Global Information Assurance Certification Paper

Netflow Analysis for the Security Professional

GSEC Practical Assignment V.1.4b, Option 1

Prepared by: Matthew Olney
August 5, 2004

NetFlow Analysis for the Security Professional

Abstract 3

NetFlow Basics 3

Security Impact of NetFlow Information 5

The flow-tools NetFlow Analysis Suite 6

Installation 7
NetFlow Data Capture: flow-capture 8
Router and Multi-Layer Switch Configuration 9
NetFlow Data Analysis: flow-print, flow-report & flow-stat 11

NetFlow Analysis Design Examples 15

Production Traffic Analysis 15
Sinkhole/Tarpit Traffic Analysis 17

Conclusion 19

References 20

Abstract

This paper looks at the security benefits of implementing a NetFlow analysis system and at one of several open source applications useful in NetFlow analysis. NetFlow statistics are generated by Cisco and Juniper routers and switches, as well as server software NetFlow probes. The Cisco implementation of NetFlow is used in the configuration examples. NetFlow data provides important information about network conversations and behavior. Each unique flow is recorded by the network devices or probes, and the flows are then reported to a data collection server. NetFlow data provides enough information to serve the needs of several different applications such as billing, network planning and traffic engineering.

The flow information, while lacking payload data, still provides enough data to the security professional to be a valuable analysis tool. The data is compact enough to be stored for multiple months, providing long term forensic capability, and is detailed enough to provide a real-time analysis of traffic flows, connection information and abnormal network behavior. This data can be used both for intrusion detection and for incident handling purposes.

NetFlow Basics

Darren Kerr and Barry Bruins developed the NetFlow technology while working for Cisco Systems in 1996. While initially developed as a switch path for Cisco routing gear, it soon became apparent that the information in the NetFlow cache carried additional value (NetFlow Overview 1). Routers and layer 3 aware switches with NetFlow capability keep track of TCP and UDP conversations and periodically report to a central logging server statistics for those conversations.

NetFlow data is kept and reported in terms of flows. A flow is unidirectional; that is, a flow is from a single source IP to a single destination IP. Therefore a typical TCP conversation will consist of two flows, one from the client to the server, and one from the server to the client. The Cisco Document “NetFlow Services Solutions Guide” states that flows are uniquely identified by seven key fields:

- “Source IP address
- Destination IP address
- Source port number
- Destination port number
- Layer 3 protocol type
- ToS byte
- Input logical interface (ifIndex)”

NetFlow statistics are exported once the flow expires. A flow expires any time a TCP close command is seen (RST or FIN), when the flow cache becomes full, when a flow has been inactive for 15 seconds or when a flow has been active for 30 minutes. Both of these timer values are the default for Cisco gear and can be modified. Note that because a flow is exported after 30 minutes, or after the flow cache becomes full, what would otherwise be reported as single flows may be reported multiple times as multiple flows. For example, if a flow lasts 90 minutes, that single flow would be reported 3 times, using the default Cisco active expiration timer.

NetFlow data is exported via UDP packets to the NetFlow monitoring server. Because of the potentially high volume of traffic, and because UDP does not guarantee delivery of the packets, Cisco recommends a dedicated connection be made available to the NetFlow monitoring server. (NetFlow Services Solutions Guide, p 12)

Data is exported in one of five versions: 1, 5, 7, 8 or 9. Version 1 provides a broad range of information regarding flows: Source and destination IPs and ports, start and end times of the flow, number of packets and octets in the flow, the IP protocol and ToS of the flow, information regarding the TCP flags (if any) and information about TCP sequencing. Version 5 adds information regarding the BGP autonomous system (AS) number, and NetFlow sequence numbers to indicate when UDP packets have been lost between the networking equipment and the NetFlow monitoring server. Version 7 is supported only on Cisco's 6500 and 7600 series switches and lacks BGP AS information, as well as information on TCP flags and ToS. Version 8 was developed to allow network administrators to export subsets of information to the monitoring server, reducing the load both on the networking equipment and the NetFlow monitoring server. Version 9 is Cisco's IETF submission that allows for an extensible format so that information can be easily added to the NetFlow export.

NetFlow data also addresses some privacy concerns that some installations may be subject to. While the data is very granular at the layer 3 data, it has no view into the actual payload data. Therefore it is possible to get a great deal of useful traffic data, while avoiding the possible issue of capturing private or sensitive data. As Daniel Medina put it in his paper, "An IDS Using NetFlow Data", "NetFlow is adequately blind to content, but still able to reveal what we need to know." (Medina 1)

While NetFlow data is generally created by Cisco routers, there are applications that monitor network traffic and then create NetFlow exports based on captured packets. These applications avoid the overhead of adding NetFlow tracking and exporting to the router by moving that functionality to a server that has a view of all the network traffic that needs to be processed into flows. Applications that generate NetFlow exports based on network traffic include Fprobe (http://fprobe.sourceforge.net/), Softflowd (http://www.mindrot.org/softflowd.html) and ntop (http://www.ntop.org/netflow.html). (Bejtlich 220-224)

Security Impact of NetFlow Information

There are several potential uses for NetFlow information in the security profession. Here is a list of all the data contained in a Version 5 record format:

- Source and Destination IP Address
- IP Address of the next hop routing device
- ifIndex of the input and output interfaces
- Total number of packets in the flow
- Total number of Layer 4 bytes in the flow
- SysUptime at the beginning of the flow, and the end of the flow
- Source and destination TCP/UDP port or equivalent
- Cumulative OR of TCP Flags
- IP Protocol Number
- IP ToS
- Source and Destination AS number
- Source and Destination prefix mask bits

This data set lends itself to several applications, including incident response, forensics, intrusion detection and even security research through the use of dark nets or sinkhole routers.

With a well developed system, NetFlow information can be used as a tool during ongoing incidents and forensics. Because the information shows activity before, during and after the incident, it can help to trace activity and define time boundaries of the incident. Network activity from different time periods can easily be compared, and abnormal or unexpected traffic can be identified. This is an excellent way to try and isolate problems. Also, network behavior of a compromised host can be reviewed, and any unusual connections can be researched. This can lead either to tracing further compromises within the organization, or help in tracking down tools, techniques or hosts used by intruders.

Note that if NetFlow information is to be used in the course of an investigation, it should be protected like any other electronic record that will be used as evidence. NetFlow records should be created by the same process each time, and have a standardized retrieval system. Each evidentiary piece of NetFlow data should be authenticated and stored securely. Protections should be in place to prevent unauthorized modifications of data. This can be done by allowing only authorized personnel to access the information, ensuring that those people with access are properly trained in the handling of that data, providing for backup and recovery of the information and ensuring that the systems are configured to resist failure (i.e. RAID and UPS systems). Also, all removable media that is part of the system should be tracked. (Schweitzer 184)

The same system can be used to create a crude intrusion detection system, looking for either incomplete SYN-SYN ACK-ACK sequences or simply unexpected connections. One tool that automates the process of scan detection with NetFlow data is “flow-dscan” from the Flow-Tools suite. This tool uses the NetFlow information to detect network scans and certain crude denial-of-service attacks. This can be combined with simple PERL scripts to provide for automatic notification of suspicious activities. Most IDS systems approach IDS in one of two ways, either misuse detection or anomaly detection (Lee & Stolfo 8). NetFlow data typically uses the anomaly detection method. For example, a system could look at abnormal TCP flag configurations such as having all flags set, or no flags set. This is a departure from the TCP protocol. A system could also look at a departure from normal operating parameters, such as a sudden increase in the number of flows from a particular IP address or block, or for an abnormally high number of packets or octets from an IP address.

An example of this would be a modified approach to Lee & Stolfo’s “Data Mining Approaches for Intrusion Detection”. In this work, Lee & Stolfo used tcpdump data to generate data records and then used statistical analysis and an inductive learning engine to build an IDS system. By looking for out-of-the-norm connections or incorrectly terminated connections, the system would decide whether the traffic was normal or suspicious. The data records were composed of:

- “Start time and duration
- Participating hosts and ports (applications)
- Statistics
- Flag: “normal” or a connection/termination error
- Protocol TCP or UDP” (Lee & Stolfo 37)

Each of these records could be generated with NetFlow data. Properly configured, this would be a more reliable collection methodology, because of the potential of data loss through tcpdump from over-subscribed spanning ports or insufficient system resources. The NetFlow data, on the other hand, is much more reliably transmitted and stored.

The flow-tools NetFlow Analysis Suite

One of the easiest tools to implement to monitor and interpret NetFlow information is the flow-tools software package, which is available at http://www.splintered.net/sw/flow-tools/. This suite of software provides an easy to setup data capture package (flow-capture) and a set of preconfigured reporting templates (flow-stat). Other tools in the package provide for data manipulation, interpretation and even a network scanning detection application (dscan).

Installation

Once the latest version of flow-tools has been obtained, the software needs to be unpacked and untared.

```
bash-2.05b# gunzip flow-tools-0.66.tar.gz
bash-2.05b# tar -xvf flow-tools-0.66.tar
flow-tools-0.66/
flow-tools-0.66/configs/
flow-tools-0.66/configs/Makefile
                .
                .
                .
bash-2.05b# cd flow-tools-0.66
```

The software then needs to be compiled onto the system.

```
bash-2.05b# ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
                .
                .
```

[NOTE: If a "configure: error: Link with "-lz" (zlib >= 1.0.2) failed!" error is received here, the zlib library needs to be installed. The zlib library is available at http://www.gzip.org/zlib/]

```
                .
                .

Now type make to continue the build process

bash-2.05b# make
Making all in lib
make[1]: Entering directory `/home/molney/flow-tools-0.66/lib'
make  all-am
                .
                .
bash-2.05b# make install
Making install in lib
make[1]: Entering directory `/home/flow/flow-tools-0.66/lib'
make[2]: Entering directory `/home/flow/flow-tools-0.66/lib'
                .
                .
bash-2.05b#
```

The software is now installed, and in a default installation it is located in '/usr/local/NetFlow'. The binary executables are located in the bin subdirectory, and the man pages are located in the man subdirectory.

NetFlow Data Capture: flow-capture

The flow-capture process is responsible for making available a network connection so that systems can send NetFlow data to the monitoring server. The flow-capture process then compresses and stores the data in the designated directory. The flow-capture process also has some data management switches that can control disk utilization and file creation.

The flow-capture process is started as follows (assuming you have created the /var/log/flows directory):

```
bash-2.05b# /usr/NetFlow/bin/flow-capture –w /var/log/flows 0/0/9500
```

The –w switch instructs the flow-capture process to write the data files to /var/log/flows. The final argument is the local IP address you want the process bound to (0 means listen on all interfaces), the remote IP address you wish to receive NetFlow information from (0 means accept NetFlow data from all IPs) and finally the UDP port you wish to listen on, in this case 9500.

-n The flow-capture application, by default, will write the data in 15 minute sets. If this data set is too large, or if the information needs to be accessed in a tighter span of time, you can modify the –n switch. This switch, which is set to 95 by default, determines the number of times per day a new data file is created. For example, if you set it to 23, then you would get a rotation once every hour. If you wanted to rotate every 5 minutes, you would set this value to 287. Note that the value is for the number of rotations, so the initial file does not count towards this total.

-e By default, each day you will generate 96 new data files containing flow information. Eventually this will reach the system limit for the number of files handled by certain system commands. For example:

```
[test@test flows]$ rm flow*
-bash: /bin/rm: Argument list too long
```

The –e command can limit the total number of files generated so that they are less than the number specified by this switch.

-E The total file size of all data files created by the flow-capture process can also be limited. The –E switch instructs the flow-capture process to ensure that the total size of the files is below the argument provided. The argument can be presented in bytes, kilobytes, megabytes or gigabytes by using the first letter of each size increment, i.e. 24K, 16b, 8M or 55G. Note that the bytes argument is a lower case b.

-N The –N switch tells the flow-capture how to build the directory structure. From the flow-capture man-page, here are the switch arguments:

-3 YYYY/YYYY-MM/YYYY-MM-DD/flow-file
-2 YYYY-MM/YYYY-MM-DD/flow-file
-1 YYYY-MM-DD/flow-file
0 flow-file
1 YYYY/flow-file
2 YYYY/YYYY-MM/flow-file
3 YYYY/YYYY-MM/YYYY-MM-DD/flow-file

-z The –z switch configures the level of compression of the data files. If you have the CPU cycles to spare, setting this to a higher number (the values are 0-9) will decrease your utilization of disk space, as the files will be smaller.

Additional switches are described in the flow-capture man file. Be sure to consult the man file (http://www.splintered.net/sw/flow-tools/docs/flow-capture.html) to ensure your setup is the best possible for your environment.

The flow-capture process will save the data files in the following naming convention:

ft-<NetFlow version>.<year>-<month>-<day>.<hour><minute><second>-<gmtoffset>

For example, the filename ft-v07.2004-07-21.023000-0400 is a flow-tools file with NetFlow version 7 data, starting at 2:30am on July 21, 2004, on a computer that is four hours behind the Greenwich Mean. The current working file for capturing flows in progress is prefixed with a “tmp” rather than the completed flow file’s “ft”. For example: tmp-v07.2004-07-30.103424-0400.

Now that the flow-capture process is running, the routers and multi-layer switches need to be configured to export their NetFlow data to the capture servers.

Now that we have the flow-capture process running, we need to configure the routers and multi-layer switches to forward their NetFlow data to the capture servers.

Router and Multi-Layer Switch Configuration

Examples in this paper were generated using a Cisco 6500 series switch with an integrated MSFC2 routing card on the supervisor module. The basic configuration steps for setting up NetFlow data export (NDE) on multi-layer switches such as the 6500 series, are as follows:

Ensure that the full flow data is being exported from the switch:

TEST> (enable) set mls flow

Set the destination for the NetFlow data (format is set mls nde <IP ADDRESS> <UDP PORT>:

TEST> (enable) set mls nde 10.1.1.1 9800

Finally, enable the export of the NetFlow data files:

TEST> (enable) set mls nde enable

At this point you should be able to see the packets from the testing system:

```
[test@test flows]# tcpdump udp port 9800
tcpdump: listening on eth0
13:31:52.679369 TEST.2376 > NetFlow.9800: udp 1428
13:31:52.682938 TEST.2376 > NetFlow.9800: udp 1428
13:31:52.685991 TEST.2376 > NetFlow.9800: udp 1428
13:31:52.689045 TEST.2376 > NetFlow.9800: udp 1428

[test@test flows]# tcpdump -c1 udp port 9800 -X
tcpdump: listening on eth0
13:33:08.186338 TEST.1027 > NetFlow.9800: udp 1428
0x0000   4500 05b0 69a3 0000 1d11 f3f8 0acb 0045        E...i..........E
0x0010   0acb 24c7 0403 2648 059c 8b93 0007 001b        ..$...&H........
0x0020   b07a beeb 4109 35ea 0000 0000 11dc ee53        .z..A.5........S
0x0030   0000 0000 0acb 2257 0acb 2087 0acb 2087        ......"W........
0x0040   011d 011d 0000 0008 0000 0170 b076 cefb        ...........p.v..
0x0050   b076                                          .v
3 packets received by filter
0 packets dropped by kernel
```

An excellent reference for all things NetFlow in the Cisco world is the “NetFlow Services Solutions Guide”, available at http://www.cisco.com/univercd/cc/td/doc/cisintwk/intsolns/netflsol/nfwhite.pdf. This document provides not only an in-depth look at all of the versions of NetFlow, but also gives several examples of different Cisco setups. For a sample router configure, this is their Configuration Example 3: NetFlow Export Destination Using Version 1, on page 15:

```
“Router(config-if)#  ip route-cache flow
Router(config)#  ip flow-export destination 172.17.246.255 9996
Router(config)#  ip flow-export version 1
Router(config)#  ip flow-export source loopback 0”
```

As always, carefully test all changes to internetworking equipment before committing the change to production. The management and transmission of the flow data adds overhead to both the internetworking device and the network.

Now that the flow-capture application has been launched, and data is being exported to the server, the NetFlow data can be analyzed using some of the other flow-tools applications.

NetFlow Data Analysis: flow-print, flow-report & flow-stat

The flow-tools application set provides three programs to analyze NetFlow data. Flow-print will display the flow data sequentially from a data file. Flow-report allows the generation of custom, comma-delimited reports of flow data. Flow-stat provides a set of pre-configured reports that are easily manipulated by shell scripts but are not fully configurable.

The flow-print application is used by piping cat output of the flow-tools data file to the command. For example:

```
[test@test flows]# cat ft-v07.2004-07-30.101500-0400 | flow-print | head
srcIP            dstIP           router_sc      prot  srcPort    dstPort    octets    packets
10.1.1.135/0     10.1.2.87/0     10.3.0.18      6     110        1701       615       10
10.1.1.135/0     10.1.2.87/0     10.3.0.18      6     110        1700       615       10
10.1.3.201/0     10.30.1,3/0     10.3.0.18      17    5112       161        76        1
10.4.64.30/0     10.1.1.254/0    10.3.0.18      17    48144      53         204       3
10.1.3.201/0     10.30.1.3/0     10.3.0.18      17    45150      161        76        1
10.224.67.20/0   10.1.1.199/0    10.3.0.18      6     59039      8010       1473      8
10.1.3.201/0     10.30.164/0     10.3.0.18      17    45241      161        77        1
10.1.3.23/0      10.3.34.87/0    10.3.0.18      6     20384      1270       528       6
192.168.0.1/0    10.3.4.25/0     10.3.0.18      6     43         60233      1675      3
```

This information shows the first nine flows of the flow-print output. The default data shows the source IP, the destination IP, the router source (in this case a multi-layer switch), the IP protocol used (6 TCP, 17 UDP), the source and destination ports used by those protocols and the total bytes and packets sent via that flow. This particular file is 6.3Mb and, when run through flow-print, has 465,103 lines of data. Because this is a lot of data to sort through, flow-stat is used to move some of the important information to the top.

For example, if a router were experiencing unusually high utilization, flow-stat could be used to find the top talkers during the last 15 minute period.

```
[test@test flows]# cat ft-v07.2004-07-30.101500-0400 | flow-stat -f 10 -S2 | head -22
#  --- ---- ---- Report Information --- --- ---
#
# Fields:    Total
# Symbols:   Disabled
# Sorting:   Descending Field 2
# Name:      Source/Destination IP
#
# Args:      flow-stat -f 10 -S2
#
#
# src IPaddr    dst IPaddr      flows          octets          packets
#
10.2.36.254     10.2.32.50      10413          1252024         10413
10.2.32.50      10.2.36.254     10136          624726          10136
10.2.0.68       10.2.36.201     5167           421623          5310
10.2.0.69       10.2.36.201     4891           397892          5024
10.2.36.201     10.1.0.164      4437           358588          4673
```

10.1.0.164	10.2.36.201	4430	371360	4668
10.2.36.201	10.1.0.162	4115	330148	4306
10.1.0.162	10.2.36.201	4108	341259	4307
10.2.0.79	10.2.36.201	4092	324987	4166
10.2.0.77	10.2.36.201	4061	327405	4169

The flow-stat command has a number of preconfigured reports, and the report is selected using the –f switch. The columns are sorted by using the –S switch, with the argument being the number of the column to be sorted. Note that the first column is column "0". In the example above, report number 10, which is the "Source/Destination IP" report is being run, and column 2, the flows column, is being sorted. If it was suspected that traffic throughput, rather than then number of connections was the problem, the following report could be run:

```
[test@test flows]# cat ft-v07.2004-07-30.101500-0400 | flow-stat -f 10 -S3 | head -22
#  --- ---- ---- Report Information --- --- ---
#
# Fields:    Total
# Symbols:   Disabled
# Sorting:   Descending Field 3
# Name:      Source/Destination IP
#
# Args:      flow-stat -f 10 -S3
#
#
# src IPaddr     dst IPaddr      flows          octets          packets
#
10.3.10.208      10.0.203.123    3              1550328822      1045696
10.3.18.8        10.3.10.201     1              781291001       765538
10.3.8.218       10.3.18.8       1              621937218       613688
10.1.247.20      10.3.18.11      3              292713324       196178
10.3.8.31        10.3.32.51      328            155543953       315138
10.3.8.31        10.3.32.50      369            141932750       291874
10.3.8.31        10.3.32.55      380            124797946       267000
10.3.8.157       10.3.32.32      18             92867587        403183
10.5.8.29        10.0.203.123    1              63243574        380098
10.3.32.32       10.3.8.157      12             50847169        224010
```

The flow-stat man page lists the complete set of reports. Another interesting report is number 5, the UDP/TCP destination port report. This is an excellent way to look at what ports are most active in your environment. The report can either be run in its default configuration: "cat ft-v07.2004-07-30.101500-0400 | flow-stat -f 5 -S3", or by using "grep" to grab only the ports of interest. For example, the following command looks for port 80, 25, 110, 443 and also includes all lines with "#" so the report information is shown:

```
[test@test flows]# cat ft-v07.2004-07-30.101500-0400 | flow-stat -f 5 -S3 | grep "^80 \|^25 \|^110 \|^443
\|#"
#  --- ---- ---- Report Information --- --- ---
#
# Fields:    Total
# Symbols:   Disabled
```

```
# Sorting:   Descending Field 3
# Name:      UDP/TCP destination port
#
# Args:      flow-stat -f 5 -S3
#
#
# port      flows           octets          packets
#
80          16707           28551720        197544
25          5163            53764008        94833
110         3012            1395399         29998
443         1242            1427027         12163
```

This information layout is excellent for basic shell scripting, but the flow-tools suite can also provide comma delimited data for use in spreadsheet data analysis or for database inserts. (Kretchmar 100-102) First a basic report configuration file has to be created. The full layout of the report file can be seen in the flow-report man page. Here is a simple report configuration file, looking again at port utilization for the first 10 ports on the list:

```
[test@test flows]# cat report.conf
stat-report portreport
        type ip-port
        output
                records 10

stat-definition portstat
        report portreport
```

And here is the output:

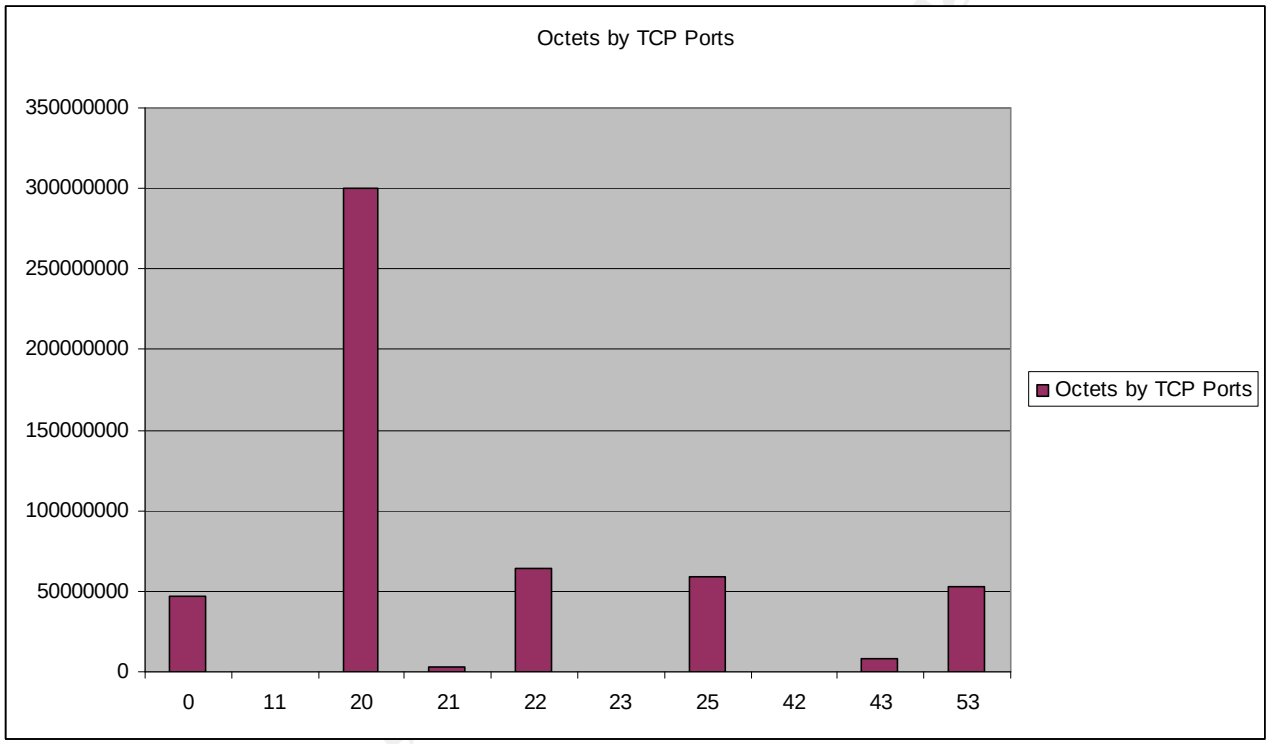
```
# recn: ip-port,flows,octets,packets,duration
0,2928,46660002,94312,1517422058
11,186,435224,1076,1383130
20,8,299832820,332792,106660
21,88,2889118,40626,2332797
22,1426,63886472,106367,32555957
23,13,38226,744,841922
25,9890,58946406,175872,139603967
42,1,1908,4,32
43,8713,7946017,41760,153687005
53,97353,52755666,427045,1500176086
# stop, hit record limit.
```

This data can easily be moved to excel, for example:

ip-port	flows	octets	packets	duration
0	2928	46660002	94312	1517422058
11	186	435224	1076	1383130
20	8	299832820	332792	106660
21	88	2889118	40626	2332797

22	1426	63886472	106367	32555957
23	13	38226	744	841922
25	9890	58946406	175872	139603967
42	1	1908	4	32
43	8713	7946017	41760	153687005
53	97353	52755666	427045	1500176086

And this data can be quickly turned into graphs for use in reports or analysis:



The flow-tools application suite provides a quick way to begin analyzing NetFlow data. In conjunction with a scripting language such as Perl or bash, or with a data analysis tool such as Excel, the basic toolset provides enough functionality for both reporting and incident investigation. The next section provides two implementation examples.

NetFlow Analysis Design Examples

Production Traffic Analysis

The standard installation of a NetFlow analysis system is most likely into a production environment. The goals of such an installation are to improve billing, network modeling and incident response assistance. A simple installation of this type might look like the network shown in Figure 1.

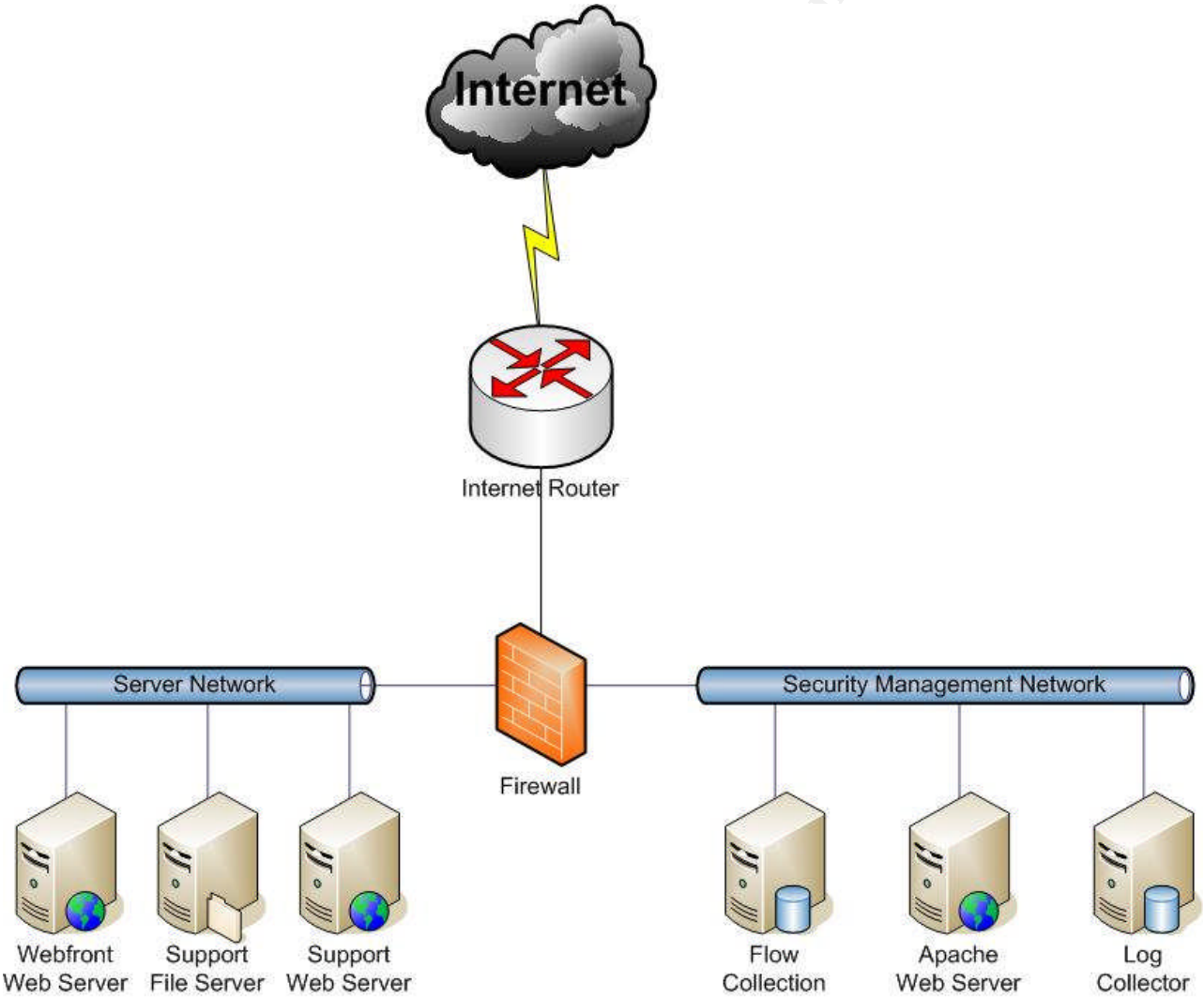


Figure 1: Production Analysis Design

In a high traffic web front, the easiest way to begin using NetFlow information in a security capacity is to look for aberrant network behavior. Some possible aberrant network behaviors are top talkers by flow, by octet and by packets, and these are easy to track.

For example, the following output is from a simple top-talker application that shows the top talker by flows every 15 minutes for the last two and a half hours. In a web commerce site, it is highly unlikely that an ongoing series of connections across two and a half hours is legitimate traffic. In this case, further investigation of 10.4.3.54 is necessary.

Filename	src IPaddr	dst IPaddr	flows	octets	packets
ft-v07.2004-08-03.171500-0400	10.4.3.54	10.1.1.111	607	1414839	6406
ft-v07.2004-08-03.170000-0400	10.4.3.54	10.1.1.111	628	1514414	6904
ft-v07.2004-08-03.164500-0400	10.4.3.54	10.1.1.111	560	1268853	5975
ft-v07.2004-08-03.163000-0400	10.4.3.54	10.1.1.111	561	1224313	5899
ft-v07.2004-08-03.161500-0400	10.4.3.54	10.1.1.111	640	1582832	7057
ft-v07.2004-08-03.160000-0400	10.4.3.54	10.1.1.111	561	1301403	6079
ft-v07.2004-08-03.154500-0400	10.4.3.54	10.1.1.111	613	1467630	6731
ft-v07.2004-08-03.153000-0400	10.4.3.54	10.1.1.111	577	1414580	6420
ft-v07.2004-08-03.151500-0400	10.4.3.54	10.1.1.111	529	1244350	5739
ft-v07.2004-08-03.150000-0400	10.4.3.54	10.1.1.111	532	1356168	6024

Incident response is also an area where a simple NetFlow analysis can assist. For example, Figure 2 is the real traffic grabber graph (http://rtg.sourceforge.net/) of an Internet circuit experiencing a denial of service attack. The incident is brief, and covered less than 10 minutes time. A tcpdump of the traffic would have been nice, but the attack is brief enough that unless an automated process was in place to start the dump, it might have been over before an engineer could begin the dump.

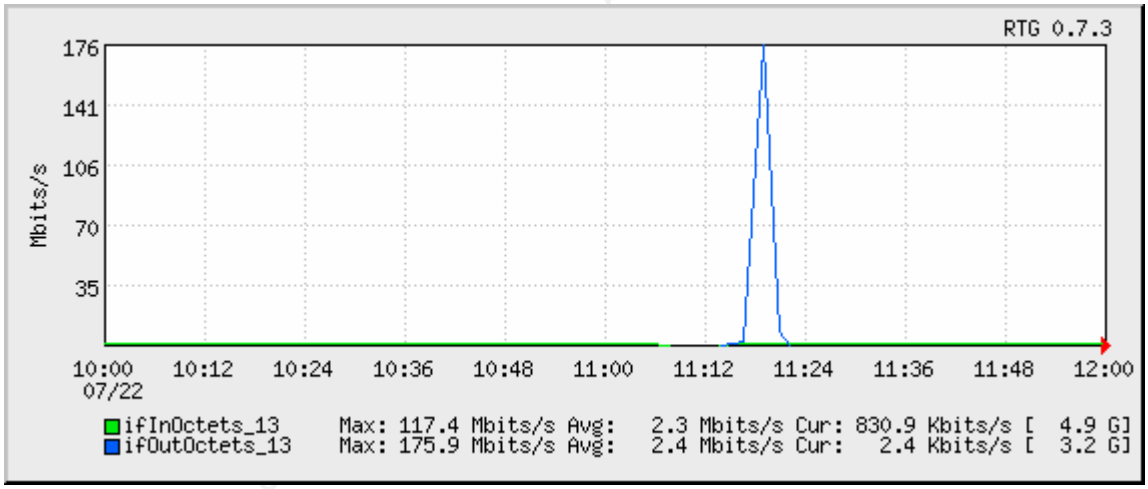


Figure 2: Internet Circuit Under Denial of Service Attack

The historical aspect of NetFlow data would probably be helpful here. You should be able to go back and look at what IP addresses were causing the most traffic, whether it was lots of packets or lots of sessions. Unfortunately there was no indication of a problem from the NetFlow point of view. However, this lack of data highlights one of the limitations of NetFlow data collection. Here is the NetFlow analysis from 11:15 to 11:30, the heart of the DOS attack:

# src IPaddr	dst IPaddr	flows	octets	packets
10.5.124.5	10.1.224.70	571	244340	2876
10.0.171.68	10.1.224.63	198	127696	954
10.2.246.9	10.1.224.69	155	1574393	3671
10.2.130.41	10.1.230.137	117	25264	523
10.4.8.173	10.1.224.70	117	61538	690
10.1.56.3	10.1.226.7	82	26541	271
10.3.27.34	10.1.225.173	74	5325	74
10.8.238.138	10.1.229.8	69	9600	223
10.3.19.88	10.1.229.8	68	8648	196
10.7.58.145	10.1.229.191	67	60489	471

There is no indication whatsoever of any abnormal traffic, yet the graph above, as well as the behavior of the network during the attack clearly showed a significant network event. During post-mortem, it was discovered that the attack, which was a variation of the Trinoo DDOS UDP flood attack (tcpdump **was** started in time to capture the information), was blocked at the Internet router because there were no UDP ports used on the destination servers. Because the traffic was dropped at the inbound interface of the router, no flows were created, and therefore no indication of the attack was passed on to the NetFlow analysis system.

Sinkhole/Tarpit Traffic Analysis

One of the pro-active activities available to a security engineer is the use of a sinkhole router to analyze emerging threats and network scans. A sinkhole could be incorporated into the existing web front diagram as shown in Figure 3.

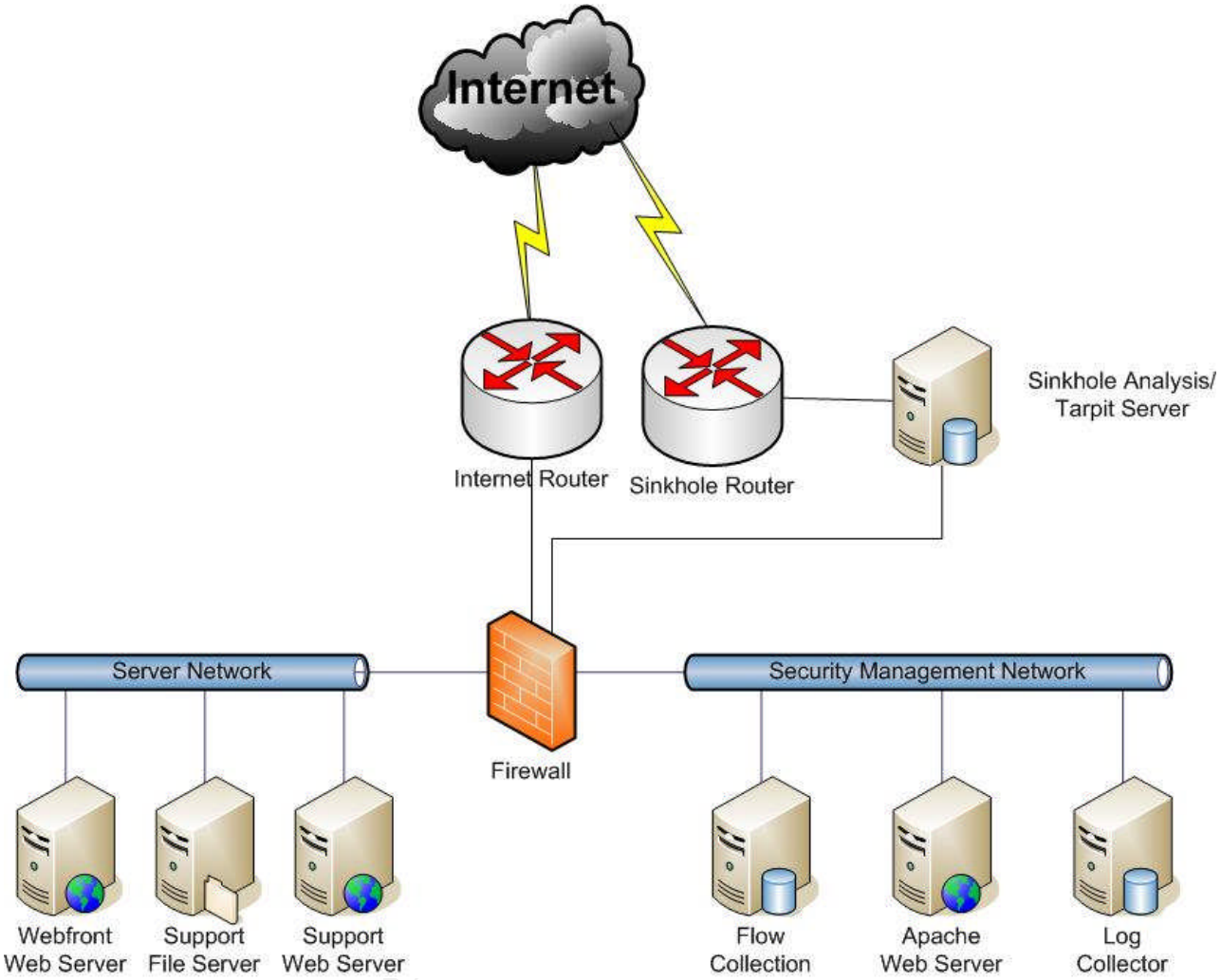


Figure 3: Production Environment with Sinkhole Router

As Barry Green and Danny McPherson point out in their presentation "Sink Holes: A Swiss Army Knife ISP Security Tool", sinkholes are essentially the network equivalent of a honey pot. A sinkhole router is a BGP speaking router that is built to analyze traffic directed towards unused network space as well as to redirect attacks away from a production environment. (Greene/McPherson 4)

One project that has successfully used sinkhole routers is the Cymru Darknet Project. The Cymru group uses the term Darknet to describe that address space that is not used. Therefore any traffic that is directed towards Darknet IP space is aberrant network traffic. The Cymru Darknet Project uses the witty worm as an example of how a sinkhole router can act as a heads up to an emerging network threat. Figure 4 is an MRTG graph on http://www.cymru.com/Darknet/index.html, showing the network traffic directed towards a Darknet during the initial stages of the witty worm release.

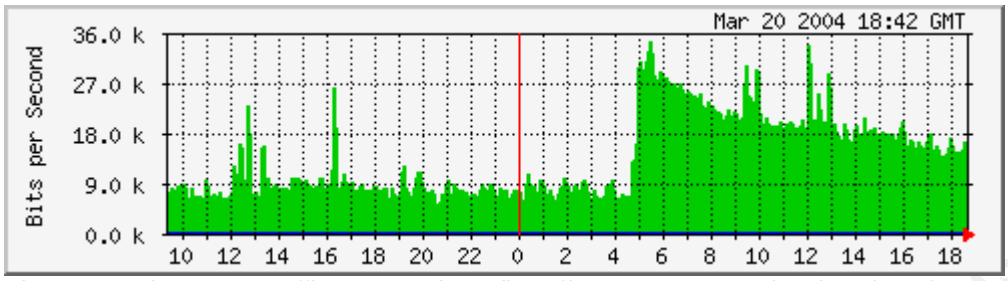


Figure 4: Witty worm traffic on a Darknet (http://www.cymru.com/Darknet/graph03.png)

The comparatively small data size of a NetFlow data file can be more appealing than the larger files created by more granular network monitoring tools. The data set provided, while not sufficient for a full analysis of emerging threats, will certainly provide statistical evidence that can point to threat origination, threat vector and can also provide some indication of the speed at which an infectious threat spreads, by looking at the increase in unique hosts hitting the sinkhole.

There is one final gotcha that might show up in a sinkhole routing scenario. Sinkholes are a common place to deploy tarpit servers. Tarpits, sometimes called “sticky honeypots”, create virtual machines and deliberately try and hold network connections open in order to slow down the rate at which a scan can be done. (Liston 1) Because a flow is terminated and reported after 15 seconds of inactivity (in default Cisco configurations) the connection itself could be held open much longer than the reported flow length indicates.

Conclusion

NetFlow analysis provides a low-impact, non-intrusive monitoring technique that is easily deployed. The flow-tools application provides a quick startup that yields data both in human-readable form as well as application-friendly comma delimited format. The information stored has all the layer 3 data necessary to provide insight into the source, destination, service and traffic load of any data stream that passes through a NetFlow enabled router.

NetFlow analysis is useful both in the intrusion detection field, through looking for abnormal traffic flows, and in the incident handling and forensics field, where its small data size allows for both historical and near-real time analysis of Internet and server traffic. It is an excellent addition to any security professional’s toolbox.

References

Bejtlich, Richard. The Tao of Network Security Monitoring. Boston: Addison Wesley, 2004.

Greene, Barry Raveendran and McPHerson, Danny. "Sink Holes: A Swiss Army Knife ISP Security Tool" URL: http://www.arbornetworks.com/downloads/research36/Sinkhole_Tutorial_June03.pdf (2 Aug, 2004).

Kretchmar, James M. Open Source Network Administration. New Jersey: Prentice Hall, 2004.

Lee, Wenke and Stolfo, Salvatore J. "Data Mining Approaches for Intrusion Detection" URL:http://www1.cs.columbia.edu/~sal/JAM/PROJECT/id-slides.ppt (2 Aug, 2004).

Liston, Tom. "LaBrea – The Tarpit" URL: http://lrp.steinkuehler.net/Packages/man/LaBrea.README (2 Aug, 2004).

Medina, Daniel. "An IDS Using NetFlow Data" URL: http://www.columbia.edu/~medina/docs/nf_ids.pdf (2 Aug, 2004).

"NetFlow Overview," February, 2003. URL:http://www.cisco.com/application/vnd.ms-powerpoint/en/us/guest/tech/tk362/c1482/ccmigration_09186a0080182b50.ppt (9 Jun. 2004).

"NetFlow Services Solutions Guide, V.2," 16 Jul. 2001. URL: http://www.cisco.com/univercd/cc/td/doc/cisintwk/intsolns/netflsol/nfwhite.pdf (9 Jun 2004).

Schweitzer, Douglas. Incident Response Indiana: Wiley, 2003.

"The Team Cymru Darknet Project" URL: http://www.cymru.com/Darknet/index.html (2 Aug, 2004).