

# XX 大学 XX 学院

## 《网络攻击与防御》

### 实验报告

课 程 名 称 : 信息安全概论

---

实 验 名 称 : Snort 与单台防火墙联动实验

---

指 导 教 师 : 翟健宏

---

学 生 姓 名 :

---

组        号 :

---

实 验 日 期 : 2021.6.18

---

实 验 地 点 :

---

实 验 成 绩 :

---

计算机科学与技术学院

计算机系网络教研室制

# 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。

## 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>

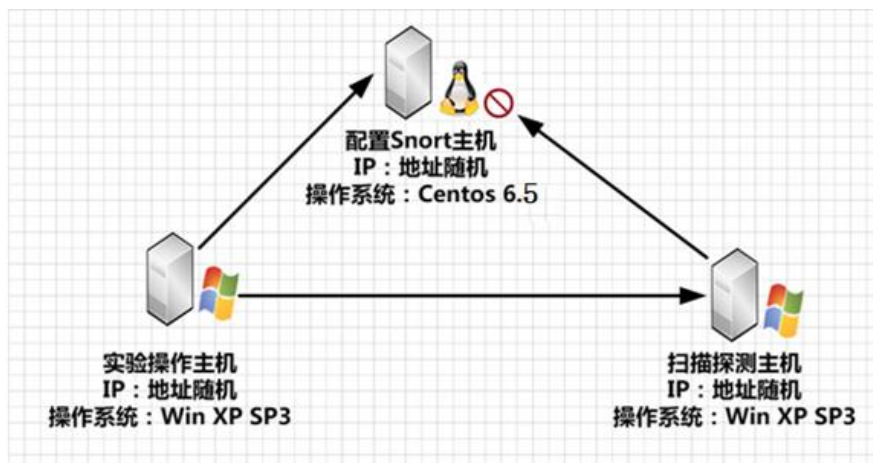


图 1 snort 实验环境

### 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。

实验内容、原理分析及具体实验要求。

#### 1. Snort 简述

Snort 是一个强大的轻量级网络入侵检测系统，它能够检测到各种不同的攻击方式，对攻击进行实时报警。此外，Snort 具有很好的扩展性和可移植性，并且这个软件遵循 GPL，这意味着只要遵守 GPL 的任何组织和个人均可以自由使用这个软件。

Snort 具有实时流量分析和日志 IP 网络数据包的能力，能够快速地检测网络攻击，及时地发出报警。Snort 的报警机制很丰富，例如：Syslog、用户指定的文件、一个 Unix 套接字，还有使用 samba 协议向 Windows 客户端发出 WinPoup 消息。利用 XML 插件，Snort 可以使用 SNML 把日志存放到一个文件或者实时报警。Snort 能够进行协议分析、内容的搜索/匹配。目前 Snort 能够分析的协议有 TCP、UDP、ICMP，将来可能支持 ARP、OSPF、IPX、RIP 等协议，它能够检测多种方式的攻击和探测，例如：缓冲区溢出、CGI 攻击、端口暴力破解、SMB 探测以及 web 应用程序的攻击现在也已经有简单规则的支持。Snort 自带的检测攻击的规则数量有限，但 Snort 支持用户自定义规则的加载，这对有能力的大型企业而言是个不错的 IDS 选择。

Snort 体系架构

#### 1.Snort 工作原理与应用场景

Snort 是一个基于模式匹配的的网络入侵检测系统，实际上目前现在市场上的大多商业入侵检测系统都是基于模式匹配的，即将恶意行为和恶意代码预定成入侵规则特征库，然后将实际数据源于规则库的特征码进行匹配，以判断其中是否包含了入侵行为。

IDS 的应用场景一般如下图所示：

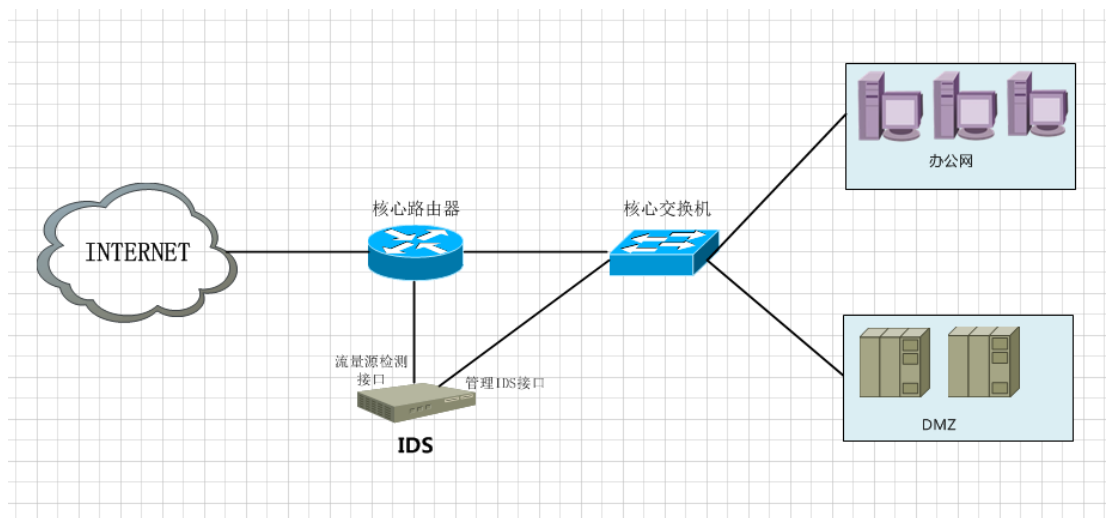


图 2 IDS 使用场景

Snort 的大致处理流程如下图所示：

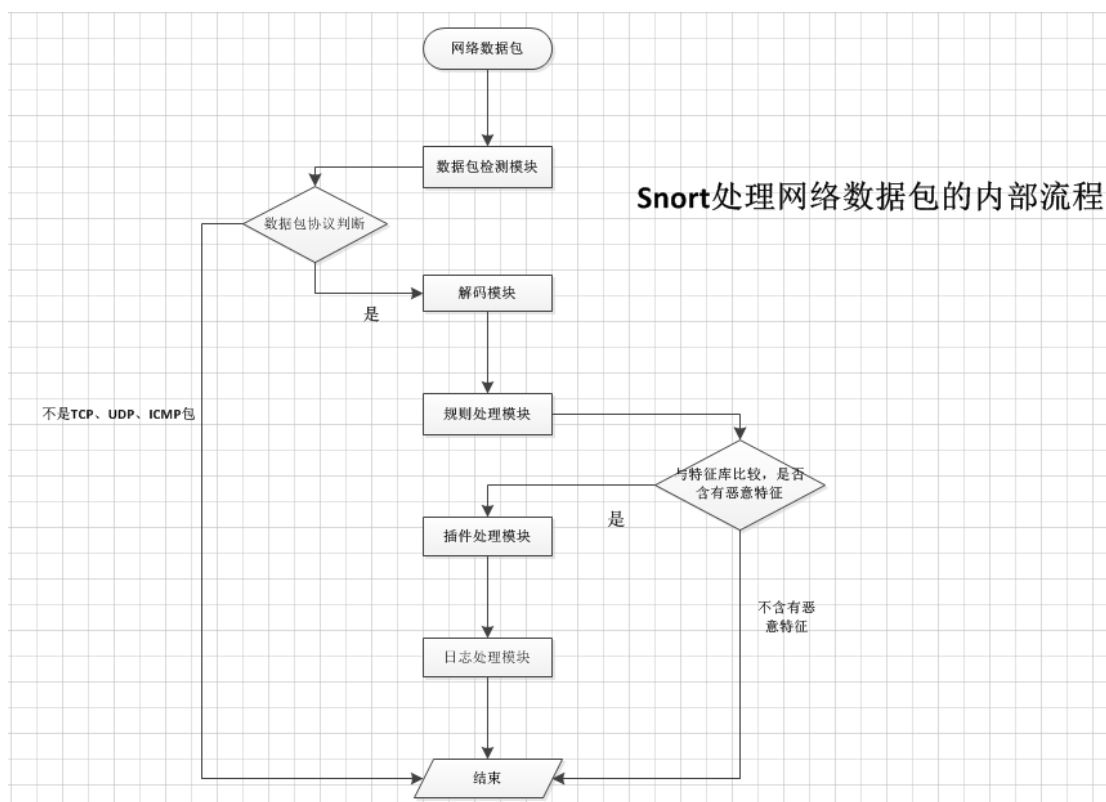


图3 snort 处理网络数据包内部流程

## 2. Snort 的主体架构

Snort 系统总体上是由规则集及 Snort 可执行程序两大部分组成。

### 1) Snort 规则集

Snort 规则集是 Snort 的攻击特征库，每条规则是一条攻击标识，Snort 通过它来识别攻击行为。

### 2) Snort 可执行程序

可执行程序由 4 个重要的子系统构成：

数据包捕获和解码子系统、检测引擎、日志/报警子系统、预处理器。

Snort 的总体结构：

序号	模块名称	源文件名称	备注
1	主控模块	Snort.cPlugbase.c...	主控模块，完成插件的管理和服务功能。
2	解码模块	Decode.c...	完成报文解码功能，把网络数据包解码成 snort 定义的 Packet 结构，用于后续分析
3	规则处理模块	Rules.cParser.c...	完成所有与规则相关的功能
4	预处理插件模块	Spp_defrag.c...	模拟 tcp、ip 堆栈功能的插件、各种解码插件
5	处理插件模块	Sp_IP_fragbits.c...	辅助完成基于规则的匹配过程
6	输出插件模块	Spo_alert_fast.c...	系统日志和告警日志的输出处理
7	日志模块	Logs.c...	完成所有与日志记录有关的功能
8	辅助模块	Ubi_Bintree.c...	辅助功能

图4 snort 内部结构

## 3. Snort 的插件机制

### 1) 预处理插件

预处理插件在规则匹配之前运行，完成的功能主要为：

- (1) 模拟 tcp、ip 堆栈功能的插件，如 IP 碎片重组、TCP 流重组插件；
- (2) 各种解码插件：http 解码插件、unicode 解码插件、rpc 解码插件、Telnet 解码插件等；

(3) 规则匹配无法进行攻击检测时所用的插件：端口扫描插件、spade 异常入侵检测插件、bo 检测插件等

## 2) 处理插件

处理插件在规则匹配阶段的 `parse rule options` 中被调用，辅助完成基于规则的匹配库。每个规则处理函数通常对应规则选项中的一个关键字，实现对这个关键字的解释。其主要功能为：

- (1) 检查协议各字段，如 TCPflag、ICMPtype、Fragbits、RPC、Dsize 等；
- (2) 辅助功能，例如关闭连接、会话记录、攻击响应等

## 3) 输出插件

输出插件在规则匹配过程中和匹配过程结束后调用，以便记录日志和告警。

## 4. 总体流程

Snort 的入侵检测流程分成两大步：

第一步是规则的解析流程，包括从规则文件中读取规则和在内存中组织规则。其过程为：

- (1) 读取规则文件；
- (2) 依次读取每条规则；
- (3) 解析规则；
- (4) 在内存中对规则进行组织，建立规则语法树；

第二步是使用这些规则进行匹配的入侵流程。其过程为对从网络上捕获的每一条数据报文和在第一步建立的规则树进行匹配，若发现存在一条规则匹配该报文，就表示检测到一个攻击，然后安装规则规定的行为进行处理；若搜索完所有的规则都没有找到匹配的规则，则视此报文正常。

## Snort 与 Iptables 的联动

### 1. 概述

为什么要让 Snort 与 Iptables 联动呢？根据前面的介绍我们已经了解了 Snort 的工作方式与原理，聪明的你一定发现了 Snort 最致命的密码——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。谁能阻断攻击呢？——Iptable。可为什么不采用 Iptables 呢？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。那能不能综合一下它们二者的优点互补对方的缺点，而达到检测到攻击即切断攻击连接这样的效果呢。答案是可以！

### 2. 实现方式和实现原理

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，很适合本次实验。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。

实现方式总结如下：

- (1).利用 Snort 的扩展功能，自定义开发集成插件(目前有 snortsam)。
- (2).利用 Snort 的告警日志，自定义开发脚本。(本次实验所采用，脚本为 guardian)
- 实现原理大致流程图：

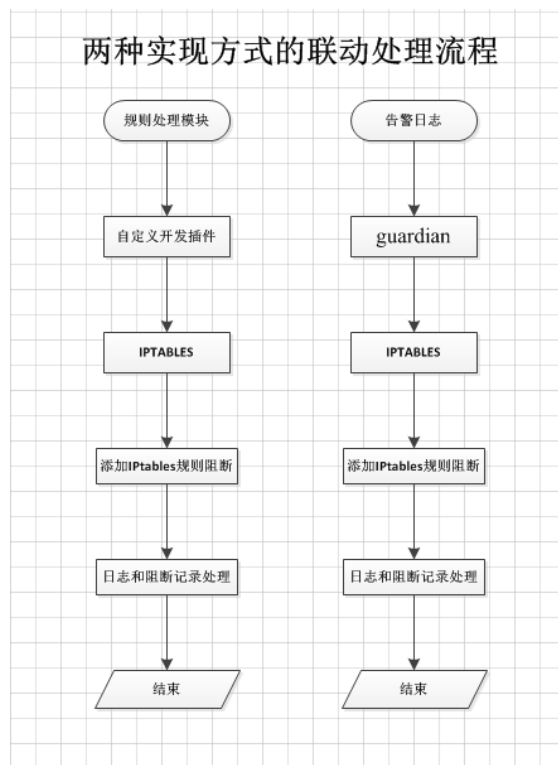


图 5 两种实现方式流程



## 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。

### 1. Snort 安装与配置

SSH 登录 centos6.5 主机

用户名: root

password:123456

① 下载 Xshell 并接入服务器，具体操作如下图所示：

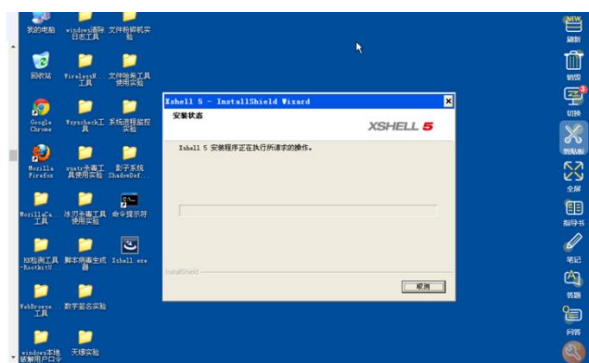


图 6 下载 xshell，接入服务器

2.接入服务器：

IP 地址: 10.1.1.12

用户名: root

password:123456

3.安装相关环境

命令：

```
cd /usr/local/src/
```

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

```
tar zxvf snort+IPtables.tar.gz
```

具体操作截图如下：

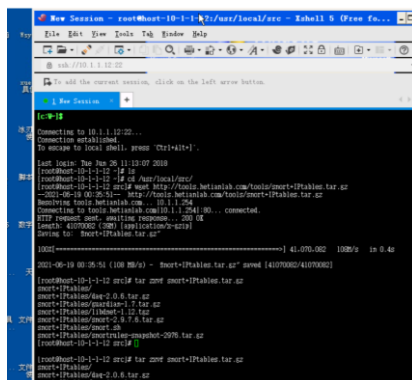


图 7 安装环境

由于接下来的操作步骤较多，不再给出截图，只展示操作代码：

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/* .
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
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 }
include $PREPROC_RULE_PATH/preprocessor.rules
include $PREPROC_RULE_PATH/decoder.rules
include $PREPROC_RULE_PATH/sensitive-data.rules

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
```

导出动态规则文件

```
snort -c /etc/snort/snort.conf --dump-dynamic-rules=/etc/snort/so_rules
```

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
```

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
```

详细截图展示如下：

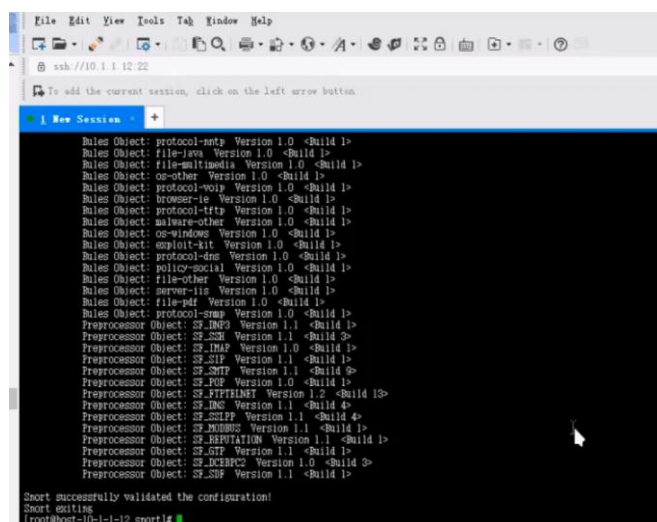


图 8 snort 环境配置

## 2. 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:

```
[root@root guardian-1.7]# /usr/bin/perl /usr/local/bin/guardian.pl -c /etc/snort/guardian.conf
OS shows Linux
Warning! HostIpAddr is undefined! Attempting to guess..
Got it.. your HostIpAddr is 10.1.1.12
My ip address and interface are: 10.1.1.12 eth0
Loaded 0 addresses from /etc/snort/guardian.ignore
Loaded 0 addresses from /etc/snort/guardian.target
Becoming a daemon..
[root@root guardian-1.7]# █
```

Guardian 停止

```
ps -ef|grep guardian
kill -9 pid 号即可杀死该进程
```

## 3. 联机测试

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，使用 x-scan 设置参数。设置参数截图示意如下：

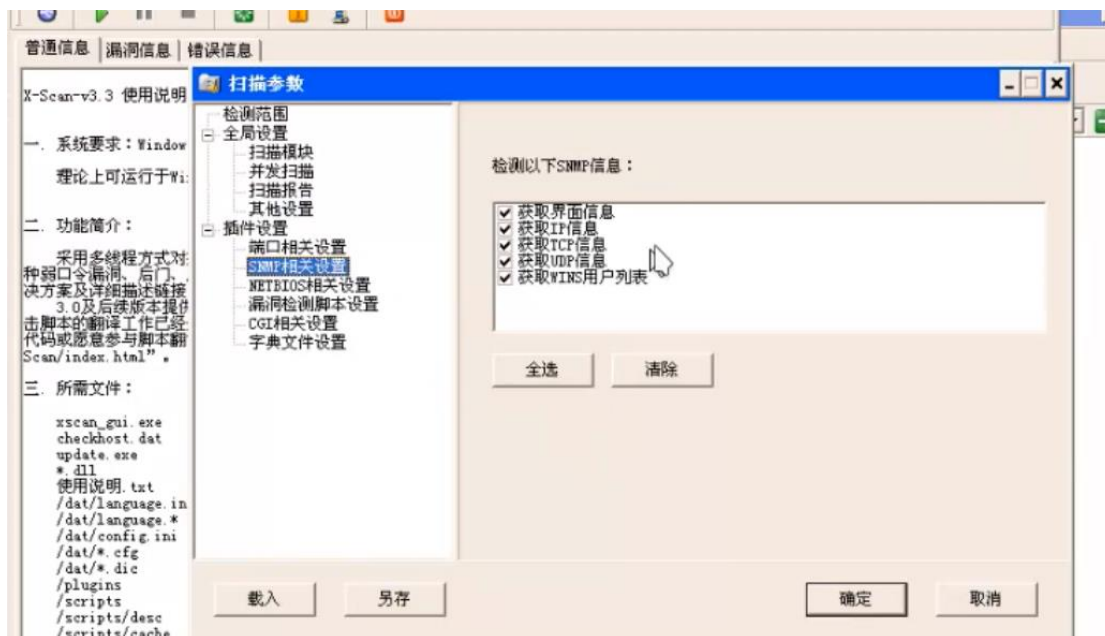


图 9 x-scan 参数设置

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

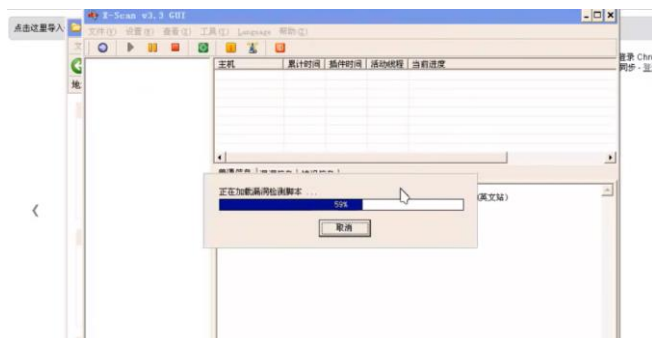


图 10 扫描过程

观察 alert 是否有告警日志

`cd /var/log/snort`

`tail -f alert`

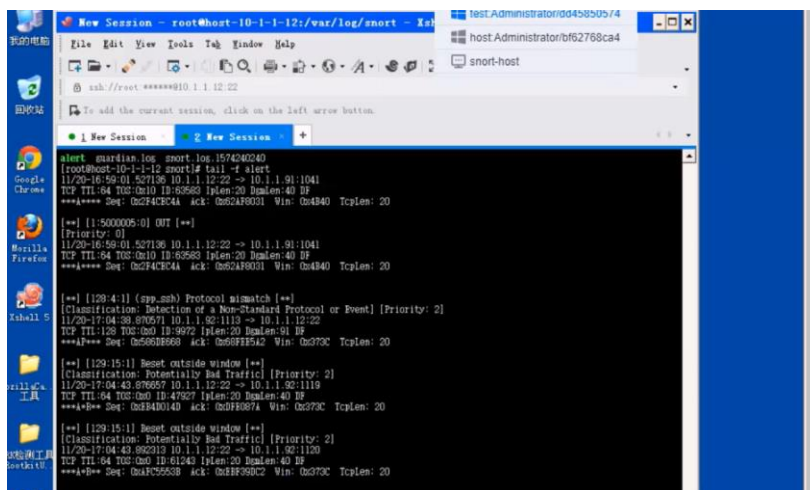


图 11 检测告警日志

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` 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。

## 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。*

### 实验结果分析：

本实验利用 snort 告警脚本，开发 guardian 脚本，实现 Snort 和 Iptables 联动，使得使用者可以识别网络中的攻击，同时使用这种方式也可以有效阻断网络攻击对于使用者的影响。

### 思考题：

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 主要起预警作用，不能阻断攻击。

Iptables：因为 Iptables 的规则过于固定，Iptables 不能识别网络中的攻击。

2、你认为 snortsam 和 guardian 的这种联动方式有哪些弊端？

第一种实现方式使用自定义插件的方法，当检测到特定行为时调用对应的防火墙，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，后者在处理速度上不如前者，但是防护效果基本相当。

### 心得体会：

本次实验使得我对于 snort 的架构和工作原理有了基本的了解，同时也掌握了 snort 和 iptables 联动的方法。从这个实验中学到了比课本上更加实际更加深入的知识，对于了解信息安全有重要的意义。

### 改进意见：

无