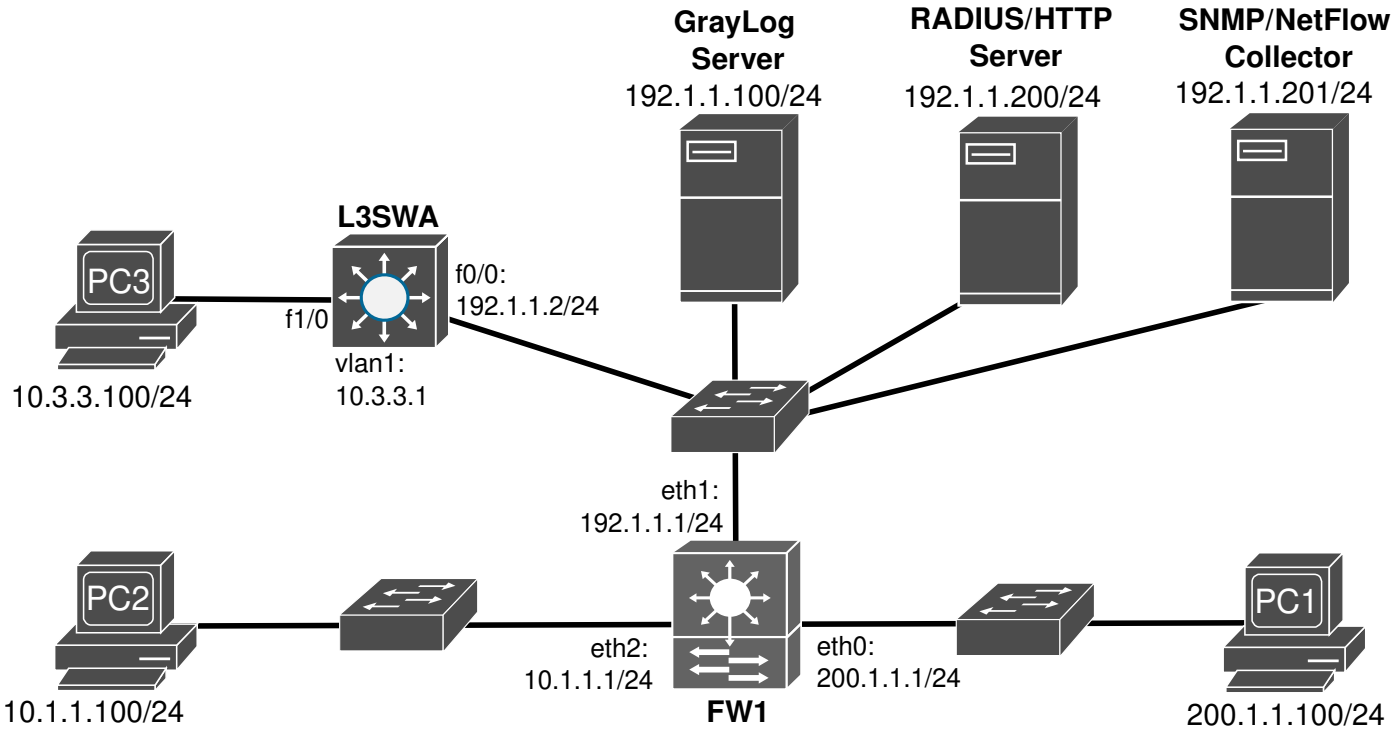
Segurança em Redes de Comunicações

Network Monitoring



Assemble the above network in GNS3, with a:
- *GrayLog server VM;*
- *Debian Linux VM server with Freeradius and Apache2 active services;*
- A layer 3 switch (L3SWA). L3SWA will be the gateway for network 192.1.1.0/24;
- PC (PC3), that should be a Linux VM;
- *(Optional) Debian Linux VM server to run SNMP and NetFlow collectors (can be the RADIUS/HTTP server);*
- (Optional) VyOS firewall (configure static or dynamic routing with SWL3A), and
- (Optional) PC1 and PC2, that can be VPCS devices.

Configure IPv4 addresses and test the full connectivity.

Syslog

1. At the Linux server, the *syslog* service should be running by default (if not, start it).

```
$ systemctl status syslog
$ systemctl restart syslog
```

Guarantee also that the Apache service is running (if not, start it).

```
$ systemctl status apache2
$ systemctl restart apache2
```

Analyze the Apache access log at /var/log/apache2/access.log. Open a terminal, and monitor new entries in the Apache access log file:

```
$ tail –f /var/log/apache2/access.log
```

From PC3 access the default web page in the HTTP server. Access multiple objects:
- *http://192.1.1.200/*
- *http://192.1.1.200/icons/openlogo-75.png*
- *http://192.1.1.200/icons//world2.png*
- *http://192.1.1.200/file1.txt*
- *http://192.1.1.200/img1.png*

>> Analyze the log entries and formats.

Rsyslog

2. At the Linux server guarantee that the *rsyslog* service is running (if not, start it).

```
$ systemctl status rsyslog
$ systemctl restart rsyslog
```

Create a rsyslog configuration file to pool the local Apache2 access log (10 sec interval) and perform a remote logging to the GrayLog server (using UDP port 1514):

Create the file **/etc/rsyslog.d/02-apache2-access.conf** with the following content:

```
module(load="imfile" PollingInterval="10")
input(type="imfile"
      File="/var/log/apache2/access.log"
      Tag="http_access"
      Severity="info"
      Facility="local6")
local6.info        @192.1.1.100:1514
```

Test the configuration file with the following command:

```
$ rsyslogd -N1 -f /etc/rsyslog.d/02-apache2-access.conf
```

If correct, restart the rsyslog service:

```
$ systemctl restart rsyslog
```

Note: the default rsyslog UPD port is 514, however it is easier in GrayLog to use a port higher than 1024, then 1514 was chosen.

Note2: without a UDP syslog input at the GrayLog server, the server will respond to the rsyslog messages with a ICMP port unreachable message.

3. Start a packet capture on the HTTP server link, and from PC3 access the default web page in the HTTP server. Access multiple objects:

- http://192.1.1.200/

- http://192.1.1.200/icons/openlogo-75.png

- http://192.1.1.200/icons//world2.png

- http://192.1.1.200/file1.txt

- http://192.1.1.200/img1.png

>> Analyze the captured rsyslog messages.

Note: Since a non-default port is being used you must configure Wireshark to decode UDP 1514 packets as rsyslog. Right-click on one Udp 1514 packet and choose option “Decode as…”, change port value to 1514 and default to syslog and save it.

4. Restrict the Apache access remote logging only to 404 errors. Rename the file **/etc/rsyslog.d/02-apache2-access.conf** to **/etc/rsyslog.d/02-apache2-404errors.conf** and add to the first line:

```
if not ($msg contains '404') then stop
```

Test the configuration file with the following command:

```
$ rsyslogd -N1 -f /etc/rsyslog.d/02-apache2-404errors.conf
```

If correct, restart the rsyslog service:

```
$ systemctl restart rsyslog
```

Start a packet capture on the HTTP server link, and from PC3 access the default web page in the HTTP server. Access multiple objects:

- http://192.1.1.200/

- http://192.1.1.200/icons/openlogo-75.png

- http://192.1.1.200/icons//world2.png

- http://192.1.1.200/file1.txt

- http://192.1.1.200/img1.png

>> Analyze the captured rsyslog messages and verify that only 404 access error are being remotely reported.

5. Enable the rsyslog report in Cisco IOS at the L3SWA:

```
L3SWA(config)# logging host 192.1.1.100 transport udp port 1514
L3SWA(config)# logging trap debugging                                    !logs all debug messages
L3SWA(config)# logging facility local6
L3SWA(config)# logging console informational                !disables debug messages to console
L3SWA(config)# exit
L3SWA# debug ip routing                                                  !enables IPv4 routing debug
```

Start a packet capture on the GrayLog server link, and add a default static route from SWL3A to the FW1:

```
L3SWA(config)# ip route 0.0.0.0 0.0.0.0 192.1.1.1
```

>> Analyze the captured rsyslog messages.

>> (Optional) test other relevant debug messages. For more details about the available debug commends see here.

(optional) 6. Enable the rsyslog report in the VyOS firewall:

```
set system syslog host 192.1.1.100 facility all level all
set system syslog host 192.1.1.100 facility all protocol udp
set system syslog host 192.1.1.100 port 1514
```

Start a packet capture on the GrayLog server link, and generate traffic between PC1 and PC2, disable one of the network interfaces (eth0 or eth2) or logout/login.

>> Analyze the captured rsyslog messages.

Log Management

7. Boot the GrayLog server. The CLI login is ubuntu/ubuntu. Configure the IPv4 address and gateway:

```
$ sudo ip addr add 192.1.1.100/24 dev enp1s0
$ sudo ip route add default via 192.1.1.2
```

Check if port TCP 9000 is open (`netstat -lnut`), if not, you may need to increase the VM RAM. From PC3, access the GrayLog web interface via http://192.1.1.100:9000 using the following credentials: admin/labcom.

8. On the GrayLog web interface, create a new GayLog data input:

System/Inputs→Inputs: "Syslog UDP", Launch new input, give a Title (e.g., syslogUDP) and change UDP port to 1514.

Check the GrayLog Streams for the rsyslog messages:

Streams→ All messages.

Generate rsyslog messages from the Apache2 server or the Layer3 switch.

>> Identify the received syslog messages.

>> Identify the message fields decoded by default.

9. Create a new data Stream:

Streams → Create Stream → name it “HTTP 404 errors” and choose “Default Index set” → Save

Manage Rules → Select input type as “Server Input” → Select as input the syslogUDP messages input

Add 2 stream rules: (i) message must contain http_access and (ii) message must contain 404.

>> Go back to the Streams windows and verify if the new stream is reporting the correct messages (generate new syslog messages on the HTTP server).

10. The client HTTP IPv4 address is not decode by default, to create a custom message field it is required to create an extractor. Go to the previously created syslog input:

System/Inputs→Manage Extractors → Add Extractor, GetStarted

Choose a received message → Extractor type: Regular expression, Source field: message, Regular expression: “([0-9a-f\.:]+) -”, Always try to extract, Store as field: client_ip_address -> Save it.

Check the GrayLog Streams for the new rsyslog messages:

Streams→ All messages (or the newly created stream).

>> Identify the message fields decoded by default and the newly one created.

>> Create more custom fields that you consider relevant.

Note: to test regular expressions use the website https://regex101.com/.

11. Create a new Streams for syslog messages from (i) RADIUS server, (ii) Layer 3 switch and (iii-optional) VyOS firewall.

Security Information and Event Management (SIEM)

12. Create an alert/event for the HTTP server when more than 10 non-existing objects are requested from the same IPv4 address (404 messages). On the GrayLog web interface:

Alerts → Event Definitions → give a name → Next → Content type: Filter & Aggregation, Streams: HTTP 404 errors, Aggregation of results reaches a threshold, Group by Field(s): client_ip_address, Condition: if count() is > than 10 → Next → Next → Done.

>> Generate a set of requests to the HTTP server and test if the alert/event is reported by the GrayLog server.

13. Propose and implement additional alerts relevant to maintaining network security.

Note that the open version of GrayLog does not allow the correlation of multiple events. However, propose the usage of multiple events to detected more complex network anomalies.

(Optional) Monitoring with SNMP

14. On the SNMP Collector server, deploy a script to (i) perform multiple SNMP queries, (ii) log the result to a local log file, and (iii) use rsyslog to report the queries results to the Gray log server.

To configure a SNMP version 3 community (using the name “private”) with Read-Only permissions, and access with authentication (MD5, password authpass) and encryption (AES128, password: privpass), for user uDDR from group gDDR:

```
L3SWA(config)# snmp-server user uDDR gDDR v3 auth md5 authpass priv aes 128 privpass
L3SWA(config)# snmp-server group gDDR v3 priv
L3SWA(config)# snmp-server community private RO
```

(Optional) Monitoring with NetFlow/IPFIX

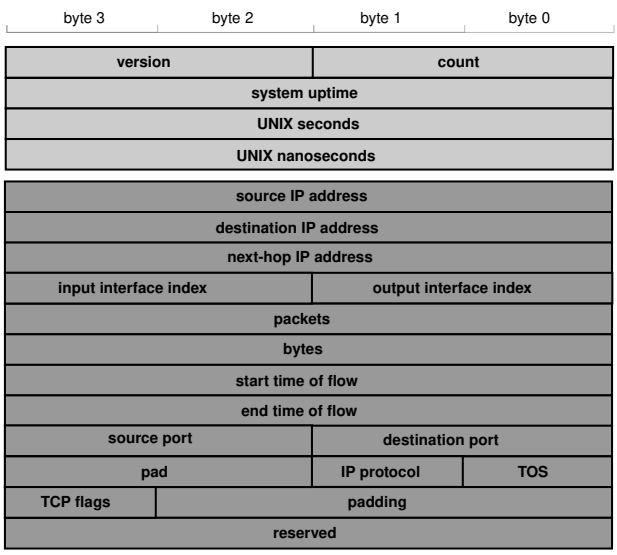
NetFlow references: NetFlow Overview

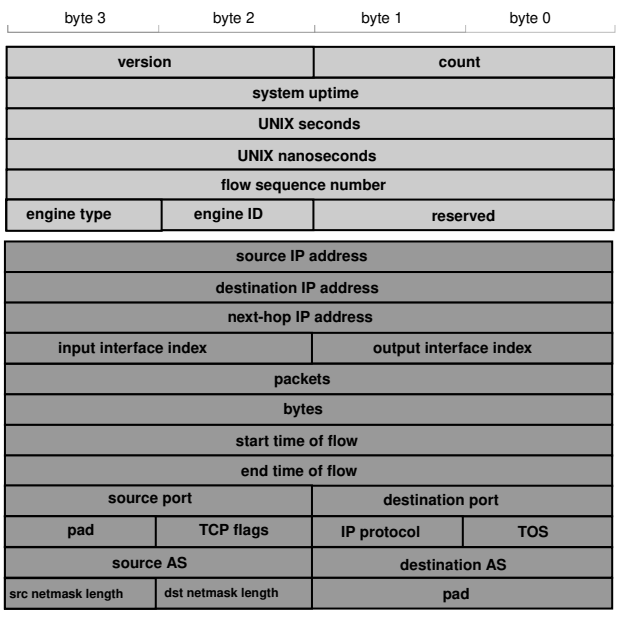
NetFlow Export Datagram Format (version 1 and 5 formats)

Python references: *socket* - Low-level networking interface, https://docs.python.org/2/library/socket.html

struct - Interpret strings as packed binary data, https://docs.python.org/2/library/struct.html

NetFlow v1 and v5 header and body formats





14. On the Netwlow Collector server, deploy a script to (i) perform receive the NetFlow messages from the network devices, (ii) log the result to a local log file, and (iii) use rsyslog to report the queries results to the Gray log server.

To configure the NetFlow in L3SWA to report all flows that egress the interface 0/0 to the NetFlow collecqtor:

```
L3SWA(config)# interface FastEthernet0/0
L3SWA(config-if)# ip flow egress
L3SWA(config)# ip flow-export destination 192.1.1.201 9996
L3SWA(config)# ip flow-export source Loopback 0
L3SWA(config)# ip flow-export version 1
```