

The FlowMatrix WebServices SOAP API

In this documentation you will find description of most of WebServices API available for integration with FlowMatrix network anomaly detection engine. Please keep in mind that architecture of FlowMatrix is structured in such a way that Web GUI communicates with anomaly engine using same API you will find documented below. So all of the information you see in FlowMatrix WebGUI interface is available using these API. You can see all of the calls and data types by browsing WSDL file located in you installation of FlowMatrix:

C:\Flow Matrix\Apache2\htdocs\flowmatrix\netflowalarm.wsdl

Some of API's have example code which is mostly taken from our PHP WebGUI interface. Some APIs, which seem to be simple enough don't have example code but if needed, it can be provided. Please send your emails to support@akmalabs.com

If you think you would like to get some information from FlowMatrix anomaly engine but you don't find it, please ask us and we will if we can add or change API call to provide more information you are looking for.

Happy integration using FlowMatrix and thank you very much for using FlowMatrix!!!

SOAP operation name: setConfigOption

Description:

Saves to FlowMatrix database value for given option name. The FlowMatrix maintains large number of options in configurations table but you can use this API call to save some options, which are needed for your integration script etc. Please keep in mind that if you specify option name same as one used by FlowMatrix you can overwrite option used by FlowMatrix anomaly engine and cause problems. To avoid such collisions it is advisable to append name space to option name, for example: myoptions.querytime. Please note that option value is in string format, so if you need to save numeric value you need to convert it to string first.

Request Parameters:

Name	Description	Notes
optionName	Name of the option	
optionValue	Value string of the options	

Response Parameters:

Name	Description	Notes
Result	Error code, 0 – success anything else – error.	

Examples:

Example XML Request:

```
<SOAP-ENV:Body>
  <ns1:setConfigOption>
    <optionName xsi:type="xsd:string">"some option name"</optionName>
    <optionValue xsi:type="xsd:string">"some option value"</optionValue>
  </ns1:setConfigOption>
</SOAP-ENV:Body>
```

Example XML Response:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:setConfigOptionResponse>
    <result xsi:type="xsd:unsignedInt">0</result>
  </ns1:setConfigOptionResponse>
</SOAP-ENV:Body>
```

Example call: php

SOAP operation name: getConfigOption

Description:

Gets option value for given option name. As described above, for set version of the function, you can use this interface to store and retrieve not only FlowMatrix name=value pairs but also some options needed for your integration script.

Request Parameters:

Name	Description	Notes
optionName	Name of option	

Response Parameters:

Name	Description	Notes
Result	Value of string type with value for given option name.	

Examples:

Example XML Request:

```

<SOAP-ENV:Body>
  <ns1:getConfigOption>
    <optionName xsi:type="xsd:string"></optionName>
  </ns1:getConfigOption>
</SOAP-ENV:Body>

```

Example XML Response:

```

<SOAP-ENV:Body>
  <ns1:getConfigOptionResponse>
    <result xsi:type="xsd:string"></result>
  </ns1:getConfigOptionResponse>
</SOAP-ENV:Body>

```

Example call: php

SOAP operation name: getRuleAlarmRecords

Description:

Gets list of events generated by the alarm rules (rule engine is addition to anomaly engine and uses only configured rules for monitoring and doesn't no statistical models used by anomaly detection engine). Set of conditions can be specified to select events of interest: [BeginTime](#), [EndTime](#), [DeviceIp](#), [RuleId](#), [RuleId](#), [EventSrcIp](#), [EventSrcIp](#), [EventDstIp](#), [EventSrcPort](#), [EventDstPort](#).

Request Parameters:

Name	Description	Notes
beginTime	Time starting from which, events must be selected in returned result sequence. In order for this field to be considered make sure you set bUseBeginTime control flag in conditions data structure of type SOAPRECORDSQUERYCONDITIONS . Please note all key fields have corresponding control flags, please set flag value to 1 in order for FlowMatrix consider provided value.	
endTime	Up to this time include rule alarms in result	
Offset	Control offset option in database SELECT command used to get list of alarm records.	
maxRecords	Maximum records to include in response.	
deviceIp	IP address of reporting NetFlow device in decimal form as signed 32 bit integer (not dot decimal form).	
ruleId	Id of the rule for which to select alarm records	
srcIp	IP address of Src IP address in alarm in decimal	

	form as signed 32 bit integer (no dot decimal form).	
dstIp	IP address of Dst IP address in alarm in decimal form as signed 32 bit integer (no dot decimal form).	
srcPort	Source port	
dstPort	Destination port	

Response Parameters:

Name	Description	Notes
totalRecords	Total number of events in sequence that follow	
eventSeq	Sequence ID of this alarm record	
	For other record parameters see example XML response or browse all the field in SOAPALARMRULERECORDSRESULT below	

Examples:

Example XML Request:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:getRuleAlarmRecords>
    <cond xsi:type="ns1:SOAPRECORDSQUERYCONDITIONS">
      <bUseBeginTime xsi:type="xsd:unsignedByte">0</bUseBeginTime>
      <bUseEndTime xsi:type="xsd:unsignedByte">0</bUseEndTime>
      <bUseDeviceIp xsi:type="xsd:unsignedByte">0</bUseDeviceIp>
      <bUseRuleId xsi:type="xsd:unsignedByte">0</bUseRuleId>
      <bUseEventSrcIp xsi:type="xsd:unsignedByte">0</bUseEventSrcIp>
      <bUseEventDstIp xsi:type="xsd:unsignedByte">0</bUseEventDstIp>
      <bUseEventSrcPort xsi:type="xsd:unsignedByte">0</bUseEventSrcPort>
      <bUseEventDstPort xsi:type="xsd:unsignedByte">0</bUseEventDstPort>
      <beginTime xsi:type="ns1:SOAPNETFLOWTIME">
        <year xsi:type="xsd:unsignedInt">0</year>
        <month xsi:type="xsd:unsignedInt">0</month>
        <day xsi:type="xsd:unsignedInt">0</day>
        <hour xsi:type="xsd:unsignedInt">0</hour>
        <minute xsi:type="xsd:unsignedInt">0</minute>
        <second xsi:type="xsd:unsignedInt">0</second>
        <ms xsi:type="xsd:unsignedInt">0</ms>
      </beginTime>
      <endTime xsi:type="ns1:SOAPNETFLOWTIME">
        <year xsi:type="xsd:unsignedInt">0</year>
        <month xsi:type="xsd:unsignedInt">0</month>
        <day xsi:type="xsd:unsignedInt">0</day>
        <hour xsi:type="xsd:unsignedInt">0</hour>
        <minute xsi:type="xsd:unsignedInt">0</minute>
      </endTime>
    </cond>
  </ns1:getRuleAlarmRecords>
</SOAP-ENV:Body>
```

```

<second xsi:type="xsd:unsignedInt">0</second>
<ms xsi:type="xsd:unsignedInt">0</ms>
</endTime>
<offset xsi:type="xsd:int">0</offset>
<maxRecords xsi:type="xsd:int">0</maxRecords>
<deviceIp xsi:type="xsd:string"></deviceIp>
<ruleId xsi:type="xsd:unsignedInt">0</ruleId>
<srcIp xsi:type="xsd:string"></srcIp>
<dstIp xsi:type="xsd:string"></dstIp>
<srcPort xsi:type="xsd:string"></srcPort>
<dstPort xsi:type="xsd:string"></dstPort>
</cond>
</ns1:getRuleAlarmRecords>
</SOAP-ENV:Body>

```

Example XML Response:

```

<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<ns1:SOAPALARMRULERRECORDSRESULT>
<returnCode xsi:type="xsd:unsignedInt">0</returnCode>
<result xsi:type="ns1:SOAPALARMRULERRECORDS">
<totalRecords xsi:type="xsd:unsignedInt">0</totalRecords>
<records xsi:type="ns1:SOAPALARMRECORD">
<eventSeq xsi:type="ns1:SOAPEVENTSEQUENCE">
<lowPart xsi:type="xsd:unsignedInt">0</lowPart>
<highPart xsi:type="xsd:unsignedInt">0</highPart>
</eventSeq>
<eventTime xsi:type="ns1:SOAPNETFLOWTIME">
<year xsi:type="xsd:unsignedInt">0</year>
<month xsi:type="xsd:unsignedInt">0</month>
<day xsi:type="xsd:unsignedInt">0</day>
<hour xsi:type="xsd:unsignedInt">0</hour>
<minute xsi:type="xsd:unsignedInt">0</minute>
<second xsi:type="xsd:unsignedInt">0</second>
<ms xsi:type="xsd:unsignedInt">0</ms>
</eventTime>
<ruleId xsi:type="xsd:unsignedInt">0</ruleId>
<devIp xsi:type="xsd:unsignedInt">0</devIp>
<inIntId xsi:type="xsd:unsignedInt">0</inIntId>
<outIntId xsi:type="xsd:unsignedInt">0</outIntId>
<srcIp xsi:type="xsd:unsignedInt">0</srcIp>
<dstIp xsi:type="xsd:unsignedInt">0</dstIp>
<srcPort xsi:type="xsd:unsignedInt">0</srcPort>
<dstPort xsi:type="xsd:unsignedInt">0</dstPort>
<dstIpCount xsi:type="xsd:unsignedInt">0</dstIpCount>
<srcIpCount xsi:type="xsd:unsignedInt">0</srcIpCount>

```

```

<dstPortCount xsi:type="xsd:unsignedInt">0</dstPortCount>
<srcPortCount xsi:type="xsd:unsignedInt">0</srcPortCount>
<totalBytes xsi:type="xsd:unsignedInt">0</totalBytes>
<ruleCondition xsi:type="ns1:SOAPFINDCONDITION">FINDCONDITIONSRCIP</ruleCondition>
</records>
</result>
</ns1:SOAPALARMRULERRECORDSRESULT>
</SOAP-ENV:Body>

```

Example call: php

```

class SOAPNETFLOWTIME
{
    public $year;
    public $month;
    public $day;
    public $hour;
    public $minute;
    public $second;
    public $ms;
    function __construct()
    {
        $this->year = 0;
        $this->month = 0;
        $this->day = 0;
        $this->hour = 0;
        $this->minute = 0;
        $this->second = 0;
        $this->ms = 0;
    }
}

class SOAPRECORDSQUERYCONDITIONS
{
    public $bUseBeginTime;
    public $bUseEndTime;
    public $bUseDeviceIp;
    public $bUseRuleId;
    public $bUseEventSrcIp;
    public $bUseEventDstIp;
    public $bUseEventSrcPort;
    public $bUseEventDstPort;
    public $beginTime;
    public $endTime;
    public $offset;
    public $maxRecords;
    public $deviceIp;
    public $ruleId;
    public $srcIp;
    public $dstIp;
    public $srcPort;
    public $dstPort;

    function __construct()

```

```

    {
        $this->beginTime = new SOAPNETFLOWTIME;
        $this->endTime = new SOAPNETFLOWTIME;
        $this->bUseBeginTime = 0;
        $this->bUseEndTime = 0;
        $this->offset = 0;
        $this->maxRecords = 0xffff;
        $this->bUseDeviceIp = 0;
        $this->bUseRuleId = 0;
        $this->bUseEventSrcIp = 0;
        $this->bUseEventDstIp = 0;
        $this->bUseEventSrcPort = 0;
        $this->bUseEventDstPort = 0;
        $this->deviceIp = 0;
        $this->srcIp = 0;
        $this->dstIp = 0;
        $this->srcPort = 0;
        $this->dstPort = 0;
    }
}

$query = new SOAPRECORDSQUERYCONDITIONS;

$query->bUseDeviceIp = $formData['devip'] ? 1 : 0;
$query->bUseEventDstIp = $formData['dstip'] ? 1 : 0;
$query->bUseEventDstPort = $formData['dstport'] ? 1 : 0;
$query->bUseRuleId = $formData['rule'] ? 1 : 0;
$query->bUseEventSrcIp = $formData['srcip'] ? 1 : 0;
$query->bUseEventSrcPort = $formData['srcport'] ? 1 : 0;

if ($formData['stime'])
{
    $query->bUseBeginTime = 1;

    $utime = gmdate(DATE_ATOM, strtotime($formData['stime']));
    $timezone = date_default_timezone_get(); // save the current time zone
    date_default_timezone_set("GMT");
    $utime = getdate(strtotime($utime));
    @list($query->beginTime->second, $query->beginTime->minute,
    $query->beginTime->hour, $query->beginTime->day,,
    $query->beginTime->month, $query->beginTime->year) = array_values($utime);
    date_default_timezone_set($timezone); // restore timezone
}
else
{
    $query->bUseBeginTime = 0;
}

if ($formData['etime'])
{
    $query->bUseEndTime = 1;

    $utime = gmdate(DATE_ATOM, strtotime($formData['etime']));
    $timezone = date_default_timezone_get(); // save the current time zone
    date_default_timezone_set("GMT");
    $utime = getdate(strtotime($utime));
}

```

```

        @list($query->endTime->second, $query->endTime->minute,
        $query->endTime->hour, $query->endTime->day,,
        $query->endTime->month, $query->endTime->year) = array_values($utime);
        date_default_timezone_set($timezone); // restore timezone
    }
else
{
    $query->bUseEndTime = 0;
}

$query->deviceIp = ip2long($formData['devip']);
$query->dstIp = ip2long($formData['dstip']);
$query->dstPort = $formData['dstport'];
$query->ruleId = $formData['rule'];
$query->srcIp = ip2long($formData['srcip']);
$query->srcPort = $formData['srcport'];

$paged_data = Pager_Wrapper_Soap("getRuleAlarmRecords", $query, $params);

if ($paged_data['data'] != false)
{
    // you got your events, now process them
}

```

SOAP operation name: getHostsProfilesTotalCount

Description:

Returns total number of Hosts profiles used for creating rules. Use total counter as limit in for loop to enumerate all hosts profiles.

Request Parameters:

None

Response Parameters:

Name	Description	Notes
result	Returns total number of host profiles in database	

Examples:

Example XML Request:

```

<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:getHostsProfilesTotalCount>
  </ns1:getHostsProfilesTotalCount>
</SOAP-ENV:Body>

```

Example XML Response:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:getHostsProfilesTotalCountResponse>
    <result xsi:type="xsd:unsignedInt">0</result>
  </ns1:getHostsProfilesTotalCountResponse>
</SOAP-ENV:Body>
```

Example call: php

SOAP operation name: getHostsProfiles

Description:

Gets sequence of hosts profiles which satisfy request conditions.

Request Parameters:

Name	Description	Notes
fromProfileId	Begin from profile id, specify 0 if don't care	
max-profiles	Maximum number of profiles to return in result	

Response Parameters:

Returns 0 or more hosts profiles in sequence.

Name	Description	Notes
profileId	Id of current profile	
profileName	Name of profile as assigned when profile was created.	
includeHosts	Include hosts IP addresses separated by space in signed integer numeric form (no dot decimal).	
excludeHosts	Exclude hosts IP addresses range pairs with dash separated by space in signed integer numeric form (no dot decimal).	
includeHostsRange	Include hosts IP addresses range pairs with dash separated by space in signed integer numeric form (no dot decimal).	
excludeHostsRange	Exclude hosts IP addresses range pairs with dash separated by space in signed integer numeric form (no dot decimal).	

Examples:

Example XML Request:

```

<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:getHostsProfiles>
    <fromProfileId xsi:type="xsd:unsignedInt">0</fromProfileId>
    <max-profiles xsi:type="xsd:unsignedInt">0</max-profiles>
  </ns1:getHostsProfiles>
</SOAP-ENV:Body>

```

Example XML Response:

```

<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:SOAPHOSTSPROFILESRESULT>
    <returnCode xsi:type="xsd:unsignedInt">0</returnCode>
    <result xsi:type="ns1:SOAPHOSTSPROFILES">
      <hostsProfiles xsi:type="ns1:SOAPHOSTPROFILE">
        <profileId xsi:type="xsd:unsignedInt">0</profileId>
        <profileName xsi:type="xsd:string"></profileName>
        <includeHosts xsi:type="xsd:string"></includeHosts>
        <excludeHosts xsi:type="xsd:string"></excludeHosts>
        <includeHostsRange xsi:type="xsd:string"></includeHostsRange>
        <excludeHostsRange xsi:type="xsd:string"></excludeHostsRange>
      </hostsProfiles>
    </result>
  </ns1:SOAPHOSTSPROFILESRESULT>
</SOAP-ENV:Body>

```

Example call: php

SOAP operation name: getHostProfile

Description:

Request Parameters:

Name	Description	Notes
profileId	Profile id of the hosts for which to return profile record.	

Response Parameters:

Returns single profile 'returnCode' is equal to 0 (success). For description of the fields in returned profile id please check description above as provided for getHostsProfile() which returns sequence of hosts profiles.

Name	Description	Notes

Examples:

Example XML Request:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:getHostProfile>
    <profileId xsi:type="xsd:unsignedInt">0</profileId>
  </ns1:getHostProfile>
</SOAP-ENV:Body>
```

Example XML Response:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:SOAPHOSTPROFILERESULT>
    <returnCode xsi:type="xsd:unsignedInt">0</returnCode>
    <result xsi:type="ns1:SOAPHOSTPROFILE">
      <profileId xsi:type="xsd:unsignedInt">0</profileId>
      <profileName xsi:type="xsd:string"></profileName>
      <includeHosts xsi:type="xsd:string"></includeHosts>
      <excludeHosts xsi:type="xsd:string"></excludeHosts>
      <includeHostsRange xsi:type="xsd:string"></includeHostsRange>
      <excludeHostsRange xsi:type="xsd:string"></excludeHostsRange>
    </result>
  </ns1:SOAPHOSTPROFILERESULT>
</SOAP-ENV:Body>
```

Example call: php

```
global $soap_client;

$host = $soap_client->getHostProfile($selectedHostId);

$host = $host["result"];
```

SOAP operation name: updateHostProfile

Description:

Saves updated fields of host profile.

Request Parameters:

Name	Description	Notes
------	-------------	-------

SOAPHOSTPROFILE - updateHostProfile	Updated hosts profile to be saved to database, see field description for this data type as provided for the getHostsProfile() above.	
---	--	--

Response Parameters:

Name	Description	Notes
result	Result of operation, 0 – on success, error anything else.	

Examples:

Example XML Request:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:updateHostProfile>
    <profile xsi:type="ns1:SOAPHOSTPROFILE">
      <profileId xsi:type="xsd:unsignedInt">0</profileId>
      <profileName xsi:type="xsd:string"></profileName>
      <includeHosts xsi:type="xsd:string"></includeHosts>
      <excludeHosts xsi:type="xsd:string"></excludeHosts>
      <includeHostsRange xsi:type="xsd:string"></includeHostsRange>
      <excludeHostsRange xsi:type="xsd:string"></excludeHostsRange>
    </profile>
  </ns1:updateHostProfile>
</SOAP-ENV:Body>
```

Example XML Response:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:updateHostProfileResponse>
    <result xsi:type="xsd:unsignedInt">0</result>
  </ns1:updateHostProfileResponse>
</SOAP-ENV:Body>
```

Example call: php

Please see example code for **createHostProfile** below.

SOAP operation name: createHostProfile

Description:

Creates new host profile and as result of the call returns success or error code for operation.

Request Parameters:

Name	Description	Notes
SOAPHOSTPROFILE	New hosts profile. Profile ID filed in data structure is ignored and created host profile ID will be returned as result of the call	

Response Parameters:

Name	Description	Notes
result	0 – success, any other values – error code	

Examples:

Example XML Request:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:createHostProfile>
    <profile xsi:type="ns1:SOAPHOSTPROFILE">
      <profileId xsi:type="xsd:unsignedInt">0</profileId>
      <profileName xsi:type="xsd:string"></profileName>
      <includeHosts xsi:type="xsd:string"></includeHosts>
      <excludeHosts xsi:type="xsd:string"></excludeHosts>
      <includeHostsRange xsi:type="xsd:string"></includeHostsRange>
      <excludeHostsRange xsi:type="xsd:string"></excludeHostsRange>
    </profile>
  </ns1:createHostProfile>
</SOAP-ENV:Body>
```

Example XML Response:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:createHostProfileResponse>
    <result xsi:type="xsd:unsignedInt">0</result>
  </ns1:createHostProfileResponse>
</SOAP-ENV:Body>
```

Example call: php

```
class SOAPHOSTPROFILE {
    public $profileId;
    public $profileName;
    public $includeHosts;
    public $excludeHosts;
    public $includeHostsRange;
    public $excludeHostsRange;
    function __construct()
```

```

    {
        $this->profileId = NULL;
        $this->profileName = "";
        $this->includeHosts = "";
        $this->excludeHosts = "";
        $this->includeHostsRange = "";
        $this->excludeHostsRange = "";
    }
}

$host = new SOAPHOSTPROFILE();

$host->profileId = $isNew ? 0 : $formData["selection"];

$host->profileName = $formData["name"];
$host->includeHosts = $formData["inhosts"];
$host->excludeHosts = $formData["exhosts"];
/* temp disable ranges
$host->includeHostsRange = $formData["inhostsrange"];
$host->excludeHostsRange = $formData["exhostsrange"];
*/

if ($isNew)
{
    $result = $soap_client->createHostProfile($host);
}
else
{
    $result = $soap_client->updateHostProfile($host);
}

```

SOAP operation name: deleteHostProfile

Description:

Deletes hosts profile given hosts profile ID.

Request Parameters:

Name	Description	Notes
profileId	Profile ID to delete.	

Response Parameters:

Name	Description	Notes
result	Indicates result of delete operation, 0-success, anything else – error.	

Examples:

Example XML Request:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:deleteHostProfile>
    <profileId xsi:type="xsd:unsignedInt">0</profileId>
  </ns1:deleteHostProfile>
</SOAP-ENV:Body>
```

Example XML Response:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:deleteHostProfileResponse>
    <result xsi:type="xsd:unsignedInt">0</result>
  </ns1:deleteHostProfileResponse>
</SOAP-ENV:Body>
```

Example call: php

SOAP operation name: getAppProfilesTotalCount

Description:

Returns total count of application profiles.

Request Parameters:

None

Response Parameters:

Name	Description	Notes
result	Contains total number of application profiles	

Examples:

Example XML Request:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:getAppProfilesTotalCount>
</ns1:getAppProfilesTotalCount>
</SOAP-ENV:Body>
```

Example XML Response:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:getAppProfilesTotalCountResponse>
```

```
<result xsi:type="xsd:unsignedInt">0</result>
</ns1:getAppProfilesTotalCountResponse>
</SOAP-ENV:Body>
```

Example call: php

SOAP operation name: getAppProfiles

Description:

Gets sequence of application profiles, beginning from 'fromProfileId' and limited by maximum number of profiles.

Request Parameters:

Name	Description	Notes
fromProfileId	Application profiles returned started from profile ID.	
max-profiles	Maximum application profiles to return in this call.	

Response Parameters:

Result returns sequence of 1 or more application profiles of type SOAPAPPLICATIONPROFILE. See below all the fields for given data type.

Name	Description	Notes
profileId	Application profile ID	
profileName	Application profile name	
includeUdpApps	List of UDP ports to include for given application profile, or * wild card to indicate that include all ports.	
excludeUdpApps	List of UDP ports to exclude from given application profile.	
includeTcpApps	List of TCP ports to include for given application profile, or * wild card to indicate that include all ports.	
excludeTcpApps	List of TCP ports to exclude from given application profile.	

Examples:

Example XML Request:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:getAppProfiles>
    <fromProfileId xsi:type="xsd:unsignedInt">0</fromProfileId>
```

```

<max-profiles xsi:type="xsd:unsignedInt">0</max-profiles>
</ns1:getAppProfiles>
</SOAP-ENV:Body>

```

Example XML Response:

```

<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<ns1:SOAPAPPLICATIONPROFILESRESULT>
<returnCode xsi:type="xsd:unsignedInt">0</returnCode>
<result xsi:type="ns1:SOAPAPPLICATIONPROFILES">
<appProfiles xsi:type="ns1:SOAPAPPLICATIONPROFILE">
<profileId xsi:type="xsd:unsignedInt">0</profileId>
<profileName xsi:type="xsd:string"></profileName>
<includeUdpApps xsi:type="xsd:string"></includeUdpApps>
<excludeUdpApps xsi:type="xsd:string"></excludeUdpApps>
<includeTcpApps xsi:type="xsd:string"></includeTcpApps>
<excludeTcpApps xsi:type="xsd:string"></excludeTcpApps>
</appProfiles>
</result>
</ns1:SOAPAPPLICATIONPROFILESRESULT>
</SOAP-ENV:Body>

```

Example call: php

```

global $soap_client;
//
// get all available hosts profiles and put them into array
$profiles = $soap_client->getAppProfiles(0, 0xffff);

```

SOAP operation name: getAppProfile

Description:

Gets application profile as specified by given application profile ID.

Request Parameters:

Name	Description	Notes
profileId	Profile ID of application to return.	

Response Parameters:

Return application profile. For fields descriptions for returned data type SOAPAPPLICATIONPROFILE see table below.

Name	Description	Notes
profileId	Application profile ID	

profileName	Application profile name	
includeUdpApps	List of UDP ports to include for given application profile, or * wild card to indicate that include all ports.	
excludeUdpApps	List of UDP ports to exclude from given application profile.	
includeTcpApps	List of TCP ports to include for given application profile, or * wild card to indicate that include all ports.	
excludeTcpApps	List of TCP ports to exclude from given application profile.	

Examples:

Example XML Request:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:getAppProfile>
    <profileId xsi:type="xsd:unsignedInt">0</profileId>
  </ns1:getAppProfile>
</SOAP-ENV:Body>
```

Example XML Response:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:SOAPAPPLICATIONPROFILERESULT>
    <returnCode xsi:type="xsd:unsignedInt">0</returnCode>
    <result xsi:type="ns1:SOAPAPPLICATIONPROFILE">
      <profileId xsi:type="xsd:unsignedInt">0</profileId>
      <profileName xsi:type="xsd:string"></profileName>
      <includeUdpApps xsi:type="xsd:string"></includeUdpApps>
      <excludeUdpApps xsi:type="xsd:string"></excludeUdpApps>
      <includeTcpApps xsi:type="xsd:string"></includeTcpApps>
      <excludeTcpApps xsi:type="xsd:string"></excludeTcpApps>
    </result>
  </ns1:SOAPAPPLICATIONPROFILERESULT>
</SOAP-ENV:Body>
```

Example call: php

SOAP operation name: updateAppProfile

Description:

Saves updated application profile to database. Sets all application profile fields to proper value and then perform update operation, all values will be saved to database, not only updated.

Request Parameters:

For fields descriptions for returned data type SOAPAPPLICATIONPROFILE see table below.

Name	Description	Notes
profileId	Application profile ID	
profileName	Application profile name	
includeUdpApps	List of UDP ports to include for given application profile, or * wild card to indicate that include all ports.	
excludeUdpApps	List of UDP ports to exclude from given application profile.	
includeTcpApps	List of TCP ports to include for given application profile, or * wild card to indicate that include all ports.	
excludeTcpApps	List of TCP ports to exclude from given application profile.	

Response Parameters:

Name	Description	Notes
result	Return result code: 0- success, any other value on error	

Examples:

Example XML Request:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:updateAppProfile>
    <profile xsi:type="ns1:SOAPAPPLICATIONPROFILE">
      <profileId xsi:type="xsd:unsignedInt">0</profileId>
      <profileName xsi:type="xsd:string"></profileName>
      <includeUdpApps xsi:type="xsd:string"></includeUdpApps>
      <excludeUdpApps xsi:type="xsd:string"></excludeUdpApps>
      <includeTcpApps xsi:type="xsd:string"></includeTcpApps>
      <excludeTcpApps xsi:type="xsd:string"></excludeTcpApps>
    </profile>
  </ns1:updateAppProfile>
</SOAP-ENV:Body>
```

Example XML Response:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:updateAppProfileResponse>
```

```
<result xsi:type="xsd:unsignedInt">0</result>
</ns1:updateAppProfileResponse>
</SOAP-ENV:Body>
```

Example call: [php](#)

See example for **createAppProfile** below.

SOAP operation name: createAppProfile

Description:

Saves newly created application profile to database. Returns profile ID for newly created profile.

Request Parameters:

Name	Description	Notes
profileId	Application profile ID	
profileName	Application profile name	
includeUdpApps	List of UDP ports to include for given application profile, or * wild card to indicate that include all ports.	
excludeUdpApps	List of UDP ports to exclude from given application profile.	
includeTcpApps	List of TCP ports to include for given application profile, or * wild card to indicate that include all ports.	
excludeTcpApps	List of TCP ports to exclude from given application profile.	

Response Parameters:

Name	Description	Notes
Result	Contains profile ID of newly created application profile.	

Examples:

Example XML Request:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:createAppProfile>
    <profile xsi:type="ns1:SOAPAPPLICATIONPROFILE">
      <profileId xsi:type="xsd:unsignedInt">0</profileId>
      <profileName xsi:type="xsd:string"></profileName>
      <includeUdpApps xsi:type="xsd:string"></includeUdpApps>
```

```

<excludeUdpApps xsi:type="xsd:string"></excludeUdpApps>
<includeTcpApps xsi:type="xsd:string"></includeTcpApps>
<excludeTcpApps xsi:type="xsd:string"></excludeTcpApps>
</profile>
</ns1:createAppProfile>
</SOAP-ENV:Body>

```

Example XML Response:

```

<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<ns1:createAppProfileResponse>
<result xsi:type="xsd:unsignedInt">0</result>
</ns1:createAppProfileResponse>
</SOAP-ENV:Body>

```

Example call: php

```

class SOAPAPPLICATIONPROFILE {
    public $profileId;
    public $profileName;
    public $includeUdpApps;
    public $excludeUdpApps;
    public $includeTcpApps;
    public $excludeTcpApps;
    function __construct()
    {
        $this->profileId = NULL;
        $this->profileName = "";
        $this->includeUdpApps = "";
        $this->excludeUdpApps = "";
        $this->includeTcpApps = "";
        $this->excludeTcpApps = "";
    }
}

$app = new SOAPAPPLICATIONPROFILE();

$app->profileId = $isNew ? 0 : $formData["selection"];

$app->profileName = $formData["name"];
$app->includeUdpApps = $formData["inudp"];
$app->excludeUdpApps = $formData["exudp"];
$app->includeTcpApps = $formData["intcp"];
$app->excludeTcpApps = $formData["extcp"];

if ($isNew)
{
    $result = $soap_client->createAppProfile($app);
}
else
{
    $result = $soap_client->updateAppProfile($app);
}

```

}

SOAP operation name: deleteAppProfile

Description:

Deletes application profile for given profile ID.

Request Parameters:

Name	Description	Notes
profileId	Profile ID of application profile to delete	

Response Parameters:

Name	Description	Notes
Result	Contains delete operation result: 0-success, anything else – error	

Examples:

Example XML Request:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">  
  <ns1:deleteAppProfile>  
    <profileId xsi:type="xsd:unsignedInt">0</profileId>  
  </ns1:deleteAppProfile>  
</SOAP-ENV:Body>
```

Example XML Response:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">  
  <ns1:deleteAppProfileResponse>  
    <result xsi:type="xsd:unsignedInt">0</result>  
  </ns1:deleteAppProfileResponse>  
</SOAP-ENV:Body>
```

Example call: php

```
global $soap_client;  
  
$result = $soap_client->deleteAppProfile($selectedAppId);
```

SOAP operation name: getAlarmRulesTotalCount

Description:

Gets total number of alarm rules in database.

Request Parameters: None

Response Parameters:

Name	Description	Notes
Result	Returns result of the call: 0-success, anything else – error	

Examples:

Example XML Request:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">  
  <ns1:getAlarmRulesTotalCount>  
  </ns1:getAlarmRulesTotalCount>  
</SOAP-ENV:Body>
```

Example XML Response:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">  
  <ns1:getAlarmRulesTotalCountResponse>  
    <result xsi:type="xsd:unsignedInt">0</result>  
  </ns1:getAlarmRulesTotalCountResponse>  
</SOAP-ENV:Body>
```

Example call: php

SOAP operation name: getAlarmRules

Description:

Gets alarm rules from database. Number of rules is limited by argument ‘max-rules’. Only rules with profile ID > then ‘ruleId’ will be included in result.

Request Parameters:

Name	Description	Notes
ruleId	Ruled ID from which to start including rules in result.	
max-rules	Maximum number of rules to include.	

Response Parameters:

Returns sequence of alarm rules, for rules fields check Example XML response below.

Examples:

Example XML Request:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:getAlarmRules>
    <ruleId xsi:type="xsd:unsignedInt">0</ruleId>
    <max-rules xsi:type="xsd:unsignedInt">0</max-rules>
  </ns1:getAlarmRules>
</SOAP-ENV:Body>
```

Example XML Response:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:SOAPNETFLOWALARMRULESRESULT>
    <returnCode xsi:type="xsd:unsignedInt">0</returnCode>
    <result xsi:type="ns1:SOAPNETFLOWALARMRULES">
      <rules xsi:type="ns1:SOAPNETFLOWALARMRULE">
        <ruleId xsi:type="xsd:unsignedInt">0</ruleId>
        <ruleName xsi:type="xsd:string"></ruleName>
        <enabled xsi:type="xsd:unsignedInt">0</enabled>
        <color xsi:type="xsd:unsignedInt">0</color>
        <find xsi:type="ns1:SOAPFINDCONDITION">FINDCONDITIONSRCIP</find>
        <byteCondition xsi:type="ns1:SOAPCOUNTCONDITION">
          <op xsi:type="ns1:SOAPCONDOPERATOR">GT-OP</op>
          <count xsi:type="xsd:unsignedInt">0</count>
        </byteCondition>
        <dstIpCondition xsi:type="ns1:SOAPRULEELEMENT">
          <uniqueCount xsi:type="ns1:SOAPCOUNTCONDITION">
            <op xsi:type="ns1:SOAPCONDOPERATOR">GT-OP</op>
            <count xsi:type="xsd:unsignedInt">0</count>
          </uniqueCount>
          <profileId xsi:type="xsd:unsignedInt">0</profileId>
        </dstIpCondition>
        <srcIpCondition xsi:type="ns1:SOAPRULEELEMENT">
          <uniqueCount xsi:type="ns1:SOAPCOUNTCONDITION">
            <op xsi:type="ns1:SOAPCONDOPERATOR">GT-OP</op>
            <count xsi:type="xsd:unsignedInt">0</count>
          </uniqueCount>
          <profileId xsi:type="xsd:unsignedInt">0</profileId>
        </srcIpCondition>
        <dstPortCondition xsi:type="ns1:SOAPRULEELEMENT">
```

```

<uniqueCount xsi:type="ns1:SOAPCOUNTCONDITION">
  <op xsi:type="ns1:SOAPCONDOPERATOR">GT-OP</op>
  <count xsi:type="xsd:unsignedInt">0</count>
</uniqueCount>
<profileId xsi:type="xsd:unsignedInt">0</profileId>
</dstPortCondition>
<srcPortCondition xsi:type="ns1:SOAPRULEELEMENT">
  <uniqueCount xsi:type="ns1:SOAPCOUNTCONDITION">
    <op xsi:type="ns1:SOAPCONDOPERATOR">GT-OP</op>
    <count xsi:type="xsd:unsignedInt">0</count>
  </uniqueCount>
  <profileId xsi:type="xsd:unsignedInt">0</profileId>
</srcPortCondition>
<resultLimit xsi:type="xsd:unsignedInt">0</resultLimit>
</rules>
</result>
</ns1:SOAPNETFLOWALARMRULESRESULT>
</SOAP-ENV:Body>

```

Example call: php

```
global $soap_client;
```

```
$keys = array();
```

```
$values = array();
```

```
$profiles = $soap_client->getAlarmRules(0, 0xffff);
```

```
if (isset($profiles['result']->rules))
```

```
{
$soapobj = is_array($profiles['result']->rules) ? $profiles['result']->rules : array($profiles['result']->rules);
```

```
    foreach ($soapobj as $rule)
```

```
    {
        $keys[] = $rule->ruleId;
        $values[] = $rule->ruleName;
    }
```

```
}
```

SOAP operation name: getAlarmRule

Description:

Gets alarm rule from database. Only rule with profile ID 'ruleId' will be included in result.

Request Parameters:

Name	Description	Notes
ruleId	ID of rule to get from database	

Response Parameters:

Returns alarm rule for given rule id, for rules fields check Example XML response below.

Name	Description	Notes

Examples:

Example XML Request:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:getAlarmRule>
    <ruleId xsi:type="xsd:unsignedInt">0</ruleId>
  </ns1:getAlarmRule>
</SOAP-ENV:Body>
```

Example XML Response:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:SOAPNETFLOWALARMRULERESULT>
    <returnCode xsi:type="xsd:unsignedInt">0</returnCode>
    <result xsi:type="ns1:SOAPNETFLOWALARMRULE">
      <ruleId xsi:type="xsd:unsignedInt">0</ruleId>
      <ruleName xsi:type="xsd:string"></ruleName>
      <enabled xsi:type="xsd:unsignedInt">0</enabled>
      <color xsi:type="xsd:unsignedInt">0</color>
      <find xsi:type="ns1:SOAPFINDCONDITION">FINDCONDITIONSRCIP</find>
      <byteCondition xsi:type="ns1:SOAPCOUNTCONDITION">
        <op xsi:type="ns1:SOAPCONDOPERATOR">GT-OP</op>
        <count xsi:type="xsd:unsignedInt">0</count>
      </byteCondition>
      <dstIpCondition xsi:type="ns1:SOAPRULEELEMENT">
        <uniqueCount xsi:type="ns1:SOAPCOUNTCONDITION">
          <op xsi:type="ns1:SOAPCONDOPERATOR">GT-OP</op>
          <count xsi:type="xsd:unsignedInt">0</count>
        </uniqueCount>
        <profileId xsi:type="xsd:unsignedInt">0</profileId>
      </dstIpCondition>
```

```

<srcIpCondition xsi:type="ns1:SOAPRULEELEMENT">
  <uniqueCount xsi:type="ns1:SOAPCOUNTCONDITION">
    <op xsi:type="ns1:SOAPCONDOPERATOR">GT-OP</op>
    <count xsi:type="xsd:unsignedInt">0</count>
  </uniqueCount>
  <profileId xsi:type="xsd:unsignedInt">0</profileId>
</srcIpCondition>
<dstPortCondition xsi:type="ns1:SOAPRULEELEMENT">
  <uniqueCount xsi:type="ns1:SOAPCOUNTCONDITION">
    <op xsi:type="ns1:SOAPCONDOPERATOR">GT-OP</op>
    <count xsi:type="xsd:unsignedInt">0</count>
  </uniqueCount>
  <profileId xsi:type="xsd:unsignedInt">0</profileId>
</dstPortCondition>
<srcPortCondition xsi:type="ns1:SOAPRULEELEMENT">
  <uniqueCount xsi:type="ns1:SOAPCOUNTCONDITION">
    <op xsi:type="ns1:SOAPCONDOPERATOR">GT-OP</op>
    <count xsi:type="xsd:unsignedInt">0</count>
  </uniqueCount>
  <profileId xsi:type="xsd:unsignedInt">0</profileId>
</srcPortCondition>
<resultLimit xsi:type="xsd:unsignedInt">0</resultLimit>
</result>
</ns1:SOAPNETFLOWALARMRULERESULT>
</SOAP-ENV:Body>

```

Example call: [php](#)

SOAP operation name: updateAlarmRule

Description:

Updates rule option with new value and saves it to database.

Request Parameters:

Name	Description	Notes
SOAPNETFLOWALARMRULE	Properly setup rule options to be saved to database. For more details about individual field's names, please check XML example for request below.	

Response Parameters:

Name	Description	Notes
result	Result of operation: 0-success, anything else on error.	

Examples:

Example XML Request:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:updateAlarmRule>
    <rule xsi:type="ns1:SOAPNETFLOWALARMRULE">
      <ruleId xsi:type="xsd:unsignedInt">0</ruleId>
      <ruleName xsi:type="xsd:string"></ruleName>
      <enabled xsi:type="xsd:unsignedInt">0</enabled>
      <color xsi:type="xsd:unsignedInt">0</color>
      <find xsi:type="ns1:SOAPFINDCONDITION">FINDCONDITIONSRCIP</find>
      <byteCondition xsi:type="ns1:SOAPCOUNTCONDITION">
        <op xsi:type="ns1:SOAPCONDOPERATOR">GT-OP</op>
        <count xsi:type="xsd:unsignedInt">0</count>
      </byteCondition>
      <dstIpCondition xsi:type="ns1:SOAPRULEELEMENT">
        <uniqueCount xsi:type="ns1:SOAPCOUNTCONDITION">
          <op xsi:type="ns1:SOAPCONDOPERATOR">GT-OP</op>
          <count xsi:type="xsd:unsignedInt">0</count>
        </uniqueCount>
        <profileId xsi:type="xsd:unsignedInt">0</profileId>
      </dstIpCondition>
      <srcIpCondition xsi:type="ns1:SOAPRULEELEMENT">
        <uniqueCount xsi:type="ns1:SOAPCOUNTCONDITION">
          <op xsi:type="ns1:SOAPCONDOPERATOR">GT-OP</op>
          <count xsi:type="xsd:unsignedInt">0</count>
        </uniqueCount>
        <profileId xsi:type="xsd:unsignedInt">0</profileId>
      </srcIpCondition>
      <dstPortCondition xsi:type="ns1:SOAPRULEELEMENT">
        <uniqueCount xsi:type="ns1:SOAPCOUNTCONDITION">
          <op xsi:type="ns1:SOAPCONDOPERATOR">GT-OP</op>
          <count xsi:type="xsd:unsignedInt">0</count>
        </uniqueCount>
        <profileId xsi:type="xsd:unsignedInt">0</profileId>
      </dstPortCondition>
      <srcPortCondition xsi:type="ns1:SOAPRULEELEMENT">
        <uniqueCount xsi:type="ns1:SOAPCOUNTCONDITION">
          <op xsi:type="ns1:SOAPCONDOPERATOR">GT-OP</op>
          <count xsi:type="xsd:unsignedInt">0</count>
        </uniqueCount>
        <profileId xsi:type="xsd:unsignedInt">0</profileId>
      </srcPortCondition>
      <resultLimit xsi:type="xsd:unsignedInt">0</resultLimit>
    </rule>
  </ns1:updateAlarmRule>
</SOAP-ENV:Body>
```

```
</rule>
</ns1:updateAlarmRule>
</SOAP-ENV:Body>
```

Example XML Response:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:updateAlarmRuleResponse>
    <result xsi:type="xsd:unsignedInt">0</result>
  </ns1:updateAlarmRuleResponse>
</SOAP-ENV:Body>
```

Example call: php

See example for **createAlarmRule** below.

SOAP operation name: createAlarmRule

Description:

Creates new alarm rule.

Request Parameters:

Name	Description	Notes
SOAPNETFLOWALARMRULE rule	Properly setup rule options to be saved to database. For more details about individual fields names, please check XML example for request below.	

Response Parameters:

Name	Description	Notes
Result	Returns result code of operation: 0-success, other value on error.	

Examples:

Example XML Request:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:createAlarmRule>
    <rule xsi:type="ns1:SOAPNETFLOWALARMRULE">
      <ruleId xsi:type="xsd:unsignedInt">0</ruleId>
      <ruleName xsi:type="xsd:string"></ruleName>
      <enabled xsi:type="xsd:unsignedInt">0</enabled>
      <color xsi:type="xsd:unsignedInt">0</color>
    </rule>
  </ns1:createAlarmRule>
</SOAP-ENV:Body>
```

```

<find xsi:type="ns1:SOAPFINDCONDITION">FINDCONDITIONSRCIP</find>
<byteCondition xsi:type="ns1:SOAPCOUNTCONDITION">
  <op xsi:type="ns1:SOAPCONDOPERATOR">GT-OP</op>
  <count xsi:type="xsd:unsignedInt">0</count>
</byteCondition>
<dstIpCondition xsi:type="ns1:SOAPRULEELEMENT">
  <uniqueCount xsi:type="ns1:SOAPCOUNTCONDITION">
    <op xsi:type="ns1:SOAPCONDOPERATOR">GT-OP</op>
    <count xsi:type="xsd:unsignedInt">0</count>
  </uniqueCount>
  <profileId xsi:type="xsd:unsignedInt">0</profileId>
</dstIpCondition>
<srcIpCondition xsi:type="ns1:SOAPRULEELEMENT">
  <uniqueCount xsi:type="ns1:SOAPCOUNTCONDITION">
    <op xsi:type="ns1:SOAPCONDOPERATOR">GT-OP</op>
    <count xsi:type="xsd:unsignedInt">0</count>
  </uniqueCount>
  <profileId xsi:type="xsd:unsignedInt">0</profileId>
</srcIpCondition>
<dstPortCondition xsi:type="ns1:SOAPRULEELEMENT">
  <uniqueCount xsi:type="ns1:SOAPCOUNTCONDITION">
    <op xsi:type="ns1:SOAPCONDOPERATOR">GT-OP</op>
    <count xsi:type="xsd:unsignedInt">0</count>
  </uniqueCount>
  <profileId xsi:type="xsd:unsignedInt">0</profileId>
</dstPortCondition>
<srcPortCondition xsi:type="ns1:SOAPRULEELEMENT">
  <uniqueCount xsi:type="ns1:SOAPCOUNTCONDITION">
    <op xsi:type="ns1:SOAPCONDOPERATOR">GT-OP</op>
    <count xsi:type="xsd:unsignedInt">0</count>
  </uniqueCount>
  <profileId xsi:type="xsd:unsignedInt">0</profileId>
</srcPortCondition>
<resultLimit xsi:type="xsd:unsignedInt">0</resultLimit>
</rule>
</ns1:createAlarmRule>
</SOAP-ENV:Body>

```

Example XML Response:

```

<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:createAlarmRuleResponse>
    <result xsi:type="xsd:unsignedInt">0</result>
  </ns1:createAlarmRuleResponse>
</SOAP-ENV:Body>

```

Example call: php

```
function processRulesForm($formData, $isNew = 0)
{
    global $soap_client;

    $objResponse = new tungusXajaxResponse();

    list($objResponse, $error) = checkInput($formData, $objResponse);

    if (!$error)
    {

        class SOAPCOUNTCONDITION
        {
            public $op; // enum type for >=, = <= or <
            public $count;
            function __construct()
            {
                $this->op = "GEQ-OP";
                $this->count = 0;
            }
        }

        class SOAPRULEELEMENT
        {
            public $uniqueCount; // will hold reference to SOAPCOUNTCONDITION
            public $profileId;
            function __construct()
            {
                $this->uniqueCount = new SOAPCOUNTCONDITION;
                $this->profileId = 1;
            }
        }

        class SOAPNETFLOWALARMRULE
        {
            public $ruleId;
            public $ruleName;
            public $enabled; // 1 when rule is enabled; 0 when rule is disabled
            public $color; // color for the rule
            public $find; // enum type for dst or host OR sr or dst port
            public $byteCondition; // will point to SOAPCOUNTCONDITION
            public $dstIpCondition; // will point to SOAPRULEELEMENT
            public $srcIpCondition; // the same as above
            public $dstPortCondition;
```

```

public $srcPortCondition;
public $resultLimit;
function __construct()
{
    $this->byteCondition = new SOAPCOUNTCONDITION;
    $this->dstIpCondition = new SOAPRULEELEMENT;
    $this->srcIpCondition = new SOAPRULEELEMENT;
    $this->dstPortCondition = new SOAPRULEELEMENT;
    $this->srcPortCondition = new SOAPRULEELEMENT;

    $this->resultLimit = 10; // temp. hardcoded
}
}

$rule = new SOAPNETFLOWALARMRULE();

$rule->ruleId = $isNew ? 0 : $formData["selection"];

$rule->ruleName = $formData["name"];
$rule->enabled = (isset($formData["disabled"]) && $formData["disabled"] == "on") ? 0 : 1;
$rule->color = hexdec($formData["color"]);
$rule->find = $formData["condition"];
$rule->byteCondition->op = $formData["bc_cond"];
$rule->byteCondition->count = $formData["bc_count"] * MB_RATIO;

$map = mapCondition($rule->find);
$rule->$map["ptr"]->uniqueCount->op = $formData["dc_cond"];
$rule->$map["ptr"]->uniqueCount->count = $formData["dc_count"];

$rule->$map["prof1_selected"]->profileId = $formData["prof1"];
$rule->$map["prof2_selected"]->profileId = $formData["prof2"];
$rule->$map["prof3_selected"]->profileId = $formData["prof3"];

/* temp. disabled in this release
$rule->resultLimit = $formData["limit"];
*/

if ($isNew)
{
    $result = $soap_client->createAlarmRule($rule);
}
else
{
    $result = $soap_client->updateAlarmRule($rule);
}

```

```

        $objResponse = loadRules($objResponse);
        $objResponse->addScriptCall("showFeedback");
    }

    return $objResponse;
}

```

SOAP operation name: deleteAlarmRule

Description:

Deletes rule from database using provided rule id.

Request Parameters:

Name	Description	Notes
ruleId	Id of rule to delete	

Response Parameters:

Name	Description	Notes
result	Returns result code of operation: 0-success, other value for error.	

Examples:

Example XML Request:

```

<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:deleteAlarmRule>
    <ruleId xsi:type="xsd:unsignedInt">0</ruleId>
  </ns1:deleteAlarmRule>
</SOAP-ENV:Body>

```

Example XML Response:

```

<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:deleteAlarmRuleResponse>
    <result xsi:type="xsd:unsignedInt">0</result>
  </ns1:deleteAlarmRuleResponse>
</SOAP-ENV:Body>

```

Example call: php

SOAP operation name: getAlarmRulesStatusRecords

Description:

Returns sequence of rule status records for every anomaly rule. Rule status included number of events generated, rule id, rule name, rule description, first time of events, last time of events by this rule and others.

Request Parameters:

None

Response Parameters:

Name	Description	Notes

Examples:

Example XML Request:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:getAlarmRulesStatusRecords>
  </ns1:getAlarmRulesStatusRecords>
</SOAP-ENV:Body>
```

Example XML Response:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:SOAPALARMRULESTATUSRECORDSRESULT>
    <returnCode xsi:type="xsd:unsignedInt">0</returnCode>
    <result xsi:type="ns1:SOAPALARMRULESTATUSRECORDS">
      <records xsi:type="ns1:SOAPALARMRULESTATUSRECORD">
        <ruleId xsi:type="xsd:unsignedInt">0</ruleId>
        <ruleName xsi:type="xsd:string"></ruleName>
        <ruleDescription xsi:type="xsd:string"></ruleDescription>
        <ruleColor xsi:type="xsd:unsignedInt">0</ruleColor>
        <eventsCount xsi:type="xsd:unsignedInt">0</eventsCount>
        <firstTime xsi:type="ns1:SOAPNETFLOWTIME">
          <year xsi:type="xsd:unsignedInt">0</year>
          <month xsi:type="xsd:unsignedInt">0</month>
          <day xsi:type="xsd:unsignedInt">0</day>
          <hour xsi:type="xsd:unsignedInt">0</hour>
          <minute xsi:type="xsd:unsignedInt">0</minute>
```

```

<second xsi:type="xsd:unsignedInt">0</second>
<ms xsi:type="xsd:unsignedInt">0</ms>
</firstTime>
<lastTime xsi:type="ns1:SOAPNETFLOWTIME">
<year xsi:type="xsd:unsignedInt">0</year>
<month xsi:type="xsd:unsignedInt">0</month>
<day xsi:type="xsd:unsignedInt">0</day>
<hour xsi:type="xsd:unsignedInt">0</hour>
<minute xsi:type="xsd:unsignedInt">0</minute>
<second xsi:type="xsd:unsignedInt">0</second>
<ms xsi:type="xsd:unsignedInt">0</ms>
</lastTime>
<deviceIp xsi:type="xsd:unsignedInt">0</deviceIp>
</records>
</result>
</ns1:SOAPALARMRULESTATUSRECORDSRESULT>
</SOAP-ENV:Body>

```

Example call: php

```

$res = $soap_client->getAlarmRulesStatusRecords();

if ( isset($res['result']->records) )
{
    // work with the result
}

```

SOAP operation name: clearAlarmRuleStatusRecords

Description:

Clears all stats counters and deletes all records for this rule, which are before date and time provided as arguments to the call.

Request Parameters:

Name	Description	Notes
ruleId	Id of rule to clear	
clearTime	Records which are older then this date-time argument will be deleted. Only records generated by this rule will be deleted.	

Response Parameters:

Name	Description	Notes
------	-------------	-------

result	Result code of operation: 0-success, anything else on error.	
--------	--	--

Examples:

Example XML Request:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:clearAlarmRuleStatusRecords>
    <ruleId xsi:type="xsd:unsignedInt">0</ruleId>
    <clearTime xsi:type="ns1:SOAPNETFLOWTIME">
      <year xsi:type="xsd:unsignedInt">0</year>
      <month xsi:type="xsd:unsignedInt">0</month>
      <day xsi:type="xsd:unsignedInt">0</day>
      <hour xsi:type="xsd:unsignedInt">0</hour>
      <minute xsi:type="xsd:unsignedInt">0</minute>
      <second xsi:type="xsd:unsignedInt">0</second>
      <ms xsi:type="xsd:unsignedInt">0</ms>
    </clearTime>
  </ns1:clearAlarmRuleStatusRecords>
</SOAP-ENV:Body>
```

Example XML Response:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:clearAlarmRuleStatusRecordsResponse>
    <result xsi:type="xsd:unsignedInt">0</result>
  </ns1:clearAlarmRuleStatusRecordsResponse>
</SOAP-ENV:Body>
```

Example call: php

SOAP operation name: reloadNetflowAlarmMonitor

Description:

Reloads new rule engine configurations. Make sure you call this function when you are done changing application, hosts or router profiles or change their associations with rules. Loading or reloading of rule engine happens only during startup and later only when triggered by this call.

Request Parameters:

None

Response Parameters:

Name	Description	Notes
result	Result code of operation: 0-success, anything else on error.	

Examples:

Example XML Request:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:reloadNetflowAlarmMonitor>
</ns1:reloadNetflowAlarmMonitor>
</SOAP-ENV:Body>
```

Example XML Response:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:reloadNetflowAlarmMonitorResponse>
    <result xsi:type="xsd:unsignedInt">0</result>
  </ns1:reloadNetflowAlarmMonitorResponse>
</SOAP-ENV:Body>
```

Example call: php

```
global $soap_client;
```

```
$result = $soap_client->reloadNetflowAlarmMonitor();
```

SOAP operation name: getVitalSignsStats

Description:

Gets sequence of records for vital statistic of the system on which FlowMatrix engine is running. Usually result includes up to 60 records for last 60 minutes. Please note, these records are stored in memory of FlowMatrix service process and not stored in data base. So you can get less then 60 records after FlowMatrix restarts even so FlowMatrix was running for long time before restart.

Request Parameters:

None

Response Parameters:

Result contains vital sign statistics, for names of the statistics field please check XML response example below. Please keep in mind all the averages are per minute.

Examples:

Example XML Request:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:getVitalSignsStats>
</ns1:getVitalSignsStats>
</SOAP-ENV:Body>
```

Example XML Response:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:SOAPAPPVITALSTATSRESULT>
    <returnCode xsi:type="xsd:unsignedInt">0</returnCode>
    <result xsi:type="ns1:SOAPAPPVITALSTATSRECORDS">
      <statsSeq xsi:type="xsd:unsignedInt">0</statsSeq>
      <records xsi:type="ns1:SOAPAPPVITALSTATS">
        <flowQueueStats xsi:type="ns1:SOAPQUEUESTATS">
          <avgIncRate xsi:type="xsd:unsignedInt">0</avgIncRate>
          <avgOutRate xsi:type="xsd:unsignedInt">0</avgOutRate>
          <avgQueueLen xsi:type="xsd:unsignedInt">0</avgQueueLen>
        </flowQueueStats>
        <eventQueueStats xsi:type="ns1:SOAPQUEUESTATS">
          <avgIncRate xsi:type="xsd:unsignedInt">0</avgIncRate>
          <avgOutRate xsi:type="xsd:unsignedInt">0</avgOutRate>
          <avgQueueLen xsi:type="xsd:unsignedInt">0</avgQueueLen>
        </eventQueueStats>
        <rulesStats xsi:type="ns1:SOAPRULESSTATS">
          <totalRuleExecuted xsi:type="xsd:unsignedInt">0</totalRuleExecuted>
          <totalExecutionTime xsi:type="xsd:unsignedInt">0</totalExecutionTime>
          <totalRulesFired xsi:type="xsd:unsignedInt">0</totalRulesFired>
          <totalRecordGenerated xsi:type="xsd:unsignedInt">0</totalRecordGenerated>
          <totalEventsGenerated xsi:type="xsd:unsignedInt">0</totalEventsGenerated>
        </rulesStats>
        <systemStats xsi:type="ns1:SOAPSYSYSTEMSTATS">
          <CPUUtilization xsi:type="xsd:unsignedInt">0</CPUUtilization>
          <MemoryUsed xsi:type="xsd:unsignedInt">0</MemoryUsed>
          <MemoryFree xsi:type="xsd:unsignedInt">0</MemoryFree>
          <MemoryUtilization xsi:type="xsd:unsignedInt">0</MemoryUtilization>
        </systemStats>
        <anomalyNetStats xsi:type="ns1:SoapNetFlowAnomlyEngStats-t">
          <totalAlarms xsi:type="xsd:unsignedInt">0</totalAlarms>
          <totalWarnings xsi:type="xsd:unsignedInt">0</totalWarnings>
          <totalFeatureSub1Alarms xsi:type="xsd:unsignedInt">0</totalFeatureSub1Alarms>
          <totalFeatureSub2Alarms xsi:type="xsd:unsignedInt">0</totalFeatureSub2Alarms>
        </anomalyNetStats>
      </records>
    </result>
  </ns1:SOAPAPPVITALSTATSRESULT>
</SOAP-ENV:Body>
```


</SOAP-ENV:Body>

Example call: php

```
$stats = $soap_client->getVitalSignsStats();
```

```
$soapobj = is_array($stats['result']->records) ? $stats['result']->records : array($stats['result']->records);
```

```
$CPUUtilization = array();
```

```
$MemoryUsed = array();
```

```
$MemoryFree = array();
```

```
$MemoryUtilization = array();
```

```
// rules stats
```

```
$totalRuleExecuted = array();
```

```
$totalExecutionTime = array();
```

```
$totalRulesFired = array();
```

```
$totalRecordGenerated = array();
```

```
$totalEventsGenerated = array();
```

```
// event queue stats
```

```
$EventavgIncRate = array();
```

```
$EventavgOutRate = array();
```

```
$EventavgQueueLen = array();
```

```
// flow queue stats
```

```
$FlowavgIncRate = array();
```

```
$FlowavgOutRate = array();
```

```
$FlowavgQueueLen = array();
```

```
foreach ($soapobj as $record)
```

```
{
```

```
    // system stats
```

```
    $CPUUtilization[] = $record->systemStats->CPUUtilization;
```

```
    $MemoryUsed[] = $record->systemStats->MemoryUsed;
```

```
    $MemoryFree[] = $record->systemStats->MemoryFree / 1000;
```

```
    $MemoryUtilization[] = $record->systemStats->MemoryUtilization;
```

```
    // rules stats
```

```
    $totalRuleExecuted[] = $record->rulesStats->totalRuleExecuted;
```

```
    $totalExecutionTime[] = $record->rulesStats->totalExecutionTime / 1000;
```

```
    $totalRulesFired[] = $record->rulesStats->totalRulesFired;
```

```
    $totalRecordGenerated[] = $record->rulesStats->totalRecordGenerated;
```

```
    $totalEventsGenerated[] = $record->rulesStats->totalEventsGenerated;
```

```
    // event queue stats
```

```
    $EventavgIncRate[] = $record->eventQueueStats->avgIncRate;
```

```
    $EventavgOutRate[] = $record->eventQueueStats->avgOutRate;
```

```
    $EventavgQueueLen[] = $record->eventQueueStats->avgQueueLen;
```

```
    // flow queue stats
```

```

$FlowavgIncRate[] = $record->flowQueueStats->avgIncRate;
$FlowavgOutRate[] = $record->flowQueueStats->avgOutRate;
$FlowavgQueueLen[] = $record->flowQueueStats->avgQueueLen;
// anomaly stats
$NetAlarms[] = $record->anomalyNetStats->totalAlarms;
$App1Alarms[] = $record->anomalyApp1Stats->totalAlarms;
$App2Alarms[] = $record->anomalyApp2Stats->totalAlarms;
$App3Alarms[] = $record->anomalyApp3Stats->totalAlarms;
}

```

SOAP operation name: getStats1RulesStats

Description:

This call returns basic NetFlow statistics for each view type. View types are: whole network, applications group 1, applications group 2 and applications group 3.

Request Parameters:

Name	Description	Notes
beginTime	Time starting from which to include NetFlow stats records in response.	
endTime	Time after which not to include NetFlow stats records in response.	
offset	Controls offset SQL select statement. So you can ask for 10 records from offset 3	
maxRecords	Maximum number of records to include in result	

Response Parameters:

Only fields for Network view are explained below, counters for Application Group 1, Group2 and Group 3 are similar to network view counters with exceptions that they only apply to traffic belonging to corresponding application group.

Name	Description	Notes
avgUcountDstIp	Total count of unique destination IP addresses observed per 5 minutes.	
avgUcountSrcIp	Total count of unique source IP addresses observed per 5 minutes.	
avgUcountDstPort	Total count of unique destination ports observed per 5 minutes.	
avgUcountSrcPort	Total count of unique source ports observed per 5 minutes.	
avgTotalKBytes	Total Kbytes observed per last 5 minutes	
avgTotalPackets	Total packets observed per last 5 minutes	

stdUcountDstIp	Number of standard deviations of unique destination IP addresses per this interval as compared to average value.	
stdUcountSrcIp	Number of standard deviations of unique source IP addresses per this interval as compared to average value.	
stdUcountDstPort	Number of standard deviations of unique destination ports per this interval as compared to average value.	
stdUcountSrcPort	Number of standard deviations of unique source ports per this interval as compared to average value.	
stdTotalKBytes	Number of standard deviations of total Kbytes per this interval per this interval as compared to average value.	
stdTotalPackets	Number of standard deviations of total packets per this interval per this interval as compared to average value.	

Examples:

Example XML Request:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:getStats1RulesStats>
    <cond xsi:type="ns1:SoapQueryRecordsConditions-t">
      <beginTime xsi:type="ns1:SOAPNETFLOWTIME">
        <year xsi:type="xsd:unsignedInt">0</year>
        <month xsi:type="xsd:unsignedInt">0</month>
        <day xsi:type="xsd:unsignedInt">0</day>
        <hour xsi:type="xsd:unsignedInt">0</hour>
        <minute xsi:type="xsd:unsignedInt">0</minute>
        <second xsi:type="xsd:unsignedInt">0</second>
        <ms xsi:type="xsd:unsignedInt">0</ms>
      </beginTime>
      <endTime xsi:type="ns1:SOAPNETFLOWTIME">
        <year xsi:type="xsd:unsignedInt">0</year>
        <month xsi:type="xsd:unsignedInt">0</month>
        <day xsi:type="xsd:unsignedInt">0</day>
        <hour xsi:type="xsd:unsignedInt">0</hour>
        <minute xsi:type="xsd:unsignedInt">0</minute>
        <second xsi:type="xsd:unsignedInt">0</second>
        <ms xsi:type="xsd:unsignedInt">0</ms>
      </endTime>
      <offset xsi:type="xsd:int">0</offset>
      <maxRecords xsi:type="xsd:int">0</maxRecords>
    </cond>
  </ns1:getStats1RulesStats>
</SOAP-ENV:Body>
```

```
<view xsi:type="ns1:NetFlowViewSource-t">ViewNet</view>
</cond>
</ns1:getStats1RulesStats>
</SOAP-ENV:Body>
```

Example XML Response:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<ns1:SOAPHOURLYSTATS1RECORDSRESULT>
<returnCode xsi:type="xsd:unsignedInt">0</returnCode>
<result xsi:type="ns1:SOAPHOURLYSTATS1RECORDS">
<maxRecords xsi:type="xsd:unsignedInt">0</maxRecords>
<totalRecords xsi:type="xsd:unsignedInt">0</totalRecords>
<records xsi:type="ns1:SOAPHOURLYSTATS1RECORD">
<time xsi:type="ns1:SOAPNETFLOWTIME">
<year xsi:type="xsd:unsignedInt">0</year>
<month xsi:type="xsd:unsignedInt">0</month>
<day xsi:type="xsd:unsignedInt">0</day>
<hour xsi:type="xsd:unsignedInt">0</hour>
<minute xsi:type="xsd:unsignedInt">0</minute>
<second xsi:type="xsd:unsignedInt">0</second>
<ms xsi:type="xsd:unsignedInt">0</ms>
</time>
<day xsi:type="xsd:unsignedInt">0</day>
<hour xsi:type="xsd:unsignedInt">0</hour>
<avgUcountDstIp xsi:type="xsd:unsignedInt">0</avgUcountDstIp>
<avgUcountSrcIp xsi:type="xsd:unsignedInt">0</avgUcountSrcIp>
<avgUcountDstPort xsi:type="xsd:unsignedInt">0</avgUcountDstPort>
<avgUcountSrcPort xsi:type="xsd:unsignedInt">0</avgUcountSrcPort>
<avgTotalKBytes xsi:type="xsd:unsignedInt">0</avgTotalKBytes>
<avgTotalPackets xsi:type="xsd:unsignedInt">0</avgTotalPackets>
<stdUcountDstIp xsi:type="xsd:unsignedInt">0</stdUcountDstIp>
<stdUcountSrcIp xsi:type="xsd:unsignedInt">0</stdUcountSrcIp>
<stdUcountDstPort xsi:type="xsd:unsignedInt">0</stdUcountDstPort>
<stdUcountSrcPort xsi:type="xsd:unsignedInt">0</stdUcountSrcPort>
<stdTotalKBytes xsi:type="xsd:unsignedInt">0</stdTotalKBytes>
<stdTotalPackets xsi:type="xsd:unsignedInt">0</stdTotalPackets>
<avgApp1UcountDstIp xsi:type="xsd:unsignedInt">0</avgApp1UcountDstIp>
<avgApp1UcountSrcIp xsi:type="xsd:unsignedInt">0</avgApp1UcountSrcIp>
<avgApp1UcountDstPort xsi:type="xsd:unsignedInt">0</avgApp1UcountDstPort>
<avgApp1UcountSrcPort xsi:type="xsd:unsignedInt">0</avgApp1UcountSrcPort>
<avgApp1TotalKBytes xsi:type="xsd:unsignedInt">0</avgApp1TotalKBytes>
<avgApp1TotalPackets xsi:type="xsd:unsignedInt">0</avgApp1TotalPackets>
<stdApp1UcountDstIp xsi:type="xsd:unsignedInt">0</stdApp1UcountDstIp>
<stdApp1UcountSrcIp xsi:type="xsd:unsignedInt">0</stdApp1UcountSrcIp>
<stdApp1UcountDstPort xsi:type="xsd:unsignedInt">0</stdApp1UcountDstPort>
```

```

<stdApp1UcountSrcPort xsi:type="xsd:unsignedInt">0</stdApp1UcountSrcPort>
<stdApp1TotalKBytes xsi:type="xsd:unsignedInt">0</stdApp1TotalKBytes>
<stdApp1TotalPackets xsi:type="xsd:unsignedInt">0</stdApp1TotalPackets>
<avgApp2UcountDstIp xsi:type="xsd:unsignedInt">0</avgApp2UcountDstIp>
<avgApp2UcountSrcIp xsi:type="xsd:unsignedInt">0</avgApp2UcountSrcIp>
<avgApp2UcountDstPort xsi:type="xsd:unsignedInt">0</avgApp2UcountDstPort>
<avgApp2UcountSrcPort xsi:type="xsd:unsignedInt">0</avgApp2UcountSrcPort>
<avgApp2TotalKBytes xsi:type="xsd:unsignedInt">0</avgApp2TotalKBytes>
<avgApp2TotalPackets xsi:type="xsd:unsignedInt">0</avgApp2TotalPackets>
<stdApp2UcountDstIp xsi:type="xsd:unsignedInt">0</stdApp2UcountDstIp>
<stdApp2UcountSrcIp xsi:type="xsd:unsignedInt">0</stdApp2UcountSrcIp>
<stdApp2UcountDstPort xsi:type="xsd:unsignedInt">0</stdApp2UcountDstPort>
<stdApp2UcountSrcPort xsi:type="xsd:unsignedInt">0</stdApp2UcountSrcPort>
<stdApp2TotalKBytes xsi:type="xsd:unsignedInt">0</stdApp2TotalKBytes>
<stdApp2TotalPackets xsi:type="xsd:unsignedInt">0</stdApp2TotalPackets>
<avgApp3UcountDstIp xsi:type="xsd:unsignedInt">0</avgApp3UcountDstIp>
<avgApp3UcountSrcIp xsi:type="xsd:unsignedInt">0</avgApp3UcountSrcIp>
<avgApp3UcountDstPort xsi:type="xsd:unsignedInt">0</avgApp3UcountDstPort>
<avgApp3UcountSrcPort xsi:type="xsd:unsignedInt">0</avgApp3UcountSrcPort>
<avgApp3TotalKBytes xsi:type="xsd:unsignedInt">0</avgApp3TotalKBytes>
<avgApp3TotalPackets xsi:type="xsd:unsignedInt">0</avgApp3TotalPackets>
<stdApp3UcountDstIp xsi:type="xsd:unsignedInt">0</stdApp3UcountDstIp>
<stdApp3UcountSrcIp xsi:type="xsd:unsignedInt">0</stdApp3UcountSrcIp>
<stdApp3UcountDstPort xsi:type="xsd:unsignedInt">0</stdApp3UcountDstPort>
<stdApp3UcountSrcPort xsi:type="xsd:unsignedInt">0</stdApp3UcountSrcPort>
<stdApp3TotalKBytes xsi:type="xsd:unsignedInt">0</stdApp3TotalKBytes>
<stdApp3TotalPackets xsi:type="xsd:unsignedInt">0</stdApp3TotalPackets>
</records>
</result>
</ns1:SOAPHOURLYSTATS1RECORDSRESULT>
</SOAP-ENV:Body>

```

Example call: php

```

$query = new SoapQueryRecordsConditions_t($viewsource[$view]);

if (isset($_GET['scale']))
{
    $endts = time();
    $startts = $endts - ($_GET['scale'] * 3600);
}
elseif (isset($_GET['st']) && isset($_GET['et']))
{
    $endts = $_GET['et'];
    $startts = $_GET['st'];
}

```

```

else
{
    $endts = time();
    $startts = $endts - (12 * 3600);
}

$span = ($endts - $startts) / 3600; // span in hours

// set the maximum for the observation window to 5 days
if ($span > 120)
{
    $startts = $endts - 432000;
}

// determine how to format time on the label depending on the scale
if ($span < 12) // observation window < 12 hours
{
    define("TIME_FORMAT", 'H:i'); // 10:20
}
else
{
    define("TIME_FORMAT", 'H:i\-\D'); // 10:20-Mon
}

$beginTime = gmdate(DATE_ATOM, $startts);
$endTime = gmdate(DATE_ATOM, $endts);
$timezone = date_default_timezone_get(); // save the current time zone
date_default_timezone_set("GMT");
$beginTime = getdate(strtotime($beginTime));
$endTime = getdate(strtotime($endTime));
date_default_timezone_set($timezone); // restore timezone
@list($query->beginTime->second, $query->beginTime->minute,
$query->beginTime->hour, $query->beginTime->day,,
$query->beginTime->month, $query->beginTime->year) = array_values($beginTime);
@list($query->endTime->second, $query->endTime->minute,
$query->endTime->hour, $query->endTime->day,,
$query->endTime->month, $query->endTime->year) = array_values($endTime);

error_reporting(0);

// get anomaly stats
$stats = $soap_client->getStats1RulesStats($query);

if ($stats['result']->totalRecords > 0)
{
    // do work with state record

```

}

SOAP operation name: **getAnamoly1AlarmRecords**

Note: please see typo in method name, make sure when you make call you use same mistyped as well..

Description:

Gets sequence anomaly alarms generated by anomaly engine. As part of the call caller provides query conditions. Only anomaly alarms records are included and not warning records.

Request Parameters:

Name	Description	Notes
beginTime	Include alarms with alarm time grater then this time	
endTime	Include alarms with alarm time less then this time.	
offset	Controls offset in SQL select statement. For example you query returns 100 records, you can specify offset 10. This helps to implement pagination in web based user interface	
maxRecords	Maximum number of records to return in this call.	
view	What type anomaly records to return. Valid enumeration types: ViewNet, ViewApp1, ViewApp2, ViewApp3	

Response Parameters:

There are a large number of fields returned as part of anomaly record, but from our opinion, you will most likely be interested in 2 fields: summary and details. Other fields are more or less self-explanatory but please keep in mind that FlowMatrix is constantly evolving program and some of the fields which are not documented here may change.

Name	Description	Notes
summary	Summary text for anomaly events. Same text you see in Das board with summary texts show for anomaly events	
details	Details of anomaly text, which includes detailed information about anomaly and if possible classification of detected anomaly	

Examples:

Example XML Request:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:getAnamoly1AlarmRecords>
    <cond xsi:type="ns1:SoapQueryRecordsConditions-t">
      <beginTime xsi:type="ns1:SOAPNETFLOWTIME">
        <year xsi:type="xsd:unsignedInt">0</year>
        <month xsi:type="xsd:unsignedInt">0</month>
        <day xsi:type="xsd:unsignedInt">0</day>
        <hour xsi:type="xsd:unsignedInt">0</hour>
        <minute xsi:type="xsd:unsignedInt">0</minute>
        <second xsi:type="xsd:unsignedInt">0</second>
        <ms xsi:type="xsd:unsignedInt">0</ms>
      </beginTime>
      <endTime xsi:type="ns1:SOAPNETFLOWTIME">
        <year xsi:type="xsd:unsignedInt">0</year>
        <month xsi:type="xsd:unsignedInt">0</month>
        <day xsi:type="xsd:unsignedInt">0</day>
        <hour xsi:type="xsd:unsignedInt">0</hour>
        <minute xsi:type="xsd:unsignedInt">0</minute>
        <second xsi:type="xsd:unsignedInt">0</second>
        <ms xsi:type="xsd:unsignedInt">0</ms>
      </endTime>
      <offset xsi:type="xsd:int">0</offset>
      <maxRecords xsi:type="xsd:int">0</maxRecords>
      <view xsi:type="ns1:NetFlowViewSource-t">ViewNet</view>
    </cond>
  </ns1:getAnamoly1AlarmRecords>
</SOAP-ENV:Body>
```

Example XML Response:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:SOAPxANAMOLYxRECORDSxRESULT>
    <returnCode xsi:type="xsd:unsignedInt">0</returnCode>
    <result xsi:type="ns1:SOAPxANAMOLYxRECORDS">
      <totalRecords xsi:type="xsd:unsignedInt">0</totalRecords>
      <records xsi:type="ns1:SOAPxNETFLOWxANAMOLYxRECORD">
        <time xsi:type="ns1:SOAPNETFLOWTIME">
          <year xsi:type="xsd:unsignedInt">0</year>
          <month xsi:type="xsd:unsignedInt">0</month>
          <day xsi:type="xsd:unsignedInt">0</day>
          <hour xsi:type="xsd:unsignedInt">0</hour>
          <minute xsi:type="xsd:unsignedInt">0</minute>
          <second xsi:type="xsd:unsignedInt">0</second>
          <ms xsi:type="xsd:unsignedInt">0</ms>
        </time>
      </records>
    </result>
  </ns1:SOAPxANAMOLYxRECORDSxRESULT>
</SOAP-ENV:Body>
```

```

</time>
<recordID xsi:type="ns1:SOAPEVENTSEQUENCE">
  <lowPart xsi:type="xsd:unsignedInt">0</lowPart>
  <highPart xsi:type="xsd:unsignedInt">0</highPart>
</recordID>
<type xsi:type="xsd:unsignedInt">0</type>
<featureQAlarmThreashold xsi:type="xsd:double">0.0</featureQAlarmThreashold>
<featureQWarningThreashold xsi:type="xsd:double">0.0</featureQWarningThreashold>
<featureQVValue xsi:type="xsd:double">0.0</featureQVValue>
<featureT2AlarmThreashold xsi:type="xsd:double">0.0</featureT2AlarmThreashold>
<featureT2WarningThreashold xsi:type="xsd:double">0.0</featureT2WarningThreashold>
<featureTVValue xsi:type="xsd:double">0.0</featureTVValue>
<volumeQAlarmThreashold xsi:type="xsd:double">0.0</volumeQAlarmThreashold>
<volumeQWarningThreashold xsi:type="xsd:double">0.0</volumeQWarningThreashold>
<volumeQVValue xsi:type="xsd:double">0.0</volumeQVValue>
<volumeT2AlarmThreashold xsi:type="xsd:double">0.0</volumeT2AlarmThreashold>
<volumeT2WarningThreashold xsi:type="xsd:double">0.0</volumeT2WarningThreashold>
<volumeTVValue xsi:type="xsd:double">0.0</volumeTVValue>
<featureNormalComp xsi:type="xsd:double">0.0</featureNormalComp>
<featureResidualComp xsi:type="xsd:double">0.0</featureResidualComp>
<volumeNormalComp xsi:type="xsd:double">0.0</volumeNormalComp>
<volumeResidualComp xsi:type="xsd:double">0.0</volumeResidualComp>
<summary xsi:type="xsd:string"></summary>
<details xsi:type="xsd:string"></details>
</records>
</result>
</ns1:SOAPxANAMOLYxRECORDSxRESULT>
</SOAP-ENV:Body>

```

Example call: php

```

$query = new SoapQueryRecordsConditions_t($viewsource[$view]);

$time = time();
$time = $time - (WINDOW * 3600);

$timezone = date_default_timezone_get(); // save the current time zone
date_default_timezone_set("GMT");
$time = getdate($time);
@list($query->beginTime->second, $query->beginTime->minute,
$query->beginTime->hour, $query->beginTime->day,,
$query->beginTime->month, $query->beginTime->year) = array_values($time);
date_default_timezone_set($timezone); // restore timezone

$timezone = date_default_timezone_get(); // save the current time zone
date_default_timezone_set("GMT");

```

```
$utime = getdate($etime);
@list($query->endTime->second, $query->endTime->minute,
$query->endTime->hour, $query->endTime->day,,
$query->endTime->month, $query->endTime->year) = array_values($utime);
date_default_timezone_set($timezone); // restore timezone
```

```
$records = $soap_client->getAnamoly1AlarmRecords($query);
```

SOAP operation name: getAnamoly1AnyRecords

Description:

Same as **getAnamoly1AlarmRecords()** documented above. The only difference is that this call includes alarms and warning anomaly records.

Request Parameters:

Please see help for **getAnamoly1AlarmRecords()**

Response Parameters:

Please see help for **getAnamoly1AlarmRecords()**

Examples:

Example XML Request:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<ns1:getAnamoly1AnyRecords>
<cond xsi:type="ns1:SoapQueryRecordsConditions-t">
<beginTime xsi:type="ns1:SOAPNETFLOWTIME">
<year xsi:type="xsd:unsignedInt">0</year>
<month xsi:type="xsd:unsignedInt">0</month>
<day xsi:type="xsd:unsignedInt">0</day>
<hour xsi:type="xsd:unsignedInt">0</hour>
<minute xsi:type="xsd:unsignedInt">0</minute>
<second xsi:type="xsd:unsignedInt">0</second>
<ms xsi:type="xsd:unsignedInt">0</ms>
</beginTime>
<endTime xsi:type="ns1:SOAPNETFLOWTIME">
<year xsi:type="xsd:unsignedInt">0</year>
<month xsi:type="xsd:unsignedInt">0</month>
<day xsi:type="xsd:unsignedInt">0</day>
<hour xsi:type="xsd:unsignedInt">0</hour>
<minute xsi:type="xsd:unsignedInt">0</minute>
<second xsi:type="xsd:unsignedInt">0</second>
```

```

    <ms xsi:type="xsd:unsignedInt">0</ms>
  </endTime>
  <offset xsi:type="xsd:int">0</offset>
  <maxRecords xsi:type="xsd:int">0</maxRecords>
  <view xsi:type="ns1:NetFlowViewSource-t">ViewNet</view>
</cond>
</ns1:getAnamoly1AnyRecords>
</SOAP-ENV:Body>

```

Example XML Response:

```

<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:SOAPxANAMOLYxRECORDSxRESULT>
    <returnCode xsi:type="xsd:unsignedInt">0</returnCode>
    <result xsi:type="ns1:SOAPxANAMOLYxRECORDS">
      <totalRecords xsi:type="xsd:unsignedInt">0</totalRecords>
      <records xsi:type="ns1:SOAPxNETFLOWxANAMOLYxRECORD">
        <time xsi:type="ns1:SOAPNETFLOWTIME">
          <year xsi:type="xsd:unsignedInt">0</year>
          <month xsi:type="xsd:unsignedInt">0</month>
          <day xsi:type="xsd:unsignedInt">0</day>
          <hour xsi:type="xsd:unsignedInt">0</hour>
          <minute xsi:type="xsd:unsignedInt">0</minute>
          <second xsi:type="xsd:unsignedInt">0</second>
          <ms xsi:type="xsd:unsignedInt">0</ms>
        </time>
        <recordID xsi:type="ns1:SOAPEVENTSEQUENCE">
          <lowPart xsi:type="xsd:unsignedInt">0</lowPart>
          <highPart xsi:type="xsd:unsignedInt">0</highPart>
        </recordID>
        <type xsi:type="xsd:unsignedInt">0</type>
        <featureQAlarmThreashold xsi:type="xsd:double">0.0</featureQAlarmThreashold>
        <featureQWarningThreashold xsi:type="xsd:double">0.0</featureQWarningThreashold>
        <featureQValue xsi:type="xsd:double">0.0</featureQValue>
        <featureT2AlarmThreashold xsi:type="xsd:double">0.0</featureT2AlarmThreashold>
        <featureT2WarningThreashold xsi:type="xsd:double">0.0</featureT2WarningThreashold>
        <featureTValue xsi:type="xsd:double">0.0</featureTValue>
        <volumeQAlarmThreashold xsi:type="xsd:double">0.0</volumeQAlarmThreashold>
        <volumeQWarningThreashold xsi:type="xsd:double">0.0</volumeQWarningThreashold>
        <volumeQValue xsi:type="xsd:double">0.0</volumeQValue>
        <volumeT2AlarmThreashold xsi:type="xsd:double">0.0</volumeT2AlarmThreashold>
        <volumeT2WarningThreashold xsi:type="xsd:double">0.0</volumeT2WarningThreashold>
        <volumeTValue xsi:type="xsd:double">0.0</volumeTValue>
        <featureNormalComp xsi:type="xsd:double">0.0</featureNormalComp>
        <featureResidualComp xsi:type="xsd:double">0.0</featureResidualComp>
        <volumeNormalComp xsi:type="xsd:double">0.0</volumeNormalComp>
      </records>
    </result>
  </ns1:SOAPxANAMOLYxRECORDSxRESULT>
</SOAP-ENV:Body>

```

```

<volumeResidualComp xsi:type="xsd:double">0.0</volumeResidualComp>
<summary xsi:type="xsd:string"></summary>
<details xsi:type="xsd:string"></details>
</records>
</result>
</ns1:SOAPxANAMOLYxRECORDSxRESULT>
</SOAP-ENV:Body>

```

Example call: php

SOAP operation name: getAnamoly1Record

Description:

This call is similar to **getAnamoly1AlarmRecords()** with only exception that with this call caller will get single anomaly record as opposed to sequence of records returned from **getAnamoly1AlarmRecords()**

Request Parameters:

Name	Description	Notes
recorded	Anomaly record id.	

Response Parameters:

Please check help for **getAnamoly1AlarmRecords()** because result type is same in both calls.

Examples:

Example XML Request:

```

<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<ns1:getAnamoly1Record>
<recordID xsi:type="ns1:SOAPEVENTSEQUENCE">
<lowPart xsi:type="xsd:unsignedInt">0</lowPart>
<highPart xsi:type="xsd:unsignedInt">0</highPart>
</recordID>
</ns1:getAnamoly1Record>
</SOAP-ENV:Body>

```

Example XML Response:

```

<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<ns1:SOAPxANAMOLYxRECORDSxRESULT>
<returnCode xsi:type="xsd:unsignedInt">0</returnCode>
<result xsi:type="ns1:SOAPxANAMOLYxRECORDS">
<totalRecords xsi:type="xsd:unsignedInt">0</totalRecords>

```

```

<records xsi:type="ns1:SOAPxNETFLOWxANAMOLYxRECORD">
  <time xsi:type="ns1:SOAPNETFLOWTIME">
    <year xsi:type="xsd:unsignedInt">0</year>
    <month xsi:type="xsd:unsignedInt">0</month>
    <day xsi:type="xsd:unsignedInt">0</day>
    <hour xsi:type="xsd:unsignedInt">0</hour>
    <minute xsi:type="xsd:unsignedInt">0</minute>
    <second xsi:type="xsd:unsignedInt">0</second>
    <ms xsi:type="xsd:unsignedInt">0</ms>
  </time>
  <recordID xsi:type="ns1:SOAPEVENTSEQUENCE">
    <lowPart xsi:type="xsd:unsignedInt">0</lowPart>
    <highPart xsi:type="xsd:unsignedInt">0</highPart>
  </recordID>
  <type xsi:type="xsd:unsignedInt">0</type>
  <featureQAlarmThreshold xsi:type="xsd:double">0.0</featureQAlarmThreshold>
  <featureQWarningThreshold xsi:type="xsd:double">0.0</featureQWarningThreshold>
  <featureQValue xsi:type="xsd:double">0.0</featureQValue>
  <featureT2AlarmThreshold xsi:type="xsd:double">0.0</featureT2AlarmThreshold>
  <featureT2WarningThreshold xsi:type="xsd:double">0.0</featureT2WarningThreshold>
  <featureTValue xsi:type="xsd:double">0.0</featureTValue>
  <volumeQAlarmThreshold xsi:type="xsd:double">0.0</volumeQAlarmThreshold>
  <volumeQWarningThreshold xsi:type="xsd:double">0.0</volumeQWarningThreshold>
  <volumeQValue xsi:type="xsd:double">0.0</volumeQValue>
  <volumeT2AlarmThreshold xsi:type="xsd:double">0.0</volumeT2AlarmThreshold>
  <volumeT2WarningThreshold xsi:type="xsd:double">0.0</volumeT2WarningThreshold>
  <volumeTValue xsi:type="xsd:double">0.0</volumeTValue>
  <featureNormalComp xsi:type="xsd:double">0.0</featureNormalComp>
  <featureResidualComp xsi:type="xsd:double">0.0</featureResidualComp>
  <volumeNormalComp xsi:type="xsd:double">0.0</volumeNormalComp>
  <volumeResidualComp xsi:type="xsd:double">0.0</volumeResidualComp>
  <summary xsi:type="xsd:string"></summary>
  <details xsi:type="xsd:string"></details>
</records>
</result>
</ns1:SOAPxANAMOLYxRECORDSxRESULT>
</SOAP-ENV:Body>

```

Example call: php

```
$query = new SOAPEVENTSEQUENCE($lowPart, $highPart);
```

```
$record = $soap_client->getAnamoly1Record($query);
```

```
if ($record['returnCode'] != 0)
```

```
    trigger_error("Error code[".$record['returnCode']."] returned by getAnamoly1Record", E_ERROR);
```

SOAP operation name: getAnomalyRecordDetails

Description:

Returns details for anomaly record as specified by record ID. Details include detail information for every network cluster (total of 28 clusters). Each cluster has summary and detail textual information plus 9 tables of statistical information: top peers records, top Src IP addresses by packets count, top Dst IP addresses by packet count, top Src IP addresses by count of unique IP's that communicated with it, top Dst IP addresses by count of unique addresses it communicated with, top Src ports by packets count, top Dst ports by packet count, top Src ports by count of unique IP's that communicated with it, top Dst ports addresses by count of unique addresses it communicated with.

Request Parameters:

Name	Description	Notes
Recorded	Anomaly event record id to return details for.	

Response Parameters:

Returns sequence of clusters details. Below is description of some key data members of FlowBucketDetailInfo data type. This data type contains details information about particular network cluster. Response contains sequence of up to 28 such records.

Name	Description	Notes
addr1	Top IP addresses from one direction (and addr2 is from other direction) sorted by number of unique IP addresses it communicated with or was contacted by during last minute.	
addr1byPack	Top IP addresses from one direction (and addr2 is from other direction) sorted by number of packets sent from/to this address during last minute.	
addr2	Top IP addresses from other direction (and addr1 is from other direction) sorted by number of unique IP addresses it communicated with or was contacted by.	
addr2byPack	Top IP addresses from other direction (and addr1 is from other direction) sorted by number of packets sent from/to this address during last minute.	
port1	Top port from one direction (and port2 is from other direction) sorted by number of unique IP addresses it communicated with or was contacted by during last minute.	
port1byPack	Top port from one direction (and port2 is from other direction) sorted by number of packets that it communicated over different connections with one	

	of the ports begin this port during last minute.	
port2	Top port from one direction (and port1 is from other direction) sorted by number of unique IP addresses it communicated with or was contacted by during last minute.	
port2byPack	Top port from one direction (and port2 is from other direction) sorted by number of packets that it communicated over different connections with one of the ports begin this port during last minute.	
Peers	Top peers over last minute sorted by number of packets communicated.	
Classified	Boolean flag indicating if cluster has been classified: 1- yes, it was classified, 0-no it was not.	
classSummary	Classification summary. In case if cluster was not classified it will contains text summary of statistical tables for this cluster.	
classDetails	Classification details. In case if cluster was not classified it will contains text summary of statistical tables for this cluster.	
classDebug	Debug information in textual form.	

Examples:

Example XML Request:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:getAnamoly1RecordDetails>
    <cond xsi:type="ns1:Anamoly1RecordQueryDetailConditions-t">
      <recordID xsi:type="ns1:SOAPEVENTSEQUENCE">
        <lowPart xsi:type="xsd:unsignedInt">0</lowPart>
        <highPart xsi:type="xsd:unsignedInt">0</highPart>
      </recordID>
      <reserved xsi:type="xsd:unsignedInt">0</reserved>
    </cond>
  </ns1:getAnamoly1RecordDetails>
</SOAP-ENV:Body>
```

Example XML Response:

```
<SOAP-ENV:Body SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <ns1:Anamoly1RecordDetailInformationResult-t>
    <returnCode xsi:type="xsd:unsignedInt">0</returnCode>
    <result xsi:type="ns1:Anamoly1RecordDetailInformation-t">
      <reserved xsi:type="xsd:unsignedInt">0</reserved>
      <detailBuckets xsi:type="ns1:FlowBucketDetailInfo">
        <addr1 xsi:type="ns1:FlowIPAddrDetailInfo-t">
```

```
<packets xsi:type="xsd:unsignedInt">0</packets>
<bytes xsi:type="xsd:unsignedInt">0</bytes>
<addr xsi:type="xsd:unsignedInt">0</addr>
<IpAddrCount xsi:type="xsd:unsignedInt">0</IpAddrCount>
<topIpList xsi:type="xsd:unsignedInt">0</topIpList>
<topPortsList xsi:type="xsd:unsignedShort">0</topPortsList>
</addr1>
<addr1byPack xsi:type="ns1:FlowIPAddrDetailInfo-t">
  <packets xsi:type="xsd:unsignedInt">0</packets>
  <bytes xsi:type="xsd:unsignedInt">0</bytes>
  <addr xsi:type="xsd:unsignedInt">0</addr>
  <IpAddrCount xsi:type="xsd:unsignedInt">0</IpAddrCount>
  <topIpList xsi:type="xsd:unsignedInt">0</topIpList>
  <topPortsList xsi:type="xsd:unsignedShort">0</topPortsList>
</addr1byPack>
<addr2 xsi:type="ns1:FlowIPAddrDetailInfo-t">
  <packets xsi:type="xsd:unsignedInt">0</packets>
  <bytes xsi:type="xsd:unsignedInt">0</bytes>
  <addr xsi:type="xsd:unsignedInt">0</addr>
  <IpAddrCount xsi:type="xsd:unsignedInt">0</IpAddrCount>
  <topIpList xsi:type="xsd:unsignedInt">0</topIpList>
  <topPortsList xsi:type="xsd:unsignedShort">0</topPortsList>
</addr2>
<addr2byPack xsi:type="ns1:FlowIPAddrDetailInfo-t">
  <packets xsi:type="xsd:unsignedInt">0</packets>
  <bytes xsi:type="xsd:unsignedInt">0</bytes>
  <addr xsi:type="xsd:unsignedInt">0</addr>
  <IpAddrCount xsi:type="xsd:unsignedInt">0</IpAddrCount>
  <topIpList xsi:type="xsd:unsignedInt">0</topIpList>
  <topPortsList xsi:type="xsd:unsignedShort">0</topPortsList>
</addr2byPack>
<port1 xsi:type="ns1:FlowPortDetailInfo-t">
  <packets xsi:type="xsd:unsignedInt">0</packets>
  <bytes xsi:type="xsd:unsignedInt">0</bytes>
  <port xsi:type="xsd:unsignedInt">0</port>
  <IpAddrCount xsi:type="xsd:unsignedInt">0</IpAddrCount>
  <topIpList xsi:type="xsd:unsignedInt">0</topIpList>
  <topPortsList xsi:type="xsd:unsignedShort">0</topPortsList>
</port1>
<port1byPack xsi:type="ns1:FlowPortDetailInfo-t">
  <packets xsi:type="xsd:unsignedInt">0</packets>
  <bytes xsi:type="xsd:unsignedInt">0</bytes>
  <port xsi:type="xsd:unsignedInt">0</port>
  <IpAddrCount xsi:type="xsd:unsignedInt">0</IpAddrCount>
  <topIpList xsi:type="xsd:unsignedInt">0</topIpList>
  <topPortsList xsi:type="xsd:unsignedShort">0</topPortsList>
```

```

</port1byPack>
<port2 xsi:type="ns1:FlowPortDetailInfo-t">
  <packets xsi:type="xsd:unsignedInt">0</packets>
  <bytes xsi:type="xsd:unsignedInt">0</bytes>
  <port xsi:type="xsd:unsignedInt">0</port>
  <IpAddrCount xsi:type="xsd:unsignedInt">0</IpAddrCount>
  <topIpList xsi:type="xsd:unsignedInt">0</topIpList>
  <topPortsList xsi:type="xsd:unsignedShort">0</topPortsList>
</port2>
<port2byPack xsi:type="ns1:FlowPortDetailInfo-t">
  <packets xsi:type="xsd:unsignedInt">0</packets>
  <bytes xsi:type="xsd:unsignedInt">0</bytes>
  <port xsi:type="xsd:unsignedInt">0</port>
  <IpAddrCount xsi:type="xsd:unsignedInt">0</IpAddrCount>
  <topIpList xsi:type="xsd:unsignedInt">0</topIpList>
  <topPortsList xsi:type="xsd:unsignedShort">0</topPortsList>
</port2byPack>
<peers xsi:type="ns1:FlowIpPeerDetailInfo">
  <packets xsi:type="xsd:unsignedInt">0</packets>
  <bytes xsi:type="xsd:unsignedInt">0</bytes>
  <addr1 xsi:type="xsd:unsignedInt">0</addr1>
  <addr2 xsi:type="xsd:unsignedInt">0</addr2>
</peers>
<status xsi:type="xsd:int">0</status>
<classified xsi:type="xsd:int">0</classified>
<classSummary xsi:type="xsd:string"></classSummary>
<classDetails xsi:type="xsd:string"></classDetails>
<classDebug xsi:type="xsd:string"></classDebug>
</detailBuckets>
</result>
</ns1:Anamoly1RecordDetailInformationResult-t>
</SOAP-ENV:Body>

```

Example call: php

```
$query = new Anamoly1RecordQueryDetailConditions_t($lowPart, $highPart);
```

```
$records = $soap_client->getAnamoly1RecordDetails($query);
```

```
if ($records['returnCode'] != 0)
```

```
    trigger_error("Error code[".$records['returnCode']."] returned by getAnamoly1RecordDetails",
E_ERROR);
```