

Powered by **plain**
concepts

dotNET 2023

Desarrollo Seguro para Muggles

#dotNET2023

dotNET2023

#dotNET2023

ORGANIZATION

**plain
concepts**

GOLD SPONSORS



intel

NTT DATA

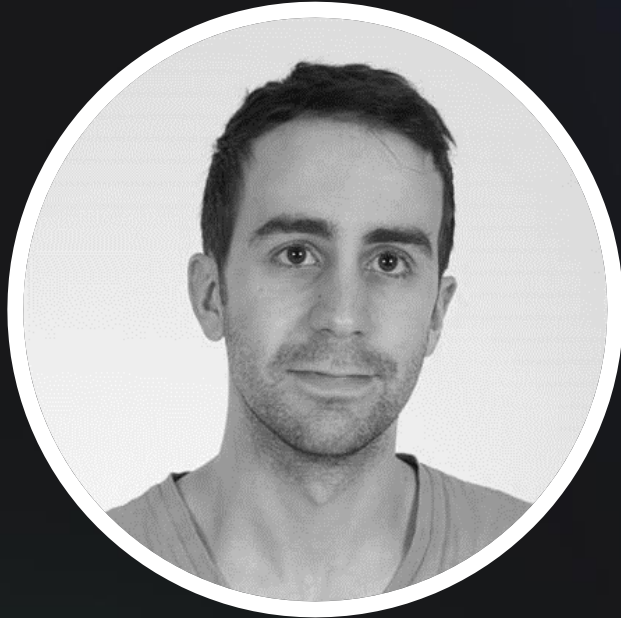
COLLABORATORS



ID boot
camps

dotNET2023

#dotNET2023



Diego Rodríguez Varela

Software Development Engineer

@diegorosec
drvarela@plainconcepts.com



Raúl Piracés Alastuey

Software Development Engineer

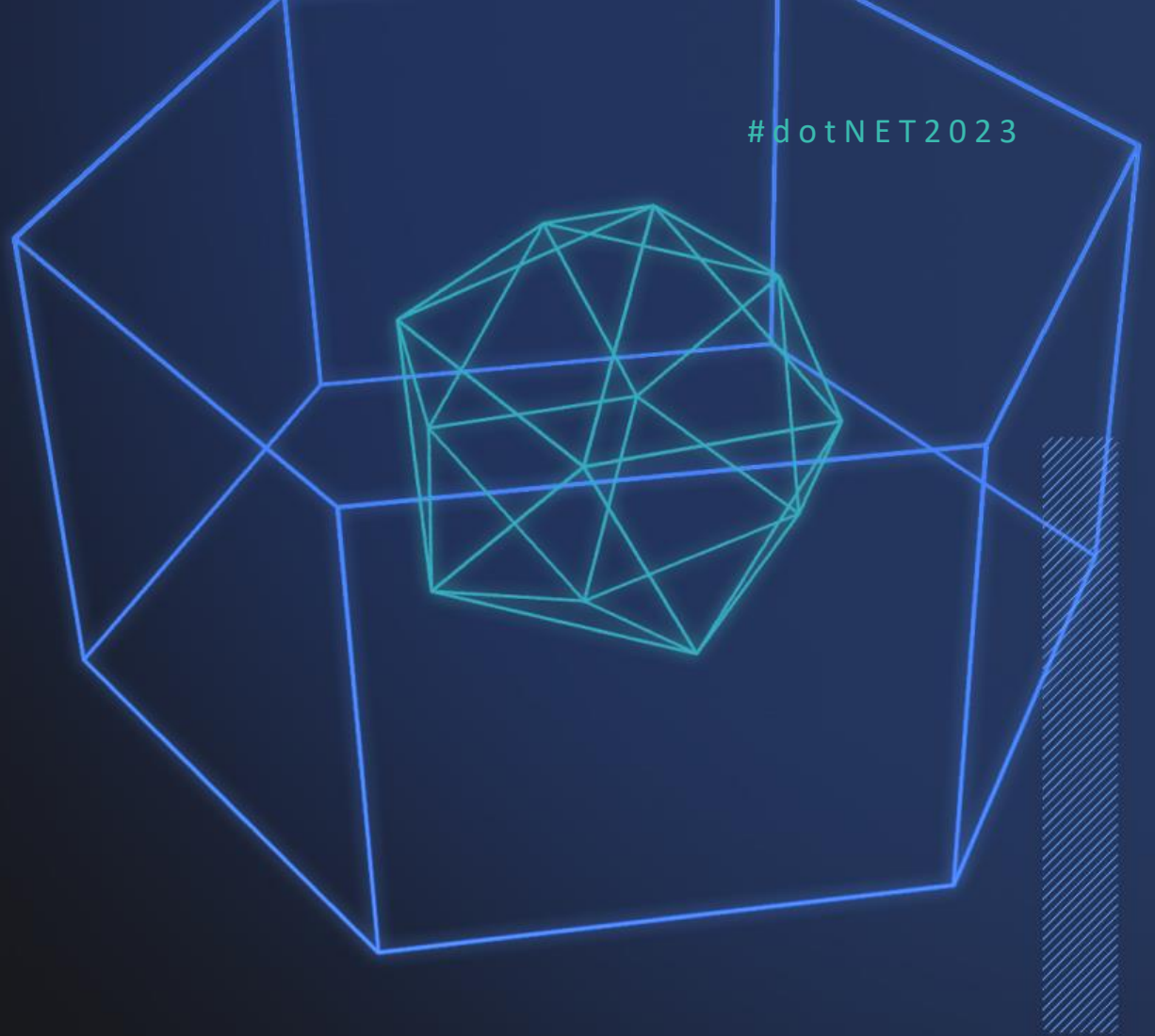
@piraces_
rpiraces@plainconcepts.com

dotNET2023

#dotNET2023

Agenda

- The State of Security Today
- Most common security errors
- Tips & Tooling



dotNET2023

#dotNET2023

Our main objective

Reducing significantly the attack surface of our applications with the right mindset, tips, tooling and a minimal effort...

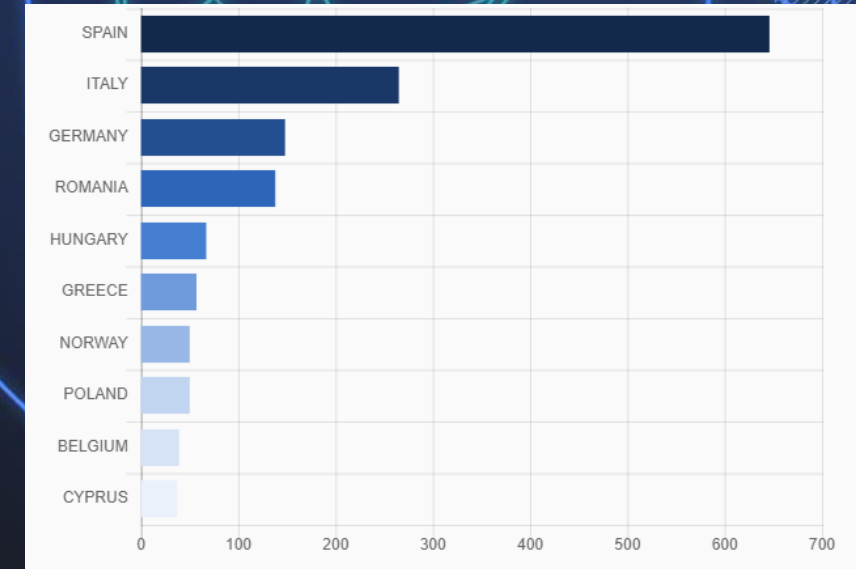
dotNET2023

#dotNET2023

The State of Security Today

The State of Security

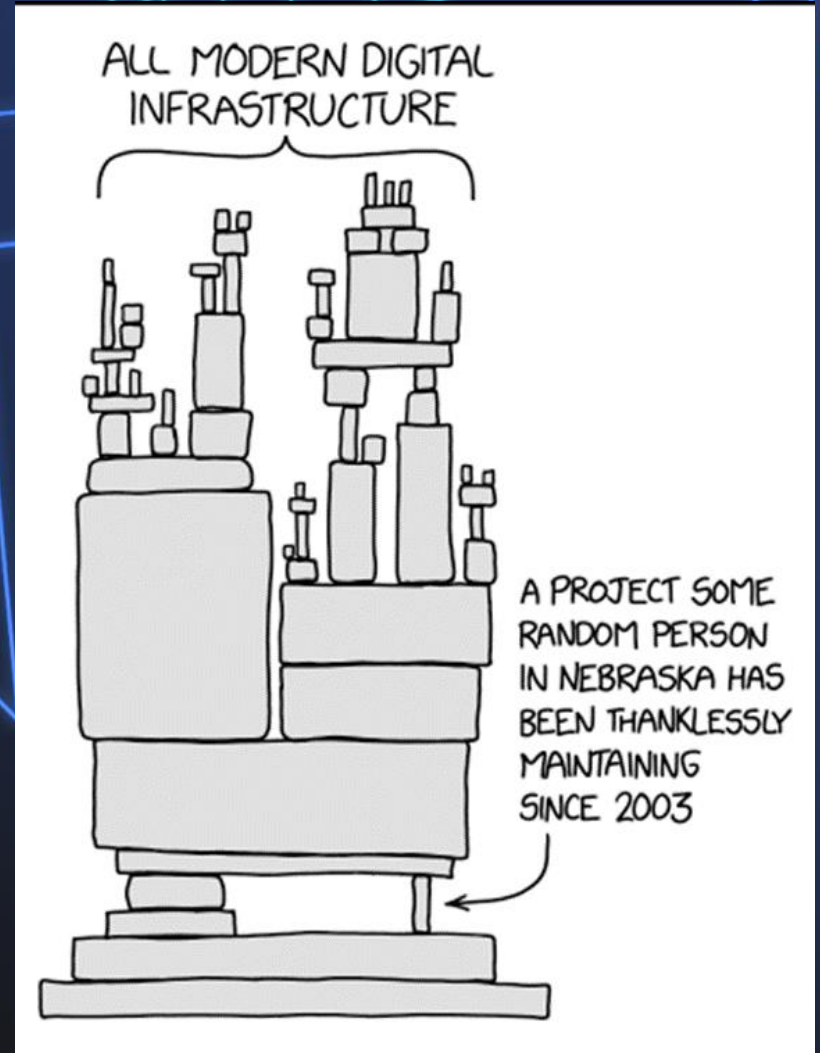
- Security numbers
 - **91.8%** orgs **compromised**
 - **85%** of applications with security issues
 - A **1:100** relation for security team vs developers
 - Allocate **11.9%** of IT budget to security
- Impact
 - Ransomware attacks **cost \$750,000** average
 - GDPR fines issued in Spain are almost €15 M (€59 M acc.)
 - 15.128 complaints in 2022 and raising (AEPD)



The State of Security Today

“The internet Relies on People Working for Free...”

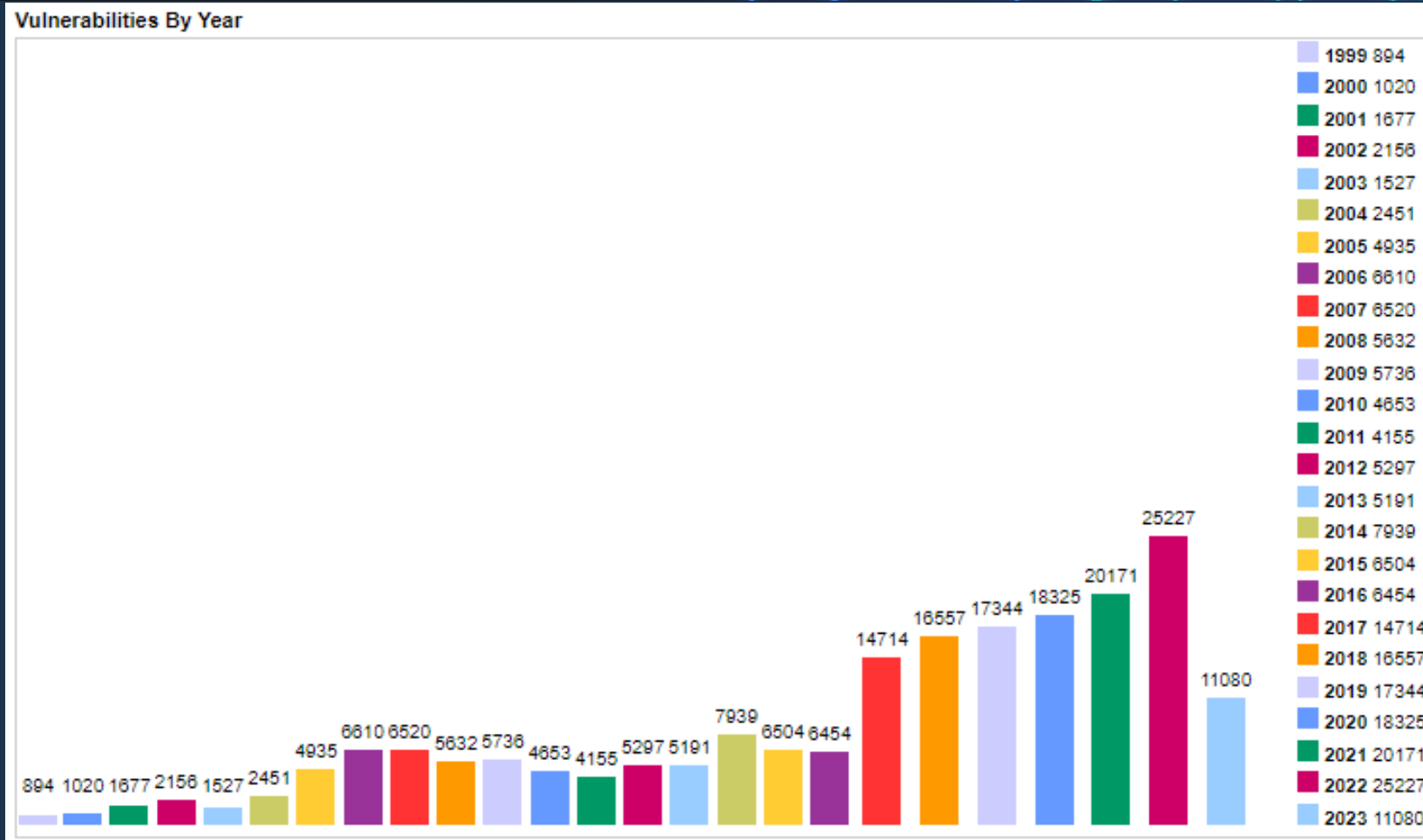
- cURL
 - 1 main developer
- OpenSSL
 - Heartbleed vulnerability
- Log4j
 - Log4Shell vulnerability
- faker.js, colors.js, core.js, left-pad



dotNET2023

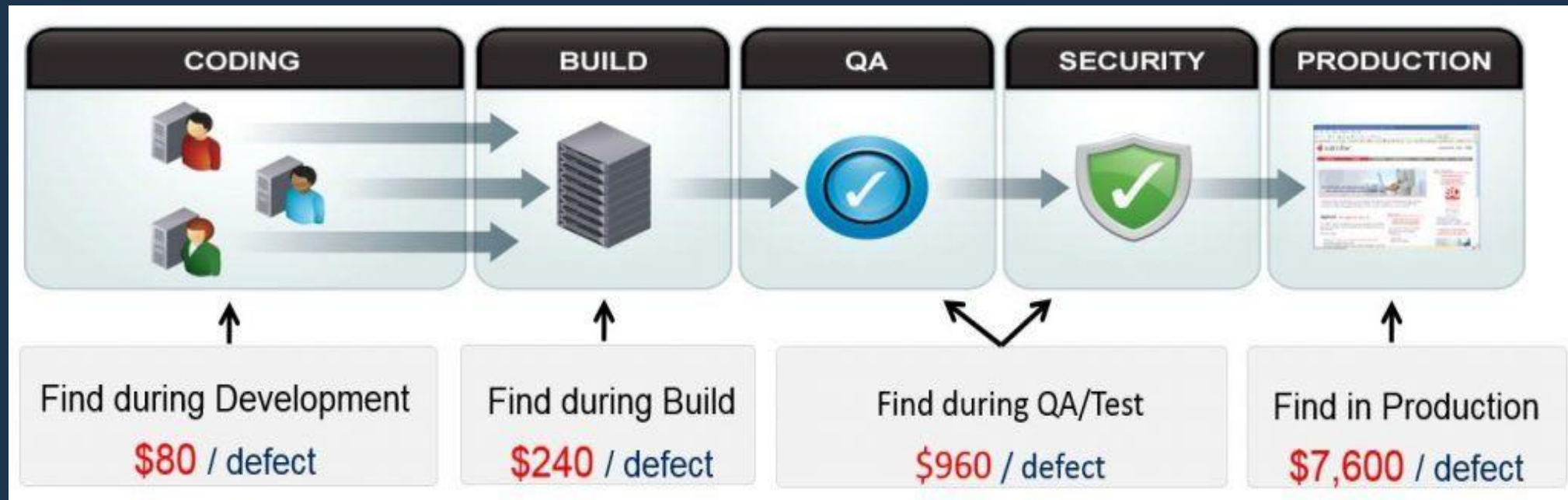
#dotNET2023

More lines of code -> More potential vulns.



Sources: cvedetails.com (as of day 19/06/2023)

Everyone wants to shift left security...



dotNET2023

#dotNET2023

Common Errors

Common Errors

Broken Access Control – Top 1

- Sequential Ids (enumeration-based attacks)
- Access to other roles, users... (guards / authorization / inverse proxy)
- Frontend vs Backend validation (Authentication / Authorization)



dotNET2023

#dotNET2023

Demo time!



Common Errors

HTTP Headers: CSP, CORS, HSTS

- Frame ancestors (anti-clickjacking, phishing attacks)
- CORS, making sure we respond to legit origins
- HSTS, ensuring connections through HTTPS
- Only loading what we need...
 - Loading libraries (XSS + External vuln. library)
- Transition to CSP: CSP Report Only

dotNET2023

#dotNET2023

Demo time!



Common Errors

CSRF, SSRF, Injections...

- Always validate user input (both in frontend and backend)
- Perform sanitization
- Parse everything
- Take care with regex...
- Use your framework's support for these kind of security threats

dotNET2023

#dotNET2023

Demo time!



Common Errors

Cookies

- SameSite, Strict or Lax?
- Secure?
- HTTPOnly?
- Do NOT allow to manage cookies via JavaScript

URL	Same Site	Same Origin
http://www.dotnetconfspain.com		
http://www.dotnetconfspain.com:80	✓	✓
http://dotnetconfspain.com	✓	✗
http://dotnetconfspain.com:8080	✓	✗
http://plain.dotnetconfspain.com	✓	✗
https://www.dotnetconfspain.com	✗	✗
http://www.dotnetconfspain.es	✗	✗



dotNET2023

#dotNET2023

Demo time!



Common Errors

Cryptography bad practices...

- **Do NOT use custom cryptography solutions...**
- Breaking weak keys in seconds...
- Key strength (How?)
- Salts + pepper
- Verify the certificate chain: SSL Labs, testssl.sh, sslscan, sslyze



dotNET2023

#dotNET2023

Demo time!



Common Errors

Insecure configurations

- Only expose publicly the minimum needed
- Read through documentation with detail
- Know your infrastructure
- Monitor, get insights and alerts (errors 401, 500, 404), unusual cases



Common Errors

Outdated components

- Keep your dependencies updated!
- Audit the dependencies & keep track of them
- Are your dependencies actively maintained?
- Do they respond to security issues?
- Supply chain security



Vulnerabilities report!

- npm audit

36 vulnerabilities (2 low, 11 moderate, 20 high, 3 critical)

To address issues that do not require attention, run:

```
npm audit fix
```

To address all issues (including breaking changes), run:

```
npm audit fix --force
```


Vulnerabilities report!

- NuGet

```
> dotnet list package --vulnerable --include-transitive
```

The following sources were used:

<https://api.nuget.org/v3/index.json>

Project ` .Host` has the following vulnerable packages

[net6.0]:

Top-level Package	Requested	Resolved	Severity	Advisory URL
> Apache.Avro	1.10.2	1.10.2	High	https://github.com/advisories/GHSA-868x-rg4c-cjqg
> Auth0.AuthenticationApi	6.5.3	6.5.3	High	https://github.com/advisories/GHSA-c9cg-q8r2-xvjq
> Azure.Storage.Blobs	12.9.1	12.9.1	Moderate	https://github.com/advisories/GHSA-64x4-9hc6-r2h6
> Azure.Storage.Queues	12.9.0	12.9.0	Moderate	https://github.com/advisories/GHSA-64x4-9hc6-r2h6

Transitive Package	Resolved	Severity	Advisory URL
> Microsoft.AspNetCore.Authentication.JwtBearer	3.1.2	Moderate	https://github.com/advisories/GHSA-q7cg-43mg-qp69
> NuGet.Common	6.3.1	High	https://github.com/advisories/GHSA-6qmf-mmc7-6c2p
> NuGet.Protocol	6.3.1	High	https://github.com/advisories/GHSA-6qmf-mmc7-6c2p
> System.Security.Cryptography.Pkcs	6.0.1	High	https://github.com/advisories/GHSA-555c-2p6r-68mm
> System.Text.RegularExpressions	4.3.0	High	https://github.com/advisories/GHSA-cmhx-cq75-c4mj

Common Errors

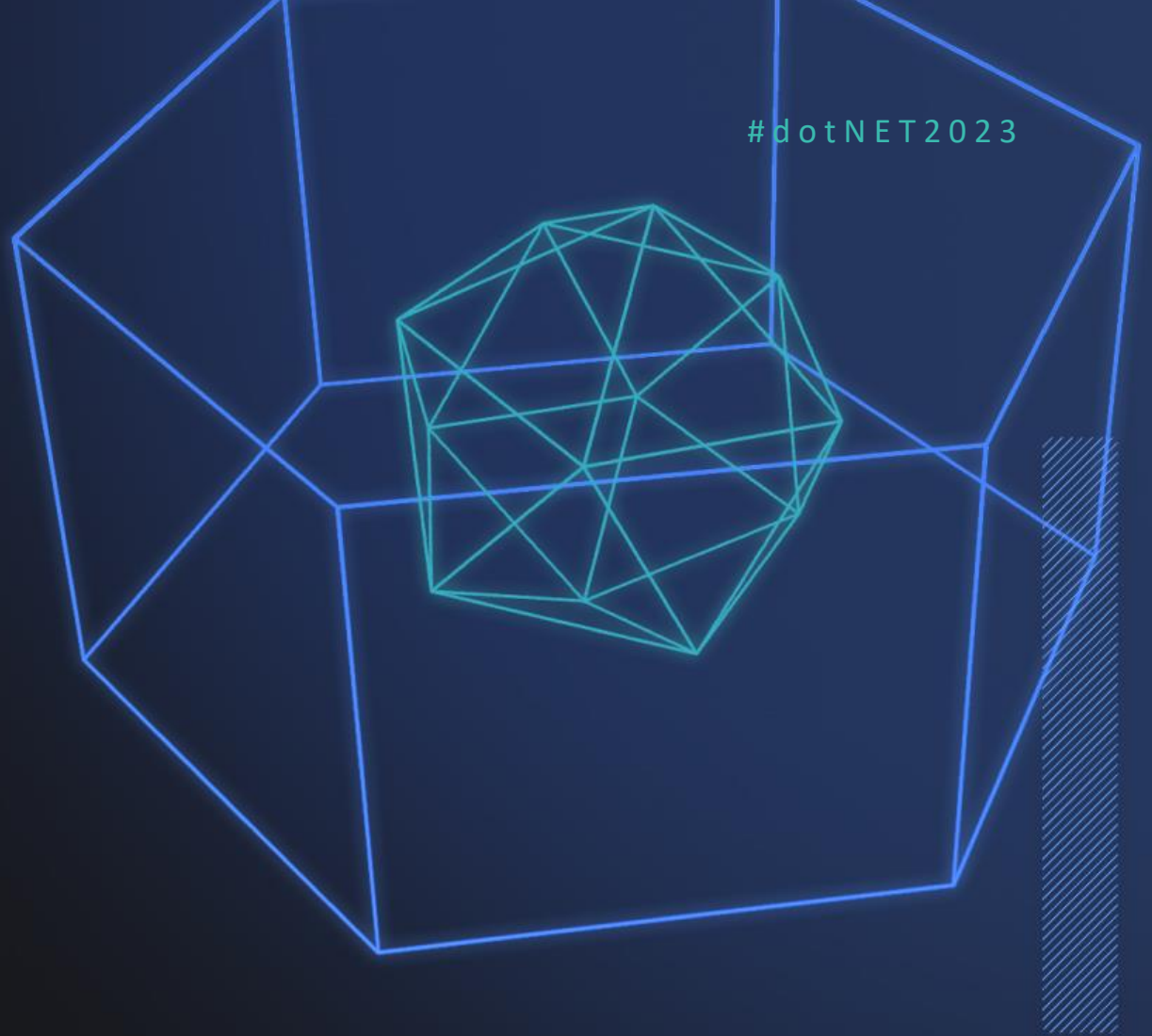
Authentication

- Minimum privilege principle
- Short-lived tokens
- Follow standards as they are (do not try to reinvent or augment them)
- Always validate signatures, use the recommended encryption methods, expose public keys

Some tips

Let's start shifting left with some tips

- **Minimum privilege** principle
- Understand/Evaluate your **attack surface**
 - Threat modelling can help
- Test for all cases (authentication/authorization)
- Do **NOT** assume anything
 - Defense in depth
- Review secondary applications, third party code
- Protect resources, variables and secrets
- Do NOT issue commands directly to the operating system



Tooling

Static Analysis vs Dynamic analysis

- Static Analysis:
 - Sonar, Snyk, Trivy (OSS)
- Dynamic Analysis:
 - Burp Suite, OWASP ZAP, OpenVAS, Acunetix, Nessus
- Check it out: <https://book.devsec.fyi/>



dotNET2023

#dotNET2023

Demo time!

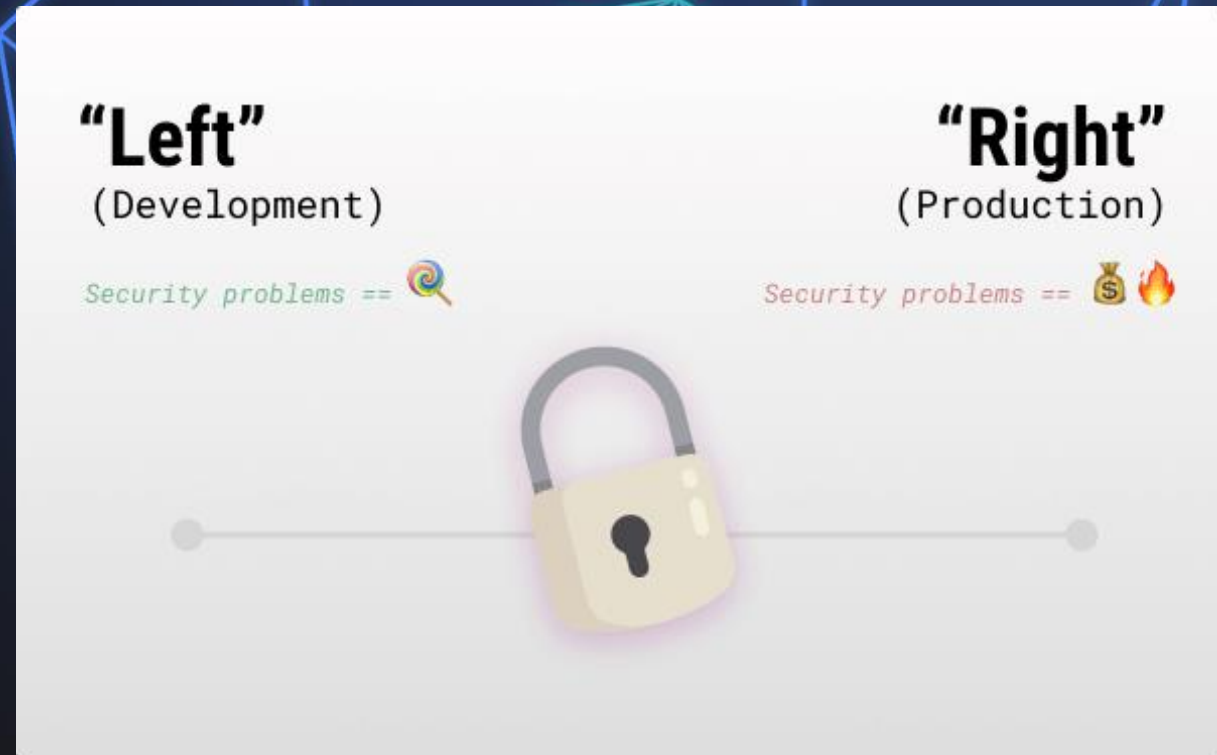


Conclusions...

- Current security “performance”
- Most common security problems
- Tips & tools to protect ourselves

Note

- There is no “silver bullet” against attacks
- “Shifting Security Left” is a mindset, not a tool



dotNET2023

Thank you!

Powered by **plain**
concepts



NTT DATA

