三级等保说明

> 本文档参考自： 信息安全技术--网络安全等级保护基本要求。
>
> 每一个项的符合程度以及重要程度本项目都提供了默认值，用户也可以根据自己的需求自己配置。
>
> 符合程度：符合， 部分符合， 不符合
>
> 重要程度：1， 2， 3
>
> 计算规则： M = 100 - (100 × Σ(重要性等级 × 不合规程度) / 项目总数)

访问控制

1. 应在网络边界或区域之间根据访问控制策略设置访问控制规则，默认情况下除允许通信外受控接口拒绝所有通信

建议：

- 启用主机防火墙（如 UFW）：sudo ufw enable
- 设置默认拒绝所有传入连接：sudo ufw default deny incoming
- 显式添加必要服务访问规则（如 SSH、Web）：sudo ufw allow 22/tcp
- 对于多区域系统，建议使用物理或虚拟防火墙、跳板机或网关进行分区隔离与规则控制。

---

2. 应删除多余或无效的访问控制规则，优化访问控制列表，并保证访问控制规则数量最小化

建议：

- 检查是否存在重复、未命中的规则。
- 保留最小必要通信规则，其余使用 deny 处理。
- 可执行：sudo ufw status numbered 查看所有规则编号并精简。
- 对于已废弃的 IP/端口，使用 sudo ufw delete <编号> 删除。

3. 应对源地址、目的地址、源端口、目的端口和协议等进行检查，以允许/拒绝数据包进出。

建议：

- 使用 iptables/nftables 设置基于五元组的规则：
  示例：iptables -A INPUT -p tcp -s 192.168.1.0/24 --dport 22 -j ACCEPT
- 禁止使用仅开放“任意 IP + 任意端口”的 allow all 规则
- 若使用 ufw，建议指定来源 IP 和协议：
  sudo ufw allow from 192.168.1.100 to any port 22 proto tcp

---

4. 应能根据会话状态信息为进出数据流提供明确的允许/拒绝访问能力。

建议配置如下 iptables 规则以满足状态访问控制要求：

- 允许已建立和相关连接：
  iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
- 丢弃无效连接：
  iptables -A INPUT -m conntrack --ctstate INVALID -j DROP
- 放行新建连接需添加明确端口规则，例如 SSH：
  iptables -A INPUT -p tcp --dport 22 -m conntrack --ctstate NEW -j ACCEPT

---

5. 应对进出网络的数据流实现基于应用协议和应用内容的访问控制

建议：

- 部署 NGINX + ModSecurity，实现基于 HTTP 方法、URL、头信息的访问控制
- 使用 Suricata 或 Snort 等 DPI 引擎检测应用协议/内容，并拦截非法请求
- 若为 HTTP 代理架构，也可使用 Squid + ACL 控制 FTP、HTTP、域名访问行为
- 对关键服务应用如 Web 应用系统启用 WAF（如 openresty、云防火墙）。

安全区域边界--入侵防范

1. 应在关键网络节点处检测、防止或限制从外部发起的网络攻击行为

建议：

- 安装并启用 fail2ban：自动阻止 SSH/Web 登录爆破行为
  sudo apt install fail2ban
- 部署 Suricata 或 Snort 实现入侵检测/阻断（IDS/IPS）
- 设置 iptables 规则阻止常见扫描器端口连接（如 nmap 默认端口）
- 配置 SSH 登录频率限制：ufw limit ssh
- 对 Web 服务建议启用 WAF 或接入云防火墙。

---

2. 应在关键网络节点处检测、防止或限制从内部发起的网络攻击行为

建议：

- 安装 Suricata 或 Snort，识别主机对外连接行为（如恶意 DNS、扫描、隧道等）
- 部署 Wazuh 代理，对主机行为做集中监控和告警
- 设置 iptables 限制主机对外访问敏感端口（如 445, 3389, SMTP 等）
- 对 curl、wget、nc 等工具的使用行为进行审计
- 若主机为出口节点，建议对外流量做 NetFlow 分析和速率限制

---

3. 应采取技术措施对网络行为进行分析实现对网络攻击特别是新型网络攻击行为的分析

建议：

- 部署 Zeek（原 Bro）进行流量行为分析，如可疑 DNS、HTTP、SSH 会话
- 部署 Suricata，配合 JSON 日志分析通信模式和告警
- 将日志汇总到 ELK + Wazuh 中台，进行图形化行为可视化和联动分析
- 若具备能力，可构建基于机器学习的异常检测模型识别新型攻击行为。

---

4. 当检测到攻击行为时，记录攻击源IP、攻击类型、攻击目标、攻击时间，在发生严重入侵事件时应提供报警

建议：

- 部署 Suricata 或 Snort，开启事件日志输出（fast.log 或 eve.json）
- 启用严重攻击邮件/Webhook 报警机制
- 使用 fail2ban 或 Wazuh 配置实时告警策略
- 定期检查事件日志完整性和准确性

- 建议将日志汇总至 SIEM 平台进行集中管理。

恶意代码和垃圾邮件防范

1. 应在关键网络节点处对恶意代码进行检测和清除，并维护恶意代码防护机制的升级和更新

建议：

- 安装 ClamAV，并配置每日自动更新：
  `sudo apt install clamav clamav-freshclam`
  `sudo systemctl enable --now clamav-freshclam`
- 安装 chkrootkit 检查 Rootkit 隐藏行为：`sudo apt install chkrootkit`
- 可选安装 Wazuh Agent + YARA 实现集中恶意文件分析与拦截
- 定期扫描关键目录，如 `/etc /var /tmp /usr/local`
- 在边界网关或跳板机上部署病毒防护尤为关键

---

2. 应在关键网络节点处对垃圾邮件进行检测和防护，并维护垃圾邮件防护机制的升级和更新

建议：

- 安装 SpamAssassin 或 Rspamd，作为邮件服务的垃圾邮件过滤组件：
  sudo apt install spamassassin
  或 sudo apt install rspamd
- 启用规则库自动更新（cron.daily 或 systemd timer）
- 启用 DNSBL（如 zen.spamhaus.org）拦截来源可疑邮件
- 配置 Postfix + Amavis 结合使用，支持病毒+垃圾邮件拦截
- 定期检测规则效果，优化白名单与误判处理机制

安全区域边界--安全审计

1. 应在网络边界、重要网络节点进行安全审计，审计覆盖到每个用户，对重要的用户行为和重要安全事件进行审计。

建议：

- 安装并启用 auditd 以记录用户行为审计：sudo apt install auditd

- 启用 rsyslog/syslog-ng 以记录用户登录、sudo 操作等事件
- 推荐部署 Wazuh Agent 进行集中审计与告警分析
- 配置审计规则，确保记录包括：登录、提权、配置变更、敏感命令执行（如 rm、chmod、scp）

---

2. 审计记录应包括事件的日期和时间、用户、事件类型、事件是否成功及其他与审计相关的信息

建议：

- 安装并启用 auditd：`sudo apt install auditd`
- 确保 /var/log/audit/audit.log 文件存在并记录关键事件
- 添加如下审计规则示例：
  - 登录/认证事件：`-w /var/log/auth.log -p wa`
  - 重要命令审计：`-a always,exit -F arch=b64 -S execve -k cmd_exec`
- 定期校验日志内容完整性（是否包含 success=、uid=、audit(time)...）

---

3. 应对审计记录进行保护，定期备份，避免受到未预期的删除 、修改或覆盖

建议：

- 在 /etc/logrotate.d/ 下创建或恢复 rsyslog 的轮转配置
- 确保 /etc/cron.daily/logrotate 存在并可执行
- 手动执行 `sudo logrotate -f /etc/logrotate.d/rsyslog` 并查看是否产生轮转文件

---

4. 应能对远程访问的用户行为、访问互联网的用户行为等单独进行行为审计和数据分析

建议：

- 启动 auditd 服务以启用系统行为审计
- 确保 /var/log/secure 文件存在并由 rsyslog 正确记录登录行为
- 安装并配置 wazuh-agent、osquery 或 suricata 等行为分析与网络访问审计工具

身份鉴别

1. 应对登录的用户进行身份标识和鉴别，身份标识具有唯一性，身份鉴别信息具有复杂度要求并定期更换

建议：

- 确保 /etc/passwd 中每个用户 UID 唯一，避免权限绕过
- 使用 `chage -M 90 <user>` 设置密码最大使用期限
- 配置 /etc/pam.d/common-password，启用 pam_pwquality 或 pam_cracklib 增强密码复杂度，如：
  `password requisite pam_pwquality.so retry=3 minlen=12 ucredit=-1 lcredit=-1 dcredit=-1 ocredit=-1`

---

2. 应具有登录失败处理功能，应配置并启用结束会话、限制非法登录次数和当登录连接超时自动退出等相关措施

建议：

- 在 /etc/pam.d/common-auth 中启用 pam_faillock，如：
  auth required pam_faillock.so preauth silent deny=5 unlock_time=600
- 在 /etc/profile 设置自动退出时间，如：export TMOUT=600
- 在 /etc/bash.bash_logout 或 profile.d 中添加登出清理命令，例如：
  trap 'rm -f ~/.bash_history' EXIT

---

3. 当进行远程管理时，应采取必要措施防止鉴别信息在网络传输过程中被窃听

建议：

- 启用并配置 SSH 服务，使用强加密算法
- 禁用 telnet、rsh、rlogin 等明文服务（可执行 systemctl disable 命令）
- 在 /etc/ssh/sshd_config 中添加安全加密配置

---

4. 应采用口令、密码技术、生物技术等两种或两种以上组合的鉴别技术对用户进行身份鉴别，且其中一种鉴别技术至少应使用密码技术来实现

建议：

- 确保启用 PAM 密码认证（如 pam_unix.so）
- 配置第二种身份鉴别技术，推荐选项：
  - Google Authenticator: 安装并启用 pam_google_authenticator.so
  - 硬件钥匙（如 YubiKey）: 配置 pam_u2f.so
  - 指纹识别：配置 pam_fprintd.so
  - TOTP/OATH: 使用 pam_oath.so

访问控制

1. 应对登录的用户分配账户和权限，确保每个用户拥有唯一身份，并依据职责授予适当权限

建议：

- 确保 UID 为 0 的账户仅限 root
- 删除或禁用默认账户（guest、test、user 等）
- 通过 usermod、visudo 或 group 添加方式，为授权用户合理分配 sudo 权限。

---

2. 应重命名或删除默认账户，修改默认账户的默认口令

建议：

- 删除不必要的默认账户（如 userdel <账户名>）
- 或使用如下方式禁用默认账户登录：
  - `usermod -L <账户>`（锁定账号）
  - `passwd -l <账户>`（锁定密码）
- 或将其改名为非默认名称：`usermod -l newname oldname`

---

3. 应及时删除或停用多余的、过期的账户，避免共享账户的存在

建议：

- 对于超过 90 天未登录的账户，若不再使用，执行 `userdel -r <user>` 删除或 `usermod -L <user>` 锁定
- 对于过期账户，可使用 `chage -E <date> <user>` 设置新的到期日期或锁定：`passwd -l <user>`

- 确保系统中仅保留一个 UID=0 的 root 账户，其它不必要的高权限账户应删除或降级。

---

4. 应授予管理用户所需的最小权限，实现管理用户的权限分离

建议：

- 使用 `visudo` 在 `/etc/sudoers` 或 `/etc/sudoers.d/` 下定义 `Cmnd_Alias`，将管理操作按功能拆分
- 为不同管理角色的用户或用户组分别分配对应 Cmnd_Alias，而非 ALL
- 确认 sudoers 中无“ALL=(ALL:ALL) ALL”或类似无限制条目，完成后测试并重启 sudo 服务。

---

5. 应由授权主体配置访问控制策略，访问控制策略规定主体对客体的访问规则

建议：

- 启用并配置 SELinux（Enforcing 模式）或 AppArmor 强制 Profile
- 在 /etc/pam.d/ 中添加 `account required pam_access.so`，并在 /etc/security/access.conf 中编写主体-客体访问规则
- 定期审计 access.conf 和 MAC 策略，确保策略由授权管理员维护。

---

6. 访问控制的粒度应达到主体为用户级或进程级，客体为文件，数据库表级

建议手动执行以下检查以确认访问控制粒度是否达标：
【文件系统部分】

- 查看关键文件（如 /etc/shadow）是否配置了用户级 ACL：
  getfacl /etc/shadow
- 检查是否启用 SELinux 或 AppArmor：
  getenforce 或 aa-status
- 检查某服务进程（如 nginx/mysql）是否运行在非 root 账号下，并限制了访问范围：
  ps -ef | grep nginx
  【数据库部分】
- 登录数据库（如 MySQL、PostgreSQL）查看用户是否具有表级权限：

- MySQL:
  SELECT user, host, table_schema, table_name, privilege_type FROM information_schema.table_privileges
- PostgreSQL:
  SELECT grantee, table_schema, table_name, privilege_type FROM information_schema.role_table_grants

- 确认是否为用户分配了细粒度的 GRANT 权限，而非全库或全实例授权。

---

7. 应对重要主体和客体设置安全标记，并控制主体对有安全标记信息资源的访问

建议：

- 安装并启用 AppArmor：sudo apt install apparmor apparmor-utils
- 检查是否存在 enforce 模式的 profile：aa-status
- 确保关键服务（如 nginx、mysql）运行在 AppArmor 限制下（查看 ps -eZ 支持有限）
- 查看已应用的策略目录：/etc/apparmor.d/
  可使用 aa-enforce /etc/apparmor.d/ 启用策略
- 可使用 aa-logprof 分析访问日志并生成自定义规则

安全计算环境--安全审计

1. 应启用安全审计功能，审计覆盖到每个用户，对重要的用户行为和重要安全事件进行审计。

建议：

- 安装并启用 auditd：sudo apt install auditd
- 开启并设为开机启动：sudo systemctl enable --now auditd
- 使用如下规则示例配置 /etc/audit/rules.d/audit.rules：
  -w /etc/sudoers -p wa -k sudo_watch
  -a always,exit -F arch=b64 -S execve -F euid>=1000 -k user_exec
- 应使用 auditctl 验证规则是否生效：auditctl -l
- 所有日志应保存在 /var/log/audit/audit.log，可使用 aureport、ausearch 分析行为。

---

2. 审计记录应包括事件的日期和时间、用户、事件类型，事件是否成功及其他审计相关的信息。

建议：

- 安装并启用 auditd：sudo apt install auditd && systemctl enable --now auditd
- 使用 auditctl 添加规则以捕捉关键操作，例如登录、命令执行：
  auditctl -a always,exit -F arch=b64 -S execve
- 审核 /var/log/audit/audit.log 日志内容是否包含用户、时间、结果等字段
- 使用 ausearch 或 aureport 工具查看解析后的审计记录：
  ausearch -m USER_AUTH,EXECVE -ts today

---

3. 应对审计记录进行保护，定期备份，避免受到未预期的删除、修改或覆盖等

建议：

- 确保 `/var/log/audit/audit.log` 权限为 600 或 640，属主属组为 root：
  `chmod 600 /var/log/audit/audit.log && chown root:root /var/log/audit/audit.log`
- 启用 logrotate：确保存在 `/etc/logrotate.d/audit`
- 修改 /etc/audit/auditd.conf：
  max_log_file_action = keep_logs
- 可选加强：定期对日志做 hash 签名或使用不可修改挂载（如 chattr +a）。

---

4. 应对审计进程进行保护，防止未经授权的中断

建议：

- 启动 auditd 并确保其开机自启：sudo systemctl enable --now auditd
- 设置 auditctl 为严格模式：sudo auditctl -e 2
- 确保内核开启审计功能：
  echo 1 > /proc/sys/kernel/audit_enabled
  或在 /etc/sysctl.conf 中添加：kernel.audit_enabled = 1
- 可使用 chattr +i 锁定配置文件，防止被篡改。
- 如需更强防护，可考虑开启 grub 审计启动参数：audit=1

安全计算环境--入侵防范

1. 应遵循最小安装原则，仅安装需要的组件和应用程序

建议：

- 卸载无业务需求的组件（如 telnet、ftp、cups、xinetd 等）：
  sudo apt remove <组件名> --purge
- 对于服务器环境，应避免安装 ubuntu-desktop/gnome/xorg 等图形界面包；
- 定期使用以下命令审查服务与包：
  - systemctl list-units --type=service
  - dpkg -l
  - netstat -tulnp 或 ss -tuln

---

2. 应关闭不需要的系统服务、默认共享和高危端口

建议：

- 禁用不需要的服务，例如：
  sudo systemctl disable --now <服务名>
- 检查是否监听高危端口：ss -tuln
- 如确需使用 Samba、NFS 等共享服务，应进行访问控制（hosts allow/deny、防火墙限制）；
- 可结合 ufw 或 iptables/firewalld 对外开放端口进行精细控制。

---

3. 应能够检测到对重要节点进行入侵的行为，并在发生严重入侵事件时提供报警

建议：

- 安装并启用 fail2ban：sudo apt install fail2ban && systemctl enable --now fail2ban
- 编辑 /etc/fail2ban/jail.local，启用 sshd 规则并配置报警：
  action = %(action_mwl)s （日志 + 邮件告警）
- 安装并启动 auditd：sudo apt install auditd
- 建议为重要节点配置 Wazuh、OSSEC 等主机入侵检测系统，以实现集中告警管理。

恶意代码防范

1. 应采用免受恶意代码攻击的技术措施或主动免疫可信验证机制，及时识别入侵和病毒行为，并将其有效阻断

建议：

- 安装并启动杀毒工具（ClamAV）：
  `sudo apt install clamav clamav-daemon && systemctl enable --now clamav-daemon`
- 安装并初始化完整性检测工具（AIDE）：
  `sudo apt install aide && aideinit`
  可定期执行 aide --check 进行变更检测
- 启用 AppArmor 并强制执行策略：
  `sudo aa-enforce /etc/apparmor.d/*`
  查看状态：sudo aa-status
- 对关键应用（如 nginx、mysql）配置自定义 AppArmor 策略以增强免疫能力。

数据完整性

1. 应采用校验技术或密码技术保证重要数据在传输过程中的完整性，包括鉴别数据、业务数据、审计数据、配置数据、视频数据、个人信息等

建议：

- 安装并启用 OpenSSL：sudo apt install openssl
- 启用 SSH 安全登录：sudo apt install openssh-server
- 启用 HTTPS，使用有效证书：如配置 nginx/apache 加密网站服务
- 审计日志应通过 rsyslog 配置 TLS 加密转发到日志服务器，参考配置：
  `action(type=\omfwd\ Target=\logserver\ Port=\6514\ Protocol=\tcp\`
  `StreamDriver=\gtls\ StreamDriverMode=\1\ StreamDriverAuthMode=\anon\)`

---

2. 应采用校验技术或密码技术保证重要数据在存储过程中的完整性，包括鉴别数据、业务数据、审计数据、配置数据、视频数据和个人信息等。

建议：

- 安装完整性检测工具 AIDE：sudo apt install aide
- 初始化数据库：sudo aideinit，并定期执行 aide --check
- 对重要文件（如审计日志、配置文件）进行签名：
  gpg --sign /var/log/audit/audit.log
- 或对关键目录生成 hash 清单：`find /etc -type f -exec sha256sum {} + > etc_hash.txt`
- 可使用 chattr +i /etc/passwd 等命令防止重要文件被修改
- 对个人数据建议使用 openssl 或 gpg 加密后存储

数据保密性

1. 应采用密码技术保证重要数据在传输过程中的保密性，包括但不限于鉴别数据、重要业务数据和重要个人信息等

建议：

- 启用 SSH 登录替代 telnet：sudo apt install openssh-server && systemctl enable --now ssh
- 启用 HTTPS 网站/接口传输（配置 Nginx/Apache TLS 证书）
- 安装 OpenSSL：sudo apt install openssl
- 配置 rsyslog TLS 加密转发审计日志
- 确保数据库连接使用 SSL 模式，如：mysql --ssl-ca=/path/ca.pem

---

2. 应采用密码技术保证重要数据在存储过程中的保密性，包括但不限于鉴别数据、重要业务数据和重要个人信息等。

建议：

- 对系统分区启用 LUKS 加密（推荐在安装系统时开启）；
- 对用户目录或业务目录启用 `ecryptfs` 或 `fscrypt`：
  sudo apt install ecryptfs-utils
- 使用 openssl/gpg 对重要文件或字段加密：
  openssl enc -aes-256-cbc -in data.txt -out data.enc
  gpg -c important-data.conf
- 不应以明文形式存储用户密码，应使用 bcrypt/sha256+salt 哈希保存；

- 对数据库表字段可实现透明加密（如 PostgreSQL pgcrypto 模块）。

数据备份恢复

1. 应提供重要数据的本地数据备份和恢复功能

建议：

- 对重要数据目录（如 /etc、/var/lib/mysql、/home）执行 tar 或 rsync 备份：
  `sudo rsync -a /etc /var/backups/etc_bak`
  `sudo tar -czf /var/backups/mysql_$(date +%F).tar.gz /var/lib/mysql`
- 数据库请定期使用 `mysqldump`、`pg_dump` 等进行结构 + 数据备份：
  mysqldump -uroot -p dbname > /var/backups/dbname.sql
- 记录恢复流程或使用脚本一键还原：tar -xzf 或 rsync -a --restore

---

3. 应提供重要数据处理系统的热冗余，保证系统的高可用性

建议：

- 对服务进行热备（建议使用 Keepalived + Nginx 实现主备漂移）
- 对数据库部署主从架构或主主复制（如 MySQL Replication）
- 使用 Pacemaker + Corosync 实现多节点集群高可用
- 对块设备采用 DRBD 或分布式文件系统如 GlusterFS
- 如使用容器服务建议部署 K8s + HAProxy/Nginx 实现高可用控制器节点

剩余信息保护

1. 应保证鉴别信息所在的存储空间在被释放或重新分配前得到完全清除。

建议：

- 安装数据销毁工具：sudo apt install coreutils secure-delete
- 使用 shred/srm 删除私钥/密码等敏感文件，如：shred -u ~/.ssh/id_rsa
- 设置加密 swap：
  - 修改 /etc/crypttab 添加 swap 加密配置
  - 或安装时启用加密 swap（默认 LUKS）

- 禁止简单 rm 删除重要认证数据；日志清理应使用 logrotate + secure-delete 工具

---

2. 应保证存有敏感数据的存储空间被释放或重新分配前得到完全清除。

建议：

- 安装数据擦除工具：sudo apt install secure-delete coreutils
- 删除敏感文件请使用：shred -u 或 srm
- 设置 swap 加密（/etc/crypttab 中配置加密 swap）
- 将 /tmp 目录挂载为 tmpfs：
  echo 'tmpfs /tmp tmpfs defaults,noatime,mode=1777 0 0' >> /etc/fstab
  mount -o remount /tmp
- 虚拟机快照和数据库导出文件也应执行 shred 后删除。

个人信息保护

1. 应仅采集和保存业务必需的用户个人信息。

请检查以下内容：

- 系统是否仅采集业务所需的用户信息字段，例如：登录验证仅需手机号/用户名，无需身份证/人脸。
- 是否存在过度收集，如不需要的人脸、指纹、家庭住址、设备唯一标识等；
- 后端数据库表结构中是否包含敏感字段但实际不使用；
- 隐私政策与实际采集字段是否一致，用户是否知情；
- 建议定期审计数据库字段和日志采集配置，避免采集冗余个人信息。
  如果是 Web 系统，可排查：form 表单字段、API 请求参数、数据库用户表字段等；
  如果是日志系统，建议避免记录完整 token、手机号等字段。

---

2. 应禁止未授权访问和非法使用用户个人信息。

建议：

- 启用系统审计：sudo apt install auditd && systemctl enable --now auditd
- 设置敏感文件权限（如密码文件）：chmod 600 /etc/shadow
- 对数据库敏感表配置基于角色的访问控制（RBAC）

- 接口返回数据时应进行脱敏处理，如显示部分手机号/证件号
- 所有敏感数据导出行为需记录日志或设置审批流程