SeMF 安装指南

1   基础环境安装（这里已 centos7.5 作为示例）

1.1   Python 安装

安装相关软件

yum   install   zlib-devel   bzip2-devel   openssl-devel   ncurses-devel   sqlite-devel

readline-devel tk-devel gcc make

```
[manage@bogon ~]$ yum install zlib-devel bzip2-devel openssl-devel ncurses-devel sqlite-devel readline-devel tk-devel gcc make
Loaded plugins: fastestmirror
You need to be root to perform this command.
[manage@bogon ~]$ su root
Password:
[root@bogon manage]# yum install zlib-devel bzip2-devel openssl-devel ncurses-devel sqlite-devel readline-devel tk-devel gcc make
Loaded plugins: fastestmirror
base                                                                        | 3.6 kB  00:00:00
extras                                                                      | 3.4 kB  00:00:00
updates                                                                     | 3.4 kB  00:00:00
updates/7/x86_64/primary_db      FAILED
http://mirrors.shu.edu.cn/centos/7.5.1804/updates/x86_64/repodata/d9936a2e437496346fb75c5c08f85346771ba1644a8714f472e164e1d4c7ccdf-primary.sqlite.b
z2: [Errno 14] curl#7 - "Failed connect to mirrors.shu.edu.cn:80; Connection refused"
Trying other mirror.
(1/4): extras/7/x86_64/primary_db                                           | 147 kB  00:00:00
(2/4): base/7/x86_64/group_gz                                               | 166 kB  00:00:00
(3/4): updates/7/x86_64/primary_db                                          | 2.0 MB  00:00:00
(4/4): base/7/x86_64/primary_db                                             | 5.9 MB  00:00:00
Determining fastest mirrors
 * base: mirrors.aliyun.com
 * extras: mirrors.tuna.tsinghua.edu.cn
 * updates: mirrors.tuna.tsinghua.edu.cn
Package 1:make-3.82-23.el7.x86_64 already installed and latest version
Resolving Dependencies
```

切换 tmp 路径（免去删除过程文件，重启自动删除）

下载 python3 安装包，这里我选择的是 3.6.5

yun install wget -y

wget https://www.python.org/ftp/python/3.6.5/Python-3.6.5.tar.xz

```
[root@bogon manage]# cd /tmp
[root@bogon tmp]# wget https://www.python.org/ftp/python/3.6.5/Python-3.6.5.tar.xz
bash: wget: command not found
[root@bogon tmp]# yum install wget
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirrors.aliyun.com
 * extras: mirrors.tuna.tsinghua.edu.cn
 * updates: mirrors.tuna.tsinghua.edu.cn
Resolving Dependencies
--> Running transaction check
---> Package wget.x86_64 0:1.14-15.el7_4.1 will be installed
--> Finished Dependency Resolution
```

解压文件包

tar -xvJf Python-3.6.5.tar.xz

cd Python-3.6.5

编译安装

./configure prefix=/usr/local/python3

make && make install

```
[root@bogon tmp]# ls
Python-3.6.5          systemd-private-7a01e384632f49e0bd4443f5f508c185-vgauthd.service-hhxbqz
Python-3.6.5.tar.xz   systemd-private-7a01e384632f49e0bd4443f5f508c185-vmtoolsd.service-Pvd9y1
[root@bogon tmp]# cd Python-3.6.5
[root@bogon Python-3.6.5]# ls
aclocal.m4    config.sub   configure.ac  Grammar  install-sh   LICENSE   Makefile.pre.in   Modules   Parser   PCbuild   pyconfig.h.in   README.rst   Tools
config.guess  configure    Doc           Include  Lib          Mac       Misc              Objects   PC       Programs  Python          setup.py
[root@bogon Python-3.6.5]# ./configure prefix=/usr/local/python3
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for python3.6... no
checking for python3... no
checking for python... python
checking for --enable-universalsdk... no
checking for --with-universal-archs... no
```

创建软链接

ln -s /usr/local/python3/bin/python3 /usr/bin/python3

```
[root@bogon Python-3.6.5]# ln -s /usr/local/python3/bin/python3 /usr/bin/python3
[root@bogon Python-3.6.5]# python
Python 2.7.5 (default, Aug  4 2017, 00:39:18)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-16)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()
[root@bogon Python-3.6.5]# python3
Python 3.6.5 (default, Jun  4 2018, 07:04:43)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-28)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

1.2 Rabitmq 安装

下载 erlang

wget

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

```
[root@bogon Python-3.6.5]# cd ..
[root@bogon tmp]# wget http://www.rabbitmq.com/releases/erlang/erlang-19.0.4-1.el7.centos.x86_64.rpm
--2018-06-04 07:10:49--  http://www.rabbitmq.com/releases/erlang/erlang-19.0.4-1.el7.centos.x86_64.rpm
Resolving www.rabbitmq.com (www.rabbitmq.com)... 104.20.63.197, 104.20.62.197, 2400:cb00:2048:1::6814:3fc5, ...
Connecting to www.rabbitmq.com (www.rabbitmq.com)|104.20.63.197|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 18580960 (18M) [application/x-redhat-package-manager]
Saving to: 'erlang-19.0.4-1.el7.centos.x86_64.rpm'

100%[===================================================================================>] 18,580,960  9.66MB/s   in 1.8s

2018-06-04 07:10:51 (9.66 MB/s) - 'erlang-19.0.4-1.el7.centos.x86_64.rpm' saved [18580960/18580960]

[root@bogon tmp]#
```

安装 rpm

rpm -ivh erlang-19.0.4-1.el7.centos.x86_64.rpm

yum -y install erlang

```
[root@bogon tmp]# rpm -ivh erlang-19.0.4-1.el7.centos.x86_64.rpm
Preparing...                          ################################# [100%]
Updating / installing...
   1:erlang-19.0.4-1.el7.centos       ################################# [100%]
[root@bogon tmp]#
```

查看版本：

erl –version

```
[root@bogon tmp]# erl -version
Erlang (SMP,ASYNC_THREADS,HIPE) (BEAM) emulator version 8.0.3
[root@bogon tmp]#
```

下载 rabbitmq

wget

http://www.rabbitmq.com/releases/rabbitmq-server/v3.6.9/rabbitmq-server-3.6.9-1.el7.noarch.rpm

```
[root@bogon tmp]# wget http://www.rabbitmq.com/releases/rabbitmq-server/v3.6.9/rabbitmq-server-3.6.9-1.el7.noarch.rpm
--2018-06-04 07:16:15--  http://www.rabbitmq.com/releases/rabbitmq-server/v3.6.9/rabbitmq-server-3.6.9-1.el7.noarch.rpm
Resolving www.rabbitmq.com (www.rabbitmq.com)... 104.20.63.197, 104.20.62.197, 2400:cb00:2048:1::6814:3ec5, ...
Connecting to www.rabbitmq.com (www.rabbitmq.com)|104.20.63.197|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4918617 (4.7M) [application/x-redhat-package-manager]
Saving to: 'rabbitmq-server-3.6.9-1.el7.noarch.rpm'

100%[===================================================================================>] 4,918,617   1.83M

2018-06-04 07:16:18 (1.83 MB/s) - 'rabbitmq-server-3.6.9-1.el7.noarch.rpm' saved [4918617/4918617]

[root@bogon tmp]#
```

安装 rabitmq

yum install rabbitmq-server-3.6.9-1.el7.noarch.rpm

```
[root@bogon tmp]# yum install rabbitmq-server-3.6.9-1.el7.noarch.rpm
Loaded plugins: fastestmirror
Examining rabbitmq-server-3.6.9-1.el7.noarch.rpm: rabbitmq-server-3.6.9-1.el7.noarch
Marking rabbitmq-server-3.6.9-1.el7.noarch.rpm to be installed
Resolving Dependencies
--> Running transaction check
---> Package rabbitmq-server.noarch 0:3.6.9-1.el7 will be installed
--> Processing Dependency: socat for package: rabbitmq-server-3.6.9-1.el7.noarch
Loading mirror speeds from cached hostfile
 * base: mirrors.aliyun.com
 * extras: mirrors.tuna.tsinghua.edu.cn
 * updates: mirrors.tuna.tsinghua.edu.cn
--> Running transaction check
---> Package socat.x86_64 0:1.7.3.2-2.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved
```

启动 rabbitmq

service rabbitmq-server start

开启 web 端

rabbitmq-plugins enable rabbitmq_management

```
[root@bogon tmp]# service rabbitmq-server start
Redirecting to /bin/systemctl start rabbitmq-server.service
[root@bogon tmp]# rabbitmq-plugins enable rabbitmq_management
The following plugins have been enabled:
  amqp_client
  cowlib
  cowboy
  rabbitmq_web_dispatch
  rabbitmq_management_agent
  rabbitmq_management

Applying plugin configuration to rabbit@bogon... started 6 plugins.
[root@bogon tmp]#
```

添加用户

rabbitmqctl add_user <user> pwd

添加 vhost

rabbitmqctl add_vhost vhost

设置标签

rabbitmqctl set_user_tags User Tag[administrator，monitoring，policymaker，management]

设置权限

rabbitmqctl set_permissions -p vhost <user> ".*" ".*" ".*" （<conf> <write> <read>，正则表达式，'.*'表示所有权限）

```
Applying plugin configuration to rabbit@bogon... started 6 plugins.
[root@bogon tmp]# rabbitmqctl add_user test test23
Creating user "test" ...
[root@bogon tmp]# rabbitmqctl add_vhost semf
Creating vhost "semf" ...
[root@bogon tmp]# rabbitmqctl  set_user_tags test administrator
Setting tags for user "test" to [administrator] ...
[root@bogon tmp]# rabbitmqctl set_permissions -p semf test ".*" ".*" ".*"
Setting permissions for user "test" in vhost "semf" ...
[root@bogon tmp]#
```

1.3 安装 nmap

yum install nmap

1.4 防火墙设置

开放端口

firewall-cmd --zone=public --add-port=5672/tcp --permanent   //rabbitmq api 端口

firewall-cmd --zone=public --add-port=15672/tcp --permanent //rabbitmq web 端口

firewall-cmd --reload   //刷新防火墙规则

firewall-cmd --zone=public --list-ports //查看所有开放端口

```
[root@bogon tmp]# firewall-cmd --zone=public --add-port=5672/tcp --permanent

success
[root@bogon tmp]#
[root@bogon tmp]# firewall-cmd --zone=public --add-port=15672/tcp --permanent
success
[root@bogon tmp]# firewall-cmd --reload
success
[root@bogon tmp]# firewall-cmd --zone=public --list-ports
5672/tcp 15672/tcp
[root@bogon tmp]#
```

systemctl stop firewalld.service   //关闭防火墙，生产环境不建议使用

2   应用程序安装

2.1   应用安装

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   SeMF

vi setting.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，详情参考 1.2 章

数据库设置

这里的话，需要注意，默认是 sqlite，也可以选用 mysql，注意使用 mysql 创建数据库时，名称要与 NAME 一致，且字符类型为 utf-8

```
# Database
# https://docs.djangoproject.com/en/2.0/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}

'''
#设置mysql数据配置信息
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'SeMF',
        'USER': 'username',
        'PASSWORD': 'password',
        'HOST': 'XX.XX.XX.XX',
        'PORT': '3306',
        'OPTIONS':{
                  'init_command':"SET sql_mode='STRICT_TRANS_TABLES' ",
                  'charset':'utf8',}
    }
}
'''
```

设置完成后，切换到项目根目录,

使用管理员权限安装依赖组件

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

创建超级账号

python3 manage.py createsuperuser

初始化漏洞库（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]$ []
```

开启异步任务，执行 celery：

创建 sh 文件

vi 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

sudo ./celery.sh

```
[root@bogon SecurityManageFramwork]# ls
ArticleManage   ChartManage   db.sqlite3   LICENSE        MappedManage   README.md        SeMFSetting   templates
AssetManage     cnvd_xml      files        logs           NoticeManage   requirements.txt static        VulnManage
celery.sh       cnvd_xml.py   initdata.py  manage.py      RBAC           SeMF             TaskManage
[root@bogon SecurityManageFramwork]# chmod u+x celery.sh
[root@bogon SecurityManageFramwork]# ./celery.sh
Start 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

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)：

python3 manage.py   runserver 0.0.0.0:8000

```
[root@bogon SecurityManageFramwork]# python3 manage.py  runserver 0.0.0.0:8080
Performing system checks...

System check identified no issues (0 silenced).
June 04, 2018 - 08:38:19
Django version 2.0.4, using settings 'SeMF.settings'
Starting development server at http://0.0.0.0:8080/
Quit the server with CONTROL-C.
[04/Jun/2018 08:38:24] "GET / HTTP/1.1" 302 0
[04/Jun/2018 08:38:24] "GET /robots.txt HTTP/1.1" 302 0
[04/Jun/2018 08:38:24] "GET /view/ HTTP/1.1" 200 2189
[04/Jun/2018 08:38:24] "GET /view/ HTTP/1.1" 200 2189
[04/Jun/2018 08:38:24] "GET /static/css/xadmin.css HTTP/1.1" 200 10224
[04/Jun/2018 08:38:24] "GET /static/css/font.css HTTP/1.1" 200 505
[04/Jun/2018 08:38:25] "GET /static/js/xadmin.js HTTP/1.1" 200 5668
[04/Jun/2018 08:38:25] "GET /static/lib/layui/layui.js HTTP/1.1" 200 6542
[04/Jun/2018 08:38:25] "GET /static/jquery/jquery-3.3.1.min.js HTTP/1.1" 200 86927
[04/Jun/2018 08:38:25] "GET /static/css/bootstrap.min.css HTTP/1.1" 200 121200
[04/Jun/2018 08:38:25] "GET /static/lib/layui/css/layui.css HTTP/1.1" 200 60227
[04/Jun/2018 08:38:25] "GET /static/js/bootstrap.min.js HTTP/1.1" 200 37045
[04/Jun/2018 08:38:25] "GET /static/images/bg.png HTTP/1.1" 200 28211
[04/Jun/2018 08:38:25] "GET /static/css/xadmin.css HTTP/1.1" 304 0
[04/Jun/2018 08:38:25] "GET /static/lib/layui/lay/modules/form.js HTTP/1.1" 200 7925
[04/Jun/2018 08:38:26] "GET /static/images/aiwrap.png HTTP/1.1" 200 3032
[04/Jun/2018 08:38:26] "GET /static/lib/layui/lay/modules/layer.js HTTP/1.1" 200 22063
[04/Jun/2018 08:38:26] "GET /static/lib/layui/css/modules/layer/default/layer.css?v=3.1.1 HTTP/1.1" 200 14425
[04/Jun/2018 08:38:26] "GET /static/lib/layui/lay/modules/element.js HTTP/1.1" 200 7264
[04/Jun/2018 08:38:26] "GET /favicon.ico HTTP/1.1" 302 0
[04/Jun/2018 08:38:26] "GET /favicon.ico HTTP/1.1" 302 0
[04/Jun/2018 08:38:26] "GET /view/ HTTP/1.1" 200 2189
[04/Jun/2018 08:38:26] "GET /view/ HTTP/1.1" 200 2189
```

2.2 扫描器设置

这里我们已 Nessus 为例，目前只支持 nessus6：

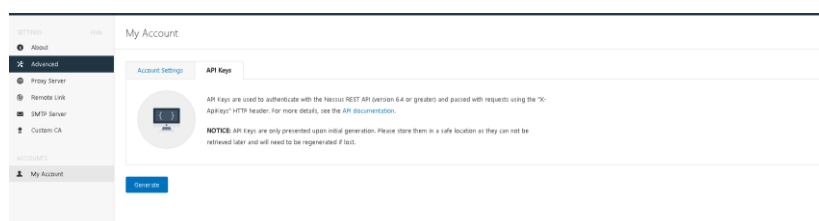访问 ip:port 登录系统，账号为 2.1 创建的用户
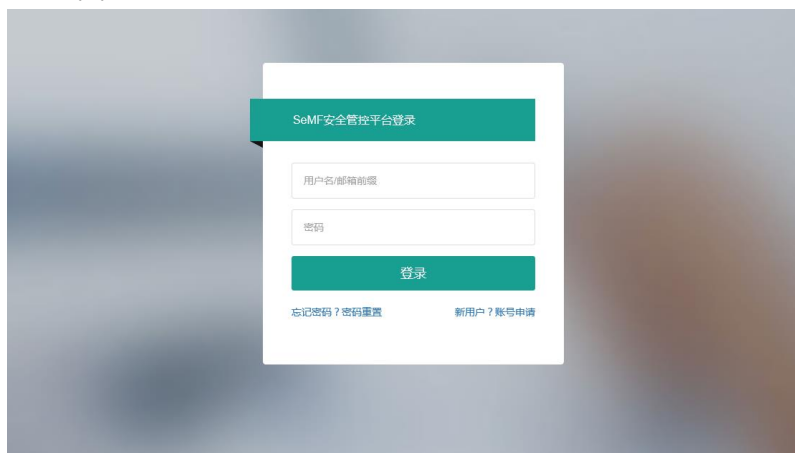
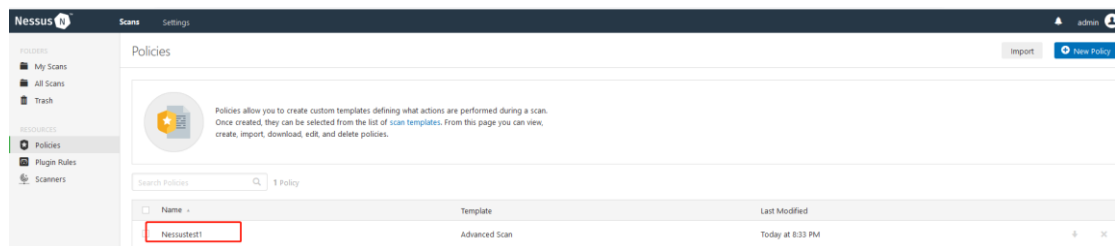登录后访问　ip:port/semf/

点击新增，选择 Nessus 相关信息

API key 的获取

登录 nessus，选择 settings/My Account /APIKeys

点击生成，即可获取一对密钥 key，复制到 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

创建完扫描策略后 ， 访问/semf/页面，添加扫描策略

2.3 AWVS 扫描器部署

AWVS 关联方式与 Nessus 相同，需要注意的是，AWVS apikey 只有一各 key，Nessus 有两。

由于 AWVS 扫描器的策略有定义好的，所以策略需要需要添加策略编号，添加参考如下：

```
11111111-1111-1111-1111-111111111112    High Risk Vulnerabilities
11111111-1111-1111-1111-111111111115    Weak Passwords
11111111-1111-1111-1111-111111111117    Crawl Only
11111111-1111-1111-1111-111111111116    CSS
11111111-1111-1111-1111-111111111113    SQL Injection Vulnerabilities
11111111-1111-1111-1111-111111111118    quick_profile_2 0
11111111-1111-1111-1111-111111111114    quick_profile_1 0
11111111-1111-1111-1111-111111111111    Full Scan   1
```