Lecture 5: Passive Testing & Network Trace Analysis

Passive Testing Techniques for Communication Protocols

Dr. Jorge López, PhD.
jorgelopezcoronado[at]gmail.com

National Research Tomsk State University

February 22, 2016

OUTLINE

INTRODUCTION & ENVIRONMENT DESCRIPTION

DEEP PACKET INSPECTION

PASSIVE TESTING WITH NETWORK TRACES

Regarding "non-intrusive" testing methods

By now you **should** know:

REGARDING "NON-INTRUSIVE" TESTING METHODS

By now you **should** know:

- Static Analysis

REGARDING "NON-INTRUSIVE" TESTING METHODS

By now you **should** know:

- Static Analysis
  - At least the basics…

REGARDING "NON-INTRUSIVE" TESTING METHODS

By now you **should** know:

- Static Analysis
  - At least the basics…
  - Lots of research opportunities, some *trending topics* include search-based software testing, or static analysis formal methods & the associated complexity issues

REGARDING "NON-INTRUSIVE" TESTING METHODS

By now you **should** know:

- Static Analysis
  - At least the basics…
  - Lots of research opportunities, some *trending topics* include search-based software testing, or static analysis formal methods & the associated complexity issues
  - The principles of parsing can be applied to **many** other fields, e.g., code optimization, etc.

REGARDING "NON-INTRUSIVE" TESTING METHODS

By now you **should** know:

- Static Analysis
  - At least the basics…
  - Lots of research opportunities, some *trending topics* include search-based software testing, or static analysis formal methods & the associated complexity issues
  - The principles of parsing can be applied to **many** other fields, e.g., code optimization, etc.
  - It is fun :)

REGARDING "NON-INTRUSIVE" TESTING METHODS

By now you **should** know:

- Static Analysis
  - At least the basics…
  - Lots of research opportunities, some *trending topics* include search-based software testing, or static analysis formal methods & the associated complexity issues
  - The principles of parsing can be applied to **many** other fields, e.g., code optimization, etc.
  - It is fun :)

Now, we don't have access to the code…

Regarding "non-intrusive" testing methods

By now you **should** know:

- Static Analysis
  - At least the basics…
  - Lots of research opportunities, some *trending topics* include search-based software testing, or static analysis formal methods & the associated complexity issues
  - The principles of parsing can be applied to **many** other fields, e.g., code optimization, etc.
  - It is fun :)

Now, we don't have access to the code…

- We know what we want to test interacts over the network, perhaps we can take a look at the exchanged protocol messages?

Network trace analysis

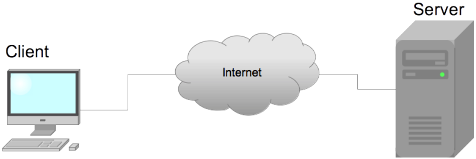
Environment description

NETWORK INTERACTION & SOURCE OF DATA

A typical network interaction looks like this:

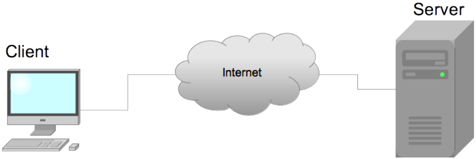
Network interaction & source of data

A typical network interaction looks like this:



NETWORK INTERACTION & SOURCE OF DATA

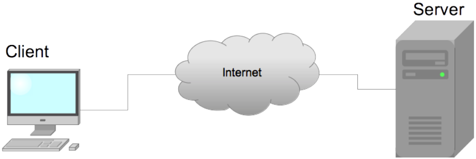
A typical network interaction looks like this:



Which one is the System Under Test (SUT)?

NETWORK INTERACTION & SOURCE OF DATA

A typical network interaction looks like this:

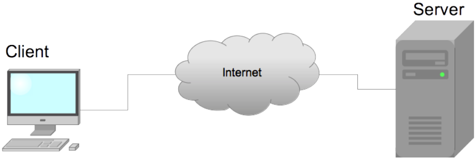


Which one is the System Under Test (SUT)?

- Where would the data be extracted?

Network interaction & source of data

A typical network interaction looks like this:

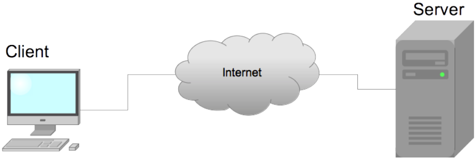


Which one is the System Under Test (SUT)?

- Where would the data be extracted?
  - Could we see all behaviors independently from the source of the data?

NETWORK INTERACTION & SOURCE OF DATA

A typical network interaction looks like this:



Which one is the System Under Test (SUT)?

- Where would the data be extracted?
  - Could we see all behaviors independently from the source of the data?
  - Or what would change?

NETWORK INTERACTION & SOURCE OF DATA II

The concept of Point of Observation (P.O.)

Network interaction & source of data II

The concept of Point of Observation (P.O.)

- The P.O. is where the network traces (sequence of network packets) is taken from

NETWORK INTERACTION & SOURCE OF DATA II

The concept of Point of Observation (P.O.)

- The P.O. is where the network traces (sequence of network packets) is taken from
- If the P.O. is not placed at the SUT (or a point where all its data goes by, e.g., its gateway), what can be observed?

NETWORK INTERACTION & SOURCE OF DATA II

The concept of Point of Observation (P.O.)

- The P.O. is where the network traces (sequence of network packets) is taken from
- If the P.O. is not placed at the SUT (or a point where all its data goes by, e.g., its gateway), what can be observed?
  - Only the data from the SUT that goes trough the P.O (duh!)

NETWORK INTERACTION & SOURCE OF DATA II

The concept of Point of Observation (P.O.)

- The P.O. is where the network traces (sequence of network packets) is taken from
- If the P.O. is not placed at the SUT (or a point where all its data goes by, e.g., its gateway), what can be observed?
  - Only the data from the SUT that goes trough the P.O (duh!)
  - Sometimes you cannot situate the P.O at the SUT, sort of *testing in context*,"little something is a lot, compared to nothing" (they say in my country…well, in Spanish…, i.e., this can still be useful)

NETWORK INTERACTION & SOURCE OF DATA II

The concept of Point of Observation (P.O.)

- The P.O. is where the network traces (sequence of network packets) is taken from
- If the P.O. is not placed at the SUT (or a point where all its data goes by, e.g., its gateway), what can be observed?
    - Only the data from the SUT that goes trough the P.O (duh!)
    - Sometimes you cannot situate the P.O at the SUT, sort of *testing in context*,“little something is a lot, compared to nothing” (they say in my country…well, in Spanish…, i.e., this can still be useful)
- If the P.O. is not situated at the SUT, but the data we are interested goes by the P.O., what changes? E.g., test a VSNP server with the P.O. placed @ a VSNP client

Network interaction & source of data II

The concept of Point of Observation (P.O.)

- The P.O. is where the network traces (sequence of network packets) is taken from
- If the P.O. is not placed at the SUT (or a point where all its data goes by, e.g., its gateway), what can be observed?
  - Only the data from the SUT that goes trough the P.O (duh!)
  - Sometimes you cannot situate the P.O at the SUT, sort of *testing in context*,"little something is a lot, compared to nothing" (they say in my country. . . well, in Spanish. . . , i.e., this can still be useful)
- If the P.O. is not situated at the SUT, but the data we are interested goes by the P.O., what changes? E.g., test a VSNP server with the P.O. placed @ a VSNP client
  - Mostly, the sense of "direction". The VSNP server responses: outgoing from local IP if P.O. @ server; incoming from a remote IP if P.O. @ client

OBTAINING THE DATA FROM THE P.O.

P.O. data collection can be done:

Obtaining the data from the P.O.

P.O. data collection can be done:

- Off-line: Capture the data and store it in a file, a network trace (or packet capture), tcpdump, wireshark, etc, move it to the tester and **later**, analyze it

OBTAINING THE DATA FROM THE P.O.

P.O. data collection can be done:

- Off-line: Capture the data and store it in a file, a network trace (or packet capture), tcpdump, wireshark, etc, move it to the tester and **later**, analyze it
- On-line: Capture the data from the P.O., and "forward" it immediately [to the tester]

OBTAINING THE DATA FROM THE P.O.

P.O. data collection can be done:

- Off-line: Capture the data and store it in a file, a network trace (or packet capture), tcpdump, wireshark, etc, move it to the tester and **later**, analyze it
- On-line: Capture the data from the P.O., and “forward” it immediately [to the tester]
  - The “forwarding” can be to a local tester application if the data traffic goes through the tester itself, mostly implementations based on libpcap (wireshark, tcpdump, snort, anything else use this library, you will too…)

OBTAINING THE DATA FROM THE P.O.

P.O. data collection can be done:

- Off-line: Capture the data and store it in a file, a network trace (or packet capture), tcpdump, wireshark, etc, move it to the tester and **later**, analyze it
- On-line: Capture the data from the P.O., and “forward” it immediately [to the tester]
  - The “forwarding” can be to a local tester application if the data traffic goes through the tester itself, mostly implementations based on libpcap (wireshark, tcpdump, snort, anything else use this library, you will too...)
  - If the tester is not the P.O., port forwarding (switch send data that goes to a port to another port, promiscuous mode needed)

OBTAINING THE DATA FROM THE P.O.

P.O. data collection can be done:

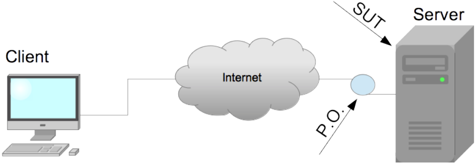
- Off-line: Capture the data and store it in a file, a network trace (or packet capture), tcpdump, wireshark, etc, move it to the tester and **later**, analyze it
- On-line: Capture the data from the P.O., and "forward" it immediately [to the tester]
  - The "forwarding" can be to a local tester application if the data traffic goes through the tester itself, mostly implementations based on libpcap (wireshark, tcpdump, snort, anything else use this library, you will too...)
  - If the tester is not the P.O., port forwarding (switch send data that goes to a port to another port, promiscuous mode needed)
  - Small tool/protocol developed in academia packet capture over IP, filter, and send(optionally using SSL/TLS) to a remote host; "somebody" planned to post this on-line as open source tool...

ENVIRONMENT – FINAL REMARKS

An image, 10^3 words…

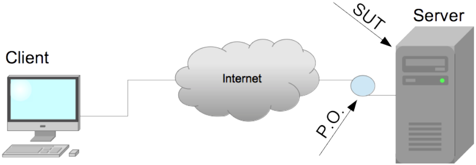
ENVIRONMENT – FINAL REMARKS

An image, 10^3 words…



ENVIRONMENT – FINAL REMARKS

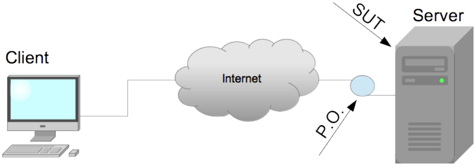
An image, 10^3 words…



- Direction: From server to client

Environment – Final Remarks

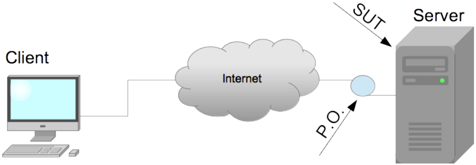
An image, 10^3 words…



- Direction: From server to client
- P.O. = **one** network interface of the server (usually a P.O. is associated with a network host, it can vary…)

Environment – Final Remarks

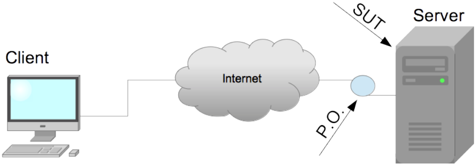
An image, 10^3 words...



- Direction: From server to client
- P.O. = **one** network interface of the server (usually a P.O. is associated with a network host, it can vary...)
- No information regarding on-line or off-line (perhaps on-line is more interesting)

Environment – Final Remarks

An image, 10^3 words…



- Direction: From server to client
- P.O. = **one** network interface of the server (usually a P.O. is associated with a network host, it can vary…)
- No information regarding on-line or off-line (perhaps on-line is more interesting)

We know about the environment and how to obtain the data, how do we test this?

Deep Packet Inspection (DPI)

Deep Packet Inspection (DPI) 101

What is DPI?

Deep Packet Inspection (DPI) 101

What is DPI?

- The process of examining the data of a network packet in the search of certain *values* in it, for instance:

DEEP PACKET INSPECTION (DPI) 101

What is DPI?

- The process of examining the data of a network packet in the search of certain *values* in it, for instance:
  - Protocol non-compliance, e.g., TCP SYN/FIN (or Christmas/xmas tree packet)

DEEP PACKET INSPECTION (DPI) 101

What is DPI?

- The process of examining the data of a network packet in the search of certain *values* in it, for instance:
  - Protocol non-compliance, e.g., TCP SYN/FIN (or Christmas/xmas tree packet)
  - Viruses, buffer overflows. How?

DEEP PACKET INSPECTION (DPI) 101

What is DPI?

- The process of examining the data of a network packet in the search of certain *values* in it, for instance:
  - Protocol non-compliance, e.g., TCP SYN/FIN (or Christmas/xmas tree packet)
  - Viruses, buffer overflows. How?
    - A packet containing a known binary sequence, the virus program itself or part of it which identifies it well, a **signature**

Deep Packet Inspection (DPI) 101

What is DPI?

- The process of examining the data of a network packet in the search of certain *values* in it, for instance:
  - Protocol non-compliance, e.g., TCP SYN/FIN (or Christmas/xmas tree packet)
  - Viruses, buffer overflows. How?
    - A packet containing a known binary sequence, the virus program itself or part of it which identifies it well, a **signature**
  - Specific application layer data, for instance:

```
a'; DROP TABLE users
```

DEEP PACKET INSPECTION (DPI) 101

What is DPI?

- The process of examining the data of a network packet in the search of certain *values* in it, for instance:
  - Protocol non-compliance, e.g., TCP SYN/FIN (or Christmas/xmas tree packet)
  - Viruses, buffer overflows. How?
    - A packet containing a known binary sequence, the virus program itself or part of it which identifies it well, a **signature**
  - Specific application layer data, for instance:

    ```
    a'; DROP TABLE users
    ```

What to do once certain value is found?

DEEP PACKET INSPECTION (DPI) 101

What is DPI?

- The process of examining the data of a network packet in the search of certain *values* in it, for instance:
  - Protocol non-compliance, e.g., TCP SYN/FIN (or Christmas/xmas tree packet)
  - Viruses, buffer overflows. How?
    - A packet containing a known binary sequence, the virus program itself or part of it which identifies it well, a **signature**
  - Specific application layer data, for instance:

```
a'; DROP TABLE users
```

What to do once certain value is found?

- Report the finding. Usually searching has the sense of searching for prohibited elements

Deep Packet Inspection (DPI) 102

Using DPI

Deep Packet Inspection (DPI) 102

Using DPI

- Mostly used in firewalls, Intrusion Detection Systems (IDS), and Intrusion Prevention Systems (IPS) given the nature of DPI (i.e., finding prohibited values)

Deep Packet Inspection (DPI) 102

Using DPI

- Mostly used in firewalls, Intrusion Detection Systems (IDS), and Intrusion Prevention Systems (IPS) given the nature of DPI (i.e., finding prohibited values)
- Off-line approaches are not very popular

DEEP PACKET INSPECTION (DPI) 102

Using DPI

- Mostly used in firewalls, Intrusion Detection Systems (IDS), and Intrusion Prevention Systems (IPS) given the nature of DPI (i.e., finding prohibited values)
- Off-line approaches are not very popular
- Off-line approaches are sometimes considered some form of computer forensics

Deep Packet Inspection (DPI) 102

Using DPI

- Mostly used in firewalls, Intrusion Detection Systems (IDS), and Intrusion Prevention Systems (IPS) given the nature of DPI (i.e., finding prohibited values)
- Off-line approaches are not very popular
- Off-line approaches are sometimes considered some form of computer forensics

How to describe these values to search?

Deep Packet Inspection (DPI) 102

Using DPI

- Mostly used in firewalls, Intrusion Detection Systems (IDS), and Intrusion Prevention Systems (IPS) given the nature of DPI (i.e., finding prohibited values)
- Off-line approaches are not very popular
- Off-line approaches are sometimes considered some form of computer forensics

How to describe these values to search?

- Many existing approaches (Cisco, Snort, etc.) , nonetheless, they tend to have common points. . .

DESCRIBING VALUES IN DPI

Based on “rules”

DESCRIBING VALUES IN DPI

Based on “rules”

- For common protocols (IP, TCP, UDP, HTTP, etc.), variables are provided (for example:

Describing values in DPI

Based on “rules”

- For common protocols (IP, TCP, UDP, HTTP, etc.), variables are provided (for example:

```
alert tcp any any -> any 21 (msg:"FTP ROOT"; content:"USER root"; nocase;)
```

Describing values in DPI

Based on “rules”

- For common protocols (IP, TCP, UDP, HTTP, etc.), variables are provided (for example:

```
alert tcp any any -> any 21 (msg:"FTP ROOT"; content:"USER root"; nocase;)
```

- A signature can be described as a set of strings(potentially binary) of a regular language.

Describing values in DPI

Based on "rules"

- For common protocols (IP, TCP, UDP, HTTP, etc.), variables are provided (for example:

```
alert tcp any any -> any 21 (msg:"FTP ROOT"; content:"USER root"; nocase;)
```

- A signature can be described as a set of strings(potentially binary) of a regular language.
  - Yes, you guessed correctly, described by a regular expression

Describing values in DPI

Based on “rules”

- For common protocols (IP, TCP, UDP, HTTP, etc.), variables are provided (for example:

```
alert tcp any any -> any 21 (msg:"FTP ROOT"; content:"USER root"; nocase;)
```

- A signature can be described as a set of strings(potentially binary) of a regular language.
  - Yes, you guessed correctly, described by a regular expression

```
alert tcp any any -> any 80 (content:"/foo.php?id=";
 pcre:"/foo.php?id=[0-9]{1,10}/iU";)
```

DESCRIBING VALUES IN DPI

Based on "rules"

- For common protocols (IP, TCP, UDP, HTTP, etc.), variables are provided (for example:

```
alert tcp any any -> any 21 (msg:"FTP ROOT"; content:"USER root"; nocase;)
```

- A signature can be described as a set of strings(potentially binary) of a regular language.
  - Yes, you guessed correctly, described by a regular expression

```
alert tcp any any -> any 80 (content:"/foo.php?id=";
 pcre:"/foo.php?id=[0-9]{1,10}/iU";)
```

(All the previous syntax were snort rules)

DPI PARTICULARITIES

Many protocols in a single P.O. is usual

DPI PARTICULARITIES

Many protocols in a single P.O. is usual

- The more detailed description of the target packet the better

DPI PARTICULARITIES

Many protocols in a single P.O. is usual

- The more detailed description of the target packet the better
  - Packet matching usually done sequentially, if part of the rule does not discard the packet, keep checking the properties

DPI PARTICULARITIES

Many protocols in a single P.O. is usual

- The more detailed description of the target packet the better
  - Packet matching usually done sequentially, if part of the rule does not discard the packet, keep checking the properties
  - A bad rule:

```
alert any any any -> any any ( pcre:"/foo.php?id=[0-9]{1,10}/iU";)
```

It will search in **all** packets for the RE. Ideally just needed to be applied for HTTP (TCP port 80) and those who's URL contain "/foo.php?id="

DPI PARTICULARITIES

Many protocols in a single P.O. is usual

- The more detailed description of the target packet the better
  - Packet matching usually done sequentially, if part of the rule does not discard the packet, keep checking the properties
  - A bad rule:

```
alert any any any -> any any ( pcre:"/foo.php?id=[0-9]{1,10}/iU";)
```

It will search in **all** packets for the RE. Ideally just needed to be applied for HTTP (TCP port 80) and those who's URL contain "/foo.php?id="

Encrypted protocols...

DPI particularities

Many protocols in a single P.O. is usual

- The more detailed description of the target packet the better
  - Packet matching usually done sequentially, if part of the rule does not discard the packet, keep checking the properties
  - A bad rule:

```
alert any any any -> any any ( pcre:"/foo.php?id=[0-9]{1,10}/iU";)
```

It will search in **all** packets for the RE. Ideally just needed to be applied for HTTP (TCP port 80) and those who's URL contain "/foo.php?id="

Encrypted protocols. . .

- Good for security, bad for DPI!

DPI & Encryption

A simple HTTP(S) request

DPI & Encryption

A simple HTTP(S) request

- This:

```
54:d7:dc:72:ea:02:54:bf:e7:4a:bb:c3:3c:80:63:73:
59:bb:ec:9a:14:15:de:ce:7a:a7:f3:f2:5e:da:72:cb
```

DPI & Encryption

A simple HTTP(S) request

- This:

```
54:d7:dc:72:ea:02:54:bf:e7:4a:bb:c3:3c:80:63:73:
59:bb:ec:9a:14:15:de:ce:7a:a7:f3:f2:5e:da:72:cb
```

- Is part of this:

```
GET / HTTP/1.1
Host kitidis.tsu.ru
```

DPI & Encryption

A simple HTTP(S) request

- This:

```
54:d7:dc:72:ea:02:54:bf:e7:4a:bb:c3:3c:80:63:73:
59:bb:ec:9a:14:15:de:ce:7a:a7:f3:f2:5e:da:72:cb
```

- Is part of this:

```
GET / HTTP/1.1
Host kitidis.tsu.ru
```

Two main solutions

DPI & Encryption

A simple HTTP(S) request

- This:

```
54:d7:dc:72:ea:02:54:bf:e7:4a:bb:c3:3c:80:63:73:
59:bb:ec:9a:14:15:de:ce:7a:a7:f3:f2:5e:da:72:cb
```

- Is part of this:

```
GET / HTTP/1.1
Host kitidis.tsu.ru
```

Two main solutions

- The elegant: look for the SSL/TLS handshake ↦ obtain the keys ↦ decrypt each packet ↦ analyze as usual

DPI & Encryption

A simple HTTP(S) request

- This:

```
54:d7:dc:72:ea:02:54:bf:e7:4a:bb:c3:3c:80:63:73:
59:bb:ec:9a:14:15:de:ce:7a:a7:f3:f2:5e:da:72:cb
```

- Is part of this:

```
GET / HTTP/1.1
Host kitidis.tsu.ru
```

Two main solutions

- The elegant: look for the SSL/TLS handshake ↦ obtain the keys ↦ decrypt each packet ↦ analyze as usual
- The fast: expect the network trace decrypted by any external entity

Stateless and Stateful DPI concepts

Stateless DPI

Stateless and Stateful DPI concepts

Stateless DPI

- Each rule is applied to each network packet and no state is saved

Stateless and Stateful DPI concepts

Stateless DPI

- Each rule is applied to each network packet and no state is saved
- Can be good if we are tying to search for a virus transmuted over SMTP, if port SMTP and signature found, then virus alert

STATELESS AND STATEFUL DPI CONCEPTS

Stateless DPI

- Each rule is applied to each network packet and no state is saved
- Can be good if we are tying to search for a virus transmuted over SMTP, if port SMTP and signature found, then virus alert

Stateful DPI

STATELESS AND STATEFUL DPI CONCEPTS

Stateless DPI

- Each rule is applied to each network packet and no state is saved
- Can be good if we are tying to search for a virus transmuted over SMTP, if port SMTP and signature found, then virus alert

Stateful DPI

- Stateful refers to store certain information regarding a connection, state of the connection, rules also check state

STATELESS AND STATEFUL DPI CONCEPTS

Stateless DPI

- Each rule is applied to each network packet and no state is saved
- Can be good if we are tying to search for a virus transmuted over SMTP, if port SMTP and signature found, then virus alert

Stateful DPI

- Stateful refers to store certain information regarding a connection, state of the connection, rules also check state
- Look for FTP data channel commands, when detected, associate to FTP session the data channel, they are related

Beyond DPI

What if we want more?

BEYOND DPI

What if we want more?

- What is more?

BEYOND DPI

What if we want more?

- What is more?
  - For every VSNP client request, a VSNP server response should follow; such response should respect the even/odd, odd/even constraint of the protocol

BEYOND DPI

What if we want more?

- What is more?
  - For every VSNP client request, a VSNP server response should follow; such response should respect the even/odd, odd/even constraint of the protocol
  - Not for typical protocols or predefined rules

Beyond DPI

What if we want more?

- What is more?
  - For every VSNP client request, a VSNP server response should follow; such response should respect the even/odd, odd/even constraint of the protocol
  - Not for typical protocols or predefined rules
  - To choose what to save and how to correlate it to future packets

BEYOND DPI

What if we want more?

- What is more?
  - For every VSNP client request, a VSNP server response should follow; such response should respect the even/odd, odd/even constraint of the protocol
  - Not for typical protocols or predefined rules
  - To choose what to save and how to correlate it to future packets

Passive Testing using Network Traces

Passive Testing using Network Traces

PASSIVE TESTING USING NETWORK TRACES 101

We are stepping into an emergent / controversial / research area

Passive Testing using Network Traces 101

We are stepping into an emergent / controversial / research area

- No fully developed solutions

Passive Testing using Network Traces 101

We are stepping into an emergent / controversial / research area

- No fully developed solutions
  - Traditionally considered that too much computational power is required to store states at all time (IDS / IPS / Firewalls lost interest)

PASSIVE TESTING USING NETWORK TRACES 101

We are stepping into an emergent / controversial / research area

- No fully developed solutions
  - Traditionally considered that too much computational power is required to store states at all time (IDS / IPS / Firewalls lost interest)
  - Some prototype implementations are now trying to make its way to an industrial environment (agency for accelerated technology transfer in France or MMT tool)

PASSIVE TESTING USING NETWORK TRACES 101

We are stepping into an emergent / controversial / research area

- No fully developed solutions
  - Traditionally considered that too much computational power is required to store states at all time (IDS / IPS / Firewalls lost interest)
  - Some prototype implementations are now trying to make its way to an industrial environment (agency for accelerated technology transfer in France or MMT tool)
- Even the term passive testing…

Passive Testing using Network Traces 101

We are stepping into an emergent / controversial / research area

- No fully developed solutions
    - Traditionally considered that too much computational power is required to store states at all time (IDS / IPS / Firewalls lost interest)
    - Some prototype implementations are now trying to make its way to an industrial environment (agency for accelerated technology transfer in France or MMT tool)
- Even the term passive testing…
    - Passive testing is said to be the technique of observing the system without interactions

Passive Testing using Network Traces 101

We are stepping into an emergent / controversial / research area

- No fully developed solutions
  - Traditionally considered that too much computational power is required to store states at all time (IDS / IPS / Firewalls lost interest)
  - Some prototype implementations are now trying to make its way to an industrial environment (agency for accelerated technology transfer in France or MMT tool)
- Even the term passive testing…
  - Passive testing is said to be the technique of observing the system without interactions
  - Many purists say testing is to apply a test to a SUT and only observing is verification / checking

PASSIVE TESTING USING NETWORK TRACES 101

We are stepping into an emergent / controversial / research area

- No fully developed solutions
  - Traditionally considered that too much computational power is required to store states at all time (IDS / IPS / Firewalls lost interest)
  - Some prototype implementations are now trying to make its way to an industrial environment (agency for accelerated technology transfer in France or MMT tool)
- Even the term passive testing…
  - Passive testing is said to be the technique of observing the system without interactions
  - Many purists say testing is to apply a test to a SUT and only observing is verification / checking
  - Oxford — a test is: "A procedure intended to establish the quality, performance, or reliability of something, especially before it is taken into widespread use"

PASSIVE TESTING USING NETWORK TRACES 102

Call it how you feel it is correct…

Passive Testing using Network Traces 102

Call it how you feel it is correct…

Nonetheless, we want to guarantee that:

PASSIVE TESTING USING NETWORK TRACES 102

Call it how you feel it is correct…

Nonetheless, we want to guarantee that:

- Certain functional and non-functional requirements hold over the network traces, referred as **properties** (or rules, or invariants, more on this later)

PASSIVE TESTING USING NETWORK TRACES 102

Call it how you feel it is correct…

Nonetheless, we want to guarantee that:

- Certain functional and non-functional requirements hold over the network traces, referred as **properties** (or rules, or invariants, more on this later)
- Those requirements go beyond single packet analysis or simple associations

Passive Testing using Network Traces 102

Call it how you feel it is correct. . .

Nonetheless, we want to guarantee that:

- Certain functional and non-functional requirements hold over the network traces, referred as **properties** (or rules, or invariants, more on this later)
- Those requirements go beyond single packet analysis or simple associations

Assume the VSNP protocol and its even/odd, odd/even property

PASSIVE TESTING USING NETWORK TRACES 102

Call it how you feel it is correct…

Nonetheless, we want to guarantee that:

- Certain functional and non-functional requirements hold over the network traces, referred as **properties** (or rules, or invariants, more on this later)
- Those requirements go beyond single packet analysis or simple associations

Assume the VSNP protocol and its even/odd, odd/even property

- Let's take a look at a potential network trace to list some properties

UNDERSTANDING CORRELATED NETWORK INTERACTIONS

Assume the following trace

Understanding correlated network interactions

Assume the following trace

ID:2	**ID:3**	**ID:4**	**ID:2**	**ID:4**	**ID:21**	**ID:21**
N:	N:	N:	N: 77	N: 89	N:	N: 101

UNDERSTANDING CORRELATED NETWORK INTERACTIONS

Assume the following trace

ID:2	**ID:3**	**ID:4**	**ID:2**	**ID:4**	**ID:21**	**ID:21**
N:	N:	N:	N: 77	N: 89	N:	N: 101

Questions

Understanding correlated network interactions

Assume the following trace

ID:2	ID:3	ID:4	ID:2	ID:4	ID:21	ID:21
N:	N:	N:	N: 77	N: 89	N:	N: 101

Questions

- How can two requests / responses be together?

Understanding correlated network interactions

Assume the following trace

ID:2	ID:3	ID:4	ID:2	ID:4	ID:21	ID:21
N:	N:	N:	N: 77	N: 89	N:	N: 101

Questions

- How can two requests / responses be together?
  - The P.O. observes as packets go through, client(s) can generate n packets before response arrives to the P.O

Understanding correlated network interactions

Assume the following trace

ID:2	**ID:3**	**ID:4**	**ID:2**	**ID:4**	**ID:21**	**ID:21**
N:	N:	N:	N: 77	N: 89	N:	N: 101

Questions

- How can two requests / responses be together?
  - The P.O. observes as packets go through, client(s) can generate *n* packets before response arrives to the P.O
- What do we do with a non-replied request?

UNDERSTANDING CORRELATED NETWORK INTERACTIONS

Assume the following trace

ID:2	**ID:3**	**ID:4**	**ID:2**	**ID:4**	**ID:21**	**ID:21**
N:	N:	N:	N: 77	N: 89	N:	N: 101

Questions

- How can two requests / responses be together?
  - The P.O. observes as packets go through, client(s) can generate *n* packets before response arrives to the P.O
- What do we do with a non-replied request?
  - It depends on one characteristic, more on this later, keep it in mind…

UNDERSTANDING CORRELATED NETWORK INTERACTIONS (CONT.)

How the trace happened / How the tester should treat it

Understanding correlated network interactions (cont.)

How the trace happened / How the tester should treat it

ID:2

N:

UNDERSTANDING CORRELATED NETWORK INTERACTIONS (CONT.)

How the trace happened / How the tester should treat it

ID:2	**ID:3**
N:	N:

Understanding correlated network interactions (cont.)

How the trace happened / How the tester should treat it

ID:2	ID:3	ID:4
N:	N:	N:

UNDERSTANDING CORRELATED NETWORK INTERACTIONS (CONT.)

How the trace happened / How the tester should treat it

ID:2	**ID:3**	**ID:4**	**ID:2**
N:	N:	N:	N: 77

Understanding correlated network interactions (cont.)

How the trace happened / How the tester should treat it

ID:3	**ID:4**
N:	N:

UNDERSTANDING CORRELATED NETWORK INTERACTIONS (CONT.)

How the trace happened / How the tester should treat it

ID:3	**ID:4**	**ID:4**
N:	N:	N: 89

Understanding correlated network interactions (cont.)

How the trace happened / How the tester should treat it

ID:3	**ID:21**
N:	N:

Understanding correlated network interactions (cont.)

How the trace happened / How the tester should treat it

ID:3	**ID:21**	**ID:21**
N:	N:	N: **101**

Understanding correlated network interactions (cont.)

How the trace happened / How the tester should treat it

ID:3

N:

Understanding correlated network interactions (cont.)

How the trace happened / How the tester should treat it

ID:3

N:

Some conclusions / questions

Understanding correlated network interactions (cont.)

How the trace happened / How the tester should treat it

ID:3

N:

Some conclusions / questions

- Given the nature of properties, matching packets cannot be expressed by a regular language (I hope you know why now)

Understanding correlated network interactions (cont.)

How the trace happened / How the tester should treat it

ID:3

N:

Some conclusions / questions

- Given the nature of properties, matching packets cannot be expressed by a regular language (I hope you know why now)
  - How do we express the properties?

UNDERSTANDING CORRELATED NETWORK INTERACTIONS (CONT.)

How the trace happened / How the tester should treat it

ID:3

N:

Some conclusions / questions

- Given the nature of properties, matching packets cannot be expressed by a regular language (I hope you know why now)
  - How do we express the properties?
- Given the network interactions, each packet can represent a connection in any state (bad, very bad...)

UNDERSTANDING CORRELATED NETWORK INTERACTIONS (CONT.)

How the trace happened / How the tester should treat it

ID:3

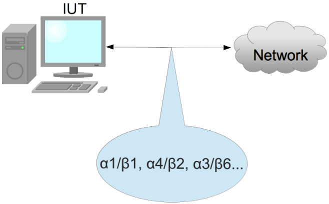
N:

Some conclusions / questions

- Given the nature of properties, matching packets cannot be expressed by a regular language (I hope you know why now)
  - How do we express the properties?
- Given the network interactions, each packet can represent a connection in any state (bad, very bad...)
  - How to avoid resource consumption?

UNDERSTANDING CORRELATED NETWORK INTERACTIONS (CONT. 2)

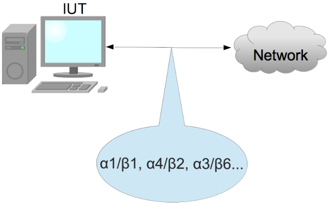
Interaction



Invariants or properties

Understanding correlated network interactions (cont. 2)

Interaction

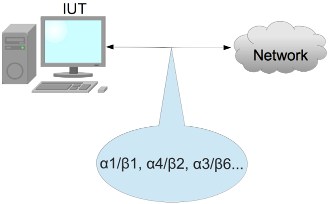


Invariants or properties

- Briefly, try to guarantee that some *test purposes* hold over the network traces

Understanding correlated network interactions (cont. 2)

Interaction

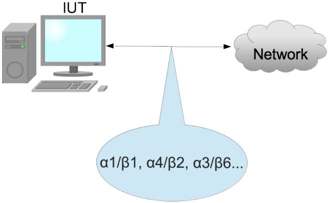


Invariants or properties

- Briefly, try to guarantee that some *test purposes* hold over the network traces
- E.g., It is not allowed to observe $\beta 6$ before an occurrence of $\alpha 4$ (this holds for our presented trace)

UNDERSTANDING CORRELATED NETWORK INTERACTIONS (CONT. 2)

Interaction



Invariants or properties

- Briefly, try to guarantee that some *test purposes* hold over the network traces
- E.g., It is not allowed to observe $\beta 6$ before an occurrence of $\alpha 4$ (this holds for our presented trace)

EXPRESSING PROPERTIES

Many languages have been proposed…

Expressing properties

Many languages have been proposed. . .

- Linear Temporal Logic (LTL)

Expressing properties

Many languages have been proposed…

- Linear Temporal Logic (LTL)
  - Describes a ω-regular language, regular language over infinite words

EXPRESSING PROPERTIES

Many languages have been proposed...

- Linear Temporal Logic (LTL)
  - Describes a ω-regular language, regular language over infinite words
  - Pretty good if assuming an interaction has a single state

EXPRESSING PROPERTIES

Many languages have been proposed…

- Linear Temporal Logic (LTL)
  - Describes a ω-regular language, regular language over infinite words
  - Pretty good if assuming an interaction has a single state
  - Not ideal if we assume many states are possible for different connections (many requests in order)

EXPRESSING PROPERTIES

Many languages have been proposed…

- Linear Temporal Logic (LTL)
  - Describes a ω-regular language, regular language over infinite words
  - Pretty good if assuming an interaction has a single state
  - Not ideal if we assume many states are possible for different connections (many requests in order)
- Languages based on CFG

Expressing properties

Many languages have been proposed…

- Linear Temporal Logic (LTL)
  - Describes a ω-regular language, regular language over infinite words
  - Pretty good if assuming an interaction has a single state
  - Not ideal if we assume many states are possible for different connections (many requests in order)
- Languages based on CFG
  - XML-based, or others…

EXPRESSING PROPERTIES

Many languages have been proposed…

- Linear Temporal Logic (LTL)
  - Describes a ω-regular language, regular language over infinite words
  - Pretty good if assuming an interaction has a single state
  - Not ideal if we assume many states are possible for different connections (many requests in order)
- Languages based on CFG
  - XML-based, or others…
  - Many proposed, adjusted to the task, but, what are the concepts behind the languages is more important

Expressing properties

Many languages have been proposed. . .

- Linear Temporal Logic (LTL)
  - Describes a ω-regular language, regular language over infinite words
  - Pretty good if assuming an interaction has a single state
  - Not ideal if we assume many states are possible for different connections (many requests in order)
- Languages based on CFG
  - XML-based, or others. . .
  - Many proposed, adjusted to the task, but, what are the concepts behind the languages is more important
  - Let's take a look at those concepts. . .

Passive Testing with Network Traces

Concepts

Based on the network traces and the desired properties that **must hold** (invariants)

PASSIVE TESTING WITH NETWORK TRACES

CONCEPTS

Based on the network traces and the desired properties that **must hold** (invariants)

- A property (or invariant) expresses (independently from the language) a sequence of premises and consequences of non-chronologically arranged, but related network packets

Passive Testing with Network Traces

Concepts

Based on the network traces and the desired properties that **must hold** (invariants)

- A property (or invariant) expresses (independently from the language) a sequence of premises and consequences of non-chronologically arranged, but related network packets
  - In English, if(A) then B (before or after), or if(if (A) then B(before or after)) then C (before or after), etc...

Passive Testing with Network Traces

Concepts

Based on the network traces and the desired properties that **must hold** (invariants)

- A property (or invariant) expresses (independently from the language) a sequence of premises and consequences of non-chronologically arranged, but related network packets
  - In English, if(A) then B (before or after), or if(if (A) then B(before or after)) then C (before or after), etc...
- A property must be able to characterize different packets

PASSIVE TESTING WITH NETWORK TRACES

CONCEPTS

Based on the network traces and the desired properties that **must hold** (invariants)

- A property (or invariant) expresses (independently from the language) a sequence of premises and consequences of non-chronologically arranged, but related network packets
    - In English, if(A) then B (before or after), or if(if (A) then B(before or after)) then C (before or after), etc...
- A property must be able to characterize different packets
    - That is, to describe a model of each packet (it uses TCP, in the port 1010, the first value is an ID, etc.)

PASSIVE TESTING WITH NETWORK TRACES

CONCEPTS

Based on the network traces and the desired properties that **must hold** (invariants)

- A property (or invariant) expresses (independently from the language) a sequence of premises and consequences of non-chronologically arranged, but related network packets
  - In English, if(A) then B (before or after), or if(if (A) then B(before or after)) then C (before or after), etc...
- A property must be able to characterize different packets
  - That is, to describe a model of each packet (it uses TCP, in the port 1010, the first value is an ID, etc.)
- A property must be able to create relationships between the different network packets

PASSIVE TESTING WITH NETWORK TRACES

CONCEPTS

Based on the network traces and the desired properties that **must hold** (invariants)

- A property (or invariant) expresses (independently from the language) a sequence of premises and consequences of non-chronologically arranged, but related network packets
  - In English, if(A) then B (before or after), or if(if (A) then B(before or after)) then C (before or after), etc...
- A property must be able to characterize different packets
  - That is, to describe a model of each packet (it uses TCP, in the port 1010, the first value is an ID, etc.)
- A property must be able to create relationships between the different network packets
  - That is, to describe how packet A relates to packet B (request port is equal to response port, etc.)

Passive Testing with Network Traces
Concepts (cont.)

How are the characterization of packets and relationships made?

Passive Testing with Network Traces

Concepts (cont.)

How are the characterization of packets and relationships made?

- Comparisons…

PASSIVE TESTING WITH NETWORK TRACES

CONCEPTS (CONT.)

How are the characterization of packets and relationships made?

- Comparisons…
  - request tcp source port = response destination port?

PASSIVE TESTING WITH NETWORK TRACES CONCEPTS (CONT.)

How are the characterization of packets and relationships made?

- Comparisons…
  - request tcp source port = response destination port?
  - Individual comparisons to constants or previous (chronological) packet

Passive Testing with Network Traces
Concepts (cont.)

How are the characterization of packets and relationships made?

- Comparisons…
  - request tcp source port = response destination port?
  - Individual comparisons to constants or previous (chronological) packet
  - The above makes relationships

PASSIVE TESTING WITH NETWORK TRACES CONCEPTS (CONT.)

How are the characterization of packets and relationships made?

- Comparisons…
  - request tcp source port = response destination port?
  - Individual comparisons to constants or previous (chronological) packet
  - The above makes relationships
  - A set of individual comparisons characterizes a packet

Passive Testing with Network Traces
Concepts (cont.)

How are the characterization of packets and relationships made?

- Comparisons…
  - request tcp source port = response destination port?
  - Individual comparisons to constants or previous (chronological) packet
  - The above makes relationships
  - A set of individual comparisons characterizes a packet

To make individual comparisons we need granular data access

Passive Testing with Network Traces Concepts (cont.)

How are the characterization of packets and relationships made?

- Comparisons…
  - request tcp source port = response destination port?
  - Individual comparisons to constants or previous (chronological) packet
  - The above makes relationships
  - A set of individual comparisons characterizes a packet

To make individual comparisons we need granular data access

- The SYN flag, of the TCP header, of *i*-th the packet

PASSIVE TESTING WITH NETWORK TRACES
CONCEPTS (CONT.)

How are the characterization of packets and relationships made?

- Comparisons…
  - request tcp source port = response destination port?
  - Individual comparisons to constants or previous (chronological) packet
  - The above makes relationships
  - A set of individual comparisons characterizes a packet

To make individual comparisons we need granular data access

- The SYN flag, of the TCP header, of *i*-th the packet
- Hierarchical as you can see…

Passive Testing with Network Traces

Concepts (cont. cont.)

Granular data access

Passive Testing with Network Traces Concepts (cont. cont.)

Granular data access

- Simple access based byte offset (TCP SYN flag is on byte offset 21 of packet, for example) is not feasible

PASSIVE TESTING WITH NETWORK TRACES CONCEPTS (CONT. CONT.)

Granular data access

- Simple access based byte offset (TCP SYN flag is on byte offset 21 of packet, for example) is not feasible
  - Many reasons, assume IHL (Internet header length), the value of it affects the offset mapping

PASSIVE TESTING WITH NETWORK TRACES CONCEPTS (CONT. CONT.)

Granular data access

- Simple access based byte offset (TCP SYN flag is on byte offset 21 of packet, for example) is not feasible
  - Many reasons, assume IHL (Internet header length), the value of it affects the offset mapping
  - In general terms, communication protocol data has semantics (meaning) behind the bytes

Passive Testing with Network Traces Concepts (cont. cont.)

Granular data access

- Simple access based byte offset (TCP SYN flag is on byte offset 21 of packet, for example) is not feasible
  - Many reasons, assume IHL (Internet header length), the value of it affects the offset mapping
  - In general terms, communication protocol data has semantics (meaning) behind the bytes
- Data access needs addressing (how to refer to the TCP SYN flag inside the TCP header of the packet)

Passive Testing with Network Traces Concepts (cont. cont.)

Granular data access

- Simple access based byte offset (TCP SYN flag is on byte offset 21 of packet, for example) is not feasible
    - Many reasons, assume IHL (Internet header length), the value of it affects the offset mapping
    - In general terms, communication protocol data has semantics (meaning) behind the bytes
- Data access needs addressing (how to refer to the TCP SYN flag inside the TCP header of the packet)

Granular data access with hierarchical key-value structure of the packet

PASSIVE TESTING WITH NETWORK TRACES CONCEPTS (CONT. CONT.)

Granular data access

- Simple access based byte offset (TCP SYN flag is on byte offset 21 of packet, for example) is not feasible
  - Many reasons, assume IHL (Internet header length), the value of it affects the offset mapping
  - In general terms, communication protocol data has semantics (meaning) behind the bytes
- Data access needs addressing (how to refer to the TCP SYN flag inside the TCP header of the packet)

Granular data access with hierarchical key-value structure of the packet

- A mapping function is needed between the raw data bytes and the structure

Passive Testing with Network Traces Concepts (cont. cont. cont.)

P packet

```
...

(TCP Header)
eb5d01bbd3e75a55cfa6
e7c0801810001cd50000

...
```

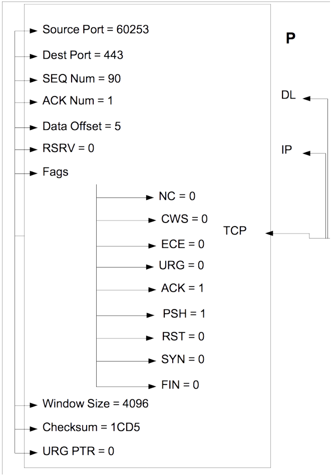
Passive Testing with Network Traces Concepts (cont. cont. cont.)

P packet

...

```
(TCP Header)
eb5d01bbd3e75a55cfa6
e7c080181000 1cd50000
```

...



Passive Testing with Network Traces
Concepts (cont. cont. cont.)

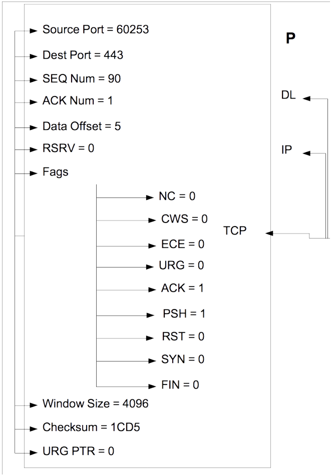
P packet

...

```
(TCP Header)
eb5d01bbd3e75a55cfa6
e7c080181001cd50000
```

...

ACK flag of TCP header of packet addressing



PASSIVE TESTING WITH NETWORK TRACES CONCEPTS (CONT. CONT. CONT.)

P packet

...

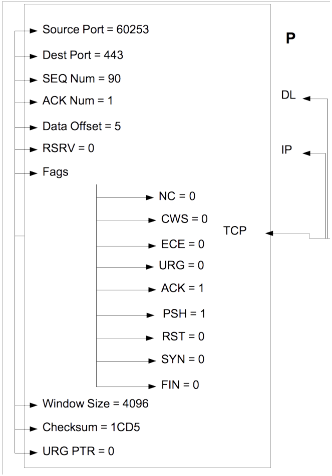
```
(TCP Header)
eb5d01bbd3e75a55cfa6
e7c0801810001cd50000
```

...

ACK flag of TCP header of packet addressing

- Many notations, assume packet is P, then value is 1 for

```
P->TCP->fags->ACK
```



P

DL

IP

TCP

Source Port = 60253

Dest Port = 443

SEQ Num = 90

ACK Num = 1

Data Offset = 5

RSRV = 0

Fags

NC = 0

CWS = 0

ECE = 0

URG = 0

ACK = 1

PSH = 1

RST = 0

SYN = 0

FIN = 0

Window Size = 4096

Checksum = 1CD5

URG PTR = 0

EXPRESSING INVARIANTS

Without a "formal" language

Expressing invariants

Without a "formal" language

- For each response with an even number a "corresponding" request with an odd ID should have been received

Expressing invariants

Without a “formal” language

- For each response with an even number a “corresponding” request with an odd ID should have been received
- For instance:

```
if RES
(
   RES->TCP->srcP = 1010 &
   RES->VSNP->Num % 2 = 0 &
   RES->IP->srcIP = REQ->IP->dstIP &
   REQ->VSNP->ID = RES->VSNP->ID &
   REQ->VSNP->ID %2 != 0
) then REQ<RES
(
   REQ->VSNP->Num = NULL
)
```

YOU ARE HERE ↓

You should know

YOU ARE HERE ↓

You should know

- The environment of passive testing using network traces

YOU ARE HERE ↓

You should know

- The environment of passive testing using network traces
- Basic concepts and how to express properties (invariants) without a formal language (which is what we will focus on)

You are here ↓

You should know

- The environment of passive testing using network traces
- Basic concepts and how to express properties (invariants) without a formal language (which is what we will focus on)

What we will discuss next Wednesday

YOU ARE HERE ↓

You should know

- The environment of passive testing using network traces
- Basic concepts and how to express properties (invariants) without a formal language (which is what we will focus on)

What we will discuss next Wednesday

- Verdicts of the properties

You are here ↓

You should know

- The environment of passive testing using network traces
- Basic concepts and how to express properties (invariants) without a formal language (which is what we will focus on)

What we will discuss next Wednesday

- Verdicts of the properties
  - Including what to do with non-replied requests :)

You are here ↓

You should know

- The environment of passive testing using network traces
- Basic concepts and how to express properties (invariants) without a formal language (which is what we will focus on)

What we will discuss next Wednesday

- Verdicts of the properties
  - Including what to do with non-replied requests :)
  - On-line vs. Off-line interpretations

YOU ARE HERE ↓

You should know

- The environment of passive testing using network traces
- Basic concepts and how to express properties (invariants) without a formal language (which is what we will focus on)

What we will discuss next Wednesday

- Verdicts of the properties
    - Including what to do with non-replied requests :)
    - On-line vs. Off-line interpretations
- Distributed architectures

YOU ARE HERE ↓

You should know

- The environment of passive testing using network traces
- Basic concepts and how to express properties (invariants) without a formal language (which is what we will focus on)

What we will discuss next Wednesday

- Verdicts of the properties
    - Including what to do with non-replied requests :)
    - On-line vs. Off-line interpretations
- Distributed architectures
- Open areas for research