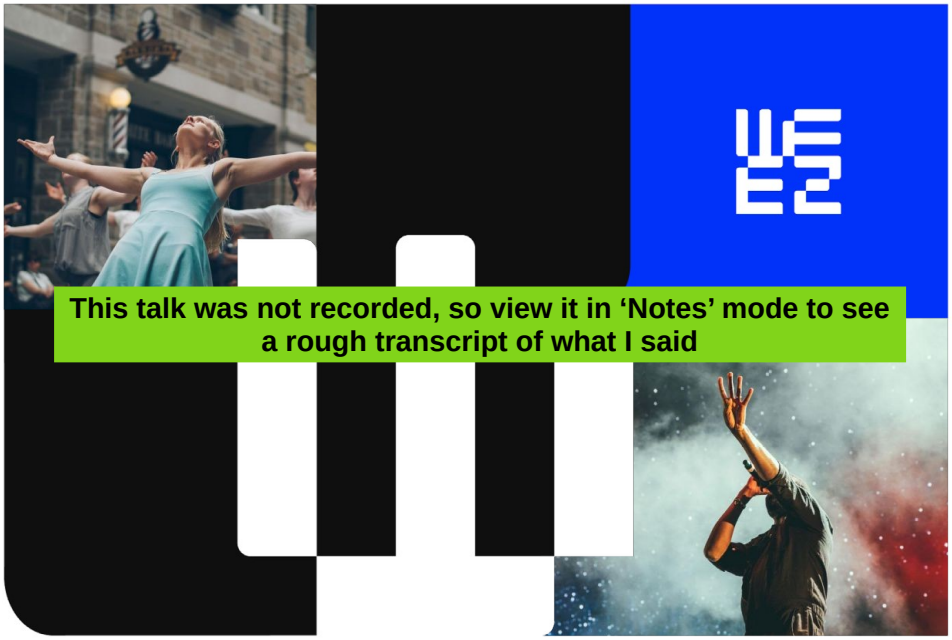


Security 101

weezcrew

0. Introduction

Security 101

Le rosbif, c’est moi

- **Ben Goldsworthy** 🇬🇧
- RunEvent/WeezCrew (*née* Recrewteer) in Montpellier since start of 2024
- IT guy™ since ~2011
  - Started programming in school (VB.NET 🤮)
  - Professionally since 2014
  - 2017: BSc (Hons) Computer Science
- Tech Day 2025: “Accessibility and Semantic HTML” [1]

weezcrew

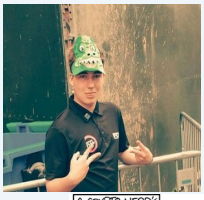
[1] Goldsworthy, B. “Accessibility and Semantic HTML” from Weez TechDay 2025 <https://bengoldsworthy.net/portfolio/presentations/accessibility-semantic-html/>

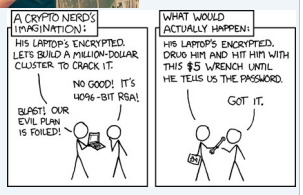
Those who saw my previous talk on accessibility know what they’re in for: a lot of information, quite dense, with the expectation that you will look through the slides later in your own time and follow the links that interest you

Security 101

My Security Background

- Started with (non-cyber) security
  - 2016–2018: Event Security
  - 2016–2026: British Army Reserves
- 2017–2022: Worked in cyber security
- 2018: MSc Cyber Security
  - Security Lancaster
  - Technology, Terrorism & Armed Conflict in the 21st Century (TTAC21)
- Now I'm back to development primarily



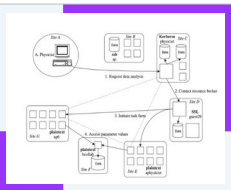


I started in non-cyber security, which I think is always important to not lose sight of — as the *xkcd* shows, you can have the fanciest technical protections possible (e.g. the Weez office fancy front door access device) but they're useless if they are trivially bypassed (e.g. someone just waits for someone else to open the door and then follows them in)

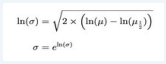
Security 101

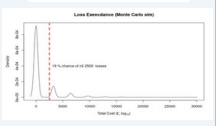
How Have I Security'd?

- ➔ Enterprise security architecture design
- ➔ Security assessments of existing systems
- ➔ Research into quantitative risk assessment and threat intelligence [1]
  - Monte Carlo simulations
- ➔ Training & education
  - E-learning
  - Tabletop exercising/wargaming [2]
- ➔ Consulting
  - Businesses
  - Charities/NGOs



CC BY-SA 4.0, Ian Foster, Carl Kesselman, Gene Tsudik, Steven Tuecke <https://commons.wikimedia.org/wiki/File:Example_from_%22A_Security_Architecture_for_Computational_Grids%22.png>





CC BY-SA 4.0, David Vose

[1] Goldsworthy, B. "Threat Intelligence Analytical Software", *bengoldsworthy.net* <https://bengoldsworthy.net/portfolio/writings/threat-intelligence-analytical-software/ >; "How to Measure Anything in Cybersecurity Risk", *bengoldsworthy.net* <https://bengoldsworthy.net/portfolio/presentations/how-to-measure-anything-in-cybersecurity-risk/>; "How to Measure Anything in Cybersecurity Risk II: Measure Harder", *bengoldsworthy.net* <https://bengoldsworthy.net/portfolio/presentations/how-to-measure-anything-in-cybersecurity-risk-ii/>
[2] Goldsworthy, B. "To Invest or Defend", *bengoldsworthy.net* <https://bengoldsworthy.net/portfolio/presentations/to-invest-or-defend/>

'Security' === blind men describing an elephant:

> The first person, whose hand landed on the trunk, said, "This being is like a thick snake". For another one whose hand reached its ear, it seemed like a kind of fan. As for another person, whose hand was upon its leg, said, the elephant is a pillar like a tree-trunk. The blind man who placed his hand upon its side said the elephant, "is a wall". Another who felt its tail, described it as a rope. The last felt its tusk, stating the elephant is that which is hard, smooth and like a spear.

Enterprise security design for a financial client: legal compliance requirements, very high-level, not worried about implementation but just looking at the whole abstract system design and identifying what security controls need to go where

Assessment of existing systems: pentesting, but also just running through basic checklists (e.g. Cyber Essentials in the UK). Assessing small projects across a huge portfolio that otherwise get ignored (see also: Weezevent GitHub org has **295** repos)

Quantitative risk assessment: commonly-used likelihood × impact risk assessment matrix is effectively rigour-washing but doesn't actually mean anything and is super subjective. I worked on a project that was investigating using tools and techniques from insurance/actuarial sciences (e.g. Monte Carlo simulations) and real-world breach data (e.g. from the UK *Cyber Security Breaches Survey*) to be able to provide evidence-backed statements like 'you have a 19% chance of a breach that costs you more than £10,000) — it's not magic! Check out the linked references for more on that.

Training & education (like right now!): producing e-learning courses, tabletop exercising simulated security incidents (also known as 'professional wargaming', which generally means there is an element of adversarial unpredictability involved)

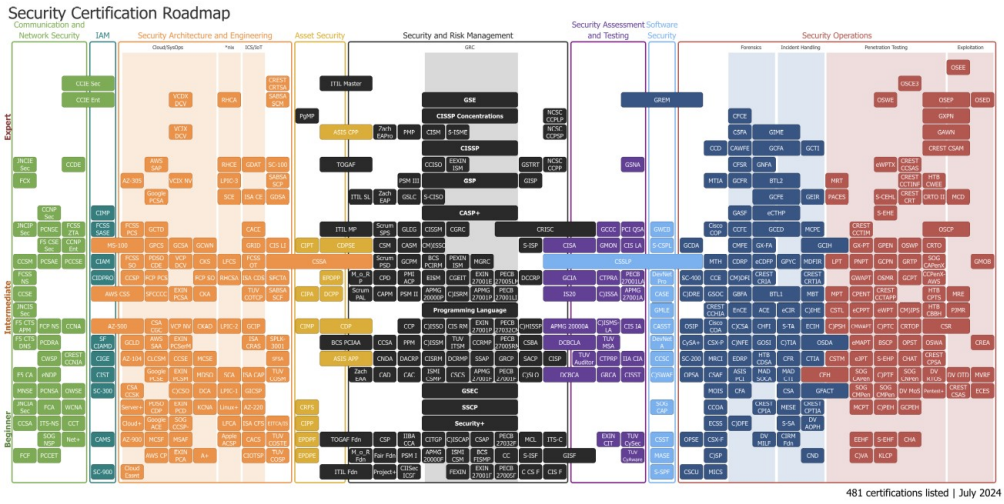
Consulting businesses: security costs money, and until something goes wrong the cost of a breach is only theoretical, so need to be able to justify why they should spend on security anyway

Consulting charities/NGOs: very different contexts, e.g. working with children, non-technical volunteers, healthcare compliance requirements

Security 101

weezcrew

Security is for Everyone



[1] Jerimy, P. “Security Certification Roadmap”, *OWASP* <https://pauljerimy.com/security-certification-roadmap/>
[2] ”Chartered Cyber Security Professional (ChCSP)”, *UK Cyber Security Council* <https://www.ukcybersecuritycouncil.org.uk/for-individuals/become-professionally-registered/professional-titles/chartered>

Cybersecurity field is super fragmented and non-standardised, this graph shows how many professional certifications exist from different providers. Advantage of that is that there is no single way in and that means you can approach the field from different angles, follow your interests, and generally be recognised for your abilities rather than your credentials (similar to development in that sense)

This has been recognised for a least a decade, see https://ieeexplore.ieee.org/document/8220378

Slight caveat for the UK: there has been a lot of work over the past few years to standardise a single ‘Chartered’ professional status, which has its pros and cons

Security 101

This Presentation

1) What is OWASP?
2) What Does ‘Security’ Entail?
3) Threat Modeling
4) Secure Development Lifecycle (SSDLC)
5) Security Requirements
6) Common Issues
7) Vulnerability Management

weezcrew

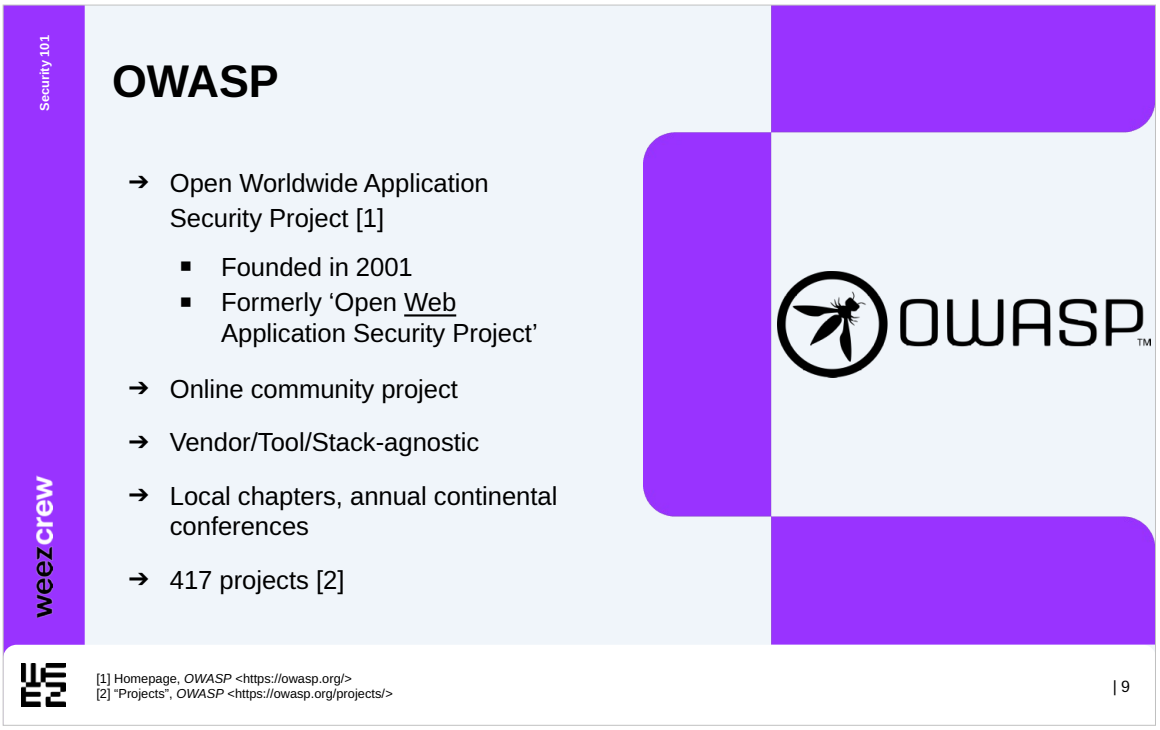
We’re going to start at the very high-level, organisational/strategic stuff.

Then work down, through things a team lead needs to think about, all the way down to duties for individual devs

We’re also going to cover issues that have existed for decades, all the way to the very latest developments in the field

weezcrew

1. What is OWASP?

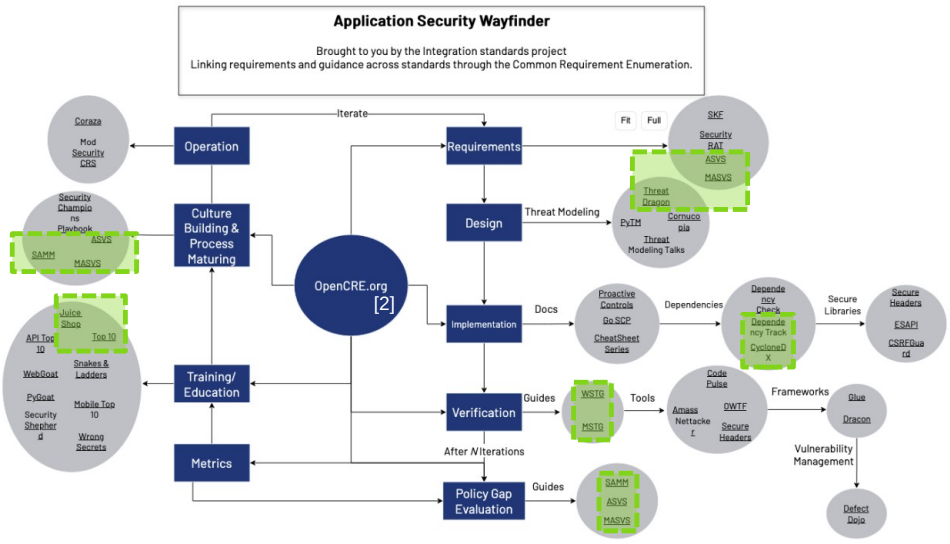


(When I asked who had heard of OWASP, only the UK devs put their hands up. So maybe not as ‘worldwide’ as I thought, but there *is* a France chapter based in Paris ht tps://owasp.org/www-chapter-france/)

For most/all OWASP tools there will exist similar alternatives from other vendors, usually paid. But by focussing on OWASP here I can highlight how they all work together, and I’m not tied to any one vendor or ecosystem. OWASP also tries to be tech-agnostic (because they’re not trying to sell you their other projects, unlike e.g. AWS) which is good because the Weez portfolio is very diverse

Huge number of projects in various states of maturity, obviously I can’t cover all of them today

OWASP Projects



[1] "OWASP Integration Standards", *OWASP* <https://owasp.org/www-project-integration-standards/>
[2] "Open Cybersecurity Requirement Enumeration", <https://opencre.org/>

Here is a map of some of the higher-profile/more-mature projects, and I've highlighted the ones I'll talk about today

Re: CRE this is an interesting project looking at creating a common mapping of security requirements across different standards (e.g. OWASP, NIST, ISO, etc.). But as always, https://xkcd.com/927/

weezcrew

2. What Does ‘Security’ Entail?

Security 101

OWASP SAMM

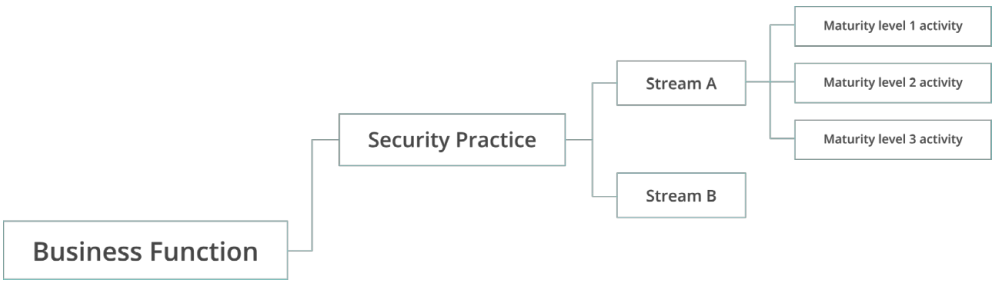
- Security Assurance Maturity Model [1]
- ‘SAMM is a prescriptive model... **The solution details are easy enough to follow even for non-security personnel.** It helps organizations analyze their current software security practices, **build a security program in defined iterations**, show progressive improvements in secure practices, and **define and measure security-related activities**.’ [2]

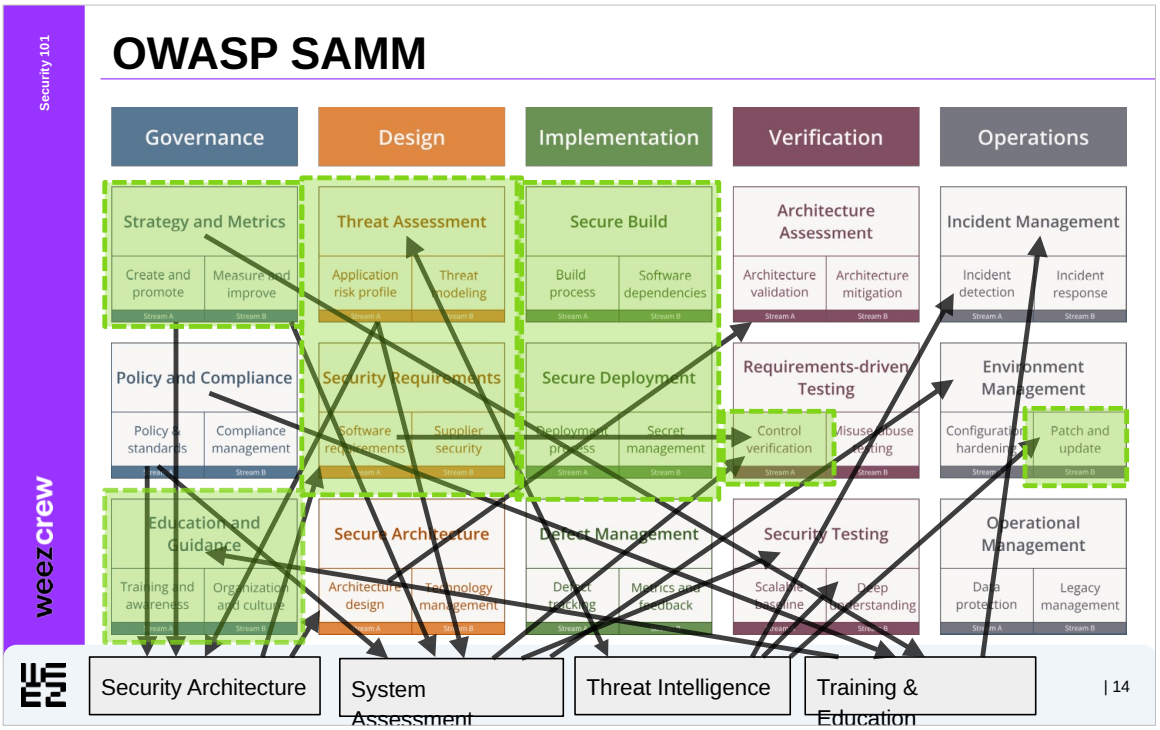
weezcrew

[1] Homepage, *OWASP SAMM* <https://owaspsamm.org/>
[2] “About”, *OWASP SAMM* <https://owaspsamm.org/about/>

Quarter way point

OWASP SAMM


Business Function
Security Practice
Stream A
Stream B
Maturity level 1 activity
Maturity level 2 activity
Maturity level 3 activity



How have some of my past projects mapped onto the SAMM areas?

Security architecture: strategy and legal requirements constrained architecture design, which in turn generated architecture and requirements against which the app could be built (and then tested)

System assessment: again strategy and legal requirements constrained (e.g. against which standard was I assessing the systems? How often?) which in turn verified that required controls were actually in place, and that the real environment in which the apps were running was configured as the security architecture expected

Threat intelligence: bi-directional with threat assessment, as initial threat assessment determines intelligence sought, which in turns informs threat assessment, which in turn informs intelligence sought, etc. etc. Intelligence highlights new risks and incidents, which informs how we respond to them, as well as vulnerable dependencies etc. to update

Training & education: again influenced by company policy and legal requirements, but also influences things like incident response because you can respond better with a well-drilled team following clearly-defined playbooks in which everybody knows their roles and responsibilities (especially if they've just been woken up by an alert at 3am)

Security 101

weezcrew

OWASP SAMM

Model | Governance | Education and Guidance

The Education and Guidance (EG) practice focuses on arming personnel involved in the software lifecycle with knowledge and resources to design, develop, and deploy secure software. With improved access to information, project teams can proactively identify and mitigate the specific security risks that apply to their organization.

One major theme for improvement across the Objectives is providing training for employees and increasing their security awareness, either through instructor-led sessions or computer-based modules. As an organization progresses, it builds a broad base of training starting with developers and moving to other roles, culminating with the addition of role-based training to ensure applicability and effectiveness.

In addition to training, this practice also requires the organization to make a significant investment in improving organizational culture to promote application security through collaboration between teams. Collaboration tools and increased transparency between technologies and tools support this approach to improve the security of the applications.

Maturity level		Stream A Training and Awareness	Stream B Organization and Culture
1	Offer staff access to resources around the topics of secure development and deployment.	Provide security awareness training for all personnel involved in software development.	Identify a "Security Champion" within each development team.
2	Educate all personnel in the software lifecycle with technology and role-specific guidance on secure development.	Offer technology and role-specific guidance, including security nuances of each language and platform.	Develop a secure software center of excellen promoting thought leadership among developers and architects.
3	Develop in-house training programs facilitated by developers across different teams.	Standardized in-house guidance around the organization's secure software development standards. [1]	Build a secure software community including all organization people involved in software security.

[1] "`Weezevent/recrewteer` Wiki", *GitHub* <https://github.com/Weezevent/recrewteer/wiki>

Here's an example of a SAMM Security Practice.

Look at maturity level 1, stream A: that's us right now!

Look at stream B: any volunteers?

For maturity level 3, stream A: look at the Crew GitHub repo, where we've made a start on documenting our own SDLC in the Wiki

weezcrew

3. Threat Modeling

Threat Modeling

‘Threat modeling is analyzing representations of a system to highlight concerns about security and privacy characteristics.

At the highest levels, when we threat model, **we ask four key questions**:

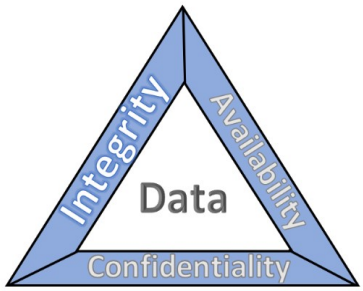
- What are we working on?
- What can go wrong?
- What are we going to do about it?
- Did we do a good enough job?’ [1]

[1] “Threat Modeling Manifesto” <https://www.threatmodelingmanifesto.org/>

Security 101

Information Security

- CIA Triad [1]
  - **C**onfidentiality
  - **I**ntegrity
  - **A**vailability
  - LIMITED
- STRIDE [2]
  - **S**poofing
  - **T**ampering
  - **R**epudiation
  - **I**nformation disclosure
  - **D**enial of Service
  - **E**levation of Privilege



weezcrew

[1] "CIA Triad", *Fortinet* <https://www.fortinet.com/uk/resources/cyberglossary/cia-triad>
[2] "Conducting a STRIDE-based threat analysis", *GOV.UK* <https://www.gov.uk/government/publications/secure-connected-places-playbook-documents/conducting-a-stride-based-threat-analysis#introduction-and-definitions>

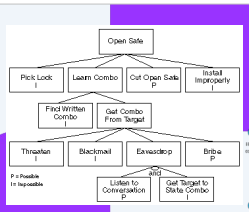
CIA is a model of what we need to protect, but very old and limited

STRIDE is a mnemonic for things an attacker might want to do

Security 101

Tactics, Techniques & Procedures (TTP)

- ➔ Attack Trees [1]
- ➔ Cyber Kill Chain [2]
- ➔ Common Attack Pattern Enumerations and Classifications (CAPEC) [3]
- ➔ MITRE ATT&CK [4]



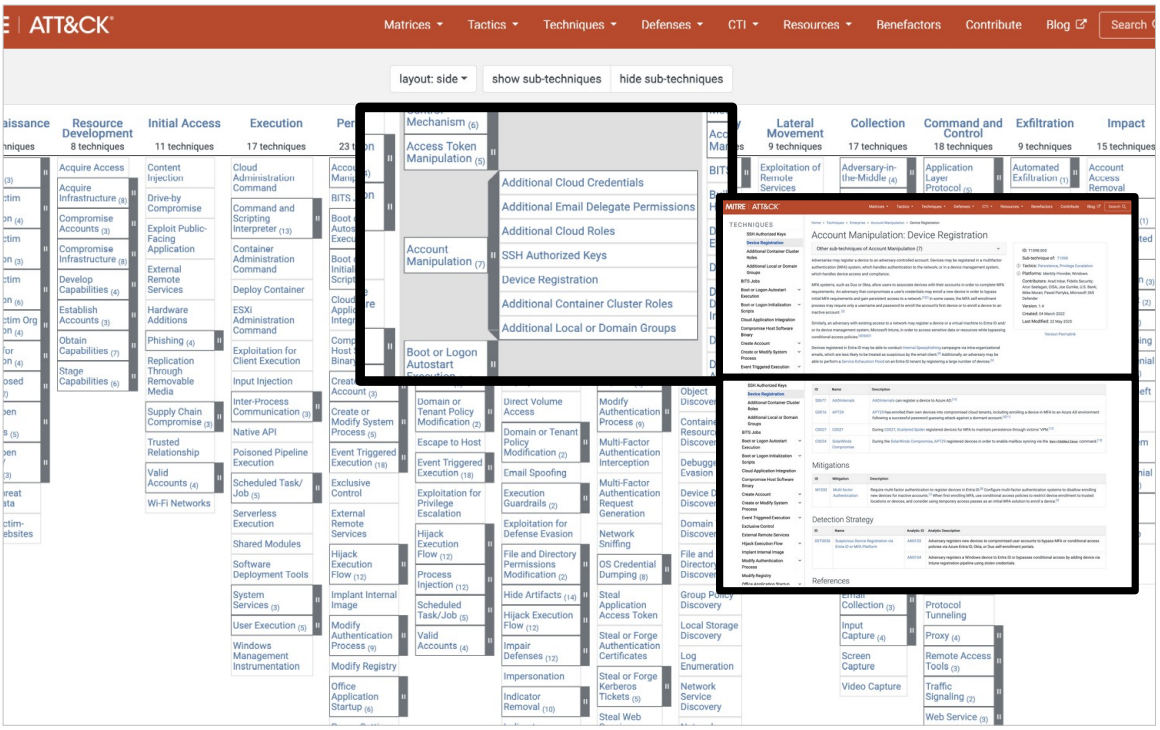
ATT&CK®

[1] Schneier, B. “Attack Trees”, *Schneier on Security* <https://www.schneier.com/academic/archives/1999/12/attack_trees.html>
[2] “Cyber Kill-Chain”, *Lockheed-Martin* <https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html>
[3] “About CAPEC”, *CAPEC* <https://capec.mitre.org/about/new_to_capec.html>
[4] Homepage, *MITRE ATT&CK* <https://attack.mitre.org/>

STRIDE tells use *what* the attacker wants, but *how* will they achieve it?

Attack trees are quite a good simple way

CAPEC I won’t go into, except to say that it’s part of a trio with CWE (which I’ll touch on again later) and CVE (which I’m sure all you devs are familiar with)

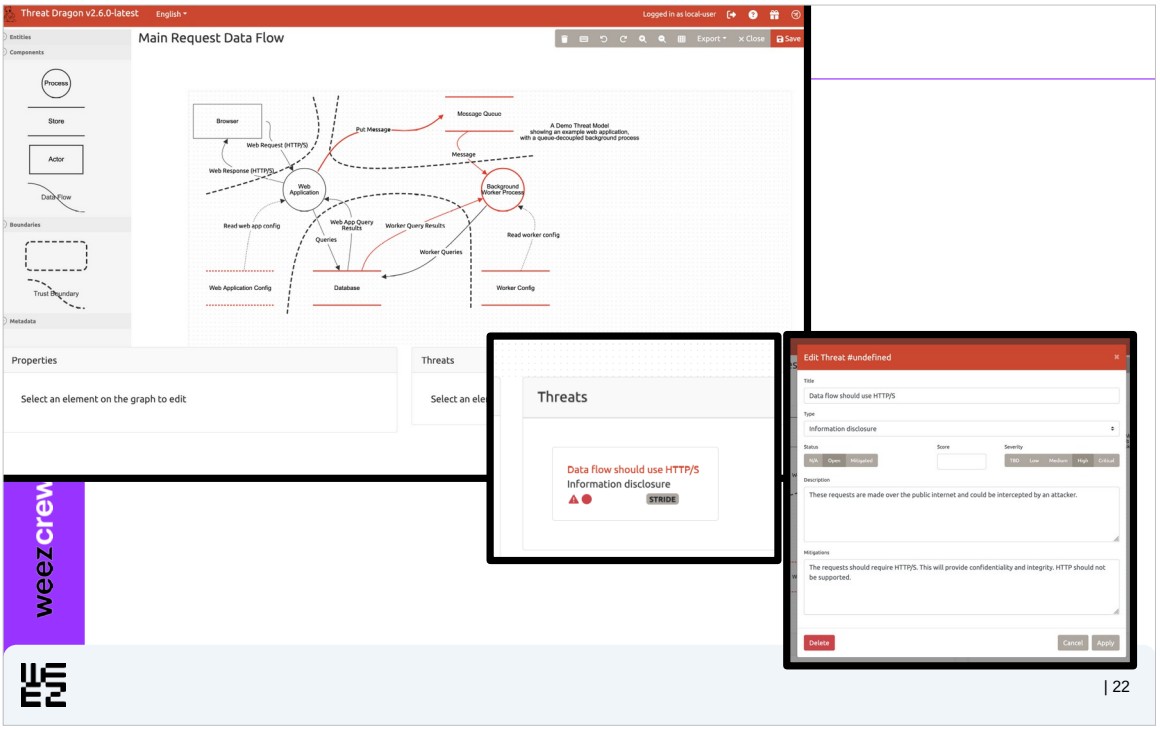


MITRE ATT&CK is a super comprehensive library of TTPs, really good learning resource

OWASP Threat Dragon

➔ ‘OWASP Threat Dragon is a modeling tool used to **create threat model diagrams** as part of a secure development lifecycle.’ [1]

[1] “OWASP Threat Dragon”, *OWASP* <https://owasp.org/www-project-threat-dragon/>



Simple example diagram

Dashed line represents trust boundary. The Internet is always untrusted, which means you need verification and validation at each boundary with it

After documenting data flows, we can identify how they might be attacked and then how we can resist that (e.g. using HTTPS for Internet traffic)

Security 101

OWASP Threat Model Library

- ‘...the first, open-sourced, structured, peer-reviewed threat modeling dataset’ [1]
- Incubator project, schema was only defined in 2025 [2]

weezcrew

[1] “OWASP Threat Model Library”, *OWASP* <https://owasp.org/www-project-threat-model-library/>
[2] Vukmirovic, P. “Breaking the Black Box: Using the OWASP Threat Model Library to End Security via Obscurity” from OWASP Global AppSec US 2025 < https://www.youtube.com/watch?v=7HTj28M05oc>

Interesting OWASP talk linked at the bottom about organisations designing and then sharing threat models between one another

weezcrew

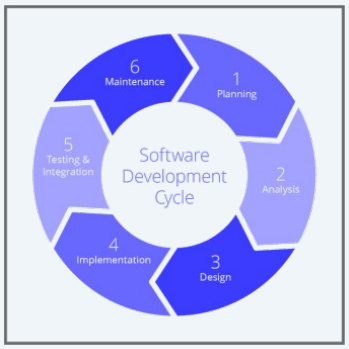
4. Secure Software Development Lifecycle (SSDLC)

Half way point

Security 101

Software Development Lifecycle (SDLC)

- ‘Almost all modern software is **developed in an iterative manner** passing through phase to phase, such as identifying customer requirements, implementation and test. These phases are revisited in a **cyclic manner** throughout the lifetime of the application.’ [1]
- ‘**Security can be embedded in a SDLC** by building on top of previous steps with policies, controls, designs, implementations and tests...’ [2]



A typical SDLC representation

weezcrew

[1] “Secure development and integration”, *OWASP Developer Guide* <https://devguide.owasp.org/en/02-foundations/02-secure-development/>
[2] “OWASP in SDLC”, *OWASP* <https://owasp.org/www-project-integration-standards/writeups/owasp_in_sdlc/>

Whether it’s documented or not, all of us devs follow some form of SDLC that repeats cyclically. Every stage of it has security considerations, and at an organisational level there are many benefits to making these explicit by actually documenting your process (but only if that documentation aligns with the actual process followed)

Security Development Lifecycle (SDL)

➔ SDL Practices: [1]

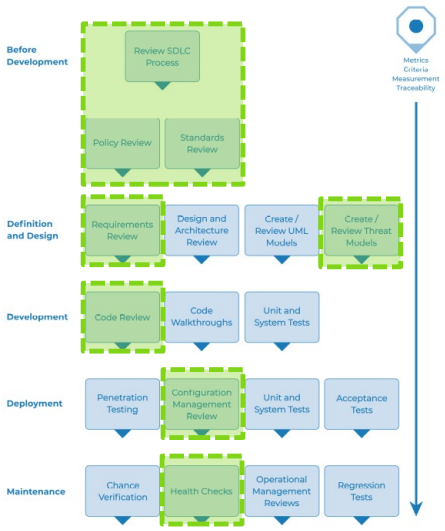
1) Establish security standards, metrics, and governance
2) **Require use of proven security features, languages, and frameworks**
3) **Perform security design review and threat modeling**
4) Define and use cryptography standards
5) **Secure the software supply chain**
6) Secure the engineering environment
7) **Perform security testing**
8) Ensure operational platform security
9) Implement security monitoring and response
10) **Provide security training**

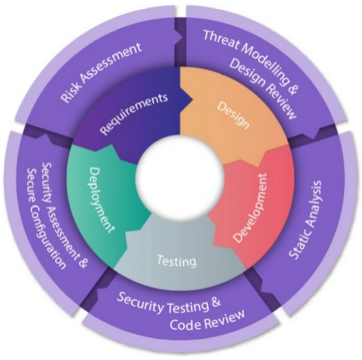
[1] "Security Development Lifecycle (SDL) Practices", *Microsoft Security Engineering* <https://www.microsoft.com/en-us/securityengineering/sdl/practices>

Security

weezcrew

Secure Software Development Lifecycle (SSDLC)





[1] "Web Security Testing Framework" in *OWASP Web Security Testing Guide (WSTG)* <https://owasp.org/www-project-web-security-testing-guide/v42/3-The_OWASP_Testing_Framework/0-The_Web_Security_Testing_Framework>
[2] "Secure Software Development Lifecycle (SSDLC)", *Snyk* <https://snyk.io/articles/secure-sdlc/>

OWASP Web Security Testing Framework through through each stage of a representative SDLC and highlights the security tasks to perform in each

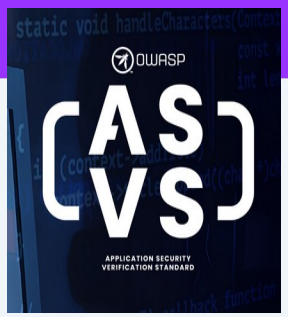
weezcrew

5. Security Requirements

Security 101

Security Requirements

- ‘A requirement that specifies the functional, assurance, and strength characteristics for a mechanism, system, or system element.’ [1]
- OWASP Application Security Verification Standard (ASVS) [2]
  - 300+ controls across 17 sections
  - Three levels (most should aim for Level 2)
  - ‘Documented security decisions’
  - Recommendations (e.g. `security.txt`/RFC 9116)
- NIST SP 800-53: 1,100+ controls [3]



ACCEPT	AVOID
REDUCE	TRANSFER

[1] *Developing Cyber-Resilient Systems: A Systems Security Engineering Approach* (NIST SP 800-16), *NIST* <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-160v2r1.pdf>
[2] “OWASP Application Security Verification Standard (ASVS)”, *OWASP* <https://owasp.org/www-project-application-security-verification-standard/>
[3] *Security and Privacy Controls for Information Systems and Organizations* (NIST SP 800-53), *NIST* <https://csrc.nist.gov/pubs/sp/800/53/r5/upd1/final>

ASVS is probably a good level of complexity for most uses (e.g. NIST 800-53 is much larger, but unwieldy)

Documented security decisions: when we encounter a risk, we have four options: to accept it as unavoidable; to avoid it completely (if possible); to reduce it through security controls; or to transfer it to another party (e.g. through cyber breach insurance)

When you make a decision to go against a requirement, it’s important to document: why, on whose authority, and a review schedule (in case something has changed in a year and you can now fulfil the requirement)

Application Security Verification Standard (ASVS)

#	Description	Level
6.2.8	Verify that the application verifies the user's password exactly as received from the user, without any modifications such as truncation or case transformation.	1
6.2.9	Verify that passwords of at least 64 characters are permitted.	2
6.2.10	Verify that a user's password stays valid until it is discovered to be compromised or the user rotates it. The application must not require periodic credential rotation.	2
6.2.11	Verify that the documented list of context specific words is used to prevent easy to guess passwords being created.	2
6.2.12	Verify that passwords submitted during account registration or password changes are checked against a set of breached passwords.	2

contra PCI DSS 4.0

Level 1 is clearly very simple, everyone should be achieving it (are we?)

Level 2 is good enough for most purposes. But see 6.2.10: PCI DSS 4.0 (which is required for anyone handling credit card data) mandates 90-day password resets unless MFA is used (https://www.compassitc.com/blog/pci-dss-4.0-password-requirements-a-guide-to-compliance#:~:text=PCI%20DSS%204%2E0%20maintains%20the,real%2Dtime%20based%20on%20behavioral%20factors%2E). There are no silver bullets in security, and you often have to make compromises between mutually exclusive conditions (see again the 'documented security decisions' in the last slide)

Although in this case, enabling mandatory MFA would be the better option

weezcrew

6. Common Issues

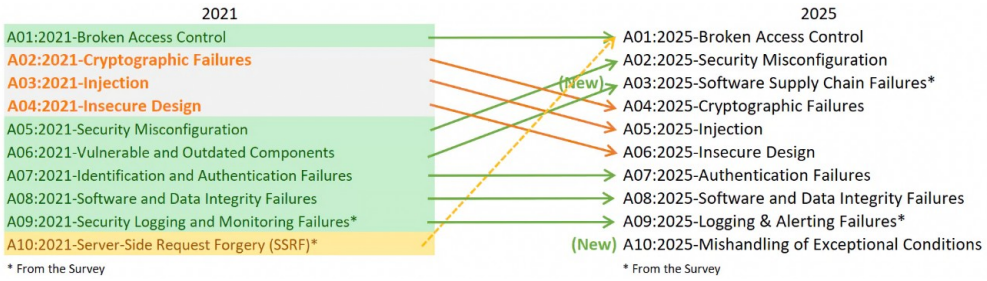
OWASP Top 10

- ‘The OWASP Top 10 is a standard awareness document for developers and web application security. It represents a **broad consensus about the most critical security risks to web applications**.’ [1]
- 2025: 8th instalment
- Mapped to Common Weakness Enumerations (CWEs)

[1] “OWASP Top Ten Web Application Security Risks”, *OWASP* <https://owasp.org/www-project-top-ten/>

Security 101

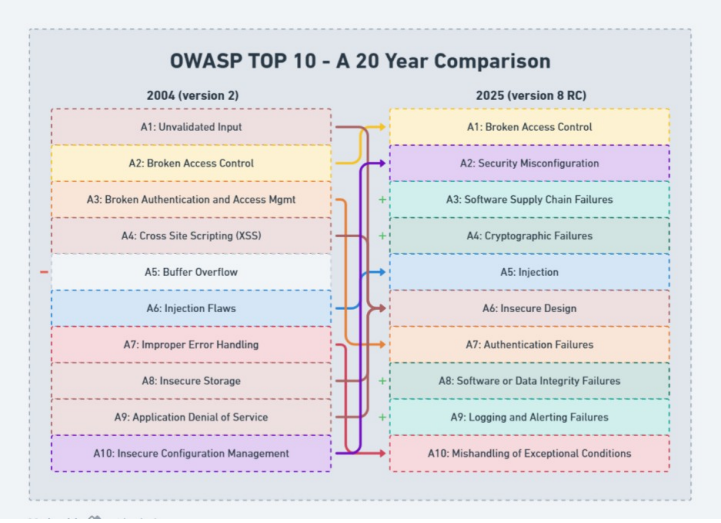
OWASP Top 10:2025



weezcrew

Some reordering from the previous version, Supply Chain Security is new(ish), but there was already something on that topic in 2021. Mishandling exceptions is new.

Plus ça change...



[1] Burger, J. "OWASP Top Ten: 20 Years of Application Security", *Octopus Deploy* <https://octopus.com/blog/20-years-of-appsec>

Since 2004 only one issue has been removed — Buffer Overflows — largely due to a shift to more memory-safe languages.

Otherwise, we've spent over 20 years with the same issues persisting. But note that these are broad *classes* of issue, e.g. there have been thousands of individual examples of injections across thousands of different software tools

Similarly, Supply Chain Security wasn't there in 2004 because this was prior to our modern dependency-heavy approach to development (e.g. well before npm)

An Example

A03: Software Supply Chain Failures

What it is

Breakdowns or compromises in building, distributing, or updating software through vulnerabilities or malicious changes in dependencies, tools, or build processes.

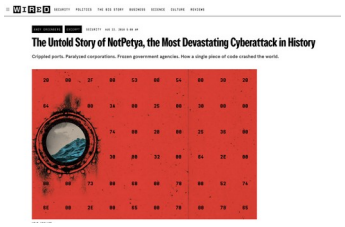
Impact on your system:

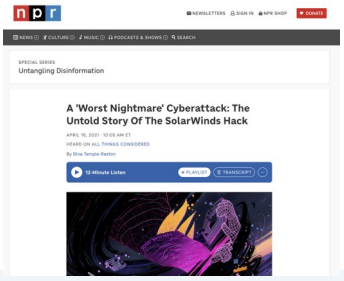
- Compromised packages introducing backdoors
- Malicious code injected during build processes
- Vulnerable dependencies cascading through your application
- Use of components from untrusted sources in production
- Changes within your supply chain are not tracked

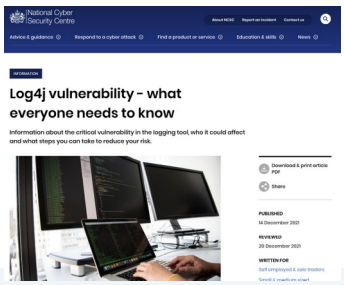
Notable CWEs

- CWE-1395: Dependency on Vulnerable Third-Party Component
- CWE-1104: Use of Unmaintained Third Party Components

Recent Incidents







Supply chain security is a big focus after a slew of impactful supply chain attacks

- NotPetya: fake ransomware (there was no way to decrypt the files, even if the ransom was paid) that originally targetted Ukrainian accounting software and ended up taking out the shipping giant Maersk

Last year – Shai-Hulud worm, first successful npm worm

weezcrew

7. Vulnerability Management

Software Bill of Materials (SBoM)
a.k.a. *nomenclature logicielle*

- ‘An SBOM is a **nested inventory** or list of ingredients that make up software components. In addition to the components themselves, SBOMs include critical information about the **libraries, tools, and processes** used to develop, build, and deploy a **software artifact**.’ [1]
- ‘A software bill of materials (SBOM) lists every component in your code so you can see what it’s built from and keep it more secure.’ [2]

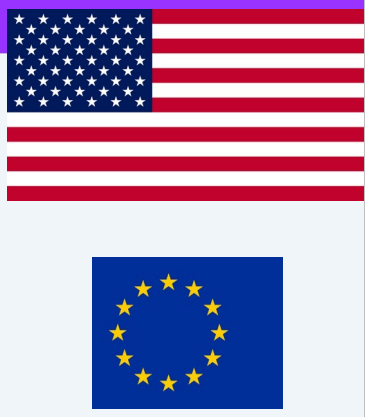
[1] Gitten, S. “The Ultimate Guide to SBOMs”, *GitLab Blog* <https://about.gitlab.com/blog/the-ultimate-guide-to-sboms/>
[2] “What is a software bill of materials (SBOM)?”, GitHub <https://github.com/resources/articles/what-is-an-sbom-software-bill-of-materials>

Security 101

Why You Should Care

- ➔ SBoMs have existed for a while, but always remained niche
- ➔ Big push by the US government in 2022 [1]
- ➔ **Legal requirement for EU digital service providers and software vendors as of Dec 2027**

> 'Manufacturers of products with digital elements shall:
> (1) **identify and document vulnerabilities and components** contained in products with digital elements, including by **drawing up a software bill of materials** in a commonly used and machine-readable format covering at the very least the **top-level dependencies** of the products;' [2]



weezcrew

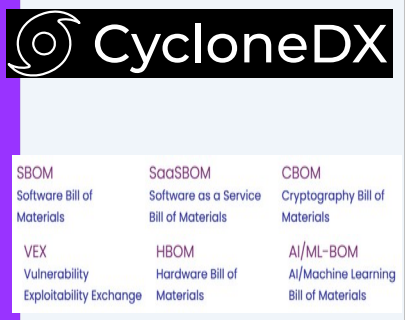
[1] *Secure Software Development Framework (SSDF)* (NIST SP 800-218), *NIST* <https://csrc.nist.gov/pubs/sp/800/218/final>
[2] Annex I, Part II, s 1 of the EU Cyber Resilience Act <https://eur-lex.europa.eu/eli/reg/2024/2847/oj>

'At the very leave the top-level dependencies' – this is the first legal step, but I would expect this to expand in the future

Security 101

OWASP CycloneDX

- ‘OWASP CycloneDX is a **full-stack Bill of Materials (BOM) standard** that provides advanced supply chain capabilities for cyber risk reduction.’ [1]
- Tools exist to generate CycloneDX SBoMs from:
  - `requirements.txt` (Python projects)
  - `package.json` (JS projects)
  - OS package managers (dev machines)
  - etc.



weezcrew

[1] Homepage, *CycloneDX* <https://cyclonedx.org/>

Can also record things like licensing information. Actually, the main alternative to CycloneDX (SPDX) began as a software licensing standard that expanded its scope

Expect to hear lots about ‘BOMs’ in the near future – AI/ML-BOM, HBOM, CBOM, etc. Slightly a marketing gimmick, but also a much-needed corrective to years of very irresponsible dependency explosion in software dev (see https://xkcd.com/2347/)

Security 101

OWASP Dependency-Track

- ‘Dependency-Track is an intelligent Component Analysis platform that allows **organizations** to identify and reduce risk in the software supply chain. Dependency-Track takes a unique … approach by **leveraging the capabilities of Software Bill of Materials (SBOM)**. This approach provides capabilities that traditional Software Composition Analysis (SCA) solutions cannot achieve.’ [1]
- Not to be confused with OWASP Dependency-*Check*

[1] “OWASP Dependency-Track”, *OWASP* <https://owasp.org/www-project-dependency-track/>
[2] *Dependency-Track* <https://dependencytrack.org/>

Dependency-Track ingests BOMs from across an organisation’s portfolio and provides a single dashboard for viewing them

OWASP Dependency-Track

Security 101

OWASP Validated Exploitable Data Flow (VXDF)

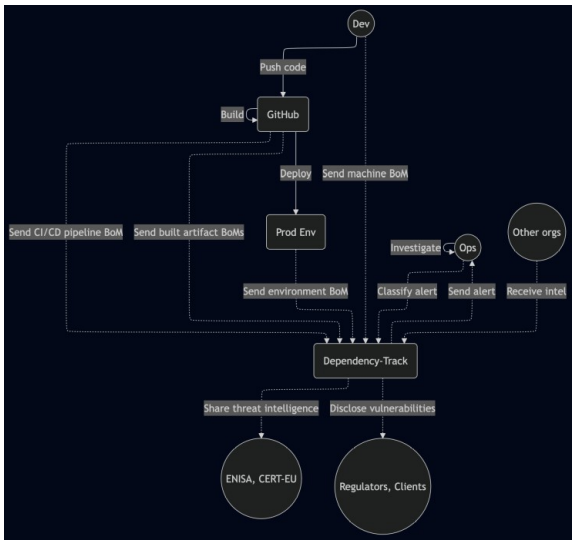
- ‘A **standardized JSON format** for documenting confirmed security vulnerabilities with the evidence needed to prove they're **real and actionable**.’ [1]
- Exploitation can be automated
- Reduce alert fatigue
- Cutting-edge [2]

weezcrew

[1] Homepage, *VXDF* <https://vxdf.org/>
[2] Shah, M. “OWASP VXDF: Is that really vulnerable? Show me the VXDF!” from OWASP Global AppSec US 2025 <https://www.youtube.com/watch?v=HY7N8pAoIDM>

VXDF – get vulnerability reports, verify that they are actually exploitable in your app, add that metadata (including things like ‘this line needs to change’). This can all be automated

Putting it All Together



We can define a per-build SBOM in our CI pipeline and ingest that in Dependency-Track

But we can go a step further and also send BOMs of our dev environments, the prod server, the CI pipeline (e.g. GitHub Actions) itself

Then, at the international/EU level, the idea is to eventually have thousands of companies all doing this sort of process internally, feeding their data to bodies like ENISA (and even sharing intelligence automatically amongst one another, using standard intelligence exchange formats like STIX or TAXII)

weezcrew

8. Further Reading

Security 101

OWASP Global AppSec US 2025 Highlights

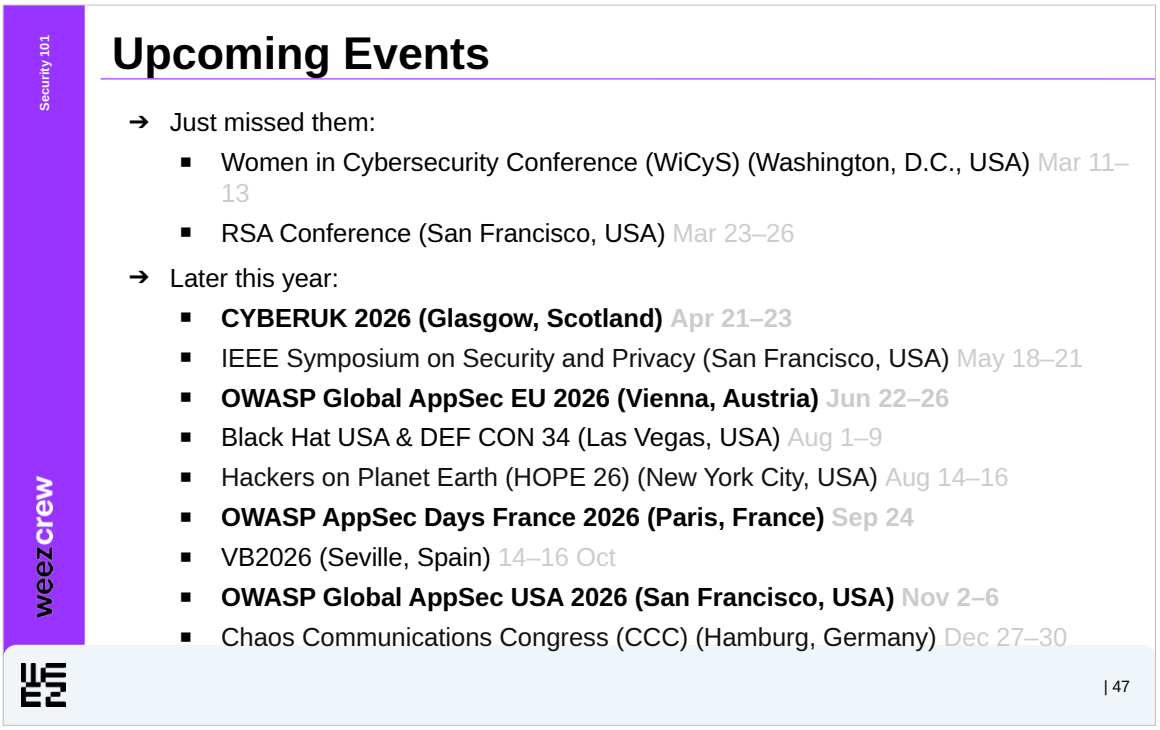
- ➔ “How Latest Browser Security Features Eliminate Bug Classes”
- ➔ “In GitHub We Trust: 10 Ways You Could Get Pwned”
- ➔ “Exploitable in the Wild CVE Appears! But Should We Fix ‘em All?”
- ➔ “The Appsec Program I Regret Building”
- ➔ **Secure Coding**
  - ▪ “Privacy by Design: What Are Engineers Supposed to Do with That?”
  - ▪ “Beyond the Checklist: Building an AppSec Culture That Engineers Don’t Dread”
- ➔ **Threat Modeling**
  - ▪ “Developer-Centric Threat Modeling: Embedding Security in Agile Workflows”
  - ▪ “How to Embed Privacy Threat Modeling in Agile Sprint Rituals Without Slowing T eams Down”
  - ▪ “Threat Modeling Developer Behavior: The Psychology of Bad Code”
- ➔ **SBoMs**
  - ▪ “SBOMs in the Real World: Practical Guidance for Managing Three Common SB OM Scenarios” (including ways to incorporate VXDF)

weezcrew

Here are some of the most interesting talks from last year’s OWASP conference

“How Latest Browser Security Features Eliminate Bug Classes” – as I mentioned before on the OWASP Top 10 slide, there are entire classes of issue that can be completely resolved with some clever technical solutions (but not always)

“In GitHub We Trust” – GitHub Actions is a security nightmare and there are many ways it can screw you over. There was a major supply chain breach via GH Actions just last week: https://unit42.paloaltonetworks.com/github-actions-supply-chain-attack/



Here are lots of big events coming up/just past. I don't expect people to fly to San Fran (nor would I recommend anyone visit the US right now), but put these dates in your calendar because you will see a lot of news articles and recorded talks on YouTube appear shortly afterwards

Re: WiCyS,

https://www.ukcybersecuritycouncil.org.uk/base/media/docs/the-equality-roadmap-final.pdf

Just 22% of the workforce across cyber is female, compared to 28% in other UK digital sectors and 48% of the total UK workforce. Just 13% of those occupying senior cyber roles are female.

In Europe:
2017 11%
2025 22%
Target: 50% by 2050

As previously mentioned, security is theoretically a field that anyone can enter, so not clear why these demographics are so imbalanced

Security 101

weezcrew

Further Reading

- ➔ Cyber Body of Knowledge (CyBOK)
- ➔ Secure coding:
  - **OWASP Development Guide, Secure Coding Quick-Reference & Cheat Sheets**
- ➔ Practise hands-on hacking:
  - OWASP Juice Shop, plus HackThisSite, HackTheBox, TryHackMe, VulnHub, etc.
- ➔ AI dangers:
  - Stenberg, D. “Death by a Thousand Slops”, *daniel.haxx.se* <https://daniel.haxx.se/blog/2025/07/14/death-by-a-thousand-slops/>
  - **Doctorow, C. “AI software assistants make the hardest kinds of bugs to spot”, *Pluralistic* <https://doctorow.medium.com/https-pluralistic-net-2025-08-04-bad-vibe-coding-maximally-codelike-bugs-8372979b3933>**

CyBOK – attempt to document the entire field of cyber security, academic experts compile summaries on each topic

For devs, practical checklists and quick-ref guides are available from OWASP

The best way to understand the methodology of an attacker is to pretend to be one, there are lots of services available that provide intentionally vulnerable apps that you can practice with

AI: curl maintainer has had to close his bug bounty programme because of AI slop reports, don’t be part of that problem

Cory Doctorow article: because of how AI tools work, they are tuned to produce buggy code that is statistically indistinguishable from functional code, so be *very* careful when reviewing AI code review and don’t use AI tools for sensitive functionality like authentication

Also, remember how little the OWASP Top 10 have changed over 20 years. AI coding tools are trained on the software written in the past 20 years, and so they are primed to regurgitate a lot of the same perennial bugs

Security 101

Further Reading

- ***Darknet Diaries* podcast**
- ***Infosecurity Magazine* and *404 Media***
- *Phrack, Paged Out!* and *UNREDACTED*
- Blogs:
  - NCSC
  - Trace Labs
  - SANS Internet Storm Center and Virus Bulletin
  - *Schneier on Security*, *Krebs on Security* and Troy Hunt
  - Cisco Talos, Crowdstrike, Google, Microsoft, etc.
  - the grugq

weezcrew

NCSC – UK National Cyber Security Centre, they do lots of great advice for both technical and non-technical audiences

Trace Labs – Focus on Open Source Intelligence (OSINT) techniques, mainly through events where you can practice those techniques to help track down missing persons

Troy Hunt: founder of Have I Been Pwned

Most tech companies have security-specific blogs that are worth a read

Put them all into your RSS reader of choice. If you don't use RSS, start: https://pluralistic.net/2026/03/07/reader-mode/

Ben GOLDSWORTHY
Software Developer
–
ben.goldsworthy@weezevent.com