A Qualitative Investigation of Insecure Code Propagation from Online Forums

Wei Bai · Omer Akgul · Michelle Mazurek

Most vulnerabilities aren’t new

- Many are “solved” problems
- But they end up in code anyway! Why?

Why do vulnerabilities happen?

- Developers don’t know enough

Why do vulnerabilities happen?

- Developers don’t know enough
- Complex security API’s

Why do vulnerabilities happen?

- Developers don't know enough
- Complex security API's

Why do vulnerabilities happen?

- Developers don’t know enough
- Complex security API’s
- Incorporating security later on

Why do vulnerabilities happen?

- Developers don’t know enough
- Complex security API’s
- Incorporating security later on
- Documentation/Material referred to[1]

[1] Acar et al. -- “You get where you’re looking for The Impact of Information sources on code security”, IEEE S&P ’16

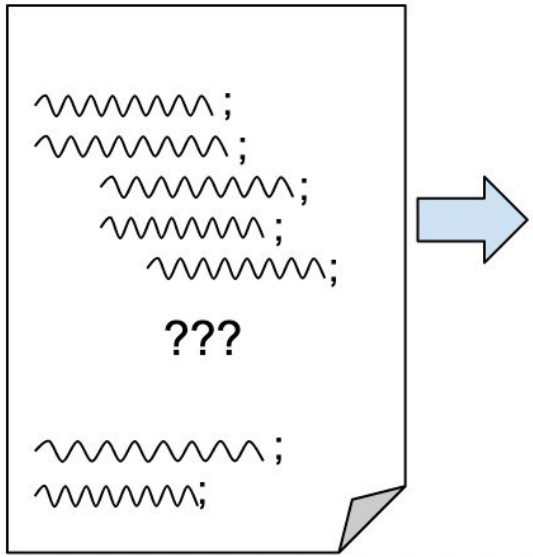
Why do vulns happen?

- Developers don't know enough
- Complex security API's
- Incorporating security later on
- Documentation/Material referred to[1]

[1] Acar et al. -- "You get where you're looking for The Impact of Information sources on code security", IEEE S&P '16

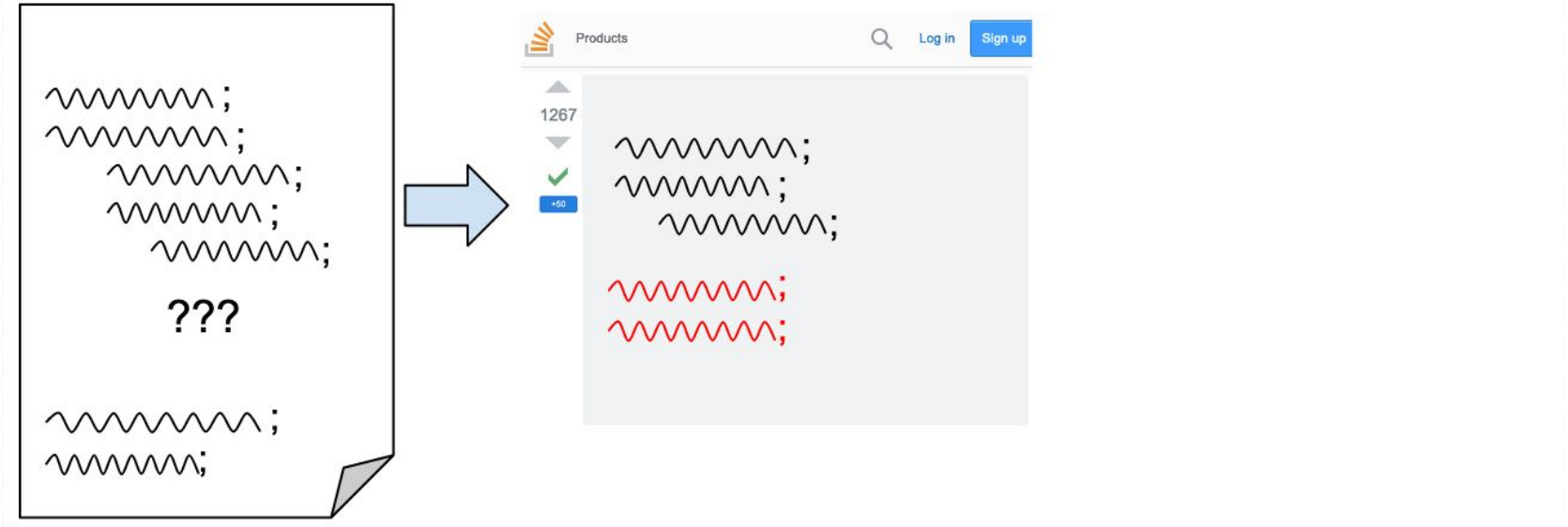
Focus: information source

What happens:



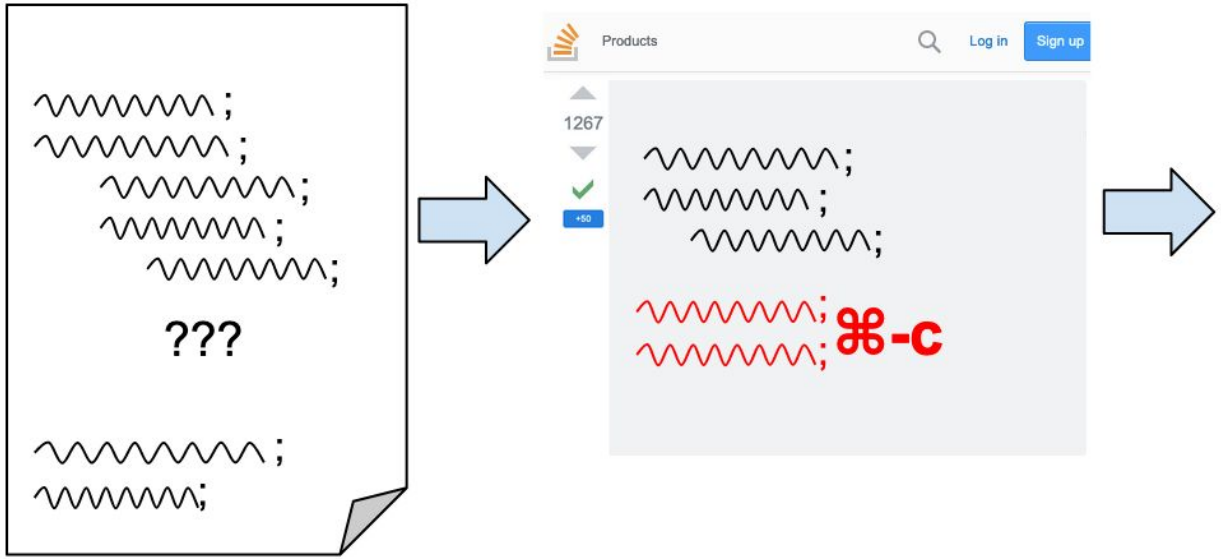
Focus: information source

What happens:



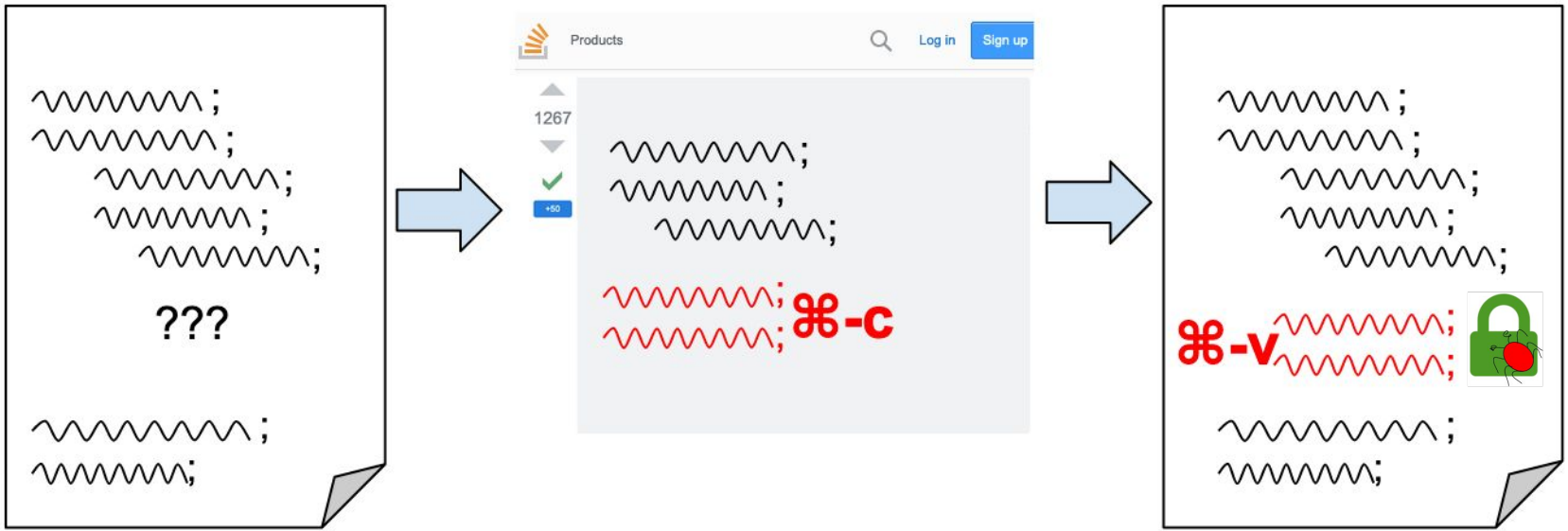
Focus: information source

What happens:



Focus: information source

What happens:



Focus: information source

What happens:

Research Questions

- Do developers realize SO can be bad?
- Do they have concerns when importing security code?
  - What are their mitigation tactics?

Research Questions

- Do developers realize SO can be bad?
- What are their mitigation tactics?

understanding developers ⇨ better mitigation design

Method Overview

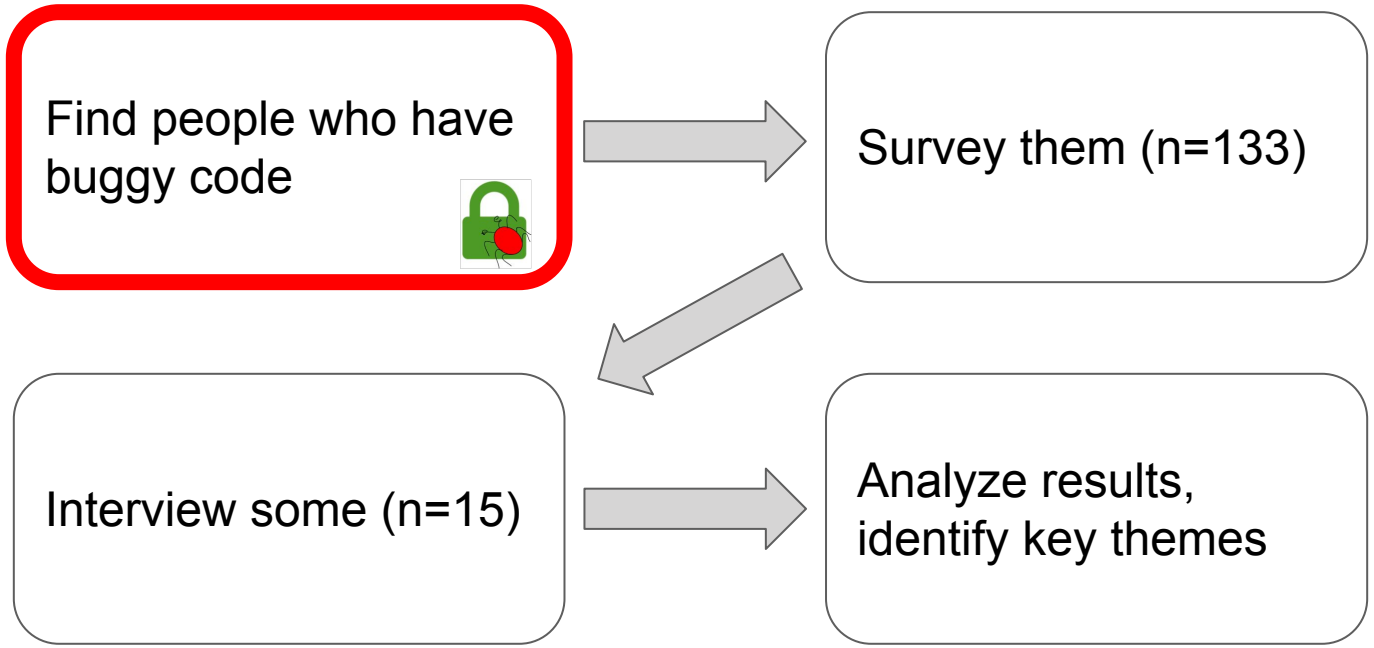
Find people who have buggy code

Survey them (n=133)

Interview some (n=15)

Analyze results, identify key themes

Method Overview

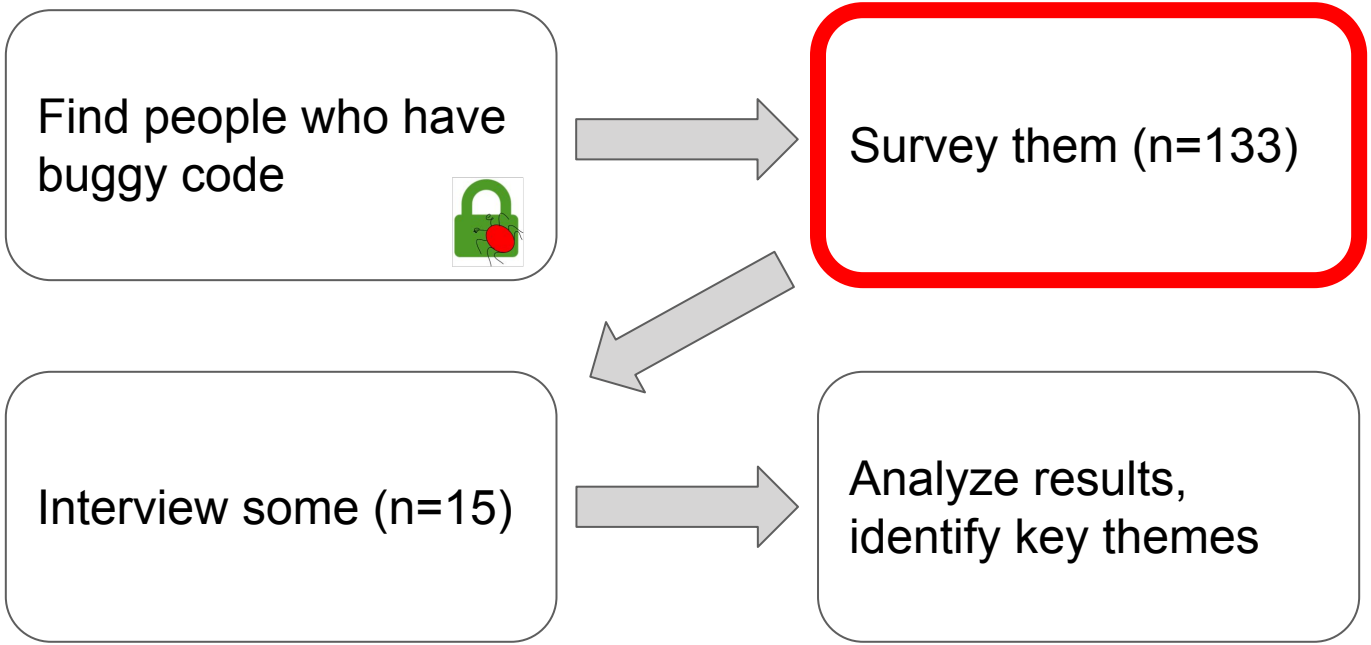


Finding buggy code

- Common crypto bugs found from prior work[1,2]
- Manually find these in SO code snippets
- Use MOSS to match with GitHub repos
- Manually inspect to be sure.

[1] Egele et al. -- “An empirical study of cryptographic misuse in android applications”, CCS ’13
[2] Lazar et al. -- “Why does cryptographic software fail?: A case study and open problems”, APSys ’14

Method Overview



The Survey

- Background: experience, education, work environment, etc.
- Usage of online programming forums in general, for security
  - How frequently?
- How do you vet code from forums?

The Survey

- Back
- Usag
  - H
- How

nt, etc.

security

"In your own words, please explain how you evaluate the quality of code from online code snippets. How do you decide whether to accept or reject the code?"

Method Overview

Find people who have buggy code 

Survey them (n=133)

Interview some (n=15)

Analyze results, identify key themes

The Interview

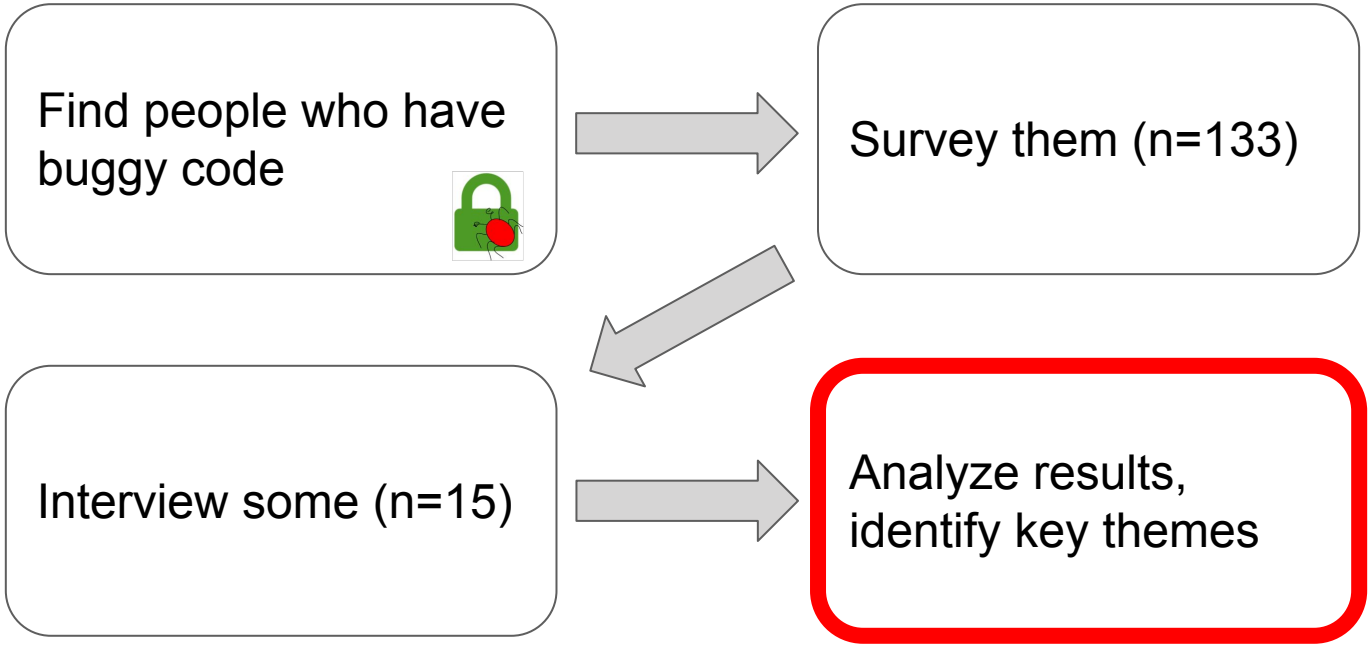
- About the project: team, deadline, etc.
- Pointing out the bugs
- Why/how did the bug happen?
- How would you fix it?
- How would you avoid this in the future? What would help?

The Interview

- About th
- Pointing
- Why/ho
- How wo
- How wou ould help?

"What would help you easily integrate security-related code into your tasks correctly and efficiently?"

Method Overview



Qualitative coding

- Rigorous social-science approach to analyzing free-text data
- Assign category labels to each statement; generate themes
- To ensure validity, two researchers work independently

Qualitative coding

- Rig… …analyzing fr… …t data
- As… …emes
- To… …validity, t…

K = 0.9, 0.82, 0.81 ⇨ “almost perfect” reliability

Bugs We Examined

- Six vulnerabilities drawn from [1,2]
- Generally involving crypto, often authentication/credentials

[1] Egele et al. -- “An empirical study of cryptographic misuse in android applications”, CCS ’13
[2] Lazar et al. -- “Why does cryptographic software fail?: A case study and open problems”, APSys ’14

Vulns	# Surveys taken	# Interviews
Bad RNG	7	1
ECB mode	11	0
IV problems	29	4
Constant keys	7	4
Constant salts	6	5
Few iterations	73	9
Totals	133	15

stackoverflow Products Customers Use cases Search…

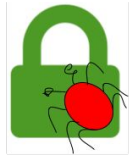
Home

PUBLIC

Stack Overflow

```
public class [redacted] {

    private static final byte[] SALT = {
        (byte) 0xA9, (byte) 0x9B, (byte) 0xC8, (byte) 0x32,
        (byte) 0x56, (byte) 0x35, (byte) 0xE3, (byte) 0x03
    };
```



Occupation	Software dev	54.1%
	Faculty member	1.5%
	Graduate students	2.3%
Years of dev. exp.	0-4	29.3%
	5-9	29.3%
	10-14	21.1%
	15-24	20.3%
Security background	Slightly know.	21.8%
	Somewhat know.	52.6%
	Very know.	21.8%

What did we observe?

- Drawing from online sources, in general
- Why did bugs happen?
- Security behaviors and justifications

Drawing from online sources

- Devs do refer to online sources (duh)
- Precautions when importing code
- Some claim they do not copy code
- Sometimes functionality is all that matters

Why did it happen?

- 3 blamed SO
- 1 blamed a book
- 4 couldn’t do security evaluation
- 8 weren’t prioritizing security
- 2 wanted the code to be more efficient.

Security behaviors and justifications

- Participants skeptical of online security code
- Some devs trust their security skills
- Majority admits they need to learn more
- Prioritize functionality over security
- Some believe it’s not their responsibility

Participants claim to be skeptical

- Many mitigation techniques indicate this
- Of people who refer to security code:
  - Survey: most described validation mechanisms
  - Interview: 8/12 did not do validation for the project

Insufficient security knowledge

- Most say security knowledge is important
- Some say bug was due to lack of knowledge
- Most would need to learn more to integrate security code properly.

Insufficient security knowledge

> *“ Well-written articles to explain the problems, explain the pitfalls, explain mistakes people commonly make. And I would love to see an article written like that. . . ”*

Insufficient security knowledge

- Validation by learning:
  - From forums, blogs, articles
  - Industry organizations, official documentation

Some trust their security skills

- Some validation mechanisms imply confidence in skills:
  - Inspect code carefully (19/43)
  - Write tests, try to break (7/43)
- When asked, most offered fixes to the bug
  - Two said they would rewrite crypto libraries!

Is this a contradiction?

- Need more knowledge vs. trust my skills
- In survey, mostly two separate groups
- But in the interview, most said both
  - They reflect on their processes and realize they need to learn more?

Security isn’t the top priority

- In line with prior work[1]
- Functionality, efficiency are higher priorities
  - Common in both interview and survey

[1] Balebako and Cranor -- “Improving app privacy: Nudging app developers to protect user privacy”, IEEE S&P ’14

Security isn’t the top priority

> *“ This was an acceptable solution. I did not search again and again to find the best solution or to find the weakness in my code. I grabbed it from some forum ... Just take, use, and go on. ”*

Security isn’t the top priority

> *“ The hard-coded thing probably is because it took less time for me to encrypt and decrypt. ”*

Security is someone else's job

- In line with prior work [1,2]
- Need code reviews to avoid similar bugs
- Use methods "trusted by the community"
  - Lots of upvotes, comments
- Completely outsource security.

[1] Mouratidis et al. -- "When security meets software engineering: a case of modelling secure information systems", Information Systems '05
[2] Redmiles et al. -- "How i learned to be secure: a census-representative survey of security advice sources and behavior", CCS '13

Security is someone else's job

> *“ Someone else in a service to do it for me, like some other company, to offload problems to someone else. I would probably use some service like Firebase from Google, they have all the authentication service. ”*

Security not important in my context

- 7/15 interviewees
  - Project is not used by many people
  - Project is used in internal offline tasks only
  - Crypto primitives for “non-security” applications

Implications for design

- Security-oriented feedback system
  - Essentially warning people about security issues.
  - For “security not my job” people
- Linking to educational material.
  - For people who want to learn

0

This question already has an answer here:

Given Final Block not properly padded while AES decryption *2 answers*

Iam encrypt in JavaScript and decrypt in Java but getting below error:

Error thrown in java: java.lang.Exception: Given final block not properly padded

Below is my Java script code:

```
var key =CryptoJS.enc.Utf8.parse("0123456789012345");
var ive  = CryptoJS.enc.Utf8.parse("0123456789012345");

var encrypted = CryptoJS.AES.encrypt(password, key, {iv: ive});
console.log('encrypted msg = ' + encrypted.toString());
```

0

This snippet has been marked **insecure**
For more information on please refer to: [Why constant encryption keys are problematic.]

Iam encrypt in JavaScript and decrypt in Java but getting below error:

Error thrown in java: java.lang.Exception: Given final block not properly padded

Below is my Java script code:

```
var key =CryptoJS.enc.Utf8.parse("0123456789012345");
var ive  = CryptoJS.enc.Utf8.parse("0123456789012345");

var encrypted = CryptoJS.AES.encrypt(password, key, {iv: ive});
console.log('encrypted msg = ' + encrypted.toString());
```

Summary

- Survey and interview study on insecure code propagation
- Several critical reasons:
  - Devs (over)trust their security skills
  - Insufficient security knowledge
  - Security is low priority
  - Security is not my job

akgul@umd.edu

go.umd.edu/sec-pro-panel

SP² SECURITY. PRIVACY. PEOPLE

NIST

UNIVERSITY OF MARYLAND 18 56