SeMF 内网安全管理平台部署指南

项目介绍

SecurityManageFramwork-SeMF,企业内网安全管理平台，包含资产管理，漏洞管理，账号管理，知识库管、安全扫描自动化功能模块，可用于企业内部的安全管理。 本平台旨在帮助安全人员少，业务线繁杂，周期巡检困难，自动化程度低的甲方，更好的实现企业内部的安全管理。

软件架构

后端系统 python3 + django2 + rabbitmq 实现。 前端显示 layui + bootstarp,使用开源模板

X-admin:http://x.xuebingsi.com/

项目特点

可自定义用户类型及权限信息，初始化中生成安全人员，运维人员，网络人员和业务人员四种类型

资产类型和资产属性可在后台自定义，根据需要进行扩展

内网资产发现和端口扫描可自动化进行

完整的漏洞跟进和扫描器漏洞过滤

一、 基础环境安装

1.1 准备工作

这里用 Centos7.8，先关闭防火墙和 selinux

```
systemctl stop firewalld.service
setenforce 0  //临时关闭 selinux
```

修改主机名及更新系统：

```
hostnamectl --static set-hostname SeMF
yum update -y
```

安装相关软件

```
yum install zlib-devel bzip2-devel openssl-devel ncurses-devel readline-devel sqlite-devel tk-devel gcc make vim lrzsz wget git gcc nmap -y
```

其它会用到的文件，如果网国下载慢可以从网盘下载

https://cloud.189.cn/t/fiANfmneMbYf（访问码：2xd3）

```
all-2.0-2020-07.tar.gz
erlang-19.0.4-1.el7.centos.x86_64.rpm
Nessus-8.10.1-debian6_amd64.deb
Python-3.6.5.tar.xz
rabbitmq-server-3.6.9-1.el7.noarch.rpm
```

1.2 Python3 安装

下载 python3 安装包，这里我选择的是 3.6.5

```
wget https://www.python.org/ftp/python/3.6.5/Python-3.6.5.tar.xz
tar -xvJf Python-3.6.5.tar.xz && cd Python-3.6.5
./configure prefix=/usr/local/python3
make && make install
//创建软链接
ln -s /usr/local/python3/bin/python3 /usr/bin/python3
```

1.3 Rabitmq 安装

下载 erlang (用迅雷下载)

```
wget http://www.rabbitmq.com/releases/erlang/erlang-19.0.4-1.el7.centos.x86_64.rpm
rpm -ivh erlang-19.0.4-1.el7.centos.x86_64.rpm
```

```
erl -version
//查看版本
wget --no-check-certificate
http://www.rabbitmq.com/releases/rabbitmq-server/v3.6.9/rabbitmq-server-3.6.9-1.el7.noarch.rpm
yum install rabbitmq-server-3.6.9-1.el7.noarch.rpm
service rabbitmq-server start
//启动 rabbitmq
/bin/systemctl status rabbitmq-server.service
systemctl enable rabbitmq-server.service
//开启 web 端
rabbitmq-plugins enable rabbitmq_management
//添加用户
rabbitmqctl add_user hzhw semf@2018
//添加 vhost
rabbitmqctl add_vhost semf
//设置标签
rabbitmqctl set_user_tags hzhw administrator
//设置权限
rabbitmqctl set_permissions -p semf hzhw ".*" ".*" ".*"
//（<conf> <write> <read>，正则表达式，'.*'表示所有权限）
```

二、 应用程序安装

2.1 SeMF 应用安装

```
useradd hzhw && passwd hzhw
//本地新建 hzhw 用户，切换到 user 权限下继续安装
su - hzhw
//创建项目路径
mkdir SeMF && cd SeMF
//克隆项目
git clone https://gitee.com/gy071089/SecurityManageFramwork.git
//进入项目目录,更改 setting 文件
cd SecurityManageFramwork/SeMF
vim settings.py
```

更改如下信息：

Email 相关为你邮箱的相关信息，参照注释进行修改

BROKER_URL 是你之前安装 rabbitmq 设置的账号和密码

```
#设置邮箱
EMAIL_HOST = 'smtp.163.com'                    #SMTP 地址
EMAIL_PORT = 25                                #SMTP 端口
```

```
EMAIL_HOST_USER = '***@163.com'      #我自己的邮箱
EMAIL_HOST_PASSWORD = '****'         #我的邮箱密码
EMAIL_SUBJECT_PREFIX = u'[SeMF]'     #为邮件 Subject-line 前缀,默认是'[django]'
EMAIL_USE_TLS = True                 #与 SMTP 服务器通信时，是否启动 TLS 链接(安全链接)。默认是 false
#管理员站点
SERVER_EMAIL = '***2007@qq.com'
DEFAULT_FROM_EMAIL = '安全管控平台<***2007@qq.com>'

#设置队列存储
BROKER_URL = 'amqp://hzhw:semf@2018@localhost/semf'   #设置与 rabbitmq 一致
```

2.2 更换 pip 源

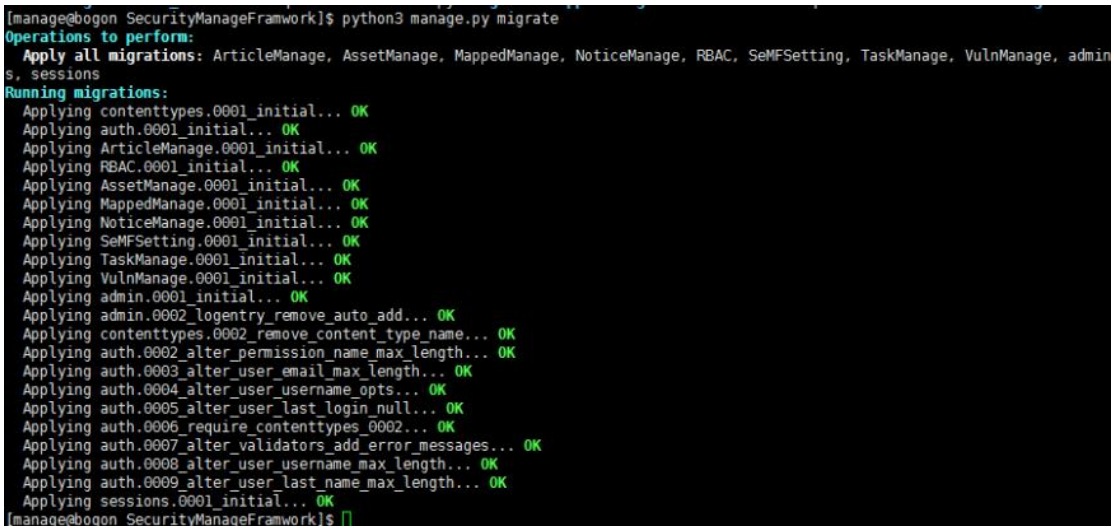
```
[root@localhost ~]# cd ~
[root@localhost ~]# mkdir .pip && cd .pip
```

```
[root@localhost .pip]# vim pip.conf
[global]  index-url=https://pypi.tuna.tsinghua.edu.cn/simple
trusted-host = pypi.tuna.tsinghua.edu.cn
```

2.3 安装依赖组件

注明：默认安装的 Django 2.2.11 有报错，在 requirements.txt 里改成 2.1.8

```
cd /home/hzhw/SeMF/SecurityManageFramwork
sudo python3 -m pip install -r requirements.txt          //安装依赖组件
python3 manage.py makemigrations                         //初始化数据表
python3 manage.py migrate                                //初始化数据库
```



2.4 创建超级账号

```
python3 manage.py createsuperuser
```

密码 Semf@2018

2.5 初始化数据

初始化漏洞库，需要半小时左右（CNNVD 漏洞数据，测试环境可不执行，执行过程中出现 fail 表明漏洞信息不完善或不重要，可忽略）

python3 cnvd_xml.py

初始化权限信息（主要包含菜单，权限以及管理员角色信息，该信息可在后台调整）

python3 initdata.py

```
[manage@bogon SecurityManageFramwork]$ python3 initdata.py
initassettype ok
initassettypeinfo ok
initrole ok
initrole ok
initsuperuser ok
initarticle ok
[manage@bogon SecurityManageFramwork]$
```

2.6 开启异步任务

执行 celery，先创建 sh 文件：

vim celery.sh

写入如下信息：

python3 -m celery -A SeMF worker -l info --autoscale=10,4 >> logs/celery.log 2>&1

&

echo 'Start celery for semf'

执行 celery

chmod u+x celery.sh

udo ./celery.sh

```
root@bogon SecurityManageFramwork]# ls
rticleManage  ChartManage  db.sqlite3   LICENSE     MappedManage  README.md         SeMFSetting  templates
ssetManage    cnvd_xml     files        logs        NoticeManage  requirements.txt  static       VulnManage
elery.sh      cnvd_xml.py  initdata.py  manage.py   RBAC          SeMF              TaskManage
root@bogon SecurityManageFramwork]# chmod u+x celery.sh
root@bogon SecurityManageFramwork]# ./celery.sh
tart celery for semf
root@bogon SecurityManageFramwork]#
```

查看 celery 情况：

ps -ef|grep celery

```
[root@bogon SecurityManageFramwork]# ps -ef|grep celery
root     19683     1  1 08:30 pts/0    00:00:01 python3 -m celery -A SeMF worker -l info --autoscale=10,4
root     19687 19683  0 08:30 pts/0    00:00:00 python3 -m celery -A SeMF worker -l info --autoscale=10,4
root     19688 19683  0 08:30 pts/0    00:00:00 python3 -m celery -A SeMF worker -l info --autoscale=10,4
root     19689 19683  0 08:30 pts/0    00:00:00 python3 -m celery -A SeMF worker -l info --autoscale=10,4
root     19690 19683  0 08:30 pts/0    00:00:00 python3 -m celery -A SeMF worker -l info --autoscale=10,4
root     19724 19593  0 08:33 pts/0    00:00:00 grep --color=auto celery
[root@bogon SecurityManageFramwork]#
```

关闭 celery :

ps -ef|grep celery|grep -v grep|awk '{print $2}'|xargs kill -9

2.7 运行应用程序

(正式环境部署，建议使用 nginx+uwsgi，部署方式自行百度)：

cd /home/hzhw/SeMF/SecurityManageFramwork/SeMF

python3 manage.py runserver 0.0.0.0:8000

2.8 开机自启

vim /home/hzhw/SeMF/SecurityManageFramwork/celery2.sh

写入如下信息：

```
#! /bin/bash
cd /home/hzhw/SeMF/SecurityManageFramwork
sh celery.sh
```

****

```
vim /etc/rc.d/rc.local
```

写入如下信息：

```
nohup python3 /home/hzhw/SeMF/SecurityManageFramwork/manage.py runserver 0.0.0.0:8000 > /tmp/log.out 2>&1 &
sh /home/hzhw/SeMF/SecurityManageFramwork/celery2.sh
```

```
chmod +x /etc/rc.d/rc.local
```

三、 SeMF 和 Nessus 对接

3.1 Nessus 扫描器设置

【Kali 上 Nessus 的安装和破解见第四步】

这里我们以 Nessus 为例，

访问 ip:port 登录系统，账号为 2.4 步创建的用户

登录后访问 ip:port/semf/

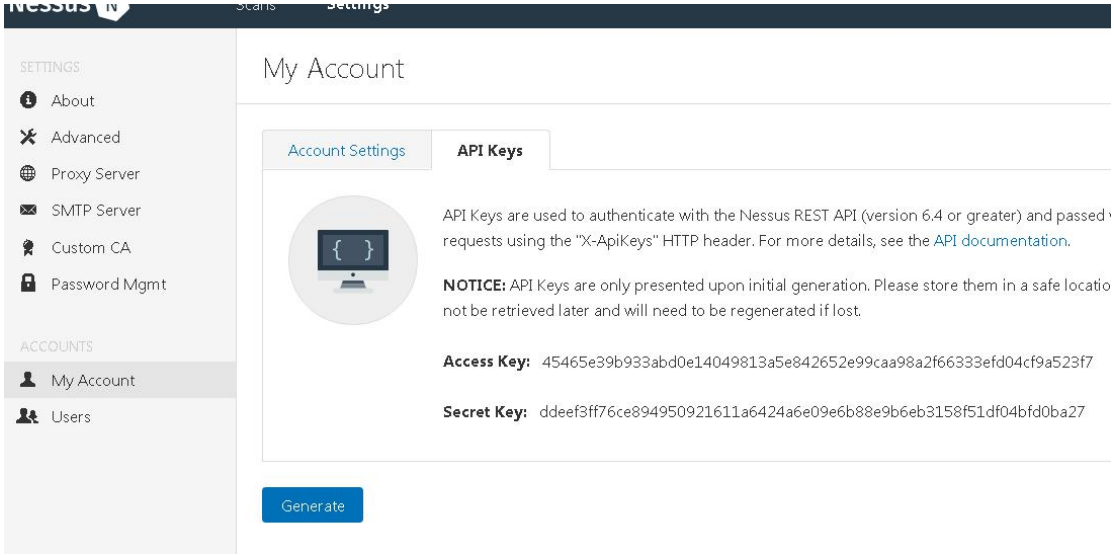
注意节点名称和 Nessus 规则中的名称统一，另外节点 URL 后不要加斜线，否则 django 会报错

3.2 在 Nessus 管理面板中获取 API key

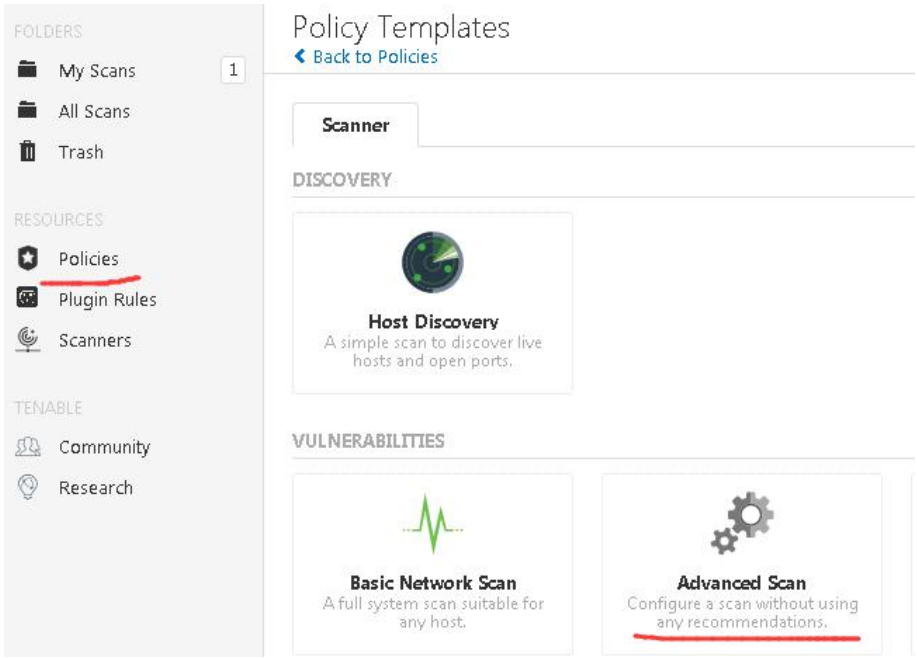
登录 nessus，选择 settings/My Account /APIKeys

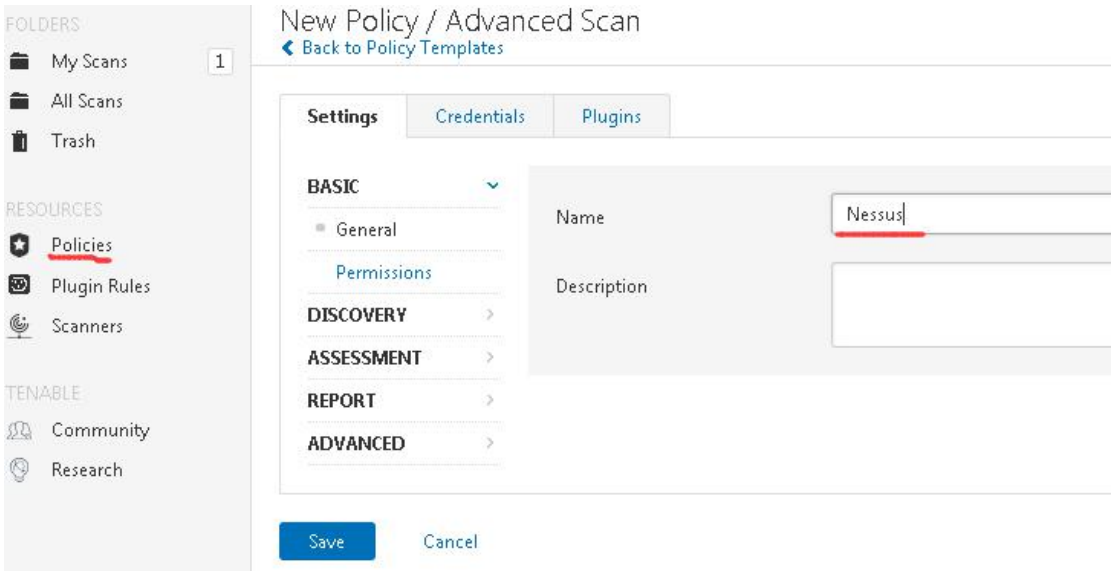
点击生成，即可获取一对密钥 key（key 每生成一次都会变更），复制到 SeMF 的 Scanner 中并保存

请注意需要选择 Nessus 扫描器可扫描的资产类型。



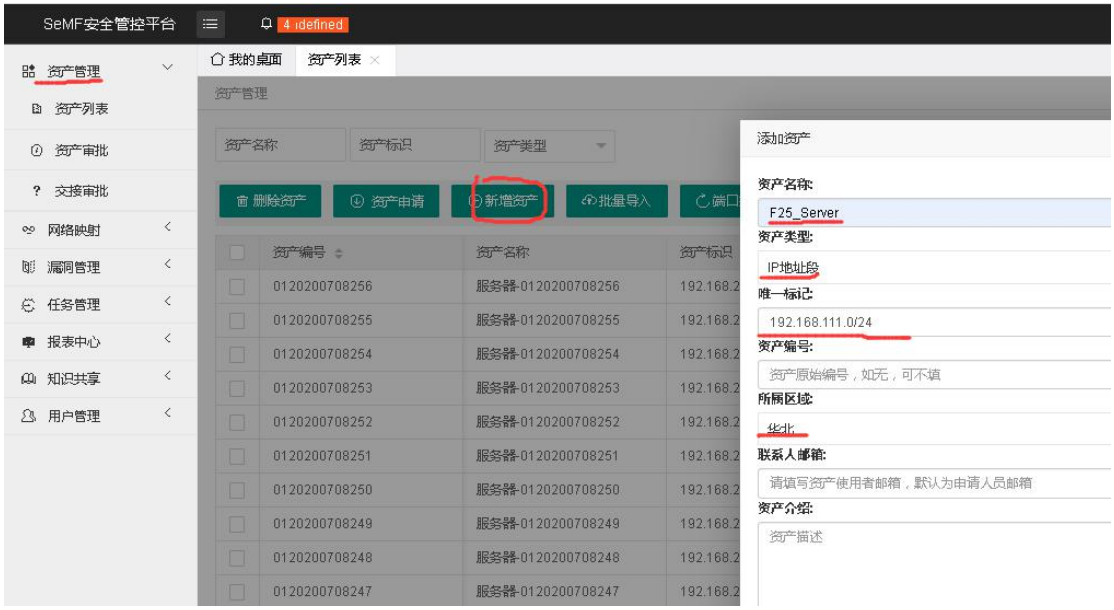
新建扫描策略

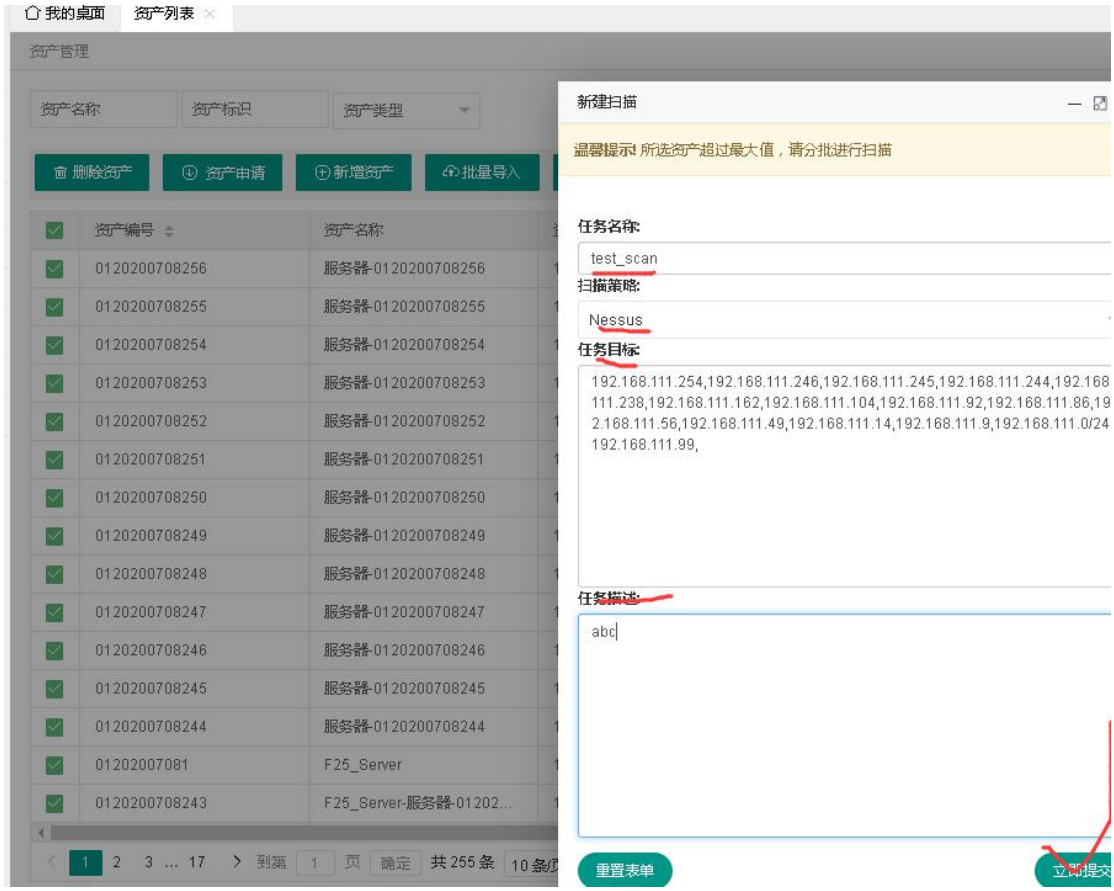




3.3 SeMF 添加资产及开启扫描

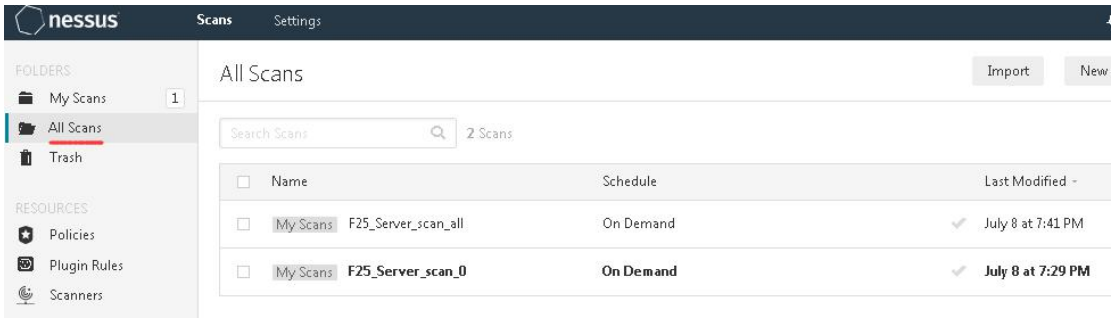
新增资产(IP 网段)，进行“资产发现”，然后“安全扫描”



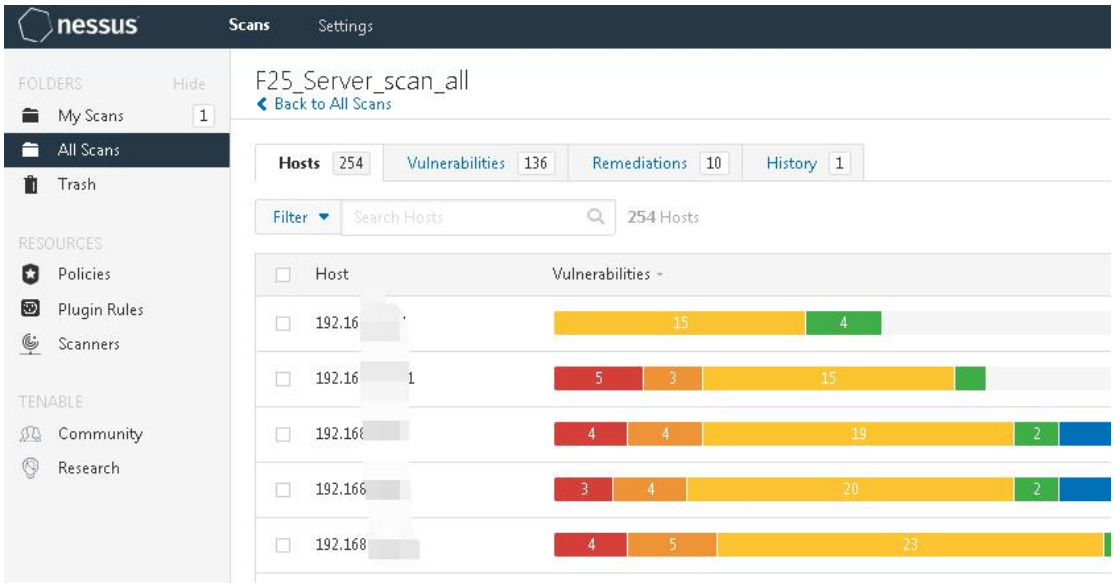


到“任务管理”中进行处理，后台交给 Nessus 扫描

可以在 Nessus 中看到推送过来的扫描任务，漏洞情况也会展示到 SeMF 的报表中心



扫描任务结束后，可以查看扫描结果



3.4 安装 Nginx 配置 ssl 证书

3.4.1 安装 nginx

```
yum -y install epel-release
yum -y install nginx
systemctl restart nginx.service
systemctl enable nginx
```

既然是安全系统，那就使用 https 吧，最后再把防火墙开起来

3.4.2 生成 ssl 证书

1、首先，进入你想创建证书和私钥的目录，例如：
cd /etc/nginx/
2、创建服务器私钥，命令会让你输入一个口令：
openssl genrsa -des3 -out server.key 1024
3、创建签名请求的证书（CSR）：
openssl req -new -key server.key -out server.csr
4、在加载 SSL 支持的 Nginx 并使用上述私钥时除去必须的口令：
cp server.key server.key.org
openssl rsa -in server.key.org -out server.key
5、最后标记证书使用上述私钥和 CSR：
openssl x509 -req -days 3650 -in server.csr -signkey server.key -out server.crt

3.4.3 配置 Nginx

```
cp nginx.conf bak_nginx.conf
cp server.crt server.key conf.d/
```

```
vim /etc/nginx/conf.d/default.conf
```

// 先备注掉 nginx.conf 中的 80 端口，使用 default.conf 配置文件。 替换下面 IP 为自己 SeMF 的 IP

```
server{
      listen      80;
      server_name  localhost;
      return 301 https://192.168.2.219$request_uri;
#      rewrite ^(.*)$ https://localhost/ permanent;
}

server {
   listen      443;
   server_name  localhost;

   ssl              on;
   ssl_certificate      /etc/nginx/conf.d/server.crt;
   ssl_certificate_key  /etc/nginx/conf.d/server.key;

   ssl_session_timeout  5m;

   location / {
      #root   html;
      #index  testssl.html index.html index.htm;
    proxy_redirect off;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_pass http://localhost:8000/;
```

```
  }
}
```

四、 在 Kali Linux 上安装 Nessus

4.1 下载 Nessus 及注册获取激活码

官网下载地址：https://www.tenable.com/downloads/nessus
注册获取激活码：https://zh-cn.tenable.com/products/nessus/nessus-essentials

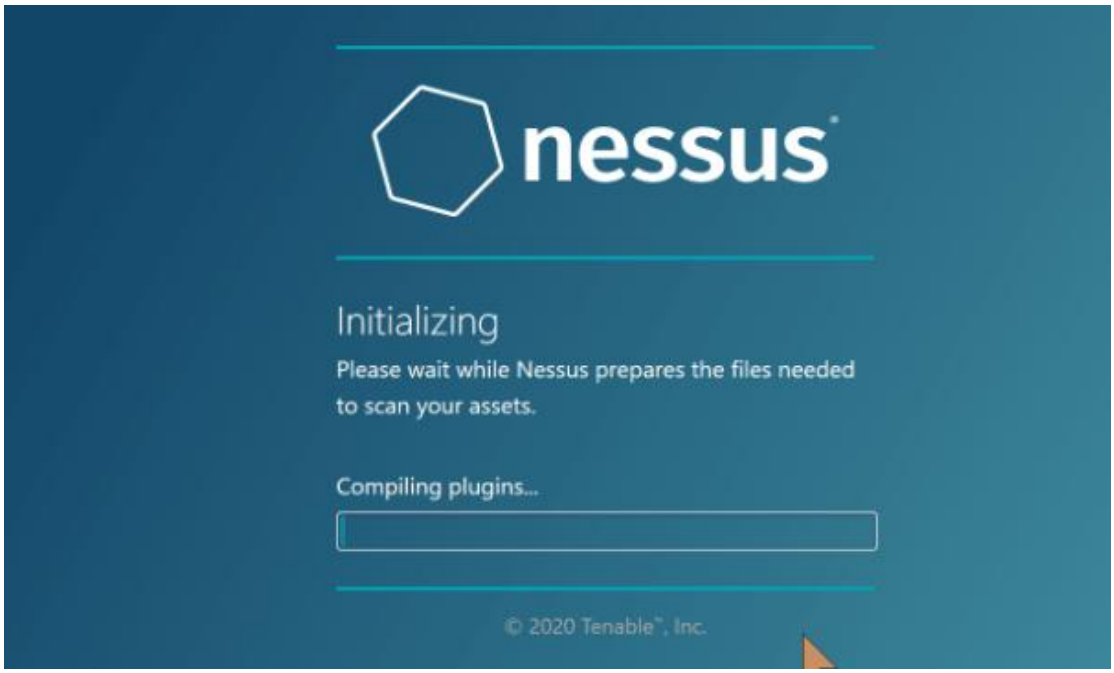
4.2 在 Kali 中安装 Nessus

```
dpkg -i Nessus-8.10.0-debian6_amd64.deb
service nessusd start
```

在浏览器中访问 https://localhost:8834，
初始化扫描器，选择 Managed Scanner→Managed by Tenable.sc，点击 Continue。

现在会要求新建账号，自己记住账号密码。

等待初始化完成，这一步会比较慢，完成后登陆进去是没有扫描界面的。



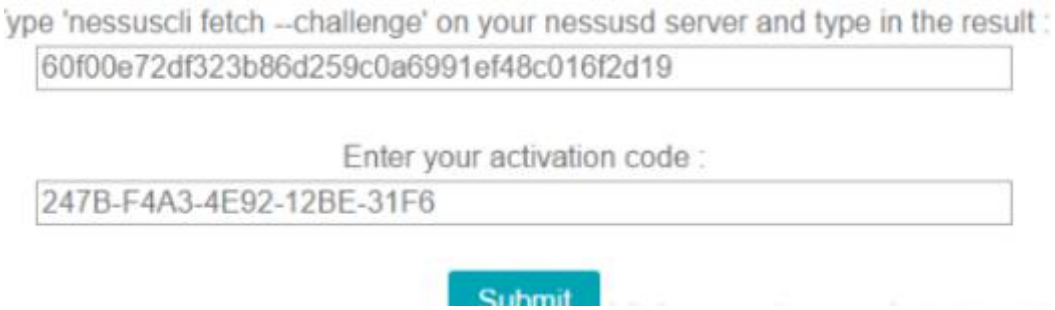
4.3 获取 Nessus 的插件包

kali 中默认把 nessus 安装在/opt 目录下，现在进入目录，执行以下操作，复制并记录 challenge code：

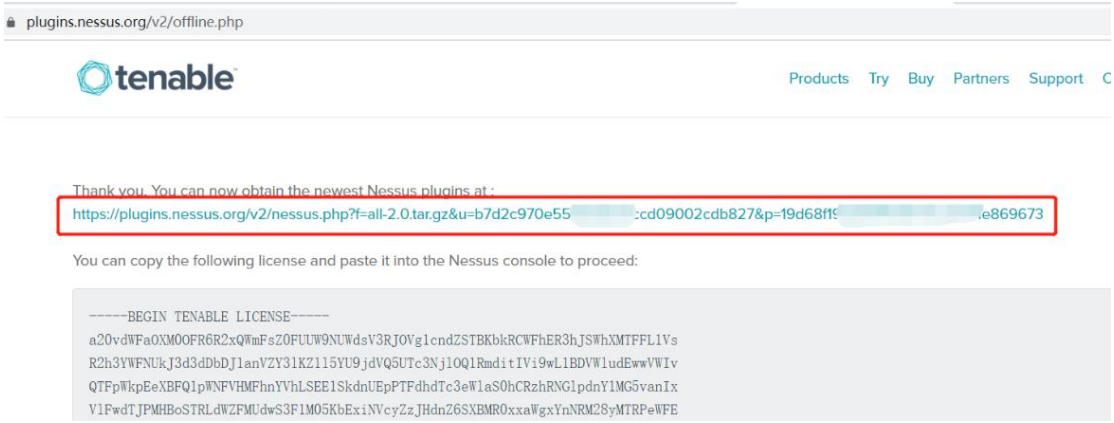
```
/opt/nessus/sbin/nessuscli fetch --challenge
```

```
root@kali:/opt/nessus# /opt/nessus/sbin/nessuscli fetch --challenge

Challenge code: b6789570934484519 93c6b447a0ba826580811f2

You can copy the challenge code above and paste it alongside your
Activation Code at:
https://plugins.nessus.org/v2/offline.php
```

访问上面输出的网址 https://plugins.nessus.org/v2/offline.php，输入 Challenge code 和收到邮件中的激活码：



注册成功后网页返回更新包的下载链接，下载速度比较慢，可以用迅雷下载（无须解压缩，直接解压会提示文件损坏，请忽略）：



4.4 更新插件

插件包下载完成后，执行更新操作：

```
/opt/nessus/sbin/nessuscli update all-2.0.tar.gz
```

```
root@kali:/opt/nessus/nessus-update# /opt/nessus/sbin/nessuscli update all-2.0.tar.gz

[info] Copying templates version 202003232053 to /opt/nessus/var/nessus/templates/tmp
[info] Finished copying templates.
[info] Moved new templates with version 202003232053 from plugins dir.
 * Update successful.  The changes will be automatically processed by Nessus.
```

到现在为止，nessus 安装完成，但只支持 16 个 IP。

4.5 破解 Nessus

修改两个文件，没有的话创建一下，再改成下面的内容。

```
export PLUGIN_SET="202004281428"   //设置一下本次安装的时间
```

```
cat >/opt/nessus/lib/nessus/plugins/plugin_feed_info.inc<<EOF
PLUGIN_SET = ${PLUGIN_SET};
PLUGIN_FEED = "ProfessionalFeed (Direct)";
PLUGIN_FEED_TRANSPORT = "Tenable Network Security Lightning";
EOF
```

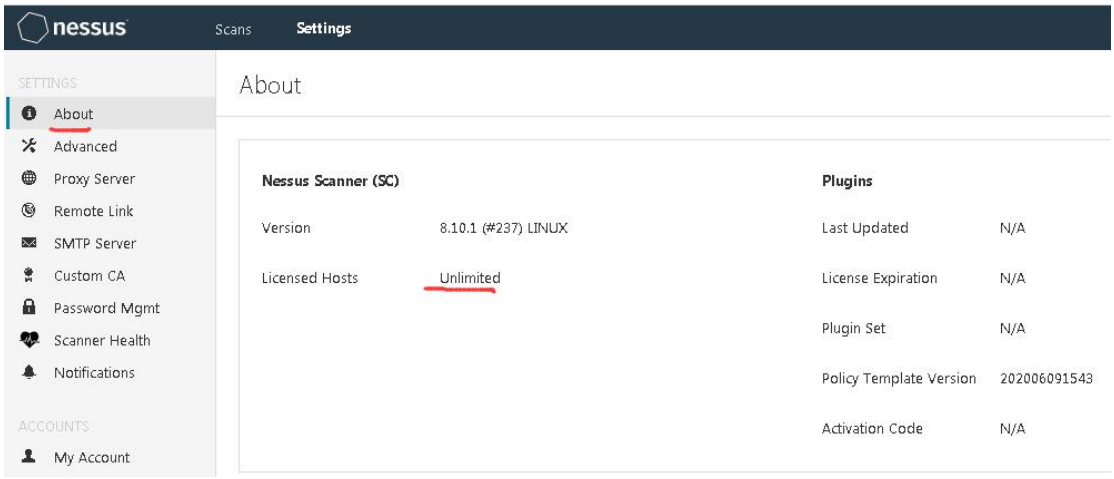
```
cat >/opt/nessus/var/nessus/plugin_feed_info.inc<<EOF
PLUGIN_SET = ${PLUGIN_SET};
PLUGIN_FEED = "ProfessionalFeed (Direct)";
PLUGIN_FEED_TRANSPORT = "Tenable Network Security Lightning";
EOF
```

```
root@kali:/opt/nessus/nessus-update# cat /opt/nessus/lib/nessus/plugins/plugin_feed_info.inc
PLUGIN_SET = 202004281428;
PLUGIN_FEED = "ProfessionalFeed (Direct)";

PLUGIN_FEED_TRANSPORT = "Tenable Network Security Lightning";
root@kali:/opt/nessus/nessus-update# cat /opt/nessus/var/nessus/plugin_feed_info.inc
PLUGIN_SET = 202004281428;
PLUGIN_FEED = "ProfessionalFeed (Direct)";

PLUGIN_FEED_TRANSPORT = "Tenable Network Security Lightning";
root@kali:/opt/nessus/nessus-update#
```

重启 nessus

```
service nessusd restart
```

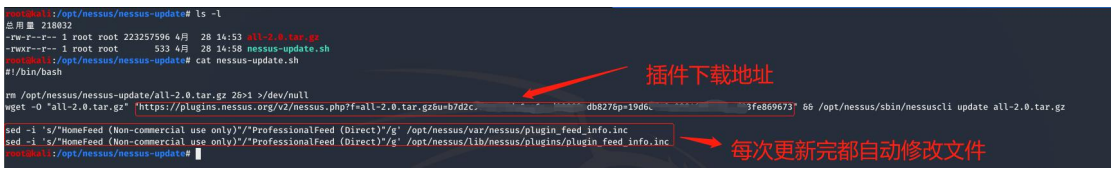
修改上面两个文件是用来把 16 个 IP 的家庭版转化成无限制的专业版的



注：每次更新完后，上述两个文件都会变回家庭版的配置（因为我们是通过下载家庭版的插件包来进行离线更新的），所以原本是破解的，一更新就又变限制版了，需要重新改成上面的内容。

进行定期自动更新脚本

自己在安装目录下新建一个文件夹 nessus-update，以后下载的插件包都放在这里，自动更新脚本也放这里。



设置 crontab 计划任务，每个月跑一次更新。