

服务器等 IT 设备托管安全责任书

甲方（机房管理方）： 网络信息中心

乙方（服务器所有方/管理员）： [托管服务器所属的学院/教师名称]

为确保托管于甲方机房内服务器等 IT 设备（以下统称服务器）的安全、稳定运行,明确双方的安全管理责任,根据国家相关法律法规及甲方机房管理制度,经甲乙双方协商一致,特订立本安全责任书。

一、 甲方的权利与责任

- 物理环境保障：**负责提供符合标准的机房物理环境,包括但不限于稳定的电力供应（市电、UPS）、稳定高效的空调系统、消防系统、防雷接地、视频监控、门禁系统等。
- 网络连通性保障：**负责提供连通校园网络的网络接入端口,并保障机房网络设备的稳定运行。
- 物理安全监管：**负责机房整体的物理安全,对进出机房的人员进行审核、登记与陪同。未经乙方授权,甲方人员不得操作乙方的服务器设备。
- 应急响应：**在发生机房级故障（如断电、网络中断、火灾等）时,立即启动应急预案,并第一时间通知乙方负责人或联系人。

5. **配合义务：**在乙方管理员进行现场维护时，提供必要的出入便利与配合。如因甲方原因需对服务器进行下电、重启等操作，须提前通知乙方并获同意后方可进行（紧急情况除外，但事后需说明）。

二、乙方的权利与责任

乙方作为服务器的所有者和管理者，对服务器内部的所有软件、数据、配置及由此产生的所有网络行为负全部安全责任。具体包括但不限于：

1. 系统安全责任：

- a. 使用甲方指定的 IP 地址，请勿占用其他 IP 地址。
- b. 负责服务器操作系统的安装、更新、补丁升级，确保系统漏洞得到及时修复。
- c. 负责服务器上所有应用软件（如 Web 服务、数据库、中间件等）的安全配置、更新与漏洞修复。
- d. 设置高复杂强度的用户密码策略（如 root/管理员密码），定期更换，并严格管理密钥、令牌等敏感信息。
- e. 合理配置防火墙（如 `nftables/iptables/firewalld/ufw`），遵循最小权限原则，仅开放必要的服务端口。
- f. 安装并维护防病毒、入侵检测/防护系统（如配置 `fail2ban`、`sshguard`、`selinux/AppArmor`）等安全软件。
- g. 设置日志的远程记录，确保能记录 180 天系统运行日志。

2. 账户与权限管理：

- a. 严格管理服务器账户，禁止使用弱密码，及时删除无用账户。
- b. 对远程登录（SSH 等）进行严格限制，建议启用 2FA 二因子验证，并可限制源 IP 地址访问。

3. 数据安全与备份责任：

- a. 负责服务器内全部数据的完整性与保密性。
- b. 制定并严格执行数据备份策略，定期将重要数据备份至安全位置（如本地、其他云存储）。甲方机房不承担因乙方未备份导致的数据丢失责任。
- c. 对敏感数据进行加密存储。

4. 内容与行为合规责任：

- a. 保证服务器上运行的服务及存储的内容完全遵守《网络安全法》、《数据安全法》、《个人信息保护法》等法律法规，不得从事任何黑客攻击、网络诈骗、传播恶意软件、发起 DDoS、hosting 非法信息等违法行为。
- b. 如服务器被黑客入侵并成为攻击源或违法信息传播源，乙方须承担全部责任，并立即配合甲方及监管部门进行处置。

5. 维护与联系责任：

- a. 提供 7x24 小时有效的紧急联系人和联系方式（至少两人），并在变更时立即通知甲方。

- b. 负责服务器的日常远程监控与维护。如需进入机房现场操作，须提前向甲方申请并遵守机房管理规定。
- c. 在接到甲方的安全告警或监管部门通知后，应立即响应并处置。

6. 设备标识与规范：

- a. 在服务器醒目位置粘贴资产标签，标明单位、联系人。
- b. 服务器主机名、网络配置等应规范设置，便于识别和管理。

三、 安全事件处理与责任划分

1. **事件定义：**安全事件包括但不限于：服务器被入侵、数据泄露、成为网络攻击跳板、感染病毒木马、端口扫描攻击、拒绝服务攻击等。
2. **乙方第一责任：**一旦发生安全事件，乙方为第一责任方，必须立即采取措施隔离、排查、修复，并按规定向甲方报告，由甲方处理并决定是否联系公安机关网络安全部门处置，请勿直接拨打 110 报警。
3. **甲方有权采取措施：**若乙方服务器对机房网络或其他托管设备造成严重影响（如大规模 DDoS 攻击、病毒扩散），为避免危害扩大，甲方有权在紧急情况下立即断开该服务器的网络连接或下电，事后再通知乙方。乙方应予以理解与配合。
4. **责任追究：**因乙方服务器安全问题导致甲方或其他第三方遭受损失的，由乙方承担全部赔偿责任及法律责任。若导致甲方机房 IP 段或声誉受损，甲方有权追究乙方责任并终止托管服务。

四、 免责条款

因以下情形导致的问题，甲方不承担责任：

- 因电力保障、电信运营商等不可控因素造成的服务中断。
- 因战争、地震、火灾等不可抗力导致的损害。
- 完全因乙方自身系统漏洞、配置错误、管理疏忽、未及时修复补丁等原因导致的安全事件及数据丢失。

五、 附则

1. 本责任书作为《服务器托管合同》的有效附件，与主合同具有同等法律效力。
2. 本责任书一式两份，甲乙双方各执一份，自双方签字盖章之日起生效。

甲方（盖章）：

乙方（盖章）：

代表签字：

代表签字：

日期：

日期：

Linux 服务器网络安全设置基线

(V20251210)

信息安全特别提醒：如出现服务器发现有异常登录、被黑客入侵、被勒索加密等事件，**请第一时间联系学校网络信息中心安全团队：**63601897、13505693311，而不是联系 110 报警（派出所不擅长处理此类安全事件）。

本文档主要针对普通的等保一级系统 Linux 服务器网络安全设置。

如果为等级保护二级或以上系统，请按照等保要求进行系统设置。

1. 使用广泛支持的系统

请使用目前仍被社区或开发商支持的主流操作系统。截至 2025 年 12 月，主流支持的操作系统包括：Ubuntu 24.04 LTS, Debian 13, Rocky Linux 9, Rocky Linux 10, Alma Linux 9, Alma Linux 10, openEuler 24.03 LTS 等。

请避免使用已经不再维护的系统，因为这些系统将不再得到安全更新。如果您的系统已经不再支持范围之内，建议尽快升级到最新的操作系统中。常见的升级命令包括：

```
# Ubuntu 系列
do-release-upgrade
# Debian 系列
apt dist-upgrade
```

升级前请注意备份数据。

2. 禁止系统中存在弱密码

一般使用的弱密码包括：纯数字、生日、ustc 等元素。密码设置建议设置至少包含大小写字母、数字和符号等元素。

特别提醒：厂商安装的系统往往存在简单密码或有规律的密码，请务必修改后再连接网线。

3. 仅仅开放必要的端口

使用外部防火墙或自带的 iptables/nftables 设置好过滤规则，仅仅对外开放需要使用的端口。

如果想测试服务器对外开放了哪些端口，可以在服务器上执行以下命令(需要大约 6 秒钟返回结果)

```
curl https://ip.ustc.edu.cn/portscan/
```

如在 202.38.64.40 上执行返回以下内容，说明 TCP 端口 80、443 是开放的：

```
正在探测 202.38.64.40 开放的端口：
=====
Discovered open port 443/tcp on 202.38.64.40
Discovered open port 80/tcp on 202.38.64.40
=====
以上是开放的端口，如果为空说明未开放端口
```

请使用 Linux 防火墙对常见端口进行管理和限制。常见命令包括：

```
# Ubuntu 服务器
ufw enable
ufw allow 80
ufw deny 80

# CentOS、Redhat、RockyLinux 等系统
firewall-cmd --add-port=80/tcp --permanent
firewall-cmd --remove-port=80/tcp --permanent

# 其他 Linux 服务器
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -p tcp --dport 80 -j DROP
```

4. ssh 登录安全防护

对 tcp 22 端口做严格限制，不允许对校园网普遍开放 22 端口，建议采用如下防护手段的组合来提高安全性：

- 限制登录 IP
- 使用 knock 敲门机制
- 使用 2FA 认证
- 使用公钥认证
- 使用 fail2ban 等安全增强程序

如果仅仅允许特定 IP 登录，可以使用如下命令对 22 端口访问进行限制：

```
# Ubuntu 服务器，允许 202.38.64.1 访问 22 端口。并禁用其他 ip 访问
ufw allow from 202.38.64.1 to any port 22
ufw deny 22
ufw reload

# CentOS、Redhat、RockyLinux 等系统
firewall-cmd --permanent --remove-service=ssh
firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source address="202.38.64.1"
port port="22" protocol="tcp" accept'
firewall-cmd --reload

# 其他 Linux 服务器
iptables -A INPUT -s 202.38.64.1 -p tcp --dport 22 -j ACCEPT
iptables -A INPUT -p tcp --dport 22 -j DROP
```

如果需要登录的 IP 变化较多，建议使用 knock 敲门机制做保护，请见附件 2。

启用 2FA 的方式请参考相关文档。

此外，建议 ssh 服务使用公钥，而非使用密码进行认证。建议将 /etc/ssh/sshd_config 配置修改禁用 SSH 密码登陆：

```
# /etc/ssh/sshd_config
PasswordAuthentication no
```

如果有任何原因需要启用密码登录，至少禁用 root 用户的密码登录 (PermitRootLogin prohibit-password)。

另外,建议使用 fail2ban 等安全增强程序,对 ssh 端口的错误登陆进行自动封禁。在默认情况下,会对 10 分钟内失败登录 5 次的 IP 地址,封禁 10 分钟。请使用如下命令安装 failban (默认情况会自动配置 ssh 服务加固)：

```
# Ubuntu/Debian 系列
apt install -y fail2ban
systemctl enable fail2ban
fail2ban-client status sshd

# Redhat、CentOS、RockyLinux 等
dnf install epel-release # (可能需要)
dnf install -y fail2ban
systemctl enable fail2ban
```

5. 设置日志的远程记录

将日志转发到日志服务器实现远程记录,以免系统被入侵后黑客删除日志无法溯源。请勿在系统日志中记录敏感信息。

网络信息中心提供日志服务器 202.38.64.45 可以接收日志并保留 180 天。请在 /etc/rsyslog.conf 最后增加以下配置,需要核查日志时与网络信息中心联系:

```
*.*@202.38.64.45
```

执行以下命令生效:

```
systemctl restart rsyslog
```

1. 数据库、NFS 等服务访问控制

MySQL、PostgreSQL、Redis 等数据库服务，应当只监听本地地址或私有网段 (localhost、127.0.0.1 或 Unix socket) 。请配置外部防火墙阻断对应端口的外部互联网连接 (例如 MySQL: 3306, PostgreSQL: 5432, Redis: 6379) 。

针对 NFS 服务器，需要特别注意利用 IP 地址进行访问控制。请检查/etc/exports 文件中的配置信息。例如：

```
/var/nfs_share 202.38.64.0/24 (rw,async)
```

将限制/var/nfs_share 目录只能通过 202.38.64.0/24 段进行访问。

2. 数据安全与备份

根据服务器内数据重要情况，规划并实现数据安全保障措施。

制定并严格执行数据备份策略，定期将重要数据备份至安全位置。

对敏感数据进行加密存储。

设置完毕后，请参附件 1: Linux 服务器安全检查清单 (V2025210) 逐项检查。

附件 1: Linux 服务器安全检查清单 (V20251210)

检查人: _____ 检查时间: _____年____月____日

■ 操作系统是广泛支持的系统, 系统版本_____

■ 系统中不存在弱密码、厂商默认密码

■ curl https://ip.ustc.edu.cn/portscan/ 检查开放了如下 TCP 端口, 均为必需

■ 端口: _____ 应用: _____

■ SSH 安全防护措施

■ 限制如下 IP 登录

■ IP 地址: _____

■ IP 地址: _____

■ IP 地址: _____

■ 已设置普通用户 ssh 登录的 knock 敲门服务

■ 已开启 fail2ban

■ 其他防护措施_____

■ 不涉及普通用户 ssh 登录

■ 日志已经设置转发，转发到如下 IP:

■ 远程日志服务器 IP: _____

■ 数据库/redis 等服务端口检查

■ 已限制，限制措施_____

■ 无这些应用

■ NFS 服务检查

■ 已限制，限制措施_____

■ 无 NFS 服务

■ 数据备份检查

■ 备份措施_____

■ 数据不重要，无需备份

附件 2: knock 敲门程序设置说明

工作原理:

Knock (端口敲击) 是一种通过发送特定序列的网络数据包来触发服务器端操作的 **隐式认证机制**。它不是直接访问服务端口，而是通过"敲门"的方式通知服务器开放特定端口。

客户端	服务器
1. 发送 SYN 包到端口 1000	
----->	
	监听端口 1000
2. 发送 SYN 包到端口 2000	
----->	
	监听端口 2000
3. 发送 SYN 包到端口 300	
----->	
	监听端口 3000
	✓ 检测到正确序列
	↓
	执行命令（如打开 SSH 端口）
4. 现在可以连接 SSH 端口 22	
<----->	

使用如下的配置，依次连接服务器的 1000、2000、3000 端口（只要发送 TCP SYN 包即可），系统将执行 start_command，允许 IP 连接服务器的 TCP 22 端口，这个过程就是“敲门”。然后等待 10 秒钟后，会自动执行 stop_command，再禁止 IP 连接。

[options]
logfile = /var/log/knockd.log
[opencloseSSH]
sequence = 1000,2000,3000
seq_timeout = 5
tcpflags = syn
start_command = /usr/bin/iptables -A TCP -s %IP% -p tcp --dport 22 -j ACCEPT
cmd_timeout = 10
stop_command = /usr/bin/iptables -D TCP -s %IP% -p tcp --dport 22 -j ACCEPT

具体设置：

Knockd 的具体使用方式，特别是 start_command、stop_command 与系统使用的包过滤机制有关，不同系统差异较大哦，可以在 deepseek 中问“如何设置 knockd”，或参考如下链接：

<https://manpages.debian.org/stretch/knockd/knockd.1.en.html>

https://wiki.archlinux.org/title/Port_knocking