

## **Intrusion Detection Systems (IDS)**

### **Objective:**

The objective of this lab is to become familiar with Intrusion Detection Systems and how to install and configure them. In order to accomplish this, the Snort IDS will be installed with the Acid interface; once installed it will be tested by using an external security scanner to actively check for vulnerabilities. The IDS should then detect these attempts and display this in the Acid interface.

### **References:**

[http://www.snort.org/docs/snort\\_acid\\_rh9.pdf](http://www.snort.org/docs/snort_acid_rh9.pdf)  
<http://www.snort.org>  
<http://acidlab.sourceforge.net/>  
<http://php.net>

### **Equipment/Programs Used:**

Dell Pentium II Desktop  
GFI LanGuard Network Scanner  
Linux RedHat 9.0  
Snort/Acid and dependencies

### **Procedures:**

This section will describe the processes involved in getting a complete IDS running. It is written in such a way that a person should be able to follow the directions exactly as they are presented, and the system should work as expected. It is broken into the following sub-sections:

- Install RedHat 9
- Download all necessary packages
- Install zlib
- Install libpcap
- Install MySQL
- Install Apache
- Install PHP
- Install Snort
- Configure the Database
- Install JpGraph
- Install ADOdb
- Install ACID
- Test and Verify Functionality

## Install RedHat 9

This installation of an IDS is based on the RedHat 9 Linux distribution. Although an IDS will theoretically work on any Linux system, it is recommended that RH 9 be used in order to ensure the correctness of this document. All of the default options for the install should be selected, except for the following:

- Select 'custom' install
- Select 'no firewall'
- Select the following packages
  - Editors
    - Emacs
    - Text-based Internet
    - Lynx
  - Development Tools
- Unselect everything else

If the noted packages are installed as shown, then the installation should show a total of 338 packages.

## Download All Necessary Packages

Download the following (wget [url]) into the `/usr/local/src` directory:

- <http://www.snort.org/dl/snort-2.0.2.tar.gz>
- <http://www.apache.org/dist/httpd/httpd-2.0.47.tar.gz>
- <http://www.php.net/distributions/php-4.3.3.tar.gz>
- <http://phplens.com/lens/dl/adodb390.tgz>
- <http://acidlab.sourceforge.net/acid-0.9.6b23.tar.gz>
- <http://flow.dl.sourceforge.net/sourceforge/libpng/zlib-1.1.4.tar.gz>
- <http://www.aditus.nu/jpgraph/downloads/jpgraph-1.13.tar.gz>
- <http://www.tcpdump.org/release/libpcap-0.7.2.tar.gz>
- <http://www.mysql.com/get/Downloads/MySQL-4.0/mysql-4.0.17.tar.gz>
- <http://mysql.cyberservers.net/>
- <http://www.snort.org/dl/rules/snortrules-stable.tar.gz>

## Install zlib

To begin the installation process, first relocate to the source directory (`cd /usr/local/src`) and then execute the following commands:

```
tar -xvzf zlib-1.1.4.tar.gz
cd zlib-1.1.4
./configure; make test
make install
cd ..
```

## Install libpcap

To install libpcap execute the following commands (from the source directory):

```
tar -xvzf libpcap-0.7.2.tar.gz
cd libpcap-0.7.2
./configure
make
make install
cd ..
```

## Install MySQL

This step is the one that will take the longest. In order to make later configuration of MySQL more simple, first create a user and group to associate with MySQL:

```
groupadd mysql
useradd -g mysql mysql
```

Next edit the `/root/.bash_profile` so the `PATH` line is as follows:

```
PATH=$PATH:$HOME/bin:/usr/local/mysql/bin
```

/\*

NOTE:

To edit files, the preferred editor for this document was Emacs. To open a file with Emacs, the command is `emacs [filename]`. To save and close the file after editing the user can press the key sequence: `Ctrl-x, Ctrl-s, Ctrl-x, Ctrl-c`

\*/

Return to the `/usr/local/src` directory (`cd /usr/local/src`)

To install the MySQL package execute the following commands:

```
tar -xvzf mysql-4.0.17.tar.gz
cd mysql-4.0.17
./configure --prefix=/usr/local/mysql
make
make install
scripts/mysql_install_db
chown -R root /usr/local/mysql
chown -R mysql /usr/local/mysql/var
chgrp -R mysql /usr/local/mysql
cp support-files/my-medium.cnf /etc/my.cnf
```

Add the lines `"/usr/local/mysql/lib/mysql"` and `"/usr/local/lib"` to the `/etc/ld.so.conf` file; and then run the `ldconfig`: `ldconfig -v`. Then test to see that it worked by running the following command (you may have to type 'Enter' to get back to the command prompt) and ensuring that there are no errors:

```
/usr/local/mysql/bin/mysqld_safe -user=mysql &
```

Next, configure MySQL to start at boot time:

```
cp support-files/mysql.server /etc/init.d/mysql
cd /etc/rc3.d
ln -s ../init.d/mysql S85mysql
ln -s ../init.d/mysql K85mysql
```

```
cd /etc/rc5.d
ln -s ../init.d/mysql S85mysql
ln -s ../init.d/mysql K85mysql
cd ../init.d
chmod 755 mysql
```

## Install Apache

```
cd /usr/local/src
tar -xvzf httpd-2.0.47.tar.gz
cd httpd_2.0.47
./configure --enable-so
make
make install
cd ..
```

## Install PHP

```
tar -xvzf php-4.3.3.tar.gz
cd php-4.3.3
./configure --enable-sockets --with-
apxs2=/usr/local/apache2/bin/apxs --with-mysql=/usr/local/mysql
make
make install
```

It is also necessary to configure Apache to parse certain extensions as PHP. In order to do this, edit the `/usr/local/apache2/conf/httpd.conf` file to have the following line:

```
AddType application/x-httpd-php .php .html
```

Also, include `index.php` in the types of files to use as a default index page. This is also done in the `httpd.conf` file. Be sure to uncomment the `ServerName` variable and set it to the ip address of the external NIC.

Finally, configure Apache to start at system boot:

```
cp /usr/local/apache2/bin/apachectl /etc/init.d/httpd
cd /etc/rc3.d
ln -s ../init.d/httpd S85httpd
ln -s ../init.d/httpd K85httpd
cd /etc/rc5.d
ln -s ../init.d/httpd S85httpd
ln -s ../init.d/httpd K85httpd
```

At this point, it may be good to restart the computer to ensure that the services are starting correctly at boot time.

## Install Snort

Begin by adding a user and group to associate with Snort:

```
groupadd snort
useradd -g snort snort
```

Complete the installation by executing the following commands:

```
mkdir /etc/snort
mkdir /var/log/snort
tar -xvzf snort-2.0.2.tar.gz
cd snort-2.0.2
./configure --with-mysql=/usr/local/mysql
make
make install
```

Now configure the rules for snort:

```
cd ..
tar xvzf snortrules-stable.tar.gz
cd rules
cp * /etc/snort
```

Next, Snort must be configured to work on the local network. This is done by editing the `/etc/snort/snort.conf` file. For the network this installation was completed on, the following configurations were changed:

```
var HOME_NET $eth0_ADDRESS,10.10.10.0/24
(the first portion is to tell it to listen to everything
attacking eth0, but also to listen to anything on the internal
addressing system 10.10.10.0/24)
var RULE_PATH /etc/snort
Output database: log, mysql, user=snort password=[password]
bname=snort host=localhost
```

The following lines of the `snort.conf` file must also be changed:

```
CONFIG=/etc/snort/snort.conf
SNORT_GID=snort
```

Finally, configure Snort to start at boot:

```
cd ../snort-2.0.2/
cp contrib/S99snort /etc/init.d/snort
cd /etc/init.d
chmod 755 snort
cd /etc/rc3.d
ln -s ../init.d/snort S99snort
ln -s ../init.d/snort K99snort
cd /etc/rc5.d
ln -s ../init.d/snort S99snort
ln -s ../init.d/snort K99snort
```

## Configure the Database

The following commands must be executed in SQL. To open SQL execute the following: `/usr/local/mysql/bin/mysql`. Once SQL has started, execute the following commands within SQL:

```
>SET PASSWORD FOR root@localhost=PASSWORD('honeybot');
>create database snort;
>grant INSERT,SELECT on root.* to snort@localhost;
>SET PASSWORD FOR snort@localhost=PASSWORD('honeybot');
```

```
>grant CREATE, INSERT, SELECT, DELETE, UPDATE on snort.* to
snort@localhost;
>grant CREATE, INSERT, SELECT, DELETE, UPDATE on snort.* to
snort;
>exit
```

Now run the Snort script to create the database tables:

```
cd /usr/local/src/snort-2.0.2
/usr/local/mysql/bin/mysql -u root -p < ./contrib./create_mysql
snort
(The password is the one that was specified, in this example,
'honeypot')
cd contrib
zcat snortdb-extra.gz | /usr/local/mysql/bin/mysql -p snort
(The password is the one that was specified, in this example,
'honeypot')
```

Check to make sure that the script ran correctly:

```
/usr/local/mysql/bin/mysql -p
(Enter password)
>SHOW DATABASES;
```

You should see the following

```
+-----+
| Database
+-----+
|mysql
|snort
|test
+-----+
```

```
>use snort
>SHOW TABLES;
```

It should list the following tables:

```
data
detail
encoding
event
flags
...
and more
...
```

## Install JPGraph

To install JPGraph, execute the following commands:

```
cd /usr/local/src/
cp jpgraph-1.13.tar.gz /usr/local/apache2/htdocs
cd /usr/local/apache2/htdocs
tar -xvzf jpgraph-1.13.tar.gz
```

```
rm -rf jpgraph-1.13.tar.gz
cd jpgraph-1.13
rm -rf README
rm -rf QPL.txt
```

## Install ADODB

To install ADODB execute the following commands:

```
cd /usr/local/src/
cp adodb390.tgz /usr/local/apache2/htdocs/
cd /usr/local/apache2/htdocs
tar -xvzf adodb390.tgz
rm -rf adodb390.tgz
```

## Install Acid

To install Acid execute the following commands:

```
cd /usr/local/src
cp acid-0.9.6b23.tar.gz /usr/local/apache2/htdocs
cd /usr/local/apache2/htdocs
tar xvzf acid-0.9.6b23.tar.gz
rm -rf acid-0.9.6b23.tar.gz
```

To configure Acid, relocate to the `/usr/local/apache2/htdocs/acid` directory and edit the `acid_conf.php` to match the following lines:

```
$DBlib_path = "/usr/local/apache2/htdocs/adodb";
$alert_dbname = "snort";
$alert_user = "snort";
$alert_password = "honeypot";
$archive_dbname = "snort";
$archive_user = "snort";
$archive_password = "honeypot";
$ChartLib_path = "/usr/local/apache2/htdocs/jpgraph-1.13/src";
```

## Test and Verify Functionality

In order to verify that the services start correctly, reboot the machine and watch to make sure there are no errors. Once the machine has restarted, on another machine connected to the same network, open a browser and go to the following URL (replacing the [ipaddress] with the IP address of the IDS): [http://\[ipaddress\]/acid/acid\\_main.php](http://[ipaddress]/acid/acid_main.php). Then click on the “setup page” hyperlink to run the database setup script. Once this has completed, click the button that says “create acid ag”. Return to the previous URL and verify that the “Analysis Console for Intrusion Detection” page is displayed.

Next, install some sort of security scanner on an external machine, and run a security scan against the IDS. This should probe for several security vulnerabilities, and in the process should set off some alerts on the IDS. Verify that this does indeed set off some alerts, and this indicates that the IDS is working properly. The next important step would be to configure the rules to apply more specifically to the needs of the network.

## Results:

The Intrusion Detection System was successfully installed by following the instructions as outlined in the procedures section of the document. Several screenshots are shown below to indicate some of the alerts displayed by the IDS and some of the functionality. These were taken after running the IDS for several days.

**Analysis Console for Intrusion Databases**

Added 0 alert(s) to the Alert cache

Queried on: Mon February 02, 2004 20:41:12  
 Database: snort@localhost (schema version: 106)  
 Time window: [2004-01-18 02:03:00] - [2004-02-02 17:57:24]

**Sensors: 1**  
**Unique Alerts: 8 ( 5 categories )**  
**Total Number of Alerts: 117**

- Source IP addresses: 40
- Dest. IP addresses: 2
- Unique IP links: 11
- Source Ports: 38
  - TCP (38) UDP (0)
- Dest. Ports: 3
  - TCP (3) UDP (0)

**Traffic Profile by Protocol**

TCP (100%)

UDP (0%)

ICMP (0%)

Portscan Traffic (0%)

- Search
- Graph Alert data
- Snapshot
  - Most recent Alerts: any protocol, TCP, UDP, ICMP
  - Today's alerts unique, listing: IP src / dst
  - Last 24 Hours: alerts unique, listing: IP src / dst
  - Last 72 Hours: alerts unique, listing: IP src / dst
  - Most recent 15 Unique Alerts
  - Last Source Ports: any, TCP, UDP
  - Last Destination Ports: any, TCP, UDP
- Most frequent 5 Alerts
- Most frequent Source Ports: any, TCP, UDP
- Most frequent Destination Ports: any, TCP, UDP
- Most frequent 15 addresses: source, destination

Graph alert detection time

Alert Group (AG) maintenance

Application cache and status

[Loaded in 4 seconds]

ACID v0.9.6a23 (by Roman Danylyk as part of the AircERT project)

**Alert Listing**

Added 0 alert(s) to the Alert cache

Queried on: Mon February 02, 2004 20:51:53

Alert Criteria: any  
 IP Criteria: any  
 Layer 3 Criteria: none  
 Payload Criteria: any

Displaying alerts 1-8 of 8 total

<input type="checkbox"/>	Signature	Classification	Total #	Sensor #	Src. Addr.	Dest. Addr.	First	Last
<input type="checkbox"/>	nessus[sever] WEB-MISC robots.txt access	web-application-activity	25 (20%)	1	27	1	2004-01-29 08:14:02	2004-02-02 08:59:22
<input type="checkbox"/>	[bugtraq][sever] MAP login literal buffer overflow attempt	misc-attack	12 (10%)	1	3	1	2004-01-29 12:58:25	2004-02-02 12:11:33
<input type="checkbox"/>	[cve][ca][bugtraq][sever] WEB-PHP directory.php access	misc-attack	52 (44%)	1	5	1	2004-01-31 18:21:53	2004-02-02 17:57:24
<input type="checkbox"/>	[arachnIDS][sever] WEB-MISC http directory traversal	attempted-recon	1 (1%)	1	1	1	2004-01-30 09:45:27	2004-01-30 09:45:27
<input type="checkbox"/>	[cve][ca][bugtraq][cve][ca][bugtraq][sever] WEB-MISC Chunked-Encoding transfer attempt	web-application-attack	5 (4%)	1	3	1	2004-01-29 03:19:55	2004-02-02 16:53:33
<input type="checkbox"/>	urfnessus[sever] WEB-IDS nrislog.dll access	web-application-activity	1 (1%)	1	1	1	2004-02-02 00:58:00	2004-02-02 00:58:00
<input type="checkbox"/>	[sever] WEB-IDS cmd.exe access	web-application-attack	1 (1%)	1	1	1	2004-01-30 22:18:56	2004-01-30 22:18:56
<input type="checkbox"/>	[sever] FTP invalid MODE	protocol-command-decode	10 (8%)	1	1	1	2004-01-18 02:03:00	2004-01-18 02:11:46

[ action ] [ Selected ] ALL on Screen

[Loaded in 3 seconds]

ACID v0.9.6a23 (by Roman Danylyk as part of the AircERT project)

## **Conclusions:**

An Intrusion Detection System can be a vital element of a network. It can help to prevent attacks by showing alerts when attempts to compromise the network are made, and will help to alert an administrator when a possible intrusion has occurred. Some of the more useful features provided in this IDS is the fact that alerts can be stored and categorized; also, the fact that alerts can automatically be sent to an administrator is also a very important feature. Another feature that might be useful to add would be some sort of policy infraction detection. This could be done by implementing an inline sniffer to analyze packets and display information as it goes across the network. This could display things such as file sharing or messenger programs to detect employees or users if they are breaking the Acceptable Use Agreement.

Obviously, if the IDS is compromised by an attacker, then it would be possible for the attacker to easily hide detection by altering the alerts in the database. Thus it is important to truly secure the IDS. One way to do this would be to actually cut or disconnect the “send” wires of the Ethernet cable. By doing this, it becomes impossible for an attacker to even detect the machine let alone compromise it. The “send” capability of the NIC is not needed, since the system only passively listens to the traffic on the network. The IDS does not need an IP address, and can possibly even be put into a bridged configuration to further avoid detection.

In conclusion, this lab was very successful in teaching the student about the process of installing and configuring an Intrusion Detection System. It is very clear that this can be a powerful way to monitor the network for malicious or harmful activity. An IDS is something that every network administrator should consider installing on every network under their stewardship.