

Advanced Monitoring with API

A Presentation for MUM Sydney, 2012

By Herry Darmawan

About ME

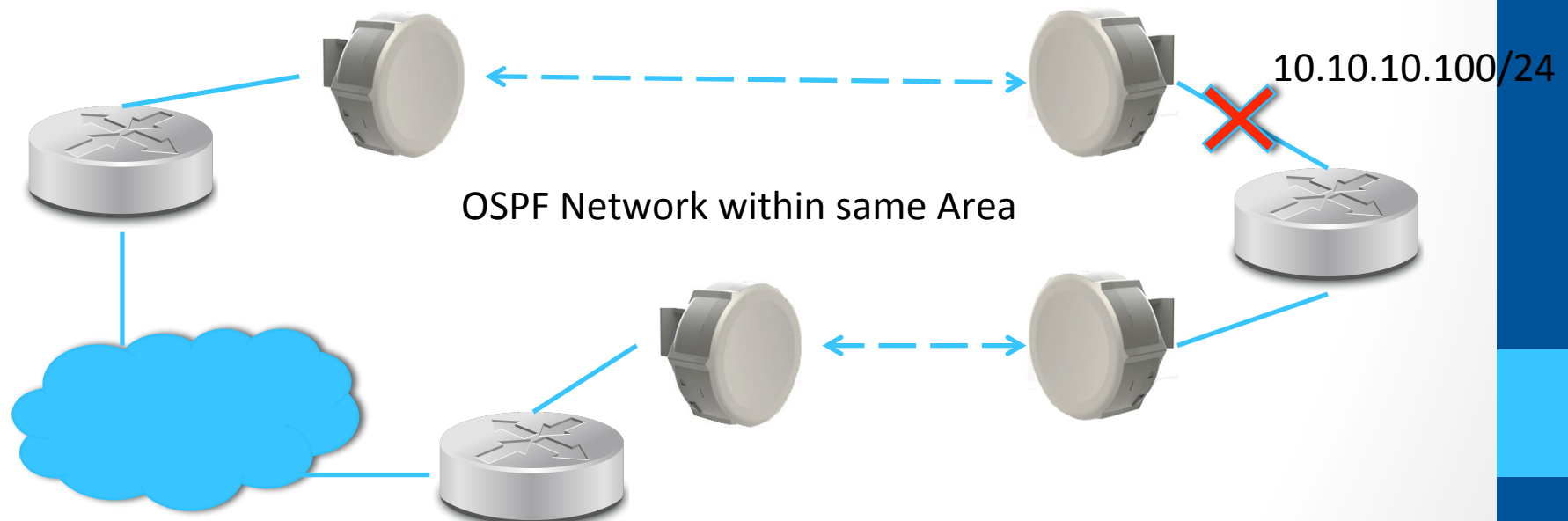
- Herry Darmawan
 - Working for : Spectrum Indonesia
 - Position : Technical and Operational Manager
 - Home base : Surabaya, Indonesia
- Has been using MikroTik since 2004
- Daily Activity
 - Train people how to use MikroTik through MikroTik Certified Training (basic and advance class)
 - Managing technical team of ISP in Surabaya for the last mile connection (Wireless and Fiber)
 - Conducting Networking Project and Consultation
 - Developing Monitoring and System for network and standard procedure particularly using MikroTik as the object

What is This Presentation About?

- Monitoring devices
- What regular method (non-API) cannot achieve?
- How to use API in polling-based method
- Case Study ...!

Regular Monitoring System

- Method
 - ICMP
 - SNMP and SNMP-Trap
 - TCP/UDP checked (based on port)
- How to monitor case like this???



Introducing Nagios

- Web based monitoring system
- Modular
 - Check plugin (in perl or c++)
 - Lots of improvement module (front-end, polling, 3rd party integration, etc)
- Database backend
 - NDOMY
 - MySQL
 - Postgres SQL
- Recommended Front-End : CENTREON

Centreon – host and service

Home

2012/10/20 4:57 ^

:: Hosts

2 Down	0 Unreachable	4 Up	0 Pending
2 Unhandled			

:: Unhandled Host problems (last 100)

Host Name	Status	IP Address	Duration	Last Check	Status Output
MR5	Down	192.168.5.1	21h 18m 31s	2012/10/19 7:41	CRITICAL - Host Unreachable (192.168.5.1)
MR4	Down	192.168.4.1	21h 18m 56s	2012/10/19 7:41	CRITICAL - Host Unreachable (192.168.4.1)

:: Services

5 Critical	0 Warning	4 OK	2 Unknown	0 Pending
2 On Problem Host			2 Unhandled	
3 Unhandled				

:: Unhandled Service problems (last 100)

Host Name	Service Name	Status	IP Address	Duration	Last Check	Status Output
MR2	ping	Critical	192.168.2.1	21h 19m 51s	2012/10/19 7:40	PING CRITICAL - From 192.168.3.2 icmp_seq=3 Destination Host Unreachable
MR3	ping	Critical	192.168.3.1	21h 20m 45s	2012/10/19 7:42	PING CRITICAL - From 192.168.3.2 icmp_seq=3 Destination Host Unreachable
MR1	ping	Critical	192.168.1.1	21h 21m 12s	2012/10/19 7:41	PING CRITICAL - From 192.168.3.2 icmp_seq=3 Destination Host Unreachable
MR3	OSPF-to-MR5	Unknown	192.168.3.1	21h 19m 37s	2012/10/19 7:40	no socket :No route to host
MR3	BGP-to-MR1	Unknown	192.168.3.1	21h 20m 59s	2012/10/19 7:41	no socket :No route to host

Centreon – host details config

Host Configuration

Relations

Data Processing

Host Extended Infos

Modify a Host

General Information

Host Name *

MR1

Alias

MetaROUTER1-BGP

IP Address / DNS

192.168.1.1

SNMP Community & Version

dsnmp

2c

Monitored from

default

Host Templates

A host can have multiple templates, their orders have a significant importance
Here is a self explanatory image.

Add a template +

generic-host

Create Services linked to the Template too

☐ Yes ☒ No

Host Check Properties

Check Period

Check Command

Args

Max Check Attempts

Normal Check Interval

* 30 seconds

Retry Check Interval

* 30 seconds

Active Checks Enabled

☐ Yes ☐ No ☒ Default

Passive Checks Enabled

☐ Yes ☐ No ☒ Default

Centreon – plugins for service

Configuration ▶ Commands ▶ Checks

More actions... Add 1 2 3 4 ➡

<input type="checkbox"/> Name	Command Line
<input type="checkbox"/> check_bgp	\$USER1\$/check_bgp.pl -m \$HOSTADDRESS\$ -u api -p te...
<input type="checkbox"/> check_centreon_cpu	\$USER1\$/check_centreon_snmp_cpu -H \$HOSTADDRESS\$ -...
<input type="checkbox"/> check_centreon_dummy	\$USER1\$/check_centreon_dummy -s \$ARG1\$ -o \$ARG2\$...
<input type="checkbox"/> check_centreon_load_average	\$USER1\$/check_centreon_snmp_loadaverage -H \$HOSTAD...
<input type="checkbox"/> check_centreon_memory	\$USER1\$/check_centreon_snmp_memory -H \$HOSTADDRESS...
<input type="checkbox"/> check_centreon_nb_connections	\$USER1\$/check_centreon_snmp_TcpCon -H \$HOSTADDRESS...
<input type="checkbox"/> check_centreon_ping	\$USER1\$/check_centreon_ping -H \$HOSTADDRESS\$ -n \$A...
check_centreon_process	
<input type="checkbox"/> check_centreon_process	\$USER1\$/check_centreon_snmp_process -H \$HOSTADDRES...
<input type="checkbox"/> check_centreon_process_exists	\$USER1\$/check_centreon_snmp_process -H \$HOSTADDRES...
<input type="checkbox"/> check_centreon_remote_storage	\$USER1\$/check_centreon_snmp_remote_storage -H \$HOS...
check_centreon_snmp_	
<input type="checkbox"/> check_centreon_snmp_proc_detailed	\$USER1\$/check_centreon_snmp_process_detailed -H \$H...
<input type="checkbox"/> check_centreon_snmp_value	\$USER1\$/check_centreon_snmp_value -H \$HOSTADDRESS\$...
check_centreon_traffic	
<input type="checkbox"/> check_centreon_traffic	\$USER1\$/check_centreon_snmp_traffic -H \$HOSTADDRES...
<input type="checkbox"/> check_centreon_traffic_limited	\$USER1\$/check_centreon_snmp_traffic -H \$HOSTADDRES...
<input type="checkbox"/> check_centreon_uptime	\$USER1\$/check_centreon_snmp_uptime -H \$HOSTADDRESS...

Plugin short-name

The real command-prompt syntax
(including the parameters)

Centreon - command

The actual command prompt
with some MACROS

Modify a Command

Check

Command Name *

Command Type ☐ Notification ☒ Check ☐ Misc ☐ Discovery

Command Line *

<< SUSER1\$ (path to the plugins) <>

<< /Centreon/SNMP

<< \$ADMINEMAILS

```
[root@localhost plugins]# ./check_centreon_snmp_uptime -H 192.168.3.1 -C dsnmp -v 2c -d
OK - Uptime (in day): 0|uptime=0day(s)
```

Centreon - attaching to service

Service Configuration

Relations

Data Processing

Service Extended Info

Add a Service

General Information

Description *

UPTIME

Service Template

generic-service

Service State

Is Volatile

☐ Yes ☐ No ☒ Default

Check Period *

Check Command *

check_centreon_uptime

Service Configuration

Relations

Add relations

Relations

Linked with Hosts *

Available

Centreon-Server
MR1
MR2
MR4
MR5






Add

Remove

Selected

MR3

Centreon – service result

<input type="checkbox"/>	 MR3	 BGP-to-MR1	150 sec / 30 sec
<input type="checkbox"/>		 OSPF-to-MR5	150 sec / 30 sec
<input type="checkbox"/>		 ping	150 sec / 30 sec
<input type="checkbox"/>		 UPTIME	150 sec / 30 sec

 Service UPTIME on host MR3 [192.168.3.1 | MetaROUTER3-BORDER]

Status Details

Service Status	OK
Status information	OK - Uptime (in day): 0
Extended status information	
Performance Data	uptime=0day(s)

Nagios Plugin Structure

- Plugins can be created using perl or c++ (compiled or not)
- For un-compiled script, this is the structure
 - Header
 - Parameter initialization
 - Help menu
 - Process
 - Processing and gathering information from device
 - Return Value
 - Result display
 - RRD result
 - Service status return

Nagios Plugin Structure

- Header
 - Taking parameters from the command prompt
 - Check whether the parameters are correct and complete (for example we need to take the username, but user didn't provide us with the username parameter)
 - Print help (if necessary)
 - Global and local variable initialization
- Process
 - Is the real process
 - All process (SNMP, Telnet, SSH, API) is happening in this part
 - Beware to check the structure

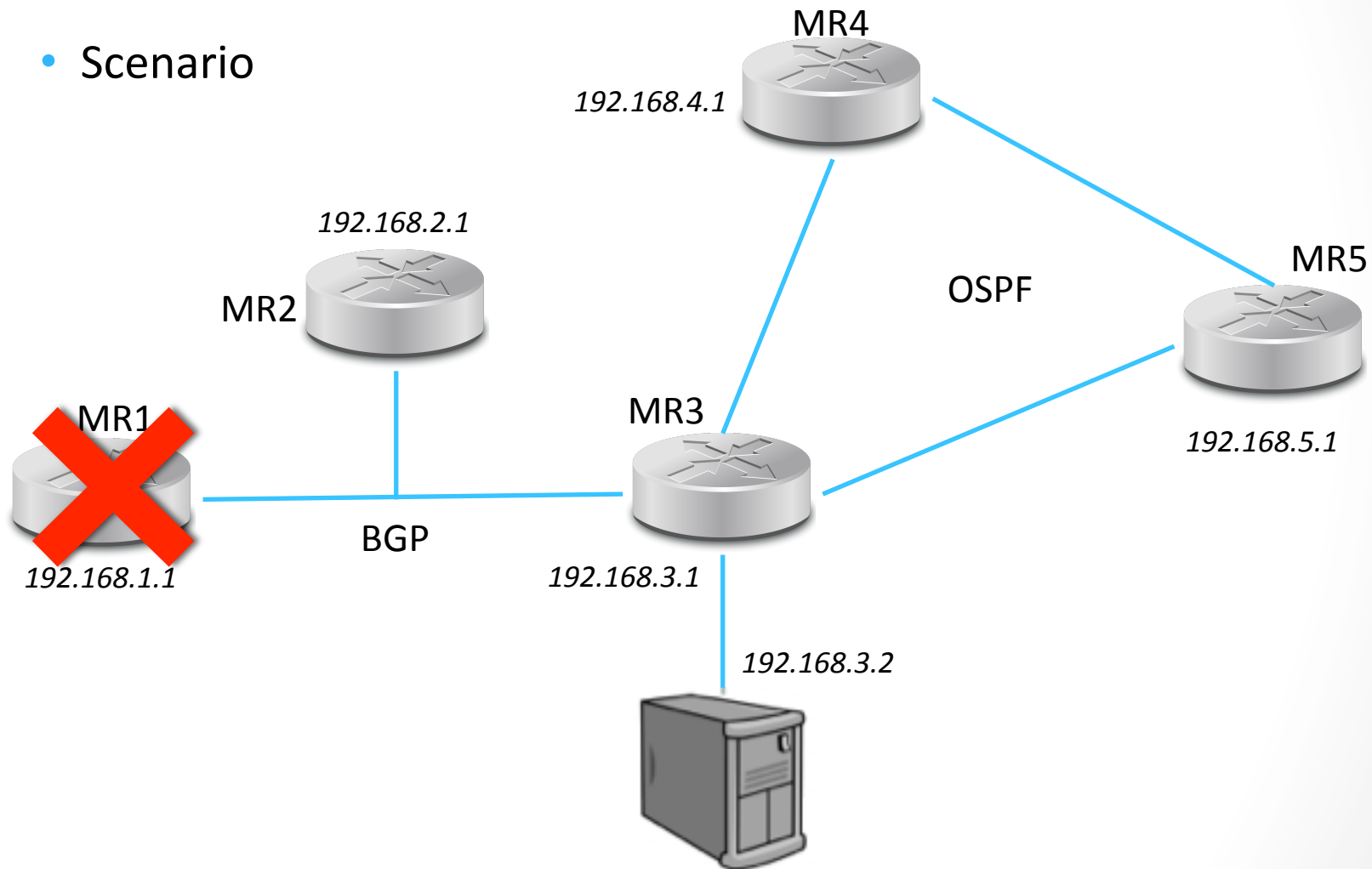
Nagios Plugin Structure

- Return Value
 - Print a line to send out to Centreon/Nagios as Status Information
 - 2nd line, if any, will be considered as Extended Information
 - Send out a performance data to be graphed using RRD Tool
 - At the end of the script, we have to send out notification whether this service state is
 - OK – 0
 - WARNING – 1
 - CRITICAL – 2
 - UNKNOWN – 3 or DEPENDENT - 4

🚩 Status Details	
Service Status	OK
Status information	Total memory used : 9% ram used : 61%, swap used 0%
Extended status information	
Performance Data	used=148750336o size=1600995328o

Case Study

- Scenario



Monitoring OSPF

- What parameter do we need?
 - Router IP
 - API Port (in this case, we use the default port)
 - Username and Password for the API
 - Interface NAME / NUMBER
 - Threshold Value
- We will create a help menu which will be shown if there is uncompleted parameters given

Monitoring OSPF through API

- Command Prompt Parameters

```
usage: $0 -m <mtik_ip> -u <user> -p <passwd>
```

```
-h : help (this message)  
-m : hostname or IP of Mikrotik router  
-u : admin username  
-p : password  
-l : list of interface  
-i : interface number  
-w : warning threshold (in Kbps)  
-c : critical threshold (in Kbps)
```

Monitoring OSPF through API

- Concept

```
./check_ospf.pl -m <RA> -u <U> -p <P> -l
```



Will list all the corresponding interface inside this router

```
./check_ospf.pl -m <RA> -u <U> -p <P> -i ether1
```



Will show the OSPF Status, along with the utilization of interface name ether1 with condition like this :

- IF the status of OSPF <> FULL, then considered CRITICAL

About API in PERL

- Created by a forum member called “cheesegrits”
- He provide some sample source-code and one of it is acting like terminal for API
- Improvement from the original module :
 - Accept “?” sign rather than only “=” for the command parameter
 - Improve output (used to be hang for more than 1kB output)
 - Adding some subprocedure
 - Sub `getall_by_key` , to list all the result based on `.id`
 - Sub `get_by_key`, to get a list of result based on `.id` as `search_key`
 - Sub `get_by_name`, to get a list of result based on custom `search_key`
 - Sub `get_by_value`, to get one single value of an item (for example to get the status of interface name “ether1”)

About API Command

- Must be started with Command Word, followed by Attribute Word (or Query Word), then terminated by zero-length Word
- API Command Word
 - It's a command in API
 - Almost the same as the terminal command syntax, but no space, instead use "/" as the replacement
 - Special API command is : *getall*, *login*, *cancel*
 - Example
 - `/interface/getall`
 - `/interface/set`
 - `/ip/address/print`
 - `/login`
 - `/interface/wireless/remove`

About API Attribute

- API Attribute Word
 - It's the value depend on the content of a command
 - Started with “=” followed by the attribute name, followed by “=” then end with the attribute value
 - Example
 - `=name=ether1`
 - `=status=enable`
 - `=.proplist=name,mtu,type,running`

About API Query

- API Query Attribute
 - Used only for “print” and “getall”
 - Start with “?”, followed by attribute name (or additional command), followed by “=” then end by attribute value
 - Example
 - `?status` (means if THERE IS a attribute named “status”)
 - `?name=ether1` (means if NAME is ether1)
 - `?-name=ether5` (means if NAME is NOT ether5)
 - `?>comment=` (means if there is non-empty comment)
 - `?#<operator>` (means popup 2 value just before this query then compare with operator)
 - The operator can be “|” (or), “&” (and), “!” change top value with opposite, etc

How to List the OSPF Interface

- In terminal, if I want to list the interface, the command is

```
/interface print
```

- In API, we convert the Terminal Command into API format

```
/interface/getall  
=.proplist=name
```

How to List the OSPF Interface

- In PERL, the command will look like this

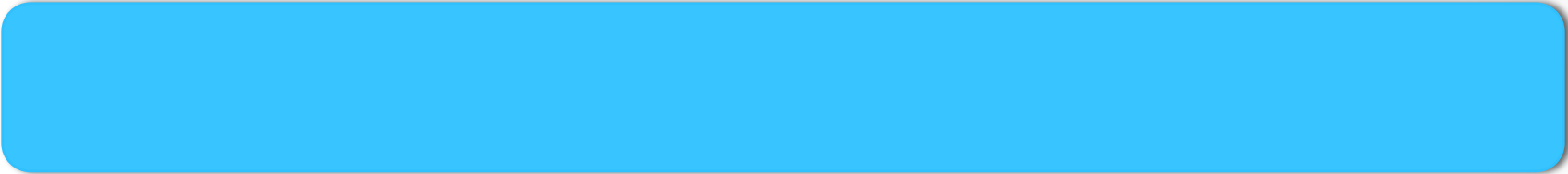
```
my(%attrs);
$attrs{'=.proplist'} = 'name';
my(%results) = Mtik::get_by_key('/interface/getall', \%attrs);
print "List of interface in router $mtik_host\n";
foreach my $item (keys(%results)) {
    my($intno) = $results{$item}{'id'};
    my($intname) = $results{$item}{'name'};
    print " $intno - $intname \n";
}
```

- And the result would be

```
[root@localhost plugins]# ./check_ospf.pl -m 192.168.3.1 -u api -p test -l
List of interface in router 192.168.3.1
*3 - ether3
*4 - ether4
*2 - ether2
*1 - ether1
```


Monitoring OSPF through API

- Concept



```
./check_ospf.pl -m <RA> -u <U> -p <P> -i ether1
```



Will show the OSPF Status, along with the utilization of interface name ether1 with condition like this :

- IF the status of OSPF \neq FULL, then considered CRITICAL

OSPF Neighbor Check

- In terminal, the command is

```
/routing ospf neighbor print
```

- In API, it looks like this

```
/routing/ospf/neighbor/getall  
?interface=<interface_name>  
=.proplist=interface,state,adjacency
```

OSPF Neighbor Check

```
#get the interface status based on interface name
$ospfattrs{'=.proplist'} = 'interface,state,adjacency';
my(%results) = Mtik::get_by_name
    ('/routing/ospf/neighbor/getall',
     'interface', $intname, \%ospfattrs);

if (%results) {
    # IF the result is non empty, then check the state
    $state = $results{$intname}{'state'};
    $adjacency = $results{$intname}{'adjacency'};
    if ($state ne "Full") {
        $errmsg = "OSPF for $intname status is $state";
        $status = "WARNING";
    } else {
        $status = "OK";
    }
} else {
    # IF the result is empty, then it might be not there
    $errmsg = "OSPF for $intname status not connected";
    $status = "CRITICAL";
}
```

Final RESULT

```
my %ERRORS=( 'OK'=>0,  
              'WARNING'=>1,  
              'CRITICAL'=>2,  
              'UNKNOWN'=>3,  
              'DEPENDENT'=>4 );
```

```
if ($errmsg) {  
    print $errmsg."\n";  
} else {  
    print "$status : "OSPF status for $intname  
           is $state for $adjacency \n";  
}  
exit $ERRORS{$status};
```

Command Prompt RESULT

LIST all the interface

```
[root@localhost plugins]# ./check_ospf.pl -m 192.168.3.1 -u api -p test -l
```

List of interface in router 192.168.3.1

*3 - ether3

*4 - ether4

*2 - ether2

*1 - ether1

RESULT for OK OSPF Status (FULL)

```
[root@localhost plugins]# ./check_ospf.pl -m 192.168.3.1 -u api -p test -i *3
```

OK : OSPF status for ether3 is Full for 00:43:30

RESULT for NOT OK OSPF (status Down or not connected)

```
[root@localhost plugins]# ./check_ospf.pl -m 192.168.3.1 -u api -p test -i *1
```

OSPF for ether1 status unknown/not connected

Integrate to NAGIOS

The screenshot shows the 'Modify a Command' interface in Nagios. The 'Check' tab is selected. The 'Command Name' is 'check_ospf'. The 'Command Type' is 'Check'. The 'Command Line' is '\$USER1\$/check_ospf.pl -m \$HOSTADDRESS\$ -u api -p test -i \$ARG1\$'. The 'Argument Example' is '\$HOSTADDRESS\$'. The 'Argument Descriptions' are 'Describe arguments' and 'ARG1 : Interface ID'. There are three buttons on the right: '<<' (path to the), '<<' (/Centreon/SNMP), and '<<' (\$ADMINEMAILS).

```
$USER1$/check_ospf.pl -m $HOSTADDRESS$ -u api -p test -i $ARG1$
```



/usr/lib/nagios/plugins



IP Address of the HOST



ARGUMENT1 – could be different for each service

Attach it to HOST

Service Configuration Relations Data Processing Service Extended Info

Modify a Service

General Information

Description • OSPF-to-MR5

Service Template generic-service

Service State

Is Volatile ☐ Yes ☐ No ☒ Default

Check Period •

Check Command • check_ospf

Args

Argument	Value	Example
Interface ID	*3	

Max Check Attempts •

Normal Check Interval • * 30 seconds

Retry Check Interval • * 30 seconds


Active Checks Enabled ☐ Yes ☐ No ☒ Default

Passive Checks Enabled ☐ Yes ☐ No ☒ Default

Command short-name

ARGUMENT1 : the interface
number

TESTING

 Status Details	
Service Status	OK
Status information	OK : OSPF status for ether3 is Full for 00:11:27
Extended status information	
Performance Data	
Current Attempt	1 / 2

 Status Details	
Service Status	CRITICAL
Status information	OSPF for ether3 status unknown/not connected
Extended status information	
Performance Data	
Current Attempt	2 / 2

Drawbacks

- API connection will constantly initiate and closed each time the monitoring tools doing polling to the device / host
- Not as fast as SNMP (since we are using TCP Socket conn)

Improvement

- Instead of just checking the OSPF status, why don't we check the traffic utilization as well and give alert if it reach some threshold?

```
./check_ospf -m <RA> -u <U> -p <P> -i ether1 -w 10 -c 100
```



Will show the OSPF Status, along with the utilization of interface name ether1 with condition like this :

- IF the traffic utilized is more than 10kbps (-w 10) then this service status is considered WARNING
- IF the traffic utilized is more than 100kbps (-c 100) then this service status is considered CRITICAL
- IF the status of OSPF <> FULL, then considered CRITICAL

GRAPH the TX and RX traffic

Traffic Utilization

- IF the traffic utilized is more than 10kbps (-w 10) then this service status is considered WARNING
- IF the traffic utilized is more than 100kbps (-c 100) then this service status is considered CRITICAL
- First of all, we will take the external value for the WARNING and CRITICAL threshold
 - WARNING threshold is taken by parameter `-w`
 - CRITICAL threshold is taken by parameter `-c`

Traffic Utilization

- In Terminal we write it like this

```
/interface monitor-traffic [ether1]
```

- In API, we write it like this

```
/interface/monitor-traffic  
=once=  
=interface=[ether1]
```

Traffic Utilization

```
### TAKING the interface number from the parameter
my($intno) = $options{'i'};

### Getting the interface name (the monitor-traffic use name)
$intattrs{'=.proplist'} = 'name';
$intattrs{'id'} = $intno;
$intname = Mtik::get_value_by_id
            ('/interface/getall', $intno, 'name', \%intattrs);

### Getting the real traffic from monitor-traffic command
$trafficator{'=.proplist'} =
            'rx-bits-per-second, tx-bits-per-second';
$trafficator{'=once'} = '';
$trafficator{'=interface'} = $intname;
my(%traffics) = Mtik::get_by_key
                ('/interface/monitor-traffic', \%trafficator);
$txbits = $traffics{$intno}{'tx-bits-per-second'};
$rxbits = $traffics{$intno}{'rx-bits-per-second'};
```

Traffic Utilization

- Now we compare the bits received with the actual Threshold

```
if ($txbits > $warningbits || $rxbits > $warningbits) {
    $retmsg .= " but the traffic exceeded the threshold";
    $status = "WARNING";
} elseif ($txbits > $criticalbits || $rxbits > $criticalbits) {
    $retmsg .= " but the traffic exceeded the threshold";
    $status = "CRITICAL";
}

print "$status : $retmsg \n";
printf("Traffic Utilization : TX : %.2f ".$txprefix."bps/
      RX : %.2f ".$rxprefix."bps\n"
      , $txdispbits, $rxdispbits);
print "|traffic_in=".$txbits."Bits/s;
      $warningbits;$criticalbits
      traffic_out=".$rxbits."Bits/s;
      $warningbits;$criticalbits\n";
exit $ERRORS{$status};
```

Traffic Utilization - COMMAND

When the OSPF is OK and the traffic is OK

```
[root@localhost]# ./check_ospf.pl -m 192.168.3.1 -u api -p test -i *4
OK : OSPF status for ether4 is Full for 00:49:37
Traffic Utilization : TX : 0.00 bps/ RX : 0.00 bps
|traffic_in=0Bits/s;100000;1000000 traffic_out=0Bits/s;100000;1000000
```

When the OSPF is OK but the traffic exceed the threshold

```
[root@localhost]# ./check_ospf.pl -m 192.168.3.1 -u api -p test -i *3
WARNING : OSPF status for ether3 is Full for 00:01:49
but the traffic exceeded the threshold
Traffic Utilization : TX : 131.97 kbps/ RX : 130.43 kbps
|traffic_in=131968Bits/s;100000;1000000
traffic_out=130432Bits/s;100000;1000000
```

Visual Result

Status Details

Service Status	UNKNOWN
Status information	no socket :No route to host
Extended status information	Couldnt log in to 192.168.3.1, probably API is not enabled
Performance Data	

Status Details

Service Status	WARNING
Status information	WARNING : OSPF status for ether3 is Full for 00:07:53 but the traffic exceeded the threshold
Extended status information	Traffic Utilization : TX : 132.80 kbps/ RX : 131.14 kbps
Performance Data	traffic_in=132800Bits/s;100000;1000000 traffic_out=131136Bits/s;100000;1000000

Status Details

Service Status	CRITICAL
Status information	OSPF for ether3 status unknown/not connected
Extended status information	

Status Details

Service Status	OK
Status information	OK : OSPF status for ether3 is Full for 00:00:45
Extended status information	Traffic Utilization : TX : 0.00 bps/ RX : 0.00 bps
Performance Data	traffic_in=0Bits/s;100000;1000000 traffic_out=0Bits/s;100000;1000000

What's NEXT?

- Basically we can monitor and graph anything
 - Graph BGP prefixes received and alert when the BGP DOWN or the prefixes reach some low threshold
 - Graph the number of Active Hotspot user, Host that connected to a Hotspot server, and the number of DHCP Lease that has been established
 - Graph the number of station that connect to an Access Point
 - Graph TX/RX Rate and CCQ of a connection and send alert once they goes below certain threshold
- Centreon and Nagios also provide
 - Passive Check
 - Lots of Modules and Plugins

<http://project.spectrumindo.com>

<http://www.mikrotiktraining.co.id>

FURTHER QUESTION

herry@spectrumindo.com