Summer School 2025

Introduction to Cybersecurity

bubu

0xDecode

Deusto Electronic Club Of Developers & Engineers

whoami

Summer School 2025

///
0x
D

Alberto Fernández-de-Retana — bubu (pronounced as boo-boo)

Interested in web security and privacy, as well as browser internals.

CTF player for: TheHackersCrew, ISwearIGoogledIt, 0xDecode …

@alberto_fdr

albertofdr.github.io

albertofdr.github.io@gmail.com

Summer School 2025

Hacking Origins

1980-1990

0xDecode

Deusto Electronic Club Of Developers & Engineers

Origins

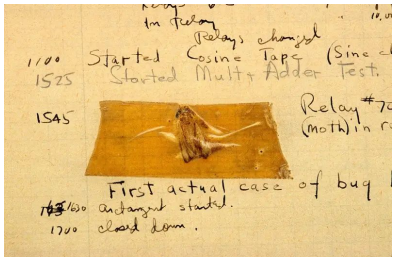
Summer School 2025

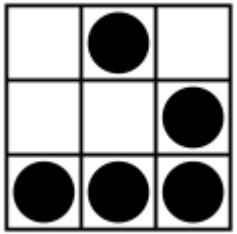
Hacker: Passion to find out how things worked and then to master them.

- **It's a "title" others gave you, you don't call yourself a hacker (expert).**
- Tech Model Railroad Club
  - "Information wants to be free"

Bug: Represents the vulnerable point (snippet of code).

- Thomas Alva Edison
- Grace Hopper - Mark II





Hacking Origins

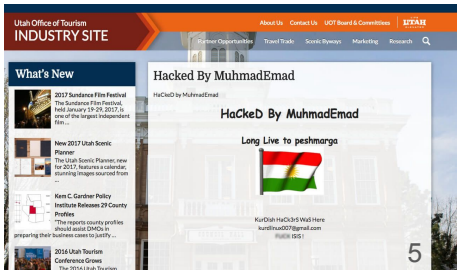
Summer School 2025

1980s

- Hacking groups
  - Legion of Doom (LOD)
  - Master of Deception (MOD)
- Phreaking
- Movie “wargames” (1983)
- Phrack Ezine (1984-now)
- BBS
- Reverse Engineering

1990s

- Hacking groups:
  - Milw0rm
- Kevin Mitnick
- BBSs to IRC
- Movie “hackers” (1995)
- Beginning of website defacements
- Reverse Engineering



Hacker Manifesto (1986)

Summer School 2025

///
0x
D

"Yes, I am a criminal. **My crime is that of curiosity**. My crime is that of **judging people by what they say and think, not what they look like**. My crime is that of outsmarting you, something that you will never forgive me for. I am a hacker, and this is my manifesto. You may stop this individual, but you can't stop us all... after all, **we're all alike**."

Summer School 2025

Cybersecurity

Introduction

Everyday Software, Everyday Risks

Summer School 2025

Consulting worker (not including phone or other gadgets):

- OS security (Linux/Mac/Windows).
- Desktop Application installed (Teams, IDE, Discord with 0xDecode Channel).
- Network connection (WiFi/Ethernet) (Active Directory).
- Browser (Chromium-based, Firefox or Safari)
- Website (bank.com).
- Videogames (Chess Online)
- Cloud (AWS/Azure)
- Phone (Android, Mac)

Example of Shooter Video Game

How to cheat?

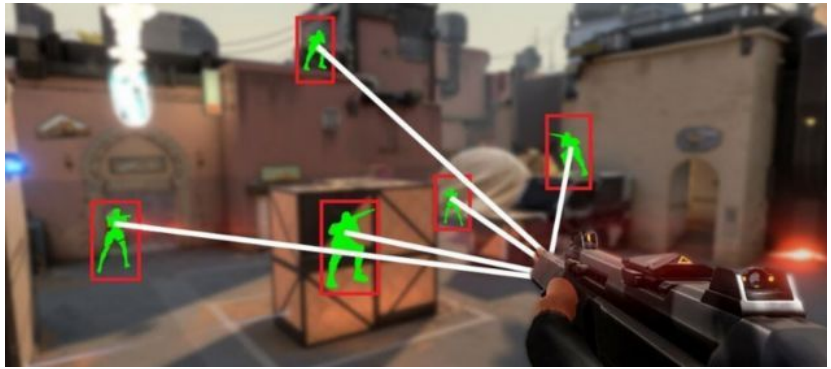
Example of Shooter Video Game

Summer School 2025

How to cheat?

Case 1:

1. You notice that user receives enemies position even when they are not in the screen.
2. Change the game to see the enemies.



Example of Shooter Video Game

Summer School 2025

How to cheat?

Case 2: Server not checking user movement data

1. You notice that user can move faster than expected or jump more one time allowing you to fly.

Example of Shooter Video Game

How to detect someone cheating?

In Server

- ???

In Client Computer

- ???

Example of Shooter Video Game

How to detect someone cheating?

In Server

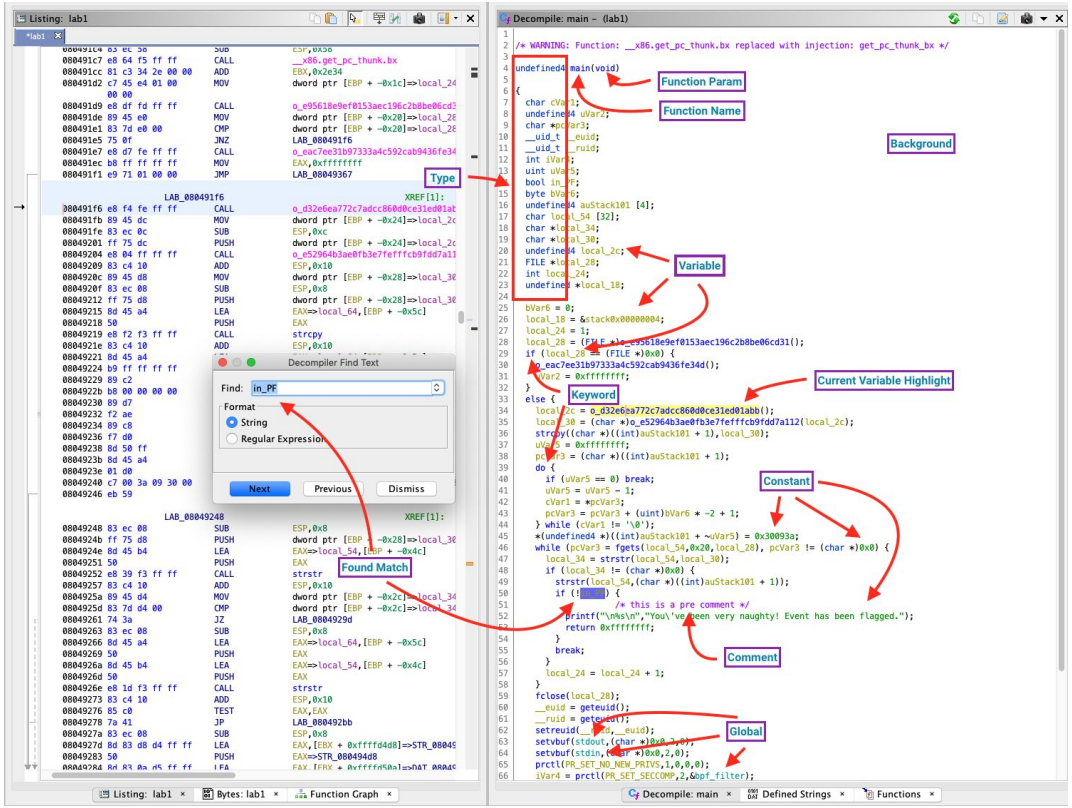
- Behavioral analysis (antinatural moving, killing someone in the back too fast)
- Results analysis (100% of headshots, wtf?)

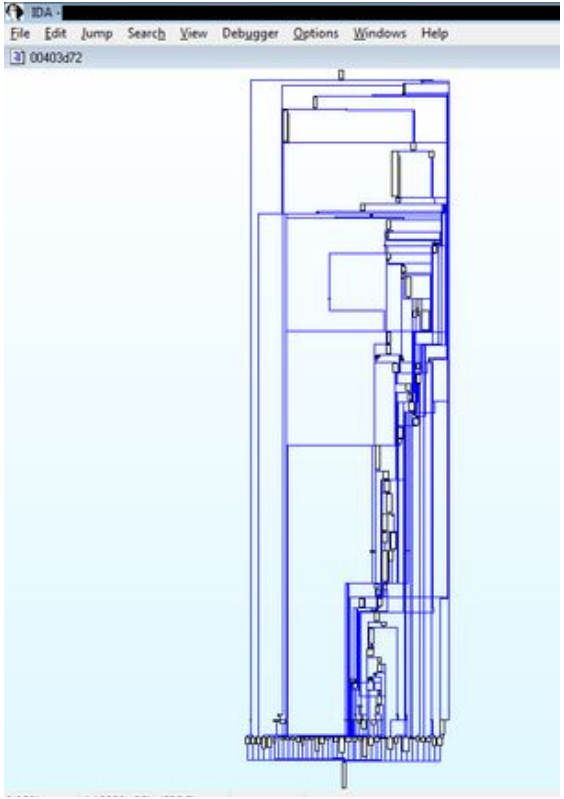
In Client Computer

- Signature or Process Name Scanning (a very common process called “definitive-cheat-tool”)
- File and Memory checks
- Kernel-monitoring
  - Computer performance is fucked up

Good blog on Vanguard: https://reversing.info/posts/guardedregions/
Hacking games: LiveOverflow Youtube

Example of Shooter Video Game





Master Thesis: Pablo Valiente

Supervised by: Antonio Nappa

[1]: https://github.com/Aspasia1337/Aspasia/tree/main-ui

Basic General concepts

- Authentication Bypass
- No user/attacker sanitization
- Race condition
- …

Basic concepts

Summer School 2025

- **Authentication Bypass**

```python
@app.route('/admin/delete/<int:uid>', methods=['GET'])
def delete_user(uid):
    # No authentication check!
    db.delete_user(uid)
    return "User deleted"
```

Web

```java
if (username.equals("admin") && password.equals("123456")) {
    loginStatus = true;
    Toast.makeText(context, "Login successful", Toast.LENGTH_SHORT).show();
}
```

APK

Basic concepts

Summer School 2025

- **No user/attacker sanitization**

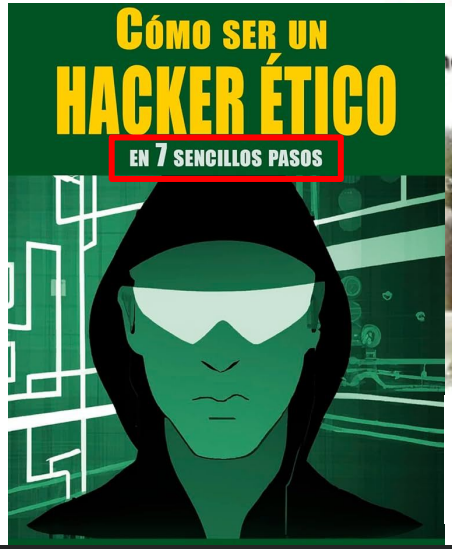
Web

Binary

Cybersecurity

How to start

How NOT to start



¿Quieres adentrarte al fascinante mundo del Hacking?, disfruta de este maravilloso curso de **53 horas** donde serás guiado paso a paso por S4vitar con el objetivo de que aprendas a detectar y explotar vulnerabilidades.

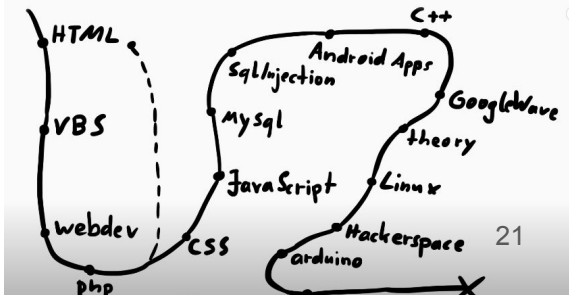
How to start - Early steps

Summer School 2025

- **Computer basics**
  - Programming
  - Networking (OSI model, TCP/IP, DNS, HTTP/S)
  - OS-basics
- **Mentality**
  - Curiosity
  - Persistance
  - Problem-solving mindset

How to make the process faster?

- Find places, communities or people with same interests where you can ask technical questions (e.g., 0xDecode)



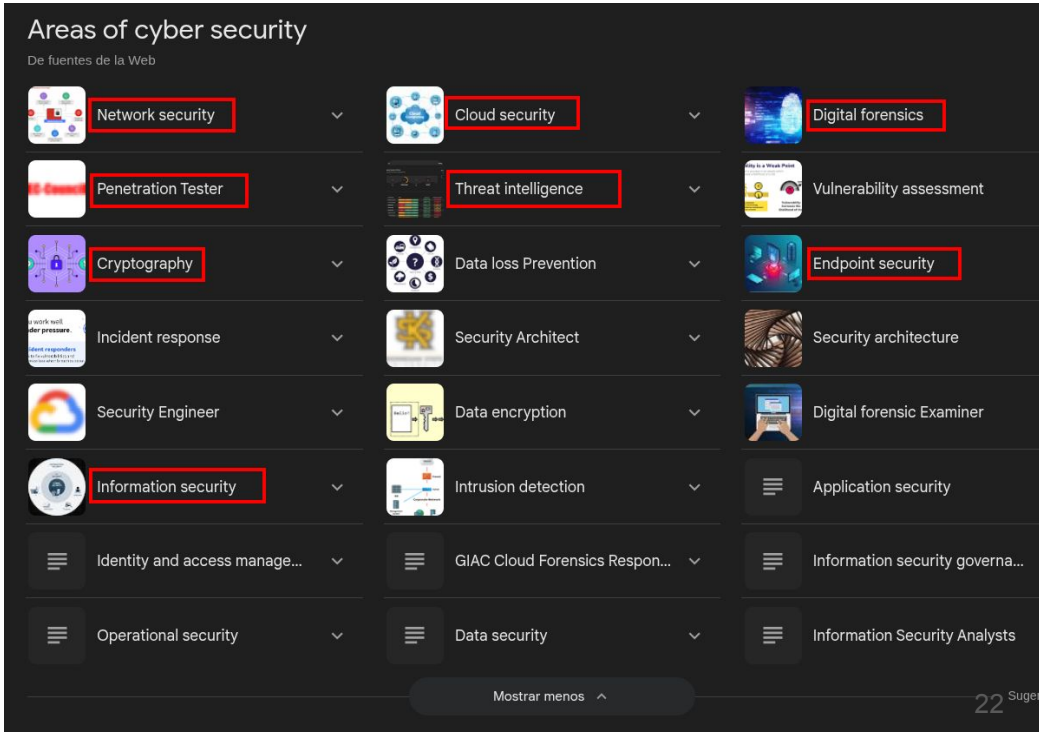
How to start in Cybersecurity?

Summer School 2025

For almost any computer-related area, including hardware, a security job exists.

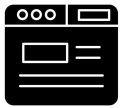
- Not only to ensure that **things work**, but also that they **work correctly.**

- Fix things when something bad happened (e.g., ransomware).



General recommendations

- Having a website with a blog usually helps.
  - CVE reviews, CTF writeups, …



- Having friends in the field helps.
  - Going to events (conferences, ctfs, …).
  - Reading and talking with other researchers (twitter, discord…)
    - *You read a blog and was really cool but still there are things that didn't understand, ask the author.*

How to learn from other researchers

Summer School 2025

Blogs

- Individual researchers blog.
- Big companies research blog (e.g., PortSwigger).

Conferences

- (Academic) IEEE S&P, NDSS, CCS and Usenix
- BlackHat, DEFCON, Chaos Computer Club (CCC), OffensiveCon, NullCon, M0lecon, No Hat, DefCamp and other random conferences all over the world.

CTF competitions

Don’t know where to start→ Lugares fantásticos y cómo encontrarlos [RootedCON 2018]

Free&Cool Platforms to start

Summer School 2025

pwn.college

OPEN SECURITY TRAINING .INFO

Root Me

Web Security Academy

pwn.college

Summer School 2025

///
0x
D

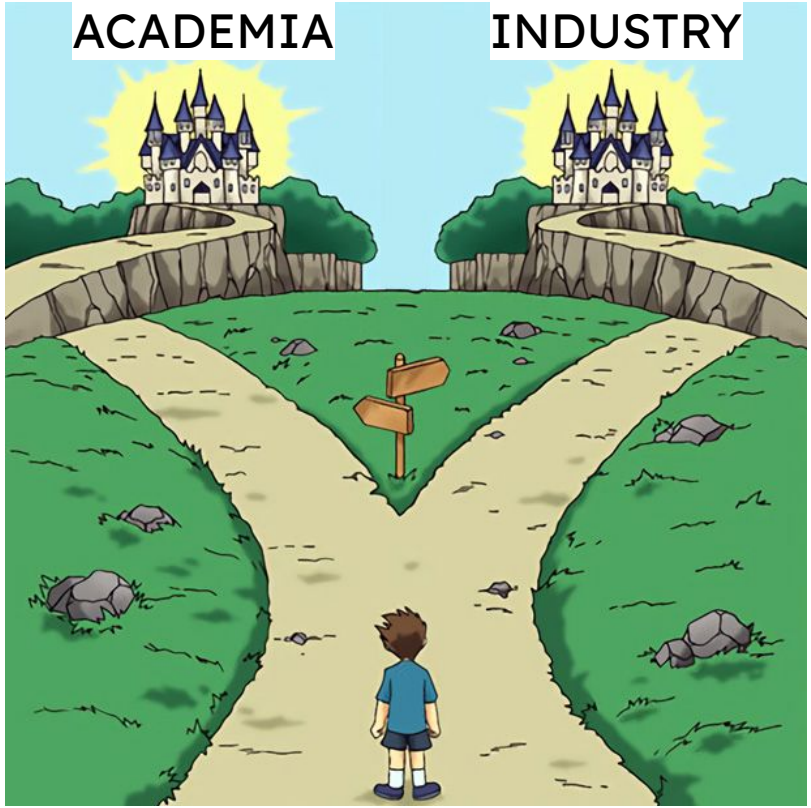
Cybersecurity

career paths

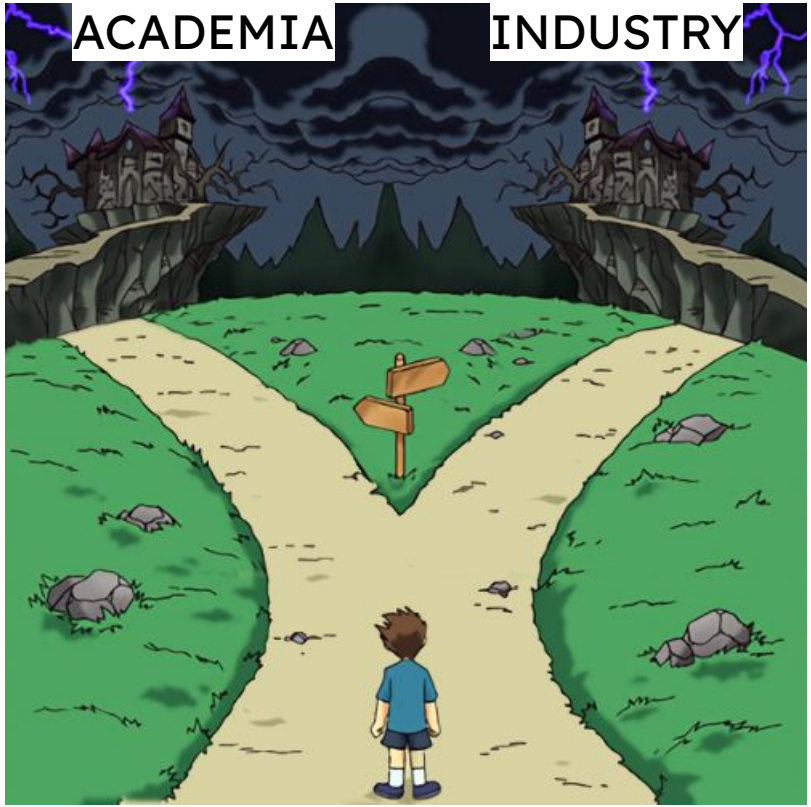
0xDecode

Deusto Electronic Club Of Developers & Engineers

Cybersecurity Career paths

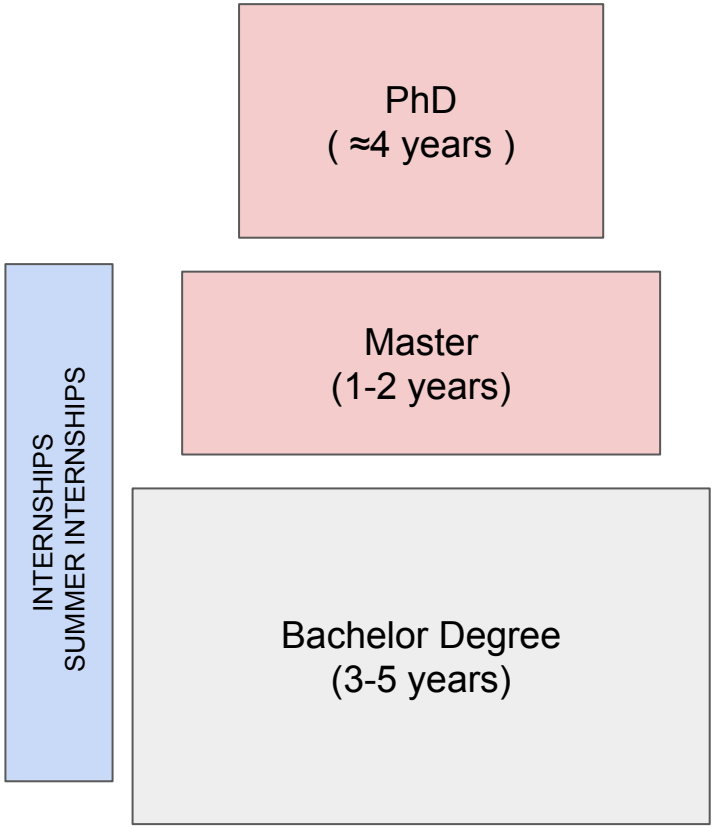


Cybersecurity Career paths



Academic Career path

Summer School 2025



Having a good supervisor (not only in the personal terms) and the supervisor's team is the key of good PhD.

Doing the master in a good university (or at least good university in your topic).

Average grade is important for Master or PhD Scholarships.

Academic Career path

PhD
(≈4 years)

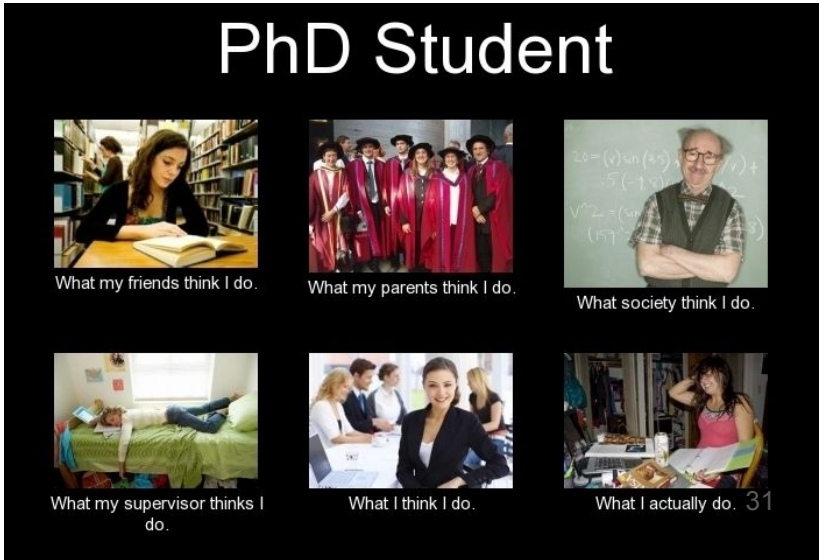
Summer School 2025

+ Freedom to learn and research (depends in the scholarship)

+ Interdisciplinary abilities (talks, discussing ideas, supervising a student…)

- money (depends on the country)

- mental health (depending on the person)

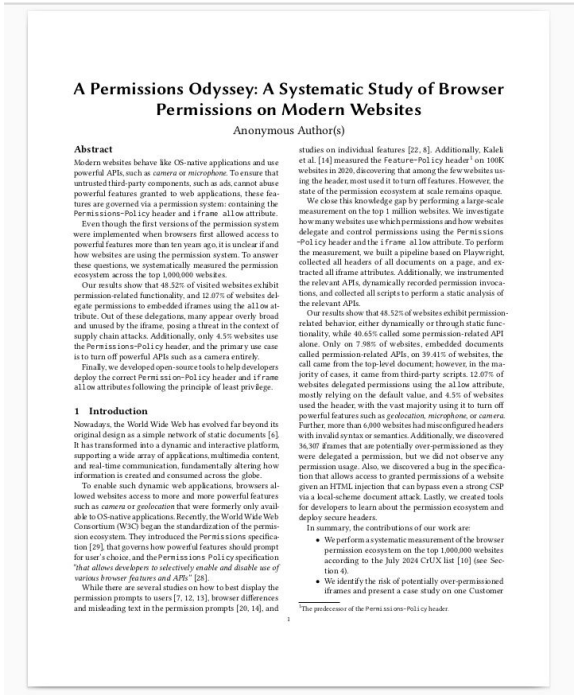


Academic Career path

PhD
(≈4 years)

Summer School 2025

///
0x
D



A Permissions Odyssey: A Systematic Study of Browser Permissions on Modern Websites

Anonymous Author(s)

Abstract

Modern websites behave like OS-native applications and use powerful APIs, such as *camera* or *microphone*. To ensure that untrusted third-party components, such as ads, cannot abuse powerful features granted to web applications, these features are governed via a permission system: containing the `Permissions-Policy` header and `iframe allow` attribute.

Even though the first versions of the permission system were implemented when browsers first allowed access to powerful features more than ten years ago, it is unclear if and how websites are using the permission system. To answer these questions, we systematically measured the permission ecosystem across the top 1,000,000 websites.

Our results show that 48.52% of visited websites exhibit permission-related functionality, and 12.07% of websites delegate permissions to embedded iframes using the `allow` attribute. Out of these delegations, many appear overly broad and unused by the iframe, posing a threat in the context of supply chain attacks. Additionally, only 4.5% websites use the `Permissions-Policy` header, and the primary use case is to turn off powerful APIs such as a camera entirely.

Finally, we developed open-source tools to help developers deploy the correct `Permission-Policy` header and `iframe allow` attributes following the principle of least privilege.

1 Introduction

Nowadays, the World Wide Web has evolved far beyond its original design as a simple network of static documents [6]. It has transformed into a dynamic and interactive platform, supporting a wide array of applications, multimedia content, and real-time communication, fundamentally altering how information is created and consumed across the globe.

To enable such dynamic web applications, browsers allowed websites access to more and more powerful features such as *camera* or *geolocation* that were formerly only available to OS-native applications. Recently, the World Wide Web Consortium (W3C) began the standardization of the permission ecosystem. They introduced the `Permissions` specification [29], that governs how powerful features should prompt for user's choice, and the `Permissions Policy` specification *"that allows developers to selectively enable and disable use of various browser features and APIs"* [28].

While there are several studies on how to best display the permission prompts to users [7, 12, 13], browser differences and misleading text in the permission prompts [20, 14], and studies on individual features [22, 8]. Additionally, Kaleli et al. [14] measured the `Feature-Policy` header[1] on 100K websites in 2020, discovering that among the few websites using the header, most used it to turn off features. However, the state of the permission ecosystem at scale remains opaque.

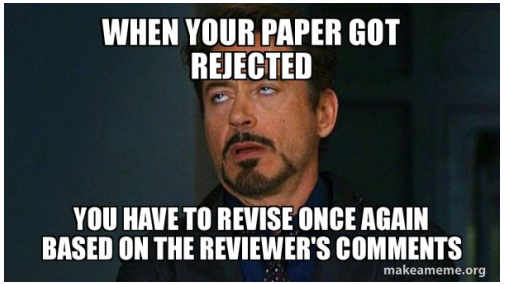
We close this knowledge gap by performing a large-scale measurement on the top 1 million websites. We investigate how many websites use which permissions and how websites delegate and control permissions using the `Permissions-Policy` header and the `iframe allow` attribute. To perform the measurement, we built a pipeline based on Playwright, collected all headers of all documents on a page, and extracted all iframe attributes. Additionally, we instrumented the relevant APIs, dynamically recorded permission invocations, and collected all scripts to perform a static analysis of the relevant APIs.

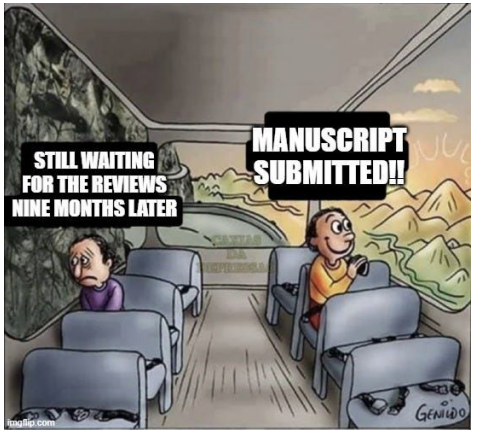
Our results show that 48.52% of websites exhibit permission-related behavior, either dynamically or through static functionality, while 40.65% called some permission-related API alone. Only on 7.98% of websites, embedded documents called permission-related APIs, on 39.41% of websites, the call came from the top-level document; however, in the majority of cases, it came from third-party scripts. 12.07% of websites delegated permissions using the `allow` attribute, mostly relying on the default value, and 4.5% of websites used the header, with the vast majority using it to turn off powerful features such as *geolocation*, *microphone*, or *camera*. Further, more than 6,000 websites had misconfigured headers with invalid syntax or semantics. Additionally, we discovered 36,307 iframes that are potentially over-permissioned as they were delegated a permission, but we did not observe any permission usage. Also, we discovered a bug in the specification that allows access to granted permissions of a website given an HTML injection that can bypass even a strong CSP via a local-scheme document attack. Lastly, we created tools for developers to learn about the permission ecosystem and deploy secure headers.

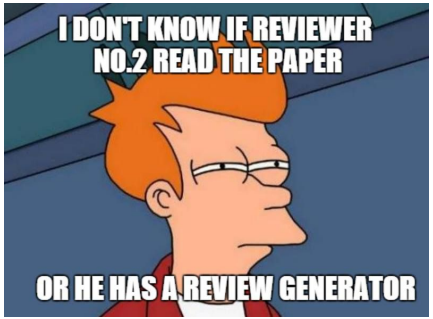
In summary, the contributions of our work are:

- We perform a systematic measurement of the browser permission ecosystem on the top 1,000,000 websites according to the July 2024 CrUX list [10] (see Section 4).
- We identify the risk of potentially over-permissioned iframes and present a case study on one Customer

[1]The predecessor of the `Permissions-Policy` header.







Academic Career path

PhD
(≈4 years)

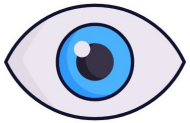
Summer School 2025

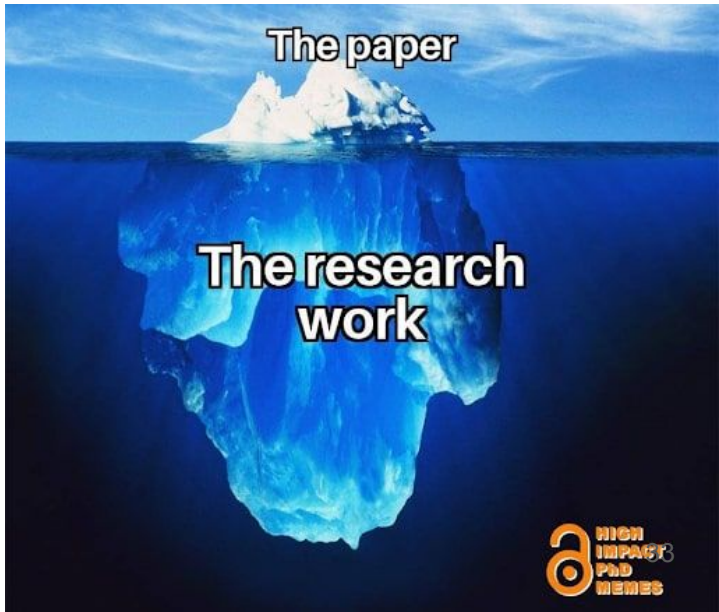
Depends in your interests, your supervisors background and what is publishable at that moment.

- Web Security/Privacy
  - Web tracking, Client-Side issues, Spec, Measurements….
- Low level
  - Fuzzing,
  - Side-channels
  - Malware
- Usable security
  - …

Conferences:

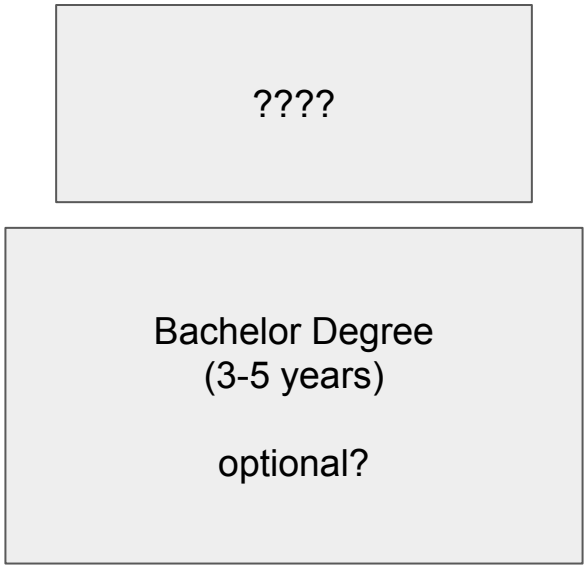
- Usenix
- IEEE Security&Privacy
- ACM CCS
- NDSS





Industry Career path

Summer School 2025



- WHAT DO I WANT TO DO?
- HOW TO FIND A JOB IN CYBERSECURITY?

Industry Career path

Summer School 2025

- Offensive/Red team and Defensive/Blue team
- Application Security (white box, grey box, black box)
- Bug Bounty
- Vulnerability Research
- Digital Forensics & Incident response
- Cloud Security
- Cryptography
- Car security
- Hardware hacking
- Web3
- Threat Intelligence & Analysis (IOCs)
- ...

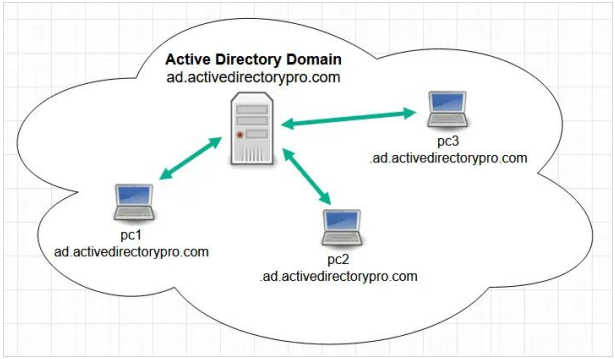
Some job example

Summer School 2025

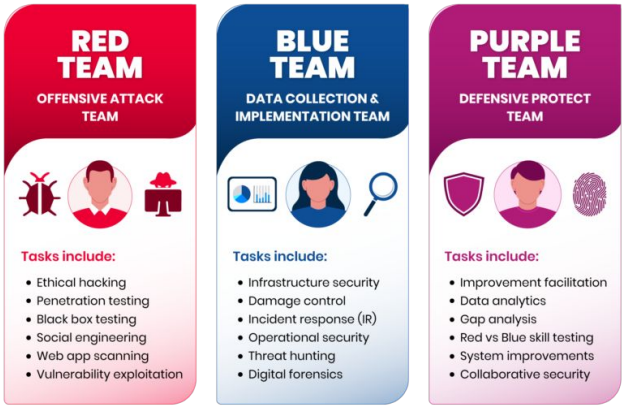
///
0x
0

Pentesting

Active Directory



Red/Blue/Purple Team



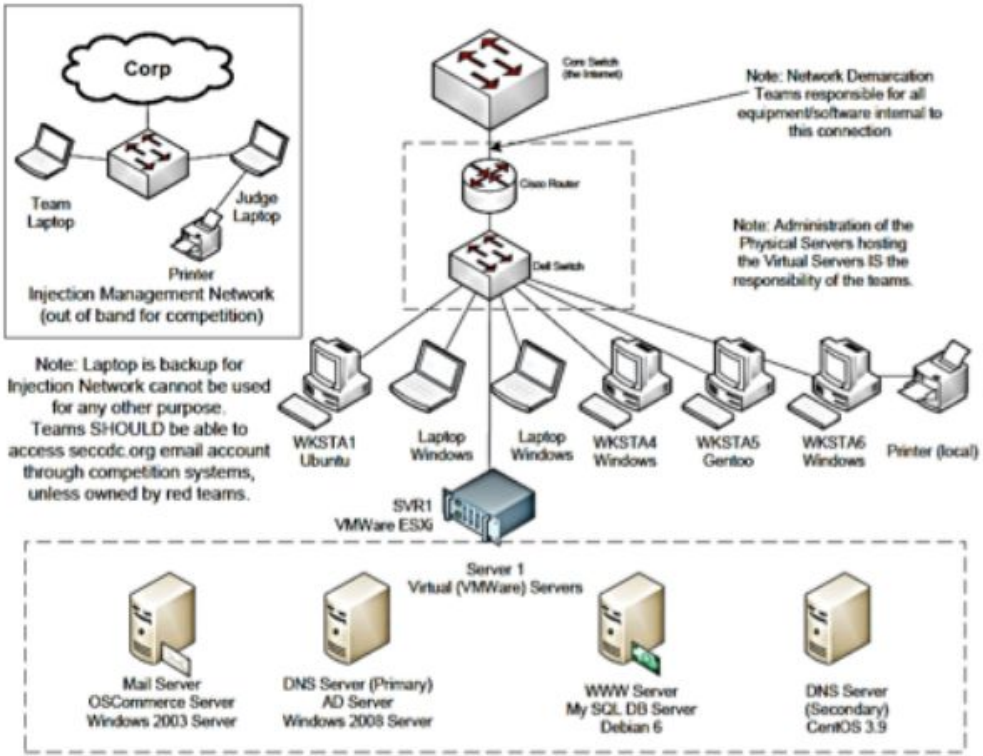
Vulnerability Research

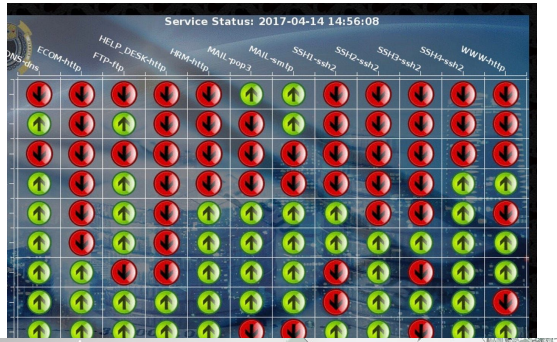
Application Security

2019	2018	2017
1. CVE-2018-15982	1. CVE-2018-8174	1. CVE-2017-0199
2. CVE-2018-8174	2. CVE-2018-4878	2. CVE-2016-0189
3. CVE-2017-11882	3. CVE-2017-11882	3. CVE-2017-0022
4. CVE-2018-4878	4. CVE-2017-8750	4. CVE-2016-7200
5. CVE-2019-0752	5. CVE-2017-0199	5. CVE-2016-7201

Red Team vs Blue Team

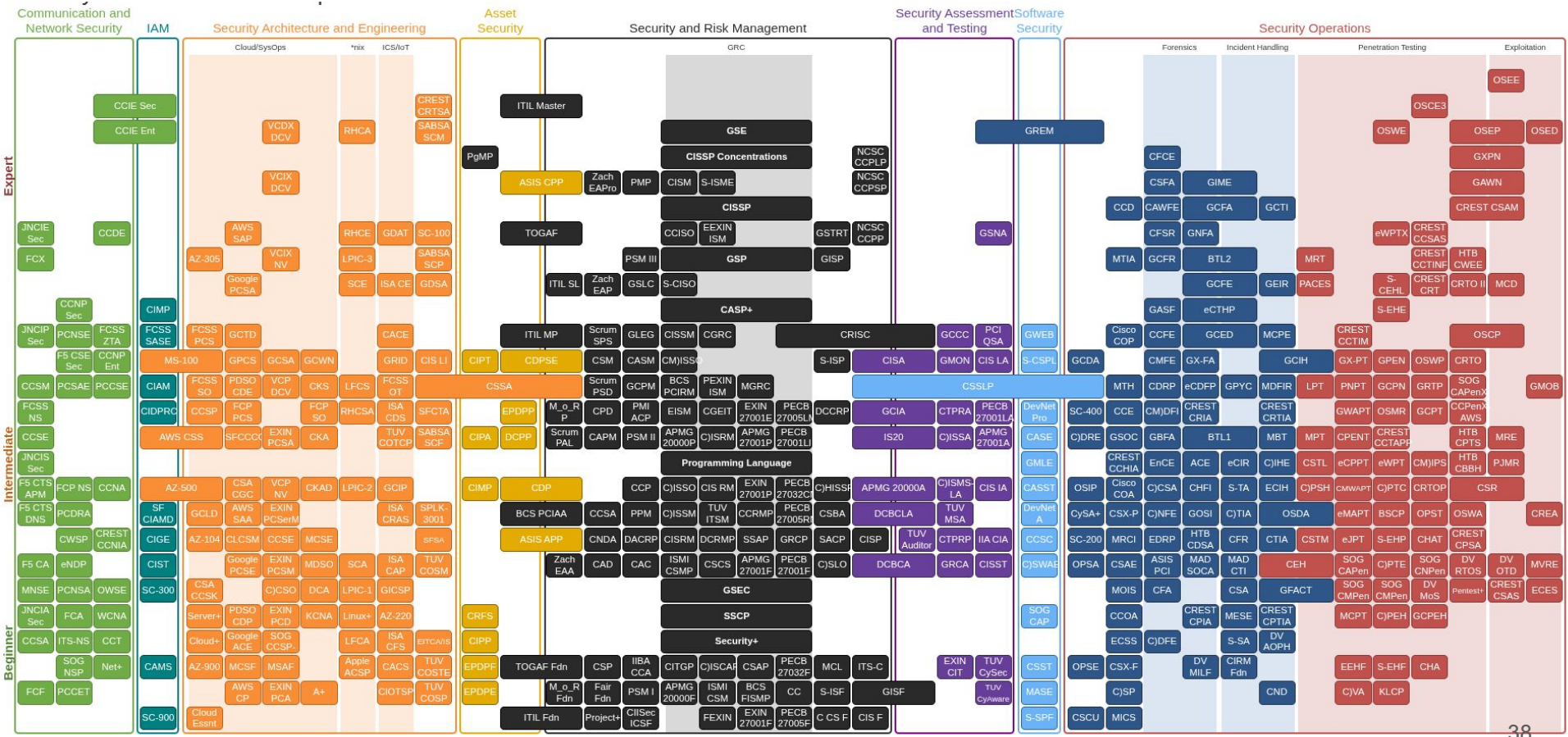
Summer School 2025





CDCC EEUU

Certs



481 certifications listed | July 2024

Cybersecurity

Learning from others is the key

0xDecode

Deusto Electronic Club Of Developers & Engineers

Summer School 2025

How to learn, motivate yourself and keep pushing

- Competing and collaborating with other researchers that have the same interest.
- Reading other researchers articles (academic or blogs).
  - They wrote about something I'm working on, write them a message!
- Explore your own ideas.
  - How common is this behavior? Is this in the spec? ….

Cool influencers/youtubers

Summer School 2025

LiveOverflow

Germany

Links

TZ: CET (GMT+1)

Content

CTF Challenges
general IT security

Schedule

Aiming for once a week

CryptoCat

United Kingdom

Links

TZ: GMT

Content

Binary Exploitation
Reverse Engineering
Offensive Security
Penetration Testing
Malware Analysis CTF Pwn
HackTheBox

Schedule

1-3 Videos Per Week

Katie / InsiderPhD

UK

Links

TZ: UTC+0

Content

Bug Bounties web app hacking

Schedule

Videos most weeks,
Wednesdays 16:00
GMT/BST

John Hammond

United States

Links

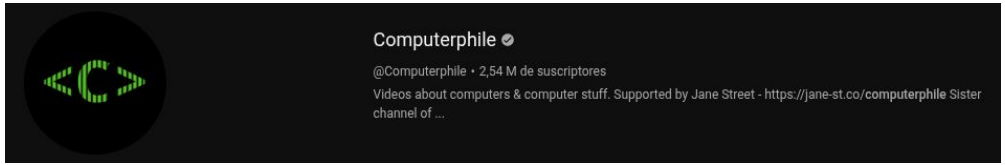
TZ: EST

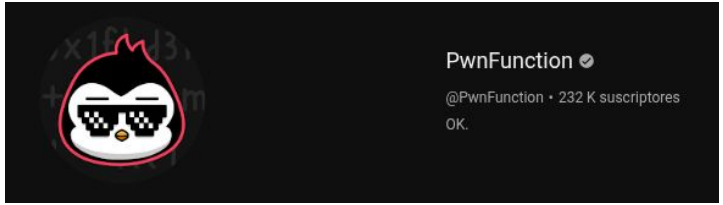
Content

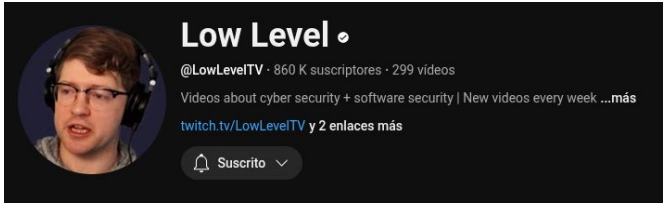
Capture the Flag
InfoSec Security Conferences
HackTheBox/TryHackMe
Online Wargames Pentesting

Schedule

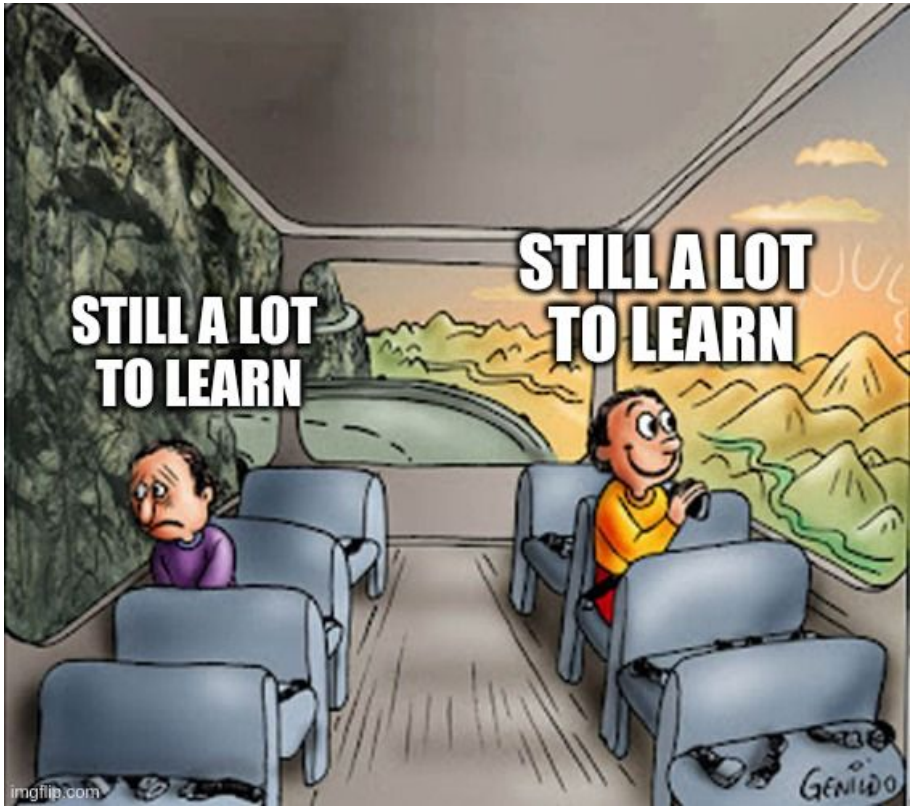
Sporadic







Any area



Summer School 2025

Hacking Anecdotes

curiosity

Information Leak

Summer School 2025

///
0x
D

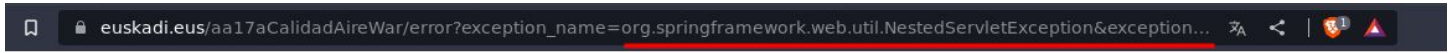
“Quieren un perfil netamente comercial… (vamos un fenicio, cuantos menos escrúpulos mejor)”

“uno tiene una ONG y el otro formación en Harvard y no se qué pero no hacen nada …”

“Tiene que ser absolutamente confidencial por el momento. La persona a la que vamos a reemplazar aún no sabe que la vamos a despedir.”

“La idea es que salga la persona que estaba contratada que ha salido imputada (K.O)”

euskadi.eus



Ayuda

ERROR

Volver al inicio

Name:

org.springframework.web.util.NestedServletException

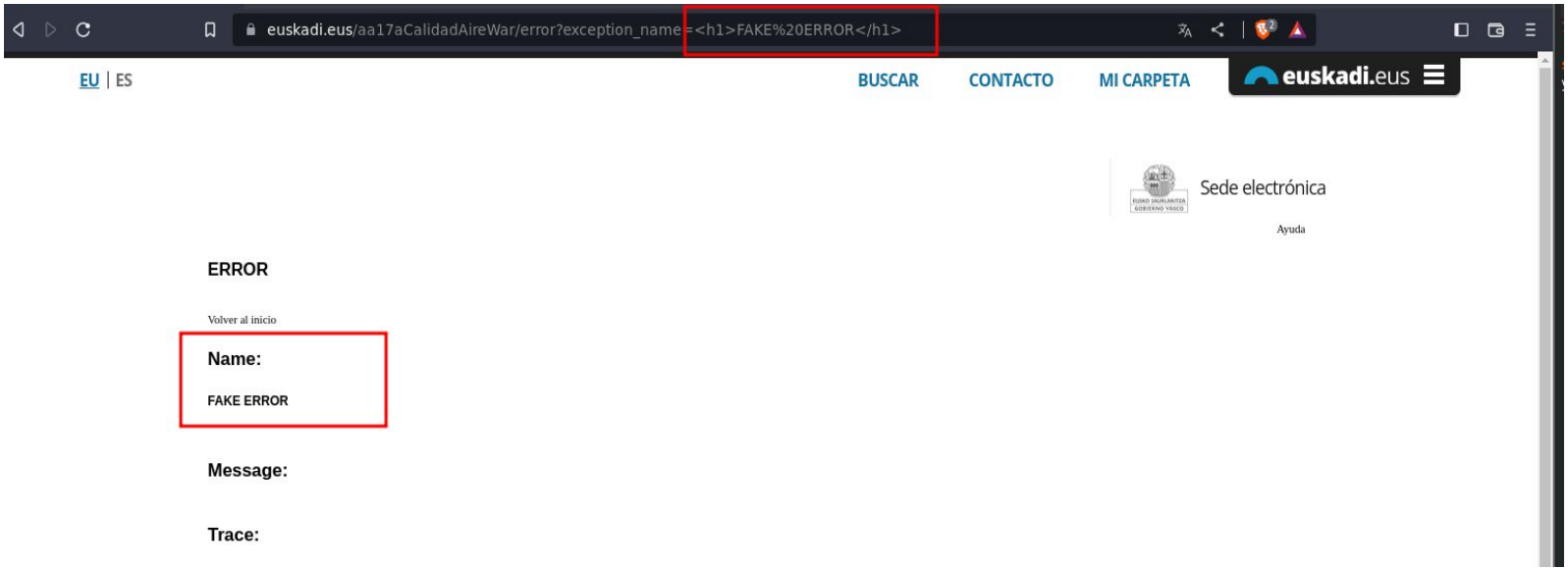
Message:

Request processing failed; nested exception is ExceptionConverter: java.io.IOException: The document has no pages.

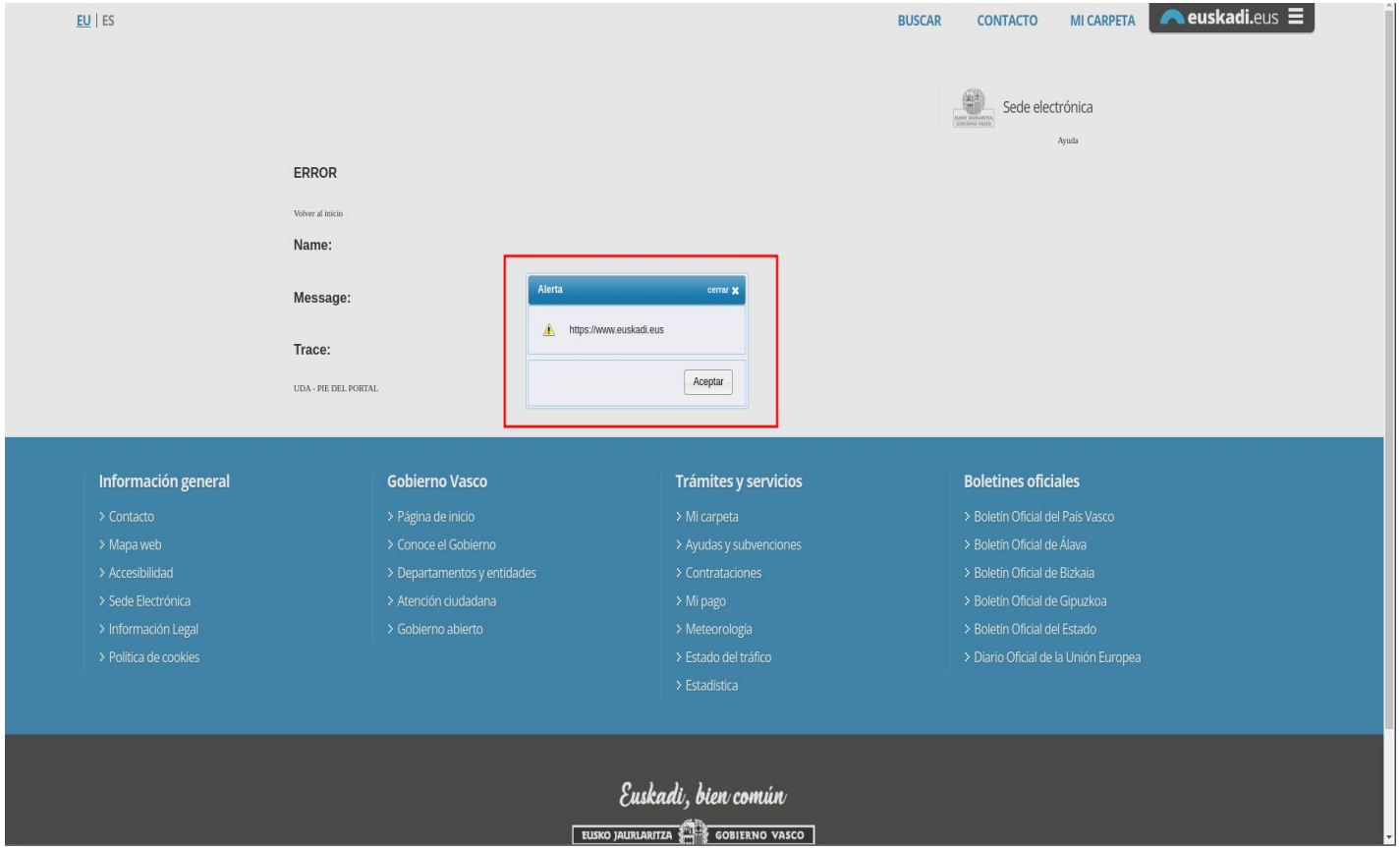
Trace:

org.springframework.web.servlet.FrameworkServlet.processRequest(FrameworkServlet.java:894)
org.springframework.web.servlet.FrameworkServlet.doGet(FrameworkServlet.java:778)
javax.servlet.http.HttpServlet.service(HttpServlet.java:707)
javax.servlet.http.HttpServlet.service(HttpServlet.java:821)
weblogic.servlet.internal.StubSecurityHelper$ServletServiceAction.run(StubSecurityHelper.java:227)
weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:301)
weblogic.servlet.internal.TailFilter.doFilter(TailFilter.java:27)
weblogic.servlet.internal.FilterChainImpl.doFilter(FilterChainImpl.java:61)
org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:76)
weblogic.servlet.internal.FilterChainImpl.doFilter(FilterChainImpl.java:61)
org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:330)
org.springframework.security.web.access.intercept.FilterSecurityInterceptor.invoke(FilterSecurityInterceptor.java:118)
org.springframework.security.web.access.intercept.FilterSecurityInterceptor.doFilter(FilterSecurityInterceptor.java:85)
org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:342)
com.ejie.x38.security.PreAuthenticateProcessingFilter.doFilter(PreAuthenticateProcessingFilter.java:94)
org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:342)
org.springframework.security.web.authentication.logout.LogoutFilter.doFilter(LogoutFilter.java:106)
org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:342)

XSS in euskadi.eus



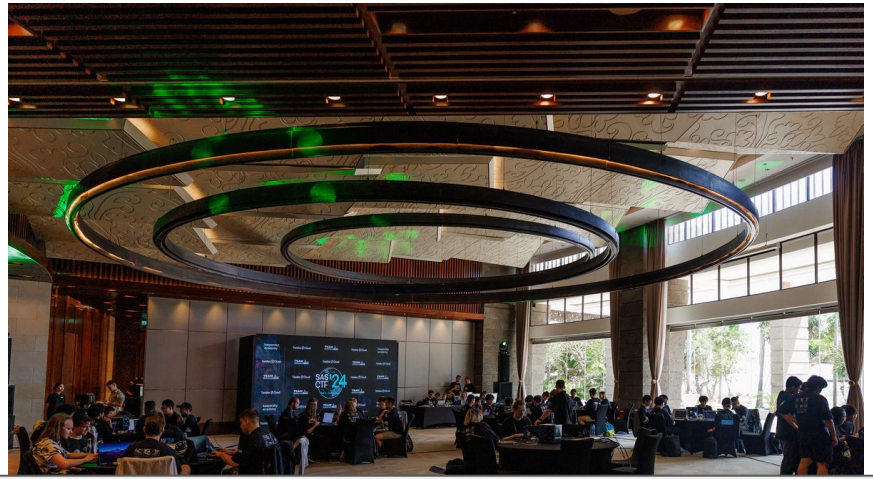
XSS in euskadi.eus

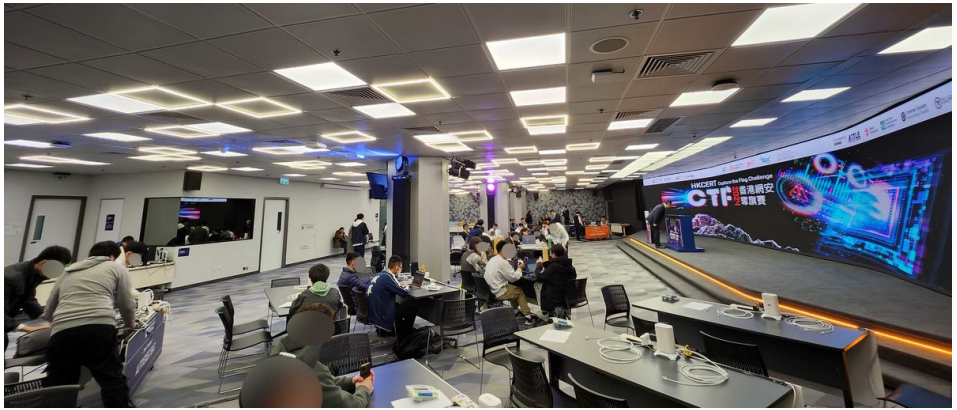


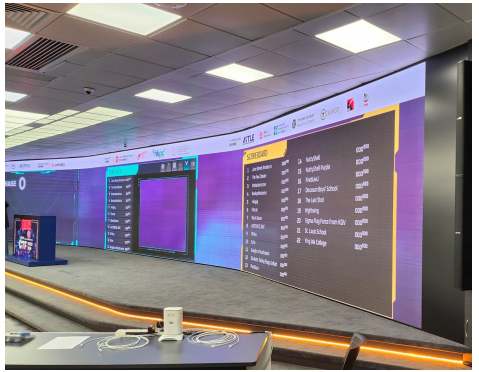
HOW TO LEARN CYBERSECURITY

Summer School 2025

Capture-The-Flag (CTF) competition







NEXT TALK!