Analysis of Authentication Code

BSc Thesis
2nd Presentation

09 November 2021

Adrian Jörg
Where is Authentication Code?

account creation

account log-in

account recovery
Authentication Code

... is a wall of defense against various attacks

... should be secure

... is transparent to the user and often forgotten in tutorials

... should always satisfy the ever-evolving safety requirements
Authentication Code In Practice
Bonobos clothing store suffers a data breach, hacker leaks 70GB database
The passwords stored in the database are hashed using SHA-256 or SHA-512 according to threat actors who have started to analyze the database. One threat actor claims to have already cracked the passwords for 158,000 SHA-256 passwords but has been unable to crack the SHA-512 passwords.
How Secure Is Authentication Code In Open-source Projects?
Where Can We Find Relevant Code?

GitHub!

26,810 repository results

- EpicGames/Signup
  - Information about signing up for a free Epic Games account, and getting access to UnrealEngine source code.
  - 1.9k stars
  - Updated on Jan 18

- sibtc/simple-signup
  - Code samples used in the blog post "How to Create User Sign Up View"
  - python, django, tutorial, tutorial-code
  - 128 stars
  - Python
  - MIT license
  - Updated on Oct 24, 2020

- abuanwar072/Welcome-Login-Signup-Page-Flutter
  - Mobile app onboarding, Login, Signup page with #flutter.
  - 473 stars
  - Dart
  - Updated on Nov 18, 2020

- aws-samples/py-flask-signup-docker
  - Sample Python application to show the capabilities of EC2 Container Service.
  - 70 stars
  - HTML, Apache 2.0 license
  - Updated 15 days ago
Initial Exploration
Collected Properties

Password policies
- Length
- Pattern

Policy inconsistencies
- Code duplication
- Impaired functionality

Names of files & functions
- Sign up
- Login
- Password reset

Used libraries
- Authentication
- Validation
- Encryption

Metadata
- Forks & stars
- Date of first & last commit

Further observations
- Insecure hashing algorithms
- Outdated encryption libraries
- Performance issues
- Risk of (No-)SQL injections
Initial Findings

Dataset:

- 54 GitHub repositories
- 3 different languages
- 5 problem categories with occurrences > 1
How can we detect such problems on GitHub?

CodeQL Queries!
Free semantic code analysis engine offered by GitHub: 
[codeql.github.com]

Supports many programming languages
C/C++, C#, Go, Java, JavaScript, Typescript, Python

Supports many frameworks
ASP.NET, Express, ...
import javascript
from Function f
where f.getName().matches("(?i)\w\d")
select f.getName(), f.getParameter(0)
Custom CodeQL Queries

Our CodeQL queries can detect ...

... insecure hashing algorithms

... outdated encryption libraries

... performance issues

... exposure to (No-)SQL injections

... duplicated authentication code
Custom CodeQL Query Example

```java
/**
 * @id synchronous-hashing
 * @kind problem
 * @precision very-high
 * @problem.severity warning
 */

import javascript

from MethodCallExpr e
where e.getMethodName().matches("hashSync")
select e,
    "This method uses synchronous hashing, which will block the event loop and prevent your application from servicing any other inbound requests or events"
```
Pipeline

- Fetching URLs from GitHub
- Cloning repositories
- Creating CodeQL databases
- Analyzing databases with CodeQL queries
- Interpreting the results
Results

Dataset:

422 GitHub repositories
(only JavaScript)

Fetched from GitHub API
“language:js NodeJS signup
in:filepushed:>2015-01-01”

3 problem categories
Takeaways

Authentication code uses ...

... outdated libraries

... different policies for different use cases

… code from popular repositories and inherits their problems
Future Work

Building **additional queries** to detect more problems

Leveraging **more data**

**Translating queries** to other programming languages
Summary

Authentication Code

- account creation
- account log-in
- account recovery

Initial Findings

Dataset:
- 31 GitHub repositories
- 3 different languages
- 5 problem categories with occurrences > 1

Custom CodeQL Query Example

```java
/**
 * @id synchronous-hashing
 * @kind problem
 * @precision very-high
 * @problem.severity warning
 */
import javascript
from MethodCallExpr e
where e.getMethodName().matches("hashSync")
select e,
"This method uses synchronous hashing, which will block the event loop and prevent your application from servicing any other inbound requests or events"
```

Results

Dataset:
- 422 GitHub repositories
  (only JavaScript)
- Fetched from GitHub API
  "language:js NodeJS signup init:pushed>2015-01-01"
- 3 problem categories