

Demonstration of The Realeyes Intrusion Detection System

Jim Sansing
May 14, 2008

For more information, see the project website and blog:

<http://realeyes.sourceforge.net>
<http://realeyes-tech.blogspot.com>

The Realeyes project consists of an analysis library that is used to build a network Intrusion Detection System (IDS). This paper describes a live demonstration that was presented to the Columbia Area Linux Users Group (<http://www.calug.org>) at the monthly meeting.

The project designers have used several security tools for monitoring networks and wanted to incorporate the most useful features of them, and some unique ones, in a single application. These features include:

- Collection of enough data to provide context for reported detects
- Reduction of false positives by supporting rule definitions that include both halves of TCP sessions
- Playback of both halves of TCP sessions
- Statistics collection
- The capability to save serious incidents for trends analysis and reporting
- Built in reports

The Realeyes IDS system consists of four components:

- IDS sensor
- Database
- Database – IDS interface (database daemon or DBD)
- User interface

All four were running on the demonstration host, but may be installed in separate hosts or other reasonable combinations. The demonstration did not include an installation of the system, but at the end the IDS installation script was run partially as an example of how the scripts eliminate the need for editing configuration files.

This demonstration was intended to show the current status of the project, which is in Beta testing. The members of CALUG in attendance were mostly familiar with security tools in general and network IDSes in particular, so the focus was on the specifics of the application.

The following screenshots were taken after the presentation, to give a sense of what was covered. A few screenshots have been added of material that was not covered during the demonstration.

```

Shell - Konsole <4>
top - 10:25:47 up 13 days, 23:13, 1 user, load average: 1.63, 0.66, 0.30
Tasks: 201 total, 3 running, 197 sleeping, 0 stopped, 1 zombie
Cpu(s): 34.4%us, 6.4%sy, 0.0%ni, 1.0%id, 57.2%wa, 0.3%hi, 0.7%si, 0.0%st
Mem: 1035776k total, 1020060k used, 15716k free, 772k buffers
Swap: 1510100k total, 1017472k used, 492628k free, 601328k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
  5596 root        16   0  891m 178m 2892 D   2.3  17.6 127:38.83 Xorg
 25045 root         5 -10  480m 7552 7312 S   0.0   0.7   0:00.01 rids_evta
 25046 root         7  -8  460m 9000 8612 S   0.0   0.9   0:00.08 rids_acta
 25043 root        15   0  458m 1556 1260 R   0.3   0.2   0:00.03 realeyesIDS
 25052 root        16   0  458m  9.8m 9816 D   1.0   1.0   0:00.29 rids_coll
 25050 root         7  -8  458m  9m 9928 S  30.6   1.0   0:05.53 rids_stra_data
 25047 root         7  -8  458m  9m 9992 S   0.3   1.0   0:00.08 rids_stra_tcp
 25049 root         7  -8  458m  9.9m 9980 S   0.0   1.0   0:00.07 rids_stra_ip4
 25051 root         8  -7  458m  9m 9.8m S   2.7   1.0   0:00.35 rids_strh
 25048 root         7  -8  458m  9.9m 9940 S   0.3   1.0   0:00.05 rids_stra_udp
 31628 jim         15   0  297m  24m 7540 S   0.0   2.5  42:22.20 soffice.bin
 19430 jim         15   0  225m  70m  28m D   0.0   7.0  3:49.10 firefox-bin

```

```

Shell - Konsole <4>
top - 10:26:33 up 13 days, 23:14, 1 user, load average: 4.13, 1.49, 0.60
Tasks: 201 total, 6 running, 194 sleeping, 0 stopped, 1 zombie
Cpu(s): 58.4%us, 13.5%sy, 0.0%ni, 13.5%id, 13.9%wa, 0.3%hi, 0.3%si, 0.0%st
Mem: 1035776k total, 1018896k used, 16880k free, 2164k buffers
Swap: 1510100k total, 1083132k used, 426968k free, 669092k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
  5596 root        16   0  887m 146m 2728 S   4.3  14.5 127:39.95 Xorg
 25045 root         5 -10  480m  23m  23m S   0.0   2.3   0:00.11 rids_evta
 25046 root         5 -10  460m  26m  26m S   0.7   2.7   0:00.50 rids_acta
 25043 root        15   0  458m 1624 1328 R   0.7   0.2   0:00.16 realeyesIDS
 25052 root        15   0  458m  26m  26m R   5.3   2.6   0:01.97 rids_coll
 25050 root         8  -7  458m  25m  24m S  41.1   2.5   0:21.05 rids_stra_data
 25047 root         7  -8  458m  26m  26m S   2.3   2.6   0:00.95 rids_stra_tcp
 25049 root         7  -8  458m  26m  26m S   1.3   2.6   0:00.55 rids_stra_ip4
 25051 root         9  -6  458m  26m  26m S  12.3   2.6   0:03.13 rids_strh
 25048 root         7  -8  458m  26m  26m S   2.3   2.6   0:00.57 rids_stra_udp
 31628 jim         15   0  297m  16m 6620 S   0.3   1.7  42:22.23 soffice.bin
 19430 jim         15   0  225m  43m  10m R   0.3   4.3   3:49.18 firefox-bin

```

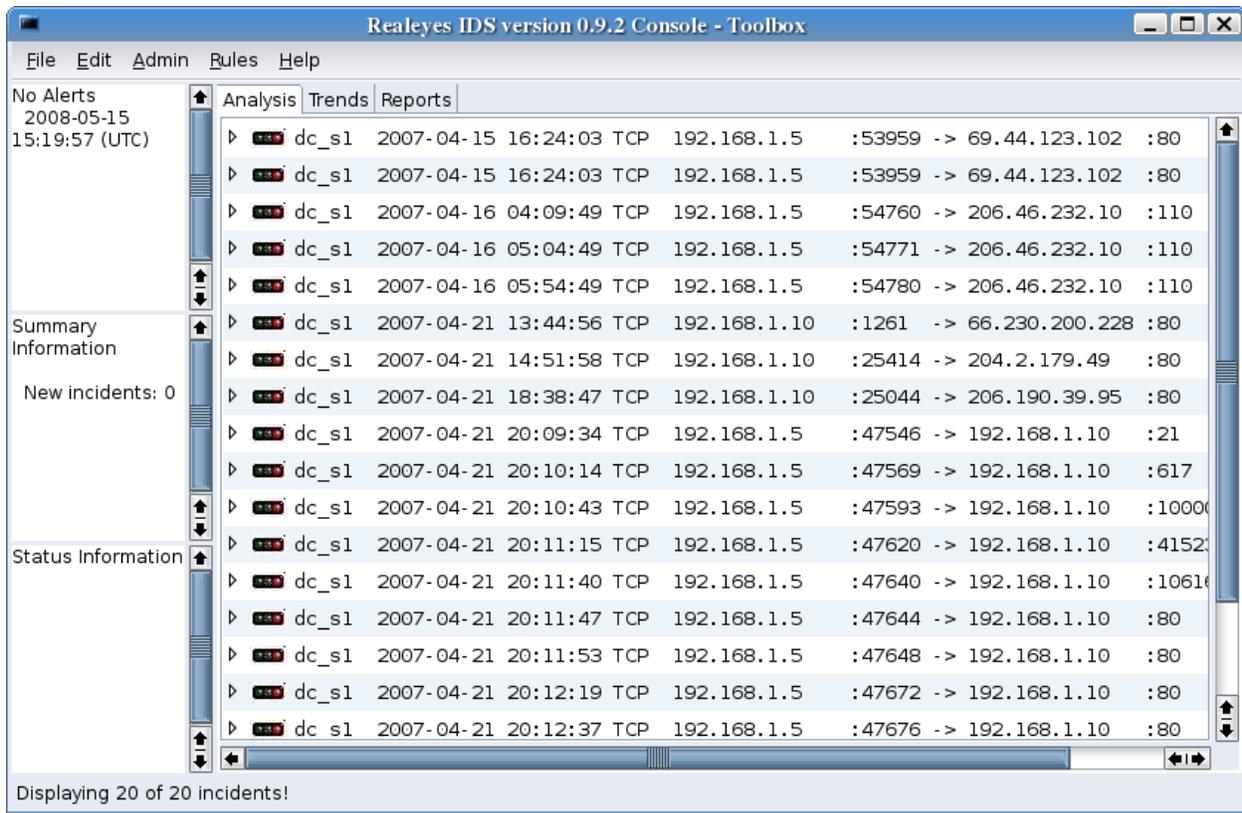
The demonstration began by running the IDS, which is a C application. In these screenshots of top, the columns to notice are NI, RES, and %CPU. The 'Nice' values of the rids_* processes are modified according to the number of buffers or structures queued for each process, versus the number that have been handled. The values are relative to each other, so that processes which are falling behind are given the most (lowest) priority.

```
Shell - Konsole <4>
11350.129371 Collector has processed 3178496 packets
*** MAIN LOOP time=47fb2c57, segs=172/1014 ***
11361.320359 Collector has processed 3211264 packets
*** MAIN LOOP time=47fb2c6a, segs=176/1014 ***
11371.877194 Collector has processed 3244032 packets
11384.860134 Collector has processed 3276800 packets
11399.160128 Collector has processed 3309568 packets
*** MAIN LOOP time=47fb2c88, segs=180/1014 ***
Streams: New 3393, Current 8924, Largest 438460222, CmpSAWEs 19694
Stream Sz 16k: 2687 85 51 29 20 7 12 5 6 3 36 (58298175)
11410.519292 Collector has processed 3342336 packets
*** MAIN LOOP time=47fb2c95, segs=184/1014 ***
11422.281238 Collector has processed 3375104 packets
*** MAIN LOOP time=47fb2ca6, segs=188/1014 ***
11432.313156 Collector has processed 3407872 packets
11445.948130 Collector has processed 3440640 packets
*** MAIN LOOP time=47fb2cc4, segs=192/1014 ***
11460.309344 Collector has processed 3473408 packets
Streams: New 3958, Current 9927, Largest 460202262, CmpSAWEs 22226
Stream Sz 16k: 2681 87 42 30 20 13 6 6 8 6 56 (27440921)
11475.477420 Collector has processed 3506176 packets
*** MAIN LOOP time=47fb2cd9, segs=196/1014 ***
11488.266662 Collector has processed 3538944 packets
*** MAIN LOOP time=47fb2cef, segs=200/1014 ***
11505.762056 Collector has processed 3571712 packets
GCB CB Start b484f000, Size 32b3000, Hd b484f000, Tl b484f000
11518.714184 Collector has processed 3604480 packets
*** MAIN LOOP time=47fb2cff, segs=204/1014 ***
Streams: New 3474, Current 10536, Largest 487496706, CmpSAWEs 15675
Stream Sz 16k: 2571 99 58 36 18 19 9 3 7 7 38 (67259671)
11533.210307 Collector has processed 3637248 packets
:
```

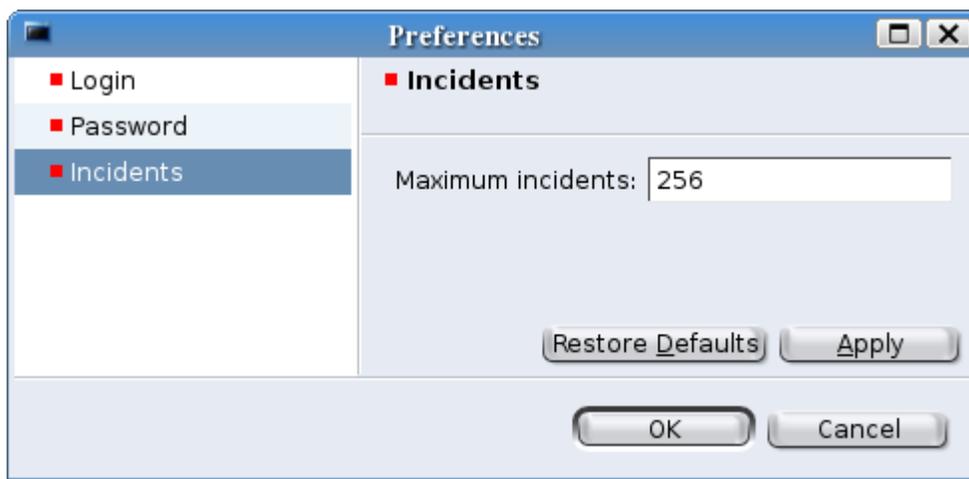
This screenshot shows the console output of the IDS. Every 32K packets, the Collector displays the total number of packets it has processed. The MAIN LOOP shows the number of 1Meg segments allocated, out of the total memory allocated (in Meg).

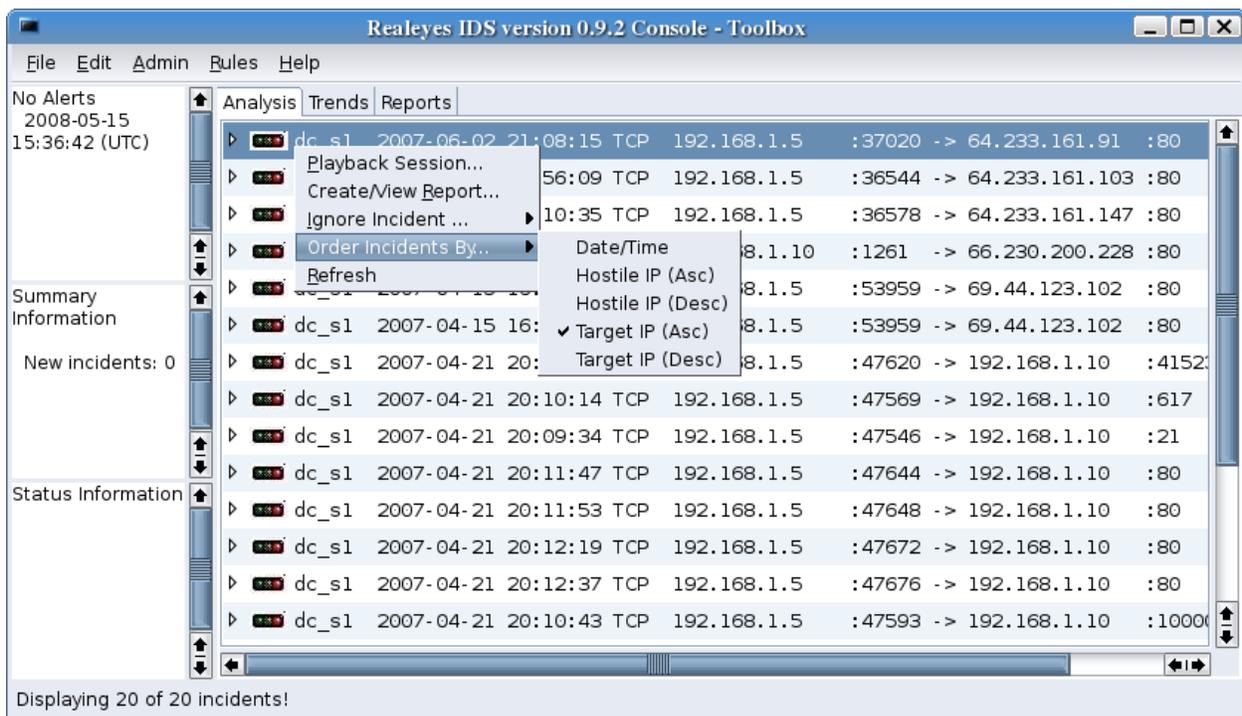
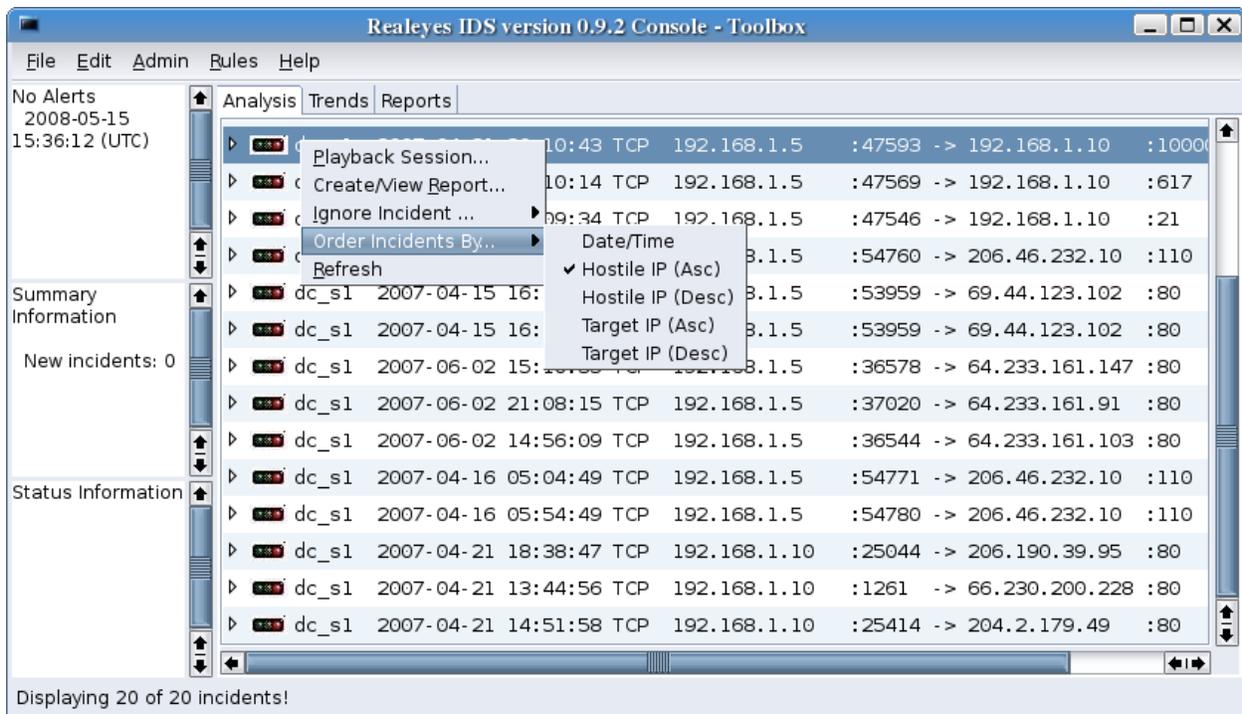
The Streams information is displayed each minute, and the interesting fields are New, Current and Largest. For the IDS, a Stream is ½ of a TCP session, so this display shows that approximately 5,000 sessions are active. The maximum number that has been recorded is just over 70,000 streams = 35,000 TCP sessions, and that is using a 733 MHz CPU and 1Gig of memory.

Near the bottom is a line that starts with GCB. This indicates that the 'Get Circular Buffer' function has been called, which means that a report is being generated for a session and needs additional buffer space.

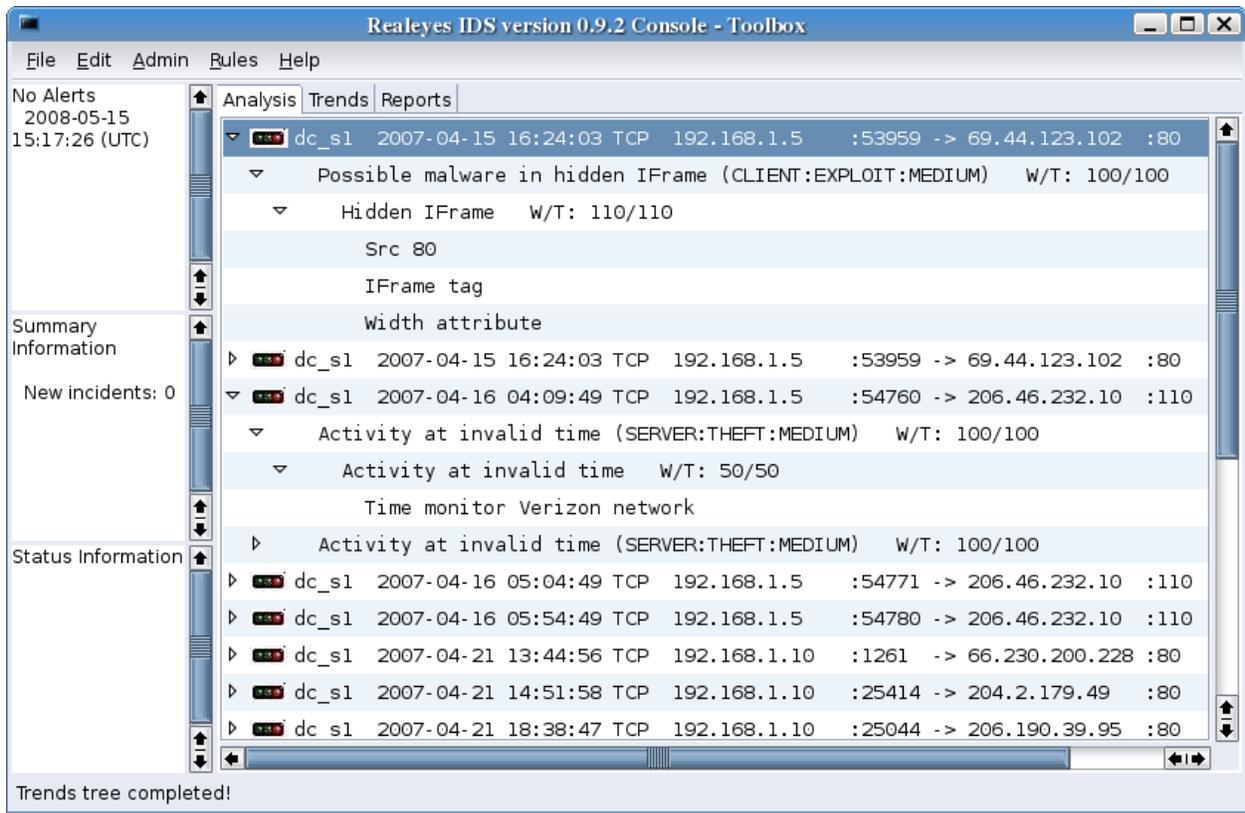


The next part of the demo was to show the results in the user interface, which is a Java application. The information for each record reported is inserted into the database, and the user interface displays it on a single line. The total number of incidents is indicated at the bottom. The maximum number displayed can be set in the preferences.



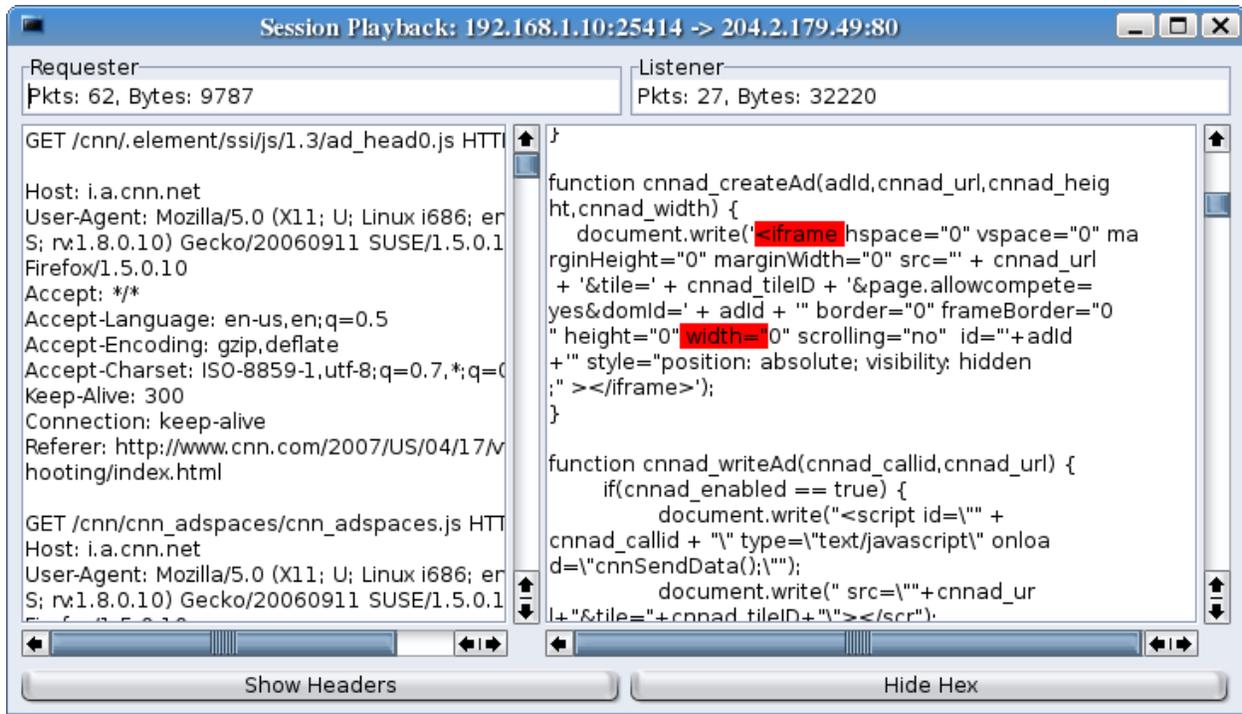


The order of the incidents can be selected from a popup menu by right clicking on the Analysis tab.



An incident is reported if the Event, Action, and Trigger thresholds in a rule definition are met or exceeded. In the Analysis tab of the main window, the Event and its Actions and its Triggers are displayed by clicking on the 'twisties'.

The 'Possible malware in hidden IFrame' Event consists of the 'Hidden IFrame' Action, which consists of the 'Src 80', 'IFrame tag', and 'Width attribute' Triggers.

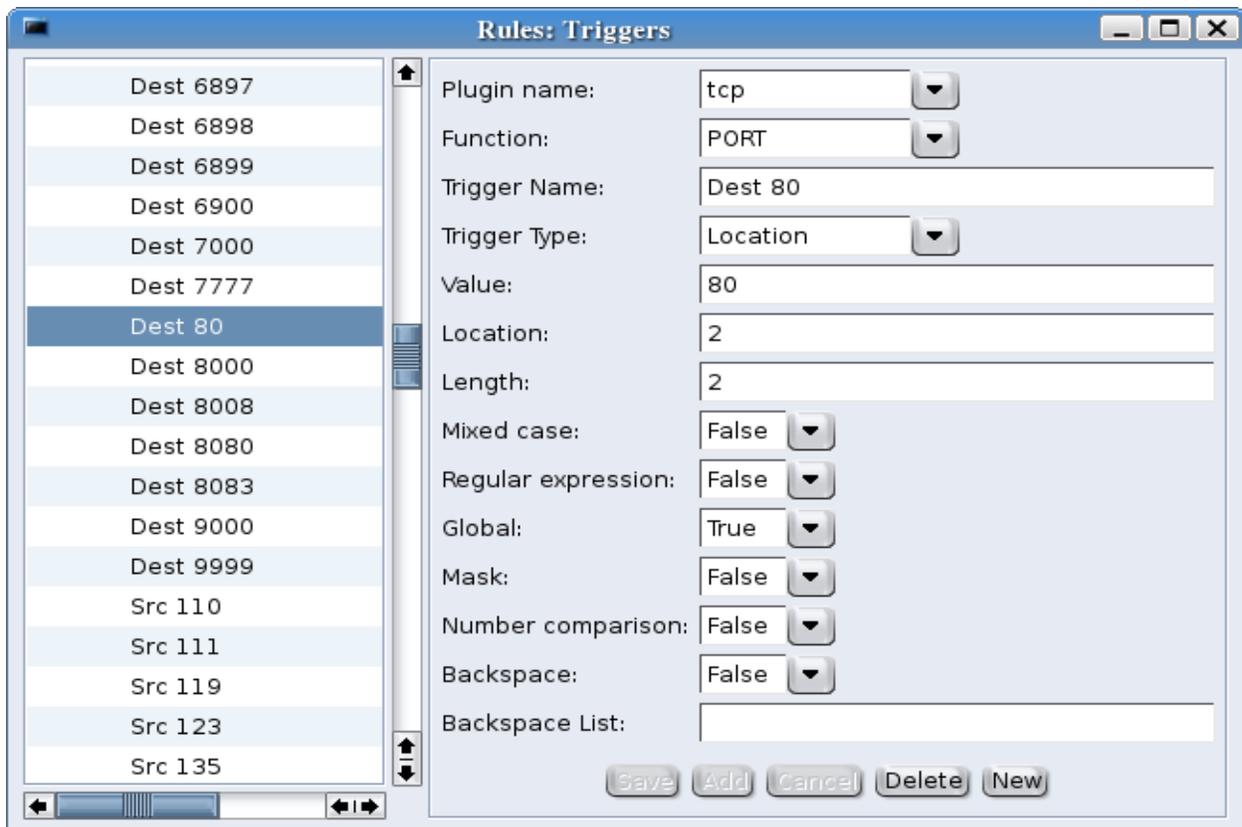
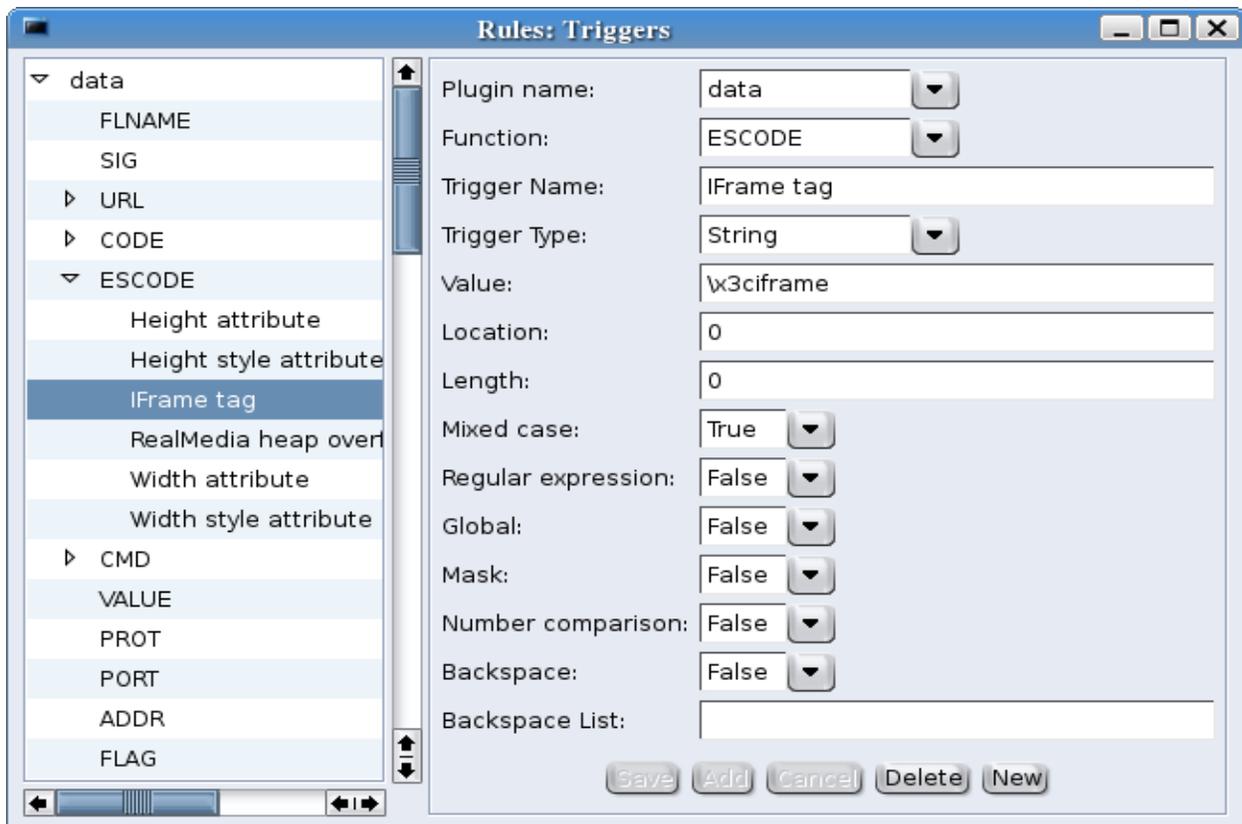


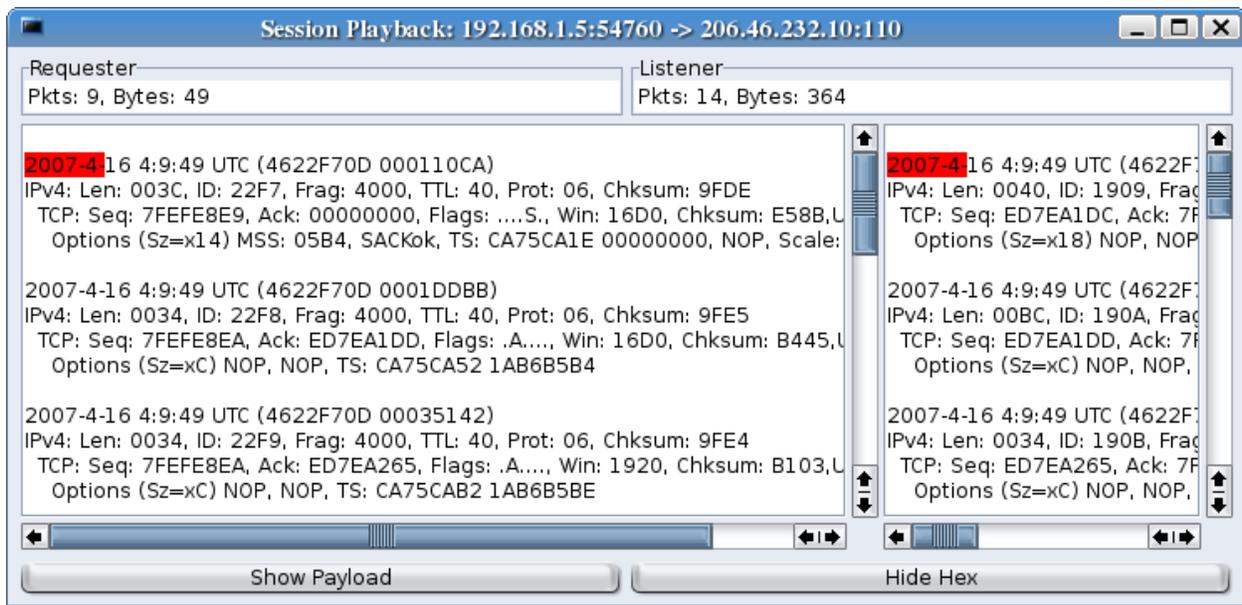
By selecting 'Playback Session' from the Analysis tab popup menu, the data captured from the TCP session is displayed. By scrolling down through the server data, the Triggers that caused the record to be reported are found to be highlighted.

Also notice that the width of the frames have been adjusted by moving the center scroll bar.

Two of the Triggers that were defined for this rule are displayed on the next page. The Plugin name defines which IDS process uses the definition, which in these cases are `rids_stra_data` and `rids_stra_tcp`. The Function allows Triggers to be organized into groups. The Trigger type defines whether it is matching on any string in a packet payload or a specific location in a TCP/IP header.

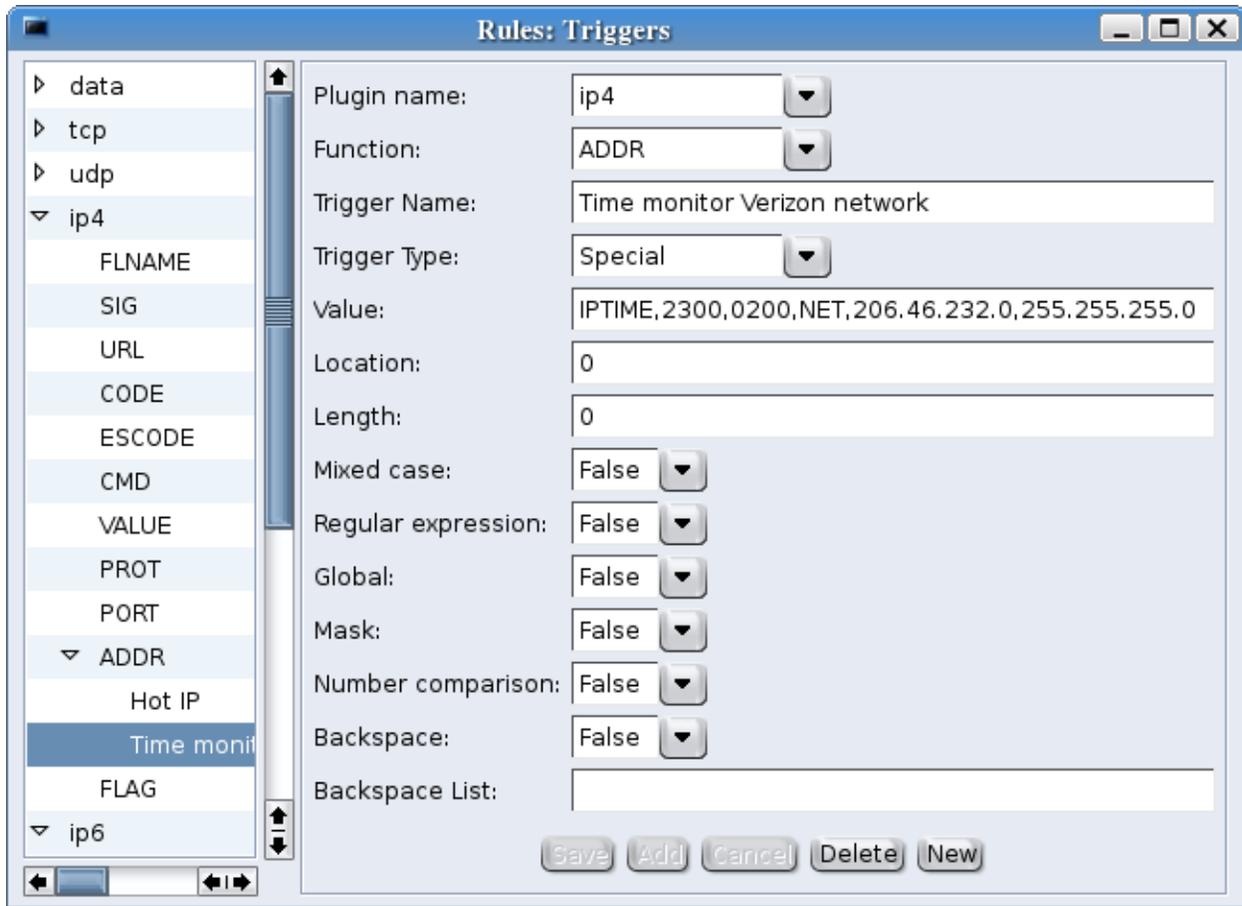
The only condition flag set for the 'IFrame tag' Trigger is to allow Mixed case. The Global flag set for the 'Dest 80' Trigger indicates that it is true for the entire session and should be reported only once.





The playback for the 'Activity at invalid time' is displaying the session headers after clicking on the 'Show Headers' button at the bottom of the window (which then becomes the 'Show Payload' button). The Trigger that was detected for this session is a Special Trigger, and is shown on the next page. It looks for activity occurring between certain times and reports everything detected.

The headers display the timestamp of the packet, the IP header, and any other appropriate header data. In this case, there are TCP headers which have options.



Special Triggers are created by writing code in the IDS plugin processes to search for conditions that are not simple string or location matches. In this case, the IPTIME Trigger looks for any activity occurring between 11:00 pm and 2:00 am, local time, on the network 206.46.232.0.

```
Shell - Konsole <4>
2008-4-21 11:23:55 UTC (480C794B 00046B5D)
IPv4: Len: 0030, ID: 0988, Frag: 4000, TTL: 6F, Prot: 06, Chksum: 007C
54110C17 00000000 7002FFFF 1B450000
020405B4 01020402

2008-3-16 21:30:39 UTC (47DD917F 000269A9)
IPv4: Len: 0040, ID: BF7E, Frag: 4000, TTL: 30, Prot: 06, Chksum: 55E3
97C159E6 00000000 B002FFFF 57CF0000
020405B4 01030300 0101080A 32108EB6 00000000 04020802
2008-3-16 21:30:42 UTC (47DD9182 0001253E)
IPv4: Len: 0034, ID: C065, Frag: 4000, TTL: 30, Prot: 06, Chksum: 5508
TCP: Seq: 97C159E7, Ack: E9B0D2EB, Flags: .A...., Win: 05B4, Chksum: D532,Urg: 0000
Options (Sz=xC) NOP, NOP, TS: 32108EBC 00000000

2008-4-5 20:32:15 UTC (47F7E1CF 000CDEA2)
IPv4: Len: 0040, ID: 5ACA, Frag: 4000, TTL: 2D, Prot: 06, Chksum: 9D13
TCP: Seq: 24091E9E, Ack: 00000000, Flags: ...S., Win: 7FFF, Chksum: AFE8,Urg: 0000
Options (Sz=x18) MSS: 05AC, EOL, EOL, EOL, EOL, NOP, NOP, TS: 00000000 00000000,
NOP, NOP, SACKok
:
```

Another Special Trigger parses the TCP options field for invalid use. This screenshot is of a few of the incidents that have been reported on this Trigger. They may be buggy code, or they may be crafted packets doing OS fingerprinting.

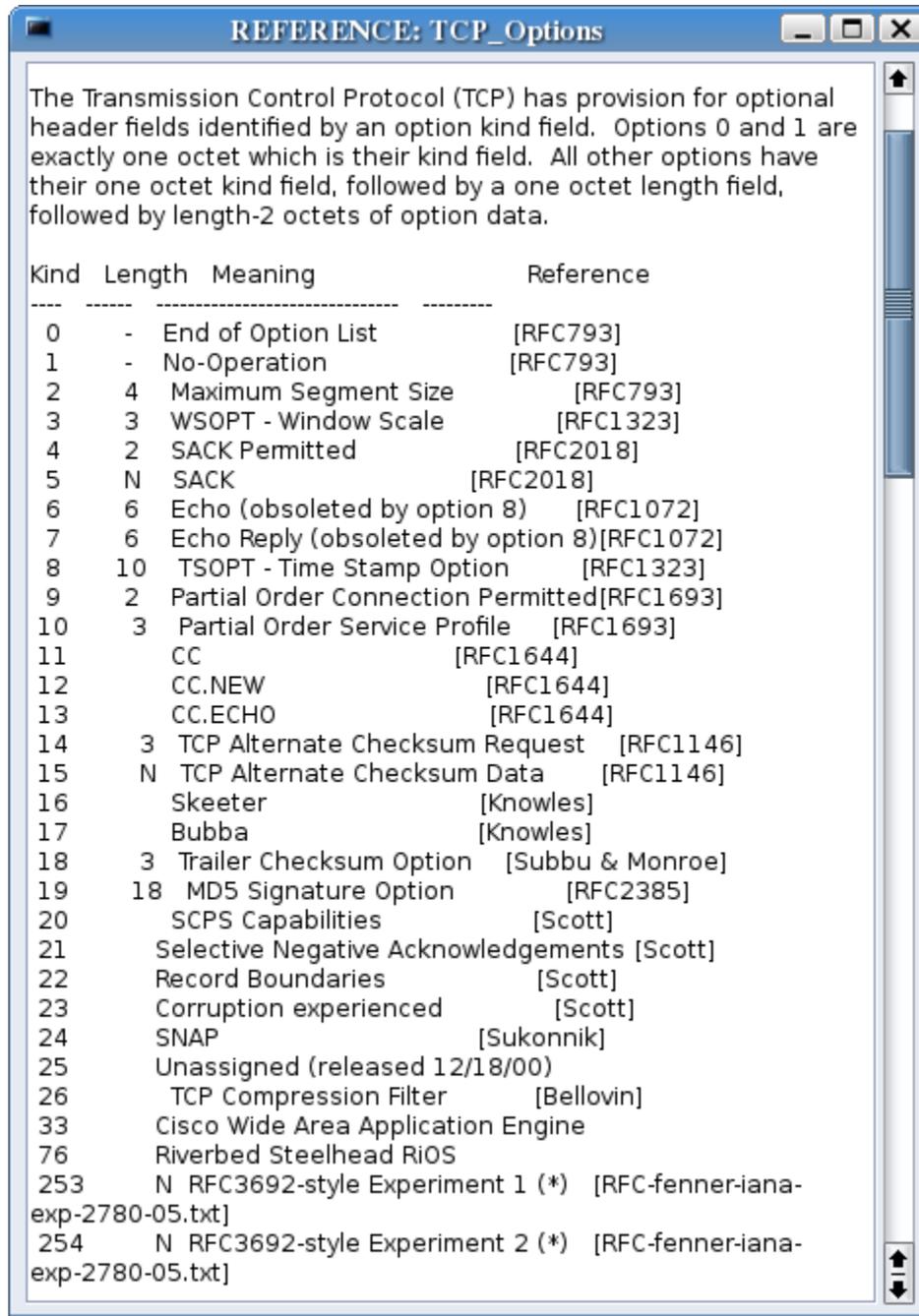
The first example displays a hex dump of the TCP header (minus the ports field). The 7 at the beginning of the third word is the number of 4 octet words in the header, including options. But the last word, '01020402' is extending beyond that length:

- 01 = NOP
- 02 = Maximum segment size option
- 04 = Length (including option and length)
- 02 = First octet of MSS value

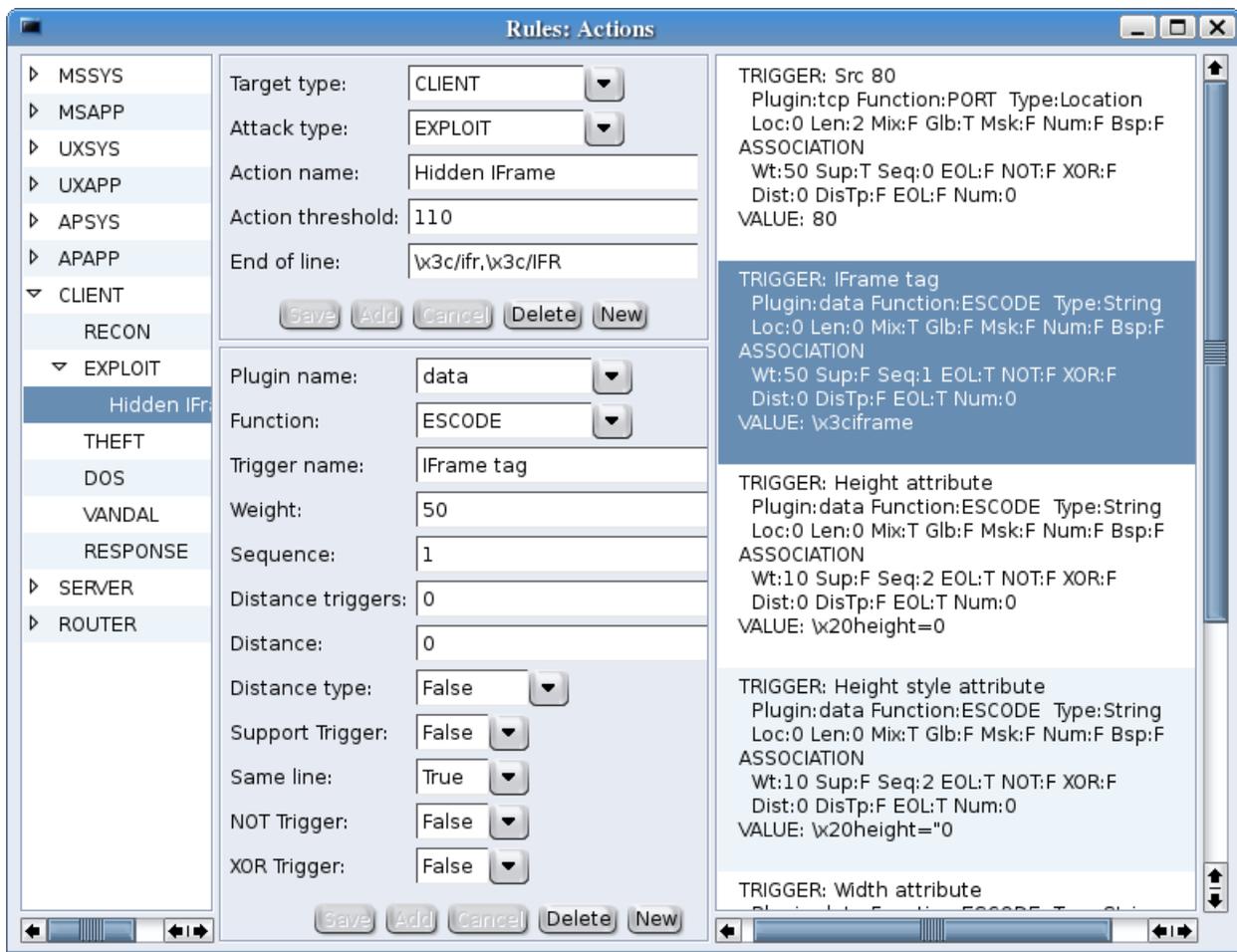
The second example option field has an invalid length in the last word, '04020802':

- 04 = SACK permitted
- 02 = Length
- 08 = Time Stamp option
- 02 = Length (should be 10 or x0A)

The third example has End of Line (EOL) options followed by more options.



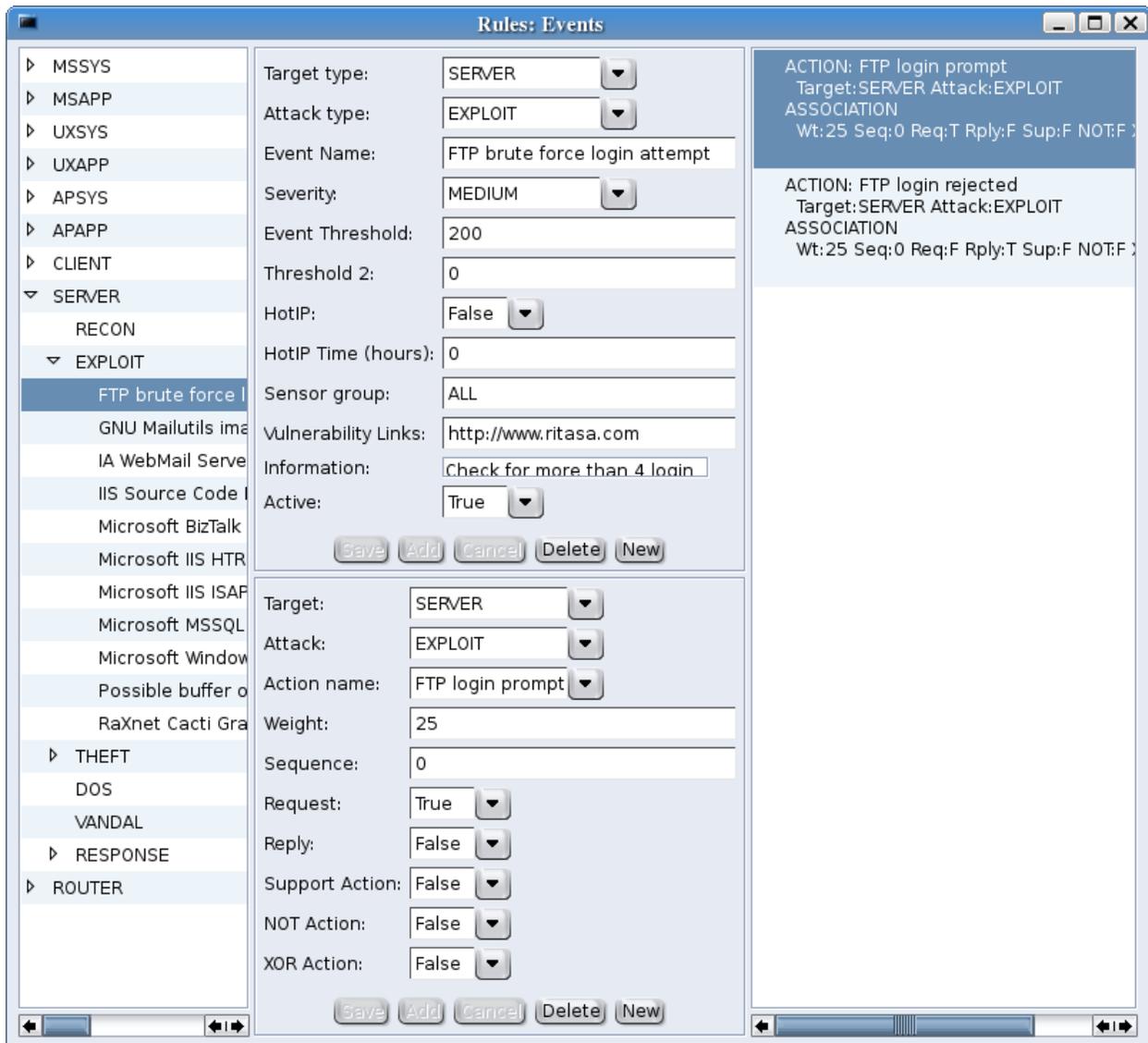
Included in the user interface help are some reference files, including the list of registered (and unregistered, ie. 33 and 76) TCP options. Any reference data may be added to the system by creating a text file, naming it something.ref, and copying it to the directory where the application code is located.



The Action definitions are more complicated because they associate Triggers with an Action. The 'Hidden IFrame' Action has six Triggers associated with it.

The highlighted 'IFrame tag' Trigger definition has a weight of 50. The total weights of detected Triggers must equal or exceed the threshold weight of 110 defined in the Action for it to be reported. This Trigger also has a sequence number defined. On the right panel, it can be seen that the 'Height attribute' Triggers have a sequence number of 2. This means that the attributes must follow the tag to be counted.

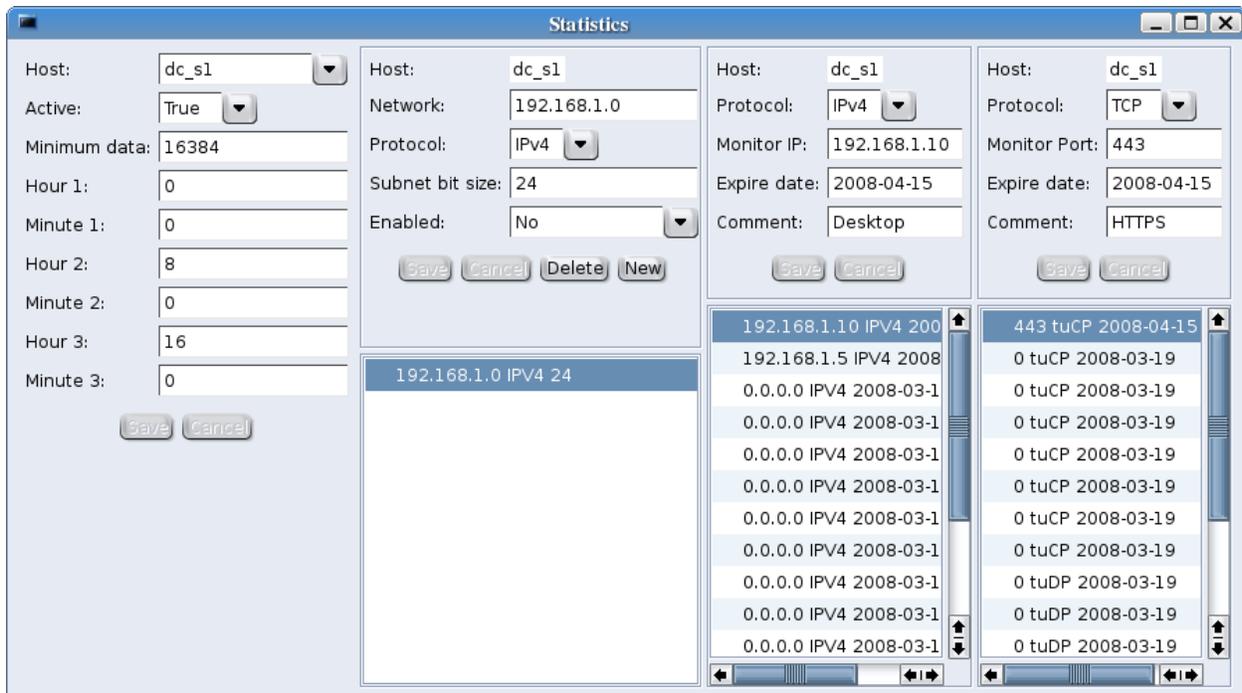
Finally, the 'IFrame tag' Trigger has the Same line flag set. The Action End of line field defines what characters end a line, in this case the end of an IFrame tag, so the attribute Triggers must be between the start and end IFrame tags.



The Event definitions are similar to the Action definitions, but with more information in the Event, which should help an analyst learn about the potential exploit.

The 'FTP brute force login attempt' Event is an example of a Request/Reply. The highlighted 'FTP login prompt' Action, is a password from a FTP client. The other Action, 'FTP login rejected', is an error reply from an FTP server. If both are detected, then the reply must immediately follow the request, based on the TCP packet's Sequence and Acknowledgment numbers.

The total weight of both rules is 50, so there must be at least 4 Request/Reply pairs for the Event to be reported. This reduces reporting user typos.



Statistics may be collected on all monitored networks. The statistics are total number of bytes transferred inbound and outbound. They are collected during three periods of the day, and the start time of each period is configurable. When the networks for statistics collection are defined, the totals for each TCP and UDP port detected are maintained until the end of the collection period, when they are transferred to be stored in the database.

It is also possible to monitor specific hosts or ports. When that is defined, the total inbound and outbound byte counts for every connection to that host or port are maintained.



The export window allows the rules definitions to be saved in a file or sent directly to the IDS hosts.

Security Report: 192.168.1.5:53959 -> 69.44.123.102:80

Report Number:

Date/Time (GMT): 2007-04-15 16:24:03

Site: DC office

Severity: MEDIUM

Protocol: TCP

Source:
 IP: 192.168.1.5
 Hostname: guitar
 Port: 53959

Destination:
 IP: 69.44.123.102
 Hostname: 69.44.123.102
 Port: 80

Events Whois Notes

Source

NetRange: 192.168.0.0 - 192.168.255.255
 CIDR: 192.168.0.0/16
 NetName: IANA-CBLK1
 NetHandle: NET-192-168-0-0-1
 Parent: NET-192-0-0-0-0
 NetType: IANA Special Use
 NameServer: BLACKHOLE-1.IANA.ORG
 NameServer: BLACKHOLE-2.IANA.ORG
 Comment: This block is reserved for special purposes.
 Comment: Please see RFC 1918 for additional information.
 Comment: <http://www.rip.net/reference/rfc/rfc1918.txt>

US - UNITED STATES

Destination

OrgName: Level 3 Communications, Inc.
 OrgID: LMLT
 Address: 1025 Eldorado Blvd.
 City: Broomfield
 StateProv: CO
 PostalCode: 80021
 Country: US

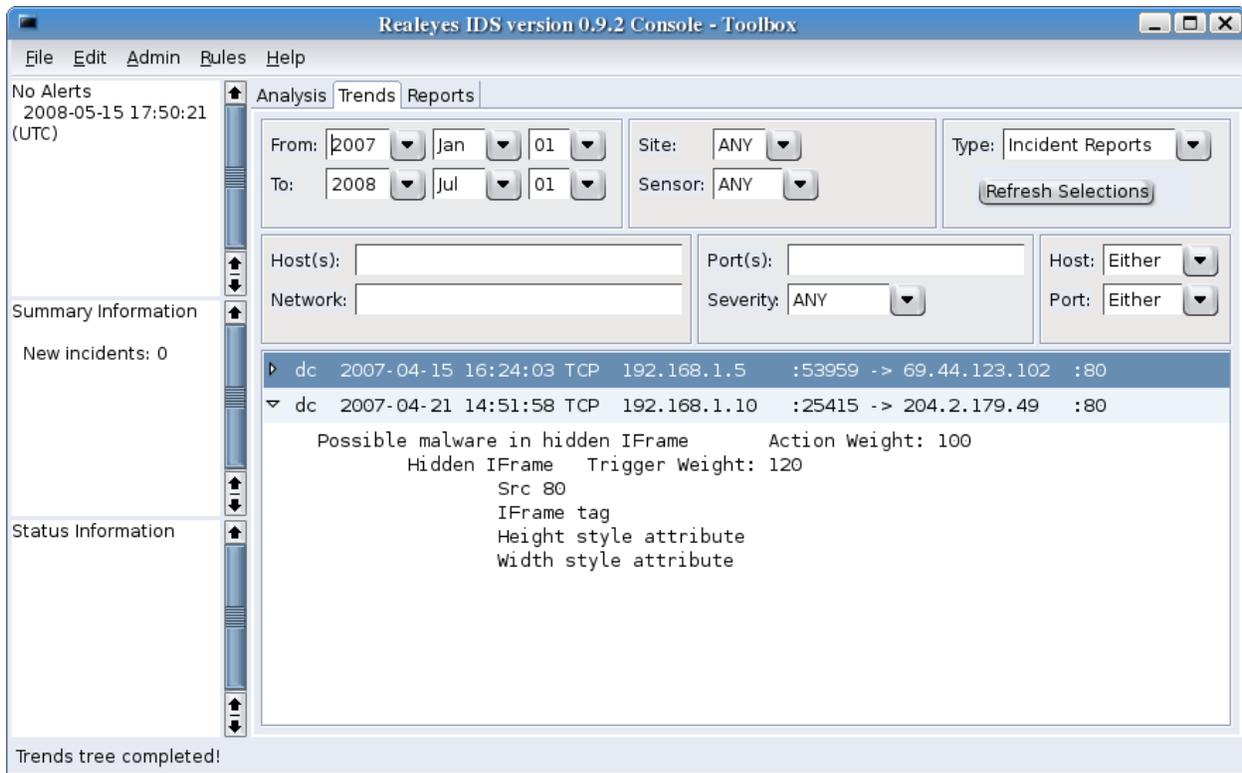
NetRange: 69.44.0.0 - 69.45.255.255
 CIDR: 69.44.0.0/15
 NetName: IANA-ORG-69.44

US - UNITED STATES

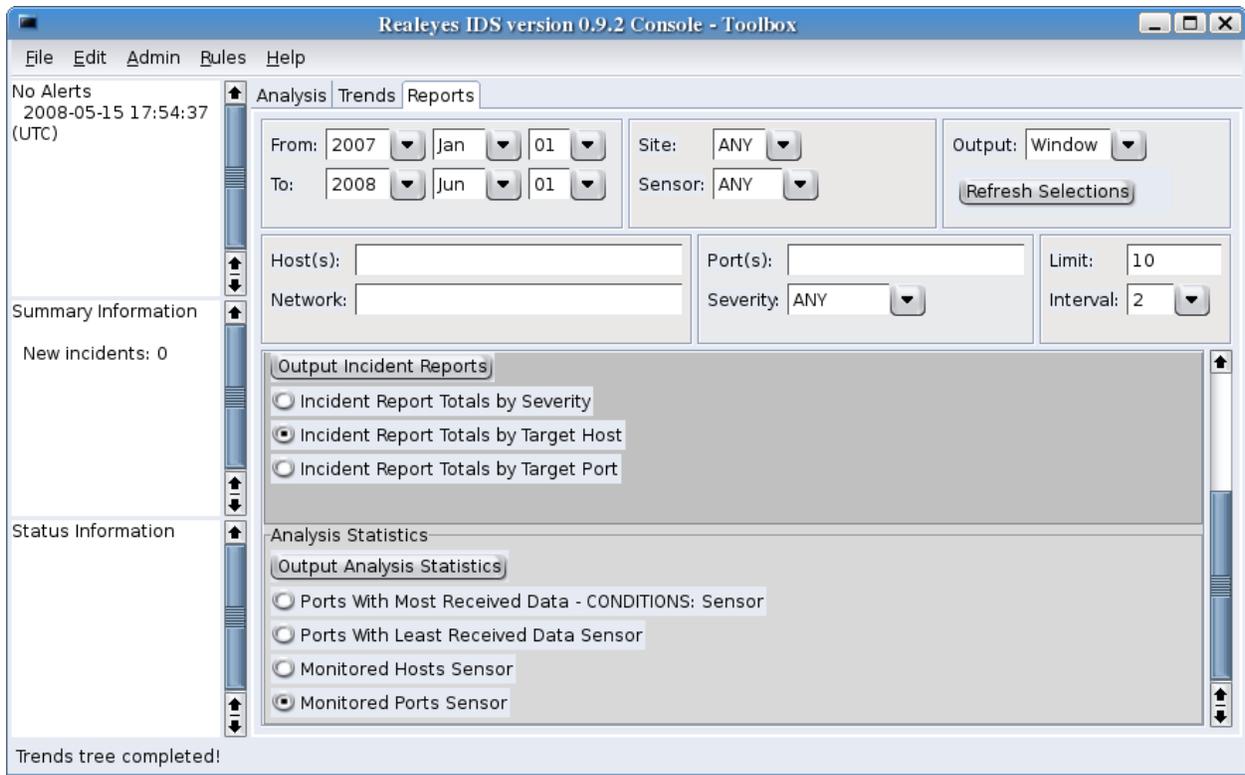
Save Report Close Report

If an incident is serious, it may be saved in a report for trends analysis. This is done by selecting the Create/View Report button on the Analysis tab. The report window has three tabs, and the Whois tab is shown in this screenshot. A whois query is run in the background to determine where both IP addresses in the reported session are registered.

The information in the left frame is automatically added, but may be modified by an analyst. Then the Save Report button is clicked to save the report to the database.



In the Trends tab of the main window, the reports may be displayed based on a variety of search criteria. The From/To dates are mandatory, but the reports may be limited to specific hosts or ports. The playback window may be displayed by using the same popup window as found in the Analysis tab.



The Report tab on the main window provides several predefined reports. The reports may display the information from the Analysis tab or Trends tab.



Report			
Ports With Most Received Data			
2007-01-01 to 2008-06-01			
Sensor: dc_s1			
2007-04-15			
Interval: 2			
TCP Ports			
Port	Data In	Data Out	
80	795240	110731	
443	508815	96538	
110	36216	3798	
Interval: 3			
TCP Ports			
Port	Data In	Data Out	
80	660636011	2147927	
8080	3516892	1499	
443	1456404	150530	
110	146147	17851	
25	2168	22122	
UDP Ports			
Port	Data In	Data Out	
34997	29938	0	
34995	20620	0	
34993	19602	0	
53	0	55878	
2007-04-16			
Interval: 1			
TCP Ports			
Port	Data In	Data Out	
80	45430186	166672	
110	47586	4197	
UDP Ports			
Port	Data In	Data Out	
35008	16393	0	
21302	0	8959680	
5000	0	103668	

The reports also display information from Statistics that may be collected. This screenshot is a display of the port statistics.

Report

Monitored Hosts _Interval 2
2007-01-01 to 2008-06-01

Sensor: dc_sl

2007-04-15

Monitored host: 192.168.1.10

Conn. Host	Port	Data In	Data Out
12.180.111.218	80	74018	13734
12.180.111.218	443	79544	17778
64.187.43.35	80	421	832
65.206.60.120	80	90187	19667
66.102.1.147	80	658	869
67.109.145.40	80	23633	924
72.9.255.178	80	44845	8806
72.14.219.99	80	1010	1464
140.90.113.200	80	0	2112
140.90.121.156	443	521	1048
140.90.121.157	443	3322	1574
140.90.121.168	443	97332	6942
206.213.211.171	80	4511	7687
206.213.211.173	443	71618	26118
206.213.253.171	80	114656	1600
206.213.253.171	443	182829	11641
206.213.253.173	443	73649	31437
209.85.165.104	80	6226	2527
216.38.80.20	80	7657	1168

This screenshot shows the results of monitoring a host. It is for the second monitoring interval of one day. This allows detailed information to be collected on connections being made to a host.

Report

Monitored Ports _Interval 2
2007-01-01 to 2008-06-01

Sensor: dc_sl

2007-04-15

Monitored port: 443 (TCP)

Conn. Host	Data In	Data Out
12.180.111.218	0	17778
140.90.121.156	0	1048
140.90.121.157	0	1574
140.90.121.168	0	6942
192.168.1.10	508815	0
206.213.211.173	0	26118
206.213.253.171	0	11641
206.213.253.173	0	31437

2007-06-02

Monitored port: 443 (TCP)

Conn. Host	Data In	Data Out
63.160.50.126	0	73053
192.168.1.10	78986	0
216.168.252.103	0	6451

This screenshot shows the results of monitoring a port. It shows all hosts external to the monitored networks that made connections using the monitored port.

```
realeyes@violin: ~ - Shell - Konsole <2>
This script will prompt for the configuration data required to
start a Realeyes IDS sensor and communicate with a
Realeyes DBD host.

!!! This script generates the following files:  !!!
!!! /etc/realeyes                               !!!
!!! realeyesIDS.conf                            !!!
!!! rae_analysis.xml                           !!!
!!! rids_collector.xml                         !!!
!!! Running it will backup existing files but  !!!
!!! create all new values from input data.     !!!

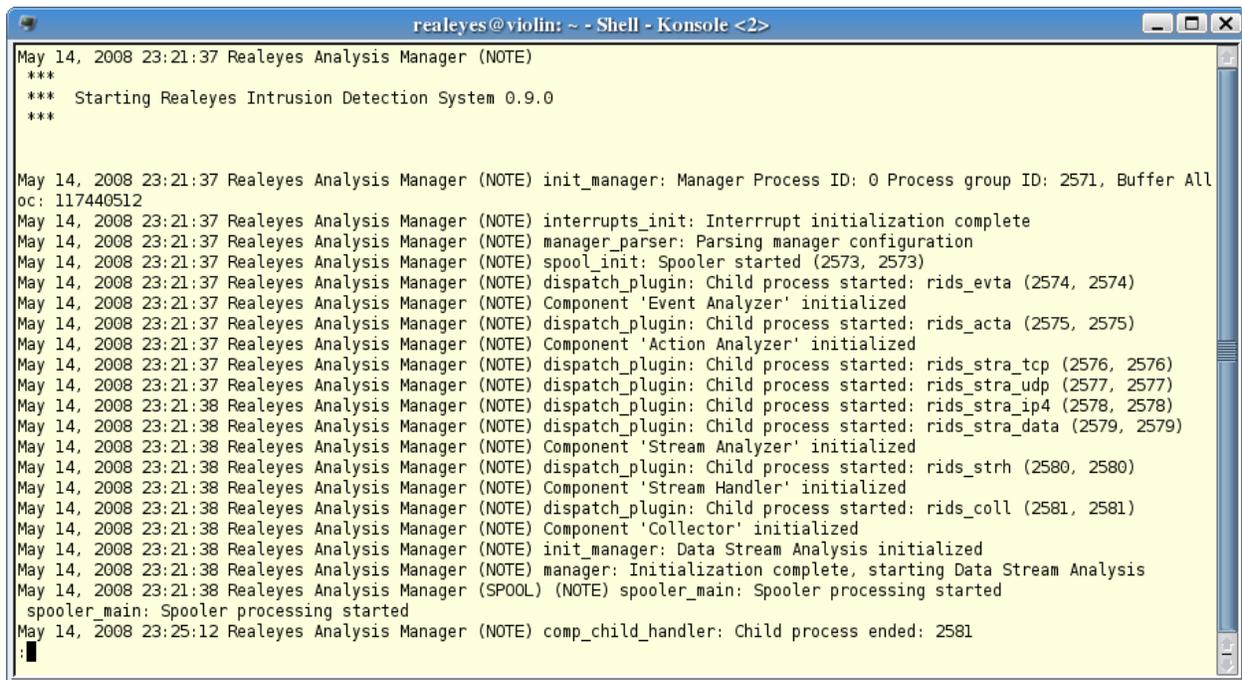
Enter 'q' to quit at any time.

Enter home directory [/usr/share/realeyes/realeyesIDS]:
Enter Memory allocation (percent of real) [33]:
Enter IDS sensor hostname: myhost
Enter IDS sensor IP address: 192.168.1.2
Enter IDS monitoring interface [<intfc>|eth0]:
The Realeyes IDS will monitor the following protocol selections:
  1) TCP only
  2) TCP and UDP only
  3) All IP protocols
Enter protocol selection [<prot>|1]: 2
Enter Manager logging options [W(arn)|I(nfo)|S(tat)|D(one)]: w
Enter Manager logging options [W(arn)|I(nfo)|S(tat)|D(one)]: d
Use encryption to DBD host [Y|n]: n
Spooler home directory [<dir>|spool]:
Port for data transmission [<port>|1332]:
Port for control transmission [<port>|1333]:
Enter DBD hostname: dbdhost
Enter DBD IP address: 192.168.1.3
Another DBD definition [y/N]:
Enter Plugin logging options [W(arn)|I(nfo)|S(tat)|D(one)]: w
Enter Plugin logging options [W(arn)|I(nfo)|S(tat)|D(one)]: d

#RIDS_CFG_DIR RIDS_CFG_DIR=/etc/realeyes
#RIDS_HOME RIDS_HOME=/usr/share/realeyes/realeyesIDS
#RIDS_MEM_ALLOC RIDS_MEM_ALLOC=33
LOC_HOST myhost
LOC_ADDR 192.168.1.2
MGR_MSG Warning
ENCRYPT NO
SPOOL_HOME spool
DATA_PORT 1332
CTL_PORT 1333
DBD_HOST dbdhost
DBD_ADDR 192.168.1.3
PLUG_MSG Warning
IDS_HOME /usr/share/realeyes/realeyesIDS
LOG_DIR /var/log/realeyes
SPOOL_USER reids
INTFC eth0
PROTOCOLS ip and (tcp or udp)

Accept these values [Y|q]: █
```

The previous screenshot shows part of the installation and configuration script for the IDS. There are scripts for each of the four components that eliminate the need for editing configuration files. However, all configuration is maintained in text files that can be edited if necessary.



```
realeyes@violin: ~ - Shell - Konsole <2>
May 14, 2008 23:21:37 Realeyes Analysis Manager (NOTE)
***
*** Starting Realeyes Intrusion Detection System 0.9.0
***

May 14, 2008 23:21:37 Realeyes Analysis Manager (NOTE) init_manager: Manager Process ID: 0 Process group ID: 2571, Buffer All
oc: 117440512
May 14, 2008 23:21:37 Realeyes Analysis Manager (NOTE) interrupts_init: Interrupt initialization complete
May 14, 2008 23:21:37 Realeyes Analysis Manager (NOTE) manager_parser: Parsing manager configuration
May 14, 2008 23:21:37 Realeyes Analysis Manager (NOTE) spool_init: Spooler started (2573, 2573)
May 14, 2008 23:21:37 Realeyes Analysis Manager (NOTE) dispatch_plugin: Child process started: rids_evta (2574, 2574)
May 14, 2008 23:21:37 Realeyes Analysis Manager (NOTE) Component 'Event Analyzer' initialized
May 14, 2008 23:21:37 Realeyes Analysis Manager (NOTE) dispatch_plugin: Child process started: rids_acta (2575, 2575)
May 14, 2008 23:21:37 Realeyes Analysis Manager (NOTE) Component 'Action Analyzer' initialized
May 14, 2008 23:21:37 Realeyes Analysis Manager (NOTE) dispatch_plugin: Child process started: rids_stra_tcp (2576, 2576)
May 14, 2008 23:21:37 Realeyes Analysis Manager (NOTE) dispatch_plugin: Child process started: rids_stra_udp (2577, 2577)
May 14, 2008 23:21:38 Realeyes Analysis Manager (NOTE) dispatch_plugin: Child process started: rids_stra_ip4 (2578, 2578)
May 14, 2008 23:21:38 Realeyes Analysis Manager (NOTE) dispatch_plugin: Child process started: rids_stra_data (2579, 2579)
May 14, 2008 23:21:38 Realeyes Analysis Manager (NOTE) Component 'Stream Analyzer' initialized
May 14, 2008 23:21:38 Realeyes Analysis Manager (NOTE) dispatch_plugin: Child process started: rids_strh (2580, 2580)
May 14, 2008 23:21:38 Realeyes Analysis Manager (NOTE) Component 'Stream Handler' initialized
May 14, 2008 23:21:38 Realeyes Analysis Manager (NOTE) dispatch_plugin: Child process started: rids_coll (2581, 2581)
May 14, 2008 23:21:38 Realeyes Analysis Manager (NOTE) Component 'Collector' initialized
May 14, 2008 23:21:38 Realeyes Analysis Manager (NOTE) init_manager: Data Stream Analysis initialized
May 14, 2008 23:21:38 Realeyes Analysis Manager (NOTE) manager: Initialization complete, starting Data Stream Analysis
May 14, 2008 23:21:38 Realeyes Analysis Manager (SPOOL) (NOTE) spooler_main: Spooler processing started
spooler_main: Spooler processing started
May 14, 2008 23:25:12 Realeyes Analysis Manager (NOTE) comp_child_handler: Child process ended: 2581
:
```

Logs of the IDS and DBD are kept in the directory, /var/log/realeyes. This screenshot shows the log of the main IDS process. All times are based on Universal Time Coordinates (UTC) so that if the system is distributed across time zones, all reports are synchronized. So the demonstration actually occurred at 7:21 Eastern USA time.

```
realeyes@violin: ~ - Shell - Konsole <2>
realeyesIDS.xml(5)           RealeyesIDS config           realeyesIDS.xml(5)
NAME
  realeyesIDS.xml - Realeyes Intrusion Detection System configuration
SYNOPSIS
  The Realeyes Intrusion Detection System configuration files are in XML
  format. The manager and each plugin level have a unique Data Type Defi-
  nition. The names of the DTDs are:
  . rae_analysis.dtd
  . rids_collector.dtd
  . rids_stream_handler.dtd
  . rids_stream_analyzer.dtd
  . rids_action_analyzer.dtd
  . rids_event_analyzer.dtd
DESCRIPTION
  ANALYSIS: The manager configuration file is contained in the Analysis
  tag and the namespace is rae_dsan.

  Manager: Defines the Manager, which initializes memory management and
  child processes

  MgrHost: The manager hostname

  Attribute Proto: IP protocol (IPV4 | IPV6)

  MgrHome: The manager home directory

  ConfigDir: Concatenated with configuration filenames to create an
  absolute path for all configuration files
Manual page realeyesIDS.xml(5) line 1
```

This screenshot shows the man page for IDS configuration files. The following man pages for the IDS and the DBD are included in the packages:

- Configuration scripts
- Configuration files
- Startup scripts

Demonstration of The Realeyes Intrusion Detection System

Jim Sansing
May 14, 2008

For more information, see the project website and blog:

<http://realeyes.sourceforge.net>

<http://realeyes-tech.blogspot.com>