哈尔滨工业大学计算机学院

《网络攻击与防御》

实验报告

计算机科学与技术学院

计算机系网络教研室制

|  |  |
| --- | --- |
| 课程名称： | 信息安全概论 |
| 实验名称： | XSS跨站脚本攻击原理与实践 |
| 指导教师： | 王彦 |
| 学生姓名： | 冯开来 |
| 组 号： | 1903010 |
| 实验日期： | 21-6-20 |
| 实验地点： | 正心11 |
| 实验成绩： |  |

实验报告撰写要求

实验操作是教学过程中理论联系实际的重要环节，而实验报告的撰写又是知识系统化的吸收和升华过程，因此，实验报告应该体现完整性、规范性、正确性、有效性。现将实验报告撰写的有关内容说明如下：

1、 实验报告模板为电子版。

2、 下载统一的实验报告模板，学生自行完成撰写和打印。报告的首页包含本次实验的一般信息：

*  组 号：例如：2-5 表示第二班第5组。
*  实验日期：例如：05-10-06 表示本次实验日期。(年-月-日)……
*  实验编号：例如：No.1 表示第一个实验。
*  实验时间：例如：2学时 表示本次实验所用的时间。

实验报告正文部分，从六个方面（目的、内容、步骤等）反映本次实验的要点、要求以及完成过程等情况。模板已为实验报告正文设定统一格式，学生只需在相应项内填充即可。续页不再需要包含首页中的实验一般信息。

3、 实验报告正文部分具体要求如下：

一、实验目的

本次实验所涉及并要求掌握的知识点。

二、实验环境

实验所使用的设备名称及规格，网络管理工具简介、版本等。

三、实验内容与实验要求

实验内容、原理分析及具体实验要求。

四、实验过程与分析

根据具体实验，记录、整理相应命令、运行结果等，包括截图和文字说明。

详细记录在实验过程中发生的故障和问题，并进行故障分析，说明故障排除的过程及方法。

五、实验结果总结

对实验结果进行分析，完成思考题目，总结实验的心得体会，并提出实验的改进意见。

六、附录

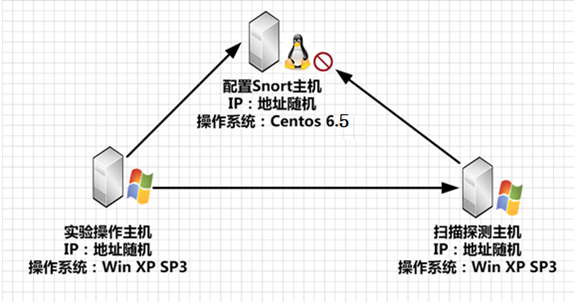
一、实验目的

*本次实验所涉及并要求掌握的知识点。*

通过该实验可以加深理解Snort的系统架构以及工作原理，掌握Snort与Iptables联动的实现方法。

二、实验环境

*实验所使用的设备名称及规格，网络管理工具简介、版本等。*

**

服务器:snort-host( Centos6.5)，IP地址: 10.1.1.12

Snort版本： 2.9.7.6(最新) Guardian版本：1.7(最新)

操作主机:host(WinXp), IP地址: 随机

测试主机:test(WinXp), IP地址: 随机

下载路径：http://tools.hetianlab.com/tools/X-Scan-v3.3-cn.rar

下载路径：http://tools.hetianlab.com/tools/Xshell.rar

三、实验内容与实验要求

*实验内容、原理分析及具体实验要求。*

实验步骤一

Snort与guardian的安装和配置

1.  操作指导（实验过程中操作命令综述）

      Vim编辑器是linux下的一款相当于记事本的编辑器。

      命令：vi 某文件名即进入编辑状态，输入:i即可插入或删除文字

      按下ESC可退出编辑。

      shift键+冒号(:) 输入q! 即不保存退出输入x即可保存退出

2.  Snort安装与配置

/\*\*\*\*\*\*\*\*\*\*更新系统、安装snort依赖包、下载snort、规则库\*\*\*\*\*\*\*\*\*/

SSH登录centos6.5主机

      用户名：root

      password:123456

查看网卡

      ifconfig（如果网卡没启动）

      ifup eth0

更新系统

      yum -y update

安装snort依赖包

      yum -y install pcre pcre-devel gcc gcc-c++ zlib zlib-devel libpcap libpcap-devel make autoconf flex byacc bison libxml2-devel wget tcpdump

下载Snort、Snort规则库、DAQ 、libdnet、guardian

(外网下载十分缓慢，因此安装包已经提前下载到实验环境服务器（http://tools.hetianlab.com/tools/）中)

下载 xshell 登陆linux

下载地址 ：http://tools.hetianlab.com/tools/Xshell.rar

cd /usr/local/src/

wget http://tools.hetianlab.com/tools/snort+IPtables.tar.gz

tar zxvf snort+IPtables.tar.gz

安装libdnet

cd /usr/local/src/snort+IPtables

tar zxvf libdnet-1.12.tgz

cd libdnet-1.12

./configure --with-pic

make && make install

cd /usr/local/lib/

ldconfig -v /usr/local/lib

安装DAQ

cd /usr/local/src/snort+IPtables

tar zxvf daq-2.0.6.tar.gz

cd daq-2.0.6

./configure

make && make install

cd /usr/local/lib/

ldconfig -v /usr/local/lib

安装snort

cd /usr/local/src/snort+IPtables/

tar zxvf snort-2.9.7.6.tar.gz

cd snort-2.9.7.6

./configure -enable-sourcefire

make && make install

cd /usr/local/lib/

ldconfig -v /usr/local/lib

Snort配置

创建snort系统必要的文件夹，导入规则库，建立黑名单和白名单文件

mkdir /etc/snort

mkdir /var/log/snort

cd /etc/snort

cp /usr/local/src/snort+IPtables/snort-2.9.7.6/etc/\* . (注意有个点号跟\*号中间有空格)

tar zxvf /usr/local/src/snort+IPtables/snortrules-snapshot-2976.tar.gz

cp ./etc/\* .

touch /etc/snort/rules/white\_list.rules /etc/snort/rules/black\_list.rules

为snort添加一个用户和组

groupadd -g 40000 snort

useradd snort -u 40000 -d /var/log/snort -s /sbin/nologin -c SNORT\_IDS -g snort

cd /etc/snort

chown -R snort.snort \*

chown -R snort.snort /var/log/snort

vi /etc/snort/snort.conf

ipvar HOME\_NET 10.1.1.0/24（本次实验为此ip段地址）

ipvar EXTERNAL\_NET any

var RULE\_PATH /etc/snort/rules

var SO\_RULE\_PATH /etc/snort/so\_rules

var PREPROC\_RULE\_PATH /etc/snort/preproc\_rules

var WHITE\_LIST\_PATH /etc/snort/rules

var BLACK\_LIST\_PATH /etc/snort/rules

preprocessor sfportscan: proto  { all } memcap { 10000000 } sense\_level { low }(检测端口扫描，不去注释也可以，去掉注释用nmap扫描即可看到扫描日志)

include $PREPROC\_RULE\_PATH/preprocessor.rules(注释去掉)

include $PREPROC\_RULE\_PATH/decoder.rules(注释去掉)

include $PREPROC\_RULE\_PATH/sensitive-data.rules(注释去掉)

为snort的文件在另一个位置创建一个同步链接

ln -s /usr/local/bin/snort /usr/sbin/snort

修改snort及daq等相关目录和文件权限

cd /usr/local/src/snort+IPtables

chown -R snort.snort daq-2.0.6

chmod -R 700 daq-2.0.6

chown -R snort.snort snort-2.9.7.6

chmod -R 700 snort-2.9.7.6

cd /usr/local/src/

chown -R snort.snort snort\_dynamicsrc

chmod -R 700 snort\_dynamicsrc

cd /var/log

chown -R snort.snort snort

chmod -R 700 snort

cd /usr/local/bin

chown -R snort.snort daq-modules-config

chmod -R 700 daq-modules-config

chown -R snort.snort u2\*

chmod -R 700 u2\*

cd /etc

chown -R snort.snort snort

chmod -R 700 snort

配置动态规则

mkdir -p /usr/local/lib/snort\_dynamicrules

cp /etc/snort/so\_rules/precompiled/RHEL-6-0/x86-64/2.9\*/\*.so /usr/local/lib/snort\_dynamicrules    (实验环境是64位系统)

如果在32位系统上，这样导入动态规则:

cp /etc/snort/so\_rules/precompiled/RHEL-6-0/i386/2.9\*/\*.so /usr/local /lib/snort\_dynamicrules/

cd /usr/local/lib

chown -R snort.snort snort\*

chmod -R 700 snort\*

chown -R snort.snort pkgconfig

chmod -R 700 pkgconfig

导出动态规则文件

snort -c /etc/snort/snort.conf --dump-dynamic-rules=/etc/snort/so\_rules

如报错，很有可能是导入动态规则时未能导入对应系统架构的规则文件。

设定告警文件权限

touch /var/log/snort/alert

cd   /var/log/snort

chown snort.snort  alert

chmod 700 alert

snort -T -c /etc/snort/snort.conf -i eth0 检测配置文件是否有错误

启动snort

snort -c /etc/snort/snort.conf -i eth0

停止snort

ps -ef |grep snort

kill -9 pid号  即可结束进程

实验步骤二

安装配置guardian

安装配置

      cd /usr/local/src/snort+IPtables

      tar zxvf guardian-1.7.tar.gz

      cd guardian-1.7

      touch /etc/snort/guardian.ignore

      touch /etc/snort/guardian.target

      touch /var/log/snort/guardian.log

      cp guardian.pl /usr/local/bin/

      cp scripts/iptables\_block.sh /usr/local/bin/guardian\_block.sh

      cp scripts/iptables\_unblock.sh /usr/local/bin/guardian\_unblock.sh

      cp guardian.conf /etc/snort

编辑guardian配置文件

      vi /etc/snort/guardian.conf

      Interface   eth0

      LogFile    /var/log/snort/guardian.log

      AlertFile   /var/log/snort/alert             //alert文件的位置

      IgnoreFile  /etc/snort/guardian.ignore        //白名单

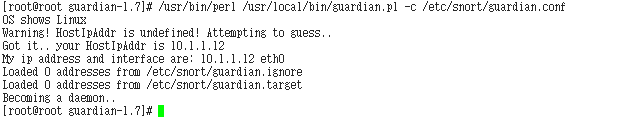
      targetFile   /etc/snort/guardian.target        //黑名单

      TimeLimit 120                       //阻断时间,以秒为单位

guardian启动

      /usr/bin/perl  /usr/local/bin/guardian.pl -c  /etc/snort/guardian.conf

启动成功会出现如下示例：



Guardian 停止

      ps -ef|grep guardian

      kill -9 pid号即可杀死该进程

实验步骤三

联动测试（在执行以下操作之前，请先停止guradian，否则会导致无法登陆试验机。）

      Snort与本地iptables联动

测试规则是否加载生效

       vi /etc/snort/rules/local.rules

添加下面两条规则

      alert tcp $HOME\_NET any -> $EXTERNAL\_NET any (msg:"OUT"; sid:5000005)

      alert tcp $EXTERNAL\_NET any -> $HOME\_NET any (msg:"IN"; sid:5000006)

规则说明：

      告警外网和内网之间的所有tcp流量，用来测试你的snort.conf配置是否有问题

启动snort，查看alert是否有日志

      snort -c /etc/snort/snort.conf -i eth0

      cd /var/log/snort

      tail -f alert

      如果可以看到日志，则表明snort.conf配置没有问题，可以继续下面的测试，否则请检查snort.conf的配置，以及snort目录和文件权限设置。

联动测试

如果之前的测试没有问题，请将/etc/snort/rules/local.rules你所添加的两条规则删除或注释掉。

      vi /etc/snort/rules/local.rules

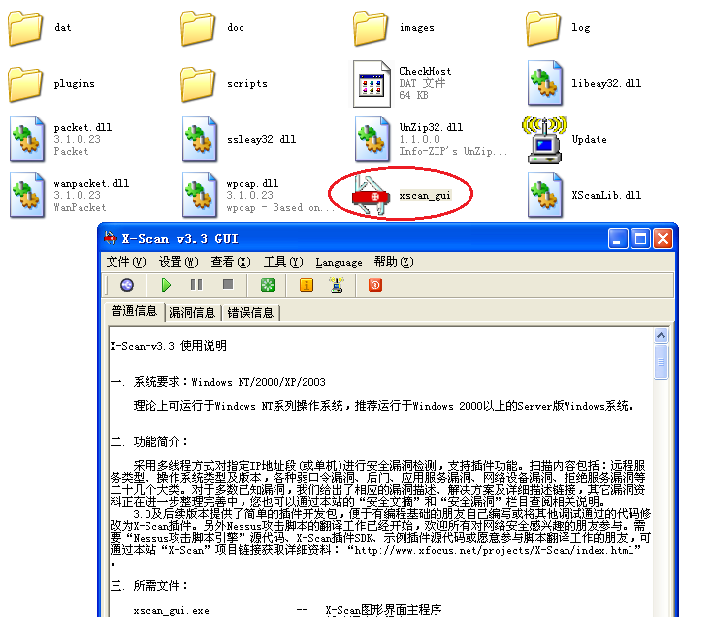
      #alert tcp $HOME\_NET any -> $EXTERNAL\_NET any (msg:"OUT"; sid:5000005)

      #alert tcp $EXTERNAL\_NET any -> $HOME\_NET any (msg:"IN"; sid:5000006)

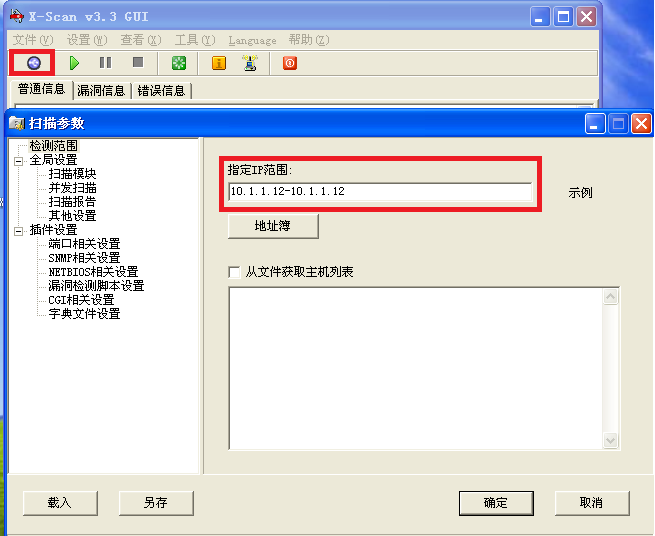
      登录另一台windows 测试主机(注意:如果用本机扫描很有可能导致snort主机无法登陆)，打开浏览器，登录http://tools.hetianlab.com/tools/X-Scan-v3.3-cn.rar下载x-scan并解压，双击文件夹



      打开x-scan

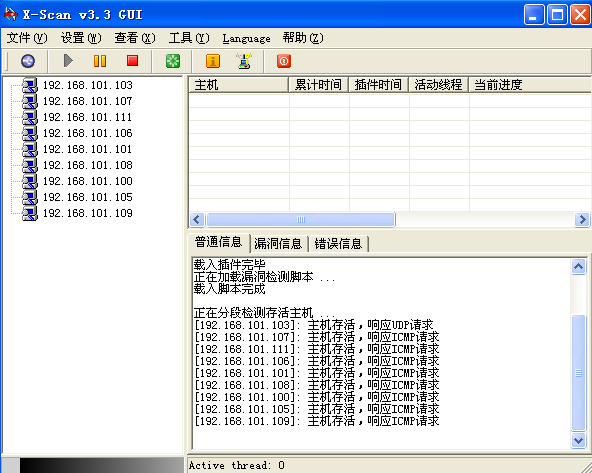


      设置扫描参数



      点击“扫描参数”按钮，将指定IP范围设置成你的snort主机ip地址，在“全局设置”的“扫描模块”中选中“全选”，“插件设置”的“SNMP相关设置”、“NETBIOS相关设置”、“漏洞检测脚本设置”均选择全部选中。

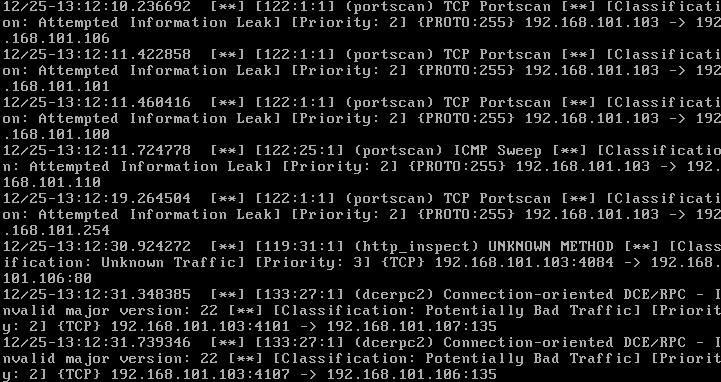
      点击开始按钮，开始扫描



观察alert是否有告警日志

      cd /var/log/snort

      tail -f alert

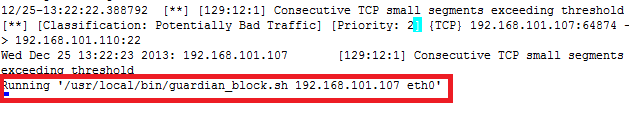


      检测到的扫描日志

发现告警日志，启动guardian与iptables联动

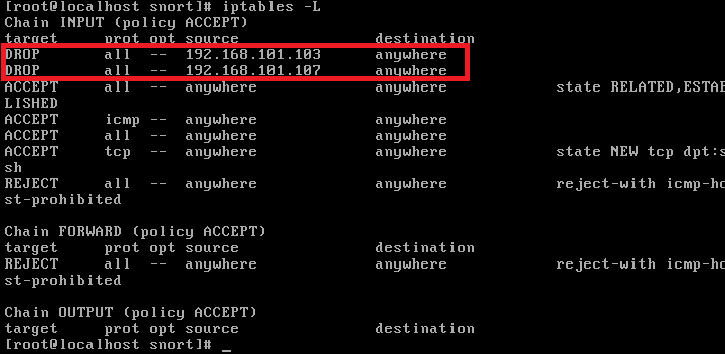
      /usr/bin/perl  /usr/local/bin/guardian.pl -c  /etc/snort/guardian.conf

在snort主机上执行iptables -L观察是否有规则加入，有规则即实验成功。



      可以看到，执行了block脚本

      在snort主机上执行，iptables -L即可看到新加入的规则。



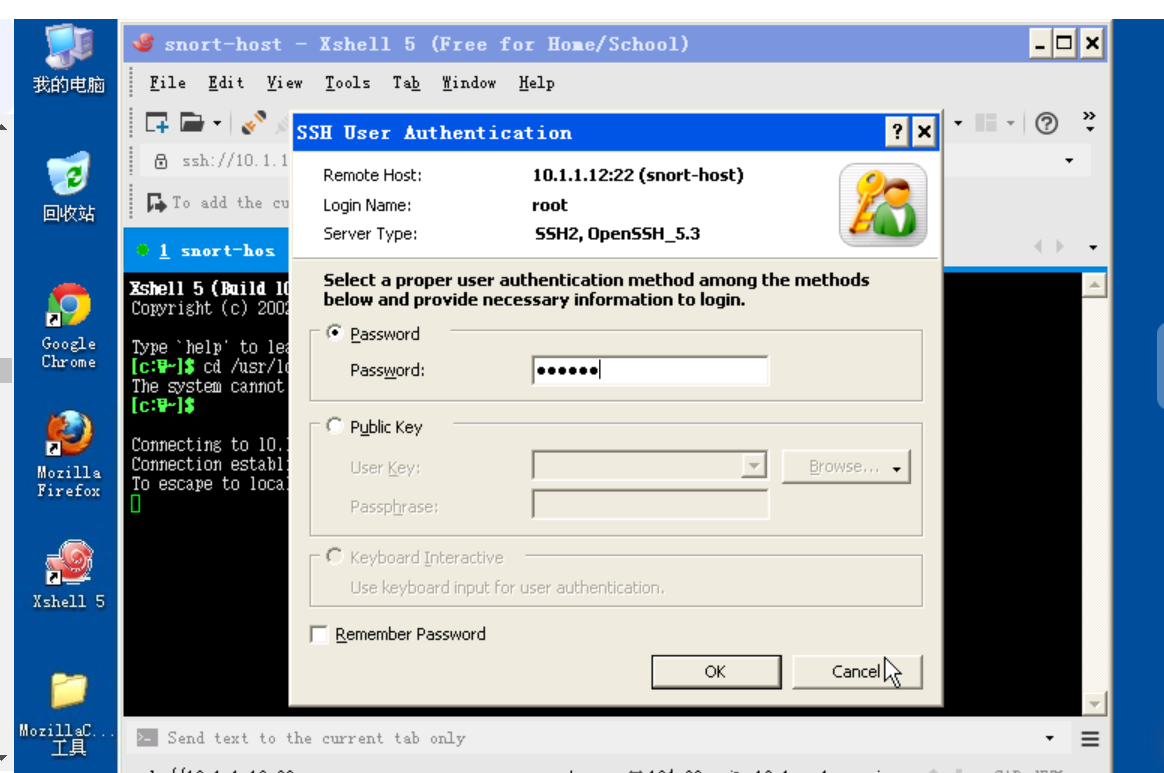
      至此，单台防火墙联动已经成功实现。

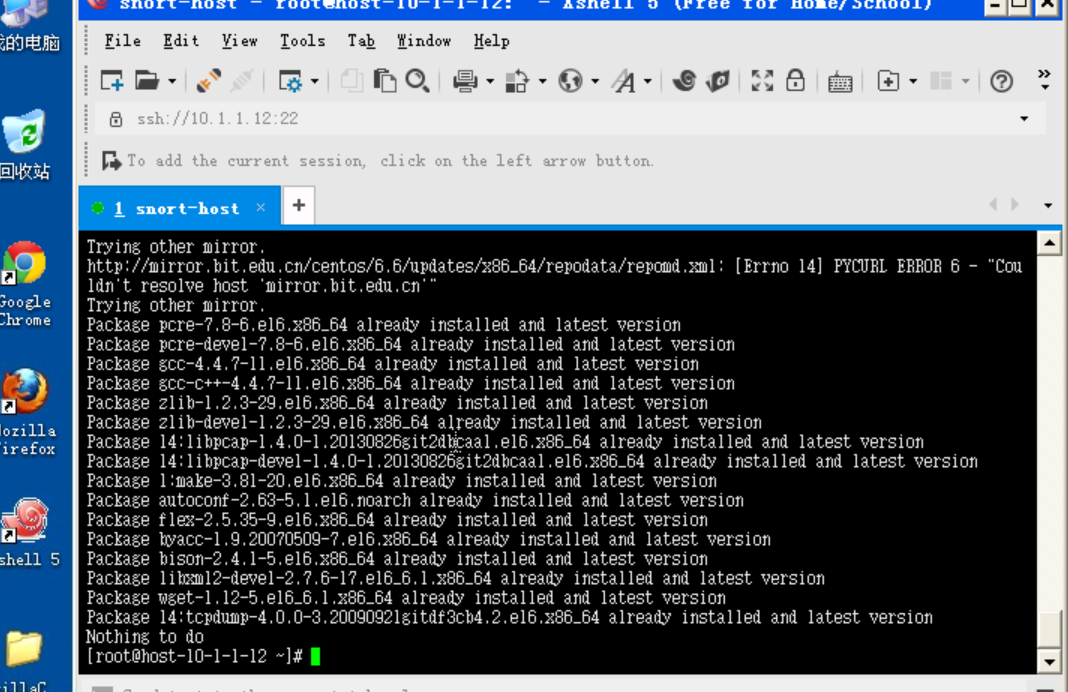
四、实验过程与分析

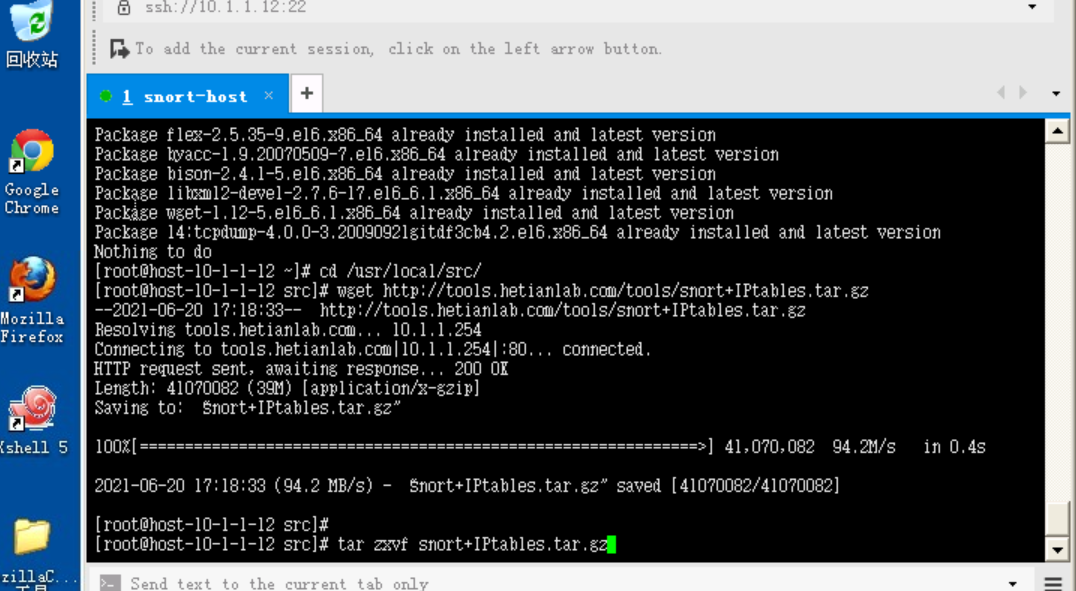
*根据具体实验，记录、整理相应命令、运行结果等，包括截图和文字说明。*

*详细记录在实验过程中发生的故障和问题，并进行故障分析，说明故障排除的过程及方法。*

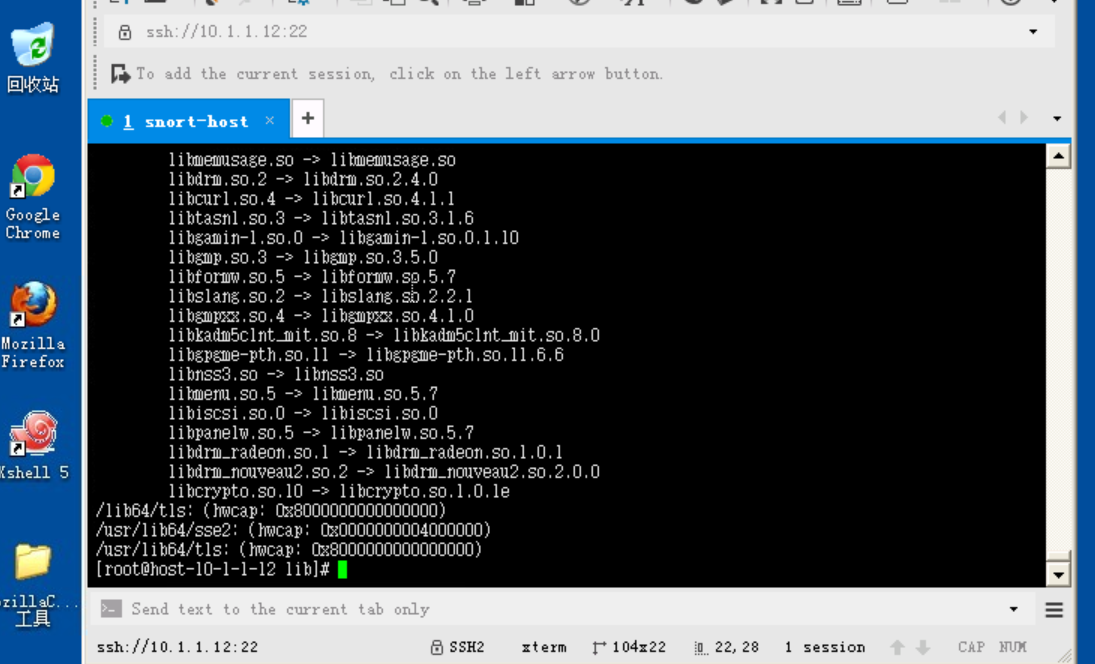
下载 xshell 登陆linux



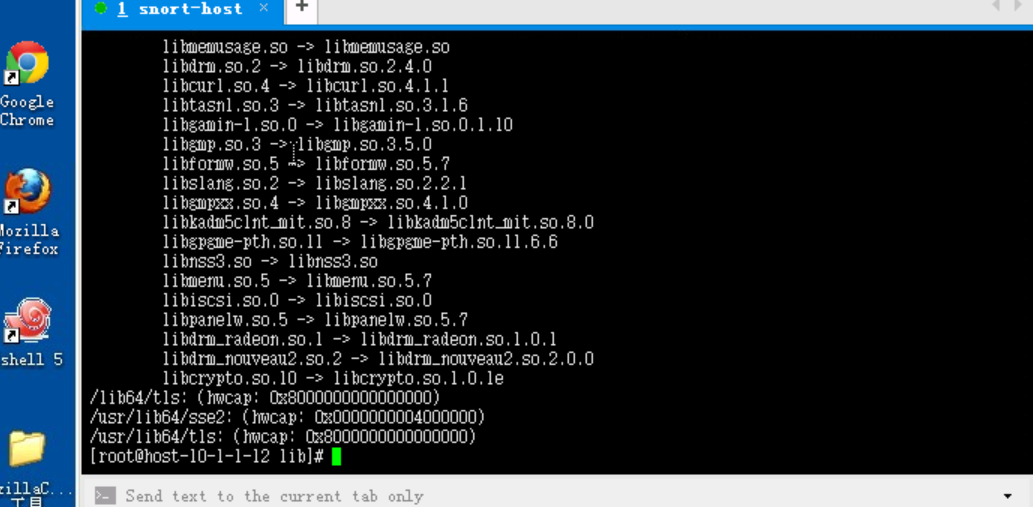




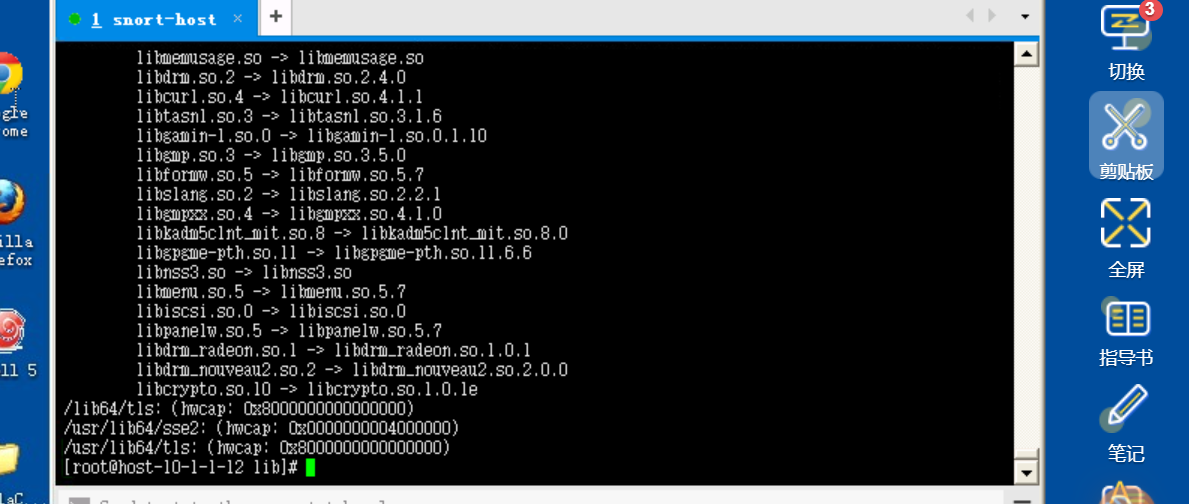
安装libdnet



安装DAQ

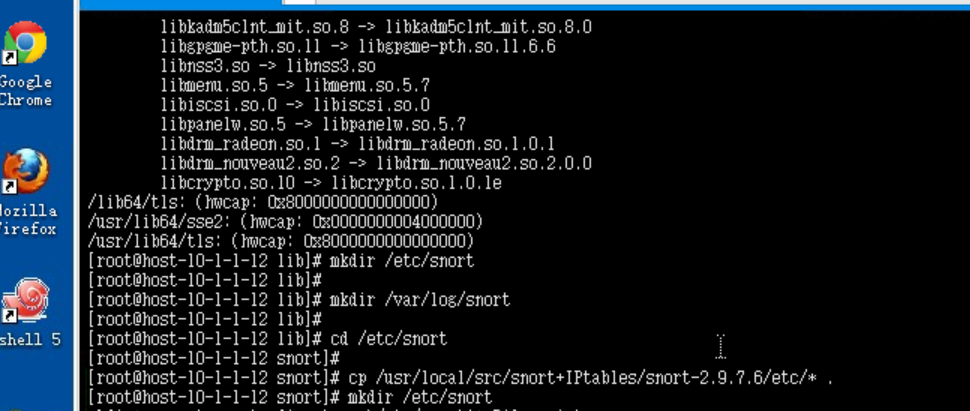


安装snort

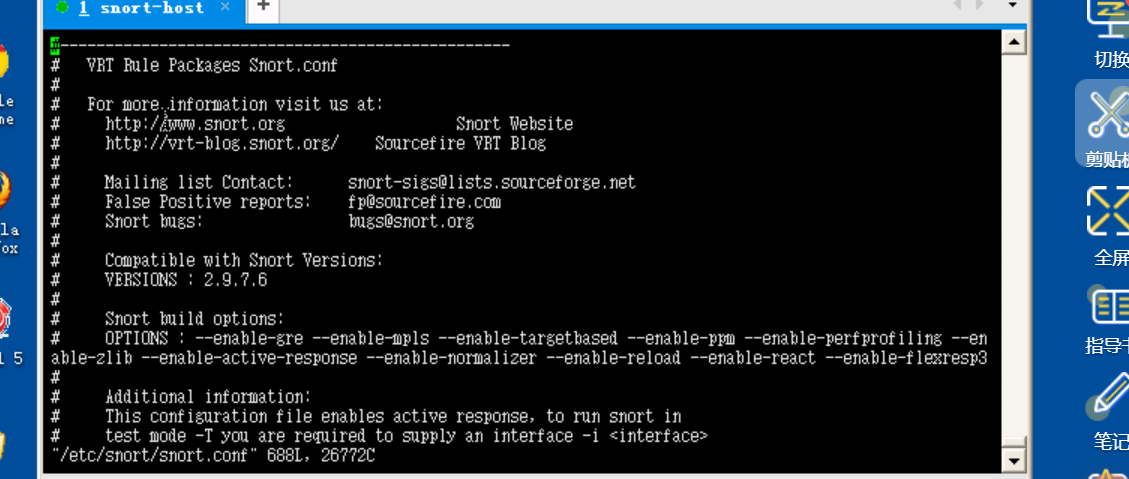


Snort配置

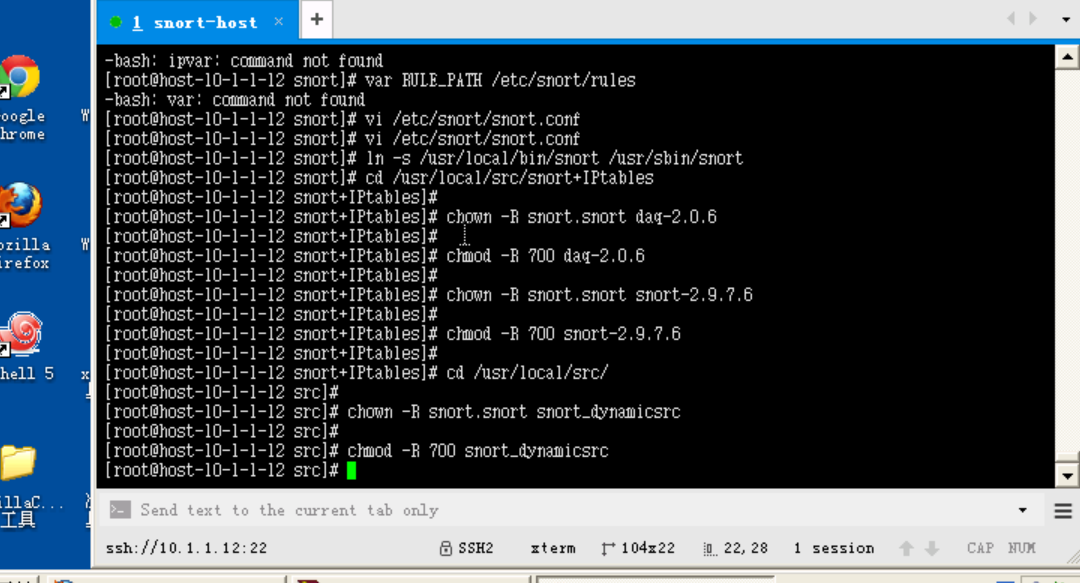
创建snort系统必要的文件夹，导入规则库，建立黑名单和白名单文件

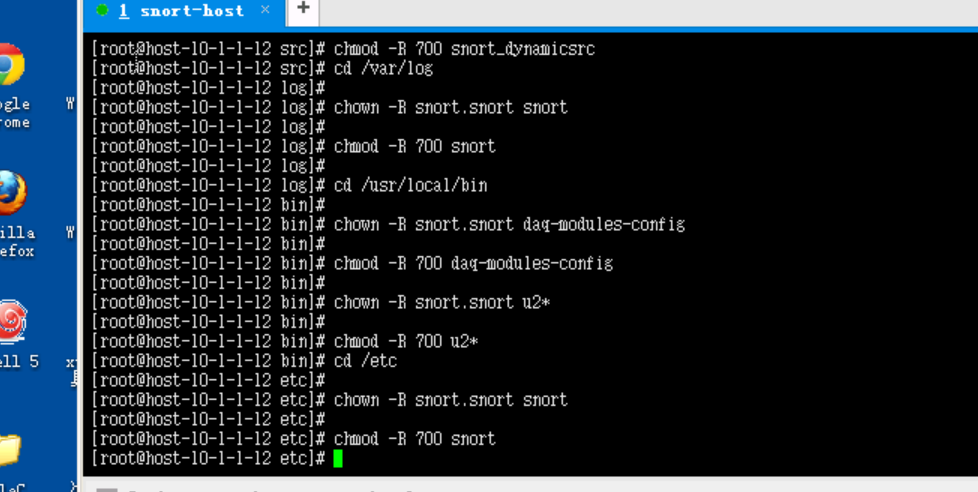


为snort添加一个用户和组



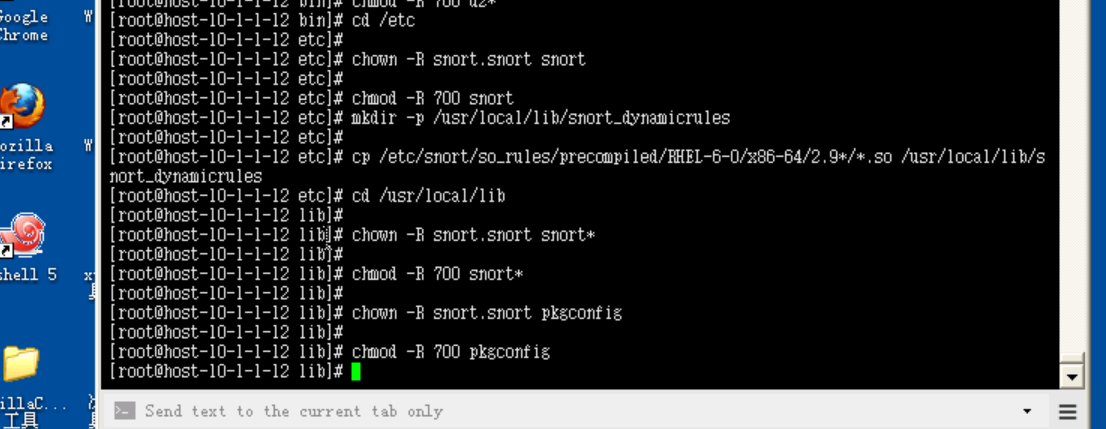
修改snort及daq等相关目录和文件权限



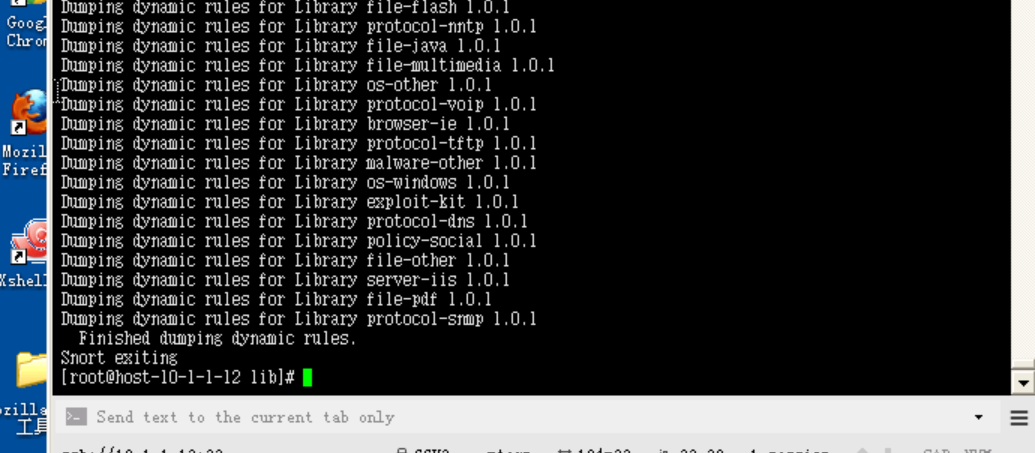


配置动态规则

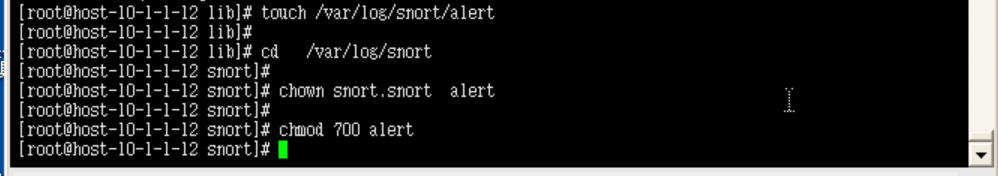


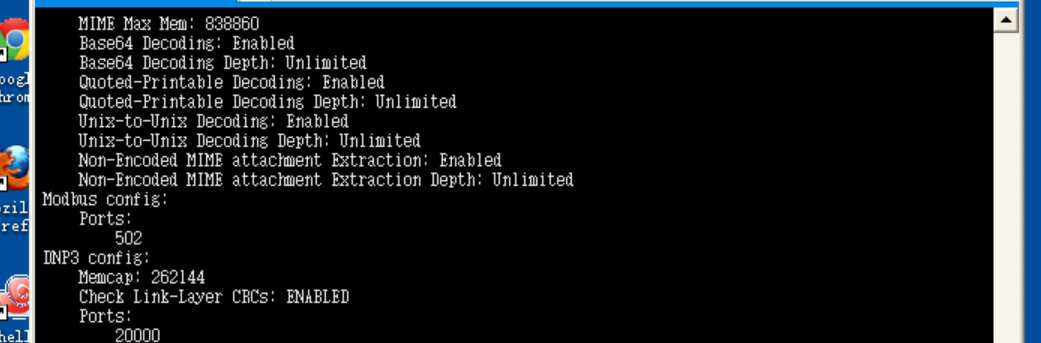


导出动态规则文件



设定告警文件权限

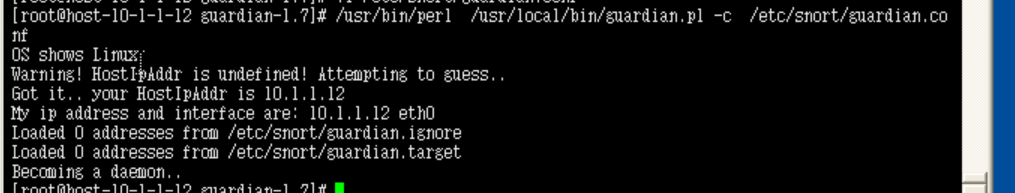




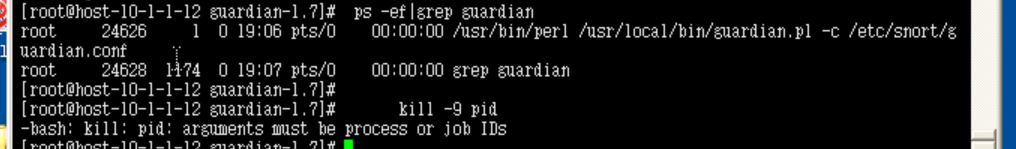
安装配置guardian



guardian启动

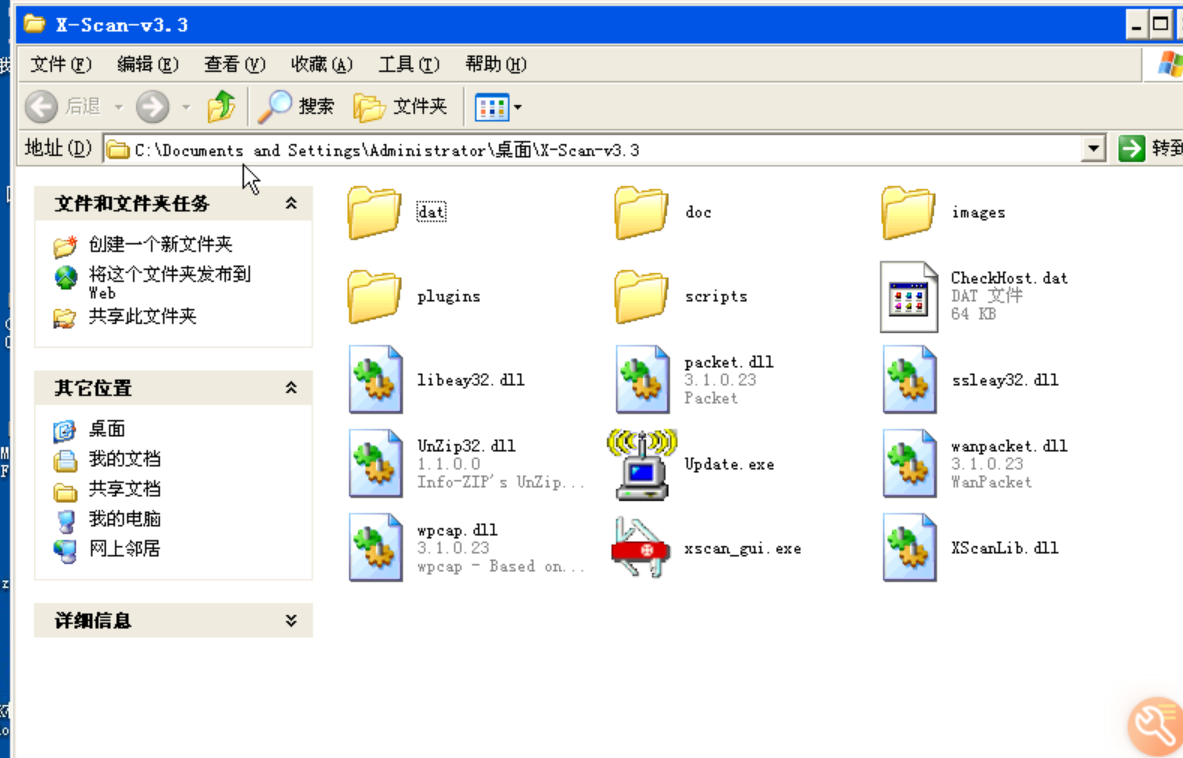


Guardian 停止

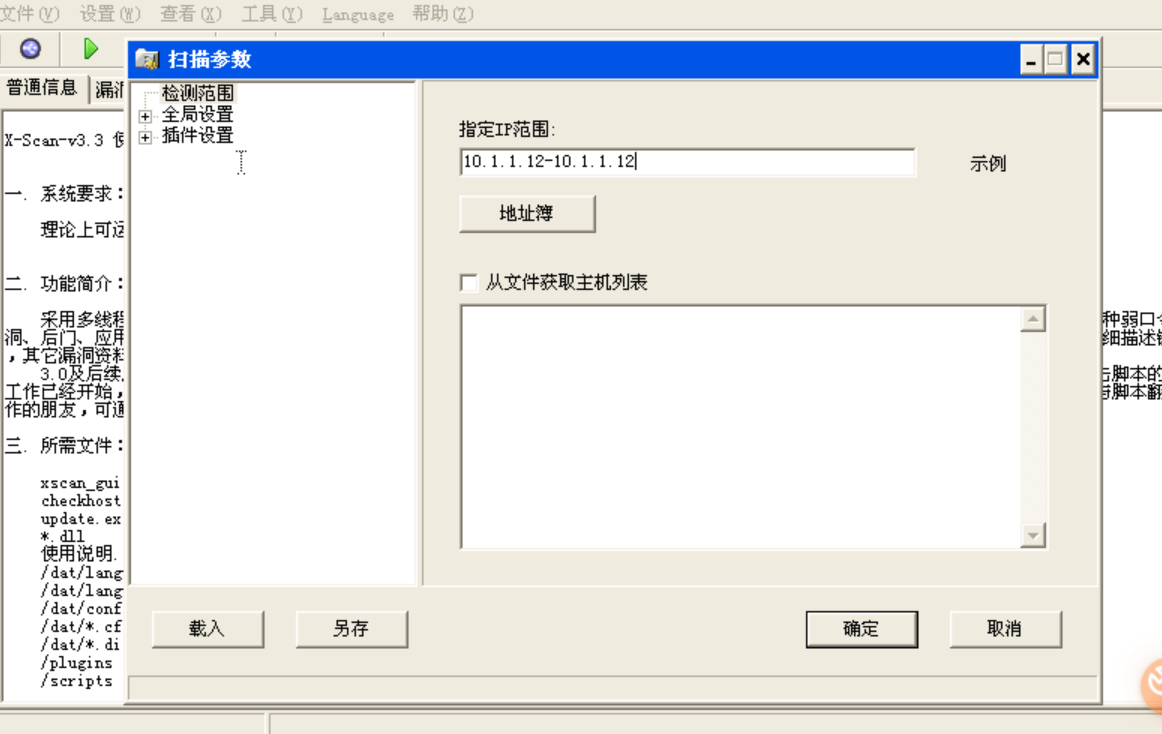


可以看到日志，则表明snort.conf配置没有问题

<http://tools.hetianlab.com/tools/X-Scan-v3.3-cn.rar下载x-scan并解压>

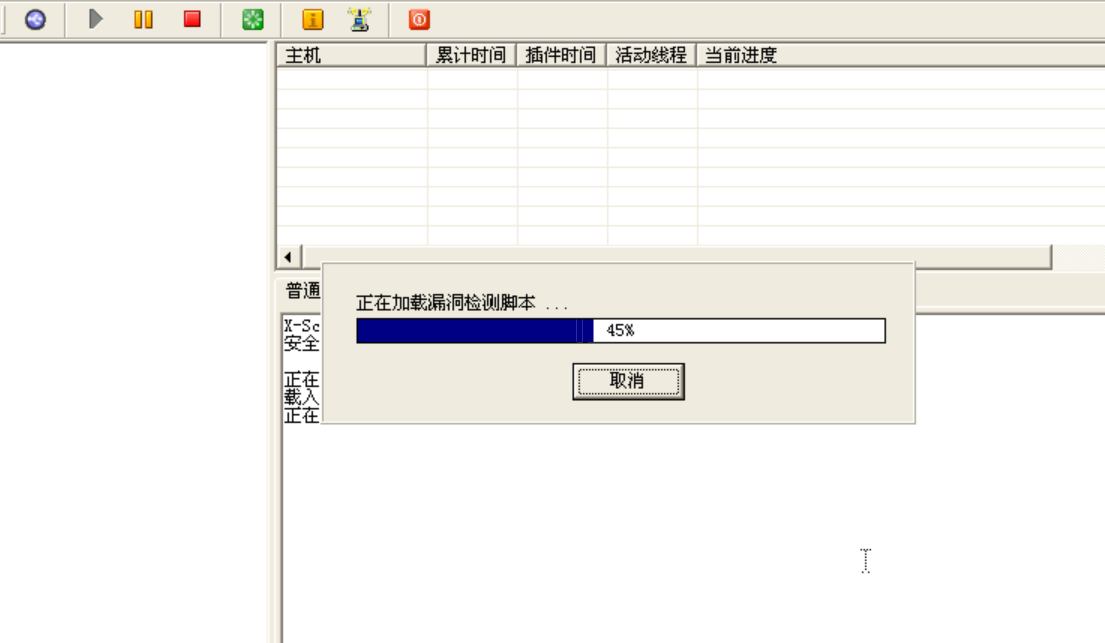


 设置扫描参数



在“全局设置”的“扫描模块”中选中“全选”，“插件设置”的“SNMP相关设置”、“NETBIOS相关设置”、“漏洞检测脚本设置”均选择全部选中。

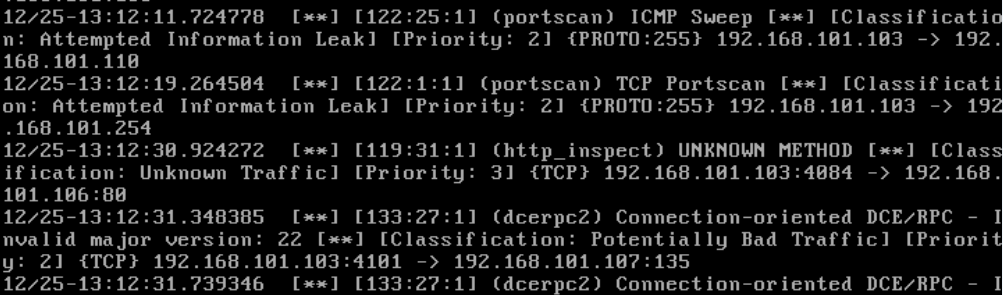
 点击开始按钮，开始扫描



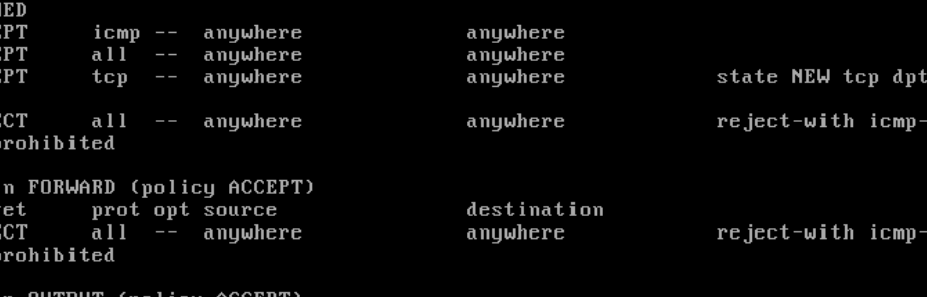
观察alert是否有告警日志

      cd /var/log/snort

      tail -f alert



在snort主机上执行iptables -L观察是否有规则加入，有规则即实验成功。



至此，单台防火墙联动已经成功实现。

五、实验结果总结

*对实验结果进行分析，完成思考题目，总结实验的心得体会，并提出实验的改进意见。*

1、 请查阅相关资料了解snortsam插件，分析并简述snortsam与guardian的优缺点。

Snort：不能阻断攻击！Snort的主要作用是对整个网络起到预警作用，从它的旁路部署模式也可以看出，它并不能阻断网络里的攻击行为。

Iptables：因为Iptables的规则过于固定，并且Iptables并不能识别网络里的攻击行为。

2、你认为snortsam和guardian的这种联动方式有哪些弊端？

通过前面的知识我们了解到，Snort有个插件机制提供了预处理插件和处理插件等方式。而这种插件在Snort里是支持自定义开发并加载的。因此第一种实现方式就是自定义开发插件，当检测到规则匹配时则调用远程或对应主机的防火墙，将有入侵行为的ip和端口，建立对应的一条Iptables规则丢弃这个连接、端口的数据包或将此ip的所有包都丢弃。

相对于Snort的插件方式，第二种的实现方式非常简单且易于实现，这种方式就是利用一个简单的脚本实时读取告警日志，将记录到的Ip和端口，创建对应的一条Iptables规则，加入到远程或对应主机的防火墙规则中，也就是实现了同第一种方式相同的功能，虽然后者在处理速度上没有第一种方式及时，但整体防护能力上并未有太大什么区别。

系统容易做到一半就奔溃…

实验资源（视频）准备好

时间太紧，这次试验两个小时有点不够用