nDPI Wrapper

Massimo Puddu

July 2019

1 Introduction

The nDPI wrapper is divided into three files. The first file is “ndpi_wrapper.c”, which is mainly used to get macros, but it can also have some more useful and interesting applications. The second file is “ndpi_Reader_wrap.py”, which simply wraps the ndpiReader.c’s main function, making the output easy to access. The last of these important files is “ndpiWrapper_example.py”, which is used to easily manipulate the data of ndpiReader.c. Indeed, we can get the statistic struct and use it to elaborate data directly in python, and of course this data can then be used to call other nDPI functions. You can run this file with the same input that you would run ndpi_Reader_wrap.py.

2 What Struct do you get?

In order to avoid confusion, I have divided the pure nDPI structure from the structure in the example. The former is stored in “ndpi_typestruct.py”, while the example struct can be found in “ndpi_util_struct.py”. Unfortunately, the example struct cannot be separated from the nDPI struct, as most structs used in the example file use the structures defined in nDPI.

2.1 nDPI Struct

- timeval
- ndpi_protocol
- ndpi_ndpi_mask
- NDPI_PROTOCOL_BITMASK
- ndpi_subprotocol_conf_struct
- ndpi_automa
- ndpi_call_function_struct
- ndpi_proto_defaults_t

- ndpi_default_ports_tree_node_t
- spinlock_t
- atomic_t
- time_t
- hash_ip4p_node
- hash_ip4p
- hash_ip4p_table
- bt_announce
- ndpi_lru_cache
- cache_entry
- cache_entry_map
- cache
- custom_categories
- ndpi_detection_module_struct
- u6_addr
- ndpi_in6_addr
- ndpi_ip_addr_t
- ndpi_id_struct
- ndpi_flow_tcp_struct
- ndpi_flow_udp_struct
- l4
- http
- dns
- ntp
- ssl
- stun
- stun_ssl
- ssh

- mdns
- ubntac2
- http2 //this one is the one used inside protos
- bittorrent
- dhcp
- protos
- tinc_cache_entry
- struct_ndpi_int_one_line_struct
- struct_ndpi_iphdr
- struct_ndpi_ip6_hdrctl
- struct_ndpi_ipv6hdr
- struct_ndpi_tcphdr
- struct_ndpi_udphdr
- ndpi_packet_struct
- ndpi_flow_struct

2.2 nDPI/example Struct

- ndpi_stats
- ndpi_workflow_prefs
- ssh_ssl
- ndpi_flow_info
- ndpi_workflow

In particular, the file "ndpiWrapper_example.py" uses the struct reader_thread, which is mainly used to get the workflow from the execution.

3 Future expansion

Should you need additional functions or feel that something is missing, please follow the instructions in this section.

3.1 Macro

Since macros could change in the future, I decided that instead of creating a pair value in python it would be better to create a new function in “ndpi_wrapper.c”, but only if the macro is a easy value type: if the macro represents a piece of code, it would not make sense to do this.

3.2 Struct

If a struct you need is missing from the list I have provided, you can add it to the relevant file according to the description of the file, or you can of course create a new file. Just remember that the file “ndpi_typestruct.py” uses the shared library, because some macros were needed to make everything work. You have plenty of examples of how to do this, so don’t hesitate to try to improve and complete the wrapper if needed.

3.3 More data of ndpiReader

Since ndpiReader uses some global variables you may need to write some C code. I provided an example of this code in the “ndpi_wrapper.c” file, but you should always remember to set the global variables first before calling the function. All you need to do is create a new C function, pass the variable you create in python as input, and then call the function directly in C code.

4 Supported Funtion

Every function that uses the struct at point 2.1 is callable. Below, I’ll list some functions that in my opinion are important:

1. nDPI
   - ndpi_detection_module_struct* ndpi_init_detection_module
   - u_int8_t ndpi_get_api_version
   - void ndpi_process_extra_packet
   - void ndpi_process_extra_packet
   - ndpi_protocol ndpi_detection_process_packet
   - ndpi_get_flow_masterprotocol
   - and much more...
2. ndpi Reader
   - int main
   - int gettimeofday
   - void parseOptions
   - void testlib

5 Conclusion

Overall, I have enjoyed this experience and I hope that this wrapper will be useful to other students and that it will help them simplify their projects.