Intrusion Detection System Lab

Geoff Vaughan

In this lab I will configure an intrusion detection system on a local machine and see if it can detect and create alert notification for various types of attacks. To do this I will configure a local machine with Snort IDS and set it up to listen to network traffic. I will configure snort to generate alerts based on a set of rules and criteria that mimic various attacks. In this case the attacks I will be trying to detect are ping of death[1] and the smurf attack[2]. Once snort has detected the attack it will output the alert to a MySQL database. In order to interpret all the alerts and view the data in a way that is readable I will use BASE[3]. BASE will create more of a user interface for the Snort alerts.

System Setup / Environment

Victim and IDS Machine

CentOS 6.0 64 bit machine with a local IP address of 192.168.139.145 and subnet mask of 192.168.139.0/24

For the most part I followed this tutorial[4] to set up the IDS machine. A couple of modifications were needed in order for me to be successful in setting up the IDS. First the tutorial suggests using the epel repository to download the libraries and applications needed. As this repository was unavailable to my machine it meant that some of the libraries needed to be installed manually. Additionally where the tutorial suggests downloading the latest versions of the Snort IDS and some of the other libraries I opted to use the older versions as used in the tutorial. I found this to be the biggest challenge in implementing this system as the new version of Snort has depreciated it support for direct output into the MySQL database (just in the last 2 weeks). In newer versions of Snort you need to use an intermediary software application called Barnyard [5] to buffer the data from Snort into the database. Given the nature of open source projects I found that the support and documentation for Barnyard to still have gaps in it to support users. For this reason I would recommend sticking with the slightly older versions until more tutorials and documentation notes are produced.

Primary Libraries and Applications Installed
(additional supporting libraries were installed with yum)

- gcc and gcc-c++
- make
- rpm-build
- autoconf
- automake
- flex
- libpcap-devel
- bison
- libdnet and libdnet-devel
- mysql and mysql-devel
- pcre-devel

- php and php-mysql
- pear and pear-image-graph
- BASE (php Script)
- DAQ
- ADODB (php script)
- Snort Rules (registration required)

Attacker

For the purpose of this lab all you need for an attacker is a computer that is network accessible to the local machine with the ability to ping it. Ideally to genuinely test the Ping of Death attack you would want to write a script that can generate packets larger than 65507 bytes as the ping command prevents you from generating packets larger than that size. For the case of generating the IDS rule to prevent the Ping of Death attack I will make the rule to alert if packets are larger than 1000 bytes and then up the number to 65507 once I have established a proof of concept.

Procedures

1. The big challenge with this lab is setting up the IDS machine completely. For that I used this tutorial[4]. It is really important that you follow this tutorial as closely as possible as any variation can result in certain aspects of the system not working properly. There was just a few things that I did differently from this tutorial that I will mention here.
    a. The Epel repository was not available on my machine so this meant that some libraries needed to be installed from the sources (pear, libdnet, and libdnet-devel to name a few).
    b. In the /etc/snort/snort.conf file I had to comment out a number of rules that were not a part of the most recent rule pack. For the purpose of this lab it wasn't necessary anyways as we are just looking for the two types of DoS attacks.
    c. I stuck with the versions listed in the tutorial rather than getting the newest versions. I resorted to this choice after many hours of headaches with the newest version of Snort that required the use of Barnyard as an intermediary between Snort and the database.
    d. In order to test to ensure that snort is actually working properly you need to write at least one rule that you can easily generate. To do this you can either make your own rule file and add it to the list of snort rule files in /ete/snort/snort.conf, or just open up an existing snort rule that is already in use and add the following line:

        alert tcp any any -> any any (msg:"Test alert"; sid:99999;)

        If you complete the tutorial successfully this line of code will trigger an alert anytime there is any tcp traffic going in or out of the IDS machine so to test this alert all you have to do is browse a web page.
    e. In order to get base to work I had to disable SeLinux:

        setenforce 0
    a. Additionally to ensure that there were no conflicts with the firewall I disabled iptables with the following commands:

        /etc/init.d/iptables save
        /etc/init.d/iptables stop

2. Writing the Snort rules - To detect the Ping of Death attack as well as the Smurf attack you need two rules in place to listen for these types of packets. Edit one of the existing rule files from /etc/snort/rules/ and add the desired rules. The rules I used were as follows:

   alert icmp any any -> any any (dsize: > 10000 msg: "Ping of Death Detected"; sid:777777)

   alert icmp any any -> any 192.168.139.255 any (msg:"stop smurfing me"; sid 888888)

   Here the first rule detects if any ICMP (ping) packets are larger than 1000 bytes. Initially I am testing this rule with 1000 bytes to prove the rule works then later I will up it to the maximum allowable size of 65507 bytes. The second rule listens for any ping requests where the broadcast address is listed as the destination which is a signature of a smurf attack.

3. Launching snort - assuming that you already have BASE, MySQL, Apache, and PHP already running from following the tutorial properly. At this point you need to relaunch Snort with the new rules. To do this type the following from the IDS machine

   snort -c /etc/snort/snort.conf -v

   This will launch snort with the preconfigured file. Leave snort running while you launch your attacks

4. Launching smurf attack - From a new terminal window on the IDS machine or an attacker machine connected to the same network as the IDS, ping the broadcast address using the following command:

   ping 192.168.139.255

5. Launching Ping of Death attack - From a new terminal window on the IDS machine or an attacker machine connected to the same networks as the IDS, make a ping request that is visible to the IDS machine (either from the IDS or directed at the IDS) that has a packet size greater than 65507

   ping -s 50000 192.168.139.145

7. Analyzing the data - At this point you should be able to load BASE and view the results from the within the base interface. To do this, in a web browser navigate to:

   http://localhost/base/

   From here you can browse through the user interface and you should see however many icmp and tcp alerts you've triggered you can also click on the icmp or tcp links to see detailed information about each alert.

Observations

Image 1 Snort running and detecting packets. Launched with command:

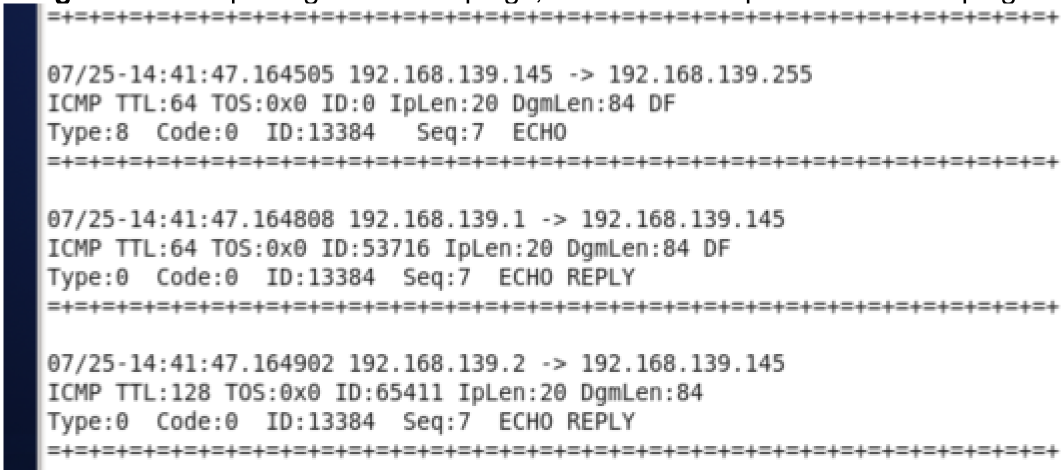
snort -c /etc/snort/snort.conf -v

```
        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.0.5 (Build 135)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/snort/snort-team
           Copyright (C) 1998-2011 Sourcefire, Inc., et al.
           Using libpcap version 1.0.0
           Using PCRE version: 7.8 2008-09-05

           Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 1.13  <Build 18>
           Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
           Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
           Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
           Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
           Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>
           Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
           Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
Commencing packet processing (pid=18472)
```

Image 2 Snort capturing large ICMP packets, note the Dgmlen on the last packet.

```
ICMP TTL:64 TOS:0x0 ID:30364 IpLen:20 DgmLen:548 MF
Frag Offset: 0x1776   Frag Size: 0x0210
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

07/25-14:40:26.716115 192.168.139.145 -> 192.168.1.141
ICMP TTL:64 TOS:0x0 ID:30364 IpLen:20 DgmLen:548 MF
Frag Offset: 0x17B8   Frag Size: 0x0210
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

07/25-14:40:26.716151 192.168.139.145 -> 192.168.1.141
ICMP TTL:64 TOS:0x0 ID:30364 IpLen:20 DgmLen:548 MF
Frag Offset: 0x17FA   Frag Size: 0x0210
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

07/25-14:40:26.716188 192.168.139.145 -> 192.168.1.141
ICMP TTL:64 TOS:0x0 ID:30364 IpLen:20 DgmLen:50028
Type:8  Code:0  ID:11848   Seq:6  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Image 3 Snort capturing broadcast pings, note the two replies for the one ping.

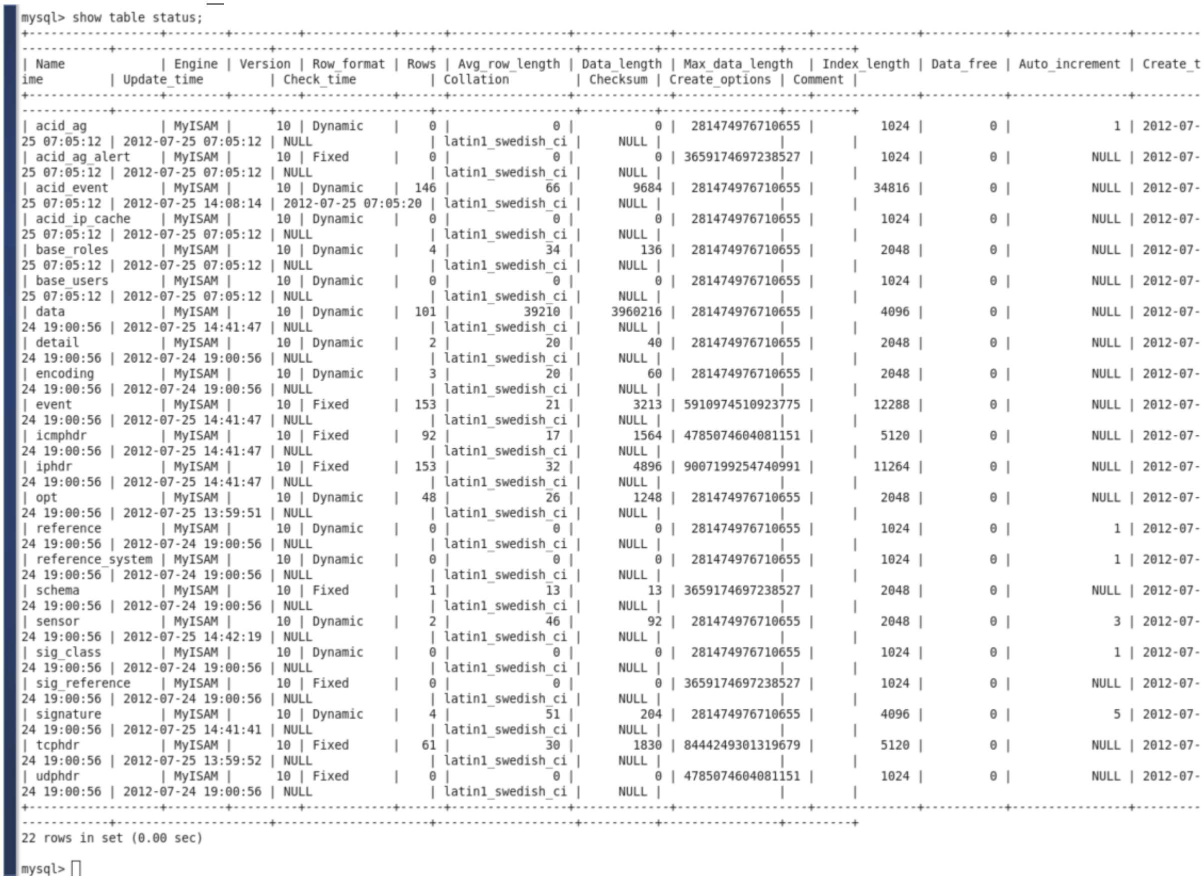


```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

07/25-14:41:47.164505 192.168.139.145 -> 192.168.139.255
ICMP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:13384   Seq:7  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

07/25-14:41:47.164808 192.168.139.1 -> 192.168.139.145
ICMP TTL:64 TOS:0x0 ID:53716 IpLen:20 DgmLen:84 DF
Type:0  Code:0  ID:13384  Seq:7  ECHO REPLY
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

07/25-14:41:47.164902 192.168.139.2 -> 192.168.139.145
ICMP TTL:128 TOS:0x0 ID:65411 IpLen:20 DgmLen:84
Type:0  Code:0  ID:13384  Seq:7  ECHO REPLY
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Image 4 Logged in to mysql database using the following commands:

mysql -u root -p
→ use snort;
→ show table status;

This shows that there is actual content being added to the mysql database. Note the number of rows in the acid_events table



```
mysql> show table status;
+------------------+--------+---------+------------+------+----------------+-------------+------------------+--------------+-----------+----------------+--------
-----------+---------------------+---------------------+-------------------+----------+----------------+---------+
| Name             | Engine | Version | Row_format | Rows | Avg_row_length | Data_length | Max_data_length  | Index_length | Data_free | Auto_increment | Create_t
ime        | Update_time         | Check_time          | Collation         | Checksum | Create_options | Comment |
+------------------+--------+---------+------------+------+----------------+-------------+------------------+--------------+-----------+----------------+--------
-----------+---------------------+---------------------+-------------------+----------+----------------+---------+
| acid_ag          | MyISAM |      10 | Dynamic    |    0 |              0 |           0 |  281474976710655 |         1024 |         0 |              1 | 2012-07-
25 07:05:12 | 2012-07-25 07:05:12 | NULL                | latin1_swedish_ci |     NULL |                |         |
| acid_ag_alert    | MyISAM |      10 | Fixed      |    0 |              0 |           0 | 3659174697238527 |         1024 |         0 |           NULL | 2012-07-
25 07:05:12 | 2012-07-25 07:05:12 | NULL                | latin1_swedish_ci |     NULL |                |         |
| acid_event       | MyISAM |      10 | Dynamic    |  146 |             66 |        9684 |  281474976710655 |        34816 |         0 |           NULL | 2012-07-
25 07:05:12 | 2012-07-25 14:08:14 | 2012-07-25 07:05:20 | latin1_swedish_ci |     NULL |                |         |
| acid_ip_cache    | MyISAM |      10 | Dynamic    |    0 |              0 |           0 |  281474976710655 |         1024 |         0 |           NULL | 2012-07-
25 07:05:12 | 2012-07-25 07:05:12 | NULL                | latin1_swedish_ci |     NULL |                |         |
| base_roles       | MyISAM |      10 | Dynamic    |    4 |             34 |         136 |  281474976710655 |         2048 |         0 |           NULL | 2012-07-
25 07:05:12 | 2012-07-25 07:05:12 | NULL                | latin1_swedish_ci |     NULL |                |         |
| base_users       | MyISAM |      10 | Dynamic    |    0 |              0 |           0 |  281474976710655 |         1024 |         0 |           NULL | 2012-07-
25 07:05:12 | 2012-07-25 07:05:12 | NULL                | latin1_swedish_ci |     NULL |                |         |
| data             | MyISAM |      10 | Dynamic    |  101 |          39210 |     3960216 |  281474976710655 |         4096 |         0 |           NULL | 2012-07-
24 19:00:56 | 2012-07-25 14:41:47 | NULL                | latin1_swedish_ci |     NULL |                |         |
| detail           | MyISAM |      10 | Dynamic    |    2 |             20 |          40 |  281474976710655 |         2048 |         0 |           NULL | 2012-07-
24 19:00:56 | 2012-07-24 19:00:56 | NULL                | latin1_swedish_ci |     NULL |                |         |
| encoding         | MyISAM |      10 | Dynamic    |    3 |             20 |          60 |  281474976710655 |         2048 |         0 |           NULL | 2012-07-
24 19:00:56 | 2012-07-24 19:00:56 | NULL                | latin1_swedish_ci |     NULL |                |         |
| event            | MyISAM |      10 | Fixed      |  153 |             21 |        3213 | 5910974510923775 |        12288 |         0 |           NULL | 2012-07-
24 19:00:56 | 2012-07-25 14:41:47 | NULL                | latin1_swedish_ci |     NULL |                |         |
| icmphdr          | MyISAM |      10 | Fixed      |   92 |             17 |        1564 | 4785074604081151 |         5120 |         0 |           NULL | 2012-07-
24 19:00:56 | 2012-07-25 14:41:47 | NULL                | latin1_swedish_ci |     NULL |                |         |
| iphdr            | MyISAM |      10 | Fixed      |  153 |             32 |        4896 | 9007199254740991 |        11264 |         0 |           NULL | 2012-07-
24 19:00:56 | 2012-07-25 14:41:47 | NULL                | latin1_swedish_ci |     NULL |                |         |
| opt              | MyISAM |      10 | Dynamic    |   48 |             26 |        1248 |  281474976710655 |         2048 |         0 |           NULL | 2012-07-
24 19:00:56 | 2012-07-25 13:59:51 | NULL                | latin1_swedish_ci |     NULL |                |         |
| reference        | MyISAM |      10 | Dynamic    |    0 |              0 |           0 |  281474976710655 |         1024 |         0 |              1 | 2012-07-
24 19:00:56 | 2012-07-24 19:00:56 | NULL                | latin1_swedish_ci |     NULL |                |         |
| reference_system | MyISAM |      10 | Dynamic    |    0 |              0 |           0 |  281474976710655 |         1024 |         0 |              1 | 2012-07-
24 19:00:56 | 2012-07-24 19:00:56 | NULL                | latin1_swedish_ci |     NULL |                |         |
| schema           | MyISAM |      10 | Fixed      |    1 |             13 |          13 | 3659174697238527 |         2048 |         0 |           NULL | 2012-07-
24 19:00:56 | 2012-07-24 19:00:56 | NULL                | latin1_swedish_ci |     NULL |                |         |
| sensor           | MyISAM |      10 | Dynamic    |    2 |             46 |          92 |  281474976710655 |         2048 |         0 |              3 | 2012-07-
24 19:00:56 | 2012-07-25 14:42:19 | NULL                | latin1_swedish_ci |     NULL |                |         |
| sig_class        | MyISAM |      10 | Dynamic    |    0 |              0 |           0 |  281474976710655 |         1024 |         0 |              1 | 2012-07-
24 19:00:56 | 2012-07-24 19:00:56 | NULL                | latin1_swedish_ci |     NULL |                |         |
| sig_reference    | MyISAM |      10 | Fixed      |    0 |              0 |           0 | 3659174697238527 |         1024 |         0 |           NULL | 2012-07-
24 19:00:56 | 2012-07-24 19:00:56 | NULL                | latin1_swedish_ci |     NULL |                |         |
| signature        | MyISAM |      10 | Dynamic    |    4 |             51 |         204 |  281474976710655 |         4096 |         0 |              5 | 2012-07-
24 19:00:56 | 2012-07-25 14:41:41 | NULL                | latin1_swedish_ci |     NULL |                |         |
| tcphdr           | MyISAM |      10 | Fixed      |   61 |             30 |        1830 | 8444249301319679 |         5120 |         0 |           NULL | 2012-07-
24 19:00:56 | 2012-07-25 13:59:52 | NULL                | latin1_swedish_ci |     NULL |                |         |
| udphdr           | MyISAM |      10 | Fixed      |    0 |              0 |           0 | 4785074604081151 |         1024 |         0 |           NULL | 2012-07-
24 19:00:56 | 2012-07-24 19:00:56 | NULL                | latin1_swedish_ci |     NULL |                |         |
+------------------+--------+---------+------------+------+----------------+-------------+------------------+--------------+-----------+----------------+--------
-----------+---------------------+---------------------+-------------------+----------+----------------+---------+
22 rows in set (0.00 sec)

mysql> 
```

Image 5 Partial contents of acid_events table in snort database, displayed using command:

select * from acid_events;

2	43	1	y U Hax my system???	0	NULL	2012-07-24 19:45:41	1311642084	3232271249	6	52745
2	44	1	y U Hax my system???	0	NULL	2012-07-24 19:45:41	1311642084	3232271249	6	52746
2	45	1	y U Hax my system???	0	NULL	2012-07-24 19:45:41	3232271249	1311642084	6	80
2	46	1	y U Hax my system???	0	NULL	2012-07-24 19:45:41	1311642084	3232271249	6	52747
2	47	2	stop smurfing me>Broadcast ping detected	0	NULL	2012-07-25 13:08:08	3232271105	3232271359	1	NULL
2	48	2	stop smurfing me>Broadcast ping detected	0	NULL	2012-07-25 13:08:09	3232271105	3232271359	1	NULL
2	49	2	stop smurfing me>Broadcast ping detected	0	NULL	2012-07-25 13:08:10	3232271105	3232271359	1	NULL
2	50	2	stop smurfing me>Broadcast ping detected	0	NULL	2012-07-25 13:08:11	3232271105	3232271359	1	NULL
2	51	2	stop smurfing me>Broadcast ping detected	0	NULL	2012-07-25 13:08:12	3232271105	3232271359	1	NULL
2	52	2	stop smurfing me>Broadcast ping detected	0	NULL	2012-07-25 13:08:13	3232271105	3232271359	1	NULL
2	53	3	Ping of Death Detected	0	NULL	2012-07-25 13:40:22	3232271249	3232235885	1	NULL
2	54	3	Ping of Death Detected	0	NULL	2012-07-25 13:40:22	3232235885	3232271249	1	NULL
2	55	3	Ping of Death Detected	0	NULL	2012-07-25 13:40:23	3232271249	3232235885	1	

Image 6 View from http://localhost/base/ The TCP alerts are from when I had the any tcp traffic rule from my original testing and the ICMP alerts are from the ping of death and smurf attacks.

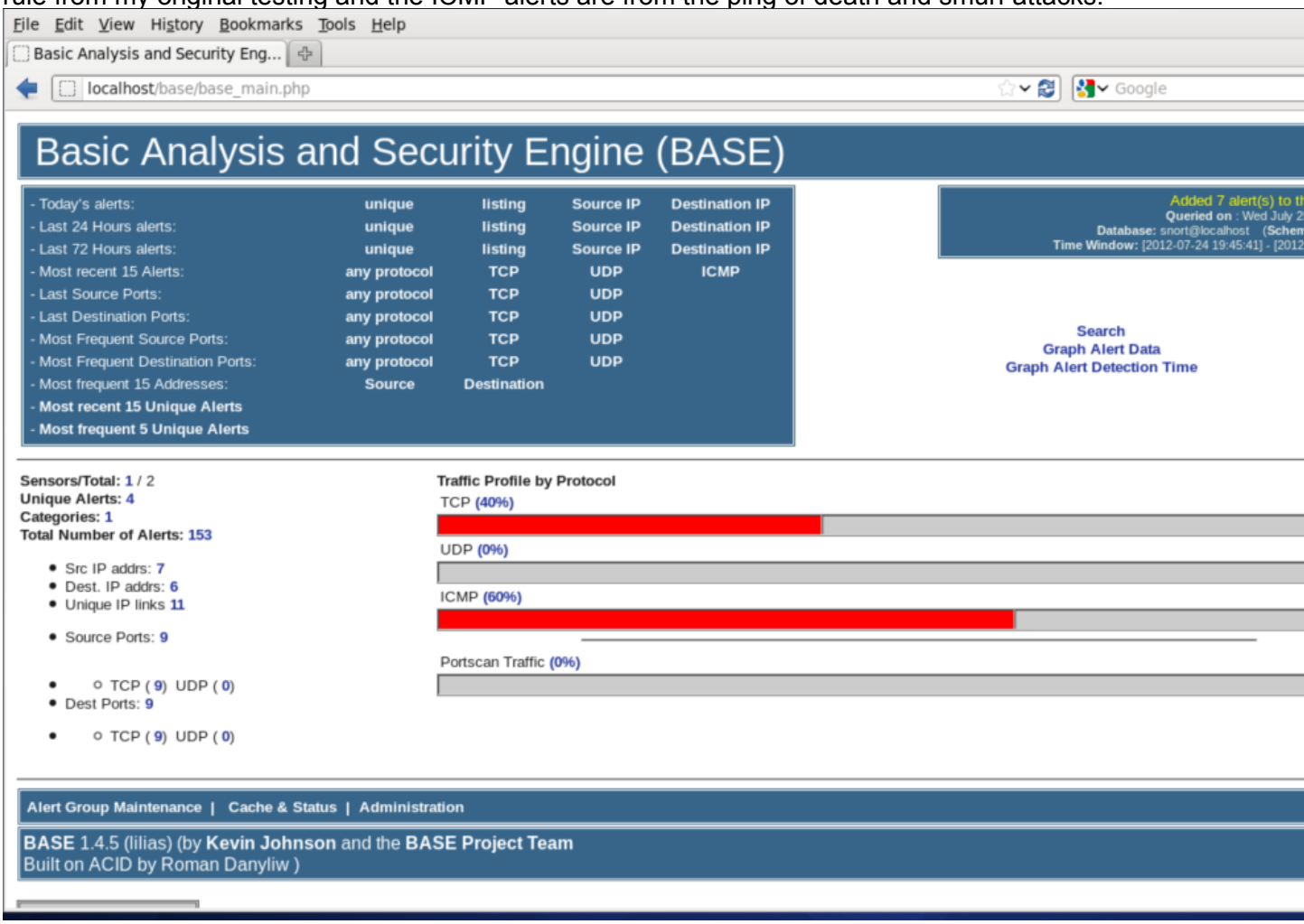
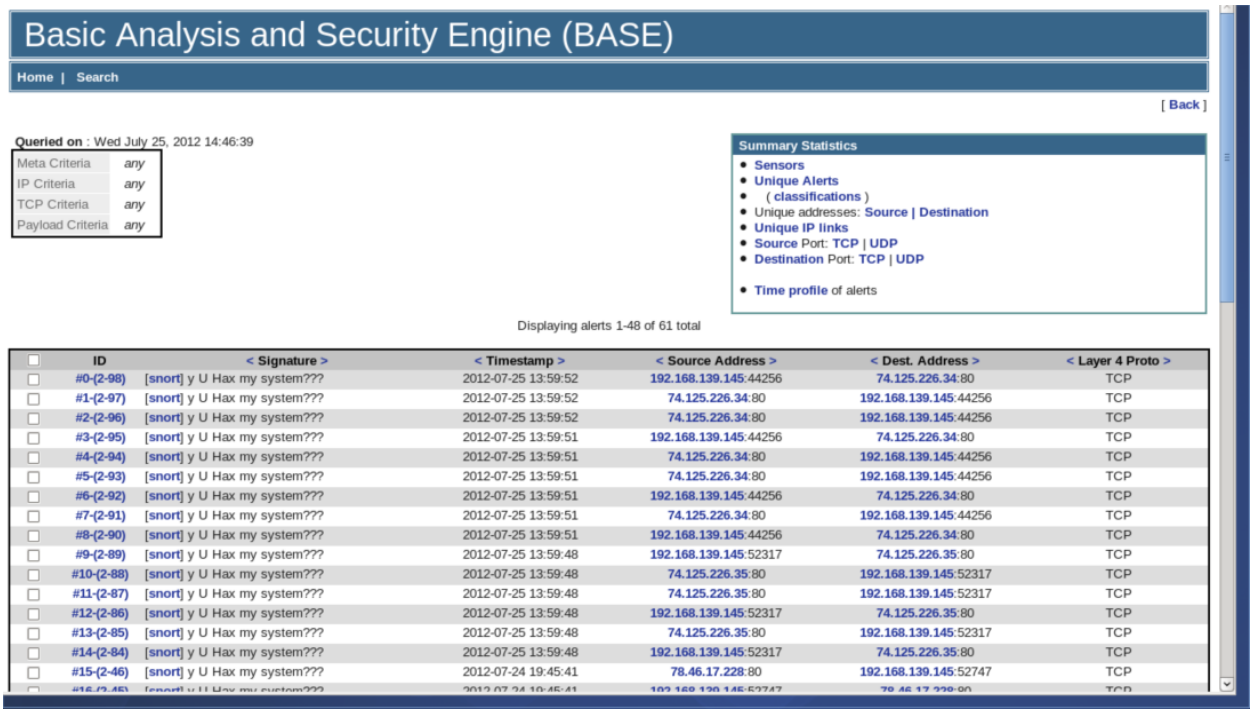


Image 7 TCP alerts from original testing rule where any tcp packet would generate an alert.



Basic Analysis and Security Engine (BASE)

Home | Search

[Back]

Queried on : Wed July 25, 2012 14:46:39

Meta Criteria	any
IP Criteria	any
TCP Criteria	any
Payload Criteria	any

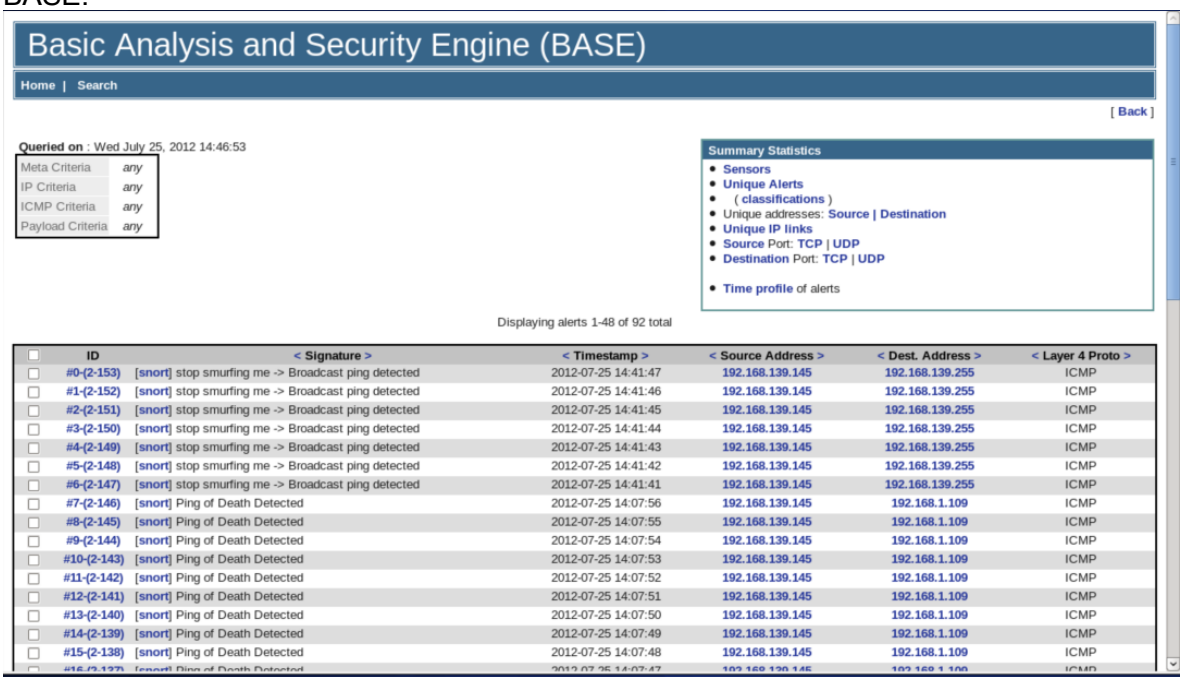
Summary Statistics

- 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 1-48 of 61 total

ID	< Signature >	< Timestamp >	< Source Address >	< Dest. Address >	< Layer 4 Proto >
#0-(2-98)	[snort] y U Hax my system???	2012-07-25 13:59:52	192.168.139.145:44256	74.125.226.34:80	TCP
#1-(2-97)	[snort] y U Hax my system???	2012-07-25 13:59:52	74.125.226.34:80	192.168.139.145:44256	TCP
#2-(2-96)	[snort] y U Hax my system???	2012-07-25 13:59:52	74.125.226.34:80	192.168.139.145:44256	TCP
#3-(2-95)	[snort] y U Hax my system???	2012-07-25 13:59:51	192.168.139.145:44256	74.125.226.34:80	TCP
#4-(2-94)	[snort] y U Hax my system???	2012-07-25 13:59:51	74.125.226.34:80	192.168.139.145:44256	TCP
#5-(2-93)	[snort] y U Hax my system???	2012-07-25 13:59:51	74.125.226.34:80	192.168.139.145:44256	TCP
#6-(2-92)	[snort] y U Hax my system???	2012-07-25 13:59:51	192.168.139.145:44256	74.125.226.34:80	TCP
#7-(2-91)	[snort] y U Hax my system???	2012-07-25 13:59:51	74.125.226.34:80	192.168.139.145:44256	TCP
#8-(2-90)	[snort] y U Hax my system???	2012-07-25 13:59:51	192.168.139.145:44256	74.125.226.34:80	TCP
#9-(2-89)	[snort] y U Hax my system???	2012-07-25 13:59:48	192.168.139.145:52317	74.125.226.35:80	TCP
#10-(2-88)	[snort] y U Hax my system???	2012-07-25 13:59:48	74.125.226.35:80	192.168.139.145:52317	TCP
#11-(2-87)	[snort] y U Hax my system???	2012-07-25 13:59:48	74.125.226.35:80	192.168.139.145:52317	TCP
#12-(2-86)	[snort] y U Hax my system???	2012-07-25 13:59:48	192.168.139.145:52317	74.125.226.35:80	TCP
#13-(2-85)	[snort] y U Hax my system???	2012-07-25 13:59:48	74.125.226.35:80	192.168.139.145:52317	TCP
#14-(2-84)	[snort] y U Hax my system???	2012-07-25 13:59:48	192.168.139.145:52317	74.125.226.35:80	TCP
#15-(2-46)	[snort] y U Hax my system???	2012-07-24 19:45:41	78.46.17.228:80	192.168.139.145:52747	TCP

Image 8 ICMP alerts from Ping of Death and Smurf attack rules in BASE.



Basic Analysis and Security Engine (BASE)

Home | Search

[Back]

Queried on : Wed July 25, 2012 14:46:53

Meta Criteria	any
IP Criteria	any
ICMP Criteria	any
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 1-48 of 92 total

ID	< Signature >	< Timestamp >	< Source Address >	< Dest. Address >	< Layer 4 Proto >
#0-(2-153)	[snort] stop smurfing me -> Broadcast ping detected	2012-07-25 14:41:47	192.168.139.145	192.168.139.255	ICMP
#1-(2-152)	[snort] stop smurfing me -> Broadcast ping detected	2012-07-25 14:41:46	192.168.139.145	192.168.139.255	ICMP
#2-(2-151)	[snort] stop smurfing me -> Broadcast ping detected	2012-07-25 14:41:45	192.168.139.145	192.168.139.255	ICMP
#3-(2-150)	[snort] stop smurfing me -> Broadcast ping detected	2012-07-25 14:41:44	192.168.139.145	192.168.139.255	ICMP
#4-(2-149)	[snort] stop smurfing me -> Broadcast ping detected	2012-07-25 14:41:43	192.168.139.145	192.168.139.255	ICMP
#5-(2-148)	[snort] stop smurfing me -> Broadcast ping detected	2012-07-25 14:41:42	192.168.139.145	192.168.139.255	ICMP
#6-(2-147)	[snort] stop smurfing me -> Broadcast ping detected	2012-07-25 14:41:41	192.168.139.145	192.168.139.255	ICMP
#7-(2-146)	[snort] Ping of Death Detected	2012-07-25 14:07:56	192.168.139.145	192.168.1.109	ICMP
#8-(2-145)	[snort] Ping of Death Detected	2012-07-25 14:07:55	192.168.139.145	192.168.1.109	ICMP
#9-(2-144)	[snort] Ping of Death Detected	2012-07-25 14:07:54	192.168.139.145	192.168.1.109	ICMP
#10-(2-143)	[snort] Ping of Death Detected	2012-07-25 14:07:53	192.168.139.145	192.168.1.109	ICMP
#11-(2-142)	[snort] Ping of Death Detected	2012-07-25 14:07:52	192.168.139.145	192.168.1.109	ICMP
#12-(2-141)	[snort] Ping of Death Detected	2012-07-25 14:07:51	192.168.139.145	192.168.1.109	ICMP
#13-(2-140)	[snort] Ping of Death Detected	2012-07-25 14:07:50	192.168.139.145	192.168.1.109	ICMP
#14-(2-139)	[snort] Ping of Death Detected	2012-07-25 14:07:49	192.168.139.145	192.168.1.109	ICMP
#15-(2-138)	[snort] Ping of Death Detected	2012-07-25 14:07:48	192.168.139.145	192.168.1.109	ICMP

Questions

1. Intrusion detection and anti-virus systems come with signature sets installed. You are always advised to keep signatures up-to-date. Why is it so important to keep signatures updated?

   New network vulnerabilities and attacks are always being detected. An updated signature set will ensure that you can always detect the most recent types of known attacks. Sure you might not be able to mitigate the risk of all attacks however you can be sure that you can detect all known ones.

2. False positives happen when the IDS detects what it's not supposed to, typically because of flawed traffic modeling or weak rules/signatures/anomalies specified. In other words, IDS flags authentic activities as malicious. Too many false positives make IDS users lose confidence in IDS and eventually start ignoring any alerts from an IDS system. As a result, real threats will be missed even if IDS detects them. Also, false negative occurs when a real attack takes place for which no alarm is raised. Which is worst? Explain why.

   The worst possible scenario of the two situation is when an attack occurs but you are unable to detect it at all. In the other situation at least if you ignore the security event, if you find out later that your system has been compromised as a result of this attack then you can look through the logs to locate the source of the attack and potentially prevent the incident from happening in the future, which is much better then not even knowing you've been hacked.

3. Does an IDS system which detects attacks with high accuracy also demonstrate high completeness rates? Explain why

   In order to detect attacks with a high degree of accuracy it would seem necessary to have an extremely large rule set as well as the ability to investigate each packet on a very thorough level. This would potentially require a lot of time and system resources. In a system where the IDS is using the same resources as the host machine it is conceivable that the machine would not be able to process high volumes of packet without either dropping packets or consuming attentional system resources that would result in slowing down the host machine. If the sensors and IDS were external from the host machine say attached to the router or in parallel in some manner it could conceivably keep up with a high volume of traffic however this form of implementation would get expensive.

Conclusions

Here I have demonstrated how to detect various types of network attacks including the Ping of Death and Smurf attacks. Smurf attacks are easily detected by writing rules to detect if any of the ICMP packets have a destination IP of that of the local broadcast address. To detect a Ping of death attack you need to look at the size of the packet. If the size of the packet is greater than the maximum allowable size of 65507. To develop this rule I tested it with a smaller packet size

limit then generated packets that violate that rule. From there you can up the limit on the rule and it is safe to assume that any packet over that size would generate an alert.

In terms of completing this assignment the most difficult task was setting up the IDS environment. The IDS has many libraries and components that all need to be installed and configured in a particular way. For the time being it is my recommendation that if you need an IDS to use a slightly older version then the current version until additional resources and support tutorials are produced to help the less experienced user.

References

[1] About Ping of Death - http://en.wikipedia.org/wiki/Ping_of_death
[2] About Smurf Attack - http://en.wikipedia.org/wiki/Smurf_attack
[3] BASE - http://base.secureideas.net/
[4] Tutorial on how to set up a Snort, MySQL, BASE IDS http://www.server-world.info/en/note?os=CentOS_6&p=snort
[5] Barnyard - http://www.securixlive.com/barnyard2/