实验六：系统监控与安全管理

一、实验目的

1、掌握系统性能监控；

2、掌握系统可视化监控。

3、了解 Linux 的安全机制；

4、掌握使用防火墙提升主机安全性；

二、实验学时

2 学时

扫码看操作演示

三、实验类型

设计性

四、实验需求

1、硬件

每人配备计算机 1 台。

2、软件

Windows 操作系统，安装 Oracle VM VirtualBox 软件，安装 MobaXterm 软件。

3、网络

本地主机与虚拟机能够访问互联网，不使用 DHCP 服务。

4、工具

无

五、实验任务

1、完成系统性能监控；

2、完成使用 Linux-dash 实现可视化监控；

4、完成防火墙配置。

六、实验环境

1、本实验需要 VM 1 台；

2、本实验 VM 配置信息如下表所示；

虚拟机配置	操作系统配置
虚拟机名称：VM-Lab-06-Task-01-172.20.1.17 内存：4GB	主机名：Lab-06-Task-01 IP 地址：172.20.1.17 子网掩码：255.255.255.0

CPU：2 颗，1 核心 虚拟磁盘：20GB 网卡：1 块，桥接	网关：172.20.1.1 DNS：8.8.8.8

3、本实验拓扑图。

无

4、本实验操作演示视频。

本实验为视频集的第 6 集：https://www.bilibili.com/video/BV1h14y1k7Gc?p=6

七、实验内容及步骤

1、使用命令工具监控系统性能

1.1 查看 CPU 信息

（1）使用 lscpu 命令工具查看 CPU 信息。

（2）使用 cat /proc/cpuinfo 命令工具查看 CPU 信息。

（3）使用 mpstat 命令工具查看 CPU 信息。

```
# 使用 lscpu 显示 CPU 详细信息
[root@Lab-06-Task-01 ~]# lscpu
# 以扩展可读的格式显示 CPU 信息
[root@Lab-06-Task-01 ~]# lscpu -e
# 显示 CPU 指定列的信息
[root@Lab-06-Task-01 ~]# lscpu -e=CPU
# 以可解析的格式显示 CPU 信息
[root@Lab-06-Task-01 ~]# lscpu -p
# 显示 online CPU 信息
[root@Lab-06-Task-01 ~]# lscpu -bp

# 使用 cat /proc/cpuinfo 显示 CPU 详细信息
[root@Lab-06-Task-01 ~]# cat /proc/cpuinfo
# 查看 CPU 型号
[root@Lab-06-Task-01 ~]# cat /proc/cpuinfo | grep name | cut -f2 -d: | uniq -c
# 查看物理 CPU 个数
[root@Lab-06-Task-01 ~]# cat /proc/cpuinfo | grep 'physical id' | sort | uniq | wc
-l
# 查看 CPU 的总线程数量
[root@Lab-06-Task-01 ~]# cat /proc/cpuinfo| grep "processor"| wc -l

# 使用 mpstat 命令工具查看 CPU 信息
# 查看多核 CPU 核心的当前运行状况信息，每两秒更新一次
[root@Lab-06-Task-01 ~]# dnf install -y sysstat
[root@Lab-06-Task-01 ~]# mpstat -P ALL 2
# 查看多核 CPU 核心的当前运行状况信息，每五秒更新一次，采样两次
[root@Lab-06-Task-01 ~]# mpstat -P ALL 5 2
```

1.2 查看磁盘信息

（1）使用 df 命令工具查看磁盘信息。

（2）使用 fdisk 命令工具查看磁盘信息。

```
# 使用 df 命令工具查看磁盘信息
[root@Lab-06-Task-01 ~]# df
# 使用 df -i 以 inode 模式来显示磁盘使用情况
[root@Lab-06-Task-01 ~]# df -i
# 使用 df -T 显示文件系统类型
[root@Lab-06-Task-01 ~]# df -T
# 使用 df -h 与更易读的方式显示目前磁盘空间和使用情况
[root@Lab-06-Task-01 ~]# df -h
# 使用 df -k 以单位显示磁盘的使用情况
[root@Lab-06-Task-01 ~]# df -k
# 使用 df -l 显示本地的分区的磁盘空间使用率
[root@Lab-06-Task-01 ~]# df -l
# 使用 df -a 显示各文件系统的使用情况
[root@Lab-06-Task-01 ~]# df -a
# 使用 df -ia 显示各文件系统的 i 节点的使用情况
[root@Lab-06-Task-01 ~]# df -ia

# 使用 fdisk -l 显示磁盘当前分区信息
[root@Lab-06-Task-01 ~]# fdisk -l
# 使用 fdisk -lu 显示 SCSI 磁盘的每个分区的情况
[root@Lab-06-Task-01 ~]# fdisk -lu
```

1.3 查看系统实时状态

（1）使用 top 命令工具查看系统实时状态。

```
# 使用 top 命令显示系统进行信息
[root@Lab-06-Task-01 ~]# top
# 使用 top -d 设置信息更新时间
[root@Lab-06-Task-01 ~]# top -d 3
# 使用 top -p 显示指定进程的信息
[root@Lab-06-Task-01 ~]# top -p 192
# 使用 top -n 显示更新 3 次后推出
[root@Lab-06-Task-01 ~]# top -n 3
# 使用 top -S 累计显示进程 CPU 使用时间
[root@Lab-06-Task-01 ~]# top -S
# 使用 top -H 显示进程中线程的详细信息
[root@Lab-06-Task-01 ~]# top -H
```

1.4 查看系统性能状态

（1）使用 htop 命令工具查看系统性能状态。

（2）使用 sar 命令工具查看系统性能状态。

（3）使用 dstat 命令工具查看系统性能状态。

```
# 使用 htop 命令工具通过图形操作界面查看系统性能状态
[root@Lab-06-Task-01 ~]# htop

# 使用 sar 命令工具查看系统性能状态
# 使用 sar -u 查看 CPU 状态，每 1s 监控一次，共监控 3 次
[root@Lab-06-Task-01 ~]# sar -u 1 3
# 使用 sar -r 查看内存使用率，每 1s 监控一次，共 3 次
[root@Lab-06-Task-01 ~]# sar -r 1 3
# 使用 sar -B 查看内存分页情况，每隔 1s 监控一次，共 3 次
[root@Lab-06-Task-01 ~]# sar -B 1 3
# 使用 sar -W 查看系统交换活动信息，每隔 1s 监控一次，共 3 次
[root@Lab-06-Task-01 ~]# sar -W 1 3
# 使用 sar -d 查看磁盘使用情况，每 1s 监控一次，共 3 次
[root@Lab-06-Task-01 ~]# sar -d 1 3 -p
# 使用 sar -b 查看 I/O 和传输率，每 1s 监控一次，共 3 次
[root@Lab-06-Task-01 ~]# sar -b 1 3
# 使用 sar -n 查看网络情况，每隔 1s 监控一次，共 3 次
[root@Lab-06-Task-01 ~]# sar -n ALL 1 3
# 使用 sar -q 查看队列的长度与负载的状态,每隔 1s 监控一次，共 3 次
[root@Lab-06-Task-01 ~]# sar -q 1 3

# 使用 dstat 命令工具查看系统性能状态
[root@Lab-06-Task-01 ~]# dstat
# 使用 dstat -c 查看 CPU 信息,每隔 1s 监控一次，共 3 次
[root@Lab-06-Task-01 ~]# dstat -c 1 3
# 使用 dstat -d 查看磁盘使用情况，每隔 1s 监控一次，共 3 次
[root@Lab-06-Task-01 ~]# dstat -d 1 3
# 使用 dstat -m 查看内存使用情况，每隔 1s 监控一次，共 3 次
[root@Lab-06-Task-01 ~]# dstat -m 1 3
# 使用 dstat -n 查看网络传送信息，每隔 1s 监控一次，共 3 次
[root@Lab-06-Task-01 ~]# dstat -n 1 3
# 使用 dstat -p 进行进程统计，每隔 1s 监控一次，共 3 次
[root@Lab-06-Task-01 ~]# dstat -p 1 3
# 使用 dstat -r 进行 io 请求统计，每隔 1s 监控一次，共 3 次
[root@Lab-06-Task-01 ~]# dstat -r 1 3
# 使用 dstat -t 进行时间和日期的输出
[root@Lab-06-Task-01 ~]# dstat -t
```

2、使用 Linux-Dash 实现可视化监控

2.1 安装

```
# 使用 dnf 工具安装 Apache
[root@Lab-06-Task-01 ~]# dnf install -y httpd
# 使用 systemctl start 命令启动 Apache 服务
[root@Lab-06-Task-01 ~]# systemctl start httpd
[root@Lab-06-Task-01 ~]# systemctl enable --now httpd
[root@Lab-06-Task-01 ~]# dnf -y install python php php-fpm
# 在/var/www/html 目录下，下载 linux-dash 的源码包
```

```
[root@Lab-06-Task-01 ~]# cd /var/www/html
[root@Lab-06-Task-01 html]# git clone https://github.com/afaqurk/linux-dash.git
# 重新加载 httpd 配置文件
[root@Lab-06-Task-01 html]# systemctl restart httpd
```

2.2 配置

```
# 配置防火墙，允许 80/tcp 端口访问
[root@Lab-06-Task-01 ~]# firewall-cmd --zone=public --add-service=http --perman
ent
[root@Lab-06-Task-01 ~]# firewall-cmd --reload
[root@Lab-06-Task-01 ~]# firewall-cmd --list-all
# 配置 SELinux
[root@Lab-06-Task-01 ~]# setenforce 0
# 修改 SELinux 配置文件
[root@Lab-06-Task-01 ~]# vi /etc/selinux/config
-------------------- config ----------------------
SELINUX=permissive
------------------------------------------------
```

2.3 访问测试

```
# 进行访问安全配置
# 修改 httpd 的配置文件，使用 httpd 启动方式进行登录保护
[root@Lab-06-Task-01 ~]# vim /etc/httpd/conf.d/dash.conf
-------------------- dash.conf ------------------
<Directory /var/www/html/linux-dash/app>
    Options FollowSymLinks
    AllowOverride All
    Order allow,deny
    allow from all
</Directory>
------------------------------------------------
# 在/var/www/html/linux-dash/app 目录下创建.htaccess 文件
[root@Lab-06-Task-01 ~]# vi /var/www/html/linux-dash/app/.htaccess
-------------------- .htaccess  ------------------
AuthType Basic
AuthName "Restricted Files"
AuthUserFile /var/www/html/linux-dash/app/.htpasswd
Require valid-user
------------------------------------------------
# 设置访问 linux-dash 页面的登录账号和密码
[root@Lab-06-Task-01 ~]# htpasswd -c /var/www/html/linux-dash/app/.htpasswd a
dmin
New password:
Re-type new password:
Adding password for user admin
# 重新服务
```

```
[root@Lab-06-Task-01 ~]# systemctl restart httpd
# 在本地主机浏览器上访问 http://172.20.1.17/linux-dash/app/，此时需要输入刚设置的账号和密码：admin，就能访问 Linux-dash 监控页面。
```

3、使用防火墙进行系统安全防护

3.1 配置防火墙

（1）管理防火墙服务

对防火墙服务的管理包括查看防火墙 Firewalld 服务状态、开启、关闭、重启、重新载入防火墙策略等。

```
# 查看防火墙 Firewalld 服务状态
[root@Lab-06-Task-01 ~]# systemctl status firewalld

# 关闭防火墙服务
[root@Lab-06-Task-01 ~]# systemctl stop firewalld

# 开启防火墙服务
[root@Lab-06-Task-01 ~]# systemctl start firewalld

# 重启防火墙服务
[root@Lab-06-Task-01 ~]# systemctl restart firewalld

# 设置防火墙为开机不自启
[root@Lab-06-Task-01 ~]# systemctl disable firewalld

# 设置防火墙为开机自启动
[root@Lab-06-Task-01 ~]# systemctl enable firewalld

# 重新载入防火墙规则
[root@Lab-06-Task-01 ~]# firewall-cmd --reload
```

（2）配置防火墙日志

对防火墙日志的配置有全局日志配置和规则日志配置两部分。全局日志配置是对防火墙日志规则进行配置，防火墙日志服务由系统 rsyslog 服务进行管理，日志默认存放在/var/log/firewalld 日志文件中，日志文件基于日期时间自动归档。规则日志配置是设置防火墙触发特定防火墙规则时记录日志的方式。

```
# 全局日志配置
# 实现防火墙对单播网络通信的日志记录。
# 防火墙日志存放目录变更为/var/log/firewalldlog。
# 防火墙日志记录等级调整为所有等级的日志均记录。
# 使用 vi 命令编辑/etc/firewalld/firewalld.conf 文件
[root@Lab-06-Task-01 ~]# vi /etc/firewalld/firewalld.conf
# ----------/etc/firewalld/firewalld.conf 文件----------
# firewalld.conf 配置文件内容较多，本部分仅显示与防火墙日志配置有关的内容
# 将 LogDenied=off 改为 LogDenied=unicast，实现对单播网络通信的日志记录
LogDenied=unicast
```

```
# ----------/etc/firewalld/firewalld.conf 文件----------

# 使用 vi 命令编辑/etc/rsyslog.conf 文件
[root@Lab-06-Task-01 ~]# vi /etc/rsyslog.conf
# ----------/etc/rsyslog.conf 文件----------
# 在配置文件中增加以下内容，kern.*表示记录所有等级的系统内核产生的日志信息
kern.*                                                  /var/log/firewalldlog/loginfo
# ----------/etc/rsyslog.conf 文件----------

# 创建防火墙日志存放目录
[root@Lab-06-Task-01 ~]# mkdir /var/log/firewalldlog
# 重新载入配置文件
[root@Lab-06-Task-01 ~]# systemctl reload firewalld
# 重启日志相关服务
[root@Lab-06-Task-01 ~]# systemctl restart rsyslog

# 规则日志配置
# 允许本地主机（172.20.1.36）通过 httpd 服务访问服务器。
# 实现触发规则的通信的日志记录。
# 设置日志记录的频率为每秒最多 3 条。
# 根据防火墙规则要求配置
[root@Lab-06-Task-01 ~]# firewall-cmd --permanent --add-rich-rule='rule family=ip
v4 source address=172.20.1.36 service name="http" log level=notice prefix="HTTP
" limit value="3/s" accept'

# 重新载入防火墙配置使其生效
[root@Lab-06-Task-01 ~]# systemctl reload firewalld
```

3.2 依据场景设计防火墙

（1）通过防火墙指定端口和协议允许访问

需求描述：

第一：打开 443/TCP 端口。

第二：永久打开 3690/TCP 端口。

第三：永久打开 100-500/TCP 端口（指定范围内端口全部打开）。

```
# 打开 443/TCP 端口
[root@Lab-06-Task-01 ~]# firewall-cmd --add-port=443/tcp

# 永久打开 3690/TCP 端口
[root@Lab-06-Task-01 ~]# firewall-cmd --add-port=3690/tcp --permanent

# 永久打开 100-500/TCP 端口（指定范围内端口全部打开）
[root@Lab-06-Task-01 ~]# firewall-cmd --remove-port=100-500/tcp --permanent

# 重新载入防火墙配置
[root@Lab-06-Task-01 ~]# firewall-cmd --reload
```

（2）通过防火墙提升远程连接服务的安全性 。

需求描述：

第一：允许地址范围 172.20.1.36/24 内的客户端远程连接服务器，进行远程管理维护。

第二：客户端远程连接服务器时，每分钟最多允许 5 次远程连接，禁止频繁请求。

```
# 使用 firewall-cmd 命令删除默认 ssh 服务规则
[root@Lab-06-Task-01 ~]# firewall-cmd --permanent --remove-service=ssh

# 使用 firewall-cmd 命令添加指定地址能够远程访问的规则
[root@Lab-06-Task-01 ~]# firewall-cmd --permanent --add-rich-rule='rule family=ip
v4 source address=172.20.1.36/24 service name="ssh" limit value="5/s" accept'

# 重新载入防火墙配置
[root@Lab-06-Task-01 ~]# firewall-cmd --reload
```

（3）通过防火墙指定 IP 地址允许/禁止访问。

需求描述：

第一：允许来自 IP 地址为 172.20.1.36 的主机的流量通过防火墙。

第二：禁止来自 IP 地址为 172.20.1.135 的主机的流量通过防火墙。

```
# 允许来自 IP 地址为 172.20.1.36 的主机的流量通过防火墙
[root@Lab-13-Task-01 ~]# firewall-cmd --add-source=172.20.1.36 --permanent

# 禁止来自 IP 地址为 172.20.1.135 的主机的流量通过防火墙
[root@Lab-13-Task-01 ~]# firewall-cmd --remove-source=172.20.1.135 --permanent
# 重新载入防火墙配置
[root@Lab-13-Task-01 ~]# firewall-cmd --reload
```

（4）通过防火墙提升文件传输服务的安全性 。

需求描述：

第一：允许地址范围 172.20.1.36/24 内的客户端通过主动与被动模式访问 FTP 服务器。

```
# 使用 firewall-cmd 命令添加通过主动模式访问 FTP 服务器
[root@Lab-06-Task-01 ~]# firewall-cmd --permanent --add-rich-rule='rule family=ip
v4 source address=172.20.1.36/24 port port=20-21 protocol=tcp limit value="10/m
" accept'
# 使用 firewall-cmd 命令添加本地客户端允许访问 phpMyAdmin
[root@Lab-06-Task-01 ~]# firewall-cmd --permanent --add-rich-rule='rule family=ip
v4 source address=172.20.1.36/24 port port=9000-9020 protocol=tcp limit value="
10/m" accept'

# 重新载入防火墙配置
[root@Lab-06-Task-01 ~]# firewall-cmd --reload
```

（5）查看防火墙日志。

```
[root@Lab-06-Task-01 ~]# cat /var/log/firewalldlog
```