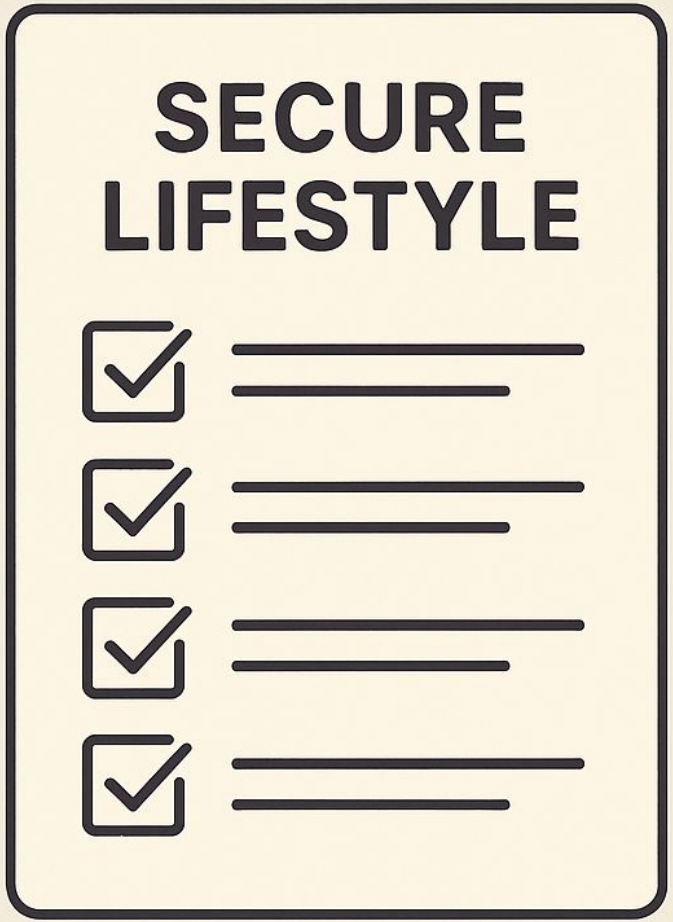

OWASP 2025
GLOBAL
AppSec
USA
Aram
Hovsepyan
The Most Common
AppSec Failures
In Fortune 500 Companies
and Beyond


JANUARY
1


SECURE
LIFESTYLE

How to sabotage your AppSec program

- Label every application as “mission critical”
- Turn threat modeling into bureaucracy
- Ignore security requirements
- Let your tools run the show
- Focus on beautiful dashboards, green is all you need

Aram Hovsepyan

- PhD in AppSec
- CEO @ Codific
- OWASP SAMM Core team member

https://www.linkedin.com/in/aramhovsep
https://appsecscience.com

Experimental methodology

- 40+ assessments
  - +-5 x 90 min interviews based on OWASP SAMM
  - Paired with another expert
- Follow up loop with most teams to validate improvements
- Qualitative insights are framework agnostic

Application risk: realities

- We have rarely seen a team with a clear and shared understanding of their risk profile
  - “All our applications are equally important”
  - “We have a tool that take care of risk”
  - “There is a CMDB record”

Application risk: why

- **Without risk you don’t need AppSec!**
- Risk defines the breadth and the depth of your program
- Risk connects strategy with execution

“Healthier” means different things to different people.

Application risk: how

Question	Scale
How sensitive is the data?	0 = Public, 10 = Top secret
What is the reputational impact to customers if breached?	0 = None, 10 = Severe
What is the reputational impact to our organization?	0 = None, 10 = Severe
What is the security knowledge of the dev team?	0 = Expert, 10 = None
…	…

Application risk: how

- Low risk: 0-15, Medium risk: 15-30, High risk: 31-40
- That’s not FAIR (or NIST 800-30)

Product	Team expertise	Data sensitivity	Impact to customer	Impact to us	Risk
Tuesday	5	10	10	10	**High**
MIRA	5	5	5	6	**Medium**

Threat modeling: realities

- The hero threat modeler
- Outsourcing to the security team

You won't get fit by watching your coach train.

Tool-driven / AI-driven threat modeling

Values

We have come to value:

- A culture of finding and fixing design issues over checkbox compliance.
- People and collaboration over processes, methodologies, and tools.
- A journey of understanding over a security or privacy snapshot.
- Doing threat modeling over talking about it.
- Continuous refinement over a single delivery.

threatmodelingmanifesto.org

Threat modeling: why

- More elaborate and technical risk profiles
- Builds lasting awareness across the team

Threat modeling: how

- Timebox it (2 hours)
- During sprint planning or whenever
- Provide context
  - Your application risk profile
  - DFD, architectural diagrams
- Hunt for threats
- Log every finding as a defect

Security requirements: realities

- Misunderstood as other activities
  - “That’s covered by training for XSS, SQLi, etc”
  - “That’s something that pen testers catch”
  - “We’re good, our tooling handles those”
- “Accidental” security requirements
  - “We had to implement access control for that feature”
  - “We have integrated the application with SSO”

You won’t get healthy by “happy accidents”

Security requirements: why

- Functional requirements define expected behavior
  - Defects are deviations from the specs
- Security requirements define CIA expectations
  - Vulnerabilities are deviations from the security specs
- **No security requirements - no vulnerabilities!**

CVE-2020-28435 Detail

This is NOT a defect
The lack of sanitization is by design

MODIFIED

This CVE record has been updated after NVD enrichment efforts were completed. Enrichment data supplied by the NVD may require amendment due to these changes.

Description

This affects all versions of package ffmpeg-sdk. The injection point is located in line 9 in index.js.

Metrics

CVSS Version 4.0 | CVSS Version 3.x | CVSS Version 2.0

NVD enrichment efforts reference publicly available information to associate vector strings. CVSS information contributed by other sources is also displayed.

CVSS 3.x Severity and Vector Strings:

NVD	**NIST:** NVD	**Base Score:** 9.8 CRITICAL	**Vector:** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
C	**CNA:** Snyk	**Base Score:** 9.4 CRITICAL	**Vector:** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:L

Security requirements: how

- Treat them like regular requirements
  - Add them to your sprint planning
  - Devs will implement them
  - Devs will write unit / integration test cases
  - QAs will verify them
- OWASP ASVS & MASVS

appsecscience.com/security_requirements

“Toolmares”

- Teams build their processes around tools
- False sense of (in)security
  - 34% of CVEs were never confirmed or disputed *
  - over 40% of CVEs got a different CVSS score after 9 months **
  - notcve.org

* *Confusing Value with Enumeration: Studying the Use of CVEs in Academia*, Schloegel et al.
** *Shedding Light on CVSS Scoring Inconsistencies: A User-Centric Study on Evaluating Widespread Security Vulnerabilities,* Wunder et al.

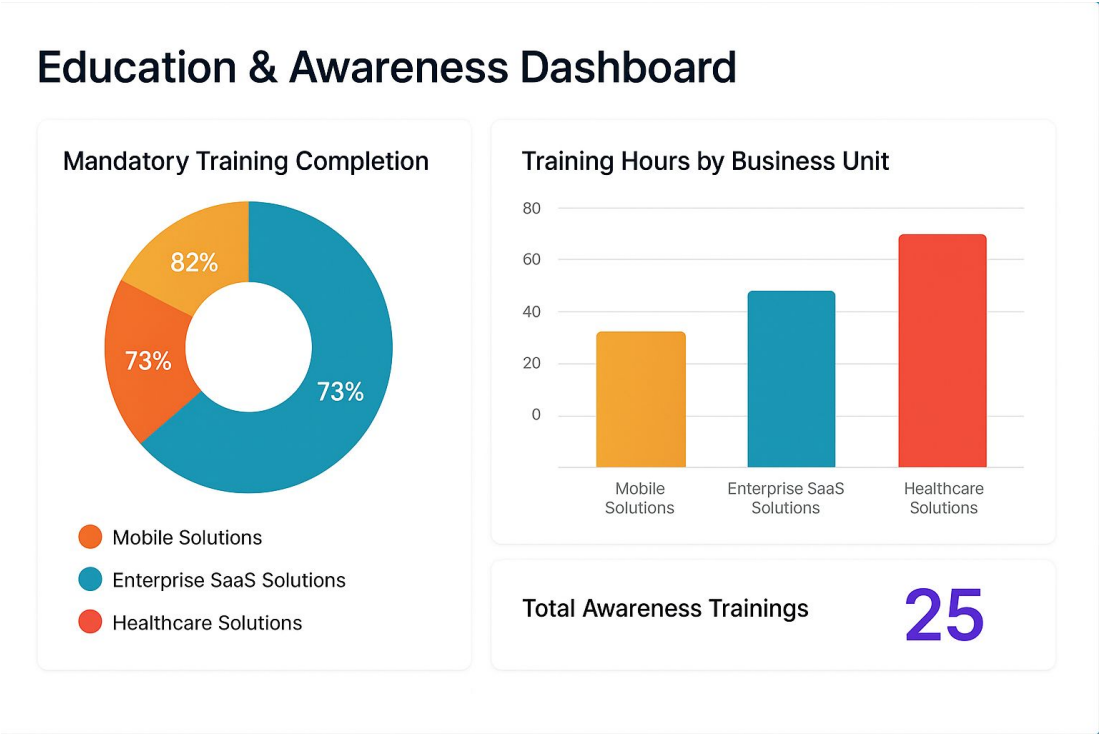
appsecscience.com/cve_video

Tooling best practices

- Start with people and process, then bring in tools
  - Handling dependency management: https://www.youtube.com/watch?v=IJQf5dHx7Mo
- Own your tool configurations

Security metrics: realities

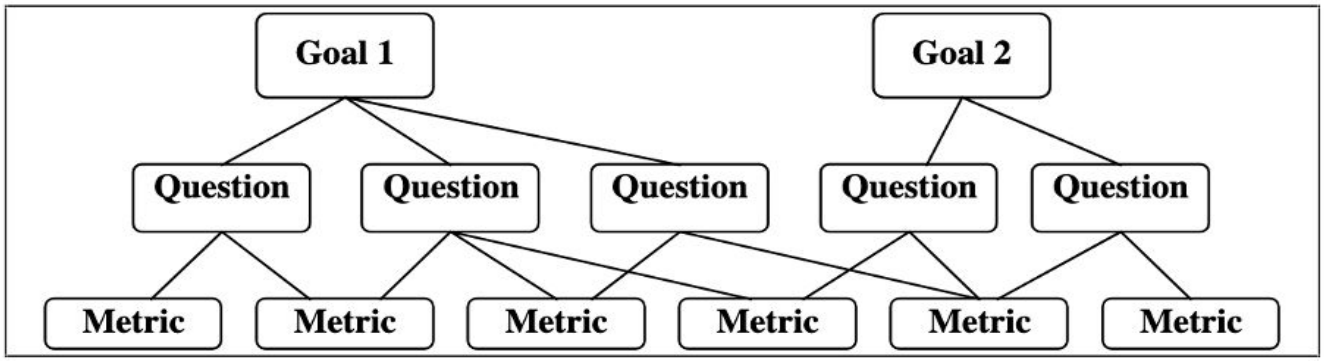
Security metrics: realities



Security metrics: why

- Measure where you are and track progress
- Validate security efforts are working
- Justify investments

Security metrics: how



appsecscience.com/metrics_video

Security metrics: how

- G: Improve the time to fix for high-risk vulnerabilities in production
- Q1: How many high-risk vulnerabilities are in production?
- Q2: What is the current mean time to fix?
- Q3: Is the time to fix improving?
- M1: Number of exploitable vulnerabilities in production
- M2: Impact score of each vulnerability
- M3: Likelihood score of each vulnerability
- M4: Time to deploy of each vulnerability fix

Key takeaways (For real this time)

- Create a shared understanding of application risk
- Make threat modeling collaborative and pragmatic
- Define security requirements
- Use tools to support your process
- Start with goals, then look for metrics

Thank you

References

- appsecscience.com
- owaspsamm.org
- *Confusing Value with Enumeration: Studying the Use of CVEs in Academia (Schloegel et al.)*
- *Shedding Light on CVSS Scoring Inconsistencies: A User-Centric Study on Evaluating Widespread Security Vulnerabilities (Wunder et al.)*