Regular Expression Filtering, as the name indicates offers the option to identify patterns inside a packet across any part of the packet, including the packet payload. These patterns can be as simple as a static string at a user configured offset, or an extremely complex PCRE regular expression at a variable offset.

With the flexibility offered by regular expression-based filters, an operator can for example:

- Identify credit card numbers, social security numbers across user-level transactions
- Identify phone numbers exchanged across SIP packets
- As part of HTTP transactions:
  - Filter on URLs
  - Filter on patterns in the user-agents, PCRE-anchors to identify packets
- Filter on DNS queries for specific URLs
- Filter on source and destination addresses in FCoE packets
**Identify Social Security Numbers in User-level Transactions**

For an incoming traffic stream of packets, we are looking for packets that contain social security numbers. Once a match is detected, the packets would be forwarded to a monitoring tool for additional analysis.

Please note: operators can always match against a specific social security number pattern if that's what they are looking for.

```
# Configure ports
no traffic all noconfirm
port 1/1/x3 type network
port 1/1/x4 type tool
port 1/1/x1 type tool
# configure gigasmart gsops
gsgroup alias gsg1 port-list 1/1/el
gsop alias gsfil apf set port-list gsg1
# configure vport.
vport alias vpl gsgroup gsg1
# forward all traffic to 2'nd level map
map alias to_vp
to vpl
from 1/1/x3
# forward any traffic to vpl
rule add pass ipver 4
# maybe this will work?
# rule add pass macsrc 00:00:00:00:00:00 bidir
exit
# configure 2'nd level map to filter on string
map alias m1
use gsop gsfil
to 1/1/x1
from vpl
# Pattern match across the entire packet
gsrule add pass pmatch RegEx "^\d{3}-\d{2}-\d{4}$" 0..1750
exit
```

**Filtering on FCoE traffic**

The flexibility offered by regular expression-based filters can be used as an infrastructure to classify traffic streams with protocol headers that are typically unsupported on traditional TAP/SPAN aggregation devices.
In this example, we are using regular expression-based filters for filtering on src address in a Fibre Channel header.

```plaintext
# Configure ports
no traffic all noconfirm
port 1/1/x3 type network
port 1/1/x4 type tool
port 1/1/x1 type tool
# configure gigasmart gsops
gsalias gsg1 port-list 1/1/e1
gsop alias gsfil apf set port-list gsg1
# configure vport.
vport alias vp1 gsgroup gsg1
# forward all traffic to 2’nd level map
map alias to_vp
to vp1
from 1/1/x3
# forward FCOE traffic to vp1
rule add pass ethertype 8906
exit
# configure 2’nd level map to filter on regular expression
map alias m1
use gsop gsfil
to 1/1/x1
from vp1
# string match at a specific offset
# search for dst addr in FCOE packet
grule add pass pmatch string “\xff\xff\xff\xe0” 29
exit
```
Multi-Encap Filtering

In order to complement the mobility brought about by the virtualized server infrastructure, network virtualization overlays like VXLAN, VN-Tag, and NVGRE are being designed and implemented in data centers and enterprise environments. Across service provider environments, huge volumes of traffic are being tunneled over GTP. Gigamon’s Visibility Fabric™ provides the option of stripping out or removing these headers, thus providing visibility to monitoring tools that do not understand these overlays and encapsulation protocol. Now with adaptive packet filtering, this capability is further enhanced. Operators have the option of making forwarding decisions based on the encapsulation and inner packet contents.

With encapsulation awareness enabled by adaptive packet filtering, operators have multiple options to act on the packet, such as the flexibility to:

- Filter on encapsulation header parameters, Layer 2 – 4 parameters in the outer or inner headers (up to 5 layers of encapsulation) and in any combination. As an example:
  - Forward traffic specific to a subset of VXLAN IDs to one or more monitoring tools
  - Distribute traffic based on MPLS label values across one or more monitoring tools
- In combination with header stripping and/or tunnel decapsulation:
  - Decapsulate a header or terminate a tunnel and then make forwarding decisions based on the original packet
  - Implement “conditional” header stripping, based on encapsulation header parameters or inner/outer packet contents
    - Forward a subset of traffic “as-is” to monitoring tools that need these encapsulations for analysis
    - Alternatively strip out the outer headers/encapsulations and distribute traffic to monitoring tools that do not require these outer headers for analysis
- Since adaptive packet filtering is implemented as a second level map, operators can also implement overlapping rules where:
  - A copy of the traffic can be distributed across a group of monitoring tools
  - A refined subset from the same incoming stream is distributed across a different set of tools
Filtering on Subscriber Device IP (User-Endpoint IP or UE-IP)

Encapsulation awareness enabled by adaptive packet filtering allows mobile operators to filter on Layer 2 – 4 header parameters found in an encapsulated packet.

This facilitates operators to filter and forward traffic specific to a mobile subscriber device or a group of subscriber devices, identified by their IP addresses (User-Endpoint IP) to one or more monitoring tools. In this example, we are

- Identifying and forwarding traffic from / to a UE-IP of 1.1.1.1 to a monitoring tool connected to 1/1/x1
- Identifying and forwarding traffic from / to a UE-IP of 1.1.1.2 to a different monitoring tool connected to tool port 1/1/x4

In many cases, the GTP control sessions are low-volume and are useful in providing some level of visibility into the quality of experience of the subscribers. To this end, operators prefer to replicate the control sessions across all the monitoring tools, while filtering and forwarding a subset of the user-plane sessions to a subset of monitoring tools. The example below also illustrates configuration commands, leveraging the patented Flow Mapping® technology to replicate the GTP control sessions across all the monitoring tools involved in the traffic analysis.

```bash
# Configure ports
no traffic all noconfirm
port 1/1/x3 type network
port 1/1/x4 type tool
port 1/1/x1 type tool
# configure gigasmart gsops
gsgroup alias gsg1 port-list 1/1/el
gsop alias gsfil apf set port-list gsg1
# configure vport.
vport alias vpl gsgroup gsg1
# forward GTP-u traffic to 2’nd level map
map alias to_vp
to vpl
from 1/1/x3
rule add pass portsrc 2152
exit
vport alias vpl gsgroup gsg1
# Replicate GTP-c traffic to all the tools
map alias to_tool
to 1/1/x1,1/1/x4
from 1/1/x3
rule add pass portsrc 2123
exit

# configure 2’nd level map to filter on src/dst (bidirectional) ip 1.1.1.1
map alias m1
use gsop gsfil
to 1/1/x1
from vpl
gsrvrule add pass ipv4 src pos 2 value 1.1.1.1 255.255.255.255
gsrvrule add pass ipv4 dst pos 2 value 1.1.1.1 255.255.255.255
exit
# configure 2’nd level map to filter on src/dst (bidirectional) ip 1.1.1.2
map alias m2
use gsop gsfil
```
Filtering on Inner Layer 2 – 4 Parameters for Unrecognized Headers

The flexibility of encapsulation awareness enables filtering on encapsulated contents even if advanced packet filtering doesn’t recognize the outer encapsulation header. The example below illustrates a packet encapsulated in FabricPath headers. FabricPath headers, as shown in the figure below, are mac-in-mac headers that are currently not recognized by advanced packet filtering; however, operators can still filter and forward traffic flows based on Layer 2 – 4 parameters found in the encapsulated packets.

- Identifying and forwarding traffic from/to ip 1.1.1.1 in the inner / original packet to monitoring tool connected to tool port 1/1/x1
- Identifying and forwarding traffic from/to ip 1.1.1.2 in the inner / original packet to monitoring tool connected to tool port 1/1/x4

```
# Configure ports
no traffic all noconfirm
port 1/1/x3 type network
port 1/1/x4 type tool
port 1/1/x1 type tool
# configure gigasmart gsops
gsgroup alias gsg1 port-list 1/1/e1
gsop alias gsfil apf set port-list gsg1
# configure vport.
vport alias vp1 gsgroup gsg1
# forward fabric path packets to 2’nd level map
map alias to_vp
to vp1 from 1/1/x3
rule add pass ethertype 8903
exit
# configure 2’nd level map to filter on src/dst (bidirectional) ip 1.1.1.1
```
map alias m1
use gsop gsfil
to 1/1/x1
from vp1
gsrule add pass ipv4 src pos 1 value 1.1.1.1 255.255.255.255
gsrule add pass ipv4 dst pos 1 value 1.1.1.1 255.255.255.255
exit
# configure 2'nd level map to filter on src/dst (bidirectional) ip 1.1.1.2
map alias m2
use gsop gsfil
to 1/1/x4
from vp1
gsrule add pass ipv4 src pos 1 value 1.1.1.2 255.255.255.255
gsrule add pass ipv4 dst pos 1 value 1.1.1.2 255.255.255.255
exit

Filtering on Encapsulation Headers

Advanced packet filtering also provides the flexibility of filtering on specific parameters found in the encapsulation headers related to GTP, VXLAN, VN-Tag, just to name a few. Intelligent filtering across advanced encapsulation headers include:

• VXLAN ID,
• ERSpan ID,
• GRE Key,
• VN-Tag src/dst vif id, list id,
• VLAN ID in QinQ,
• MPLS labels
• GTP tunnel ID

GTP Tunnel ID-Based Filtering

The example given below illustrates filtering and forwarding traffic based on tunnel IDs that are included as part of the GTP user-plane messages. It also illustrates the concept of a shared collector where traffic that does not match any of the configured filters can be optionally sent out to. GTP control sessions are being forwarded to all the monitoring tools leveraging the power of Flow Mapping by filtering on Layer 4 UDP port 2132.

• For GTP-u
  – Filter and forward teid ranges 0x001e8480..0x001e8489 to monitoring tool connected to tool port 1/1/x1
  – Filter and forward teid ranges 0x001e8490..0x001e8499 to monitoring tool connected to tool port 1/1/x4
  – Forward the rest of the traffic to a shared collector

# Configure ports
no traffic all noconfirm
port 1/1/x3 type network
port 1/1/x4 type tool
port 1/1/x1 type tool
port 1/1/x2 type tool
# configure gigasmart gsops
gsgroup alias gsg1 port-list 1/1/e1
gsop alias gsfil apf set port-list gsg1
# configure vport.
vport alias vp1 gsgroup gsg1
# forward GTP-u traffic to 2’nd level map
map alias to_vpl
  to vpl
  from 1/1/x3
  rule add pass portsrc 2152
exit
map alias ctrl_to_tool
  to 1/1/x1..2,1/1/x4
  from 1/1/x3
  rule add pass portsrc 2123
exit

# configure 2'nd level map to filter on teid range 0x001e8480..0x001e8489
map alias m1
  use gsop gsfil
  to 1/1/x1
  from vpl
  gsrule add pass gtp gtpu-teid range 0x001e8480..0x001e8489 subset none
exit
# configure 2'nd level map to filter on teid range 0x001e8490..0x001e8499
map alias m2
  use gsop gsfil
  to 1/1/x4
  from vpl
  gsrule add pass gtp gtpu-teid range 0x001e8490..0x001e8499 subset none
exit

## Add collector for unmatched data
map-scollector alias scoll
  from vpl
  collector 1/1/x2
exit

**MPLS Label-based Filtering**

Multiprotocol Label Switching (MPLS) is a mechanism in high-performance telecommunications networks that directs data from one network node to the next based on short path labels rather than long network addresses, avoiding complex lookups in a routing table. The labels identify virtual links (paths) between distant nodes rather than endpoints.

MPLS is a scalable, protocol-independent transport. In an MPLS network, data packets are assigned labels. Packet-forwarding decisions are made solely on the contents of this label, without the need to examine the packet itself. This allows one to create end-to-end circuits across any type of transport medium, using any protocol.

However in the context of visibility nodes, traffic flows encapsulated in MPLS labels cannot be filtered and forwarded. With the wide-scale adoption of MPLS as a technology across enterprise and service provider environments, the ability to classify traffic flows based on MPLS labels would be a huge value add to granularly control the flow of traffic to the monitoring tools. Advanced packet filtering can be leveraged to filter and forward traffic flows based on MPLS label values. MPLS can stack multiple labels to form tunnels within tunnels. The flexibility of advanced packet filtering facilitates traffic classifications across up to 5 levels of MPLS label stacks in addition to the capability to filter and forward based on Layer 2 – 4 parameters found in the encapsulated packet. The example given below illustrates filtering and forwarding traffic based on MPLS labels.

- Filter and forward traffic flows specific to mpls label = 4 at the second level in the MPLS label stack to tool 1
- Filter and forward traffic flows specific to mpls label = 3 at the first level in the MPLS label stack to tool 2
# Configure ports
no traffic all noconfirm
port 1/1/x3 type network
port 1/1/x4 type tool
port 1/1/x1 type tool
# configure gigasmart gsops
gsgroup alias gsg1 port-list 1/1/e1
gsop alias gsfil apf set port-list gsg1
# configure vport.
vport alias vp1 gsgroup gsg1
# forward all traffic to 2’nd level map
map alias to_vp
to vp1
from 1/1/x3
# forward any traffic to vp1
rule add pass ipver 4
# maybe this will work?
# rule add pass macsrc 00:00:00:00:00:00 bidir
exit
# configure 2’nd level map to filter on mpls label
map alias m1
use gsop gsfil
to 1/1/x1
from vp1
gsrule add pass mpls label pos 1 value 4
exit
# configure 2’nd level map to filter on mpls label
map alias m2
use gsop gsfil
to 1/1/x4
from vp1
gsrule add pass mpls label pos 1 value 3
exit
Combining Advanced Packet Filtering with GigaSMART® Operations

Alternative to Second-level Map

Advanced packet filtering can also be combined with other GigaSMART functions including header stripping, packet slicing/masking, de-duplication and FlowVUE™. This allows network administrators and operators to perform a second layer of filtering in combination with the GigaSMART tool optimization and packet manipulation operations. In the example illustrated below, operators can distribute traffic to monitoring tools based on decapsulated contents, more specifically, after header stripping VXLAN:

- Identifying and forwarding traffic from/to ip 1.1.1.1 from the decapsulated packets to monitoring tool connected to tool port 1/1/x1
- Identifying and forwarding traffic from/to ip 1.1.1.2 in the decapsulated packets to monitoring tool connected to tool port 1/1/x4

Note: This can be applied to any protocol that is supported via header stripping, for example:
- GTP, VXLAN, ISL, MPLS, VLAN, VN-Tag, fabric-path
- This is also supported for Gigamon tunnel decapsulation

```bash
# Configure ports
no traffic all noconfirm
port 1/1/x3 type network
cr 1/1/x4 type tool
cr 1/1/x1 type tool

# configure gigasmart gsops
# configure gigasmart gsops
gsgroup alias gsg1 port-list 1/1/e1
gsop alias gsfil_vxlanhs apf set strip-header vxlan 0 port-list gsg1

# configure vport.
vport alias vp1 gsgroup gsg1

# forward VXLAN traffic to 2’nd level map
map alias to_vp
to vp1
from 1/1/x3
rul3 add pass portsrc 8472
exit

# configure 2’nd level map to filter on src/dst (bidirectional) ip 1.1.1.1
map alias m1
use gsop gsfil_vxlanhs
to 1/1/x1
from vp1
gsrul3 add pass ipv4 src pos 2 value 1.1.1.1 255.255.255.255
gsrul3 add pass ipv4 dst pos 2 value 1.1.1.1 255.255.255.255
exit

# configure 2’nd level map to filter on src/dst (bidirectional) ip 1.1.1.2
map alias m2
use gsop gsfil_vxlanhs
to 1/1/x4
from vp1
gsrul3 add pass ipv4 src pos 2 value 1.1.1.2 255.255.255.255
gsrul3 add pass ipv4 dst pos 2 value 1.1.1.2 255.255.255.255
exit
```
**Conditional Header Stripping**

Another use case that can be addressed leveraging the flexibility of advanced packet filtering would be the capability to header strip packets based on specific contents found across the packet including the inner packet contents. Since the advanced packet filtering rules are enforced before any other GigaSMART operation, operators can filter based on encapsulation protocol values and/or encapsulated (original) packet contents and apply conditional header stripping operations. The example illustrated below shows how an end-user can filter and strip out outer VXLAN headers for a subset of the traffic based on inner IP addresses, while sending the rest of the traffic "as-is" to monitoring tools for traffic analysis.

- Identifying and forwarding traffic from/to ip 1.1.1.1 in the inner / encapsulated packets to monitoring tool connected to tool port 1/1/x1 after header stripping VXLAN
- Identifying and forwarding traffic from/to ip 1.1.1.2 in the inner / encapsulated packets to monitoring tool connected to tool port 1/1/ x4 without stripping the VXLAN header

Note: This can be applied to any GigaSMART operation. While this example shows filtering based on inner packet contents, conditional GigaSMART operations can be applied by filtering on encapsulation headers as well.

**VXLAN Encapsulation**

<table>
<thead>
<tr>
<th>Outer MAC DA</th>
<th>Outer MAC SA</th>
<th>Outer IP 802.1Q</th>
<th>Outer IP DA</th>
<th>Outer UDP</th>
<th>VXLAN ID (24 Bit)</th>
<th>Inner MAC DA</th>
<th>Inner MAC SA</th>
<th>Optional Inner 802.1Q</th>
<th>Original Ethernet Payload</th>
</tr>
</thead>
</table>

Note: This can be applied to any protocol that is supported via header stripping, i.e.:

- GTP, VXLAN, ISL, mpls, mpls, vlan, vntag, fabric-path
- This is also supported for Gigamon tunnel decapsulation

```bash
# Configure ports
no traffic all nocofirm
port 1/1/x3 type network
port 1/1/x4 type tool
port 1/1/x1 type tool
# configure gigasmart gsops
gsop alias gsg1 port-list 1/1/e1
gsop alias gsfil_vxlanhs apf set strip-header vxlan 0 port-list gsg1
gsop alias gsfil apf set port-list gsg1
# configure vport.
vport alias vp1 gsgroup gsg1
# forward VXLAN traffic to 2’nd level map
map alias to_vp
to vp1
from 1/1/x3
# filter by UDP port #
rule add pass portsrc 8472
exit
```

Copyright © 2013 Gigamon. All rights reserved.
# configure 2'nd level map to filter on src/dst (bidirectional) ip 1.1.1.1
map alias m1
  use gsop gsfil_vxlanhs
to 1/1/x1
from vp1
gsrule add pass ipv4 src pos 2 value 1.1.1.1 255.255.255.255
gsrule add pass ipv4 dst pos 2 value 1.1.1.1 255.255.255.255
exit

# configure 2'nd level map to filter on src/dst (bidirectional) ip 1.1.1.2
map alias m2
  use gsop gsfil
to 1/1/x4
from vp1
gsrule add pass ipv4 src pos 2 value 1.1.1.2 255.255.255.255
gsrule add pass ipv4 dst pos 2 value 1.1.1.2 255.255.255.255
exit

Combining Advanced Packet Filtering with Gigamon Tunnel Decapsulation and Forward-specific IPs in the Original/Inner Packets to Specific Tools

The flexibility of advanced packet filtering can also be leveraged to conditionally forward traffic based on original/decapsulated packet contents for Gigamon tunnel decapsulation operation. In the example illustrated below, advanced packet filtering is being used for filtering on the IP address of the decapsulated packets to distribute traffic across a group of monitoring tools.

- Tunnel Decap and forward packets with src ip 1.1.1.1 in inner/original packet to tool 1
- Tunnel Decap and forward packets with src ip 1.1.1.2 in inner/original packet to tool 2

# Configure ports
no traffic all noconfirm
port 1/1/x3 type network
port 1/1/x4 type tool
port 1/1/x1 type tool
# configure gigasmart gsops
gsgroup alias gsg1 port-list 1/1/e1
gsop alias gsfil_decap apf set tunnel-decap type portdst 4000 port-list gsg1
# configure vport.
vport alias vp1 gsgroup gsg1
# forward gmip traffic to 2’nd level map
map alias to_vp
to vp1
  from 1/1/x3
# filter by UDP port #
  rule add pass portdst 4000
exit
# configure 2’nd level map to filter on src/dst (bidirectional) ip 1.1.1.1.
# decap and send it to 1/1/x1
map alias m1
  use gsop gsfil_decap
to 1/1/x1
from vp1
gsrule add pass ipv4 src pos 2 value 1.1.1.1 255.255.255.255
gsrule add pass ipv4 dst pos 2 value 1.1.1.1 255.255.255.255
exit
Facilitating Overlapping Rules

Since advanced packet filtering is implemented as a second-level map operation, advanced packet filtering can also be leveraged for implementing basic overlapping rules. For the same incoming input stream, a copy of the traffic can be sent out to a group of monitoring tools while a refined subset of the traffic stream can be sent to a different set of monitoring tools. Typically overlapping rules would be implemented by combining advanced packet filtering with the patented Flow Mapping technology.

Please note that Role-Based Access Control in case of advanced packet filtering is applied at the gsgroup/e-port. In the example illustrated below, for the same input stream:
- HTTP traffic is identified and distributed to a monitoring tool connected to tool port 1/1/x1
- At the same time, the same stream of HTTP packets are being sent out after slicing unwanted packet contents to a different monitoring tool connected to tool port 1/1/x4

```bash
# Configure ports
no traffic all noconfirm
port 1/1/x3 type network
port 1/1/x4 type tool
port 1/1/x1 type tool
# configure gigasmart gsops
gsgroup alias gsg1 port-list 1/1/e1
gsop alias gsfil apf set port-list gsg1
    gsop alias gsfil_slice apf set slicing protocol none offset 150 port-list gsg1
# configure vport.
vport alias vp1 gsgroup gsg1
# forward all traffic to 2'nd level map
map alias to_vp
to vp1,1/1/x4
from 1/1/x3
# forward all traffic to vp1 and tool2
rule add pass portsrc 2152
exit
# configure 2'nd level map to filter on HTTP traffic and slice it
map alias m1
    use gsop gsfil_slice
to 1/1/x1
from vp1
gsrule add pass l4port dst pos 2 value 80
gsrule add pass l4port src pos 2 value 80
exit
```
# configure 2’nd level map to forward the rest of the traffic
map alias m2  
use gsop gsfil  
to 1/1/x1  
from vp1  
gsrule add pass ipver pos 1 value 4  
exit

In this example, for the same traffic stream, we are sending TCP traffic to one monitoring tool while forwarding a subset of TCP flows specific to HTTP to another monitoring tool connected to tool port 1/1/x4.

# Configure ports  
no traffic all noconfirm  
port 1/1/x3 type network  
port 1/1/x4 type tool  
port 1/1/x1 type tool  
# configure gigasmart gsops  
gsgroup alias gsg1 port-list 1/1/e1  
gsop alias gsfil apf set port-list gsg1  
# configure vport.  
vport alias vp1 gsgroup gsg1  
# forward TCP traffic to 2’nd level map  
map alias to_vp  
to vp1,1/1/x4  
from 1/1/x3  
# forward TCP traffic to vp1  
rule add pass protocol tcp  
exit  
# configure 2’nd level map to filter on HTTP  
map alias m1  
use gsop gsfil  
to 1/1/x1  
from vp1  
gsrule add pass l4port dst pos 1 value 80  
gsrule add pass l4port src pos 1 value 80  
exit

**Common Debug Commands Used to Verify Configuration**

# Map Statistics #  
show map stats alias <map-name>

# Port Statistics #  
show port stats port-list <port-list>

# GigaSMART Statistics #  
show gsop stats all  
show vport stats all  
show gsgroup stats all  
show gsgroup flow-ops-report alias <gs-group> type flow-filtering any summary  
show gsgroup flow-ops-report alias <gs-group> type flow-filtering gtp-imsi-pattern * upload  
tftp://192.168.6.37/IMSI_list1


About Gigamon

Gigamon provides an intelligent Visibility Fabric™ architecture to enable the management of increasingly complex networks. Gigamon technology empowers infrastructure architects, managers and operators with pervasive visibility and control of traffic across both physical and virtual environments without affecting the performance or stability of the production network. Through patented technologies, centralized management and a portfolio of high availability and high density fabric nodes, network traffic is intelligently delivered to management, monitoring and security systems. Gigamon solutions have been deployed globally across enterprise, data centers and service providers, including over half of the Fortune 100 and dozens of government and state and local agencies.

For more information about the Gigamon Visibility Fabric architecture visit: www.gigamon.com