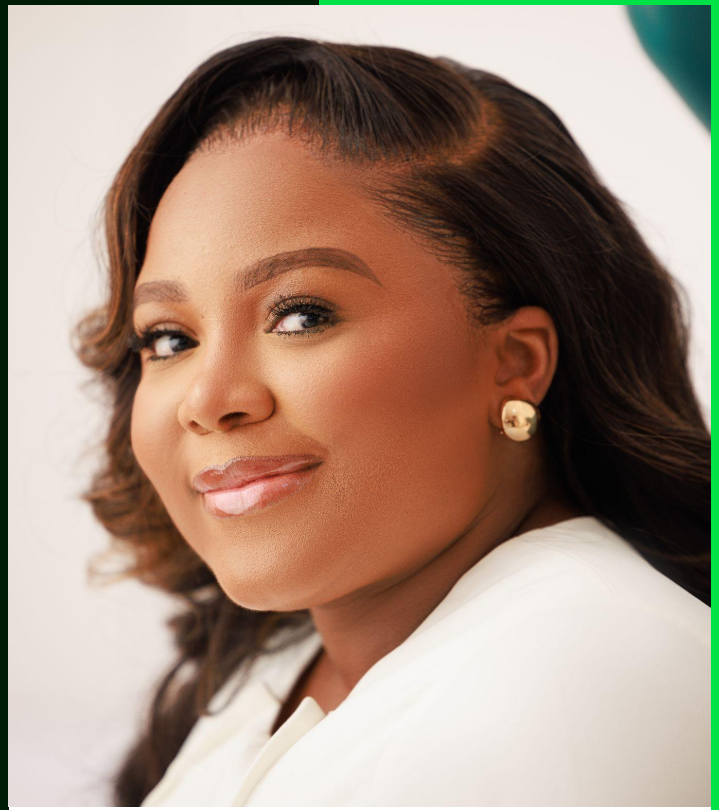
Security Principles for the Proactive Dev

Do these 10 things and you would have taken care of the most prevalent security challenges.



CONFIDENCE STAVELEY

CONFIDENCE STAVELEY

She is a multi-award winning cybersecurity leader, best selling author of API Security For White Hat Hackers, talent developer, gender inclusion actionist and top 40 global thought leader in life safety & security.

Her super power? The ability to communicate complex security concepts to audiences of all types.

Beyond her advisory roles on boards, she is the founder of CyberSafe Foundation and MerkleFence.

Why is security so hard?

1. You dislike working with security professionals.
2. Security professionals are unreasonable… sometimes.
3. You are not proactively consuming security updates.
4. You are not incentivized to have a security-first mindset.
5. Problems are found too late.

10

things you can do to prevent the most impactful security challenges…

Do Threat Modelling

1

What if we could brainstorm and catch security problems early before they are used to attack our software?

* Use STRIDE

Do Input Validation

2

Process inputs (from user, API, file stream, database, etc.) ONLY after encoding and validation on the Server Side.

Authenticate and Authorize.

3

Follow ONLY best practices like centralized implementation of authentication, separating authentication logic, etc..

50%

of successful cloud attacks start with compromised credentials and phishing.

Source: IBM Cost of Data Breach 2023

Protect your cloud credentials

4

The bad guys want your identity!

Attackers are going after dev infrastructure more than vulnerability in your code.

12%

Of all data breaches originated from software supply chain attacks.

Source: IBM Cost of Data Breach 2023

Do Software Composition Analysis…automate if possible!

5

Are there known problems with that framework/library/third-party code you didn’t write, but your code needs to run?

Don’t build these things…

6

Cryptographic algorithms, Identity and session management, authorization & authentication.

It’s called a secret for a reason…

7

Properly store authorization tokens, access keys, passwords, SSH keys …preferably in a vault. Definitely NOT in your code.

Where possible, pick type and memory safe languages.

8

Some programming languages are more secure than others.

Handle errors gracefully & Log!

9

Tackle system misconfiguration

...

10

oops! Don’t forget encryption and using all applicable security headers too.

Test for
security issues!

Bonus

Some Helpful Resources

1. Confidence Staveley LinkedIn - https://www.linkedin.com/in/confidencestaveley/
2. API Kitchen - https://www.youtube.com/@SisiNerdTV/
3. API Sec University - https://www.apisecuniversity.com
4. SemGrep - https://academy.semgrep.dev/order?ct=dd265eb0-626a-41cf-bd1d-fb25e80fe1e8
5. OWASP Secure Coding Practices Checklist - https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/stable-en/02-checklist/05-checklist
6. API Security For White Hat Hackers https://packt.link/ConfidenceS
7. OWASP Zap - Free DAST Tool ZAPzaproxy.orghttps://www.zaproxy.org
8. OWASP Cheat Sheet - https://cheatsheetseries.owasp.org/index.html
9. Google Phishing Quiz https://phishingquiz.withgoogle.com

Thank You.