BCNET 2019

# Monitoring and Alerting on Blackhole Routes

(with NETCONF and Perl)

Craig Tomkow, BCNET

# Remotely Triggered Blackhole Routing

- BCNET working on deploying RTBH Routing
- Downstream advertises prefix to BCNET with community 271:666 (including from our own private AS's)
- Monitor and Alerting?
  - Don't forget about our own blackhole routes
  - Visibility

# How? A couple different options

- Have a server setup a BGP peer with each core router (e.g. ExaBGP)
  - Pro: Instant knowledge of when a route is blackholed
  - Con: Additional setup required

- Periodically poll all routers for blackhole routes
  - Pro: Simple setup and integration with existing software tool
  - Con: Slower updates (5 minute polling)

# Periodic Polling Plugin in CMDB

- Existing CMDB software has ability to add custom plugins
  (shameless plug: https://github.com/netharbour/netharbour)

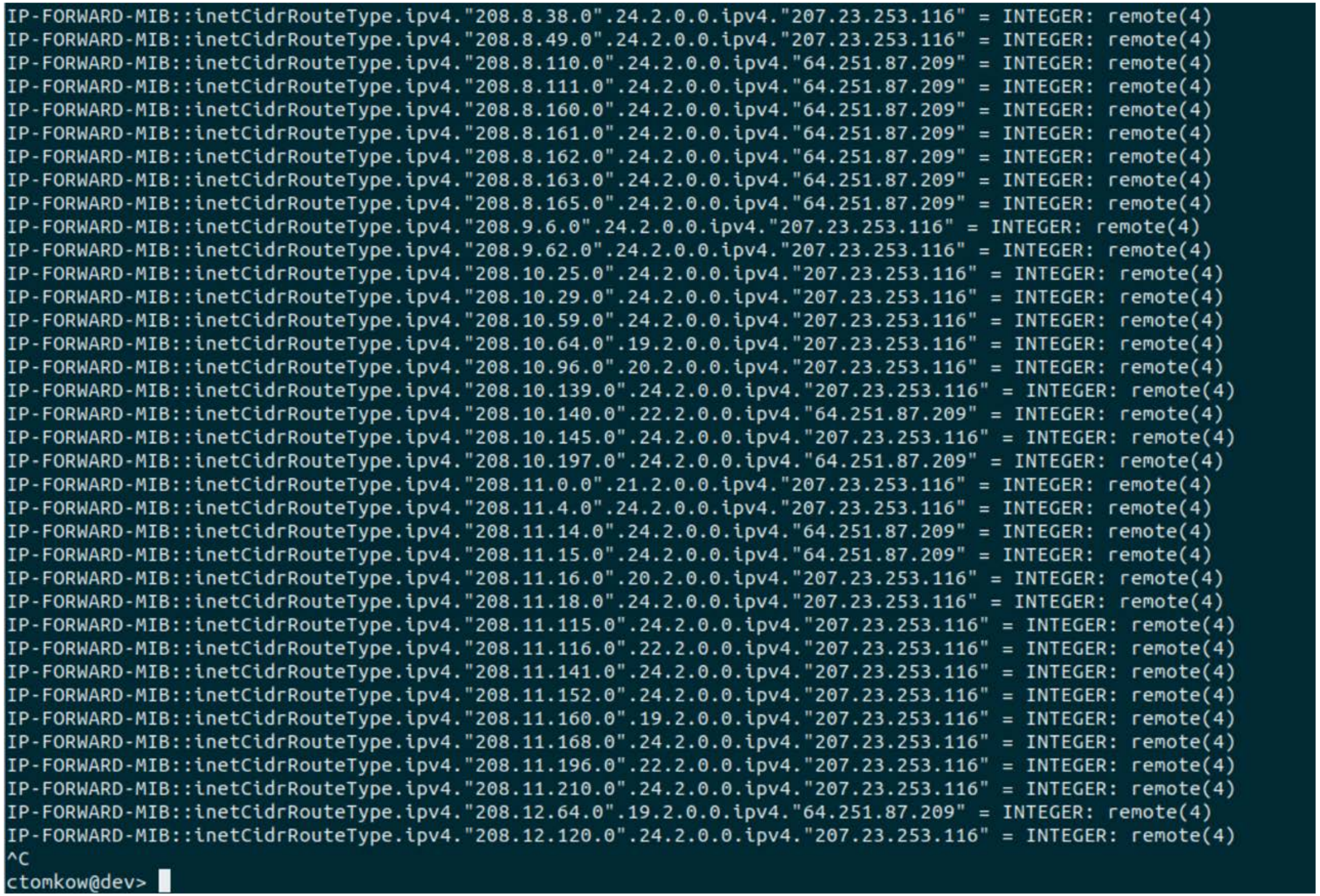
- Leverages existing database
  - Existing inventory of routers
  - Ability to add DB tables for custom plugin easily

- CMDB scheduled to run plugin script every 5 minutes

# Polling Script

- Perl. Why not Python, or any other language??
  - All CMDB plugins are in Perl. Keep the codebase consistent.
  - Simplifies system dependencies
- SNMP vs NETCONF
  - All CMDB polling uses SNMP. So keep using SNMP?
  - Only way to get blackhole routes was to walk the routing table...

# 5 minutes of walking the routing table...



```
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.8.38.0".24.2.0.0.ipv4."207.23.253.116" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.8.49.0".24.2.0.0.ipv4."207.23.253.116" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.8.110.0".24.2.0.0.ipv4."64.251.87.209" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.8.111.0".24.2.0.0.ipv4."64.251.87.209" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.8.160.0".24.2.0.0.ipv4."64.251.87.209" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.8.161.0".24.2.0.0.ipv4."64.251.87.209" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.8.162.0".24.2.0.0.ipv4."64.251.87.209" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.8.163.0".24.2.0.0.ipv4."64.251.87.209" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.8.165.0".24.2.0.0.ipv4."64.251.87.209" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.9.6.0".24.2.0.0.ipv4."207.23.253.116" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.9.62.0".24.2.0.0.ipv4."207.23.253.116" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.10.25.0".24.2.0.0.ipv4."207.23.253.116" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.10.29.0".24.2.0.0.ipv4."207.23.253.116" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.10.59.0".24.2.0.0.ipv4."207.23.253.116" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.10.64.0".19.2.0.0.ipv4."207.23.253.116" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.10.96.0".20.2.0.0.ipv4."207.23.253.116" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.10.139.0".24.2.0.0.ipv4."207.23.253.116" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.10.140.0".22.2.0.0.ipv4."64.251.87.209" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.10.145.0".24.2.0.0.ipv4."207.23.253.116" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.10.197.0".24.2.0.0.ipv4."64.251.87.209" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.11.0.0".21.2.0.0.ipv4."207.23.253.116" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.11.4.0".24.2.0.0.ipv4."207.23.253.116" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.11.14.0".24.2.0.0.ipv4."64.251.87.209" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.11.15.0".24.2.0.0.ipv4."64.251.87.209" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.11.16.0".20.2.0.0.ipv4."207.23.253.116" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.11.18.0".24.2.0.0.ipv4."207.23.253.116" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.11.115.0".24.2.0.0.ipv4."207.23.253.116" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.11.116.0".22.2.0.0.ipv4."207.23.253.116" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.11.141.0".24.2.0.0.ipv4."207.23.253.116" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.11.152.0".24.2.0.0.ipv4."207.23.253.116" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.11.160.0".19.2.0.0.ipv4."207.23.253.116" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.11.168.0".24.2.0.0.ipv4."207.23.253.116" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.11.196.0".22.2.0.0.ipv4."207.23.253.116" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.11.210.0".24.2.0.0.ipv4."207.23.253.116" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.12.64.0".19.2.0.0.ipv4."64.251.87.209" = INTEGER: remote(4)
IP-FORWARD-MIB::inetCidrRouteType.ipv4."208.12.120.0".24.2.0.0.ipv4."207.23.253.116" = INTEGER: remote(4)
^C
ctomkow@dev>
```

# A Better Way - NETCONF

- Remote procedure call (RPC) over SSH as XML

- Enable NETCONF on Junos

```
set system services netconf ssh port 22
```

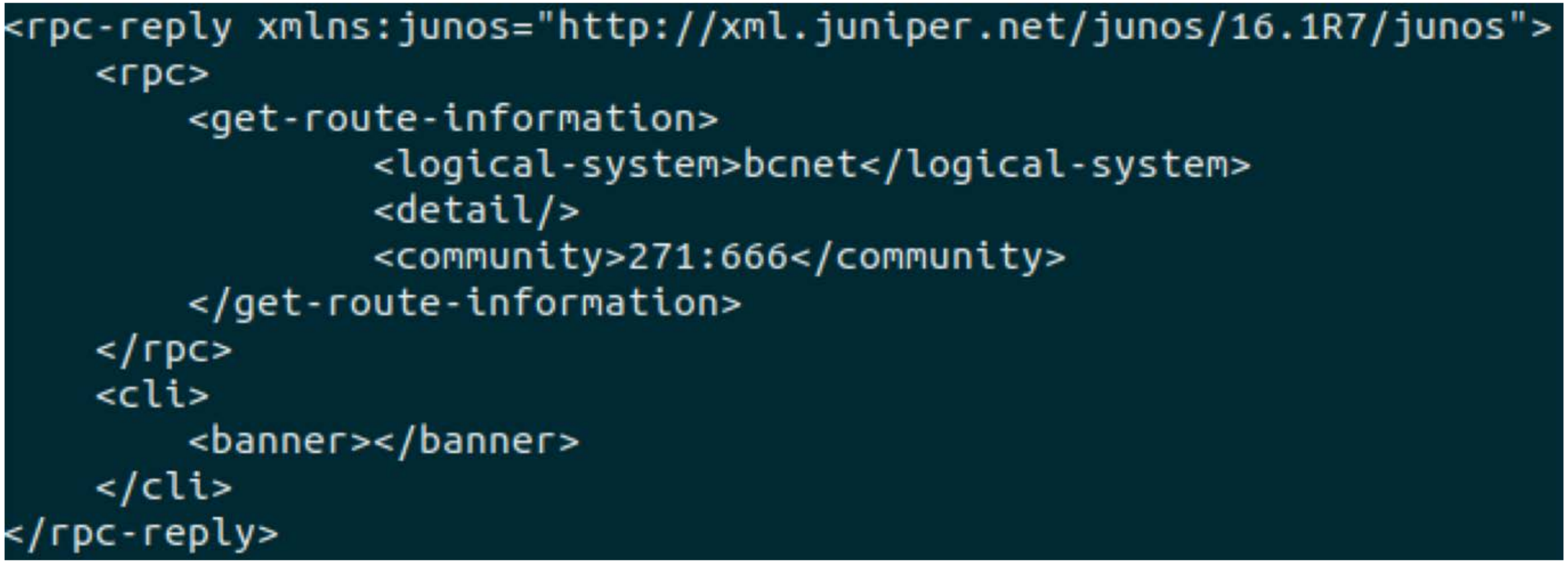
- Use Juniper maintained NETCONF library (in CPAN!)

```
Net::Netconf
```

# More NETCONF

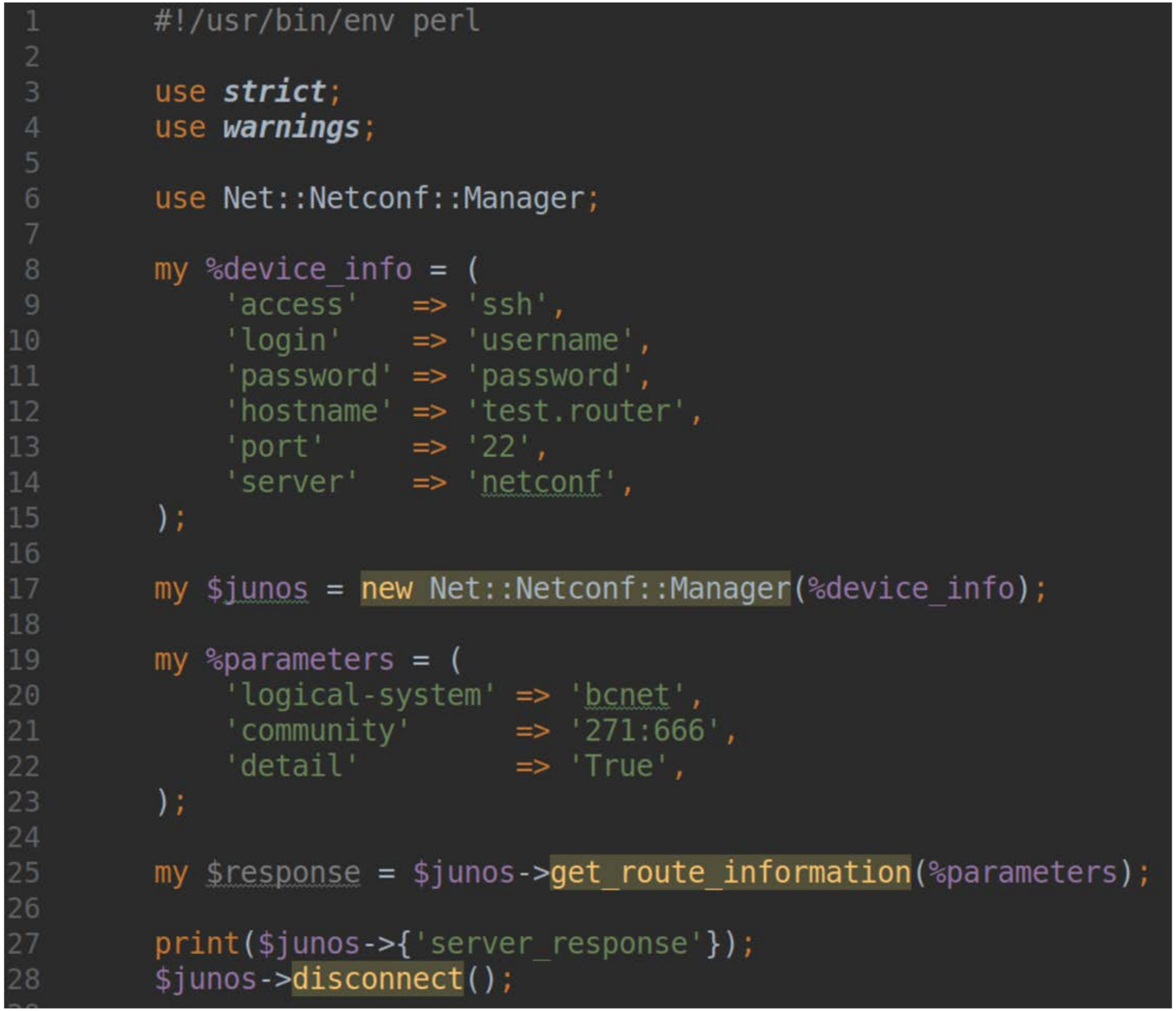
- Convert Junos CLI command into RPC XML

```
show route logical-system bcnet detail community 271:666 | display xml rpc
```



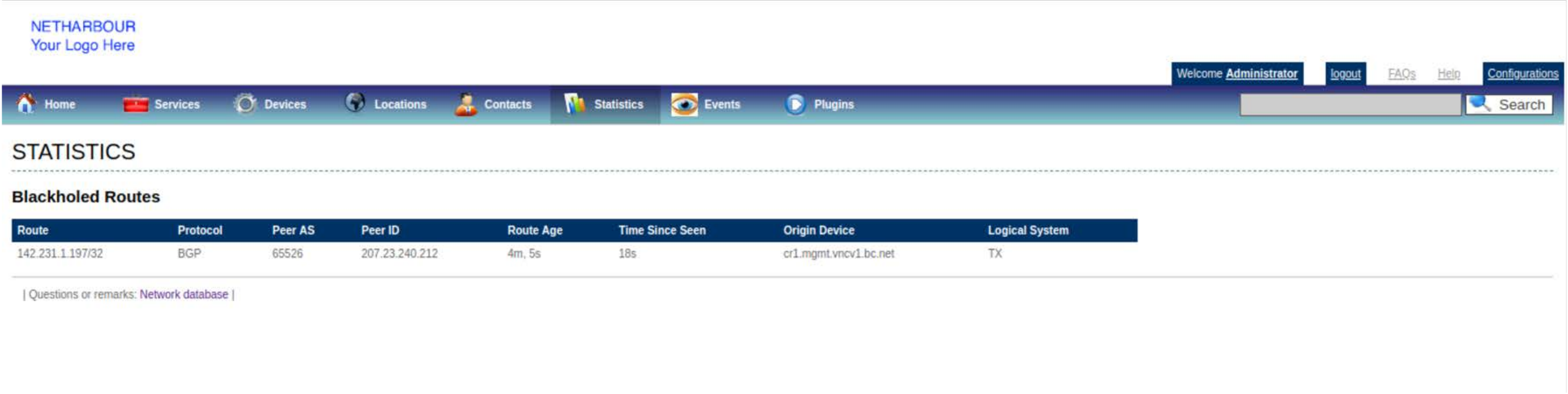
```
<rpc-reply xmlns:junos="http://xml.juniper.net/junos/16.1R7/junos">
    <rpc>
        <get-route-information>
                <logical-system>bcnet</logical-system>
                <detail/>
                <community>271:666</community>
        </get-route-information>
    </rpc>
    <cli>
        <banner></banner>
    </cli>
</rpc-reply>
```

# Send XML, or use NETCONF library



```
#!/usr/bin/env perl

use strict;
use warnings;

use Net::Netconf::Manager;

my %device_info = (
    'access'   => 'ssh',
    'login'    => 'username',
    'password' => 'password',
    'hostname' => 'test.router',
    'port'     => '22',
    'server'   => 'netconf',
);

my $junos = new Net::Netconf::Manager(%device_info);

my %parameters = (
    'logical-system' => 'bcnet',
    'community'      => '271:666',
    'detail'         => 'True',
);

my $response = $junos->get_route_information(%parameters);

print($junos->{'server_response'});
$junos->disconnect();
```

# CMDB Plugin GUI



## STATISTICS

### Blackholed Routes

| Route | Protocol | Peer AS | Peer ID | Route Age | Time Since Seen | Origin Device | Logical System |
|---|---|---|---|---|---|---|---|
| 142.231.1.197/32 | BGP | 65526 | 207.23.240.212 | 4m, 5s | 18s | cr1.mgmt.vncv1.bc.net | TX |

| Questions or remarks: Network database |

# What about Alerting?

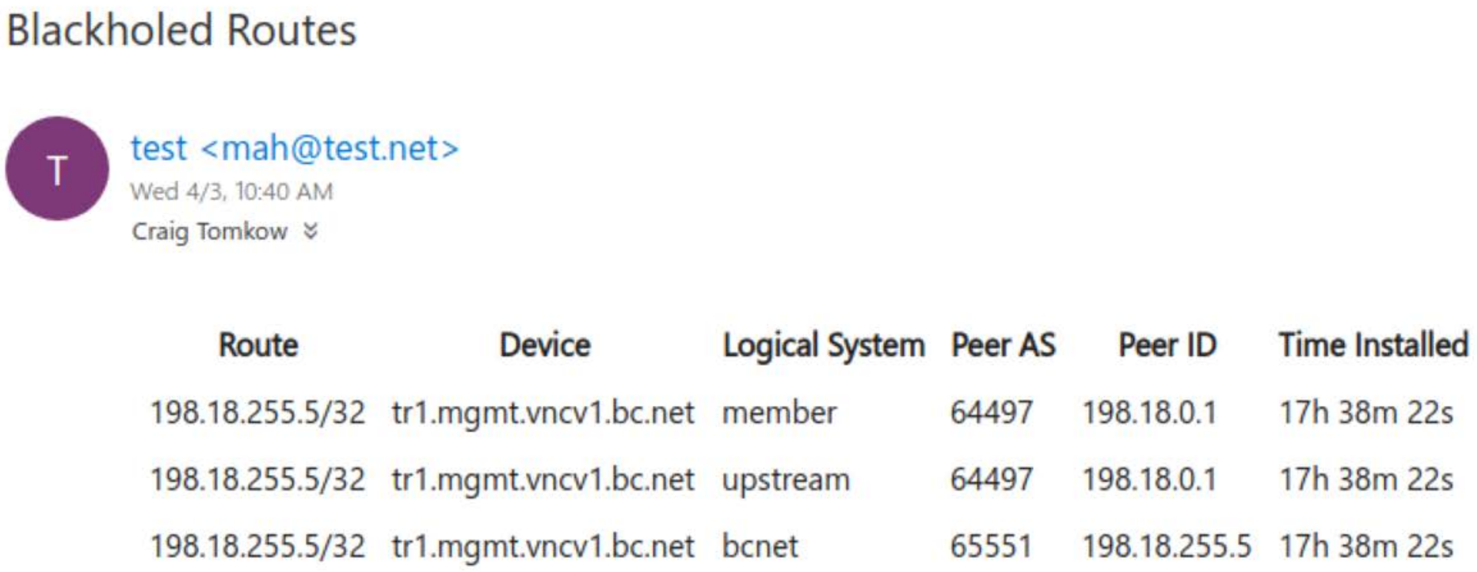
- Email notification of blackhole routes
  - Notify routes that match peer AS(s)
  - Notify after *X* seconds (e.g. 24 hours or 86400 seconds)
  - Repeat notification every *X* seconds

  - Alerting options configurable in the plugin's .conf file

  - Currently only notifies our internal network team email

# Email Alert

- Simple HTML table



Blackholed Routes

test <mah@test.net>
Wed 4/3, 10:40 AM
Craig Tomkow

| Route | Device | Logical System | Peer AS | Peer ID | Time Installed |
|---|---|---|---|---|---|
| 198.18.255.5/32 | tr1.mgmt.vncv1.bc.net | member | 64497 | 198.18.0.1 | 17h 38m 22s |
| 198.18.255.5/32 | tr1.mgmt.vncv1.bc.net | upstream | 64497 | 198.18.0.1 | 17h 38m 22s |
| 198.18.255.5/32 | tr1.mgmt.vncv1.bc.net | bcnet | 65551 | 198.18.255.5 | 17h 38m 22s |

# Thank You!

Email: craig.tomkow@bc.net

Github: https://github.com/ctomkow

Netharbour: https://github.com/netharbour/netharbour

BCNET 2019