Securing Public-key Cryptography on the Android Platform

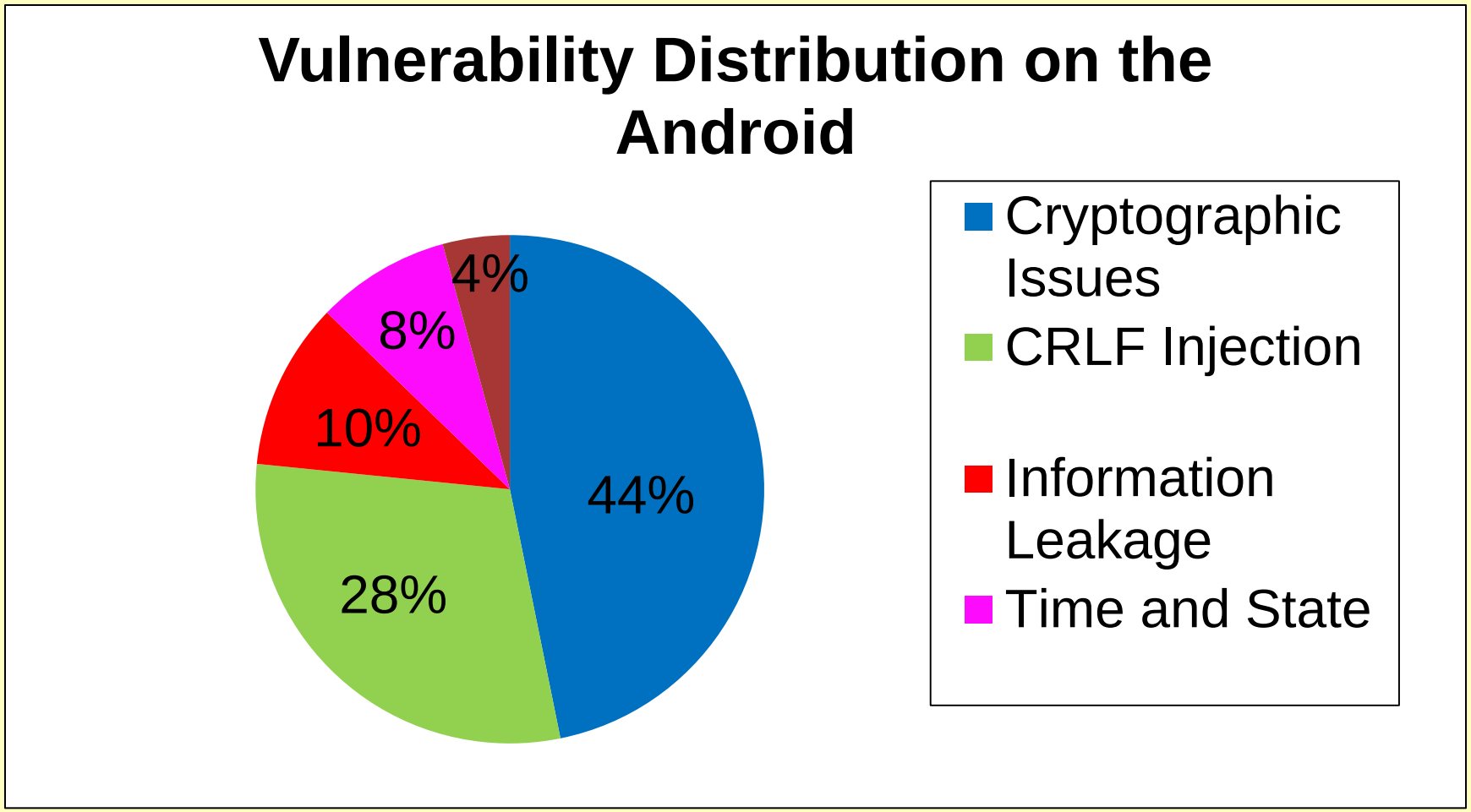
Hannah Gommerstadt

Harvard University

The Weak Link

Out of 10,000 analyzed Android applications, 40% were found to have a hard coded cryptographic key.

Motivation


Vulnerability Distribution on the Android
44%
28%
10%
8%
4%
Cryptographic Issues
CRLF Injection
Information Leakage
Time and State

Cryptographic Issues

- Major vulnerabilities:
  - Cryptographic keys hardcoded into the program text
  - Output of private keys onto insecure channels

Theoretical Basis

- Defined a semantic security condition for public-key cryptography
- Developed an enforcement mechanism
- Proved that the type system enforces the security condition

Implementation

- Developed Cryptflow, which is an information flow analysis of Java code that detects misuses of cryptography
- Cryptflow can currently analyze snippets of Java code and identify simple vulnerabilities
  - Output of private keys to System.out.println()
  - Cryptographic keys hardcoded into the program text

Future Work & Conclusion

- We are currently working on getting Cryptflow to analyze the source code of entire Android applications to detect misuses of public-key cryptography
- It is necessary to guarantee and enforce the security of public-key cryptographic protocols, especially on the Android platform.