ntopng User Guide

IT Project One

Team Unclassified

Loren Molloy, Matthew Phelps, Sarah Currie, Zeeshan Ali

Client: Kent O’Sullivan

June 17

Contents

Introduction

Scope and Purpose

ntopng is a web-based network traffic analysis and flow collection owned and maintained by ntop, and has been released and distributed under the GNU GPLv3 Licence. This user guide will detail the modifications and additions our project has contributed with the end user in mind being an Australian Army unit.

The intention for its development is to enable identification of applications running by users on a network. This is achieved through utilisation of both live captured and manual PCAP traffic files. When this data is input into the program an overview summary page will display the relevant information to the user.

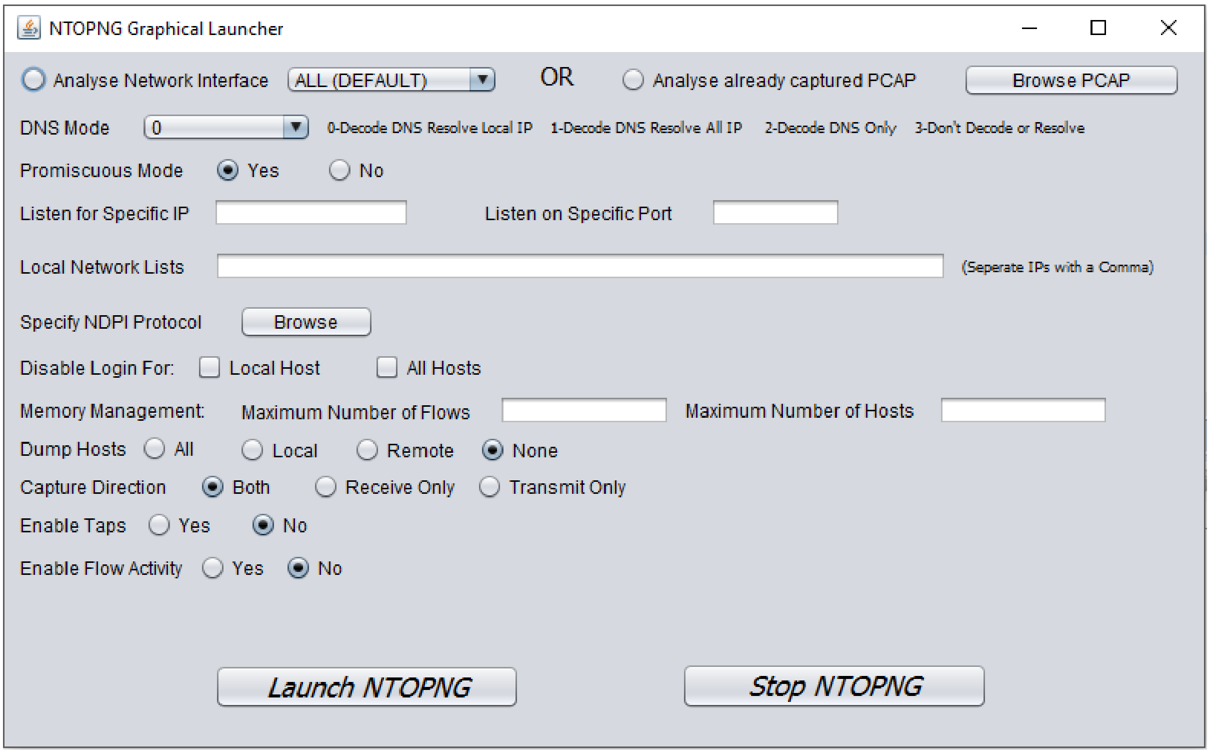
1.2 Process Overview

This user guide will include instructions, guidelines and descriptions for the following features of 'ntopng':

- Use of a GUI Wrapper to launch the ntopng service
  - How to perform additional functions with the GUI Wrapper
- Navigating the summary view
  - Changes to the community edition
  - Professional edition
- Overview of the interface points for developers
- Installation of ntopng
  - on Raspberry Pi and Netbeans
- Maintenance of ntopng
  - Non-technical
  - Technical

2. Using the GUI Wrapper

The GUI Wrapper provides a user friendly method of launching and configuring NTOPNG that is easily used by non-technical persons. It offers several simple specifications on the way in which NTOPNG monitors a network. This includes but is not limited to listening on specific network interfaces, using specific IP's and ports and analysing already captured PCAP files. If further configuration is required; instruction can be found in the original NTOPNG User Guide. This requires an operator to execute commands from the command prompt or terminal and should only be done by more technical personnel. The GUI Wrapper has been designed to be extremely user friendly and self-explanatory.



NTOPNG Graphical Launcher

○ Analyse Network Interface ALL (DEFAULT) ▼ OR ○ Analyse already captured PCAP Browse PCAP

DNS Mode 0 ▼ 0-Decode DNS Resolve Local IP 1-Decode DNS Resolve All IP 2-Decode DNS Only 3-Don't Decode or Resolve

Promiscuous Mode ● Yes ○ No

Listen for Specific IP Listen on Specific Port

Local Network Lists (Seperate IPs with a Comma)

Specify NDPI Protocol Browse

Disable Login For: □ Local Host □ All Hosts

Memory Management: Maximum Number of Flows Maximum Number of Hosts

Dump Hosts ○ All ○ Local ○ Remote ● None

Capture Direction ● Both ○ Receive Only ○ Transmit Only

Enable Taps ○ Yes ● No

Enable Flow Activity ○ Yes ● No

Launch NTOPNG Stop NTOPNG

The GUI wrapper as shown above gives a minimalist display with several options that can be changed by the operator.

2.1 Network Interface or PCAP

To run NTOPNG on a specific interface the user must first click the ‘Analyse Network Interface’ option. Then clicking the drop down menu adjacent the operator should see a list of network devices that can be selected as the focus of the NTOPNG program.

To run NTOPNG on an already captured PCAP the user must click the ‘Analyse already captured PCAP’ option. Then clicking the ‘Browse PCAP’ button will open a file explorer in which the user can select a PCAP for analysis.

2.2 DNS Mode

DNS mode has 4 options. These can be accessed in the drop down menu on the second line of the GUI launcher. Some information is given in the program about the 4 DNS modes and how they differ. The user can select the DNS mode best corresponding to their needs. DNS modes are;

0 - Decode DNS responses and resolve local numeric IPs only,

1 - Decode DNS responses and resolve all numeric IPs,

2 - Decode DNS responses and don’t resolve numeric IPs and

3 - Don’t decode DNS responses and don’t resolve numeric IPs

2.3 Promiscuous Mode

Promiscuous mode is by default enabled. To disable it, the user must simply click the 'No' option on line 3 of the GUI Launcher.

2.4 Listening for Specific IPs on Specific Ports

For the user to specify IPs and Ports they must enter these into the text boxes on line 4 of the program. The IP will go in the first text box and the port in the second text box. A user can specify an IP without a port or a port without an IP and these will still be valid at runtime.

2.5 Local Network Lists

If the user wishes to specify a Local Network List of IP Addresses before running NTOPNG they may do so on line 5 of the GUI Launcher. The IPs that are entered must be separated with a comma e.g

192.168.0.1,195.124.23.1,255.255.255.2

2.6 Specify nDPI Protocol

NTOPNG can be tuned so that it only analyses network from one application. To do this the user must have the nDPI protocol file corresponding to the application which they are analysing. The user must click the 'Browse' button on line 6 and then locate the protocol file in the file explorer that opens.

2.7 Disabling Login

The user may wish to disable login for certain hosts. They can do this on line 7 of the GUI Launcher by clicking the appropriate check boxes.

2.8 Memory Management

If NTOPNG is running on extensive networks it may be beneficial to place limits on the traffic it analyses to manage memory appropriately. The GUI Launcher offers the ability to limit the maximum number of flows and hosts.

2.9 Dump Hosts

Line 9 of the GUI Launcher offers the capability to dump certain hosts. The option for this is by default none. The operator can specify All hosts, Local hosts or Remote hosts.

2.10 Capture Direction

The operator can specify the direction of the traffic which NTOPNG will be analysing. NTOPNG can either analyse traffic in both directions, only incoming or only outgoing traffic.

2.11 Enabling Taps and Enabling Flow Activity

Lines 11 and 12 of the GUI Launcher give the user control on whether taps will be enabled or disabled and whether flow activity will be enabled or disabled.

2.12 Launching NTOPNG

After the user has inputted all of their specifications into the GUI Launcher they need to launch NTOPNG. This can be done simply by clicking the 'Launch NTOPNG' button at the bottom left of the program. The GUI Launcher will then save the users inputs into a configuration file and execute the program. The NTOPNG GUI will open in the default browser on the computer.

2.13 Stopping NTOPNG

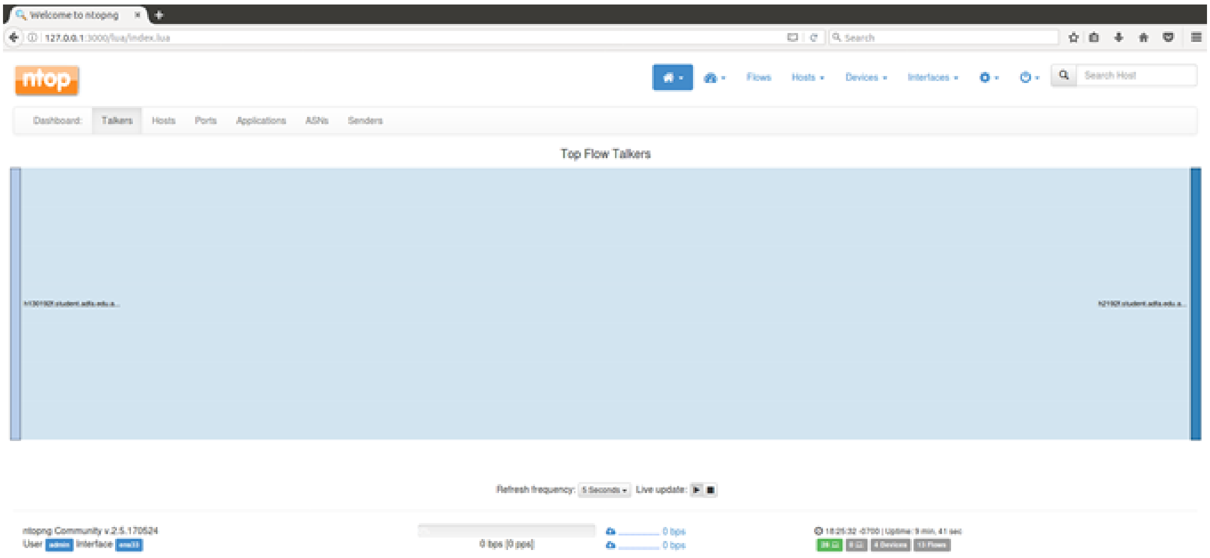
NTOPNG will continue to run in the background when the web browser has been closed and even when the GUI launcher has been closed. To properly stop the NTOPNG service from running the user must click the ‘Stop NTOPNG’ button on the bottom right of the GUI Launcher. If the GUI Launcher has been closed but NTOPNG is still running then it can be stopped by opening a new instance of the GUI Launcher and clicking the ‘Stop NTOPNG’ button.

3. Navigating the summary view

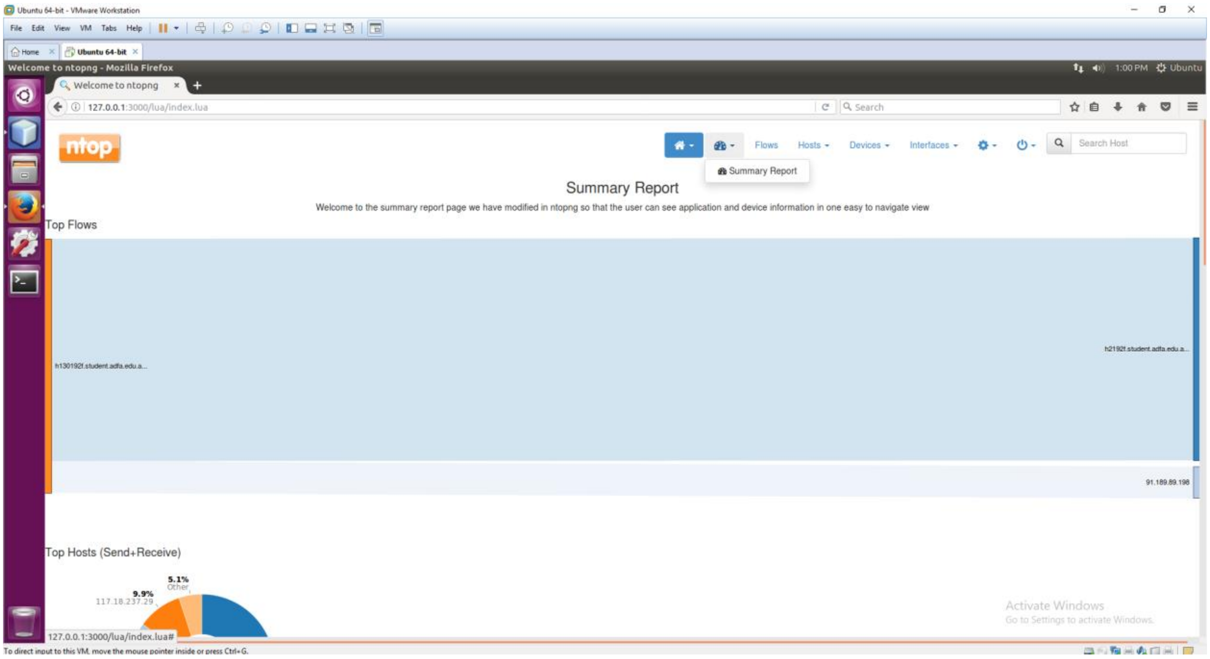
The summary page is the overview information of Talkers, Hosts, Ports, Applications, ASNs, and Senders combined into one view. This overview information is represented mostly in the form of pie chart statistics, and this is the default page that is launched after logging into ntopng. The script for this page is found in /ntopng/scripts/lua/index.lua

Each graph still has the same functionality as it did from ntop's version 2.5, which is detailed in their User Guide January 2017 (found in /ntopng/doc/UserGuide.pdf).

The benefit of combining all the information onto one page is that this page can be saved as an overview capture of the network at that time. The summary guide was intended to contain more information about the devices and applications (ports and protocols), and so the index.lua source script is easily modifiable and contains detailed code commenting.



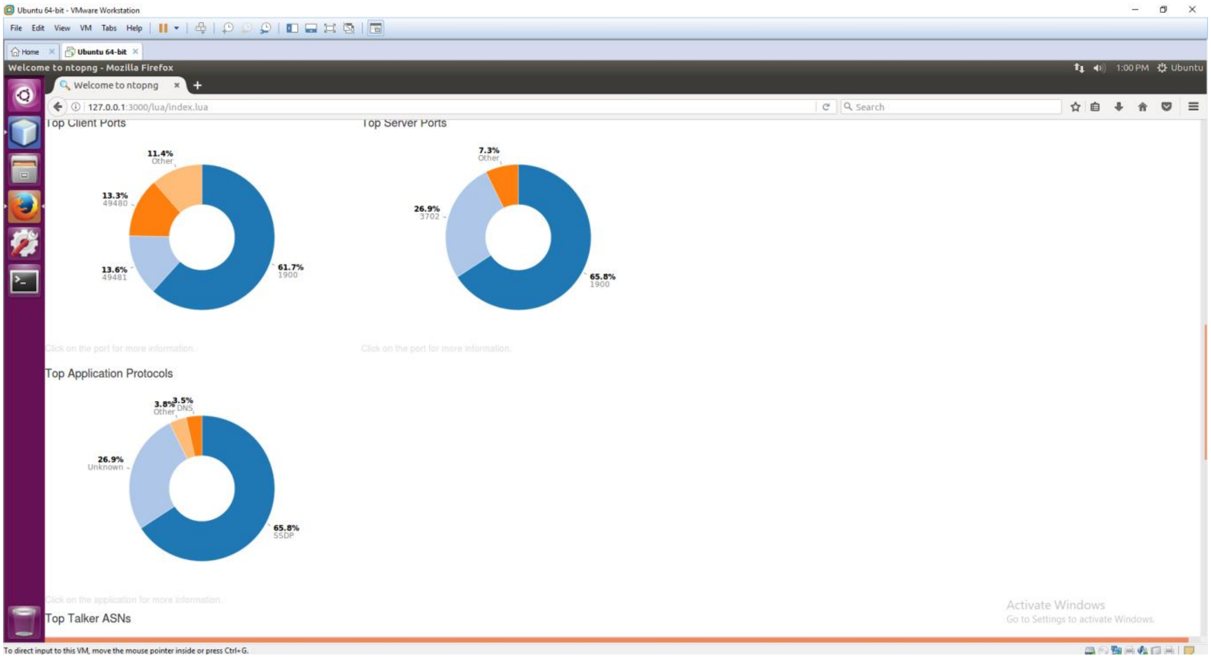
Original ntopng community dashboard view


Ubuntu 64-bit - VMware Workstation
File Edit View VM Tabs Help
Home
Ubuntu 64-bit
Welcome to ntopng - Mozilla Firefox
1:00 PM Ubuntu
Welcome to ntopng
127.0.0.1:3000/lua/index.lua
Search
ntop
Flows
Hosts
Devices
Interfaces
Search Host
Summary Report
Top Flows
h130192f.student.adfa.edu.a...
h2192f.student.adfa.edu.a...
91.189.89.198
Top Hosts (Send+Receive)
5.1%
Other
9.9%
117.18.237.29
Activate Windows
Go to Settings to activate Windows.
127.0.0.1:3000/lua/index.lua#
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Summary Report

Welcome to the summary report page we have modified in ntopng so that the user can see application and device information in one easy to navigate view

New summary view page


Ubuntu 64-bit - VMware Workstation
File Edit View VM Tabs Help
Home
Ubuntu 64-bit
Welcome to ntopng - Mozilla Firefox
1:00 PM Ubuntu
Welcome to ntopng
127.0.0.1:3000/lua/index.lua
Search
Top Client Ports
11.4%
Other
13.3%
49480
13.6%
49481
61.7%
1900
Click on the port for more information.
Top Server Ports
7.3%
Other
26.9%
3702
65.8%
1900
Click on the port for more information.
Top Application Protocols
3.8%
Other
3.5%
DNS
26.9%
Unknown
65.8%
SSDP
Click on the application for more information.
Top Talker ASNs
Activate Windows
Go to Settings to activate Windows.
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

3.1 Device and Application Information

Further detailed device information (MAC addresses, time on net, GeoIP, etc) and Application information (protocols, ports) can be found from the summary view by clicking on the written text under the pie chart statistic.

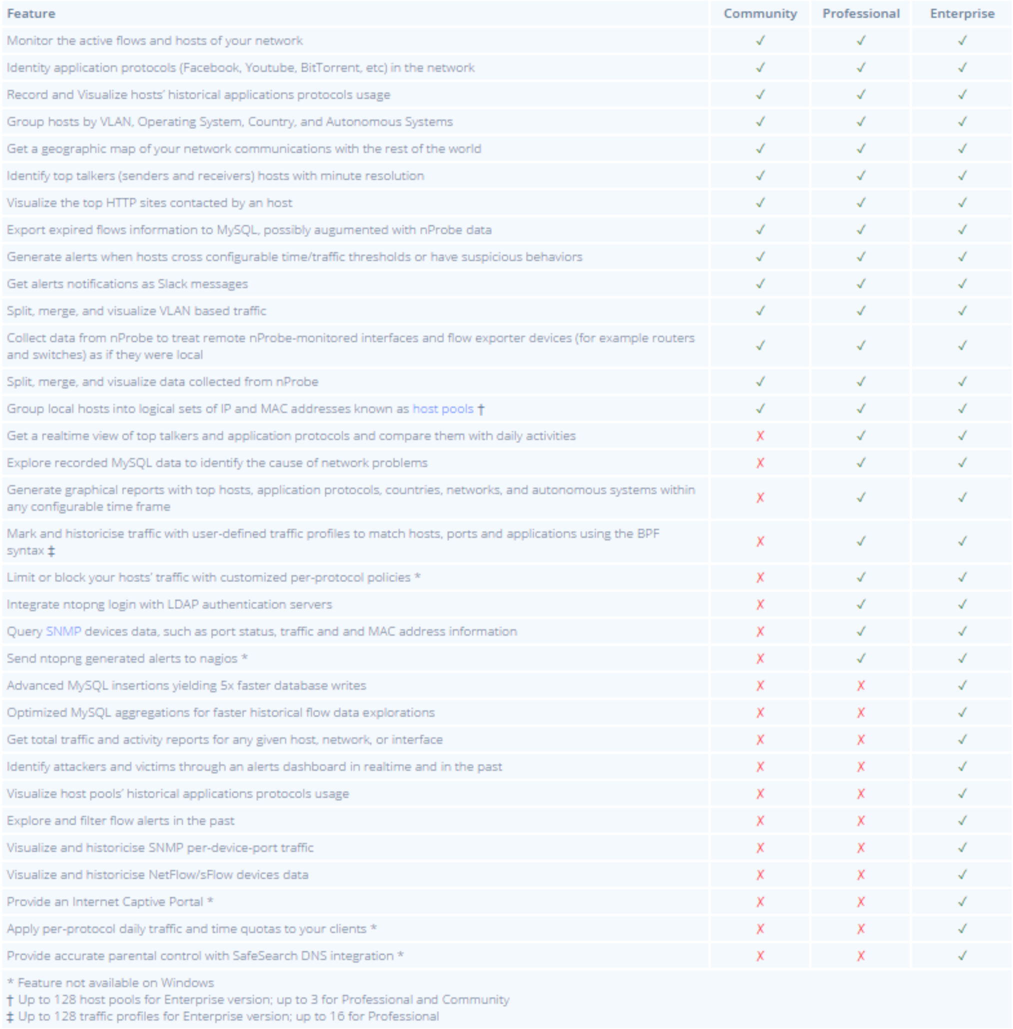
During the project we did a bit of research into the relationship between applications, protocols and ports and we found that directly linking ports to an application is not always true or reliable. nDPI is continually extended by ntop to support more protocols, and this should be updated as per the maintenance sections we have detailed at the end of this document.

3.2 Professional 'Dashboard' Version of ntopng

It is worth exploring the feasibility of purchasing the professional or enterprise version of ntopng, as it features a different dashboard page to generate custom reports, and display/filter more in depth information of real-time traffic on user specified interfaces and application protocols.

Similar to what we have done of turning the community version dashboard into a summary page, the professional dashboard could also be modified to suit the unit's needs once purchased.

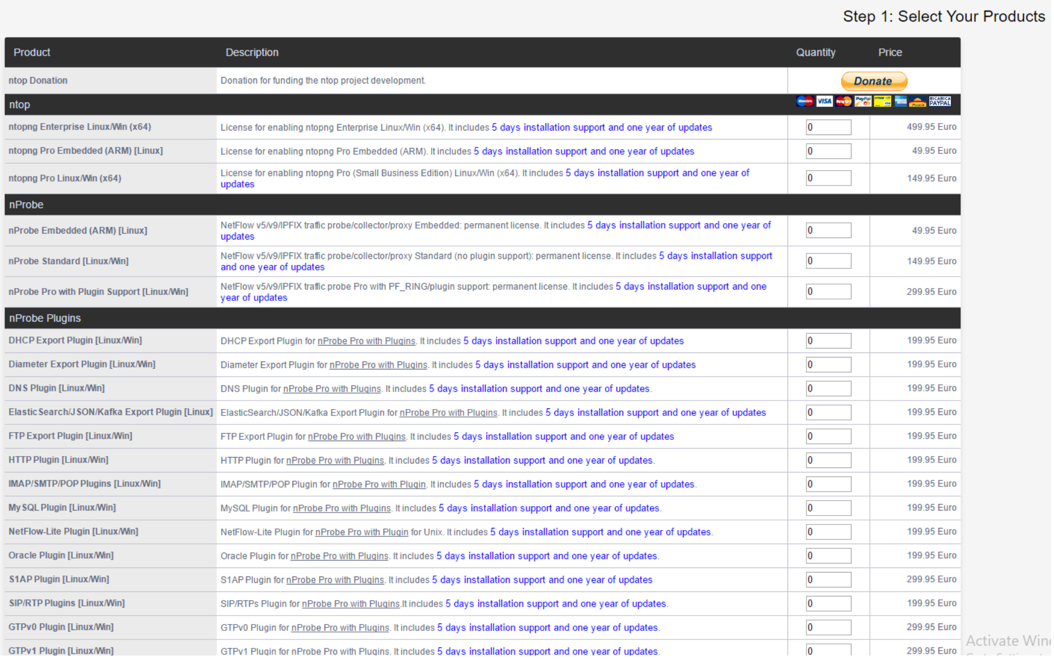
The figure below is taken from ntop.org's website and provides a comparison of the different features available across ntop community, professional, and enterprise editions. It can be noted that although our project has tried to provide our client with the best summary view of the network, it is somewhat limited by the features of the community version.



Feature	Community	Professional	Enterprise
Monitor the active flows and hosts of your network	✓	✓	✓
Identity application protocols (Facebook, Youtube, BitTorrent, etc) in the network	✓	✓	✓
Record and Visualize hosts' historical applications protocols usage	✓	✓	✓
Group hosts by VLAN, Operating System, Country, and Autonomous Systems	✓	✓	✓
Get a geographic map of your network communications with the rest of the world	✓	✓	✓
Identify top talkers (senders and receivers) hosts with minute resolution	✓	✓	✓
Visualize the top HTTP sites contacted by an host	✓	✓	✓
Export expired flows information to MySQL, possibly augumented with nProbe data	✓	✓	✓
Generate alerts when hosts cross configurable time/traffic thresholds or have suspicious behaviors	✓	✓	✓
Get alerts notifications as Slack messages	✓	✓	✓
Split, merge, and visualize VLAN based traffic	✓	✓	✓
Collect data from nProbe to treat remote nProbe-monitored interfaces and flow exporter devices (for example routers and switches) as if they were local	✓	✓	✓
Split, merge, and visualize data collected from nProbe	✓	✓	✓
Group local hosts into logical sets of IP and MAC addresses known as host pools †	✓	✓	✓
Get a realtime view of top talkers and application protocols and compare them with daily activities	✗	✓	✓
Explore recorded MySQL data to identify the cause of network problems	✗	✓	✓
Generate graphical reports with top hosts, application protocols, countries, networks, and autonomous systems within any configurable time frame	✗	✓	✓
Mark and historicise traffic with user-defined traffic profiles to match hosts, ports and applications using the BPF syntax ‡	✗	✓	✓
Limit or block your hosts' traffic with customized per-protocol policies *	✗	✓	✓
Integrate ntopng login with LDAP authentication servers	✗	✓	✓
Query SNMP devices data, such as port status, traffic and and MAC address information	✗	✓	✓
Send ntopng generated alerts to nagios *	✗	✓	✓
Advanced MySQL insertions yielding 5x faster database writes	✗	✗	✓
Optimized MySQL aggregations for faster historical flow data explorations	✗	✗	✓
Get total traffic and activity reports for any given host, network, or interface	✗	✗	✓
Identify attackers and victims through an alerts dashboard in realtime and in the past	✗	✗	✓
Visualize host pools' historical applications protocols usage	✗	✗	✓
Explore and filter flow alerts in the past	✗	✗	✓
Visualize and historicise SNMP per-device-port traffic	✗	✗	✓
Visualize and historicise NetFlow/sFlow devices data	✗	✗	✓
Provide an Internet Captive Portal *	✗	✗	✓
Apply per-protocol daily traffic and time quotas to your clients *	✗	✗	✓
Provide accurate parental control with SafeSearch DNS integration *	✗	✗	✓

* Feature not available on Windows
† Up to 128 host pools for Enterprise version; up to 3 for Professional and Community
‡ Up to 128 traffic profiles for Enterprise version; up to 16 for Professional

The current prices and options for ntopng can be found from their shop at https://shop.ntop.org/cart.php, and the below image gives an example of what is available:



Step 1: Select Your Products

Product	Description	Quantity	Price
ntop Donation	Donation for funding the ntop project development.	Donate	
ntop			
ntopng Enterprise Linux/Win (x64)	License for enabling ntopng Enterprise Linux/Win (x64). It includes 5 days installation support and one year of updates	0	499.95 Euro
ntopng Pro Embedded (ARM) [Linux]	License for enabling ntopng Pro Embedded (ARM). It includes 5 days installation support and one year of updates	0	49.95 Euro
ntopng Pro Linux/Win (x64)	License for enabling ntopng Pro (Small Business Edition) Linux/Win (x64). It includes 5 days installation support and one year of updates	0	149.95 Euro
nProbe			
nProbe Embedded (ARM) [Linux]	NetFlow v5/v9/IPFIX traffic probe/collector/proxy Embedded: permanent license. It includes 5 days installation support and one year of updates	0	49.95 Euro
nProbe Standard [Linux/Win]	NetFlow v5/v9/IPFIX traffic probe/collector/proxy Standard (no plugin support): permanent license. It includes 5 days installation support and one year of updates	0	149.95 Euro
nProbe Pro with Plugin Support [Linux/Win]	NetFlow v5/v9/IPFIX traffic probe Pro with PF_RING/plugin support: permanent license. It includes 5 days installation support and one year of updates	0	299.95 Euro
nProbe Plugins			
DHCP Export Plugin [Linux/Win]	DHCP Export Plugin for nProbe Pro with Plugins. It includes 5 days installation support and one year of updates	0	199.95 Euro
Diameter Export Plugin [Linux/Win]	Diameter Export Plugin for nProbe Pro with Plugins. It includes 5 days installation support and one year of updates	0	199.95 Euro
DNS Plugin [Linux/Win]	DNS Plugin for nProbe Pro with Plugins. It includes 5 days installation support and one year of updates.	0	199.95 Euro
ElasticSearch/JSON/Kafka Export Plugin [Linux]	ElasticSearch/JSON/Kafka Export Plugin for nProbe Pro with Plugins. It includes 5 days installation support and one year of updates	0	199.95 Euro
FTP Export Plugin [Linux/Win]	FTP Export Plugin for nProbe Pro with Plugins. It includes 5 days installation support and one year of updates	0	199.95 Euro
HTTP Plugin [Linux/Win]	HTTP Plugin for nProbe Pro with Plugins. It includes 5 days installation support and one year of updates.	0	199.95 Euro
IMAP/SMTP/POP Plugins [Linux/Win]	IMAP/SMTP/POP Plugin for nProbe Pro with Plugin. It includes 5 days installation support and one year of updates.	0	199.95 Euro
MySQL Plugin [Linux/Win]	MySQL Plugin for nProbe Pro with Plugins. It includes 5 days installation support and one year of updates.	0	199.95 Euro
NetFlow-Lite Plugin [Linux/Win]	NetFlow-Lite Plugin for nProbe Pro with Plugin for Unix. It includes 5 days installation support and one year of updates.	0	199.95 Euro
Oracle Plugin [Linux/Win]	Oracle Plugin for nProbe Pro with Plugins. It includes 5 days installation support and one year of updates.	0	199.95 Euro
S1AP Plugin [Linux/Win]	S1AP Plugin for nProbe Pro with Plugins. It includes 5 days installation support and one year of updates	0	299.95 Euro
SIP/RTP Plugins [Linux/Win]	SIP/RTPs Plugin for nProbe Pro with Plugins.It includes 5 days installation support and one year of updates.	0	199.95 Euro
GTPv0 Plugin [Linux/Win]	GTPv0 Plugin for nProbe Pro with Plugins. It includes 5 days installation support and one year of updates.	0	299.95 Euro
GTPv1 Plugin [Linux/Win]	GTPv1 Plugin for nProbe Pro with Plugins. It includes 5 days installation support and one year of updates.	0	299.95 Euro

4. Overview of the Interface Points for developers

For the further development of the code it's necessary to identify the source code packages and the main interface points. The ones which we detail here are ones which we have identified and come to understand during the project time and are only a small part of the entire ntop system.

Ntopng is fully scriptable using the scripts engines. This means that via Lua the user can extract the monitoring information in HTML pages or export it as a third party applications. The ntopng Lua API consists of two main classes and is fairly simple. Scripting has been used for the interaction with the Web GUI, which is more efficient and fast as compared to any other languages. Generally you build up an interface to allow some flexibility with the scripting language of your choice so you can use the scripting language in a series of scenarios.

The embedded HTTP(S) web server is responsible for parsing GET/POST parameters (if any) and places them into the _GET global scripts variable so that they can be accessed from Lua. Scripts are also used in ntopng to execute periodic actions such as dumping throughput and top talkers to disk. In ntopng multiple scripts can be executed simultaneously simultaneously.

The good thing about scripting languages like Lua is they are interpreted instead of compiled so the user can modify the scripts without the need of compilation. Lua commands only get called when the C program wants to execute it. As such, the code will only be interpreted when it is called. Lua scripts executes separately like as a bash script that executes separately from the main program.

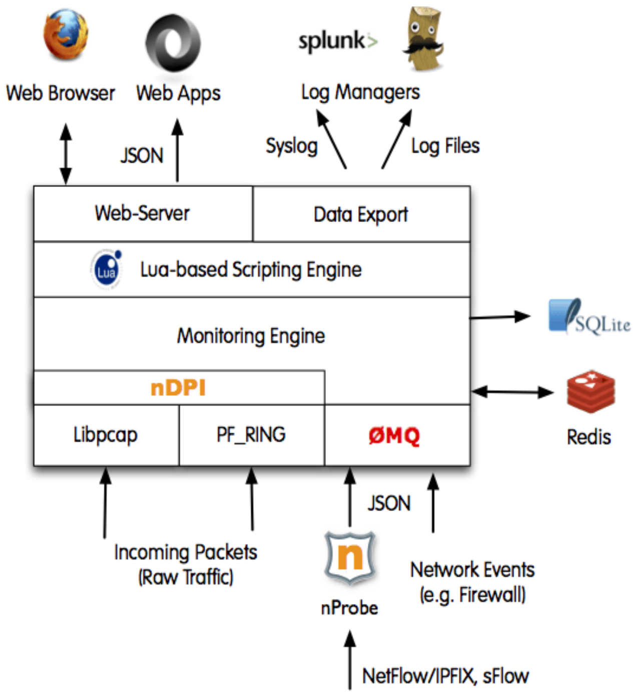
The figure below will explain how the different layers of the application work together and how they interact.

From the ntopng guide owned by ntop, the ntopng architecture is described as being divided into three main layers:

Ingress layer (flow or packet capture)

Monitoring and Lua based scripting engines: the ntopng core

Data export layer (via web, syslog or log files)



The web GUI design repositories are located under the file ntopng/httpdocs, and the scripts are running from the scripts folder also in the main ntopng repository.

The design of the webpage has been done through css, the content has been added through HTML, and javascript along with Lua has been used for the interaction with GUI. The source code for these has been located in the following directories:

- ntopng/httpdocs/css
- ntopng/httpdocs/inc
- ntopng/httpdocs/js

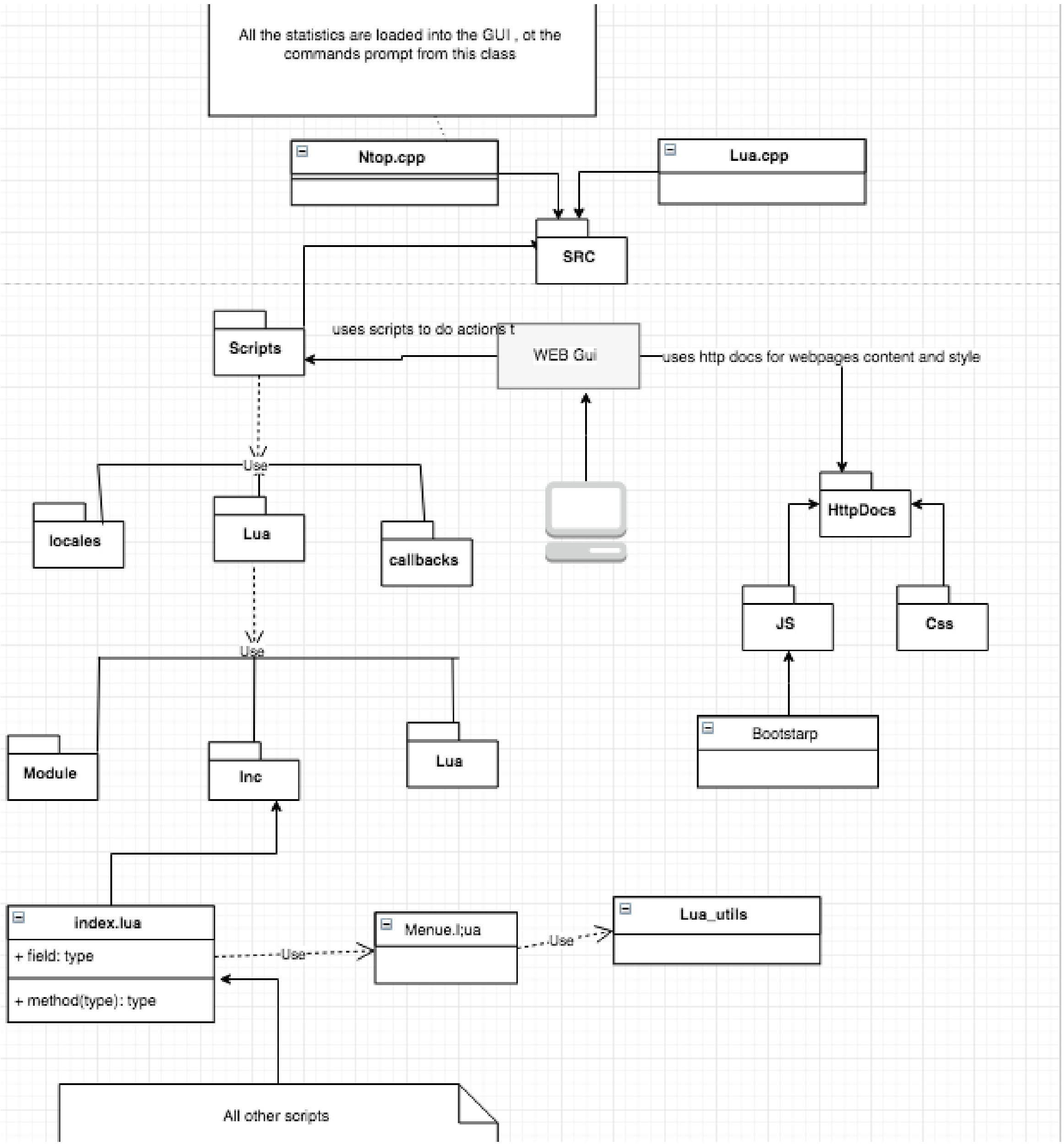
All the statistics of the the analysis, and live capturing are displayed using the ntop.cpp class located in in ntopng/src/ntop.cpp, which loads the all the details of the ip address, hosts, devices, and other relevant details.

All the scripting has been done through Lua, and it will be worth mentioning for further development of the Web Interface, that how it can be done. There is a class in the main source code of the directory under the name lua.cpp, this file act as an intersection point between the c code and the Lua scripts. So once the interface has loaded, the interaction of the user occurs through the files located in scripts.

The other classes that extract the data and show it as in the home page are menu.lua and index.lua, both of which uses lua_utill.lua. Most of the information about the data captured and analysis has been extracted and are loaded in our interface using these classes.

As mentioned in the summary view section, the index.lua file has been modified to contain all the Top Scripts from the httpdocs package (eg index_TopHosts.inc) so that these are no longer required and the index.lua file can be easily modified to contain more network traffic information. This is the default page that the running ntopng server loads too (127.0.0.1/3000)

The figure below below is overview of the overall content of the web GUI and the summary page.



5. Installation of ntopng

The following steps will detail the relevant information for installation of our ntopng version onto Raspberry Pi with Linux/Ubuntu, and the required nDPI libraries. ntop support for windows and further documentation in Linux and MacOSX can be found in the projects documents at

- /ntopng/doc/UserGuide.pdf
- /ntopng/doc/README.windows

Or on the ntop website at:http://www.ntop.org/support/documentation/documentation/

5.1. On Raspberry Pi with Ubuntu and Netbeans

We chose Raspberry Pi with Linux as our primary platform for this project to demonstrate how a small tool placed as an access point could analyse and control the large amounts of network traffic. ntopng can run on cross platforms (Unix, MacOSX, Windows), however it has a larger amount of support from the Linux community.

For this project we tested ntopng on a Raspberry Pi Zero W and a Raspberry Pi 2 model B. We found that the raspberry pi zero did not have the capabilities/processing power required to run ntopng efficiently, but the Raspberry Pi 2 made for a great tool (more details in Annex A - Testing Plan).

To start with working on the code we used Ubuntu Mate (16.04.2 LTS), which has been made for a generic aarch32 (AMRv7) based system like the raspberry pi, in a virtual machine and installed the Netbeans IDE.

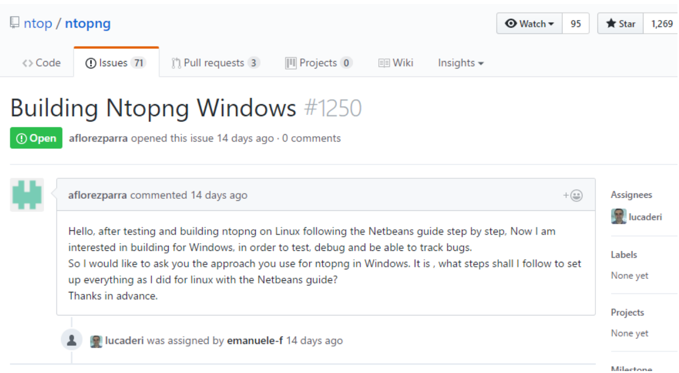
We forked the ntop/ntopng github repository (https://github.com/ntop/ntopng) into our own (https://github.com/madizer13/ntopng) for building, testing and debugging and to collaborate on the project on the default branch dev. From here it could then be cloned onto the Pi for installation or further use with Netbeans.

The supporting documentation provided by ntop can be followed to achieve this and the only changes necessary is using the Github repo link ‘https://github.com/madizer13/ntopng’ instead of that owned by ntop.

- ntopng/doc/README.compilation
- ntopng/doc/README.netbeans.pdf
- ntopng/doc/README.raspberry

Our GUI Wrapper also requires Java, so Java must be installed onto the Raspberry Pi. This is relatively simple and the guides are on the Java Oracle website. The guide we used for this was: http://www.rpiblog.com/2014/03/installing-oracle-jdk-8-on-raspberry-pi.html

The only problems we encountered were very much to do with not understanding the compilation process and not having any previous experience using virtual machines, however it is recommended to test, build and debug the code on Linux (i.e. a virtual machine if necessary) because of the support on the platform. Many issues like the one to the right have been opened in ntopng for building on



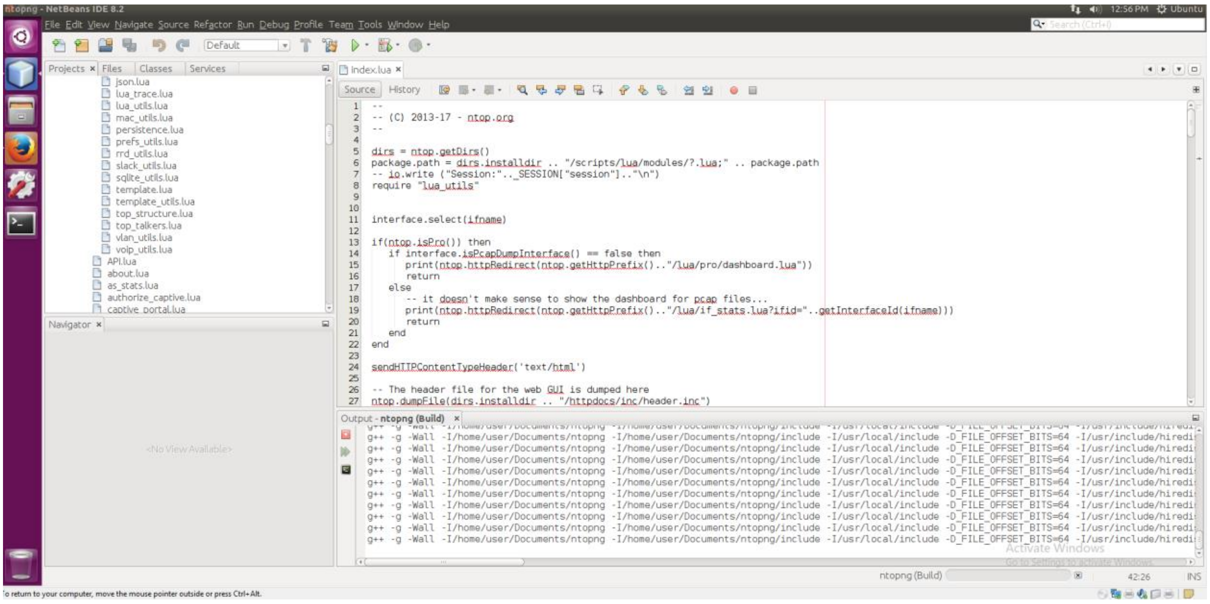
Building Ntopng Windows #1250

aflorezparra opened this issue 14 days ago · 0 comments

aflorezparra commented 14 days ago

Hello, after testing and building ntopng on Linux following the Netbeans guide step by step, Now I am interested in building for Windows, in order to test, debug and be able to track bugs.
So I would like to ask you the approach you use for ntopng in Windows. It is , what steps shall I follow to set up everything as I did for linux with the Netbeans guide?
Thanks in advance.

lucaderi was assigned by emanuele-f 14 days ago windows, so if you are interested this issue could be followed for a solution.



```lua
--
-- (C) 2013-17 - ntop.org
--

dirs = ntop.getDirs()
package.path = dirs.installdir .. "/scripts/lua/modules/?.lua;" .. package.path
-- io.write ("Session:".._SESSION["session"].."\n")
require "lua_utils"


interface.select(ifname)

if(ntop.isPro()) then
   if interface.isPcapDumpInterface() == false then
      print(ntop.httpRedirect(ntop.getHttpPrefix().."/lua/pro/dashboard.lua"))
      return
   else
      -- it doesn't make sense to show the dashboard for pcap files...
      print(ntop.httpRedirect(ntop.getHttpPrefix().."/lua/if_stats.lua?ifid="..getInterfaceId(ifname)))
      return
   end
end

sendHTTPContentTypeHeader('text/html')

-- The header file for the web GUI is dumped here
ntop.dumpFile(dirs.installdir .. "/httpdocs/inc/header.inc")
```

```
g++ -g -Wall -I/home/user/Documents/ntopng -I/home/user/Documents/ntopng/include -I/usr/local/include -D_FILE_OFFSET_BITS=64 -I/usr/include/hiredi
g++ -g -Wall -I/home/user/Documents/ntopng -I/home/user/Documents/ntopng/include -I/usr/local/include -D_FILE_OFFSET_BITS=64 -I/usr/include/hiredi
g++ -g -Wall -I/home/user/Documents/ntopng -I/home/user/Documents/ntopng/include -I/usr/local/include -D_FILE_OFFSET_BITS=64 -I/usr/include/hiredi
g++ -g -Wall -I/home/user/Documents/ntopng -I/home/user/Documents/ntopng/include -I/usr/local/include -D_FILE_OFFSET_BITS=64 -I/usr/include/hiredi
g++ -g -Wall -I/home/user/Documents/ntopng -I/home/user/Documents/ntopng/include -I/usr/local/include -D_FILE_OFFSET_BITS=64 -I/usr/include/hiredi
g++ -g -Wall -I/home/user/Documents/ntopng -I/home/user/Documents/ntopng/include -I/usr/local/include -D_FILE_OFFSET_BITS=64 -I/usr/include/hiredi
g++ -g -Wall -I/home/user/Documents/ntopng -I/home/user/Documents/ntopng/include -I/usr/local/include -D_FILE_OFFSET_BITS=64 -I/usr/include/hiredi
g++ -g -Wall -I/home/user/Documents/ntopng -I/home/user/Documents/ntopng/include -I/usr/local/include -D_FILE_OFFSET_BITS=64 -I/usr/include/hiredi
g++ -g -Wall -I/home/user/Documents/ntopng -I/home/user/Documents/ntopng/include -I/usr/local/include -D_FILE_OFFSET_BITS=64 -I/usr/include/hiredi
g++ -g -Wall -I/home/user/Documents/ntopng -I/home/user/Documents/ntopng/include -I/usr/local/include -D_FILE_OFFSET_BITS=64 -I/usr/include/hiredi
```

ntopng project open in Netbeans on our Ubuntu virtual machine.

6. Maintenance of ntopng

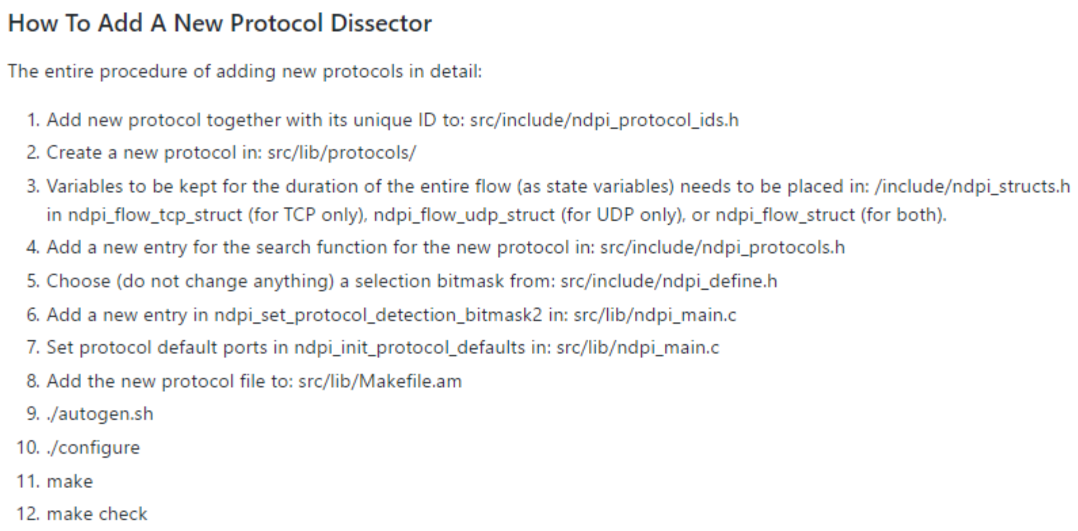
6.1 Non-Technical

nDPI is the deep packet inspection library maintained by ntop and has been released under the open source LGPLv3 license. It's application-layer detection of protocols must be maintained and updated so that new protocols on the network can be detected and analysed in ntopng.

More information of the protocols that are currently supported can be found on ntop's website at: http://www.ntop.org/products/deep-packet-inspection/ndpi/

Because we have not made any changes to the nDPI repository, the update is the same as per the repository maintained by ntop: https://github.com/ntop/nDPI. This should be routinely checked and updated on a schedule to ensure the latest protocols are supported.

If you want to add in your own protocols (e.g. known protocols from a local network) to nDPI to be able to see in ntopng traffic analysis, then the site also contains the instructions on how to add a new protocol as per the image below.



How To Add A New Protocol Dissector

The entire procedure of adding new protocols in detail:

1. Add new protocol together with its unique ID to: src/include/ndpi_protocol_ids.h
2. Create a new protocol in: src/lib/protocols/
3. Variables to be kept for the duration of the entire flow (as state variables) needs to be placed in: /include/ndpi_structs.h in ndpi_flow_tcp_struct (for TCP only), ndpi_flow_udp_struct (for UDP only), or ndpi_flow_struct (for both).
4. Add a new entry for the search function for the new protocol in: src/include/ndpi_protocols.h
5. Choose (do not change anything) a selection bitmask from: src/include/ndpi_define.h
6. Add a new entry in ndpi_set_protocol_detection_bitmask2 in: src/lib/ndpi_main.c
7. Set protocol default ports in ndpi_init_protocol_defaults in: src/lib/ndpi_main.c
8. Add the new protocol file to: src/lib/Makefile.am
9. ./autogen.sh
10. ./configure
11. make
12. make check

6.2 Technical

To be able to fully keep the ntopng source library up to date with fixes from bugs, the ntop community repo must be merged into ours. This requires knowing how to deconflict any changes in files from the base repo (ntop/ntopng) to our head fork so that we keep our changes in our code within our repository.

Instead we recommend that only the nDPI needs to be updated as per the section above, and the ntopng only needs to be updated, by manually copying across necessary files that don’t conflict, when a new version is released.

However, both these methods have not been tested yet due to the time constraints of the project. We assume that from here the code will be further developed to be more specific to the unit and bought in house under the appropriate licence.

7. Test Plan

Test Plan Identifier

This test plan is for our changes to ntopng version 2.5, a network traffic analysis tool. This is at the test stage level (IEEE 829) within our IT Project, and the testing phase went from the 1 June 17 to 6 June 17.

References

- Statement of Work (Client Contract) version 2
- User Guide

Test Items (Functions/Features)

- GUI Wrapper functions
- Summary Page with live capture and refresh of graphs
- The running of ntopng on a Raspberry Pi Device
- Structural UML component/package diagram

Software Risk Issues

The largest risk came from a change in the project when we decided to use ntopng as a tool that the client took too. This late change and the ability to learn a new package that has been in development for the past 19 years put us significantly at risk of not being able to value add for the client in the time frame for the IT Project course.

We also don't know many of the systems/interfaces that ntopng will be used with at the unit, thus the testing plan will not be able to reveal any defects in this area.

Test Plan

The below matrix describes the functionality being tested, what it was tested on and the results. All the results have been accepted by the project team and are to be handed over to the client for further approval at the business process level.

ID	Description	Manual Testing		Result
		Device	Virtual Machine	
1	Running ntopng This was the first test we had to conduct to decide if we would change to ntopng for our project, and it consisted of installing the source code onto a platform then compiling and running it	Y	Y	Tested on the Rasperry Pi, Ubuntu virtual machine and Windows 10. For windows we could only install from the prebuilt executable packages provided by ntop as the compilation and make files could not be done from the cmd prompt.
2	Opening as a pre-existing project in Netbeans	N	Y	Only opened in Virtual Machine to work on the code and packages. The virtual machine was set up to replicate the Raspberry Pi and the README.netbeans file was followed. This worked successfully and enabled us to easily be able to build/develop and test the code.

3	Test GUI Wrapper functions	N	Y	Whilst building the code for the GUI Wrapper it was tested iteratively to ensure the code had no defects. On final testing the GUI Wrapper worked fine with the latest version of Java in an Ubuntu Virtual Machine. The tests revealed that 'Browse PCAP' function did not check if it was a valid pcap file, and the 'Listen for Specific IP' did also not check if this IP address exists. Any of these functions could throw any error in the background that would cause the program to stop. As a short term fix we have put the technical knowledge for this into the user guide.
4	Set up Raspberry Pi to prepare for ntopng installation			First had problems installing the Ubuntu flavor of linux that we wanted, so we changed to Ubuntu Mate which is supported by the Raspberry Pi 2. Next we had to install a wifi dongle (EW-7811) driver to enable wireless. After failing many attempts to listen for wireless networks we found this solution to update the driver: https://askubuntu.com/questions/551522/netis-wf2120-wifi-adapter-drops-signal-within-seconds/551648#551648 The Raspberry Pi was then ready for the installation of ntopng and nDPI.
5	Install our ntopng version and nDPI onto the Raspberry Pi 2	Y	N	The README.compilation first had to be followed to install the necessary packages that ntopng depends on and all were still up to date for download onto the Pi. After changing the github clone link to ours (https://github.com/madizer13/ntopng) the ./autogen.sh and ./configure scripts ran fine and ntopng was made and installed successfully.
6	Run Graphical Wrapper on Raspberry Pi to start ntopng service.			As the GUI Wrapper already worked on java in the Ubuntu virtual machine, porting it across was easy and no functionality was lost. After placing it in the same file location as ntopng, it could write settings to the config file, start the service and automatically open the web browser GUI to the default summary page.
7	Navigate summary view in Rasperry Pi browser	Y	N	The Raspberry Pi used the firefox web browser, which ntop provides a large amount of support for, and it displayed all the scripts correctly. It integrated fine into the Raspberry Pi environment with an admin user.

Schedule

As per our the maintence in our ntopng user guide, we recommend that the ntop owned repository be followed for updates/fixes and then the testing plan should be redone with each new version or platform.

Further integration testing is also required at the client end for use with their in-house systems. The client may need to seek approval for ntopng to be installed onto these networks before commencing testing.