网络攻击与防御技术实验报告

**姓名：周美廷 学号：76066002 所在班级：160611**

**实验名称：Linux 上 Snort 的安装与配置 实验日期：2019年6月5日**

**指导老师： 实验评分：**

**验收评语：**

**参与人员：**

**实验目的：**

通过安装和配置 Linux 系统下 Snort，掌握各种常用软件的安装和使用方法，熟悉 Snort 的各种功能。

**实验内容：**

在 Linux 操作系统中，下载并安装 Apache，PHP，MySQL，Snort，Barnyard2，BASE 等源码库，查阅相关资料，将 Snort 配置为网络入侵检测模式，利用 BASE 作为 入侵检测的分析控制台，利用 Apache 服务器的 Web 页面进行管理，将 Snort 的日志、报警、 权限等信息写入 MySQL 数据库。

**实验环境：**

Windows 10 – 64 bit

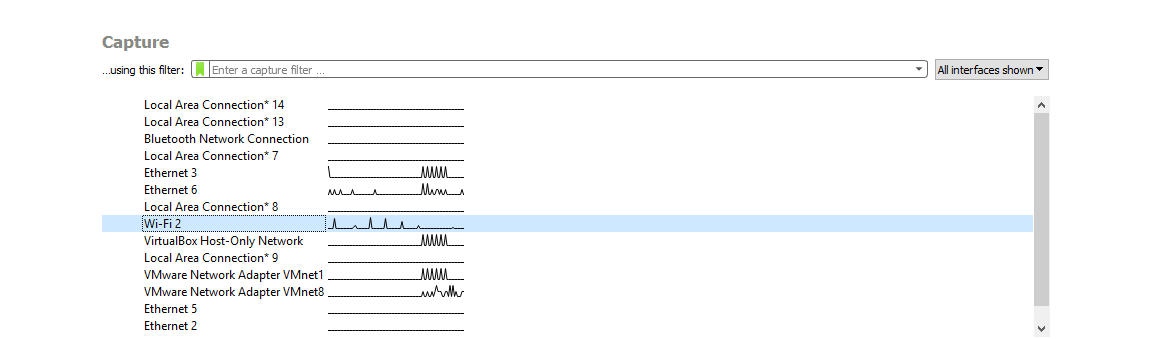
**实验设计：**

使用的软件：

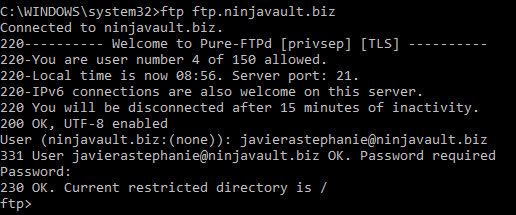
* Wireshark
* Filezilla
* 拥有FTP服务的网站
* Command Prompt

**详细过程：**

1. 开启Wireshark，选择在链接的Wi-Fi开始抓包

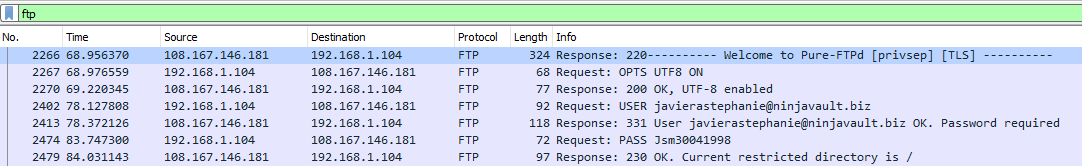


1. 打开网站（管理员）cpanel获取ftp信息
2. 使用Command Prompt进行unecrypted FTP访问



通过cmd访问ftp是不支持SSL Certificate的，所以访问过程也比较危险

1. 使用FTP过滤分析抓包信息

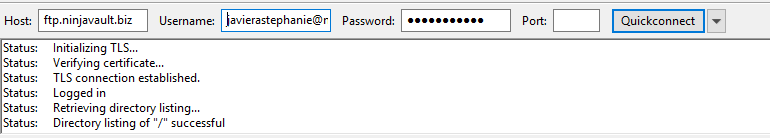


以上可以看到，这次可以获取到访问的用户名以及密码

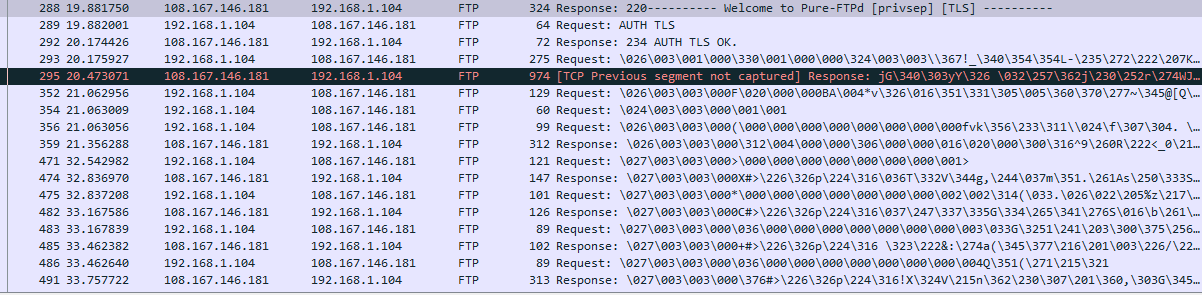




1. 使用FTP客户端Filezilla进行SSL ecrypted FTP访问



1. 使用FTP过滤分析抓包信息



可以看到这次用户名和密码并没有被嗅探到，因为Filezilla（以及其他FTP客户端）都有开启SSL Certificate访问的安全措施，有了SSL Certificate的访问用户信息的地方有写着AUTH TLS，这TLS就是说明登录时的用户authorization有SSL Certificate加密的意思。



网络攻击与防御技术实验报告

**姓名：周美廷 学号：76066002 所在班级：160611**

**实验名称：网络嗅探与防御 实验日期：2019年6月5日**

**指导老师： 实验评分：**

**验收评语：**

**参与人员：**

**实验目的：**

通过使用 Sniffer 软件来掌握 Sniffer 工具的使用方法，实现捕捉 HTTP、FTP、SMTP、 TELNET 等协议的数据包，理解 TCP/IP 主要协议的数据结构、会话连接建立和终止的过程、 TCP 序列号应答号的变化规律。

通过实验了解 HTTP、FTP、SMTP 等协议明文传输的特性，以建立安全意识。

**实验内容：**

使用一种 Sniffer 工具进行网络监听，主要集中在 HTTP、FTP、SMTP、TELNET 协议 方面，并分析监听结果。

通过熟悉该 Sniffer 工具，完成：

1. 用该工具监听用户通过网页登陆邮箱的过程，分析捕获数据包，了解数据包所使用的网络协议。
2. 在主机上进行 FTP 登陆，并进行嗅探，把嗅探到的重要信息找出来，尤其是用户名和密码。

为了减少分析的数据量，必须有效地设置过滤规则。

完成嗅探工作之后，设计一种安全方案对局域网嗅探器进行防范，付诸实施，然后再打开嗅探器，看是否还能够捕获到上述数据。

**实验环境：**

Ubuntu 16.04 – 64 bit

**实验设计：**

使用的软件：

* VirtualBox
* Apache
* PHP
* MySQL
* Snort 2.9.13
* DAQ 2.0.6

**详细过程：**

**基本工具**

1. 下载Ubuntu资源更新

sudo apt-get update

sudo apt-get dist-upgrade -y

sudo apt-get install -y openssh-server

sudo reboot // 重启是为了确认所有更新都已安装好

1. 下载Snort依赖的文件

sudo apt-get install -y build-essential // 这里面是make，make install这种用来编译的工具

sudo apt-get install -y libpcap-dev libpcre3-dev libdumbnet-dev // Snort的依赖文件

sudo apt-get install -y bison flex // DAQ的依赖文件

1. 由于需要下载的包有好几个，所以我们建一个文件夹存这些包

mkdir ~/snort\_src

cd ~/snort\_src

**DAQ2.0.6**

1. 下载并安装DAQ 2.0.6

cd ~/snort\_src

wget https://snort.org/downloads/snort/daq-2.0.6.tar.gz

tar -xvzf daq-2.0.6.tar.gz

cd daq-2.0.6

./configure

make

sudo make install

1. 若DAQ的配置已成功，那在执行./configure的时候应该有显示以下信息

Build AFPacket DAQ module.. : yes

Build Dump DAQ module...... : yes

Build IPFW DAQ module...... : yes

Build IPQ DAQ module....... : no

Build NFQ DAQ module....... : no

Build PCAP DAQ module...... : yes

Build netmap DAQ module.... : no

**SNORT2.9.13**

1. 下载Snort之前，以下是可以提高使用效率的包

sudo apt-get install -y zlib1g-dev liblzma-dev openssl libssl-dev

1. 因为实验中使用的是Ubuntu 16.04，于是需要执行以下指令下载Nghttp2的library

sudo apt-get install -y libnghttp2-dev

1. 下载Snort

cd ~/snort\_src

wget https://snort.org/downloads/snort/snort-2.9.13.tar.gz

tar -xvzf snort-2.9.13.tar.gz

cd snort-2.9.13

./configure --enable-sourcefire

make

sudo make install

这时，只要没有看到 error: “Fatal!” 的编译反馈就代表Snort已经安装好了，并可以继续。但是在实验时却因为遇到词错误。执行 ./configure | grep "... no" 可以看到错误来源是因为LuaJIT library没有安装好，所以从<http://www.luajit.org手动下载luajit2.0.5>：

cd ~/snort\_src

wget http://luajit.org/download/LuaJIT-2.0.5.tar.gz

tar zxf LuaJIT-2.0.5.tar.gz

cd LuaJIT-2.0.5

make

sudo make install

LuaJIT的问题解决后再次执行以下命令：

cd ~/snort\_src/snort-2.9.13

./configure --enable-sourcefire

make

sudo make install

1. 更新shared libraries

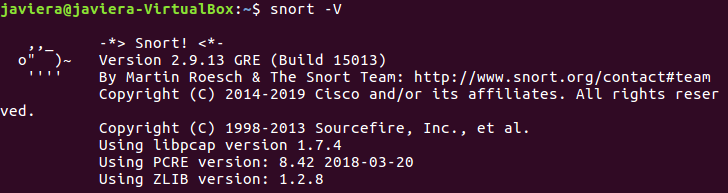
sudo ldconfig

1. 放置一个symlink在Snort binary（在/usr/sbin）里

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

1. 若Snort已配置好，可以使用以下命令查看Snort的信息

snort -V



1. 配置Snort，使Snort可以在NIDS模式跑

# Create the snort user and group:

sudo groupadd snort

sudo useradd snort -r -s /sbin/nologin -c SNORT\_IDS -g snort

# Create the Snort directories:

sudo mkdir /etc/snort

sudo mkdir /etc/snort/rules

sudo mkdir /etc/snort/rules/iplists

sudo mkdir /etc/snort/preproc\_rules

sudo mkdir /usr/local/lib/snort\_dynamicrules

sudo mkdir /etc/snort/so\_rules

# Create some files that stores rules and ip lists

sudo touch /etc/snort/rules/iplists/black\_list.rules

sudo touch /etc/snort/rules/iplists/white\_list.rules

sudo touch /etc/snort/rules/local.rules

sudo touch /etc/snort/sid-msg.map

# Create our logging directories:

sudo mkdir /var/log/snort

sudo mkdir /var/log/snort/archived\_logs

# Adjust permissions:

sudo chmod -R 5775 /etc/snort

sudo chmod -R 5775 /var/log/snort

sudo chmod -R 5775 /var/log/snort/archived\_logs

sudo chmod -R 5775 /etc/snort/so\_rules

sudo chmod -R 5775 /usr/local/lib/snort\_dynamicrules # Change Ownership on folders:

sudo chown -R snort:snort /etc/snort

sudo chown -R snort:snort /var/log/snort

sudo chown -R snort:snort /usr/local/lib/snort\_dynamicrules

1. 复制配置文件以及dynamic preprocessors

cd ~/snort\_src/snort-2.9.13/etc/

sudo cp \*.conf\* /etc/snort

sudo cp \*.map /etc/snort

sudo cp \*.dtd /etc/snort

cd ~/snort\_src/snort-2.9.13/src/dynamic-preprocessors/build/usr/local/lib/snort\_dynamicpreprocessor/

sudo cp \* /usr/local/lib/snort\_dynamicpreprocessor/

1. 执行以下指令注释掉snort.conf的一些内容

sudo sed -i "s/include \$RULE\\_PATH/#include \$RULE\\_PATH/" /etc/snort/snort.conf

1. 现在我们将手动修改snort.conf文件，这个过程可以用自己喜欢的工具，在这个实验中我使用的是nano

sudo nano /etc/snort/snort.conf

1. 修改的有这些内容

ipvar HOME\_NET 192.168.1.0/24

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/iplists

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

去掉以下代码的注释

include $RULE\_PATH/local.rules

1. 执行以下命令检查Snort配置

sudo snort -T -i enp0s3 -c /etc/snort/snort.conf

配置已成功所以编译结果的最后两行会显示

Snort successfully validated the configuration!

Snort exiting

1. 填写一些rules

local.rules:

sudo nano /etc/snort/rules/local.rules

然后写下面这一行：

alert icmp any any -> $HOME\_NET any (msg:"ICMP test detected"; GID:1; sid:10000001; rev:001; classtype:icmpevent;)（必须在一行中）

sid-msg.map

sudo nano /etc/snort/sid-msg.map

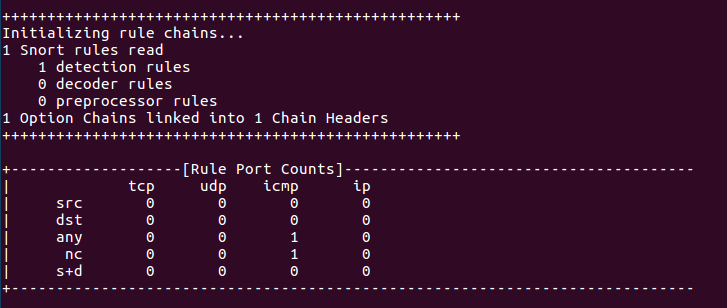
然后填写下面这一行：

1 || 10000001 || 001 || icmp-event || 0 || ICMP Test detected || url,tools.ietf.org/html/rfc792

1. 填写rules之后，Snort的配置就是有一些变化，所以需要再次检查是否能编译通过

sudo snort -T -c /etc/snort/snort.conf -i enp0s3

成功后，在编译反馈信息中会有这些内容

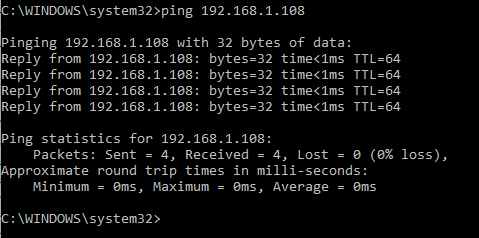


1. 执行Snort

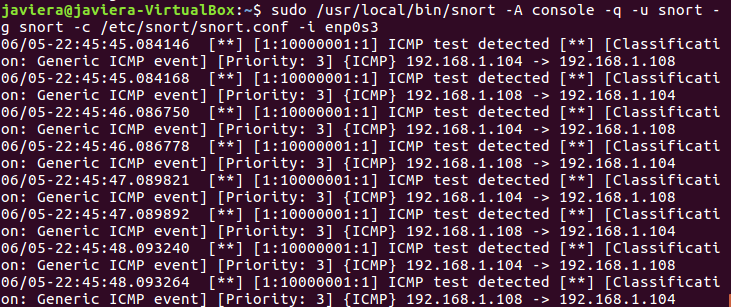
sudo /usr/local/bin/snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i enp0s3

按回车后，Snort已经在执行中了，但是还没看到任何信息，然后使用另一台PC ping这个PC的IP，这个PC的IP是192.168.1.108，于是在另一台PC

ping 192.168.1.108



这个时候，Snort的命令行会显示一些信息



**Barnyard2**

1. 下载Barnyard2的依赖文件

sudo apt-get install -y mysql-server libmysqlclient-dev mysql-client autoconf libtool

1. 修改snort.conf文件，使Snort可以一二进制的形式输出警告

在第522行（也就是# output unified2: filename merged.log, limit 128, nostamp, mpls\_event\_types, vlan\_event\_types} 后）加上这一行

output unified2: filename snort.u2, limit 128

1. 下载并安装Barnyard2 2.1.14

cd ~/snort\_src

wget https://github.com/firnsy/barnyard2/archive/master.tar.gz -O barnyard2-Master.tar.gz

tar zxvf barnyard2-Master.tar.gz

cd barnyard2-master

autoreconf -fvi -I ./m4

sudo ln -s /usr/include/dumbnet.h /usr/include/dnet.h

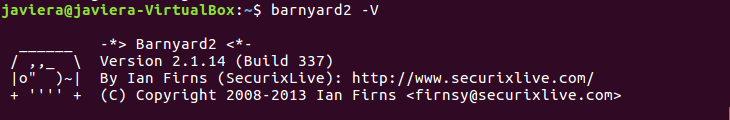
sudo ldconfig

./configure --with-mysql --with-mysql-libraries=/usr/lib/x86\_64-linux-gnu

make

sudo make install

1. 若Barnyard2已安装成功可以执行barnyard2 -V进行检查



1. 新建一些Barnyard2依赖的文件

sudo cp ~/snort\_src/barnyard2-master/etc/barnyard2.conf /etc/snort/

sudo mkdir /var/log/barnyard2

sudo chown snort.snort /var/log/barnyard2

sudo touch /var/log/snort/barnyard2.waldo

sudo chown snort.snort /var/log/snort/barnyard2.waldo

1. 执行以下代码新建MySQL用户用来访问Barnyard2保存警告的数据库

mysql -u root -p

mysql> create database snort;

mysql> use snort;

mysql> source ~/snort\_src/barnyard2-master/schemas/create\_mysql

mysql> CREATE USER 'snort'@'localhost' IDENTIFIED BY 'Jsm30041998\_';

mysql> grant create, insert, select, delete, update on snort.\* to 'snort'@'localhost';

mysql> exit

1. 在Barnyard2的配置文件最后一行加上这一行，用来告诉Barnyard2如何访问MySQL数据库

sudo nano /etc/snort/barnyard2.conf

output database: log, mysql, user=snort password=Jsm30041998\_ dbname=snort host=localhost sensor\_name=sensor01

因为这里有密码这种秘密信息，所以执行以下命令提防别的用户访问此文件

sudo chmod o-r /etc/snort/barnyard2.conf

1. 再次执行Snort（在alert模式）

sudo /usr/local/bin/snort -q -u snort -g snort -c /etc/snort/snort.conf -I enp0s3

这时也跟之前一样，使用另一台PC ping到这台PC，这次不会看到任何Debug信息因为那些信息会被输出到/var/log/snort中的snort.u2.1559739447

1. 执行Barnyard2

sudo barnyard2 -c /etc/snort/barnyard2.conf -d /var/log/snort -f snort.u2 -w /var/log/snort/barnyard2.waldo -g snort -u snort

这时会输出以下内容

1. 执行以下命令检查Barnyard2是否把events写到数据库内

mysql -u snort -p -D snort -e "select count(\*) from event"

输出的这个内容代表成功：

+----------+

| count(\*) |

+----------+

| 8 |

+----------+

**BASE**

1. 安装PHP 5.6（BASE是依赖于PHP 5）

sudo add-apt-repository ppa:ondrej/php

sudo apt-get update

sudo apt-get install -y apache2 libapache2-mod-php5.6 php5.6-mysql php5.6-cli php5.6 php5.6-common php5.6-gd php5.6-cli php-pear php5.6-xml

1. 安装Pear image Graph：

sudo pear install -f --alldeps Image\_Graph

1. 下载并安装ADODB

cd ~/snort\_src

wget https://sourceforge.net/projects/adodb/files/adodb-php5-only/adodb-520-for-php5/adodb-5.20.8.tar.gz

tar -xvzf adodb-5.20.8.tar.gz

sudo mv adodb5 /var/adodb

sudo chmod -R 755 /var/adodb

1. 下载BASE

cd ~/snort\_src

wget http://sourceforge.net/projects/secureideas/files/BASE/base-1.4.5/base-1.4.5.tar.gz

tar xzvf base-1.4.5.tar.gz

sudo mv base-1.4.5 /var/www/html/base/

1. 配置BASE

cd /var/www/html/base

sudo cp base\_conf.php.dist base\_conf.php

sudo nano /var/www/html/base/base\_conf.php 填写以下内容

$BASE\_urlpath = '/base'; # line 50

$DBlib\_path = '/var/adodb/'; # line 80

$alert\_dbname = 'snort'; # line 102

$alert\_host = 'localhost';

$alert\_port = '';

$alert\_user = 'snort';

$alert\_password = 'MySqlSNORTpassword'; # line 106

把这些内容注释掉：

//$graph\_font\_name = "Verdana";

//$graph\_font\_name = "DejaVuSans";

//$graph\_font\_name = "Image\_Graph\_Font";

$graph\_font\_name = "";

1. 设置配置文件以及BASE文件夹的访问权

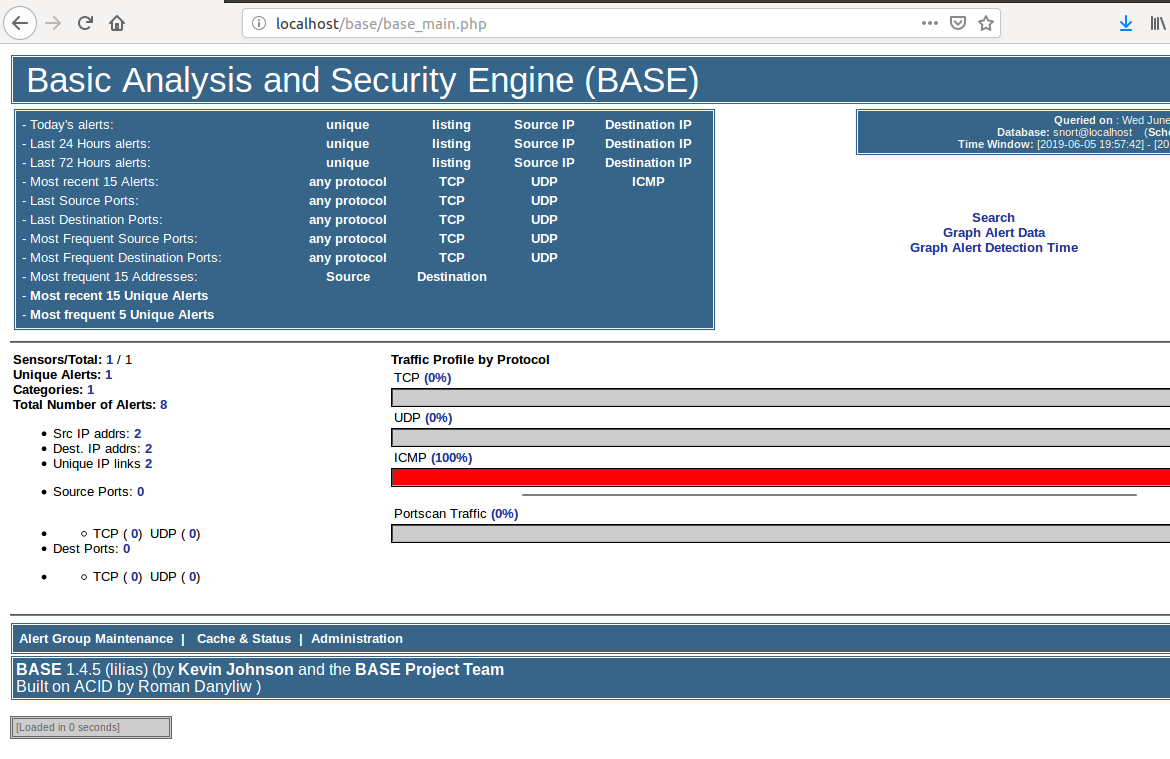
sudo chown -R www-data:www-data /var/www/html/base

sudo chmod o-r /var/www/html/base/base\_conf.php

1. 重启Apache

sudo service apache2 restart

1. 打开<http://localhost/base/index.php>然后点击Setup page 按钮
2. 点击Create BASE AG
3. 点击Main page会显示以下内容



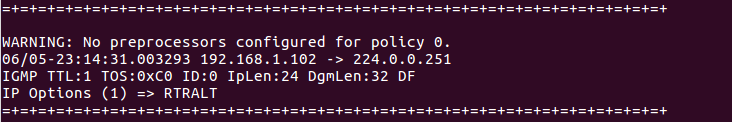
**使用Snort**

Snort有三个模式：

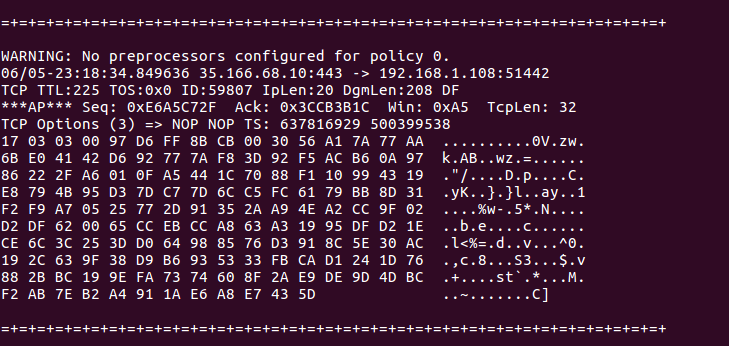
* Sniffer mode, which simply reads the packets off of the network and displays them for you in a continuous stream on the console (screen).
* Packet Logger mode, which logs the packets to disk.
* Network Intrusion Detection System (NIDS) mode, which performs detection and analysis on network traffic.

Sniffer模式：

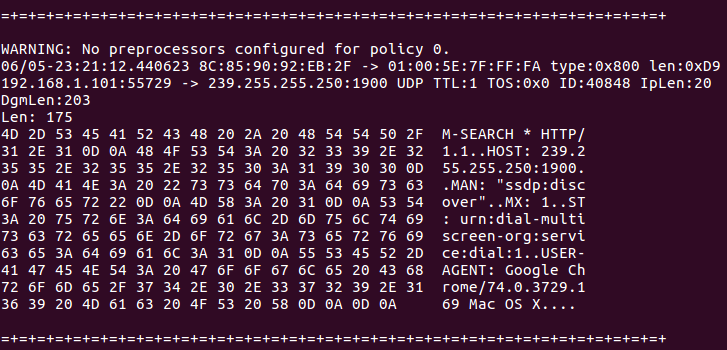
1. sudo snort -v //输出TCP/IP报文信息



1. sudo snort -vd // 输出IP 和TCP/UDP/ICMP报文信息

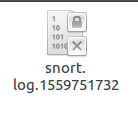


1. sudo snort -vde // 输出更详细的报文信息



Packet Logger模式：

1. 新建一个名为log的文件夹
2. 执行sudo snort -dev -l ./log 会把信息输出到相应的日志文件



1. 执行sudo snort -dev -l ./log -h 192.168.1.0/24过滤其他不相干的IP，只关注这一个IP的信息

NIDS模式：

1. 执行sudo snort -dev -l ./log -h 192.168.1.0/24 -c /etc/snort/snort.conf

网络攻击与防御技术实验报告

**姓名：周美廷 学号：76066002 所在班级：160611**

**实验名称：Nessus扫描工具的使用 实验日期：2019年6月5日**

**指导老师： 实验评分：**

**验收评语：**

**参与人员：**

**实验目的：**

通过 Nessus 扫描工具的使用，了解漏洞扫描的常用方法，掌握插件扫描的原理，了解针对扫描工具的防范措施。

**实验内容：**

（1）安装Nessus。

（2）配置Nessus，分配具体用户。

（3）用Nessus对局域网或者公网主机进行扫描。

（4）分析扫描结果报告，获取系统的有用信息，发现网络系统的安全漏洞。

（5）提出防范扫描工具的具体措施，并付诸实施。再用Nessus进行扫描，比较两次扫描结果的异同。

**实验环境：**

Windows 10 – 64 bit

**实验设计：**

使用的软件：

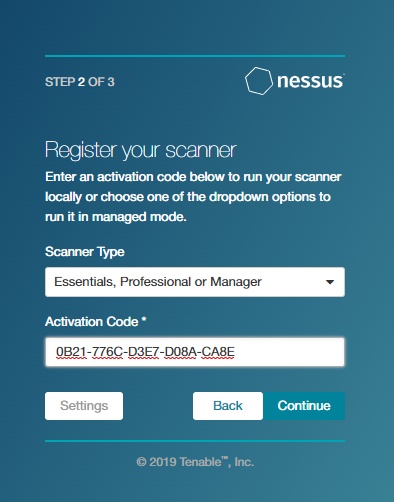
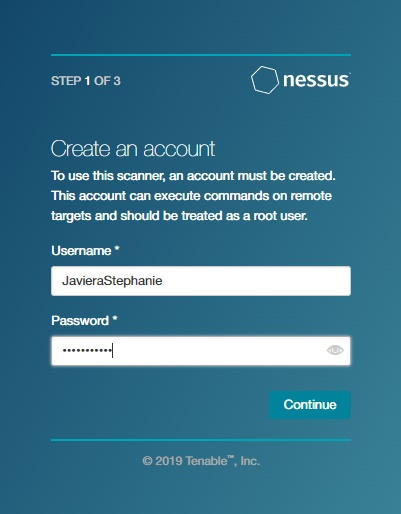
* Nessus

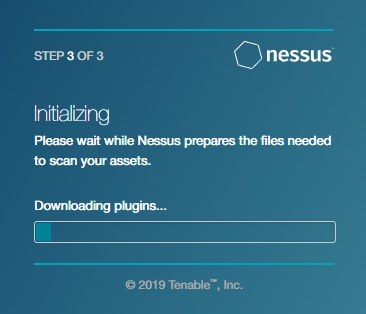
**详细过程：**

1. 从<https://www.tenable.com/downloads/nessus> 下载Nessus（同时有通过邮件收到activation code）
2. 安装Nessus -> Nessus-8.4.0-x64.msi

安装时有WinPcap的问题（我已安装的是Winpcap4.1.12而这版本的Nessus需要4.1.13，但是又不能直接替换，所以需要先到C:/Windows/SysWOW64 把packet.dll -> packet.dll.old, wpcap.dll -> wpcap.dll.old然后重新安装

1. 安装成功后，会自动打开localhost:8834配置用户等信息





1. 登录到Nessus