

# expo IQA 24

MADRID  
May 28th,  
29th, 30th  
2024



[expoqqa.com](http://expoqqa.com)

Make a **fearless**  
start with  
security testing





# Who am I?

Sander van Beek

bartosz









# Sander van Beek

Home Lists About

Jan 8

## Test Automation Roadmap

Test automation is a complex topic where diverse approaches often lead to confusion and misalignment among teams. The Test Automation Roadmap introduces a structured model to help teams,...



Test Automation 6 min read



Sep 25, 2023

## Automatically detecting redundant tests

It's common for test suites to grow over time, but removing tests is rare. Because of this, some tests run every time without adding any value. All these tests do is waste time and increase the test...



Testing 7 min read



Published in ITNEXT · Nov 28, 2022

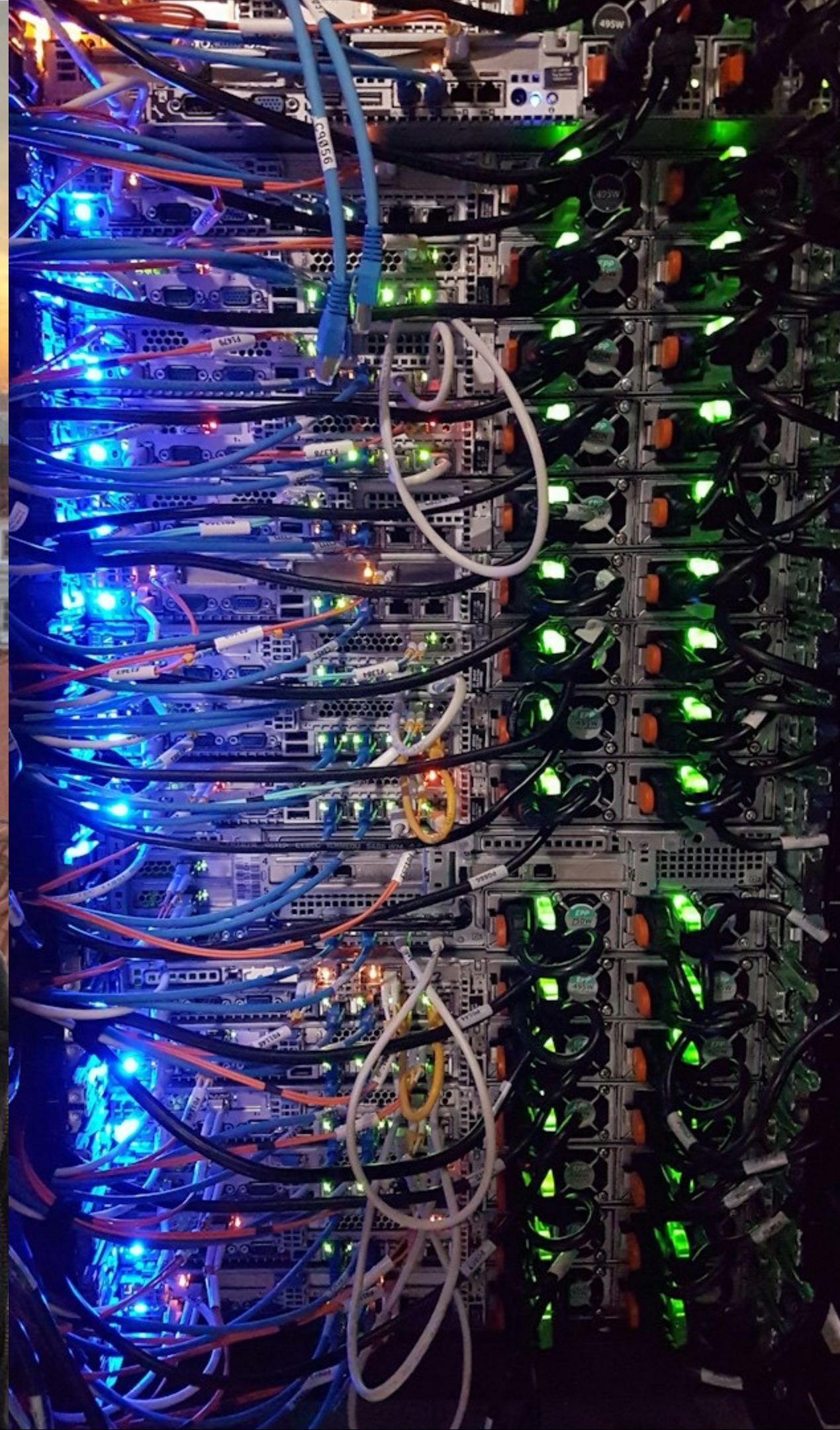
## When is software correct

Developing software should always go hand in hand with testing it. With tests, we can prove that our software works correctly. There are many ways to prove this, but all of them require us to define what...



Software Testing 5 min read





Search Write

# Sander van Beek

Home Lists About

Jan 8

## Test Automation Roadmap

Test automation is a complex topic where diverse approaches often lead to confusion and misalignment among teams. The Test Automation Roadmap introduces a structured model to help teams,...

Test Automation 6 min read

Sep 25, 2023

## Automatically detecting redundant tests

It's common for test suites to grow over time, but removing tests is rare. Because of this, some tests run every time without adding any value. All these tests do is waste time and increase the test...

Testing 7 min read

Published in ITNEXT · Nov 28, 2022

## When is software correct

Developing software should always go hand in hand with testing it. With tests, we can prove that our software works correctly. There are many ways to prove this, but all of them require us to define what...

Software Testing 5 min read

The slides will be shared



## In scope

- Getting started with security
- Assisting security experts
- Hard targets
- Security of your application

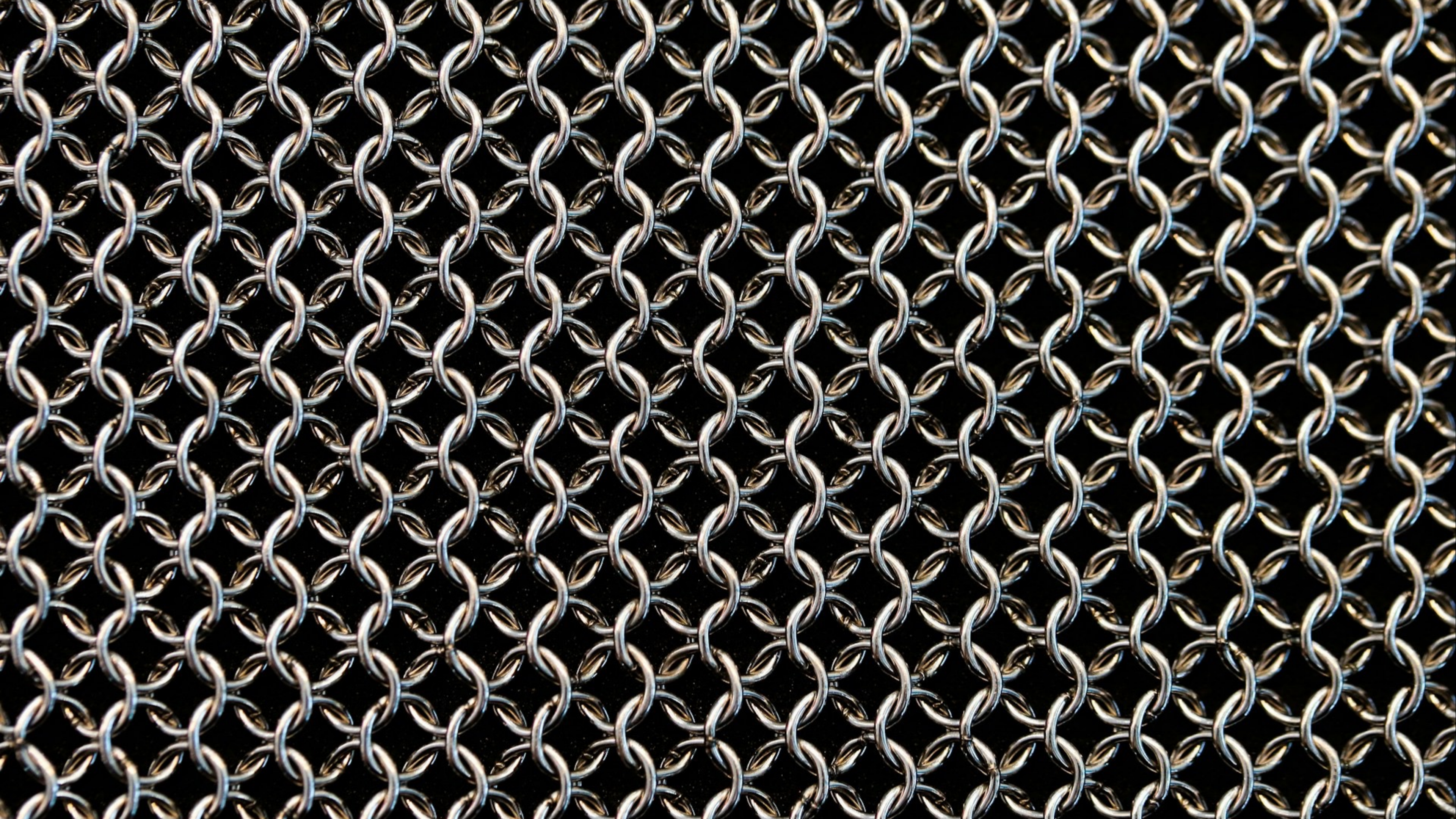
## Out of scope

- Becoming a security expert
- Hacking
- Human targets
- Organization-wide security

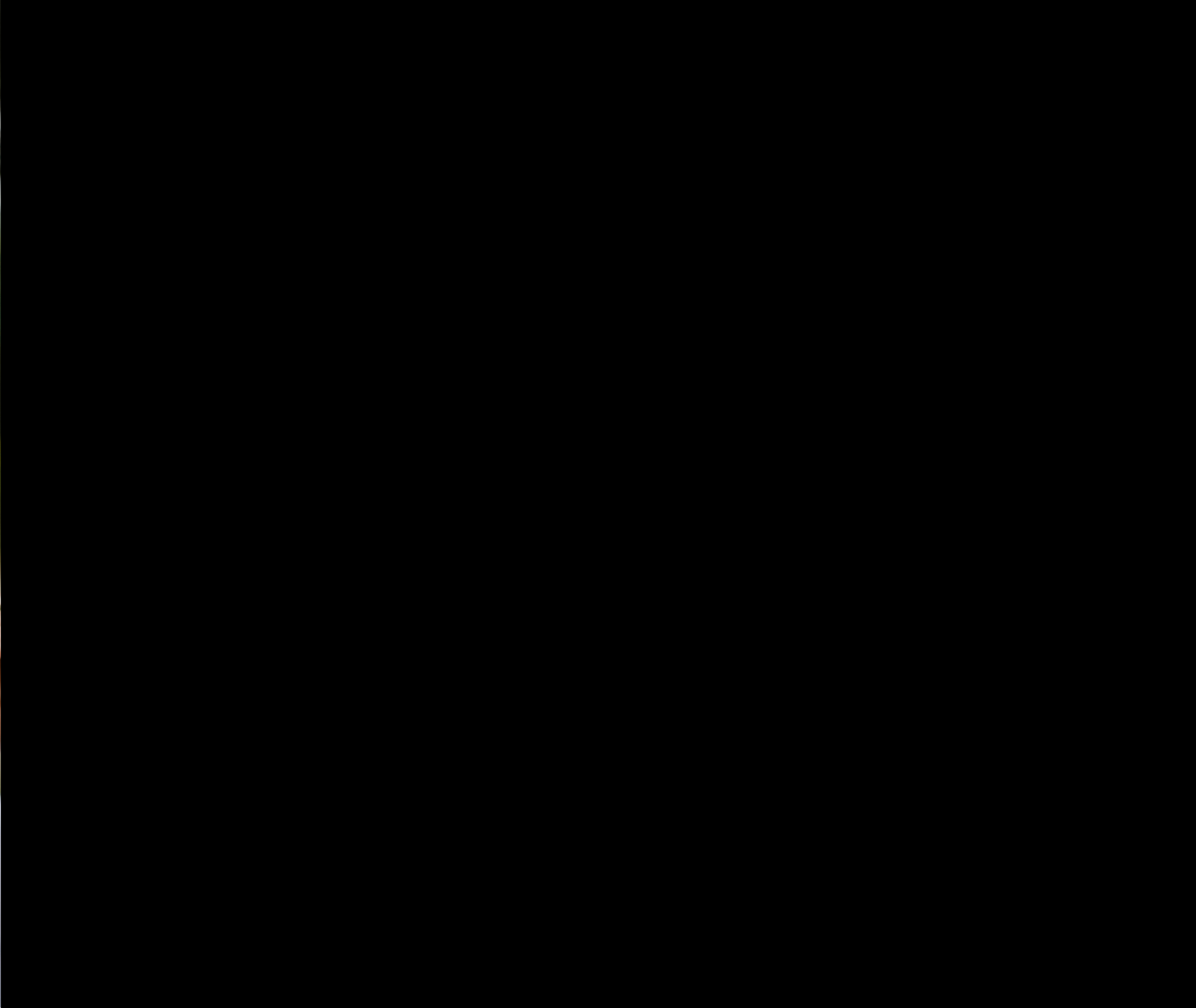
Security = Defense







Who are the attackers?











End goals



# End goals

- Money



Wana Decrypt0r 2.0

**Ooops, your files have been encrypted!** English

**What Happened to My Computer?**  
Your important files are encrypted. Many of your documents, photos, videos, databases and other files are no longer accessible because they have been encrypted. Maybe you are busy looking for a way to recover your files, but do not waste your time. Nobody can recover your files without our decryption service.

**Can I Recover My Files?**  
Sure. We guarantee that you can recover all your files safely and easily. But you have not so enough time. You can decrypt some of your files for free. Try now by clicking <Decrypt>. But if you want to decrypt all your files, you need to pay. You only have 3 days to submit the payment. After that the price will be doubled. Also, if you don't pay in 7 days, you won't be able to recover your files forever. We will have free events for users who are so poor that they couldn't pay in 6 months.

**How Do I Pay?**  
Payment is accepted in Bitcoin only. For more information, click <About bitcoin>. Please check the current price of Bitcoin and buy some bitcoins. For more information, click <How to buy bitcoins>. And send the correct amount to the address specified in this window. After your payment, click <Check Payment>. Best time to check: 9:00am - 11:00am GMT from Monday to Friday.

**Payment will be raised on**  
5/16/2017 00:47:55  
Time Left  
02:23:57:37

**Your files will be lost on**  
5/20/2017 00:47:55  
Time Left  
06:23:57:37

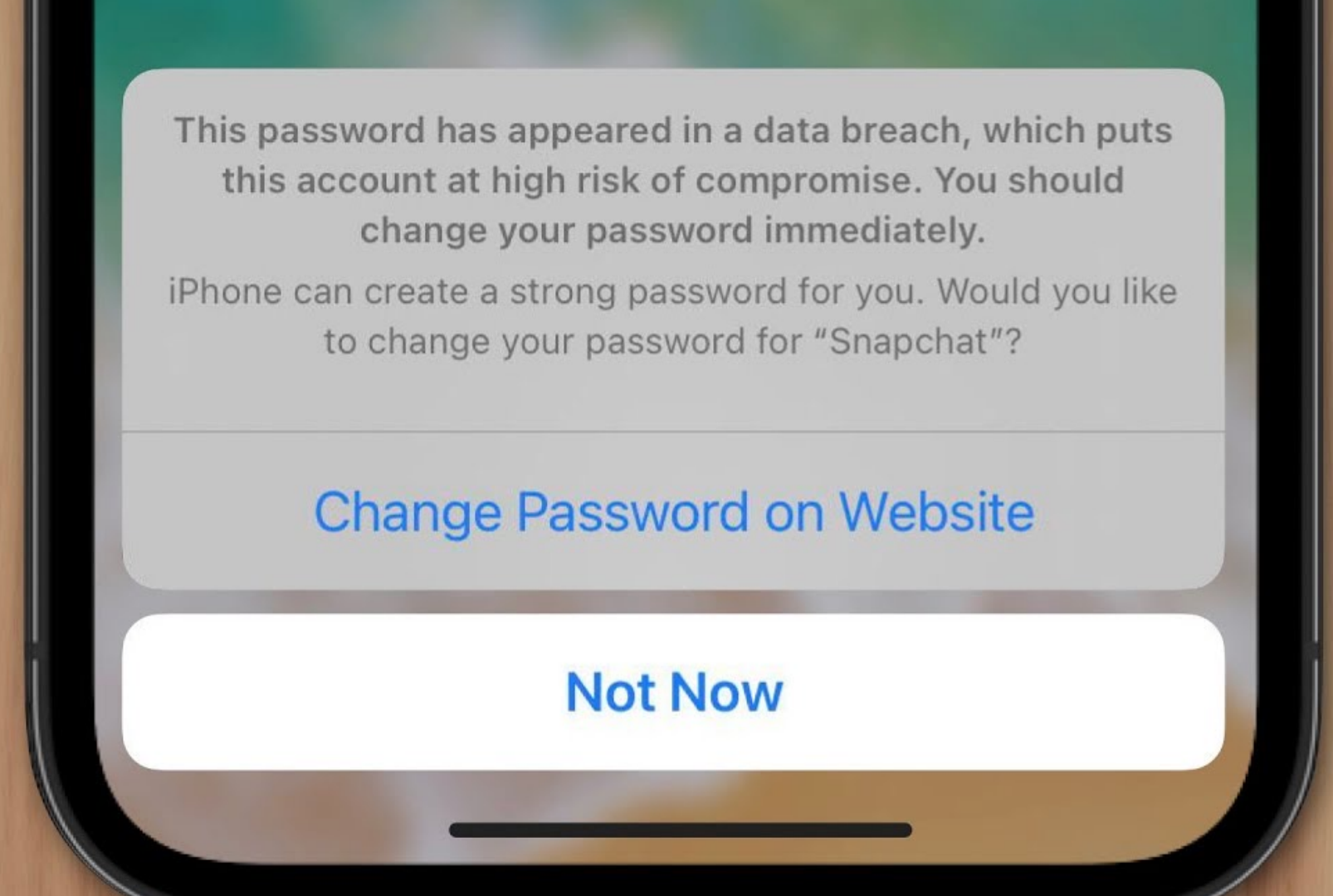
[About bitcoin](#)  
[How to buy bitcoins?](#)  
[Contact Us](#)

**Send \$300 worth of bitcoin to this address:**  
12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw Copy

**Check Payment** **Decrypt**

# End goals

- Money
- Data



# End goals

- Money
- Data
- Make you suffer



# End goals

- Money
- Data
- Make you suffer
- Thrill

***Ha Ha!***



# Intermediate goals

- How does your system work?
- Collect data
- Run their code on your machine







OWASP

O

pen

W

orldwide

A

pplication

S

ecurity

P

roject

# OWASP top 10

# OWASP top 10

- Attackers can run their code on your machine
- Attackers can get information
- Non-code attack vectors
- You can't detect attacks

Let's dive in

**Attackers** can run  
their code on your machine

# Injection



# Injection

Inject malicious code into the applications runtime.

SQL — NoSQL — JS (XSS) — XML (XPath) — OS command — ORM — LDAP — EL — ...

# Server-side



Inject malicious code into the applications runtime

*Your back-end*

# Client-side



Inject malicious code into the **user's runtime**

*Your front-end*

1-48 of over 100,000 results for "amazing"

Expand all Collapse all Results Learn about these results.

HI, THIS IS  
YOUR SON'S SCHOOL.  
WE'RE HAVING SOME  
COMPUTER TROUBLE.



OH, DEAR - DID HE  
BREAK SOMETHING?  
IN A WAY - )



DID YOU REALLY  
NAME YOUR SON  
Robert'); DROP  
TABLE Students;-- ?



OH, YES. LITTLE  
BOBBY TABLES,  
WE CALL HIM.

WELL, WE'VE LOST THIS  
YEAR'S STUDENT RECORDS.  
I HOPE YOU'RE HAPPY.



AND I HOPE  
YOU'VE LEARNED  
TO SANITIZE YOUR  
DATABASE INPUTS.

# Injection

## Mitigation

- Validate user input
- Sanitize user input

# Injection

What is sanitization?

```
<h1>amazing</h1>
```

```
&lt;h1&gt;amazing&lt;/h1&gt;
```

**amazing**



```
<h1>amazing</h1>
```



# Injection

Test for

- Sanitize form inputs
- Sanitize API input
- Sanitize front-end URLs
- Sanitize or reject file uploads

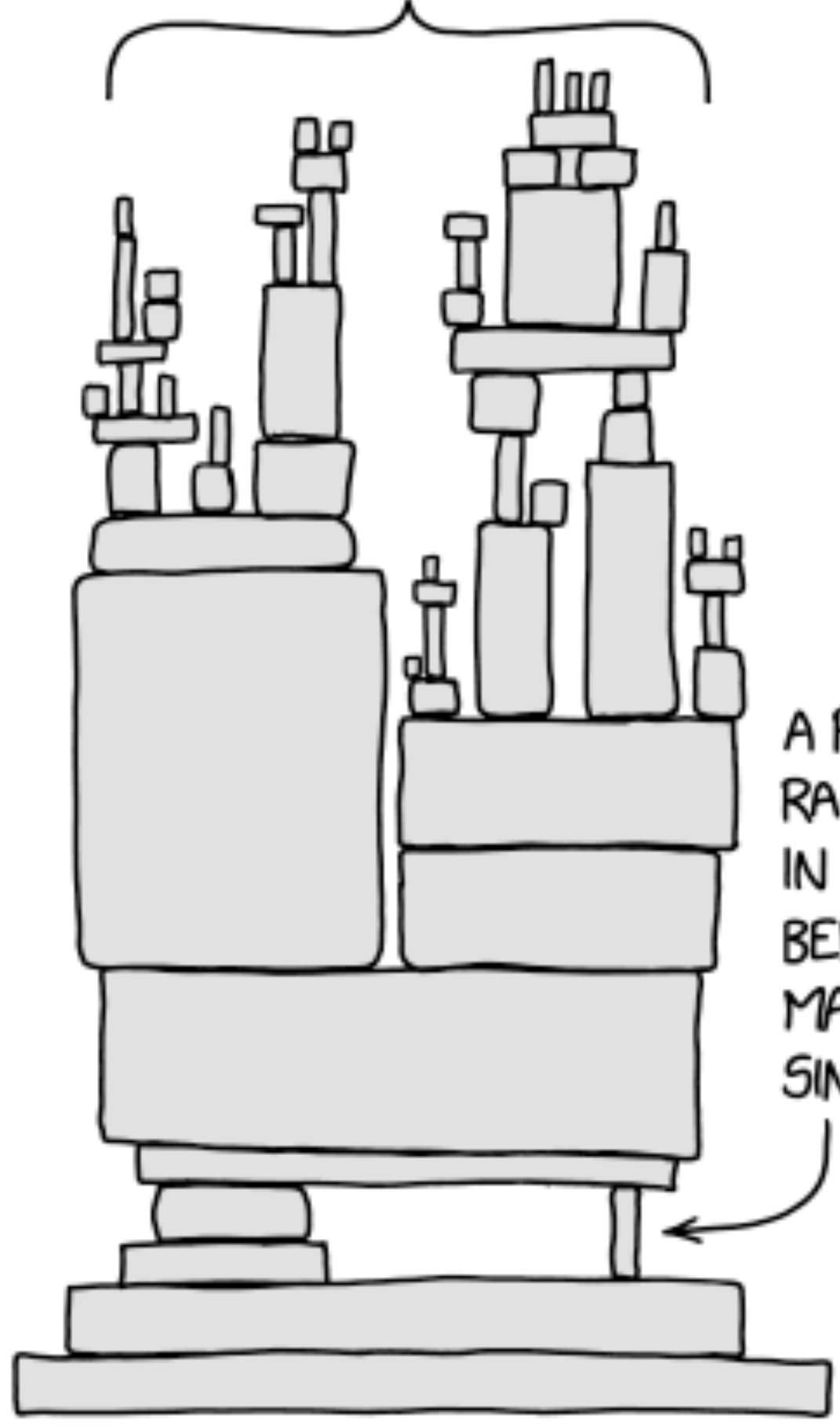
# Injection

[https://cheatsheetseries.owasp.org/cheatsheets/Injection\\_Prevention\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Injection_Prevention_Cheat_Sheet.html)



# Vulnerable and Outdated Components

ALL MODERN DIGITAL  
INFRASTRUCTURE



A PROJECT SOME  
RANDOM PERSON  
IN NEBRASKA HAS  
BEEN THANKLESSLY  
MAINTAINING  
SINCE 2003

We stand on the  
shoulders of giants

```
> npm view
```

```
http-server@14.1.1 | MIT | deps: 13 | versions: 49
```

```
A simple zero-configuration command-line http server
```

```
> npm list --omit dev
http-server@14.1.1
+-- basic-auth@2.0.1
+-- chalk@4.1.2
+-- corser@2.0.1
+-- he@1.2.0
+-- html-encoding-sniffer@3.0.0
+-- http-proxy@1.18.1
+-- mime@1.6.0
+-- minimist@1.2.6
+-- opener@1.5.2
+-- portfinder@1.0.28
+-- secure-compare@3.0.1
+-- union@0.5.0
`-- url-join@4.0.1
```

```
> npm list --omit dev --all
http-server@14.1.1
+-- basic-auth@2.0.1
| `-- safe-buffer@5.1.2
+-- chalk@4.1.2
| +-- ansi-styles@4.3.0
| | `-- color-convert@2.0.1
| |   `-- color-name@1.1.4
| `-- supports-color@7.2.0
|   `-- has-flag@4.0.0
+-- corser@2.0.1
+-- he@1.2.0
+-- html-encoding-sniffer@3.0.0
| `-- whatwg-encoding@2.0.0
|   `-- iconv-lite@0.6.3
|     `-- safer-buffer@2.1.2
...
```

# Vulnerable and Outdated Components

Test for

- Outdated dependencies
- Dependencies with known vulnerabilities
  
- Outdated infrastructure
- Infrastructure with known vulnerabilities

# Vulnerable and Outdated Components

[https://cheatsheetseries.owasp.org/cheatsheets/Vulnerable\\_Dependency\\_Management\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Vulnerable_Dependency_Management_Cheat_Sheet.html)



**Attackers** can run  
their code on your machine

**Attackers** can get information

# Broken Access Control


# Broken Access Control

Users can act outside of their intended permissions.

# Broken Access Control

- A user can see some data intended for their supervisor
- An employee can see their own customer data
- An article is made public too soon
- A page is obscured, but not secured
- ...

Missouri Governor Mike Parson is threatening legal action against a reporter and newspaper that found and responsibly disclosed a security vulnerability that left teacher and educational staffs' social security numbers exposed and easily accessible.



**Seeing employee's  
SSNs was reportedly as  
easy as clicking View  
Source**

<https://www.theverge.com/2021/10/14/22726866/missouri-governor-department-elementary-secondary-education-ssn-vulnerability-disclosure>

Access Control VS GDPR

# Broken Access Control

Test for

- Can you access things without login?
- Can you access things with wrong login?
- Can you expose unlisted data by changing URLs?



# Broken Access Control

[https://cheatsheetseries.owasp.org/cheatsheets/Authorization\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Authorization_Cheat_Sheet.html)

[https://cheatsheetseries.owasp.org/cheatsheets/Insecure\\_Direct\\_Object\\_Reference\\_Prevention\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Insecure_Direct_Object_Reference_Prevention_Cheat_Sheet.html)

# Identification and Authentication Failures

# Identification and Authentication Failures

Issues with confirmation of the user's identity.

# Identification and Authentication Failures

- Allow weak passwords
- Weak password recovery
- No/bad password encryption

# Identification and Authentication Failures

- Allow weak passwords
- Weak password recovery
- No/bad password encryption
- No multi-factor authentication
- Bad user session handling

# Identification and Authentication Failures

- Allow weak passwords
- Weak password recovery
- No/bad password encryption
- No multi-factor authentication
- Bad user session handling
- Hard-coded secrets
- Secrets in Git at some point in time



A CRYPTO NERD'S  
IMAGINATION:

HIS LAPTOP'S ENCRYPTED.  
LET'S BUILD A MILLION-DOLLAR  
CLUSTER TO CRACK IT.



A CRYPTO NERD'S  
IMAGINATION:

HIS LAPTOP'S ENCRYPTED.  
LET'S BUILD A MILLION-DOLLAR  
CLUSTER TO CRACK IT.

BLAST! OUR  
EVIL PLAN  
IS FOILED!

NO GOOD! IT'S  
4096-BIT RSA!



WHAT WOULD  
ACTUALLY HAPPEN:

HIS LAPTOP'S ENCRYPTED.  
DRUG HIM AND HIT HIM WITH  
THIS \$5 WRENCH UNTIL  
HE TELLS US THE PASSWORD.

GOT IT.





# Identification and Authentication Failures

[https://cheatsheetseries.owasp.org/cheatsheets/Authentication\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html)

[https://cheatsheetseries.owasp.org/cheatsheets/Session\\_Management\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html)

[https://cheatsheetseries.owasp.org/cheatsheets/Forgot\\_Password\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Forgot_Password_Cheat_Sheet.html)

Attackers can get information

# Non-code attack vectors

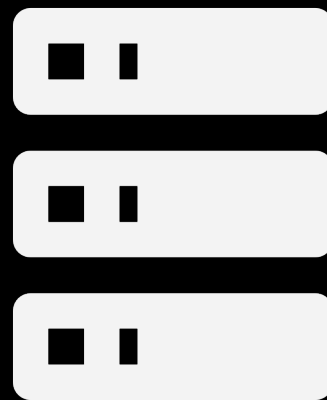
# Security Misconfiguration

# Security Misconfiguration

Secure design and tools with insecure configuration.



```
username: postgres  
password: postgres
```



192.168.178.10:443

HTTPS

192.168.178.10:80

HTTP

192.168.178.10:20

FTP

192.168.178.10:22

SSH





# Security Misconfiguration

## Mitigation

- Simplify infrastructure
- Infrastructure as code
- Periodically review and update configuration

# Security Misconfiguration

[https://cheatsheetseries.owasp.org/cheatsheets/Infrastructure\\_as\\_Code\\_Security\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Infrastructure_as_Code_Security_Cheat_Sheet.html)  
[https://cheatsheetseries.owasp.org/cheatsheets/Docker\\_Security\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Docker_Security_Cheat_Sheet.html)

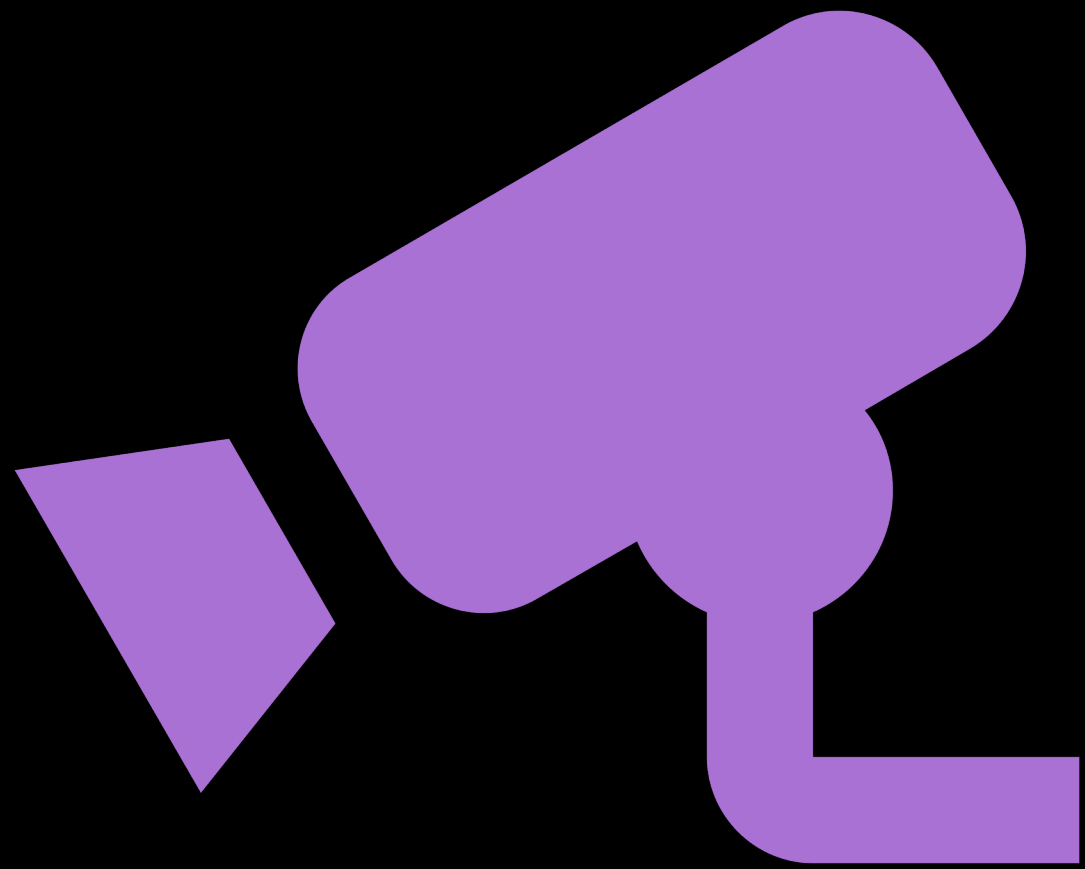
Non-code attack vectors

You can't detect attacks

# Security Logging and Monitoring Failures

# Security Logging and Monitoring Failures

Unable to detect, escalate, and respond to active breaches.



# Security Logging and Monitoring Failures

## Mitigation

- Log events such as login, important transactions, etc.
- Prevent log tampering
- Send logs to central logging platform
- Give security experts access to your logs



# Security Logging and Monitoring Failures

[https://cheatsheetseries.owasp.org/cheatsheets/Logging\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Logging_Cheat_Sheet.html)

[https://cheatsheetseries.owasp.org/cheatsheets/Logging\\_Vocabulary\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Logging_Vocabulary_Cheat_Sheet.html)

You can't detect attacks

## Attackers can run their code on your machine

- Injection
- Software and Data Integrity Failures
- Server-side Request Forgery
- Vulnerable and Outdated Components

## Attackers can get information

- Broken access control
- Cryptographic Failures
- Identification and Authentication failures

## Non-code attack vectors

- Insecure design
- Security Misconfiguration

## You can't detect attacks

- Security Logging and Monitoring Failures

## Test for

### Attackers can run their code on your machine

- Injection
- Software and Data Integrity Failures
- Server-side Request Forgery
- Vulnerable and Outdated Components

### Attackers can get information

- Broken access control
- Cryptographic Failures
- Identification and Authentication failures

### Non-code attack vectors

- Insecure design
- Security Misconfiguration

### You can't detect attacks

- Security Logging and Monitoring Failures

# Test for

- User input sanitization
  - GUI
  - API
  - URL

# Test for

- User input sanitization
- Broken access control
  - With account
  - Without account

# Test for

- User input sanitization
- Broken access control
- Outdated dependencies

# Test for

- User input sanitization
  - GUI
  - API
  - URL
- Broken access control
  - With account
  - Without account
- Outdated dependencies



# Discuss with developers

## Attackers can run their code on your machine

- Injection
- Software and Data Integrity Failures
- Server-side Request Forgery
- Vulnerable and Outdated Components

## Attackers can get information

- Broken access control
- Cryptographic Failures
- Identification and Authentication failures

## Non-code attack vectors

- Insecure design
- Security Misconfiguration

## You can't detect attacks

- Security Logging and Monitoring Failures

# Discuss with developers

- Reduce dependencies
- Subscribe to dependency vulnerability alerts
- Remove secrets from code
- Secure runtime configuration
- Enable security monitoring

What about  
test automation security?

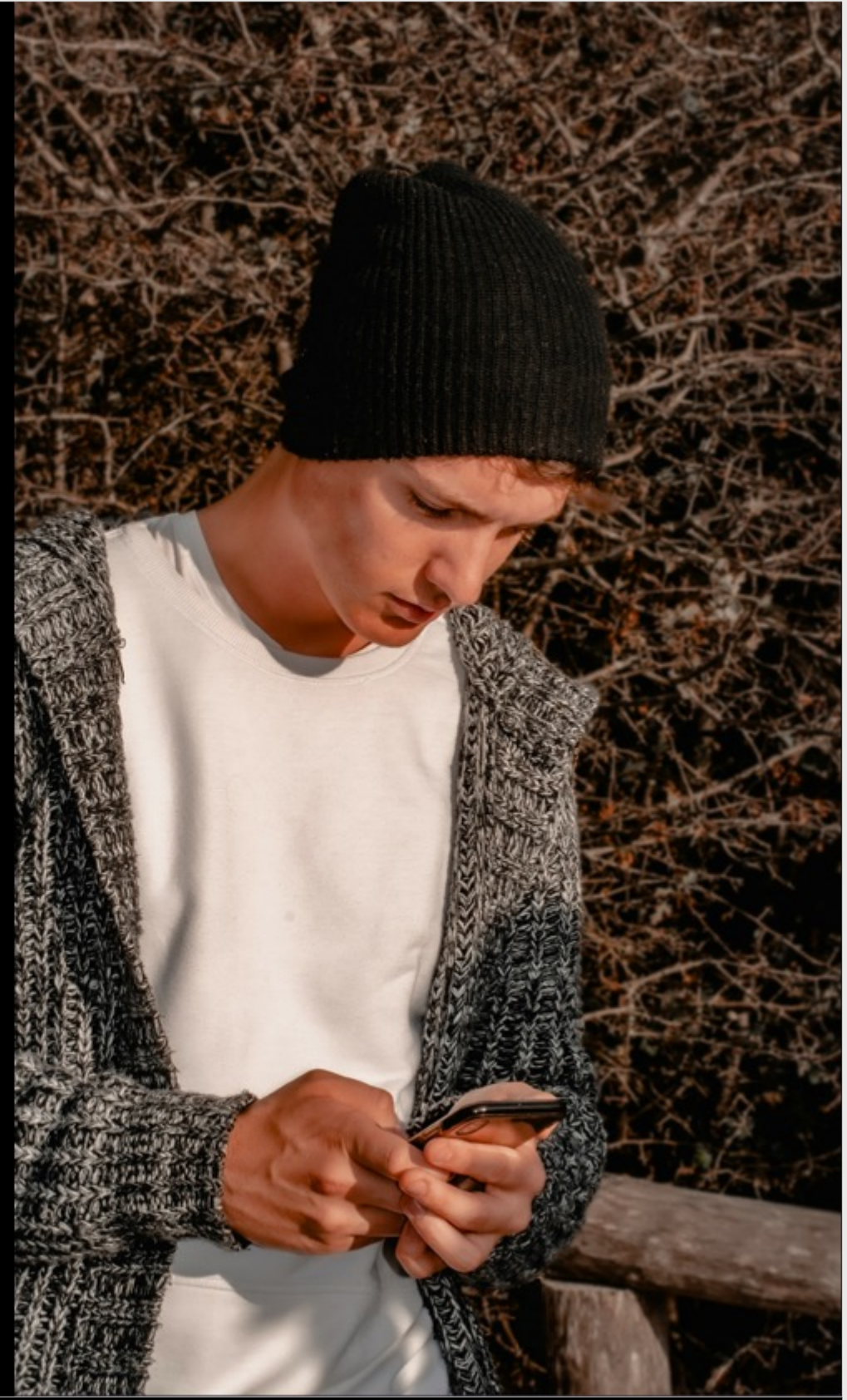
Is test automation  
security even useful?

Yes

Make security a habit

# Intermediate goals

- How does your system work?
- Collect data
- Run their code on your machine



# Test automation security

- Reduce dependencies
- Subscribe to dependency vulnerability alerts
- Remove secrets from code
- Secure runtime configuration
- ~~Enable security monitoring~~

# Production

- Reduce dependencies
- Subscribe to dependency vulnerability alerts
- Remove secrets from code
- Secure runtime configuration
- Enable security monitoring

# Test automation

- Reduce dependencies
- Subscribe to dependency vulnerability alerts
- Remove secrets from code
- Secure runtime configuration



## Production

## Test automation

- Reduce dependencies

- Subscribe to dependency

vul

Make security a habit

- Secure runtime configuration

- Enable security monitoring

- Reduce dependencies

- Subscribe to dependency

- Secure runtime configuration

# Summary

Security is defense



# Hat colors



# OWASP top 10

## Attackers can run their code on your machine

- Injection
- Software and Data Integrity Failures
- Server-side Request Forgery
- Vulnerable and Outdated Components

## Attackers can get information

- Broken access control
- Cryptographic Failures
- Identification and Authentication failures

## Non-code attack vectors

- Insecure design
- Security Misconfiguration

## You can't detect attacks

- Security Logging and Monitoring Failures

# Test for

- User input sanitization
  - GUI
  - API
  - URL
- Broken access control
  - With account
  - Without account
- Outdated dependencies

# Discuss with developers

- Reduce dependencies
- Subscribe to dependency vulnerability alerts
- Remove secrets from code
- Secure runtime configuration
- Enable security monitoring

Make security a habit



What are **you** going to  
test **next week**?

Questions?



# References and slides

<https://lakitna.nl/talks/2024/expoqa>

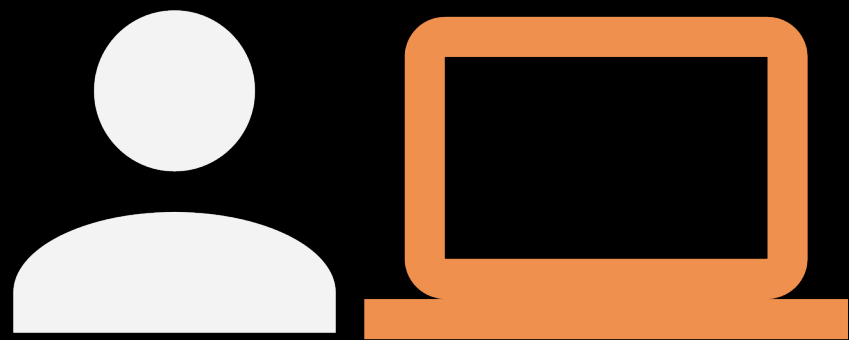


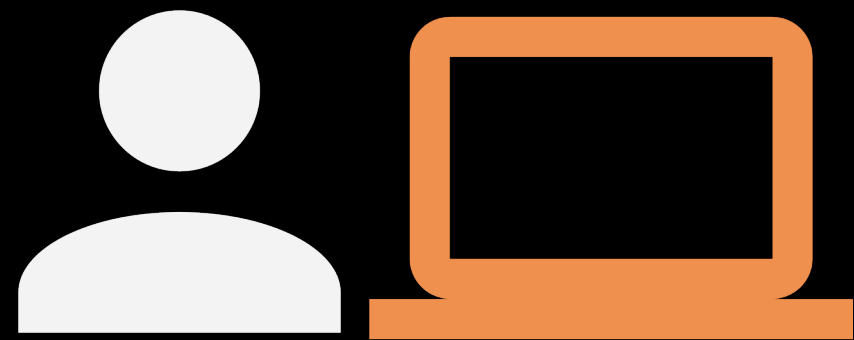
**bartosz**

expo:QA<sup>®</sup>

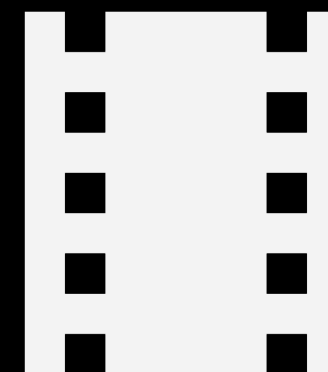


# Software and Data Integrity Failures



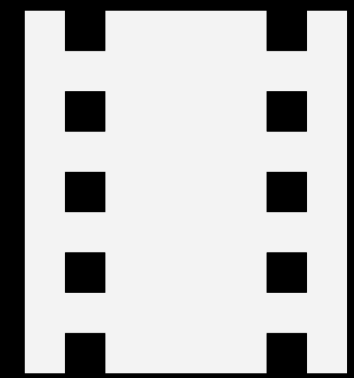


*Movie please!*



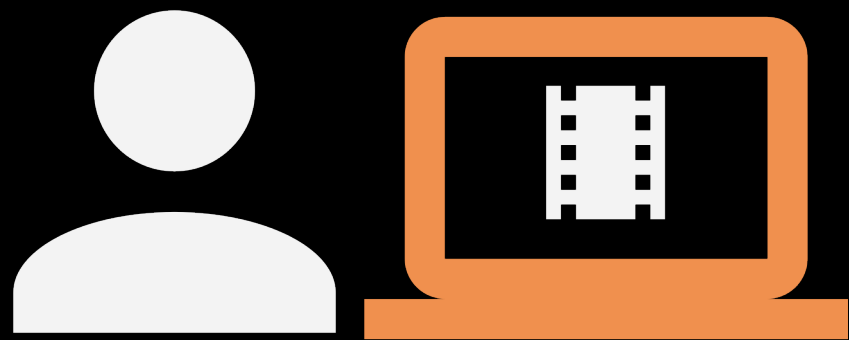


*Sure, have fun*



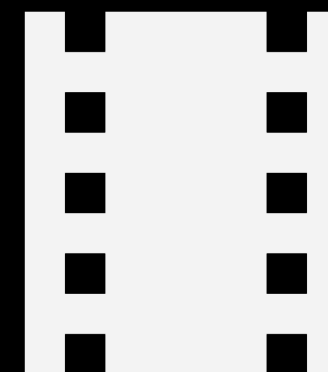


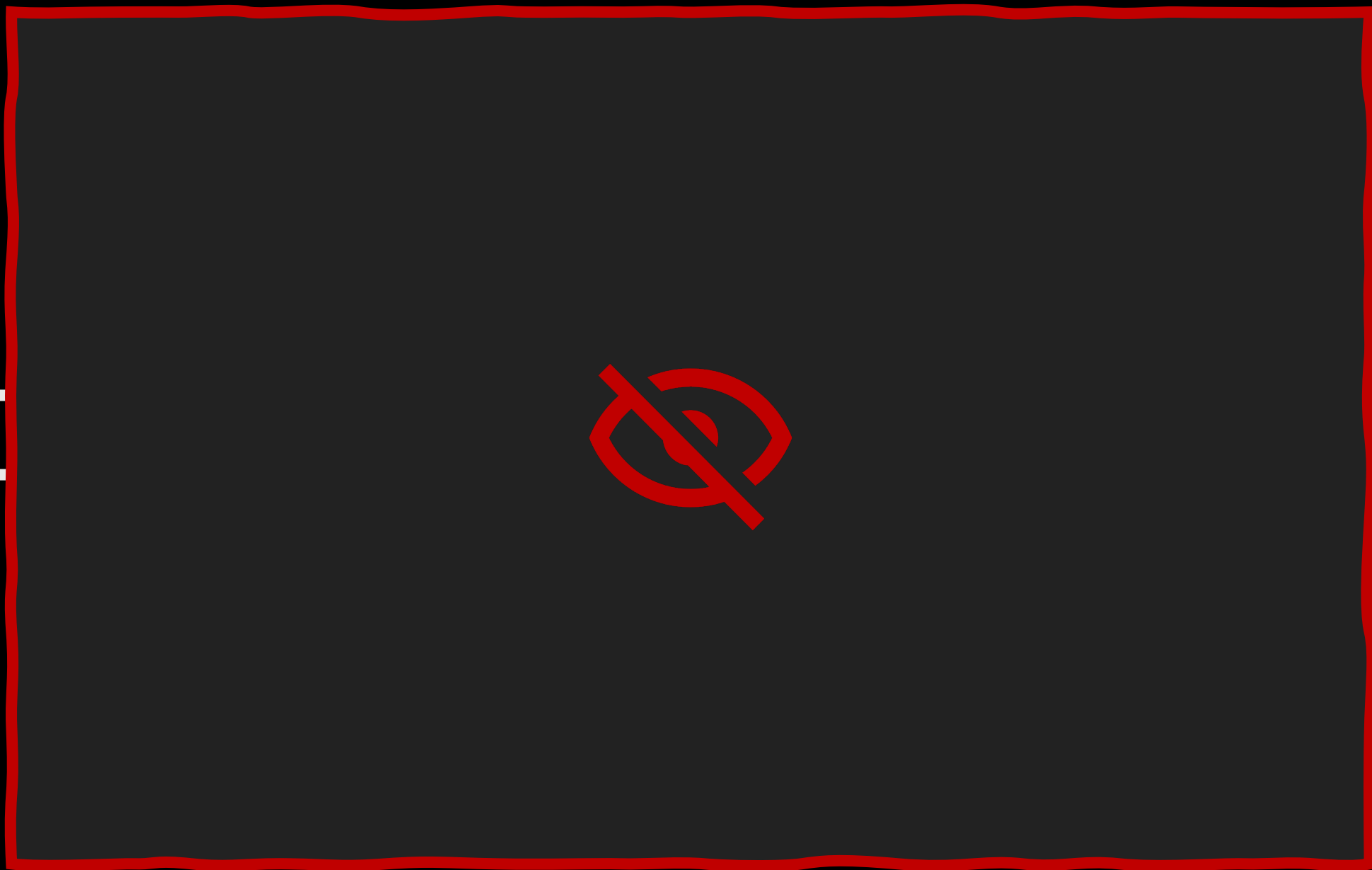
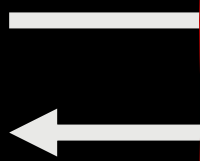




or







# Software and Data Integrity Failures

Code and infrastructure that does not protect against integrity violations.

## Examples

- Using dependencies of untrusted sources
- Auto-update from untrusted source

# Software and Data Integrity Failures

## Mitigation

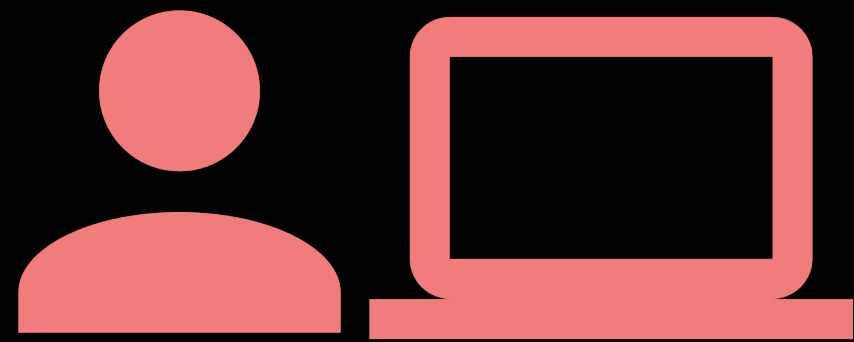
- Integrity checks
- Download from trusted repositories
- Review downloaded code

# Software and Data Integrity Failures

[https://owasp.org/Top10/A08\\_2021-Software\\_and\\_Data\\_Integrity\\_Failures/#references](https://owasp.org/Top10/A08_2021-Software_and_Data_Integrity_Failures/#references)

# Server-side Request Forgery

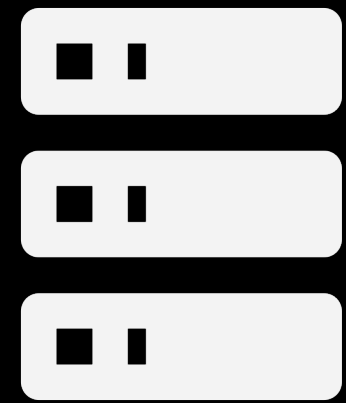


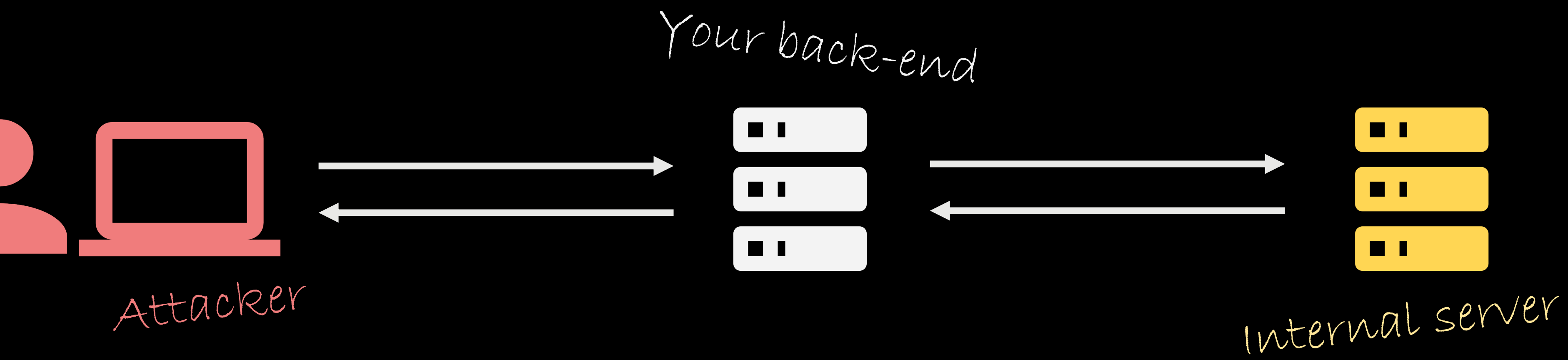


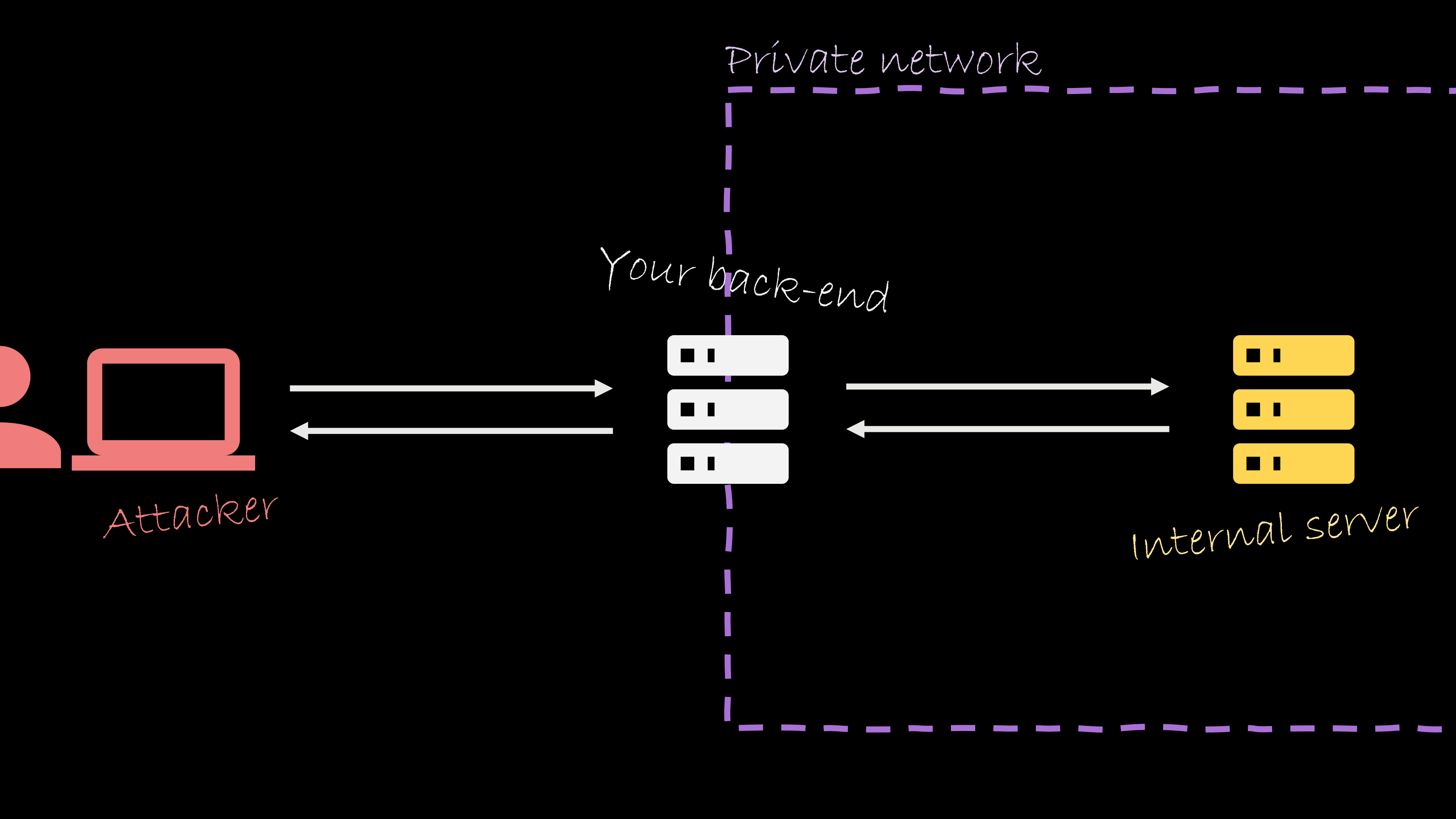
Attacker

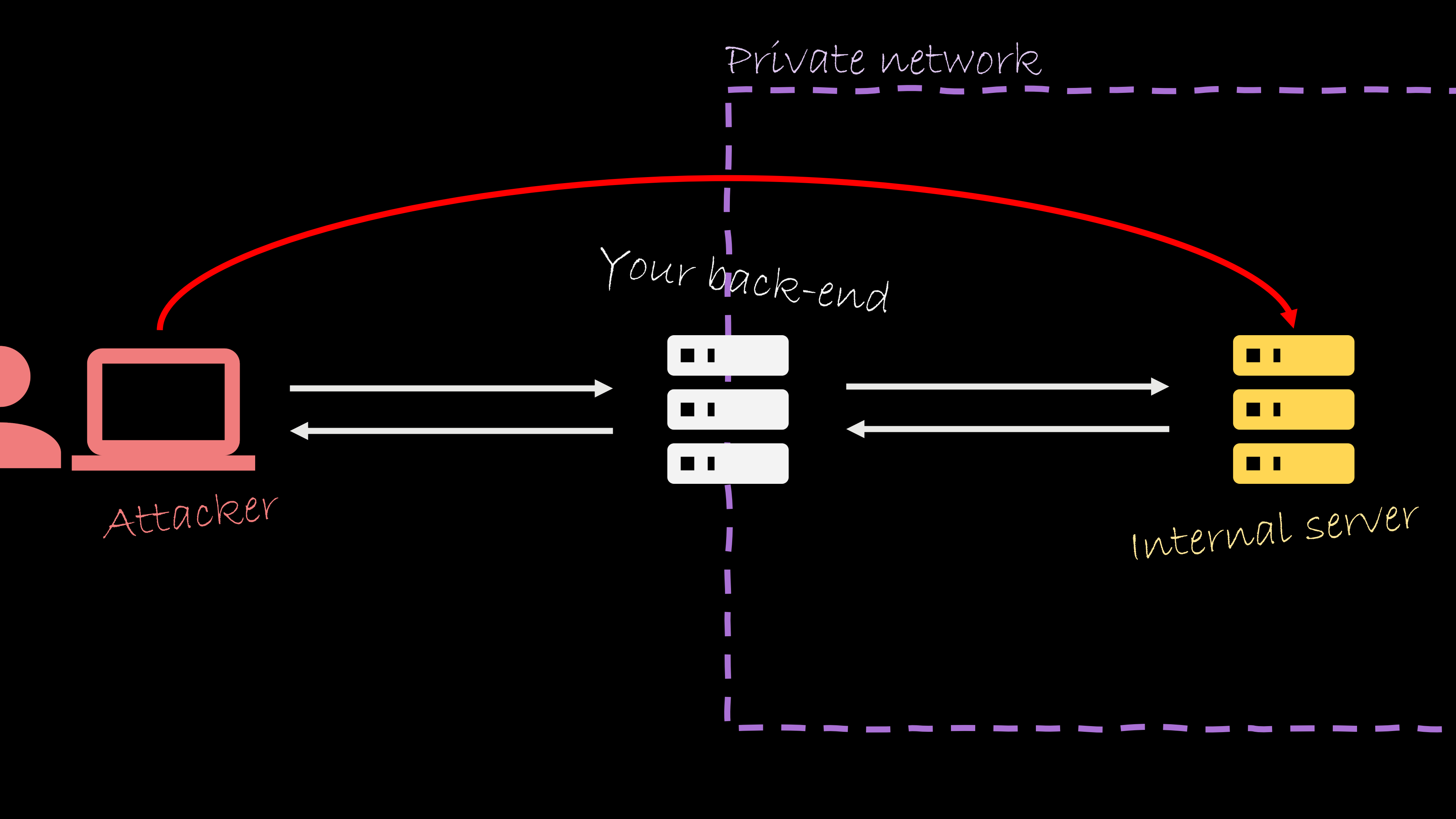


Your back-end







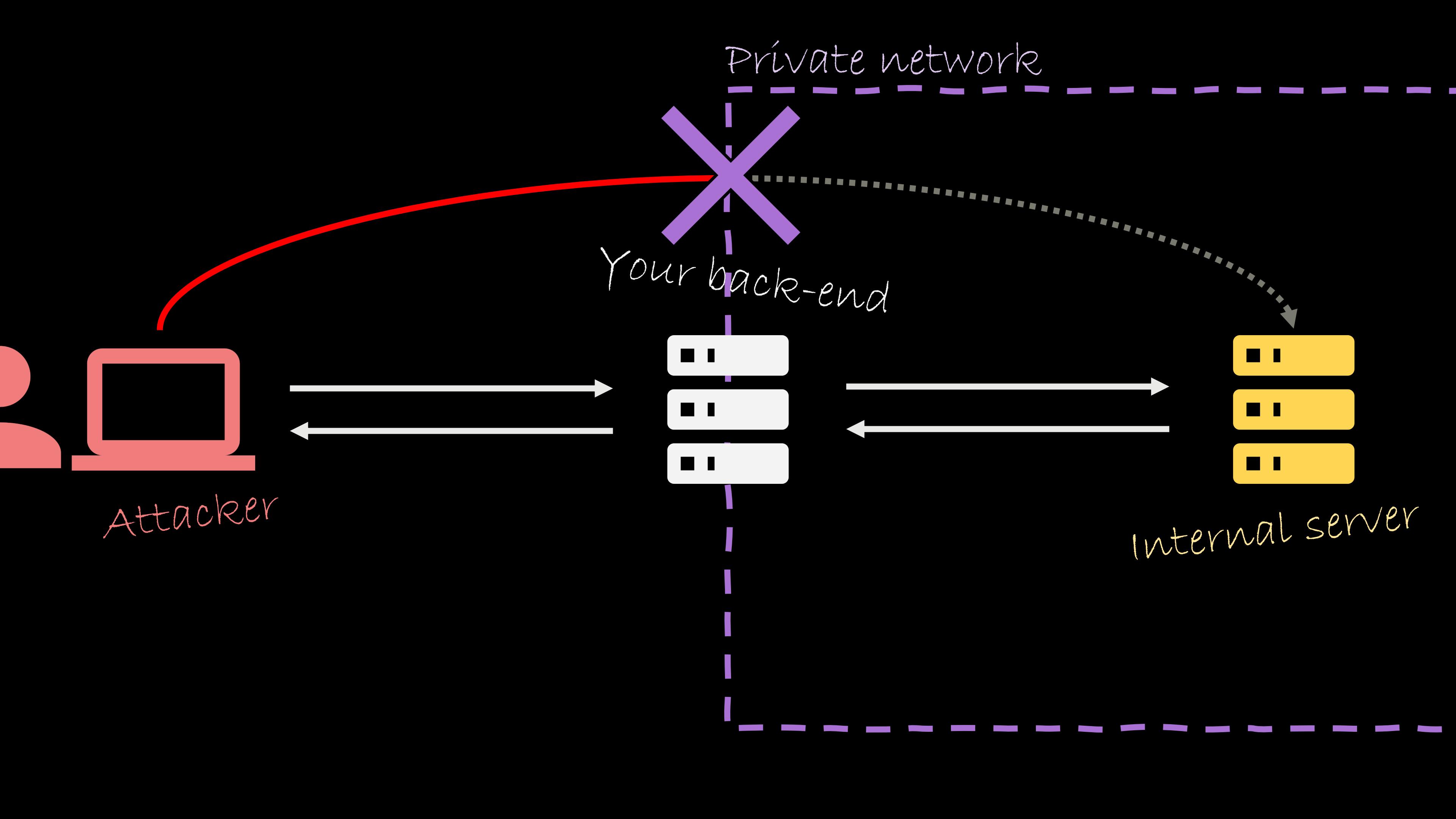


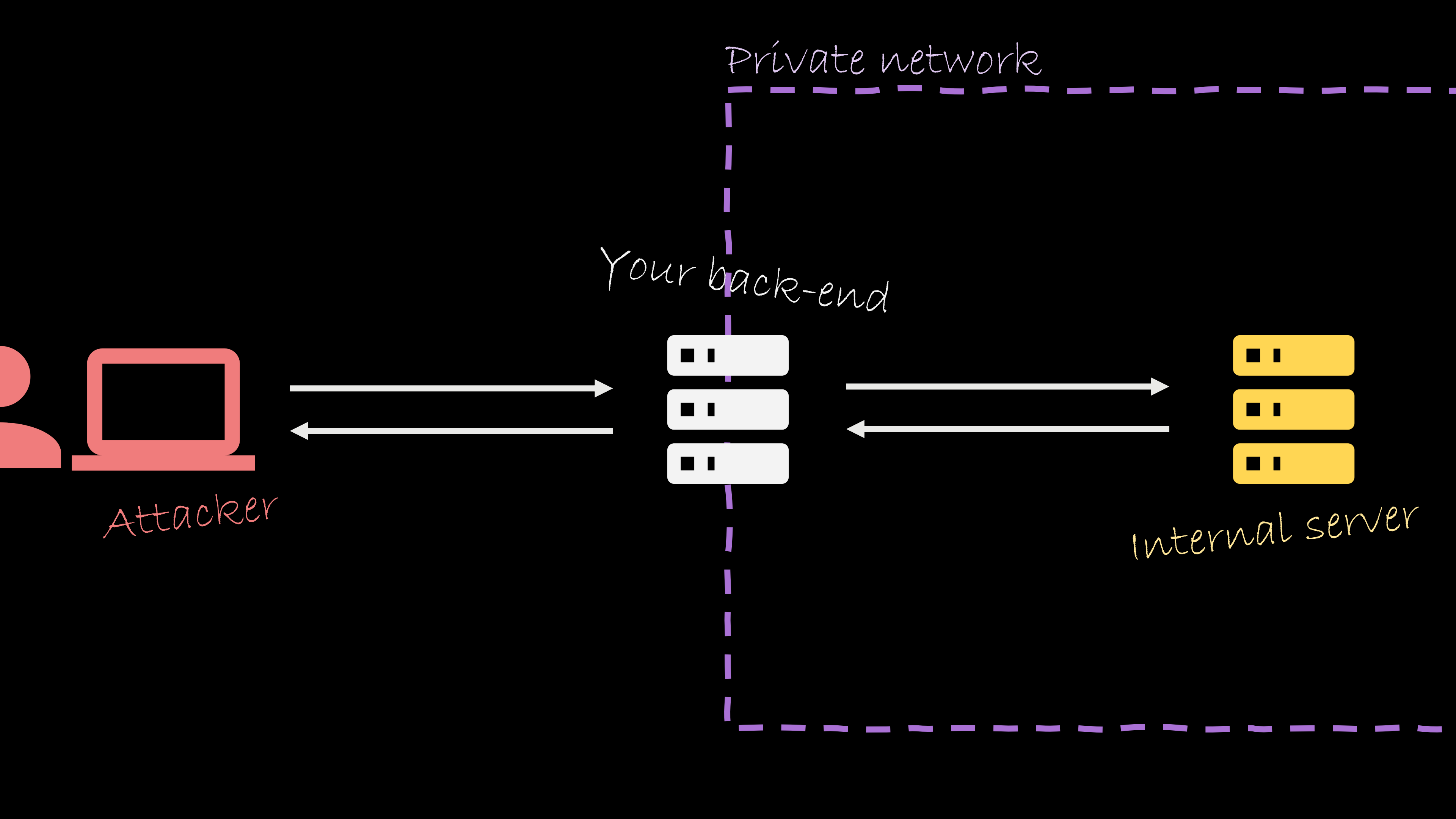
Private network

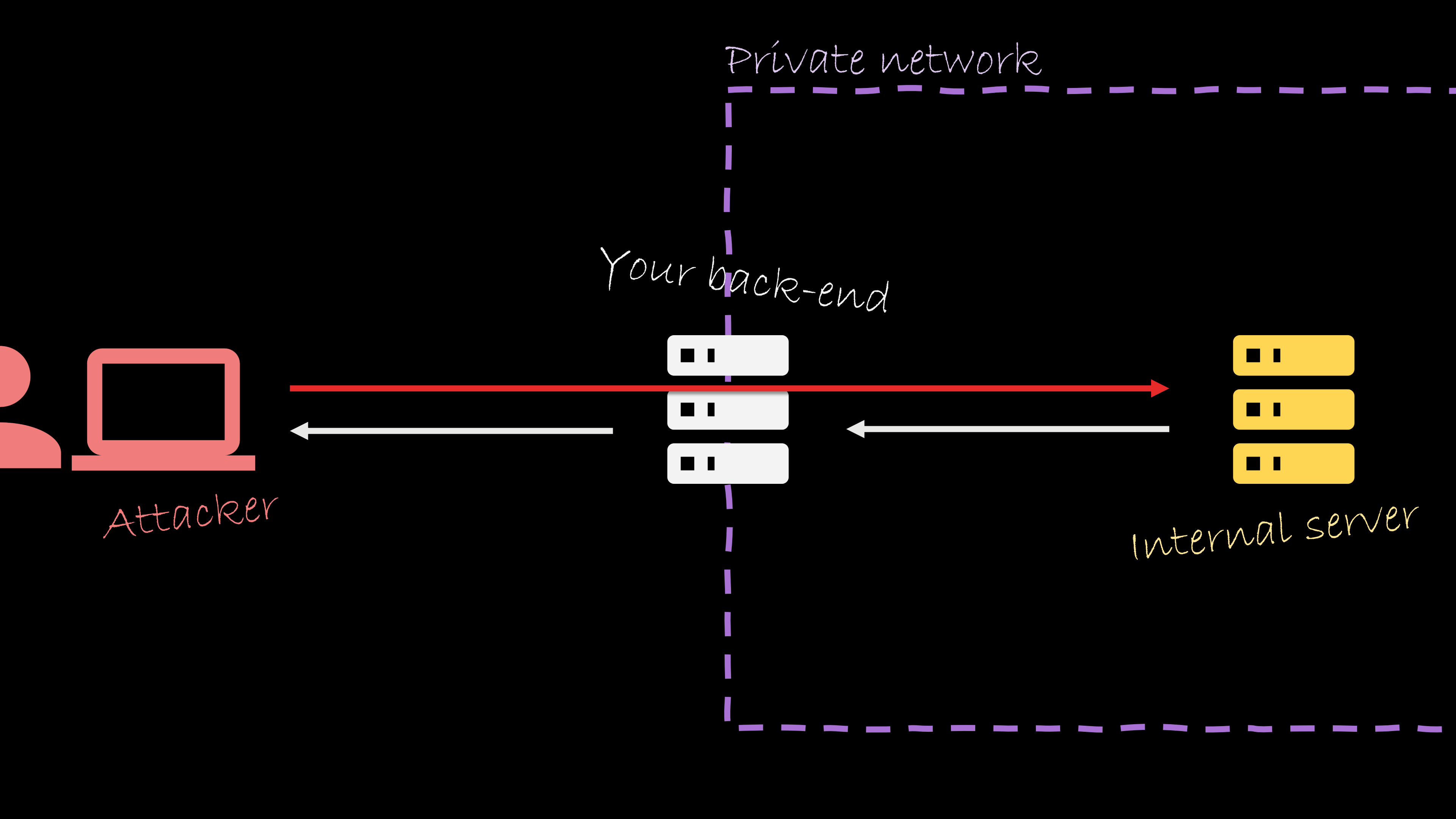
Your back-end

Attacker

Internal server







Private network

Your back-end

Attacker

Internal server

# Server-side Request Forgery

## Mitigation

- Sanitize user input
- Deny by default in firewall



# Server-side Request Forgery

[https://cheatsheetseries.owasp.org/cheatsheets/Server\\_Side\\_Request\\_Forgery\\_Prevention\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Server_Side_Request_Forgery_Prevention_Cheat_Sheet.html)

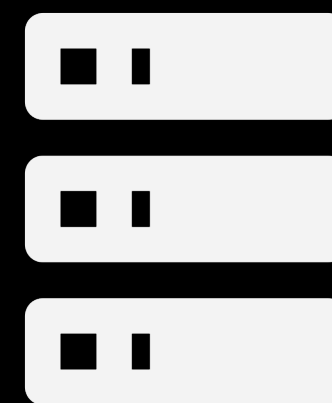
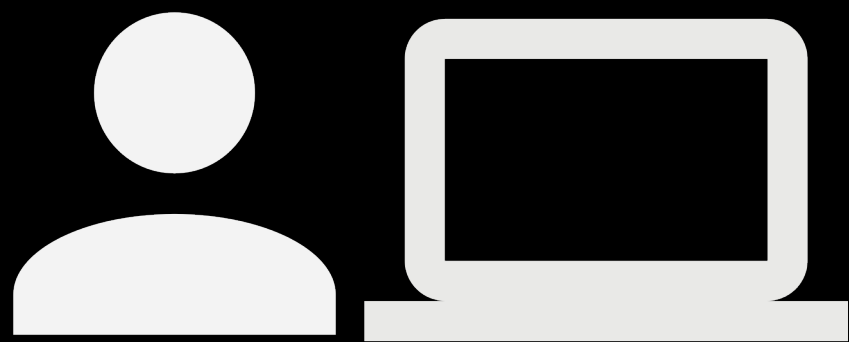
Attackers can get information

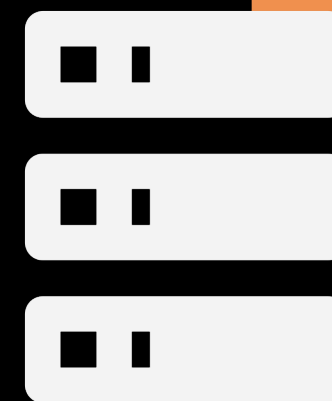
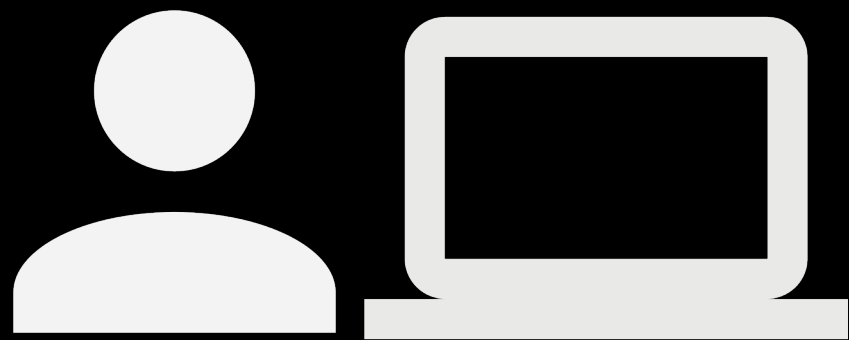
A02:2021

# Cryptographic Failures

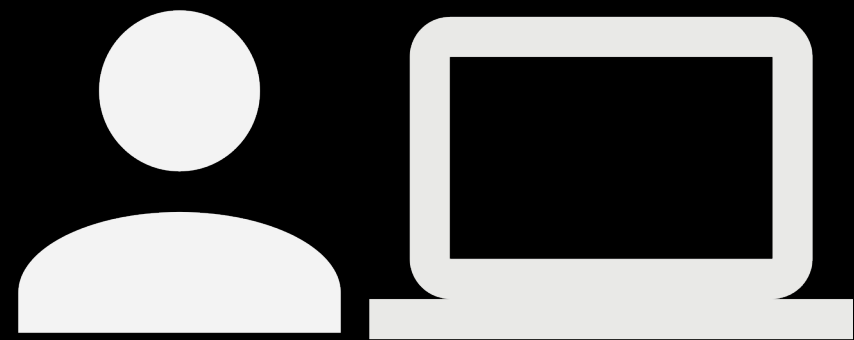
# Cryptographic Failures

Failures related to cryptography (or lack thereof).

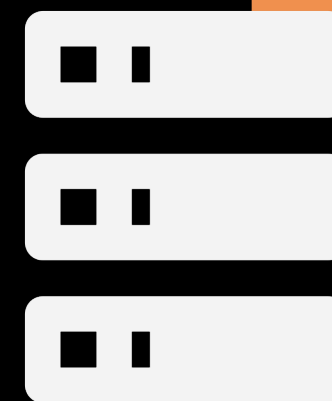




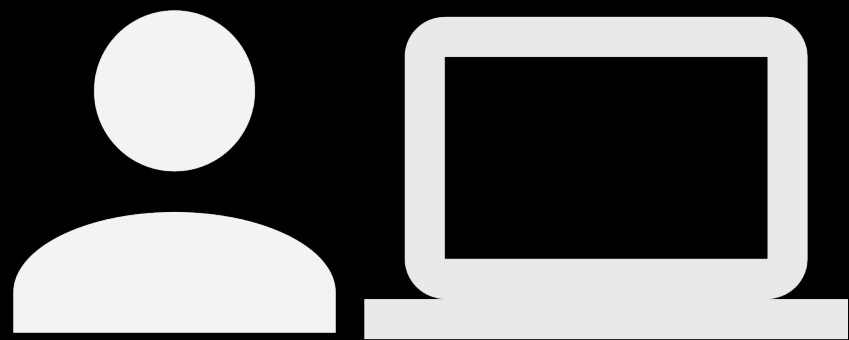
Security by  
obscurity is  
no security



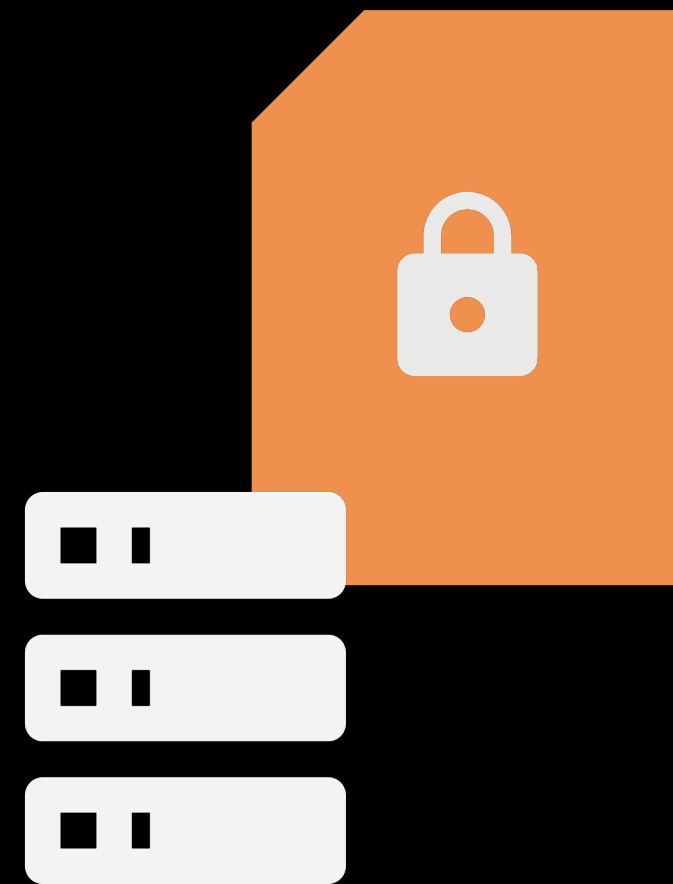
*Give me file*

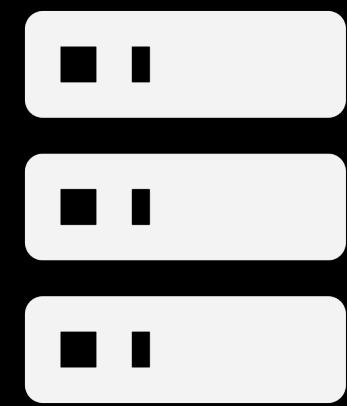


Security by  
obscurity is  
no security

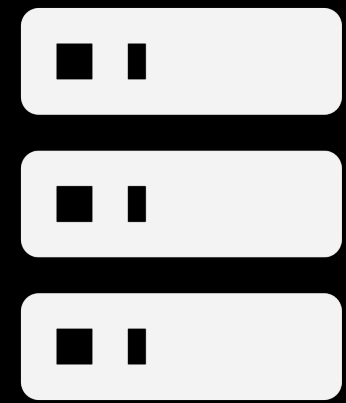


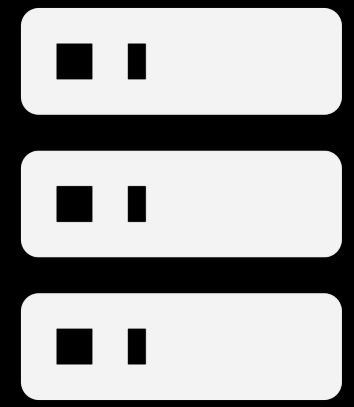
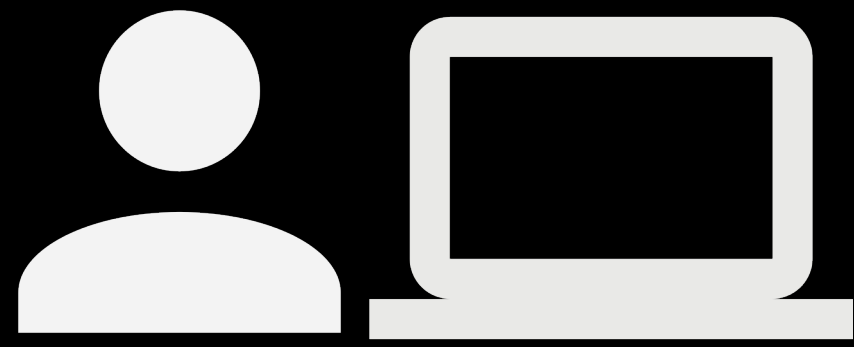
*Give me file*

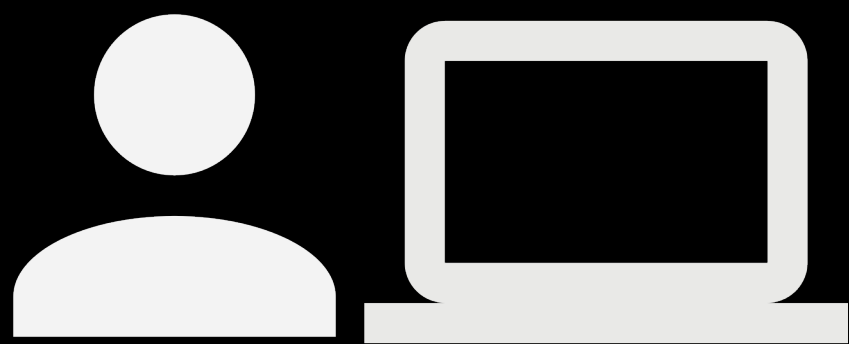


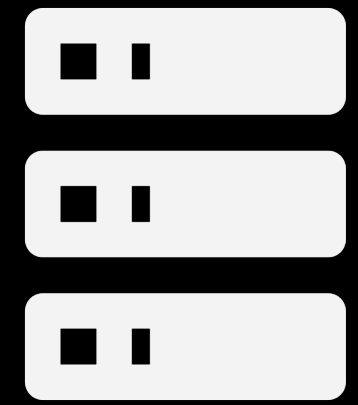


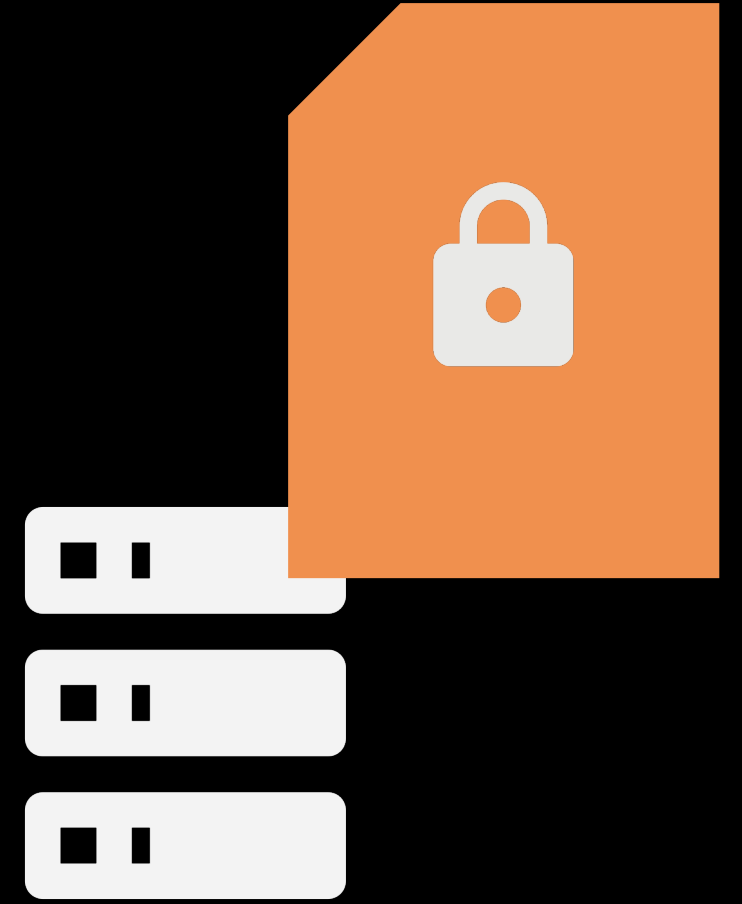


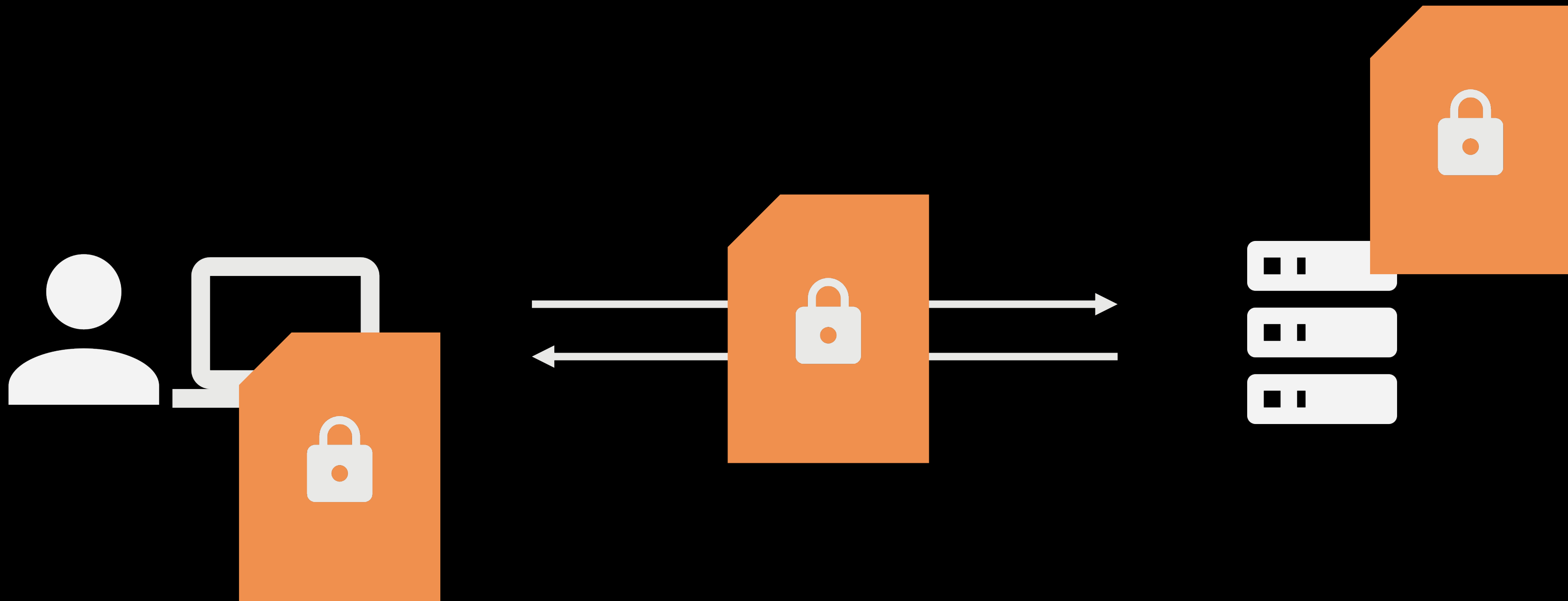












# Cryptographic Failures

## Mitigation

- Require encryption everywhere. No exceptions
- Keep encryption protocols up-to-date
- Don't use email (SMTP) or FTP

# Cryptographic Failures

[https://cheatsheetseries.owasp.org/cheatsheets/Cryptographic\\_Storage\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Cryptographic_Storage_Cheat_Sheet.html)

[https://cheatsheetseries.owasp.org/cheatsheets/Transport\\_Layer\\_Security\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Transport_Layer_Security_Cheat_Sheet.html)





# Insecure Design

Design → Implementation

# Insecure Design

Weaknesses caused by the software architecture and/or design.

	Secure implementation	Insecure implementation
Secure design		
Insecure design		

	Secure implementation	Insecure implementation
Secure design		
Insecure design		




Secure implementation





Insecure implementation

Secure design



Insecure design

	Secure implementation	Insecure implementation
Secure design		
Insecure design		

	Secure implementation	Insecure implementation
Secure design		
Insecure design		

# Insecure Design

Discuss with developers / architect

- Do you use company standard design patterns?



# Insecure Design

Discuss with developers / architect

- ~~Do you use company standard design patterns?~~
- What is the attack surface?
- What are the threats?
- How can you defend?

# Insecure Design

[https://cheatsheetseries.owasp.org/cheatsheets/Threat\\_Modeling\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Threat_Modeling_Cheat_Sheet.html)  
[https://cheatsheetseries.owasp.org/cheatsheets/Abuse\\_Case\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Abuse_Case_Cheat_Sheet.html)

# expo IQA 24

MADRID  
May 28th,  
29th, 30th  
2024

Thank you for attending

[expoqqa.com](https://expoqqa.com)