



# Global Information Assurance Certification Paper

Copyright SANS Institute  
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

## Interested in learning more?

Check out the list of upcoming events offering  
"Implementing and Auditing CIS Controls (Security 566)"  
at <http://www.giac.org/registration/gccc>

# Automating Provisioning of NetFlow Analyzers

*GIAC (GCCC) Gold Certification*

Author: Sumesh Shivdas, msumesh@hotmail.com

Advisor: Adam Kliarsky

Accepted: September 7, 2016

## Abstract

NetFlow is an embedded instrumentation within Cisco IOS software (Introduction to Cisco IOS NetFlow). NetFlow tracks every network conversation and thus provides insight into the network traffic. Third party NetFlow analyzers are available to store, analyze, alert and report on the NetFlow data. NetFlow analyzers allow users to create custom alerts and reports based on the network traffic. To maximize the benefits from custom alerting and reporting the analyzers must be configured with details of the network environment. Manual configuration of the analyzer can soon be out of sync with the actual setup thus creating false negatives and false positives. This paper proposes an option to automate the configuration of the NetFlow analyzer from a central repository.

## 1. Introduction

NetFlow is rapidly gaining popularity among security practitioners. This is mainly due to its ability to identify and alert on network-based anomalies, analyze security incidents, and minimal overhead to process and store information. In a research conducted by Hewlett Packard [9]; 47.3% of the responders include “NetFlow” as a data source regularly monitored by their security operations Center. This paper intends to use NetFlow to detect and report non-compliance to Center for Internet Security (CIS) version 6, specifically the six controls defined in section 2 of the document.

Organizations using NetFlow based monitoring typically have three main components a NetFlow exporter, a NetFlow collector and a NetFlow analyzer. The exporters are usually network gears like routers, switches, firewalls which can be configured to export the NetFlow data to one or more NetFlow collectors. The NetFlow collector receives the NetFlow data via User Datagram Protocol (UDP). The collector does pre-processing like de-duplication of flows and stores the flow information. The analyzers can analyze the flow information and can generate alerts and report any anomaly based on the network traffic metadata of NetFlow. A NetFlow record will at the minimum have the following metadata related to any network traffic between two IP-based devices

- Source IP address
- Destination IP address
- IP protocol
- Source port
- Destination port
- Number of Bytes and Packets
- Timestamp
- ICMP type and code
- TCP flags for TCP flows

Third party vendors like Riverbed, Lancope, Plixer, ManageEngine, and Solarwinds have developed NetFlow analyzer tools. These analyzers utilize NetFlow records and provide users capability to store and analyze the network traffic. The analyzers provide the ability to create alerts based on the traffic and generate reports. Most of the products include out of the box security rules to trigger alarms. These typically include protocol tunneling, port and host scan detection, as well as DoS (denial of service) identification. These products also allow the end-users to define custom security alerting policies. To optimally use the "user defined" security alerting and reporting feature, the user must provision the NetFlow analyzer tool with details of the user's environment. These configurations also enhance the reporting capability. For example, the user can now create reports based on host/application name instead of IP/port.

Standard intelligence feeds like blacklisted IP addresses, known file transfer sites, known e-mail exfiltration websites, known command and control sites, known bad DNS servers can be configured in the NetFlow analyzers to generate alerts and reports of any traffic to and from these sites.

Environment specific provisioning can include some of the following attributes:

- Application fingerprint
- Ports and protocols used
- Grouping of hosts based on its role, location, network segment
- Application(s) hosted
- Data sensitivity
- System owner and type

The customization can include any attributes that can be used to create a rule based on the network traffic. NetFlow analyzers provide GUIs (Graphical User Interfaces) for administrators to provision environment specific configuration. In most organizations, the configuration - if manually provisioned - can quickly become out of sync with the central repository. This paper proposes to automate the environment specific provisioning on a regular basis from the central repository.

## 2. Critical Controls

Center for Internet Security (CIS) releases a prioritized list of twenty security controls, named as Critical Security Controls (CSC). Cyber security experts from around the world contribute to the development and evolution of these recommendations. Implementing the recommended security controls can dramatically reduce the exposure of the organization to cyber-attacks.

This document focuses on using NetFlow analyzer as a tool to address portions of the following CIS Security Controls.

### 2.1. CSC #1 Inventory of Authorized and Unauthorized Devices

An accurate inventory of the hardware is one of the most important security hygiene. The idea behind the control is for organizations to know their assets and manage them. An unauthorized device must not be allowed in the network. Unauthorized devices must be detected and isolated as soon as possible.

NetFlow can passively detect any IP device in the network. Routers and switches will export the metadata for all network conversations and forward it to the NetFlow collector. IP addresses are available from the source and destination IP address fields in the NetFlow record. The NetFlow analyzer will use the flow information from the collector and generate an alert if it detects any unknown IP device (not configured in the analyzer) in the network.

### 2.2. CSC #3 Secure Configuration of Hardware and Software

The goal of this control is to ensure that the organization has hardened configuration for hardware and software configuration for their servers, workstations, and other network devices. The control recommends automated monitoring and alerting on unexpected open ports or use of insecure protocols like telnet, VNC, RDP, and other insecure protocols.

NetFlow records contain the network layer-4 attributes, source and destination ports. This information from the NetFlow can be used by the NetFlow analyzer to detect and report on any new listening port in the Network. The source and destination ports can also be used to alert on the usage of any

insecure protocol based on the standard port numbers used for such communications.

### **2.1. CSC #8 Malware Defenses**

The goal of this control is to detect and contain the installation, execution, and spread of Malware.

NetFlow can be used to detect malware based on its behavior or anomaly in the network behavior. Communication attempts or actual communication from IP-devices to known command and control or a blacklisted site can be alerted based on the IP address stored in the NetFlow records. Other typical malware behavior like a lateral movement across the network, host or port scans originating from unexpected network devices, attempt to use an unauthorized DNS server, network traffic pattern deviations indicating possible data exfiltration's, Denial of Service or Distributed Denial of Service can be alerted.

### **2.2. CSC #9 Limitations and Control of Network Ports, Protocols, and Services**

The goal of this control is to monitor and manage the use of ports, protocol, and services on any networked device. The intention is to remediate the use of unknown port or service immediately to reduce the attack surface by removing any potentially vulnerable services or remote listening ports.

The metadata available from NetFlow contains all the information needed to detect the network ports and protocols used. A NetFlow analyzer configured with the known ports it can detect the anomaly and alert as soon as it sees any attempted or successful flow reported by the routers or switches. A NetFlow analyzer can also be configured to alert if it detects critical network services like DNS, Mail, or NTP is running on unexpected IP devices. NetFlow based detection can complement the standard network scanner based on local listening port detection. Network scanners or local port scanners have to run at scheduled intervals. It can be several weeks or months of elapsed time between two scans in a large organization. It will be months before an open port is detected if NetFlow is not used to complement the detection. Advanced malicious software can evade detection by port scanners by closing the ports during the scan window. This evasion is not possible if NetFlow is used to complement the port scans.

### 2.3. CSC #12 Boundary Defense

The goal of this control is to detect, prevent and correct any anomalous information transferring across network boundaries at different trust level. The focus in this control is the data transferred across the trust boundary which can damage the security posture of the organization.

NetFlow adds another layer of detection for this control. NetFlow adds another layer of detection for this control. A NetFlow analyzer configured with flow whitelist across the trust boundaries can alert on any network anomaly. NetFlow analyzers can alert based on the number of packets, bytes of data transferred, rate and thresholds of ingress or egress connections, long running TCP sessions, egress or ingress connections from blacklisted IP to or from the DMZ systems, or any other network behavior anomaly.

### 2.4. CSC #13 Data Protection

This critical control is about tools and processes to prevent data exfiltration, mitigate the effect of exfiltrated data and ensure the confidentiality and integrity of sensitive data.

NetFlow does not have access to the actual network packets traversing the network. It only has the metadata of the network traffic. NetFlow cannot inspect traffic and hence cannot look for patterns in the network traffic to detect data exfiltration however it can detect anomalous traffic. The NetFlow analyzer can learn the traffic behavior over time and can alert on any deviation from the standard traffic. NetFlow can also be configured to generate an alert if it detects traffic to a known file transfer or e-mail exfiltration websites.

## 3. Background – Network Behavior Setup

This paper uses Riverbed Cascade Express, a NetFlow analyzer tool, as an example and will focus on the automation tasks using the Riverbed Cascade Express. The automation concepts apply to any other NetFlow analytics tool. The figure below depicts a screen for querying network traffic from Cascade Express. The lack of any custom grouping and application definitions limits the traffic reporting or alerting based on the ports, protocols or IP address instead of using names.

Trace: Dashboard » Traffic

Traffic

Hosts Interfaces Applications Advanced

☐ Report Criteria (default by Host)

Applications:  Browse...

Protocols or ports:  Browse...

Servers, subnets or groups:  Browse...

Clients, subnets or groups:  Browse...

Quality of Service (QoS):  Browse...

Time frame: ☐ Starting  Hour(s) ago

☐ From:

To:

Data resolution:

Report by:  ☐ Break out MAC-IP assignments

Report Format

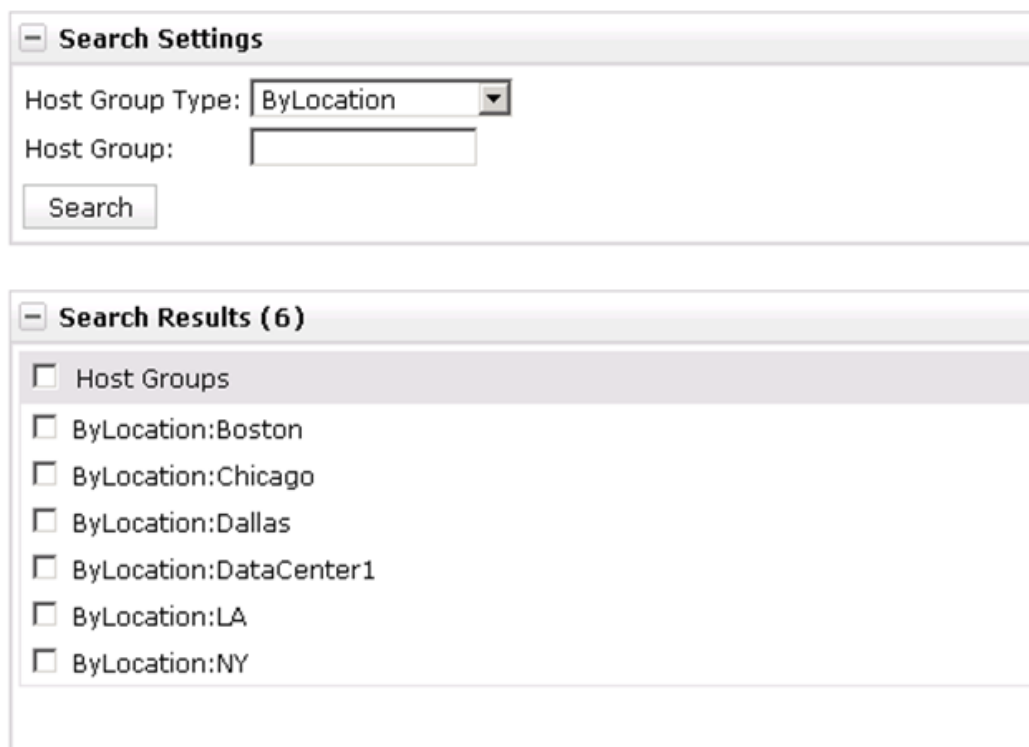
Figure 1: Traffic Report

If the user has created custom groups based on the environment, then these can be used in reporting and alerting policies. As shown in the figure below the user can run on the traffic report based on the selected location. The user can create groupings based on the environment specific requirement for example

- Function of the servers (mail, dns, web, db, etc.),
- Customer (Staples, OfficeDepot, Target, etc.),
- Location of host in the network (DMZ, ServerFarm, Workstations, etc.),



## Look Up Host Groups



**Search Settings**

Host Group Type: ByLocation

Host Group:

**Search Results (6)**

- ☐ Host Groups
- ☐ ByLocation:Boston
- ☐ ByLocation:Chicago
- ☐ ByLocation:Dallas
- ☐ ByLocation:DataCenter1
- ☐ ByLocation:LA
- ☐ ByLocation:NY

Figure 2: Host grouped 'ByLocation'

The screenshot in the following figures shows manual configuration options for host group type. Riverbed provides a graphical user interface for administrators to define the host, port groups, port names, application with layer seven fingerprints and layer four details.

New Group Type


---

**Group Type Information**

To create a view, provide a name. The name cannot contain spaces.

Group type name:

Description:

☐ Available to Dashboard 

Profiler can be configured for up to 10 different group types for reporting, Dashboard views are limited to 4 group types. You have 3 more available for dashboard viewing.

---

**Identify Groups in this Group Type**

A host can appear in only one group (within this group type). A host must be contained within the set of 'Inside Addresses'. To create a group, specify a group definition and name, using the format shown below. Click Import to import group definitions from a file ([view the proper file syntax](#)).

```
# This is a comment - Below are some examples for groups.
# Use CIDR notation, a space, and a group name or
# Use IP address and subnet mask in dotted decimal format, a space, and a group name
# Examples:
#
#192.168.0.0/16 Boston
#192.168.0.100/255.255.0.255 Phoenix
#
```

Figure 3: Manual Definition of Host Group Type and Hosts

Figure 3: Manual Definition of Host Group Type and Hosts

The screenshot in the following figures shows riverbed administrator manually configuring the port groups and port names.

Trace: Dashboard » Applications » Manage Host Group Types » Traffic » Port Groups

Port Groups ⓘ

**New Port Group**

Name:

Services:   
(e.g., tcp/21-23, udp/21-23)

Figure 4: Manual Definition of Ports Groups

As previously mentioned, the main issue with the manual provisioning of the tool is it is hard to manage and can quickly be out of sync with the central configuration database. Configuration not aligned with existing setup makes the system unreliable. Many security tools, such as Security Incident and Event Managers (SIEMs), Intrusion Detection Systems (IDS), Network Intrusion Prevention Systems (NIPS), Network Behavior Analysis (NBA) and other systems require similar configuration. Configuration items described above like host groups, hosts, port groups, port names, and application (with its corresponding signature) must be in the Central Configuration Management Database (CMDB). Automated provisioning tools must use the information in CMDB to generate specific provisioning information as required by the product. Automation of provisioning will reduce the manual provisioning errors and ensure that the tools are in sync with the latest information from the CMDB.

NetFlow information, supported in most of the routers, switches, and firewalls, with the automated configuration of NetFlow Analyzer, helps organizations continuously monitor six of the top 20 Critical Controls. NetFlow based monitoring and alerting is near real time with minimal overhead and can be setup easily. NetFlow based alerts provide Security operations another layer of visibility into their environment to efficiently detect and investigate a security incident.

## 4. Proposed Automation

The flow chart below shows the recommended setup for automation.

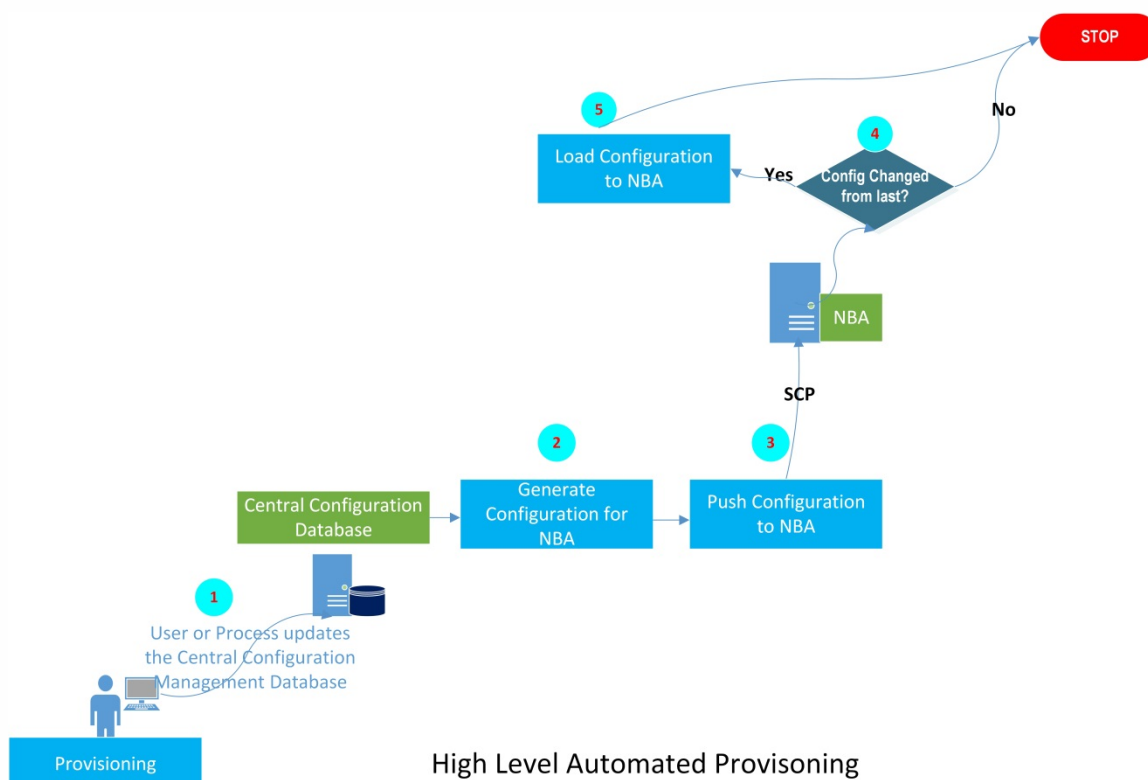


Figure 5: Proposed Automation flow

1. A user or a process triggers a provisioning process which updates the central configuration database
2. A process running on the central configuration server frequently checks for any updates to the database. If it detects an update, it dumps configuration files in the format required by the NetFlow Analyzer tool (server groups, application mapping, ports, and protocols, etc.)
3. The process copies the file to the NetFlow Analyzer server using SSH PKA-based authentication using a low privileged account in a particular directory
4. A Process running on the NetFlow analyzer polls for any updated provisioning files then it compares with previous and checks for any update
5. If the provisioning files are updated, it loads the new information into the NetFlow Analyzer using the interface provided

#### 4.1. Host Group Update

Riverbed expects the host grouping to be in the following format. The name of the file containing this information will be the unique host group with an

extension of "\_hostgroup.dat". In our example, we are classifying the systems based on their function. The file will be named ByFunction\_hostgroup.dat (<Group Type>\_hostgroup.dat)

```
192.12.232.0/30 Mail
192.12.52.0/30 Web
192.15.92.0/30 Web
10.121.0.32/29 DB
10.121.0.64/29 DB
10.125.1.96/30 LOG
```

The sample code to load the above group file will look like below

```
$PROVISIONING_DIR=/var/mazu/provisioning
$MAZU_BIN_DIR=/usr/mazu
$MAZU_HOSTGROUPING=$MAZU_BIN_DIR/mazu-hostgrouping
for file in $(/bin/ls -1 $PROVISIONING_DIR/*_hostgroup.dat)
do
    $HOST_GROUP=$(echo ${file%%*_hostgroup.dat})
    $PROCESSED_FILE=$PROVISIONING_DIR/$file
    $comment="Auto provisioning of $HOST_GROUP:${/bin/date}"
    $MAZU_HOSTGROUPING -o add --group-type-name $HOST_GROUP --description "$comment" --definitions-file $PROCESSED_FILE
done
```

The code above will update the Riverbed profiler with the latest host grouping. Any reporting or alerting can use these automatically provisioned group names.

## 4.2. Port Group Update

The port grouping must be in the following XML format. Each port group containing a unique id and port group name.

```
<?xml version="1.0" encoding="UTF-8" ?>
<portgroups version='1'>
  <portgroup id='1' name='web'>
    <protoport proto='6' port='3128' />
    <protoport proto='17' port='443' />
    <protoport proto='6' port='443' />
    <protoport proto='17' port='80' />
    <protoport proto='6' port='80' />
  </portgroup>
  <portgroup id='2' name='mail'>
    <protoport proto='6' port='1109' />
    <protoport proto='17' port='995' />
    <protoport proto='6' port='995' />
    <protoport proto='6' port='465' />
    <protoport proto='17' port='220' />
    <protoport proto='6' port='220' />
    <protoport proto='17' port='209' />
    <protoport proto='6' port='209' />
    <protoport proto='17' port='174' />
    <protoport proto='6' port='174' />
    <protoport proto='17' port='143' />
    <protoport proto='6' port='143' />
    <protoport proto='17' port='110' />
  </portgroup>
</portgroups>
```

```
<protoport proto='6' port='110' />
<protoport proto='17' port='109' />
<protoport proto='6' port='109' />
<protoport proto='17' port='25' />
<protoport proto='6' port='25' />
</portgroup>
</portgroups>
```

The port grouping file can be an individual file with all the ports defined, or it could be multiple based on the categorization of the application within the customer environment. The suffix "\_portgroup.xml" identifies the portgroup files. Riverbed interface "mazu-portgroup" must be used to upload the port definition.

Below you can see the sample code to automate the upload of the definitions

```
$PROVISIONING_DIR=/var/mazu/provisioning
$MAZU_BIN_DIR=/usr/mazu
$MAZU_PORTGROUPING=$MAZU_BIN_DIR/mazu-portgrouping
for file in $( /bin/ls -1 $PROVISIONING_DIR/*_portgroup.xml)
do
    $PORTGROUP_FILE=$PROVISIONING_DIR/$file
    $MAZU_PORTGROUPING -load $PORTGROUP_FILE
done
```

### 4.3. Application Group Update

Riverbed expects the application grouping file as XML. The application groups can use the port names defined in the port groups. Below is the sample of a mapping file

```
<?xml version="1.0" encoding="UTF-8" ?>
<AppMappings version='2'>
  <AppMapping id='1' app='CIFS' enabled='true' priority='1' override='1' >
    <hosts>
      <cidr value='0.0.0.0/0' />
    </hosts>
    <ports>
      <protoport proto='tcp' port='139' />
      <protoport proto='tcp' port='445' />
    </ports>
  </AppMapping>
  <AppMapping id='2' app='FTP' enabled='true' priority='2' override='1' >
    <hosts>
      <cidr value='0.0.0.0/0' />
    </hosts>
    <ports>
      <protoport proto='tcp' port='20' />
      <protoport proto='tcp' port='21' />
    </ports>
  </AppMapping>
</AppMappings>
```

The sample code to automate the load of the application group file(s) is shown below

Sumesh Shivdas msumesh@hotmail.com

```
$PROVISIONING_DIR=/var/mazu/provisioning
$MAZU_BIN_DIR=/usr/mazu
$MAZU_APPMAPPING=$MAZU_BIN_DIR/mazu-appmappings
for file in $(/bin/ls -1 $PROVISIONING_DIR/*_appmappings.xml)
do
    $APP_MAPPING_FILE=$PROVISIONING_DIR/$file
    $MAZU_APPMAPPING -load $APP_MAPPING_FILE
done
```

## 4.4. Rules Update

The XML format can be used to define the rules in Riverbed. The rules can use the application mapping, host group(s), and the port groups defined.

Here is a sample rule. The rule will trigger when the number of active connection to the user-defined "web" port groups exceeds the maximum threshold of 5,000 in a minute.

```
<?xml version="1.0" encoding="UTF-8" ?>
<rules version='7'>
<rule id='1598' name='New Connection Rate to Web too High' s1_role='0' direction='0' time_start='0' time_end='86399'
threshold='5000' type='upper'
unit='12' period='1' duration='60' severity='100' deleted='' enabled='true' rule_type='0' >
<description>Higher than Expected new connections to the web servers</description>
<notification type='low' recipient='1' recipient_name='Default' />
<notification type='med' recipient='1' recipient_name='Default' />
<notification type='high' recipient='1' recipient_name='Default' />
<set_a count='aggregate' negated='false'>
</set_a>
<set_b count='aggregate' negated='false'>
<group name='Web' type_name='ByFunction' type='100' id='1' />
</set_b>
<ports negated='false'>
<portgroup name='web' id='1' />
</ports>
<days>
<day value='Sunday' />
<day value='Monday' />
<day value='Tuesday' />
<day value='Wednesday' />
<day value='Thursday' />
<day value='Friday' />
<day value='Saturday' />
</days>
<apps negated='false'>
</apps>
<interfaces negated='false'>
</interfaces>
<interfaces_path negated='false'>
</interfaces_path>
<qos_set negated='false'>
</qos_set>
</rule>
</rules>
```

The sample code to automate the load of the user defined rule definitions in riverbed profiler is shown below

```
$PROVISIONING_DIR=/var/mazu/provisioning
```

Sumesh Shivdas msumesh@hotmail.com

```

$MAZU_BIN_DIR=/usr/mazu
$MAZU_RULES=$MAZU_BIN_DIR/mazu-rules
for file in $(ls -1 $PROVISIONING_DIR/*_rules.xml)
do
    $RULES_FILE=$PROVISIONING_DIR/$file
    $MAZU_RULES -load $RULES_FILE
done

```

## 5. Solution Demonstration

### 5.1. Grouping Auto Updated

The administrator and other users can see the updates reflected in the configuration after the automatic upload task completes. Any new hosts, group, ports, applications will be visible and will be available to be used by the existing or new reporting or alerting. Performing manual configuration of any network behavior analysis tool is slow and error prone. Most of the security and Operations tool require the information on the Assets being managed or monitored by them. Automating the provisioning of these devices from the central configuration management database makes the process quick and reliable.

Edit Group Type **ByFunction**

+
Group Type Information

-
Identify Groups in this Group Type

A host can appear in only one group (within this group type). A host must be contained within the set of 'Inside Addresses'. To create a group, specify a group definition and name, using the format shown below. Click Import to import group definitions from a file ([view the proper file syntax](#)).

192.12.232.0/30 Mail  
192.12.52.0/30 Web  
192.15.92.0/30 Web  
10.121.0.32/29 DB  
10.121.0.64/29 DB  
10.125.1.96/30 LOG

Import...

OK
Cancel

Figure 6: Updated Group by Function seen by Riverbed

Sumesh Shivdas msumesh@hotmail.com

© 2016 The SANS Institute

Author retains full rights.



## 5.2. Using Group in User Query

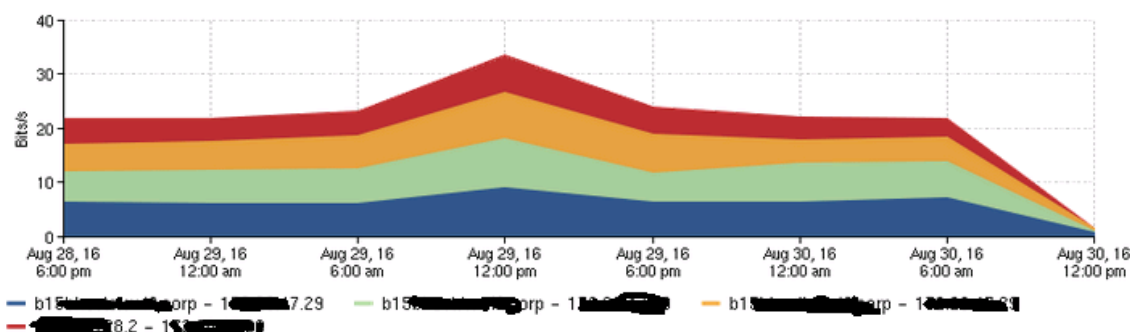
The picture below shows a query with a group "ByFunction:Web". It will report the traffic to all the hosts in the Web group using port tcp/443. The group ByFunction:Web is defined automatically using the information from the central configuration database. The end-user does not have to worry about the hosts included in the collection definition; the automation ensures that the system reflects the latest available information from the central database.

The screenshot shows the 'Traffic' tab in the NetFlow Analyzer interface. The breadcrumb trail is 'Trace: Dashboard » Manage Host Group Types » ByFunction » Traffic'. The 'Applications' sub-tab is selected. Under 'Report Criteria', the 'Servers, subnets or groups' field is populated with 'ByFunction:Web' and is highlighted with a red box. Other fields include 'Applications' (empty), 'Protocols or ports' (tcp/443), 'Clients, subnets or groups' (empty), and 'Quality of Service (QoS)' (empty). To the right, there are 'Browse...' links for each field. The 'Time frame' section shows 'Starting' at 1 hour(s) ago, with 'From' and 'To' dates set to Jul 10, 2016 at 10:39 AM and 11:39 AM respectively. The 'Data resolution' is set to 'automatic'. The 'Report by' dropdown is set to 'Host Pairs with Ports', and the 'Break out MAC-IP assignments' checkbox is unchecked. At the bottom, there are 'Run now' and 'Run in background...' buttons.

Figure 7: Query using the "ByFunction:Web" group

The screenshot below shows a part of the output from the query above. The figure shows multiple servers from the "ByFunction:Web" group the user selected. The user neither had to remember and select individual hosts or IP addresses nor had to worry about the number of servers in the group.

## Top 10 Host Pairs by Avg Bits/s



## Host Pair 1 - 4 of 4

Server	Server Group	Client	Client Group	Avg Bits/s ↓	Avg Packets/s	Avg Active Connections/s	%
b15...corp		1...		1.98 (29%)	< 0.01 (29%)	< 0.01 (28%)	
b15...corp		1...		1.93 (28%)	< 0.01 (28%)	< 0.01 (28%)	
b15...corp		1...		1.75 (26%)	< 0.01 (25%)	< 0.01 (25%)	
1...8.2		1...		1.21 (18%)	< 0.01 (18%)	< 0.01 (19%)	
Total				6.87 (100%)	< 1 (100%)	< 0.01 (100%)	

Figure 8: Query output automatically showing all servers from "ByFunction:Web" group

### 5.3. Using Group in Alerting

The following screens show the creation of user-defined rule in Riverbed profiler. The example here uses the automatically created "ByFunction:Web" group and the "web" port group name. The rule will trigger an alert, if there are more than 5000 active connections on any days of the week to the web servers on the "web" port. The event will be assigned severity of "100" and forwarded to the SIEM, which is the default action.

**Host Policy**

**▼ Policy Identification / Schedule**

Name:  Days to run rule: ☒ Mon ☒ Tue ☒ Wed ☒ Thu  
 Description:  ☒ Fri ☒ Sat ☒ Sun  
 Start time:  End time:   
☒ Enabled Select Time Zone:  ⓘ

**▼ Hosts / Groups**

Host/Group A:  ☒ Any ☐ Within ☐ Outside  
 Role:  [Browse...](#)  
 Statistics:

Host/Group B:  ☐ Any ☒ Within ☐ Outside  
 Role:  [Browse...](#)  
 Statistics:

**▼ Applications / Ports**

Applications:  ☒ Any ☐ Within ☐ Outside  
[Browse...](#)

Protocols/Ports:  ☐ Any ☒ Within ☐ Outside  
[Browse...](#)

**► Reporting Interfaces / Quality of Service (QoS)**

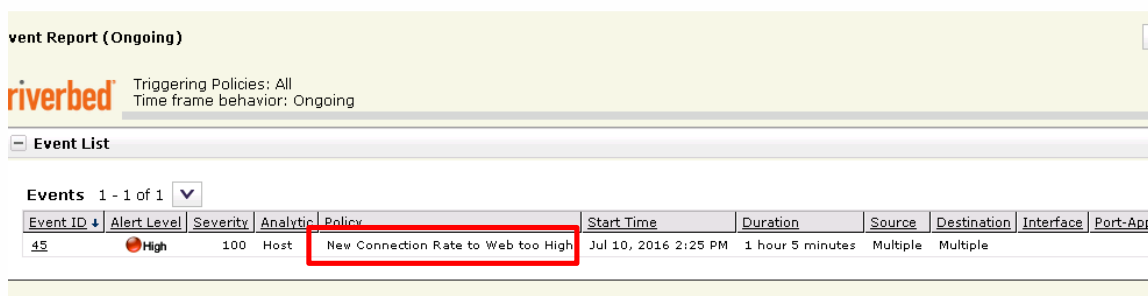
**► Interfaces in Network Path**

**▼ Threshold**

Trigger:      
 Direction:   
 Duration:  hours  minutes  
 Severity:   
 Notification: Low  Medium  High

Figure 9: User defined rule using "ByFunction:Web" group

The screenshot below from the alert window of the riverbed profiler shows a security event generated from the "New Connection Rate Too High" rule we defined above.



The screenshot shows the Riverbed Event Report interface. At the top, it says 'Event Report (Ongoing)' and 'riverbed'. Below that, it indicates 'Triggering Policies: All' and 'Time frame behavior: Ongoing'. The main section is titled 'Event List'. Under 'Events', it shows '1 - 1 of 1'. A table lists the event details:

Event ID	Alert Level	Severity	Analytic	Policy	Start Time	Duration	Source	Destination	Interface	Port-App
45	High	100	Host	New Connection Rate to Web too High	Jul 10, 2016 2:25 PM	1 hour 5 minutes	Multiple	Multiple		

Figure 10: User-defined policy using "ByFunction:Web" group alert

Alerting for "Insecure" flows, for example will include a user defined group named say "InsecureProtocolGroup". This group will include the telnet, remsh, rlogin ports. The rule will be to comparing the number of bits > 1 (0 is not allowed in riverbed) from any to any host/group during any time of the day, seven days a week. The "InsecureProtocolGroup" will be a configuration item in the CMDB will be used to create the rule. If the InsureProtocolGroup is updated to include http and ftp in the CMDB, the next automated update will ensure that the riverbed rule includes the latest updates and alerts for any violation.

The rule above for monitoring and alerting based on abnormal active connections to the Web servers in DMZ is an example of detective control for CSC #12 - Boundary Defense. It is looking at the network behavior (rate of new connections higher than normal) and creating an alert which will be forwarded to SIEM for further investigation by Security Operation Center. The rule to monitor use of insecure protocols covers the detection of CSC #2 – Secure configuration of Hardware and Software. It will alert on any use of insecure flow in the network. Similarly a rule to alert on traffic to or from the blacklisted IP, known malware, or known command and control can be used to detect controls in CSC #8 – Malware defenses. Any traffic destined to known file transfer site, e-mail exfiltration sites can be alerted as detection for CSC #13 – Data protection. Alerts generated on seeing a flow to a new network device or a new listening port covers detection for control #1 – Inventory of Authorized and Unauthorized devices and control #9 – Limitation and control of Network ports, protocols, and services. As we can see NetFlow can be used to continuously monitor many critical security controls, in near real time.

## 6. Conclusion

The solution proposed automates the provisioning of the NetFlow analyzer groups using the central repository. In an environment which is dynamic or has a lot of host's, automation is a must to avoid the configurations being out of sync. Riverbed also provides utilities for updating the information on the port groupings, application mappings and even creation of custom user defined rules for alerting. If the central database contains detailed information about all the allowed network flows, then this solution can be extended to automate the generation of customer defined flow whitelisting rule. The NetFlow analyzer will alert on any flows which are anomalous. This rule can be very useful to detect any lateral or abnormal flows by malicious users or programs communicating using an unknown protocol or host. NetFlow analyzer, in conjunction with automation, is a great tool to implement detective controls to address Center for Internet Security control #9: Limitation and Control of Network Ports, Protocols, Services as well as control #12: Boundary Defense.

## References

- Introduction to Cisco IOS NetFlow. (n.d.). Retrieved from <http://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/ios-NetFlow>
- CIS Top 20 Controls Version 6.0. Retrieved July 6, 2016, from <https://www.cisecurity.org/critical-controls>
- Successfully delivering mission critical, performance- sensitive services and applications with NetFlow. Retrieved July 6, 2016, from [http://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/ios-NetFlow/prod\\_white\\_paper0900aecd80406232.html](http://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/ios-NetFlow/prod_white_paper0900aecd80406232.html)
- ROI for Flow Monitoring Solutions: Five Use Cases. Retrieved July 6, 2016, from <https://cdn.plixer.com/support/whitepapers/ROI-%20For-Flow-Monitoring-Solutions.pdf>
- Successful ways to use NetFlow. Retrieved July 6, 2016, from [https://cdn.plixer.com/support/whitepapers/wp\\_successful\\_NetFlow.pdf](https://cdn.plixer.com/support/whitepapers/wp_successful_NetFlow.pdf)
- Cisco ASA Guide to NetFlow Event Logging and Cyber Threat Detection. Retrieved July 6, 2016, from <https://cdn.plixer.com/support/whitepapers/cisco-asa-guide-to-NetFlow-security-event-logging-and-cyber-threat-detection.pdf>
- NetFlow – A case Study. Retrieved July 6, 2016, from [http://www.cisco.com/c/dam/en/us/products/collateral/ios-nx-os-software/ios-NetFlow/prod\\_case\\_study0900aecd80311fc2.pdf](http://www.cisco.com/c/dam/en/us/products/collateral/ios-nx-os-software/ios-NetFlow/prod_case_study0900aecd80311fc2.pdf)
- Manage Engine NetFlow Analyzer. Retrieved July 6, 2016, from <https://www.manageengine.com/products/NetFlow/>
- Lancope – Stealthwatch, NetFlow Traffic Analyzer. Retrieved July 6, 2016, from <https://www.lancope.com/>
- Solarwinds – NetFlow Traffic Analyzer. Retrieved July 6, 2016, from [http://www.solarwinds.com/NetFlow-traffic-analyzer?&CMP=KNC-TAD-GGL-SW\\_NA\\_X\\_PP\\_CPC\\_LD\\_EN\\_PRODB\\_DWA-NTA-X\\_X\\_X\\_X-X&kwid=OHslTwWT&gclid=CIC6p\\_7R2s0CFcNehgodpWMGzw](http://www.solarwinds.com/NetFlow-traffic-analyzer?&CMP=KNC-TAD-GGL-SW_NA_X_PP_CPC_LD_EN_PRODB_DWA-NTA-X_X_X_X-X&kwid=OHslTwWT&gclid=CIC6p_7R2s0CFcNehgodpWMGzw)
- Hewlett Packard Enterprise Security Research – Cyber Risk Report 2016. Retrieved July 6, 2016, from [https://www.thehaguesecuritydelta.com/media/com\\_hsd/report/57/document/4aa6-3786enw.pdf](https://www.thehaguesecuritydelta.com/media/com_hsd/report/57/document/4aa6-3786enw.pdf)
- Cascade profiler and Cascade Express User Guide. Retrieved July 6, 2016 from

[https%3A%2F%2Fsupport.riverbed.com%2Fbin%2Fsupport%2Fdownload%3Fdi%3D3nbecm52pjur8blokkp6jjul6g&usg=AFQjCNGNxVdeetDFyiiZd2ULNLN\\_JktxZg&bvm=bv.126130881,d.ZGg](https%3A%2F%2Fsupport.riverbed.com%2Fbin%2Fsupport%2Fdownload%3Fdi%3D3nbecm52pjur8blokkp6jjul6g&usg=AFQjCNGNxVdeetDFyiiZd2ULNLN_JktxZg&bvm=bv.126130881,d.ZGg)

©2016 SANS Institute, Author retains full rights.