Stack Overflow Considered Harmful?  
The Impact of Copy&Paste on Android Application Security  

F. Fischer*, K. Böttinger*, H. Xiao*, C. Stransky†, Y. Acar†, M. Backes†, S. Fahl†  
*Fraunhofer AISEC †CISPA, Saarland University  

Presentation by Kevin Liao
Code copypasta insecure?
How prolific are security-related code snippets from Stack Overflow in Android applications?
This talk

Rather than discuss results at end...

Present results first, then analyze the methodology

Does the methodology convince us of the results?
The high-level approach
The high-level approach

Extract security-related snippets
The high-level approach

Security analysis
The high-level approach

Identify code reuse
Results: Alarming (potentially)
Extracted snippets

30 million posts

2 million Android-related posts

~4,000 security-related snippets
Security classification

- Secure: 70%
- Insecure: 30%
Prevalence of code reuse

2,673 secure snippets
1,161 insecure snippets

1.3 million free apps
Prevalence of code reuse

Security-related

Secure

Insecure

200,672 (15.4%)
Prevalence of code reuse

- Security-related: 200,672 (15.4%)
- Secure: 73,839 (5.56%)
- Insecure

Bar chart showing the number of instances of code reuse in different categories.
Prevalence of code reuse

- Security-related: 200,672 (15.4%)
- Secure: 73,839 (5.56%)
- Insecure: 196,403 (15%)
Apps with security-related snippets

- Secure: 2%
- Insecure: 98%
Top-offender? TLS...

- 180k apps w/ empty Trust Manager
- Deactivates server verification
- Can lead to MITM
Next top-offender? Symmetric crypto

- 18k apps with AES in ECB mode
- Hard-coded keys

AES/ECB

91%
Next top-offender? Symmetric crypto

- 18k apps with AES in ECB mode
- Hard-coded keys
Do insecure snippets have lower scores?
Do insecure snippets with a warning have lower scores?
Are high view count/score snippets copy&pasted more?
Are high view count/score snippets with a warning copy&Pasted less?
Discussion of methodology

Extract security-related snippets
Extract security related-snippets

1. Get all posts with ‘Android’ tag
2. Filter code-snippets that use security APIs
   • TLS/SSL
   • Symmetric/asymmetric crypto
   • RNG
   • Signatures
   • Message digests
   • Authentication/access control
Discuss snippet extraction
Discussion of methodology

Security analysis
Security analysis

1. Manually label snippets as secure or insecure
2. Train a binary classifier to automatically determine security/insecurity of all snippets
tl;dr for labeling rules

• SSL/TLS: Use TLS v1.1 or greater; don’t use old crypto

• Symmetric: Don’t use old crypto; don’t use ECB; don’t use static/zeroed/derived keys or IVs

• Asymmetric: Use >=2048 bit RSA; use >= 244 bit ECC

• Hashing: Don’t use MD-family

• RNG: Use crypto-secure RNG; securely random seed
Security score of training set
Train SVM binary classifier
Feature selection

• Based on tf-idf
• “The features rely merely on the vocabulary level of input code snippets, without even understanding how they are functioning.”
• Claim: Can be more accurate and more scalable than rule-based methods
TF-IDF

TF-IDF is a measure of originality of a word by comparing the number of times a word appears in a doc with the number of docs the word appears in.

\[
\text{TF-IDF} = \text{TF}(t, d) \times \text{IDF}(t)
\]

- **Term frequency**
  - Number of times term \( t \) appears in a doc, \( d \)

- **Inverse document frequency**
  - \[ \log \frac{1 + n}{1 + df(d, t) + 1} \]
    - \( n \): # of documents
    - \( df(d, t) \): Document frequency of the term \( t \)
Security classification

- Secure: 70%
- Insecure: 30%
Discuss security classification
Discussion of methodology

Identify code reuse
Identify code reuse

1. Transform source code and Dalvik executables into same IR
2. Identify similar code snippets using Program Dependency Graphs (PDGs)
IR transformation

Source code

<...>

PPA

Typed AST

Dalvik executable

EXE

Lift

Bytecode

JVM

WALA

T. J. WATSON LIBRARIES FOR ANALYSIS
Program Dependency Graphs

• Generate PDG for each method
• Nodes: Statements in methods
• Edges: Data and control dependence
Dependency edges

Data: S2 depends on S1, since A read in S2.

\[
\begin{align*}
S1: & \quad A = B \times C \\
S2: & \quad D = A \times E + 1
\end{align*}
\]

Control: S2 depends on A, since A determines S2’s execution.

\[
\begin{align*}
S1: & \quad \text{if (A) then} \\
S2: & \quad B = C \times D \\
& \quad \text{endif}
\end{align*}
\]
Examples of PDGs

Program `Main`

```c
program Main
    sum := 0;
    i := 1;
    while i < 11 do
        sum := sum + i;
        i := i + 1
    od
    od
end
```

Program `Main`

```c
program Main
    i := 1;
    while i < 11 do
        i := i + 1
    od
end
```
Prevalence of code reuse

- Security-related: 200,672 (15.4%)
- Secure: 73,839 (5.56%)
- Insecure: 196,403 (15%)
Discuss identification of code reuse
Final discussion

• About results?
• About methodology?
• About future work?