jumpserver

跳板机，堡垒机的概念

跳板机就是一台服务器而已，运维人员在使用管理服务器的时候，必须先连接上跳板机，然后才能去操控内网中的服务器，才能登录到目标设备上进行维护和操作。

开发小张 > 登录跳板机 > 再登录开发服务器

测试小王 > 登录跳板机 > 再登录测试服务器。

跳板机的缺点就是，仅仅实现了服务器登录安全，但是没有实现对于运维人员的行为操控和审计。

跳板机的优缺点

优点：集中式对服务器进行管理

缺点：没有实现对于运维人员的行为操作监控和审计，使用跳板机的过程中，还有可能在服务器上进行错误操作，一旦出现错误操作，很难定位到实施人。

猿来教育

堡垒机运维思想

- 审计也只是事后的行为，审计能够发现问题以及责任人，但是无法防止问题的发生。
- 只有实现了事先严格监控，才能够源头上解决服务器误操作的事故。
- 堡垒机能够创建系统账号，该系统账号功能是属于角色区分的作用，但是也无法确认该账号的执行人。

堡垒机的作用

由于跳板机的不足，企业需要更新，更好，更安全的技术理念去管理服务器的运维操作，需要一种能够满足角色管理，角色授权，信息资源访问控制，操作记录和审计，系统更变和维护控制等等需求，且还能生成服务器资产统计报表等功能的一个IT堡垒机。

1.核心系统运维和安全审计管理

2.过滤和拦截非法请求访问，恶意攻击，拒绝不合法命令，进行审计口监控，报警和责任追踪。

3.报警，记录，分析，处理。

堡垒机核心功能

1.单点登录功能

2.账号管理

3.身份认证

4.资源授权

5.访问控制

6.操作审计

猿来教育

堡垒机应用的场景

1.多个用户使用同一个账号

2.一个用户使用多个账户

3.缺少统一的权限管理平台，难以实现高粒度的命令权限控制

4.对于传统的网络设备无法对运维人员的远程连接命令进行加密，审计。

企业角度看堡垒机

通过更加细致的粒度对企业IT资产设备进行管理，保证企业的it设备资产安全，可靠运行，降低人为操作的风险，避免风险性，保证企业的资源资金安全。

管理角度来看堡垒机

运维人员只需要记录堡垒机的账号密码，一次登录，即可快捷访问多个管理的设备，无须记忆多个账户密码，提升工作效率，且能够对于服务器最大化的安全性操作。

企业真实堡垒机案例

1.运维管理人员手段落后，导致难以发现问题的因素，以及问题的责任制

2.设备的账户管理缺失，连锁酒店的每一个运维人员都能够直接操控所有的服务器，账户密码是及其不安全的，一套完整的信息管理系统，一般需要多个运维人员去管理，因此也就存在了多个账户密码信息，因此存在些问题隐患，比如密码丢失，密码忘记，密码被破解等等，还有就是第三方运维人员，对于服务器的操作，需要有效的进行账号管理，以及账号监控

猿来教育

如何解决

简单的总结堡垒机

就是解决，运维权限混乱，操作无审计。

Jumpserver服务的部署

```
1.linux服务器准备
```

```
硬件配置如下
2cpu   4G内存  50G硬盘

2.想要运行jumpserver，后台相关，需要软件如下

python2解释器
linux的命令，bash解释器  ls命令 >  交给bash解释器，进行翻译之后 > 再告诉linux内核去执行

jumpserver是由python编程语言开发的，旧的jumpserver是由python2开发的，新版本是python3开发的

得准备如下软件版本
python  = 3.6.x
mysql server  必须大于等于 5.6
mariadb  也得是大于等于 5.6  ， 在centos7系统上，mysql由于收费了，开源社区就诞生了mariadb数据库，是开源的
redis 数据库，缓存型数据库


```

猿来教育

部署jumpserver实践

1.非常重要的环境初始化

```
1.环境准备，关闭防火墙服务
[root@teach_jumpserver ~]# iptables -F
[root@teach_jumpserver ~]# systemctl disable firewalld
[root@teach_jumpserver ~]# systemctl stop firewalld

[root@teach_jumpserver ~]# getenforce
Disabled

```

```
2.配置yum源，准备好阿里云的yum源，以及epel源
  wget -O /etc/yum.repos.d/CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-7.repo
  wget -O /etc/yum.repos.d/epel.repo http://mirrors.aliyun.com/repo/epel-7.repo

yum cleann all  #清空原有的yum缓存
yum makecache  # 生成新的yum缓存，便于加速软件下载


3.安装系统初始化所需的软件
  yum install -y bash-completion vim lrzsz wget expect net-tools nc nmap tree dos2unix htop iftop iotop unzip telnet sl psmisc nethogs glances bc ntpdate  openldap-devel gcc


4.安装jumpserver运行所需的依赖环境
yum -y install git python-pip  gcc automake autoconf python-devel vim sshpass lrzsz readline-devel  zlib zlib-devel openssl openssl-devel

git 我们获取jumpserver代码，是在一个全球最大的代码托管平台下载的，github

5.修改系统的字符集，改为是中文的
localedef -c -f UTF-8 -i zh_CN zh_CN.UTF-8
export LC_ALL=zh_CN.UTF-8
# 吧修改字符集的命令，写入全局配置文件
echo 'LANG="zh_CN.UTF-8"' > /etc/locale.conf

6.检查系统编码

```

猿来教育

部署数据库mysql5.6

```
1.获取mysql5.6的软件包
wget https://cdn.mysql.com//Downloads/MySQL-5.6/MySQL-5.6.49-1.el7.x86_64.rpm-bundle.tar

[root@teach_jumpserver ~]# mkdir /teach_jmp
[root@teach_jumpserver ~]# cd /teach_jmp/
[root@teach_jumpserver teach_jmp]# wget https://cdn.mysql.com//Downloads/MySQL-5.6/MySQL-5.6.49-1.el7.x86_64.rpm-bundle.tar

2.解压缩该mysql压缩包
[root@teach_jumpserver teach_jmp]# mkdir mysql_rpm
[root@teach_jumpserver teach_jmp]#
[root@teach_jumpserver teach_jmp]#
[root@teach_jumpserver teach_jmp]# tar -xf MySQL-5.6.49-1.el7.x86_64.rpm-bundle.tar -C ./mysql_rpm/


3.使用yum命令，安装一系列的rpm包
[root@teach_jumpserver teach_jmp]# cd mysql_rpm/
[root@teach_jumpserver mysql_rpm]#
[root@teach_jumpserver mysql_rpm]# ls
MySQL-client-5.6.49-1.el7.x86_64.rpm    MySQL-server-5.6.49-1.el7.x86_64.rpm         MySQL-test-5.6.49-1.el7.x86_64.rpm
MySQL-devel-5.6.49-1.el7.x86_64.rpm     MySQL-shared-5.6.49-1.el7.x86_64.rpm
MySQL-embedded-5.6.49-1.el7.x86_64.rpm  MySQL-shared-compat-5.6.49-1.el7.x86_64.rpm
[root@teach_jumpserver mysql_rpm]#
[root@teach_jumpserver mysql_rpm]#
[root@teach_jumpserver mysql_rpm]# yum localinstall ./*
```

猿来教育

```

4.安装完毕后，检查mysql的配置文件，做如下的修改
[root@teach_jumpserver mysql_rpm]# cat /etc/my.cnf
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0
# Settings user and group are ignored when systemd is used.
# If you need to run mysqld under a different user or group,
# customize your systemd unit file for mariadb according to the
# instructions in http://fedoraproject.org/wiki/Systemd

# 注意这里，要修改此2行配置
[mysqld_safe]
log-error=/var/log/mysql/mysql.log
pid-file=/var/run/mysql/mysql.pid

#
# include all files from the config directory
#
!includedir /etc/my.cnf.d


4.1 启动mysql服务端
[root@teach_jumpserver mysql_rpm]# systemctl start mysql
[root@teach_jumpserver mysql_rpm]#
[root@teach_jumpserver mysql_rpm]#
[root@teach_jumpserver mysql_rpm]# netstat -tunlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
```

猿来教育

```
tcp        0      0 10.0.1.100:1194         0.0.0.0:*               LISTEN      1168/openvpn
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1155/sshd
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      1433/master
tcp6       0      0 :::3306                 :::*                    LISTEN      2721/mysqld
tcp6       0      0 :::22                   :::*                    LISTEN      1155/sshd
tcp6       0      0 ::1:25                  :::*                    LISTEN      1433/master



5.对mysql进行初始化，mysql5.6版本在安装完毕后，会默认生成一个root的随机密码，如下
[root@teach_jumpserver mysql_rpm]# cat ~/.mysql_secret
# The random password set for the root user at Thu Aug 27 17:49:42 2020 (local time): Dg37dxfIM041dfI6

6.是否要修改原有的密码，自行决定
mysqladmin -uroot -pDg37dxfIM041dfI6  password chaoge666
# 更为安全的修改root密码的操作
mysql> update mysql.user  set password=password('chaoge888') where user='root';
Query OK, 4 rows affected (0.01 sec)
Rows matched: 4  Changed: 4  Warnings: 0

mysql>
mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

7.再次用新密码登录mysql5.6
```

```
[root@teach_jumpserver mysql_rpm]# mysql -uroot -p

8.登录数据库后，创建运行jumpserver所需的用户信息
mysql> create database  jumpserver default charset 'utf8' collate 'utf8_bin';
Query OK, 1 row affected (0.00 sec)

# 创建完毕数据库后，再创建用户，且设置密码
mysql> create user 'jumpserver'@'%' IDENTIFIED BY 'chaoge888';
Query OK, 0 rows affected (0.00 sec)

9.给该用户授予访问数据库的权限
mysql> grant all privileges on jumpserver.* to 'jumpserver'@'%' identified by 'chaoge888';
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

```

猿来教育

部署python3.6

由于新版jumpserver是python3.6开发的，因此我们得准备好python3.6的环境

```
1.下载python3.6的源代码，可以在线下载，也可以向超哥索要软件包都行
cd /teach_jmp && \
wget https://www.python.org/ftp/python/3.6.10/Python-3.6.10.tgz

2.开始源码安装python3，进行编译三部曲
[root@teach_jumpserver teach_jmp]# tar -zxf Python-3.6.10.tgz

# 指定python3的安装目录
# 编译第一曲，指定安装路径， 与编译参数
[root@teach_jumpserver Python-3.6.10]# ./configure  --prefix=/teach_jmp/python3.6.10/

# 第二曲，第三曲
[root@teach_jumpserver Python-3.6.10]# make && make install

# 运行python3的两种方式：
方式1：使用绝对路径
[root@teach_jumpserver teach_jmp]# /teach_jmp/python3.6.10/bin/python3
Python 3.6.10 (default, Aug 27 2020, 18:31:37)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-39)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>> print("hello chaoge~~~~")
hello chaoge~~~~
>>>
>>>
>>> exit()

# 方式2，省事，使用环境变量的形式







3.配置python3的环境变量，可以直接使用python3的命令
[root@teach_jumpserver bin]# tail -1 /etc/profile
PATH="/teach_jmp/python3.6.10/bin:$PATH"

```

猿来教育

```
[root@teach_jumpserver bin]# source /etc/profile
[root@teach_jumpserver bin]#
[root@teach_jumpserver bin]#
[root@teach_jumpserver bin]#
[root@teach_jumpserver bin]# echo $PATH
/teach_jmp/python3.6.10/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin

# 使用环境变量方式，启动python3解释器
[root@teach_jumpserver bin]# python3
Python 3.6.10 (default, Aug 27 2020, 18:31:37)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-39)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()




4. 创建python运行所需的虚拟环境
为什么需要用虚拟环境?
  答案是：因为你不希望，运行一个程序，缺搞乱你的环境变量

PATH变量

ls
cd
默认是去PATH里面寻找，是否有ls cd等命令，which ls一样 ，会得到ls的命令绝对路径

ls
mkdir
echo

```

猿来教育

```
# linux系统内置了python2解释器，那么在你安装了python3之后，你的系统上有多个解释器版本共存
你得明确，你的项目用的是哪一个解释器去运行

# python的程序，运行，需要安装很多的模块，





```

图1，理解linux上有多个版本的python解释器，去运行项目的概念

图2，为什么要用虚拟环境 virtualenv工具

猿来教育

python3创建虚拟环境

```
1.安装虚拟环境工具，python3是一个解释器，还有一个工具叫做 pip3 ，这是给python3安装模块的
可以理解为，linux上我们需要使用各种系统软件，可以方便的用yum自动化下载安装

python3程序代码，在运行的时候，必须也下载一些软件模块，才能运行，使用的是pip3安装


```

```
2.如果你的python3在安装模块的时候，像超哥一样报错了，由于缺少ssl，python3无法使用，解决方式如下

删掉编译安装的python3，然后安装openssl工具，然后重新编译安装python3才行
yum install openssl openssl-devel -y


3.再次编译安装完成python3后，再次尝试，安装python3的模块
# 先更新一下pip3的下载源，就如同更换yum源一个概念
# 操作步骤如下
mkdir ~/.pip
touch ~/.pip/pip.conf

# 最终pip3的源，文件内容如下
[root@teach_jumpserver ~]# cat ~/.pip/pip.conf
[global]
index-url =  https://mirrors.aliyun.com/pypi/simple/

4.下载虚拟环境工具
[root@teach_jumpserver ~]# pip3 install virtualenv

5.使用虚拟环境工具，再创建出一个python3解释器，用于运行代码
[root@teach_jumpserver teach_jmp]# virtualenv --python=python3 jmp_venv1
created virtual environment CPython3.6.10.final.0-64 in 618ms
  creator CPython3Posix(dest=/teach_jmp/jmp_venv1, clear=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/root/.local/share/virtualenv)
    added seed packages: pip==20.2.2, setuptools==49.6.0, wheel==0.35.1
```

源来教育

```
  activators BashActivator,CShellActivator,FishActivator,PowerShellActivator,PythonActivator,XonshActivator


6.此时你的linux服务器上就有2个python3解释器了
解释器本体是：/teach_jmp/python3.6.10/bin/python3
我们创建了一个虚拟的解释器，路径是：/teach_jmp/jmp_venv1/bin/python3

7.激活虚拟环境，其实是默认修改了环境变量
[root@teach_jumpserver bin]# source /teach_jmp/jmp_venv1/bin/activate
(jmp_venv1) [root@teach_jumpserver bin]#


# 可以退出虚拟环境，查看解释器的路径，效果
(jmp_venv1) [root@teach_jumpserver bin]# deactivate
[root@teach_jumpserver bin]#
[root@teach_jumpserver bin]#
[root@teach_jumpserver bin]#
[root@teach_jumpserver bin]#
[root@teach_jumpserver bin]#
[root@teach_jumpserver bin]# which python3
/teach_jmp/python3.6.10/bin/python3


```

部署redis数据库

mysql关系型数据库，磁盘型数据库，数据是以文件形式，存储在磁盘上的，可以持久化长期存储

redis内存性数据库，缓存性数据库。

```
1.安装redis的形式
rpm包手动安装，需要手动解决依赖，不推荐使用
yum自动化安装，适合软件调试学习使用，安装自动解决依赖，很好用
源代码编译安装redis


2.选择yum自动化安装即可
配置好yum源才行，epel源

yum install redis -y

systemctl start redis

netstat -tunlp|grep 6379
```

部署jumpserver服务

一个后台程序，基本上都是需要依赖于数据库才能运行，后台程序在启动的时候，代码就回去连接数据库，保证数据库，正确启动，且可以正确连接，否则后台程序是起不来的。

```
1.获取jumpserver程序的代码，github有公共仓库，所有人都可以下载，私有仓库，只有企业内部人员，用账号密码登录后下载
wget https://github.com/jumpserver/jumpserver/releases/download/v2.1.0/jumpserver-v2.1.0.tar.gz

2.解压缩源码，且安装运行jumpserver系统必须的依赖组件
[root@teach_jumpserver teach_jmp]# jumpserver-v2.1.0.tar.gz
[root@teach_jumpserver teach_jmp]# ln -s /teach_jmp/jumpserver-v2.1.0 /teach_jmp/jumpserver
```

猿来教育

```

安装依赖关系
yum install -y bash-completion vim lrzsz wget expect net-tools nc nmap tree dos2unix htop iftop iotop unzip telnet sl psmisc nethogs glances bc ntpdate  openldap-devel

3.安装运行jumpserver所需要的模块（由python开发的程序，必须安装该程序使用到的一些模块，才能正确运行）

注意
注意
注意
安装jumpserver模块，听老师的，先激活虚拟环境，然后再安装
[root@teach_jumpserver requirements]# source /teach_jmp/jmp_venv1/bin/activate
# 安装模块
(jmp_venv1) [root@teach_jumpserver requirements]# pip3 install -r /teach_jmp/jumpserver/requirements/requirements.txt

```

猿来教育

修改jumpserver程序运行的配置文件

```
1.修改配置文件，默认未修改的配置文件如下，我们需要做一些定制修改
(jmp_venv1) [root@teach_jumpserver jumpserver]# grep -Ev '^#|^$' config.yml
SECRET_KEY:
BOOTSTRAP_TOKEN:
DB_ENGINE: mysql
DB_HOST: 127.0.0.1
DB_PORT: 3306
DB_USER: jumpserver
DB_PASSWORD:
DB_NAME: jumpserver
```

```
HTTP_BIND_HOST: 0.0.0.0
HTTP_LISTEN_PORT: 8080
WS_LISTEN_PORT: 8070
REDIS_HOST: 127.0.0.1
REDIS_PORT: 6379



2.生成密钥
if [ "$SECRET_KEY" = "" ]; then SECRET_KEY=`cat /dev/urandom | tr -dc A-Za-z0-9 | head -c 50`; echo "SECRET_KEY=$SECRET_KEY" >> ~/.bashrc; echo $SECRET_KEY; else echo $SECRET_KEY; fi


# 生成密钥，都是随机的
(jmp_venv1) [root@teach_jumpserver jumpserver]# if [ "$SECRET_KEY" = "" ]; then SECRET_KEY=`cat /dev/urandom | tr -dc A-Za-z0-9 | head -c 50`; echo "SECRET_KEY=$SECRET_KEY" >> ~/.bashrc; echo $SECRET_KEY; else echo $SECRET_KEY; fi
iKsMR9P7b6nYq3J02vFvj1KZBuqc8vQDRv9975rLN5KKmiYZ4w


# 生成token密钥
(jmp_venv1) [root@teach_jumpserver jumpserver]# if [ "$BOOTSTRAP_TOKEN" = "" ]; then BOOTSTRAP_TOKEN=`cat /dev/urandom | tr -dc A-Za-z0-9 | head -c 16`; echo "BOOTSTRAP_TOKEN=$BOOTSTRAP_TOKEN" >> ~/.bashrc; echo $BOOTSTRAP_TOKEN; else echo $BOOTSTRAP_TOKEN; fi
LLyyAZ8dbcg0nQ9m


# 修改，写入配置文件，至此配置文件就修改完毕了
(jmp_venv1) [root@teach_jumpserver jumpserver]# grep -Ev '^#|^$' config.yml
```

```
SECRET_KEY: iKsMR9P7b6nYq3J02vFvj1KZBuqc8vQDRv9975rLN5KKmiYZ4w
BOOTSTRAP_TOKEN: LLyyAZ8dbcg0nQ9m
DB_ENGINE: mysql
DB_HOST: 127.0.0.1
DB_PORT: 3306
DB_USER: jumpserver
DB_PASSWORD: chaoge888
DB_NAME: jumpserver
HTTP_BIND_HOST: 0.0.0.0
HTTP_LISTEN_PORT: 8080
WS_LISTEN_PORT: 8070
REDIS_HOST: 127.0.0.1
REDIS_PORT: 6379
(jmp_venv1) [root@t
```

猿来教育

对python程序进行数据库迁移

jumpserver这个程序是由python的web框架django开发而来，必须得先进行数据库迁移，生成库表的信息，才能运行程序

```
1.jumpserver后台程序，数据库迁移命令
(jmp_venv1) [root@teach_jumpserver apps]# python3 /teach_jmp/jumpserver/apps/manage.py makemigrations
Migrations for 'tickets':
  tickets/migrations/0002_auto_20200830_2020.py
    - Alter field type on ticket

2.数据库迁移命令

python3  /teach_jmp/jumpserver/apps/manage.py migrate
```

启动jms服务

```
(jmp_venv1) [root@teach_jumpserver jumpserver]#
/teach_jmp/jumpserver/jms start -d
```

部署koko组件

koko是用golang编程语言开发的一个组件，和之前的coco组件相比（python开发的）相比而言，性能，效率，系统资源利用率都更高了。

```
1.下载koko源代码
wget
https://github.com/jumpserver/koko/releases/download/v2.1.0/koko-v2.1.0-linux-amd64.tar.gz

2.解压缩配置koko软件
(jmp_venv1) [root@teach_jumpserver teach_jmp]# tar -zxf koko-v2.1.0-linux-amd64.tar.gz
(jmp_venv1) [root@teach_jumpserver teach_jmp]# chown -R
root:root koko-v2.1.0-linux-amd64
(jmp_venv1) [root@teach_jumpserver teach_jmp]#
(jmp_venv1) [root@teach_jumpserver teach_jmp]#
(jmp_venv1) [root@teach_jumpserver teach_jmp]# ln -s
/teach_jmp/koko-v2.1.0-linux-amd64 /teach_jmp/koko

3.修改koko配置文件信息
(jmp_venv1) [root@teach_jumpserver koko]# grep -Ev '^#|^$'
/teach_jmp/koko/config.yml
CORE_HOST: http://127.0.0.1:8080
BOOTSTRAP_TOKEN: LLyyAZ8dbcg0nQ9m
LOG_LEVEL: INFO
REDIS_HOST: 127.0.0.1
REDIS_PORT: 6379
REDIS_PASSWORD:
REDIS_CLUSTERS:
```

```
REDIS_DB_ROOM:

4.启动koko程序
启动命令：/teach_jmp/koko/koko -d

(jmp_venv1) [root@teach_jumpserver koko]# ps -ef|grep koko
root       8440   1840  0 21:25 pts/0    00:00:00 tail -f data/logs/koko.log
root       8486      1  0 21:25 ?        00:00:00 /teach_jmp/koko/koko -d
root       8495   8443  0 21:25 pts/1    00:00:00 grep --color=auto koko

5.可以检查koko的日志，明确koko是否正确启动
(jmp_venv1) [rot@teach_jumpserver koko]# tail /teach_jmp/koko/data/logs/koko.log
2020-08-30 21:18:01 [ERRO] POST http://127.0.0.1:8080/api/v2/terminal/terminal-registrations/ failed, get code: 401, {"detail":"身份认证信息未提供。"}
2020-08-30 21:18:01 [ERRO] register access key failed
2020-08-30 21:25:39 [INFO] Exchange share room type: local
2020-08-30 21:25:39 [INFO] Start HTTP server at 0.0.0.0:5000
2020-08-30 21:25:39 [INFO] Start SSH server at 0.0.0.0:2222

6.检查koko的端口
(jmp_venv1) [root@teach_jumpserver koko]# netstat -tunlp|grep 2222
tcp6       0      0 :::2222                 :::*                    LISTEN      8486/koko
```

部署Guacamole组件

```
1.获取软件代码
```

```
(jmp_venv1) [root@teach_jumpserver teach_jmp]# ll 2020-07-22-16-48-00-docker-guacamole-v2.1.0.tar.gz

2.解压缩配置
(jmp_venv1) [root@teach_jumpserver teach_jmp]# tar -zxvf 2020-07-22-16-48-00-docker-guacamole-v2.1.0.tar.gz
(jmp_venv1) [root@teach_jumpserver teach_jmp]# mv docker-guacamole-2.1.0/ guacamole
(jmp_venv1) [root@teach_jumpserver teach_jmp]# cd guacamole/
(jmp_venv1) [root@teach_jumpserver guacamole]# ls
Dockerfile           guacamole-auth-jumpserver-1.0.0.jar  README.md  s6-overlay-amd64.tar.gz
guacamole-1.0.0.war  guacamole-server-1.2.0.tar.gz        root       ssh-forward.tar.gz

3.继续解压执行程序
(jmp_venv1) [root@teach_jumpserver guacamole]# tar -zxf guacamole-server-1.2.0.tar.gz
(jmp_venv1) [root@teach_jumpserver guacamole]# tar -zxf ssh-forward.tar.gz

4.编译安装该软件程序
(jmp_venv1) [root@teach_jumpserver guacamole]# cd guacamole-server-1.2.0/
(jmp_venv1) [root@teach_jumpserver guacamole-server-1.2.0]# ls
aclocal.m4  build-aux    configure     CONTRIBUTING  Dockerfile  m4           Makefile.in  README  util
bin         config.h.in  configure.ac  doc           LICENSE     Makefile.am  NOTICE       src

# 编译软件之前，基本上都要吧编译环境准备好
yum install cairo-devel libjpeg-turbo-devel     libjpeg-devel    libpng-devel libtool uuid-devel -y
# 可选的软件依赖
```

猿来教育

```
yum install  freerdp-devel pango-devel     libssh2-devel libtelnet-devel libvncserver-devel libwebsockets-devel pulseaudio-libs-devel openssl-devel libvorbis-devel libwebp-devel -y


5.安装FFmpeg工具

sudo yum install epel-release -y
sudo rpm -v --import http://li.nux.ro/download/nux/RPM-GPG-KEY-nux.ro
sudo rpm -Uvh http://li.nux.ro/download/nux/dextop/el7/x86_64/nux-dextop-release-0-5.el7.nux.noarch.rpm

yum install ffmpeg ffmpeg-develI -y

6.编译安装guacamole
(jmp_venv1) [root@teach_jumpserver guacamole-server-1.2.0]# ./configure --with-init-dir=/etc/init.d

(jmp_venv1) [root@teach_jumpserver guacamole-server-1.2.0]# make && make install

7.部署java开发环境
yum install -y java-1.8.0-openjdk

8.创建运行guacamole所需的文件夹
mkdir -p /config/guacamole /config/guacamole/extensions /config/guacamole/record /config/guacamole/drive && \
chown daemon:daemon /config/guacamole/record /config/guacamole/drive && \
cd /config

```

猿来教育

```
9.下载tomcat工具,用于运行java项目
cd /opt && \
wget http://mirrors.tuna.tsinghua.edu.cn/apache/tomcat/tomcat-9/v9.0.36/bin/apache-tomcat-9.0.36.tar.gz

10.部署guacamole和tomcat工具的结合，需要修改他们的配置文件
cd /opt && \
tar -xf apache-tomcat-9.0.36.tar.gz && \
mv apache-tomcat-9.0.36 tomcat9 && \
rm -rf /opt/tomcat9/webapps/* && \
sed -i 's/Connector port="8080"/Connector port="8081"/g' /opt/tomcat9/conf/server.xml && \
echo "java.util.logging.ConsoleHandler.encoding = UTF-8" >> /opt/tomcat9/conf/logging.properties && \
ln -sf /teach_jmp/guacamole/guacamole-1.0.0.war /opt/tomcat9/webapps/ROOT.war && \
ln -sf /teach_jmp/guacamole/guacamole-auth-jumpserver-1.0.0.jar /config/guacamole/extensions/guacamole-auth-jumpserver-1.0.0.jar && \
ln -sf /teach_jmp/guacamole/root/app/guacamole/guacamole.properties /config/guacamole/guacamole.properties

11.设置guacamole的运行环境变量
export JUMPSERVER_SERVER=http://127.0.0.1:8080
echo "export JUMPSERVER_SERVER=http://127.0.0.1:8080" >> ~/.bashrc
export BOOTSTRAP_TOKEN=zxffNymGjP79j6BN
echo "export BOOTSTRAP_TOKEN=zxffNymGjP79j6BN" >> ~/.bashrc
export JUMPSERVER_KEY_DIR=/config/guacamole/keys
echo "export JUMPSERVER_KEY_DIR=/config/guacamole/keys" >> ~/.bashrc
export GUACAMOLE_HOME=/config/guacamole
echo "export GUACAMOLE_HOME=/config/guacamole" >> ~/.bashrc
```

猿来教育

```
export GUACAMOLE_LOG_LEVEL=ERROR
echo "export GUACAMOLE_LOG_LEVEL=ERROR" >> ~/.bashrc
export JUMPSERVER_ENABLE_DRIVE=true
echo "export JUMPSERVER_ENABLE_DRIVE=true" >> ~/.bashrc

12.启动服务
(jmp_venv1) [root@teach_jumpserver opt]# /etc/init.d/guacd start
Starting guacd: guacd[13375]: INFO: Guacamole proxy daemon (guacd) version 1.2.0 started
SUCCESS
(jmp_venv1) [root@teach_jumpserver opt]#
(jmp_venv1) [root@teach_jumpserver opt]#
(jmp_venv1) [root@teach_jumpserver opt]#
(jmp_venv1) [root@teach_jumpserver opt]# sh /opt/tomcat9/bin/startup.sh
Using CATALINA_BASE:   /opt/tomcat9
Using CATALINA_HOME:   /opt/tomcat9
Using CATALINA_TMPDIR: /opt/tomcat9/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /opt/tomcat9/bin/bootstrap.jar:/opt/tomcat9/bin/tomcat-juli.jar
Tomcat started.



```

猿来教育

Lina组件部署

```
# 提前准备好nginx服务
yum install nginx -y

```

```
1.获取代码
wget https://github.com/jumpserver/lina/releases/download/v2.1.0/lina-v2.1.0.tar.gz

2.解压缩linna组件
(jmp_venv1) [root@teach_jumpserver teach_jmp]# tar -zxf lina-v2.1.0.tar.gz
(jmp_venv1) [root@teach_jumpserver teach_jmp]# mv lina-v2.1.0 lina
(jmp_venv1) [root@teach_jumpserver teach_jmp]#
(jmp_venv1) [root@teach_jumpserver teach_jmp]#
(jmp_venv1) [root@teach_jumpserver teach_jmp]#
(jmp_venv1) [root@teach_jumpserver teach_jmp]#
(jmp_venv1) [root@teach_jumpserver teach_jmp]#
(jmp_venv1) [root@teach_jumpserver teach_jmp]# chown -R nginx:nginx lina
```

部署Luna组件

猿来教育

```
1.获取luna代码
wget https://github.com/jumpserver/luna/releases/download/v2.1.1/luna-v2.1.1.tar.gz
(jmp_venv1) [root@teach_jumpserver teach_jmp]# tar -zxf luna-v2.1.1.tar.gz
(jmp_venv1) [root@teach_jumpserver teach_jmp]# mv luna-v2.1.1 luna
(jmp_venv1) [root@teach_jumpserver teach_jmp]#
(jmp_venv1) [root@teach_jumpserver teach_jmp]#
(jmp_venv1) [root@teach_jumpserver teach_jmp]# chown -R root.root luna


```

部署nginx

nginx作用在处理静态文件，以及用于对jumpserver后台程序的反向代理

猿来教育

```
1.安装nginx
yum install nginx -y

2。修改nginx配置文件，删除一些默认的配置，然后添加新的配置
sed -i  '38,58d' /etc/nginx/nginx.conf

3.加入新的虚拟主机配置
server {
    listen 80;

    client_max_body_size 100m;  # 录像及文件上传大小限制

    location /ui/ {
        try_files $uri / /index.html;
        alias /teach_jmp/lina/;
    }

    location /luna/ {
        try_files $uri / /index.html;
        alias /teach_jmp/luna/;  # luna 路径，如果修改安装目录，此处需要修改
    }

    location /media/ {
        add_header Content-Encoding gzip;
        root /teach_jmp/jumpserver/data/;  # 录像位置，如果修改安装目录，此处需要修改
    }

    location /static/ {
        root /teach_jmp/jumpserver/data/;  # 静态资源，如果修改安装目录，此处需要修改
    }

    location /koko/ {
        proxy_pass       http://localhost:5000;
        proxy_buffering off;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        access_log off;
    }

```

猿来教育

```
    location /guacamole/ {
        proxy_pass       http://localhost:8081/;
        proxy_buffering off;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection $http_connection;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        access_log off;
    }

    location /ws/ {
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_pass http://localhost:8070;
        proxy_http_version 1.1;
        proxy_buffering off;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
    }

    location /api/ {
        proxy_pass http://localhost:8080;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }

    location /core/ {
        proxy_pass http://localhost:8080;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }

    location / {
        rewrite ^/(.*)$ /ui/$1 last;
    }
}
```

启动nginx服务

```
nginx -t
nginx
```

猿来教育

补充koko启动

新版jumpserver启动koko组件时，经常会出现问题

```
2020-08-30 21:18:01 [ERRO] POST http://127.0.0.1:8080/api/v2/terminal/terminal-registrations/ failed, get code: 401, {"detail":"身份认证信息未提供。"}

```

想要彻底解决，可以按超哥如下方案

```
1.删除koko的data目录下的.access_key文件
(jmp_venv1) [root@teach_jumpserver keys]# pwd
/teach_jmp/koko/data/keys
(jmp_venv1) [root@teach_jumpserver keys]# ls -a
```

```
.  ..  .access_key  data

2.重新生成获取两个重要的密钥，然后修改jumpserver后台所有组件的配置，重启服务
SECRET_KEY
BOOTSTRAP_TOKEN

重新生成这2个key
第一步：修改环境变量配置文件
vim ~/.bashrc
删除该2个变量
SECRET_KEY
BOOTSTRAP_TOKEN

第二步：重新登陆linux会话，检查该变量是否存在，没有几正确
[root@teach_jumpserver ~]# echo $SECRET_KEY

[root@teach_jumpserver ~]# echo $BOOTSTRAP_TOKEN


第三步：重新生成这2个密钥
[root@teach_jumpserver ~]# if [ "$SECRET_KEY" = "" ]; then SECRET_KEY=`cat /dev/urandom | tr -dc A-Za-z0-9 | head -c 50`; echo "SECRET_KEY=$SECRET_KEY" >> ~/.bashrc; echo $SECRET_KEY; else echo $SECRET_KEY; fi
bJbFvkfOpW04RWjYim1aEtazbdC1YXlZa2Q4VrS3w4nnXsvai3


[root@teach_jumpserver ~]# if [ "$BOOTSTRAP_TOKEN" = "" ]; then BOOTSTRAP_TOKEN=`cat /dev/urandom | tr -dc A-Za-z0-9 | head -c 16`; echo "BOOTSTRAP_TOKEN=$BOOTSTRAP_TOKEN" >> ~/.bashrc; echo $BOOTSTRAP_TOKEN; else echo $BOOTSTRAP_TOKEN; fi
3JZr7GpSr0Loeoyu
```

泉来教育

```


第四步：修改jumpserver后台配置文件 config.yml
(jmp_venv1) [root@teach_jumpserver jumpserver]# grep -Ev '^#|^$' config.yml
SECRET_KEY: bJbFvkfOpW04RWjYim1aEtazbdC1YXlZa2Q4VrS3w4nnXsvai3
BOOTSTRAP_TOKEN: 3JZr7GpSr0Loeoyu
DB_ENGINE: mysql
DB_HOST: 127.0.0.1
DB_PORT: 3306
DB_USER: jumpserver
DB_PASSWORD: chaoge888
DB_NAME: jumpserver
HTTP_BIND_HOST: 0.0.0.0
HTTP_LISTEN_PORT: 8080
WS_LISTEN_PORT: 8070
REDIS_HOST: 127.0.0.1
REDIS_PORT: 6379

第五步：重新启动jumpserver核心后台程序
(jmp_venv1) [root@teach_jumpserver jumpserver]# /teach_jmp/jumpserver/jms stop
(jmp_venv1) [root@teach_jumpserver jumpserver]# /teach_jmp/jumpserver/jms start -d

第六步：修改koko的配置文件，准备启动koko
此时最新的koko配置文件，长这样
(jmp_venv1) [root@teach_jumpserver koko]# grep -Ev '^#|^$' config.yml
CORE_HOST: http://127.0.0.1:8080
BOOTSTRAP_TOKEN: 3JZr7GpSr0Loeoyu
LOG_LEVEL: INFO
REDIS_HOST: 127.0.0.1
REDIS_PORT: 6379
```

源来教育

```
REDIS_PASSWORD:
REDIS_CLUSTERS:
REDIS_DB_ROOM:

第七步：见证koko的正确启动
(jmp_venv1) [root@teach_jumpserver koko]# ./koko  -d
(jmp_venv1) [root@teach_jumpserver koko]#
(jmp_venv1) [root@teach_jumpserver koko]#
(jmp_venv1) [root@teach_jumpserver koko]# netstat -tunlp|grep 2222
tcp6       0      0 :::2222                 :::*                    LISTEN      36246/./koko
(jmp_venv1) [root@teach_jumpserver koko]# netstat -tunlp|grep 5000
tcp6       0      0 :::5000                 :::*                    LISTEN      36246/./koko

第八步：由于修改了密钥，还会影响到其他的服务，需要修改配置
设置guacamole的运行环境变量
export JUMPSERVER_SERVER=http://127.0.0.1:8080
echo "export JUMPSERVER_SERVER=http://127.0.0.1:8080" >> ~/.bashrc
export BOOTSTRAP_TOKEN=3JZr7GpSr0Loeoyu
echo "export BOOTSTRAP_TOKEN=3JZr7GpSr0Loeoyu" >> ~/.bashrc
export JUMPSERVER_KEY_DIR=/config/guacamole/keys
echo "export JUMPSERVER_KEY_DIR=/config/guacamole/keys" >> ~/.bashrc
export GUACAMOLE_HOME=/config/guacamole
echo "export GUACAMOLE_HOME=/config/guacamole" >> ~/.bashrc
export GUACAMOLE_LOG_LEVEL=ERROR
echo "export GUACAMOLE_LOG_LEVEL=ERROR" >> ~/.bashrc
export JUMPSERVER_ENABLE_DRIVE=true
echo "export JUMPSERVER_ENABLE_DRIVE=true" >> ~/.bashrc

```

```

第九步: 重启服务
(jmp_venv1) [root@teach_jumpserver koko]# /etc/init.d/guacd restart
(jmp_venv1) [root@teach_jumpserver koko]# /opt/tomcat9/bin/shutdown.sh
(jmp_venv1) [root@teach_jumpserver koko]# /opt/tomcat9/bin/startup.sh


```

Jumpserver实践

猿来教育

```
1.启动数据库,mysql,redis
systemctl start mysqld
systemctl start redis

2.激活python的虚拟环境, 然后启动jms核心后台
source /teach_jmp/jmp_venv1/bin/activate

/teach_jmp/jumpserver/jms start -d

3.启动koko程序
/teach_jmp/koko/koko -d

4.guacd程序启动
 /etc/init.d/guacd start

 5.启动tomcat程序
 bash /opt/tomcat9/bin/startup.sh
```

```

6.web服务器启动
nginx


```

给目标机器添加防火墙规则

```
1.只允许jumnpserver机器的ip可以登陆，其他机器拒绝
[root@web01 ~ 14:49:42]$iptables -A INPUT -s 10.0.1.100 -p tcp --dport 22 -j ACCEPT
iptables -A INPUT -p tcp --dport 22 -j REJECT
```

这一节的内容：

- 修改admin密码
- 设置防火墙规则，只允许堡垒机登陆linux

猿来教育

Jumpserver用户管理

Jumpserver资产管理

资产：服务器，路由器，交换机等设备，资产

管理用户

Root 超级管理员用户

sudo命令，伪管理员，默认以root身份去执行命令，因此要慎用，我们可以基于sudo命令作更多的权限控制

zhangsan 系统等普通用户，权限很低

admin jumpserver管理员用户

Chaoge jumpserver普通用户，权限较低

管理用户，【客户端 > jumpserver > 目标服务器】

管理用户值得就是 `被管理机器` 上的root用户，或者是可以使用sudo权限的用户，jumpserver利用该管理用户在目标机器上，进行远程的命令执行，推送系统用户，获取资产硬件信息，指标等等。

系统用户

猿来教育

/etc/passwd 是系统级的超级用户，普通用户，等等，有些事可以允许登陆服务器的，使用ssh协议

那么jumpserver的系统用户，针对jumpserver操控，登陆普通机器，所使用的一些特有用户

web终端功能

luna提供web终端界面

命令行跳板机

koko的功能就来了

猿来教育