FlowViewer

Powered by flow-tools and SiLK

Advancing Network Traffic Situational Awareness

Version 4.6

1. Introduction

FlowViewer is an open source set of tools that provide a convenient web-based user interface to Mark Fullmer's flow-tools suite, and now with version 4.0 and above, Carnegie Mellon NetSA group's netflow data capture/analyzer, SiLK. The inclusion of the underlying SiLK tool set enables FlowViewer users to continue to use the tool with the newer IPFIX netflow data protocol. FlowViewer has been developed for NASA's Earth Sciences Data and Information System (ESDIS) networks, and credit goes to NASA for their usual outstanding support of innovation.

The FlowViewer tools provide additional graphing and monitoring features by utilizing open source software including Thomas Boutrell's gd, Lincoln Stein's GD, Martien Verbruggen's GD::Graph, and Tobias Oetiker's RRDtool packages.

FlowViewer version 4.0 introduced an entirely new user interface which features an updating dashboard to provide the user with the ability to quickly establish a "network traffic situational awareness." The new main page for FlowViewer is shown in Figure 1-1 below. Different HTTP browsers will render the user interface slightly differently. Recent versions of Firefox, Opera, and Chrome are preferred.

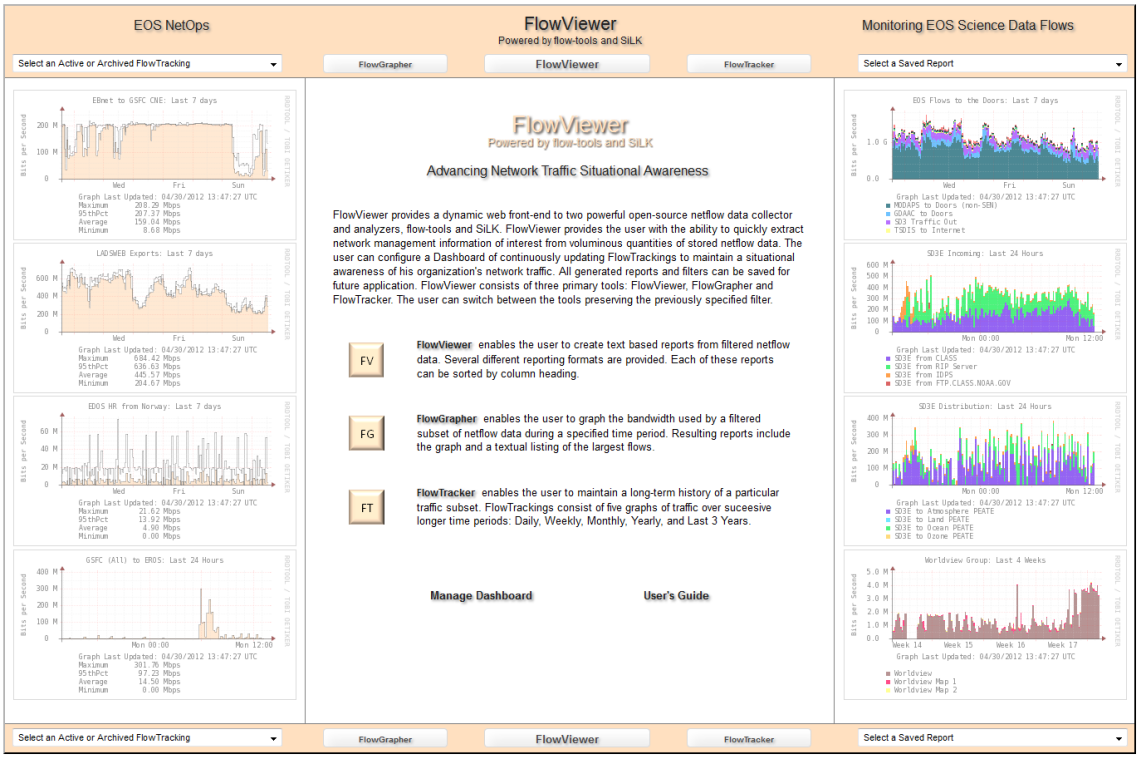

Figure 1-1 – FlowViewer 4.2 Main Page

The umbrella FlowViewer package consists of FlowViewer, FlowGrapher, and FlowMonitor. Each of these tools uses a web interface to collect filtering information and apply the filter to netflow data captured and stored by flow-tools and/or SiLK, resident on the same host. The processing of each of the tools is configured via a common configuration file. Guidance for using each of these tools is presented in separate sections below.

2. Installation

Special upgrade notes for v4.6

Version 4.6 fixes local timezone difficulties that were not fixed as advertised in version 4.5 for FlowGrapher and FlowViewer. Thanks goes to Randy Feeney. Also note that version 4.6 removes the "$time_zone" configurable parameter from FlowViewer_Configuration.pm. Timezone is now exclusively extracted from the system, using the 'date' function. This version fixes a problem with FlowGrapher not correctly displaying the smallest flows when requested (e.g., Detail Lines: -100 for smallest 100 flows.) The version fixes improper listing of very old Saved files.

Special upgrade notes for v4.5

SiLK version 3.9, together with libfixbuf version 1.6, now provides support for sFlow exports. I have not tested the interface with FlowViewer but have heard that it should be transparent as SiLK puts the sFlow data into SiLK-formatted records.

Due to a name clash in commercial space, FlowViewer version 4.5 changes the name of FlowTracker to FlowMonitor everywhere. There are a couple of things to beware of:

1. The new FM_button.png file will need to be copied into $reports_directory
2. Double check that these FlowViewer_Configuration.pm parameters are correct for you:

```
$monitor_directory       = "/var/www/html/FlowMonitor";
$monitor_short           = "/FlowMonitor";
$filter_directory        = "/var/www/cgi-bin/FlowMonitor_Files/FlowMonitor_Filters";
$rrdtool_directory       = "/var/www/cgi-bin/FlowMonitor_Files/FlowMonitor_RRDtool";
```

If this is an upgrade of an existing install and you have ongoing FlowMonitors, you'll either have to reset these parameter values (not the parameter names) back to the values you have, or change the names of the existing directories to these names.

All documentation will now use the terminology 'FlowMonitor' in place of 'FlowTracker' and 'FlowMonitor'. Note that because of the difficulty in securitizing graphics (e.g., "anonymyzing" IP addresses and names) the "FlowTracker" text on bullets and reports may still exist within the graphics in this document.

New FlowViewer_Configuration.pm parameters in v4.5:

```
$silk_compiled_localtime   - "Y" if SiLK compiled with local timezone
$ipset_directory           - Directory where FlowViewer can find IPsets
$use_bottom_pulldowns      - Will exclude pulldowns on bottom of UI
$ipfix_default_device      - Controls the default in device_name pulldown
$sensor_config_file        - Was $sensor_config_directory
$site_config_file          - Used to specify the location of silk.conf file
```

See the README release notes below for version 4.5 fixes and new functionality.

Quick Instructions for an Upgrade

If using version 4.6 to analyze IPFIX (e.g., Cisco v9) (not required):

1. Acquire, install and configure SiLK software (version 3.8.0 or more recent)

For FlowViewer:

1. Untar the package into new cgi-bin subdirectory
2. Configure FlowViewer_Configuration.pm variables as necessary
3. Create necessary directories with adequate permissions for web-server to write into
4. Copy the FV_, FG_, and FM_button.png files into $reports_directory
5. Copy the FlowViewer.css, FlowViewer.pdf files into the $reports_directory
6. Use 'convert_pre40_filters' script against old (pre v4.0) saved filters if any
7. Configure FlowViewer_Configuration.pm to point to existing FlowMonitor_Filter and FlowMonitor_RRDtool directories (either change the configuration parameter values or change the names of the directories themselves.)
8. Stop old FlowTracker_Collector and FlowTracker_Grapher
9. Start new FlowMonitor_Collector and FlowMonitor_Grapher
10. Use included FV_Relay.cgi 'User Relay' script if desired (see below)
11. Point browser to FV.cgi

Quick Instructions for Installation:

Install software dependencies:

flow-tools	*http://code.google.com/p/flow-tools*	(If collecting v5 only)
silk	*http://tools.netsa.cert.org/silk*	(If collecting IPFIX – v3.8 or newer)
libfixbuf	*http://tools.netsa.cert.org/silk*	(If collecting IPFIX)
gd	*http://www.libgd.org/Downloads*	
GD	*http://search.cpan.org/~lds/GD-2.30*	
GD::Graph	*http://search.cpan.org/~mverb/GDGraph-1.43*	
GD::Text	*http://search.cpan.org/~mverb/GDTextUtil-0.86/Text*	
RRDtool	*http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/pub*	

FlowViewer itself …

FlowViewer *https://sourceforge.net/projects/flowviewer*

For FlowViewer

2. Configure FlowViewer_Configuration.pm variables as necessary
3. Create necessary directories with adequate permissions for web-server to write into
4. Copy the FV_, FG_, and FM_button.png files into $reports_directory
5. Copy the FlowViewer.css, FlowViewer.pdf files into the $reports_directory
6. Point browser to FV.cgi

For FlowGrapher

7. Install gd (C), GD, GD::Graph, GD::Text::Align (Perl)
8. Configure graphing related FlowViewer_Configuration.pm variables as necessary
9. Point browser to FV.cgi

For FlowMonitor

10. Install RRDtool (at least version 1.4)
11. Create FlowMonitor_Filter and FlowMonitor_RRDtool directories
12. Configure FlowViewer_Configuration.pm variables as necessary
13. Start FlowMonitor_Collector, FlowMonitor_Grapher in background
14. Point browser to FV.cgi

More detailed information:

Getting and un-tarring the package:

Obtain the latest version of FlowViewer from one of the FlowViewer websites:

http://ensight.eos.nasa.gov/FlowViewer/ (deprecated)
https://sourceforge.net/projects/flowviewer

From your system's cgi-bin directory:

```
host: /var/www/cgi-bin/>tar -xvf FlowViewer_4.6.tar

FlowViewer_4.6/FG_button.png
FlowViewer_4.6/FlowGrapher.cgi
FlowViewer_4.6/FlowGrapher_Analyze.cgi
FlowViewer_4.6/FlowGrapher_Colors
FlowViewer_4.6/FlowGrapher_Ports
FlowViewer_4.6/FlowGrapher_Main.cgi
FlowViewer_4.6/FlowGrapher_Replay.cgi
FlowViewer_4.6/FlowGrapher_Sort.cgi
FlowViewer_4.6/FlowMonitor.cgi
FlowViewer_4.6/FlowMonitor_Collector
FlowViewer_4.6/FlowMonitor_Dashboard.cgi
FlowViewer_4.6/FlowMonitor_Display.cgi
FlowViewer_4.6/FlowMonitor_DisplayPublic.cgi
FlowViewer_4.6/FlowMonitor_Dumper.cgi
FlowViewer_4.6/FlowMonitor_Grapher
FlowViewer_4.6/FlowMonitor_Group.cgi
FlowViewer_4.6/FlowMonitor_Main.cgi
FlowViewer_4.6/FlowMonitor_Management.cgi
FlowViewer_4.6/FlowMonitor_Recreate
FlowViewer_4.6/FlowMonitor_Replay.cgi
FlowViewer_4.6/FlowMonitor_Thumbnail
FlowViewer_4.6/FlowViewer.cgi
FlowViewer_4.6/FlowViewer.css
```

FlowViewer_4.6/FlowViewer_CleanASCache
FlowViewer_4.6/FlowViewer_CleanFiles
FlowViewer_4.6/FlowViewer_CleanHostCache
FlowViewer_4.6/FlowViewer_CleanSiLK
FlowViewer_4.6/FlowViewer_Configuration.pm
FlowViewer_4.6/FlowViewer_Main.cgi
FlowViewer_4.6/FlowViewer_Replay.cgi
FlowViewer_4.6/FlowViewer_Save.cgi
FlowViewer_4.6/FlowViewer_SaveManage.cgi
FlowViewer_4.6/FlowViewer_Sort.cgi
FlowViewer_4.6/FlowViewer_UI.pm
FlowViewer_4.6/FlowViewer_Utilities.pm
FlowViewer_4.6/FM_button.png
FlowViewer_4.6/FV.cgi
FlowViewer_4.6/FV_button.png
FlowViewer_4.6/FV_Relay.png
FlowViewer_4.6/NamedInterfaces_Devices
FlowViewer_4.6/NamedInterfaces_Exporters
FlowViewer_4.6/README
FlowViewer_4.6/Flow_Working
FlowViewer_4.6/logs
FlowViewer_4.6/tools
FlowViewer_4.6/tools/analyze_flowmonitor_debug
FlowViewer_4.6/tools/analyze_netflow_packets
FlowViewer_4.6/tools/convert_pre40_filters
FlowViewer_4.6/tools/create_ports_file
FlowViewer_4.6/tools/date_to_epoch_gm
FlowViewer_4.6/tools/date_to_epoch_local
FlowViewer_4.6/tools/epoch_to_date_gm
FlowViewer_4.6/tools/epoch_to_date_local
FlowViewer_4.6/tools/flowcapture_restart
FlowViewer_4.6/tools/flow-capture-table.conf
FlowViewer_4.6/tools/flowmonitor_archive_restore
FlowViewer_4.6/tools/flowmonitor_grapher_nonlazy
FlowViewer_4.6/tools/flowmonitor_restart
FlowViewer_4.6/tools/performance_check
FlowViewer_4.6/tools/resize_rrdtools
FlowViewer_4.6/tools/rsync_flows
FlowViewer_4.6/tools/rsync_htmls
FlowViewer_4.6/tools/rsync_monitors
FlowViewer_4.6/tools/rwflowpack_start

This has created a cgi-bin subdirectory called FlowViewer_4.6 which includes the whole package. It may be the case that you have created this directory as a user that is not the same as the owner of the web server process. The web server may, depending on your configuration (more later) need to write into this directory. If that is the case, you must give this directory adequate ‘write’ permissions. For example, provide it with ‘0775’ (e.g., chmod 0775

/var/www/cgi-bin/FlowViewer_4.6) to allow a web server whose process owner is 'apache.' 'apache' would be placed in the group that owns the directory.

If you plan to use FlowMonitor, you'll need to establish directories to hold the permanent filter files and rrdtool databases that will be created. These are defined by the $filter_directory and $rrdtool_directory parameters. If you've been using an earlier version of FlowViewer, and you've been using the FlowTracker tool, you'll want to either set the $filter_directory and $rrdtool_directory parameters to the existing directories, or rename the existing directories.

2.1 Latest Release Information

```
# # README (this file) FlowViewer V4.6  Date: 04/06/2015
#
# Emergency fix to FlowMonitor_Collector to fix $no_devices_or_exporters
# not setting times (thanks Vladimir Stepanov) - 04/06/2105
#
# FlowViewer is a set of three tools (FlowViewer, FlowGrapher,
# FlowMonitor) that create text reports, graph reports, and
# long-term monitor reports from flow-tools and SiLK captured
# and stored netflow data. FlowViewer can run with both flow-tools
# and SiLK simultaneously. Flow-tools can handle up to v7; SilK
# can handle v5, v9, and IPFIX. The User's Guide is very helpful.
#
# Software Dependencies:
#
#   flow-tools  http://code.google.com/p/flow-tools (If collecting v5 only)
#   SiLK        http://tools.netsa.cert.org/silk    (If collecting IPFIX)
#   libfixbuf   http://tools.netsa.cert.org/silk    (If collecting IPFIX)
#   gd          http://www.libgd.org/Downloads
#   GD          http://search.cpan.org/~lds/GD-2.30
#   GD::Graph   http://search.cpan.org/~mverb/GDGraph-1.43
#   GD::Text    http://search.cpan.org/~mverb/GDTextUtil-0.86/Text
#   RRDtool     http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/pub
#
# Quick Upgrade
#
#   0. If using SiLK, must upgrade to v3.8.0 or newer
#   1. Untar the package into a new cgi-bin subdirectory
#   2. Configure FlowViewer_Configuration.pm variables to your environment
#      and create all necessary directories with proper permissions
#   3. Replace old logos with new buttons (will be done automatically)
#   4. Copy FlowViewer.css, FlowViewer.pdf to $reports_directory
#   5. Configure FlowViewer_Configuration.pm to point to existing
#      FlowMonitor_Filter and FlowMonitor_RRDtool directories
#      [For prior v4.5 you can rename these directories to:
#      FlowMonitor_Filter and FlowMonitor_RRDtool or change the
#      configuration variables to point to the old directories.]
#   6. Configure new FlowViewer_Configuration.pm
#   7. Stop old FlowMonitor_Collector and FlowMonitor_Grapher
#      [Upgrading from v4.4 and prior you will be stopping FlowTracker_Collector
#      and FlowTracker_Grapher.]
#   8. Start new FlowMonitor_Collector and FlowMonitor_Grapher
#   9. Copy NamedInterfaces_Devices, names file, user logo to new directory
#  10. (If upgrading from pre v4.0) Run convert_pre40_filters against existing filters
#      (ie FlowViewer_SavedFilters)
#  11. Use included 'User Relay' scripts if desired (recommended - see below)
#
# Quick Install
#
#   1. Untar into cgi-bin subdirectory
#
#   For netflow v5 and older (option):
#
#   2. Download, install, configure flow-tools
#
#   For IPFIX (e.g., v9 - also handles v5):
#
#   3. Download, install, configure SiLK (v3.8.0 or newer) and libfixbuf
#
#   For sflow
#
#   4. From SiLK FAQ:
#
#      "Support for sFlow v5 is available as of SiLK 3.9.0 when you configure
#      and build SiLK to use v1.6.0 or later of the libfixbuf library."
#
#   For FlowViewer
#
#   5. Configure FlowViewer_Configuration.pm variables as necessary
```

```
#    6. Create all necessary directories with proper permissions
#    7. Copy FlowViewer.css, FlowViewer.pdf to $reports_directory
#    8. Point browser to FV.cgi
#
#    For FlowGrapher
#
#    9. Install gd (C), GD (Perl), GD::Graph (Perl) GD::Text (Perl)
#   10. Configure FlowViewer_Configuration.pm variables as necessary
#   11. Point browser to FV.cgi
#
#    For FlowMonitor
#
#   12. Install RRDtool (at least version 1.4)
#   13. Create FlowMonitor_Filter and FlowMonitor_RRDtool directories
#   14. Configure FlowViewer_Configuration.pm variables as necessary
#   15. Start FlowMonitor_Collector, FlowMonitor_Grapher in background
#   16. Point browser to FV.cgi
#
#    For all FlowViewer tools
#
#   17. Review all FlowViewer directories and files for proper permissions
#
# Version 4.6 Release Notes
#
#  Version 4.6 fixes local timezone difficulties that were not fixed as
#  advertised in version 4.5 for FlowGrapher and FlowViewer. Thanks goes
#  to Randy Feeney. Also note that version 4.6 removes the "$time_zone"
#  configurable parameter from FlowViewer_Configuration. Timezone is now
#  exclusively extracted from the system, using the 'date' function. This
#  version fixes a problem with FlowGrapher not correctly displaying the
#  smallest flows when requested (e.g., Detail Lines: -100 for smallest
#  100 flows.) Fixes improper listing of very old Saved files.
#
# Version 4.5 Release Notes
#
#  Version 4.5 resolves an unfortunate name clash in commercial space and
#  renames FlowTracker to FlowMonitor. The situation where SiLK data is
#  saved in UTC (GMT) time, but the system is left in local time has been
#  fixed (thanks to Kees Leune.) A new configuration variable
#  "$silk_compiled_localtime" has been added for the environment where SiLK
#  has been comipled with the --enable-localtime switch. FlowGrapher_Analyze
#  has been fixed to handle hyper-links to IPv6 hosts properly. SiLK IPsets
#  can now be input through the various tool menus. A problem with
#  multi-word Dashboards and Group creation has been fixed. Corrected
#  flows/second initiated calculation. Added the ability to bypass the
#  printing of pulldowns on the bottom service bar. Fixed an error with
#  filtering on port equal to '0'. Fixed 'Len' field output for some
#  FlowGrapher reports. New parameter: $ipfix_default_device allows IPFIX
#  users to pre-select a primary device (e.g., using one sensor only.)
#  Extended pie-charts to some Printed reports. A new parameter
#  $site_config_file is added to make it easier to accomodate various
#  SiLK stored data file structures.
#
#  New FlowViewer_Configuration.pm parameters in v4.5:
#  $silk_compiled_localtime   - "Y" if SiLK compiled with local timezone
#  $ipset_directory           - Directory where IPsets can be found
#  $use_bottom_pulldowns      - Will exclude pulldowns on bottom of UI
#  $ipfix_default_device      - Controls the default in device_name pulldown
#  $sensor_config_file        - Changed from $sensor_config_directory
#  $site_config_file          - Left blank (= "";) will look in rootdir
#
#  Note: the rename of FlowTracker to FlowMonitor includes default names
#  for FlowMonitor related directories. The defaults that will previal if
#  no changes are made are:
#
#  $monitor_directory       = "/var/www/html/FlowMonitor";
#  $monitor_short           = "/FlowMonitor";
#  $filter_directory        = " ... /FlowMonitor_Files/FlowMonitor_Filters";
#  $rrdtool_directory       = " ... /FlowMonitor_Files/FlowMonitor_RRDtool";
#
#  For users who are upgrading, these can be revised back to 'FlowTracker'
```

```
#  (or whatever) with no problem. The alternative is to simply rename the
#  existing directories.
```

[Note: Please see README file in distribution for full history and credits.]

2.2 Dependencies

FlowViewer requires that you have flow-tools, SiLK (if you want to collect IPFIX (e.g., v9) netflow data, flow data files, a web-server, Perl, and the FlowViewer package all installed on the same machine.

You will need flow-tools if you choose to use it as a collector for netflow versions v7 and earlier. Written by Mark Fullmer, flow-tools versions up to 0.68 are available at:

http://www.splintered.net/sw/flow-tools/

Paul Komkoff Jr., et. al. are keeping a newer 'fork' of flow-tools at:

http://code.google.com/p/flow-tools/

See the next section for installation installs.

The Software Engineering Institute (SEI) at Carnegie Mellon University has formed a CERT NetSA group that has developed netflow collection and analysis software that can handle netflow v5, v9 and IPFIX exports. You will need:

- SiLK, at least version 3.8
- libfixbuf, at least version 1.1.0.

Note that SiLK version 3.9 together with libfixbuf version 1.6 now collect sFlow data exports. The SiLK web site has the software for download and an excellent library of SiLK documentation. The web site is:

http://tools.netsa.cert.org/silk/

If you are planning on using FlowGrapher, you will need to install Thomas Boutrell's gd, Lincoln Stein's GD, and Martien Verbruggen's GD::Graph packages. They can be found at:

gd: *http://www.libgd.org/Downloads*
GD package: *http://search.cpan.org/~lds/GD-2.30/*
GD::Graph package: *http://search.cpan.org/~mverb/GDGraph-1.43/*
GD::Text::Align*: http://search.cpan.org/~mverb/GDTextUtil-0.86/Text/Align.pm*

If you are planning on using FlowMonitor, you will need to install Toby Oetiker's RRDtool package. This package can be found at:

http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/pub/ (version 1.4.4 or beyond)

For each of these you should make sure you have the latest stable versions.

2.2.1 flow-tools installation

The flow-tools software is excellent. It is very stable and has great flexibility through so many options. With FlowViewer, however, the only component that you have to work with is flow-capture. FlowViewer will automatically invoke several of the other components for you.

The man pages are very informative. The most recent version:

flow-tools man pages: http://www.splintered.net/sw/flow-tools/docs/

A typical flow-capture command may look like this:

```
flow-capture -p /var/flows/pids/flowtool.pid -w /var/flows/router_1 -E5G -S3 0/0/2050
```

The case above stores netflow data separately for each device instead of collecting from multiple exporters into a single directory structure. You can see this by the fact that the directory is identified using the name of a single device, 'router_1'. There would be a second, similar command for a second device (e.g., 'router_2') where the only difference in the command syntax would be to replace 'router_1' with 'router_2' and to increment the receiving port number from '2050' to '2051', say. Both commands would be executed resulting in two flow-captures running simultaneously. Actually, here at NASA GSFC, we've run 23 flow-captures simultaneously. Each one takes a surprisingly little amount of CPU (actually they're more I/O bound), with four of them receiving from very busy devices.

The -p parameter identifies a directory where flow-capture will store the process identifier (PID) for the flow-capture process. The -w parameter identifies the location for depositing the netflow data. The -E parameter identifies how much disk space (5 Gigabytes) should be allocated to this collection, with flow-capture aging out netflow data once the limit is reached.

The -S3 parameter informs flow-capture to write a status message to the log file (generally e.g., /var/log/cflowd.log) every 3 minutes. The 0/0/2050 notation informs flow-capture to expect netflow data from any device IP address (use of '0') and to capture it with any destination IP address. These can be specific IP addresses as well. The UDP port number for receiving packets from the device is '2050.'

At this point you are ready to modify the @devices field in the FlowViewer_Configuration.pm file to match the collection directory name (i.e., 'router_1') and you are ready to go.

If you wish to collect from multiple exporters, all exporting to the same UDP port, your flow-capture syntax might look like this:

```
flow-capture -p /var/flows/pids/flowtool.pid -w /var/flows/all_routers -E5G -S3 0/0/2050
```

In this case you would set up the following relevant parameters in FlowViewer_Configuration.pm:

```
$exporter_directory = "/var/flows/all_routers";
@exporters = ("192.168.100.1:New York Router","192.168.100.2:Prague Router");
```

Finally, you may simply collect all netflow data (from one or more devices) into a single directory structure and not use named devices or exporters. The flow-capture command might look like:

```
flow-capture -p /var/flows/pids/flowtool.pid -w /var/flows/all_flows -E5G -S3 0/0/2050
```

In this case you would set up the following relevant parameters in FlowViewer_Configuration.pm:

```
$exporter_directory = "/var/flows/all_flows";
$no_devices_or_exporters = "Y";
```

2.2.1 SiLK installation

The first thing to note is that the SiLK documentation is excellent, comprehensive, and should be reviewed. The key documents for understanding and installation are:

The SiLK Installation Handbook SiLK-3.0 –
http://tools.netsa.cert.org/silk/install-handbook.html

Analyst's handbook - *Using SiLK for Network Traffic Analysis –*
http://tools.netsa.cert.org/silk/analysis-handbook.pdf

The user must first install libfixbuf from the SiLK web site. This is a straight *./configure, make, make install.* The only possible difficulty may be that your *glib* version is not higher than v2.4 which libfixbuf requires. Installing libglib2.4 might not be straightforward; this worked for me:

sudo apt-get install libglib2.0-dev libpango1.0-dev

For SiLK, FlowViewer version 4.6 requires SiLK version 3.8.0 or newer. Note: You'll probably want IPv6 flow record support, so when compiling, a typical. /configure statement might look like (had to specify location of libfixbuf package config file):

./configure --enable-data-rootdir=/var/flows --enable-ipv6
--with-libfixbuf=/usr/local/lib/pkgconfig

Typical configure results:

```
    * Configured package:           SiLK 3.10.0
    * Host type:                    x86_64-unknown-linux-gnu
    * Source files ($top_srcdir):   .
    * Install directory:            /usr/local
    * Root of packed data tree:     /data/flows
    * Packing logic:                via run-time plugin
    * Timezone support:             UTC
    * Default compression method:   SK_COMPMETHOD_NONE
    * IPv6 network connections:     YES
    * IPv6 flow record support:     YES
    * IPFIX collection support:     YES (-pthread -L/usr/local/lib -L/lib64 -lfixbuf -
lpthread -lgthread-2.0 -lglib-2.0)
    * NetFlow9 collection support:  YES
    * sFlow collection support:     YES
    * Fixbuf compatibility:         libfixbuf-1.6.2 >= 1.6.0
    * Transport encryption support: NO (gnutls not found)
```

```
    * IPA support:                 NO
    * ZLIB support:                YES (-lz)
    * LZO support:                 NO
    * LIBPCAP support:             NO
    * C-ARES support:              NO
    * ADNS support:                NO
    * Python interpreter:          /usr/bin/python
    * Python support:              NO
    * Build analysis tools:        YES
    * Build packing tools:         YES
    * Compiler (CC):               gcc
    * Compiler flags (CFLAGS):     -I$(srcdir) -I$(top_builddir)/src/include -
I$(top_srcdir)/src/include -DNDEBUG -D_ALL_SOURCE=1 -D_GNU_SOURCE=1 -g -DDEBUG -
I../include -fno-strict-aliasing -Wall -W -Wmissing-prototypes -Wformat=2 -Wdeclaration-
after-statement -Wpointer-arith
    * Linker flags (LDFLAGS):
    * Libraries (LIBS):            -lz -ldl -lm
```

You may need to run '*ldconfig*' to 'create the necessary links and cache to the most recent shared libraries.'

Below is a quick guide to what is required of SiLK to run FlowViewer. For users with experience with flow-tools, SiLK raw netflow storage can be arranged in a manner very similar to flow-tools. The only real difference is an "extra" layer between the device and the years. It is straightforward to integrate SiLK storage along with flow-tools (particularly for existing flow-tools users.). flow-tools and (the flow-tools-like) SiLK storage essentially look like this:

```
/var/flows/FT_device_1/
                       2011/
                       2012/
                            2012-01/
                            2012-02/
                            2012-03/
                            2012-04/
                            2012-05/
                                    2012-05-01/
                                    2012-05-02/
                                    2012-05-03/
                                    2012-05-04/
                                               ft-v05.2012-05-04.000002+0000
                                               ft-v05.2012-05-04.001501+0000
                                               ft-v05.2012-05-04.003001+0000
                                               ft-v05.2012-05-04.004501+0000
                                               ft-v05.2012-05-04.010001+0000
                                               ft-v05.2012-05-04.011501+0000
                                               ft-v05.2012-05-04.013000+0000
                                               ft-v05.2012-05-04.014501+0000
                                               ft-v05.2012-05-04.020001+0000

/var/flows/silk_device_1/ext2ext/
                         /in/
                              2011/
                              2012/
                                   01/
                                   02/
                                   03/
                                   04/
                                   05/
                                      01/
                                      02/
                                      03/
                                      04/
```

```
                                in-S1_20120504.00
                                in-S1_20120504.01
                                in-S1_20120504.02
                                in-S1_20120504.03
                                in-S1_20120504.04
                    /int2int/
                    /inweb/
                    /out/
                    /outweb/
                    silk.conf
```

In the above case, FlowViewer will look for SiLK exporting devices in the same $flow_data_directory that that flow-tools exporting devices are found. In the layout above, $flow_data_directory = '/var/flows'. The example shows a directory layout for a flow-tools device (FT_device_1) and a SiLK device (silk_device_1) expanded for May 4, 2012. Notice the six directories (ext_to_ext, in, int2int, inweb, out, outweb) immediately beneath the SiLK device in the layout. This is the standard SiLK directory structure.

As of version 4.4, much more flexibility has been provided for the experienced SiLK user. The FlowViewer, FlowGrapher, and FlowMonitor input screens (e.g., see figure 4-1 below) now permit the user to specify the following fields (used to accommodate a great variety of SiLK file structures.):

- Data rootdir
- Class
- Flowtype
- Type
- Sensors
- Other Switches

As of version 4.5, users with a single rootdir structure (i.e., not set up with devices as in the example above, and with 'in', 'inweb', etc. subdirectories appearing directly below the rootdir) the user can simplify data input by specifying the following:

$silk_data_directory = (user's SiLK rootdir);
@ipfix_devices = ("Site");
$ipfix_default_device = "Site";

'Site' is a special device_name that when used will prevent the user from having to select a device each time when they are actually working without devices. If FlowViewer sees that this variable is non-empty, it will automatically adjust the device_name pulldown and selection appropriately. Then the user can control all SiLK data selection via the text fields (described above) together with their default definitions in the FlowViewer_Configuration.pm file.

The minimal configuration that will get SiLK working involves modifying two basic file templates, silk.conf and sensor.conf.

The sensor.conf file can look as simple as:

```
probe S1 netflow-v9
    listen-on-port 9997
```

```
    protocol udp
end probe

group Main-Internal
    ipblocks 192.168.0.0/18
    ipblocks 172.16.0.0/16
end group

group Second-Internal
    ipblocks 192.168.128.0/18
    ipblocks 172.17.0.0/16
end group

sensor S1
    netflow-v9-probes S1
    internal-ipblocks Second-Internal
    external-ipblocks remainder
end sensor
```

This file goes in the $sensor_config_directory. The silk.conf file can look as simple as:

```
# silk.conf for the "twoway" site
# RCSIDENT("$SiLK: silk.conf 15669 2010-04-21 21:18:05Z mthomas $")

# The syntactic format of this file
#    version 2 supports sensor descriptions, but otherwise identical to 1
# version 2

sensor 0 v9-exporter   "
sensor 1 S1    "My v9 Testing Router"
sensor 2 S2
sensor 3 S3
sensor 4 S4
sensor 5 S5
sensor 6 S6
sensor 7 S7
sensor 8 S8
sensor 9 S9
sensor 10 S10
sensor 11 S11
sensor 12 S12
sensor 13 S13
sensor 14 S14

class all
    sensors S0 S1 S2 S3 S4 S5 S6 S7 S8 S9 S10 S11 S12 S13 S14
end class

# Editing above this line is sufficient for sensor definition.

# Be sure you understand the workings of the packing system before
# editing the class and type definitions below.  In particular, if you
# change or add-to the following, the C code in packlogic-twoway.c
# will need to change as well.

class all
    type  0 in      in
    type  1 out     out
    type  2 inweb   iw
    type  3 outweb  ow
    type  4 innull  innull
    type  5 outnull outnull
```

```
    type  6 int2int int2int
    type  7 ext2ext ext2ext
    type  8 inicmp  inicmp
    type  9 outicmp outicmp
    type 10 other   other

    default-types in inweb inicmp
end class

default-class all

# The layout of the tree below SILK_DATA_ROOTDIR.
# Use the default, which assumes a single class.
# path-format "%T/%Y/%m/%d/%x"

# The plug-in to load to get the packing logic to use in rwflowpack.
# The --packing-logic switch to rwflowpack will override this value.
# If SiLK was configured with hard-coded packing logic, this value is
# ignored.
packing-logic "packlogic-twoway.so"
```

This file goes in the SiLK device directory (e.g. /var/flows/silk_device_1).

A general SiLK environment is shown in figure 2.1 below.

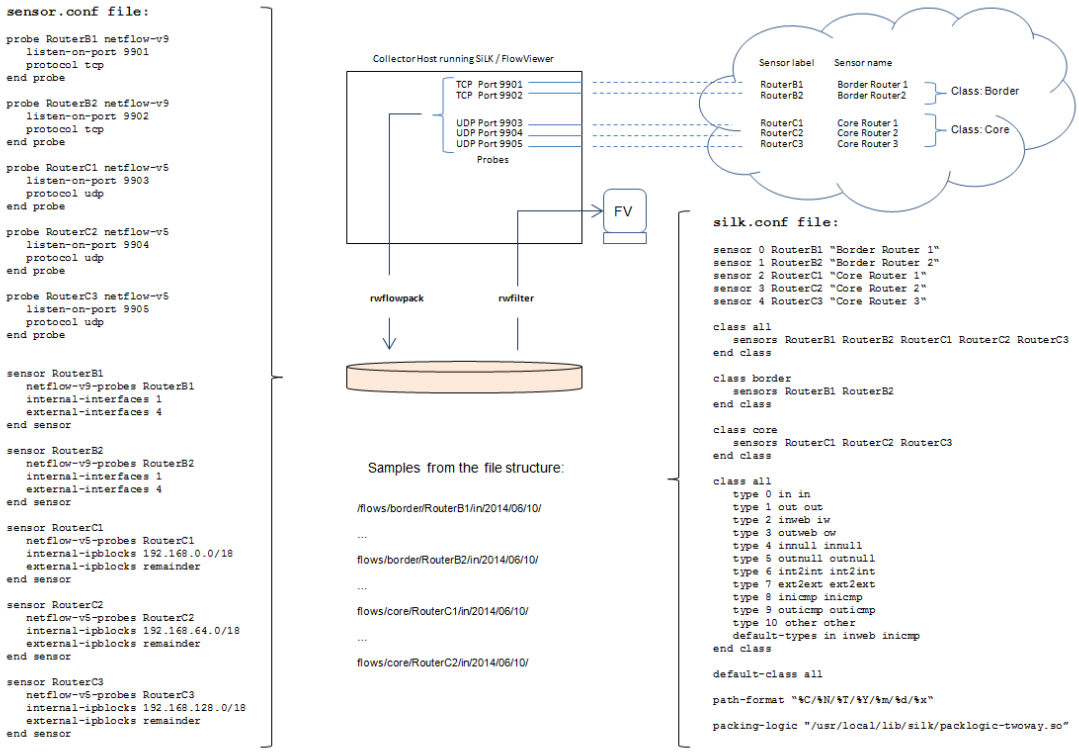

Figure 2-1 – General SiLK environment

It is hoped that the version 4.6 FlowViewer will be able to accommodate any user created variations of the general SiLK environment. To initiate the SiLK capture, here's an example script (provided with the distribution: *rwflowpack_start*, see section 7) with one command:

rwflowpack --no-daemon --root-directory=/data/V9-exporter --site-config-file=/data/silk.conf --sensor-configuration=/data/sensor.conf --log-directory=/var/log/V9-exporter --pack-interfaces --sensor-name=V9-exporter 2>&1

The script is run in the background: *host>./rwflowpack_script_S1&*

The relevant SiLK FlowViewer_Configuration.pm variables, explained in more detail in section 4 include:

```
# SiLK parameters

$silk_data_directory     = "/data/flows";
$silk_bin_directory      = "/usr/local/bin";
$sensor_config_directory = "/data/flows";
$silk_compiled_localtime = "";          # Set to "Y" if you compiled SiLK with --enable-localtime switch

$silk_capture_buffer_pre = (125 * 60);  # Start of SiLK file concatenation
$silk_capture_buffer_post= (5 * 60);    # End of SiLK file concatenation

$silk_init_loadscheme    = 1;           # For Flows Initiated/Second - see SiLK rwcount documentation
$silk_active_loadscheme  = 5;           # For Flows Active/Second - see SiLK rwcount documentation
$silk_class_default      = "";          # General SiLK file structure info. silk.conf, sensor.conf
$silk_flowtype_default   = "";          # General SiLK file structure info. silk.conf, sensor.conf
$silk_type_default       = "all";       # General SiLK file structure info. silk.conf, sensor.conf
$silk_sensors_default    = "";          # General SiLK file structure info. silk.conf, sensor.conf
$silk_switches_default   = "";          # General SiLK file structure info. silk.conf, sensor.conf
```

2.2.3 Time Zones

Network monitor tools that collect and analyze data for networks that span time zones should always maintain data in UTC time. This becomes useful when comparing netflow or log times when tracking down an event. However for single timezone networks (e.g., a campus network) UTC is not as handy as the local system time.

Flow-tools will always store data in the system's local time zone. Oftentimes the local time zone is UTC, but not always of course. This winds up making processing straightforward for FlowViewer whether the system time is UTC or not.

SiLK is a little different. By default SiLK will store captured netflow data in the UTC timezone whether the local system time is UTC or some other local time. Prior to FlowViewer version 4.6 it was not easy to deal with environments where the system time was kept local. Because of its heritage from flow-tools, FlowViewer always assumed stored data was in the system time zone. This would produce incorrect results whenever a user kept the system time zone local, but used the SiLK default compile process which will set up to *store* data in UTC. The easiest method for dealing with the difficulty became to convert the local system to UTC which is not optimal for some users.

The SiLK compile process gives users the option to use system local time instead of UTC for *storing* the data. This worked as well for resolving the situation provided users were willing to re-compile SiLK with that option. FlowViewer v4.6 however is now able to discern whether the system time zone is different from the data storage time zone and handle the differences. FlowViewer will assume UTC for stored data unless the $silk_compiled_localtime configuration parameter is set to "Y", and it will then use the local time for accessing the data. Version 4.6 has removed the $time_zone parameter from FlowViewer_Configuration.pm and will obtain the local time zone from a `date` system call.

3. Configuring for your environment

3.1 Configuration Parameters

The file FlowViewer_Configuration.pm is used to configure each of FlowViewer, FlowGrapher, and FlowMonitor. If you are using SiLK only (not also using flow-tools) leave the flow-tools variables unchanged (do not remove.) Most of the parameters in the file do not need to be changed. Those that might require change are discussed below:

Parameter	Description	Example
$ENV{PATH}	Set this variable to include directories to your basic system commands (e.g., rm, mv, etc.)	$ENV{PATH} = ':/usr/local/bin:/usr/sbin';
$FlowViewer_server	This variable should be set to the IP address of the machine that is running your flow-tools, web-server, and the FlowViewer software. Used only for the link that appears in Threshold Notification emails.	$FlowViewer_server = “192.168.0.1";
$FlowViewer_service	Set this parameter according the service which your web-server is running. The options are ‘http’ or the encrypted ‘https.’ Used only for the link that appears in Threshold Notification emails.	$FlowViewer_service = "https";
$reports_directory	This is the directory into which you will put the FlowViewer.css file, FlowViewer.pdf (User’s Guide) and the FV_Button.png, FG_Button.png and the FM_Button.png files which may be placed there automatically. IMPORTANT: Must have adequate ‘write’ permissions so that the web-server can write into this directory as necessary.	$reports_directory = "/var/www/html/FlowViewer_4.6";
$reports_short	This parameter is used within scripts that are being run by the web-server to locate directories off of the web-server default short-cuts. Typically, the web-server omits the directory information pointing to the root of the ‘htdocs’ and ‘cgi-bin’ directories. See this in comparison to the parameter above.	$reports_short = "/FlowViewer_4.6";
$graphs_directory	This is the directory into which the FlowViewer scripts will put FlowViewer and FlowGrapher graphical output (e.g., graphs and pie-charts). The directory should be somewhere beneath your …/htdocs directory. IMPORTANT: Must have adequate ‘write’ permissions so that the web-server can write into this directory as necessary.	$graphs_directory = "/var/www/html/FlowGrapher_4.6";
$graphs_short	This parameter is used within scripts that are being run by the web-server to locate directories off of the web-server default short-cuts. Typically, the web-server omits the directory information pointing to the root of the ‘htdocs’ and ‘cgi-bin’ directories. See this in comparison to the parameter above.	$graphs_short = "/FlowGrapher_4.6";

$monitor_directory	This is the directory which will be used to store your FlowMonitor graphs. Each Monitor will be structured as a subdirectory of this directory, where the subdirectory contains the five RRDtool graphs. IMPORTANT: Must have adequate 'write' permissions so that the web-server can write into this directory as necessary. The directory should be somewhere beneath your …/htdocs directory.	$monitor_directory = "/var/www/html/FlowMonitor_4.0";
$monitor_short	This parameter is used within scripts that are being run by the web-server to locate directories off of the web-server default short-cuts. Typically, the web-server omits the directory information pointing to the root of the 'htdocs' and 'cgi-bin' directories. See this in comparison to the parameter above.	$monitor_short = "/FlowMonitor_4.0";
$cgi_bin_directory	This is the directory into which you have placed the FlowViewer scripts. It should somewhere beneath you system's main cgi-bin directory. IMPORTANT: Must have adequate 'write' permissions so that the web-server can write into this directory as necessary.	$cgi_bin_directory = "/var/www/cgi-bin/FlowViewer_4.0";
$cgi_bin_short	This parameter is used within scripts that are being run by the web-server to locate directories off of the web-server default short-cuts. Typically, the web-server omits the directory information pointing to the root of the 'htdocs' and 'cgi-bin' directories. See this in comparison to the parameter above.	$cgi_bin_short = "/FlowViewer_4.0";
$work_directory	This directory is used to hold intermediate files generated during processing, including save files created in case someone wants to save the file. IMPORTANT: Must have adequate 'write' permissions so that the web-server can write into this directory as necessary. Also, some intermediate files are quite large, so the size of the partition that holds this directory should be of adequate size.	$work_directory = "/tmp";
$names_directory	This directory specifies where you would like to store the 'names' file created in the process of resolving IP addresses to hosts names. The file is used to cache names for much quicker retrieval than using the 'dig' function to get them. It is a good idea to keep this file in a more permanent place (e.g., not /tmp) since temporary directories are cleaned out on system reboots, etc.. IMPORTANT: Must have adequate 'write' permissions so that the web-server can write into this directory as necessary.	$names_directory = "/var/www/cgi-bin/FlowViewer_4.0";
$ipset_directory	Points to the directory where a user stores his IPset files. IPsets are files that contain large numbers of IP addresses. SiLK use these to optimize processing against large lists of IP addresses.	$ipset_directory = "/var/www/cgi-bin/FlowViewer_4.6";
$filter_directory	This directory is used to store permanent filter files associated with the long-term Monitors established using FlowMonitor.	$filter_directory = "/var/www/cgi-bin/FlowMonitor_Files/FlowMonitor_Filters";

	IMPORTANT: Must have adequate 'write' permissions so that the web-server can write into this directory as necessary. This directory must be kept around through FlowViewer version updates if the user wishes to continue with the existing Monitors.	
$rrdtool_directory	This directory is used to store permanent RRDtool files associated with the long-term Monitors established using FlowMonitor. IMPORTANT: Must have adequate 'write' permissions so that the web-server can write into this directory as necessary. This directory must be kept around through FlowViewer version updates if the user wishes to continue with the existing Monitors.	$rrdtool_directory = "/var/www/cgi-bin/FlowMonitor_Files/FlowMonitor_RRDtool";
$save_directory	This directory will hold all saved reports and graphs.	$save_directory = "/var/www/html/FlowViewer_Saves";
$save_short	Short reference to save Directory used by scripts.	$save_short = "/FlowViewer_Saves";
$dashboard_directory	This directory stores the thumbnail graphs (updated every 5 minutes) that make up the dashboard.	$dashboard_directory = "/var/www/html/FlowViewer_Dashboard";
$dashboard_short	This parameter is used within scripts that are being run by the web-server to locate directories off of the web-server default short-cuts.	$dashboard_short = "/FlowViewer_Dashboard";
@other_dashboards	Array is configured to contain fully specified directory locations for each ***additional*** dashboard. If using only one dashboard set this to the empty set (e.g., @other_dashboards = ();)	@other_dashboards = ("/var/www/html/SOC","/var/www/html/NetOps");
@dashboard_titles	Array is configured with ***all*** dashboard titles when using more than one dashboard. Note that when using a single dashboard only, the Left and Right Titles are used and this array should be set to empty (e.g., @dashboard_titles = ();)	@dashboard_titles = ("Performance","SOC","NetOps"); Or @dashboard_titles = ("John","Robert","Rama");
$flow_data_directory	This is the directory that sits at the top of subdirectories that store raw flow-tools netflow data. Note that if you are using EXPORTER_ID to distinguish your devices, instead of storing each device's netflow data in a separate directory, then you can ignore this field and use the $exporter_directory.	$flow_data_directory = "/var/flows";
$silk_data_directory	This is the directory that sits at the top of subdirectories that store raw SiLK netflow data. It is the default value for the "Data Rootdir" (--data-rootdir=)input field.	$silk_data_directory = "/var/flows";

$exporter_directory (flow-tools only)	This is the directory that stores all of the netflow data that you are exporting when you are capturing data from more than one device onto the same port. This is opposed to capturing data from different devices on different ports (i.e., multiple instantiations of *flow-capture*), and then storing each device's netflow data into a different directory, distinguished by *device_name*.	$exporter = "/var/flows/all_routers";
$flow_bin_directory (flow-tools only)	This directory contains all of the flow-tools programs.	$flow_bin_directory = "/usr/bin";
$silk_bin_directory	This directory contains all of the SiLK programs.	$silk_bin_directory = "/usr/local/bin";
$sensor_config_file	This is the full filename of the sensor.conf file.	$sensor_config_file = "/data/flows/sensor.conf";
$site_config_file	This is the full filename of the silk.conf file. Note that if this parameter is left blank (i.e., $site_config_file = "";) the SiLK commands issued by FlowViewer will look for the silk.conf file in the Data Rootdir as specified by (the default) $silk_rootdir variable, or as specified within the "Data Rootdir" text input field on the input screens. This latter case permits the user to place different silk.conf files in different root directories if desired.	$site_config_file = "/data/flows/silk.conf";
$silk_compiled_localtime	Indicates whether you compiled SiLK with 'enable-localtime' switch. By default SiLK stores data in UTC time. If you set this switch when you compile SiLK, it will store data in the system localtime zone and FlowViewer will adapt. ***See Section 2.2.3 above.***	$silk_compiled_localtime = "";
$rrdtool_bin_directory	This directory holds the rrdtool binary.	$rrdtool_bin_directory = "/usr/local/rrdtool-1.4/bin";
$version	Simply the current FlowViewer version number, used to differentiate between versions.	$version = "4.0";
$left_title	Text for the title that appears on the left side of the top frame in the user interface.	$left_title = "SWAN Network Operations Center";
$left_title_link	A hyper-link for quick linking to another web site.	$left_title_link = "http://swan.net/noc/";
$right_title	Text for the title that appears on the right side of the top frame in the user interface.	$right_title = "SWAN Network Device Bandwidth Site";
$right_title_link	A hyper-link for quick linking to another web site.	$right_title_link = "http://swan.net/noc/bandwidth";
$use_bottom_pulldowns	If over time a user creates a lot of FlowMonitors, and saves a lot of files, setting this variable to "N" will prevent pulldowns for those files occurring for a second time in the bottom frame of the display. This keeps the user interface HTML files smaller and speeds page loading somewhat.	$use_bottom_pulldowns = "Y";

@devices (flow-tools only)	This array holds a list of all of the different devices you are collecting netflow data from. FlowViewer can use a flow-tools data directory layout that has a particular device at the top. A typical flow-tools directory looks like: /flows/router_1/2012/2012-07/2012-07-04 The device name (router_1) is obtained from this array. Populate this array with your device names. If your flow-data file structure does not include a device name, for example you are collecting only from one device, set the @devices array to empty (i.e., @devices = ("");) and set: $no_devices_or_exporters = "Y"; Note that version 3.3 introduced the "Exporter" option which allows users to collect all devices on a single port and separate them via EXPORTER_ID. If you are taking the "Exporter" approach exclusively (i.e., you are not also using devices as described here) you may comment out this parameter. See next parameter **Note**: *If you add a device, you must restart any active FlowMonitor_Collector and FlowMonitor_Grapher jobs running in the background.*	@devices = ("router_1","router_2","router_3");
@exporters (flow-tools only)	If you are collecting from all of your devices onto a single flow-capture port, you may use $exporter[n] to separate the data. If so, uncomment this parameter. Each entry in this array is formed like this: exporter_ip_address : exporter_name On the FlowViewer input screens you will then see a pulldown with each exporter defined by exporter name. Internal searches will be based on the associated IP address.	@exporters = ("192.168.100.1:New York Router","192.168.100.2:Prague Router");
@ipfix_devices	This array holds a list of all of the different SiLK devices (sensors) you are collecting netflow data from. FlowViewer uses a SiLK data directory layout that has the particular device at the top. An example SiLK directory looks like: /flows/silk_router_1/in/2012/07/04 Note that SiLK contains subdirectories between the device_name and the daily directories (e.g., in, inweb, int2int, out, outweb, ext2ext) The SiLK device name (silk_router_1) is obtained from this array. Populate this array with your device names and create the directories with the same names. Note that version 4.4 introduces greater flexibility	@ipfix_devices = ("Router_v9_1","Router_v9_2","Test_6509_v9");

	for interfacing with SiLK environments that don't necessarily resemble the flow-tools directory configuration. This flexibility is accomplished through new SiLK Selection Parameter fields on the input screens together with their defaults as specified in the FlowViewer_Configuration.pm file. (See below.) **Note**: *If you add a device, you must restart any active FlowMonitor_Collector and FlowMonitor_Grapher jobs running in the background.*	
@ipfix_storage	Array holds storage limits per SiLK device. The FlowViewer_CleanSiLK script will check this variable before examining and trimming the oldest data from SiLK device data stores to remain under the storage limit.	@ipfix_storage = ("ipfix_rtr1:15G,"ipfix_rtr2:20G")
$ipfix_default_device	All initial, blank forms will have this selected instead of "Select Device". SiLK users with a single sensor can optimize default input screens by setting the "silk_data_directory" to their SiLK Rootdir, and setting this field to "Site".	$ipfix_default_device = "Site";
$no_devices_or_exporters (flow-tools only)	You need to set this parameter to "Y" if you are using neither devices nor exporters, you are simply collecting data (probably from just one device) into one directory. If you have devices and/or exporters, this field should be left at "N".	$no_devices_or_exporters = "N";
$flow_capture_interval (flow-tools only)	This variable defines the length of time beyond your specified end_time up to which the script will continue to parse through the flow_data looking for flows that occurred during your specified time period, but were exported from the router after the time_period. Some long flows, with a lot of data, may not complete and be exported from the router until well after your specified end_time.	$flow_capture_interval = 35 * 60; # Continue to look for flows 35 minutes beyond
$flow_file_length (flow-tools only)	This parameter defines how long each of your flow data files is. This is set via *the flow-tools flow-capture* command and defaults to 15 minutes.	$flow_file_length = 15 * 60
$start_offset	This parameter specifies how far back before the current time to specify the start_time for your FlowViewer or FlowGrapher run. These are the default start and end times that appear on your filter input screens.	$start_offset = (90 * 60); # e.g., 90 minutes ago
$end_offset	This parameter specifies how far back before the current time to specify the end_time for your FlowViewer or FlowGrapher run. The example below and the one above will specify a one hour period occurring approximately 30 minutes ago.	$end_offset = (30 * 60); # e.g., 30 minutes ago
$use_even_hours	If set to "Y" this parameter will cause default start and end times of your report period to be set on the hour.	$use_even_hours = "Y";
$N (flow-tools only)	Flow-tools only. Different organizations store captured flow-tools netflow data differently according to the 'N' setting on the flow-capture	$N = 3;

	statement. However, there is a bug in the flow-tools documentation such that the default value is truly '3' and not '0' as indicated. The default has been set to $N = 3 to reflect the more common setting. The directory structure associated with $N = 3 is shown below: /var/flows/router_1/2006/2006-07/2006-07-04 Setting $N=0, would cause the data to accumulate into a single directory without any subdirectories for date organization.	
$silk_capture_buffer_pre	Number of minutes used to set the –start-date field for invocations of rwcount, etc. SiLK's rwflowpack place the flow record into the file associated with the start time. Thus in order to account for long flows that cross a time period examined, the rwfilter process must include older files.	$silk_capture_buffer_pre = (125 * 60)
$silk_capture_buffer_post	Number of minutes used to set the –end-date field for invocations of rwcount, etc.	$silk_capture_buffer_post = (5 * 60)
$silk_init_loadscheme	Selects the loadscheme (--load-scheme=) used when graphing or monitoring 'Flows Initiated' only (e.g., not used for "Bits" or "Packets"). Assigns the whole flow to just the time the flow initiated.	$silk_init_loadscheme = 1
$silk_active_loadscheme	Selects the loadscheme (--load-scheme=) used when graphing or monitoring 'Flows Active' only (e.g., not used for "Bits" or "Packets") Evenly pro-rates the flows across all buckets for which it is active.	$silk_active_loadscheme = 5
$silk_class_default	Default value of SiLK Class (--class=)for use when presenting the Input screens.	$silk_class_default = "”;
$silk_flowtype_default	Default value of SiLK Flowtype (--flowtype=)for use when presenting the Input screens.	$silk_flowtype_default = "”
$silk_type_default	Default value of SiLK Type (--type=)for use when presenting the Input screens.	$silk_type_default = "in, inweb";
$silk_sensors_default	Default value of SiLK Sensors (—sensors=) for use when presenting the Input screens.	$silk_sensors_default = "”;
$silk_switches_default	Additional user specified (field name and value) selection switches	$silk_switches_default = "”;
$use_NDBM	FlowViewer offers the ability to resolve *netflow* IP addresses into their host names on the fly. This process is speeded up by caching names into a 'names' file which resides in the directory specified by the 'names_directory' parameter. As you are building up your 'names' file with early runs, you will notice the speed increase dramatically as the 'names' file is used more. The process of resolving names is the primary reason for slower overall FlowViewer performance. You should preferably use the GDBM array database which is	$use_NDBM = "N";

	fastest. However, not all Perl distributions support GDBM but most do support NDBM. The '$use_NDBM' flag will cause the FlowViewer_Main.cgi and FlowGrapher_Main.cgi scripts to use NDBM.	
$pie_chart_default	The parameter defines which pie-chart option appears as the default on the Pie Chart pulldown on the FlowViewer input screen. The “With Others” option means that the Pie Chart will show an “Others” slice which includes “everything else”. The “Without Others” option will not show an “everything else” slice.	$pie_chart_default = 0; # 0 = None; 1 = With Others; 2 = Without Others
$number_slices	Defines the number of slices included in the Pie Chart.	$number_slices = 6;
$pie_colors	List containing color descriptors (from FlowGrapher_Colors) to be used when graphing Pie charts using the FlowViewer tool. Note the exact syntax used.	$pie_colors = ['pie2 color1','pie2 color2','pie2 color3','pie2 color4','pie2 color5','pie2 color6','pie2 color7','pie2 color8','pie2 color9','pie2 color10'];
$maximum_days	This parameter defines a maximum number of days for the length of user created FlowViewer reports and FlowGrapher graphs.	$maximum_days = 91;
$remove_workfiles_time	This parameter defines the age at which to remove intermediate files from the $flow_working directory when running the FlowViewer_CleanFile utility from *crontab*. (In seconds- the example shows 1 day.)	$remove_workfiles_time = 86400;
$remove_graphfiles_time	This parameter defines the age at which to remove intermediate files from the $graphs_directory directory when running the FlowViewer_CleanFile utility from *crontab.* (In seconds- the example shows 7 days.)	$remove_graphfiles_time = 7*86400;
$remove_reportfiles_time	This parameter defines the age at which to remove intermediate files from the $reports directory when running the FlowViewer_CleanFile utility from *crontab*. (In seconds- the example shows 7 days.)	$remove_reportfiles_time = 7*86400;
$time_zone_dst_offset	Number of seconds of the Daylight Savings adjustment in your time-zone. This is important if your system time is not GMT. The example shows a one-hour offset.	$time_zone_dst_offset = (60 * 60); # i.e., 1 hour
$date_format	Allows the user to enter dates and view dates in reports in their chosen format. Options include: Month/Day/Year (MDY) Ex.: 12/25/2013 Day/Month/Year (DMY) Ex.: 25/12/2013 Day.Month.Year (DMY2) Ex.: 25.12.2013 Year-Month-Day (YMD) Ex.: 2013-12-25	$date_format = “DMY2”;
$labels_in_titles	This parameter controls whether to display the Monitor title in the title of the graph itself. Setting this to “1” will include titles, setting it to “0” will not.	$labels_in_titles = "1";

$left_title	Title will appear at top left of all screens when using only one dashboard.	$left_title = "SWAN NOC Management System";
$left_title_link	This link will be visited when a user clicks on the title in the left top of all screens. Can be any HREF.	$left_title_link = "$cgi_bin_short/FV.cgi";
$right_title	Title will appear at top right of all screens when using only one dashboard.	$right_title = "Monitoring SWAN Network Data Flows";
$right_title_link	This link will be visited when a user clicks on the title in the right top of all screens. Can be any HREF.	$right_title_link = "$cgi_bin_short/FV.cgi";
$debug_viewer	This parameter, if set to “Y”, will turn on debugging for FlowViewer. The debug output can be found in $flow_working/DEBUG_VIEWER.	$debug_viewer = “Y”;
$debug_grapher	This parameter, if set to “Y”, will turn on debugging for FlowGrapher. The debug output can be found in $flow_working/DEBUG_GRAPHER.	$debug_grapher = “Y”;
$debug_monitor	This parameter, if set to “Y”, will turn on debugging for FlowMonitor. The debug output can be found in $flow_working/DEBUG_MONITOR.	$debug_monitor = “Y”;
$debug_group	This parameter, if set to “Y”, will turn on debugging for FlowMonitor_Group. The debug output can be found in $flow_working/DEBUG_GROUP.	$debug_group = “Y”;
$debug_files	This parameter controls whether to save intermediate files for debugging purposes. A value of “Y” will leave the files around for inspection. This defaults to “N”.	$debug_files = “N”;
$log_directory	The location for the logging output files. Some of the logging files, when set to full logging, can get big. Also, if you want the files around for a while, don’t place them in a directory that will get cleaned by one of the FlowViewer_Clean scripts.	$log_directory = "/var/www/cgi-bin/FlowViewer_4.0/log";
$log_collector_short	Provides for a minimal amount of logging for FlowMonitor_Collector. A timer is printed which tells how long it has taken to collect the data. This might be useful if you have a lot of Monitors and you want to see if they are still being completed in a timely manner.	$log_collector_short= "Y";
$log_collector_med	Provides for a medium amount of logging for FlowMonitor_Collector. A timer is printed which tells how long it has taken to collect the data. This might be useful if you have a lot of Monitors and you want to see if they are still being completed in a timely manner.	$log_collector_med= "N";
$log_collector_long	Provides for a full amount of logging for FlowMonitor_Collector. This includes collected data for each active Monitor. A timer is printed which tells how long it has taken to collect the data. This might be useful if you have a lot of Monitors and you want to see if they are still being completed in a timely manner.	$log_collector_long= "N";

$log_grapher_short	Provides for a medium amount of logging for FlowMonitor_Grapher. The logs have timers showing how long it takes to complete the graphs (e.g., usually under 1 second per Monitor).	$log_grapher_short= "Y";
$log_grapher_long	Provides for a full amount of logging for FlowMonitor_Grapher. This includes graph data for each active Monitor. The logs have timers showing how long it takes to complete the graphs (e.g., usually under 1 second per Monitor).	$log_grapher_long= "N";
$collection_offset	Defines how many minutes into the past you want to use to collect data. At 1800 (30 minutes) this will cause FlowMonitor_Collector to examine a period 30 minutes in the past. This is useful for allowing all flows that may have crossed that period to be exported from the device. Some flows can last 30 minutes or more and will be excluded for consideration if they haven't been exported yet.	$collection_offset = 1800;
$collection_period	This parameter controls how often data is collected for Monitors by FlowMonitor_Collector. It is probably a good idea to leave this at its default value.	$collection_period = 300;
$graphing_period	Frequency at which FlowMonitor_Grapher is executed to generate new Monitor graphs.	$graphing_period = 300;
$use_existing_concats (flow-tools only)	DEPRECATED (now used always) When set to "Y" this parameter will cause FlowMonitor_Collector to re-use concatenated flow-tools files for different Monitors that are based on the same device. This dramatically speeds things up.	$use_existing_concats = "Y";
$use_existing_prefilters (SiLK only)	DEPRECATED (now used always) When set to "Y" this parameter will cause FlowMonitor_Collector to use the file created by rwfilter INPUT for each subsequent FlowMonitor that has the same sensor/class pair. This dramatically speeds things up.	$use_existing_prefilters = "Y";
$recreate_cat_length	Sets the length of the concatenation period during a FlowMonitor_Recreate. The concatenation is performed and then data is extracted according to the filter for each 5-minute period in the concatenation. Used also for SiLK and informs SiLK how many 1 hour files to perform rwfilter on before completing the 5-minute period extractions.	$recreate_cat_length = 6*(60*60);
$rrd_dir_perms	Controls the UNIX permissions applied to directories of the type defined by the parameter.	$rrd_dir_perms = 0775;
$filter_dir_perms	Controls the UNIX permissions applied to directories of the type defined by the parameter.	$filter_dir_perms = 0775;
$work_dir_perms	Controls the UNIX permissions applied to directories of the type defined by the parameter.	$work_dir_perms = 0775;

$html_dir_perms	Controls the UNIX permissions applied to directories of the type defined by the parameter.	$html_dir_perms = 0775;
$html_file_perms	Controls the UNIX permissions applied to files of the type defined by the parameter.	$html_file_perms = 0775;
$graph_file_perms	Controls the UNIX permissions applied to files of the type defined by the parameter.	$graph_file_perms = 0775;
$rrd_file_perms	Controls the UNIX permissions applied to files of the type defined by the parameter.	$rrd_file_perms = 0775;
$filter_file_perms	Controls the UNIX permissions applied to files of the type defined by the parameter.	$filter_file_perms = 0775;
$analyze_count	Number of hosts or ports that will be made available for Analysis graphs and statistics.	$analyze_count = 5;
$analyze_peak_width	Number of observations to examine for peaks (per period.) Each $analyze_peak_width observations are examined to see which hosts or ports dominated those flows. The highest $analyze_count number of hosts or ports are maintained and presented in the graph and statistics.	$analyze_peak_width = 1000;
$analyze_colors	List containing color descriptors (from FlowGrapher_Colors) to be used when graphing Analysis graphs when using the FlowGrapher tool. Note the exact syntax used.	$analyze_colors = ['gray95','pale green','pale brown','pale red','pale blue','pale yellow'];
$monitor_file_perms	Controls the UNIX permissions applied to files of the type defined by the parameter.	$monitor_file_perms = 0775;
$saved_filters_perms	Controls the UNIX permissions applied to files of the type defined by the parameter.	$saved_filters_perms= 0775;
$rrd_slope_mode	Controls the effect at the top of the FlowMonitor graphs. The tops can be either 'square' or 'sloped'. Default = ""; (square)	$rrd_slope_mode = "--slope-mode";
$sigma_type_1 $sigma_type_2 $sigma_type_3	Allows the user to configure FlowMonitor thresholds for email alerts that are based on statistical analysis of FlowMonitor observations. Looking at the example to the right, the first part is the number of previous observations considered (12) and the second part is the multiplier. In this case an alert is generated if the current FlowMonitor observation is greater than the average plus 2.67 times the standard deviation as derived from the last 12 observations. There are three options the user can construct; each will be available on the 'Alert Frequency' pulldown.	$sigma_type_1 = "12:2.67" This means: "Create an average out of the most recent 12 observations and alert the user if the current observation is more than 2.67 sigma higher than the average."
$dscan_parameters	Permits the user to modify the flow-dscan command that identifies scanners. See flow-tool's *flow-dscan* documentation. Note that FlowViewer will "touch" two files in $cgi_bin_directory that flow-dscan requires: dscan.suppress.dst, and dscan.suppress.src. These	$dscan_parameters = "-w –W";

	files permit the user to specify hosts and ports that should be exempted when determining whether a host is scanning or not. They can be left empty.	
$scan_model	Selects the scan detection model for SiLK Detect Scanning reports. 0=TRW&BLR; 1=TRW only; 2=BLR only	$scan_model = 2;
$trw_internal_set	Used for SiLK Detect Scanning reports that invoke the TRW model. The parameter identifies the file that contains internal IP addresses.	$trw_internal_set = "/tmp/trw_internal_ips
$dns_column_width	Allows for variable widths for source and destination address columns in reports.	$dns_column_width = 40;
$detail_lines	Controls the default value of printed lines in FlowGrapher reports. Will find the largest "detail_lines" flows that occur in the period. A new capability is added in version 4.4 that permits a negative number which will find the "smallest" flows in terms of bytes.	$detail_lines = 200;
$asn_width	Controls the name length for AS Names printed out for the Detect Scanning report.	$asn_width = 60;
$default_identifier	Controls the default presentation of host identifiers (either by IP address, or resolved DNS hostname.)	$default_identifier = "DNS";
$sip_prefix_length	For Prefix Aggregation reports, specifies the number of most significant bits of the source address to keep.	$sip_prefix_length = "16";
$dip_prefix_length	For Prefix Aggregation reports, specifies the number of most significant bits of the destination address to keep.	$dip_prefix_length = "16";
$default_report	Allows the user to identify a particular FlowViewer report that will be pre-selected. This enables quicker execution after switching tools. See section 4 for the numbers to use to set this parameter.	$default_report = 10;
$default_graph	Allows the user to specify a particular graph type that will thereafter be pre-selected. Choose from: • "bps" (Bits/Second) • "pps" (Packets/Second) • "fpsi" (Flows Initiated/Second) • "fpsa" (Flows Active/Second)	$default_graph = "bps";
$dig	This parameter points to the location of DNS utility 'dig' (set this to nslookup if required.) The parameter is set to do inverse DNS lookups, hence the –x in the example.	$dig = "/usr/bin/dig +time=1 -x ";
$dig_forward	This parameter points to the location of DNS utility 'dig' (set this to nslookup if required.) The parameter is set to do forward DNS lookups.	$dig_forward = "/usr/bin/dig +time=1 +tries=1 ";

3.2 Browsers

Various browsers interpret the FlowViewer.css cascading style sheet differently. This shows up in minor differences in rendering objects (e.g., buttons) and in the extent to which a browser can reduce the size of the user interface. The following table documents the screen reduction percentages that are workable (not “Zoom text only”.)

Browser	100%	90%	80%	75%	70%	67%	60%	50%	40%	33%	30%	25%	20%
Firefox	Y			Y		Y		Y					
Opera	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Safari	Y			Y		Y							
Chrome	Y	Y		Y		Y							
Internet Explorer	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		

An important browser differentiator is the ability to ignore the warning to ‘Confirm Form Resubmission”. Recent versions of Chrome and Opera do not have this option and require the user to manually resubmit forms thus preventing ‘backing up’ or copying tabs without resubmitting the form. With FlowViewer this is very inconvenient as many submissions take several seconds or longer to process and having to repeat them needlessly is quite burdensome.

For best effect, the browser option “Allow pages to choose their own color instead of me” should be selected.

Version 4.5 introduces “$use_bottom_pulldowns” a configuration variable used to control whether the FlowMonitor and Saved Reports pulldowns are repeated at the bottom of the user interface. Once a user accumulates a lot of FlowMonitors and Saved Reports, the HTML becomes a bit hefty and setting this variable to “N” speeds things up somewhat. I have noticed that I have very rarely used the pulldowns on the bottom frame.

4. FlowViewer Operation and Usage

The FlowViewer specific tool consists of two parts: FlowViewer.cgi, and FlowViewer_Main.cgi. The user invokes FlowViewer by clicking on one of the FlowViewer buttons on the Main screen or in the top or bottom frames. Once the user has clicked on the Generate Textual Report button, the FlowViewer_Main.cgi script is invoked which runs several flow-tools or SiLK commands to generate the data for the report.

The execution of the flow-tools is as follows:

flow-cat
flow-nfilter
flow-print (or)
flow-stat (or)
flow-dscan

The execution of the SiLK is as follows:

rwfilter
rwstats (or)
rwsort (or)
rwcut (or)
rwnetmask (or)
rwscan

The parameters for these commands are derived from the user's input, including filtering criteria and report selection. For flow-tools, the filtering criteria are collected and used to create a flow-filter file which is provided to *flow-nfilter*. The script captures the output from either *flow-stat* or *flow-print* (for flow-tools) or *rwstats* or *rwscan* (for SiLK) and formats it for web-page output.

The FlowViewer input screen (created by FlowViewer.cgi) is shown (for an IPFIX, SiLK device) in Figure 4-1 below. Notice also the three different Dashboard options available as active links at the top for EISOC, Net Ops, and Performance, with the Performance Dashboard active.

Figure 4-1 - FlowViewer Input Screen

The user will complete input fields as necessary to define a filter for viewing stored netflow data. Users may select from a previously defined filter. Filters are saved for future reference after a report has been produced by clicking on the “Save Filter” link that will appear in the bottom frame after a report has been generated.

Several new fields have been added with version 4.4 to accommodate more experienced SiLK users. The new fields should allow these users to now use FlowViewer with environments that are more complex than the straightforward, flow-tools mimicking, environments earlier versions of FlowViewer were chained to.

FlowViewer will accept unlimited entries for each field separated by commas. However, flow-tools has some limits (e.g., won’t accept 500 individually specified ports, but will accept ranges, e.g., 50000-50500.) Fields may be preceded by a dash or minus sign (-), which will cause the script to ignore such flows. For example, providing the value -1776 to the Source AS field will eliminate from the report any flows that originated in AS 1776.

The “Device Name” field allows the user to select from a collection of devices that he may be collecting netflow from (‘@devices field in the FlowViewer_Configuration.pm file.) As of version 3.3, the user is also able to differentiate netflow data from different exporters based on the ‘Exporter ID’ field. This is used in the situation where the user is collecting from multiple devices onto the same *flow-capture* port. Version 4.0 introduces a new configuration parameter called @ipfix_devices which will invoke SiLK processing when one of the devices from the array is selected.

If you are not using any devices or exporters, you will have to set: $no_devices_or_exporters = "Y"; In this case, no device or exporter pulldown will appear.

The Source and Destination IP fields can accept individual IP addresses, network base and range (e.g., 192.169.100.0/24), or fully qualified domain names (e.g., www.abccompany.com.) The user may exclude a sub-network from a specified network by following the network with a negated sub-network. An example is shown:

Source Address: 192.168.100.0/24, 192.168.200.0/24, -192.168.200.64/27

The example removes from consideration any hosts in the range 192.168.200.64 – 192.168.200.95, but permits all others in 192.168.200.0/24 to pass.

With version 4.6 SiLK users can provide IPset names into the Source and Destination IP Address fields. The IPset file name must have the '.set' suffix. The $ipset_directory configuration parameter tells FlowViewer where to look for the IPset files.

The Source Port and Destination Port entry fields will accept a range value. By accident it has been discovered that flow-tools indeed has an (undocumented) port range capability. Simply use a dash (e.g., 50000-60000; a preceding dash, e.g., -1024-65535, will exclude that range.) The previous work-around (i.e., a range value created by separating the range end values with a colon (e.g., 40100:40200) will continue to be supported. SiLK also accepts port ranges.

The Source and Destination Interface values expect the SNMP index value for the device's interfaces. Note that these can change over time (e.g., when a new interface card is added to the device.) For FlowMonitor this becomes important. If an interface index value should change for an active Monitor, use the 'Revise' option on the Manage All Active and Archived FlowMonitors page (see below) to modify the filter. This will maintain the integrity of the FlowMonitor. FlowViewer offers the option to use Named Interfaces. These must be configured in advance in either the NamedInterfaces_Devices, or NamedInterfaces_Exporters files. Only one interface is available per report via the NamedInterfaces pulldown; however this may be combined with numeric values in the numeric Interfaces text box to filter on multiple interfaces.

The TCP Flags filter criteria is specified by first identifying the TCP flags one is interested in and then identifying a mask which will specify whether the flag is set or unset. Flow-tools expects an integer between 0 and 256, (e.g., 0x2/0x2). Flow-print (used to generate reports that include this field, e.g., the '132 Column' report) treats the flags as 'the cumulative or of the TCP control bits'. In other words, for every packet in the flow examined, if the flag is set, it will contribute to this value. Thus the value of '6' would indicate that one packet had a SYN bit set (2) and one had the RST (4) flag set.

For SiLK, the flags are represented internally as: F=FIN; S=SYN; R=RST; P=PSH; A=ACK; U=URG; E=ECE; C=CWR. Use the approach of specifying the flags and the mask via hex (e.g., 0x0A/0xAA which will yield the rwfilter parameter: --flags-all=/C,/U,P/P,S/S, the C and U flags must be unset, and the P and S flags must be set.)

```
SiLK Flags Denominators:     CEUAPRSF
 Flags Requested (0x0A):     00001010
  Mask Requested (0xAA):     10101010
```

```
SiLK rwfilter parameter:     --flags-all=/C,/U,P/P,S/S
```

If the selected device is a SiLK collecting device, the user can select which of the data types he would like to search from (e.g., in, inweb, int2int, out, outweb, ext2ext, or all). The screen will default to "all." The user can shorten the length of processing time by limiting the input data types.

After completing the Filter Criteria, the user selects either a Statistics Report, or a Printed Report. SiLK adds IPv6 processing and several associated Printed reports. Some reports are only available to flow-tools devices and some only to SiLK devices (alert messages will be presented.) The names of the reports are preserved from flow-tools, with SiLK reports created to match. FlowViewer provides the same output that flow-tools and SiLK tools report from the command line. The current reports that are available include:

Statistical Reports:

Name	Number used to set default
Summary	99
Detect Scanning	30
UDP/TCP Destination Port	5
UDP/TCP Source Port	6
UDP/TCP Port	7
Destination IP	8
Source IP	9
Source/Destination IP	10
Source or Destination IP	11
IP Protocol	12
Input Interface	17
Output Interface	18
Source AS	19
Destination AS	20
Source/Destination AS	21
IP ToS	22
Input/Output Interface	23
Source Prefix	24
Destination Prefix	25
Source/Destination Prefix	26
Exporter IP	27

Printed Reports (cannot set default for these reports):

Flow Times	(Note: flow_times will be printed according to $date_format)
AS Numbers	
132 Columns	(Note: flow_times will be printed according to $date_format)
1 Line with Tags	
AS Aggregation	
Protocol Port Aggregation	
Source Prefix Aggregation	

Destination Prefix Aggregation
Prefix Aggregation
Source Prefix Aggregation v6
Destination Prefix Aggregation v6
Prefix Aggregation v6
Full (Catalyst)

The user can select a default Statistics report which will be pre-selected when the FlowViewer screen is presented. The $default_report value is manually set accordingly in FlowViewer_Configuration.pm file. This value is selected using the numbers shown for each report above.

The “Include Flow If” parameter allows the user several options for controlling which flows are included in the report. Because flows do not completely lie within a specified period, the user has the option to define the conditions for including the flow. These include:

Any Part in Specified Time Span
End Time in Specified Time Span
Start Time in Specified Time Span
Entirely in Specified Time Span

The “Sort Field” parameter controls the ordering of the report based on which reported value has been selected.

The user has the option to view FlowViewer results in text form together with a pie-chart representation of the data for the Statistics Reports. The user would select a particular “Pie Charts” option.

The “Cutoff Lines” parameter controls how many lines will be printed. The first ‘cutoff_lines’ are printed. The ’Cutoff Octets’ parameter controls the point at which to end the report based on the number of octets displayed in the output line. No additional lines will be printed which contain an Octets value less than ‘cutoff octets’. The “Oct Conv.” option if selected will display octets in a shorthand notation (e.g., 10.3 MB instead of 10300000.)

The “Sampling Multiplier” field can be set to a value greater than one whereby the data in all reports and graphs will be multiplied by this number. This field is used to give an approximation of real traffic flow levels for devices that export sampled netflow data. The sampling multiplier is not available for some of the Printed Reports.

The “Resolve Addresses” parameter informs the script whether or not to resolve IP addresses into their full host names. Resolving addresses is a little slow the first time through, but builds up a cache as the number of runs increases and soon becomes as fast as not resolving addresses.

A ‘Flow Analysis’ switch (On, Off) has been added to provide additional information with respect to flows (e.g., octets/flow, etc.) which has proved useful for security analysis.

A typical FlowViewer report (in this case Prefix Aggregation v6) is shown in figure 2 below:

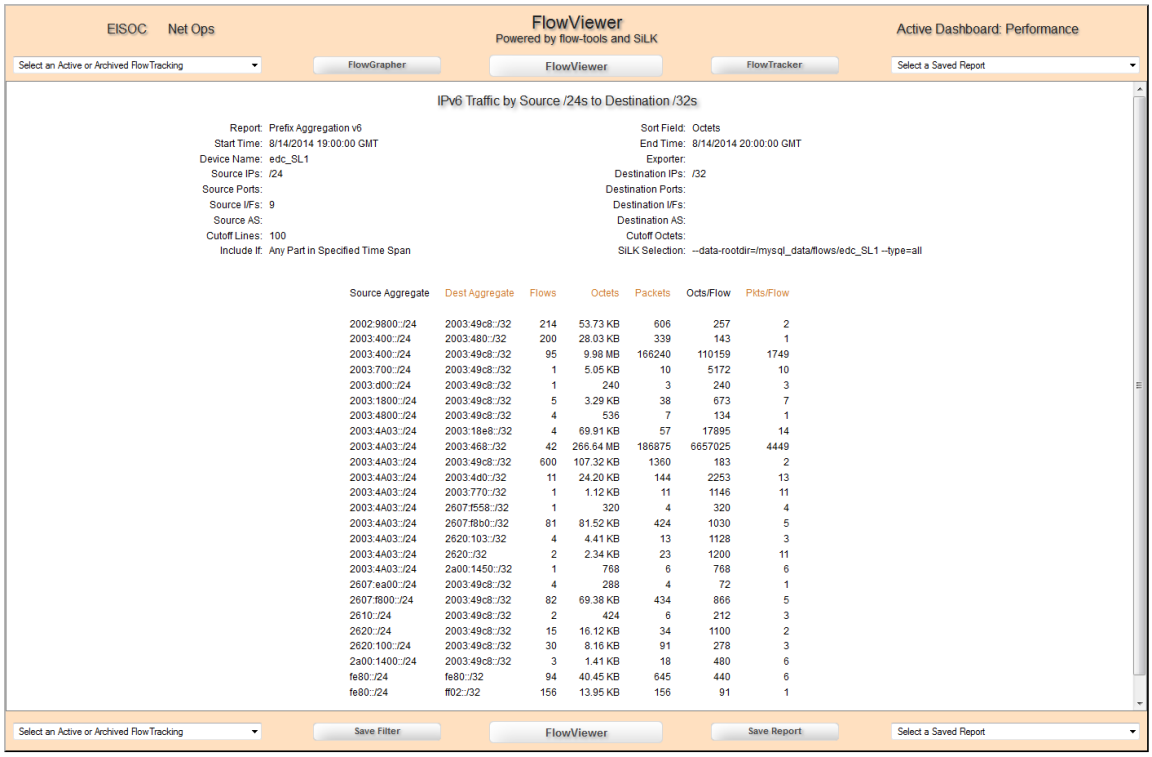

Figure 2 - FlowViewer Report

The user can sort the report by clicking on any of the column headers. From the FlowViewer Report output page the user can switch between this report and any of the three tool input screens simply by clicking on one of the other tool's buttons in the top or bottom frame. The existing filter parameters are saved and the new input screen is pre-filled.

The user can 'Save Report' or 'Save Filter' via links in the bottom frame. Filters can be saved for future retrieval by any of the tools. Reports are saved by clicking on the "Save Report" button. The user provides a meaningful name to the report so that it can be readily identified later.

The user can review a Saved Report by selecting it from the Saved Reports pulldown to the right in either the top or bottom frames. The option for Manage All Saved Reports will provide a screen for managing (e.g., "Removing") saved reports (Figure 3.) The Manage option will produce a screen similar to the Manage All FlowMonitors screen in Figure 9 below.

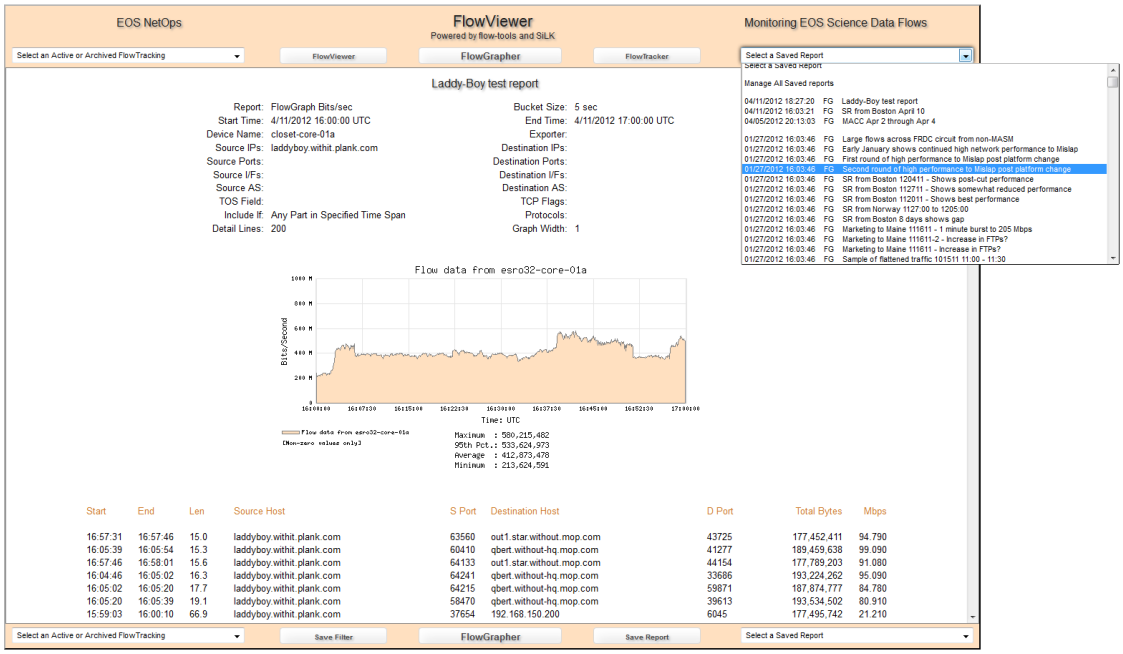

Figure 3 – Selecting a Previously Saved Report for Viewing

A new Detect Scanning report has been added to take advantage of the existing scan detection tools in flow-tools and SiLK. The outputs are slightly different. A typical flow-tools Detect Scanning report is shown in figure 4 below. Clicking on the IP Address link will create a FlowGrapher report for that host. See the section 3.1 discussion of $dscan_parameters and $scan_model for more information on configuring scans.

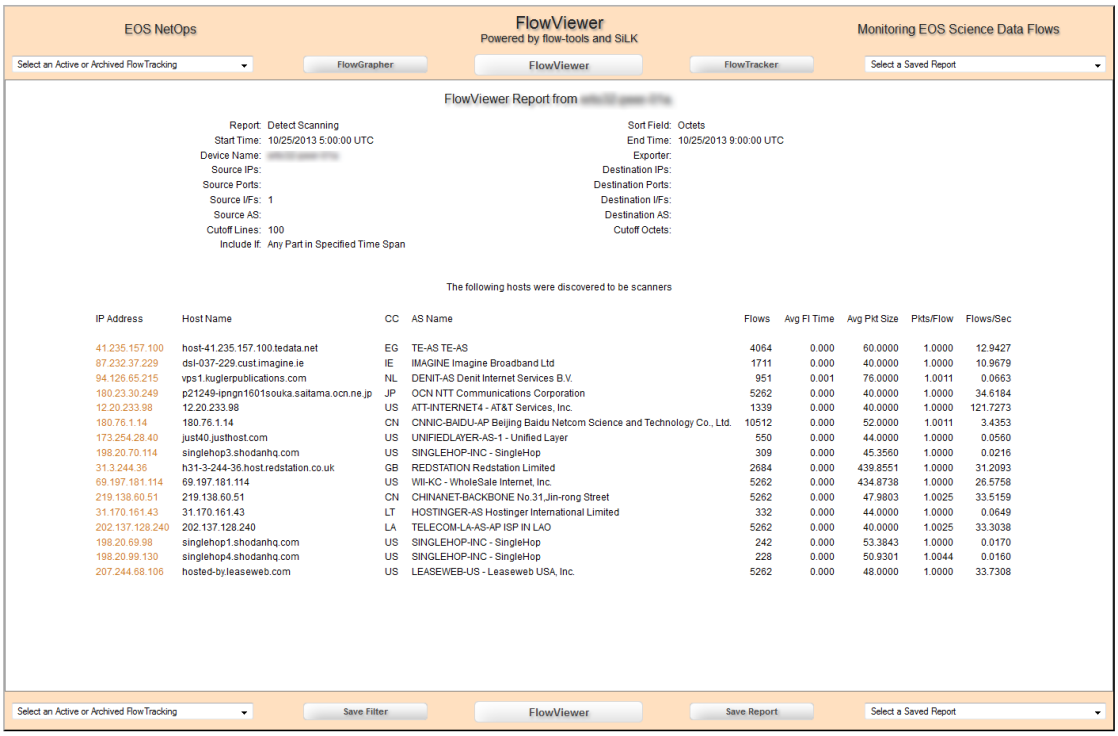

Figure 4 – Detect Scanning Report

FlowViewer Tips

- The 132 Column Printed Report option is very useful for understanding individual flows through your network. The report provides source and destination information and interface and port information in the same output. This length of this report is constrained primarily by the 'Cutoff Lines' parameter, but is not slowed down by large values

- The Input and Output interfaces are represented by the SNMP index assigned to each interface by the device. Named interfaces are available if the NamedInterface files have been pre-configured for the device or exporter selected.

- The nature of the way RRDtool consolidates data may sometimes make it desirable to save an existing FlowMonitor page to preserve detail that may vanish soon due to consolidation.

5. FlowGrapher Operation and Usage

FlowGrapher consists of two parts: FlowGrapher.cgi, and FlowGrapher_Main.cgi. The user invokes FlowGrapher by clicking on the FlowGrapher button on the Main page or from the top or bottom frames. This approach is different from earlier versions and provides the added benefit of updating input date and time periods automatically. Once the user has clicked on the Generate Graph button, the FlowGrapher_Main.cgi script is invoked which runs several flow-tools or SiLK commands to generate the report.

The execution of the flow-tools is as follows:

flow-cat
flow-nfilter
flow-report (or)
flow-print (for prorated)

The execution of the SiLK is as follows:

rwfilter
rwcount (or)
rwcut (or)
rwsort

The parameters for each of these commands are derived from the user's input, primarily the filtering criteria. For flow-tools, the filtering criteria are collected and a used to create a flow-filter file which is provided to *flow-nfilter*. The script captures the output from the *flow-report* Linear option and parses it to build the graph. For the Flows Active or Analysis options FlowGrapher will parse through *flow-print's* 132 column option. For SiLK, the processing uses *rwcount* to map flows to buckets, and *rwsort* to find the top $detail_lines flows for printing in the text section below the graph.

The script builds an array of times and values depending on the "Bucket Size" parameter (in seconds.) This parameter defines the width of the 'buckets' into which segments of flow-data is accumulated. Note a new option, 300E, will map the value to the End time of the bucket to match the way FlowMonitor assigns the value. When all of the 132-column output is parsed (for flow-tools), or the *rwcount* report returns (for SiLK), an array (of flow amounts in time buckets) is created and provided to Lincoln Stein's GD::Graph software to produce the graph.

The FlowGrapher input screen (FlowGrapher.cgi) is shown in Figure 5-1 below. This environment depicted here has only one Dashboard available. The user will complete input fields as necessary to define a filter for limiting the netflow data. The filtering criteria are identical to those from FlowViewer described above.

The "Detail Lines" parameter controls how many lines of flow detail information will be printed. FlowGrapher will select the largest '$detail_lines' number of flows to present below the graph. The "Graph Width" parameter is used to scale the resulting graph image. This is useful

sometimes for viewing detailed graphs. The “Bucket Size” parameter controls the time bucket into which flows are parsed to build up the graph. The default is 5 seconds. The “Resolve Addresses” and “Include Flow If” parameters are the same as with FlowViewer, and are described above.

Figure 5-1 - FlowGrapher Input Screen

The “Graph Types” option allows the user to graph octets (Bits/Second), packets (Packets/Second), or Flows (Flow Initiated/Second or Flows Active/Second). For flow-tools, with the exception of Flows Active or Analysis options, this will engage the flow-report “linear-interpolated-flows-octets-packets" option (Linear method) which is the default as of version 4.2.1. This can speed up FlowGrapher reports sometimes up to 4X. Or, for flow-tools, the user may select one of “Bits/Second - Analysis”, “Packets/Second - Analysis”, “Flow Initiated/Second - Analysis” or “Flow Active/Second - Analysis” which will engage the older “prorated” option (see discussion below.) This method is required to initiate the new Analysis tools. The differences in completion times between the two methods become starker with longer examined time periods.

There are now two Analysis methods for looking at Flows. The Flows Active/Second option will add a whole flow to each bucket during which the flow is active, prorating the first and last buckets as necessary. This gives the user a look at the number of flows active at a time. The Flows Initiated/Second option will map the flow to the bucket that contains its start time. FlowGrapher will then gather flow-tools one-second buckets into the user specified bucket size. SiLK FlowGrapher processing for Flows/second offers the user the ability to look at flows in a variety of ways as controlled by the *$silk_init_loadscheme and $silk_active_loadscheme* configurable parameters.

The “Sampling Multiplier” field allows the user to expand the graphed output in compliance with the sampling rate for sampled netflow data in order to simulate actual traffic flows.

A typical FlowGrapher report is shown in Figure 5-2 below:

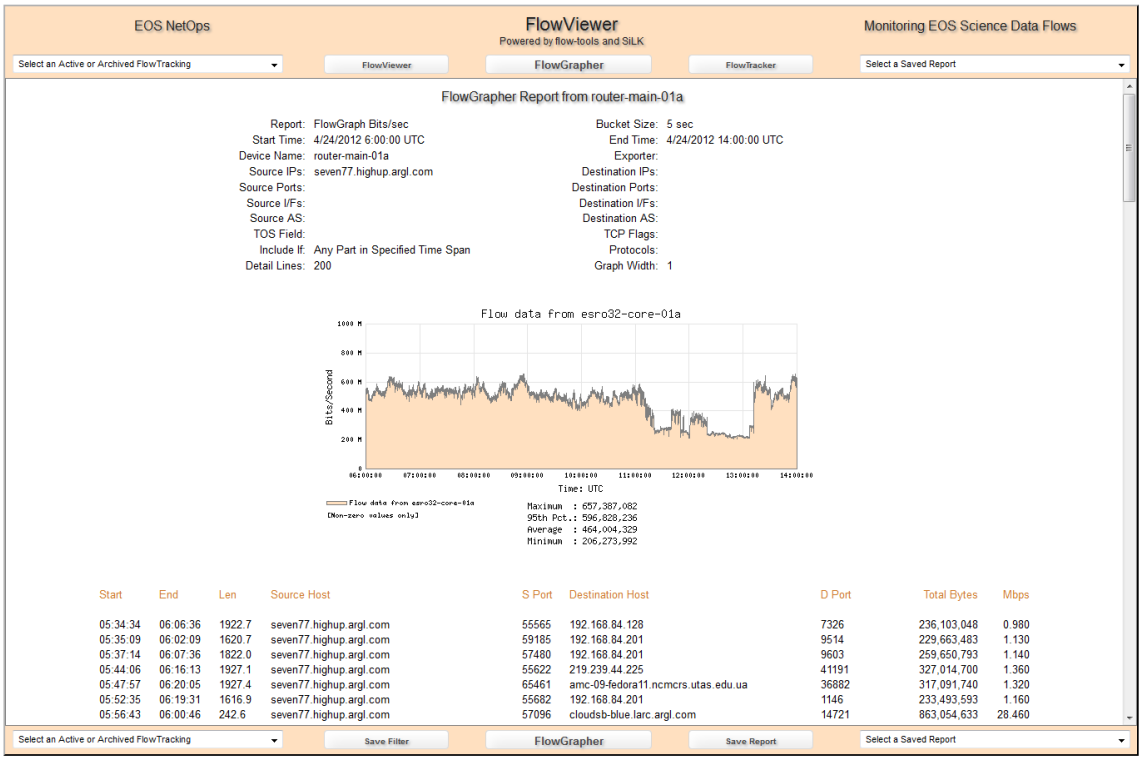

Figure 5-2 - FlowGrapher Report

FlowGrapher displays statistical information about the data flows for the time period graphed. The information includes the maximum, minimum, average, and 95th percentile values of those data points plotted. The text data section of the report is sortable by column header.

From the FlowGrapher report output page the user can ‘Save Report’ or ‘Save Filter’ via links in the bottom frame. Filters can now be saved for future retrieval by either of FlowViewer, FlowGrapher, or FlowMonitor. The user provides a meaningful name to the filter or report so that it can be more readily identified later. The user is provided options to manage saved reports on the Manage Saved Reports pulldown option.

FlowGrapher Tips

- FlowGrapher completion speed is only modestly effected by the ‘Detail Lines’ input variable. FlowGrapher will select the largest ‘Detail Lines’ number of flows to display. For example the largest 100 flows.
- Varying the ‘Sample Time’ parameter (which effects the size of the bucket into which flows are parsed) does not have a significant impact on report completion speed.

Discussion of the difference between the Linear and the Prorated Methods

FlowViewer 4.2 introduced the use of a heretofore un-noticed flow-tools capability, known as the flow-report "linear-interpolated-flows-octets-packets" option. The use of this option has enabled significant performance improvements in generating FlowGrapher reports. As of version 4.2.1, this capability has been extended to FlowMonitor with significant decreases in FlowMonitor generation times for recreated FlowMonitors as well as improvements in FlowMonitor_Collector completion times which enables more FlowMonitors to complete in each collection cycle (at NASA we collect for over 230 FlowMonitors in a little over 30 seconds.)

The new method is significantly faster and the results are very close to the old method so it is now the only method for all but the Flows Active and Analysis options. The FlowGrapher detail lines are slightly different (aggregated by flows,) and the statistics are slightly different particularly with the Maximum. The previous method (i.e., now selected via an analysis Graph Type: Bits/Second – Analysis, etc.) will include only the fractional portion of the total that actually occurs in the first and last buckets using the actual start and end times. The new method divides a single flow into equal pieces (total flow bytes divided by number of seconds) and places the subdivided portion into each bucket that the flow crosses in time. The total flow bytes are equally accounted for, however depending on the alignment of flows, the two methods can produce slightly different statistical values as noted on the graph. A comparison of Averages between the methods is usually very close. Figure 5-3 below attempts to shed some light on this phenomenon.

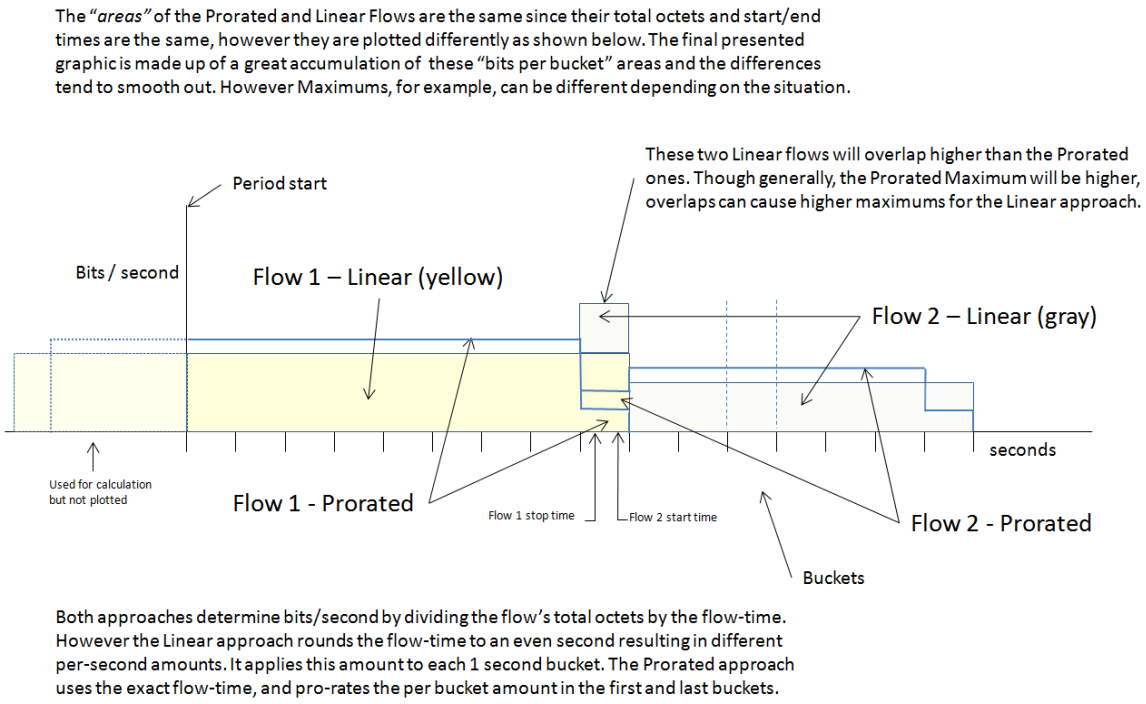

Figure 5-3 – Comparing the Linear and Prorated methods of graphing

Note that when graphing *flows* themselves, Graph Type "Flows Active/Second – Analysis" takes on a different meaning. In this case a single flow is mapped whole into each bucket through which it is active (with first and last buckets pro-rated.)

FlowGrapher Analysis

Version 4.4 introduces a new capability called FlowGrapher Analysis which enables the user to analyze FlowGraphs by examining the more prominent subsets of the traffic visually. This is particularly useful in isolating source and destination endpoints of peak flow periods, thus helping to identify possible Denial of Service (DOS) attacks or preparations. New options have been added to the 'Graph Type' pulldown on the FlowGrapher input screen. The new options are:

- Bits/Sec – Analysis — Analyze by largest users in bits/second
- Packets/Sec - Analysis — Analyze by largest users in packets/second
- Flows Initiated/Sec – Analysis — Analyze by largest flows initiated per second
- Flows Active/Sec – Analysis — Analyze by largest concurrent flows per second

The figures below demonstrate how this feature can be used. The first (Figure 5-4) shows a FlowMonitor that is monitoring incoming 'Flows/Sec'. The eclipsing of a previously configured threshold (a new 'Sigma' threshold, see section 6) has generated an alert email for the anomalies starting with the 19:15 to 19:20 value.

Figure 5-4 – FlowMonitor depicting anomalous Flows/second behavior

Selecting the FlowGrapher tool with a tighter time frame and the 'Flows Initiated/Second – Analysis' graph type selected isolates the anomalies, as seen in the FlowGraph in figure 5-5 below.

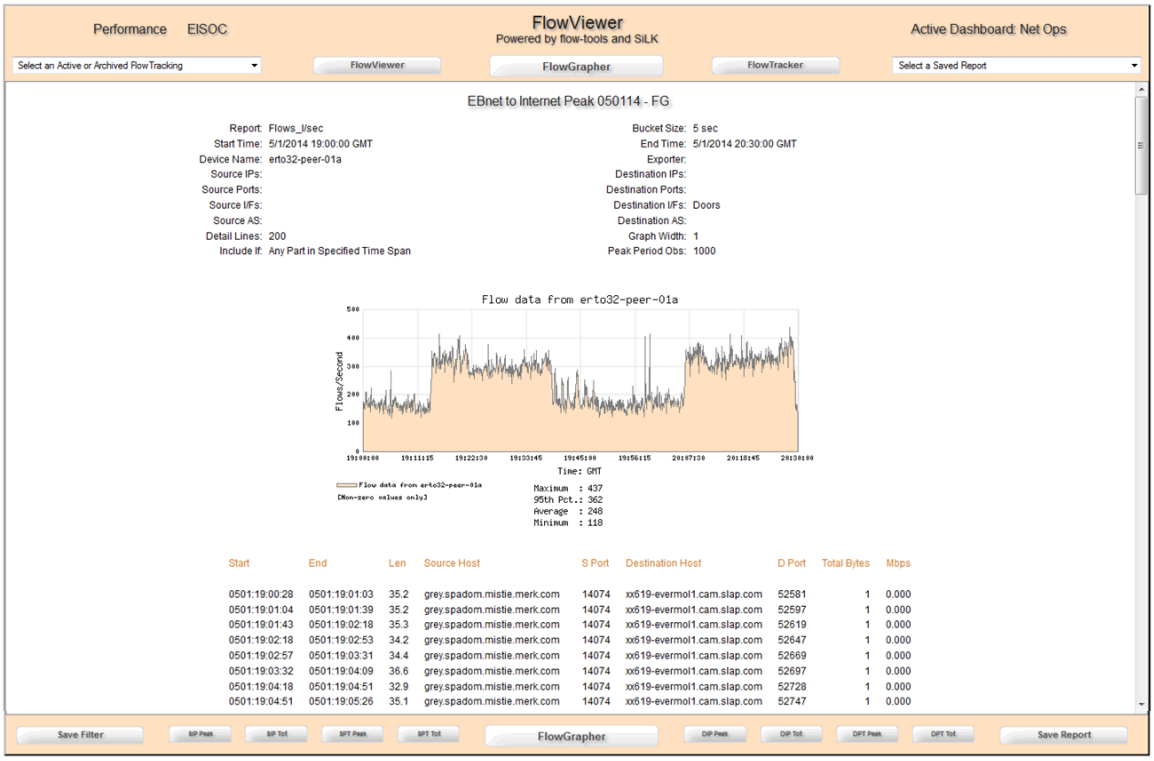

Figure 5-5 – FlowGraph isolates the anomalies and presents user with Analysis buttons

Notice the new Analysis option buttons on the bottom frame. They include:

- SIP Peak — Isolate those source hosts that have the highest flow/sec rates
- SIP Total — Isolate those source hosts that have the highest total flows
- SPT Peak — Isolate those source ports that have the highest flow/sec rates
- SPT Total — Isolate those source ports that have the highest total flows
- DIP Peak — Isolate those destination hosts that have the highest flow/sec rates
- DIP Total — Isolate those destination hosts that have the highest total flows
- DPT Peak — Isolate those destination ports that have the highest flow/sec rates
- DPT Total — Isolate those destination ports that have the highest total flows

Selecting one of these options will generate a FlowAnalysis graph. Figure 5-6 shows the associated 'Source – Total' analysis for the above FlowGraph. Several sources are seen contributing to the flows/second peaks. The FlowAnalysis isolates the top five sources or destinations and provides a row for the remainder (All Others) as well. Each isolated host and the remainder are hyper-linked for further analysis. Clicking on a host's link will create a FlowGraph with just that single source or destination. Clicking on the remainder will create a FlowGraph which has eliminated the previously isolated five hosts. This is useful if the initial Analysis did not isolate the peak sufficiently. Notice that the graph is twice the normal width –

this is selected on the input screen. A combination of increasing the graph width and increasing the bucket size can reduce the gray 'froth' that sometimes appears at the top of graphs.

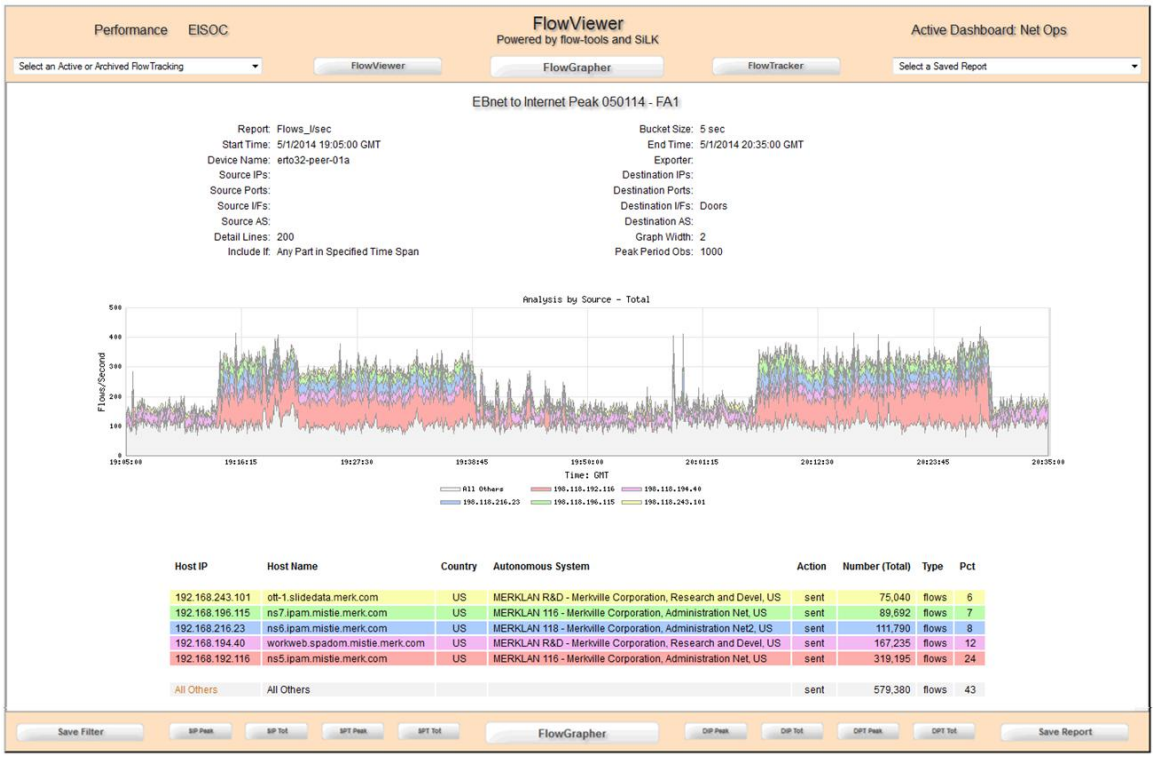

Figure 5-6 – Source Total analysis of FlowGraph which isolated anomalies

When determining Peak Sources or Destinations, FlowViewer will measure the number of observations against the configurable 'Peak Period Observations'. Whichever hosts or ports record the highest in any specified period (e.g., 1000 observations) will be graphed and linked. In the case of Bits/Second or Packets/Second, the observation is the amount (octets or packets) occurring in the flow and is summed for each 'Peak Period Observations' number of flows.

After the Analysis report is produced the user continues to have the analysis option buttons available on the bottom part of the screen. Figure 5-7 below is shows the user switching to the Destination Port Total (DPT Total) report. The report shows high traffic on port 53, DNS. This peak was probably produced by a (legitimate) DNS zone transfer.

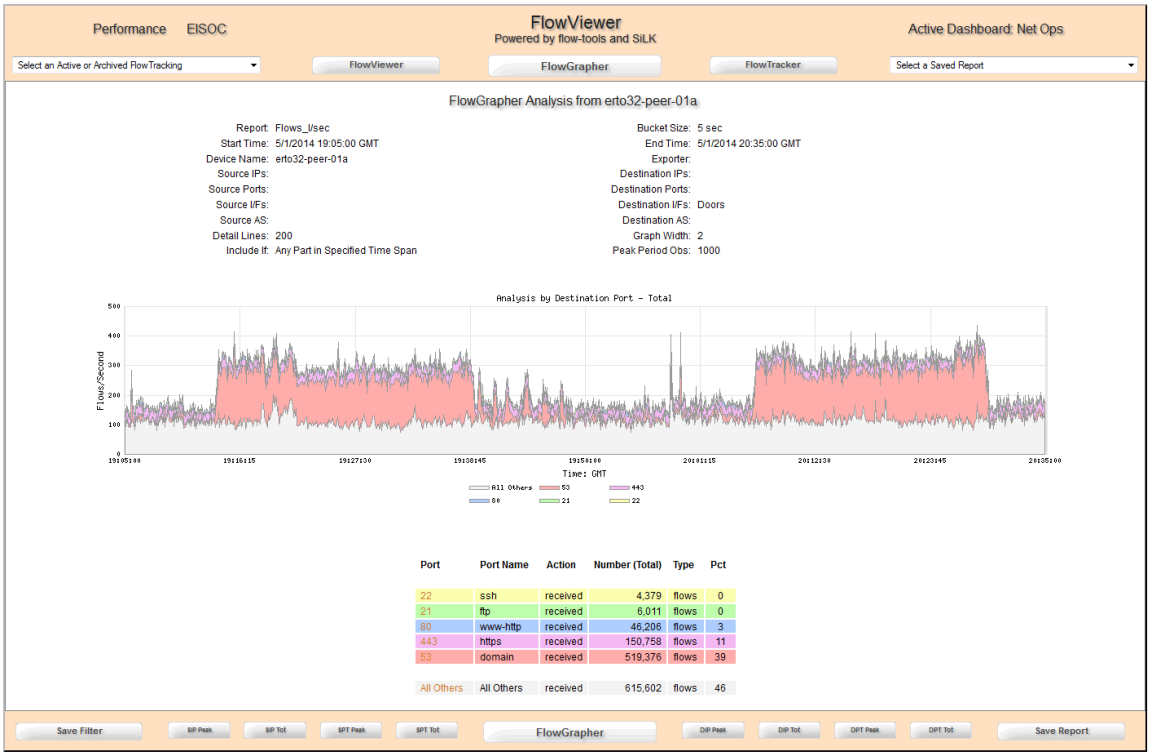

Figure 5-7 – Destination Port Total analysis of FlowGraph which isolated anomalies

The analysis 'Graph Type' options invoke a slower method of working through the all of the flows in the specified time period, so it is prudent to limit those periods to just that which surrounds your point of interest. Also sometimes peaks are very 'skinny' and it is helpful to increase the 'Graph Width' option. In figures 5-6 and 5-7 the Graph Width was set to 2.

Probably the most efficient way to determine DOS preparation (scanning, etc.) or attack is via Flows/Second. Note that the FlowViewer tool has the capability to sort on Flows which is useful in this type of analysis also. Sometimes however the perpetrator of a peak is not the same as the source or destination with the highest number of flows and it can be frustrating to find the peaks this way. The FlowGrapher Analysis option makes it easier to isolate the peaks.

6. FlowMonitor Operation and Usage

FlowMonitor consists of three parts for setting up a FlowMonitor: FlowMonitor.cgi, FlowMonitor_Main.cgi, and FlowMonitor_Group.cgi. The user invokes FlowMonitor by clicking on one of the FlowMonitor buttons on the main page or the top or bottom frames. The FlowMonitor input screen offers the same basic filtering criteria, with the exception that only a Start Time is available. Generally the user will ignore the Start Time for normal FlowMonitors. The Start Time is used when a user wishes to "recreate" a FlowMonitor (discussed below.) FlowMonitors established by FlowMonitor will be updated independently by the FlowMonitor_Collector background script. This script runs every five minutes and extracts flow data amounts, based on the established FlowMonitor filter, for a 5-minute period approximately 30 minutes in the past.

The FlowMonitor input screen (Figure 6-1 below) permits the user to define a filter for a long term Monitor or to create a Group (select "Group" from Monitor Type) from pre-existing Individual FlowMonitors. The user is prompted to provide a Monitor Label and to supply any comments that might help describe the FlowMonitor. The user can track bits, flows, or packets by selecting from the Monitor Type pulldown.

It is useful to experiment with different filters in FlowViewer or FlowGrapher before settling on the right filter for creating a FlowMonitor. When you are satisfied, simply click on the FlowMonitor button in either the top or bottom frame and the filter criteria you finally ended up with will pre-fill the FlowMonitor input screen. Although you can make any number of modifications to a Monitor once it is created, the last three of these modifications can be indicated on the graph itself, creating a vertical line to mark the change.

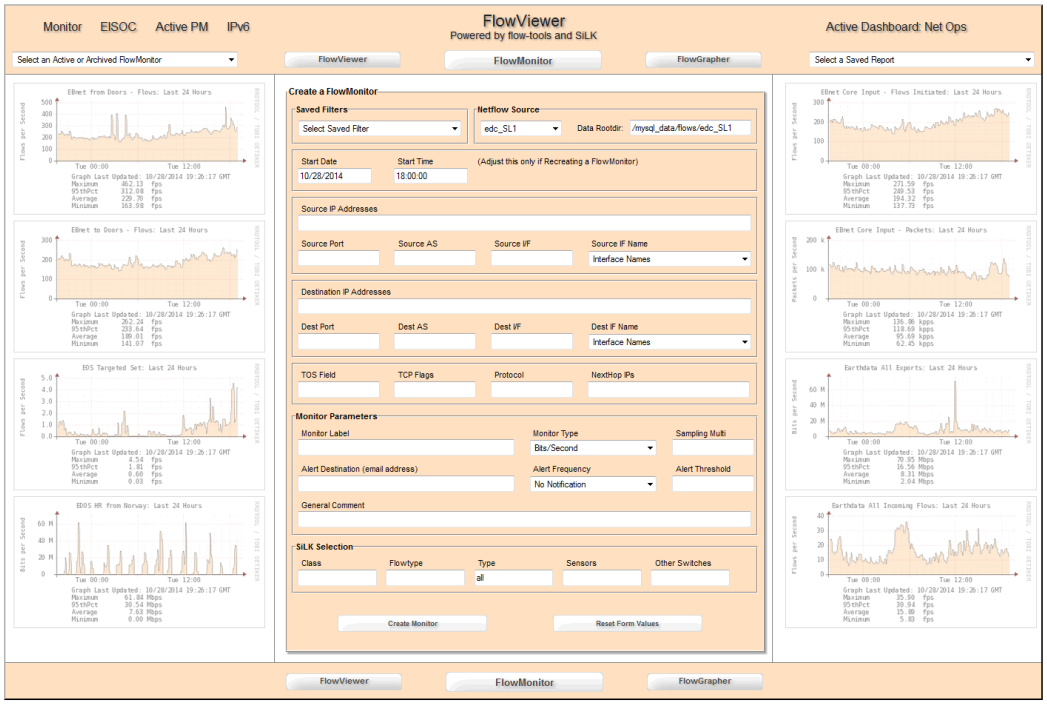

Figure 6-1 - FlowMonitor Input Screen

There is currently no means by which to suppress or eliminate a previously established change notation (i.e., vertical bar), but this can be accomplished by modifying the notification indicator in the filter file (e.g., /var/www/cgi-bin/FlowMonitor_Files/FlowMonitor_Filters/my_world.fil) . For example, change the "Y" to "N":

Display line: input: revision: **Y**|1332424500|IP address migration to 192.168.100.0/24
Suppress line: input: revision: **N**|1332424500|IP address migration to 192.168.100.0/24

The Source and Destination Interface values expect the SNMP index value for the device's interfaces. Note that these can change over time (e.g., when a new interface card is added to the device.) For FlowMonitor this becomes important. If an interface index value should change for an active Monitor, use the 'Revise' option on the FlowMonitor main page to modify the filter. This will maintain the integrity of the Monitor.

As of version 4.2.1, users may control the FlowMonitor_Collector method by selecting the desired Monitor Type. As mentioned in the "Discussion of the difference between the Linear and the Prorated Methods" subsection of section 5 above, users can now invoke a new faster method for generating both new and recreated FlowMonitors. The new Linear method is the default in the Monitor Type pulldown. If the user wishes to use the older Prorated method, this can be selected by, for example, selecting the "Bits/second – Prorated" method for a particular Monitor.

The user may establish an Alert Threshold and be alerted via email whenever this threshold has been exceeded; or for negative values, whenever the Monitor does not meet the threshold. Multiple recipients may be identified as email addresses separated by commas. The user may elect to be notified in the following optional ways:

Alerting Option	Explanation
No Notification	Default – user(s) will never be notified about this Monitor
Single Miss – Once a Day	Upon a single threshold violation (miss), user(s) will receive an email, but only once every 24 hours
3 Consecutive – Once a day	Once 3 consecutive periods are missed (i.e., 15 minutes), user(s) will receive an email, but only once every 24 hours
6 Consecutive – Once a day	Once 6 consecutive periods are missed (i.e., 30 minutes), user(s) will receive an email, but only once every 24 hours
12 Consecutive – Once a day	Once 12 consecutive periods are missed (i.e., 60 minutes), user(s) will receive an email, but only once every 24 hours
Single miss – Every Occurrence	Upon a single threshold violation (miss), user(s) will receive an email. Emails will be sent for every violation.
3 Consecutive – Every Occurrence	Every time 3 consecutive periods are missed (i.e., 15 minutes), user(s) will receive an email. After 3 consecutive misses the count is reset to zero.
6 Consecutive – Every Occurrence	Every time 6 consecutive periods are missed (i.e., 30 minutes), user(s) will receive an email. After 6 consecutive misses the count is reset to zero.
12 Consecutive – Every Occurrence	Every time 12 consecutive periods are missed (i.e., 60 minutes), user(s) will receive an email. After 12 consecutive misses the count is reset to zero.
Sigma Type 1, 2, 3 Check	Configurable in FlowViewer_Configuration.pm. Sets both the number of observations and the sigma multiplier to be applied. Example $sigma_type_1 = '6:2.67' means compute the Average and Standard Deviation based on the previous 6 observations and alert if the current observation is greater than Average + 2.67 Sigma. The user configures three types of his choosing.

If the user is not familiar with MRTG (or RRDtool) it is recommended to study the way in which succeeding graphs are derived from averages of the previous graph's data points. The FlowMonitor graphs maintain the original maximum 5-minute data point from the group that makes up the succeeding data point. It is plotted as the thin gray line at the top of the graphs. This can certainly be confusing (but valuable), hence the reference to the RRDtool web site.

When the user clicks on the "Create Monitor" button, the FlowMonitor_Main.cgi script is invoked. When the Monitor Type is set to 'Individual', this script creates a filter file to preserve the filter criteria, and an RRDtool database to maintain the 5-minute flow data readings for each Monitor, based on the filter data. The script will establish a directory to hold the five graphs which will be updated by the FlowMonitor_Grapher script every 5 minutes.

A FlowMonitor is filter-driven and basically results in five MRTG-like graphs (Last 24 Hours, Weekly, Monthly, Last 12 Months, Last Three Years) which track flow amounts over the time periods. A sample FlowMonitor is shown in Figure 6-2 below. In the figure, the window has been scrolled down and is showing only two of the five graphs: the Last 12 Months and Last Three Years.

Figure 6-2 – Last Two Graphs of a FlowMonitor (scrolled down)

Users can generate a textual listing of the data making up each graph in the FlowMonitor. Embedded in each graph is a [List Values] option. The resulting web page lists the values and also does an approximation (extrapolation: bits/seconds * seconds) of Bytes transferred during the period. An example from a "Yearly" graph showing bytes per day is shown in Figure 6-3 below.

To view FlowMonitors, the user selects the one of interest from the Active and Archived FlowMonitors pulldown to the left in the top frame (see Figure 6-4 below.) The user can also select from a listing in a different format from the Manage All FlowMonitors option (when selected see Figure 6-5 below.)

RRDtool, which is invoked to generate the set of FlowMonitor graphs, is an excellent tool that has been used by network management staff for years. In fact, the time series of graphs it produces have become a de-facto standard for representing network data traffic levels over time. Sometimes the user may want to preserve short period information in the graphs that over time will 'average out into consolidation'. In a situation like this the user can 'freeze' a FlowMonitor simply by Saving the Report.

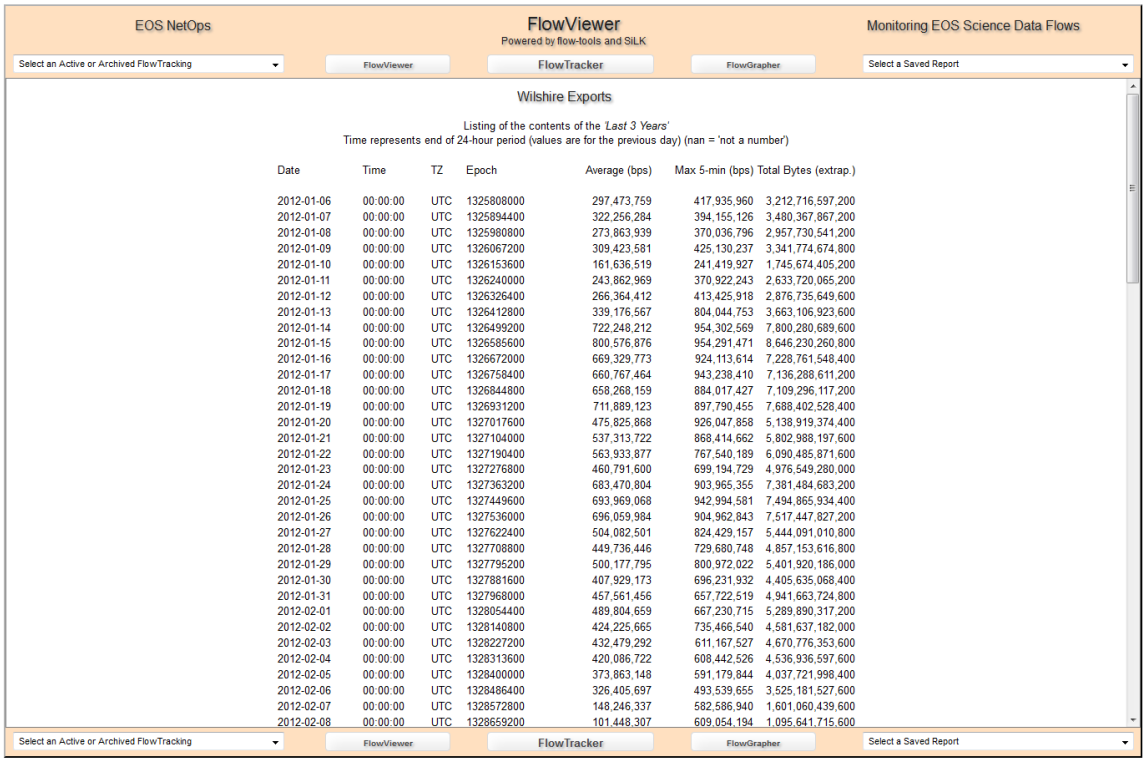

Figure 6-3 - List Values option from a "Yearly" FlowMonitor graph

Access to FlowMonitors is through the Active or Archived FlowMonitors pulldown (Figure 6-4.)

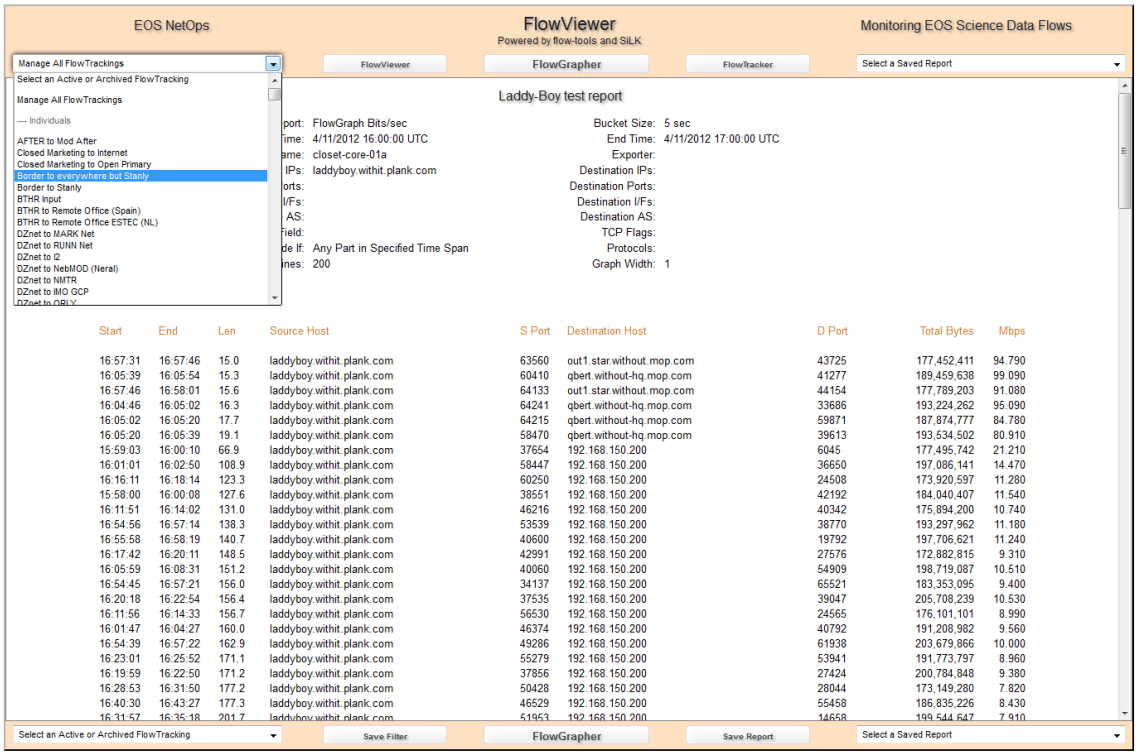

Figure 6-4 – Access to Active or Archived FlowMonitors

The Manage All FlowMonitors option on the Active or Archived FlowMonitors pulldown allows the user to manage all of the active or archived FlowMonitors and Groups. For existing FlowMonitors, the user has the ability to “Revise”, “Rename”, “Archive”, or “Remove” them. The “Revise” feature permits the user to adjust an existing FlowMonitor to use a modified filter, or to change the comment associated with a Monitor. The FlowMonitor will continue with these modifications, thus preserving historical data. The user can rename an existing Individual, Group, or Archived FlowMonitor. The user can “Archive” a FlowMonitor and “Restart” archived FlowMonitors. This can be useful for removing inactive FlowMonitors from both the collecting and graphing processes. When a user “Removes” a FlowMonitor, the script moves the Monitor files that were created (see below) to the working directory for deletion later. This allows the user a chance to recover if he has done this by mistake. Existing FlowMonitors and Groups are listed on the Manage All FlowMonitors page for management. Figure 6-5 shows a portion of the page.

Figure 6-5 – Manage All FlowMonitors Page (partial)

When the user wishes to create a Group from previously defined existing (Individual) Monitors, he selects 'Group' from the Monitor Type pulldown. No Monitor filter criteria are required and in fact are ignored if provided since a Group has no filter criteria or RRDtool databases associated with it directly. The FlowMonitor_Main.cgi script will invoke the FlowMonitor_Group.cgi script which will handle the user's creation of a Group. The Group monitor input screen is shown in Figure 6-6 below.

Figure 6-6 - FlowMonitor Group Input Screen (partial)

When defining a group, the user identifies which existing Individual FlowMonitor he would like to add to the Group next. He identifies whether it should be placed above or below the x-axis, and which color it should be. There are eight automatic colors (red, green, blue, violet, etc.) which if selected will inform the script to automatically use the next color in a range of similar colors. Each time a new component (i.e., a selected existing Individual FlowMonitor) is added to the graph, a sample RRDtool graph is created. Note that the RRDs.pm component of the RRDtool distribution is used to speed things up. If you are having problems creating these sample graphs, make sure your RRDs.pm is installed and compatible with your RRDtool version (it usually is.)

The FlowMonitor Group input screen allows the user to move components around and change their colors until a satisfactory Group is achieved. Groups can be revised just like Individual monitors can and the Group graph can be notated with a vertical bar for a particular revision if desired.

Once a Group is established it will appear with the next execution of FlowMonitor_Grapher (defaults to every 5 minutes.) The Group will appear in the Group area of the Active and Archived FlowMonitors pulldown and also in the Group area of Manage All FlowMonitors page. Users have the ability to archive monitors, and restore them to active collection and graphing later if they wish.

A typical FlowMonitor Group web page is shown in Figure 6-7 below. Only the Last 24 Hours and Last 7 Days graphs are shown. The remainder is visible by scrolling down. Group FlowMonitors have the same ability for revision as Individual FlowMonitors, with vertical bars placed on the

Group graphs when the user requests that revisions to the Group be notated. Each of the Individual FlowMonitors that make up the Group is listed beneath the graph and each is an embedded link back to the Individual FlowMonitor web page.

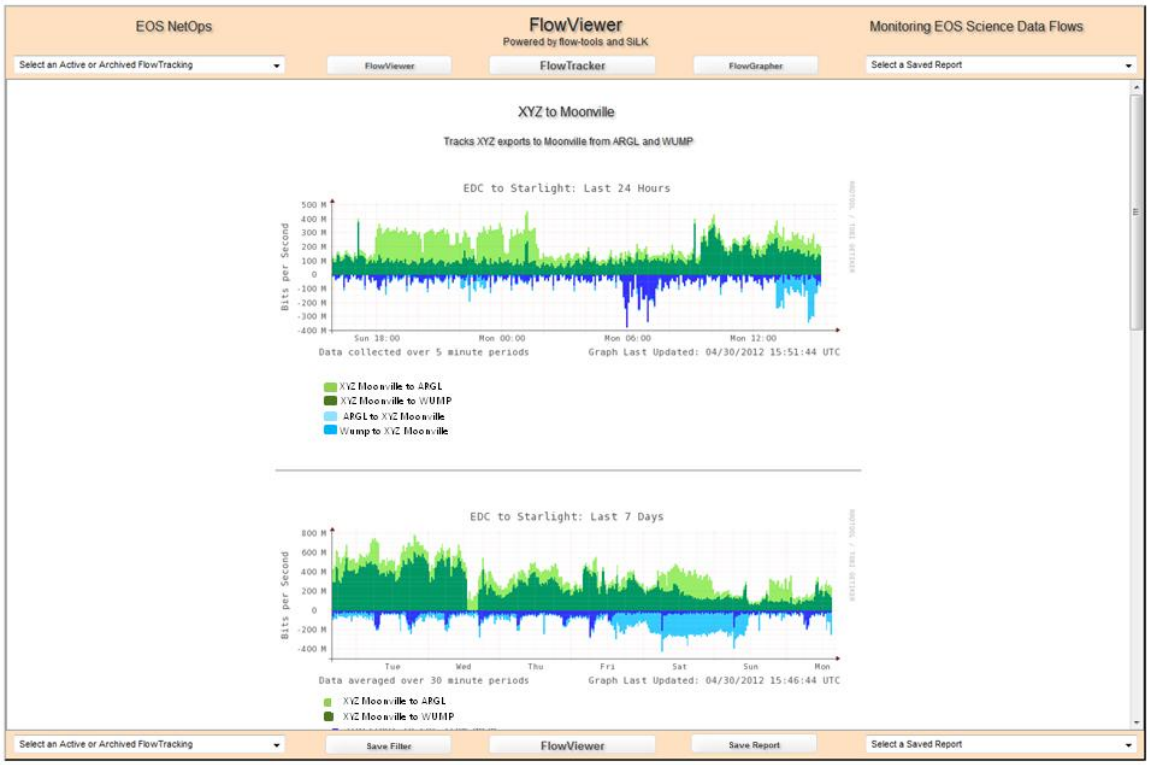

Figure 6-7 - Typical FlowMonitor Group (partial)

Running FlowMonitor_Collector and FlowMonitor_Grapher

After un-tarring the FlowViewer package and modifying the FlowViewer_Configuration.pm file for their environment, the user should initiate the FlowMonitor_Collector and FlowMonitor_Grapher programs from the command line. Each of these programs is intended to run continuously so they should be put into the 'background' (e.g. "host>FlowMonitor_Collector&.") The user does not need to start these programs on any particular minute, as they will self-adjust to collect on even five minute intervals and graph when first started and every graphing_period seconds thereafter.

Note that these processes will have to interact with files that have been established by the web server, so that the permissions on the web-server created files, (i.e., in particular the RRD files) must allow the owner of the FlowMonitor_Collector and FlowMonitor_Grapher processes to be able to write to these files.

FlowMonitor_Collector controls itself to run every five minutes. Once started, the script first looks to see if has been started less than five minutes from its previous execution. If so, it will go to sleep until a full five minutes has elapsed since its last execution. FlowMonitor_Collector will

then parse through each of the established Monitors (identified by the presence of a Monitor filter file in the FlowMonitor_Filters subdirectory) reading the filter file and

for flow-tools, invoking:

flow-cat
flow-nfilter
flow-report

and for SiLK invoking:

rwfilter
rwcount

For flow-tools, FlowMonitor_Collector will reuse previously created flow-tools concatenation files to speed up the processing. This is accomplished on a device-by-device basis, that is, concatenation files will be generated for each device one time only.

The output file is parsed and flow data for the 5-minute period is extracted (the Linear method is used exclusively as of version 4.4). The time period specified internally for all flows under consideration for the FlowMonitor 5-minute period is configurable, but should go at least 35 minutes past the end of the 5-minute period because many long flows do not get exported until a 30-minute timer expires and they would otherwise be missed. Parsing individual flows in this manner is necessary to accumulate the correct portion of flows that cross either or both ends of the 5-minute time period. The script then invokes *rrdtool_update* with this latest data point. The RRDtool file for each active monitor is stored in the FlowMonitor_RRDtool subdirectory established by the user.

FlowMonitor_Collector will compare the rate it collected for a monitor against the Alert Threshold if this has been established. If a threshold has be set up and the value determined for the collection period exceeds it (or has not met it for negative threshold values), an email is sent to the email address established along with the Alert Threshold.

After FlowMonitor_Collector has completed this process for each active Monitor, it determines how much time it took to do this and subtracts that from five minutes and puts itself to sleep for what remains of the period. If the user has elected to log FlowMonitor_Collector activity, the script will output information to the FlowMonitor_Collector.log file.

After starting up FlowMonitor_Collector, the user should invoke FlowMonitor_Grapher from the command line, placing it also in the 'background' (e.g., "host >FlowMonitor_Grapher&".) FlowMonitor_Grapher simply invokes *rrdtool_graph* to create the five MRTG-like graphs for each active monitor and goes back to sleep for a parameter adjustable period (e.g., $graphing_period = 300;.) FlowMonitor_Grapher updates only the MRTG-like graphs that have changed if the $lazy-mode variable is set.

From the FlowMonitor page, the user can click on the FlowViewer and FlowGrapher button in either the top or bottom frame. This will invoke either FlowViewer.cgi, or FlowGrapher.cgi with filter criteria pre-filled with the Monitor filter criteria. This permits additional analysis.

FlowMonitor_Recreate

The new FlowMonitor Input screen provides an entry for Start Time. The field will default to a recent time and can be ignored for normal FlowMonitors. The field is there to allow the user to "recreate" a FlowMonitor from an earlier time. The user may now "go back in time" to create a FlowMonitor they may have 'wished" they had created earlier. If the user specifies a Start Time more than 2 hours earlier than the current time, the FlowMonitor_Main.cgi script will invoke a background script called FlowMonitor_Recreate which will run in the background and return control to the user's browser. In the background, the FlowMonitor_Recreate script will determine a value for each 5-minute period (using the Linear method exclusively as of v4.4) starting at the Start time until the present time. This could take a while and hence runs in the background. When it completes, it will create a FlowMonitor filter file and FlowMonitor_Collector will now take over, updating it every 5 minutes and treating it from that point forward as a normal FlowMonitor.

7. Dashboards

The FlowViewer dashboard consists of the left and right panels that can contain up to eight "thumbnail" copies of any existing individual or group FlowMonitor. These dashboard thumbnails are automatically updated with every FlowMonitor_Grapher cycle (every 5 minutes.) Each of the thumbnails is a "hot" link and can be clicked on to display the associated FlowMonitor. The installation and placement of the dashboard component thumbnails are controlled via the "Manage Dashboard" button from the main FlowViewer screen (see bottom of Figure 1-1, above.) Figure 7-1 below shows the manage Dashboard screen.

Figure 7-1 – Dashboard Management screen

The user has the option to install, replace, move up, move down, or remove any of the eight possible "thumbnails." The user first selects the FlowMonitor and desired type (e.g., Last 24 Hours, Last 7 Days, etc.) and then completes the action. Thumbnails will automatically fill space from the top so that there are no empty slots.

FlowViewer version 4.4 introduces the ability to have multiple dashboards available from links embedded in the top panel of all FlowViewer screens. Notice in Figure 7-1 links to three dashboards (EISOC, Net Ops and the active dashboard, Performance.) Simply clicking on one of these links makes that dashboard active and provides a Manage Dashboard option for that dashboard on its main screen. Multiple dashboards can be used to separate data centers, network regions, users, or any other division the user might have in mind.

The relevant FlowViewer_Configuration parameters are shown below:

```
$dashboard_directory     = "/var/www/html/FlowViewer_Dashboard";
$dashboard_short         = "/FlowViewer_Dashboard";
#@other_dashboards       = ();
@other_dashboards        = ("/var/www/html/SOC","/var/www/html/NetOps");
#@dashboard_titles       = ();
@dashboard_titles        = ("Performance","SOC","NetOps");
```

The structure of this new option is intended to preserve older single dashboard if the user wishes to keep things that way, hence the $dashboard_directory parameter is retained as the primary dashboard. If the user wishes to have multiple dashboards, he creates a first dashboard ($dashboard_directory) and creates other dashboard directories and includes their full value in the @other_dashboards array. Finally he creates the dashboard titles for each dashboard (including the first, referenced by the $dashboard_directory parameter) in the @dashboard_titles array. In the example above the title Performance, being the first in the array, will point to the dashboard identified in the $dashboard_directory parameter. The following titles will point *in the same order* to the dashboards specified in the @other_dashboards array. Order is important. If you are using only one dashboard, be sure to set @other_dashboards, and @dashboard_titles to empty arrays:

```
@other_dashboards       = ();
@dashboard_titles       = ();
```

FlowViewer version 4.1 introduced the capability to have more than one active Dashboard but that particular approach is now deprecated.

8. Support Tools

The FlowViewer distribution includes a few tools that provide general assistance.

- ***analyze_flowmonitor_debug***

This script is used to analyze the DEBUG_TACKER_C file created in the $work_directory during each FlowMonitor_Collector loop. The script focuses on performance.

- ***analyze_netflow_packets***

This script is used to analyze previously captured netflow data. Particularly with version 9 and IPFIX, the user may want to gain a deeper understanding of the netflow packet stream. The IPFIX protocols involve occasional netflow Template packets and dynamic length data packets. The user will capture a representative sample using tcpdump and then analyze the capture file with this script. The script documentation is shown below:

```
# This script will analyze netflow export packets
#
#  Step 1 - Capture raw exports (example port 9997):
#
#  >tcpdump -s1500 port 9997 -w capture_file.raw
#
#  Step 2 - Expand the raw captured data ...
#
#  >tcpdump -X -r capture_070414.raw > capture_file.exp
#
#  Step 3 - Analyze the expanded data ...
#
#  >analyze_netflow_packets 2 v9 capture_file.exp > capture_file.txt
#
#  Step 4 - View the results ....
#
#  >vi capture_file.txt
#
# Usage: analyze_netflow_packets <debug_level> <suspected_version, e.g., v5, v9)> <capture_file>
#
# Example:
#
# analyze_netflow_packets 0 v9 capture_070414.exp > capture_070414.txt
```

Note: The network protocol analyzer, Wireshark, has the capability to decode netflow data including IPFIX and version 9. The capability is obtained by using the “Anaylze/Decode As” pulldown and selecting the UDP->CFLOW option.

- ***convert_pre40_filters***

This script is invoked prior to using FlowViewer 4.0 for the first time if the user has filters saved under an earlier version of FlowViewer.

- ***create_ports_file***

This script creates a cache file of TCP/UDP port numbers to common names. It is used by FlowGrapher_Analysis. Modify the FlowGrapher_Ports file to your liking.

- ***date_to_epoch_gm (or _local)***

These scripts convert a date (e.g., 07/12/2013 12:30:00) to its equivalent epoch representation (1373632200).

- ***epoch_to_date_gm (or _local)***

These scripts convert an epoch (e.g., 1373632200) to its equivalent date representation (07/12/2013 12:30:00).

- ***flowmonitor_restart, flowcapture_restart, flow-capture-table.conf***

For flow-tools and SiLK, *flowmonitor_restart* can be used to restart FlowMonitor_Collector and FlowMonitor_Grapher. For flow-tools, *flowcapture_restart* is a handy script for restarting all flow-captures according to a configuration file. See the README file for more information.

- ***flowmonitor_archive_restore***

A tool for restoring archived FlowMonitors that may have gotten lost in the shuffle. It simply recreates the graphs as of the latest date and time that the FlowMonitor was updated. For Groups it will recreate the graphs from the latest update time from all of the Group components.

- ***flowmonitor_grapher_nonlazy***

This script can be run from the command and it will simply recreate all FlowMonitor graphs (i.e., all five timeframes) from their RRDtool databases. It is useful if you want to see a graph before the next 5-minute run of FlowMonitor_Grapher.

- ***flowmonitor_grapher_recent***

Re-graph all recently created FlowMonitors newer than $include_period (line 54 of script).

- ***performance_check***

The *performance_check* script can be used from the command line to keep track of how well your implementation is performing. It can be run against the FlowMonitor_Collector.log file to see how things are going. The NASA Earth Observing System network has over 200 FlowMonitors and they complete in an average of 25 seconds. FlowMonitor_Collector runs every five minutes, so watch for runs that take longer than five minutes. Even in those situations, however, FlowMonitor_Collector seems to continue on with no real visible effects.

- ***resize_rrdtool***

This script is typically used to extend RRDtool databases created prior to the 3-Year capability.

- ***rsync_flows, rsync_html, rsync_monitors***

The rsync_flows, rsync_html and rsync_monitors scripts are useful for efficiently backing up all raw netflow data, HTML, and FlowMonitor state information (Filters and RRDtool databases.)

- ***rwflowpack_start***

A one-line script used to start up rwflowpack. Useful as it contains the necessary command-line parameters.

9. Cleaning Up

The following files are provided in the distribution for cleaning up caches and directories of Reports and Graphs that have lost their usefulness:

- ***FlowViewer_CleanASCache***

This script is used to remove Autonomous System resolutions that may have changed externally, but remain in the FlowViewer AS Cache file. It is invoked from the command line.

- ***FlowViewer_CleanHostCache***

This script is used to remove host name resolutions that may have changed in DNS, but remain in the FlowViewer Names Cache file. It is invoked from the command line.

- ***FlowViewer_CleanSiLK***

This script is used to monitor and adjust the disk space usage by IPFIX (i.e., SiLK) devices. This script performs what the flow-capture "-E" option accomplished (e.g., flow-capture … -E5G …) It is invoked from the command line or crontab and examines the @ipfix_storage environment parameter (e.g., @ipfix_storage = ("ipfix_rtr1:15G","ipfix_rtr2:15G");) In the example just shown, each device is limited to 15 Gigabytes of storage.

- ***FlowViewer_CleanFiles***

This script is used to clean up older files that remain in the Graph, and Work directories. This file can be invoked from the command line or daily from crontab.

```
# crontab file for flowviewer
#
# min        hr  dom  moy  dow   command
#
# Clean up files that have been lying around

12            0    *    *    *   cd /var/www/cgi-bin/FlowViewer_4.2 && ./FlowViewer_CleanFiles > logs/cleanfiles.log 2> logs/cleanfiles.log

# Clean up SiLK directories that have exceeded specified storage limit

17            0    *    *    *   cd /var/www/cgi-bin/FlowViewer_4.2 && ./FlowViewer_CleanSiLK > logs/FlowViewer_CleanSiLK.log 2> logs/FlowViewer_CleanSiLK.log

# rsync the raw flow data, html files, and the Flowtracker_Files state data

50            *    *    *    *   cd /var/www/cgi-bin/FlowViewer_4.2/tools && ./rsync_htmls > ../logs/rsync_htmls.log 2>> ../logs/rsync_htmls.log
5,20,35,50    *    *    *    *   cd /var/www/cgi-bin/FlowViewer_4.2/tools && ./rsync_flows > ../logs/rsync_flows.log 2>> ../logs/rsync_flows.log
10,25,40,55   *    *    *    *   cd /var/www/cgi-bin/FlowViewer_4.2/tools && ./rsync_trackings > ../logs/rsync_trackings.log 2>> ../logs/rsync_trackings.log
```

10. Backing Up

- ***rsync_flows, rsync_html, rsync_monitors***

The rsync_flows, rsync_html and rsync_monitors scripts are useful for efficiently backing up all raw netflow data, HTML, and FlowMonitor state information (Filters and RRDtool databases.)

11. Contents of the FlowViewer Distribution

Each of the elements of the FlowViewer 4.0 distribution are described below.

FV.cgi

This script produces the main web page for FlowViewer. It includes links to each tool, and a link for managing the Dashboard which is new in version 4.0. The user can configure two links in the top header to web sites of their choice. There is a link to this User's Guide. Place a copy of this file (FlowViewer.pdf) into the $reports_directory.

FlowViewer.cgi

This script produces the web page which provides the user the form for entering analysis selection criteria for FlowViewer. The form includes SiLK source selection criteria.

FlowViewer_Main.cgi

This script responds when the user completes the selection criteria form and submits the 'Generate Report' command. For flow-tools, the script creates a flow-tools filter file based on the selection criteria. Based on the input time period, the script concatenates the relevant flow-tools or SiLK data files for the selected device. The SiLK tools embed the filter criteria in the command line set up by FlowViewer_Main.cgi. The location of the flow-tools and SiLK raw data files is specified via the 'flow_data_directory' parameter. The script then invokes the selected statistics/print report flow-tools or SiLK programs and reformats the output into HTML. An option is available in FlowViewer_Configuration.pm to have this script use the NDBM capability (for caching resolved host names) instead of the default GDBM capability for users whose Perl distribution does not have GDBM.

FlowViewer_Save.cgi

This script moves temporary FlowViewer or FlowGrapher save files into a permanent residence as defined by the $save_directory environment variable.

FlowViewer_SaveManage.cgi

This script creates a web page for the user to manage saved FlowViewer and FlowGrapher reports. They user may link to or remove a listed report.

FlowViewer_Replay.cgi

This script will re-create a FlowViewer report that had been saved at an earlier time.

FlowGrapher.cgi

This script produces the web page which provides the user the form for entering analysis selection criteria for FlowGrapher. The form includes SiLK source selection criteria.

FlowGrapher_Main.cgi

This script responds when the user completes the FlowGrapher selection criteria form and submits the 'Generate Graph' command. The script creates intermediate processing files exactly like FlowViewer above. The script then parses intermediate output, fills time buckets, and generates a graphic image. Textual output accompanies the graph. An option is available in FlowViewer_Configuration.pm to have this script use the NDBM capability (for caching resolved host names) instead of the default GDBM capability for users whose Perl distribution does not have GDBM.

FlowGrapher_Analyze.cgi

This script responds when the user clicks on one of the analyze buttons on the bottom of a FlowGrapher report when a users has selected an Analysis graph_type (e.g., Flow Initiated / Second – Analysis, etc.) The script will graph the $analyze_count largest hosts or ports that were detected during the period.

FlowGrapher_Replay.cgi

This script will re-create a FlowGrapher report that had been saved at an earlier time.

FlowMonitor.cgi

This script produces the web page which provides the user the form for entering analysis selection criteria for FlowMonitor. The form includes SiLK source selection criteria.

FlowMonitor_Main.cgi

This script responds when the user completes the FlowMonitor selection criteria form and creates a filter file in the FlowMonitor_Filters directory. Every five minutes, FlowMonitor_Collector will parse through the directory and collect 5-minute samples for each FlowMonitor established by the filter file.

FlowMonitor_Group.cgi

This script is invoked by FlowMonitor_Main.cgi whenever a user wishes to define a Monitor group. When this is the case (i.e., user has selected the ‘Group’ pulldown) any filter criteria entered is ignored since it is not required for a group. A group simply points to existing FlowMonitors for which filter criteria has already been defined.

FlowMonitor_Management.cgi

This script creates a web page for the user to manage FlowMonitors. They may be edited, archived, restarted or removed. The script also handles the user requests.

FlowMonitor_Display.cgi

This script display a FlowMonitor selected either from the Active FlowMonitor pulldown, or from the FlowMonitors management page.

FlowMonitor_DisplayPublic.cgi

This script display a FlowMonitor selected from the $actives_webpage. The $actives_webpage permits external users to view FlowMonitors without having the ability to query any other data, maintaining access to the full set of netflow data only to those with access to the FV.cgi script.

FlowMonitor_Dashboard.cgi

This script creates a web page for controlling the creation, removal and replacement of FlowMonitor Thumbnail graphs which create the dashboard on both sides of the tool.

FlowMonitor_Thumbnail

This script creates a Thumbnail version of a selected FlowMonitor graph. Up to eight Thumbnails may populate the FlowViewer Dashboard. The script is also called to re-create the Thumbnails every five minutes with the latest data point.

FlowViewer_Configuration.pm

This file contains parameters that configure and control the FlowViewer, FlowGrapher, and FlowMonitor environments. This package should remain in the same directory that the CGI scripts are in.

FlowViewer_Utilities.pm

This file contains processing utilities used by multiple programs (e.g., 'epoch_to_date' which converts between typical date formats and 'seconds since 1972') that are invoked by other scripts. This package should be placed in the same directory as the CGI scripts.

FlowViewer_UI.pm

This file contains processing utilities used to create the FlowViewer User Interface (UI.) They are used by all scripts that present forms or reports.

FV_button.png

A simple button on the main page invokes FlowViewer.cgi which presents the input form for creating a FlowViewer report. The file needs to be copied into the $reports_directory.

FG_button.png

A simple button on the main page invokes FlowGrapher.cgi which presents the input form for creating a FlowGrapher report. The file needs to be copied into the $reports_directory.

FM_button.png

A simple button on the main page invokes FlowMonitor.cgi which presents the input form for establishing a FlowMonitor. The file needs to be copied into the $reports_directory.

FlowGrapher_Sort.cgi

This script is invoked when the user clicks on a column header for the Detail Lines of a FlowGrapher report. The textual data on the page is sorted and re-presented.

FlowViewer_Sort.cgi

This script is invoked when the user clicks on a column header of a FlowViewer report. The textual data on the page is sorted and re-presented. This capability is new in version 4.0.

FlowGrapher_Colors

This file contains a translation between textual color names and their RGB value counterparts. This file controls colors for FlowGrapher, FlowGrapher_Analysis and FlowMonitor_Grapher. The colors that start with ‘auto’ will enable you to create Groups more easily by automatically selecting the next color from a pre-defined family of colors. The user may add as many colors as desired. If you add colors, you must restart the FlowMonitor_Grapher script.

FlowGrapher_Ports

This file contains a translation between TCP/UDP port numbers and their usual service name. It is used to provide better information when looking at a FlowGrapher_Analysis run showing highest quantity or peak source or destination port traffic during the time frame.

FlowMonitor_Collector

The script is started once by the user and placed in the 'background'. The script will execute and then sleep for the duration of a five minute period, essentially running every five minutes. For each existing Monitor, the script applies the associated filter to the flow data and extracts the amount that occurred during a 5-minute window approximately 30 minutes earlier. This is to permit long-running flows to have been exported and available to the collector. The script then divides the total bits by 300 seconds to get an average bits-per-second rate during the period. The data point is then provided to RRDtool for storage.

FlowMonitor_Recreate

This script is invoked through the FlowMonitor tool web page, but will run in the background once initiated allowing the user to continue using FlowViewer. The script recreates a FlowMonitor starting at the date and time provided. If the date and time on the FlowMonitor

input screen are not modified FlowMonitor will respond as normal, setting up a FlowMonitor to be initiated with the next wake-up and run of FlowMonitor_Collector. If the recreate script is invoked, it will extract 5-minute values beginning at the start date and time and up to the current time. Each 5-minute value is computed separately and added to the RRDtool database. This results in an identical FlowMonitor to one that would have been created had it been set up in the normal way in the past.

FlowMonitor_Grapher

The script is started once by the user and placed in the 'background'. The script will execute and then sleep for the duration of a five minute period, essentially running every five minutes. The script runs the RRDtool graph function for each existing Monitor. Daily, Weekly, Monthly, Yearly and Three Years graphs are updated with the latest information. The script also updates each FlowMonitor Thumbnail that has been added to the Dashboard.

FlowMonitor_Dumper.cgi

This script is invoked when the user clicks on a link within the FlowMonitor graph labeled '[List values]'. The script dumps the RRDtool contents onto a web page.

FlowViewer.css

This file is the Cascading Style Sheet (CSS) for FlowViewer. This technique permits a richer user interface than FlowViewer had before. The file needs to be copied into the $reports_directory.

FlowViewer_CleanFiles

A utility for cleaning out temporary files that have been left over from debugging (e.g. $debug_files = 'Y'). Files older than the following configurable parameters are removed:

```
$remove_workfiles_time   = 86400;
$remove_graphfiles_time = 7*86400;
```

See above for *crontab* settings for running this automatically.

FlowViewer_CleanSiLK

This script is used to monitor and adjust the disk space usage by IPFIX (i.e., SiLK) devices. This script performs what the flow-capture “-E” option accomplished (e.g., flow-capture … -E5G …) It is invoked from the command line or crontab and examines the @ipfix_storage environment parameter (e.g., @ipfix_storage = ("ipfix_rtr1:15G","ipfix_rtr2:15G");) In the example just shown, each device is limited to 15 Gigabytes of storage.

FlowViewer_CleanASCache

A utility for cleaning out from the AS resolving cache ($as_file) a resolved AS name that is no longer valid.

FlowViewer_CleanHostCache

This script is a utility for cleaning out from the DNS resolving cache ($names_file) a resolved host name that is no longer valid.

FV_Relay.cgi

This short script refers users from older versions to the current version. This keeps you from having to notify users to go to a different web site.

Tools

analyze_flowmonitor_debug – See section 8

analyze_netflow_packets – See section 8

convert_pre40_filters – See section 8

create_ports_file – See section 8

date_to_epoch_gm (or _local) – See section 8

epoch_to_date_gm (or _local) – See section 8

flowcapture_restart– See section 8

flowmonitor_restart – See section 8

flowmonitor_archive_restore – See section 8

flowmonitor_grapher_nonlazy – See section 8

flowmonitor_grapher_recent – See section 8

performance_check – See section 8

resize_rrdtools – See section 8

rsync_flows – See section 8

rsync_htmls – See section 8

rsync_monitors – See section 8

rwflowpack_start – See section 8

12. References

1. Author contact: jloiacon@csc.com
2. FlowViewer: https://sourceforge.net/projects/flowviewer
3. flow-tools: http://www.splintered.net/sw/flow-tools/docs/flow-tools.html
4. SiLK: http://tools.netsa.cert.org/silk
5. libfixbuf: http://tools.netsa.cert.org/fixbuf
6. RRDtool: http://oss.oetiker.ch/rrdtool
7. gd Library: http://www.boutell.com/gd
8. GD: http://search.cpan.org/~lds/GD-2.46/GD.pm
9. GD::Graph: http://search.cpan.org/~mverb/GDGraph-1.43/Graph.pm
10. GD::Text: http://search.cpan.org/~mverb/GDTextUtil-0.86/Text

13. Troubleshooting

Below is the FlowViewer FAQ (http://sourceforge.net/p/flowviewer/wiki/FAQ)

1. The v3.0 package is different. Why?
2. With v3.0 my bookmarks don't work Why not?
3. I start a report, but nothing comes back. Why not?
4. I start a report, but nothing comes back. Why not? Part II
5. I get a report back, but it has no data. What's up?
6. I get a report back, but it has no data. Part II
7. I get a report back, but it has no data. Part III
8. The settings are correct, but it still has no data. What's up?
9. On long queries the browser seems to 'time out.' Why?
10. FlowViewer works, but is slow. Why?
11. FlowViewer stops unexpectedly, general unspecified problems, weirdnesses?
12. Will new versions of FlowViewer mess up my existing Monitors?
13. Why do I sometimes get "*** attempt to put segment in horiz list twice"?
14. I'm having problems and I'm running on a 64-bit system. Any known issues?
15. I want to change netflow formats, any problems?
16. FlowMonitor is not letting me create Groups
17. I'm seeing: flow-cat: Warning, partial inflated record before EOF
18. Getting: "Must select a device or an exporter.", but I'm not using devices
19. Does FlowViewer support IPFIX or netflow v9?
20. FlowViewer takes a long time to complete. Why?
21. The FlowMonitor input screen is blank. Why?
22. FlowGrapher will not generate a graph. Why not?
23. flow-capture starts, but is not writing files. Why not?
24. Why are the embedded links to Monitors not lining up on Group graphs?
25. I point my browser to FlowViewer, but only see broken image symbols. Why?
26. What is a good way to set up flow-tools?
27. I'd like to replicate flows to another host. How do I do that?
28. No graphs. HTTP error_log: Illegal division by zero at .../axestype.pm?
29. Sometimes FlowMonitor_Collector takes more than 5 minutes, and freezes. Why?
30. A FlowMonitor name got messed up and I can't remove it. How can I delete it?
31. FlowViewer returns empty for Prefix reports (e.g., Src, Dest prefix, etc.)
32. FlowViewer, FlowGrapher hang in the middle of listing reports or flows?
33. The graphs from my Archived FlowMonitors are missing. Where are they?
34. Appears FlowMonitor and FlowGrapher yield slightly different results. True?
35. I've added an IPFIX (SiLK) device and FlowMonitors are zero. Why?
36. I've added a Dashboard and Thumbnails are not updating. Why not?
37. I can’t get the SiLK implementation going. Why not?
38. Why doesn’t the Port information print for a FlowGrapher_Analysis run?
39. I’m not seeing any data for SiLK runs, but I know it is there?
40. I’m not seeing data. The SiLK command in DEBUG files does not match my environment?
41. My User Interface is all garbled - why?

1. The v3.0 package is different. Why?

Version 3.0 introduces FlowMonitor, but also provides an improvement that several users requested. They were tired of inputting the day and time with each invocation of FlowViewer or FlowGrapher. The new architecture does away with create_FlowViewer_webpage and create_FlowGrapher_webpage and has the user point his browser instead to FlowViewer.cgi or FlowGrapher.cgi. Now the start and end times are pre-filled according to how you would like it by the start_offset and end_offset parameters in the FlowViewer_Configuration.pm file.

2. With v3.0 my bookmarks don't work. Why not?

See FAQ #1 above. The structure of the scripts has changed and now the user should point the browser (and make a bookmark) to FlowViewer.cgi, FlowGrapher.cgi, and now FlowMonitor.cgi instead of /htp/htdocs/FlowViewer/index.html, etc. (or however your http environment was set up.)

3. I start a report, but nothing comes back. Why not?

This could be caused by your web server CGI settings. Examine the httpd.conf file to make sure that the web server is set up to execute CGI. Make sure that the FlowViewer_Configuration parameters $cgi_bin_directory and $cgi_short are set correctly with respect to your web server environment. Typically, the cgi-bin directory is aliased. Here is an example from Apache:

```
#
# ScriptAlias: This controls which directories contain server scripts.
# ScriptAliases are essentially the same as Aliases, except that
# documents in the realname directory are treated as applications and
# run by the server when requested rather than as documents sent to the client.
# The same rules about trailing "/" apply to ScriptAlias directives as to
# Alias.
# ScriptAlias /cgi-bin/ "/htp/cgi-bin/"
```

In this case, provided that the contents of FlowViewer package now resided in the /htp/cgi-bin/FlowViewer_4.0 directory, the relevant parameters and settings would be:

$cgi_bin_directory = "/htp/cgi-bin/FlowViewer_4.0"; $cgi_bin_short = "/cgi-bin/FlowViewer_4.0";

And, as always, make sure that all relevant directories have been created and permit the web-server process to write into them. This includes the 'reports', 'graphs', 'monitor', 'names', 'work', and 'log' (if you're logging) directories.

The following can help you get started. Afterwards you can tighten things up as you want.

From the $cgi_bin_directory issue a 'chmod -R 0777 *' From the $flow_data_directory issue a 'chmod -R 0777 *' From the $reports_directory issue a 'chmod -R 0777 *' From the

$graphs_directory issue a 'chmod -R 0777 *' From the $monitor_directory issue a 'chmod -R 0777 *'

Turn on debug ($debug_viewer = "Y";, etc.), make a run, and examine the DEBUG_VIEWER output. The output will have the text of the flow-tools command that was created. Cut and paste this command to a command prompt, run the command, and review the results. This may give you a clue to what is happening.

You can also simply run FlowViewer.cgi, FlowGrapher.cgi, or FlowMonitor.cgi from the command line. This may provide a good hint. For example:

'cannot mkdir /var/www/FlowGrapher_3.2/: Permission denied at FlowGrapher.cgi line 58.'

This would mean that you have to loosen permissions on /var/www, or create the subdirectory yourself with adequate permissions (e.g., 0777).

4. I start a report, but nothing comes back. Why not? Part II

Perhaps you haven't created the directory pointed to by $work_directory. This would prevent processing from completing.

5. I get a report back, but it has no data. What's up?

Make sure the FlowViewer scripts are reading flow-data from the correct directory. FlowViewer will look for flow-data according to three settings in the FlowViewer_Configuration.pm file. These are:

a. $flow_data_directory
b. @devices
c. $N

For example, here we track netflow data from several devices using the default flow-tools nesting value. Our file structure looks like:

```
/htp/flows/ecs_edc/2006/2006-01/2006-01-19/ft-v05.2006-01-19.000001-0500
<-- a --->|<- b ->|<---------- c -------->|<-- actual flow-data file -->
```

In this case a) '/htp/flows' is our flow_data_directory, b) 'ecs_edc' is one of our devices, and c) the three levels of nested date-ordered directories are addressed by setting $N = 3 (the FlowViewer default.) Note that $N can be confusing because the flow-tools documentation indicates that -N0 is the default, but if you do not put a '-N' modifier on your flow-capture statement, it will behave as if -N3 has been set.

In our FlowViewer_Configuration.pm, the variables are set as follows:

```
$flow_data_directory = "/htp/flows";
@devices = ("ecs_edc","router_1","router_2","router_3");
 $N = 3;
```

Also, verify that the flow-tools are in the $flow_bin_directory you have specified. This can be accomplished by, e.g., 'which flow-stat';

6. I get a report back, but it has no data. Part II.

Another possibility for this problem is that the timestamps on the flows are not what you are expecting, and hence the data is completely filtered out. For example, you may wish to see everything from 10:00:00 to 11:00:00 but the report is empty, and you're sure you have data because there are plenty of non-zero sized ft... files in your flow-data directory. It may be that the flows are time stamped quite differently from the file timestamp.

In this case a simple "flow-print -f5 < ft-v05.2006-01-19.100001-0500" will list the flows with embedded time stamps. The output could be long so you might want to redirect it to a file first. Compare the flow timestamps to what you are expecting. If they are off - then perhaps your router's time setting is off, or your computer time setting is off.

7. I get a report back, but it has no data. Part III.

In the situation where you generated a large FlowViewer or FlowGrapher report you may have generated a temporary intermediate file (e.g., /tmp/FlowGrapher_output_070406) that exceeds the amount of space available to the partition that holds your working directory (e.g., you used up all of /tmp space.) To fix this, remove the offending file and either run a smaller report, or increase the size of your working directory, or move it to a directory on a larger partition.

8. The settings are correct, but it still has no data. What's up?

Another possibility for an empty report is that the web server (e.g., Apache) that is running the CGI scripts does not have adequate permission to read from the flow-data directory or files. Review the permissions of the flow-data directories and files to make sure they are 'open' enough.

Make sure that Apache can get access to the flow-tools specified by the $flow_bin_directory parameter.

The following can help you get started. Afterwards you can tighten things up as you want.

From the $cgi_bin_directory issue a 'chmod -R 0777 *'
From the $flow_data_directory issue a 'chmod -R 0777 *'
From the $reports_directory issue a 'chmod -R 0777 *'
From the $graphs_directory issue a 'chmod -R 0777 *'
From the $monitor_directory issue a 'chmod -R 0777 *'

If you are running a version of Security Enhanced Linux (SELinux), verify that there are no file or directory access controls that are preventing Apache from accessing either the flow-data directory and files, or the flow-tools themselves.

Since everything in the stock configuration (original FlowViewer_Configuration.pm) is below /var/www, one can issue the following command to free things up:

host> chcon –R –t public_content_rw_t /var/www

Or you could disable SELinux functionality:

In /etc/selinux/config file, set SELINUX=disabled.

9. On long queries the browser seems to 'time out.' Why?

When you have requested a time period that requires the analysis of many flows, while flow-tools is cranking away no data is being sent to the browser. As a consequence, the connection drops. This closes the data path and no data is sent back to the browser.

Reset either the web server or web browser setting that is controlling this. For example, with Apache there is a timeout value that controls this and is set to 300 seconds. Adjust this to 1800 which will permit browser-to-server connections to stay open for 30 minutes.
Apache example, in the httpd.conf file:

```
#
# Timeout: The number of seconds before receives and sends time out.
#
#Timeout 300
Timeout 1800
```

Remember to stop/restart your web server in order to read the new httpd.conf settings.
Some have had to modify a similar setting on their browsers.

10. FlowViewer works, but is slow. Why?

Most likely the FlowViewer script is not taking advantage of the caching that the 'names' file provides. Make sure that your web server process owner (e.g., Apache) has adequate permission to write into the directory identified by the $names_directory parameter in the FlowViewer_Configuration.pm file. For example, set the permissions to 0777 for the $names_directory.

Also, make sure that the permissions on the 'names' file itself are open enough for the web server process owner to write to the file.

Note also that queries over long time periods cause the flow-tools flow-cat process to really crank through a lot of data for busy routers. So if you are looking at a busy device, you will get better response times for shorter queries.

11. FlowViewer stops unexpectedly, general unspecified problems, weirdnesses?

Permissions. Many problems are caused by restrictive file permission settings. This is particularly important with FlowMonitor. With FlowMonitor you have the web process owner (e.g., apache)

taking care of creating, modifying, and deleting Monitors, but you may have a different user (perhaps your own account) starting and running the FlowMonitor_Collector and FlowMonitor_Grapher scripts. Inadequate permissions will stop things in their tracks.

There are at least two ways out of this jam. The first is to set up and run everything as the web server process owner (e.g., apache); installing, creating directories, and executing scripts (e.g., FlowMonitor_Collector) as that user.

The other way out is to make sure all scripts, directories, subdirectories, files, etc. have permissions that permit the owner AND the web server process the equal ability to read, write, and execute all files, directories, scripts, etc. A good way to aid in this is to put both accounts in the same group, and provide the group with write permissions.

You might have to reset the umask for each of these accounts.

12. Will new versions of FlowViewer mess up my existing Monitors?

No. Care has been taken to preserve existing Monitors with new versions of FlowViewer including new versions of FlowMonitor_Collector and FlowMonitor_Grapher. After configuring the FlowViewer_Configuration.pm file for your environment, and making sure that the $filter_directory and the $rrdtool_directories contain the existing filters and RRDtool databases, the user can simply 'kill' the running versions of FlowMonitor_Collector and FlowMonitor_Grapher, and start up the new versions.

13. Why do I sometimes get "* attempt to put segment in horiz list twice"?**

Occasionally FlowMonitor_Grapher will output this error message into the shell it was launched from. The best I can tell this is caused by a bug in some old versions of librsvg (or similar) which fails to cope with some SVG images during RRDtool's generation of the graph. It appears to be harmless.

14. I'm having problems and I'm running on a 64-bit system. Any known issues?

This is fixed in the new 'forked' version of flow-tools found here:

flow-tools v 0.68

If you are installing the previous version, the following applies:

Yes, some 64-bit users are having problems. The best I can tell at this point is there is a bug in flow-tools when deployed on a 64-bit platform. There are three solutions that I'm aware of. The first is a patch to flow-tools by Mike Hunter:

flow-tools 64-bit patch 1

The second is a more extended patch (Paul Komkoff Jr.) that uses ia temporary variable:

flow-tools 64-bit patch 2

The third approach (Ryan Gerdes) was to use binaries for key flow-tools components: flow-cat, flow-print, flow-nfliter, and flow-stat compiled for the 32-bit version of the OS.

15. I want to change netflow formats, any problems?

The flow-tools flow-cat process does not concatenate across varying netflow type boundaries. That is, if you run a FlowViewer report that includes v5 and v7 data (for example) no report will be generated. If you use the DEBUG feature, cut and paste the flow-run command string onto a command prompt, and run it, you will get the following error message:

flow-cat: data version or sub version changed!

flow-tools will work on either type by itself, so as long as you confine the requested time period to one or the other, you'll be OK. Or you can have flow-tools store the v7 data as v5:

flow-capture -V 5 0/0/*** -w /blah/blah/blah

As far as losing any data, according to Mark Fullmer:

"You'll lose the router_sc field. AFAIK unless there are multiple routers providing shortcut paths to the switching module this field will never change."

Flow-tools mailing list email that discusses this, including how to use flow-xlate to merge: Version change discussion

16. FlowMonitor is not letting me make Groups

If the FlowMonitor Group page appears but there is no sample graph at the top, or you receive an "Internal Server Error" (most likely Perl compilation problem, it could be that you haven't correctly installed RRDs.pm. A quick way to check for this is to issue a

'perl -c FlowMonitor_Group.cgi'

from a command line. If there is a RRDs.pm location problem the script will not compile.

This problem can be tricky and I will try to make it easier in the next version. In the meantime the easiest way to fix this is:

1. Do a 'perl -V' from a command line, and look at the @INC array

@INC: /usr/lib/perl5/5.8.5/i386-linux-thread-multi /usr/lib/perl5/5.8.5
/usr/lib/perl5/site_perl/5.8.5/i386-linux-thread-multi (... more)

2. Identify the most likely directory into which to put a copy of RRDs.pm

probably: /usr/lib/perl5/site_perl/5.8.5/i386-linux-thread-multi

3. Copy RRDs.pm into that directory

from: /usr/local/rrdtool-1.2.26/lib/perl/5.8.5/i386-linux-thread-multi/RRDs.pm
to: /usr/lib/perl5/site_perl/5.8.5/i386-linux-thread-multi/RRDs.pm

4. Copy the RRDs and RRDp 'auto' subdirectories and their contents into the Perl 'auto' subdirectory

from: /usr/local/rrdtool-1.2.26/lib/perl/5.8.5/i386-linux-thread-multi/auto/RRDp
to: /usr/lib/perl5/site_perl/5.8.5/i386-linux-thread-multi/auto/RRDp

from: /usr/local/rrdtool-1.2.26/lib/perl/5.8.5/i386-linux-thread-multi/auto/RRDs
to: /usr/lib/perl5/site_perl/5.8.5/i386-linux-thread-multi/auto/RRDs

Note: the above can be accomplished using links instead of copying.

Another thing to watch for is using a special character in the FlowMonitor 'Monitor Set Label' text box. This field is used to create a file-name and is allergic to many special characters.

17. I'm seeing: "flow-cat: Warning, partial inflated record before EOF"

This error message may indicate that you are trying to read an empty directory. This error would appear for Exporter users in the very initial release of version 3.3. This was was caused by FlowMonitor_Collector trying to read a device_name directory even though the user was not using devices. This was fixed in version 3.3.1.

This error may also occur during the processing of data in normal directories. It is not understood at this point why this happens, however it appears to be mostly harmless. I have seen it occur on every third FlowMonitor_Collector run (every 15 minutes) which coincides with the end of a typical 15-minute flow-tools ft file.

18. I'm getting: "Must select a device or an exporter.", but I'm not using devices

This would happen for early users of version 3.3 if they were not using devices or exporters. This was fixed in version 3.3.1.

19. Does FlowViewer support IPFIX or netflow v9?

Yes. The user must install the SiLK software from CMU's NetSA group. Note that you must use at least version 3. Silk can be downloaded from tools.netsa.cert.org. SiLK is wonderful software. FlowViewer version 4.0 continues to happily work with that other outstanding netflow software we all love: flow-tools.

20. FlowViewer takes a long time to complete. Why?

This could be because your environment does not have a properly working DNS resolution capability. FlowViewer (and FlowGrapher) default to "Y" for Resolve Addresses on the input screen, so FlowViewer is attempting to resolve each IP address, there is no resolving capability, and it takes a while to complete this task. You should set this field to "N". You could modify the

FlowViewer.cgi and FlowGrapher.cgi scripts to select "N" instead of "Y" if you like. I will try to put such a switch in the next version.

21. The FlowMonitor input screen is blank. Why?

This could be caused by uncreated directories for $filter_directory, or $rrdtool_directory. Create these directories and provide them with adequate permissions to solve this problem.

22. FlowGrapher will not generate a graph. Why not?

This may be caused by a non-optimal installation of GD (and libgd). A good way to test this out is to issue a 'perl -c FlowGrapher_Main.cgi' from a command line. If you get the follwing message, or something like it, you have probably installed the GD components in a location that Perl is not familiar with.

"Can't load '/usr/lib/perl5/site-perl/5.8.5/i386-linux-thread-multi/auto/GD/GD.so' for module GD:libgd.so.2: cannot open shared object file. No such file or directory at /usr/lib/perl5/5.8.5/i386-linux-thread-multi/DynaLoader.pm line 230. at FlowGrapher_Main.cgi line 82."

A soft link or a reinstall will help solve this.

23. flow-capture starts, but is not writing files. Why not?

Everything looks fine: flow-capture is running in the background, 'netstat -an' shows it listening OK on the specified port, and tcpdump shows netflow UDP packets arriving, yet no capture files are being created.

Have you created the directory that the netflow data is supposed to go into? For example if your flow-capture command looks like this:

flow-capture -p /var/flows/pids/flowtool.pid -w /var/flows/cwrouter_1 -E5G -S3 0/0/2050

You must manually create the /var/flows/router_1 directory and give the flow-capture process owner adequate permissions to write into the directory.

Also, this could be caused by a host firewall blocking the packets from going into the TCP stack. Turns out that the firewall will stop the packets after tcpdump sees them. Simply adjust the firewall rules (e.g., iptables) to permit the netflow exports.

24. Why are the embedded links to Monitors not lining up on Group graphs?

It seems that newer versions of RRDtool handle the COMMENT command that produces a line break a little differently. It involves FlowMonitor_Grapher only and is fixed in later versions of FlowViewer_3.3.1.

25. I point my browser to FlowViewer.cgi, but see only broken image symbols. Why?

This could be due to a number of things, mostly related to permissions.

Make sure that the process owner for your web server (e.g., apache, www-data etc.) has write permissions into the htdocs directory immediately above the $reports_directory. Sometimes this is the root htdocs file as defined in the httpd.conf (or with Debian, the apache2.conf file.) FlowViewer.cgi will try to create your $reports_directory if you haven't already created it, and it will try to copy the FlowViewer.png and User Logo graphics into the directory. The web server process owner must be able to write into the directory.

Make sure you have manually created the $work_directory and that the web server process owner has write permissions into it. Same for the $names_directory.

In general, make sure all directories defined in FlowViewer_Configuration.pm have been created and have write permissions for the web server process owner. To be very safe, do this manually ahead of time.

Finally, make sure your browser (i.e., desktop IP address) can acccess the web server htdocs directory structure. Sometimes access controls may be blocking this.

26. What is a good way to set up flow-tools?

The flow-tools software is excellent. It is very stable and has great flexibility through so many options. With FlowViewer, however, the only component that you have to work with is flow-capture. FlowViewer will automatically invoke several of the other components for you.

The man pages are very informative. I think this is the most recent version:

flow-tools man pages

A typical flow-capture command may look like this:

flow-capture -p /var/flows/pids/flowtool.pid -w /var/flows/router_1 -E5G -S3 0/0/2050

In the case above, I am storing netflow data separately for each device instead of collecting from multiple exporters into a single directory structure. You can see this by the fact that I have identified the directory using the name of a single device, 'router_1'. I would have a second, similar command for a second device (e.g., 'router_2') where the only difference in the command syntax would be to replace 'router_1' with 'router_2" and to increment the receiving port number from '2050' to '2051', say. I would execute both commands and have two flow-captures running simultaneously. Actually, here at NASA GSFC, I'm running 23 flow-captures simultaneously. Each one takes a surprisingly little amount of CPU, with four of them receiving from very busy devices.

The -p parameter identifies a directory where flow-capture will store the process identifier (PID) for the flow-capture process. The -w parameter identifies the location for depositing the netflow data. The -E parameter identifies how much disk space (5 Gigabytes) should be allocated to this collection, with flow-capture aging out netflow data once the limit is reached.

The -S3 parameter informs flow-capture to write a status message to the log file (generally e.g., /var/log/cflowd.log) every 3 minutes. The 0/0/2050 notation informs flow-capture to expect netflow data from any device IP address (use of '0') and to capture it with any destination IP address. These can be specific IP adresses as well. The UDP port number for receiving packets from the device is 2050.

At this point you are ready to modify the @devices field in the FlowViewer_Configuration.pm file to match the collection directory name (i.e., 'router_1') and you are ready to go.

If you wish to collect from multiple exporters, all exporting to the same UDP port, your flow-capture syntax might look like this:

flow-capture -p /var/flows/pids/flowtool.pid -w /var/flows/all_routers -E5G -S3 0/0/2050

In this case you would set up the following relevant parameters in FlowViewer_Configuration.pm:

$exporter_directory = "/var/flows/all_routers"; @exporters = ("192.168.100.1:New York Router", "192.168.100.2:Prague Router");

Finally, you may simply collect all netflow data (from one or more devices) into a single directory structure and not use named devices or exporters. The flow-capture command might look like:

flow-capture -p /var/flows/pids/flowtool.pid -w /var/flows/all_flows -E5G -S3 0/0/2050

In this case you would set up the following relevant parameters in FlowViewer_Configuration.pm:

$exporter_directory = "/var/flows/all_flows"; $no_devices_or_exporters = "Y";

If you are having having problems capturing netflow data, see FAQ #23 above.

27. I'd like to replicate flows to another host. How do I do that?

Sometimes it is useful to be able to replicate a netflow stream coming to your normal capturing host on to another host. The flow-tools flow-fanout tool will do this. I, and others, have found the example on the flow-fanout man page to be confusing.

So, after playing with it for awhile, the following command seemed to do the trick (also thanks to Victor Wiebe):

>flow-fanout -s -V5 -S3 -p/var/flows/pids 0/0/2095 0/127.0.0.1/2195 0/192.168.100.10/2095

In the above example, any flows received at the capturing host on port 2095 will be replicated to the local host (127.0.0.1) on port 2195, and a new stream reflected to host 192.168.100.10 and received there on port 2095. The -s parameter will wind up substituting the exporter IP address as the source address on packets sent to the local host and to the reflected host. The -V5 ensures that all reflected PDUs continue to be in version 5 format. You may need to change

this for other versions. The -S3 parameter tells flow-fanout to record status messages every 3 minutes.

You'll probably need 'root' to be able to get flow-fanout going as it requires the ability to open a socket via the 'setsockopt' command.

Once the replication is working, you would need to start up a flow-capture on the local host which is listening on port 2195, and a flow-capture on the reflected host listening on port 2095.

I think what makes the man page examples confusing are two things. The localip and remoteip fields make different sense in different contexts depending on whether the 'triplet' in question is the orginal capturing host, the local host, or the reflected host. Also, the example provided receives packets on port 9500, and then resends them to the local host on this same port (and to port 9200 on the reflected host.) When I tried that I wound up sending an endless loop of packets to the remote device as the replicator was essentially listening to itself, caught in a feedback loop.

Here's a way to preserve the same port across the fan-out (thanks to Paul Fuller, NASA.) The receiving host is 192.168.10.10. It fans out to two other hosts.

>flow-fanout 192.168.10.10/0/2061 0/0/2061 0/192.168.20.20/2061 0/192.168.30.30/2061

>flow-capture -p /flows/pids/ -w /flows/router_1 -E120G -S3 127.0.0.1/0/2061

28. No FlowGrapher graphs. HTTP error_log: Illegal division by zero at .../axestype.pm?

This could be caused by a number of variants around the functioning of GD::Graph. In some cases GD::Graph has not been installed quite properly and a re-install did the trick. In another case, version 1.43 had a bug, and an install of verion 1.44 fixed the problem. Also, there may be an inconsistency in how GD::Graph, GD, and fonts (particularly FreeType fonts) interact.

29. Sometimes FlowMonitor_Collector takes more than 5 minutes, and freezes. Why?

It may happen that the host is overburdened sometime and FlowMonitor_Collector winds up taking more than 5 minutes (300 seconds) to complete. For me, this normally has not been a problem, as it simply starts back up for the next period immediately after the long period completes. This has worked quite well on Red Hat Linux. Recently we upgraded our hardware and also switched to a Debian OS.

Normally here FlowMonitor_Collector will complete somewhere around 60 seconds for our 124 Monitors. However, recently at midnight we've experienced an excessive use of system resources, and as a consequence, FlowMonitor_Collector would take more than 300 seconds to complete. I was quite surprised to see that even though the FlowMonitor daemon was still running, the collection was not taking place at all. FlowMonitor_Collector times itself and determines how long it should sleep before running again at the next 5-minute mark. It turns out the Debian Perl 'sleep' function hangs on values less than zero. Red Hat Perl would convert these to zero and continue happily along. Of course, it could be different versions of Perl causing this, but I haven't got that far in post-mortem analysis.

30. A FlowMonitor name got messed up and I can't remove it. How can I delete it?

FlowMonitor names are allergic to special characters. It might be the case that you have created a FlowMonitor with a label that contains such a special character. In fact, the label got so messed up that the FlowMonitor web page 'Remove' link won't actually remove the FlowMonitor. It is easy to remove any Monitor by manually removing both the filter file from your $filter_directory, and the RRDtool file from your $rrdtool_directory. At that point all trace of the Monitor is gone.

31. FlowViewer returns nothing for Prefix reports (e.g., Source, Dest prefix, etc.)

This may be caused by the type of netflow data that you are receiving. If you run the debug generated FlowViewer command string (e.g., from Flow_Working/DEBUG_VIEWER) from a command line, you may receive the following message from flow-tools:

```
# --- ---- ---- Report Information --- --- ---
#
# Fields: Total
# Symbols: Disabled
# Sorting: Descending Field 3
# Name: Destination Prefix
#
# Args: /usr/local/flow-tools/bin/flow-stat -f25 -S3
# flow-stat: Flow record missing required field for format.
```

32. FlowViewer, or FlowGrapher, seems to hang in the middle of listing reports or flows

For example, A FlowGrapher report returns with a nice graph, but in the middle of listing the associated flows it hangs. A browser, like Firefox, will report 'Done'. This could be caused by the 'dig' name resolution function hanging while trying to resolve an IP address. This has been found to be caused by a corrupt 'names' file. Remove this file and retry the report. Note that you will lose all name caching you have established to that point and you will have to start anew. Still researching how the file gets corrupted.

33. The graphs from my Archived FlowMonitors are missing. Where are they?

When clicking on a link on the FlowMonitor page to an archived link, the page appears but the graphics are missing. This may have been caused during a transition to a new FlowViewer version (e.g., v3.4). FlowMonitor directories have been established, but the individual contents (i.e., long-term graphs) have not been copied over. Either the user can manually copy these over to the new directories, or the user can set up the FlowMonitor_Relay.cgi script as described in the README file. Once a user clicks on the FlowMonitor web page (set up by the FlowMonitor_Relay.cgi script now named simply FlowMonitor.cgi and in the old directory), all of the individual Archive files are copied into the new directory.

34. It appears FlowMonitor and FlowGrapher yield slightly different results. Is this true?

When one compares a 5-minute FlowMonitor result to a FlowGrapher report for the same 5-minute period a slight difference can be seen when looking at the FlowGrapher Average. The FlowMonitor value for the 5-minut period is the sum of all bits that passed in that period divided by 300 seconds. This should agree with the Average computed by FlowGrapher for the same period and shown as one of the statistics in the graph itself. These can be off slightly though most of my observations has the difference at most around 0.5%. As part of the development of v4.0 (which can handle IPFIX and v9) I decided to iron out this annoying discrepancy. It turns out that most of the difference is attributable to round-off resulting from ignoring or rounding off flow time milliseconds. This was not wise, as after a little thought, of course milliseconds can sometimes be significant on the network (!) Anyways, this is polished up post v4.0.

35. Why are FlowMonitors for the new device I added coming up all zeroes?

Whenever you add a device to the @devices or @ipfix_devices arrays in FlowViewer_Configuration.pm, you must restart FlowMonitor_Collector and FlowMonitor_Grapher that have been running in the background. They load the information from the FlowViewer_Configuration.pm file only on start-up.

36. I've added a Dashboard and Thumbnails are not updating. Why not?

When FlowMonitor_Collector is running in the background it does not have access to changes to the FlowViewer_Configuration.pm file. Restart FlowMonitor_Collector and FlowMonitor_Grapher.

37. I can't get the SiLK implementation going. Why not?

Have you placed the silk.conf file in the device directory (section 2.2.1)? Also, version 4.4 attempts to make SiLK implementations easier to get going with a much more flexible user interface which hopefully permits more SiLK environments than just those that are set up to mimic flow-tools.

38. Why doesn't the Port information print for a FlowGrapher_Analysis run?

There may have developed an error in the 'ports' file created to cache port number to name mappings. Remove the 'ports' file from $names_directory. After modifying the FlowGrapher_Ports file to your environment, use the /tools/create_ports_file script to create the 'ports' file.

39. I'm not seeing any data for SiLK runs, but I know it is there?

Check to see if your system local time is not UTC. If it is not, you should upgrade FlowViewer to version 4.6 which will allow the stored SiLK data to be in a different time zone that the specified system time zone. Note that this is the default situation for users whose system timezone is not UTC. SiLK by default keeps data in UTC time whatever the system time is. You can compile SiLK with the –enable-localtime switch which will then maintain stored data in the system local time.

40. I'm not seeing data. The SiLK command in DEBUG files does not match my environment?

Prior to version 4.4, FlowViewer had a very strong flow-tools heritage which was not lining up well with some SiLK environments, particularly those that had a 'singular' structure with the normal subdirectories (e.g., 'in', 'inweb', etc.) appearing directly below the 'rootdir'. Described differently, there was no 'device' level as in the recommended data directory structure presented in section 2.2.1 above. This has been addressed in version 4.6. Users with such an environment can set the following configuration parameters:

$silk_data_directory = (user's SiLK rootdir);
$site_config_file = (user's silk.conf file)
$ipfix_default_device = "Site";

'Site' is a special device_name that when used will allow a "no device" environment and prevent the user from having to select a device each. If FlowViewer sees that this variable is non-empty, it will automatically adjust the device_name pulldown and selection appropriately. Then the user can control all SiLK data selection via the text fields (see section 2.1.1 above) together with their default definitions in the FlowViewer_Configuration.pm file.

41. My User Interface is all garbled - why?

Please verify that you have copied the FlowViewer.css file from distribution you're working with into the $reports_directory.

Appendix A – Typical Cisco Flexible Netflow (ASR 1000)

For a single export to Primary NMS:

```
flow exporter PRIMARY_NMS
 description FNF export to Primary NMS
 destination 192.168.100.100
 source Loopback0
 transport udp 9996
 template data timeout 60
!
flow monitor MONITOR_V4
 description IPv4 netflow monitor
 record netflow ipv4 original-input
 exporter PRIMARY_NMS
 cache timeout active 900
 cache entries 200000
!
flow monitor MONITOR_V6
 description IPv6 netflow monitor
 record netflow ipv6 original-input
 exporter PRIMARY_NMS
 cache timeout active 900
 cache entries 200000
!
!For each interface ....
!
interface GigabitEthernet0/0/0
 ip flow monitor MONITOR_V4 input
 ipv6 flow monitor MONITOR_V6 input
!
interface GigabitEthernet1/0/0
 ip flow monitor MONITOR_V4 input
 ipv6 flow monitor MONITOR_V6 input
```

For a second export to Backup NMS:

```
flow exporter PRIMARY_NMS
 description FNF export to Primary NMS
 destination 192.168.100.100
 source Loopback0
 transport udp 9996
 template data timeout 60
!
flow exporter BACKUP_NMS
 description FNF export to Backup NMS
```

```
 destination 192.168.200.100
 source Loopback0
 transport udp 9997
 template data timeout 60
!
flow monitor MONITOR_V4
 description IPv4 netflow monitor
 record netflow ipv4 original-input
 exporter PRIMARY_NMS
 exporter BACKUP_NMS
 cache timeout active 900
 cache entries 200000
!
flow monitor MONITOR_V6
 description IPv6 netflow monitor
 record netflow ipv6 original-input
 exporter PRIMARY_NMS
 exporter BACKUP_NMS
 cache timeout active 900
 cache entries 200000
!
!For each interface ....
interface GigabitEthernet0/0/0
 ip flow monitor MONITOR_V4 input
 ipv6 flow monitor MONITOR_V6 input
!
interface GigabitEthernet1/0/0
 ip flow monitor MONITOR_V4 input
 ipv6 flow monitor MONITOR_V6 input
```

Appendix B – Transitioning flow-tools created FlowMonitors to SiLK

At some point you may wish to switch a device from exporting netflow version 5 (or v1, or v7) to version 9 or IPFIX. However you have many FlowMonitors that have been created using a flow-tools capture of v5 netflow and you would like to continue these. This can be done by the following steps. In the example below, a device called router_1 has been exporting v5 for many years and will now be exporting v9

Example of re-configuring FlowViewer for a new v9 export from an existing device

1. Stop the active flow-capture associated with the device you are configuring to IPFIX. This is needed also because SiLK (rwflowpack) requires a clean port with nothing listening on it.

2. Rename the existing capture directory (e.g., >mv /flows/router_1 /flows/router_1_v5)

3. Change the device name in @devices to "router_1_v5"

4. Create a new capture directory (e.g., >mkdir router-1)

Note: SiLK does not like underscores – you must instead use dashes in your device names

5. Start up the rwflowpack script for "router-1" (see above)

6. Stop and restart FlowMonitor_Collector and FlowMonitor_Grapher

Notes:

1) The FlowMonitor_Collector in the FlowViewer version 4.6 distributions will convert all FlowMonitor_Collector filters into the new format for SiLK processing.

2) For archived FlowMonitors that were created by flow-tools and whose device is now an IPFIX exporting device (e.g., v9), they will be converted with the first FlowMonitor_Collector run after being restarted.

3) When a filter is converted from flow-tools to SiLK, the flow-tools information is retained so a switch back should be as simple as moving the device name from @ipfix_devices to @devices. This has not been tested however.