SECAI

AI-supported Security Testing

Supervised by

Michael Schlichtig, Markus Schmidt

Secure Software Engineering

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

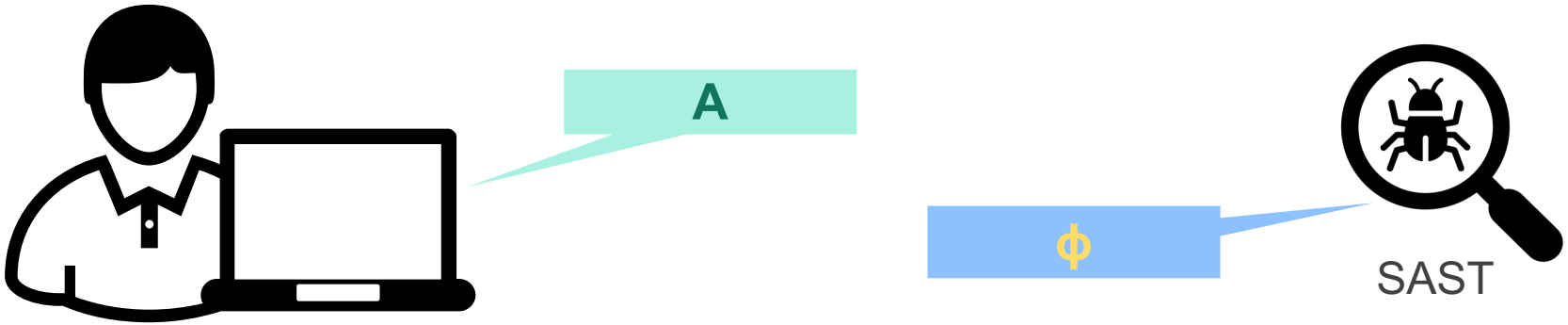
Fraunhofer
IEM

Motivation

Static Application Security Testing (SAST) Tools

63.7% applications ≥ 1 misuse cryptography
State of Software Security Volume 11, VeraCode Inc.

Code issues are widespread



Support developers in secure software development

Preparing slides on Friday …

… and then …

BBC

Home News Sport Business Innovation Culture Travel Earth Video Live

CrowdStrike IT outage affected 8.5 million Windows devices, Microsoft says

2 days ago

Share

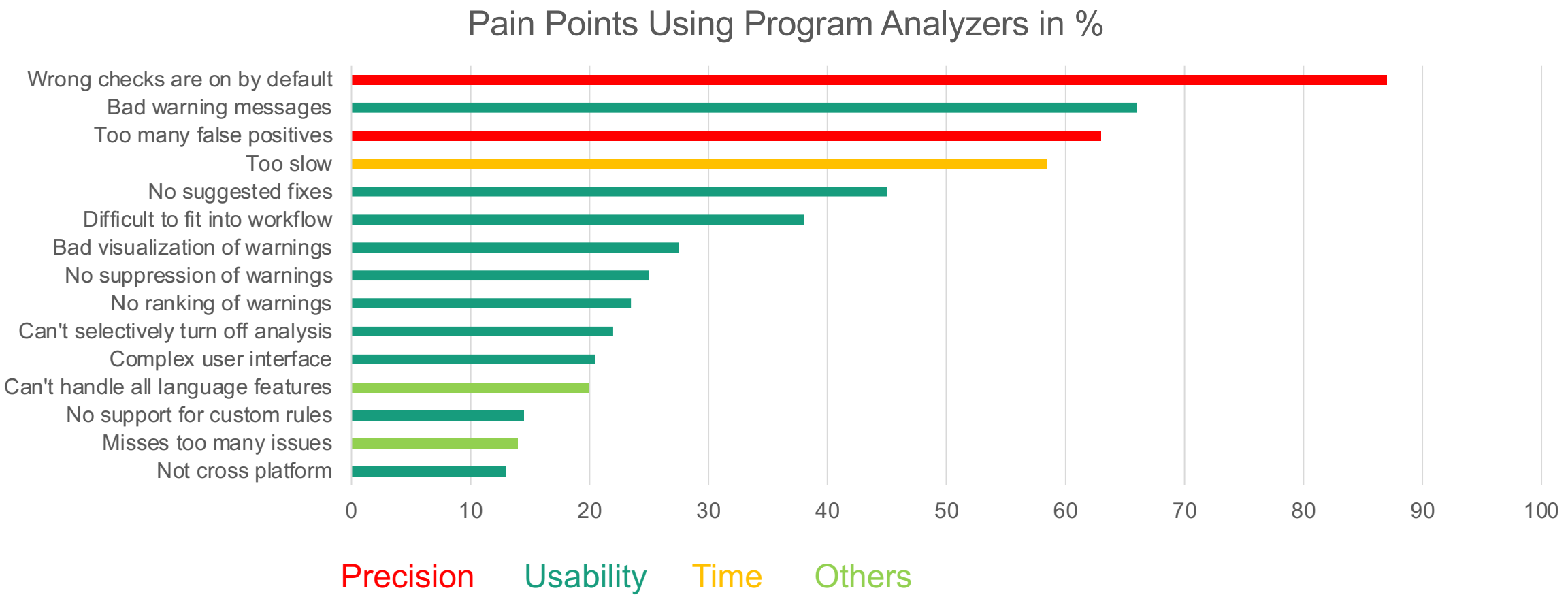
By Joe Tidy, Cyber correspondent, BBC News

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

Fraunhofer IEM

Problem

Usability issues



[1] Christakis, Maria, and Christian Bird. "What developers want and need from program analysis: an empirical study." ASE*'16*.

How warning reports often look like

GitHub Action example of CogniCrypt

Run CogniCrypt 1m 10s

```
38086
38087  Findings in Java Class: org.cambench.cap.fieldsensitivity.advanced.fielddepth3.truepositive.pbeparameters.PBEParameters1
38088
38089          in Method: void main(java.lang.String[])
38090                  RequiredPredicateError violating CrySL rule for javax.crypto.spec.PBEParameterSpec
38091                          First parameter was not properly generated as randomized
38092                          at statement: specialinvoke $r13.<javax.crypto.spec.PBEParameterSpec: void <init>(byte[],int)>($r16, i0)
38093                          at line: 35
38094
38095
38096  ======================== CryptoAnalysis Summary ==========================
38097          Number of CrySL rules: 51
38098          Number of Objects Analyzed: 2319
38099
38100          CryptoAnalysis found the following violations. For details see description above.
38101          ConstraintError: 327
38102          RequiredPredicateError: 607
38103          IncompleteOperationError: 457
38104
38105          Additional analysis statistics:
38106                  SoftwareID:
38107                  SeedObjectCount: 2319
38108                  CryptoAnalysisTime (in ms): 38870
38109                  CallgraphConstructionTime (in ms): 1501
38110                  CallgraphReachableMethods: 1885
38111                  CallgraphReachableMethodsWithActiveBodies: 1852
38112                  DataflowVisitedMethods: 781
38113  ============================================================================
38114  Error: Process completed with exit code 1.
```

https://github.com/CROSSINGTUD/CryptoAnalysis

https://github.com/CROSSINGTUD/CryptoAnalysis-demo/actions/runs/9414947239/job/25934880677

Your code for file encryption compiles?

But is it secure?


```
1        public class FileEncrypt {
2            private static Key generateKey(String password) throws Exception {
3                DESKeySpec dks = new DESKeySpec(password.getBytes("utf-8"));
4                SecretKeyFactory keyFactory = SecretKeyFactory.getInstance("DES");
5                return keyFactory.generateSecret(dks);
6            }
7
8            public static String encryptFile(String password, String srcFile, String destFile) {
9                IvParameterSpec iv = new IvParameterSpec("123456".getBytes("utf-8"));
10               Cipher cipher = Cipher.getInstance("DES/CBC/PKCS5Padding");
11               cipher.init(Cipher.ENCRYPT_MODE, generateKey(password), iv);
12               InputStream is = new FileInputStream(srcFile);
13               CipherInputStream cis = new CipherInputStream(is, cipher);
14               // write cis into destFile and return;
15           }
16       }
```

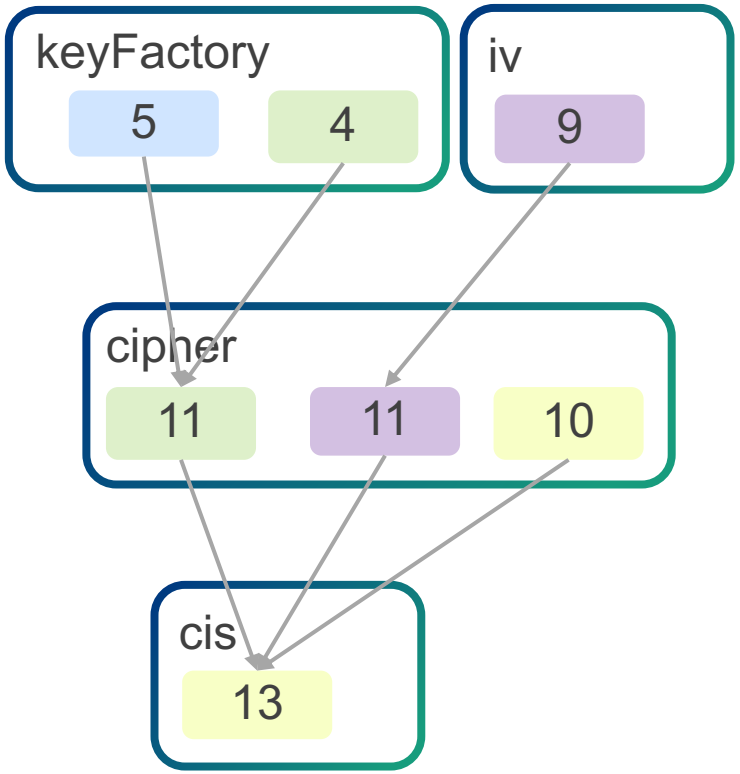
Simplified real-world example.

[2] Wickert, Anna-Katharina, et al. "Supporting Error Chains in Static Analysis for Precise Evaluation Results and Enhanced Usability.“ SANER 24

How can we help developers?

Developing secure code is difficult

```
1                public class FileEncrypt {
2                        private static Key generateKey(String password) throws Exception {
3                                DESKeySpec dks = new DESKeySpec(password.getBytes("utf-8"));
4   🐞                           SecretKeyFactory keyFactory = SecretKeyFactory.getInstance("DES");
5   🐞                           return keyFactory.generateSecret(dks);
6                        }
7
8                        public static String encryptFile(String password, String srcFile, String destFile) {
9   🐞                           IvParameterSpec iv = new IvParameterSpec("123456".getBytes("utf-8"));
10  🐞                           Cipher cipher = Cipher.getInstance("DES/CBC/PKCS5Padding");
11  🐞 🐞                        cipher.init(Cipher.ENCRYPT_MODE, generateKey(password), iv);
12                               InputStream is = new FileInputStream(srcFile);
13  🐞                           CipherInputStream cis = new CipherInputStream(is, cipher);
14                               // write cis into destFile and return;
15                       }
16               }
```



Goal: Provide developers with a useful interface for using SAST tools such as CogniCrypt.

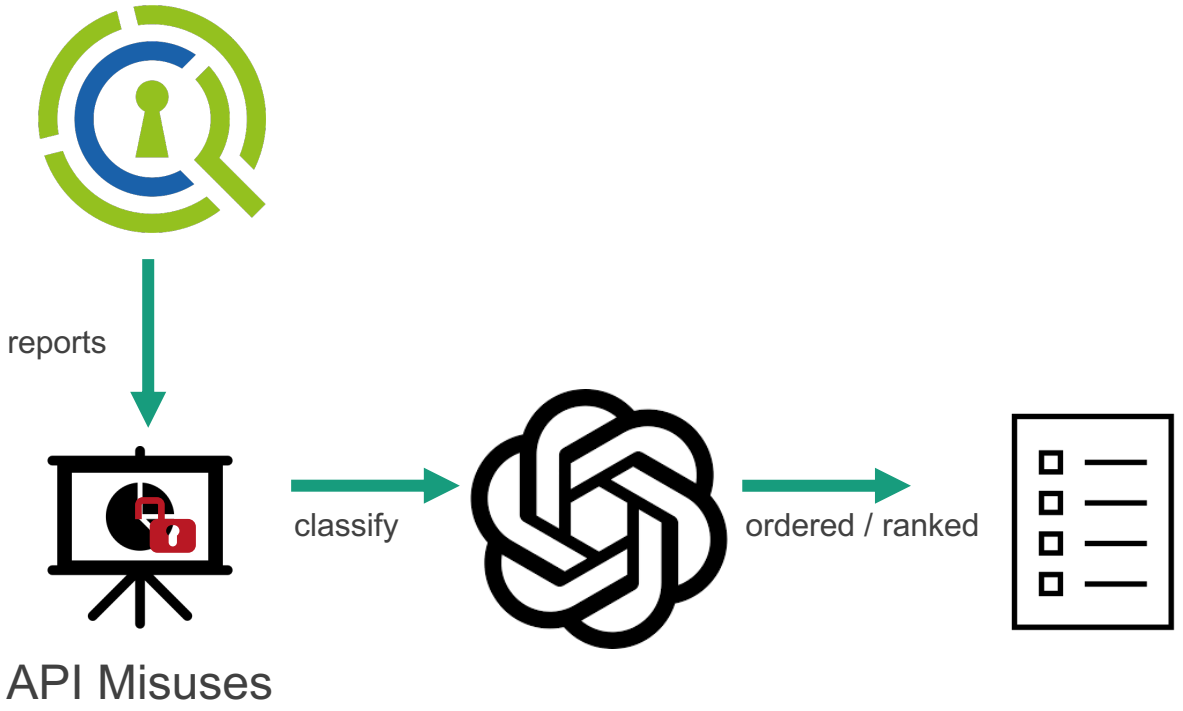
Simplified real-world example.

[1] Wickert, Anna-Katharina, et al. "Supporting Error Chains in Static Analysis for Precise Evaluation Results and Enhanced Usability.“ SANER 24

What else can we do?

Leveraging AI

- Ranking Warning Messages
  - Filter False Positives
  - Severity Score
  - Difficulty, …



[3] Tripp, Omer, et al. "Aletheia: Improving the usability of static security analysis." CCS 14.

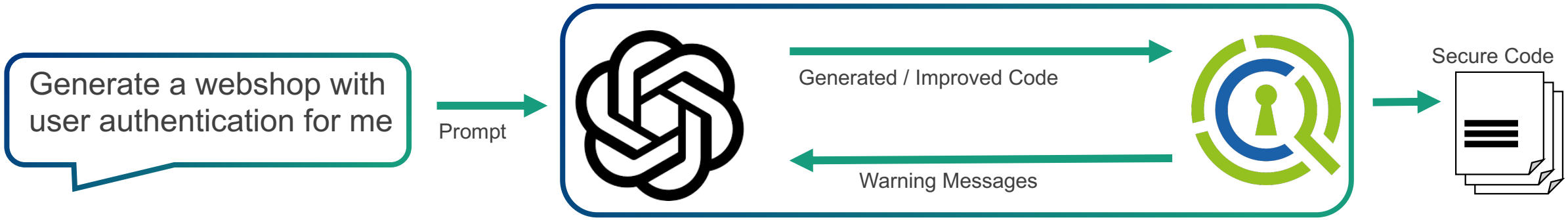
HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

Fraunhofer IEM

What else can we do?

Current student theses

- Secure code generation by combining LLMs and SAST



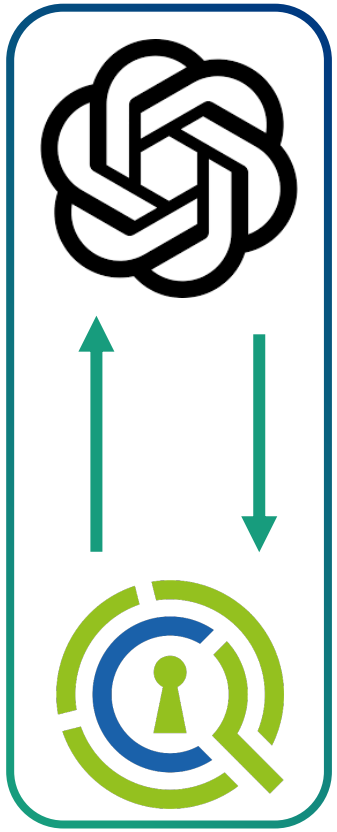
- Use LLMs to explain warning messages

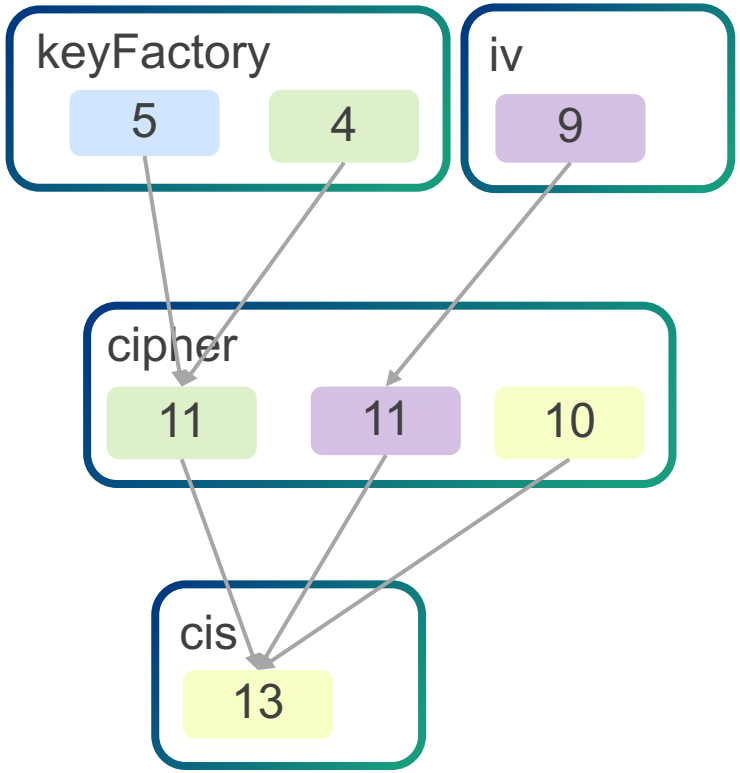
Summary

Goals of our Project Group

- Provide developers with a useful interface for using SAST tools such as CogniCrypt.

- Build the **SecAI** plugin with valuable features!





HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

Fraunhofer IEM

SecAI

Join our Project Group

Interested in any of these topics?

- Security & Cryptography
- Machine Learning & AI
- User Interface Design
- Software Engineering

SCAN ME

Michael Schlichtig

Research Associate
Secure Software Engineering
michael.schlichtig@uni-paderborn.de

Heinz Nixdorf Institut

Markus Schmidt

Employee
Secure Software Engineering
markus.schmidt@uni-paderborn.de

Heinz Nixdorf Institut

go.upb.de/pg-sse-SecAI

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

Fraunhofer IEM