Suricata - Feature #511
Port independent protocol identification
07/26/2012 02:04 PM - David André

Status: New
Priority: Normal
Assignee: Community Ticket
Category: 
Target version: TBD
Effort: 
Difficulty: 

Label: 

Description
nDPI open-source GPL library allows to detect the protocol no matter what port is used.
It could be used to implement other Protocol keywords for suricata

http://www.ntop.org/products/ndpi/

Protocols supported according to nDPI documentation:
FTP POP SMTP IMAP DNS IPP HTTP MDNS NTP NETBIOS NFS SSDP BGP SNMP XDMCP SMB SYSLOG DHCP PostgreSQL MySQL TDS DirectDownloadLink I23V5 AppleJuice DirectConnect Socrates WinMX MANOLITO PANDO Filetopia iMESh Kontiki OpenFT Kazaa/Fasttrack Gnutella eDonkey Bittorrent (Extended) OFF AVI Flash OGG MPEG QuickTime RealMedia Windowsmedia MMS XBOX QQ MOVE RTSP Feidian Icecast PPLive PPSream Zattoo SHOUTCast SopCast TVAnts TVUser VeohTV QQLive Thunder/Webthunder Soulseek GaduGadu IRC Popo Jabber MSN Oscar Yahoo Battlefield Quake Second Life Steam Halflife2 World of Warcraft Teletnet STUN IPSEC GRE ICMP IGMP EGP SCTP OSPF IP in IP RTP RDP VNC PCAnywhere SSL SSH USENET MGCP IAX TFTP AIP StealthNet Aimini SIP Truphone ICMPv6 DHCPv6 Armagetron CrossFire Dofus Fiesta Florenzia Guildwars HTTP Application Activesync Kerberos LDAP MapleStory mSQL PPTP WARCRAFT3 World of Kung Fu MEEBO FaceBook Twitter DropBox Gmail Google Maps YouTube Skype Google DCE RPC NetFlow_iPFIX sFlow HTTP Connect (SSL over HTTP) HTTP Proxy Netflix Citrix CitrixOnline/GotoMeeting Apple (iMessage, FaceTime...) Webex WhatsApp Apple iCloud Viber Apple iTunes Radius

Related issues:
Related to Feature #2757: improve protocol detection

In Review

History
#1 - 08/06/2012 10:08 AM - Victor Julien
  - Target version set to TBD

OpenDPI/nDPI is licensed LGPLv3, Suricata is GPLv2. According to http://www.gnu.org/licenses/gpl-faq.html#AllCompatibility these are incompatible.

#2 - 08/27/2012 02:20 AM - Victor Julien
  - Assignee set to Victor Julien

#3 - 09/06/2012 10:11 AM - Victor Julien
In the process of asking for legal advice on this.

#4 - 09/08/2012 03:48 PM - Victor Julien
Another lib: http://research.wand.net.nz/software/libprotoident.php

#5 - 12/22/2015 04:02 PM - Andreas Herz
Any update on the legal issue? the lib from wand looks not really up to date

#6 - 04/28/2016 06:18 AM - Vito Piserchia
I'll try to put and add more information about different protocol inspection projects, for now:

- libprotoident
- nDPI
Generalities (contact page, main development site, etc etc)

**libprotoident**

Description:
(from the project page)

Libprotoident is a library that performs application layer protocol identification for flows. Unlike many techniques that require capturing the entire packet payload, only the first four bytes of payload sent in each direction, the size of the first payload-bearing packet in each direction and the TCP or UDP port numbers for the flow are used by libprotoident.

- source code: [https://github.com/wanduow/libprotoident](https://github.com/wanduow/libprotoident)
- blog and news: [https://secure.wand.net.nz/projects/details/libprotoident](https://secure.wand.net.nz/projects/details/libprotoident)
- wiki: [https://secure.wand.net.nz/trac/libprotoident/](https://secure.wand.net.nz/trac/libprotoident/)

**nDPI**

Description
(from the project page):

nDPI is a ntop-maintained superset of the popular OpenDPI library. Released under the LGPL license, its goal is to extend the original library by adding new protocols that are otherwise available only on the paid version of OpenDPI. In addition to Unix platforms, we also support Windows, in order to provide you a cross-platform DPI experience. Furthermore, we have modified nDPI do be more suitable for traffic monitoring applications, by disabling specific features that slow down the DPI engine while being them un-necessary for network traffic monitoring.

- source code: [https://github.com/ntop/nDPI](https://github.com/ntop/nDPI)
- blog and news: [http://ntop.org](http://ntop.org)

**Specificities**

**libprotoident**

- Language: C++
- Protocol Identification:
  - Payload Matches (mostly), pattern matching on the first four bytes of the payload on each direction of the traffic
  - Payload Size
  - Port Number, used in case of ambiguity and only for well known ports
  - IP Matching, very few cases
  - protocols are checked in order depending on the confidence the author have on the rules and the popularity of the protocol
- Multi thread support: no, but can be added in the user application
- Flow-aware: no, but can use the wand libflowmanager, from [http://research.wand.net.nz/software/libflowmanager.php](http://research.wand.net.nz/software/libflowmanager.php). Only needed for building the tools
- Test Pcap: none

**nDPI**

- Language: C
- Dependencies: none except for the build environment
- Protocol Identification:
  - Payload Matches
  - Payload Size
  - Port Number
  - IP Matching
  - ability to specify custom ports for protocol in specific environment through a configuration file
- Multi thread support: yes
- Flow-aware: yes (embedded)
- Test Pcap: few in the code base
Community support

- **libprotoident**
  The source code is openly available on GitHub, at the moment there is only one author active.

- **nDPI**
  The source code is openly available on GitHub.

Licence

- **libprotoident**
  GPLv2.

- **nDPI**
  LGPLv3. This is an issue if you want to include its source into suricat, according to http://www.gnu.org/licenses/gpl-faq.html#AllCompatibility

Supported Protocols

- **libprotoident**
  A list is here (updated?): https://secure.wand.net.nz/trac/libprotoident/wiki/SupportedProtocols

- **nDPI**
  A list is present on the main project page: http://www.ntop.org/products/deep-packet-inspection/ndpi/

Interesting Papers

  Very long (more than 400 pages) comparison of the most popular and public available DPI engine. A shorter article version from the same author also exists (Nov. 2014, http://tomasz.bujlow.com/publications/2014_journal_elsevier_comnet_independent_comparison.pdf)

#7 - 04/29/2016 02:45 AM - Vito Piserchia
Important statement about the **libprotoident** and its future from his author can be found here: https://github.com/wanduow/libprotoident/issues/12

#8 - 05/02/2016 04:58 PM - Andreas Herz
Does anyone have some real experience with those projects and could share that knowledge? Might be worth to take a look at it but seems like a more time consuming task :)

#9 - 05/03/2016 09:08 AM - Victor Julien
- Assignee changed from Victor Julien to Anonymous

I think we can rule out nDPI for the licensing issue. The libprotoident might be worth looking into, although I'm a bit worried about it's continued development. Also the further dependencies the lib has might be an issue.

#10 - 05/03/2016 10:15 AM - Vito Piserchia
The libprotoident comes out with a few examples, in the tools folder of the code. From the README https://github.com/wanduow/libprotoident

- **lpi_protoident**
  Description:
  This tool attempts to identify each individual flow within the provided trace. Identification only occurs when the flow has concluded or expired, so it is not very effective for real-time applications.

- **lpi_live** (DEPRECATED)
  Description:
  This tool reports byte and packet counts (both inbound and outbound) for each identified protocol in real-time. Identification of a flow occurs as soon as possible, so that the statistics reported are as up-to-date as possible.
  lpi_live has been deprecated and will not be built by default. The code is still available in our git repository, but we will not update or support this tool anymore. Instead, please use the lpicollector (https://github.com/wanduow/lpicollector) for real-time flow analysis with libprotoident.
  In combination with the included lpi.py example client, lpicollector can produce output similar to that produced by lpi_live.
Libprotoident calls (to external libs) and mappings to suricata's (tentative)

from libtrace:

- types
  
  libtrace_t | NOT NEEDED
  libtrace_tcp_t | combination of PKT_IS_TCP(p) && (p)->tcph
  libtrace_udp_t | combination of PKT_IS_UDP(p) && (p)->udph
  libtrace_packet_t | analogous to the Packet struct
  libtrace_filter_t | use the suricata's builtin BPF code

- calls

  trace_get_layer3 | IP_GET_IPPROTO(p)
  trace_get_direction | FlowGetPacketDirection(f, p) == TOSERVER ? 1 : 0; **NOTE:** they are reversed
  trace_get_payload | (Packet *)p->payload
  trace_get_payload_length | (Packet *)p->payload_len
  trace_read_packet | NOT NEEDED, use suricata source modules
  trace_get_seconds | XXX
  trace_create | NOT NEEDED, use suricata source modules
  trace_destroy | NOT NEEDED, use suricata source modules
  trace_create_filter | use suricata's BPF filtering
  trace_perror | NOT NEEDED
  trace_is_err | NOT NEEDED

from libflowmanager

NEEDED | lfm_match_packet_to_flow | FlowReference(&p->flow, f);
NEEDED | lfm_update_flow_expiry_timeout | this is done via the flow-timeouts values
NOT NEEDED | lfm_expire_next_flow
NOT NEEDED | lfm_set_config_option

#11 - 07/13/2017 05:14 AM - Fanny Dwargee
Do we have any progress with this feature?

It seems that the GitHub repository of libprotoident is updated frequently (at least in the last months) and this would make Suricata another step ahead of Snort ;)

#12 - 12/21/2018 12:07 PM - Victor Julien
- Related to Feature #2757: improve protocol detection added

#13 - 12/21/2018 06:23 PM - Vito Piserchia
Since from 6th August 2016 libprotoident is GNUv3 licensed [1], putting it in the same situation as nDPI

There is an open ticket [2] but it got no reply since Mar 9, 2017

[1] https://github.com/wanduow/libprotoident/commit/de8e2ca6d6eb04526912dbcc433b0c6003b965e1#diff-7116ef0705885343c9e1b2171a06be0e

#14 - 02/23/2019 10:09 PM - Andreas Herz
- Assignee set to Community Ticket