

PacketX GRISM T20



繁體中文操作手冊

V1.0

**目 錄**

[壹、 前言 1](#_Toc41943828)

[貳、 外觀介面說明 2](#_Toc41943829)

[相關連接埠 ： 3](#_Toc41943830)

[參、 網路管理 4](#_Toc41943831)

[肆、 功能說明 6](#_Toc41943832)

[伍、 GRISM XML格式 16](#_Toc41943833)

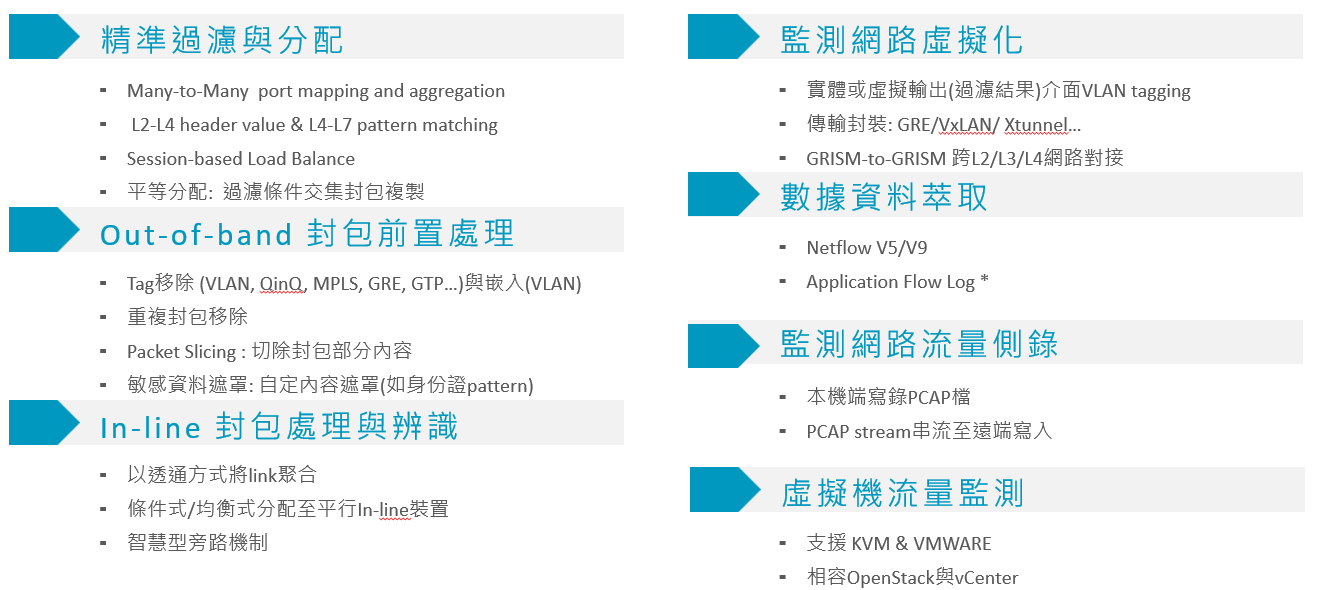
[陸、 RESTful API 26](#_Toc41943834)

[柒、 黑名單阻擋功能 27](#_Toc41943835)

# 前言

PacketX GRISM系列是一個網路可視化平台，可針對流量進行精細的分流，主要應用於

1. 精準分流：將所收到的網路流量導引至多樣且多台的網路分析設備或網路安全設備，導引過程會有封包去除與過濾篩選的機制
2. Metadata擷取：自網路流量中萃取出metdadata，比如非取樣的Netflow
3. IP/Domain偵測與阻斷：針對輸入的封包進行比對，若偵測到屬於黑名單內的一員，將視使用者設定以執行阻斷或告警。黑名單的清單可由外部以API的方式輸入。

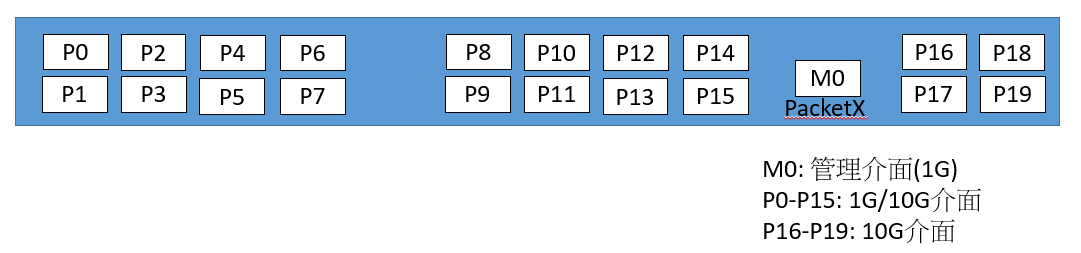


# 外觀介面說明

* 實體外觀



* 埠位示意圖



M0：管理介面(1G)

P0-P15: 1G/10G介面(SFP/SFP+選配)

P16-P19: 10G介面(SFP+選配)

* 硬體外觀規格

|  |  |  |  |
| --- | --- | --- | --- |
| Specifications | | | |
| Height | 1.75 in (44.45 mm) | Operating Humidity | 10-90% RH, non-condensing |
| Length | 21 in (533.40 mm) | Storage Humidity | 5-95% RH, non-condensing |
| Width | 16.80 in (126.72 mm) | Max Raw Power | 356W |
| Weight | 20.70 Ibs. (9.39 kg) | Air Flow | Front to Back |
| Operating Temp | 0-40 Celsius (32-104 Fahrenheit) | AC | 100-240V, 8-4A, 50-60Hz |

## 相關連接埠 ：

* **Console/Serial port連線設定需依照下列參數進行設定：**
  + - **Baud rate:**115200
    - **Data:** 8-bit
    - **Parity:** None
    - **Stop:** 1-bit
    - **Flow Control:**None

# 網路管理

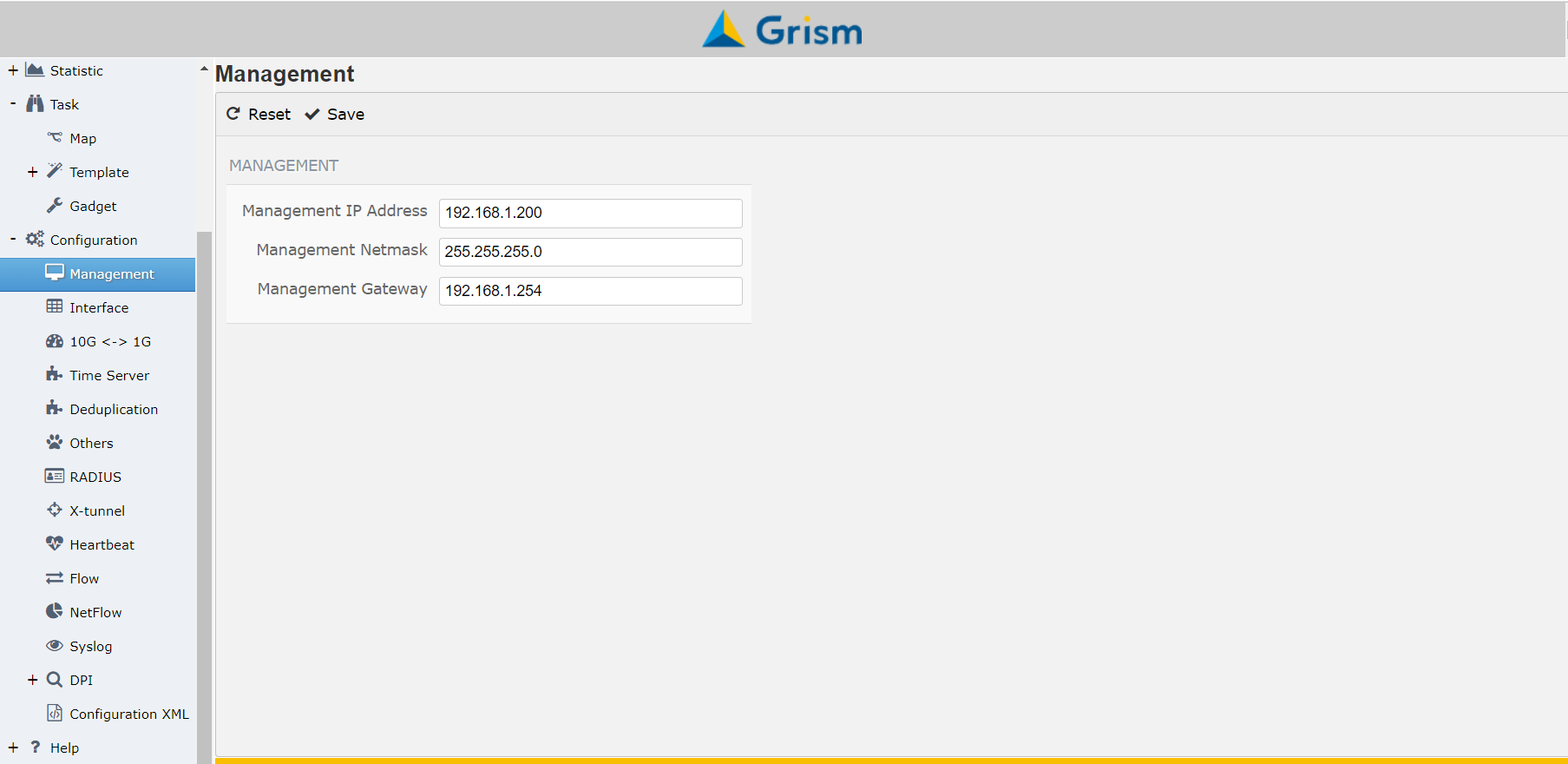
1. 將網路線插在M0的位置，使用瀏覽器開啟 https://192.168.1.160網頁，即可進行登入作業，預設帳號/密碼為：packetx/packetx



1. 登入後，畫面如下

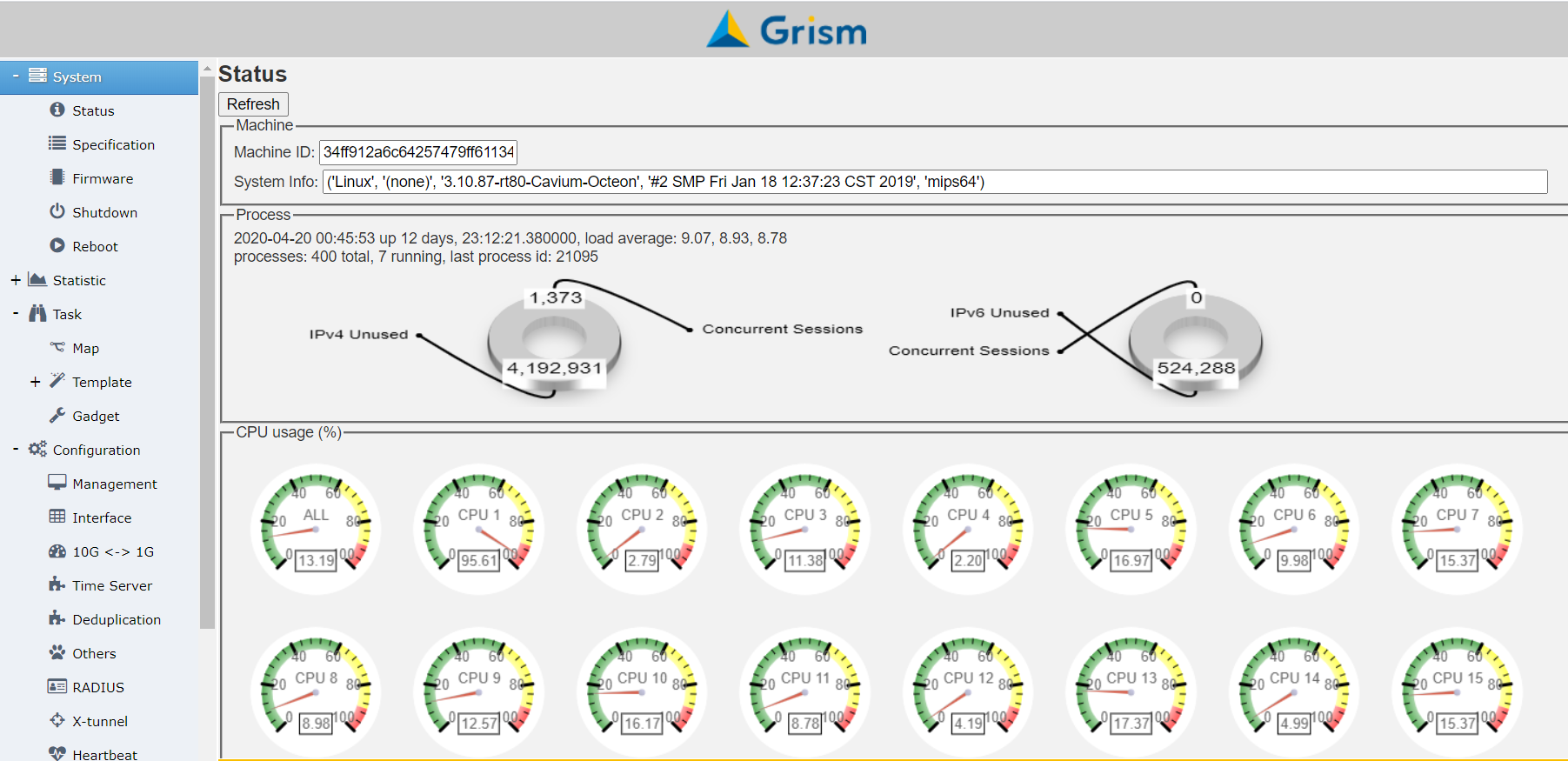


1. 進入後點選**Configuration**的**Management**設定更改Management IP Address、Management Netmask、Management Gateway

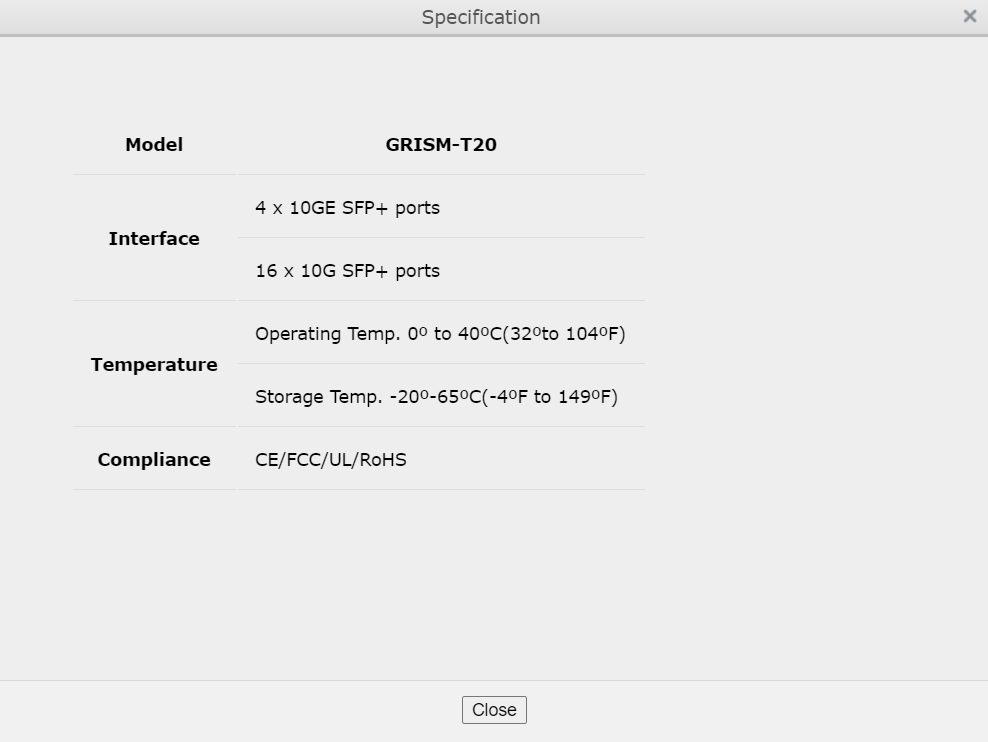


# 功能說明

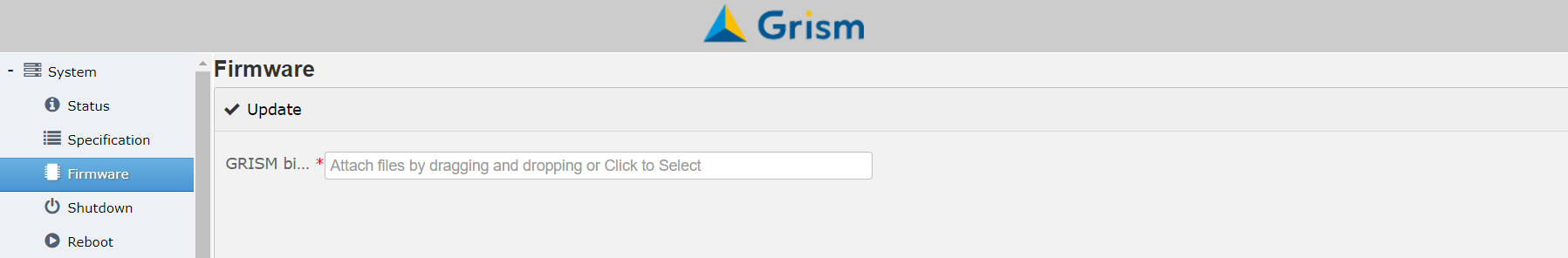
1. 系統狀態，顯示設備硬體之Machine ID、系統中呈現之IPv4/IPv6/Session之統計值及每個CPU之使用狀況。



1. Specification之硬體細部規格。



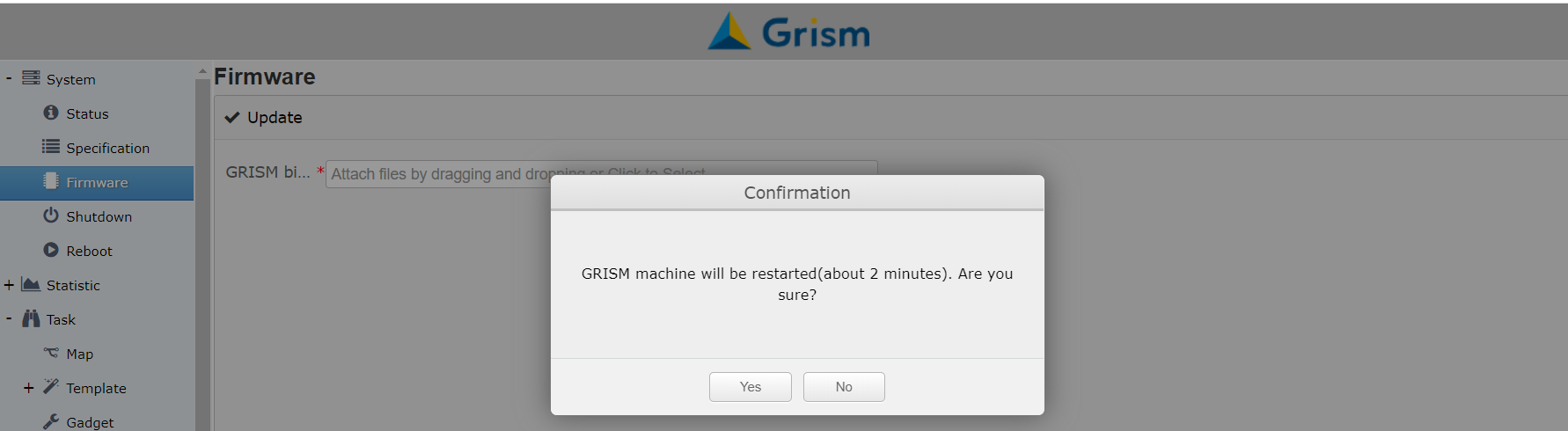
1. Firmware可以手動更新韌體。



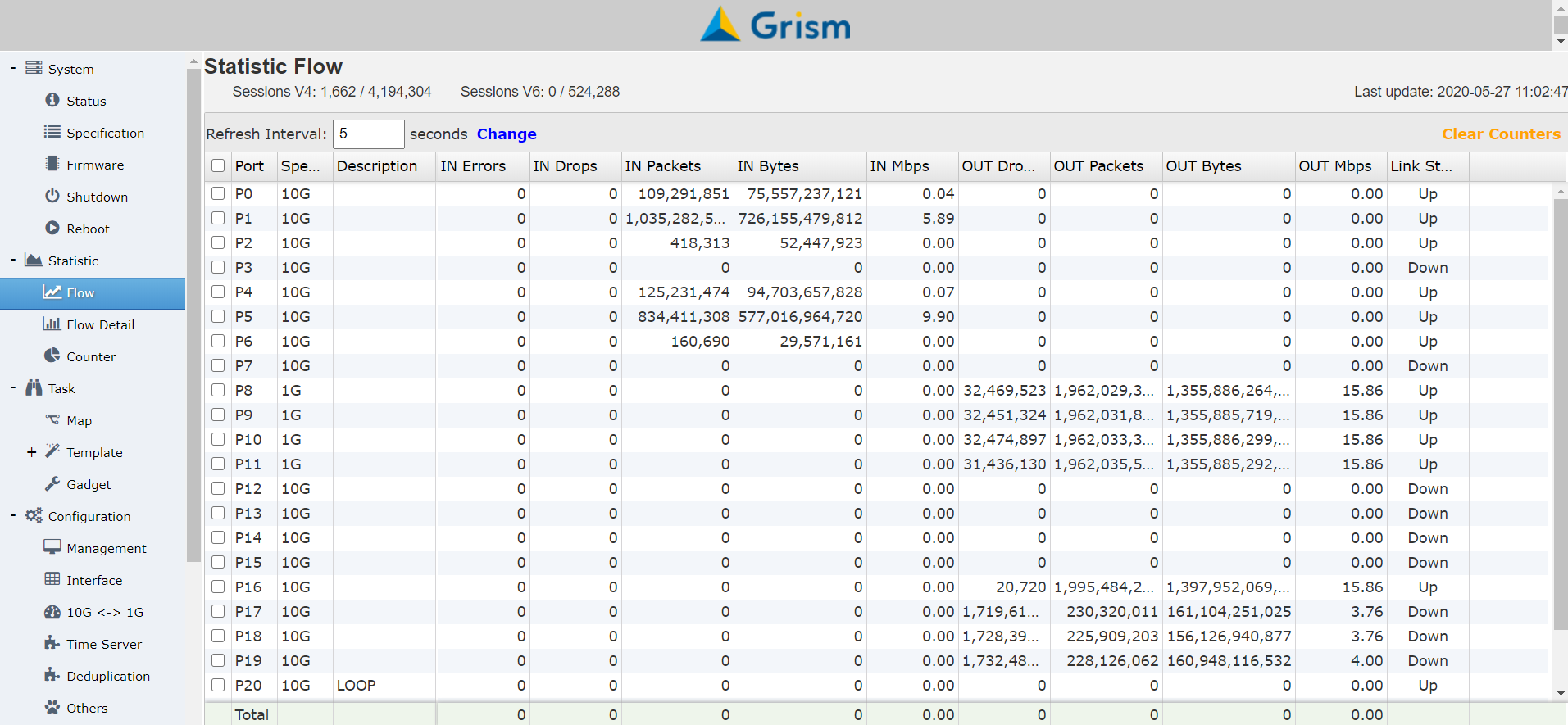
1. Shutdown為關閉此設備。



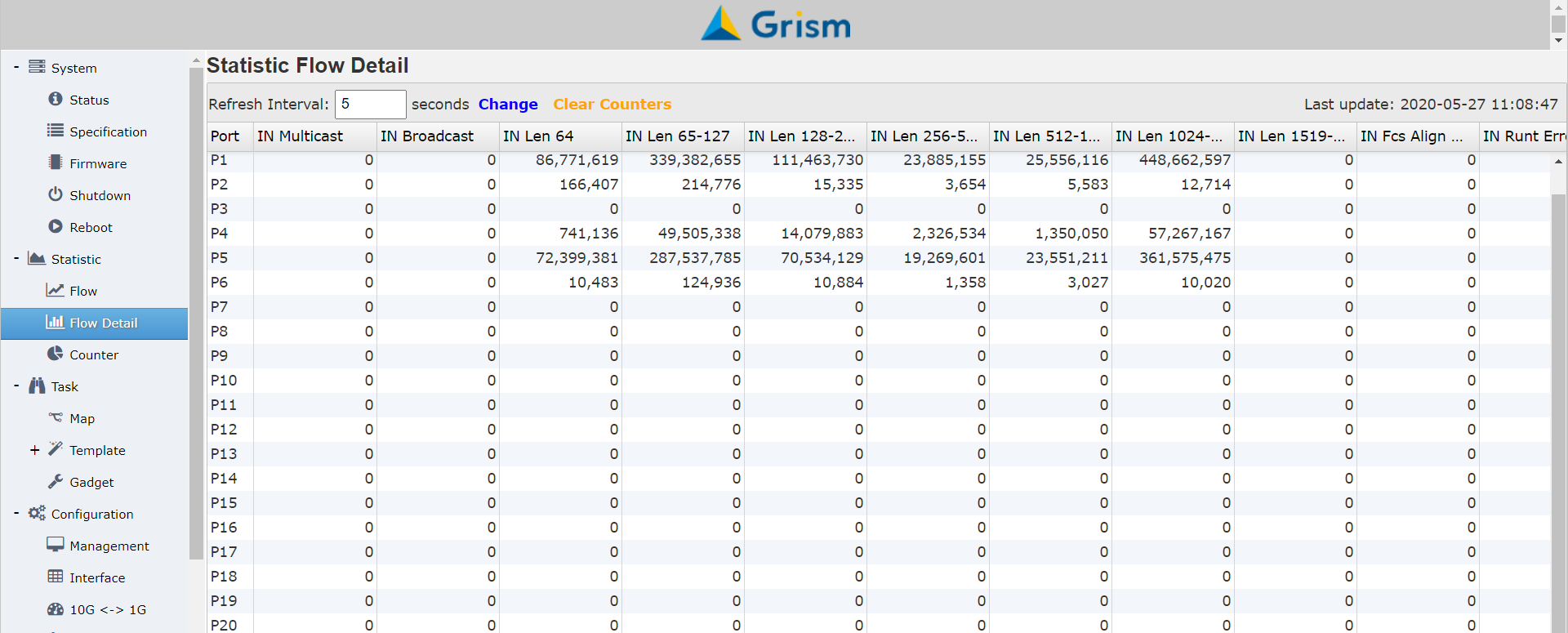
1. Reboot為重新啟動此設備，需要等待約2分鐘。



1. Statistic為呈現目前系統設定之每個埠狀態，Up及代表有皆線在上，並且有流量，IN Errors/IN Drops/IN Packets/In Bytes/IN Mbps代表目前線路上之輸入流量，OUT Drops/ OUT Packets/ OUT Bytes/ OUT Mbps代表目前線路上之輸出流量。



1. Statistic Flow Detail資訊



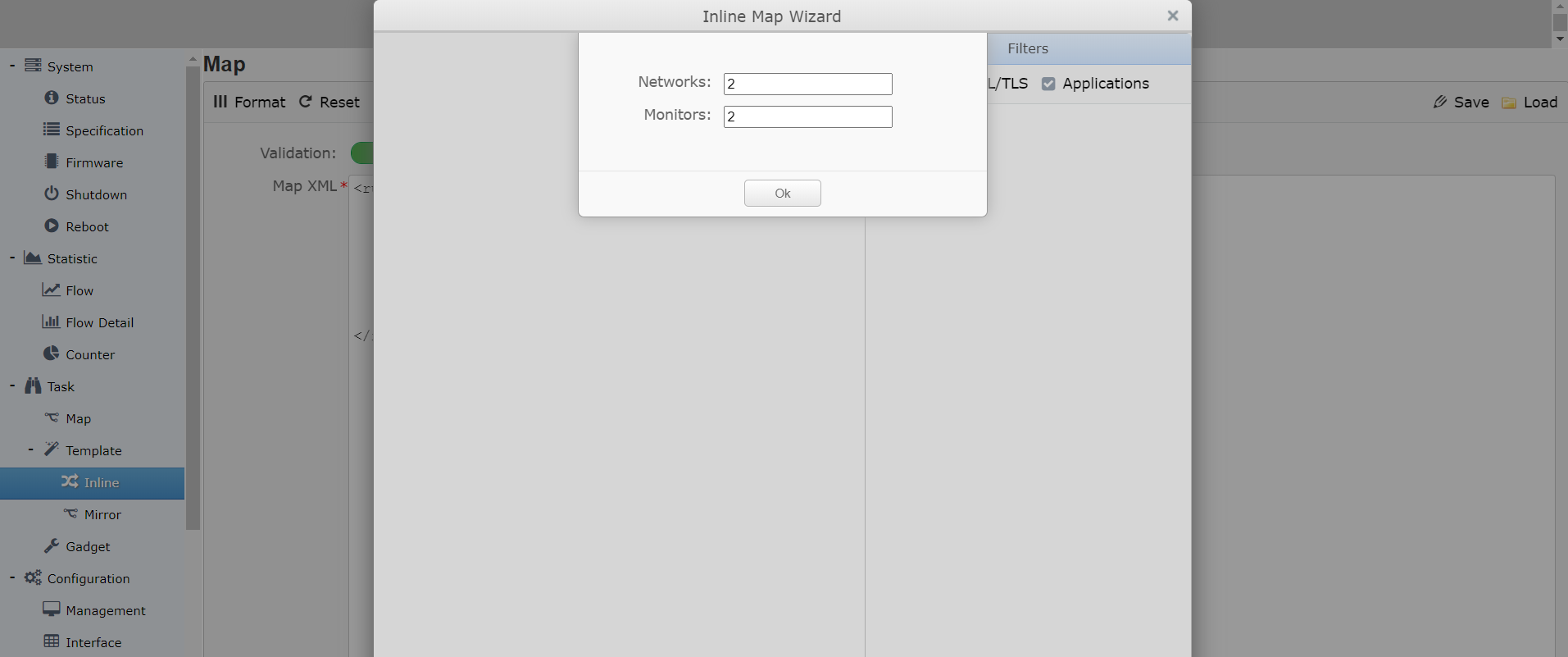
1. Statistic Counter，呈現流量內之TCP/UDP/ICMP協定之統計，以及每個服務埠之統計



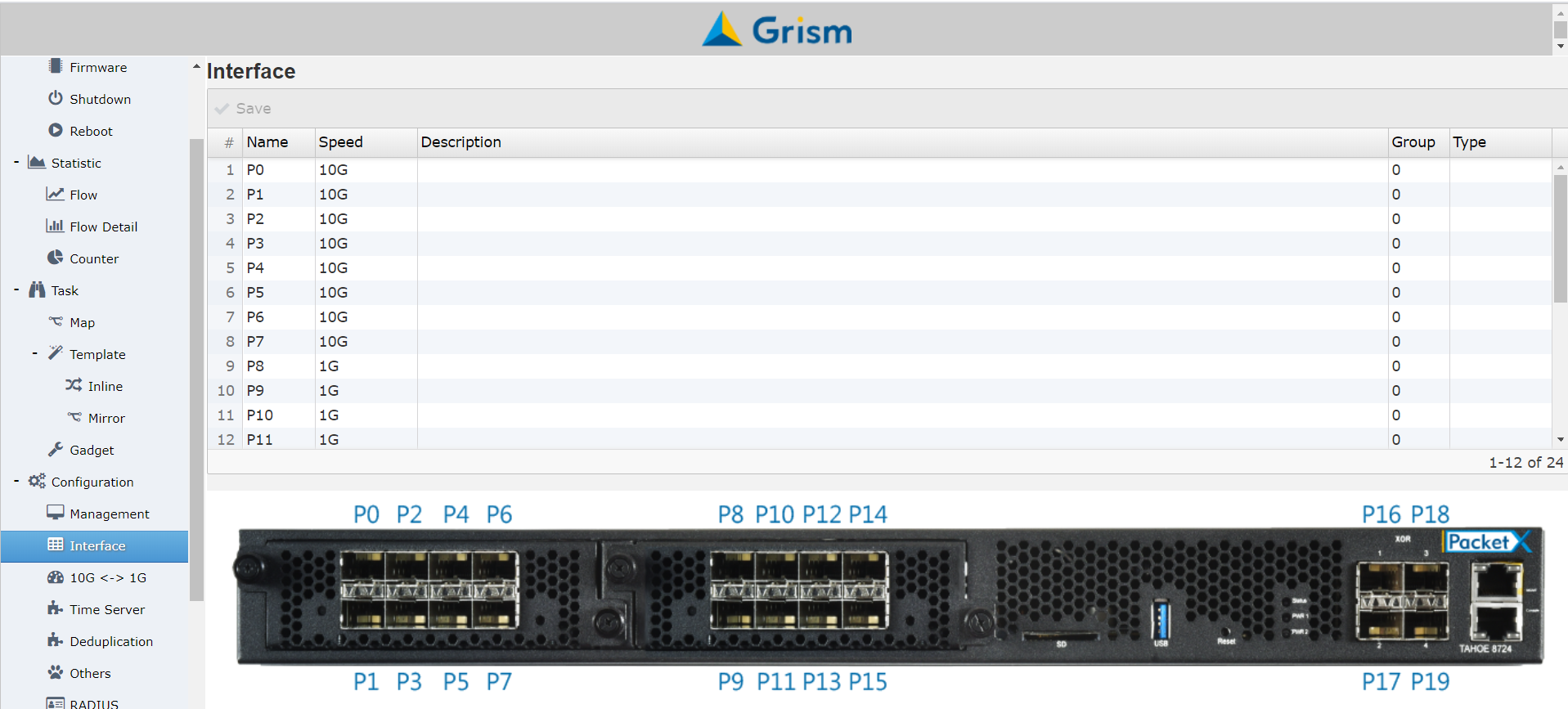
1. **Task**的 **Map**為主要核心，所有功能設定均參考此頁之XML進行設定，範例即為



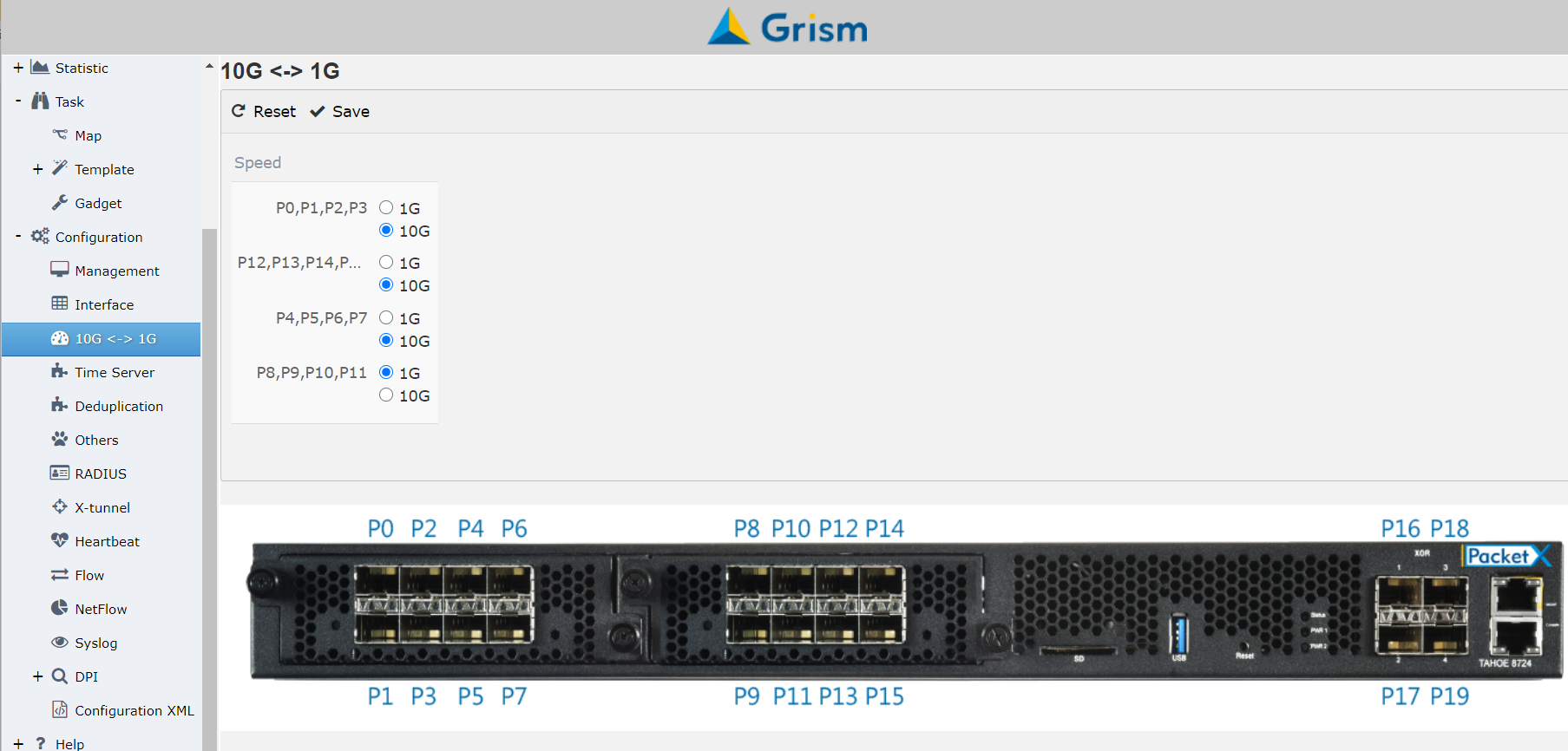
1. **Task**的Inline Map Wizard有工具可以產生設定檔



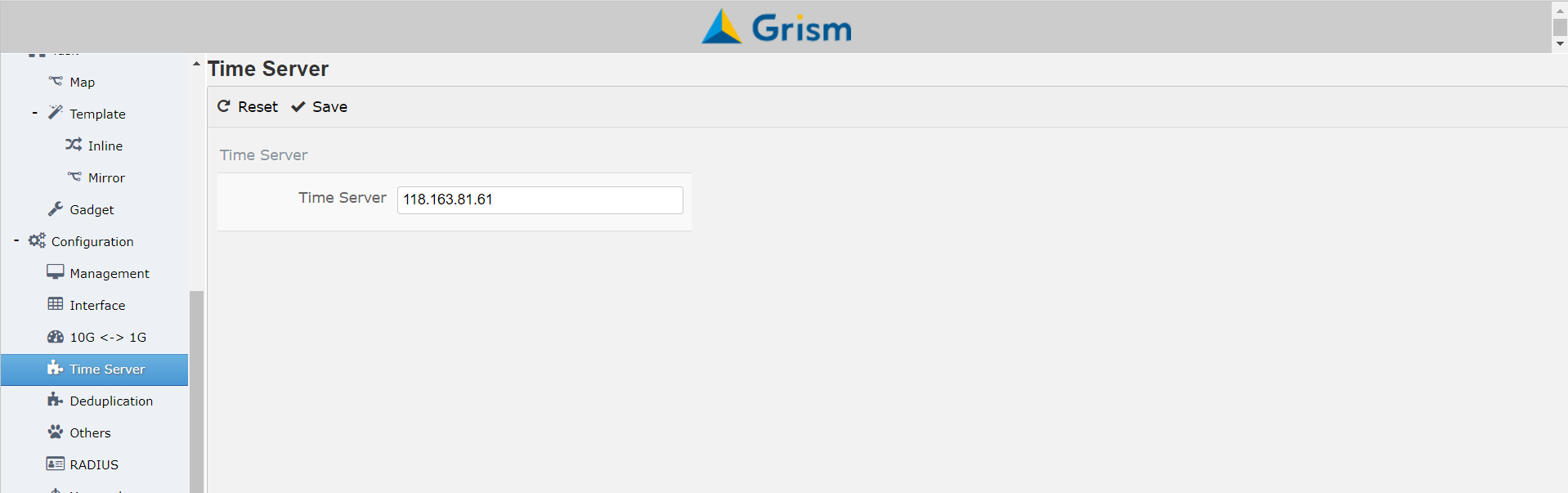
1. **Configuration**內的**Interface**可以設定每個埠之描述，以利後續識別。



1. 若是有特定埠需要由10G介面轉換為1G介面，需要由**Configuration**內的**10G <-> 1G**進行設定轉換，每次轉換為四個埠為一組。



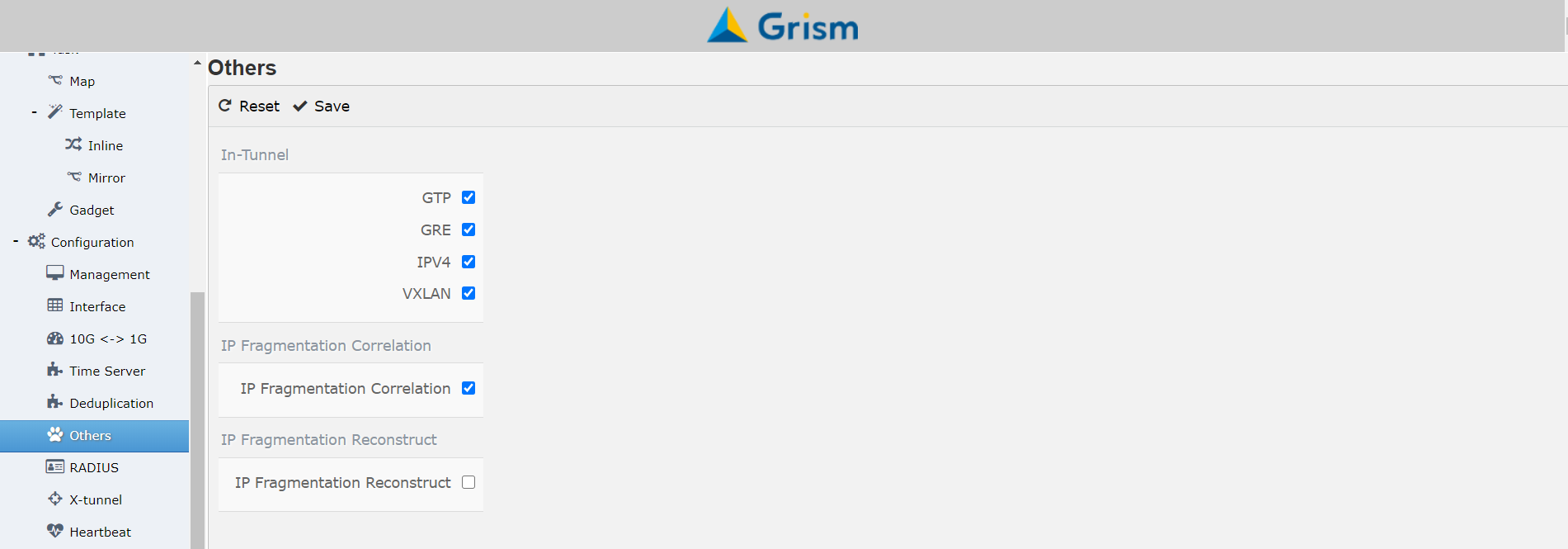
1. **Configuration**內的**Time Server**可以校時之伺服器網路位置。



1. **Configuration**內的**Deduplication**可以設定是否要去除重複之封包。



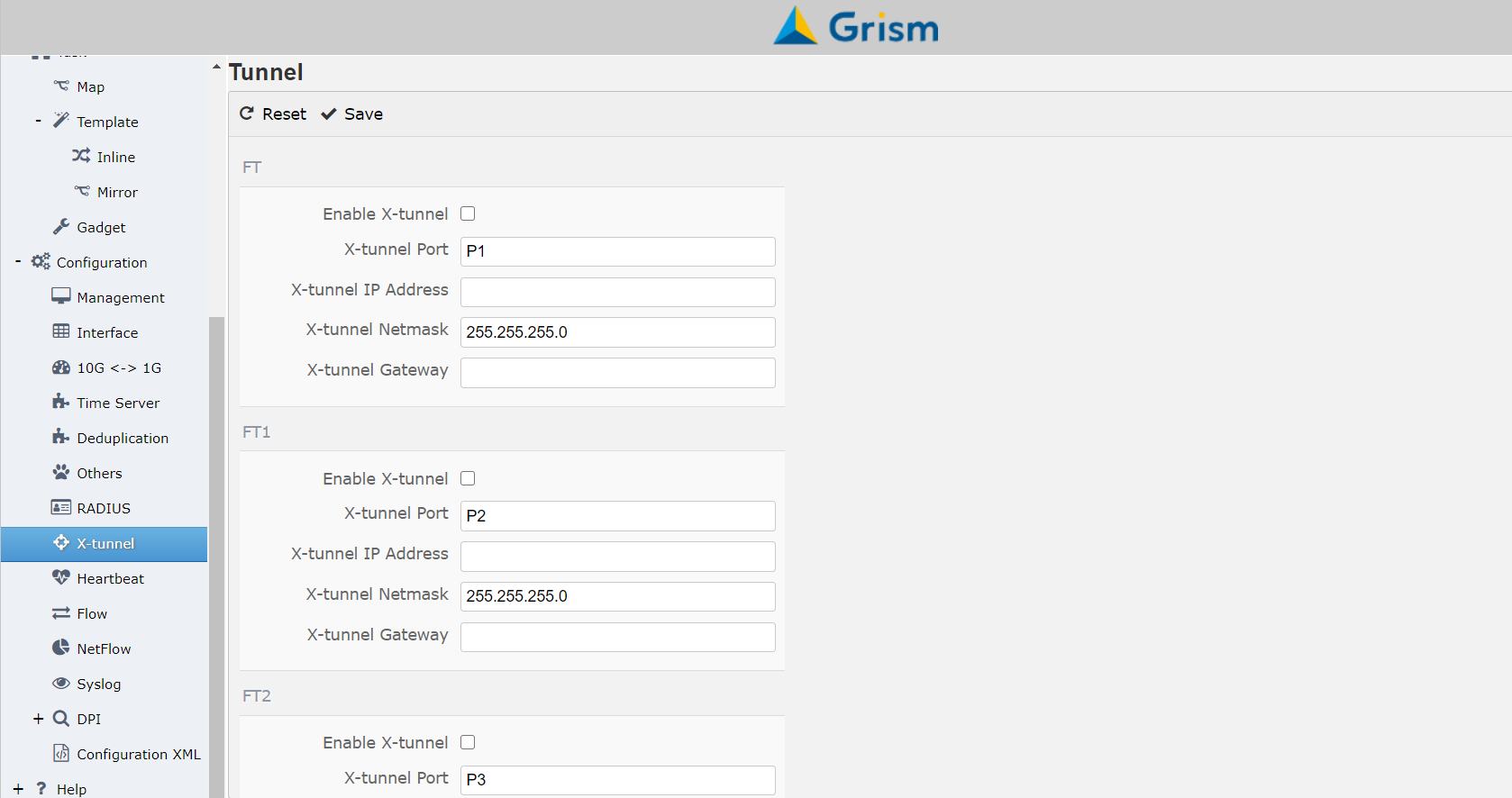
1. **Configuration**內的**Others**可以設定是否有要啟用GTP、GRE、IPv4、VLAN、IP Fragmentation Correlation、IP Fragmentation Reconstruct等識別功能。



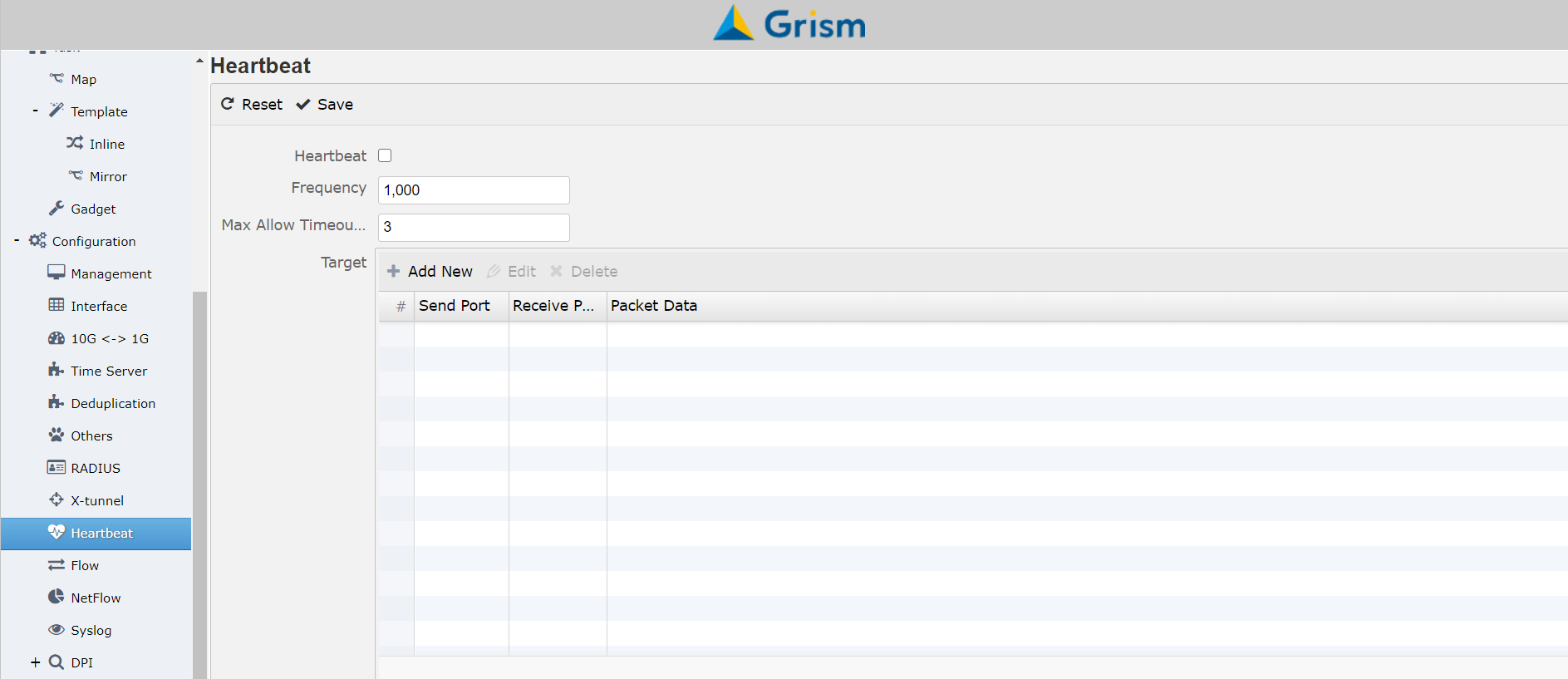
1. **Configuration**內的**Radius**參數設定。



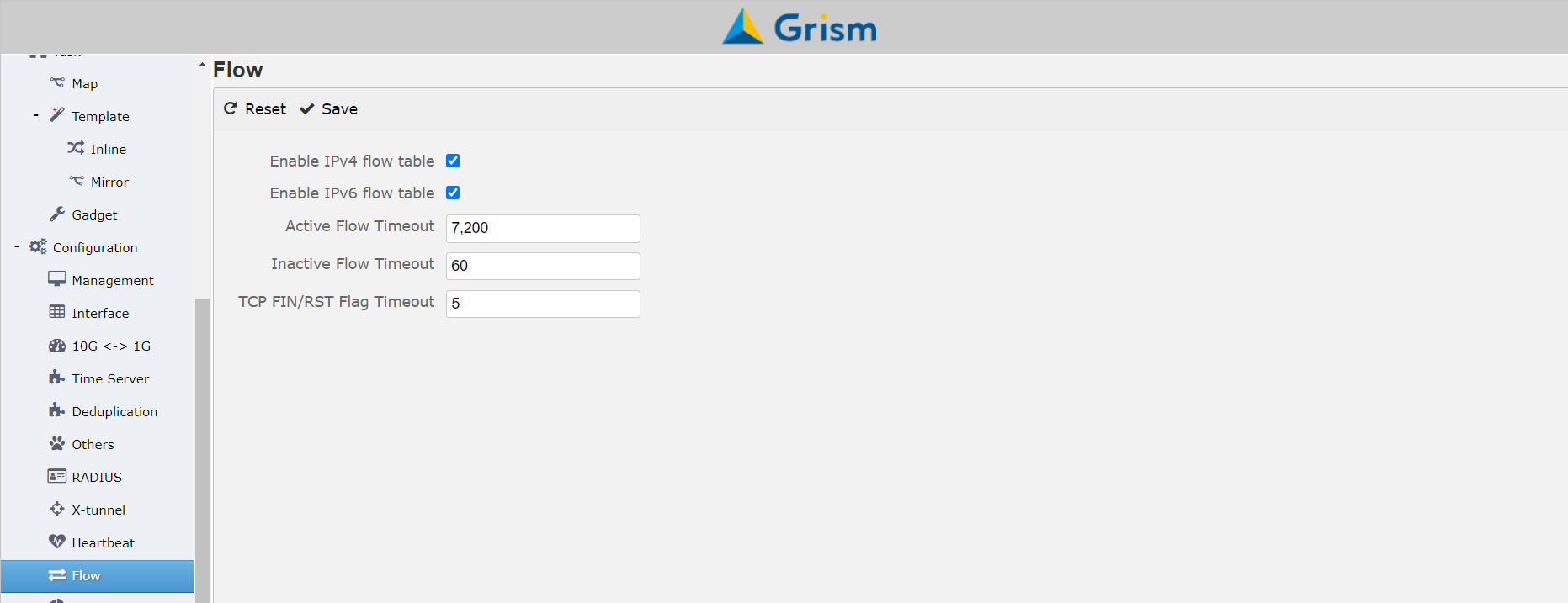
1. **Configuration**內的**X-Tunnel**設定參數。



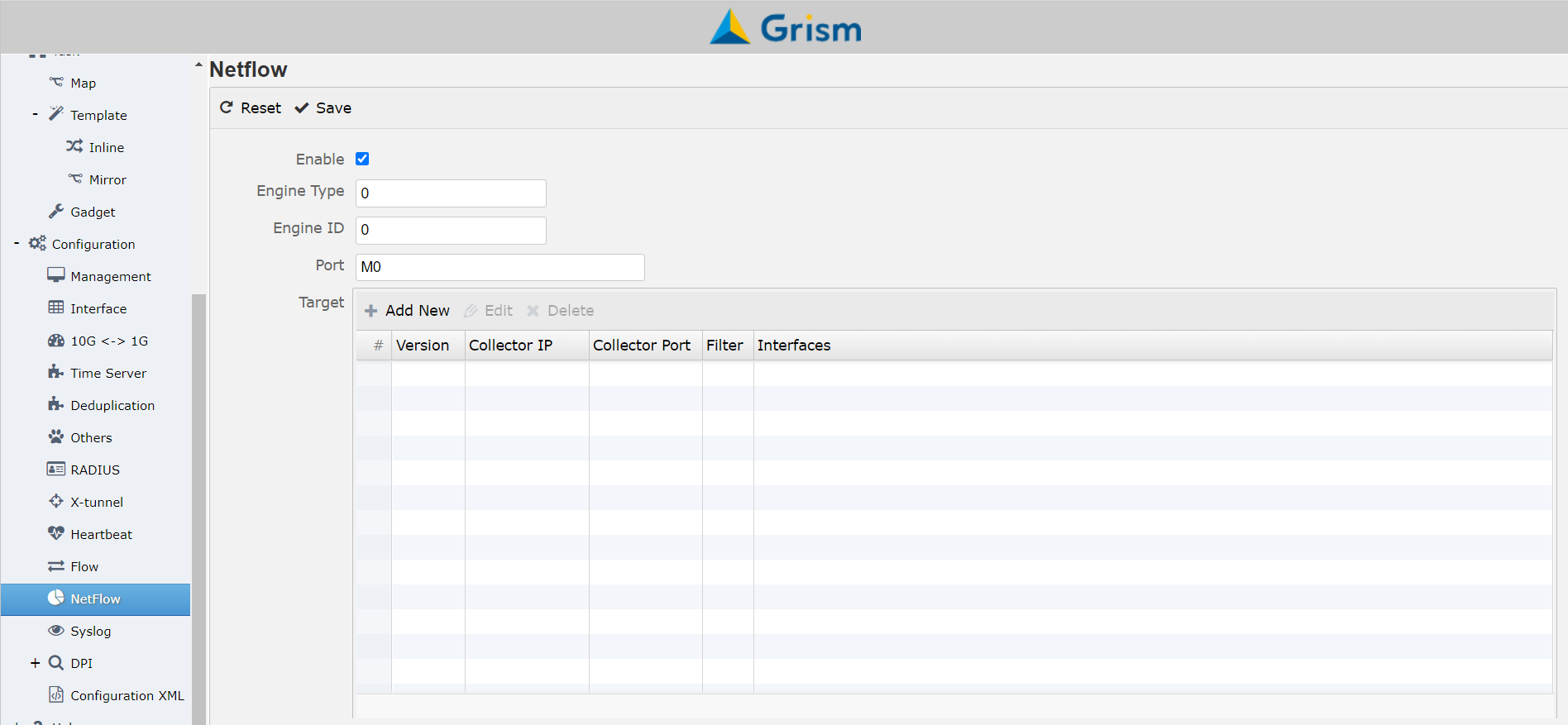
1. **Configuration**內的**Heartbeat**可以設定並監控設備健康存活狀態。



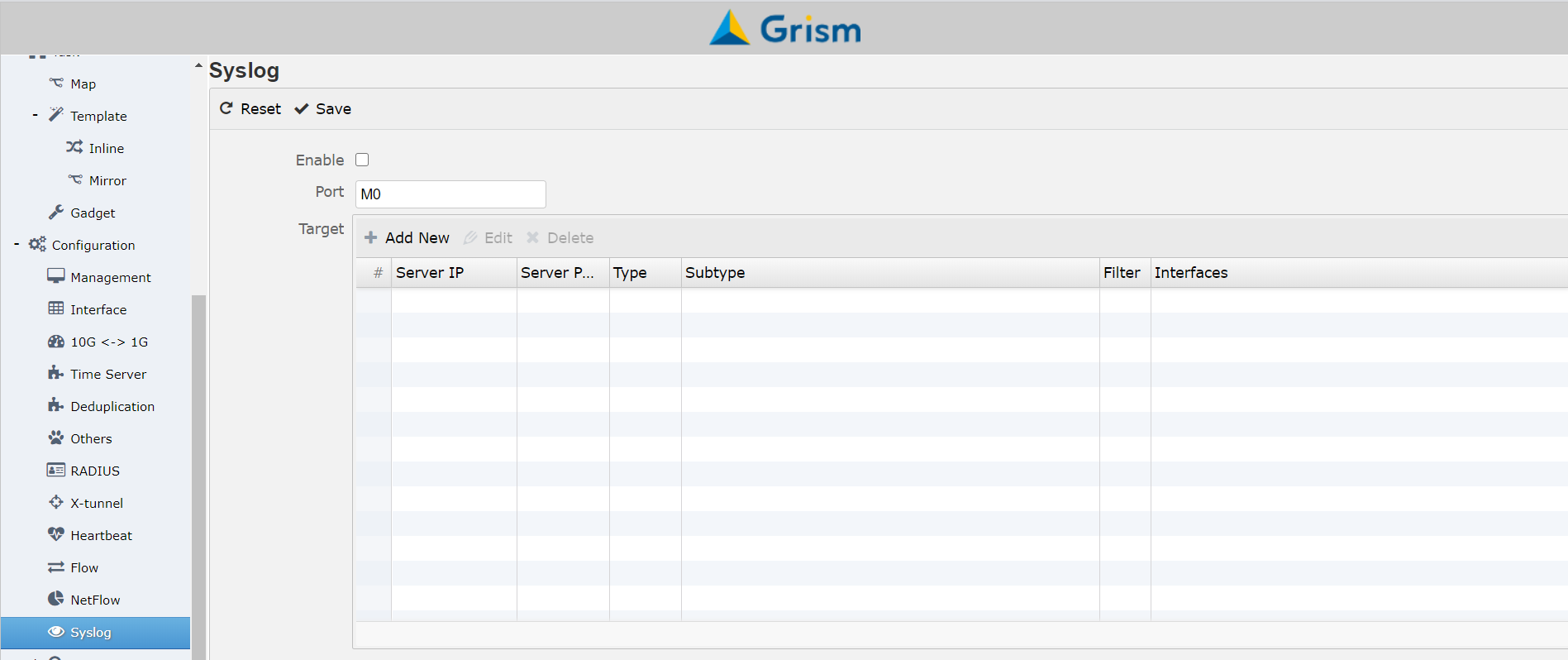
1. **Configuration**內的**Flow**可以設定Flow之取樣參數。



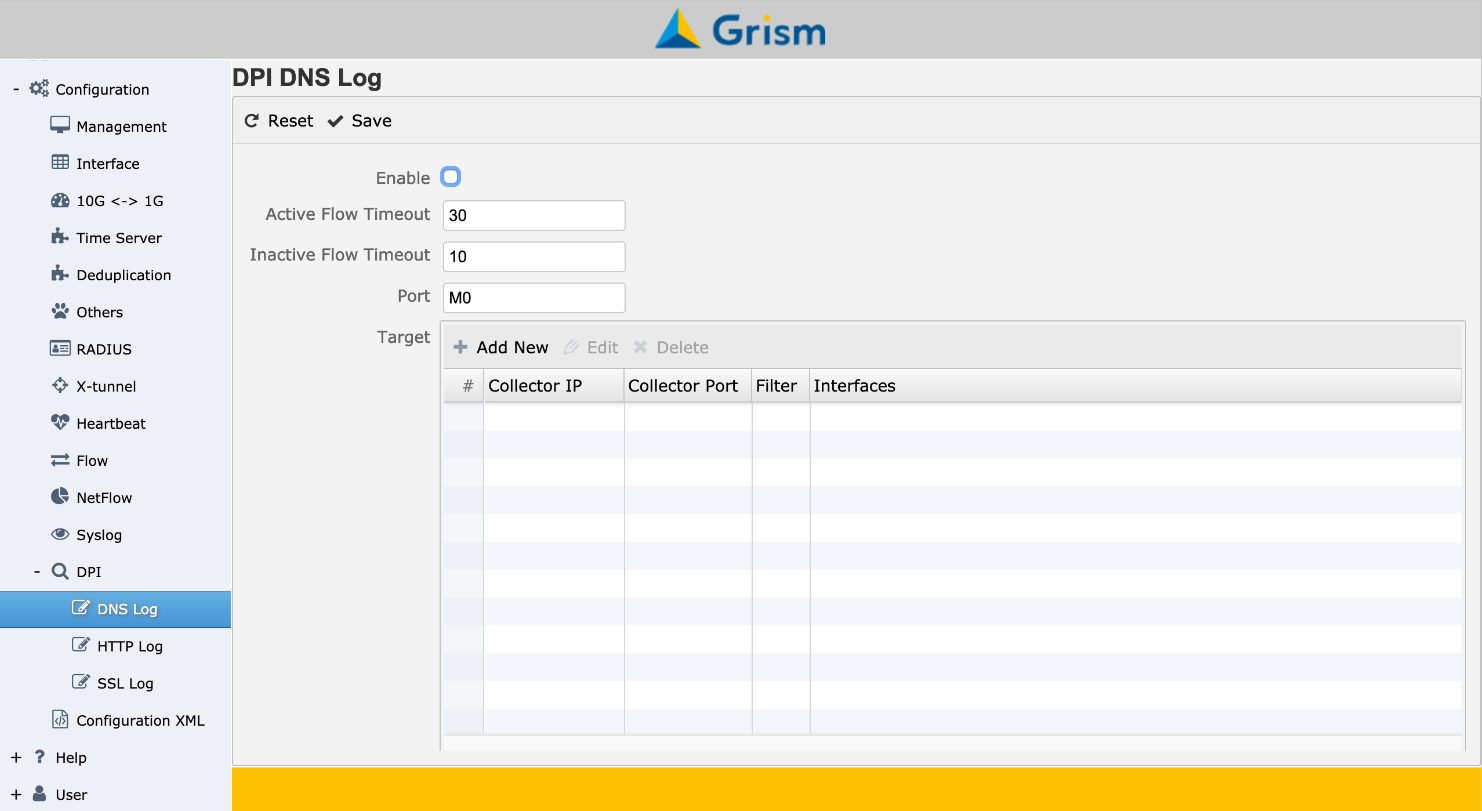
1. **Configuration**內的**Netflow**可以設定接收Netflow資訊目的地之網路地址。



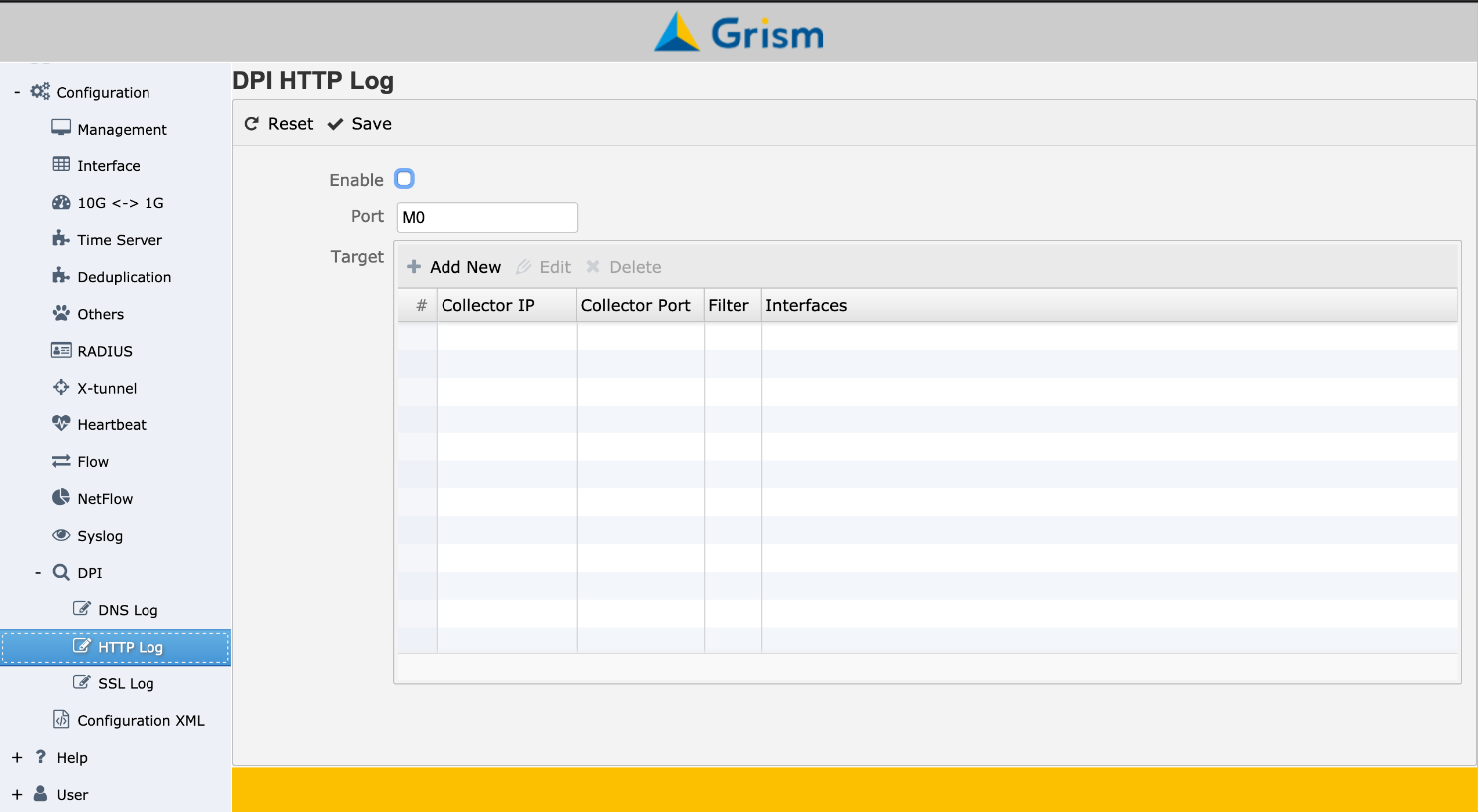
1. **Configuration**內的**Syslog**可以設定接收Syslog目的地之網路地址。



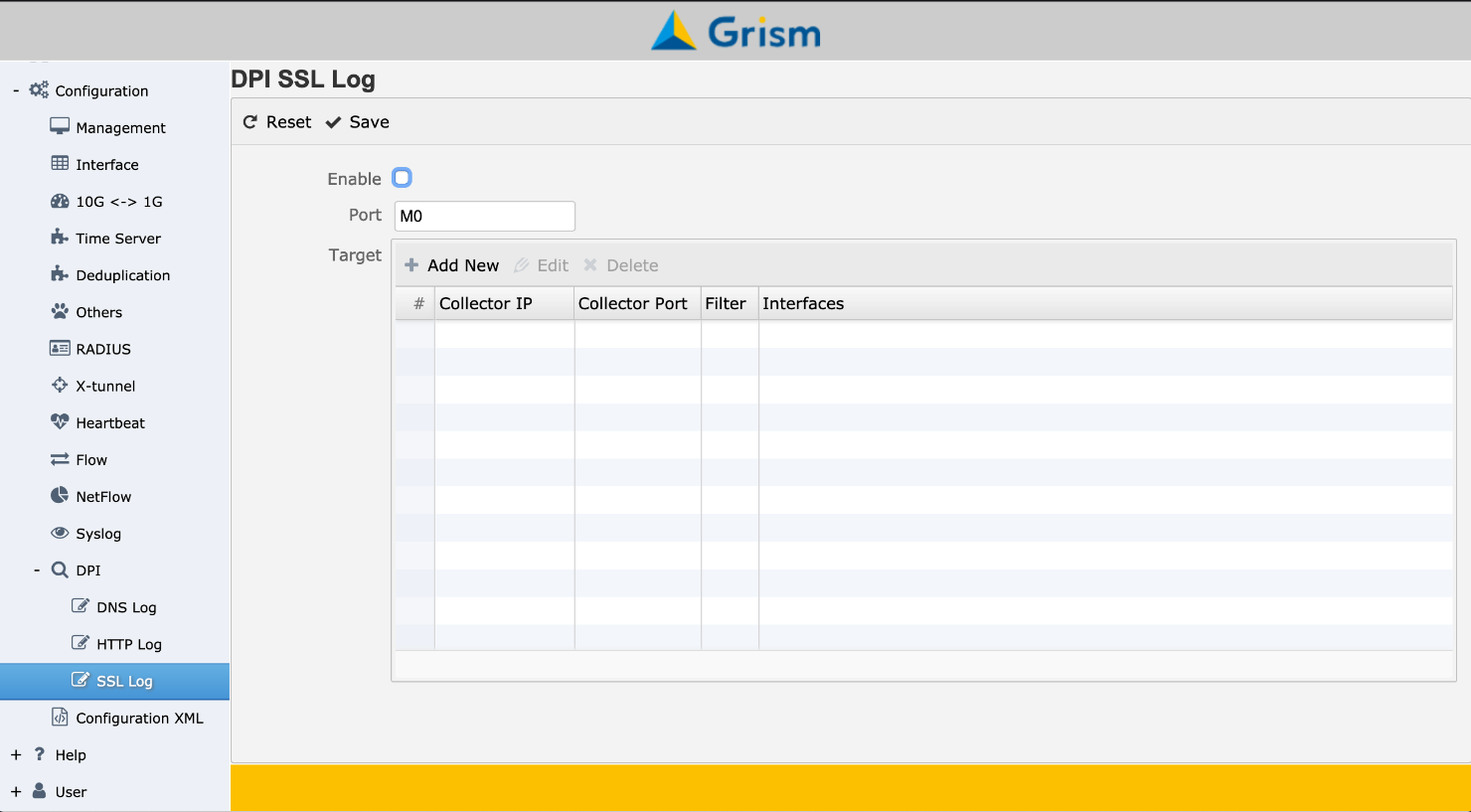
1. **DPI**內的**DPI DNS Log**可以產出訊務中與DNS相關之DPI資訊。



1. **DPI**內的**DPI HTTP Log**可以產出訊務中與HTTP相關之DPI資訊。



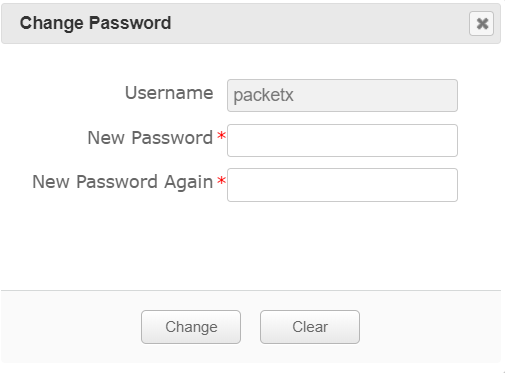
1. **DPI**內的**DPI SSL Log**可以產出訊務中與SSL相關之DPI資訊。



1. **Configuration**內的**Configuration XML**可以讓使用者確認目前系統之相關設定是否正確。

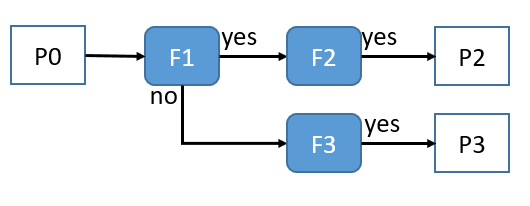


1. **User**內的**Password**可以修改預設之帳號/密碼，預設帳號/密碼為：packetx/packetx。



# GRISM XML語法與撰寫範例

1. 欄位格式說明
   * <run>🡪開始
   * <filter>🡪過濾規則
   * <and>、<or>🡪布林邏輯運算
   * <chain>🡪鏈結
   * <in>🡪輸入
   * <fid>🡪功能編號
   * <out>🡪輸出
   * <next type=“match”>🡪 符合規則
   * <next type=“notmatch”>🡪不符合規則
2. in/out流量路徑規劃



<chain id="0">

<in>P0</in>

<fid>F1</fid>

<next type="match">

<fid>F2</fid>

<out>P2</out>

</next>

<next type="notmatch">

<fid>F3</fid>

<out>P3</out>

</next>

</chain>

1. 針對DNS流量過濾

<filter id=“1” name=“ip.src list” sessionBase=“no”>

<or>

<find name=“ip.src” relation=“==” content=“8.8.8.8”>

<find name=“ip.src” relation=“==” content=“168.95.1.1”>

</or>

</filter>

1. 針對L7流量過濾

<filter id=“2” name=“l7 test” sessionBase=“yes”>

<or>

<find name=“regex” relation=“==” content=“\x0bgooglevideo\x03com”>

<find name=“regex” relation=“==” content=“{s}\/.\*Host: yahoo\.com”>

</or>

</filter>

* + ps1：\x0bgooglevideo\x03com是要比對DNS訊息內的domain googlevideo.com
  + ps2：{s}\/.\*Host: yahoo\.com是要比對HTTP host yahoo.com比對成功後接下來屬於相同session之封包會被濾出

XML範例一：L2/L3過濾

過濾特定VLAN/IP/IP Rang(Class C)之流量

<?xml version="1.0" encoding="UTF-8"?>

<run>

<!-- empty rule, for aggregation -->

<filter id="0" name="empty" sessionBase="no">

<or></or>

</filter>

<!-- vlan id is 100 -->

<filter id="1" name="vlan" sessionBase="no">

<or>

<find name="vlan.id" relation="==" content="100" />

</or>

</filter>

<!-- one ip -->

<filter id="2" name="ip" sessionBase="no">

<or>

<find name="ip.addr" relation="==" content="192.168.1.203" />

</or>

</filter>

<!-- class c ip -->

<filter id="4" name="ip" sessionBase="no">

<or>

<find name="ip.addr" relation="==" content="192.168.3.0/24" />

</or>

</filter>

<!-- multiple ip -->

<filter id="3" name="ip" sessionBase="no">

<or>

<find name="ip.addr" relation="==" content="192.168.2.10" />

<find name="ip.addr" relation="==" content="192.168.2.11" />

<find name="ip.addr" relation="==" content="192.168.2.12" />

<find name="ip.addr" relation="==" content="192.168.2.13" />

</or>

</filter>

XML範例二：過濾YouTube等串流訊務

過濾特定過濾Streaming訊務(續)流量導流過濾：P0、P9、P1、P2為流量輸入埠，符合F1規則的由P3輸出，符合F2規則的由P4輸出，符合F3規則的由P5輸出，符合F4規則的由P6輸出，符合F5及F5規則的由P7輸出，符合F5及F2規則的由P8輸出。

<!-- Streaming -->

<filter id="5" name="ssl client\_hello" sessionBase="no">

<or>

<find id="2" name="ssl.handshake.type" relation="==" content="1" />

<find id="3" name="quic.tag" relation="==" content="CHLO" />

<find id="4" name="http.request" relation="==" content="" />

</or>

</filter>

<filter id="6" name="stream host" sessionBase="yes">

<or>

<find id="122" name="regex" relation="==" content="{s}\.googlevideo\.com" />

<find id="123" name="regex" relation="==" content="{s}\.youtube\-nocookie\.com" />

<find id="124" name="regex" relation="==" content="{s}\.iqiyi\.com" />

<find id="125" name="regex" relation="==" content="{s}v\-df1a4a46\.71edge\.com" />

<find id="126" name="regex" relation="==" content="{s}v\-df1a4a4a\.71edge\.com" />

<find id="127" name="regex" relation="==" content="{s}Qiyi\.com" />

<find id="128" name="regex" relation="==" content="{s}pps\.tv" />

<find id="129" name="regex" relation="==" content="{s}google\-analytics\.com" />

<find id="130" name="regex" relation="==" content="{s}vimeo\.akamaized\.net" />

<find id="131" name="regex" relation="==" content="{s}cf\-hls\-media\.sndcdn\.com" />

<find id="132" name="regex" relation="==" content="{s}\.fbcdn\.net" />

<find id="133" name="regex" relation="==" content="{s}video\.twimg\.com" />

<find id="134" name="regex" relation="==" content="{s}vtt\.tumblr\.com" />

<find id="135" name="regex" relation="==" content="{s}upos\-hz\-mirrorcos\.acgvideo\.com" />

<find id="136" name="regex" relation="==" content="{s}p\-events\-delivery\.akamaized\.net" />

<find id="137" name="regex" relation="==" content="{s}aod\.itunes\.apple\.com" />

<find id="138" name="regex" relation="==" content="{s}vdn\-obs\.line\-scdn\.net" />

<find id="139" name="regex" relation="==" content="{s}dev\-asia\-vod\-linetv\.line\-apps\.com" />

<find id="140" name="regex" relation="==" content="{s}sstatic\-linetv\.line\-apps\.com" />

<find id="141" name="regex" relation="==" content="{s}dev\-gb\-vod\-linetv\.line\-apps\.com" />

<find id="142" name="regex" relation="==" content="{s}asia\-vod\-linetv\.line\-apps\.com" />

<find id="143" name="regex" relation="==" content="{s}gb\-vod\-linetv\.line\-apps\.com" />

</or>

</filter>

XML範例三：匯聚鏡像流量後再分配輸出(Load Balance)

<chain id="1" name="01">

<in>P0,P9,P1,P2</in>

<fid>F0</fid>

<next type="match">

<fid>F1<fid>

<out>P3</out>

</next>

<next type="match">

<fid>F2<fid>

<out>P4</out>

</next>

<next type="match">

<fid>F3<fid>

<out>P5</out>

</next>

<next type="match">

<fid>F4<fid>

<out>P6</out>

</next>

<next type="match">

<fid>F5<fid>

<next type="match">

<fid>F6<fid>

<out>P7</out>

<next type="match">

<fid>F2<fid>

<out>P8</out>

</next>

</next>

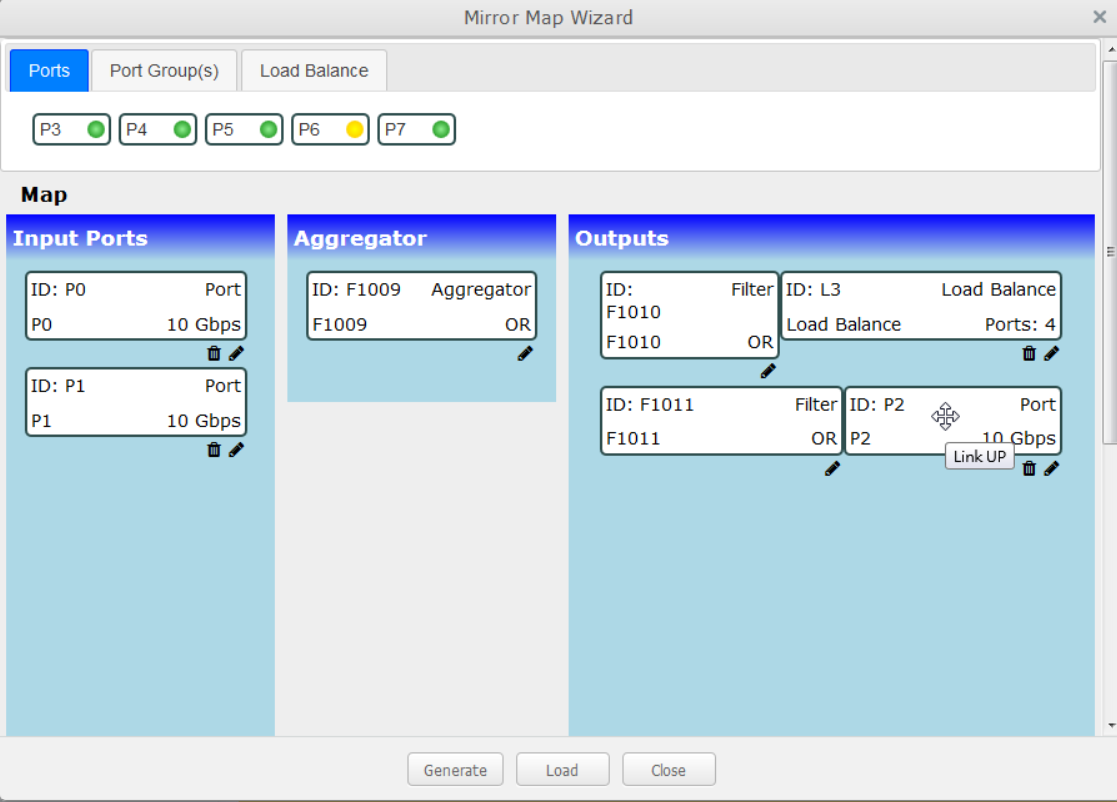
</next>

</chain>

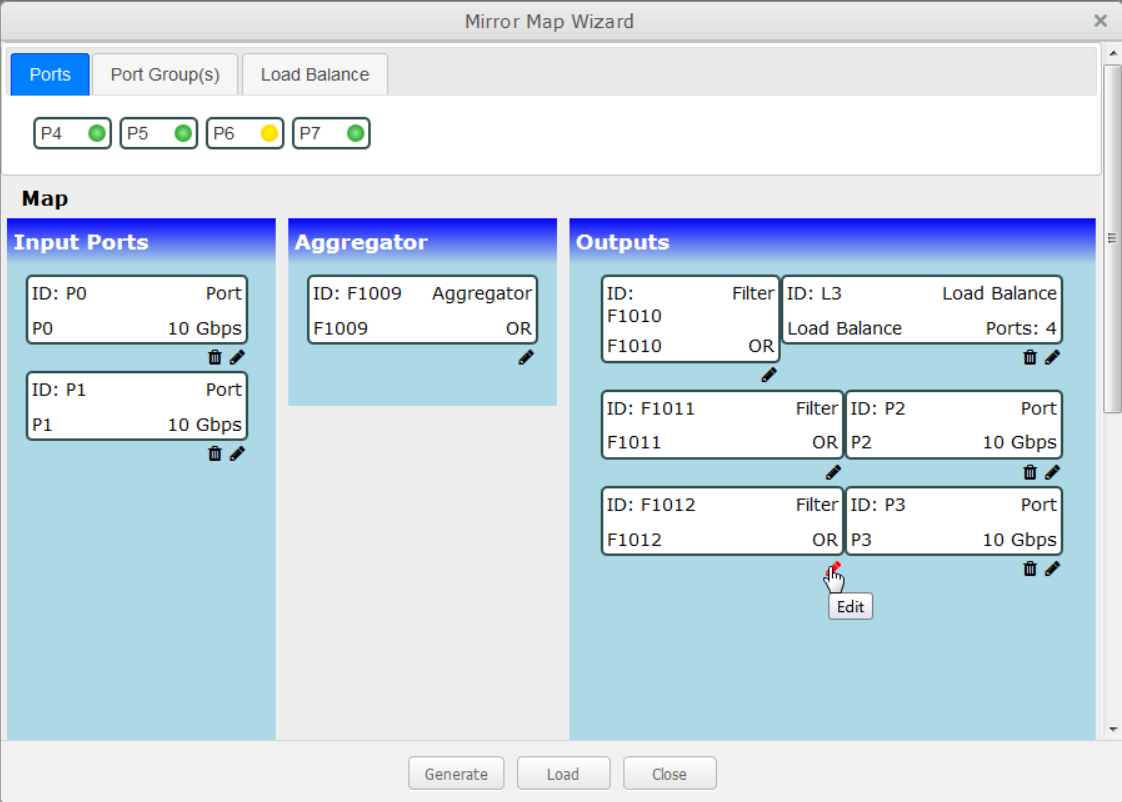
</run>

接收流量再送給流量分析設備時，可能有多個輸入，且支援多種分析功能；某些分析功能要用多台設備或多個網路介面來收納。底下的範例將輸入流量送至兩種分析設備，其中一種用4個介面來收納所有流量，另一種用1個介面來收納且只針對tcp port 443的流量來分析

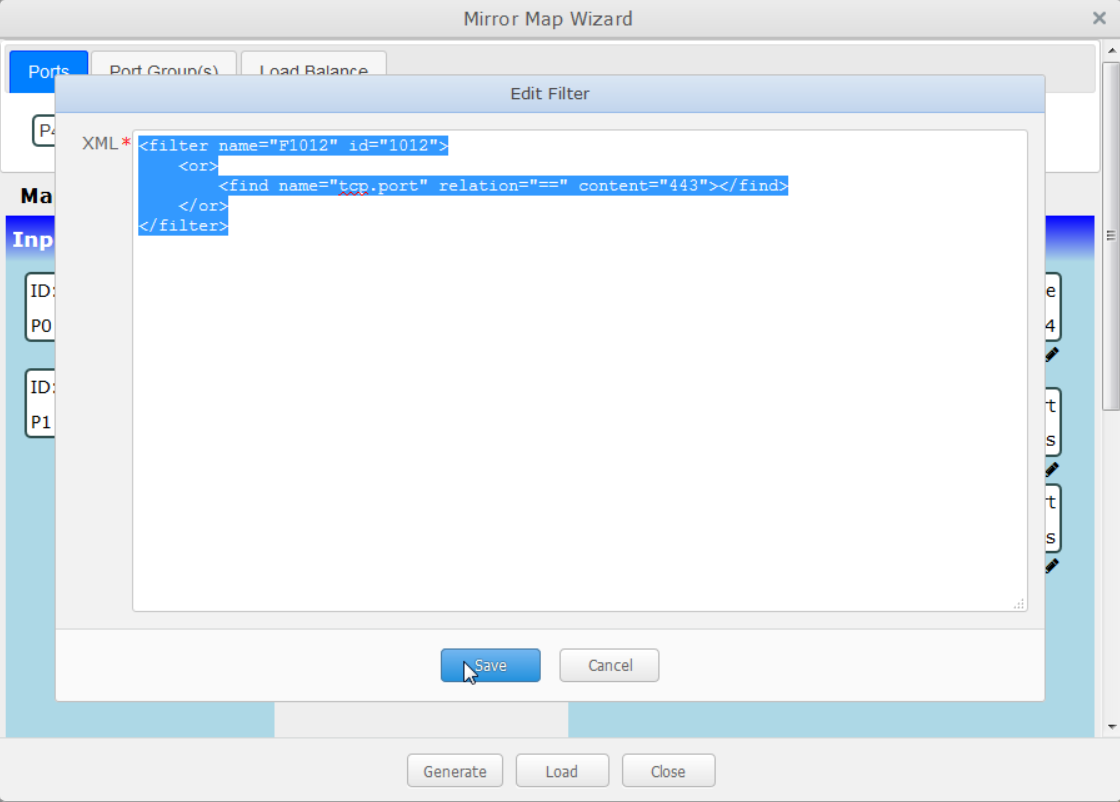
1. 可以透過GUI方式拖拉產生設定檔，將P0、P1拖拉到下方Input Port區域當作輸入埠，並設定一個Rule ID:F1009，將此二埠流量設定成Aggregator在一起，在Outputs輸出埠設定二個Rule，Rule ID:F1010及Rule ID:F1011，Rule ID:F1010為做Load Balance前之一個規則，Rule ID:F1011為做一個過濾規則，再將流量輸出至P2埠。



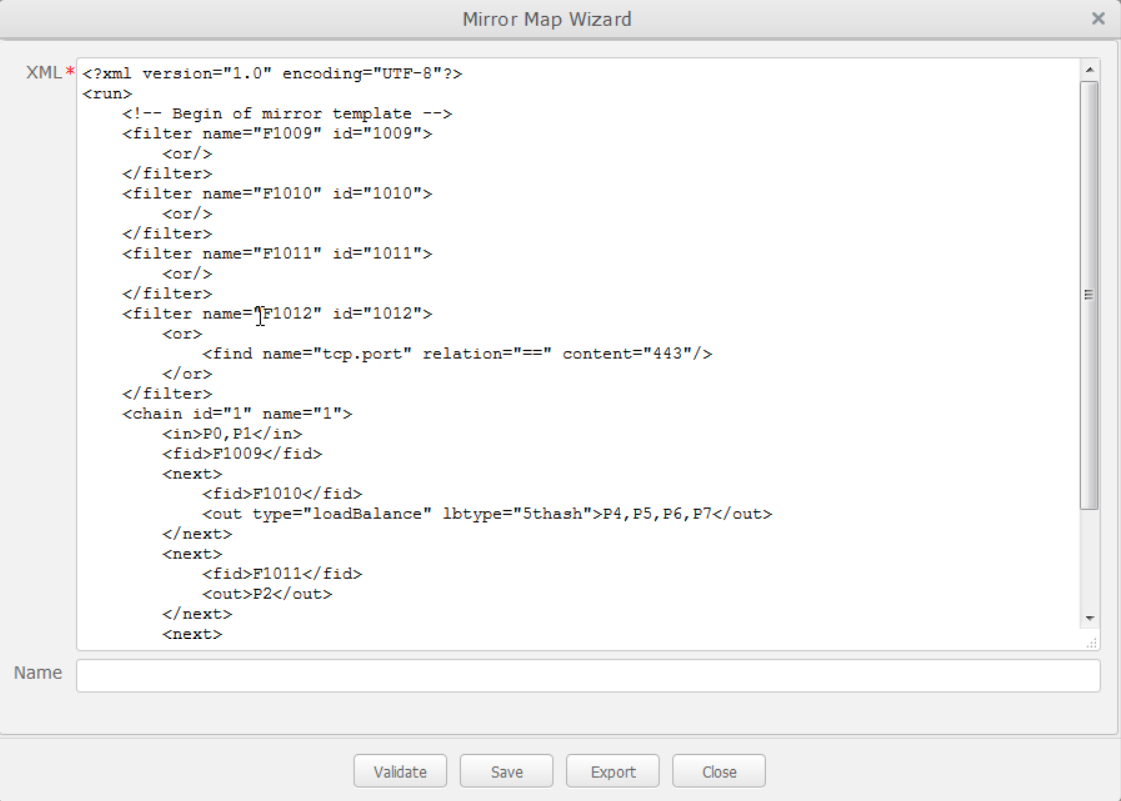
1. 在Outputs輸出埠再設定一個Rule，Rule ID:F1012，Rule ID:F1012為做一個過濾規則，再將流量輸出至P3埠。



1. 點選Rule ID:F1012下方的鉛筆圖案標示，可以直接進行過濾規則之撰寫，範例為針對將TCP 443埠之流量進行過濾。每個Rule ID可以特別撰寫所需之過濾規則



1. 下圖及為每個過濾規則撰寫完後之XML檔案，Rule ID:F1009為空之規則，主要為將P0、P1訊務匯集起來；Rule ID:F1010為空之規則，主要為將匯集後之訊務，使用5thash方式做負載平衡，並將流量平均分給P4、P5、P6、P7埠；Rule ID:F1011為空之規則，主要為將P0、P1訊務匯集至P2埠；；Rule ID:F1012為將P0、P1訊務匯集後，針對將TCP 443埠之流量進行過濾，過濾完後之訊務導至P3埠輸出。



1. 下圖為完整XML範例

<?xml version="1.0" encoding="UTF-8"?>

<run>

<!-- Begin of mirror template -->

<filter name="F1009" id="1009">

<or/>

</filter>

<filter name="F1010" id="1010">

<or/>

</filter>

<filter name="F1011" id="1011">

<or/>

</filter>

<filter name="F1012" id="1012">

<or>

<find name="tcp.port" relation="==" content="443"/>

</or>

</filter>

<chain id="1" name="1">

<in>P0,P1</in>

<fid>F1009</fid>

<next>

<fid>F1010</fid>

<out type="loadBalance" lbtype="5thash">P4,P5,P6,P7</out>

</next>

<next>

<fid>F1011</fid>

<out>P2</out>

</next>

<next>

<fid>F1012</fid>

<out>P3</out>

</next>

</chain>

<!-- End of mirror template -->

</run>

# RESTful API

1. 取得機器ID

https://192.168.1.160/grism/get\_machine\_id

1. 取得系統設定檔

https://192.168.1.160/grism/task/get\_config

1. 取得網路介面流量資訊

https://192.168.1.160/grism/task/get\_statistics\_json

1. 取得xml task讀取狀態

https://192.168.1.160/grism/task/get\_status

回傳值如下

{"loading":true, "message":"compile regular expressions into a graph"}

或

{"loading":false}

loading: true 表示正在讀取，這時候可顯示message呈現目前進度

loading: false 表示沒有在讀取

# 黑名單阻擋功能

1. 上傳黑名單

* 產生一個 run1.xml 檔案，內容為IP/domain黑名單

<?xml version="1.0" encoding="UTF-8"?>

<run>

<filter id="1" name="ip" sessionBase="yes" matchedlog="yes">

<or>

<find name="ip.addr" relation="==" content="110.250.158.62" />

<find name="ip.addr" relation="==" content="113.128.212.234" />

<find name="ip.addr" relation="==" content="113.161.194.221" />

</or>

</filter>

<filter id=" 2" name="domain list" matchedlog=“yes”>

<or>

<find name="dns.qry.name" relation="==" content=" google.com" />

<find name="dns.qry.name" relation="==" content=" facebook.com" />

<find name="dns.qry.name" relation="==" content=" twitter.com" />

</or>

<chain id="0">

<in>P0</in>

<fid>F1</fid>

<next>

<fid>F2</fid>

<out>P2</out>

</next>

</chain>

</run>

* <filter> 欄位說明
  1. id: 範圍10000- 2^31, 依使用者需求給予適當值。為唯一值，不能重複
  2. sessionBase:
     1. no: 以封包為單位來過濾內容
     2. yes: 以連線(flow/session)為單位來過濾內容，舉例來說如連線中有一個封包符合過濾條件，整條連線都會被視為符合條件
  3. matchedlog:
     1. no: 當符合條件時不送出syslog
     2. yes: 當符合條件時送出syslog
  4. <or></or>, <and></and>: 包在<find>外面，決定find要用or還是and
  5. <find> 欄位說明

1. id: 範圍10000- 2^31, 依使用者需求給予適當值。為唯一值，不能重複
2. name: 過濾條件，例如ip.addr, ip.src, ip.dst等共60幾種，
3. relation: == 或 !=
4. content: 依過濾條件會有不同內容

* <chain>欄位說明

1. id: 範圍10000- 2^31, 依使用者需求給予適當值。為唯一值，不能重複
2. <in></in>: 設定從哪個Port進來的流量，可以多個，例如P1,P2
3. <fid></fid>: 設定要經過哪個filter例如F10000就對應到filter id=10000
4. <out></out>: 經過filter後，設定從哪個Port出去，可以多個，例如P3,P4。0為特殊寫法，代表丟掉
5. <next></next>: 根據filter條件決定是否繼續往裡面走
   1. type:
      1. match: 如果符合filter條件就繼續往裡面走(default value)
      2. notmatch: 如果不符合filter條件就繼續往裡面走

* GRISM內部有sftp server，使用者可使用sftp上傳run1.xml到主機 /data/ftproot目錄下，接下來

(sftp server之IP即為管理介面IP)

1. 產生run1.xml.ok檔案(不需要內容)，上傳run1.xml.ok到主機 /data/ftproot目錄下
2. 如果要加密run1.xml檔案，產生run1.xml.okc檔案(不需要內容)，上傳run1.xml.okc到主機 /data/ftproot目錄下
3. 送出命中黑名單syslog

產生一個configSet.xml檔案，檔案內容為

<configSet reboot="no">

<log>

<enable>True</enable>

<syslog>

<enable>True</enable>

<port>M0</port>

<target>

<enable>True</enable>

<dip>192.168.1.201</dip>

<dport>514</dport>

<interfaces>all</interfaces>

<filter></filter>

<type>matched</type>

<subtype>

<sip>True</sip>

<dip>True</dip>

<sport>True</sport>

<dport>True</dport>

<protocol>True</protocol>

<find\_id>True</find\_id>

<find\_content>True</find\_content>

</subtype>

</target>

</syslog>

</log>

</configSet>

其中log->syslog->target->dip要改成syslog receiver的ip

其中log->syslog->target->dport要改成syslog receiver的port

使用sftp上傳configSet.xml到主機 /data/ftproot目錄下，產生configSet.xml.ok檔案(不需要內容)，上傳configSet.xml.ok到主機 /data/ftproot目錄下後，如果有成功，可觀察由restful API取得之系統設定中的log->syslog->target觀察設定是否成功

1. 黑名單syslog欄位說明

|  |  |  |
| --- | --- | --- |
| Tag | 說明 | 備註 |
| datetime | Time | Format: yyyy-mm-dd hh:mm:ss  Ex. 2019-09-09 17:15:10 |
| type | Grism Type Number | Ex. 1: (matched符合過濾條件) |
| sip | Source IP Address | Ex. 192.168.1.10 |
| dip | Destination IP Address | Ex. 8.8.8.8 |
| sport | Source Port | Ex. 56789 |
| dport | Destination Port | Ex. 53 |
| protocol | IP Protocol Number | Ex. 17 |
| find\_id | Grism Find Id Number | Ex. 30030 |
| find\_content | Grism Find Content | Ex. .8.8.8.8 or www.google.com |