nDPI Encrypted Traffic Analysis

Luca Deri <deri@ntop.org>
@lucaderi

Who am I

- ntop founder (http://www.ntop.org): company that develops open-source network security and visibility tools including
  - ntopng: web-based traffic monitoring and security
  - nDPI: deep packet inspection toolkit
  - n2n: peer-to-peer VPN
- Intel Software Innovator
- Author of various open source software tools.
- Former member of nic.it

Monitoring Requirements

- Network administrators need to enforce network policies hence:
  - Limit the bandwidth of specific protocols (e.g. BitTorrent).
  - Block malicious communications that might travel over encrypted traffic channels.
  - Prioritise specific traffic protocols (e.g. WhatsApp/ Skype) or cloud protocols.
  - Traffic decryption is not an option for many reasons, in particular as it is useless in many reason while violating privacy.

What Do We Want to Accomplish?

- Fingerprint network traffic to detect if both the protocol (e.g. the certificate) has changed or its behaviour.
- Prevent specific traffic flows (e.g. unsafe TLS communications) to happen on our network.
- Provide metrics for measuring the nature of specific communications (e.g. HTTPS) while not being able to inspect the content.
- Identify malware in network communications.

What is Deep Packet Inspection?

- Technique that inspects the packet payload.
- Computationally intensive with respect to simple packet header analysis.
- Concerns about privacy and confidentiality of inspected data.
- Encryption is becoming pervasive, thus challenging DPI techniques.
- No false positives unless statistical methods or IP range/flow analysis are used by DPI tools.

Welcome to nDPI

- In 2012 we decided to develop our own GNU LGPL DPI toolkit (based on a unmaintained project named OpenDPI) in order to build an open DPI layer for ntop and third-party applications (Wireshark, netfilter, ML tools…).
- Protocols supported exceed 240 and include:
  - P2P (Skype, BitTorrent)
  - Messaging (Viber, Whatsapp, Telegram, Facebook)
  - Multimedia (YouTube, Last.gm, iTunes)
  - Conferencing (Webex, CitrixOnLine)
  - Streaming (Zattoo, Icecast, Shoutcast, Netflix)
  - Business (VNC, RDP, Citrix, Webex)

What is a Protocol in nDPI? [1/2]

- Each protocol is identified as <major>.<minor> protocol. Example:
  - DNS.Facebook
  - QUIC.YouTube and QUIC.YouTubeUpload
- Caveat: Skype or Facebook are protocols in the nDPI world but not for IETF.
- The first question people ask when they have to evaluate a DPI toolkit is: how many protocol do you support? This is not the right question.

What is a Protocol in nDPI? [2/2]

- Today most protocols are HTTP/TLS-based.
- nDPI includes support for string-based protocols detection:
  - DNS query name
  - HTTP Host/Server header fields
  - TLS/QUIC SNI (Server Name Indication)

- Example: NetFlix detection

```
{ "netflix.com", NULL,   "netflix" TLD,   "NetFlix",  NDPI_PROTOCOL_NETFLIX, NDPI_PROTOCOL_CATEGORY_STREAMING, NDPI_PROTOCOL_FUN },
{ "nflxext.com", NULL,   "nflxext" TLD,   "NetFlix",  NDPI_PROTOCOL_NETFLIX, NDPI_PROTOCOL_CATEGORY_STREAMING, NDPI_PROTOCOL_FUN },
{ "nflximg.com", NULL,   "nflximg" TLD,   "NetFlix",  NDPI_PROTOCOL_NETFLIX, NDPI_PROTOCOL_CATEGORY_STREAMING, NDPI_PROTOCOL_FUN },
{ "nflximg.net", NULL,   "nflximg" TLD,   "NetFlix",  NDPI_PROTOCOL_NETFLIX, NDPI_PROTOCOL_CATEGORY_STREAMING, NDPI_PROTOCOL_FUN },
{ "nflxvideo.net", NULL, "nflxvideo" TLD, "NetFlix",  NDPI_PROTOCOL_NETFLIX, NDPI_PROTOCOL_CATEGORY_STREAMING, NDPI_PROTOCOL_FUN },
{ "nflxso.net", NULL,    "nflxso" TLD,    "NetFlix",  NDPI_PROTOCOL_NETFLIX, NDPI_PROTOCOL_CATEGORY_STREAMING, NDPI_PROTOCOL_FUN },
```

Traffic Classification Lifecycle

- Based on traffic type (e.g. UDP traffic) dissectors are applied sequentially starting with the one that will most likely match the flow (e.g. for TCP/80 the HTTP dissector is tried first).
- Each flow maintains the state for non-matching dissectors in order to skip them in future iterations.
- Analysis lasts until a match is found or after too many attempts (8 packets is the upper-bound in our experience).

nDPI: Packet Processing Performance

```
nDPI Memory statistics:
     nDPI Memory (once):        203.62 KB
     Flow Memory (per flow):    2.01 KB
     Actual Memory:             95.60 MB
     Peak Memory:               95.60 MB
     Setup Time:                1001 msec
     Packet Processing Time:    813 msec

Traffic statistics:
     Ethernet bytes:          1090890957    (includes ethernet CRC/IFC/trailer)
     Discarded bytes:         247801
     IP packets:              1482145       of 1483237 packets total
     IP bytes:                1055319477    (avg pkt size 711 bytes)
     Unique flows:            36703
     TCP Packets:             1338624
     UDP Packets:             143521
     VLAN Packets:            0
     MPLS Packets:            0
     PPPoE Packets:           0
     Fragmented Packets:      1092
     Max Packet size:         1480
     Packet Len < 64:         590730
     Packet Len 64-128:       67824
     Packet Len 128-256:      66380
     Packet Len 256-1024:     157623
     Packet Len 1024-1500:    599588
     Packet Len > 1500:       0
     nDPI throughput:         1.82 M pps / 9.99 Gb/sec
     Analysis begin:          04/Aug/2010 04:15:23
     Analysis end:            04/Aug/2010 18:31:30
     Traffic throughput:      28.85 pps / 165.91 Kb/sec
     Traffic duration:        51367.223 sec
     Guessed flow protos:     0
```

← Single Core (E3 1241v3)

Behaviour and Fingerprinting

- nDPI is not only about application recognition but also:
  - Traffic classification: is this TLS connection a HTTPS connection, a VPN, or something else?
  - Malware recognition: traffic bins (time and packet size)
  - Content enforcement: bytes entropy (measure how bytes are distributed)

Server Entropy (SCP)

PDF	PNG	TEXT
6,418	7,014	7,008

nDPI Encrypted Traffic Analysis

- $ ./example/ndpiReader -J -i ./tests/pcap/instagram.pcap -v 2 -f "port 49355"

Behaviour

```
TCP 192.168.2.17:49355 <-> 31.13.86.52:443 [byte_dist_mean:
125.398474][byte_dist_std: 67.665465][entropy: 0.997011]
[total_entropy: 5609.185931][score: 1.0000][proto: 91.211/
TLS.Instagram][cat: SocialNetwork/6][456 pkts/33086 bytes <->
910 pkts/1277296 bytes][Goodput ratio: 9.0/95.3][14.29 sec]
[ALPN: http/1.1][TLS Supported Versions: TLSv1.3;TLSv1.3
(Fizz)][bytes ratio: -0.950 (Download)][IAT c2s/s2c min/avg/
max/stddev: 0/0 37.7/0.7 10107/274 546.6/11.8][Pkt Len c2s/s2c
min/avg/max/stddev: 66/66 72.6/1403.6 657/1454 57.2/231.0]
[TLSv1.3 (Fizz)][Client: scontent-mxp1-1.cdninstagram.com]
[JA3C: 7a29c223fb122ec64d10f0a159e07996][JA3S:
f4febc55ea12b31ae17cfb7e614afda8][Cipher:
TLS_AES_128_GCM_SHA256]
```

Fingerprint

Malware Analysis: Trickbot [1/2]

- See https://unit42.paloaltonetworks.com/wireshark-tutorial-examining-trickbot-infections/
- ndpiReader -J -v2 -i 2019-09-25-Trickbot-gtag-ono19-infection-traffic.pcap
- Many TLS flows on non-standard ports, self-signed certificate, same JA3

```
TCP 10.9.25.101:49184 <-> 187.58.56.26:449 [byte_dist_mean: 124.148883][byte_dist_std: 58.169660]
[entropy: 5.892724][total_entropy: 7124.302784][score: 0.9973][proto: 91/TLS][cat: Web/5][97 pkts/36053
bytes <-> 159 pkts/149429 bytes][Goodput ratio: 85/94][111.31 sec][bytes ratio: -0.611 (Download)][IAT
c2s/s2c min/avg/max/stddev: 0/0 1129/662 19127/19233 2990/2294][Pkt Len c2s/s2c min/avg/max/stddev:
54/54 372/940 1514/1514 530/631][Risk: ** Self-signed Certificate **** Obsolete TLS version (< 1.1) **]
[TLSv1][JA3S: 623de93db17d313345d7ea481e7443cf][Issuer: C=AU, ST=Some-State, O=Internet Widgits Pty Ltd]
[Subject: C=AU, ST=Some-State, O=Internet Widgits Pty Ltd][Certificate SHA-1: DD:EB:4A:36:6A:2B:50:DA:
5F:B5:DB:07:55:9A:92:B0:A3:52:5C:AD][Validity: 2019-07-23 10:32:39 - 2020-07-22 10:32:39][Cipher:
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA]

TCP 10.9.25.101:49165 <-> 144.91.69.195:80 [byte_dist_mean: 95.694525][byte_dist_std: 25.418150]
[entropy: 0.000000][total_entropy: 0.000000][score: 0.9943][proto: 7/HTTP][cat: Web/5][203 pkts/11127
bytes <-> 500 pkts/706336 bytes][Goodput ratio: 1/96][5.18 sec][Host: 144.91.69.195][bytes ratio: -0.969
(Download)][IAT c2s/s2c min/avg/max/stddev: 0/0 23/9 319/365 49/37][Pkt Len c2s/s2c min/avg/max/stddev:
54/54 55/1413 207/1514 11/134][URL: 144.91.69.195/solar.php[StatusCode: 200][ContentType: application/
octet-stream][UserAgent: pwtyyEKzNtGatwnJjmCcBLbOveCVpc][Risk: ** Binary application transfer **][PLAIN
TEXT (GET /solar.php HTTP/1.1)]
```

Malware Analysis: Trickbot [2/2]

```
00:08:02:1C:47:AE|20:E5:2A:B6:93:F1|0|0|0|10.9.25.101|5.53.125.13|49469|447|::|::|4|6|91|971|10|2732|11|
1589100502|1589100502|27|27|0.002|0.021|0.024|0|0|0|0|0|0||||||3,1,1,1,1,1|8,0,0,0,0,0|2.406|0.000|
95,-1419,134,-59,293,-773,-37,37,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
00:08:02:1C:47:AE|20:E5:2A:B6:93:F1|0|0|0|10.9.25.101|185.90.61.116|49482|447|::|::|4|6|91|931|9|2692|10|
1589100502|1589100502|27|27|0.002|0.025|0.026|0|0|0|0|0|0||||||3,1,1,1,1,1|8,0,0,0,0,0|2.406|0.000|
95,-1419,134,-59,293,-773,-37,37,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
00:08:02:1C:47:AE|20:E5:2A:B6:93:F1|0|0|0|10.9.25.101|195.123.221.104|49498|447|::|::|4|6|91|979|9|2692|10|
1589100502|1589100502|27|27|0.003|0.021|0.016|0|0|0|0|0|0||||||3,1,1,1,1,1|8,0,0,0,0,0|2.406|0.000|
127,-1419,134,-59,309,-773,-37,37,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
00:08:02:1C:47:AE|20:E5:2A:B6:93:F1|0|0|0|10.9.25.101|195.123.221.178|49515|447|::|::|4|6|91|947|9|2692|10|
1589100502|1589100502|27|27|0.002|0.024|0.026|0|0|0|0|0|0||||||3,1,1,1,1,1|8,0,0,0,0,0|2.406|0.000|
95,-1419,134,-59,309,-773,-37,37,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
```

- Same packet sequence, same packet len and time distribution (using bins to detect similarities), same entropy…

https://github.com/ntop/nDPI