1 -F key=delete
```

如输出结果符合预期，则视为通过此项检查。

参考

2.22 确保收集对系统管理范围（sudoers）的更改

安全等级

- Level 3

描述

对系统管理范围（sudoers）的更改操作进行审计记录

修复建议

目标：确保收集对系统管理范围（sudoers）的更改。

运行以下命令，配置审计服务，确保收集对系统管理范围（sudoers）的更改：

```
# echo -e "-w /etc/sudoers -p wa -k scope\n-w /etc/sudoers.d/ -p wa -k scope" >> /etc/
 ↪  audit/rules.d/audit.rules
```

扫描检测

确保收集对系统管理范围（sudoers）的更改。

1. 执行以下命令，检查对系统管理范围（sudoers）的审计收集是否正确配置：

```
# grep -E "\-w\s/etc/sudoers\s\-p\swa\s\-k\sscope
\-w\s/etc/sudoers.d/\s\-p\swa\s\-k\sscope" /etc/audit/rules.d/*.rules /etc/audit/
 ↪  *.rules

/etc/audit/rules.d/audit.rules:-w /etc/sudoers -p wa -k scope
/etc/audit/rules.d/audit.rules:-w /etc/sudoers.d/ -p wa -k scope
/etc/audit/audit.rules:-w /etc/sudoers -p wa -k scope
/etc/audit/audit.rules:-w /etc/sudoers.d/ -p wa -k scope
```

2. 执行以下命令，检查对系统管理范围（sudoers）的审计收集是否正确运行：

```
# auditctl -l | grep -P "^\-w\s+\/etc\/sudoers\s+\-p\s+wa\s+\-k\s+scope" && auditctl -
  ↪ l | grep -P "^\-w\s+\/etc\/sudoers.d\s+\-p\s+wa\s+\-k\s+scope"
-w /etc/sudoers -p wa -k scope
-w /etc/sudoers.d -p wa -k scope
```

如输出结果符合预期，则视为通过此项检查。### 参考

2.23 确保收集修改用户/组信息的事件

安全等级

- Level 3

描述

收集修改用户/组信息的事件

修复建议

目标：确保收集修改用户/组信息的事件。

运行以下命令，配置审计服务，确保收集对用户/组信息的修改事件：

```
# echo -e "-w /etc/group -p wa -k identity\n-w /etc/passwd -p wa -k identity\n-w /etc/
 ↪  gshadow -p wa -k identity\n-w /etc/shadow -p wa -k identity\n-w /etc/security/
 ↪  opasswd -p wa -k identity\n" >> /etc/audit/rules.d/audit.rules
```

扫描检测

确保收集修改用户/组信息的事件。

1. 执行以下命令，检查对用户/组信息的修改审计收集是否正确配置：

```
# grep -E "\-w\s/etc/group\s\-p\swa\s\-k\sidentity
\-w\s/etc/passwd\s\-p\swa\s\-k\sidentity
\-w\s/etc/gshadow\s\-p\swa\s\-k\sidentity
\-w\s/etc/shadow\s\-p\swa\s\-k\sidentity
\-w\s/etc/security/opasswd\s\-p\swa\s\-k\sidentity" /etc/audit/rules.d/*.rules /etc/
 ↪  audit/*.rules

/etc/audit/rules.d/audit.rules:-w /etc/group -p wa -k identity
/etc/audit/rules.d/audit.rules:-w /etc/passwd -p wa -k identity
/etc/audit/rules.d/audit.rules:-w /etc/gshadow -p wa -k identity
```

```
/etc/audit/rules.d/audit.rules:-w /etc/shadow -p wa -k identity
/etc/audit/rules.d/audit.rules:-w /etc/security/opasswd -p wa -k identity
/etc/audit/audit.rules:-w /etc/group -p wa -k identity
/etc/audit/audit.rules:-w /etc/passwd -p wa -k identity
/etc/audit/audit.rules:-w /etc/gshadow -p wa -k identity
/etc/audit/audit.rules:-w /etc/shadow -p wa -k identity
/etc/audit/audit.rules:-w /etc/security/opasswd -p wa -k identity
```

2. 执行以下命令，检查对用户/组信息的修改审计收集是否正确运行：

```
# auditctl -l | grep "\-w /etc/group -p wa -k identity
-w /etc/passwd -p wa -k identity
-w /etc/gshadow -p wa -k identity
-w /etc/shadow -p wa -k identity
-w /etc/security/opasswd -p wa -k identity"

-w /etc/group -p wa -k identity
-w /etc/passwd -p wa -k identity
-w /etc/gshadow -p wa -k identity
-w /etc/shadow -p wa -k identity
-w /etc/security/opasswd -p wa -k identity
```

如输出结果符合预期，则视为通过此项检查。

参考

2.24 确保记录成功或不成功使用 chsh 命令

安全等级

- Level 3

描述

在成功/不成功使用 chsh 命令时，操作系统必须生成的相应的审计记录。

修复建议

任何成功/不成功的使用 chsh 命令都需要配置审计系统来生成审计事件。

1. 运行以下命令，在 `/etc/audit/rules.d/stig.rules` 文件中添加或更新审计规则：

```
# grep -qs "\-a always,exit -F path=/usr/bin/chsh -F perm=x -F auid>=1000 -F auid!
 ↪  =4294967295 -k priv_cmd" /etc/audit/rules.d/stig.rules || echo -e "-a always,exit
 ↪  -F path=/usr/bin/chsh -F perm=x -F auid>=1000 -F auid!=4294967295 -k priv_cmd" >>
 ↪  /etc/audit/rules.d/stig.rules
```

2. 执行以下命令，加载审计规则：

```
# augenrules --load
```

扫描检查

1. 执行以下命令，检测 chsh 命令的审计规则是否正确写入审计规则文件：

```
# grep chsh /etc/audit/rules.d/*.rules /etc/audit/*.rules

/etc/audit/rules.d/stig.rules:-a always,exit -F path=/usr/bin/chsh -F perm=x -F
 ↪  auid>=1000 -F auid!=4294967295 -k priv_cmd
/etc/audit/audit.rules:-a always,exit -F path=/usr/bin/chsh -F perm=x -F auid>=1000 -F
 ↪  auid!=4294967295 -k priv_cmd
```

2. 执行以下命令，检测 chsh 命令的审计规则是否已正确加载：

```
# auditctl -l | grep chsh

-a always,exit -S all -F path=/usr/bin/chsh -F perm=x -F auid>=1000 -F auid!=-1 -F
 ↪ key=priv_cmd
```

如输出结果符合预期，则视为通过此项检查。

参考

2.25 确保审计日志不会自动删除

安全等级

- Level 3

描述：

出于安全考虑，维护长时间的审计历史记录利大于弊。在 `/etc/audit/auditd.conf` 中，`max_log_file_action` 配置项指定 `max_log_file` 容量达到设定的值时采取的动作。当该配置项设置为 `keep_logs` 时，系统将循环日志文件但会忽略 num_logs 参数（也就是不删除日志文件）。

修复建议

在 `/etc/audit/auditd.conf` 中将 `max_log_file_action` 选项设置为 `keep_logs` ：

```
max_log_file_action = keep_logs
```

扫描检测

运行以下命令，并验证是否有如下输出：

```
# grep max_log_file_action /etc/audit/auditd.conf
max_log_file_action = keep_logs
```

参考：

2.26 确保审计系统内存配置信息和磁盘配置信息相同

安全等级

- Level 1

描述

审计系统在磁盘上的系统配置信息可能和内存中的系统配置信息不相同。

通过对审计系统磁盘配置信息和内存配置信息的一致性进行检查，保证审计系统正常运行。

修复建议

目标：确保内存中的配置信息和磁盘上存放的配置信息是一致的。

1. 运行以下命令来合并和加载所有的规则：

```
# augenrules --load
```

2. 检查是否需要重启：

```
if [[ $(auditctl -s | grep "enabled") =~ "2" ]]; then
    echo "Reboot required to load rules";
fi
```

扫描检测

目标：确保内存中的配置信息和磁盘上存放的配置信息是一致的。

1. 检查 `/etc/audit/rules.d` 目录下的规则改动是否已合并至 `/etc/audit/audit.rules` ：

```
# augenrules --check

/usr/sbin/augenrules: No change
```

如返回 `Rules have changed and should be updated` 则需执行 `augenrules --load` 命令来合并并加载所有规则。

2. 在合并规则集之后，再次进行测试：

```
# augenrules --check | grep -Psiq "No\s+change" && echo 'pass' || echo 'fail'


pass
```

如输出 pass，则视为通过此项检查。

参考

2.27 确保开启防火墙日志记录功能

安全等级

- Level 3

描述

`firewalld` 的默认配置是不记录日志，可通过对配置文件的修改来开启日志记录功能。

日志记录下防火墙过滤时拒绝的非法 ip，有利于日后排查和细化防火墙规则。也可主动把这些有攻击性的 ip 加入到黑名单，防患于未然。但开启防火墙日志可能会增加系统资源消耗，请按实际需求决定是否开启此功能。

修复建议

目标：开启防火墙日志记录功能

1. 修改 `firewalld` 配置文件，添加或修改 `LogDenied` 参数，以开启防火墙日志功能。

```
# vim /etc/firewalld/firewalld.conf
LogDenied=all
```

2. 重启 `firewalld` 服务，使日志配置生效

```
systemctl restart firewalld.service
```

扫描检测

1. 使用以下命令，检查防火墙日志是否正确启用：

```
# firewall-cmd --get-log-denied
all
```

如执行结果为 `all` 则视为通过此项检查。

参考

2.28 确保收集登录和注销事件

安全等级

- Level 3

描述

创建审计规则，收集用户的登录和注销事件。

记录登录系统的用户，有助于审查是否有未授权或非法用户入侵。此功能会增加系统负载和占用磁盘空间，请按实际需求决定是否开启此功能。

修复建议

确保收集登录和注销事件

1. 在/etc/audit/rules.d/ 目录下，创建.rules 文件，并写入以下审计规则：

```
# echo "-w /var/log/lastlog -p wa -k logins" >> /etc/audit/rules.d/audit-root.rules
# echo "-w /var/run/faillock -p wa -k logins" >> /etc/audit/rules.d/audit-root.rules
```

2. 执行以下命令，加载审计规则：

```
# augenrules --load
```

扫描检查

1. 执行以下命令，检查审计规则是否正确写入配置文件：

```
# grep -P "(lastlog|faillock)" /etc/audit/rules.d/*.rules /etc/audit/*.rules

/etc/audit/rules.d/audit-root.rules:-w /var/log/lastlog -p wa -k logins
/etc/audit/rules.d/audit-root.rules:-w /var/run/faillock -p wa -k logins
/etc/audit/audit.rules:-w /var/log/lastlog -p wa -k logins
/etc/audit/audit.rules:-w /var/run/faillock -p wa -k logins
```

2. 执行以下命令，审计规则是否正确加载：

```
# auditctl -l | grep -P "(lastlog|faillock)"


-w /var/log/lastlog -p wa -k logins
-w /var/run/faillock -p wa -k logins
```

如输出结果符合预期，则视为通过此项检查。

参考

2.29 确保收集 sudo 日志

安全等级

- Level 3

描述

启用 sudo 日志可以审计用户在使用 `sudo` 时执行了什么命令等相关信息。此功能会增加系统负载和占用磁盘空间，请按实际需求决定是否开启此功能。

修复建议

确保收集 sudo 日志

1. 使用 `visudo` 命令，在 `/etc/sudoers` 中添加日志规则：

```
# visudo

Defaults logfile=/var/log/sudo.log
```

扫描检查

1. 执行以下命令，检查审计规则是否正确写入配置文件：

```
# grep -Ps "^Defaults\slogfile\=.*\.log$" /etc/sudoers

Defaults logfile=/var/log/sudo.log
```

如输出结果符合预期，则视为通过此项检查。

参考

2.30 确保收集 sudo 日志的改动记录

安全等级

- Level 3

描述

创建审计规则，收集 sudo 日志的改动记录。

开启此审计规则，能够有效记录非授权用户篡改或删除日志记录，方便后期进行排查。此功能会增加系统负载和占用磁盘空间，请按实际需求决定是否开启此功能。

修复建议

确保收集 sudo 日志的改动记录

1. 首先确认当前系统环境下是否配置了 `sudolog` ，如未配置，则无法进行审计，可跳过之后步骤：

```
# grep -r logfile /etc/sudoers* | sed -e 's/.*logfile=//;s/,? .*//' -e 's/"//g'


/var/log/sudo.log
```

   - 如返回了 `sudolog` 路径，则继续进行修复，否则请暂停本项修复。
   - 可参考 `2.29-ensure-sudo-log-are-collected.md` 配置 sudolog 路径后，再进行本项修复。

2. 执行以下命令，添加审计规则：

```
sudoLogfile=$(grep -r logfile /etc/sudoers* | sed -e 's/.*logfile=//;s/,? .*//' -e
 ↪  's/"//g')


printf "
-w $sudoLogfile -p wa -k sudo_log_file
" >> /etc/audit/rules.d/50-sudo.rules
```

   - 其中 `$sudoLogfile` 为第 1 步查询到的 `sudolog` 路径，可根据实际情况进行修改替换。

3. 执行以下命令，加载审计规则：

```
# augenrules --load
```

扫描检查

1. 首先确认当前系统环境下是否配置了 `sudolog` ，如未配置，则视为本项检测失败，可跳过之后步骤：

```
# grep -r logfile /etc/sudoers* | sed -e 's/.*logfile=//;s/,? .*//' -e 's/"//g'

/var/log/sudo.log
```

- 如返回了 `sudolog` 路径，则继续进行之后的检测步骤，否则视为本项检测失败。
- 可参考 `2.29-ensure-sudo-log-are-collected.md` 配置 sudolog 路径后，再进行本项检测。

2. 执行以下命令，检查审计规则是否正确写入配置文件：

```
# sudoLogFilePath=$(grep -r logfile /etc/sudoers* | sed -e 's/.*logfile=//;s/,? .*//'
 ↪  -e 's/"//g' -e 's|/|\\/|g')
# awk "/^ *-w/ &&/"${sudoLogFilePath}"/ &&/ +-p *wa/ &&(/ key= *[!-~]* *$/||/ -k *[!-
 ↪  ~]* *$/)" /etc/audit/rules.d/*.rules /etc/audit/*.rules

-w /var/log/sudo.log -p wa -k sudo_log_file
-w /var/log/sudo.log -p wa -k sudo_log_file
```

3. 执行以下命令，审计规则是否正确加载：

```
# sudoLogFilePath=$(grep -r logfile /etc/sudoers* | sed -e 's/.*logfile=//;s/,? .*//'
 ↪  -e 's/"//g' -e 's|/|\\/|g')
# auditctl -l | awk "/^ *-w/ &&/"${sudoLogFilePath}"/ &&/ +-p *wa/ &&(/ key= *[!-~]*
 ↪  *$/||/ -k *[!-~]* *$/)"

-w /var/log/sudo.log -p wa -k sudo_log_file
```

如输出结果均符合预期，则视为通过此项检查。

参考

2.31 确保收集特权命令的使用记录

安全等级

- Level 3

描述

创建审计规则，收集特权命令，如：sudo、su、umount、passwd、selinux_child 等对系统有重大影响的命令使用记录。

开启此审计规则，能够有效记录非授权用户滥用或误用特权命令，方便后期进行排查优化。此功能会增加系统负载和占用磁盘空间，请按实际需求决定是否开启此功能。

修复建议

确保收集特权命令的使用记录

1. 执行以下命令，添加审计规则：

```
build_audit_rules()
(
  UID_MIN=$(awk '/^\s*UID_MIN/{print $2}' /etc/login.defs)
  AUDIT_RULE_FILE="/etc/audit/rules.d/50-privileged.rules"
  NEW_DATA=()
  for PARTITION in $(findmnt -n -l -k -it $(awk '/nodev/ { print $2 }' /proc/
    ↪  filesystems | paste -sd,) | grep -Pv "noexec|nosuid" | awk '{print $1}'); do
    readarray -t DATA < <(find "${PARTITION}" -xdev -perm /6000 -type f | awk -v
      ↪  UID_MIN=${UID_MIN} '{print "-a always,exit -F path=" $1 " -F perm=x -F
      ↪  auid>="UID_MIN" -F auid!=unset -k privileged" }')
      for ENTRY in "${DATA[@]}"; do
        NEW_DATA+=("${ENTRY}")
      done
  done
  readarray &> /dev/null -t OLD_DATA < "${AUDIT_RULE_FILE}"
  COMBINED_DATA=( "${OLD_DATA[@]}" "${NEW_DATA[@]}" )
  printf '%s\n' "${COMBINED_DATA[@]}" | sort -u > "${AUDIT_RULE_FILE}"
```

```
)
build_audit_rules
```

2. 执行以下命令，加载审计规则：

```
# augenrules --load
```

扫描检查

1. 执行以下命令，检查审计规则是否正确写入配置文件：

```
for PARTITION in $(findmnt -n -l -k -it $(awk '/nodev/ { print $2 }' /proc/filesystems
 ↪  | paste -sd,) | grep -Pv "noexec|nosuid" | awk '{print $1}'); do
    for PRIVILEGED in $(find "${PARTITION}" -xdev -perm /6000 -type f); do
        grep -qr "${PRIVILEGED}" /etc/audit/rules.d && printf "OK: '${PRIVILEGED}'
         ↪  found in auditing rules.\n" || printf "Warning: '${PRIVILEGED}' not found
         ↪  in on disk configuration.\n"
    done
done
```

- 可能出现的输出结果：
  - OK: '/usr/sbin/pam_timestamp_check' found in auditing rules.
  - Warning: '/usr/sbin/unix_chkpwd' not found in on disk configuration.

2. 执行以下命令，检查审计规则是否正确加载：

```
RUNNING=$(auditctl -l)
[ -n "${RUNNING}" ] && for PARTITION in $(findmnt -n -l -k -it $(awk '/nodev/ { print
 ↪  $2 }' /proc/filesystems | paste -sd,) | grep -Pv "noexec|nosuid" | awk '{print
 ↪  $1}'); do
    for PRIVILEGED in $(find "${PARTITION}" -xdev -perm /6000 -type f); do
        printf -- "${RUNNING}" | grep -q "${PRIVILEGED}" && printf "OK:
         ↪  '${PRIVILEGED}' found in auditing rules.\n" || printf "Warning:
         ↪  '${PRIVILEGED}' not found in running configuration.\n"
    done
done \
```

```
|| printf "ERROR: Variable 'RUNNING' is unset.\n"
```

- 可能出现的输出结果：
  - OK: '/usr/sbin/pam_timestamp_check' found in auditing rules.
  - Warning: '/usr/sbin/unix_chkpwd' not found in running configuration.

以上检查脚本执行后，将自动检测本文档"描述"中提到的对系统有重大影响的命令是否正确配置了审计规则。输出结果应为多行，每个命令对应一行。

如输出结果为 `OK: ...` ，则表示此命令已正确配置了审计规则。如为 `Warning: ...` 则表示此命令暂未配置审计规则，还需检查并修复。

当输出结果中所有条目均为 `OK: ...` ，且未出现任何 `Warning: ...` 信息时，视为通过此项检查。

参考

2.32 确保收集访问控制权限修改事件

安全等级

- Level 3

描述

创建审计规则，收集访问控制权限修改事件:

- chmod
- fchmod
- fchmodat
- chown
- fchown
- fchownat
- lchown
- setxattr
- lsetxattr
- fsetxattr
- removexattr
- lremovexattr
- fremovexattr

收集以上命令的使用记录，能够有效记录非授权用户滥用或误用，方便后期进行排查优化。此功能会增加系统负载和占用磁盘空间，请按实际需求决定是否开启此功能。

修复建议

确保收集访问控制权限修改事件

1. 执行以下命令，添加审计规则:

```
UID_MIN=$(awk '/^\s*UID_MIN/{print $2}' /etc/login.defs)
[ -n "${UID_MIN}" ] && printf "
-a always,exit -F arch=b64 -S chmod,fchmod,fchmodat -F auid>=${UID_MIN} -F auid!=unset
 ↪  -F key=perm_mod
```

```
-a always,exit -F arch=b64 -S chown,fchown,lchown,fchownat -F auid>=${UID_MIN} -F
 ↪  auid!=unset -F key=perm_mod
-a always,exit -F arch=b32 -S chmod,fchmod,fchmodat -F auid>=${UID_MIN} -F auid!=unset
 ↪  -F key=perm_mod
-a always,exit -F arch=b32 -S lchown,fchown,chown,fchownat -F auid>=${UID_MIN} -F
 ↪  auid!=unset -F key=perm_mod
-a always,exit -F arch=b64 -S
 ↪  setxattr,lsetxattr,fsetxattr,removexattr,lremovexattr,fremovexattr -F
 ↪  auid>=${UID_MIN} -F auid!=unset -F key=perm_mod
-a always,exit -F arch=b32 -S
 ↪  setxattr,lsetxattr,fsetxattr,removexattr,lremovexattr,fremovexattr -F
 ↪  auid>=${UID_MIN} -F auid!=unset -F key=perm_mod
" >> /etc/audit/rules.d/50-perm_mod.rules \
|| printf "ERROR: Variable 'UID_MIN' is unset.\n"
```

2. 执行以下命令，加载审计规则：

```
# augenrules --load
```

扫描检查

1. 执行以下命令，检查审计规则是否正确写入配置文件：

```
UID_MIN=$(awk '/^\s*UID_MIN/{print $2}' /etc/login.defs)
[ -n "${UID_MIN}" ] && awk "/^ *-a *always,exit/ \
&&/ -F *arch=b[2346]{2}/ \
&&(/ -F *auid!=unset/||/ -F *auid!=-1/||/ -F *auid!=4294967295/) \
&&/ -S/ \
&&/ -F *auid>=${UID_MIN}/ \
&&(/chmod/||/fchmod/||/fchmodat/ \
  ||/chown/||/fchown/||/fchownat/||/lchown/ \
  ||/setxattr/||/lsetxattr/||/fsetxattr/ \
  ||/removexattr/||/lremovexattr/||/fremovexattr/) \
&&(/ key= *[!~]* *$/||/ -k *[!~]* *$/)" /etc/audit/rules.d/*.rules /etc/audit/
 ↪  *.rules \
```

```
|| printf "ERROR: Variable 'UID_MIN' is unset.\n"
```

- 预期输出结果

```
- a always,exit - F arch=b64 - S chmod,fchmod,fchmodat - F auid>=1000 - F auid! =unset -
F key=perm_mod
- a always,exit - F arch=b64 - S chown,fchown,lchown,fchownat - F auid>=1000 - F auid!
=unset -F key=perm_mod
- a always,exit - F arch=b32 - S chmod,fchmod,fchmodat - F auid>=1000 - F auid! =unset -
F key=perm_mod
- a always,exit - F arch=b32 - S lchown,fchown,chown,fchownat - F auid>=1000 - F auid!
=unset -F key=perm_mod
-a always,exit -F arch=b64 -S setxattr,lsetxattr,fsetxattr,removexattr,lremovexattr,fremovexattr -
F auid>=1000 -F auid!=unset -F key=perm_mod
-a always,exit -F arch=b32 -S setxattr,lsetxattr,fsetxattr,removexattr,lremovexattr,fremovexattr -
F auid>=1000 -F auid!=unset -F key=perm_mod
```

2. 执行以下命令，审计规则是否正确加载：

```
UID_MIN=$(awk '/^\s*UID_MIN/{print $2}' /etc/login.defs)
[ -n "${UID_MIN}" ] && auditctl -l | awk "/^ *-a *always,exit/ \
&&/ -F *arch=b[2346]{2}/ \
&&(/ -F *auid!=unset/||/ -F *auid!=-1/||/ -F *auid!=4294967295/) \
&&/ -S/ \
&&/ -F *auid>=${UID_MIN}/ \
&&(/chmod/||/fchmod/||/fchmodat/ \
  ||/chown/||/fchown/||/fchownat/||/lchown/ \
  ||/setxattr/||/lsetxattr/||/fsetxattr/ \
  ||/removexattr/||/lremovexattr/||/fremovexattr/) \
&&(/ key= *[!~]* *$/||/ -k *[!~]* *$/)" \
|| printf "ERROR: Variable 'UID_MIN' is unset.\n"
```

- 预期输出结果

```
- a always,exit - F arch=b64 - S chmod,fchmod,fchmodat - F auid>=1000 - F auid! =-1 -
F key=perm_mod
-a always,exit -F arch=b64 -S chown,fchown,lchown,fchownat -F auid>=1000 -F auid!=-1 -
```

```
F key=perm_mod
- a always,exit - F arch=b32 - S chmod,fchmod,fchmodat - F auid>=1000 - F auid! =-1 -
F key=perm_mod
-a always,exit -F arch=b32 -S lchown,fchown,chown,fchownat -F auid>=1000 -F auid!=-1 -
F key=perm_mod
-a always,exit -F arch=b64 -S setxattr,lsetxattr,fsetxattr,removexattr,lremovexattr,fremovexattr -
F auid>=1000 -F auid!=-1 -F key=perm_mod
-a always,exit -F arch=b32 -S setxattr,lsetxattr,fsetxattr,removexattr,lremovexattr,fremovexattr -
F auid>=1000 -F auid!=-1 -F key=perm_mod
```

如输出结果均符合预期，则视为通过此项检查。

参考

3 services

3.1 禁用 HTTP Server

安全等级

- Level 1

描述

HTTP 或 Web 服务器提供托管网站内容的能力。

除非需要将系统作为 Web 服务器运行，否则建议禁用软件包以减少潜在的攻击面。

修复建议

运行以下命令来禁用 `httpd`

```
# systemctl --now disable httpd
```

扫描检测

运行以下命令来检查 `httpd` 是否被禁用

```
# systemctl is-enabled httpd
```

期待的输出结果为 `disabled`

参考

3.2 禁用 FTP Server

安全等级

- Level 1

描述

文件传输协议 (FTP) 为联网计算机提供了传输文件的能力。

FTP 不保护数据或身份验证凭据的机密性。如果需要文件传输，建议使用 SFTP。除非需要将系统作为 FTP 服务器运行（例如，允许匿名下载），否则建议禁用对应的服务以减少潜在的攻击面。

修复建议

运行以下命令来禁用 `vsftpd`

```
# systemctl --now disable vsftpd
```

扫描检测

运行以下命令来检查 `vsftpd` 是否被禁用

```
# systemctl is-enabled vsftpd
```

期待的输出结果**不是** `enabled`

参考

3.3 禁用 DNS Server

安全等级

- Level 1

描述

域名系统 (DNS) 是一种分层命名系统，它将名称映射到计算机、服务和其它联网资源的 IP 地址。

除非系统被专门指定用作 DNS 服务器，否则建议禁用 DNS Server 以减少潜在的攻击面。

修复建议

运行以下命令来禁用 `named`

```
# systemctl --now disable named
```

扫描检测

运行以下命令来检查 `named` 是否被禁用

```
# systemctl is-enabled named
```

期待的输出结果**不是** `enabled`

参考

3.4 禁用 NFS

安全等级

- Level 1

描述

网络文件系统 (NFS) 是 UNIX 环境中最早也是分布最广泛的文件系统之一。它为系统提供了通过网络挂载其他服务器的文件系统的能力。

如果系统不导出 NFS 共享，建议禁用 NFS 以减少远程攻击面。

修复建议

运行以下命令来禁用 `nfs-server`

```
# systemctl --now disable nfs-server
```

扫描检测

运行以下命令来检查 `nfs-server` 是否被禁用

```
# systemctl is-enabled nfs-server
```

期待的输出结果**不是** `enabled`

参考

3.5 禁用 RPC

安全等级

- Level 1

描述

rpcbind 服务将远程过程调用 (RPC) 服务映射到它们侦听的端口。RPC 进程在启动时通知 rpcbind，注册它们正在侦听的端口以及它们期望服务的 RPC 程序编号。然后，客户端系统使用特定的 RPC 程序号联系服务器上的 rpcbind。rpcbind 服务将客户端重定向到正确的端口号，以便它可以与请求的服务进行通信。

如果系统不需要基于 rpc 的服务，建议禁用 rpcbind 以减少远程攻击面

修复建议

运行以下命令来禁用 `rpcbind`

```
# systemctl stop rpcbind.service
# systemctl stop rpcbind.socket
# systemctl mask rpcbind
```

扫描检测

运行以下命令来检查 `rpcbind` 是否被禁用

```
# systemctl is-enabled rpcbind
```

期待的输出结果**不是** `enabled`

参考

3.6 禁用 LDAP Server

安全等级

- Level 1

描述

轻量级目录访问协议 (LDAP) 被引入作为 NIS/YP 的替代品。它是一种提供从中央数据库查找信息的方法的服务。

如果系统不需要充当 LDAP Server，建议禁用 LDAP Server 以减少潜在的攻击面。

修复建议

运行以下命令来禁用 `slapd`

```
# systemctl --now disable slapd
```

扫描检测

运行以下命令来检查 `slapd` 是否被禁用

```
# systemctl is-enabled slapd
```

期待的输出结果**不是** `enabled`

参考

3.7 禁用 DHCP Server

安全等级

- Level 1

描述

动态主机配置协议 (DHCP) 是一项允许为机器动态分配 IP 地址的服务。

除非系统专门设置为充当 DHCP Server，否则建议禁用该服务以减少潜在的攻击面。

修复建议

运行以下命令来禁用 `dhcpd`

```
# systemctl --now disable dhcpd
```

扫描检测

运行以下命令来检查 `dhcpd` 是否被禁用

```
# systemctl is-enabled dhcpd
```

期待的输出结果**不是** `enabled`

参考

3.8 禁用 CUPS

安全等级

- Level 1

描述

通用 Unix 打印系统 (CUPS) 提供了打印到本地和网络打印机的能力。运行 CUPS 的系统还可以接受来自远程系统的打印作业并将它们打印到本地打印机。它还提供基于 Web 的远程管理功能。

如果系统不需要打印作业或接受来自其他系统的打印作业，建议禁用 CUPS 以减少潜在的攻击面。

修复建议

运行以下命令来禁用 `cups`

```
# systemctl --now disable cups
```

扫描检测

运行以下命令来检查 `cups` 是否被禁用

```
# systemctl is-enabled cups
```

期待的输出结果**不是** `enabled`

参考

3.9 禁用 NIS Server

安全等级

- Level 1

描述

网络信息服务 (NIS)（以前称为黄页）是一种用于分发系统配置文件的客户端-服务器目录服务协议。NIS 服务器是允许分发配置文件的程序的集合。

NIS Server 本质上是一个不安全的系统，容易受到 DOS 攻击、缓冲区溢出以及查询 NIS 映射的身份验证较差。NIS 通常已被轻量级目录访问协议 (LDAP) 等协议取代。建议禁用该服务并使用其他更安全的服务。

修复建议

运行以下命令来禁用 `ypserv`

```
# systemctl --now disable ypserv
```

扫描检测

运行以下命令来检查 `ypserv` 是否被禁用

```
# systemctl is-enabled ypserv
```

期待的输出结果**不是** `enabled`

参考

3.10 禁用 Rsync Server

安全等级

- Level 1

描述

rsyncd 服务可用于在网络连接的系统之间同步文件。

rsyncd 服务存在安全风险，因为它使用未加密的协议进行通信。

修复建议

运行以下命令来禁用 `rsyncd`

```
# systemctl --now disable rsyncd
```

扫描检测

运行以下命令来检查 `rsyncd` 是否被禁用

```
# systemctl is-enabled rsyncd
```

期待的输出结果**不是** `enabled`

参考

3.11 禁用 Avahi Server

安全等级

- Level 1

描述

Avahi 是一个免费的 zeroconf（零配置网络服务规范）实现，包括用于多播 DNS/DNS-SD 服务发现的系统。Avahi 允许程序在没有特定配置的情况下发布和发现在本地网络上运行的服务和主机。例如，用户可以将计算机插入网络，Avahi 会自动查找要打印到的打印机、要查看的文件和要交谈的人，以及机器上运行的网络服务。

系统功能通常不需要自动发现网络服务。建议禁用该服务以减少潜在的攻击面。

修复建议

运行以下命令来禁用 `avahi-daemon.socket` 和 `avahi-daemon.service`

```
# systemctl --now disable avahi-daemon.socket
# systemctl --now disable avahi-daemon.service
```

扫描检测

1. 运行以下命令来检查 `avahi-daemon.socket` 是否被禁用

```
# systemctl is-enabled avahi-daemon.socket
```

期待的输出结果**不是** `enabled` 。

2. 运行以下命令来检查 `avahi-daemon.serivce` 是否被禁用

```
# systemctl is-enabled avahi-daemon
```

期待的输出结果**不是** `enabled` 。

参考

3.12 禁用 SNMP Server

安全等级

- Level 1

描述

简单网络管理协议 (SNMP) 服务器用于侦听来自 SNMP 管理系统的 SNMP 命令，执行命令或收集信息，然后将结果发送回请求系统。

SNMP 服务器可以使用 SNMP v1 进行通信，它以明文方式传输数据，并且不需要身份验证即可执行命令。除非绝对必要，否则建议不要使用 SNMP 服务。如果需要 SNMP，则应将服务器配置为禁止 SNMP v1。

修复建议

运行以下命令来禁用 `snmpd`

```
# systemctl --now disable snmpd
```

扫描检测

运行以下命令来检查 `snmpd` 是否被禁用

```
# systemctl is-enabled snmpd
```

期待的输出结果**不是** `enabled`

参考

3.13 禁用 HTTP Proxy Server

安全等级

- Level 1

描述

Squid 是在许多发行版和环境中使用的标准代理服务器。

如果不需要代理服务器，建议禁用 squid 代理以减少潜在的攻击面。

修复建议

运行以下命令来禁用 `squid`

```
# systemctl --now disable squid
```

扫描检测

运行以下命令来检查 `squid` 是否被禁用

```
# systemctl is-enabled squid
```

期待的输出结果**不是** `enabled`

参考

3.14 禁用 Samba

安全等级

- Level 1

描述

Samba 守护程序允许系统管理员配置他们的 Linux 系统以与 Windows 桌面共享文件系统和目录。Samba 将通过服务器消息块 (SMB) 协议公布文件系统和目录。Windows 桌面用户将能够将这些目录和文件系统作为盘符挂载在他们的系统上。

如果不需要将目录和文件系统挂载到 Windows 系统，则可以删除该服务以减少潜在的攻击面。

修复建议

运行以下命令来禁用 `smb`

```
# systemctl --now disable smb
```

扫描检测

运行以下命令来检查 `smb` 是否被禁用

```
# systemctl is-enabled smb
```

期待的输出结果**不是** `enabled`

参考

3.15 禁用 IMAP 和 POP3 Server

安全等级

- Level 1

描述

dovecot 是基于 Linux 系统的开源 IMAP 和 POP3 服务器。

除非该系统要提供 POP3 或 IMAP 服务器，否则建议禁用该服务以减少潜在的攻击面。

修复建议

运行以下命令来禁用 `dovecot`

```
# systemctl --now disable dovecot
```

扫描检测

运行以下命令来检查 `dovecot` 是否被禁用

```
# systemctl is-enabled dovecot
```

期待的输出结果**不是** `enabled`

参考

3.16 禁用使用 smtp 协议的 postfix 服务

安全等级

- Level 1

描述

允许垃圾邮件发送者发送未经授权的 email 的邮件服务器使用 25 端口进行邮件传递，而 smtp 协议中 25 端口无需认证，这样邮件可以无障碍得在邮件传输代理中传输，存在安全风险。除非该系统要提供 postfix 服务器，否则建议禁用该服务以减少潜在的攻击面。

修复建议

运行以下命令来禁用 `postfix`

```
# systemctl --now disable postfix.service
```

扫描检测

运行以下命令来检查 `postfix` 是否被禁用:

```
# systemctl is-enabled postfix.service
```

期待的输出结果 `disabled` 。

参考

3.17 禁用或卸载 telnet

安全等级

- Level 1

描述

telnet 客户端允许用户通过 telnet 协议启动与其他系统的连接。然而 telnet 协议不安全且未加密，使用未加密的传输介质可能允许未经授权的用户窃取凭据。

修复建议

目标：禁用 telnet 的 23 端口或确保 telnet 被卸载

- 运行以下命令来禁用 telnet 。

```
# systemctl --now disable telnet.socket
```

或者:

- 运行以下命令来卸载 telnet 。

```
# dnf remove telnet telnet-server -y
```

扫描检测

1. 运行以下命令来检查是否安装 telnet 。

```
# rpm -qa | grep telnet
```

若输出为空则表示未安装 telnet ，满足预期目标，扫描结束通过检查。

如果已安装 telnet 则:

2. 运行以下命令来检查 telnet 是否被禁用。

```
# systemctl is-enabled telnet.socket
disabled
```

输出结果为 `disabled` 则表示已禁用 telnet 。

如 telnet 服务未安装或已禁用，则视为通过此项检查。

参考

3.18 卸载 Avahi

安全等级

- Level 1

描述

Avahi 是一个免费的 zeroconf（零配置网络服务规范）实现，包括用于多播 DNS/DNS-SD 服务发现的系统。Avahi 允许程序在没有特定配置的情况下发布和发现在本地网络上运行的服务和主机。例如，用户可以将计算机插入网络，Avahi 会自动查找要打印到的打印机、要查看的文件和要交谈的人，以及机器上运行的网络服务。

系统功能通常不需要自动发现网络服务。建议卸载该服务以减少潜在的攻击面。

修复建议

运行以下命令来卸载 `avahi`

```
# yum remove -y --noautoremove avahi
```

扫描检测

确保未安装 avahi。

1. 执行以下命令，检查 avahi 软件包是否安装：

```
# rpm -q avahi
package avahi is not installed
```

如输出结果符合预期，则视为通过此项检查。

参考

3.19 卸载 kexec-tools

安全等级

- Level 3

描述

kexec 工具是 Linux 内核的一个补丁，让您可以从当前正在运行的内核直接引导到一个新内核。在上面描述的引导序列中，kexec 跳过了整个引导装载程序阶段（第一部分）并直接跳转到我们希望引导到的内核。不再有硬件的重启，不再有固件操作，不再涉及引导装载程序。完全避开了引导序列中最弱的一环：固件。这一功能部件带来的最大益处在于，系统现在可以极其快速地重新启动。

但由于其直接跳过了引导阶段，可以滥用此功能来加载恶意内核并在内核模式下获得任意代码执行能力。因此应当卸载此工具。

修复建议

运行以下命令来卸载 `kexec-tools`

```
# yum remove -y --noautoremove kexec-tools
```

扫描检测

确保未安装 kexec-tools。

1. 执行以下命令，检查 kexec-tools 软件包是否安装：

```
# rpm -q kexec-tools
package kexec-tools is not installed
```

如输出结果符合预期，则视为通过此项检查。

参考

3.20 卸载 firstboot

安全等级

- Level 1

描述

Linux 在安装完之后第一次启动会启动 firstboot 服务。Firstboot 只能在使用图形安装或者安装了桌面和 X 视窗系统，并启用图形登录的 kickstart 安装中使用。如果执行文本安装或者没有包括桌面和 X 视窗系统的 kickstart 安装，则不会出现 firstboot 配置工具。

修复建议

运行以下命令来卸载 `firstboot`

```
# yum remove -y --noautoremove firstboot
```

扫描检测

确保未安装 firstboot。

1. 执行以下命令，检查 firstboot 软件包是否安装：

```
# rpm -q firstboot
package firstboot is not installed
```

如输出结果符合预期，则视为通过此项检查。

参考

3.21 卸载 wpa_supplicant

安全等级

- Level 1

描述

wpa_supplicant 是 wifi 客户端（client）加密认证工具，和 iwconfig 不同，wpa_supplicant 支持 wep、wpa、wpa2 等完整的加密认证，而 iwconfig 只能支持 wep。如对无线网络没有需求，应卸载此服务，以减少潜在的攻击面。

修复建议

运行以下命令来卸载 `wpa_supplicant`

```
# yum remove -y --noautoremove wpa_supplicant
```

扫描检测

确保未安装 wpa_supplicant。

1. 执行以下命令，检查 wpa_supplicant 软件包是否安装：

```
# rpm -q wpa_supplicant
package wpa_supplicant is not installed
```

如输出结果符合预期，则视为通过此项检查。

参考

3.22 确保 NIS 客户端被卸载

安全等级

- Level 1

描述

Network Information Service（NIS）是一种采用客户端-服务器架构的目录服务协议，早期也被称为“黄页”服务。它用于分发系统配置文件。NIS 客户端（ypbind）用于将设备连接到 NIS 服务器，并从服务器获取分发下来的配置文件。

NIS 服务从本质上讲是一个不安全的系统，它很容易遭到 DOS 攻击、缓冲区溢出攻击等，而其用于查询 NIS 目录的身份认证机制也不可靠。一般来说，NIS 服务已经被轻量级目录访问协议（LDAP）等替代。建议将其卸载。

修复建议

目标：确保 ypbind 被卸载

1. 运行以下命令卸载 ypbind。

```
# dnf remove -y ypbind
```

扫描检测

执行修复前检测 ypbind 是否安装

1. 运行以下命令以检测是否安装 ypbind。

```
# rpm -q ypbind
package ypbind is not installed
```

输出结果为 `package ypbind is not installed` 则表示未安装 ypbind。

参考

3.23 禁用 rsh

安全等级

- Level 1

描述

Rsh 是远程外壳 (remote shell) 的缩写 (外壳是操作系统的一种命令接口)。运行于远程计算机上的 rshd 后台程序，接受 rsh 命令，验证用户名和主机名信息，并执行该命令。当用户不愿或不需要与远程计算机建立远程会话时，可以使用 rsh 工具执行输入的命令。Rsh 工具允许用户在远程计算机上执行单条命令，而无需在该远程计算机上进行登录。

因 rsh 使用明文传输，且没有密钥的机制，有极大的安全隐患。所以应当禁用 rsh 服务，使用更加安全的远程链接方式，如 SSH 等。

修复建议

运行以下命令来禁用 `rsh`

```
# systemctl --now disable rsh.socket
```

扫描检测

运行以下命令来检查 `rsh` 是否被禁用

```
# systemctl is-enabled rsh.socket
disabled
```

如输出结果为 `disabled` ，或提示未安装此服务，则视为通过此项检查。

参考

3.24 禁用 ntalk

安全等级

- Level 1

描述

talk/ntalk 是一个用于 Linux 用户之间交流的程序，write 也可以实现用户交流，但是 write 一次只能发送一条信息。而 talk 是基于 socket 实现的，用户可以实时交流。

因 ntalk 使用明文传输，且没有密钥的机制，有极大的安全隐患。所以应当禁用 ntalk 服务，使用更加安全的远程链接方式，如 SSH 等。

修复建议

运行以下命令来禁用 `ntalk`

```
# systemctl --now disable ntalk
```

扫描检测

运行以下命令来检查 `ntalk` 是否被禁用

```
# systemctl is-enabled ntalk
disabled
```

如输出结果为 `disabled` ，或提示未安装此服务，则视为通过此项检查。

参考

3.25 确保 xinetd 被卸载

安全等级

- Level 1

描述

eXtended InterNET 守护进程 (xinetd) 是一个开源的超级守护进程，用于取代原始 inetd 守护进程。xinetd 能够监听许多常用服务并调度合适的守护进程以正确响应服务请求。

若 xinetd 服务非必要，建议将其卸载。

修复建议

目标：确保 xinetd 被卸载

1. 运行以下命令卸载 xinetd。

```
# dnf remove -y xinetd
```

扫描检测

执行修复前检测 xinetd 是否安装

1. 运行以下命令以检测是否安装 xinetd。

```
# rpm -q xinetd
package xinetd is not installed
```

输出结果为 `package xinetd is not installed` 则表示未安装 xinetd。

参考

3.26 禁用 USB 存储

安全等级

- level 1

描述

`USB` 存储是一种传输和存储文件的方式，它与网络连接状态无关，能够确保文件的持久性和可用性。它的流行和实用性导致基于 `USB` 的恶意软件成为进行网络渗透的简单常见手段，并使它成为在网络环境中建立持久威胁的第一步。限制系统上的 `USB` 访问能够有效地缩小设备的物理攻击面，降低恶意软件的引入可能性。

通过在模块装载程序 `modprobe` 读取并加载内核模块时，修改系统上的 `usb-storage` 内核模块的安装指向，使其指向 `/bin/true` ，可使该内核模块成功装载但实际上不执行任何实际操作，由此可以禁用 `USB` 存储。

禁用 `usb-storage` 模块的另一种方案可以在 `USBGuard` 中找到。使用 `USBGuard` 并制定 `USB` 设备策略应遵循站点政策。

修复建议

修改 `usb-storage` 模块安装指向，并卸载 `usb-storage` 模块。

1. 在 `/etc/modprobe.d/` 目录下创建（或编辑）一个 `.conf` 文件（文件可任意命名，示例： `usb_storage.conf` ），并在其中输入以下内容，以将 `usb-storage` 模块安装指向 `/bin/true` ：

创建/编辑文件：

```
# vim /etc/modprobe.d/usb_storage.conf
```

输入如下内容并保存：

```
install usb-storage /bin/true
```

2. 执行以下命令，以立即卸载 `usb-storage` 模块：

```
# rmmod usb-storage
```

扫描检测

确保 `usb-storage` 模块已经卸载。

1. 执行以下命令，验证 `usb-storage` 模块的安装指向并验证 `usb-storage` 模块是否已经卸载：

```
# modprobe -n -v usb-storage
install /bin/true
# lsmod | grep usb-storage
<No output>
```

如果第一条命令执行后返回 `install /bin/true` ，且第二条命令执行后，没有返回任何结果，则视为通过此项检查。

参考

3.27 确保时间同步服务已安装

安全等级

- Level 1

描述

同一环境中的所有系统之间应同步系统时间。这通常是通过建立一个或一组权威的时间服务器，并使所有系统的时钟与之同步来实现的。时间同步对于支持时间敏感的安全机制如 Kerberos 非常重要，它还可以确保日志文件在整个企业中具有一致的时间记录，这有助于取证调查。

修复建议

目标：安装 chrony 软件包。

1. 运行以下命令安装 chrony。

```
# dnf install chrony -y
```

扫描检测

验证是否正确安装了 chrony。

1. 运行以下命令以检测是否安装 chrony。

```
# rpm -q chrony
chrony-<VERSION>
```

输出结果为 `chrony-<VERSION>` 则表示安装了 chrony。其中 `<VERSION>` 为版本信息。

参考

3.28 禁用自动挂载

安全等级

- Level 1

描述

`autofs` 允许自动安装外设，通常包括 `CD/DVD` 和 `USB` 驱动器。当启用自动挂载后，任何可以访问设备的用户，即使他们没有权限装载外设，也可以在服务器或主机上连接 `USB` 驱动器或光盘，并在系统中读取、改动他们的内容。协作工作的企业用户习惯于使用便携式存储设备，如果用户的管理者允许其在工作站上使用便携式存储和媒体设备，并且对工作站服务设备的访问权限是足够的，那么关闭自动挂载几乎没有什么价值。

修复建议

1. 如果有其他安装包依赖于 `autofs` ，请使用以下命令禁用 `autofs` 服务：

```
# systemctl --now disable autofs
```

2. 如果没有其他依赖于 `autofs` 的包，请使用以下命令卸载 autofs 软件包：

```
# dnf remove autofs -y
```

扫描检测

除非其他包依赖于 `autofs` 或是缺失需要自动挂载功能，否则不建议安装 `autofs` 。

1. 运行以下命令来检测 `autofs` 是否被成功禁用或移除：

```
# systemctl is-enabled autofs
disabled
```

```
# systemctl is-enabled autofs
Failed to get unit file state for autofs.service: No such file or directory
```

- 如返回 `disabled` ，则表示当前系统已安装了 `autofs` 软件包，但未启用。**通过检查（pass）**
- 如返回 `Failed to get unit file state for autofs.service: No such file or directory` ，则表示当前系统未安装 `autofs` 软件包。**通过检查（pass）**
- 如返回 `enabled` ，则表示当前系统已安装了 `autofs` 软件包，且已启用。**未通过检查，需处理（fail）**

参考

4 system-configurations

4.1 确保登录提示消息的内容符合要求

安全等级

- Level 1

描述

`/etc/motd` 文件的内容会在用户登录后展示给用户。

类 Unix 系统通常会在登录系统时显示当前操作系统的版本信息和补丁信息。这些信息对于为特定操作系统平台开发软件的开发者来说是很有用的。但这样做也有一个副作用：会将系统版本的详细提供给针对特定系统漏洞的攻击者。而这些信息，对于已授权的用户来说，可以通过 `uname -a` 命令轻松获得。

综上，登录提示信息不应展示操作系统版本信息及补丁信息。

修复建议

编辑或删除 `/etc/motd` 文件，确保系统信息不展示在登录消息内。

1. 可根据使用环境，自行编辑登录提示消息，但要确保其内容没有系统版本或补丁信息。如不需要登录提示消息，直接删除 `/etc/motd` 文件即可。

- 参考内容:

```
# echo "Authorized uses only. All activity may be monitored and reported." > /etc/motd
```

扫描检测

确保登录提示消息的内容符合要求。

1. 执行以下命令，检查输出内容是否有不符合安全要求的信息（系统版本信息、补丁信息等）:

```
# cat /etc/motd
Authorized uses only. All activity may be monitored and reported.
```

2. 执行以下命令，确保没有返回任何信息：

```
# grep -E -i "(\\v|\\r|\\m|\\s|$(grep '^ID=' /etc/os-release | cut -d= -f2 | sed -
 ↪  e 's/"//g'))" /etc/motd
No information is returned.
```

如 `/etc/motd` 内没有任何不符合安全要求的信息（系统版本信息、补丁信息等），且第 2 条命令执行后没有返回任何内容，则视为通过此项检查。

参考

4.2 确保本地登录提示消息的内容符合要求

安全等级

- Level 1

描述

在**本地终端**登录时会展示 `/etc/issue` 文件中的内容。

类 Unix 系统通常会在登录系统时显示当前操作系统的版本信息和补丁信息。这些信息对于为特定操作系统平台开发软件的开发者来说是很有用的。但这样做也有一个副作用：会将系统版本的详细提供给针对特定系统漏洞的攻击者。而这些信息，对于已授权的用户来说，可以通过 `uname -a` 命令轻松获得。

综上，登录提示信息不应展示操作系统版本信息及补丁信息。

`/etc/issue` 文件的内容解析：

```
# cat /etc/issue
\S
Kernel \r on an \m
```

其中 `\*` 参数对应输出内容如下：

`\m` 给出当前操作系统的位数

`\r` 详细的内核版本

`\s` 操作系统的名称

`\v` 操作系统的版本

修复建议

编辑或删除 `/etc/issue` 文件，确保系统信息不展示在登录消息内。

1. 可根据使用环境，自行编辑登录提示消息，但要确保其内容没有系统版本或补丁信息。删除任何含有 `\m` 、 `\r` 、 `\s` 、 `\v` 的信息。

- 参考内容：

```
# echo "Authorized uses only. All activity may be monitored and reported." > /etc/
 ↪ issue
```

扫描检测

确保登录提示消息的内容符合要求。

1. 执行以下命令，检查输出内容是否有不符合安全要求的信息（系统版本信息、补丁信息等）：

```
# cat /etc/issue
Authorized uses only. All activity may be monitored and reported.
```

2. 执行以下命令，确保没有返回任何信息：

```
# grep -E -i "(\\v|\\r|\\m|\\s|$(grep '^ID=' /etc/os-release | cut -d= -f2 | sed -
 ↪ e 's/"//g'))" /etc/issue
No information is returned.
```

如 `/etc/issue` 内没有任何不符合安全要求的信息（系统版本信息、补丁信息等），且第 2 条命令执行后没有返回任何内容，则视为通过此项检查。

参考

4.3 确保远程登录提示消息的内容符合要求

安全等级

- Level 1

描述

在**远程终端**登录时会展示 `/etc/issue.net` 文件中的内容。

类 Unix 系统通常会在登录系统时显示当前操作系统的版本信息和补丁信息。这些信息对于为特定操作系统平台开发软件的开发者来说是很有用的。但这样做也有一个副作用：会将系统版本的详细提供给针对特定系统漏洞的攻击者。而这些信息，对于已授权的用户来说，可以通过 `uname -a` 命令轻松获得。

综上，登录提示信息不应展示操作系统版本信息及补丁信息。

`/etc/issue.net` 文件的内容解析：

```
# cat /etc/issue.net
\S
Kernel \r on an \m
```

其中 `\*` 参数对应输出内容如下：

`\m` 给出当前操作系统的位数

`\r` 详细的内核版本

`\s` 操作系统的名称

`\v` 操作系统的版本

修复建议

编辑或删除 `/etc/issue.net` 文件，确保系统信息不展示在登录消息内。

1. 可根据使用环境，自行编辑登录提示消息，但要确保其内容没有系统版本或补丁信息。删除任何含有 `\m` 、 `\r` 、 `\s` 、 `\v` 的信息。

- 参考内容：

```
# echo "Authorized uses only. All activity may be monitored and reported." > /etc/
 ↪ issue.net
```

扫描检测

确保登录提示消息的内容符合要求。

1. 执行以下命令，检查输出内容是否有不符合安全要求的信息（系统版本信息、补丁信息等）：

```
# cat /etc/issue.net
Authorized uses only. All activity may be monitored and reported.
```

2. 执行以下命令，确保没有返回任何信息：

```
# grep -E -i "(\\v|\\r|\\m|\\s|$(grep '^ID=' /etc/os-release | cut -d= -f2 | sed -
 ↪ e 's/"//g'))" /etc/issue.net
No information is returned.
```

如 `/etc/issue.net` 内没有任何不符合安全要求的信息（系统版本信息、补丁信息等），且第 2 条命令执行后没有返回任何内容，则视为通过此项检查。

参考

4.4 确保 /etc/motd 的权限配置正确

安全等级

- Level 1

描述

`/etc/motd` 文件的内容会在用户登录后展示给用户。

如果 `/etc/motd` 文件的权限及所有者没有正确配置，其内容就可能被未经授权的用户篡改，从而展示不正确或误导性的登录信息。

在最新版本的 Anolis OS 8 中，`/etc/motd` 文件为一个软连接文件，指向 `/var/lib/update-motd/motd` 文件，所以在检查和修复过程中，也需要对 `/var/lib/update-motd/motd` 文件进行操作。

修复建议

目标：正确配置 `/etc/motd` 及 `/var/lib/update-motd/motd` 文件的权限和所有者。

1. 使用以下代码，配置 `/etc/motd` 及 `/var/lib/update-motd/motd` 文件的权限和所有者：

```
# chown root:root /etc/motd
# chmod u-x,go-wx /etc/motd
# chown root:root /var/lib/update-motd/motd
# chmod u-x,go-wx /var/lib/update-motd/motd
```

扫描检测

确保 `/etc/motd` 及 `/var/lib/update-motd/motd` 文件的权限配置正确。

1. 执行以下命令，检查 `/etc/motd` 文件是否是以软连接的形式存在：

```
## 正常文件形式
# ll /etc/motd
-rw-r--r--. 1 root root 53 Aug 16 15:17 /etc/motd
```

```
## 软连接形式
# ll /etc/motd
lrwxrwxrwx 1 root root 25 Oct 31 10:15 /etc/motd -> /var/lib/update-motd/motd
```

2. 如 `/etc/motd` 文件为正常文件，则需检查 `/etc/motd` 文件的权限属性。

```
# stat /etc/motd
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
```

3. 如 `/etc/motd` 文件为软连接文件，则需检查 `/etc/motd` 及 `/var/lib/update-motd/motd` 文件的权限属性：

```
# stat /etc/motd
Access: (0777/lrwxrwxrwx)  Uid: (    0/    root)   Gid: (    0/    root)

# stat /var/lib/update-motd/motd
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
```

如果输出结果中：`Uid` 与 `Gid` 均为 `0/root` ，且 `Acces` 为 `0644` 或更加严格（软连接文件 `/etc/motd` 除外），则视为通过此项检查。

参考

4.5 确保 /etc/issue 的权限配置正确

安全等级

- Level 1

描述

在本地终端登录时会展示 `/etc/issue` 文件中的内容。

如果 `/etc/issue` 文件的权限及所有者没有正确配置，其内容就可能被未经授权的用户篡改，从而展示不正确或误导性的登录信息。

修复建议

目标：正确配置 `/etc/issue` 文件的权限和所有者。

1. 使用以下代码，配置 `/etc/issue` 文件的权限和所有者：

```
# chown root:root /etc/issue
# chmod u-x,go-wx /etc/issue
```

扫描检测

确保 `/etc/issue` 目录的权限配置正确。

1. 执行以下命令，检查 `/etc/issue` 目录的权限属性：

```
# stat /etc/issue
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
```

如果输出结果中：`Uid` 与 `Gid` 均为 `0/root` ，且 `Acces` 为 `0644` 或更加严格，则视为通过此项检查。

参考

4.6 确保 /etc/issue.net 的权限配置正确

安全等级

- Level 1

描述

在远程终端登录时会展示 `/etc/issue.net` 文件中的内容。

如果 `/etc/issue.net` 文件的权限及所有者没有正确配置，其内容就可能被未经授权的用户篡改，从而展示不正确或误导性的登录信息。

修复建议

目标：正确配置 `/etc/issue.net` 文件的权限和所有者。

1. 使用以下代码，配置 `/etc/issue.net` 文件的权限和所有者：

```
# chown root:root /etc/issue.net
# chmod u-x,go-wx /etc/issue.net
```

扫描检测

确保 `/etc/issue.net` 目录的权限配置正确。

1. 执行以下命令，检查 `/etc/issue.net` 目录的权限属性：

```
# stat /etc/issue.net
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
```

如果输出结果中：`Uid` 与 `Gid` 均为 `0/root` ，且 `Acces` 为 `0644` 或更加严格，则视为通过此项检查。

参考

4.7 确保 gpgcheck 全局激活

安全等级

- Level 1

描述

在 `/etc/dnf/dnf.conf` 文件及 `/etc/yum.repos.d/*` 目录下的部分文件中，包含 `gpgcheck` 配置参数。`gpgcheck` 参数的配置决定了在安装软件前是否检查 RPM 包的签名。

在安装软件之前，一定要检查 RPM 包的签名，以确保软件是从可信的来源获得的，这一点非常重要。

修复建议

目标：激活 gpgcheck 签名检查。

1. 编辑 `/etc/dnf/dnf.conf` 文件，在 `[main]` 部分设置 `gpgcheck=1` ：

```
# sed -i 's/^gpgcheck\s*=\s*.*/gpgcheck=1/' /etc/dnf/dnf.conf
```

2. 编辑 `/etc/yum.repos.d/` 目录下所有文件内 `gpgcheck` 参数的值为 1：

```
# find /etc/yum.repos.d/ -name "*.repo" -exec echo "Checking:" {} \; -exec sed -i 's/
↪  ^gpgcheck\s*=\s*.*/gpgcheck=1/' {} \;
```

扫描检测

确保 gpgcheck 全局激活。

1. 执行以下命令，检查 `/etc/dnf/dnf.conf` 文件中 `gpgcheck` 参数配置是否正确：

```
# grep ^gpgcheck /etc/dnf/dnf.conf
gpgcheck=1
```

2. 配置在 `/etc/yum.repos.d/` 目录下的 `gpgcheck` 参数，优先级大于全局配置。执行以下命令，确认没有 `gpgcheck=0` 的配置项。

```
# grep -P "^gpgcheck\h*=\h*[^1].*\h*$" /etc/yum.repos.d/*
No information is returned.
```

如果以上 2 条检查项目的输出结果均符合要求，则视为通过此项检查。

参考

4.8 确保正确安装 AIDE

安全等级

- Level 1

描述

AIDE (Advanced Intrusion Detection Environment) 是一种入侵检测工具，它在 Linux 操作系统下使用预定义的规则对文件和目录的完整性进行检测。

AIDE 有自己的数据库来检查文件和目录的完整性：AIDE 获取文件和目录的快照，包括修改时间、权限和文件哈希值，然后使用该快照与文件系统的当前状态进行比较，以检测文件系统的修改。

修复建议

目标：正确安装 `AIDE` 。

1. 执行以下命令，安装 `AIDE` 软件：

```
# dnf install aide -y
```

2. 初始化 `AIDE` 服务：

```
# aide --init
# mv /var/lib/aide/aide.db.new.gz /var/lib/aide/aide.db.gz
```

扫描检测

确保正确安装 AIDE。

1. 执行以下命令，验证是否正确安装了 AIDE：

```
# rpm -q aide
aide-<version>
```

`<version>` 为版本号，如：`aide-0.16-14.an8_5.1.x86_64` 。如有返回 `aide` 及版本号，则视为通过此项检查。

参考

4.9 确保定期检查文件系统完整性

安全等级

- Level 1

描述

定期进行文件系统完整性检查，有助于系统管理员跟踪了解关键文件的变化，及时发现关键文件是否有被未经授权的更改或删除。

文件系统完整性检查依赖于 aide 工具，请在执行修复前，确认当前环境是否正确安装并初始化了 aide 工具。

修复建议

建立对文件系统的定期检查机制。

1. 检查是否安装了 aide：

```
# rpm -q aide
aide-<version>
```

2. 检查 aide 是否已正确初始化:

```
# ls /var/lib/aide/aide.db.gz
/var/lib/aide/aide.db.gz
```

如以上检查输出结果均符合预期，则可进行第 3 步，否则请检查是否正确安装并初始化了 aide 工具。

3. 使用 `cron` 工具调度和执行文件系统检查：

- 打开定时任务编辑：

```
# crontab -u root -e
```

- 写入以下内容：

```
0 5 * * * /usr/sbin/aide --check
```

以上内容表示：每 5 小时执行一次文件系统检查。

扫描检测

确保定期检查文件系统完整性。

1. 执行以下命令，验证 `cron` 作业项目：

```
# crontab -u root -l | grep aide
0 5 * * * /usr/sbin/aide --check
```

如返回 aide 的定时任务，则视为通过此项检查。

参考

4.10 确保设置了 bootloader 密码

安全等级

- Level 2

描述

需要配置引导加密，要求所有人在 grub 设置启动参数之前必须进行密码验证。

这样做可以防止未经授权的用户进行启动参数修改或改变启动分区。（例如在启动时关闭 `SELinux` ）。

修复建议

设置 bootloader 密码。

1. 使用以下命令，配置 bootloader 密码：

```
# grub2-setpassword
Enter password: <password>
Confirm password: <password>
```

需自定义符合要求的密码。

2. 执行以下脚本更新 `grub2` 配置:

```
##!/usr/bin/env bash
GFCU()
{
 grubfile=$(find /boot -type f \( -name 'grubenv' -o -name 'grub.conf' -o -name
  ↪  'grub.cfg' \) -exec grep -Pl '^\h*(kernelopts=|linux|kernel)' {} \;)
 grubdir=$(dirname "$grubfile")
 grub2-mkconfig -o "$grubdir/grub.cfg"
}
GFCU
```

扫描检测

确保设置了 bootloader 密码。

1. 执行以下命令，验证是否配置了 bootloader 密码：

```
# grep -P '^\h*GRUB2_PASSWORD\h*=\h*.+$' /boot/grub2/user.cfg
GRUB2_PASSWORD=grub.pbkdf2.sha512.....
```

如返回结果符合预期，则视为通过此项检查。如返回为空或无此文件，则未通过此项检查。

参考

4.11 确保 bootloader 配置文件的权限配置正确

安全等级

- Level 1

描述

grub 文件包含启动信息和 bootloader 密码信息。grub2 的配置通常在 `grub.cfg` 文件中。

如系统使用 UEFI，则 `/boot/efi` 为 vfat 文件系统。vfat 文件系统本身没有权限的概念，但是可以在 Linux 下使用所需的任何权限进行挂载。

应将 grub 配置文件的权限配置为，仅 root 用户可访问，防止其他未授权用户读取或修改启动参数。

修复建议

修改 grub 配置文件的权限。

1. 使用以下命令，配置 `/boot/grub2/` 目录下，配置文件的权限及所有权：

```
## [ -f /boot/grub2/grub.cfg ] && chown root:root /boot/grub2/grub.cfg
## [ -f /boot/grub2/grub.cfg ] && chmod og-rwx /boot/grub2/grub.cfg
## [ -f /boot/grub2/grubenv ] && chown root:root /boot/grub2/grubenv
## [ -f /boot/grub2/grubenv ] && chmod og-rwx /boot/grub2/grubenv
## [ -f /boot/grub2/user.cfg ] && chown root:root /boot/grub2/user.cfg
## [ -f /boot/grub2/user.cfg ] && chmod og-rwx /boot/grub2/user.cfg
# osID=$(cat /etc/os-release | grep -Pi "^ID=" | cut -f2 -d= | sed -rn "s/\"//gp") ; [
 ↪  -f /boot/efi/EFI/$osID/grubenv ] && chown root:root /boot/efi/EFI/$osID/grubenv
# osID=$(cat /etc/os-release | grep -Pi "^ID=" | cut -f2 -d= | sed -rn "s/\"//gp") ; [
 ↪  -f /boot/efi/EFI/$osID/grubenv ] && chmod og-rwx /boot/efi/EFI/$osID/grubenv
```

2. 如为 UEFI 模式，则需编辑 `/etc/fstab` 文件，添加以下参数：

```
<device> /boot/efi vfat defaults,umask=0027,fmask=0077,uid=0,gid=0 0 0
```

* 可能需要重启系统以启用更改。

扫描检测

确保 bootloader 配置文件的权限配置正确。

1. 执行以下命令，验证 bootloader 配置文件的权限配置正确：

```
##!/usr/bin/env bash
GFPT()
{
 tst1="" tst2="" tst3="" tst4="" tst5="" tst6="" output="" output2="" output3=""
  ↪ output4="" output5="" output6=""
 grubfile=$(find /boot -type f \( -name 'grubenv' -o -name 'grub.conf' -o -name
  ↪ 'grub.cfg' \) -exec grep -Pl '^\h*(kernelopts=|linux|kernel)' {} \;)
 grubdir=$(dirname "$grubfile")
 stat -c "%a" "$grubfile" | grep -Pq '^\h*[0-7]00$' && tst1=pass output="Permissions
  ↪ on \"$grubfile\" are \"$(stat -c "%a" "$grubfile")\""
 stat -c "%u:%g" "$grubfile" | grep -Pq '^\h*0:0$' && tst2=pass output2="\"$grubfile\"
  ↪ is owned by \"$(stat -c "%U" "$grubfile")\" and belongs to group \"$(stat -c "%G"
  ↪ "$grubfile")\""
 if [ -f "$grubdir/user.cfg" ]; then
 stat -c "%a" "$grubdir/user.cfg" | grep -Pq '^\h*[0-7]00$' && tst3=pass
  ↪ output3="Permissions on \"$grubdir/user.cfg\" are \"$(stat -c "%a" "$grubdir/
  ↪ user.cfg")\""
 stat -c "%u:%g" "$grubdir/user.cfg" | grep -Pq '^\h*0:0$' && tst4=pass
  ↪ output4="\"$grubdir/user.cfg\" is owned by \"$(stat -c "%U" "$grubdir/
  ↪ user.cfg")\" and belongs to group \"$(stat -c "%G" "$grubdir/user.cfg")\""
 else
 tst3=pass;tst4=pass
 fi
 if [ -f "$grubdir/grub.cfg" ]; then
 stat -c "%a" "$grubdir/grub.cfg" | grep -Pq '^\h*[0-7]00$' && tst5=pass
  ↪ output5="Permissions on \"$grubdir/grub.cfg\" are \"$(stat -c "%a" "$grubdir/
  ↪ grub.cfg")\""
 stat -c "%u:%g" "$grubdir/grub.cfg" | grep -Pq '^\h*0:0$' && tst6=pass
  ↪ output6="\"$grubdir/grub.cfg\" is owned by \"$(stat -c "%U" "$grubdir/
  ↪ grub.cfg")\" and belongs to group \"$(stat -c "%G" "$grubdir/grub.cfg")\""
```

```
 else
 tst5=pass;tst6=pass
 fi
 if [ "$tst1" = "pass" ] && [ "$tst2" = "pass" ] && [ "$tst3" = "pass" ] && [ "$tst4"
  ↪  = "pass" ] && [ "$tst5" = "pass" ] && [ "$tst6" = "pass" ]; then
 echo "PASSED: "
 else
 echo "FAILED: "
 fi
 [ -n "$output" ] && echo "$output";[ -n "$output2" ] && echo "$output2";[ -n
 ↪  "$output3" ] && echo "$output3" [ -n "$output4" ] && echo "$output4";[ -n
 ↪  "$output5" ] && echo "$output5";[ -n "$output6" ] && echo "$output6"
}
GFPT
```

如返回 `PASSED` ，则视为通过此项检查。如返回 `FAILED` ，则未通过此项检查。

参考

4.12 确保进入单用户模式需要进行身份验证

安全等级

- Level 1

描述

单用户模式（救援模式）：Linux 的单用户模式有些类似 Windows 的安全模式，只启动最少的程序用于系统修复。在单用户模式（运行级别为 1）中，Linux 引导进入根 Shell，网络被禁用，只有少数进程运行。单用户模式可以用来修改文件系统损坏、还原配置文件、移动用户数据等。

要求在进入单用户模式（救援模式）时进行身份验证，以防止未经授权用户进入此模式，对系统进行破坏。

修复建议

配置单用户模式身份验证。

1. 编辑 `/usr/lib/systemd/system/rescue.service` 文件，并添加以下参数：

```
ExecStart=-/usr/lib/systemd/systemd-sulogin-shell rescue
```

2. 编辑 `/usr/lib/systemd/system/emergency.service` 文件，并添加以下参数：

```
ExecStart=-/usr/lib/systemd/systemd-sulogin-shell emergency
```

扫描检测

确保进入单用户模式需要进行身份验证。

1. 执行以下命令，单用户模式身份验证是否正确配置：

```
# grep /systemd-sulogin-shell /usr/lib/systemd/system/rescue.service
ExecStart=-/usr/lib/systemd/systemd-sulogin-shell rescue
```

```
# grep /systemd-sulogin-shell /usr/lib/systemd/system/emergency.service
ExecStart=-/usr/lib/systemd/systemd-sulogin-shell emergency
```

如返回值符合要求，则视为通过此项检查。

参考

4.13 确保核心转储受到限制

安全等级

- Level 1

描述

当程序运行的过程中异常终止或崩溃，操作系统会将程序当时的内存状态记录下来，保存在一个文件中，这种行为就叫做核心转储 (core dumps)。它还可以用于从核心文件中收集机密信息。

系统自身提供了核心转储的限制功能，但用户可能会重写这个限制。对核心转储设置硬性限制可以防止其被用户覆盖。

修复建议

配置单用户模式身份验证。

1. 编辑 `/etc/security/limits.conf` 文件，修改或添加 `* hard core 0` 参数：

```
egrep -q "^(\s*)\*\s+hard\s+core\s+\S+(\s*#.*)?\s*$" /etc/security/limits.conf && sed
 ↪  -ri "s/^(\s*)\*\s+hard\s+core\s+\S+(\s*#.*)?\s*$/\1* hard core 0\2/" /etc/
 ↪  security/limits.conf || echo "* hard core 0" >> /etc/security/limits.conf
```

2. 编辑 `/etc/sysctl.conf` 文件，修改或添加 `fs.suid_dumpable = 0` 参数：

```
egrep -q "^(\s*)fs.suid_dumpable\s*=\s*\S+(\s*#.*)?\s*$" /etc/sysctl.conf && sed -ri
 ↪  "s/^(\s*)fs.suid_dumpable\s*=\s*\S+(\s*#.*)?\s*$/\1fs.suid_dumpable = 0\2/" /etc/
 ↪  sysctl.conf || echo "fs.suid_dumpable = 0" >> /etc/sysctl.conf
```

3. 执行以下命令，设置活动内核参数：

```
# sysctl -w fs.suid_dumpable=0
```

扫描检测

确保核心转储受到限制。

1. 执行以下命令，检查输出结果是否匹配：

```
# grep -E "^\s*\*\s+hard\s+core" /etc/security/limits.conf
* hard core 0
# sysctl fs.suid_dumpable
fs.suid_dumpable = 0
# grep "fs\.suid_dumpable" /etc/sysctl.conf /etc/sysctl.d/*
/etc/sysctl.conf:fs.suid_dumpable = 0
/etc/sysctl.d/99-sysctl.conf:fs.suid_dumpable = 0
```

如返回值符合要求，则视为通过此项检查。

参考

4.14 确保地址空间布局随机化（ASLR）被启用

安全等级

- Level 1

描述

ASLR (Address Space Layout Randomization，地址空间布局随机化)，是一种针对缓冲区溢出的安全保护技术。借助 ASLR，PE 文件每次加载到内存的起始地址都会随机变化。

修复建议

启用 ASLR。

1. 编辑 `/etc/sysctl.conf` 或 `/etc/sysctl.d/*` 文件，添加 `kernel.randomize_va_space = 2` 参数：

```
printf "
kernel.randomize_va_space = 2
" >> /etc/sysctl.d/50-kernel_sysctl.conf
```

2. 执行以下命令，设置活动内核参数：

```
# sysctl -w kernel.randomize_va_space=2
```

扫描检测

确保地址空间布局随机化（ASLR）被启用。

1. 执行以下命令，检查输出结果是否匹配：

```
# sysctl kernel.randomize_va_space
kernel.randomize_va_space = 2
# grep -s -- "kernel\.randomize_va_space" /run/sysctl.d/*.conf /etc/sysctl.d/*.conf /
 ↪  usr/local/lib/sysctl.d/*.conf /usr/lib/sysctl.d/*.conf /lib/sysctl.d/*.conf /etc/
 ↪  sysctl.conf
```

```
/etc/sysctl.d/99-sysctl.conf:kernel.randomize_va_space = 2
/etc/sysctl.conf:kernel.randomize_va_space = 2
```

如返回值符合要求，则视为通过此项检查。

参考

4.15 确保系统全局加密策略符合要求

安全等级

- Level 1

描述

系统全局加密策略，目前有以下等级：

- LEGACY
- DEFAULT
- FUTURE
- FIPS

Legacy 等级的加密策略包含（TLS 1.0、TLS 1.1、SSH2 协议或更高版本、DSA、3DES、RC4）等。但这些加密策略目前已不够安全，建议使用 DEFAULT 等级。

修复建议

配置系统加密策略为 DEFAULT。

1. 执行以下命令，配置系统加密策略等级：

```
# update-crypto-policies --set DEFAULT
```

2. 执行以下命令，激活加密策略：

```
# update-crypto-policies
```

默认系统加密策略为：`DEFAULT` 。

扫描检测

确保系统全局加密策略符合要求。

1. 执行以下命令，检查系统加密策略是否符合要求：

```
# grep -E -i '^\s*LEGACY\s*(\s+#.*)?$' /etc/crypto-policies/config
No information is returned.
```

如没有任何返回值，则视为通过此项检查。

参考

4.16 确保所有全局可写目录都设置了 sticky 位

安全等级

- Level 1

描述

设置全局可写目录的 sticky 位可以防止用户删除或重命名该目录中不属于自己的文件。

修复建议

配置全局可写目录 sticky 位。

1. 执行以下命令，配置全局可写目录 sticky 位：

```
# df --local -P | awk '{if (NR!=1) print $6}' | xargs -I '{}' find '{}' -xdev -type d
 ↪  \( -perm -0002 -a ! -perm -1000 \) 2>/dev/null | xargs -I '{}' chmod a+t '{}'
```

扫描检测

确保所有全局可写目录都设置了 sticky 位。

1. 执行以下命令，检查全局可写目录 sticky 位是否正确配置：

```
# df --local -P | awk '{if (NR!=1) print $6}' | xargs -I '{}' find '{}' -xdev -type d
 ↪  \( -perm -0002 -a ! -perm -1000 \) 2>/dev/null
```

如没有任何返回值，则视为通过此项检查。

参考

4.17 确保 /etc/passwd 文件权限配置正确

安全等级

- Level 1

描述

`/etc/passwd` 文件中也存储了所有用户账户信息，所有用户都需要读取此文件。也包含了许多系统级应用所使用的用户账户信息（虚拟用户），这些应用对 `passwd` 文件也必须有可读权限，才可正常运行。

正由于其对所有用户都可读，因此，`/etc/passwd` 文件的安全性要求是非常高的。需要确保 `/etc/passwd` 文件不被未经授权的用户访问或修改。

修复建议

配置 `/etc/passwd` 文件的所有者与权限。

1. 执行以下命令，配置 `/etc/passwd` 文件的所有者与权限：

```
# chown root:root /etc/passwd
# chmod 644 /etc/passwd
```

扫描检测

确保 `/etc/passwd` 文件权限配置正确。

1. 执行以下命令， `/etc/passwd` 文件权限配置是否正确：

```
# stat /etc/passwd
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
```

如返回结果中：`Uid` 与 `Gid` 均为 `(0/root)` ，且 `Access` 为 `0644` 或更严格的限制，则视为通过此项检查。

参考

4.18 确保 /etc/shadow 文件权限配置正确

安全等级

- Level 1

描述

`/etc/shadow` 文件用于存储系统内用户账户的重要安全信息，如哈希密码等。

如果攻击者获得了 `/etc/shadow` 文件的读取权限，就可以轻易的获得用户的哈希密码，并运行密码破解程序来破解它。存储在 `/etc/shadow` 文件中的其他安全信息（如过期时间等）也可以用来对用户账户进行破坏。

因此，`/etc/shadow` 文件的安全性要求是非常高的。需要确保 `/etc/shadow` 文件不被未经授权的用户访问或修改。

修复建议

配置 `/etc/shadow` 文件的所有者与权限。

1. 执行以下命令，配置 `/etc/shadow` 文件的所有者与权限：

```
# chown root:root /etc/shadow
# chmod 0000 /etc/shadow
```

扫描检测

确保 `/etc/shadow` 文件权限配置正确。

1. 执行以下命令，`/etc/shadow` 文件权限配置是否正确：

```
# stat /etc/shadow
Access: (0000/----------)  Uid: (    0/    root)   Gid: (    0/    root)
```

如返回结果中：`Uid` 与 `Gid` 均为 `(0/root)`，且 `Access` 为 `0000`，则视为通过此项检查。

参考

4.19 确保 /etc/group 文件权限配置正确

安全等级

- Level 1

描述

`/etc/group` 文件包含了系统中所有有效用户组的列表。其权限应配置为，`root` 用户对其有读写权限，其余用户对其仅有读权限。

修复建议

配置 `/etc/group` 文件的所有者与权限。

1. 执行以下命令，配置 `/etc/group` 文件的所有者与权限：

```
# chown root:root /etc/group
# chmod u-x,g-wx,o-wx /etc/group
```

扫描检测

确保 `/etc/group` 文件权限配置正确。

1. 执行以下命令，`/etc/group` 文件权限配置是否正确：

```
# stat /etc/group
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
```

如返回结果中：`Uid` 与 `Gid` 均为 `(0/root)`，且 `Access` 为 `0644` 或更严格的限制，则视为通过此项检查。

参考

4.20 确保 /etc/gshadow 文件权限配置正确

安全等级

- Level 1

描述

`/etc/gshadow` 文件用于存储系统内用户组的重要安全信息，如哈希密码等。

如果攻击者获得了 `/etc/gshadow` 文件的读取权限，就可以轻易的获得用户组的哈希密码，并运行密码破解程序来破解它。存储在 `/etc/gshadow` 文件中的其他安全信息（如组管理员等）也可以用来对用户组进行破坏。

因此，`/etc/gshadow` 文件的安全性要求是非常高的。需要确保 `/etc/gshadow` 文件不被未经授权的用户访问或修改。

修复建议

配置 `/etc/gshadow` 文件的所有者与权限。

1. 执行以下命令，配置 `/etc/gshadow` 文件的所有者与权限：

```
# chown root:root /etc/gshadow
# chmod 0000 /etc/gshadow
```

扫描检测

确保 `/etc/gshadow` 文件权限配置正确。

1. 执行以下命令，`/etc/gshadow` 文件权限配置是否正确：

```
# stat /etc/gshadow
Access: (0000/----------)  Uid: (    0/    root)   Gid: (    0/    root)
```

如返回结果中：`Uid` 与 `Gid` 均为 `(0/root)`，且 `Access` 为 `0000`，则视为通过此项检查。

参考

4.21 确保 /etc/passwd- 文件权限配置正确

安全等级

- Level 1

描述

`/etc/passwd-` 文件中包含备份的用户帐号信息。

需要确保 `/etc/passwd-` 文件不被未经授权的用户访问或修改。

修复建议

配置 `/etc/passwd-` 文件的所有者与权限。

1. 执行以下命令，配置 `/etc/passwd-` 文件的所有者与权限：

```
# chown root:root /etc/passwd-
# chmod u-x,go-wx /etc/passwd-
```

扫描检测

确保 `/etc/passwd-` 文件权限配置正确。

1. 执行以下命令， `/etc/passwd-` 文件权限配置是否正确：

```
# stat /etc/passwd-
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
```

如返回结果中： `Uid` 与 `Gid` 均为 `(0/root)` ，且 `Access` 为 `0644` 或更严格的限制，则视为通过此项检查。

参考

4.22 确保 /etc/shadow- 文件权限配置正确

安全等级

- Level 1

描述

`/etc/shadow-` 文件用于备份系统内用户账户安全信息，如哈希密码等。

如果攻击者获得了 `/etc/shadow-` 文件的读取权限，就可以轻易的获得用户的哈希密码，并运行密码破解程序来破解它。存储在 `/etc/shadow-` 文件中的其他安全信息（如过期时间等）也可以用来对用户账户进行破坏。

需要确保 `/etc/shadow-` 文件不被未经授权的用户访问或修改。

修复建议

配置 `/etc/shadow-` 文件的所有者与权限。

1. 执行以下命令，配置 `/etc/shadow-` 文件的所有者与权限：

```
# chown root:root /etc/shadow-
# chmod 0000 /etc/shadow-
```

扫描检测

确保 `/etc/shadow-` 文件权限配置正确。

1. 执行以下命令， `/etc/shadow-` 文件权限配置是否正确：

```
# stat /etc/shadow-
Access: (0000/----------)  Uid: (    0/    root)   Gid: (    0/    root)
```

如返回结果中： `Uid` 与 `Gid` 均为 `(0/root)` ，且 `Access` 为 `0000` ，则视为通过此项检查。

参考

4.23 确保 /etc/group- 文件权限配置正确

安全等级

- Level 1

描述

`/etc/group-` 文件包含了系统中所有有效用户组列表的备份。其权限应配置为，`root` 用户对其有读写权限，其余用户对其仅有读权限。

修复建议

配置 `/etc/group-` 文件的所有者与权限。

1. 执行以下命令，配置 `/etc/group-` 文件的所有者与权限：

```
# chown root:root /etc/group-
# chmod u-x,go-wx /etc/group-
```

扫描检测

确保 `/etc/group-` 文件权限配置正确。

1. 执行以下命令，`/etc/group-` 文件权限配置是否正确：

```
# stat /etc/group-
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
```

如返回结果中：`Uid` 与 `Gid` 均为 `(0/root)`，且 `Access` 为 `0644` 或更严格的限制，则视为通过此项检查。

参考

4.24 确保 /etc/gshadow- 文件权限配置正确

安全等级

- Level 1

描述

`/etc/gshadow-` 文件备份了系统内用户组的重要安全信息，如哈希密码等。

如果攻击者获得了 `/etc/gshadow-` 文件的读取权限，就可以轻易的获得用户组的哈希密码，并运行密码破解程序来破解它。

需要确保 `/etc/gshadow-` 文件不被未经授权的用户访问或修改。

修复建议

配置 `/etc/gshadow-` 文件的所有者与权限。

1. 执行以下命令，配置 `/etc/gshadow-` 文件的所有者与权限：

```
# chown root:root /etc/gshadow-
# chmod 0000 /etc/gshadow-
```

扫描检测

确保 `/etc/gshadow-` 文件权限配置正确。

1. 执行以下命令， `/etc/gshadow-` 文件权限配置是否正确：

```
# stat /etc/gshadow-
Access: (0000/----------)  Uid: (    0/    root)   Gid: (    0/    root)
```

如返回结果中： `Uid` 与 `Gid` 均为 `(0/root)` ，且 `Access` 为 `0000` ，则视为通过此项检查。

参考

4.25 确保没有所有人可写的文件

安全等级

- Level 2

描述

所有人可写的文件中的数据可以被系统中的任何用户修改和破坏，极易被未授权用户注入恶意脚本。所有人可写的文件很有可能是一个恶意脚本或程序，在其被执行后，可能对系统造成破坏。

所以需要扫描出系统内所有人可写的文件，对其进行甄别与清理。

修复建议

建议删除所有人可写的文件，但在操作前，应阅读供应商或软件服务文档，以免破坏正常程序的依赖文件。

扫描检测

确保没有所有人可写的文件。

1. 执行以下命令，检查是否有返回结果：

```
# df --local -P | awk '{if (NR!=1) print $6}' | xargs -I '{}' find '{}' -xdev -type f
 ↪  -perm -0002
Nothing should be returned
```

如没有任何返回结果，则视为通过此项检查。

参考

4.26 确保所有文件或目录都配置了所有者

安全等级

- Level 2

描述

当系统管理员从 `/etc/passwd` 中删除用户时，如果忘记从系统中删除这些用户拥有的文件，那么这些被删除的用户 ID 在重新分配给新用户后，新用户可能会直接拥有这些文件，从而获得比预期更多的文件访问权限。

所以，定期对这些无主文件或目录进行权限的重新分配或清理是很有必要的。

修复建议

建议对无主文件或目录的拥有者权限进行重新分配：可使用 `chown` 命令，将这些文件或目录的拥有者重置为系统上的某个活动用户。但在操作前，应阅读供应商或软件服务文档，以免破坏正常程序的依赖文件。

扫描检测

确保所有文件或目录都配置了所有者。

1. 执行以下命令，检查是否有返回结果：

```
# df --local -P | awk {'if (NR!=1) print $6'} | xargs -I '{}' find '{}' -xdev -nouser
Nothing should be returned
```

如没有任何返回结果，则视为通过此项检查。

参考

4.27 确保所有文件或目录都配置了所属组

安全等级

- Level 2

描述

当系统管理员删除用户组时，如果忘记从系统中删除这些用户组拥有的文件，那么这些被删除的用户组 ID 在重新分配给新用户后，新用户可能会直接拥有这些文件，从而获得比预期更多的文件访问权限。

所以，定期对这些无所属组的文件或目录进行权限的重新分配或清理是很有必要的。

修复建议

建议对无所属组的文件或目录的组权限进行重新分配：可使用 `chown` 命令，将这些文件或目录的所属组重置为系统上的某个活动用户组。但在操作前，应阅读供应商或软件服务文档，以免破坏正常程序的依赖文件。

扫描检测

确保所有文件或目录都配置了所属组。

1. 执行以下命令，检查是否有返回结果：

```
# df --local -P | awk '{if (NR!=1) print $6}' | xargs -I '{}' find '{}' -xdev -nogroup
Nothing should be returned
```

如没有任何返回结果，则视为通过此项检查。

参考

4.28 确保所有用户的密码不为空

安全等级

- Level 2

描述

如果一个用户的密码为空，则代表任何人都可以以该用户身份登录系统。这对于操作系统来说非常不安全，未经授权登录的用户可能会对系统进行恶意破坏或盗取敏感信息。

所以，系统内所有用户账户必须配置了密码，没有密码的用户，必须为锁定状态。

修复建议

锁定没有密码的用户账户，不允许其登录。

1. 可使用以下命令，对用户进行锁定。

```
# passwd -l <username>
```

`<username>` 为需要被锁定的用户名。

扫描检测

确保所有用户的密码不为空。

1. 执行以下命令，检查是否有返回结果：

```
# awk -F: '($2 == "" ) { print $1 " does not have a password "}' /etc/shadow
Nothing should be returned
```

如没有任何返回结果，则视为通过此项检查。

参考

4.29 确保 root 用户 PATH 环境变量内所有目录的权限配置符合要求

安全等级

- Level 2

描述

PATH 环境变量的内容是由一个个目录组成的，各目录之间用冒号 `:` 隔开。当执行某个命令时，Linux 会依照 PATH 中包含的目录依次搜寻该命令的可执行文件，一旦找到，则会立即执行。

在 Linux 系统中，root 用户的权限极大，几乎可执行任何命令。攻击者可通过替换或修改 root 用户 PATH 变量涉及的目录下的可执行文件，导致 root 用户在执行基础命令时，无意识的使用 root 权限执行恶意程序，对系统进行攻击。

所以 root 用户 PATH 变量内所有目录的权限配置极为重要，需保证仅有 root 用户对这些目录有完全的管理权限，对于其他用户来说没有可写权限，避免其中可执行文件被未授权用户替换或篡改。

修复建议

规范 root 用户 PATH 变量中所有目录的权限。

- 需根据检查情况，自行判断，对不符合安全要求的目录权限进行规范。

1. 如目录权限不符合要求，则执行以下命令，将目录权限配置为 `755` 或更加严格：

```
# chmod 755 <path>
```

`<path>` 为需要修改权限的目录路径。

2. 如目录所有者或所属组不符合要求，则执行以下命令，将目录所有者和所属组配置为 `root` ：

```
# chown root:root <path>
```

`<path>` 为需要修改权限的目录路径。

3. 如 PATH 变量内有不存在的目录，则执行以下命令，创建该目录，并执行第 1、2 步命令，对其权限进行配置：

```
# mkdir -p <path>
```

`<path>` 为需要创建的目录路径。

扫描检测

确保 root 用户 PATH 环境变量内所有目录的权限配置符合要求。

1. 执行以下命令，检查是否有返回结果：

```
##!/bin/bash
RPCV="$(sudo -Hiu root env | grep '^PATH=' | cut -d= -f2)"
echo "$RPCV" | grep -q "::" && echo "root's path contains a empty directory (::)"
echo "$RPCV" | grep -q ":$" && echo "root's path contains a trailing (:)"
for x in $(echo "$RPCV" | tr ":" " "); do
 if [ -d "$x" ]; then
 ls -ldH "$x" | awk '$9 == "." {print "PATH contains current working directory (.)"}
 $3 != "root" {print $9, "is not owned by root"}
 substr($1,6,1) != "-" {print $9, "is group writable"}
 substr($1,9,1) != "-" {print $9, "is world writable"}'
 else
 echo "$x is not a directory"
 fi
done
```

如没有任何返回结果，则视为通过此项检查。

参考

4.30 确保 UID 为 0 的用户只有 root

安全等级

- Level 2

描述

在 Linux 操作系统中，任何 UID 为 0 的用户，都具有超级用户权限。

需确保系统中，只有 root 用户的 UID 为 0。

修复建议

检查 `/etc/passwd` 文件中所有用户的 UID：删除除 root 用户以外，所有 UID 为 0 的用户，或为他们分配一个新的 ID。

例：

```
# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
```

`/etc/passwd` 文件格式：`用户名：密码:UID:GID：描述信息：主目录：默认 shell`

扫描检测

确保 UID 为 0 的用户只有 root。

1. 执行以下命令，检查返回结果：

```
# awk -F: '($3 == 0) { print $1 }' /etc/passwd
root
```

如返回结果为 `root` ，则视为通过此项检查。

参考

4.31 确保用户的主目录权限为 750 或更严格

安全等级

- Level 1

描述

在创建新用户时，系统管理员会为用户的主目录配置符合安全要求的权限，但后期用户可以很容易地修改这些权限。

如果用户主目录的权限对 group 与 other 可写，可能使恶意用户窃取或修改其他用户的数据，或获得其他用户的系统特权。

建议对所有用户的主目录权限进行限制，配置为 750 或更严格权限策略。

修复建议

配置用户主目录的权限。

- 在不通知用户的情况下对用户主目录进行全局修改可能会导致程序意外中断和用户不满。因此建议建立监控策略，及时上报用户文件权限，并根据实际情况，判断采取的措施。

1. 执行以下脚本，删除用户主目录 750 以上的权限：

```
##!/bin/bash
awk -F: '($1!~/(halt|sync|shutdown|nfsnobody)/ &&
$7!~/^(\/usr)?\/sbin\/nologin(\/)?$/ && $7!~/(\/usr)?\/bin\/false(\/)?$/) {print $6}'
 ↪  /etc/passwd | while read -r dir; do
 if [ -d "$dir" ]; then
    dirperm=$(stat -L -c "%A" "$dir")
    if [ "$(echo "$dirperm" | cut -c6)" != "-" ] || [ "$(echo "$dirperm" | cut -c8)" !
     ↪  = "-" ] || [ "$(echo "$dirperm" | cut -c9)" != "-" ] || [ "$(echo "$dirperm" |
     ↪  cut -c10)" != "-" ]; then
        chmod g-w,o-rwx "$dir"
    fi
```

```
 fi
done
```

扫描检测

确保用户的主目录权限为 750 或更严格。

1. 执行以下脚本，检查返回结果：

```
##!/bin/bash
awk -F: '($1!~/(halt|sync|shutdown|nfsnobody)/ &&
$7!~/^(\/usr)?\/sbin\/nologin(\/)?$/ && $7!~/(\/usr)?\/bin\/false(\/)?$/) {print $1 "
 ↪  " $6}' /etc/passwd | while read -r user dir; do
 if [ ! -d "$dir" ]; then
 echo "User: \"$user\" home directory: \"$dir\" doesn't exist"
 else
    dirperm=$(stat -L -c "%A" "$dir")
    if [ "$(echo "$dirperm" | cut -c6)" != "-" ] || [ "$(echo "$dirperm" | cut -c8)" !
     ↪  = "-" ] || [ "$(echo "$dirperm" | cut -c9)" != "-" ] || [ "$(echo "$dirperm" |
     ↪  cut -c10)" != "-" ]; then
 echo "User: \"$user\" home directory: \"$dir\" has permissions: \"$(stat -L -c "%a"
  ↪  "$dir")\""
    fi
 fi
done
```

如没有任何返回值，则视为通过此项检查。

参考

4.32 确保用户拥有自己的主目录

安全等级

- Level 1

描述

用户主目录（家目录）是为每个普通用户定义的主工作目录，用于设置用户本地环境变量和存储用户个人文件。

每个用户对存储在自己主目录中的文件负责，因此所有用户（包括 root ）必须是自己主目录的所有者。

修复建议

配置用户主目录的所有权。

1. 执行以下脚本，将对所有权配置错误的用户主目录进行修正：

```
##!/bin/bash
awk -F: '($1!~/(halt|sync|shutdown|nfsnobody)/ && $7!~/^(\/usr)?\/sbin\/nologin(\/)?$/
 ↪  && $7!~/(\/usr)?\/bin\/false(\/)?$/) { print $1 " " $6 }' /etc/passwd | while read
 ↪  -r user dir; do
 if [ ! -d "$dir" ]; then
    echo "User: \"$user\" home directory: \"$dir\" does not exist, creating home
     ↪  directory"
    mkdir "$dir"
    chmod g-w,o-rwx "$dir"
    chown "$user" "$dir"
 else
    owner=$(stat -L -c "%U" "$dir")
        if [ "$owner" != "$user" ]; then
            chmod g-w,o-rwx "$dir"
            chown "$user" "$dir"
        fi
```

```
 fi
done
```

扫描检测

确保用户拥有自己的主目录。

1. 执行以下脚本，检查返回结果：

```
##!/usr/bin/env bash
UHOC()
{
for i in $( awk -F: '($1!~/(halt|sync|shutdown|nfsnobody)/ && $7!~/^(\/usr)?\/sbin\/
 ↪  nologin(\/)?$/ && $7!~/(\/usr)?\/bin\/false(\/)?$/) {print $1":"$6}' /etc/passwd);
 ↪  do
 output=''
 output2=''
 user=$(echo "$i" | cut -d: -f1)
 dir=$(echo "$i" | cut -d: -f2)
 if [ ! -d "$dir" ]; then
 [ -z "$output2" ] && output2="The following users' home directories don't exist:
 ↪  \"$user\"" || output2="$output2, \"$user\""
 echo  $output2
 else
    owner="$(stat -L -c "%U" "$dir")"
    if [ "$owner" != "$user" ] && [ "$owner" != "root" ]; then
    [ -z "$output" ] && output="The following users' don't own their home directory:
 ↪  \"$user\" home directory is owned by \"$owner\"" || output="$output, \"$user\"
 ↪  home directory is owned by \"$owner\""
    echo  $output
    fi
 fi
done
}
UHOC
```

如没有任何返回值，则视为通过此项检查。

参考

4.33 确保用户的 dot 文件权限配置正确

安全等级

- Level 1

描述

在 Linux 下，各种软件的配置文件大多存储于以 `.` 开头以 `rc` 结尾的文件中并存放于用户的主目录 `~/` 中，也就是俗称的 dotfile 者 rcfile，例如 zsh 的配置文件 `.zshrc` ，vim 的配置文件 `.vimrc` 等等。

如果用户的 dotfile 权限对 group 与 other 可写，可能使恶意用户窃取或修改其他用户的配置数据。

修复建议

- 在不通知用户的情况下对用户的 dotfile 进行全局修改可能会导致程序意外中断和用户不满。因此建议建立监控策略，及时上报用户 dotfile 权限，并根据实际情况，判断采取的措施。

1. 执行以下脚本，删除用户 dotfile 多余的权限：

```
##!/bin/bash
awk -F: '($1!~/(halt|sync|shutdown|nfsnobody)/ && $7!~/^(\/usr)?\/sbin\/nologin(\/)?$/
 ↪  && $7!~/(\/usr)?\/bin\/false(\/)?$/) { print $6 }' /etc/passwd | while read -r
 ↪  dir; do
if [ -d "$dir" ]; then
    for file in "$dir"/.*; do
        if [ ! -h "$file" ] && [ -f "$file" ]; then
            fileperm=$(stat -L -c "%A" "$file")
            if [ "$(echo "$fileperm" | cut -c6)" != "-" ] || [ "$(echo "$fileperm" |
             ↪  cut -c9)" != "-" ]; then
                chmod go-w "$file"
            fi
        fi
    done
```

```
fi
done
```

扫描检测

确保用户的 dot 文件权限配置正确。

1. 执行以下脚本，检查返回结果：

```
##!/bin/bash
awk -F: '($1!~/(halt|sync|shutdown|nfsnobody)/ && $7!~/^(\/usr)?\/sbin\/nologin(\/)?$/
 ↪  && $7!~/(\/usr)?\/bin\/false(\/)?$/) { print $1 " " $6 }' /etc/passwd | while read
 ↪  -r user dir; do
if [ -d "$dir" ]; then
    for file in "$dir"/.*; do
        if [ ! -h "$file" ] && [ -f "$file" ]; then
            fileperm=$(stat -L -c "%A" "$file")
            if [ "$(echo "$fileperm" | cut -c6)" != "-" ] || [ "$(echo "$fileperm" |
             ↪  cut -c9)" != "-" ]; then
                echo "User: \"$user\" file: \"$file\" has permissions: \"$fileperm\""
            fi
        fi
    done
fi
done
```

如没有任何返回值，则视为通过此项检查。

参考

4.34 确保没有用户拥有.forward 文件

安全等级

- Level 1

描述

`.forward` 文件指定了一个电子邮件地址来转发用户的邮件。

使用 `.forward` 文件会带来额外的安全风险，容易导致敏感数据在无意中被转发到可信组织外。`.forward` 文件本身也有被用来执行恶意代码的风险。

修复建议

- 在不通知用户的情况下对用户 `.forward` 文件进行全局修改可能会导致程序意外中断和用户不满。因此建议建立监控策略，及时上报用户 `.forward` 文件使用情况，并根据实际情况，判断采取的措施。

1. 执行以下脚本，删除用户主目录下的 `.forward` 文件：

```
##!/bin/bash
awk -F: '($1!~/(halt|sync|shutdown|nfsnobody)/ && $7!~/^(\/usr)?\/sbin\/nologin(\/)?$/
 ↪  && $7!~/(\/usr)?\/bin\/false(\/)?$/) {
print $6 }' /etc/passwd | while read -r dir; do
 if [ -d "$dir" ]; then
    file="$dir/.forward"
    [ ! -h "$file" ] && [ -f "$file" ] && rm -rf "$file"
 fi
done
```

扫描检测

确保没有用户拥有 `.forward` 文件。

1. 执行以下脚本，检查返回结果：

```
##!/bin/bash
awk -F: '($1!~/(halt|sync|shutdown|nfsnobody)/ && $7!~/^(\/usr)?\/sbin\/nologin(\/)?$/
 ↪  && $7!~/(\/usr)?\/bin\/false(\/)?$/) { print $1 " " $6 }' /etc/passwd | while read
 ↪  -r user dir; do
 if [ -d "$dir" ]; then
    file="$dir/.forward"
    if [ ! -h "$file" ] && [ -f "$file" ]; then
        echo "User: \"$user\" file: \"$file\" exists"
    fi
 fi
done
```

如没有任何返回值，则视为通过此项检查。

参考

4.35 确保没有用户拥有.netrc 文件

安全等级

- Level 1

描述

`.netrc` 文件包含用于登录远程主机的数据，以便通过 FTP 进行文件传输。

由于 `.netrc` 文件以明文形式存储密码，因此存在很大的安全风险。

修复建议

- 在不通知用户的情况下对用户 `.netrc` 文件进行全局修改可能会导致程序意外中断和用户不满。因此建议建立监控策略，及时上报用户 `.netrc` 文件使用情况，并根据实际情况，判断采取的措施。

1. 执行以下脚本，删除用户主目录下的 `.netrc` 文件：

```
##!/bin/bash
awk -F: '($1!~/(halt|sync|shutdown|nfsnobody)/ && $7!~/^(\/usr)?\/sbin\/nologin(\/)?$/
 ↪  && $7!~/(\/usr)?\/bin\/false(\/)?$/) {
print $6 }' /etc/passwd | while read -r dir; do
 if [ -d "$dir" ]; then
    file="$dir/.netrc"
    [ ! -h "$file" ] && [ -f "$file" ] && rm -rf "$file"
 fi
done
```

扫描检测

确保没有用户拥有 `.netrc` 文件。

1. 执行以下脚本，检查返回结果：

```
##!/bin/bash
awk -F: '($1!~/(halt|sync|shutdown|nfsnobody)/ && $7!~/^(\/usr)?\/sbin\/nologin(\/)?$/
 ↪  && $7!~/(\/usr)?\/bin\/false(\/)?$/) { print $1 " " $6 }' /etc/passwd | while read
 ↪  -r user dir; do
 if [ -d "$dir" ]; then
    file="$dir/.netrc"
    if [ ! -h "$file" ] && [ -f "$file" ]; then
       echo "User: \"$user\" file: \"$file\" exists"
    fi
 fi
done
```

如没有任何返回值，则视为通过此项检查。

参考

4.36 确保用户.netrc 文件权限配置正确

安全等级

- Level 1

描述

`.netrc` 文件包含用于登录远程主机的数据，以便通过 FTP 进行文件传输。由于 `.netrc` 文件以明文形式存储密码，因此存在很大的安全风险。

如果确实需要一个 `.netrc` 文件，则需正确配置其权限，使其符合安全要求。应将其权限配置为 `600` 或更加严格。

修复建议

- 在不通知用户的情况下对用户 `.netrc` 文件进行全局修改可能会导致程序意外中断和用户不满。因此建议建立监控策略，及时上报用户 `.netrc` 文件使用情况，并根据实际情况，判断采取的措施。

1. 执行以下脚本，配置用户 `.netrc` 文件的权限：

```
##!/bin/bash
awk -F: '($1!~/(halt|sync|shutdown|nfsnobody)/ &&
$7!~/^(\/usr)?\/sbin\/nologin(\/)?$/ && $7!~/(\/usr)?\/bin\/false(\/)?$/) {
print $6 }' /etc/passwd | while read -r dir; do
 if [ -d "$dir" ]; then
    file="$dir/.netrc"
    [ ! -h "$file" ] && [ -f "$file" ] && chmod 600 "$file"
 fi
done
```

扫描检测

确保用户 `.netrc` 文件权限配置正确。

1. 执行以下脚本，检查返回结果：

```
##!/bin/bash
awk -F: '($1!~/(halt|sync|shutdown|nfsnobody)/ && $7!~/^(\/usr)?\/sbin\/nologin(\/)?$/
 ↪  && $7!~/(\/usr)?\/bin\/false(\/)?$/) { print $1 " " $6 }' /etc/passwd | while read
 ↪  -r user dir; do
 if [ -d "$dir" ]; then
    file="$dir/.netrc"
    if [ ! -h "$file" ] && [ -f "$file" ]; then
        if stat -L -c "%A" "$file" | cut -c4-10 | grep -Eq '[^-]+'; then
            echo "FAILED: User: \"$user\" file: \"$file\" exists with permissions: \"$
             ↪  (stat -L -c "%a" "$file")\", remove file or excessive permissions"
        else
            echo "WARNING: User: \"$user\" file: \"$file\" exists with permissions:
             ↪  \"$(stat -L -c "%a" "$file")\", remove file unless required"
        fi
    fi
 fi
done
```

查看返回值：

- FAILED：表示 `.netrc` 文件的权限不符合要求，应尽快对其权限进行限制，使其权限为 `600` 或更加严格。如非必要应删除此文件。
- WARNING：表示有用户主目录下存在 `.netrc` 文件，其权限符合要求，如非必要应删除此文件。

如返回值为 `WARNING` ，则视为通过此项检查。但仍需根据实际情况判断，如非必要应删除 `.netrc` 文件。删除文件的方法请参考 `4.35` 项目内的修复代码。

参考

4.37 确保没有用户拥有.rhosts 文件

安全等级

- Level 1

描述

`.rhosts` 文件是 `/etc/hosts.equiv` 文件的用户等效文件。此文件包含主机-用户组合列表，而不包含一般意义的主机。如果此文件中列出了主机-用户组合，则指定用户将被授予从指定主机登录而不必提供口令的权限。注意，`.rhosts` 文件必须驻留在用户起始目录的顶层。如果 `.rhost` 文件位于子目录中，则不会生效。

遗憾的是，`.rhosts` 文件存在严重的安全问题。`/etc/hosts.equiv` 文件受系统管理员的控制并且可以有效地管理，但任何用户都可以创建属于自己的 `.rhosts` 文件，从而可以在系统管理员不知情时对其选择的任何人授予访问权限。

如果所有用户起始目录都在一台服务器上，并且只有某些人员才在该服务器上具有超级用户权限，则防止用户使用 `.rhosts` 文件的一种好方法就是以超级用户身份在用户起始目录中创建一个空文件。然后，将此文件的权限更改为 `000` ，这样即使作为超级用户也很难更改它。这种方式可有效地防止用户因不负责任地使用 `.rhosts` 文件而导致的系统安全风险。但是，如果用户能够更改指向其起始目录的有效路径，则此更改将不能解决任何问题。

管理 `.rhosts` 文件的唯一安全方法是完全删除并禁用它们。作为系统管理员，需要经常检查系统以了解 `.rhosts` 文件的违规情况。

可能存在一种例外情况：即超级用户帐户可能需要使用 `.rhosts` 文件来执行网络备份和其他远程服务。

修复建议

- 在不通知用户的情况下对用户 `.rhosts` 文件进行全局修改可能会导致程序意外中断和用户不满。因此建议建立监控策略，及时上报用户 `.rhosts` 文件使用情况，并根据实际情况，判断采取的措施。

1. 执行以下脚本，删除用户主目录下的 `.rhosts` 文件：

```
##!/bin/bash
awk -F: '($1!~/(halt|sync|shutdown|nfsnobody)/ && $7!~/^(\/usr)?\/sbin\/nologin(\/)?$/
 ↪  && $7!~/(\/usr)?\/bin\/false(\/)?$/) { print $6 }' /etc/passwd | while read -r
 ↪  dir; do
    if [ -d "$dir" ]; then
        file="$dir/.rhosts"
        [ ! -h "$file" ] && [ -f "$file" ] && rm -rf "$file"
    fi
done
```

扫描检测

确保没有用户拥有 `.rhosts` 文件。

1. 执行以下脚本，检查返回结果：

```
##!/bin/bash
awk -F: '($1!~/(halt|sync|shutdown|nfsnobody)/ && $7!~/^(\/usr)?\/sbin\/nologin(\/)?$/
 ↪  && $7!~/(\/usr)?\/bin\/false(\/)?$/) { print $1 " " $6 }' /etc/passwd | while read
 ↪  -r user dir; do
    if [ -d "$dir" ]; then
        file="$dir/.rhosts"
        if [ ! -h "$file" ] && [ -f "$file" ]; then
            echo "User: \"$user\" file: \"$file\" exists"
        fi
    fi
done
```

如没有任何返回值，则视为通过此项检查。

参考

4.38 确保 /etc/passwd 中所有组都存在于 /etc/group 中

安全等级

- Level 2

描述

随着时间的推移，系统管理员的失误或更改可能导致有些在 `/etc/passwd` 中定义的组，在 `/etc/group` 中没有。

这种情况会对系统安全造成威胁，因为这种组的权限没有得到正确的监管与配置。

修复建议

1. 对出现异常的组，根据实际情况进行修复，如删除或重新正确配置。

扫描检测

确保 `/etc/passwd` 中所有组都存在于 `/etc/group` 中。

1. 执行以下脚本，检查返回结果：

```
##!/bin/bash
for i in $(cut -s -d: -f4 /etc/passwd | sort -u ); do
    grep -q -P "^.*?:[^:]*:$i:" /etc/group
    if [ $? -ne 0 ]; then
    echo "Group $i is referenced by /etc/passwd but does not exist in /etc/group"
    fi
done
```

如没有任何返回值，则视为通过此项检查。

参考

4.39 确保没有重复的 UID

安全等级

- Level 2

描述

使用 `useradd` 命令创建用户时，UID 为自动分配且不会重复。但系统管理员可以手动编辑 `/etc/passwd` 文件并更改 UID 字段。如在工作中出现失误，就有可能导致 UID 的重复。

在正常的系统中，必须为所有用户分配唯一的 UID，以正确监控用户的权限和确保审计责任信息的准确性。

修复建议

1. 对出现异常的 UID，根据实际情况进行修复：删除无用用户或重新分配正确的 UID。

扫描检测

确保没有重复的 UID。

1. 执行以下脚本，检查返回结果：

```
##!/bin/bash
cut -f3 -d":" /etc/passwd | sort -n | uniq -c | while read x ; do
 [ -z "$x" ] && break
 set - $x
 if [ $1 -gt 1 ]; then
 users=$(awk -F: '($3 == n) { print $1 }' n=$2 /etc/passwd | xargs)
 echo "Duplicate UID ($2): $users"
 fi
done
```

如没有任何返回值，则视为通过此项检查。

参考

4.40 确保没有重复的 GID

安全等级

- Level 2

描述

使用 `groupadd` 命令创建用户组时，GID 为自动分配且不会重复。但系统管理员可以手动编辑 `/etc/group` 文件并更改 GID 字段。如在工作中出现失误，就有可能导致 GID 的重复。

在正常的系统中，必须为所有用户组分配唯一的 GID，以正确监控用户组的权限和确保审计责任信息的准确性。

修复建议

1. 对出现异常的 GID，根据实际情况进行修复：删除无用用户组或重新分配正确的 GID。

扫描检测

确保没有重复的 GID。

1. 执行以下脚本，检查返回结果：

```
##!/bin/bash
cut -d: -f3 /etc/group | sort | uniq -d | while read x ; do
 echo "Duplicate GID ($x) in /etc/group"
done
```

如没有任何返回值，则视为通过此项检查。

参考

4.41 确保没有重复的用户名

安全等级

- Level 2

描述

使用 `useradd` 命令创建用户时，不允许创建同名用户。但系统管理员可以手动编辑 `/etc/passwd` 文件并更改用户名字段。如在工作中出现失误，就有可能导致用户名的重复。

在正常的系统中，必须为所有用户分配唯一的用户名，以正确监控用户的权限和确保审计责任信息的准确性。

修复建议

1. 对出现异常的用户名，根据实际情况进行修复：删除无用用户或重新分配正确的用户名。

扫描检测

确保没有重复的用户名。

1. 执行以下脚本，检查返回结果：

```
##!/bin/bash
cut -d: -f1 /etc/passwd | sort | uniq -d | while read x; do
 echo "Duplicate login name ${x} in /etc/passwd"
done
```

如没有任何返回值，则视为通过此项检查。

参考

4.42 确保没有重复的组名

安全等级

- Level 2

描述

使用 `groupadd` 命令创建用户组时，不允许创建同名用户组。但系统管理员可以手动编辑 `/etc/group` 文件并更改组名字段。如在工作中出现失误，就有可能导致组名的重复。

在正常的系统中，必须为所有用户组分配唯一的组名，以正确监控用户组的权限和确保审计责任信息的准确性。

修复建议

1. 对出现异常的组名，根据实际情况进行修复：删除无用用户组或重新分配正确的组名。

扫描检测

确保没有重复的组名。

1. 执行以下脚本，检查返回结果：

```
##!/bin/bash
cut -d: -f1 /etc/group | sort | uniq -d | while read -r x; do
 echo "Duplicate group name ${x} in /etc/group"
done
```

如没有任何返回值，则视为通过此项检查。

参考

4.43 确保所有用户的主目录都存在

安全等级

- Level 1

描述

管理员有可能在 `/etc/passwd` 中定义没有主目录的用户，或定义一个实际上不存在的主目录。

如果用户的主目录不存在或未分配，则该用户将被放置在根目录（`/`）中，且不能写入任何文件或设置本地环境变量。所以为每个用户配置一个正确的主目录是非常重要的。

修复建议

对主目录异常的用户进行修复。

1. 使用以下脚本，对没有配置主目录的用户创建主目录：

```
##!/bin/bash
awk -F: '($1!~/(halt|sync|shutdown|nfsnobody)/ && $7!~/^(\/usr)?\/sbin\/nologin(\/)?$/
 ↪  && $7!~/(\/usr)?\/bin\/false(\/)?$/) { print $1 " " $6 }' /etc/passwd | while read
 ↪  -r user dir; do
    if [ ! -d "$dir" ]; then
        mkdir "$dir"
        chmod g-w,o-wrx "$dir"
        chown "$user" "$dir"
    fi
done
```

扫描检测

确保所有用户的主目录都存在。

1. 执行以下脚本，检查返回结果：

```
##!/bin/bash
awk -F: '($1!~/(halt|sync|shutdown|nfsnobody)/ && $7!~/^(\/usr)?\/sbin\/nologin(\/)?$/
 ↪  && $7!~/(\/usr)?\/bin\/false(\/)?$/) { print $1 " " $6 }' /etc/passwd | while read
 ↪  -r user dir; do
    if [ ! -d "$dir" ]; then
        echo "User: \"$user\" home directory: \"$dir\" does not exist."
    fi
done
```

如没有任何返回值，则视为通过此项检查。

参考

4.44 确保禁用 SCTP

安全等级

- Level 1

描述

流控制传输协议（SCTP）是一个传输层协议，用于支持面向消息的通信，在一个连接中有多个消息流。它的功能类似于 TCP 和 UDP，并融合了二者的特性：它像 UDP 一样是面向消息的，并像 TCP 一样通过拥塞控制来确保可靠的消息顺序传输。

如果不使用该协议，建议禁用该服务以减少潜在的攻击面。

修复建议

禁用 SCTP。

1. 在 `/etc/modprobe.d/` 目录下，创建一个以 `.conf` 结尾的文件，如：`sctp.conf`，并执行以下代码，在文件内添内容：

```
printf "
install sctp /bin/true
" >> /etc/modprobe.d/sctp.conf
```

扫描检测

确保禁用 SCTP。

1. 执行以下命令，检查返回结果：

```
# modprobe -n -v sctp
install /bin/true
# lsmod | grep sctp
<No output>
```

如返回结果符合要求，则视为通过此项检查。

参考

4.45 确保禁用 DCCP

安全等级

- Level 1

描述

数据拥塞控制协议 (Datagram Congestion Control Protocol, DCCP) 是由 (因特网工程工作小组 IETF) 提出一个针对传输层中 UDP 的新传输的协议而发展出来, 用来传输实时业务。它是一个可以进行拥塞控制的非可靠传输协议，并同时提供多种拥塞控制机制，在通信开始时由用户进行协商选择。除预留和自定义方式外，目前 DCCP 定义了两种拥塞控制机制：TCP － Like 和 TFRC。TCP － Like 类似 TCP 的 AIMD 机制，而 TFRC 是 TCP 友好的速率控制机制。建立、维护和拆卸不可靠连接的数据流以及对不可靠性数据流进行拥塞控制，是 DCCP 主要提供的两大功能。实时业务需要快速且低开销的传输协议，要使包头带来的开销和终端处理的工程量尽量小。因此，DCCP 尽可能做到简单合理、低延迟和快速响应，避免提供更高层的传输功能。DCCP 没有 TCP 的可靠性和顺序发送的特性。基于单播的应用功能也被涵盖在 DCCP 中。

如果不使用 DCCP ，建议不安装驱动程序，以减少潜在的攻击面。

修复建议

禁用 DCCP。

1. 在 `/etc/modprobe.d/` 目录下，创建一个以 `.conf` 结尾的文件，如：`dccp.conf` ，并执行以下代码，在文件内添内容：

```
printf "
install dccp /bin/true
" >> /etc/modprobe.d/dccp.conf
```

扫描检测

确保禁用 DCCP。

1. 执行以下命令，检查返回结果：

```
# modprobe -n -v dccp
install /bin/true
# lsmod | grep dccp
<No output>
```

如返回结果符合要求，则视为通过此项检查。

参考

4.46 确保禁用无线网卡接口

安全等级

- Level 1

描述

如对无线网络没有需求，应禁用无线设备，以减少潜在的攻击面。

修复建议

禁用无线网卡接口。

1. 执行以下脚本，禁用所有无线网络接口：

```
##!/usr/bin/env bash
wireless_disable()
{
    if command -v nmcli >/dev/null 2>&1 ; then
        nmcli radio all off
    else
        if [ -n "$(find /sys/class/net/*/ -type d -name wireless)" ]; then
            mname=$(for driverdir in $(find /sys/class/net/*/ -type d -name wireless |
             ↪ xargs -0 dirname); do basename "$(readlink -f "$driverdir"/device/
             ↪ driver/module)";done | sort -u)
            for dm in $mname; do
                echo "install $dm /bin/true" >> /etc/modprobe.d/disable_wireless.conf
            done
        fi
    fi
}
wireless_disable
```

扫描检测

确保禁用无线网卡接口。

1. 执行以下命令，检查返回结果：

```
##!/usr/bin/env bash
wireless_chk()
{
    if command -v nmcli >/dev/null 2>&1 ; then
        if nmcli radio all | grep -Eq '\s*\S+\s+disabled\s+\S+\s+disabled\b';
then
            echo "Wireless is not enabled"
        else
            nmcli radio all
        fi
    elif [ -n "$(find /sys/class/net/*/ -type d -name wireless)" ]; then
        t=0
        mname=$(for driverdir in $(find /sys/class/net/*/ -type d -name
wireless | xargs -0 dirname); do basename "$(readlink -f "$driverdir"/device/driver/
 ↪  module)";done | sort -u)
 for dm in $mname; do
        if grep -Eq "^\s*install\s+$dm\s+/bin/(true|false)" /etc/modprobe.d/*.conf;
          ↪  then
            /bin/true
        else
            echo "$dm is not disabled"
            t=1
        fi
    done
    [ "$t" -eq 0 ] && echo "Wireless is not enabled"
    else
    echo "Wireless is not enabled"
    fi
}
wireless_chk
```

如返回：`Wireless is not enabled`，则视为通过此项检查。

参考

4.47 确保禁用 IP 转发功能

安全等级

- Level 1

描述

`net.ipv4.ip_forward` 和 `net.ipv6.conf.all.forwarding` 转发标志是用来告诉系统是否可以转发此数据包。

将这些标志设置为 `0` 可以确保有多个接口的系统（例如，一个硬代理），不能够转发数据包，不能作为一个路由器使用。这对于信息安全来说是非常重要的。

修复建议

禁用 IP 转发功能。

1. 执行以下代码，禁用 IP 转发功能：

```
# grep -Els "^\s*net\.ipv4\.ip_forward\s*=\s*1" /etc/sysctl.conf /etc/sysctl.d/*.conf
 ↪  /usr/lib/sysctl.d/*.conf /run/sysctl.d/*.conf | while read filename; do sed -ri
 ↪  "s/^\s*(net\.ipv4\.ip_forward\s*)(=)(\s*\S+\b).*$/# *REMOVED* \1/" $filename;
 ↪  done; sysctl -w net.ipv4.ip_forward=0; sysctl -w net.ipv4.route.flush=1

# grep -Els "^\s*net\.ipv6\.conf\.all\.forwarding\s*=\s*1" /etc/sysctl.conf /etc/
 ↪  sysctl.d/*.conf /usr/lib/sysctl.d/*.conf /run/sysctl.d/*.conf | while read
 ↪  filename; do sed -ri "s/^\s*(net\.ipv6\.conf\.all\.forwarding\s*)(=)(\s*\S+\b).*$/
 ↪  # *REMOVED* \1/" $filename; done; sysctl -w net.ipv6.conf.all.forwarding=0; sysctl
 ↪  -w net.ipv6.route.flush=1
```

扫描检测

确保禁用 IP 转发功能。

1. 执行以下命令，检查返回结果：

```
# sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 0
# grep -E -s "^\s*net\.ipv4\.ip_forward\s*=\s*1" /etc/sysctl.conf /etc/sysctl.d/*.conf
 ↪  /usr/lib/sysctl.d/*.conf /run/sysctl.d/*.conf
No value should be returned
# sysctl net.ipv6.conf.all.forwarding
net.ipv6.conf.all.forwarding = 0
# grep -E -s "^\s*net\.ipv6\.conf\.all\.forwarding\s*=\s*1" /etc/sysctl.conf /etc/
 ↪  sysctl.d/*.conf /usr/lib/sysctl.d/*.conf /run/sysctl.d/*.conf
No value should be returned
```

如返回结果均符合预期，则视为通过此项检查。

参考

4.48 确保禁用报文重定向发送

安全等级

- Level 1

描述

ICMP 重定向用来向其他主机发送路由信息。如当前环境下，此主机没有充当路由器的需求，就没有必要发送重定向。

攻击者可以利用被攻击的主机向其他路由器设备发送无效的 ICMP 重定向，破坏路由指向，将正常用户的访问流量导向至一个由攻击者设置的系统，而不是用户期望访问的系统。

修复建议

禁用报文重定向发送功能。

1. 修改 `/etc/sysctl.conf` 文件及 `/etc/sysctl.d` 路径下所有后缀为 `.conf` 文件中以下参数的值。如没有以下参数，则需在 `/etc/sysctl.conf` 文件中添加：

```
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.send_redirects = 0
```

2. 执行以下命令，设置活动内核参数：

```
# sysctl -w net.ipv4.conf.all.send_redirects=0
# sysctl -w net.ipv4.conf.default.send_redirects=0
# sysctl -w net.ipv4.route.flush=1
```

扫描检测

确保禁用报文重定向发送。

1. 执行以下命令，检查返回结果：

```
# sysctl net.ipv4.conf.all.send_redirects
net.ipv4.conf.all.send_redirects = 0
# sysctl net.ipv4.conf.default.send_redirects
net.ipv4.conf.default.send_redirects = 0
# grep "net\.ipv4\.conf\.all\.send_redirects" /etc/sysctl.conf /etc/sysctl.d/*
net.ipv4.conf.all.send_redirects = 0
# grep "net\.ipv4\.conf\.default\.send_redirects" /etc/sysctl.conf /etc/sysctl.d/*
net.ipv4.conf.default.send_redirects = 0
```

如返回结果均符合预期，则视为通过此项检查。

参考

4.49 确保不接受源路由报文

安全等级

- Level 1

描述

源路由的用户可以指定他所发送的数据包沿途经过的部分或者全部路由器。它区别于由主机或者路由器的互联层（IP）软件自行选择路由后得出的路径。

禁止接受源路由报文可防止黑客利用 IP 地址欺骗对系统进行攻击。源路由在过去被广泛使用，以防止单一网络故障引起重大网络波动，但今天的互联网路由协议使得这种技术不再必要。

修复建议

不接受源路由报文。

1. 修改 `/etc/sysctl.conf` 文件及 `/etc/sysctl.d` 路径下所有后缀为 `.conf` 文件中以下参数的值。如没有以下参数，则需在 `/etc/sysctl.conf` 文件中添加：

```
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.default.accept_source_route = 0
net.ipv6.conf.all.accept_source_route = 0
net.ipv6.conf.default.accept_source_route = 0
```

2. 执行以下命令，设置活动内核参数：

```
# sysctl -w net.ipv4.conf.all.accept_source_route=0
# sysctl -w net.ipv4.conf.default.accept_source_route=0
# sysctl -w net.ipv6.conf.all.accept_source_route=0
# sysctl -w net.ipv6.conf.default.accept_source_route=0
# sysctl -w net.ipv4.route.flush=1
# sysctl -w net.ipv6.route.flush=1
```

扫描检测

确保不接受源路由报文。

1. 执行以下命令，检查返回结果：

```
# sysctl net.ipv4.conf.all.accept_source_route
net.ipv4.conf.all.accept_source_route = 0
# sysctl net.ipv4.conf.default.accept_source_route
net.ipv4.conf.default.accept_source_route = 0
# grep "net\.ipv4\.conf\.all\.accept_source_route" /etc/sysctl.conf /etc/sysctl.d/*
net.ipv4.conf.all.accept_source_route = 0
# grep "net\.ipv4\.conf\.default\.accept_source_route" /etc/sysctl.conf /etc/sysctl.d/
 ↪ *
net.ipv4.conf.default.accept_source_route = 0
# sysctl net.ipv6.conf.all.accept_source_route
net.ipv6.conf.all.accept_source_route = 0
# sysctl net.ipv6.conf.default.accept_source_route
net.ipv6.conf.default.accept_source_route = 0
# grep "net\.ipv6\.conf\.all\.accept_source_route" /etc/sysctl.conf /etc/sysctl.d/*
net.ipv4.conf.all.accept_source_route = 0
# grep "net\.ipv6\.conf\.default\.accept_source_route" /etc/sysctl.conf /etc/sysctl.d/
 ↪ *
net.ipv6.conf.default.accept_source_route = 0
```

如返回结果均符合预期，则视为通过此项检查。

参考

4.50 确保不接受 ICMP 重定向

安全等级

- Level 1

描述

ICMP 重定向报文是 ICMP 控制报文中的一种。在特定的情况下，当路由器检测到一台机器使用非优化路由的时候，它会向该主机发送一个 ICMP 重定向报文，请求主机改变路由。路由器也会把初始数据报向它的目的地转发。

攻击者可以使用虚假的 ICMP 重定向消息恶意地改变系统路由表，使被攻击者的报文发送向不正确的路径，并截获其发出的报文。

修复建议

不接受 ICMP 重定向。

1. 修改 `/etc/sysctl.conf` 文件及 `/etc/sysctl.d` 路径下所有后缀为 `.conf` 文件中以下参数的值。如没有以下参数，则需在 `/etc/sysctl.conf` 文件中添加：

```
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv6.conf.all.accept_redirects = 0
net.ipv6.conf.default.accept_redirects = 0
```

2. 执行以下命令，设置活动内核参数：

```
# sysctl -w net.ipv4.conf.all.accept_redirects=0
# sysctl -w net.ipv4.conf.default.accept_redirects=0
# sysctl -w net.ipv6.conf.all.accept_redirects=0
# sysctl -w net.ipv6.conf.default.accept_redirects=0
# sysctl -w net.ipv4.route.flush=1
# sysctl -w net.ipv6.route.flush=1
```

扫描检测

确保不接受源路由报文。

1. 执行以下命令，检查返回结果：

```
# sysctl net.ipv4.conf.all.accept_redirects
net.ipv4.conf.all.accept_redirects = 0
# sysctl net.ipv4.conf.default.accept_redirects
net.ipv4.conf.default.accept_redirects = 0
# grep "net\.ipv4\.conf\.all\.accept_redirects" /etc/sysctl.conf /etc/sysctl.d/*
net.ipv4.conf.all.accept_redirects = 0
# grep "net\.ipv4\.conf\.default\.accept_redirects" /etc/sysctl.conf /etc/sysctl.d/*
net.ipv4.conf.default.accept_redirects = 0
# sysctl net.ipv6.conf.all.accept_redirects
net.ipv6.conf.all.accept_redirects = 0
# sysctl net.ipv6.conf.default.accept_redirects
net.ipv6.conf.default.accept_redirects = 0
# grep "net\.ipv6\.conf\.all\.accept_redirects" /etc/sysctl.conf /etc/sysctl.d/*
net.ipv6.conf.all.accept_redirects = 0
# grep "net\.ipv6\.conf\.default\.accept_redirects" /etc/sysctl.conf /etc/sysctl.d/*
net.ipv6.conf.default.accept_redirects = 0
```

如返回结果均符合预期，则视为通过此项检查。

参考

4.51 确保不接受安全的 ICMP 重定向

安全等级

- Level 1

描述

ICMP 重定向报文是 ICMP 控制报文中的一种。在特定的情况下，当路由器检测到一台机器使用非优化路由的时候，它会向该主机发送一个 ICMP 重定向报文，请求主机改变路由。路由器也会把初始数据报向它的目的地转发。

安全 ICMP 重定向与 ICMP 重定向基本相同，不同之处在于它们来自已知的且可信的网关，这类报文很可能是安全的。但即使是已知的网关也有可能被劫持或破坏。攻击者可以使用虚假的 ICMP 重定向消息恶意地改变系统路由表，使被攻击者的报文发送向不正确的路径，并截获其发出的报文。

综上，配置不接受安全的 ICMP 重定向。进一步加强系统的安全性。

修复建议

不接受 ICMP 重定向。

1. 修改 `/etc/sysctl.conf` 文件及 `/etc/sysctl.d` 路径下所有后缀为 `.conf` 文件中以下参数的值。如没有以下参数，则需在 `/etc/sysctl.conf` 文件中添加:

```
net.ipv4.conf.all.secure_redirects = 0
net.ipv4.conf.default.secure_redirects = 0
```

2. 执行以下命令，设置活动内核参数:

```
# sysctl -w net.ipv4.conf.all.secure_redirects=0
# sysctl -w net.ipv4.conf.default.secure_redirects=0
# sysctl -w net.ipv4.route.flush=1
```

扫描检测

确保不接受源路由报文。

1. 执行以下命令，检查返回结果：

```
# sysctl net.ipv4.conf.all.secure_redirects
net.ipv4.conf.all.secure_redirects = 0
# sysctl net.ipv4.conf.default.secure_redirects
net.ipv4.conf.default.secure_redirects = 0
# grep "net\.ipv4\.conf\.all\.secure_redirects" /etc/sysctl.conf /etc/sysctl.d/*
net.ipv4.conf.all.secure_redirects = 0
# grep "net\.ipv4\.conf\.default\.secure_redirects" /etc/sysctl.conf /etc/sysctl.d/*
net.ipv4.conf.default.secure_redirects = 0
```

如返回结果均符合预期，则视为通过此项检查。

参考

4.52 确保对可疑报文进行日志记录

安全等级

- Level 1

描述

启用该特性后，系统会将源地址不可达的报文记录到内核日志。

启用此功能并记录这些数据包，能够使管理员了解和防范攻击者向系统发送欺骗数据包。

修复建议

开启可疑报文日志记录功能。

1. 修改 `/etc/sysctl.conf` 文件及 `/etc/sysctl.d` 路径下所有后缀为 `.conf` 文件中以下参数的值。如没有以下参数，则需在 `/etc/sysctl.conf` 文件中添加：

```
net.ipv4.conf.all.log_martians = 1
net.ipv4.conf.default.log_martians = 1
```

2. 执行以下命令，设置活动内核参数：

```
# sysctl -w net.ipv4.conf.all.log_martians=1
# sysctl -w net.ipv4.conf.default.log_martians=1
# sysctl -w net.ipv4.route.flush=1
```

扫描检测

确保对可疑报文进行日志记录。

1. 执行以下命令，检查返回结果：

```
# sysctl net.ipv4.conf.all.log_martians
net.ipv4.conf.all.log_martians = 1
# sysctl net.ipv4.conf.default.log_martians
```

```
net.ipv4.conf.default.log_martians = 1
# grep "net\.ipv4\.conf\.all\.log_martians" /etc/sysctl.conf /etc/sysctl.d/*
net.ipv4.conf.all.log_martians = 1
# grep "net\.ipv4\.conf\.default\.log_martians" /etc/sysctl.conf /etc/sysctl.d/*
net.ipv4.conf.default.log_martians = 1
```

如返回结果均符合预期，则视为通过此项检查。

参考

4.53 确保忽略 ICMP 广播请求

安全等级

- Level 1

描述

接受带有广播或多播目的地的 ICMP echo 和时间戳请求可能会欺骗您的主机启动（或参与）Smurf 攻击。Smurf 攻击者使用虚假的源地址发送大量 ICMP 广播消息，所有接收到此消息并作出响应的主机都会发送响应报文到此虚假的地址，此地址可能是不可路由的。如果响应这些数据包的主机达到一定数量，那么网络上的通信量将大大增加，引起通信阻塞。

应配置系统忽略所有向广播和多播地址发送的 ICMP echo 和时间戳请求。

修复建议

忽略 ICMP 广播请求。

1. 执行以下命令，修改配置文件，并设置活动内核参数。

```
# grep -Els "^\s*net\.ipv4\.icmp_echo_ignore_broadcasts\s*=\s*0" /etc/sysctl.conf /
 ↪  etc/sysctl.d/*.conf /usr/lib/sysctl.d/*.conf /run/sysctl.d/*.conf | while read
 ↪  filename; do sed -ri "s/^\s*(net\.ipv4\.icmp_echo_ignore_broadcasts\s*)(=)(\s*\S+
 ↪  \b).*$/# *REMOVED* \1/" $filename; done; sysctl -w
 ↪  net.ipv4.icmp_echo_ignore_broadcasts=1; sysctl -w net.ipv4.route.flush=1
```

扫描检测

确保忽略 ICMP 广播请求。

1. 执行以下命令，检查返回结果：

```
# sysctl net.ipv4.icmp_echo_ignore_broadcasts
net.ipv4.icmp_echo_ignore_broadcasts = 1
```

```
# grep -E -s "^\s*net\.ipv4\.icmp_echo_ignore_broadcasts\s*=\s*0" /etc/sysctl.conf /
 ↪  etc/sysctl.d/*.conf /usr/lib/sysctl.d/*.conf /run/sysctl.d/*.conf
Nothing should be returned
```

如返回结果均符合预期，则视为通过此项检查。

参考

4.54 确保忽略伪造的 ICMP 响应

安全等级

- Level 1

描述

一些攻击者会伪装为路由器发送违反 RFC-1122 的响应，并试图用大量无用的错误消息填充日志文件系统。

禁止内核记录来自广播帧的虚假响应 (RFC-1122 不合规)，从而避免文件系统被无用的日志消息填满，影响系统性能与日志准确性。

修复建议

忽略伪造的 ICMP 响应。

1. 执行以下命令，修改配置文件，并设置活动内核参数：

```
# grep -Els "^\s*net\.ipv4\.icmp_ignore_bogus_error_responses\s*=\s*0" /etc/
 ↪ sysctl.conf /etc/sysctl.d/*.conf /usr/lib/sysctl.d/*.conf /run/sysctl.d/*.conf |
 ↪ while read filename; do sed -ri "s/
 ↪ ^\s*(net\.ipv4\.icmp_ignore_bogus_error_responses\s*)(=)(\s*\S+\b).*$/# *REMOVED*
 ↪ \1/" $filename; done; sysctl -w net.ipv4.icmp_ignore_bogus_error_responses=1;
 ↪ sysctl -w net.ipv4.route.flush=1
```

扫描检测

确保忽略伪造的 ICMP 响应。

1. 执行以下命令，检查返回结果：

```
# sysctl net.ipv4.icmp_ignore_bogus_error_responses
net.ipv4.icmp_ignore_bogus_error_responses = 1
```

```
# grep -E -s "^\s*net\.ipv4\.icmp_ignore_bogus_error_responses\s*=\s*0" /etc/
  ↪ sysctl.conf /etc/sysctl.d/*.conf /usr/lib/sysctl.d/*.conf /run/sysctl.d/*.conf
Nothing should be returned
```

如返回结果均符合预期，则视为通过此项检查。

参考

4.55 确保启用反向路径过滤

安全等级

- Level 1

描述

反向路径过滤：强制 Linux 内核对接收到的数据包进行反向路径过滤来验证数据包的有效性。如果返回的数据包与源数据包来自不同的接口，此数据包将被丢弃 (如果设置了 log_martians，则会记录日志)。此功能能够有效的阻止攻击者向您的系统发送无法回应的虚假数据包。

此功能在不对称路由的环境中是不适用的。如果您的系统上使用了不对称路由（bgp, ospf 等），启用此功能将影响您的路由通信。

修复建议

开启反向路径过滤功能。

1. 执行以下命令，修改配置文件，并设置活动内核参数：

```
# grep -Els "^\s*net\.ipv4\.conf\.all\.rp_filter\s*=\s*0" /etc/sysctl.conf /etc/
 ↪  sysctl.d/*.conf /usr/lib/sysctl.d/*.conf /run/sysctl.d/*.conf | while read
 ↪  filename; do sed -ri "s/^\s*(net\.ipv4\.conf\.all\.rp_filter\s*)(=)(\s*\S+\b).*$/#
 ↪  *REMOVED* \1/" $filename; done; sysctl -w net.ipv4.conf.all.rp_filter=1; sysctl -w
 ↪  net.ipv4.route.flush=1
```

2. 修改 `/etc/sysctl.conf` 文件及 `/etc/sysctl.d` 路径下所有后缀为 `.conf` 文件中以下参数的值。如没有以下参数，则需在 `/etc/sysctl.conf` 文件中添加：

```
net.ipv4.conf.default.rp_filter = 1
```

3. 设置活动内核参数：

```
# sysctl -w net.ipv4.conf.default.rp_filter=1
# sysctl -w net.ipv4.route.flush=1
```

扫描检测

确保启用反向路径过滤。

1. 执行以下命令，检查返回结果：

```
# sysctl net.ipv4.conf.all.rp_filter
net.ipv4.conf.all.rp_filter = 1
# sysctl net.ipv4.conf.default.rp_filter
net.ipv4.conf.default.rp_filter = 1
# grep -E -s "^\s*net\.ipv4\.conf\.all\.rp_filter\s*=\s*0" /etc/sysctl.conf /etc/
 ↪ sysctl.d/*.conf /usr/lib/sysctl.d/*.conf /run/sysctl.d/*.conf
Nothing should be returned
# grep -E -s "^\s*net\.ipv4\.conf\.default\.rp_filter\s*=\s*1" /etc/sysctl.conf /etc/
 ↪ sysctl.d/*.conf /usr/lib/sysctl.d/*.conf /run/sysctl.d/*.conf
net.ipv4.conf.default.rp_filter = 1
```

如返回结果均符合预期，则视为通过此项检查。

参考

4.56 确保已启用 TCP SYN cookie

安全等级

- Level 1

描述

SYN Cookie 是对 TCP 服务器端的三次握手协议作一些修改，专门用来防范 SYN Flood 攻击的一种手段。它的原理是，在 TCP 服务器收到 TCP SYN 包并返回 TCP SYN+ACK 包时，不分配一个专门的数据区，而是根据这个 SYN 包计算出一个 cookie 值。在收到 TCP ACK 包时，TCP 服务器再根据那个 cookie 值检查这个 TCP ACK 包的合法性。如果合法，再分配专门的数据区进行处理未来的 TCP 连接。

SYN Flood 是一种非常危险而常见的 DoS 攻击方式。到目前为止，能够有效防范 SYN Flood 攻击的手段并不多，SYN Cookie 就是其中最著名的一种。

修复建议

启用 TCP SYN cookie 功能。

1. 执行以下命令，修改配置文件，并设置活动内核参数：

```
# grep -Els "^\s*net\.ipv4\.tcp_syncookies\s*=\s*[02]*" /etc/sysctl.conf /etc/
 ↪  sysctl.d/*.conf /usr/lib/sysctl.d/*.conf /run/sysctl.d/*.conf | while read
 ↪  filename; do sed -ri "s/^\s*(net\.ipv4\.tcp_syncookies\s*)(=)(\s*\S+\b).*$/#
 ↪  *REMOVED* \1/" $filename; done; sysctl -w net.ipv4.tcp_syncookies=1; sysctl -w
 ↪  net.ipv4.route.flush=1
```

扫描检测

确保已启用 TCP SYN cookie。

1. 执行以下命令，检查返回结果：

```
# sysctl net.ipv4.tcp_syncookies
net.ipv4.tcp_syncookies = 1
```

```
# grep -E -r "^\s*net\.ipv4\.tcp_syncookies\s*=\s*[02]" /etc/sysctl.conf /etc/
 ↪  sysctl.d/*.conf /usr/lib/sysctl.d/*.conf
Nothing should be returned
```

如返回结果均符合预期，则视为通过此项检查。

参考

4.57 确保不接受 IPv6 路由器通告

安全等级

- Level 1

描述

建议在系统内配置不接受路由器的广播通告，因为这种路由很可能是一种攻击手段，用于劫持或恶意引导主机的流量。在系统内配置一条可信的默认路由可以保护系统不受恶意路由的影响。

应禁用 IPv6 路由通告的接受功能。

修复建议

禁用 IPv6 路由通告的接受功能。

1. 修改 `/etc/sysctl.conf` 文件及 `/etc/sysctl.d` 路径下所有后缀为 `.conf` 文件中以下参数的值。如没有以下参数，则需在 `/etc/sysctl.conf` 文件中添加：

```
net.ipv6.conf.all.accept_ra = 0
net.ipv6.conf.default.accept_ra = 0
```

2. 执行以下命令，设置活动内核参数：

```
# sysctl -w net.ipv6.conf.all.accept_ra=0
# sysctl -w net.ipv6.conf.default.accept_ra=0
# sysctl -w net.ipv6.route.flush=1
```

扫描检测

确保不接受 IPv6 路由器通告。

1. 执行以下命令，检查返回结果：

```
# sysctl net.ipv6.conf.all.accept_ra
net.ipv6.conf.all.accept_ra = 0
# sysctl net.ipv6.conf.default.accept_ra
net.ipv6.conf.default.accept_ra = 0
# grep "net\.ipv6\.conf\.all\.accept_ra" /etc/sysctl.conf /etc/sysctl.d/*
net.ipv6.conf.all.accept_ra = 0
# grep "net\.ipv6\.conf\.default\.accept_ra" /etc/sysctl.conf /etc/sysctl.d/*
net.ipv6.conf.default.accept_ra = 0
```

如返回结果均符合预期，则视为通过此项检查。

参考

4.58 确保已安装防火墙软件包

安全等级

- Level 1

描述

应选择一个防火墙包安装。大多数防火墙配置工具是作为 `nftables` 或 `iptables` 的前端运行。

防火墙包是防火墙管理和配置所必需的。

修复建议

安装防火墙软件包。

1. 根据实际环境，执行以下 3 条命令中任一 1 条，安装一种防火墙软件包：

- firewalld:

```
# dnf install firewalld -y
```

- nftables:

```
# dnf install nftables -y
```

- iptables:

```
# dnf install iptables -y
```

扫描检测

确保已安装防火墙软件包。

1. 执行以下命令，检查以下 3 款软件包，是否正确安装了其中任一 1 款：

- firewalld:

```
# rpm -q firewalld
firewalld-<version>
```

- nftables:

```
# rpm -q nftables
nftables-<version>
```

- iptables:

```
# rpm -q iptables
iptables-<version>
```

返回值中 `<version>` 为软件版本，如以上 3 款软件包，正确安装了其中 1 款，则视为通过此项检查。

参考

4.59 确保防火墙服务已启用且运行状态正常

安全等级

- Level 1

描述

确保启用了防火墙服务以保护系统及网络环境。

- firewald、iptables、nftables 三种防火墙工具，可根据实际生产使用环境，使用其中一种即可。

修复建议

启用防火墙服务。

1. 执行以下命令，启用防火墙服务：

```
# systemctl --now enable firewalld
```

扫描检测

确保防火墙服务已启用且运行状态正常。

1. 执行以下命令，检查防火墙是否开启：

```
# systemctl is-enabled firewalld
enabled
```

2. 执行以下命令，检查防火墙是否正常运行：

```
# firewall-cmd --state
running
```

参考

4.60 确保 iptables 未启用

安全等级

- Level 1

描述

iptables 是集成在 Linux 内核中的包过滤防火墙系统。使用 iptables 可以添加、删除具体的过滤规则，iptables 默认维护着 4 个表和 5 个链，所有的防火墙策略规则都被分别写入这些表与链中。

同时运行 firewald 和 iptables 可能会导致冲突，因此 iptables 应该在使用防火墙时被停止和屏蔽。

- firewald、iptables、nftables 三种防火墙工具，可根据实际生产使用环境，使用其中一种即可。

修复建议

停止 iptables 服务。

1. 执行以下命令，停止 iptables 服务：

```
# systemctl --now mask iptables
```

扫描检测

确保 iptables 未启用。

1. 执行以下命令，检查 iptables 是否已停止：

```
# systemctl status iptables
 Loaded: disabled (/dev/null; bad)
 Active: inactive (dead)
```

2. 执行以下命令，检查 iptables 是否已停止：

```
# systemctl is-enabled iptables
(disabled|masked)
```

如输出结果均符合预期，则视为通过此项检查。

参考

4.61 确保 nftables 未启用

安全等级

- Level 1

描述

nftables 提供了一个新的包过滤框架，该框架基于特定于网络的虚拟机（VM），一个新的用户空间实用程序（nft）和一个用于 {ip，ip6} 表的兼容层。从 Linux 内核 3.13 版本开始 (2013)，nftables 已经成为 Linux 内核主线的一部分。正在逐渐替换 iptables。

相比于 iptables，nftables 有了如下改进：

- 查表取代线性处理
- ipv4 ipv6 使用同一个框架
- 以原子方式应用规则，而不是获取、更新和存储完整的规则集
- 支持在规则集中 debug 和 trace
- 更一致紧凑的语法，没有特定协议的扩展
- 为第三方应用提供 Netlink API

在运行了 firewalld 的系统内，同时运行 nftables 可能会导致冲突，因此 nftables 应该会在使用 firewalld 时被停止和屏蔽。

- firewald、iptables、nftables 三种防火墙工具，可根据实际生产使用环境，使用其中一种即可。

修复建议

停止 nftables 服务。

1. 执行以下命令，停止 nftables 服务：

```
# systemctl --now mask nftables
```

扫描检测

确保 nftables 未启用。

1. 执行以下命令，检查 nftables 是否已停止：

```
# systemctl status nftables
 Loaded: masked (/dev/null; bad)
 Active: inactive (dead)
```

2. 执行以下命令，检查 nftables 是否已停止：

```
# systemctl is-enabled nftables
(disabled|masked)
```

如输出结果均符合预期，则视为通过此项检查。

参考

4.62 确保 nftables 服务已启用

安全等级

- Level 1

描述

nftables 提供了一个新的包过滤框架，该框架基于特定于网络的虚拟机（VM），一个新的用户空间实用程序（nft）和一个用于 {ip，ip6} 表的兼容层。从 Linux 内核 3.13 版本开始 (2013)，nftables 已经成为 Linux 内核主线的一部分。正在逐渐替换 iptables。

相比于 iptables，nftables 有了如下改进：

- 查表取代线性处理
- ipv4 ipv6 使用同一个框架
- 以原子方式应用规则，而不是获取、更新和存储完整的规则集
- 支持在规则集中 debug 和 trace
- 更一致紧凑的语法，没有特定协议的扩展
- 为第三方应用提供 Netlink API

firewald、iptables、nftables 三种防火墙工具，可根据实际生产使用环境，使用其中一种即可。

修复建议

开启 nftables 服务。

1. 执行以下命令，开启 nftables 服务：

```
# systemctl --now enable nftables
```

扫描检测

确保 nftables 服务已启用。

1. 执行以下命令，检查 nftables 是否已启用：

```
# systemctl is-enabled nftables
enabled
```

如输出结果符合预期，则视为通过此项检查。

参考

4.63 确保正确安装 iptables 软件包

安全等级

- Level 1

描述

iptables 是集成在 Linux 内核中的包过滤防火墙系统。使用 iptables 可以添加、删除具体的过滤规则，iptables 默认维护着 4 个表和 5 个链，所有的防火墙策略规则都被分别写入这些表与链中。

需确保系统中安装了 iptables 软件包。

- **firewald、iptables、nftables 三种防火墙工具，可根据实际生产使用环境，使用其中一种即可。**

修复建议

安装 iptables 软件包。

1. 执行以下命令，安装 iptables 软件包：

```
# yum install -y iptables iptables-services
```

扫描检测

确保正确安装 iptables 软件包。

1. 执行以下命令，检查 iptables 软件包是否已安装：

```
# rpm -q iptables iptables-services
iptables-<version>
iptables-services-<version>
```

输出结果中 `<version>` 为软件版本。如输出结果符合预期，则视为通过此项检查。

参考

4.64 确保未安装 nftables

安全等级

- Level 1

描述

nftables 提供了一个新的包过滤框架，该框架基于特定于网络的虚拟机（VM），一个新的用户空间实用程序（nft）和一个用于 {ip，ip6} 表的兼容层。从 Linux 内核 3.13 版本开始 (2013)，nftables 已经成为 Linux 内核主线的一部分。正在逐渐替换 iptables。

相比于 iptables，nftables 有了如下改进：

- 查表取代线性处理
- ipv4 ipv6 使用同一个框架
- 以原子方式应用规则，而不是获取、更新和存储完整的规则集
- 支持在规则集中 debug 和 trace
- 更一致紧凑的语法，没有特定协议的扩展
- 为第三方应用提供 Netlink API

在运行了 iptables 的系统内，同时运行 nftables 可能会导致冲突，因此 nftables 应该会在使用 iptables 时被停止和屏蔽。

firewald、iptables、nftables 三种防火墙工具，可根据实际生产使用环境，使用其中一种即可。

修复建议

卸载 nftables 软件包。

1. 执行以下命令，卸载 nftables 软件包：

```
# yum remove -y nftables
```

扫描检测

确保未安装 nftables。

1. 执行以下命令，检查 nftables 软件包是否未安装：

```
# rpm -q nftables
package nftables is not installed
```

如输出结果符合预期，则视为通过此项检查。

参考

4.65 确保防火墙没有安装或服务已停止

安全等级

- Level 1

描述

同时运行 firewalld 和 iptables 可能会导致冲突，因此 firewalld 应该在使用 iptables 时被停止和屏蔽。

firewald、iptables、nftables 三种防火墙工具，可根据实际生产使用环境，使用其中一种即可。

修复建议

卸载 firewalld 软件包或停止 firewalld 服务。

1. 执行以下命令，卸载 firewalld 软件包：

```
# yum remove -y firewalld
```

OR

2. 执行以下命令，停止 firewalld 服务：

```
# systemctl --now mask firewalld
```

以上 2 条命令，根据实际生产使用环境，选择其中 1 条执行即可。

扫描检测

确保防火墙没有安装或服务已停止。

1. 执行以下命令，检查 firewalld 软件包是否未安装：

```
# rpm -q firewalld
package nftables is not installed
```

OR

2. 执行以下命令，检查 firewalld 服务是否已停止：

```
# systemctl status firewalld | grep "Active: " | grep -v "active (running) "
Active: inactive (dead)
# systemctl is-enabled firewalld
masked
```

如输出结果符合预期，则视为通过此项检查。

参考

4.66 限制历史命令记录数量

安全等级

- Level 1

描述

建议系统限制查看历史命令的数量，建议 50 或 100(参考三级标准)

修复建议

查看 profile 文件中环境变量 HISTSIZE 的值，运行以下命令设置历史命令记录数量为 100 并生效:

```
# grep -qiP "^HISTSIZE" /etc/profile && sed -i "/^HISTSIZE/cHISTSIZE=100" /etc/profile
 ↪  || echo -e "HISTSIZE=100" >> /etc/profile
# source /etc/profile
```

扫描检测

1. 查看环境变量 HISTSIZE 设置的值:

```
# echo $HISTSIZE
100
```

2. 查看 profile 文件 HISTSIZE 设置的值:

```
# grep -iP "^HISTSIZE" /etc/profile
HISTSIZE=100
```

如果检测 1 中输出为 100，且检测 2 中输出为 HISTSIZE=100，说明则通过检查，否则检测未通过。

参考

4.67 限制历史命令存储文件的保存数量

安全等级

- Level 1

描述

建议系统对 bash_history 文件保存命令条数进行限制

修复建议

查看 profile 文件中环境变量 HISTFILESIZE 的值，运行以下命令设置历史命令保存条数为 100 并生效：

```
# grep -qiP "^HISTFILESIZE" /etc/profile && sed -i "/^HISTFILESIZE/cexport
 ↪  HISTFILESIZE=100" /etc/profile || echo -e "export HISTFILESIZE=100" >> /etc/
 ↪  profile
# source /etc/profile
```

扫描检测

1. 查看环境变量 HISTFILESIZE 设置的值:

```
# echo $HISTFILESIZE
100
```

2. 查看 profile 文件 HISTFILESIZE 设置的值:

```
# grep -iP "HISTFILESIZE" /etc/profile
HISTFILESIZE=100
```

如果检测 1 中输出为 100，且检测 2 中输出为 HISTFILESIZE=100，说明则通过检查，否则检测未通过。

参考

4.68 为公共目录/tmp 添加粘贴位

安全等级

- Level 1

描述

建议为公共目录/tmp 设置粘贴位，防止非属主用户进行删除操作

修复建议

查看系统/tmp 目录权限中 other 组中是否存在粘贴位，运行以下命令为/tmp 目录添加粘贴位：

```
# ls -l / | grep tmp | grep rwt || chmod o+t /tmp/
```

扫描检测

查看系统/tmp 目录的 other 组中是否存在 t 权限位:

```
# ls -l / | grep tmp | grep rwt;echo $?
drwxrwxrwt.  10 root root 4096 Nov  6 16:22 tmp
0
```

如果检测最后输出为 0，说明则通过检查，否则检测未通过。

参考

4.69 严格要求 SSH 公私钥文件权限配置正确

安全等级

- Level 3

描述

避免在操作系统免密环境中随意被导入攻击机密钥，从而免密登陆被攻击机器。将公钥文件权限设置为 400，确保非属主 (root) 用户对各类公钥文件进行任何操作。

修复建议

查看系统 `/etc/ssh/*key` ， `/etc/ssh/*key.pub` 文件权限，确保权限位均为 400，运行以下命令为上述文件设置正确权限位：

```
# chmod 400 /etc/ssh/*key
# chmod 400 /etc/ssh/*key.pub
```

扫描检测

查看 `/etc/ssh/*key` ， `/etc/ssh/*key.pub` 文件权限位是否符合要求:

```
# stat -c "%a-%U-%G" {/etc/ssh/*key,/etc/ssh/*key.pub}
400-root-root
```

如果检测最后输出每行以 “-” 作为分隔符分隔的第一个字段显示的权限位均为 400，则说明通过检查，否则检测未通过。

参考

4.70 确保没有启用 XDMCP

安全等级

- Level 1

描述

X 显示监控协议 (X Display Manager Control Protocol, XDMCP)：管理与操控 X server 的显示内容，并提供登录验证。

但 XDMCP 服务是不安全的：- XDMCP 不是加密协议，这可能导致用户的输入内容被攻击者捕获；- XDMCP 容易受到中间人攻击：攻击者伪装为 XDMCP 服务器，截获合法用户的登录请求，从而导致凭据泄露等问题。

修复建议

编辑文件 `/etc/gdm/custom.conf` 并删掉以下行：

```
Enable=true
```

扫描检测

运行以下命令并验证输出结果:

```
# grep -Eis '^\s*Enable\s*=\s*true' /etc/gdm/custom.conf
Nothing should be returned
```

如果没有任何输出，则说明通过检查，否则检测未通过。

参考

4.71 确保/var 分区上设置 nosuid 选项

安全等级

- Level 3

描述

`nosuid` 加载选项的让指定文件系统不包含 `setuid` 文件。由于 `/var` 路径下一般用于存放日志之类的可变文件，因此在此文件系统下增加 `nosuid` 配置，以确保用户不能在 `/var` 下创建可执行文件。

修复建议

目标：编辑 `/etc/fstab` 文件，在 `/var` 分区的第四个字段 (加载选项) 中添加 `nosuid` 。

1. 使用 root 权限打开 `/etc/fstab` 文件

```
sudo vim /etc/fstab
```

2. 修改 `/var` 分区配置，在第四个字段（defaults...）中增加 `nosuid` 参数。具体参考以下配置：

```
<device> /var <fstype> defaults,nosuid 0 0
```

注意，要用 `,` 隔开该选项的其他参数。

3. 保存并关闭文件。
4. 运行以下命令以使用配置的选项重新挂载 /var：

```
# mount -o remount /var
```

扫描检测

验证是否为 `/var` 分区配置了 `nosuid` 参数。

1. 使用以下命令来验证 `/var` 分区是否已经配置了 `nosuid` 参数：

```
# mount | grep '/var.*nosuid'
/dev/nvme0n2p1 on /var type ext4 (rw,nosuid,relatime)
```

如果输出中包含了 `/var` 并且有 `nosuid` 字段，则表示已经为 `/var` 挂载设置了 `nosuid` 选项。

2. 还可以使用以下命令来检查 `/etc/fstab` 文件是否包含了 `nosuid` 选项：

```
# grep '/var' /etc/fstab | grep -o nosuid
nosuid
```

如果输出中有 `nosuid` 字段，则表示已经为 /var 挂载设置了 `nosuid` 选项。

参考

5 mandatory-access-control

5.1 确保 SElinux 工具已安装

安全等级

- Level 1

描述

SELinux 提供强制访问控制。强制访问控制系统，增强系统安全性。

修复建议

目标：确保 SElinux 工具已安装。

1. 使用以下命令安装 SElinux 工具：

```
# dnf install libselinux selinux-policy-mls selinux-policy-targeted
```

扫描检测

1. 执行以下命令，检查 SElinux 工具是否安装：

```
# rpm -q libselinux selinux-policy-mls selinux-policy-targeted
libselinux-<version>
selinux-policy-mls-<version>
selinux-policy-targeted-<version>
```

为软件版本信息。如输出结果符合预期，则视为通过此项检查。

参考

5.2 确保 SELinux 调用 mls 策略

安全等级

- Level 3

描述

配置 SELinux 以满足或超过默认目标策略，该策略限制守护进程和系统软件。SELinux 默认定义了两个策略：- targeted：这是 SELinux 的默认策略，这个策略主要是限制网络服务的，对本机系统的限制极少。- mls：多级安全保护策略，这个策略限制得更为严格。

修复建议

目标：确保 SELinux 调用 mls 策略，最大限度地减小系统中服务进程可访问的资源（最小权限原则）。

1. 修改/etc/selinux/config 文件，设置 SELINUXTYPE 参数

```
SELINUXTYPE=mls
```

如果您需要限制较少的策略，可在 /etc/selinux/config 文件中设置它们 **此配置需重启才可生效**

扫描检测

1. 使用以下命令查看 SELinux 使用的默认策略：

```
# grep -E '^\s*SELINUXTYPE=(targeted|mls)\b' /etc/selinux/config
SELINUXTYPE=mls
# sestatus | grep Loaded
Loaded policy name:             mls
```

如输出结果符合预期，则视为通过此项检查

参考

5.3 确保 SELinux 不是禁用模式

安全等级

- Level 3

描述

强烈建议不要在禁用模式下运行 SELinux；避免系统强制执行 SELinux 策略，还避免标记任何持久性对象，例如文件，从而使未来难以启用 SELinux。SELinux 提供了三种模式，分别是：Enforcing：强制模式。违反 SELinux 规则的行为将被阻止并记录到日志中。Permissive：宽容模式。违反 SELinux 规则的行为只会记录到日志中。一般为调试用。Disabled：关闭 SELinux。每种模式都为 Linux 系统安全提供了不同的好处。

修复建议

目标：确保系统不在禁用模式下运行 SELinux。

1. 以下任意一个命令设置 SELinux 的运行模式设置成 Enforcing 模式:

```
# setenforce 1
```

或者设置成 Permissive 模式:

```
# setenforce 0
```

2. 也可以修改 /etc/selinux/config 文件进行 SELinux 模式的设置设置成 Enforcing 模式：

```
SELINUX=enforcing
```

或者设置成 Permissive 模式:

```
SELINUX=permissive
```

此配置需重启才可生效

扫描检测

1. 执行以下命令，检查当前 SELinux 的模式：

```
# getenforce
Enforcing
-OR-
Permissive
```

2. 执行以下命令，检查 SELinux 配置文件的参数是否符合要求：

```
# grep -Ei '^\s*SELINUX=(enforcing|permissive)' /etc/selinux/config
SELINUX=enforcing
-OR-
SELINUX=permissive
```

如以上两条检测命令输出结果均为 enforcing 或 permissive，则视为通过此项检查。

参考

SELinux States and Modes:https:// access.redhat.com/ documentation/ en-us/ red_hat_enterprise_linux/ 7/ html/ selinux_users_and_administrators_guide/sect-security-enhanced_linux-introduction-selinux_modes

5.4 确保 SELinux 是 Enforcing 模式

安全等级

- Level 3

描述

避免系统强制执行 SELinux 策略，还避免标记任何持久性对象，例如文件，从而使未来难以启用 SELinux。SELinux 提供了三种模式，分别是：Enforcing：强制模式。违反 SELinux 规则的行为将被阻止并记录到日志中。Permissive：宽容模式。违反 SELinux 规则的行为只会记录到日志中。一般为调试用。Disabled：关闭 SELinux。

修复建议

目标：确保 SELinux 是 Enforcing 模式。

1. 运行以下命令设置成 Enforcing 模式:

```
# setenforce 1
```

2. 也可以修改 /etc/selinux/config 文件进行设置成 Enforcing 模式：

```
SELINUX=enforcing
```

扫描检测

1. 命令查看当前 SELinux 的模式：

```
# getenforce
Enforcing
```

2. 运行以下命令查看当前 SELinux 的模式：

```
# grep -Ei '^\s*SELINUX=(enforcing|permissive)' /etc/selinux/config
SELINUX=enforcing
```

如输出结果为 enforcing 那么符合预期，则视为通过此项检查

参考

SELinux States and Modes:https:// access.redhat.com/ documentation/ en-us/ red_hat_enterprise_linux/ 7/ html/ selinux_users_and_administrators_guide/sect-security-enhanced_linux-introduction-selinux_modes

5.5 确保没有未限制的服务存在

安全等级

- Level 4

描述

避免不受限制的进程在不受限制的域中运行，应对于不受限制的进程，使用 SELinux 分配相对应的安全上下文以及构建策略规则。

修复建议

查看审计日志的信息发现的任何不受限制的信息。并依据实际情况分配给它们现有安全上下文或为它们构建的策略。

扫描检测

1. 运行以下命令并验证未产生输出：

```
# ps -eZ | grep unconfined_service_t
```

如未产生输出，那么符合预期，则视为通过此项检查

参考

5.6 使用 SELinux 实现三权分离-用户创建

安全等级

- Level 4

描述

- 当前，Linux 操作系统已广泛应用于各种设备和产品中，如服务器、PC 机、机顶盒及路由器等。随着 Linux 系统的不断发展和广泛应用，Linux 系统的安全问题也引起越来越多的关注。
- 在 Linux 操作系统中，存在一个超级用户即 root 用户。root 也称为系统管理员，它拥有管理系统的一切权限。当一个非法用户获得 root 用户口令后，他就可以以超级用户的身份登录系统，然后做任何他想做的事情：如任意添加、删除用户，终止进程，删除重要文件甚至更改 root 用户的口令。因此，一旦 root 权限被恶意用户利用，就可能导致系统数据遭到泄密和破坏。
- 该问题也引起了国家的重点关注，如国家保密标准 BMB20-2007《涉及国家秘密的信息系统分级保护管理规范》中明确提出：涉密信息系统应配备系统管理员、安全保密管理员和安全审计员这三类安全保密管理人员，三员应该相互独立、相互制约、不得兼任。三个管理员之间的工作机制分为协作和制约两种机制，行使的是原超级用户的权力，即系统管理员、安全管理员和审计管理员间相互协作，共同维护系统的正常运行。制约机制指只有在当前管理员操作不影响其他管理员正在进行的操作时才被允许，从而保证了管理员行为的可预期性，避免超级用户的误操作或其身份被假冒而带来的安全隐患，增强了系统的安全性。该规范可以有效防止由系统管理员权力过大所带来的系统安全威胁和隐患。

- SELinux (security-enhanced Linux) 是安全增强的 Linux，以强制访问控制 (mandatory access control, MAC) 技术为基础，应用类型增强 (type enforcement, TE) 和基于角色访问控制 (role-base access control, RBAC) 两种安全策略模型。通过 MAC 技术可以实现对用户和进程权限的最小化，即使在系统受到攻击或者进程和用户的权限被剥夺的情况下，也不会对整个系统的安全造成重大影响。SELinux 对访问的控制更彻底，它对系统中的所有文件、目录、端口资源的访问控制都基于一定的安全策略而设定。只有管理员才能定制安全策略，一般用户没有权限更改。因此 SELinux 为三权分离思想的实现奠定了基础。

前置条件

SElinux 服务正确安装，且服务正常开启。具体可参考以下文档：

- 5.1-ensure-selinux-is-installed.md -> 安装 SElinux 工具
- 5.3-ensure-the-selinux-mode-is-enabled.md -> 启用 SElinux 服务

修复建议

- 创建系统、安全、审计管理员用户

1. 创建 SElinux 系统管理员用户

```
# useradd -m -s /bin/bash sysadm
# passwd sysadm
# echo 'export PATH="/sbin:/bin:/usr/sbin:/usr/bin:/usr/X11R6/bin:/sbin:/usr/sbin"' >>
 ↪  /home/sysadm/.bashrc
# usermod sysadm -aG wheel
# semanage user -m -R 'sysadm_r system_r' sysadm_u
# semanage login -a -s sysadm_u sysadm
# restorecon -FR -v /home/sysadm
```

2. 创建 SElinux 安全管理员用户

```
# useradd -m -s /bin/bash secadm
# passwd secadm
# echo 'export PATH="/sbin:/bin:/usr/sbin:/usr/bin:/usr/X11R6/bin:/sbin:/usr/sbin"' >>
 ↪  /home/secadm/.bashrc
# usermod secadm -aG wheel
# semanage user -a -R "staff_r secadm_r" -P staff secadm_u
# semanage login -a -s secadm_u secadm
# restorecon -FR -v /home/secadm
```

3. 创建 SElinux 审计管理员用户

```
# useradd -m -s /bin/bash audadm
# passwd audadm
# echo 'export PATH="/sbin:/bin:/usr/sbin:/usr/bin:/usr/X11R6/bin:/sbin:/usr/sbin"' >>
 ↪  /home/audadm/.bashrc
```

```
# usermod audadm -aG wheel
# semanage user -a -R "staff_r auditadm_r" -P staff auditadm_u
# semanage login -a -s auditadm_u audadm
# restorecon -FR -v /home/audadm
```

- 应用 `mls` 策略

1. 安装 `selinux-policy-mls` 策略软件包:

```
# yum install -y selinux-policy-mls
```

2. 修改 `/etc/selinux/config` 文件中 `SELINUXTYPE` 变量的值，将安全策略配置为 `mls` :

```
SELINUXTYPE=mls
```

- 重启系统

1. 重启系统，使配置生效:

```
# reboot
```

扫描检测

1. 查看 selinux 状态

```
# getenforce
Permissive
OR
Enforcing
```

2. 查看 selinux 策略

```
# sestatus | grep Loaded
Loaded policy name:             mls
```

3. 查看创建三个用户状态

```
# ls -Z /home
auditadm_u:object_r:user_home_dir_t:s0 audadm
secadm_u:object_r:user_home_dir_t:s0 secadm
sysadm_u:object_r:user_home_dir_t:s0 sysadm
```

参考

5.7 使用 SELinux 实现三权分离-系统管理员登录权限配置

安全等级

- Level 4

描述

- SELinux (security-enhanced Linux) 是安全增强的 Linux，以强制访问控制 (mandatory access control, MAC) 技术为基础，应用类型增强 (type enforcement, TE) 和基于角色访问控制 (role-base access control, RBAC) 两种安全策略模型。通过 MAC 技术可以实现对用户和进程权限的最小化，即使在系统受到攻击或者进程和用户的权限被剥夺的情况下，也不会对整个系统的安全造成重大影响。SELinux 对访问的控制更彻底，它对系统中的所有文件、目录、端口资源的访问控制都基于一定的安全策略而设定。只有管理员才能定制安全策略，一般用户没有权限更改。因此 SELinux 为三权分离思想的实现奠定了基础。
- 当使用 SElinux 实现三权分离后，初步分离的系统管理员、安全管理员、审计管理员的所具备的权力策略需要进一步完善，增加系统管理员从图形界面和 ssh 方式登录系统的相关策略，以便系统管理员可以正常登录系统。

前置条件

SElinux 服务正确安装，且服务正常开启。具体可参考以下文档：

- 5.1-ensure-selinux-is-installed.md -> 安装 SElinux 工具
- 5.3-ensure-the-selinux-mode-is-enabled.md -> 启用 SElinux 服务
- 5.6-use-selinux-for-separation-of-powers-user-created.md -> 使用 SELinux 实现三权分离-用户创建

审计功能正确安装，且服务正常开启。具体可参考以下文档：

- 2.19-ensure-audit-is-installed.md -> 安装审计功能
- 2.20-ensure-audit-service-is-enabled.md -> 启用审计功能

修复建议

- 目的：添加 SELinux 系统管理员登录系统的策略模块，以便系统管理员可成功登录系统

1. 添加 SELinux 系统管理员 ssh 登录的策略模块

- 添加生成并插入策略的脚本内容，并增加其可执行权限

```
# vim run_te.sh
##!/bin/bash
  cat $1   | audit2allow  -m $1 > $1.te
  checkmodule -M -m -o $1.mod $1.te
  semodule_package -o $1.pp -m $1.mod
  semodule -i $1.pp
# chmod a+x run_te.sh
```

- root 用户打开审计日志，以便查看系统管理员登录失败的信息

```
# tail -f /var/log/audit/audit.log
```

在执行系统管理员登录的操作后，预期输出登录失败的信息。

- 系统管理员用户使用 ssh 方式登录系统

```
# ssh sysadm@ip
```

预期结果：登录失败

- 将审计日志中输出的登录失败的信息复制到 sysadm_log 中，并使用 run_te.sh 添加策略规则

```
# vim sysadm_log
# sh run_te.sh sysadm_log
```

预期输出 sysadm_log.mod、sysadm_log.te、sysadm_log.pp 等文件。并且系统管理员可通过 ssh 方式登录系统。

2. 添加 SELinux 系统管理员从图形界面登录的策略模块

- 添加生成并插入策略的脚本内容，并增加其可执行权限

```
# vim run_te.sh
##!/bin/bash
  cat $1   | audit2allow  -m $1 > $1.te
  checkmodule -M -m -o $1.mod $1.te
  semodule_package -o $1.pp -m $1.mod
  semodule -i $1.pp
# chmod a+x run_te.sh
```

- 使用 ssh 登录 root 用户并打开审计日志，以便查看系统管理员登录失败的信息

```
# ssh root@ip
# tail -f /var/log/audit/audit.log
```

预期结果：在执行系统管理员登录的操作后，可得到系统管理员登录失败的信息。

- 系统管理员用户使用图形界面方式登录系统

预期结果：登录失败。

- 将审计日志中输出的登录失败的信息复制到 sysadm_log 中，并使用 run_te.sh 添加策略规则

```
# vim sysadm_log1
# sh run_te.sh sysadm_log1
```

预期结果：当前目录存在 sysadm_log1.mod、sysadm_log1.te、sysadm_log1.pp 等文件；并且系统管理员可通过图形界面登录系统。

扫描检测

1. 系统管理员通过 ssh 方式登录系统

```
# ssh sysadm@ip
```

2. 系统管理员通过图形界面方式登录系统

预期结果：系统管理员通过以上 2 种方式都可以成功登录系统，即符合预期结果。

参考

5.8 创建普通、审计、安全用户

安全等级

- Level 2

描述

- 当前，Linux 操作系统已广泛应用于各种设备和产品中，如服务器、PC 机、机顶盒及路由器等。随着 Linux 系统的不断发展和广泛应用，Linux 系统的安全问题也引起越来越多的关注。
- 在 Linux 操作系统中，存在一个超级用户即 root 用户。root 也称为系统管理员，它拥有管理系统的一切权限。当一个非法用户获得 root 用户口令后，他就可以以超级用户的身份登录系统，然后做任何他想做的事情：如任意添加、删除用户，终止进程，删除重要文件甚至更改 root 用户的口令。因此，一旦 root 权限被恶意用户利用，就可能导致系统数据遭到泄密和破坏。
- 该问题也引起了国家的重点关注，如国家保密标准 BMB20-2007《涉及国家秘密的信息系统分级保护管理规范》中明确提出：涉密信息系统应配备系统管理员、安全保密管理员和安全审计员这三类安全保密管理人员，三员应该相互独立、相互制约、不得兼任。三个管理员之间的工作机制分为协作和制约两种机制，行使的是原超级用户的权力，即系统管理员、安全管理员和审计管理员间相互协作，共同维护系统的正常运行。制约机制指只有在当前管理员操作不影响其他管理员正在进行的操作时才被允许，从而保证了管理员行为的可预期性，避免超级用户的误操作或其身份被假冒而带来的安全隐患，增强了系统的安全性。该规范可以有效防止由系统管理员权力过大所带来的系统安全威胁和隐患。

修复建议

使用 `useradd` `passwd` 创建系统、安全、审计管理员用户并配置密码。

```
## 创建自定义用户
# useradd [username]
# passwd
```

扫描检测

使用以下命令，查看用户是否创建成功。

```
cat /etc/passwd |cut -f 1 -d :
```

参考

5.9 确保 SETroubleshoot 被卸载

安全等级

- Level 1

描述

SETroubleshoot 服务通过用户友好的界面通知桌面用户 SELinux 拒绝服务。该服务提供关于配置错误的重要信息以及未经授权的入侵和其他潜在错误。

修复建议

目标：确保 SETroubleshoot 被卸载

1. 运行以下命令卸载 setroubleshoot。

```
# dnf remove setroubleshoot -y
```

扫描检测

验证 setroubleshoot 是否安装

1. 运行以下命令以检测是否安装 setroubleshoot。

```
# rpm -q setroubleshoot
package setroubleshoot is not installed
```

输出结果为 `package setroubleshoot is not installed` 则表示未安装 setroubleshoot。

参考