CHAPTER 6

Using ACID and SnortSnarf with Snort

Analysis Console for Intrusion Databases (ACID) is a tool used to analyze and present Snort data using a web interface. It is written in PHP. It works with Snort and databases like MySQL, as you have learned in the last chapter, and makes information available in the database to the user through a web server. In addition to Snort, the tool can be used with other security-related products like firewalls and networking monitoring.

This chapter provides information about ACID and discusses how to install it with MySQL and Snort to view and analyze the intrusion detection data logged by Snort into the database. You will go through a step-by-step procedure to install ACID and use it. The graphical representation of captured data is very useful for analysis purposes.

In addition to ACID, the chapter also provides basic information about SnortSnarf, another tool that can be used with a web server. SnortSnarf is able to parse Snort log files and generate HTML pages that can be viewed using a web browser. I assume that you are able to install and run Apache web server as well as MySQL database server, which are required in order to use the tools discussed in this chapter.

6.1 What is ACID?

ACID consists of many Pretty Home Page (PHP) scripts and configuration files that work together to collect and analyze information from a database and present it through a web interface. A user will use a web browser to interact with ACID. You have to have a web server, database server, PHP and some other tools installed on your system to make it work. For the sake of this book, I am using a RedHat Linux 7.1 machine. I have installed Apache web server, PHP, and MySQL, which are part of the RedHat distribution. The database is configured to work with Snort as explained in Chapter 5. The latest version of ACID is available from http://www.cert.org/kb/acid/.

ACID offers many features:

1. Searching can be done on a large number of criteria like source and destination addresses, time, ports and so on, as shown in Figure 6-7.
2. Packet viewing is used to view different parts of packet. You can view different header parts as well as the payload. Refer to Figure 6-6 for an example of ICMP packet.
3. Alerts can be managed by creating alert classes, exporting and deleting and sending them to an e-mail address.
4. Graphical representation includes charts based upon time, protocol, IP addresses, port numbers and classifications.
5. Snapshots can be taken of the alerts database. As an example, you can view alerts for the last 24 hours, unique alerts, frequent alerts and so on. Refer to Figure 6-7 for detail on snapshots.
6. You can go to different whois databases on the Internet to find out who owns a particular IP address that is attacking your network. You can then contact the responsible person to stop it. The whois database contains information about owners of domain names and IP addresses.

All of these facilities are available through the web browser. You point the web browser to a URL to access ACID screens. For example, I can use http://www.conformix.com/acid/ on my intranet site to view logs. The web pages are written in PHP. Support packages like GD library and PHPLOT are used to print graphs on the web pages. PHP connects to the backend MySQL database to get and update data. For this purpose, you have to provide the database user name and password.

The big picture of the whole system including Snort, MySQL, Web server, PHP and web browser is shown in Figure 1-1 in Chapter 1. The following is a brief, step-by-step description of what happens when an intruder attempts to get into your network.

- An intruder tries to get into your network.
- A Snort sensor installed in your network detects intruder activity based on its rules. It then uses information in the `snort.conf` file to log data into MySQL database. You have to provide the database user name, password, hostname or IP address of the database server and database name in `snort.conf` file.
- A web server is installed where MySQL server is running.
- A user starts the browser, connects to the web server and starts requesting PHP web pages.
- The PHP engine connects to the database using the database user name, password, and database name and gets information from the database server.
- The web server processes this information and sends back a reply to the web browser, where a user can view intrusion data.
- A user can then perform different operations on this data via the web pages.

The rest of this chapter describes how to install and configure all of these tools to build a web-based user interface.

6.2 Installation and Configuration

Since ACID needs additional packages, like PHPLOT, GD library and so on, to work, you need to make sure that everything is installed properly. Fortunately you can install different components independently from each other in no particular order. The following step-by-step process makes it easy to put everything in place.

- Install and test Snort. You have already done it in Chapter 2.
- Install and test MySQL. Please see Chapter 5 for reference. After installing MySQL, you have to create a database and tables so that Snort can log its activity into the database. After that you have to configure Snort using `snort.conf` file so that it logs its data to the database server.
- Install Apache. I would suggest using the RPM package that is part of RedHat installation media. You can also download the latest version of Apache web server from http://www.apache.org.
- Download ACID from http://www.cert.org/kb/acid/ and uncompress it in `/var/www/html` directory. This process creates a directory named `acid` under `/var/www/html` directory. The Apache package that is part of the RedHat distribution has its HTML files under `/var/www/html` directory. Depending on your distribution, the directory may be different on your

machine. If you download Apache in source code form and compiled it yourself, you can choose a particular directory for this purpose during the compilation process. Just keep in mind that you have to install ACID under the directory where Apache is looking for HTML files.

- Get and Install PHP. You can download it from http://www.php.net or you can use the RPM package that is part of the RedHat distribution. Set `display_errors` variable in `/etc/php.ini` to Off. If you are using a precompiled or RPM version of Apache, PHP may already have been built into it as a module.
- Get and install GD library from http://www.boutell.com/gd/. This is also available on RedHat installation CDs in the RPM form and I would recommend using the RPM file. It is installed as `/usr/lib/libgd.so` file.
- Download PHPLOT from http://www.phplot.com and uncompress it in `/var/www/html` directory. This is used to create graphics in the web pages.
- Download ADODB from http://php.weblogs.com/adodb and install it in `/var/www/html` directory. ADODB is an object oriented library written in PHP and is used to connect to the database. ADODB Frequently Asked Questions (FAQ) are available at http://php.weblogs.com/adodb_faq.

Let us carry out the process of installing these components. At this point I assume that you have:

- Installed MySQL database server as discussed in the last chapter.
- Installed and configured Snort so that it logs data into the Snort database.
- Installed Apache, GD library, and PHP as part of RedHat Linux installation.

Now download and install the software as mentioned below:

- Download ACID file `acid-0.9.6b21.tar.gz` from http://www.cert.org/kb/acid/ and put it in `/opt` directory.
- Download ADODB file `adodb221.tgz` from http://php.weblogs.com/adodb and put it in `/opt` directory.
- Download PHPLOT file `phplot-4.4.6.tar.gz` from http://www.phplot.com and put it in `/opt` directory.
- Move to `/var/www/html` directory.
- Use the command "`tar zxvf /opt/acid-0.9.6b21.tar.gz`." This will create a directory /var/www/html/acid and put all ACID files under it.

- Use the cd command to go to /var/www/html/acid directory.
- Use the command "tar zxvf /opt/adodb221.tgz" to extract ADODB files. The command will create a directory /var/www/html/acid/adodb and put all ADODB files under this directory.
- Use the command "tar zxvf /opt/phplot-4.4.6.tar.gz" to extract PHPLOT files. This will create a directory /var/www/html/acid/phplot-4.4.6 and put all PHPLOT files under this directory.
- Create another database snort_archive using "create database snort_archive;" command after starting mysql client using the procedure described in Chapter 5. You have already created a database with the name "snort" and a user with the name "rr" as discussed in Chapter 5. The new snort_archive database is used by ACID to archive old data. The new database is not required by Snort to log data. If you don't want to archive old data using ACID, you can skip this step and the next step as well.
- Grant permissions to user rr to manage snort_archive database using the command "grant CREATE,INSERT,DELETE,UPDATE,SELECT on snort_archive.* to rr@localhost;".
- Create tables in this database using the command "mysql -u rr -p snort_archive <contrib/create_mysql" as described in Chapter 5.
- Set display_errors variable in /etc/php.ini to Off.

Now you have to configure ACID so that it can interact with the MySQL database. The configuration process also enables Snort to use the PHPLOT package. The configuration process is simple and includes setting up different parameters in the acid_conf.php configuration file which is located in the same directory where you uncompressed the ACID files. For the examples in this book, the file is located in the /var/www/html/acid directory. You have to put information about the following items in this file:

- Location of ADODB files. In our case this path is ./adodb. This is because all ADODB files are located in adodb directory under the directory where ACID files are located.
- Type of database server. For the example in this book the type of server is "mysql".
- MySQL database name for Snort log data.
- MySQL database server name or IP address.
- MySQL database user name and password.

- Name of the archive database if you are using one.
- Database server name where archive database is located. In our case both snort and snort_archive databases are located on localhost.
- Database user name and password to access snort_archive database.
- Location of PHPLOT files. In our case this is ./phplot-4.4.6. This is because all PHPLOT files are located in phplot-4.4.6 directory under the directory where ACID files are located.

This information is present in the start of the acid_conf.php file. The typical opening lines of this file in my installation are as follows:

```
<?php

$ACID_VERSION = "0.9.6b21";

/* Path to the DB abstraction library
 *  (Note: DO NOT include a trailing backslash after the
 *   directory)
 *   e.g. $foo = "/tmp"      [OK]
 *        $foo = "/tmp/"     [OK]
 *        $foo = "c:\tmp"    [OK]
 *        $foo = "c:\tmp\"   [WRONG]
 */
$DBlib_path = "./adodb";

/* The type of underlying alert database
 *
 *  MySQL       : "mysql"
 *  PostgresSQL : "postgres"
 *  MS SQL Server : "mssql"
 */
$DBtype = "mysql";

/* Alert DB connection parameters
 *   - $alert_dbname   : MySQL database name of Snort
                       : alert DB
 *   - $alert_host     : host on which the DB is stored
 *   - $alert_port     : port on which to access the DB
 *   - $alert_user     : login to the database with
                       : this user
 *   - $alert_password : password of the DB user
 *
 *  This information can be gleaned from the Snort database
 *  output plugin configuration.
 */
```

```
$alert_dbname   = "snort";
$alert_host     = "localhost";
$alert_port     = "";
$alert_user     = "rr";
$alert_password = "rr78x";

/* Archive DB connection parameters */
$archive_dbname   = "snort_archive";
$archive_host     = "localhost";
$archive_port     = "";
$archive_user     = "rr";
$archive_password = "rr78x";

/* Type of DB connection to use
 *   1  : use a persistant connection (pconnect)
 *   2  : use a normal connection (connect)
 */
$db_connect_method = 1;

/* Path to the graphing library
 *  (Note: DO NOT include a trailing backslash after the
directory)
 */
$ChartLib_path = "./phplot-4.4.6";
```

Note that you have used the same user name, password, and database name as we used in `snort.conf` file. The following is a description of data located in the `acid_conf.php` file.

The following line in `acid_conf.php` file sets up the location of ADODB files:

```
$DBlib_path = "./adodb";
```

The following line in `acid_conf.php` file sets up the type of database:

```
$DBtype = "mysql";
```

The following lines in `acid_conf.php` file set up the main Snort database information where Snort logs its data:

```
$alert_dbname   = "snort";
$alert_host     = "localhost";
$alert_port     = "";
$alert_user     = "rr";
$alert_password = "rr78x";
```

The following lines in `acid_conf.php` file set up archive database information where ACID archives data. This part is not necessary for Snort or ACID operation. It is required only if you want to archive data using ACID.

```
$alert_dbname   = "snort_archive";
$alert_host     = "localhost";
$alert_port     = "";
$alert_user     = "rr";
$alert_password = "rr78x";
```

The following line in `acid_conf.php` file sets up the location of PHPLOT files.

```
$ChartLib_path = "./phplot-4.4.6";
```

After going through this practice, make sure that Snort, MySQL server, and Apache server are running. Now you are ready to start using the web interface of ACID.

6.3 Using ACID

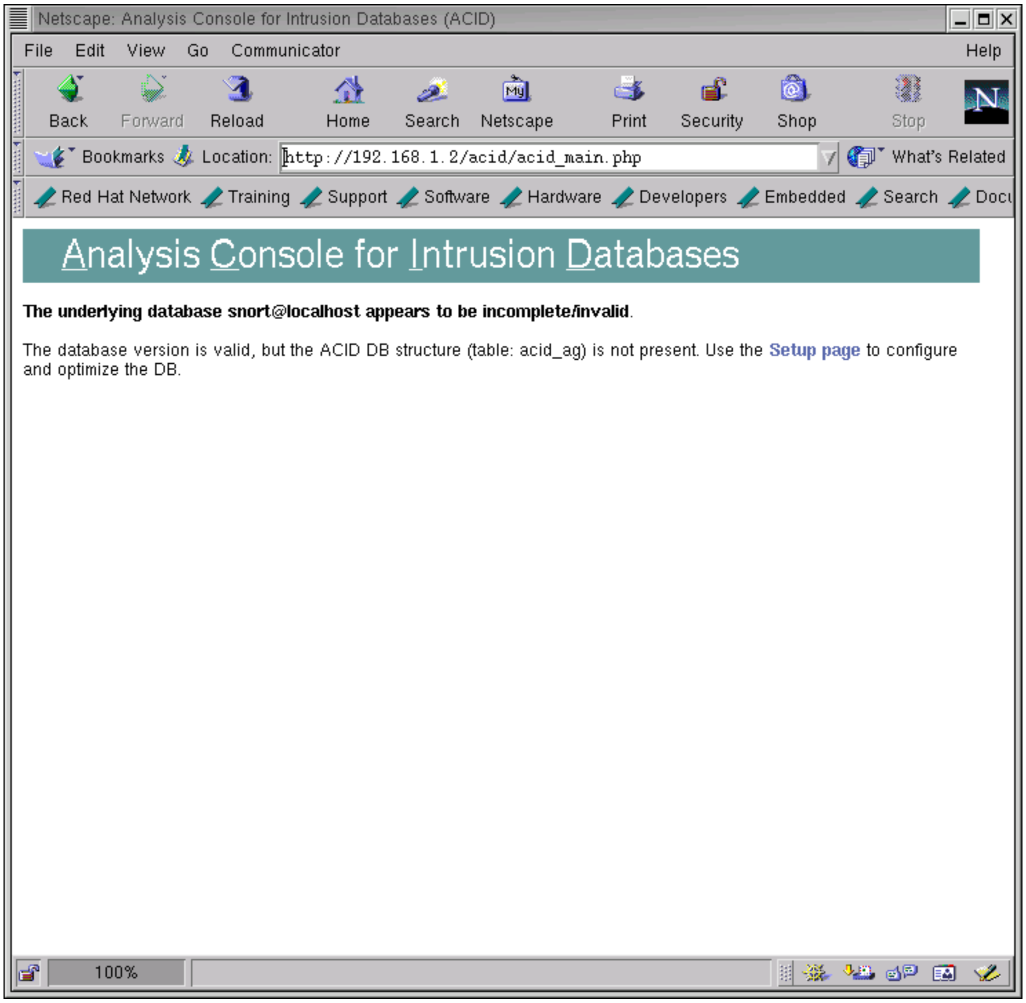
If you have installed everything as mentioned above, you should be able to access ACID by going to URL http:/<your_web_server>/acid/. My web server is running on IP address 192.168.1.2, so I can go the URL http://192.168.1.2/acid/.

The first time you go to this URL, ACID needs to do some setup tasks and you will see a web window like the one shown in Figure 6-1.

At this screen, click the Setup page link and you will move to the DB Setup page shown in Figure 6-2.

In Figure 6-2, click the “Create ACID AG” link so that ACID can create its own table to support Snort. ACID creates its own tables in the main Snort database and uses these tables for its own housekeeping data. More discussion about ACID tables is presented later in this chapter. Figure 6-3 shows the result of creating these new tables.

As shown in Figure 6-3, you can click the “Main Page” link towards the bottom of the page to go to the main ACID page. Web pages shown in Figures 6-1, 6-2 and 6-3 will not be displayed the next time you start using ACID.

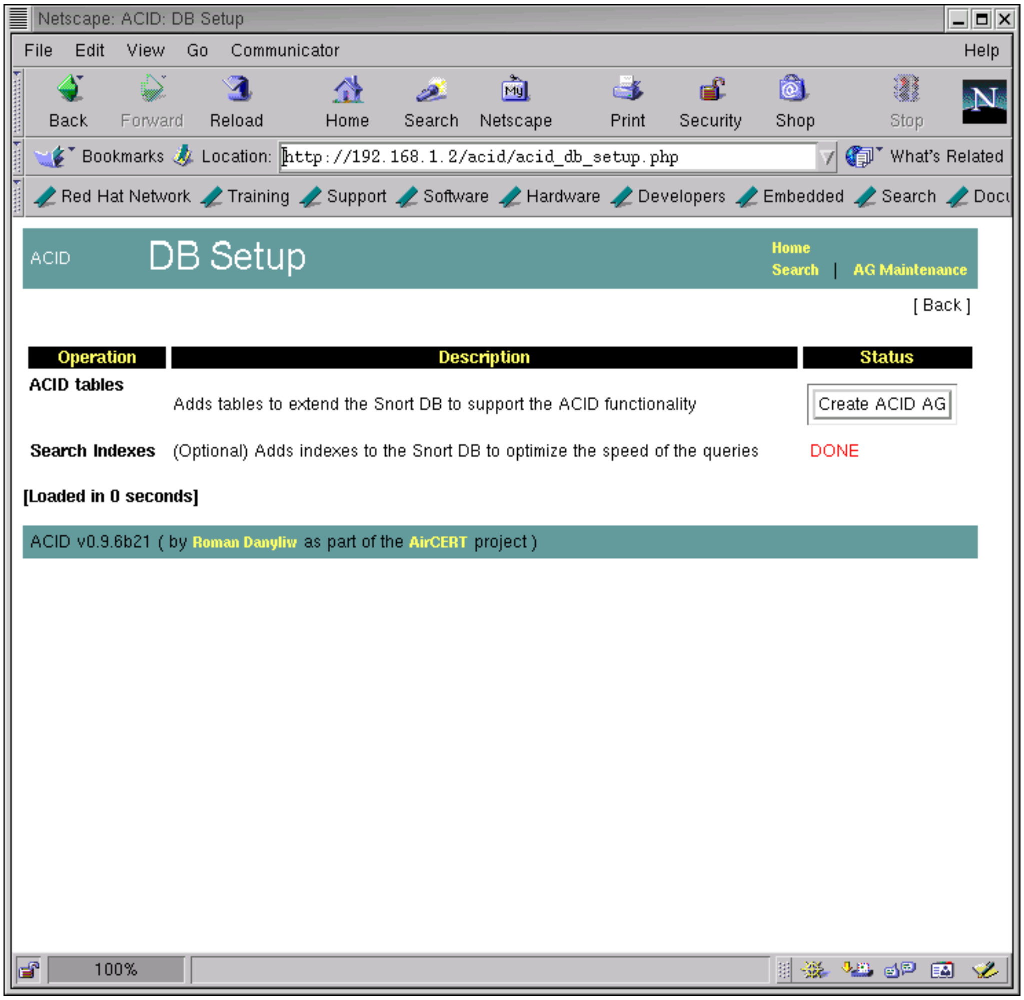

Netscape: Analysis Console for Intrusion Databases (ACID)
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: http://192.168.1.2/acid/acid_main.php What's Related
Red Hat Network Training Support Software Hardware Developers Embedded Search Docu
100%

Analysis Console for Intrusion Databases

The underlying database snort@localhost appears to be incomplete/invalid.

The database version is valid, but the ACID DB structure (table: acid_ag) is not present. Use the **Setup page** to configure and optimize the DB.

Figure 6-1 Invoking ACID for the first time.


Netscape: ACID: DB Setup
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location:
What's Related
Red Hat Network Training Support Software Hardware Developers Embedded Search Docu
100%

Location: http://192.168.1.2/acid/acid_db_setup.php

ACID **DB Setup**

Home
Search | AG Maintenance

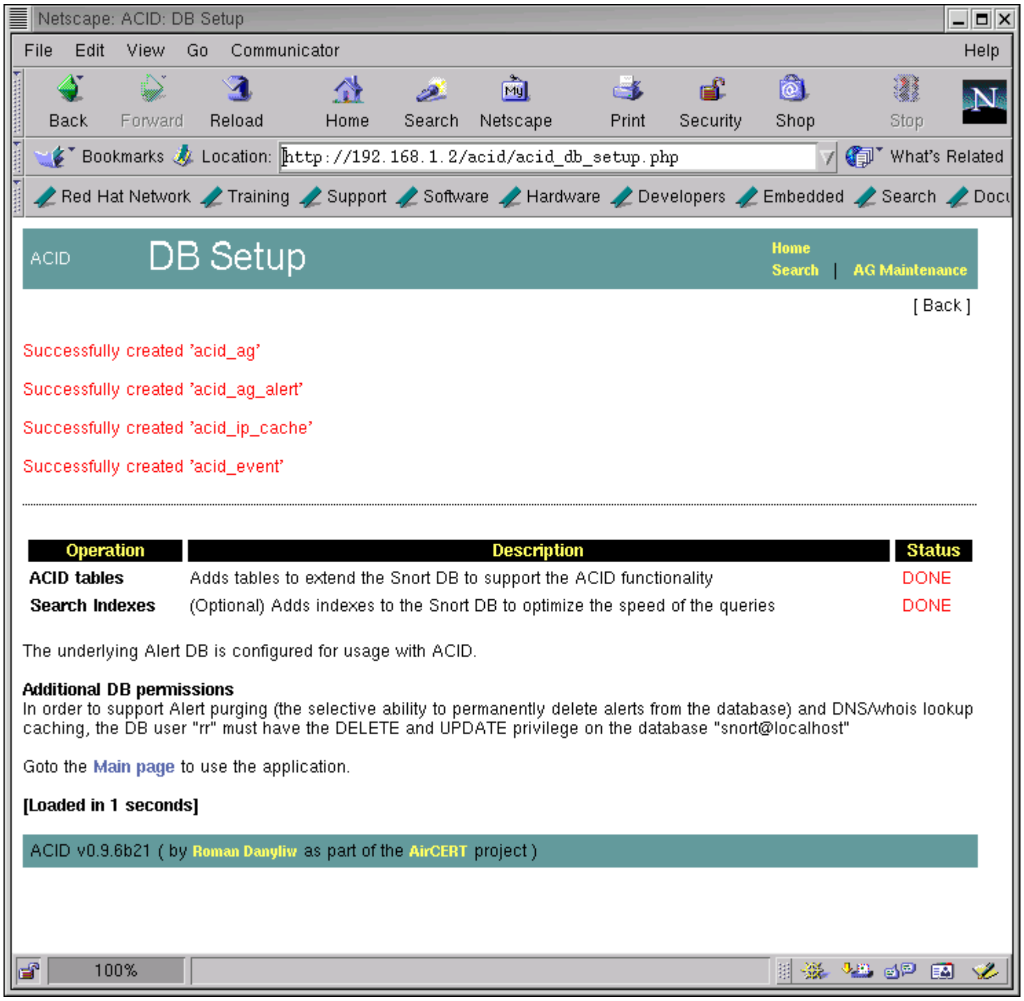
[Back]

Operation	Description	Status
ACID tables	Adds tables to extend the Snort DB to support the ACID functionality	Create ACID AG
Search Indexes	(Optional) Adds indexes to the Snort DB to optimize the speed of the queries	DONE

[Loaded in 0 seconds]

ACID v0.9.6b21 (by Roman Danyliw as part of the AirCERT project)

Figure 6-2 Creating ACID tables to existing database.


Netscape: ACID: DB Setup
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: http://192.168.1.2/acid/acid_db_setup.php
What's Related
Red Hat Network Training Support Software Hardware Developers Embedded Search Docu
ACID
DB Setup
Home
Search | AG Maintenance
[Back]
100%

Successfully created 'acid_ag'

Successfully created 'acid_ag_alert'

Successfully created 'acid_ip_cache'

Successfully created 'acid_event'

Operation	Description	Status
ACID tables	Adds tables to extend the Snort DB to support the ACID functionality	DONE
Search Indexes	(Optional) Adds indexes to the Snort DB to optimize the speed of the queries	DONE

The underlying Alert DB is configured for usage with ACID.

Additional DB permissions
In order to support Alert purging (the selective ability to permanently delete alerts from the database) and DNS/whois lookup caching, the DB user "rr" must have the DELETE and UPDATE privilege on the database "snort@localhost"

Goto the Main page to use the application.

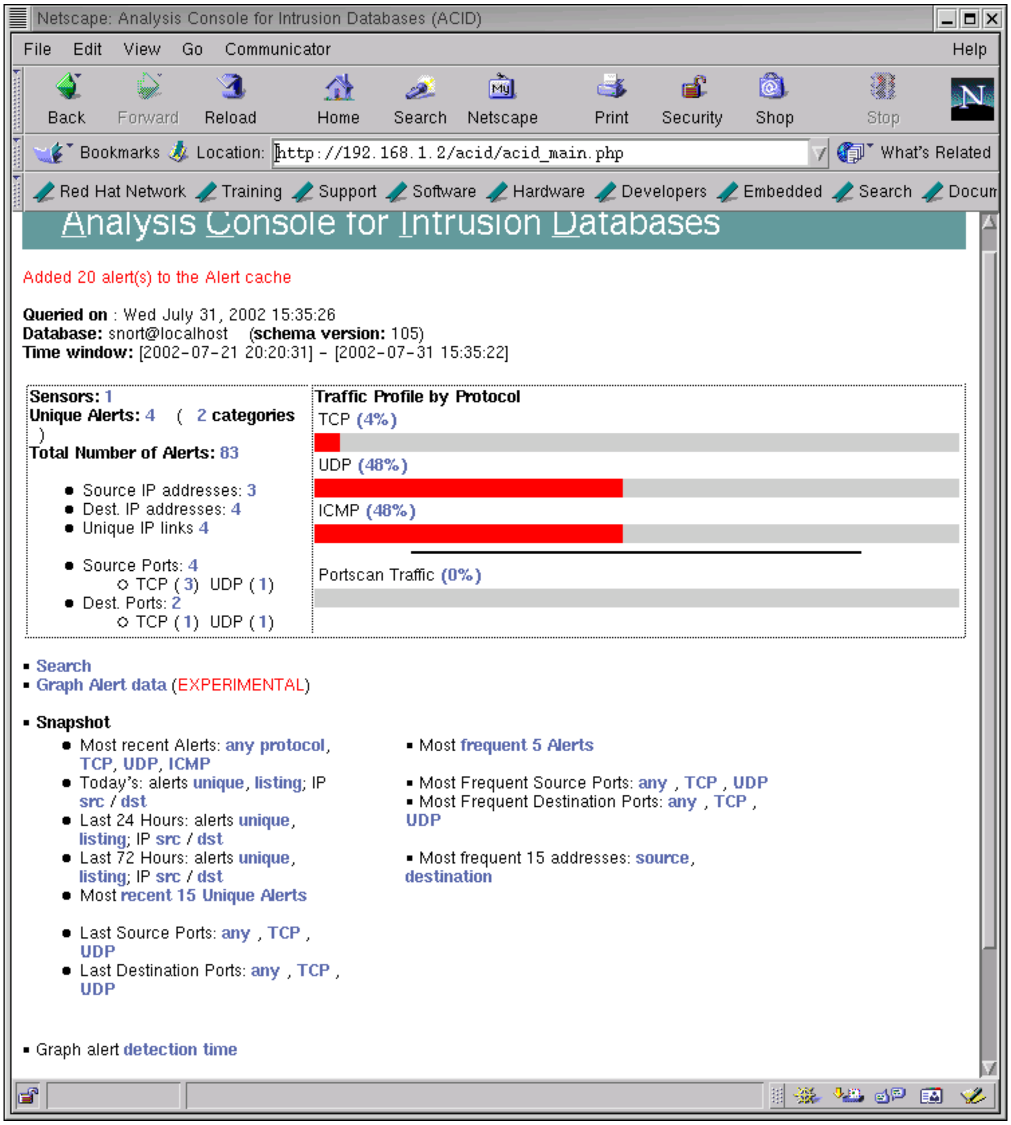
[Loaded in 1 seconds]

ACID v0.9.6b21 (by Roman Danyliw as part of the AirCERT project)

Figure 6-3 The result of creating additional tables in the Snort database to support ACID.

6.3.1 ACID Main Page

The ACID main page provides an overview of currently available data. It has different sections to display information in groups. You can view traffic profiles by different protocols, get a snapshot of sensors, search data and so on, as shown in Figure 6-4. You are encouraged to explore the different links found on this page.


Netscape: Analysis Console for Intrusion Databases (ACID)
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: What's Related
Red Hat Network Training Support Software Hardware Developers Embedded Search Docum

Location: http://192.168.1.2/acid/acid_main.php

Analysis Console for Intrusion Databases

Added 20 alert(s) to the Alert cache

Queried on : Wed July 31, 2002 15:35:26
Database: snort@localhost (**schema version:** 105)
Time window: [2002-07-21 20:20:31] - [2002-07-31 15:35:22]

Sensors: 1
Unique Alerts: 4 (2 **categories**)
Total Number of Alerts: 83

- Source IP addresses: 3
- Dest. IP addresses: 4
- Unique IP links 4

- Source Ports: 4
  - TCP (3) UDP (1)
- Dest. Ports: 2
  - TCP (1) UDP (1)

Traffic Profile by Protocol

TCP (4%)

UDP (48%)

ICMP (48%)

Portscan Traffic (0%)

- Search
- Graph Alert data (EXPERIMENTAL)

- **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

Figure 6-4 ACID main page.

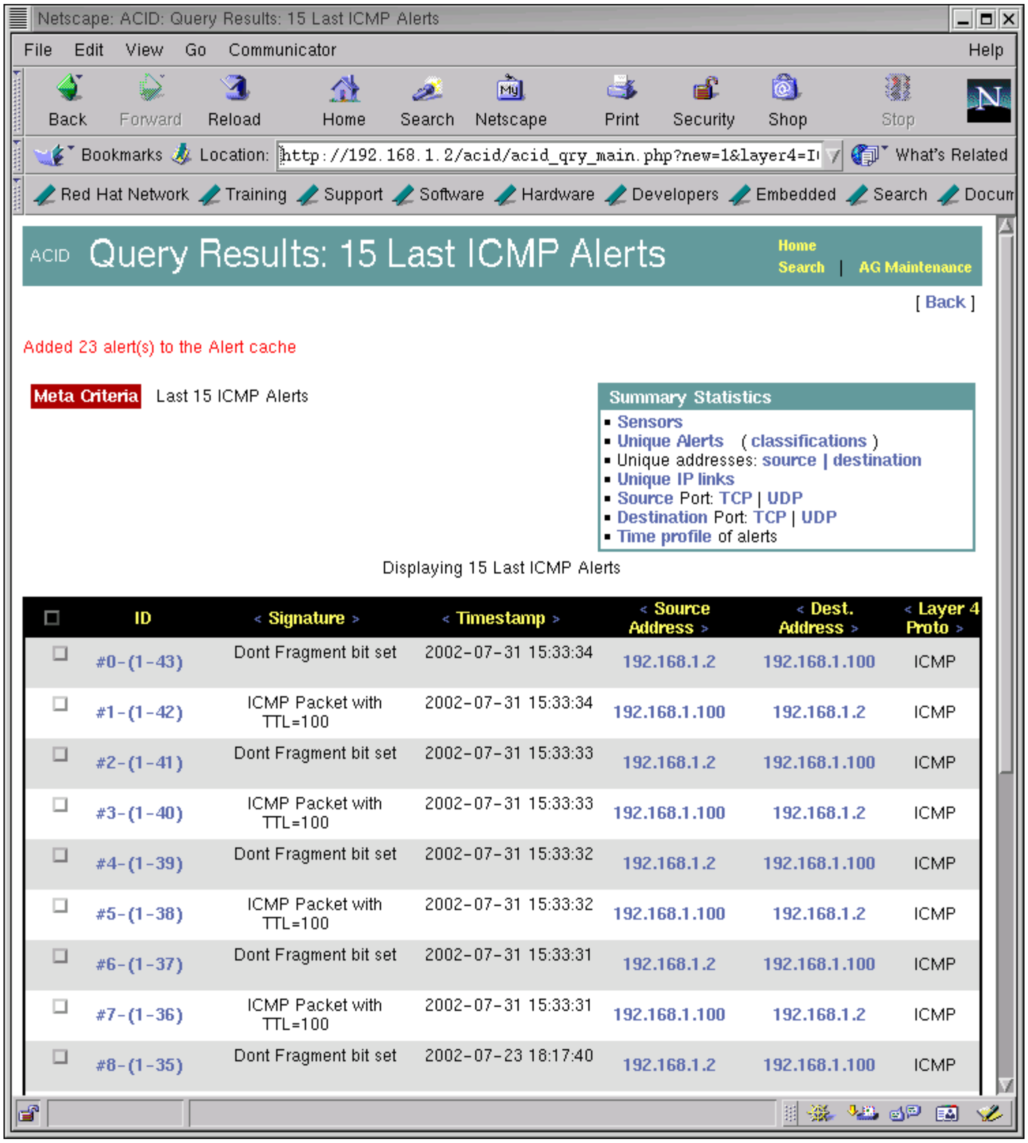
By clicking different links on the web page shown in Figure 6-4, you can view a great deal of information.

- List of sensors that are logging data to the database.
- Number of unique alerts and their detail.
- Total number of alerts and their detail.
- Source IP addresses for the captured data. This shows who is trying to hack into your network. By following the subsequent links, you can also find the owner of the source IP address by looking up whois databases.
- Destination IP addresses for captured data.
- Source and destination ports.
- Alerts related to a particular protocol, like TCP alerts, UDP alerts and ICMP alerts.
- Search alert and log data for particular entries.
- Most frequent alerts.
- Plot alert data, which is still experimental.

In the following screen shots, you will learn a few important things. But this is just an overview of what ACID can do for you. The more time you spend using ACID, the more you will learn about different methods of analyzing Snort data. As you learn new things, you will appreciate how arranging Snort data in different ways makes a lot more sense compared to just looking at log files.

6.3.2 Listing Protocol Data

From the main page, you can click on a protocol to get information about packets logged for that particular protocol. Figure 6-5 shows a screen shot for ICMP protocol. The bottom part of the screen shows the last fifteen individual packets that have been logged into the database. You can click on any one of these lines at the bottom to find out more details about a particular packet.



Location: http://192.168.1.2/acid/acid_qry_main.php?new=1&layer4=I

ACID Query Results: 15 Last ICMP Alerts

Home
Search | AG Maintenance

[Back]

Added 23 alert(s) to the Alert cache

Meta Criteria Last 15 ICMP Alerts

Summary Statistics

- Sensors
- Unique Alerts (classifications)
- Unique addresses: source | destination
- Unique IP links
- Source Port: TCP | UDP
- Destination Port: TCP | UDP
- Time profile of alerts

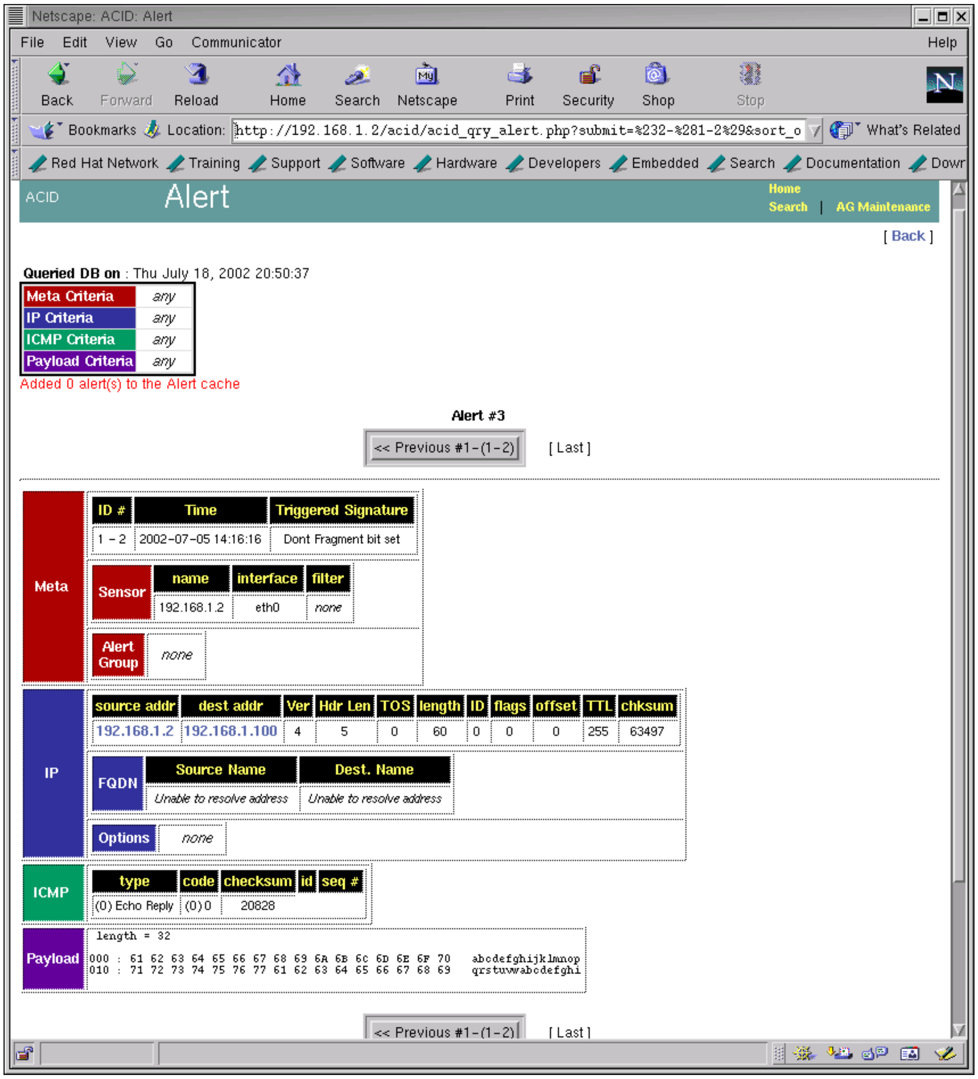
Displaying 15 Last ICMP Alerts

□	ID	< Signature >	< Timestamp >	< Source Address >	< Dest. Address >	< Layer 4 Proto >
□	#0-(1-43)	Dont Fragment bit set	2002-07-31 15:33:34	192.168.1.2	192.168.1.100	ICMP
□	#1-(1-42)	ICMP Packet with TTL=100	2002-07-31 15:33:34	192.168.1.100	192.168.1.2	ICMP
□	#2-(1-41)	Dont Fragment bit set	2002-07-31 15:33:33	192.168.1.2	192.168.1.100	ICMP
□	#3-(1-40)	ICMP Packet with TTL=100	2002-07-31 15:33:33	192.168.1.100	192.168.1.2	ICMP
□	#4-(1-39)	Dont Fragment bit set	2002-07-31 15:33:32	192.168.1.2	192.168.1.100	ICMP
□	#5-(1-38)	ICMP Packet with TTL=100	2002-07-31 15:33:32	192.168.1.100	192.168.1.2	ICMP
□	#6-(1-37)	Dont Fragment bit set	2002-07-31 15:33:31	192.168.1.2	192.168.1.100	ICMP
□	#7-(1-36)	ICMP Packet with TTL=100	2002-07-31 15:33:31	192.168.1.100	192.168.1.2	ICMP
□	#8-(1-35)	Dont Fragment bit set	2002-07-23 18:17:40	192.168.1.2	192.168.1.100	ICMP

Figure 6-5 ICMP protocol data.

6.3.3 Alert Details

Figure 6-6 shows details about a particular ICMP packet that you would see when you click on an alert as shown in Figure 6-5. As you can see, there are different sections on the page. Each section displays a particular layer of the data packet. The topmost section provides general information about the alert. The IP section displays all parts of the IP header. The ICMP header displays ICMP data, followed by the payload. Payload is displayed both in hexadecimal and ASCII text. Refer to Appendix C for information about different protocol headers.



http://192.168.1.2/acid/acid_qry_alert.php?submit=%232-%281-2%29&sort_o

Queried DB on : Thu July 18, 2002 20:50:37

Meta Criteria	*any*
IP Criteria	*any*
ICMP Criteria	*any*
Payload Criteria	*any*

Added 0 alert(s) to the Alert cache

Alert #3

Meta

ID #	Time	Triggered Signature
1 - 2	2002-07-05 14:16:16	Dont Fragment bit set

Sensor	name	interface	filter
	192.168.1.2	eth0	*none*

Alert Group	*none*

IP

source addr	dest addr	Ver	Hdr Len	TOS	length	ID	flags	offset	TTL	chksum
192.168.1.2	192.168.1.100	4	5	0	60	0	0	0	255	63497

FQDN	Source Name	Dest. Name
	Unable to resolve address	*Unable to resolve address*

Options	*none*

ICMP

type	code	checksum	id	seq #
(0) Echo Reply	(0) 0	20828		

Payload

```
 length = 32

000 : 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70    abcdefghijklmnop
010 : 71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69    qrstuvwabcdefghi
```

Figure 6-6 Alert detail.

Navigation buttons are provided in this window that can be used to move to next and previous alerts. Different colors are used to indicate different headers of the packet, which makes it very easy to understand visually.

6.3.4 Searching

One important feature of ACID is that it can be used to search the captured log and alert data based on parameters such as:

- A particular sensor when you are using a central database to log data from many Snort sensors.
- Time of alert using start and ending time. This is very useful if you want to look at alerts that occurred within a specific period of time.
- Source and destination addresses.
- Different fields in the IP packet header.
- Transport layer protocols.
- String of data in the payload area of the IP packet.

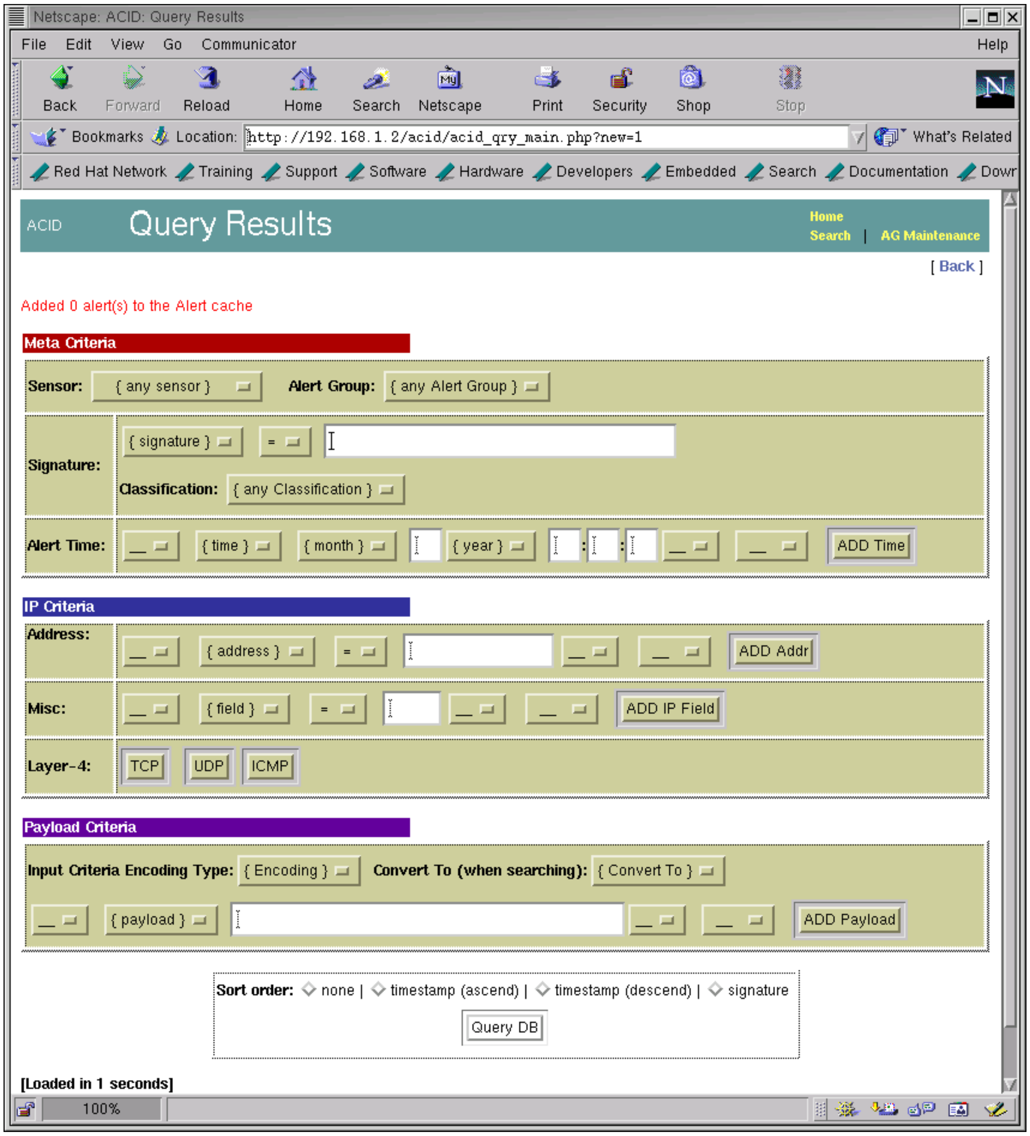
If you look at the screen shot shown in Figure 6-7, you can see that searching for data in the database is very easy. All the criteria that you specify in this screen are translated to a SQL statement that is passed to the MySQL database server. Results of your query are displayed when you click the “Query DB” button.

For example, if you want to search all alerts for which the signature field contains the string “ATTACK RESPONSE”, you can fill out information as shown in Figure 6-8.

The result of this search is shown in Figure 6-9, where all alerts containing this string are displayed. You can click a particular alert line to find out more information about that alert.

I would strongly recommend spending some time with the search methods of ACID to get acquainted to it.

Snort can also be used to find fully qualified names for source and destination addresses found in captured data. Figure 6-10 shows unique destination IP addresses and hostnames. For the sake of this screen shot and to create some data in the database, I had to use a rule that creates an alert for all outgoing HTTP requests. Of course it is not intrusion activity, but it does provide some data in the Snort database.



Netscape: ACID: Query Results

Location: http://192.168.1.2/acid/acid_qry_main.php?new=1

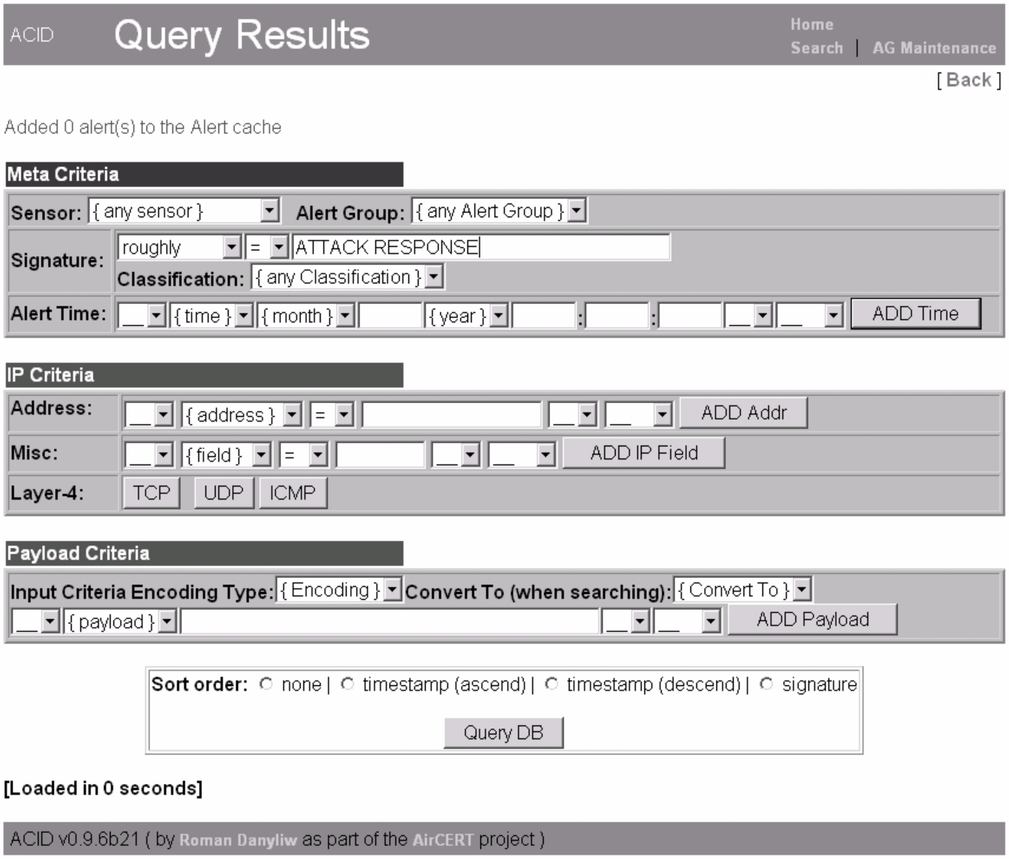
ACID Query Results

Added 0 alert(s) to the Alert cache

[Loaded in 1 seconds]

File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks What's Related
Red Hat Network Training Support Software Hardware Developers Embedded Search Documentation Dowr
Home
Search | AG Maintenance
[Back]
Meta Criteria
Sensor: { any sensor } Alert Group: { any Alert Group }
Signature: { signature } =
Classification: { any Classification }
Alert Time: { time } { month } { year } ADD Time
IP Criteria
Address: { address } = ADD Addr
Misc: { field } = ADD IP Field
Layer-4: TCP UDP ICMP
Payload Criteria
Input Criteria Encoding Type: { Encoding } Convert To (when searching): { Convert To }
{ payload } ADD Payload
Sort order: ◇ none | ◇ timestamp (ascend) | ◇ timestamp (descend) | ◇ signature
Query DB
100%

Figure 6-7 Searching database using ACID.



Added 0 alert(s) to the Alert cache

Meta Criteria

Sensor: { any sensor } Alert Group: { any Alert Group }

Signature: roughly = ATTACK RESPONSE

Classification: { any Classification }

Alert Time: __ { time } { month } { year } : : __ __

IP Criteria

Address: __ { address } = __ __

Misc: __ { field } = __ __

Layer-4:

Payload Criteria

Input Criteria Encoding Type: { Encoding } Convert To (when searching): { Convert To }

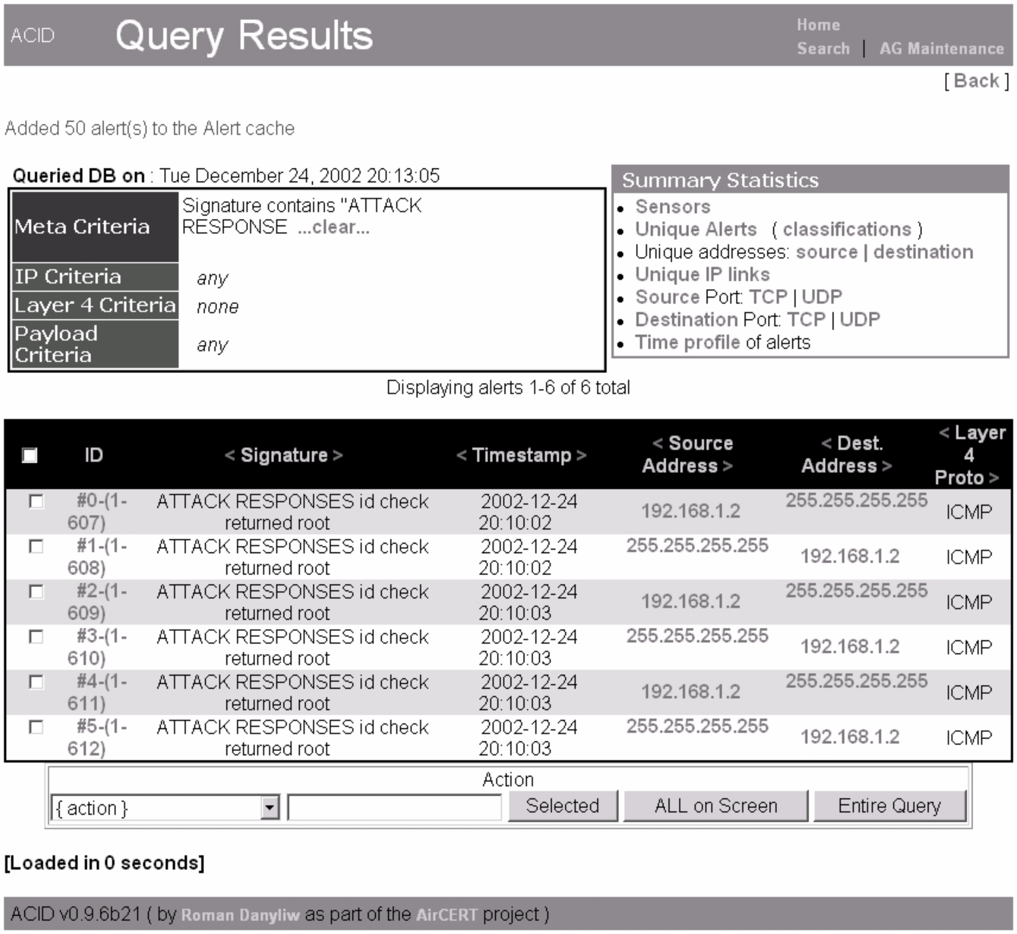
__ { payload } __ __

Sort order: ○ none | ○ timestamp (ascend) | ○ timestamp (descend) | ○ signature

[Loaded in 0 seconds]

ACID v0.9.6b21 (by Roman Danyliw as part of the AirCERT project)

Figure 6-8 Searching for all alerts that contain “ATTACK RESPONSE” string in the signature.



Query Results

Added 50 alert(s) to the Alert cache

Queried DB on : Tue December 24, 2002 20:13:05

Meta Criteria	Signature contains "ATTACK RESPONSE ...clear...
IP Criteria	*any*
Layer 4 Criteria	*none*
Payload Criteria	*any*

Summary Statistics

- Sensors
- Unique Alerts (classifications)
- Unique addresses: source | destination
- Unique IP links
- Source Port: TCP | UDP
- Destination Port: TCP | UDP
- Time profile of alerts

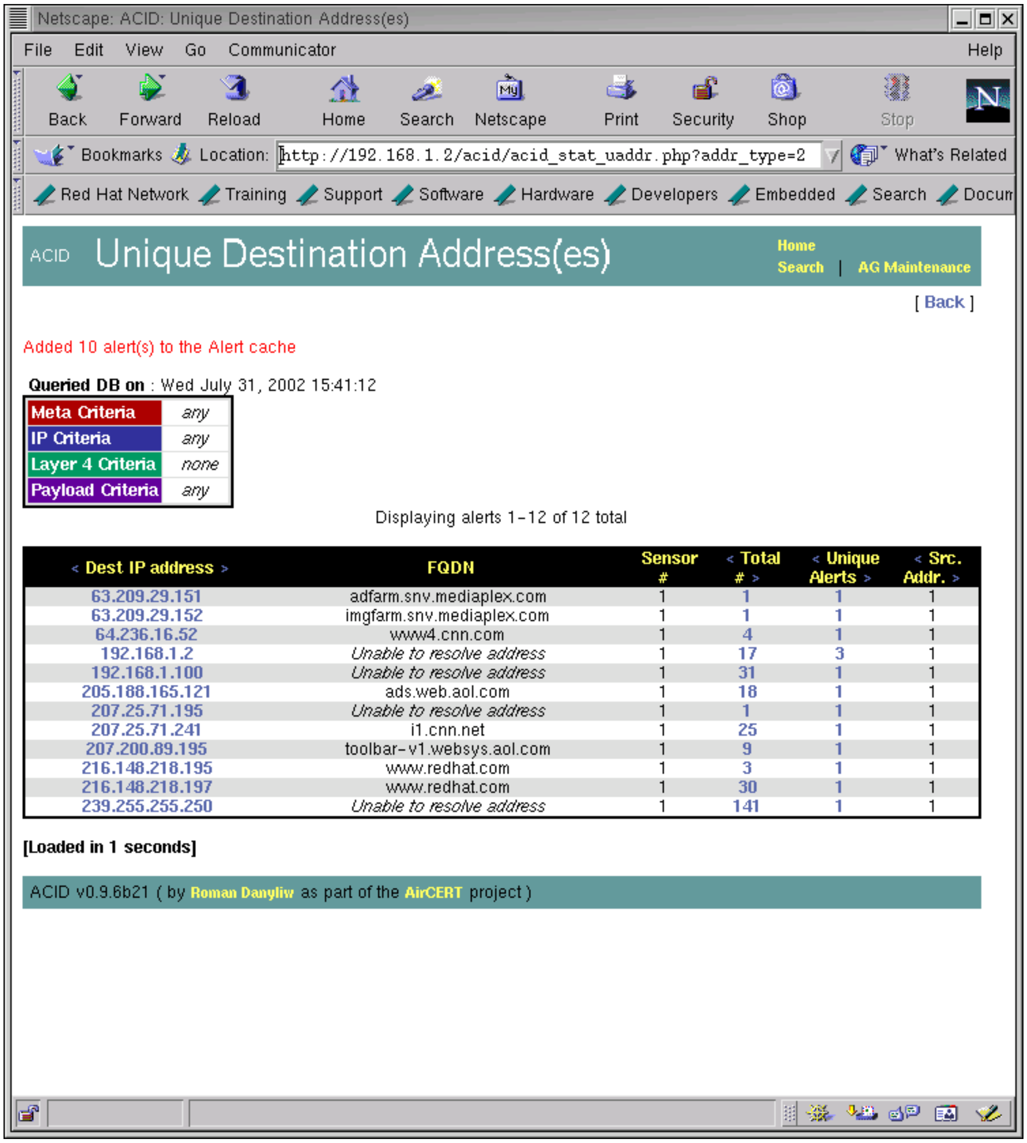
Displaying alerts 1-6 of 6 total

☐	ID	< Signature >	< Timestamp >	< Source Address >	< Dest. Address >	< Layer 4 Proto >
☐	#0-(1-607)	ATTACK RESPONSES id check returned root	2002-12-24 20:10:02	192.168.1.2	255.255.255.255	ICMP
☐	#1-(1-608)	ATTACK RESPONSES id check returned root	2002-12-24 20:10:02	255.255.255.255	192.168.1.2	ICMP
☐	#2-(1-609)	ATTACK RESPONSES id check returned root	2002-12-24 20:10:03	192.168.1.2	255.255.255.255	ICMP
☐	#3-(1-610)	ATTACK RESPONSES id check returned root	2002-12-24 20:10:03	255.255.255.255	192.168.1.2	ICMP
☐	#4-(1-611)	ATTACK RESPONSES id check returned root	2002-12-24 20:10:03	192.168.1.2	255.255.255.255	ICMP
☐	#5-(1-612)	ATTACK RESPONSES id check returned root	2002-12-24 20:10:03	255.255.255.255	192.168.1.2	ICMP

[Loaded in 0 seconds]

ACID v0.9.6b21 (by Roman Danyliw as part of the AirCERT project)

Figure 6-9 Result of query used in Figure 6-8.



Netscape: ACID: Unique Destination Address(es)
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: http://192.168.1.2/acid/acid_stat_uaddr.php?addr_type=2 What's Related
Red Hat Network Training Support Software Hardware Developers Embedded Search Docum
Home
Search | AG Maintenance
[Back]

ACID Unique Destination Address(es)

Added 10 alert(s) to the Alert cache

Queried DB on : Wed July 31, 2002 15:41:12

Meta Criteria	*any*
IP Criteria	*any*
Layer 4 Criteria	*none*
Payload Criteria	*any*

Displaying alerts 1-12 of 12 total

< Dest IP address >	FQDN	Sensor #	< Total # >	< Unique Alerts >	< Src. Addr. >
63.209.29.151	adfarm.snv.mediaplex.com	1	1	1	1
63.209.29.152	imgfarm.snv.mediaplex.com	1	1	1	1
64.236.16.52	www4.cnn.com	1	4	1	1
192.168.1.2	*Unable to resolve address*	1	17	3	1
192.168.1.100	*Unable to resolve address*	1	31	1	1
205.188.165.121	ads.web.aol.com	1	18	1	1
207.25.71.195	*Unable to resolve address*	1	1	1	1
207.25.71.241	i1.cnn.net	1	25	1	1
207.200.89.195	toolbar-v1.websys.aol.com	1	9	1	1
216.148.218.195	www.redhat.com	1	3	1	1
216.148.218.197	www.redhat.com	1	30	1	1
239.255.255.250	*Unable to resolve address*	1	141	1	1

[Loaded in 1 seconds]

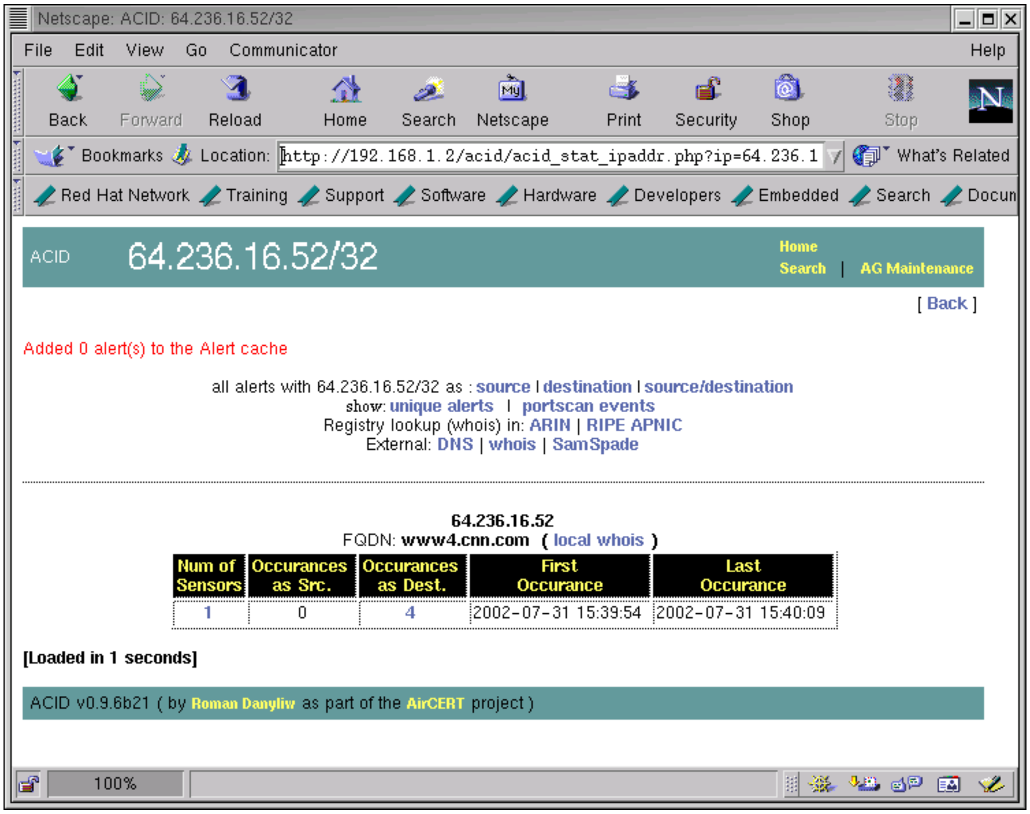
ACID v0.9.6b21 (by Roman Danyliw as part of the AirCERT project)

Figure 6-10 Unique destination addresses for alerts in Snort database.

6.3.5 Searching whois Databases

To get whois information about a particular address, you can click on any address and select a particular whois database, like American Registry for Internet Numbers (ARIN) at http://www.arin.net. The response to such a query for IP address 66.236.16.52 is shown in Figure 6-11.

This information is very important for incident response. This is usually the first step to finding out the owner of the attacking IP address and his/her contact number. After finding this information, you can contact the owner to stop bad guys from probing your network.



Location: http://192.168.1.2/acid/acid_stat_ipaddr.php?ip=64.236.1

ACID 64.236.16.52/32

Home
Search | AG Maintenance

[Back]

Added 0 alert(s) to the Alert cache

all alerts with 64.236.16.52/32 as : source | destination | source/destination
show: unique alerts | portscan events
Registry lookup (whois) in: ARIN | RIPE APNIC
External: DNS | whois | SamSpade

64.236.16.52
FQDN: www4.cnn.com (local whois)

Num of Sensors	Occurances as Src.	Occurances as Dest.	First Occurance	Last Occurance
1	0	4	2002-07-31 15:39:54	2002-07-31 15:40:09

[Loaded in 1 seconds]

ACID v0.9.6b21 (by Roman Danyliw as part of the AirCERT project)

Figure 6-11 Response to whois query.

6.3.6 Generating Graphs

Generating graphs is still experimental in ACID at the time of writing this book. I have included it for the sake of introducing this interesting feature. You can go to the ACID main page where a link is provided to generate graphs. When generating graphs, you can select data and type of graph. For example, you can generate a line or bar graph for alerts in the last five days. Figure 6-12 shows a sample bar graph for the alert data.

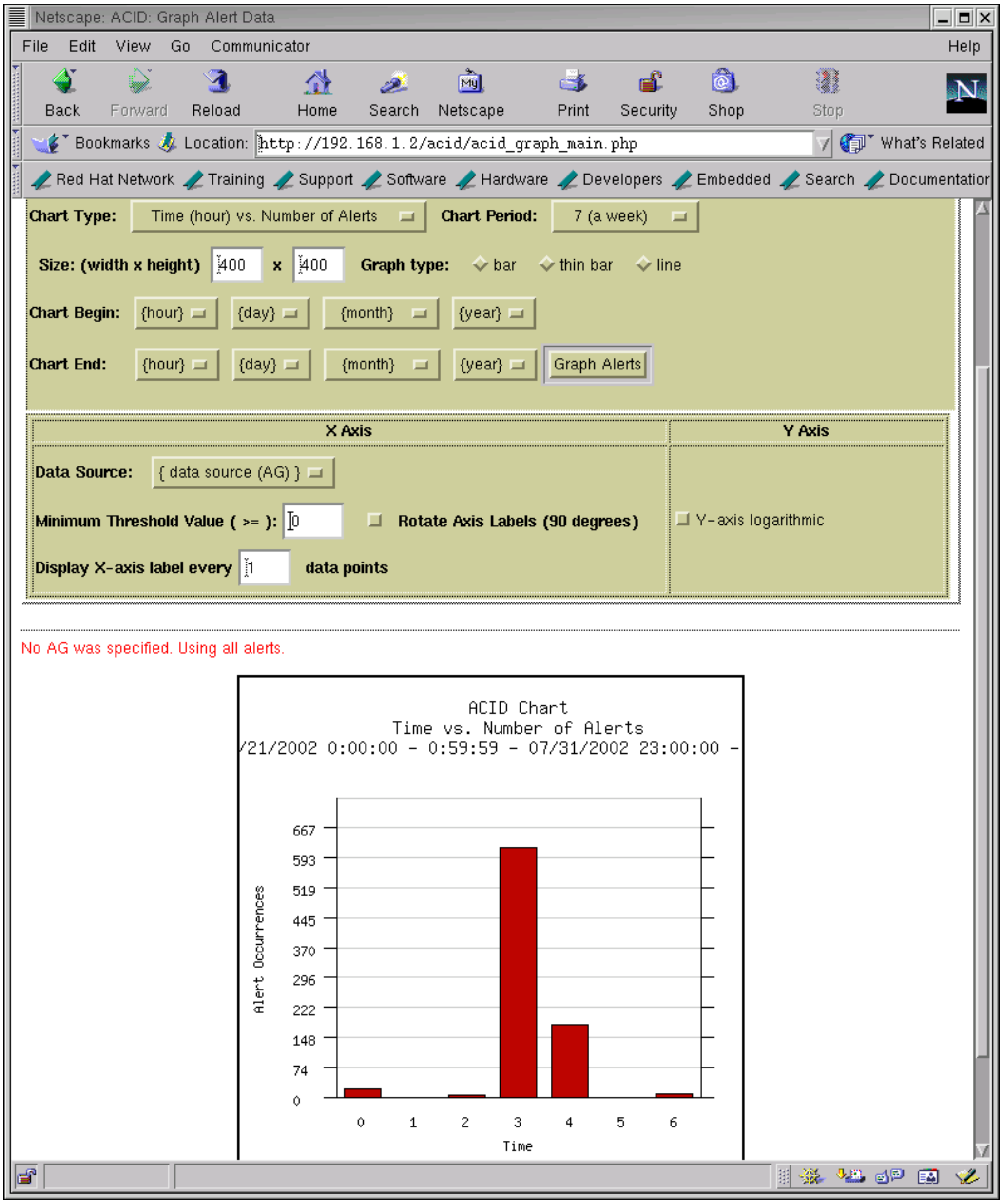
ACID uses the PHPLOT package on the backend side to generate these graphs. You can also use another package, JPGRAPH in place of PHPLOT. JPGRAPH has a different licensing scheme and there may be some restrictions for using it in commercial environment.

NOTE The functionality described in this section is just an overview of ACID capabilities. In addition to the tasks presented here, you can also use ACID to archive data, delete data from the database and so on.

6.3.7 Archiving Snort Data

You have created a new database called `snort_archive` in the previous sections to archive the data from the main Snort database. Using ACID, you can either move alerts from the main database to the archive database or just copy them. For example, if you want to move all alerts from the main database to the archive database, click the number next to “Total Number of Alerts” on the main ACID page. The next page displays all of the alerts in the database. If the number of alerts is more than 50, then only the first 50 alerts are displayed. Now you can use the bottom part of the screen to archive the alerts as shown in Figure 6-13. Note that only the bottom part of the browser window is shown in this figure.

If you click the “Entire Query” button in Figure 6-13, all alerts will be moved to the archive database. The result of this action is shown in Figure 6-14.



Netscape: ACID: Graph Alert Data
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: http://192.168.1.2/acid/acid_graph_main.php What's Related
Red Hat Network Training Support Software Hardware Developers Embedded Search Documentation
Graph Alerts
X Axis
Y Axis

Chart Type: Time (hour) vs. Number of Alerts Chart Period: 7 (a week)

Size: (width x height) 400 x 400 Graph type: ◇ bar ◇ thin bar ◇ line

Chart Begin: {hour} {day} {month} {year}

Chart End: {hour} {day} {month} {year}

Data Source: { data source (AG) }

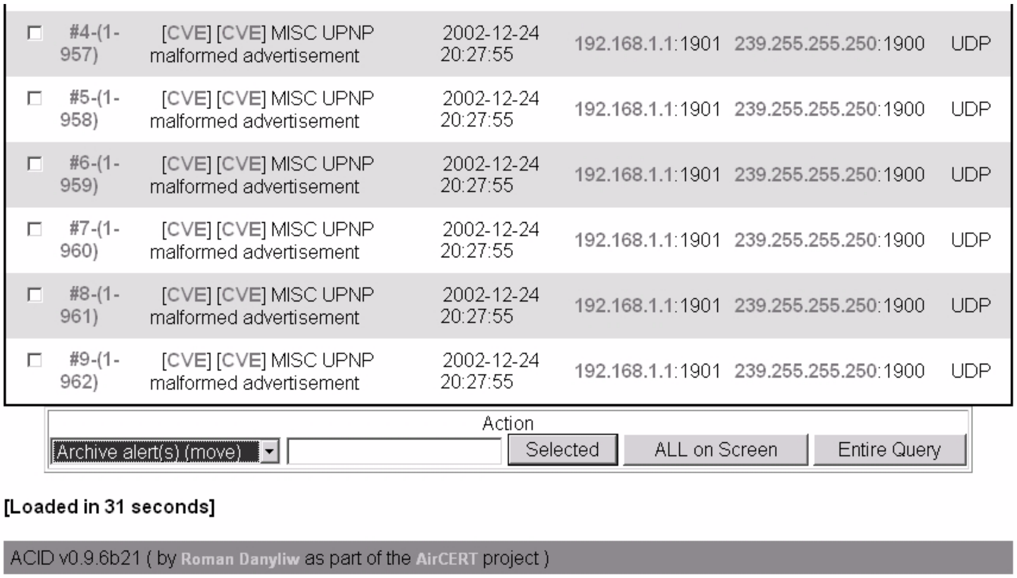
Minimum Threshold Value (>=): 0 □ Rotate Axis Labels (90 degrees) □ Y-axis logarithmic

Display X-axis label every 1 data points

No AG was specified. Using all alerts.

ACID Chart
Time vs. Number of Alerts
/21/2002 0:00:00 - 0:59:59 - 07/31/2002 23:00:00 -
Alert Occurrences
667
593
519
445
370
296
222
148
74
0
0
1
2
3
4
5
6
Time

Figure 6-12 Graph of alert data.

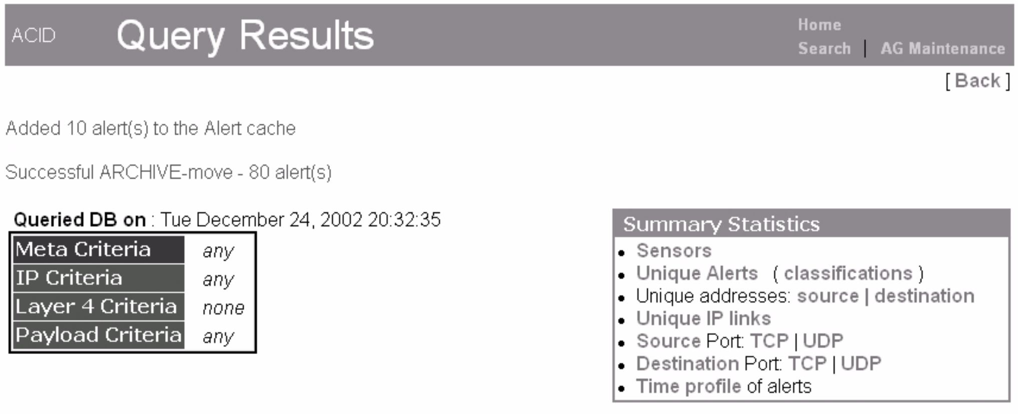


□	#4-(1-957)	[CVE] [CVE] MISC UPNP malformed advertisement	2002-12-24 20:27:55	192.168.1.1:1901	239.255.255.250:1900	UDP
□	#5-(1-958)	[CVE] [CVE] MISC UPNP malformed advertisement	2002-12-24 20:27:55	192.168.1.1:1901	239.255.255.250:1900	UDP
□	#6-(1-959)	[CVE] [CVE] MISC UPNP malformed advertisement	2002-12-24 20:27:55	192.168.1.1:1901	239.255.255.250:1900	UDP
□	#7-(1-960)	[CVE] [CVE] MISC UPNP malformed advertisement	2002-12-24 20:27:55	192.168.1.1:1901	239.255.255.250:1900	UDP
□	#8-(1-961)	[CVE] [CVE] MISC UPNP malformed advertisement	2002-12-24 20:27:55	192.168.1.1:1901	239.255.255.250:1900	UDP
□	#9-(1-962)	[CVE] [CVE] MISC UPNP malformed advertisement	2002-12-24 20:27:55	192.168.1.1:1901	239.255.255.250:1900	UDP

[Loaded in 31 seconds]

ACID v0.9.6b21 (by Roman Danyliw as part of the AirCERT project)

Figure 6-13 Moving alerts to the archive database.



Added 10 alert(s) to the Alert cache

Successful ARCHIVE-move - 80 alert(s)

Queried DB on : Tue December 24, 2002 20:32:35

Meta Criteria	*any*
IP Criteria	*any*
Layer 4 Criteria	*none*
Payload Criteria	*any*

Summary Statistics

- Sensors
- Unique Alerts (classifications)
- Unique addresses: source | destination
- Unique IP links
- Source Port: TCP | UDP
- Destination Port: TCP | UDP
- Time profile of alerts

Figure 6-14 Result of moving alert data to archive database.

6.3.8 ACID Tables

When you start using ACID for the first time, it creates its own tables in the Snort database. These tables are used for housekeeping functions of ACID. For example, you can create new alert groups called (AG) in ACID and ACID keeps a record in its own tables. This section shows a list of MySQL database tables before and after configuring ACID. The following is a list of tables as they appear before using ACID for the first time.

```
mysql> show tables;
+------------------+
| Tables_in_snort  |
+------------------+
| data             |
| detail           |
| encoding         |
| event            |
| flags            |
| icmphdr          |
| iphdr            |
| opt              |
| protocols        |
| reference        |
| reference_system |
| schema           |
| sensor           |
| services         |
| sig_class        |
| sig_reference    |
| signature        |
| tcphdr           |
| udphdr           |
+------------------+
19 rows in set (0.01 sec)

mysql>
```

The following is a list of tables after the creation of ACID tables in the database. The user name that was used for ACID must have permission to create new tables. Refer to Chapter 5 for information about granting permissions.

```
mysql> show tables;
+------------------+
| Tables_in_snort  |
+------------------+
| acid_ag          |
| acid_ag_alert    |
| acid_event       |
| acid_ip_cache    |
| data             |
| detail           |
| encoding         |
| event            |
| flags            |
| icmphdr          |
| iphdr            |
| opt              |
| protocols        |
| reference        |
| reference_system |
| schema           |
| sensor           |
| services         |
| sig_class        |
| sig_reference    |
| signature        |
| tcphdr           |
| udphdr           |
+------------------+
23 rows in set (0.00 sec)

mysql>
```

The first four tables in the list show the newly created ACID tables.

6.4 SnortSnarf

SnortSnarf is another tool to display Snort data using a web interface. It is available from its web site at http://www.silicondefense.com/software/snortsnarf/index.htm. Basically it is a Perl script and you can run it after downloading without going through any compilation process. It can parse Snort log files as well as extract data from MySQL database. The following command parses `/var/log/snort/alert` file and places the newly generated HTML files in the `/var/www/html/snortsnarf` directory where they can be viewed later using a web browser.

```
snortsnarf.pl /var/log/snort/alert -d /var/www/html/snortsnarf
```

The following command extracts data from MySQL database running on the `localhost`. It uses a user name `rr` and password `rr78x` to login to the database.

```
snortsnarf.pl rr:rr78x@snort@localhost -d /var/www/html/snortsnarf
```

To get data from a database, you have to define the following parameters on the command line:

- Database user name
- Password
- Database name
- Host where database server is running
- Port number for the database server. By default the port number is 3306 and this parameter is optional.

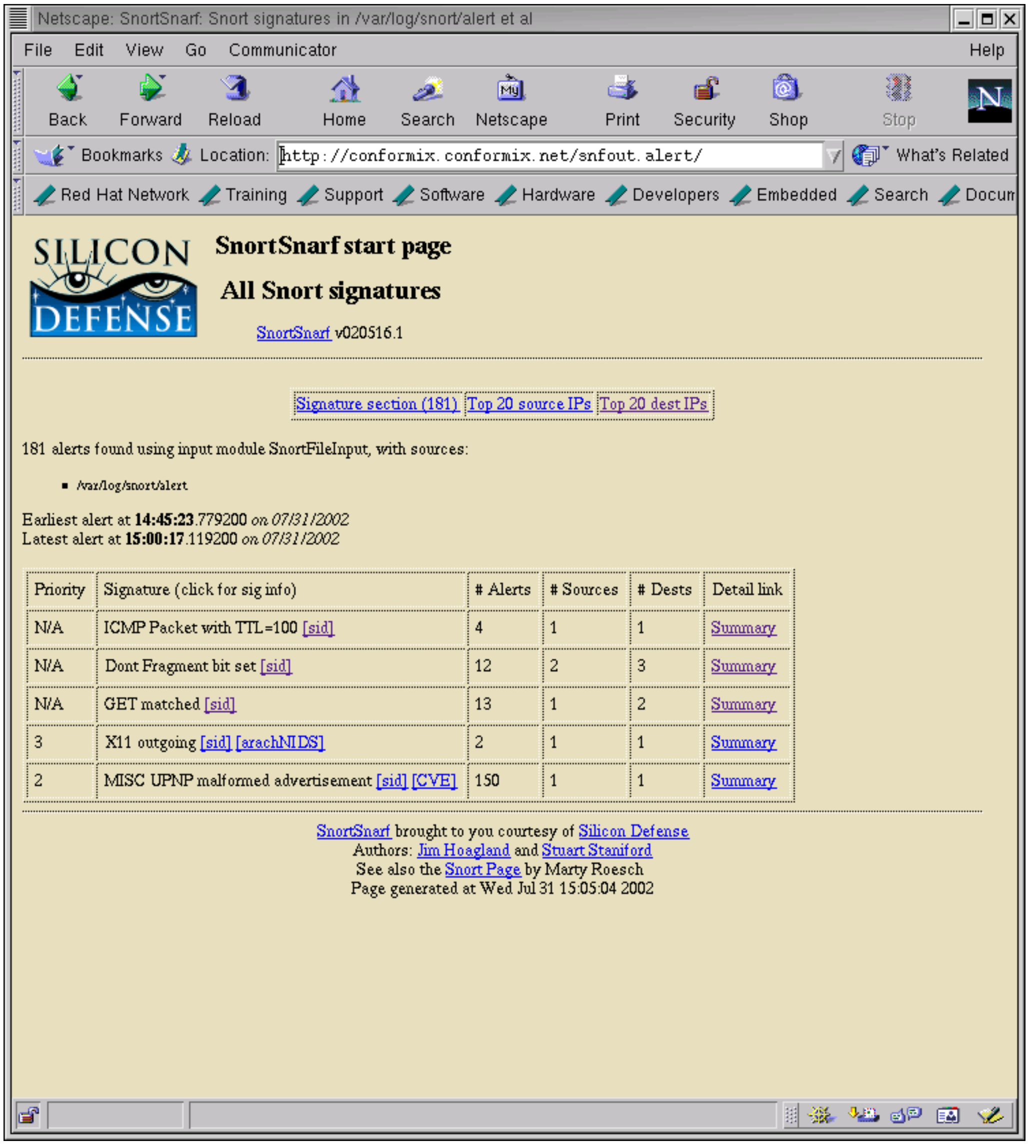
The general format of defining these parameters is:

```
    user:passwd@dbname@host:port
```

You can run SnortSnarf from a cron script on a periodic basis. Figure 6-15 shows the main page created by SnortSnarf. It provides basic information about alert data.

Figure 6-16 shows the information about a particular alert that is displayed when you click a link as shown in Figure 6-15.

Figure 6-17 shows a screen shot for searching whois databases or DNS lookup when you need to get more information about an IP address.


Netscape: SnortSnarf: Snort signatures in /var/log/snort/alert et al
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: http://conformix.conformix.net/snfout.alert/ What's Related
Red Hat Network Training Support Software Hardware Developers Embedded Search Docum
SILICON DEFENSE

SnortSnarf start page

All Snort signatures

SnortSnarf v020516.1

Signature section (181) Top 20 source IPs Top 20 dest IPs

181 alerts found using input module SnortFileInput, with sources:

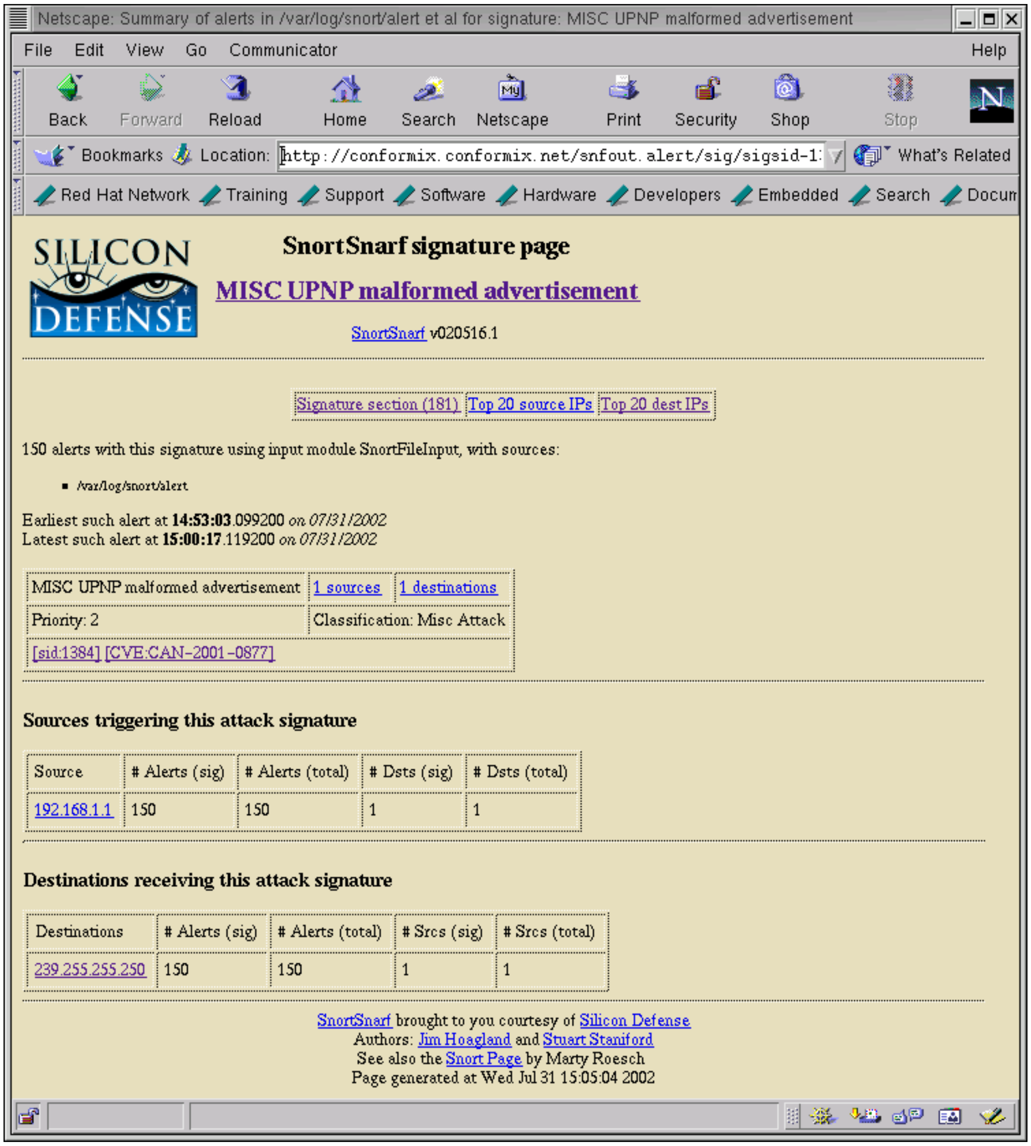
- /var/log/snort/alert

Earliest alert at **14:45:23**.779200 *on 07/31/2002*
Latest alert at **15:00:17**.119200 *on 07/31/2002*

Priority	Signature (click for sig info)	# Alerts	# Sources	# Dests	Detail link
N/A	ICMP Packet with TTL=100 [sid]	4	1	1	Summary
N/A	Dont Fragment bit set [sid]	12	2	3	Summary
N/A	GET matched [sid]	13	1	2	Summary
3	X11 outgoing [sid] [arachNIDS]	2	1	1	Summary
2	MISC UPNP malformed advertisement [sid] [CVE]	150	1	1	Summary

SnortSnarf brought to you courtesy of Silicon Defense
Authors: Jim Hoagland and Stuart Staniford
See also the Snort Page by Marty Roesch
Page generated at Wed Jul 31 15:05:04 2002

Figure 6-15 SnortSnarf main page.



SnortSnarf signature page

MISC UPNP malformed advertisement

SnortSnarf v020516.1

Signature section (181) | Top 20 source IPs | Top 20 dest IPs

150 alerts with this signature using input module SnortFileInput, with sources:

- /var/log/snort/alert

Earliest such alert at **14:53:03**.099200 *on 07/31/2002*
Latest such alert at **15:00:17**.119200 *on 07/31/2002*

MISC UPNP malformed advertisement	1 sources	1 destinations
Priority: 2	Classification: Misc Attack	
[sid:1384] [CVE:CAN-2001-0877]		

Sources triggering this attack signature

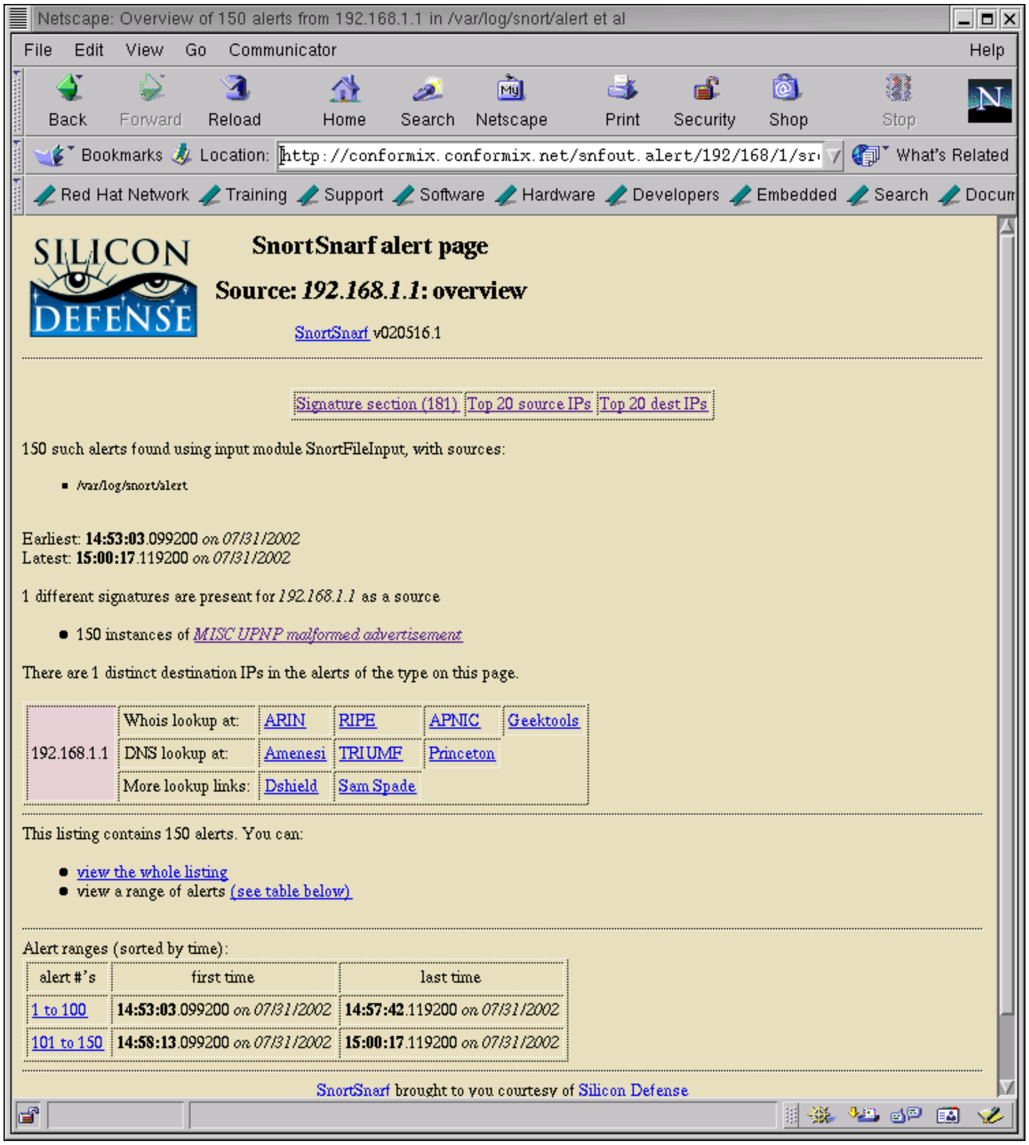
Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
192.168.1.1	150	150	1	1

Destinations receiving this attack signature

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
239.255.255.250	150	150	1	1

SnortSnarf brought to you courtesy of Silicon Defense
Authors: Jim Hoagland and Stuart Staniford
See also the Snort Page by Marty Roesch
Page generated at Wed Jul 31 15:05:04 2002

Figure 6-16 Detail of a particular alert in SnortSnarf.



SnortSnarf alert page

Source: *192.168.1.1*: overview

SnortSnarf v020516.1

Signature section (181) | Top 20 source IPs | Top 20 dest IPs

150 such alerts found using input module SnortFileInput, with sources:

- /var/log/snort/alert

Earliest: **14:53:03**.099200 *on 07/31/2002*
Latest: **15:00:17**.119200 *on 07/31/2002*

1 different signatures are present for *192.168.1.1* as a source

- 150 instances of *MISC UPNP malformed advertisement*

There are 1 distinct destination IPs in the alerts of the type on this page.

192.168.1.1	Whois lookup at:	ARIN	RIPE	APNIC	Geektools
	DNS lookup at:	Amenesi	TRIUMF	Princeton	
	More lookup links:	Dshield	Sam Spade		

This listing contains 150 alerts. You can:

- view the whole listing
- view a range of alerts (see table below)

Alert ranges (sorted by time):

alert #'s	first time	last time
1 to 100	**14:53:03**.099200 *on 07/31/2002*	**14:57:42**.119200 *on 07/31/2002*
101 to 150	**14:58:13**.099200 *on 07/31/2002*	**15:00:17**.119200 *on 07/31/2002*

SnortSnarf brought to you courtesy of Silicon Defense

Figure 6-17 Getting more information about an IP address.

6.5 Barnyard

Barnyard is a new tool which is intended to parse binary log files generated by Snort when you use the unified logging module. Barnyard is still in experimental form at the time of writing this book. You can download the latest version from the Snort web site and read the included file about installation and use of the tool. Basically you have to carry out the following three steps to compile and install it.

1. Run the configure script with a prefix command line parameter to define the directory where you intend to install it. A typical command line may be "`configure --prefix=/opt/barnyard`".
2. Run the `make` command.
3. Run the `make install` command to install it.

You also need to edit the `barnyard.conf` file before using the tool. I am omitting a detailed discussion because the process may change significantly by the time you read this book.

WARNING At the time of writing this book, Barnyard is still in the development process and the installation may differ significantly in the final release of the package.

6.6 References

1. ACID is available from http://www.cert.org/kb/acid/
2. Apache web site at http://www.apache.org
3. PHP web site at http://www.php.net
4. GD library at http://www.boutell.com/gd/
5. PHPLOT package at http://www.phplot.com
6. ADODB package at http://php.weblogs.com/adodb
7. SnortSnarf at http://www.silicondefense.com/software/snortsnarf/index.htm
8. ADODB FAQ at http://php.weblogs.com/adodb_faq