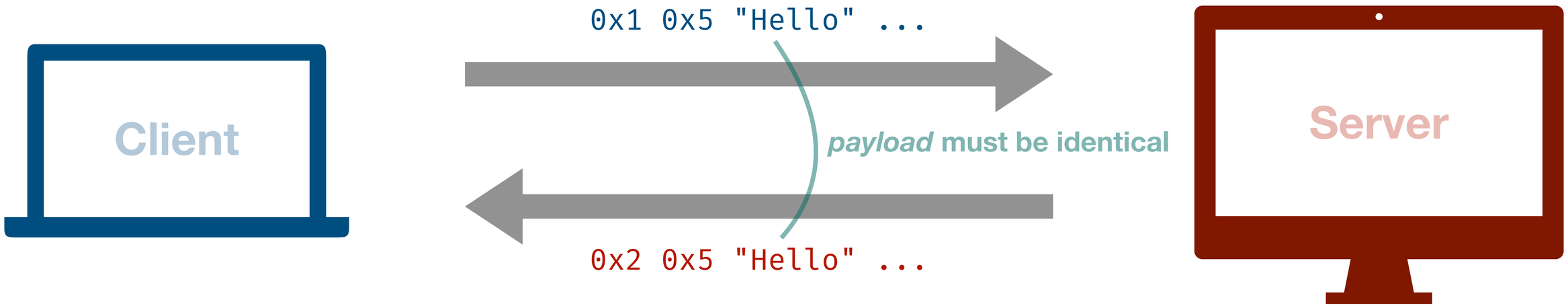
CISPA
HELMHOLTZ CENTER FOR INFORMATION SECURITY

How to Create the Nastiest Test Inputs Ever

Inputs on Demand with ISLa

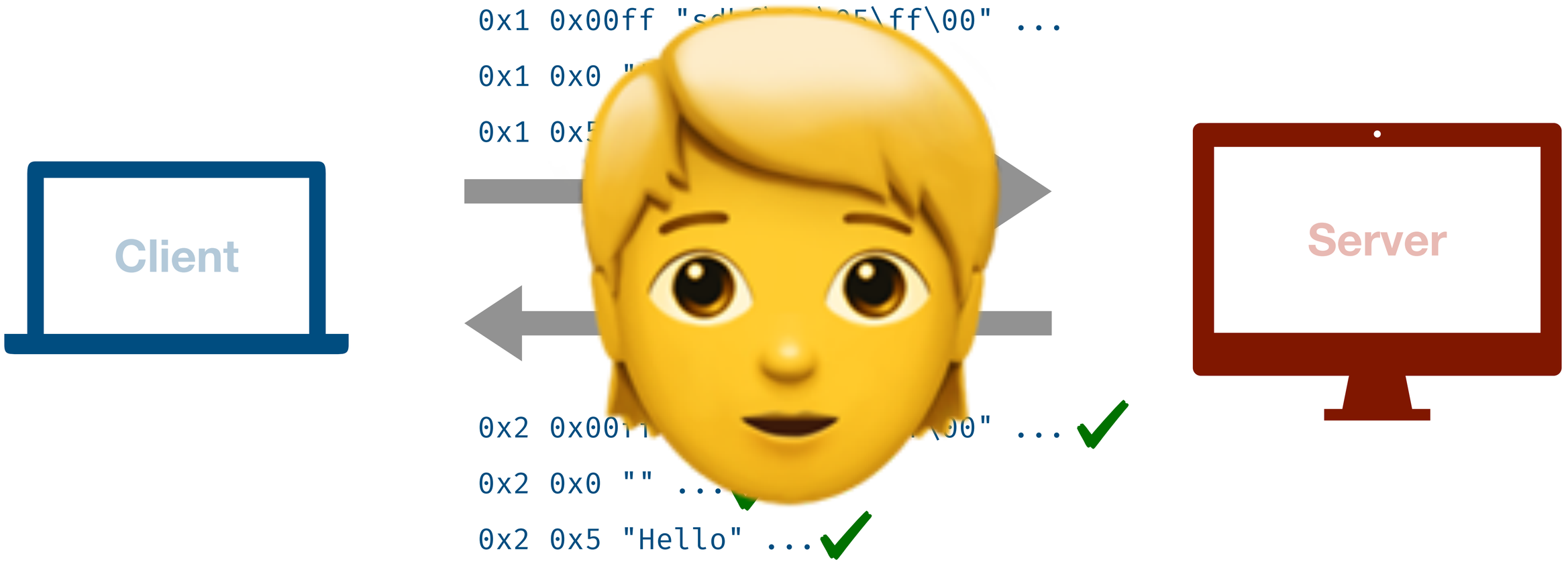
Andreas Zeller • Never work in Theory • Spring 2023

Testing a Server

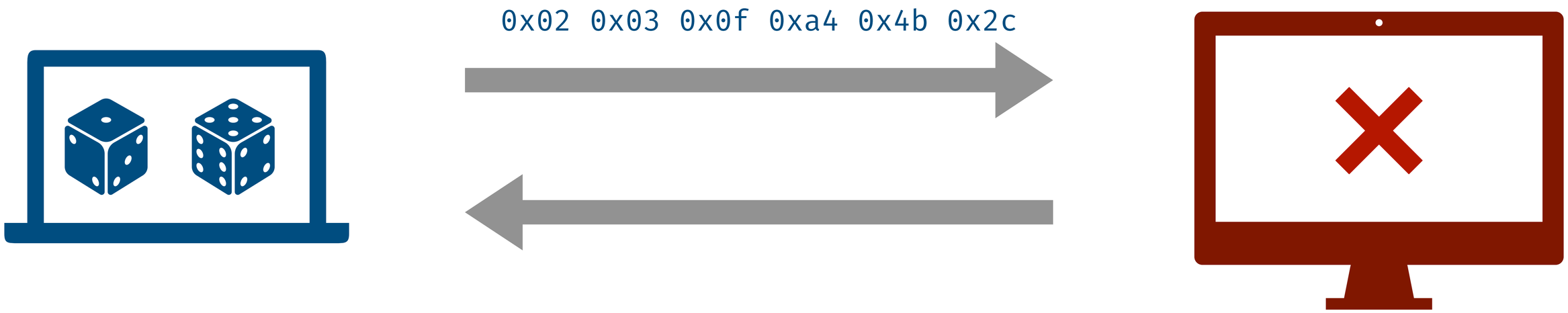


SSL/TLS Heartbeat Protocol

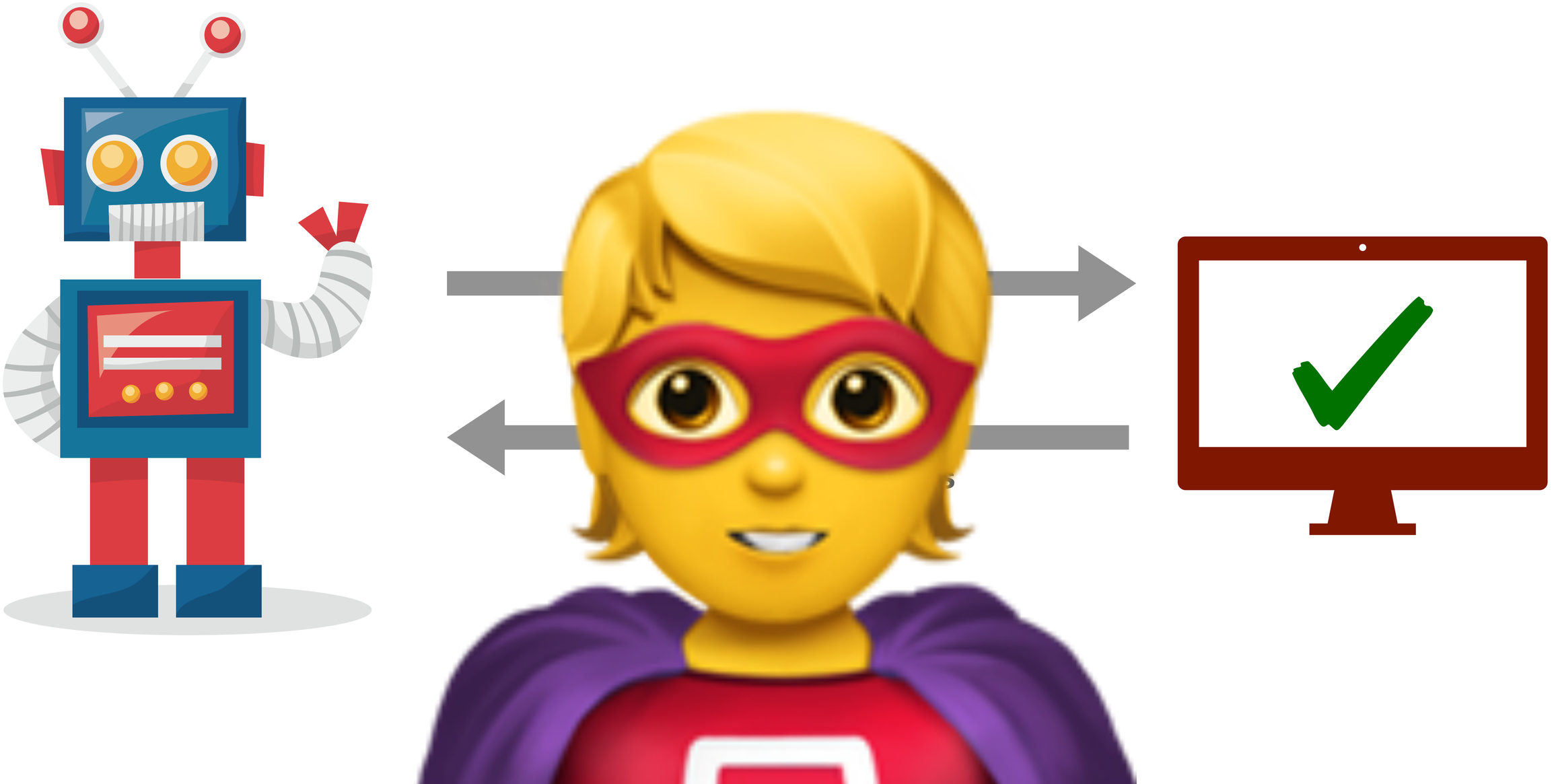
Testing with Handcrafted Inputs



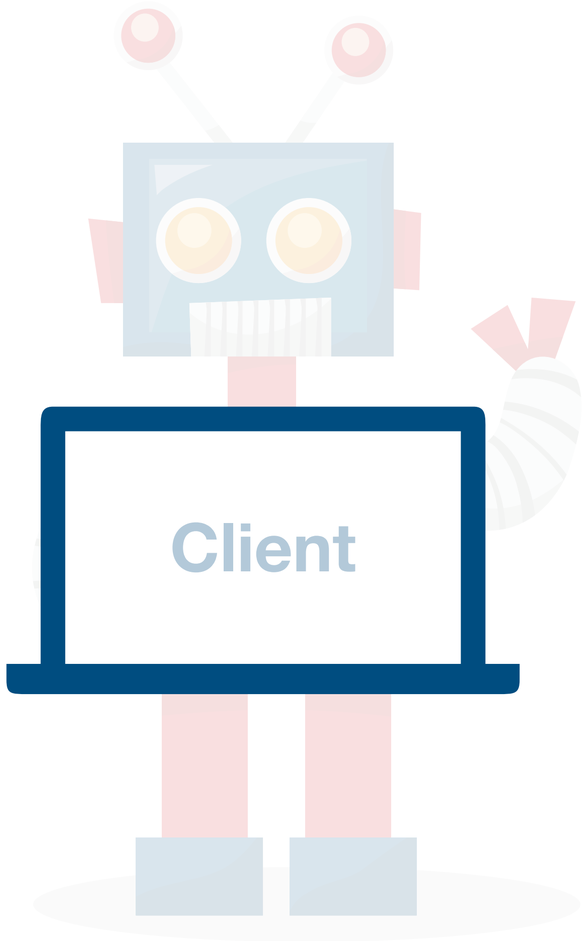
Testing with Random Inputs



How you can Become a Testing Superhero

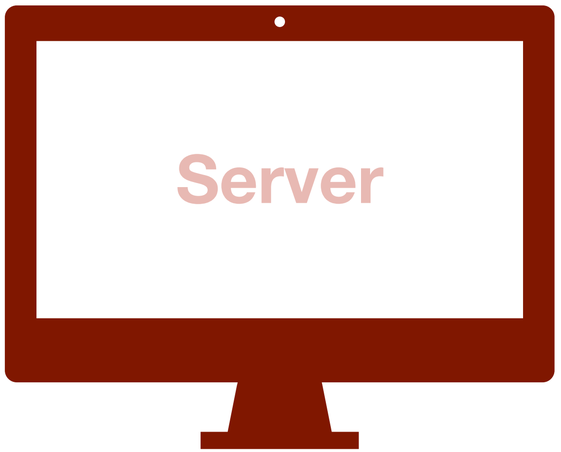


Leveraging Languages

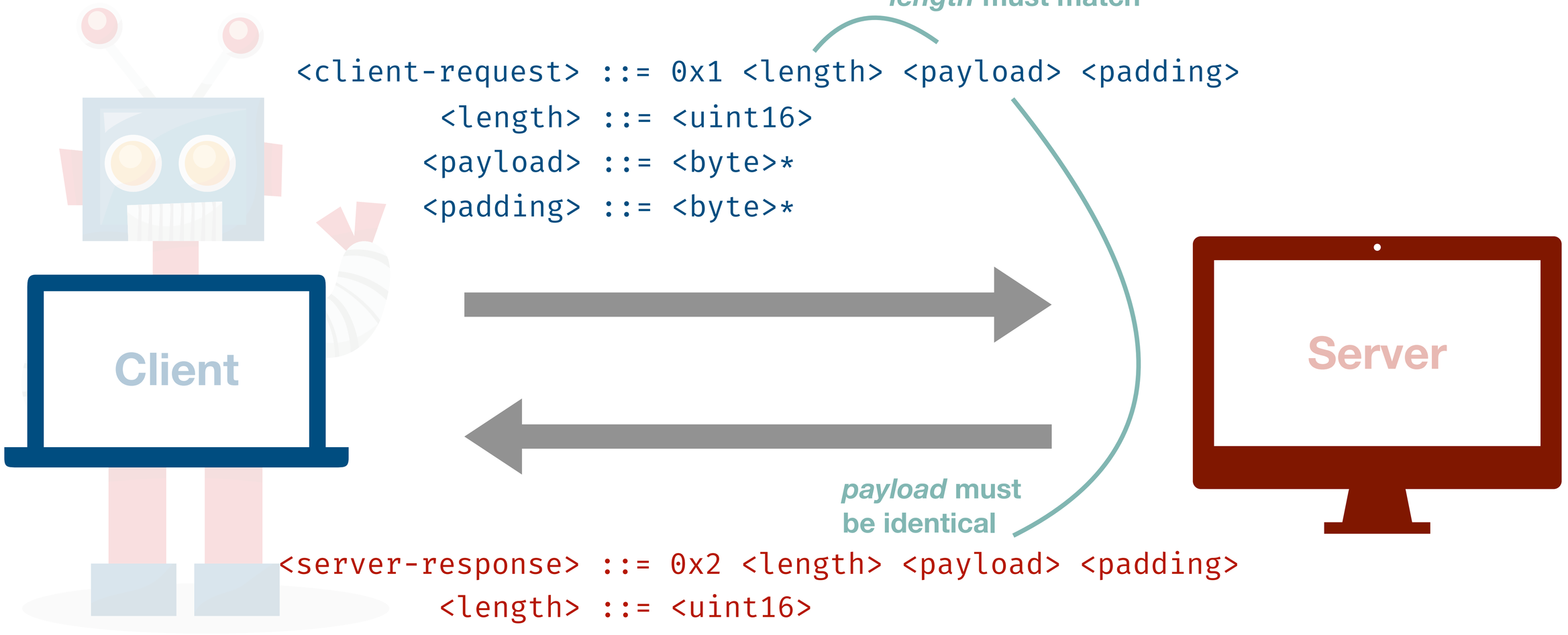


0x1 <length> <payload> <padding>

0x2 <length> <payload> <padding>



Leveraging Languages



Specifying Languages

Syntax
I/O Grammar

```
<exchange> ::= <client-request> <server-response>
<client-request> ::= 0x1 <length> <payload> <padding>
<server-response> ::= 0x2 <length> <payload> <padding>
<length> ::=
<payload>
<padding>
```

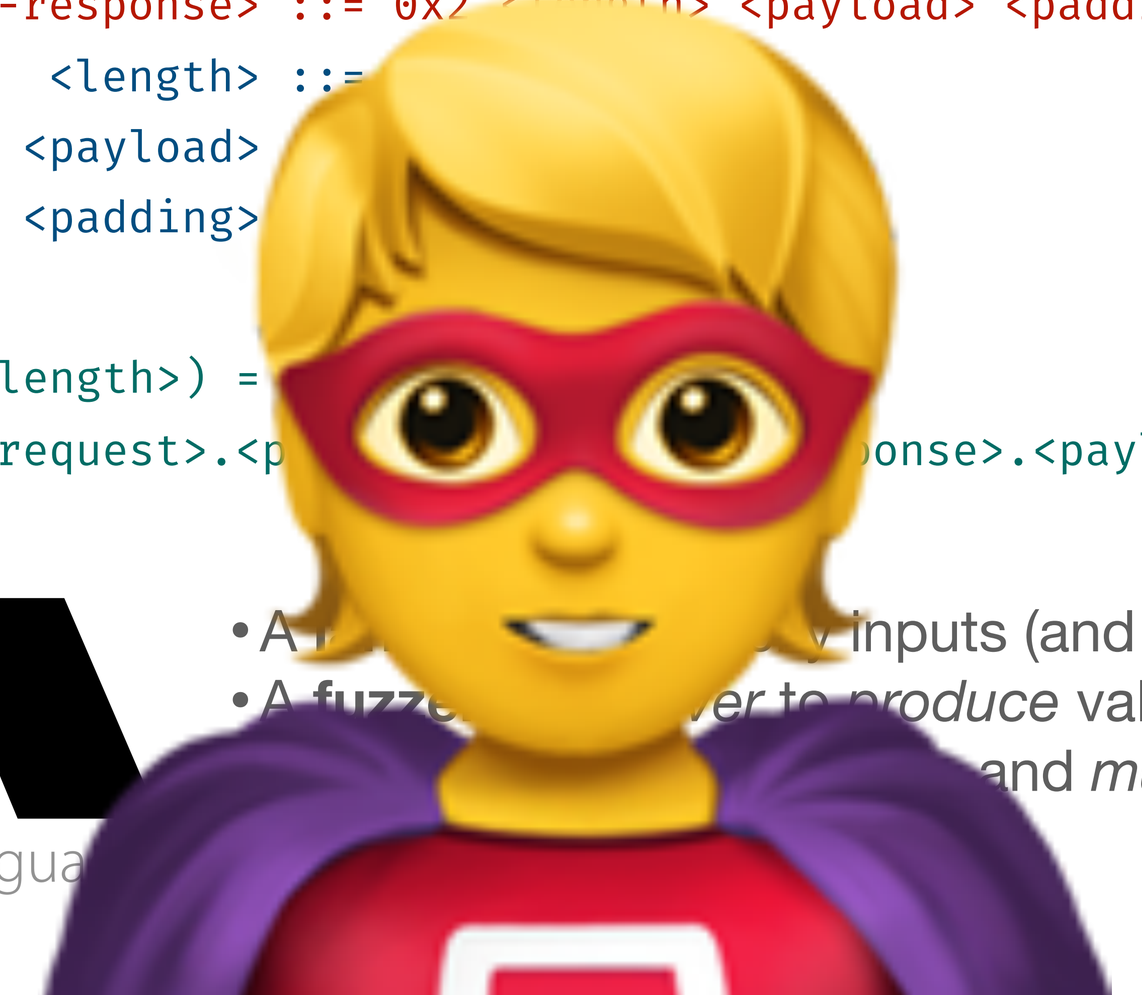
syntax alone does not suffice

Semantics
Constraints

```
uint16(<length>) =
<client-request>.<p                    onse>.<payload>
```

length must match

payload must be identical



ISLA
Input Specification Langua

- A [illegible] inputs (and outputs)
- A fuzz[illegible]er to *produce* valid inputs
  [illegible] and *mutate* following constraints

Specifying Languages

Syntax
I/O Grammar

```
<exchange> ::= <client-request> <server-response>
<client-request> ::= 0x1 <length> <payload> <padding>
<server-response> ::= 0x2 <length> <payload> <padding>
<length> ::= <uint16>
<payload> ::= <byte>*
<padding> ::= <byte>*
```

Semantics
Constraints

```
uint16(<length>) = len(<payload>)
<client-request>.<payload> = <server-response>.<payload>
```

ISLA
Input Specification Language

Producing Inputs

Syntax
I/O Grammar

```
<exchange> ::= <client-request> <server-response>
<client-request> ::= 0x1 <length> <payload> <padding>
<server-response> ::= 0x2 <length> <payload> <padding>
<length> ::= <uint16>
<payload> ::= <byte>*
<padding> ::= <byte>*
```

Semantics
Constraints

```
uint16(<length>) = len(<payload>)
<client-request>.<payload> = <server-response>.<payload>
```

ISLA
Input Specification Language

Producing Inputs

Syntax
I/O Grammar

```
<exchange> ::= <client-request> <server-response>
<client-request> ::= 0x1 <length> <payload> <padding>
<server-response> ::= 0x2 <length> <payload> <padding>
<length> ::= <uint16>
<payload> ::= <byte>*
<padding> ::= <byte>*
```

Semantics
Constraints

```
uint16(<length>) = len(<payload>)
<client-request>.<payload> = <server-response>.<payload>
```

ISLA
Input Specification Language

Producing Inputs

Syntax
I/O Grammar

```
<exchange> ::= <client-request> <server-response>
<client-request> ::= 0x1 <length> <payload> <padding>
<server-response> ::= 0x2 <length> <payload> <padding>
<length> ::= <uint16>
<payload> ::= <byte>*
<padding> ::= <byte>*
```

Semantics
Constraints

```
uint16(<length>) = len(<payload>)
<client-request>.<payload> = <server-response>.<payload>
```

ISLA
Input Specification Language

<client-request>

Producing Inputs

Syntax
I/O Grammar

```
<exchange> ::= <client-request> <server-response>
<client-request> ::= 0x1 <length> <payload> <padding>
<server-response> ::= 0x2 <length> <payload> <padding>
<length> ::= <uint16>
<payload> ::= <byte>*
<padding> ::= <byte>*
```

Semantics
Constraints

```
uint16(<length>) = len(<payload>)
<client-request>.<payload> = <server-response>.<payload>
```

ISLA
Input Specification Language

<client-request> <server-response>

Producing Inputs

Syntax
I/O Grammar

```
<exchange> ::= <client-request> <server-response>
<client-request> ::= 0x1 <length> <payload> <padding>
<server-response> ::= 0x2 <length> <payload> <padding>
<length> ::= <uint16>
<payload> ::= <byte>*
<padding> ::= <byte>*
```

Semantics
Constraints

```
uint16(<length>) = len(<payload>)
<client-request>.<payload> = <server-response>.<payload>
```

ISLA
Input Specification Language

<server-response>

Producing Inputs

Syntax
I/O Grammar

```
<exchange> ::= <client-request> <server-response>
<client-request> ::= 0x1 <length> <payload> <padding>
<server-response> ::= 0x2 <length> <payload> <padding>
<length> ::= <uint16>
<payload> ::= <byte>*
<padding> ::= <byte>*
```

Semantics
Constraints

```
uint16(<length>) = len(<payload>)
<client-request>.<payload> = <server-response>.<payload>
```

ISLA
Input Specification Language

```
0x1 <length> <payload> <padding> <server-response>
```

Producing Inputs

Syntax
I/O Grammar

```
<exchange> ::= <client-request> <server-response>
<client-request> ::= 0x1 <length> <payload> <padding>
<server-response> ::= 0x2 <length> <payload> <padding>
<length> ::= <uint16>
<payload> ::= <byte>*
<padding> ::= <byte>*
```

Semantics
Constraints

```
uint16(<length>) = len(<payload>)
<client-request>.<payload> = <server-response>.<payload>
```

ISLA
Input Specification Language

0x1 <payload> <padding> <server-response>

Producing Inputs

Syntax
I/O Grammar

```
<exchange> ::= <client-request> <server-response>
<client-request> ::= 0x1 <length> <payload> <padding>
<server-response> ::= 0x2 <length> <payload> <padding>
<length> ::= <uint16>
<payload> ::= <byte>*
<padding> ::= <byte>*
```

Semantics
Constraints

```
uint16(<length>) = len(<payload>)
<client-request>.<payload> = <server-response>.<payload>
```

ISLA
Input Specification Language

```
0x1 <uint16> <payload> <padding> <server-response>
```

Producing Inputs

Syntax
I/O Grammar

```
<exchange> ::= <client-request> <server-response>
<client-request> ::= 0x1 <length> <payload> <padding>
<server-response> ::= 0x2 <length> <payload> <padding>
<length> ::= <uint16>
<payload> ::= <byte>*
<padding> ::= <byte>*
```

Semantics
Constraints

```
uint16(<length>) = len(<payload>)
<client-request>.<payload> = <server-response>.<payload>
```

ISLA
Input Specification Language

```
0x1 0x0005 <payload> <padding> <server-response>
```

Producing Inputs

Syntax
I/O Grammar

```
<exchange> ::= <client-request> <server-response>
<client-request> ::= 0x1 <length> <payload> <padding>
<server-response> ::= 0x2 <length> <payload> <padding>
<length> ::= <uint16>
<payload> ::= <byte>*
<padding> ::= <byte>*
```

Semantics
Constraints

```
uint16(<length>) = len(<payload>)
<client-request>.<payload> = <server-response>.<payload>
```

ISLA
Input Specification Language

```
0x1 0x0005 "hello" <padding> <server-response>
```

Producing Inputs

Syntax
I/O Grammar

```
<exchange> ::= <client-request> <server-response>
<client-request> ::= 0x1 <length> <payload> <padding>
<server-response> ::= 0x2 <length> <payload> <padding>
<length> ::= <uint16>
<payload> ::= <byte>*
<padding> ::= <byte>*
```

Semantics
Constraints

```
uint16(<length>) = len(<payload>)
<client-request>.<payload> = <server-response>.<payload>
```

ISLA
Input Specification Language

complete and valid input

```
0x1 0x0005 "hello" 0x0 0x0... <server-response>
```

Parsing Outputs

Syntax
I/O Grammar

```
<exchange> ::= <client-request> <server-response>
<client-request> ::= 0x1 <length> <payload> <padding>
<server-response> ::= 0x2 <length> <payload> <padding>
<length> ::= <uint16>
<payload> ::= <byte>*
<padding> ::= <byte>*
```

Semantics
Constraints

```
uint16(<length>) = len(<payload>)
<client-request>.<payload> = <server-response>.<payload>
```

ISLA
Input Specification Language

0x2 0x0005 "hello" 0x0 0x0...

0x1 0x0005 "hello" 0x0 0x0... <server-response>

Parsing Outputs

Syntax
I/O Grammar

```
<exchange> ::= <client-request> <server-response>
<client-request> ::= 0x1 <length> <payload> <padding>
<server-response> ::= 0x2 <length> <payload> <padding>
<length> ::= <uint16>
<payload> ::= <byte>*
<padding> ::= <byte>*
```

Semantics
Constraints

```
uint16(<length>) = len(<payload>)
<client-request>.<payload> = <server-response>.<payload>
```

ISLA
Input Specification Language

```
                                    0x2  0x0005   "hello"  0x0 0x0...
0x1 0x0005 "hello" 0x0 0x0... 0x2 <length> <payload> <padding>
```

Parsing Outputs

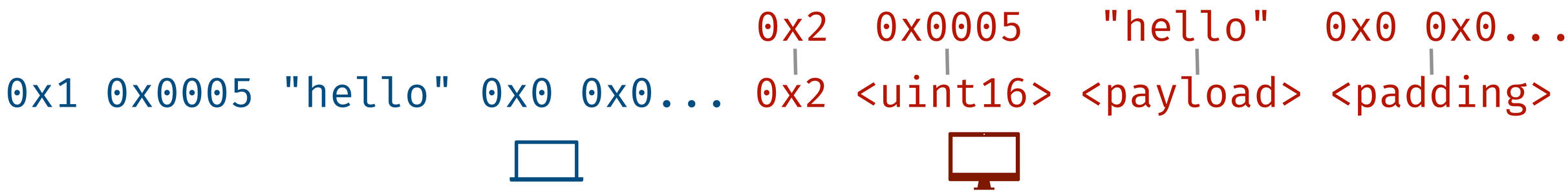
Syntax
I/O Grammar

```
<exchange> ::= <client-request> <server-response>
<client-request> ::= 0x1 <length> <payload> <padding>
<server-response> ::= 0x2 <length> <payload> <padding>
<length> ::= <uint16>
<payload> ::= <byte>*
<padding> ::= <byte>*
```

Semantics
Constraints

```
uint16(<length>) = len(<payload>)
<client-request>.<payload> = <server-response>.<payload>
```

ISLA
Input Specification Language



Parsing Outputs

Syntax
I/O Grammar

```
<exchange> ::= <client-request> <server-response>
<client-request> ::= 0x1 <length> <payload> <padding>
<server-response> ::= 0x2 <length> <payload> <padding>
<length> ::= <uint16>
<payload> ::= <byte>*
<padding> ::= <byte>*
```

Test generation problem ✓

Oracle problem ✓

Semantics
Constraints

```
uint16(<length>) = len(<payload>)
<client-request>.<payload> = <server-response>.<payload>
```

ISLA
Input Specification Language

```
0x1 0x0005 "hello" 0x0 0x0... 0x2 0x0005 "hello" 0x0 0x0...
```

Nasty Inputs: Buffer Overflows

Syntax
I/O Grammar

```
<exchange> ::= <client-request> <server-response>
<client-request> ::= 0x1 <length> <payload> <padding>
<server-response> ::= 0x2 <length> <payload> <padding>
<length> ::= <uint16>
<payload> ::= <byte>*
<padding> ::= <byte>*
```

Semantics
Constraints

```
uint16(<length>) = len(<payload>)
len(<payload>) > 1000000000
```

0x1 0x1 0x1 0x1 0x1 0x1 0x1 0x1 0x1 0x1 0x1 0x1 0x1 0x1 0x1 0x1 0x1 0x1 0x1 0x1 0x1 0x1 0x

Nasty Inputs: SQL Injections

Syntax
I/O Grammar

```
<exchange> ::= <client-request> <server-response>
<client-request> ::= 0x1 <length> <payload> <padding>
<server-response> ::= 0x2 <length> <payload> <padding>
<length> ::= <uint16>
<payload> ::= <byte>*
<padding> ::= <byte>*
```

Semantics
Constraints

```
uint16(<length>) = len(<payload>)
```

```
<client-request>.<payload> =
"'); DROP TABLE CUSTOMERS --"
```

```
INSERT INTO LOG VALUES ('payload: ');
DROP TABLE CUSTOMERS --')
```

Nasty Inputs: HTML Injections

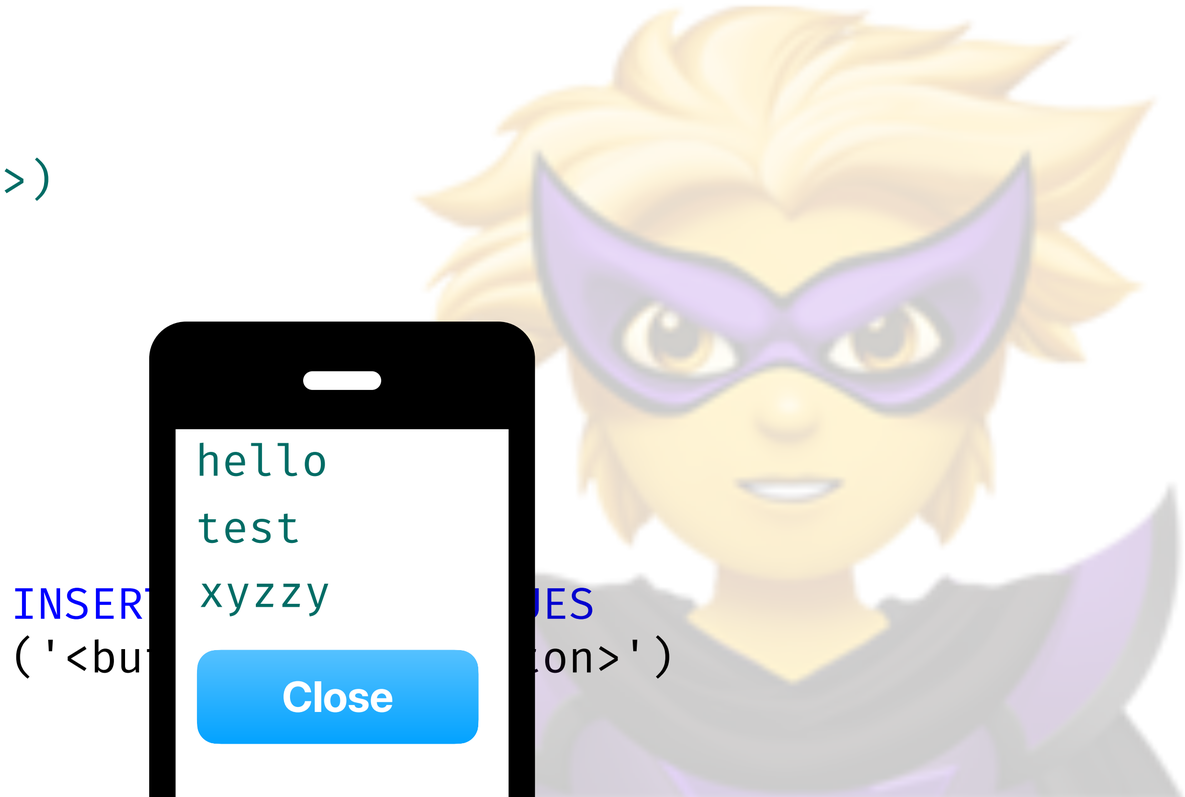
Syntax
I/O Grammar

```
<exchange> ::= <client-request> <server-response>
<client-request> ::= 0x1 <length> <payload> <padding>
<server-response> ::= 0x2 <length> <payload> <padding>
<length> ::= <uint16>
<payload> ::= <byte>*
<padding> ::= <byte>*
```

Semantics
Constraints

```
uint16(<length>) = len(<payload>)
```

`<client-request>.<payload> = "<button>Close<button>"`



Nasty Inputs: All Together

Syntax
I/O Grammar

```
<exchange> ::= <client-request> <server-response>
<client-request> ::= 0x1 <length> <payload> <padding>
<server-response> ::= 0x2 <length> <payload> <padding>
<length> ::= <uint16>
<payload> ::= <byte>*
<padding> ::= <byte>*
```

Semantics
Constraints

```
uint16(<length>) = len(<payload>)
<client-request>.<payload> = <nasty-input>
```

Nasty Inputs: All Together

Syntax
I/O Grammar

```
<exchange> ::= <client-request> <server-response>
<client-request> ::= 0x1 <length> <payload> <padding>
<server-response> ::= 0x2 <length> <payload> <padding>
<length> ::= <uint16>
<payload> ::= <byte>*
<padding> ::= <byte>*
```

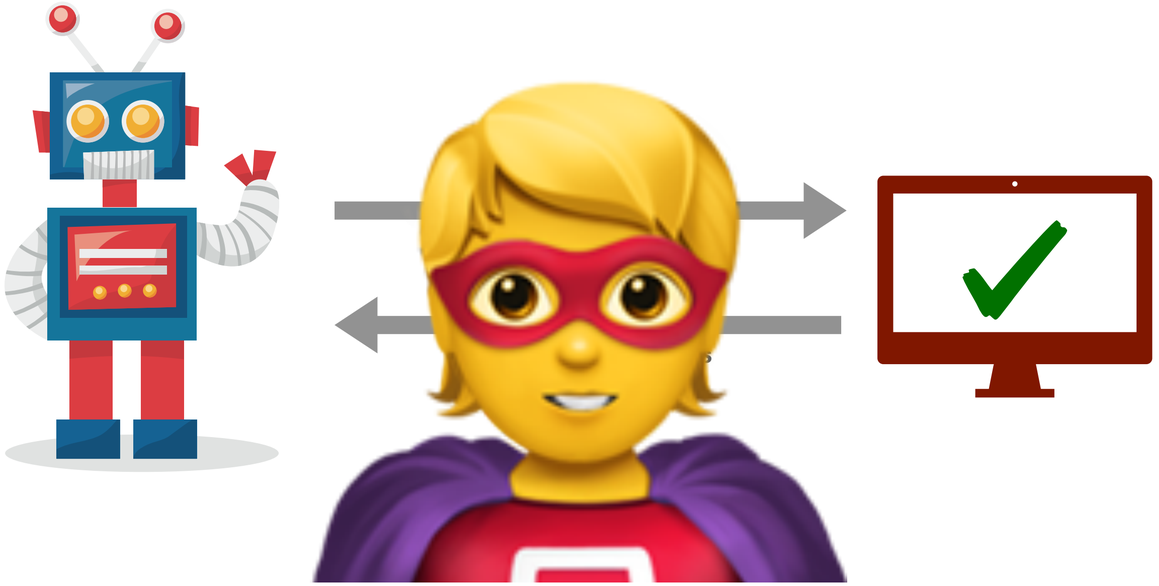
Semantics
Constraints

```
uint16(<length>) = len(<payload>)
<client-request>.<payload> = <nasty-input>
```

Nasty Inputs
Attacks

```
<nasty-input> ::= <buffer-overflow-input> |
                  <sql-injection-input> |
                  <html-injection-input> |
                  ...
```

How to Become a Testing Superhero



Testing with Language Specs

Syntax
I/O Grammar

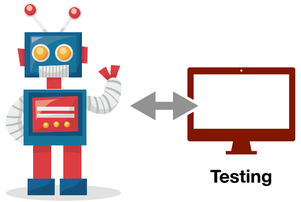
```
<exchange> ::= <client-request> <server-response>
<client-request> ::= 0x1 <length> <payload> <padding>
<server-response> ::= 0x2 <length> <payload> <padding>
<length> ::= <uint16>
<payload> ::= <byte>*
<padding> ::= <byte>*
```

Test generation problem ✓

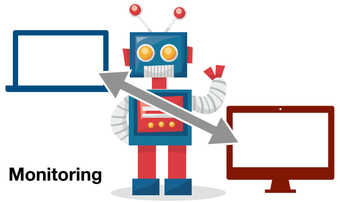
Oracle problem ✓

Semantics
Constraints

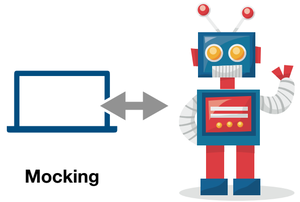
```
uint16(<length>) = len(<payload>)
<client-request>.<payload> = <server-response>.<payload>
```



Testing



Monitoring



Mocking

Nasty Inputs

Syntax
I/O Grammar

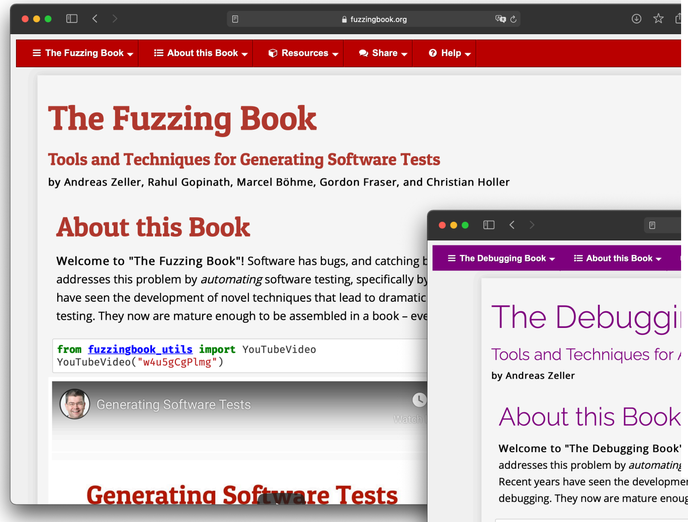
```
<exchange> ::= <client-request> <server-response>
<client-request> ::= 0x1 <length> <payload> <padding>
<server-response> ::= 0x2 <length> <payload> <padding>
<length> ::= <uint16>
<payload> ::= <byte>*
<padding> ::= <byte>*
```

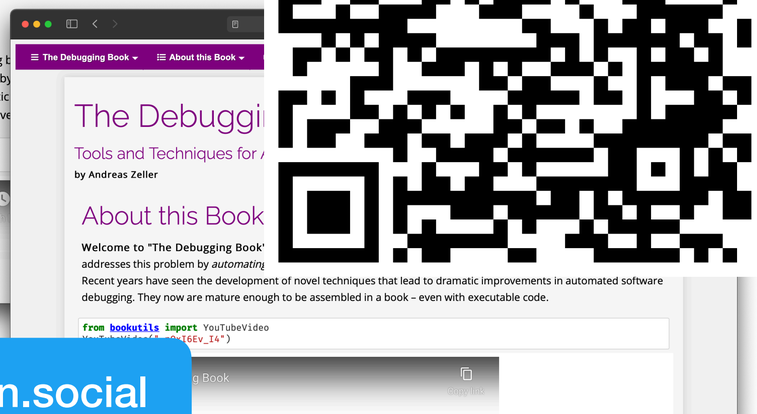
Semantics
Constraints

```
uint16(<length>) = len(<payload>)
<client-request>.<payload> = <nasty-input>
```

Nasty Inputs
Attacks

```
<nasty-input> ::= <buffer-overflow-input> |
                  <sql-injection-input> |
                  <html-injection-input> |
                  ...
```





@AndreasZeller @AndreasZeller@mastodon.social