Enhancing Suricata with Deep Packet Inspection

Luca Deri <deri@ntop.org>, @lucaderi
Alfredo Cardigliano <cardigliano@ntop.org>

SURICON MADRID 2024

ntop

About ntop

Pioneering software company (we recently celebrated 25 years) dedicated to providing innovative, high-performance network monitoring solutions. Founded with a vision to make network visibility accessible/possible, cheap, and efficient.

Our flagship products, including ntopng, nProbe, and PF_RING, are used globally by IT professionals, enterprises and telecommunications companies.

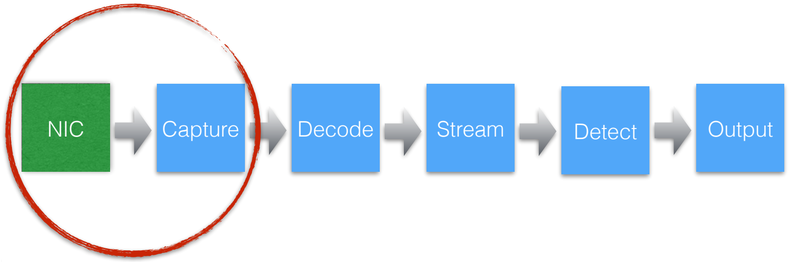
- Innovation: We continuously develop new technologies to keep up with the fast-evolving network landscape.

- Performance: Speed and reliability are central to all our solutions.

- Open-Source Commitment: We believe in collaboration and transparency as the path to powerful software solutions. Suricata and Wireshark are our favourite tools/communities but many other open source tools include our work (e.g. zmap/masscan, Moloch/Arkime, Ostinato, netsniff-ng, OpenWRT/LEDE, Softflowd, OpenMPTCPRouter).

ntop and Suricata

- Capture acceleration with PF_RING and ZC drivers
- L7 filtering with PF_RING FT
- Suricata EVE ingestion in ntopng



Actions	Begin	End	Duration	Protocol	Application	Score	Status	Flow	Info	Pkts	Bytes	Thpt
	17:33:40	17:33:44	00:04	TCP	SSH	100	External Alert	192.168.2.153 L :64084 192.168.2.134 L :ssh		1,185	116.51 KB	190.90

Motivation [1/4]

- Suricata supports (out of the box) ~20 protocols most of which are cleartext and RFC-based (e.g. NFS, DHCP…).

- Suricata rules for “modern protocols” such as Facebook become a nightmare to write (11+ domains) and increase rules number:

  pass tls any any → $EXTERNAL_NET 443 (tls.sni; dotprefix; content: “.facebook.com”; nocase; endswith; msg:“TLS Allowlisted access to facebook.com”; flow:to_server, established; sid:1; rev:1;)
  pass tls any any → $EXTERNAL_NET 443 (tls.sni; dotprefix; content: “.fbcdn.net”; nocase; endswith; msg:“TLS Allowlisted access to fbcdn.net”; flow:to_server, established; sid:1; rev:1;)

Motivation [2/4]

- “Dynamic” protocols such as Tor, crypto-miners etc are interesting in cybersecurity but are difficult to handle with signatures.
- Non trivial (i.e. a != b) detections such as Domain Generation Algorithm (DGA) or Punycode IDN is very complicated today, if possible at all.
- Traffic intelligence (e.g. non corporate VPNs, tunnelling, encrypted flows, gaming) is unlikely to be feasible with the current Suricata implementation.

Motivation [3/4]

- It could be very useful to trigger rules/alerts **out-of-the box** on:
  - Suspicious patterns (e.g. a flow with an unknown protocol that is probably encrypted).
  - Insecure/obsolete protocols/apps/ciphers versions.
  - Obfuscated, double-encrypted traffic.
  - Malware hosts (blacklists and JA3/JA4).
  - Traffic fingerprints.

Motivation [4/4]

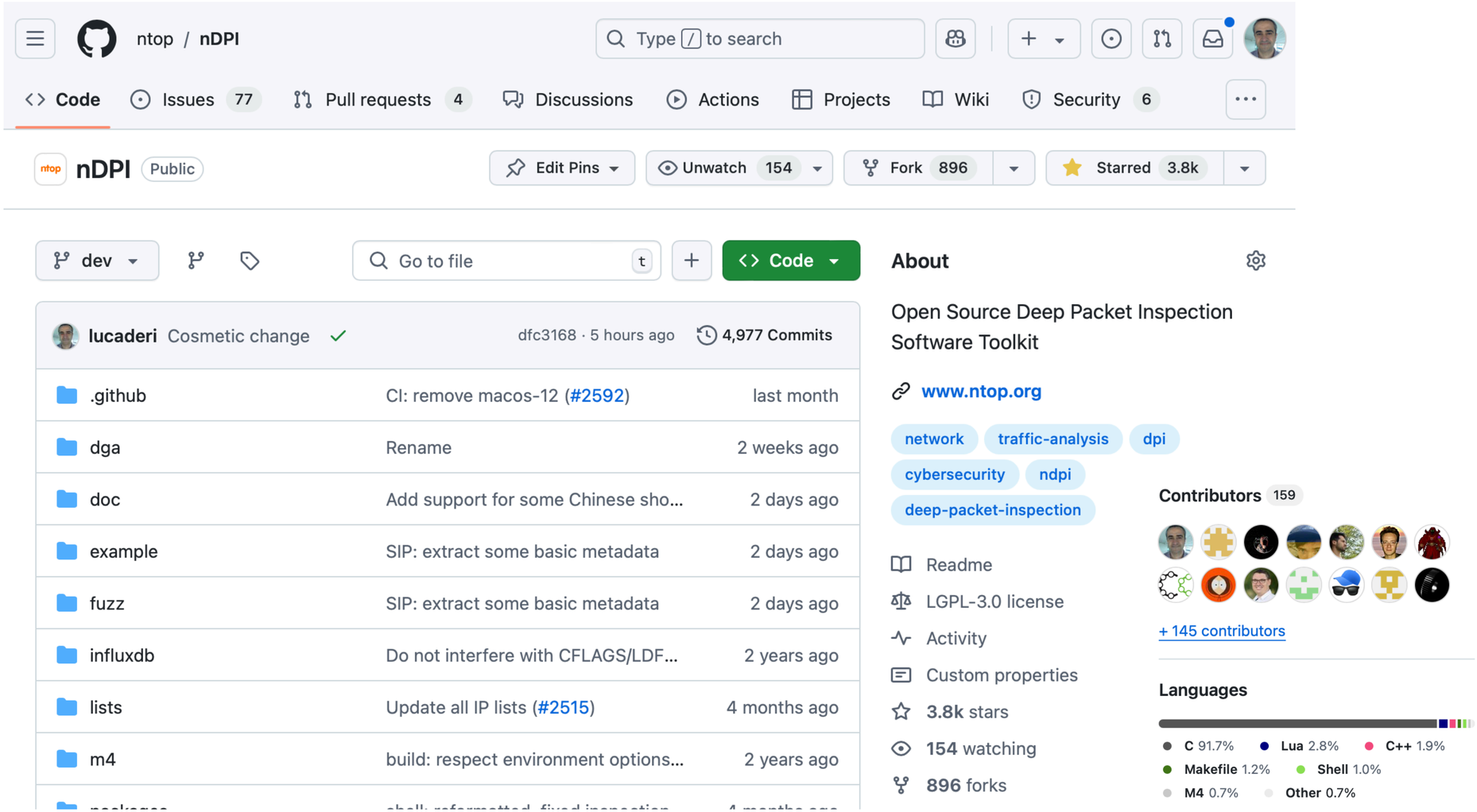
- In essence there are many good reasons to enhance traffic visibility in Suricata:
  - Simpler/more effective rules with DPI.
  - Complement Suricata **signature-based** engine with **behavioural traffic analysis**.
  - Better protocol visibility (from ~20 to ~500).
  - Enhance Suricata logs to implement detailed traffic visibility with respect to simple bytes/packet counters.

Previous Art

Nice but … it's proprietary code based on a licensed (and probably costly?) toolkit.

We want an open source and community developed/ maintained solution.

Welcome to nDPI [1/2]



https://github.com/ntop/nDPI

Welcome to nDPI [2/2]

- In 2012 we decided to develop our own GNU LGPL DPI toolkit in order to build an opensource DPI layer.
- Written in C, portable (ARM, Intel etc), fuzz-checked (thanks to Philippe Antoine of Catenacyber), many users (e.g. Sophos).
- Protocols supported exceed 430+ and include:
  - P2P (Skype, BitTorrent)
  - Messaging (Viber, Whatsapp, Telegram, Facebook)
  - Multimedia (YouTube, Last.gm, iTunes)
  - Conferencing (Webex, CitrixOnLine)
  - Streaming (Disney+, Hulu, Zattoo, Icecast, Shoutcast, Netflix)
  - Business (VNC, RDP, Citrix, Webex)
  - VPN (CiscoVPN, WireGuard, OpenVPN, Softether, NordVPN, Proton VPN…)
  - Gaming (WorldOfWarcraft, RiotGames, Nintendo, Playstation…
  - Mining (Ethereal, Bitcoin…)

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

- Each protocol is identified as <major>.<application> protocol. Example:
  - DNS.Facebook
  - QUIC.YouTube and QUIC.YouTubeUpload

- Caveat: Skype or Facebook are application 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 as:
  - you can define them via a configuration file.
  - better to ask how many categories nDPI supports.

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
  - SSL Certificate
  - 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 },
```

Custom Protocols in nDPI [1/2]

```
#  Subprotocols
#  Format:
#  host:"<value>",host:"<value>",.....@<subproto>

host:"disneyplus.com",host:"cdn.registerdisney.go.com",host:"disney-portal.my.onetrust.com",host:"disneyplus.bn5x.net",host:"disney-plus.net"@DisneyPlus
host:"*.lvlt.dash.us.aiv-cdn.net.c.footprint.net"@AmazonVideo
host:"api-global.netflix.com"@Netflix

ip:213.75.170.11/32:443@CustomProtocol
ip:8.248.73.247:443@AmazonPrime
ip:54.80.47.130@AmazonPrime

#You can specify a protocol Id. In that case you probably want to avoid conflict with internal ids.
#You can use any number up to 65535

ip:3.3.3.3:443@CustomProtocolA
ip:3.3.3.3:444@CustomProtocolB
ip:3.3.3.3:446@CustomProtocolC=800

ipv6:[3ffe:507:0:1:200:86ff:fe05:80da]@CustomProtocolD=1024
ipv6:[247f:855b:5e16:3caf::]/64:100@CustomProtocolE=2048
ipv6:[247f:855b:5e16:3caf::]/64@CustomProtocolF=2049
ipv6:[fe80::76ac:b9ff:fe6c:c124]:12717@CustomProtocolG=2050
ipv6:[fe80::76ac:b9ff:fe6c:c124]:12718@CustomProtocolH=65535
ipv6:[fe80::76ac:b9ff:fe6c:c124]:12719@CustomProtocolI=65534

#
# You can use symbolic IP addreses if you want
#
ip:www.ntop.org@ntop
ipv6:www.ntop.org@ntop
```

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: Flow Risks

- Built-in engine used to trigger meaningful (56 so far) events whenever a “potential risk” is found.

- HTTP suspicious user-agent
- HTTP numeric IP host contacted
- HTTP suspicious URL
- HTTP suspicious protocol header
- TLS connections not carrying HTTPS (e.g. a VPN over TLS)
- Suspicious DGA domain contacted
- Malformed packet
- SSH/SMB obsolete protocol or application version
- TLS suspicious ESNI usage
- Unsafe Protocol used
- Suspicious DNS traffic
- TLS with no SNI.
- XSS (Cross Site Scripting)
- SQL Injection
- Arbitrary Code Injection/Execution
- Binary/.exe application transfer (e.g. in HTTP)
- Known protocol on non standard port
- TLS self-signed certificate
- TLS obsolete version
- TLS weak cipher
- TLS certificate expired
- TLS certificate mismatch
- DNS suspicious traffic
- HTTP suspicious content

Encrypted Traffic Analysis

- Encryption is often perceived as a problem/limitation in particular in tools such as Suricata that are traditionally based on clear-text analysis.
- While this is not completely wrong, ETA (Encrypted Traffic Analysis) is an opportunity as
  - Protocol fingerprints are useful for detecting specific attacks.
  - nDPI has the ability to detect 'edge' uses of encryption such as double-encryption, obfuscated traffic etc that are definitively interesting for security analysers.
  - nDPI features **behavioural ETA** (i.e. not limited at the first few flow packets) for detecting misbehaving applications or for identifying similarities in traffic that is useful when comparing unknown traffic with know patterns.

nDPI Integration

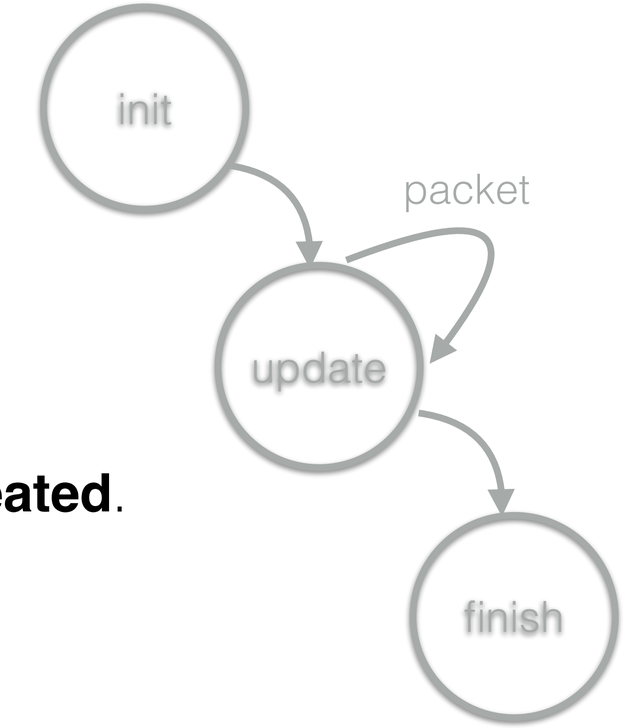
- First iteration:

(Proof of Concept initially implemented as a patch set adding hooks across the code)

nDPI as a Plugin

- Introduction of new callbacks in Suricata required by nDPI-like plugins:
  - Flow Callbacks (hooks in the flow life cycle)
  - Thread and Flow storage to keep nDPI data
  - EVE JSON Builder to extend the logs
- Thanks Jason for implementing all this!

Flow Callbacks



- New flow callbacks for nDPI-like plugins:
- `SCFlowRegisterInitCallback` (new)

  Register a callback to be called every time a flow is **created**.

- `SCFlowRegisterUpdateCallback` (new)

  Register a callback to be called every time a packet is seen on a flow and the flow is **updated**.

- `SCFlowRegisterFinishCallback` (new)

  Register a callback to be called when Suricata is **done** with a flow.

EVE JSON Builder

- New callback to extend the EVE JSON.
- *`SCEveRegisterCallback`* `(new)`

  Register a callback to be called for each EVE log to be produced.

  The callback is called just before closing the JSON object, for appending new fields from the plugin (nDPI metadata!).

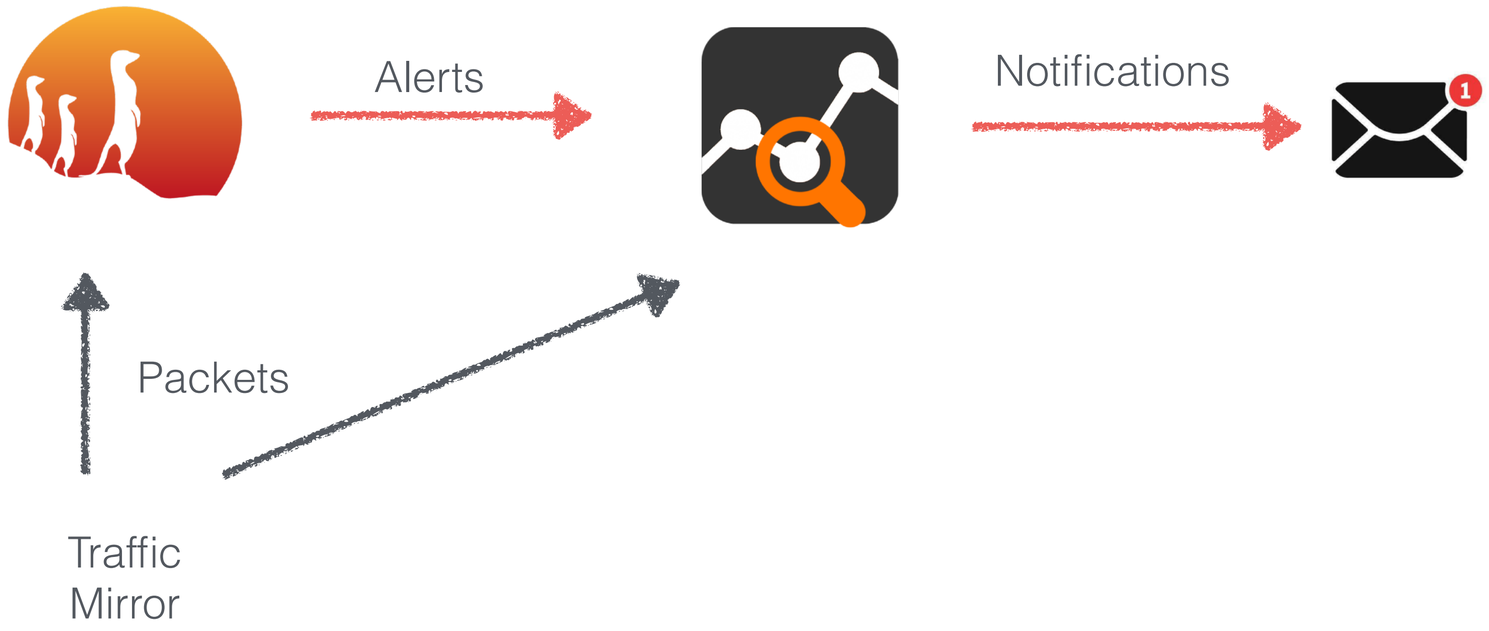
Enriching Traffic Information

- Augment EVE output with nDPI metadata

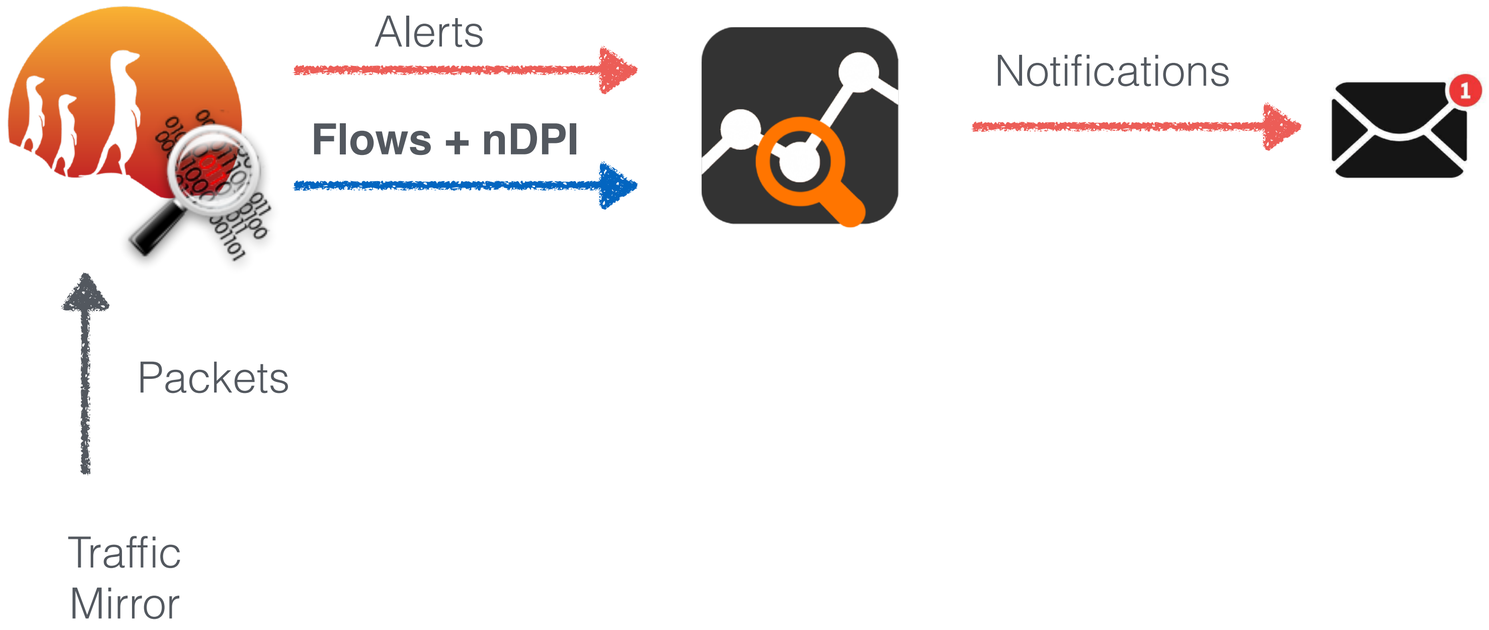

```
"proto": "TLS.FbookReelStory",
"proto_id": "91.337",
"category_id": 6,
"category": "SocialNetwork",
"proto_by_ip": "Facebook",
"proto_by_ip_id": 119,
"confidence": {
    "6": "DPI"
},
"breed": "Fun",
"encrypted": 1,
"hostname": "static.xx.fbcdn.net",
"tls": {
    "version": "TLSv1.3",
    "tls_supported_versions": "TLSv1.3,TLSv1.2",
    "ja3": "d39e1be3241d516b1f714bd47c2bc968",
    "ja3s": "475c9302dc42b2751db9edcac3b74891",
    "ja4": "t13d311100_e8f1e7e78f70_d41ae481755e",
    "cipher": "TLS_CHACHA20_POLY1305_SHA256",
    "unsafe_cipher": 0,
    "blocks": 0
},
"flow_risk": {
```

Merging Visibility and Security

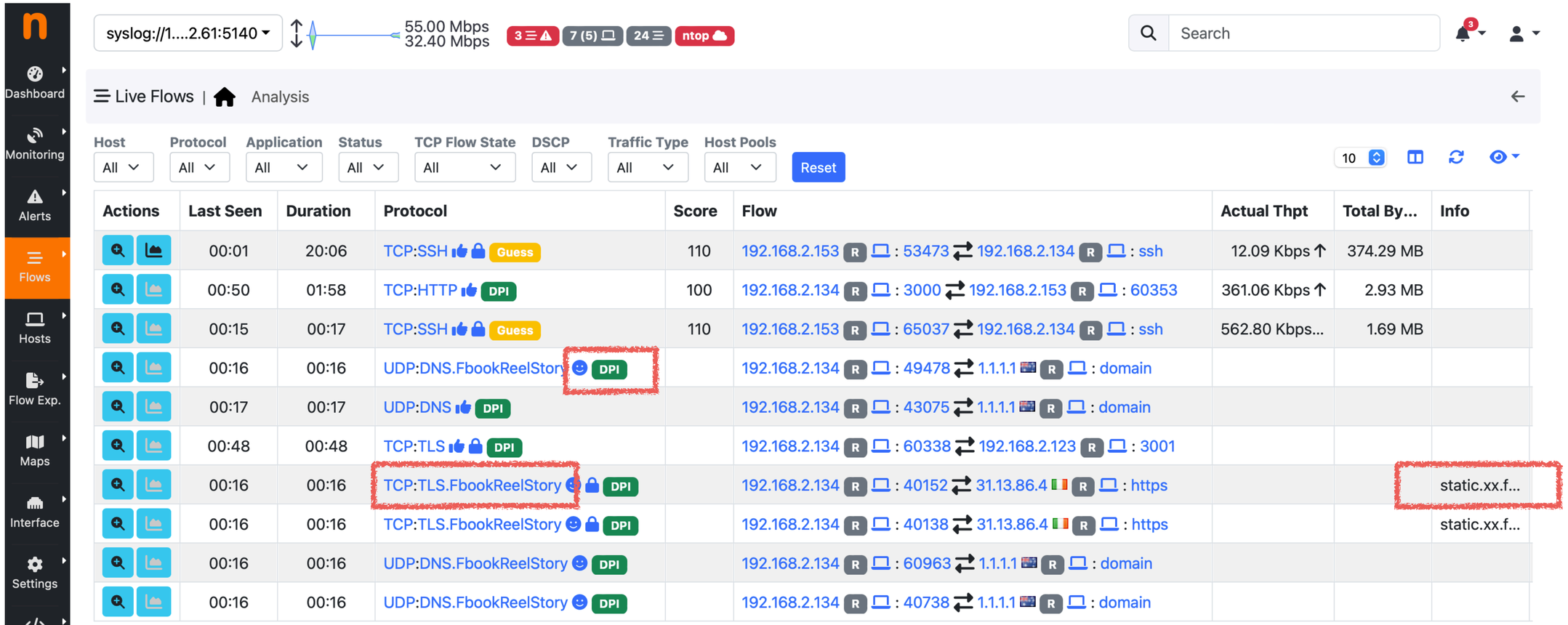
(presented at Suricon 2019)



Suricata as First-Class ntopng Sensor



Suricata as First-Class ntopng Sensor



Description	Score	Info / Remediation
External Alert ntopng	100	Detected HTTP-Susp-Entropy alert: Suspicious entropy detected on HTTP [NTOP]

L7-Powered Signatures

- New ***ndpi-protocol*** keyword to match traffic based on the application protocol.
- Syntax:
  - ndpi-protocol:[!]<protocol>;
- Protocol can be major, application, or major.application
- Example:
  - `alert tcp any any -> any any (msg:"NTOP DoT-DoH DNS over TLS or HTTPS detected"; ndpi-protocol: DoH_DoT; sid:1;)`

Application Protocols (Built-in)

FTP_CONTROL
POP3
SMTP
IMAP
DNS
IPP
HTTP
MDNS
NTP
NetBIOS
NFS
SSDP
BGP
SNMP
XDMCP
SMBv1
Syslog
DHCP
PostgreSQL
MySQL
Outlook
VK
POPS
Tailscale
Yandex
ntop
COAP
VMware
SMTPS
DTLS
UBNTAC2
BFCP
YandexMail
YandexMusic
Gnutella
eDonkey
BitTorrent
Skype_TeamsCall
Signal
Memcached
SMBv23
Mining
NestLogSink
Modbus
WhatsAppCall
DataSaver
Xbox
QQ
TikTok

RTSP
IMAPS
IceCast
CPHA
iQIYI
Zattoo
YandexMarket
YandexDisk
Discord
AdobeConnect
MongoDB
Pluralsight
YandexCloud
OCSP
VXLAN
IRC
MerakiCloud
Jabber
Nats
AmongUs
Yahoo
DisneyPlus
HART-IP
VRRP
Steam
HalfLife2
WorldOfWarcraft
Telnet
STUN
IPSec
GRE
ICMP
IGMP
EGP
SCTP
OSPF
IP_in_IP
RTP
RDP
VNC
Tumblr
TLS
SSH
Usenet
MGCP
IAX
TFTP
AFP
YandexMetrika

YandexDirect
SIP
TruPhone
ICMPV6
DHCPV6
Armagetron
Crossfire
Dofus
ADS_Analytic_Tra
ck
AdultContent
Guildwars
AmazonAlexa
Kerberos
LDAP
MapleStory
MsSQL-TDS
PPTP
Warcraft3
WorldOfKungFu
Slack
Facebook
Twitter
Dropbox
GMail
GoogleMaps
YouTube
Skype_Teams
Google
MS-RPCH
NetFlow
sFlow
HTTP_Connect
HTTP_Proxy
Citrix
NetFlix
LastFM
Waze
YouTubeUpload
Hulu
CHECKMK
AJP
Apple
Webex
WhatsApp
AppleiCloud
Viber
AppleiTunes
Radius

WindowsUpdate
TeamViewer
EthernetGlobalDa
ta
LotusNotes
SAP
GTP
WSD
LLMNR
TocaBoca
Spotify
FacebookMessen
ger
H323
OpenVPN
NOE
CiscoVPN
TeamSpeak
Tor
CiscoSkinny
RTCP
RSYNC
Oracle
Corba
UbuntuONE
Whois-DAS
SD-RTN
SOCKS
Nintendo
RTMP
FTP_DATA
Wikipedia
ZeroMQ
Amazon
eBay
CNN
Megaco
RESP
Pinterest
VHUA
Telegram
CoD_Mobile
Pandora
QUIC
Zoom
EAQ
Ookla
AMQP
KakaoTalk

KakaoTalk_Voice
Twitch
DoH_DoT
WeChat
MPEG_TS
Snapchat
Sina
GoogleMeet
IFLIX
Github
BJNP
Reddit
WireGuard
SMPP
DNScrypt
TINC
Deezer
Instagram
Microsoft
Starcraft
Teredo
HotspotShield
IMO
GoogleDrive
OCS
Microsoft365
Cloudflare
MS_OneDrive
MQTT
RX
AppleStore
OpenDNS
Git
DRDA
PlayStore
SOMEIP
FIX
Playstation
Pastebin
LinkedIn
SoundCloud
SteamDatagramR
elay
LISP
Diameter
ApplePush
GoogleServices
AmazonVideo
GoogleDocs

WhatsAppFiles
TargusDataspeed
DNP3
IEC60870
Bloomberg
CAPWAP
Zabbix
S7Comm
Teams
WebSocket
AnyDesk
SOAP
AppleSiri
SnapchatCall
HP_VIRTGRP
GenshinImpact
Activision
FortiClient
Z3950
Likee
GitLab
AVASTSecureDNS
Cassandra
AmazonAWS
Salesforce
Vimeo
FacebookVoip
SignalVoip
Fuze
GTP_U
GTP_C
GTP_PRIME
Alibaba
Crashlytics
Azure
iCloudPrivateRela
y
EthernetIP
Badoo
AccuWeather
GoogleClassroom
HSRP
Cybersec
GoogleCloud
Tencent
RakNet
Xiaomi
Edgecast
Cachefly

Softether
MpegDash
Dazn
GoTo
RSH
1kxun
PGM
IP_PIM
collectd
TunnelBear
CloudflareWarp
i3D
RiotGames
Psiphon
UltraSurf
Threema
AliCloud
AVAST
TiVoConnect
Kismet
FastCGI
FTPS
NAT-PMP
Syncthing
CryNetwork
Line
LineCall
AppleTVPlus
DirecTV
HBO
Vudu
Showtime
Dailymotion
Livestream
Tencentvideo
IHeartRadio
Tidal
TuneIn
SiriusXMRadio
Munin
Elasticsearch
TuyaLP
TPLINK_SHP
Source_Engine
BACnet
OICQ
Heroes_of_the_St
orm
FbookReelStory

SRTP
OperaVPN
EpicGames
GeForceNow
Nvidia
BITCOIN
ProtonVPN
Thrift
Roblox
Service_Location_
Protocol
Mullvad
HTTP2
HAProxy
RMCP
Controller_Area_N
etwork
Protobuf
ETHEREUM
TelegramVoip
SinaWeibo
TeslaServices
PTPv2
RTPS
OPC-UA
S7CommPlus
FINS
EtherSIO
UMAS
BeckhoffADS
ISO9506-1-MMS
IEEE-C37118
Ether-S-Bus
Monero
DCERPC
PROFINET_IO
HiSLIP
UFTP
OpenFlow
JSON-RPC
WebDAV
Kafka
NoMachine
IEC62056
HL7
Ceph
GoogleChat
Roughtime

PrivateInternetAc
cess
KCP
Dota2
Mumble
Yojimbo
ElectronicArts
STOMP
Radmin
Raft
CIP
Gearman
TencentGames
GaijinEntertainme
nt
ANSI_C1222
Huawei
HuaweiCloud
DLEP
BFD
NetEaseGames
PathofExile
GoogleCall
PFCP
FLUTE
LoLWildRift
TES_Online
LDP
KNXnet_IP
Bluesky
Mastodon
Threads
ViberVoip
ZUG
JRMI
RipeAtlas
HLS
ClickHouse
Nano
OpenWire
CNP-IP
ATG
TRDP
Lustre
NordVPN
Shein
Temu
Taobao
...

nDPI Flow Risks

- New ***ndpi-risk*** keyword to match nDPI flow risks
- Syntax:
  - ndpi-risk:[!]<risk>;
- Example:
  - `alert tcp any any -> any any (msg:"NTOP Binary-over-HTTP Binary application transfer over HTTP"; ndpi-protocol:HTTP; ndpi-risk: NDPI_BINARY_APPLICATION_TRANSFER; sid:1;)`

Risks

NDPI_URL_POSSIBLE_XSS
NDPI_URL_POSSIBLE_SQL_INJECTION
NDPI_URL_POSSIBLE_RCE_INJECTION
NDPI_BINARY_APPLICATION_TRANSFER
NDPI_KNOWN_PROTOCOL_ON_NON_STANDARD_PORT
NDPI_TLS_SELFSIGNED_CERTIFICATE
NDPI_TLS_OBSOLETE_VERSION
NDPI_TLS_WEAK_CIPHER
NDPI_TLS_CERTIFICATE_EXPIRED
NDPI_TLS_CERTIFICATE_MISMATCH
NDPI_HTTP_SUSPICIOUS_USER_AGENT
NDPI_NUMERIC_IP_HOST
NDPI_HTTP_SUSPICIOUS_URL
NDPI_HTTP_SUSPICIOUS_HEADER
NDPI_TLS_NOT_CARRYING_HTTPS
NDPI_SUSPICIOUS_DGA_DOMAIN
NDPI_MALFORMED_PACKET
NDPI_SSH_OBSOLETE_CLIENT_VERSION_OR_CIPHER
NDPI_SSH_OBSOLETE_SERVER_VERSION_OR_CIPHER
NDPI_SMB_INSECURE_VERSION

NDPI_TLS_SUSPICIOUS_ESNI_USAGE
NDPI_UNSAFE_PROTOCOL
NDPI_DNS_SUSPICIOUS_TRAFFIC
NDPI_TLS_MISSING_SNI
NDPI_HTTP_SUSPICIOUS_CONTENT
NDPI_RISKY_ASN
NDPI_RISKY_DOMAIN
NDPI_MALICIOUS_FINGERPRINT
NDPI_MALICIOUS_SHA1_CERTIFICATE
NDPI_DESKTOP_OR_FILE_SHARING_SESSION
NDPI_TLS_UNCOMMON_ALPN
NDPI_TLS_CERT_VALIDITY_TOO_LONG
NDPI_TLS_SUSPICIOUS_EXTENSION
NDPI_TLS_FATAL_ALERT
NDPI_SUSPICIOUS_ENTROPY
NDPI_CLEAR_TEXT_CREDENTIALS
NDPI_DNS_LARGE_PACKET
NDPI_DNS_FRAGMENTED
NDPI_INVALID_CHARACTERS
NDPI_POSSIBLE_EXPLOIT

NDPI_TLS_CERTIFICATE_ABOUT_TO_EXPIRE
NDPI_PUNYCODE_IDN
NDPI_ERROR_CODE_DETECTED
NDPI_HTTP_CRAWLER_BOT
NDPI_ANONYMOUS_SUBSCRIBER
NDPI_UNIDIRECTIONAL_TRAFFIC
NDPI_HTTP_OBSOLETE_SERVER
NDPI_MINOR_ISSUES
NDPI_TCP_ISSUES
NDPI_FULLY_ENCRYPTED
NDPI_TLS_ALPN_SNI_MISMATCH
NDPI_MALWARE_HOST_CONTACTED
NDPI_BINARY_DATA_TRANSFER
NDPI_PROBING_ATTEMPT
NDPI_OBFUSCATED_TRAFFIC

◉ detected on encrypted traffic

Wrapping Up

- The nDPI integration, besides augmenting metadata, simplifies and extends signatures with behaviour analysis, making threat detection even more effective.
- Low impact on memory footprint (max 1.16 KB/flow during detection) and performance (nDPI adds negligible additional CPU overhead being DPI limited to the first few flow packets).
- Code on Github
  - https://github.com/OISF/suricata/pull/12120
- Ticket on Redmine
  - https://redmine.openinfosecfoundation.org/issues/7231

Future Work

- Still a lot of work to be done:
  - More signature keywords (e.g. *ndpi-category:VPN*, *ndpi-breed:Unsafe*)
  - Support for writing signatures matching on any nDPI metadata and fingerprints (e.g. *ndpi-metadata:quic.unsafe_cipher=1*)
  - Be alerted when contacting a malware host, suspicious signature, invalid protocol (e.g. high entropy on ICMP/DNS traffic) or OS (e.g. a Windows host that contacts connectivitycheck.android.com) combination.
  - nDPI configuration for specifying nDPI settings and enabling/disabling features (add an ndpi section in the yaml?)

See You at ntopConf 2025

Zürich, May 7th-8th 2025 - https://www.ntop.org/ntopconf25/

SURICON MADRID2024

ntop