Global Information Assurance Certification Paper

Augmenting Intrusion Detection Through Network Analysis

GIAC Security Essentials
Certification (GSEC)
Practical Assignment
Version 1.4c

Option 1 - Research on Topics
in Information Security

Submitted by: Lance Megyesi
Location: SANS Seattle, Oct 2004

Table of Contents

List of Figures

Abstract/Summary

"An intrusion detection system (IDS) inspects all inbound and outbound network activity and identifies suspicious patterns that may indicate a network or system attack from someone attempting to break into or compromise a system."[1]

Using the above definition, IDS systems are analogous to the classic detective story. Like the victim going through a book of mug shots these systems search through volumes of data, comparing it to a list of known profiles or signatures. When the system matches a signature an alarm is sent to a console and alerts an analyst. What happens if the IDS system is not aware of a 'new criminal'? Until someone reports the existence of this 'new criminal' and the signature database is updated, the IDS system is blind to this new threat. This situation forces the security analyst to resort to other means of detection or they won't even realize that an event may have happened.

By understanding the baseline traffic patterns in the network the security analyst can enhance the identification of attacks when used with signature based intrusion detection. The security analyst must have insight to what constitutes normal traffic and what is abnormal traffic. When this information is known for any given host it makes identifying attacks much easier. This paper explores the benefits of such analysis and introduces some of the tools used to perform network behavioral analysis or anomaly detection.

For the purposes of this paper all listed IP addresses and host names have been edited or blurred to provide some level of anonymity. This paper contains examples of numerous commands and programs. Commands and programs that must be entered from a Linux command-line-interface are presented in an italicized-bolded font. All references to commercial products within this paper do not constitute an endorsement of the product and are only referenced to provide examples.

Introduction

Over time the network has become the backbone of any business. With the globalization of the marketplace even small companies found that they must connect to the Internet in order to conduct business. As the Internet grew and more companies made this connection, new threats targeted their most important asset, information. Information assets exposed to the Internet, became vulnerable to theft or destruction from a whole new group of individuals with varying purposes.

1. Webopedia.com. 2004. URL: http://www.webopedia.com/TERM/I/intrusion_detection_system.html (Jan 2005)

To help combat these threats, technologies were developed to reduce the risk of attack. Current security technologies fit into the following categories:

- Firewalls
- Network and Host vulnerability assessment tools
- Network and Host intrusion detection systems
- Anti-virus systems
- Access Controls and Identity Management systems

Each of these technologies was built to combat a specific security problem. The most basic of these is 'how do I keep someone out of my data"? The solution
was the firewall. Next, technologies were introduced to look for changes on networks or hosts; then systems to report on known vulnerabilities and finally anti-virus solutions appeared to aid in the combat of worms, viruses and Trojans.

As the security market matured, technologies were built independently of each other. Each device deployed was distinct and had its own console for management and event reporting. Event logging was limited and reporting often suffered. The security analyst had to be an expert in packet analysis to fully understand what was happening. In addition the security analyst had to be fluent with multiple systems and the way they worked in order to understand what was happening on the network.

Security Event Monitoring

As these security technologies were placed within the network infrastructure analysts soon became overwhelmed with the amount of raw data being collected on a daily basis. Much of this data contained 'false positives' and the analyst had to determine whether to act on an event. With the large number of potential events occurring on a daily basis the security analyst could effectively review only a small portion of the presented data. This often led to missed attacks because they were not readily visible, a higher alert showed up or the analyst reacted to an event that was already over.

To overcome this challenge, security vendors introduced event correlation systems. The purpose of this type of system is to aggregate security data from different network devices and determines the relative importance of a single event. The most important events are escalated to the security analyst for analysis and review. Although these systems do provide some relief to the amount of information presented, there is still a constant battle of having to manually assess the data and seek critical events. Correlation engines are expensive to implement and like traditional IDS systems require a great quantity of time to properly tune it.

The type of data collected from security systems is still 'packet centric', meaning

as data packets enter the network they are compared against a set of rules to determine what to do with the packet. Firewalls look at the packet to determine if it should be allowed to pass. Intrusion Detection systems look for specific signature patterns within a packet or data stream to determine if an attack is occurring. Anti-Virus systems look for signature patterns with data stream to determine if a virus is present. While effective, this type of analysis only presents half of the picture as no two systems look at the same packet for the same reason.

Network Traffic Monitoring

The other half of the picture can be found with the network analysts. For years they have been faced with the problem of squeezing more packets through a limited network pipe. End users complained that their applications were under performing and attributed it to some sort of network problem. Trying to get a handle on what was happening at the application layer, networking companies participated in the development of standards to monitor traffic patterns. These standards were then incorporated into network devices and slowly placed within the infrastructure as businesses replaced aging equipment.

Three main choices for traffic monitoring are RMON, NetFlow® and sFlow. Each of these standards provides a method for a network device to monitor and decode IP packets, to collect measurements on the host conversations and to provide near or real time analysis. While each standard is implemented in a different fashion they all provide similar statistics on network traffic that can be useful to the security analyst.

These statistics fall into the following categories:

- Real-time Segment statistics – these statistics track and trend traffic patterns for a given network segment. This includes but is not limited to: Bandwidth Utilization%; Bytes and Octets; Packets; Broadcasts, Multicasts, and Unicasts; CRCs; and Jabbers.

- Real-Time Top Talkers – gives information on top stations receiving, sending and total traffic. This includes percent of total traffic, total packets sent or received, broadcasts, multicasts (per second) for each station on your LAN.

- Traffic Matrices and Protocol Statistics - Breakdown of all traffic by protocols and sub-protocols that represent a snapshot of the load on the network.

The primary focus of the network monitoring tools is geared towards solving network performance problems. Network analysts create all kinds of graphs showing average throughput, bandwidth utilization, error rates, etc. While they collect a lot of information the network analyst only looks at a portion of the data

on an ongoing basis. With the right tools in place and given access to the detailed network traffic data available, the security analyst could add another tool to his/her arsenal for intrusion detection.

Normally the security analyst would typically collaborate with the network analyst when a situation occurred, for example a zero day virus outbreak. This interaction usually occurred after the virus was detected either through the detection of some network performance issue or other mechanism. Once a problem was identified the security group would begin to pour over the graphs provided by the network group. The two teams would try to determine when the traffic started, where it first showed up and where the traffic was flowing. Packet analyzers would be started, performance data would be reviewed and access lists would be installed, all in an effort to contain the virus. Once contained the process of getting the infection cleaned would begin. In addition the team would establish methods to mitigate future events.

This exercise has occurred many times in many different IT shops around the world. If the security group had been knowledgeable about current traffic patterns and had views into the communications flows they may have spotted the event before it escalated into a scale infection. Inevitably security teams always go back to network performance data to solve or combat an event. By implementing systems that collect and baseline long term network performance data, anomalies can be identified, reviewed and mitigated.

Combining Security Events with Traffic Monitoring

When looking at network performance data from both a traffic and security perspective, a whole new picture of network communications is presented. Each team is given a comprehensive view of all traffic that is flowing. As baseline information begins to develop, things will pop out and abnormalities will be looked at. Each of these abnormalities will be corrected or deemed normal. By continual using this form of a review process new items will be presented as they appear, not just after an IDS signature update. The security and networking groups will become proactive in keeping things in check and resolving issues that were not visible before the baselines were established.

Behavioral Analysis: Anomaly Detection

Anomaly detection is generally watching something and comparing that against an expected behavior, if it does not match this good behavior then we alert on the difference. This is different than normal signature base detection where we are watching something bad and then reporting on it. The something being watched can be a device (host, router, and switch), an application, a network (traffic pattern, protocol, addresses) or a set of users. One of the main differences between anomaly detection and other forms of detection is that a baseline for normal patterns must be first established. Anomaly detection will only work when there is something to compare to. You must establish the

baseline patterns so you can detect the differences.

Anomaly detection can be broken into three different categories: behavioral, traffic pattern or protocol.

Behavioral detection looks for changes in usage, for example a use; Jim is approved and normally logs in to host Gumby. When Jim attempts to log into host Pokey an alert would be triggered. Another form of behavioral detection is a credit card fraud detection system. This system will compare the current transaction against past purchases. If the pattern is different, a very large purchase or a number of quick transactions on a seldom used card, a red flag will be raised.

Traffic pattern anomaly detection uses statistics against current traffic loads for a known device and watches for anomalies. For example a system that normally transmits about 500Mb of data via ftp on a daily basis is now generating 700Mb of data. Using statistics this system would trigger an alert when the traffic generated exceeded a set standard deviation. This would allow for some flexibility in the quantity of data that could normally flow out of the system but would definitely let you know when things were really out of norm.

Protocol anomaly systems generally are focused on what is running on the network and compares the traffic against a list of approved protocol ports. For example a given web server is only allowed to talk on an SSL port. When this web server begins to respond on http traffic the detection system flags this as an anomaly. This could indicate that there is an invalid configuration, an unapproved change was implemented or at the very least someone brought up an unauthorized protocol for that host and further investigation is required.

Because anomaly detection systems are based on normal patterns that are constantly updated and do not require predefined signature files, they can help in the detection of zero day attacks. These systems also can defend against attacks that change patterns as they spread through out the network.

The downside of these systems is getting them to watch the information you are most interested in. You must be able to identify and build a baseline that the detection system can compare against. Because the environment is constantly changing you must be able to keep the baseline up-to-date as well. Another area to be aware of is that these systems only give an indication that something is amiss; their alerts may be vague or categorized.

Available tools used in anomaly detection

There are a number of tools available both commercially and from open-source, which can be implemented to generate network baselines statistics. These statistics can then be used on a daily basis to watch for anomalies. While

commercial tools are more polished and are easier to implement, the cost can be prohibitive. We will look at a few open source packages that provide a base set of tools with which to build an anomaly detection system.

Open Source Tools

Flowtools

Flowtools is a collection of programs that provides a command line mechanism for the collection, processing, reporting and exporting of data generated by using NetFlow® records. NetFlow® records are typically generated by routers and switches manufactured by Cisco® and Juniper®. For sites that do not use routers and switches that can natively generate NetFlow® records, a probe can be setup on a stand-alone Linux system using the application, Fprobe. Fprobe is a daemon that watches current network traffic, generates NetFlow® records and forwards them to a NetFlow® collector.

Flowtools is a set of approximately 20 command line applications used to process flow records, the output is text based and requires a fair amount of analysis to provide trending results. Other applications can be layered on top of Flowtools to provide graphical trending or can save flow records to a database. Flowtools are executed on a Linux or BSD environment. Some of the programs contained in the Flowtools suite are:

- Flow-capture – this program is used to collect NetFlow® records and store them in a file.
- Flow-filter – this program is used to filter flow data based on user criteria.
- Flow-print – this program is used to display flow data in ASCII format either to a screen or text file.
- Flow-cat – this program is used to concatenate multiple files and/or directories of flow data.
- Flow-stat – this program is used to generate various reports from the collected flow data.

It is important to note that flow data only shows unidirectional information and does not include both sides of the conversation. In the examples below, displayed flow records are from the perspective of source address flowing to the destination address. The flow data was stored on a Linux host named labhost and files were written to the following location:

/var/flow/*hostname*/*year-month*/*year-month-day*

Flow data for a router named labrtr on Jan 10, 2005 is stored in the directory:

/var/flow/labrtr/2005-01/2005-01-10/

Once flow data records are being generated, either via a properly configured

Cisco router or Fprobe application, collection can begin. Using a Linux system with Flowtools installed, issue the following command to begin capturing flow records sent to UDP port 2055 for any source and destination:

labhost:/var/flow# ***flow-capture –w labrtr/2005-01/2005-01-10 0/0/2055***

The flow records are captured directly to disk for storage. To view flows captured you combine the flow-cat and flow-print tools. For example to view flows for 01/10/05 issue the following command:

labhost:/var/flow# ***flow-cat labrtr/2005-01/2005-01-10/*|flow-print***

```
srcIP           dstIP           prot  srcPort  dstPort  octets      packets
172.27.1.1      192.168.1.2     1     0        771      224         1
192.168.10.10   192.168.1.3     1     0        2048     84          1
192.168.1.3     192.168.10.10   1     0        0        84          1
192.168.1.62    192.168.132.7   6     1266     20       1017178     1071
192.168.132.7   192.168.1.62    6     445      1266     93          1
192.168.10.10   204.153.128.34  1     0        2048     84          1
192.168.10.10   204.153.128.34  17    36581    161      104         1
192.168.10.10   204.153.128.34  17    36582    161      104         1
192.168.10.10   192.168.1.2     1     0        2048     168         2
```

The above listing displays conversations between source and destination hosts, protocol, source and destination ports, number of bytes transferred and packet count. For example the line that has been italicized shows that there was an ftp initiated by host 192.168.1.62 to host 192.168.132.7 and over 1 megabyte of data was transferred.

By combining flow-cat with flow-stat we can begin to pull additional information out of the flow data. The flow-stat utility has 33 different report formats that can be displayed. Please see the appropriate man page for a list of all the different formats. In the examples below, we will look at the top services being used, the top destination services and top talkers.

The following report is a list of the top services used on Jan 10, 2005. This list shows the port name or number, the number of flows associated with the port, the amount of data transferred to/from the port and the number of packets associated. This report is generated by concatenating all flow records stored in the /var/flow/labrtr/2005-01/2005-01-10/ subdirectory using the flow-cat utility. The –p switch tells flow-cat to preload the headers and preserve stored metadata. Once the records are concatenated they are piped to flow-stat. Flow-stat is manipulated by a number of switches; the –f switch selects the report format, in this case the –f 7 selects all UDP/TCP ports. The –n switch tells flow-stat to display symbolic names where appropriate and the –S switch determines what field will be used for sorting. Finally the entire output is piped through the utility, more. The more utility displays output one screen at a time.

Top Service Ports being used:

labhost:/var/flow# ***flow-cat -p labrtr/2005-01/2005-01-10 | flow-stat –f 7 -n –S 2 | more***

```
#  --- ---- ---- Report Information --- --- ---
#
# Fields:    Total
# Symbols:   Enabled
# Sorting:   Descending Field 2
# Name:      UDP/TCP port
#
# Args:      flow-stat -f 7 -n -S 2
#
#
# port      flows              octets             packets
#
http       90799              2281766202          19064320
jetdirect   8732               686843840          666814
snmp        46952               266221344          522377
1594        14916               242925481          304485
1100        960                66355120           251945
1089        2653                64270649           570141
1085        3236                59940551           835415
1092        3175                58093525           511869
1091        1496                52953324           458927
1087        2611                50880918           517897
1088        2078                44250542           670429
1232        534                42011491           87017
```

Below is a report on the top destination ports. Again the syntax for the commands are similar to the example above except the format selected is for UDP/TCP destination ports. This is indicated by changing the –f switch and including a 5 for the format type. The same information is displayed as in the previous example.

Top Destination Service Ports being used:

labhost:/var/flow# ***flow-cat -p labrtr/2005-01/2005-01-10 | flow-stat –f 5 –n –S 2 | more***

```
#  --- ---- ---- Report Information --- --- ---
#
# Fields:    Total
# Symbols:   Enabled
# Sorting:   Descending Field 2
# Name:      UDP/TCP port
#
# Args:      flow-stat -f 5 -n -S 2
#
# port      flows              octets             packets
#
jetdirect   8667               686535549          666458
snmp        46236               263902673          517117
```

```
ansoft-lm-  4053          78442454        860500
1100        955           66354865        251940
1089        2646          63994585        565603
1085        3229          59940009        835405
1092        3164          53973745        505397
1091        1495          52785551        457968
1087        2602          50783508        516343
1088        2067          44230649        670152
```

Below is a report that contains a list of the top talking IP addresses. The listing show the unidirectional conversation with the most traffic transmitted from the source IP address to the destination IP address. The syntax is the same as before except a different format is chosen. In addition to the source and destination IP address the number of flows, the number of bytes transferred and number of packets are displayed.

Top Talking Hosts:

labhost:/var/flow# ***flow-cat -p labrtr/2005-01/2005-01-10 | flow-stat -f 10 -S 3 | more***

```
#  --- ---- ---- Report Information --- --- ---
#
# Fields:    Total
# Symbols:   Disabled
# Sorting:   Descending Field 3
# Name:      Source/Destination IP
#
# Args:      flow-stat -f 10 -S 3
#
#
# src IPaddr     dst IPaddr       flows   octets     packets
#
192.168.140.101  192.168.173.33   143     52116041   48805
192.168.138.18   192.168.181.18   346     46998168   782403
192.168.200.96   192.168.175.53   97      41488401   78375
192.168.140.101  192.168.181.6    291     37109959   35153
192.168.200.133  192.168.182.53   283     31770197   40781
192.168.200.133  192.168.224.31   144     31743879   69797
192.168.200.93   192.168.204.58   85      31709423   32161
```

The above lists only show information collected for 1 day. By collecting this information and storing it in a database you will begin to create a baseline on the type and amount of traffic that is moving across the network. The analyst can easily find or build a set of tools to generate graphs or tables with which he/she could use to find anomalies in the traffic patterns. Because the data has been collected the analyst can start to ask questions and receive answers through a minimal analysis effort.

For example, looking at the top talkers list above, we could ask if the host at 192.168.140.101 was supposed to be sending or receiving 52Mb worth of data

from the host at 192.168.173.33. If this was pc to server traffic that might be ok, but if this was pc to pc traffic then we would go a little deeper in the analysis.

First we would have to determine what these 2 hosts were. Being internal systems it was relatively easy to determine that they were both personal computers. Using the flow statistics collected we would query for the source IP address to determine what ports were being used. To filter out a specific IP address we would first have to setup up a filter in a file named flow.acl. This file will be stored in the same subdirectory as the Flowtool suite. The filter is built very similar to an access list used in a Cisco router. The contents of the flow.acl file that would filter out all IP addresses except for 192.168.140.101 would contain the following command:

```
ip access-list standard badguy permit host 192.168.140.101
```

This filter creates a standard access list named badguy and would only allow host 192.168.140.101 data to be displayed. Next we would invoke a series of Flowtool utilities to display the information using the following command:

labhost:/var/flow# ***flow-cat labrtr/2005-01/2005-01-10/* | flow-filter –S badguy | flow-print***

First we concatenate the flow records using flow-cat and pipe them into the utility flow-filter. Using the –S switch with flow-filter, we would provide the name of the acl that was defined in the flow.acl file. The name of the access list in this case is badguy. The output of the flow-filter utility is next piped into flow-print and the following output was displayed.

srcIP	dstIP	prot	srcPort	dstPort	octets	packets	
192.168.140.101	192.168.173.33	6	20401	1214	4029224	2686	
192.168.140.101	192.168.173.33	6	20401	1214	3929562	2620	
192.168.140.101	192.168.173.33	6	20401	1214	3127379	2085	

Going to the IANA registry and looking up port 1214, we find that it is commonly used by Kazaa. Kazaa is a peer-to-peer program normally used in trading MP3 music files. We could now go to the end user and further investigate what was happening. Using similar queries we could look for other odd destination ports or high traffic heading towards other hosts. By having this information available and setting up some automated script we can easily create some useful tools for anomaly detection.

The Flowtools suite is purely a set of command line tools that allow you to collect and generate tabular reports based on the network activity. Even having automated scripts to generate daily reports it does not allow you to see long term trending, for that we need to send the captured data to some sort of database.

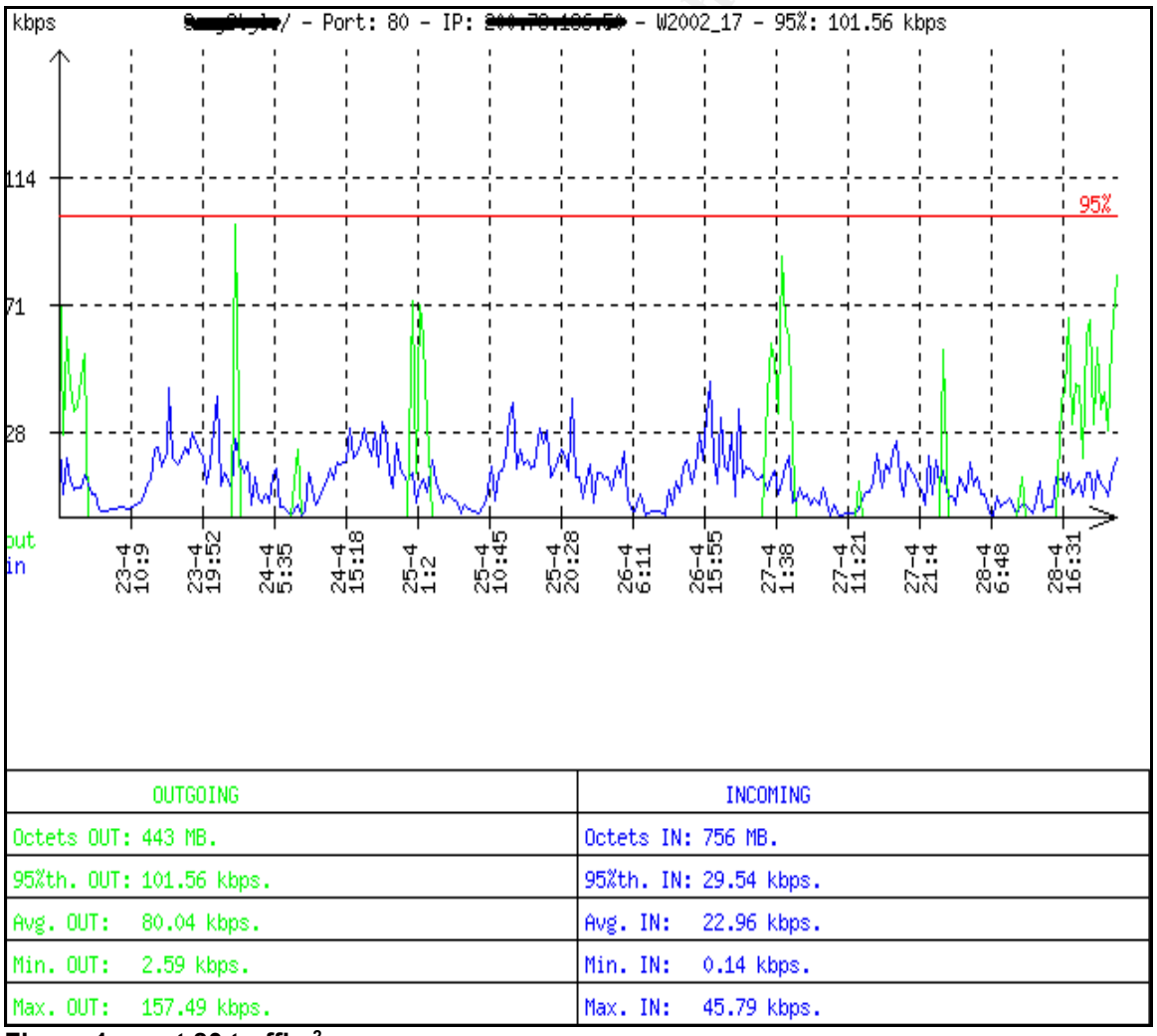
Two other open source applications that will capture Netflow® records and store

them in a MySQL data base are NEye and Flow Loader And Virtual Information Output (F.L.A.V.I.O.).

NEye is purely a capture tool that collects flow records and stores them into a MySQL database. You are required to setup the environment to retrieve the data and create the needed graphics. This requires more work but you have the flexibility to create any view of the data you need.

F.L.A.V.I.O. on the other hand can collect records, store the information in MySQL and generate graphs. F.L.A.V.I.O. can do its own data collection or interface with records captured with Flowtools. Graphs can be generated for weekly, monthly and yearly time frame. Again using other open source tools and a bit of programming you can create any view of the data you need. Below are some basic graphs generated by the F.L.A.V.I.O. system.

Below Figure 1 shows port 80 traffic over a 1 week time-frame.

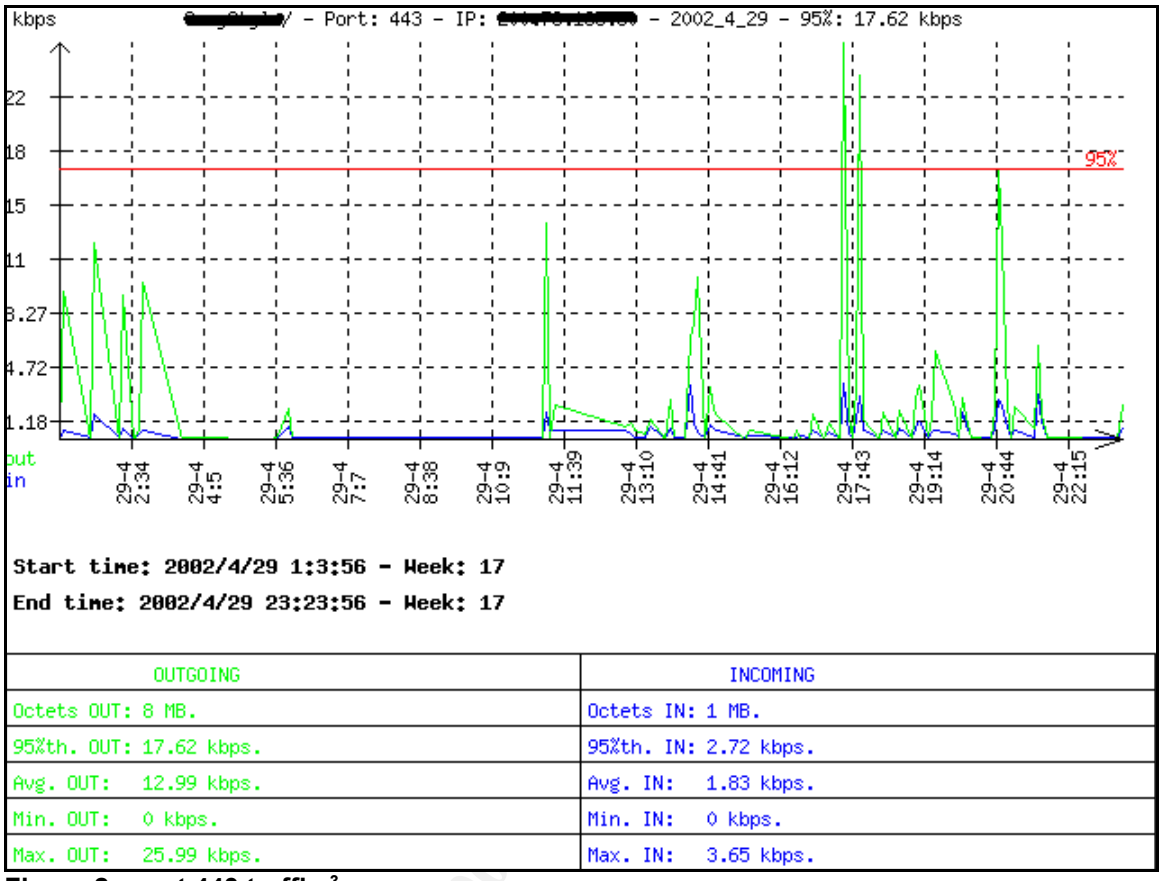


OUTGOING	INCOMING
Octets OUT: 443 MB.	Octets IN: 756 MB.
95%th. OUT: 101.56 kbps.	95%th. IN: 29.54 kbps.
Avg. OUT: 80.04 kbps.	Avg. IN: 22.96 kbps.
Min. OUT: 2.59 kbps.	Min. IN: 0.14 kbps.
Max. OUT: 157.49 kbps.	Max. IN: 45.79 kbps.

Figure 1 - port 80 traffic [2]

2. Villanustre, Flavio. flavio.sourceforge.net. URL: http://flavio.sourceforge.net/images/charts/80.png (Feb 2005)

Below Figure 2 shows port 443 traffic for a 24 hour time-frame.



OUTGOING	INCOMING
Octets OUT: 8 MB.	Octets IN: 1 MB.
95%th. OUT: 17.62 kbps.	95%th. IN: 2.72 kbps.
Avg. OUT: 12.99 kbps.	Avg. IN: 1.83 kbps.
Min. OUT: 0 kbps.	Min. IN: 0 kbps.
Max. OUT: 25.99 kbps.	Max. IN: 3.65 kbps.

Figure 2 - port 443 traffic [3]

When continually watched, trending will be readily apparent and anomalies will become visible.

Ntop

Ntop is a network statistical and monitoring application that passively collects traffic statistics and presents the information using a WWW front-end. Ntop provides a number of different statistics that includes traffic measurement for each host. Below are the types of host statistics that are collected:

- DATA SENT /RECEIVED - The total traffic (volume and packets) generated or received by the host. Classified according to network protocol (IP,IPX, AppleTalk, etc.) and IP protocol (FTP, HTTP, NFS,etc.)
- USED BANDWIDTH - Actual, average and peak bandwidth usage.
- IP MULTICAST - Total amount of multicast traffic generated or received by the host.
- TCP SESSIONS HISTORY - Currently active TCP sessions

3. Villanustre, Flavio. flavio.sourceforge.net.. URL: http://flavio.sourceforge.net/images/charts/443.png

established/accepted by the host and associated traffic statistics.

- UDP TRAFFIC - Total amount of UDP traffic sorted by port.
- TCP/UDP USED SERVICES - List of IP-based services (e.g. open and active ports) provided by the host with the list of the last five hosts that used them.
- TRAFFIC DISTRIBUTION - Local traffic, local to remote traffic, remote to local traffic (local hosts are attached to the broadcast network).
- IP TRAFFIC DISTRIBUTION - UDP vs. TCP traffic, relative distribution of the IP protocols according to the host name.

Ntop provides multiple graphs over a wide variety of time periods including yearly, monthly, daily, last 12 hours, last 6 hours and last hour. In addition Ntop can also collect NetFlow® and SFlow® records. Ntop is configurable and has some customizable features. Ntop provides access to a large variety of information that is very useful to the security analyst. Because of the graphical capabilities of Ntop the analyst can quickly spot trends or anomalies with relative ease. Below are just 2 screen shots that show some of the network information collected from a test system.

Below Figure 3 shows bandwidth usage by TCP/UDP protocol.

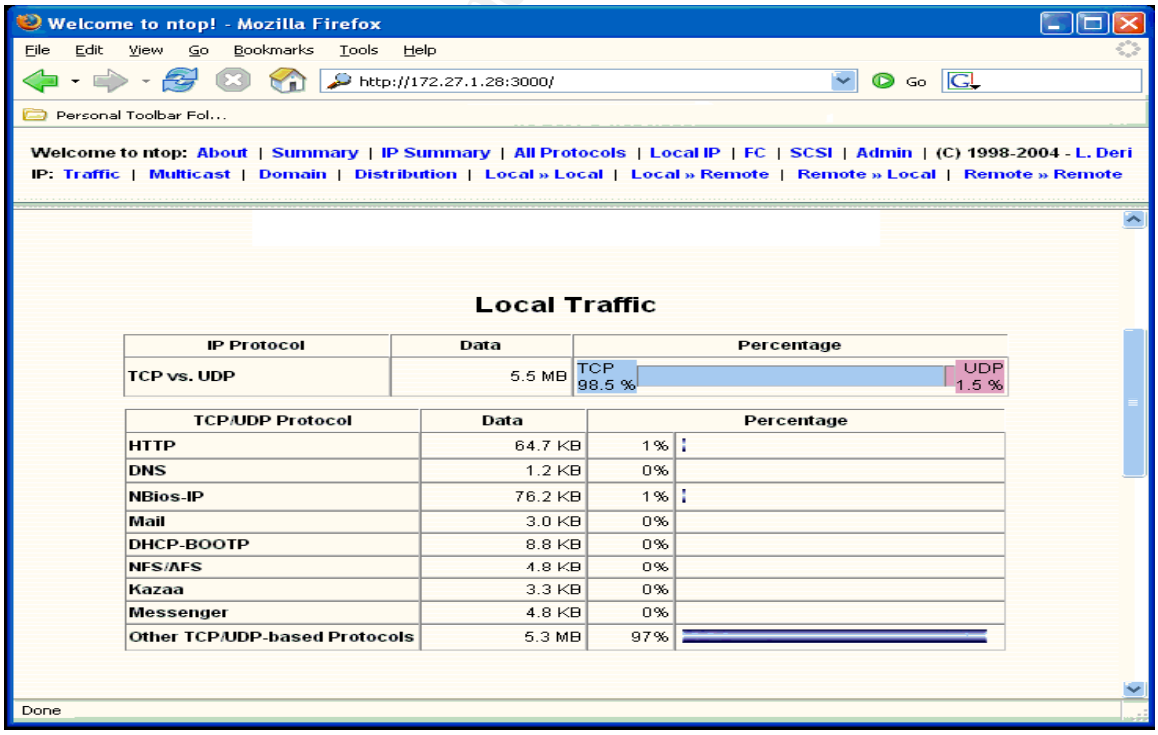
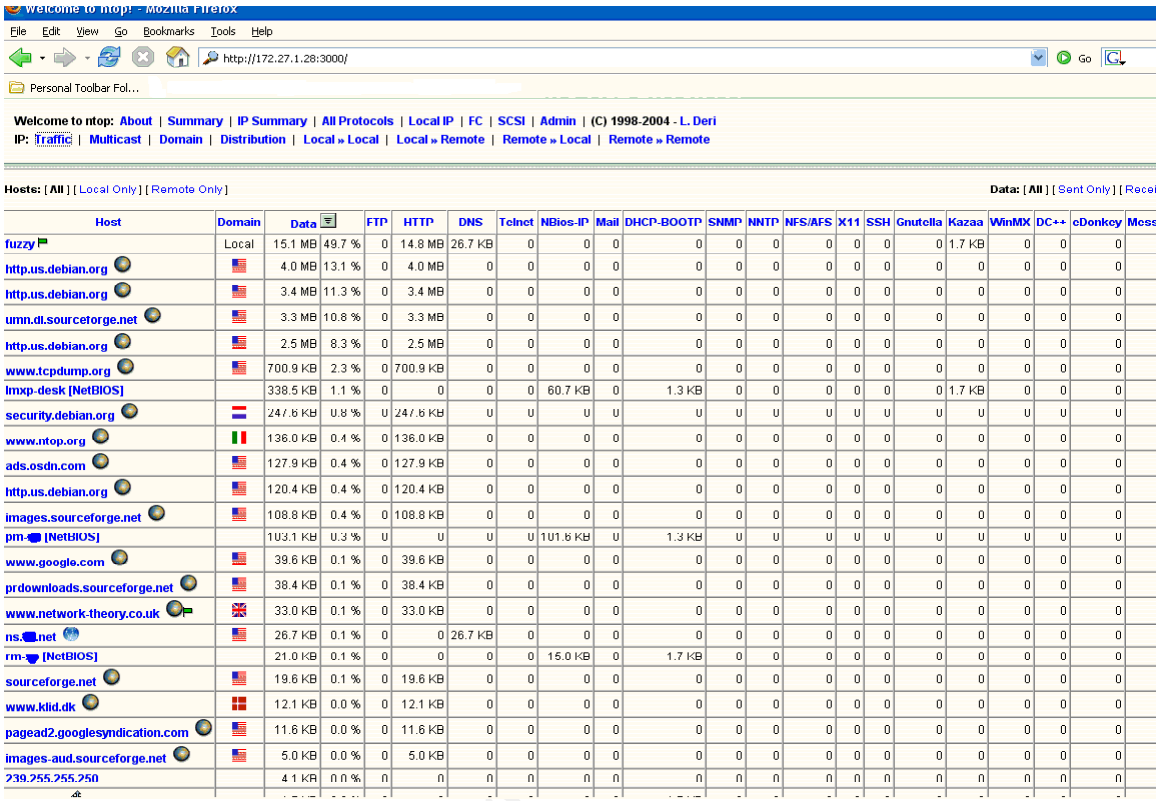


Figure 3 - Bandwidth by Protocol

Below Figure 4 shows traffic for protocols listed by all hosts that have been observed since monitoring began.



Welcome to ntop! - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://172.27.1.28:3000/

Personal Toolbar Fol...

Welcome to ntop: About | Summary | IP Summary | All Protocols | Local IP | FC | SCSI | Admin | (C) 1998-2004 - L. Deri

IP: Traffic | Multicast | Domain | Distribution | Local » Local | Local » Remote | Remote » Local | Remote » Remote

Hosts: [All] [Local Only] [Remote Only] Data: [All] [Sent Only] [Recei

Host	Domain	Data	FTP	HTTP	DNS	Telnet	NBios-IP	Mail	DHCP-BOOTP	SNMP	NNTP	NFS/AFS	X11	SSH	Gnutella	Kazaa	WinMX	DC++	eDonkey	Mess
fuzzy	Local	15.1 MB	49.7 %	0	14.8 MB	26.7 KB	0	0	0	0	0	0	0	0	0	1.7 KB	0	0	0	
http.us.debian.org		4.0 MB	13.1 %	0	4.0 MB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
http.us.debian.org		3.4 MB	11.3 %	0	3.4 MB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
umn.dl.sourceforge.net		3.3 MB	10.8 %	0	3.3 MB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
http.us.debian.org		2.5 MB	8.3 %	0	2.5 MB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
www.tcpdump.org		700.9 KB	2.3 %	0	700.9 KB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
lmxp-desk [NetBIOS]		338.5 KB	1.1 %	0	0	0	0	60.7 KB	0	1.3 KB	0	0	0	0	0	0	1.7 KB	0	0	0
security.debian.org		247.6 KB	0.8 %	0	247.6 KB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
www.ntop.org		136.0 KB	0.4 %	0	136.0 KB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ads.osdn.com		127.9 KB	0.4 %	0	127.9 KB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
http.us.debian.org		120.4 KB	0.4 %	0	120.4 KB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
images.sourceforge.net		108.8 KB	0.4 %	0	108.8 KB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
pm- [NetBIOS]		103.1 KB	0.3 %	0	0	0	0	101.8 KB	0	1.3 KB	0	0	0	0	0	0	0	0	0	0
www.google.com		39.6 KB	0.1 %	0	39.6 KB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
prdownloads.sourceforge.net		38.4 KB	0.1 %	0	38.4 KB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
www.network-theory.co.uk		33.0 KB	0.1 %	0	33.0 KB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ns. .net		26.7 KB	0.1 %	0	0	26.7 KB	0	0	0	0	0	0	0	0	0	0	0	0	0	0
rm- [NetBIOS]		21.0 KB	0.1 %	0	0	0	0	15.0 KB	0	1.7 KB	0	0	0	0	0	0	0	0	0	0
sourceforge.net		19.6 KB	0.1 %	0	19.6 KB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
www.klid.dk		12.1 KB	0.0 %	0	12.1 KB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
pagead2.googlesyndication.com		11.6 KB	0.0 %	0	11.6 KB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
images-aud.sourceforge.net		5.0 KB	0.0 %	0	5.0 KB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
239.255.255.250		4.1 KB	0.0 %	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4 - Protocol Bandwidth by Host

Snort with SPADE

SPADE is short for Statistical Packet Anomaly Detection Engine. It's a Snort preprocessor that uses statistical methods to create a baseline of what types and flow rates of traffic exist on your network. When SPADE finds a packet of interest it will compare it to the established baseline. If the packet is different from the baseline SPADE will send an alert along with an anomaly score through to Snort. The anomaly score that is assigned is based on the observed history of the network. The fewer times that a particular kind of packet has occurred in the past, the higher its anomaly score will be.

For example, if your network has never had IRC traffic and a user gets infected with a Trojan that uses an IRC backdoor SPADE would alert you to the unusual traffic, regardless of the port it's on.

As with any implementation of Snort, other package must be used in order to display alerts received and generate reports. There are number of different packages which include ACID, SNORTSNARF and BASE. This can be a fairly complicated installation and is outside of the scope of this paper. These tools are mentioned to provide and additional source of applications used in anomaly detections.

Commercial products

Over the last few years commercial network behavioral analysis tools have begun to appear. These products can be expensive to implement and require network and security expertise in the installation. Once these systems have been installed they are fairly easy to use and administer. Three products which play in the area of network and security analysis are Lancope's Stealthwatch, Netscout's nGenius Performance Management System and SourceFire's Realtime Network Analyzer.

Lancope's Stealthwatch

Stealthwatch is a passive monitoring appliance that can be placed at different points in the network. Similar to any type of monitoring device it is best to deploy Stealthwatch at network chokepoints. Once installed Stealthwatch begins profiling hosts and network traffic and over the first 30 days develops a baseline. During the building of the baseline, Stealthwatch collects information for each observed host including protocol usage, packet rate, bandwidth consumption and traffic history statistics.

In addition Stealthwatch creates flow-based statistical analysis and applies it to traffic on a host by host basis. Once host traffic flows have been properly categorized and thresholds have been set, any anomalous traffic can be identified and reported. Stealthwatch assigns a numerical rating called a Concern Index. The Concern Index will accumulate 'points' for any given host as traffic patterns emerge. An alarm will be sent to the console when the Concern Index passes a threshold there by notifying the analyst of a host's activity. At this point further investigation can be initiated or if the pattern is valid the Concern Index threshold can be adjusted. Policies can be developed for each host, network segment or zone (a group of network segments).

Each Stealthwatch appliance has a web-based front-end that allows the analyst to quickly view a dashboard of statistics, alarms and it provides access to the administration tools. In addition there is a separate management console that will consolidate information from multiple appliances.

One of the advantages of Steathwatch is that it automatically develops baseline patterns and allows the analyst to adjust them as needed. Steathwatch then provides console alarms based upon traffic patterns observed for any host. One of the disadvantages is that it requires tuning the alarm thresholds.

nGenius Performance Management System

nGenius Performance Management System started out as an RMON based product used to report on network performance traffic. There had to be a number of RMON probes placed with the network infrastructure so visibility into the packets could be seen. Netscout has greatly improved the capabilities of the product by allowing other types of traffic data to be pulled from various

devices. nGenius extracts information out of different data sources including: SNMP devices, flow records, network probes, and NetScout's nGenius Probes and Active Agents.

Data is collected and sent to a central console where it is sliced and diced into different views. Information is presented via a web interface and can be updated on a scheduled or adhoc basis. The *nGenius* Performance Management System's NetFlow® tools can collect, analyze, and report on NetFlow®. All of the RMON and NetFlow® statistics can be reported on. Data can be broken down and reported on by utilization, top talkers, protocol statistics, host performance statistics, etc. *nGenius* can provide a high-level summary as well as allowing the analyst to view any detail of a network packet. The analyst can access multiple views of applications, metrics, flows and data sources to quickly determine where or what a problem is. In addition this system provides workflow automation and alarming.

nGenius was developed for network performance monitoring in mind but, based on the type of data collected and reporting of traffic patterns it has quickly become a tool for network security monitoring.

SourceFire's Realtime Network Analyzer

Built on the open source Snort® rules-based detection engine, the Sourcefire system uses a combination of signature, protocol and anomaly-based inspection methods to provide a real time view on network traffic. The RNA (Real-time Network Awareness™) Sensors provide a view of security events, and host traffic patterns using a combination of passive network discovery and behavioral profiling technologies.

Sourcefire is a combination of appliances and software that can be deployed as a traditional IDS system. The added features of the RNA Sensor provide a real-time view of monitored network assets. This view presents baseline profiles and includes device configurations, behavioral information and potential vulnerabilities that may exist. RNA builds a 'host record' of every device it discovers on the network which includes a vulnerability database associated with its respective operating system. In addition profiles include traffic flow, traffic type and traffic volumes.

The Sourcefire system integrates and correlates threat information provided by intrusion sensors and combines this data with the network baseline profiles to present a picture of the over all network security. The central console has the ability to prioritize security events and to determine the most critical events that need to be reviewed. Network policies can be established for networks and hosts. This system also has the ability to interface with firewalls and change rules to block anomalous traffic based on observed patterns.

Sourcefire can assign an 'Impact' value to an event by comparing attacks

targeting a host against its particular baseline database record. This impact value is used to establish priority for threat response. An alarm is generated and sent to a centralized console for remediation. The console is web based and provides the ability to drill down from the alarm through to the specific host. The analyst has host profile and the current state of network communications available for review. Using the information available the analyst can determine if the alarm needs to be immediately acted upon or can be resolved at a later time.

Sourcefire takes the approach of combining traditional IDS sensors and couples that with real-time traffic monitoring. Baseline profiles are generated for each host and networks. Real-time network traffic is then compared against profiles to generate prioritized alarms that can be acted upon. Sourcefire uses a combination of anomaly and behavioral analysis to generate console alarms. An advantage of Sourcefire is that a properly installed system monitors, adjusts and maintains host profiles automatically. This reduces the amount of work that is required to implement anomaly detections schemes.

Open Source vs. Commercial

Whether a system is built using open-source tools or purchased commercial products, it is important to understand the benefit that network analysis brings to the security analyst. Each type of system has advantages and disadvantages. With open-source tools you have the flexibility to develop what you feel is most important but, the disadvantage is that you have to build the tools. With commercial products much of the work is already done but the cost can be significant and you may want to watch something that is not supported.

An advantage to either is that systems can be implemented in a small way and added upon. There can be a mix of open-source and commercial products that will create the base tools needed for behavioral and anomaly analysis. Being aware of what should and should not be present on the network, the security analyst has a much better chance of providing benefit to the organization verses being viewed as a necessary expense.

Conclusion

In an article on SearchSecurity, Michael J. Martin stated, “If you’re unable to discern between what should and should not be running on a network, you cannot secure it.”[4]

The above statement really describes the role of security analyst, identifying threats and securing the network against them. Various tools are needed to truly

4. Martin, Michael J. “Router Expert: Why you need a network services audit” Nov 2004. URL: http://searchsecurity.techtarget.com/tip/1,289483,sid14_gci1026349,00.html?FromTaxonomy=%2Fpr%2F292187

get a picture of what is really happening on the network. Multiple layers of devices providing information and having that data presented in a manageable format is necessary for the security analyst to fulfill this mandate. But limiting one-self to traditional IDS tools and not understanding what should be on the network is really looking at half the picture.

The role of the attacker has become less specialized and exploits are easier to use. Script kiddies can assemble and run attacks from pre-made sources without giving thought to how the attack really works. In addition, the proliferation of networked home computers and general lack of awareness by the owners has created an environment where they have become unsuspecting attackers. The security analyst must deploy systems and tools that help them understand what is supposed to be allowed and what is not.

When combined with traditional IDS systems, behavioral network analysis and anomaly detection systems provide a very powerful intrusion detection mechanism. This defense-in-depth approach gives the security and network analysts a set of tools to quickly determine and isolate an attack. When all of the information for anomaly detection is available and with the right tools put in place, this type of system can be used to quickly identify areas of concern and can move the analysts from a reactive state to a proactive response.

References

The following materials were consulted in the preparation of this document.

InMon Corp. “Traffic Monitoring in a Switched Environment” 2001. URL: http://www.inmon.com/pdf/EmbeddedTM.pdf (Jan 2005)

InMon Corp. “Using sFlow and InMon Traffic Server for Intrusion Detection and other Security Applications” 2001. URL: http://www.inmon.com/pdf/sFlowSecurity.pdf (Jan 2005)

Cisco System Inc. “nGenius Real Time Monitor 1.4 Q&A” 1992-2004. URL: http://www.cisco.com/en/US/products/sw/cscowork/ps2803/products_qanda_item09186a0080088839.shtml (Jan 2005)

Reves, Joseph and Panchen, Sonia. “Traffic Monitoring with Packet-Based Sampling of Defense against Security Threats” 2002. URL: http://www.sflow.org/SamplingforSecurity.pdf (Jan 2005)

sFlow.org. “Using sFlow” 2003-2004. URL: http://www.sflow.org/using_sflow/index.php (Jan 2005)

Reilly, Rob. “Network Intrusion Detection, Neighborhood Watch Style” Oct 2004. URL: http://www.linuxplanet.com/linuxplanet/reports/5605/1/ (Jan 2005)

Schwartz, Mathew. “Case Study: Outsourced Network Security Uses Behavioral Modeling” Dec 2004. URL: http://www.esj.com/Security/article.aspx?EditorialsID=1216 (Jan 2005)

Schwartz, Mathew. “Case Study: Securing Network Bandwidth” Dec 2004. URL: http://www.esj.com/Security/article.aspx?EditorialsID=1207 (Jan 2005)

Martin, Michael J. “Router Expert: Why you need a network services audit” Nov 2004. URL: http://searchsecurity.techtarget.com/tip/1,289483,sid14_gci1026349,00.html?FromTaxonomy=%2Fpr%2F292187 (Jan 2005)

Fullmer, Mark. www.slpintered.net. URL: http://www.splintered.net/sw/flow-tools/docs/flow-tools.html (Feb 2005)

Persico, Gilberto. URL http://neye.unsupported.info/ (Feb 2005)

Villanustre, Flavio. flavio.sourceforge.net. URL http://flavio.sourceforge.net/images/charts/ (Feb 2005)

Deir, Luca. www.ntop.org. 1998-2004. URL: http://www.ntop.org/ntop-

overview.pdf (Feb 2005)

Bejtlich, Richard. The Tao of Network Security Monitoring, Beyond Intrusion Detection. Boston: Addison Wesley Professional, 2004.

Additional Information the Open Source and commercial tools presented can be found at:

Galloway, Robert. “How to build detailed network usage reports using RRDTool, Flowtools, FlowScan and CUFlow” 2003. URL: http://www.linuxgeek.org/netflow-howto.php (Feb 2005)
Blundell, Rick. “NetFlow Guide” URL: http://www.netflowguide.com (Feb 2005)

www.splintered.net. “flow-tools information” URL: http://www.splintered.net/sw/flow-tools (Feb 2005)

Caswell, Brian and Roesch, Marty. “Official Documentation” 2004 URL: http://www.snort.org/docs/ (Feb 2005)

www.netscout.com. nGenius Product Information. URL: http://www.netscout.com/products/ (Feb 2005)

www.lancope.com. Stealthwatch Product Information. URL: http://www.lancope.com/products/ (Feb 2005)

www.sourcefire.com. RNA Product Information. URL: http://www.sourcefire.com/products/rna.html (Feb 2005)