pcap-analysis Documentation

Release 0.1.1

Tyler N. Thieding

Jun 29, 2020

Contents

1 About **3**
- 1.1 Contributors . . . 3
- 1.2 Development . . . 3
- 1.3 License . . . 3

2 API Reference **5**
- 2.1 Classes . . . 5
- 2.2 Analyzers . . . 6

3 Installation **9**
- 3.1 Requirements . . . 9
- 3.2 Installation Steps . . . 9

4 Release Notes **11**
- 4.1 [0.1.1] Fix analyzer class internal attribute logic. (2020-06-29) . . . 11
- 4.2 [0.1.0] Initial beta release. (2020-06-28) . . . 11

5 Known Limitations **13**

6 Usage **15**

Python Module Index **17**

Index **19**

Analyze packet capture format (`.pcap` or `.pcapng`) files.

CHAPTER 1

About

1.1 Contributors

- Tyler N. Thieding (Primary Author)

1.2 Development

Repository https://gitlab.com/TNThieding/pcap-analysis/

1.3 License

```
Copyright 2020 Tyler N. Thieding

Permission is hereby granted, free of charge, to any person obtaining a copy of this
software and associated documentation files (the "Software"), to deal in the Software
without restriction, including without limitation the rights to use, copy, modify,␣
↪merge,
publish, distribute, sublicense, and/or sell copies of the Software, and to permit␣
↪persons
to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies␣
↪or
substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A␣
↪PARTICULAR
PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE␣
↪LIABLE
```

(continues on next page)

(continued from previous page)

```
FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE SOFTWARE.
```

CHAPTER 2

API Reference

- *Classes*
  - *PacketAnalyzer*
- *Analyzers*
  - *ARP*
  - *BOOTP*
  - *DHCP*
  - *ICMP*

Analyze packet capture format (pcap) files.

2.1 Classes

2.1.1 PacketAnalyzer

class pcap_analysis.**PacketAnalyzer**(*pcap_file*)

Packet analyzer interface.

Parameters **pcap_file** (*str*) – path to packet capture file (i.e., *pcap* or‘pcapng‘)

arp

Address resolution protocol (ARP) analyzer accessor.

bootp

Bootstrap protocol (BOOTP) analyzer accessor.

dhcp

Dynamic host configuration protocol (DHCP) analyzer accessor.

icmp
Internet control message protocol (ICMP) analyzer accessor.

2.2 Analyzers

Access analyzer class instances through the `PacketAnalyzer` class. They should not be instantiated directly and used standalone!

2.2.1 ARP

class pcap_analysis._analyzers.arp.**Arp**
Address resolution protocol (ARP) analyzer.

did_device_arp_for(*mac_address*, *target_ip*)
Check if the specified device ARPed for the specified target IP address.

Parameters

- **mac_address** (*str*) – device MAC address
- **target_ip** (*str*) – target IP address

Returns device sent ARP packet(s)

Return type bool

did_device_receive_response(*mac_address*, *target_ip*)
Check if the specified device received an ARP reply from the specified target IP address.

If the device received a response, the IP and MAC address are included in the ARP table accessible with the `get_arp_table` method.

Parameters

- **mac_address** (*str*) – device MAC address
- **target_ip** (*str*) – target IP address

Returns device received ARP reply

Return type bool

get_arp_table(*mac_address*, *include_gratuitous=True*)
Generate a hypothetical ARP table based on network traffic.

Parameters

- **mac_address** (*str*) – device MAC address
- **include_gratuitous** (*bool*) – include gratuitous ARP entries

Returns generated ARP table

Return type dict

Raises **ValueError** – specified MAC address not observed in network traffic

get_gratuitous_arp_ips(*mac_address*)
Get set of IP address(es) announced via gratuitous ARP for the specified device.

Parameters **mac_address** (*str*) – device MAC address

Returns announced IP addresses

Return type set of str

Raises **ValueError** – no gratuitous ARP packets sent from specified MAC address

get_probed_ips(*mac_address*)
Get set of IP address(es) probed by the specified device.

Parameters **mac_address** (*str*) – device MAC address

Returns probed IP addresses

Return type set of str

Raises **ValueError** – no probe ARP packets sent from specified MAC address

2.2.2 BOOTP

class pcap_analysis._analyzers.bootp.**Bootp**
Bootstrap protocol (BOOTP) analyzer.

did_client_make_request(*mac_address*)
Check if a device requested an IP address using BOOTP.

Parameters **mac_address** (*str*) – client device MAC address

Returns client made BOOTP request

Return type bool

did_client_receive_ip_address(*mac_address*)
Check if a device received an IP address using BOOTP.

Parameters **mac_address** (*str*) – client device MAC address

Returns client received IP address

Return type bool

get_received_ip_address(*mac_address*)
Get IP address assigned to device via BOOTP.

Parameters **mac_address** (*str*) – client device MAC address

Returns assigned IP address

Return type str

Raises **ValueError** – no IP address assigned to specified MAC address

2.2.3 DHCP

class pcap_analysis._analyzers.dhcp.**Dhcp**
Dynamic host configuration protocol (DHCP) analyzer.

did_client_make_request(*mac_address*)
Check if a device requested an IP address using DHCP.

Parameters **mac_address** (*str*) – client device MAC address

Returns client made DHCP request

Return type bool

did_client_receive_ip_address(*mac_address*)
Check if a device received an IP address using DHCP.

Parameters **mac_address** (*str*) – client device MAC address

Returns client received IP address

Return type bool

get_received_ip_address(*mac_address*)
Get IP address assigned to device via DHCP.

Parameters **mac_address** (*str*) – client device MAC address

Returns assigned IP address

Return type str

Raises **ValueError** – no IP address assigned to specified MAC address

2.2.4 ICMP

class pcap_analysis._analyzers.icmp.**Icmp**
Internet control message protocol (ICMP) analyzer.

did_device_ping(*source_host_ip*, *target_host_ip*)
Check if the specified source device pinged the specified target IP address.

Parameters

- **source_host_ip** (*str*) – source IP address
- **target_host_ip** (*str*) – target IP address

Returns device pined specified target

Return type bool

get_mean_rtt(*source_host_ip*, *target_host_ip*)
Calculate average round-trip time for the specified source and target hosts.

Parameters

- **source_host_ip** (*str*) – source IP address
- **target_host_ip** (*str*) – target IP address

Returns average round-trip time

Return type float

get_ping_count(*source_host_ip*, *target_host_ip*)
Count ping requests from source host to target host that received a response.

Parameters

- **source_host_ip** (*str*) – source IP address
- **target_host_ip** (*str*) – target IP address

Returns number of ping requests with a corresponding response

Return type int

CHAPTER 3

Installation

3.1 Requirements

- Python 2.7 or 3.5+
- Wireshark

3.2 Installation Steps

Install `pcap-analysis` from the command line using pip:

```
pip install pcap-analysis
```

CHAPTER 4

Release Notes

4.1 [0.1.1] Fix analyzer class internal attribute logic. (2020-06-29)

Previously, analyzer classes attached internal-use attributes to the class itself. Now, these attributes are instance attributes as expected.

4.2 [0.1.0] Initial beta release. (2020-06-28)

Release initial beta version of *pcap-analysis* package with analyzers for the following protocols:

- ARP
- BOOTP
- DHCP
- ICMP (Pings Only)

CHAPTER 5

Known Limitations

This package contains the following known limitations:

- None

CHAPTER 6

Usage

Under construction! Coming soon…

Python Module Index

p

pcap_analysis, 5

Index

A

Arp (*class in pcap_analysis._analyzers.arp*), 6
arp (*pcap_analysis.PacketAnalyzer attribute*), 5

B

Bootp (*class in pcap_analysis._analyzers.bootp*), 7
bootp (*pcap_analysis.PacketAnalyzer attribute*), 5

D

Dhcp (*class in pcap_analysis._analyzers.dhcp*), 7
dhcp (*pcap_analysis.PacketAnalyzer attribute*), 5
did_client_make_request() (*pcap_analysis._analyzers.bootp.Bootp method*), 7
did_client_make_request() (*pcap_analysis._analyzers.dhcp.Dhcp method*), 7
did_client_receive_ip_address() (*pcap_analysis._analyzers.bootp.Bootp method*), 7
did_client_receive_ip_address() (*pcap_analysis._analyzers.dhcp.Dhcp method*), 7
did_device_arp_for() (*pcap_analysis._analyzers.arp.Arp method*), 6
did_device_ping() (*pcap_analysis._analyzers.icmp.Icmp method*), 8
did_device_receive_response() (*pcap_analysis._analyzers.arp.Arp method*), 6

G

get_arp_table() (*pcap_analysis._analyzers.arp.Arp method*), 6
get_gratuitous_arp_ips() (*pcap_analysis._analyzers.arp.Arp method*), 6
get_mean_rtt() (*pcap_analysis._analyzers.icmp.Icmp method*), 8
get_ping_count() (*pcap_analysis._analyzers.icmp.Icmp method*), 8
get_probed_ips() (*pcap_analysis._analyzers.arp.Arp method*), 7
get_received_ip_address() (*pcap_analysis._analyzers.bootp.Bootp method*), 7
get_received_ip_address() (*pcap_analysis._analyzers.dhcp.Dhcp method*), 8

I

Icmp (*class in pcap_analysis._analyzers.icmp*), 8
icmp (*pcap_analysis.PacketAnalyzer attribute*), 5

P

PacketAnalyzer (*class in pcap_analysis*), 5
pcap_analysis (*module*), 5