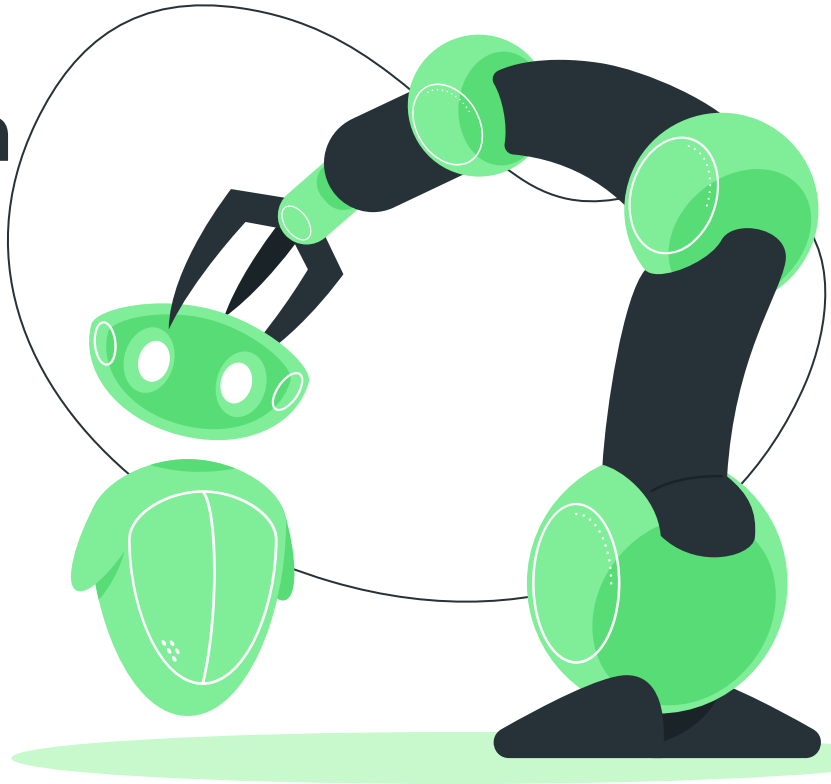
eiannone@unisa.it
https://emaiannone.github.io/

# Toward Automated Exploit Generation for Known Vulnerabilities in Open-Source Libraries

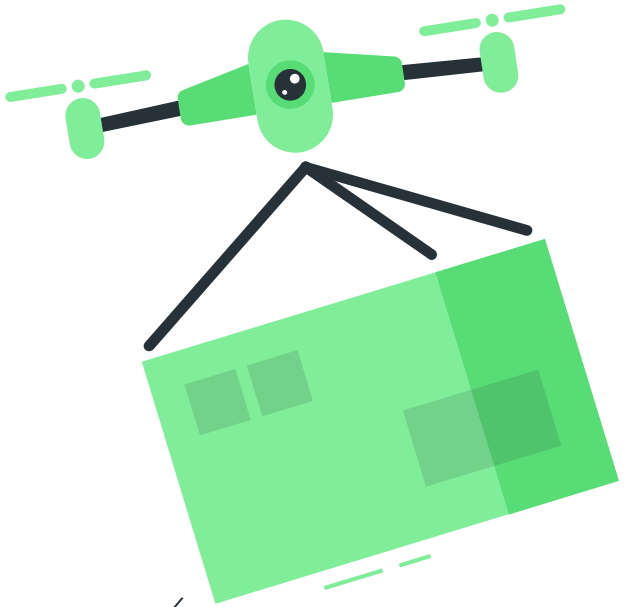
**Emanuele Iannone**[*], Dario Di Nucci[†], Antonino Sabetta[‡] and Andrea De Lucia[*]



[*]*SeSa Lab* - University of Salerno, Fisciano, Italy
[†]*Tilburg University*, JADS, 's-Hertogenbosch, The Netherlands
[‡]*SAP Security Research*, France

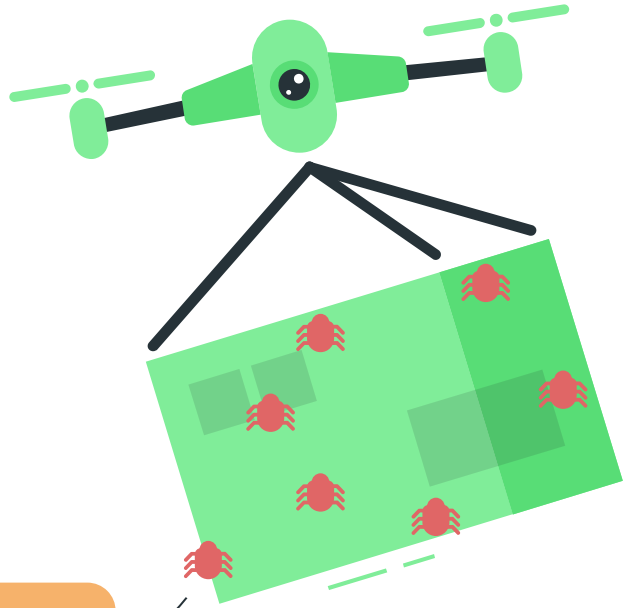
# Software Reuse

**+90%** Software products on the market use OSS components.



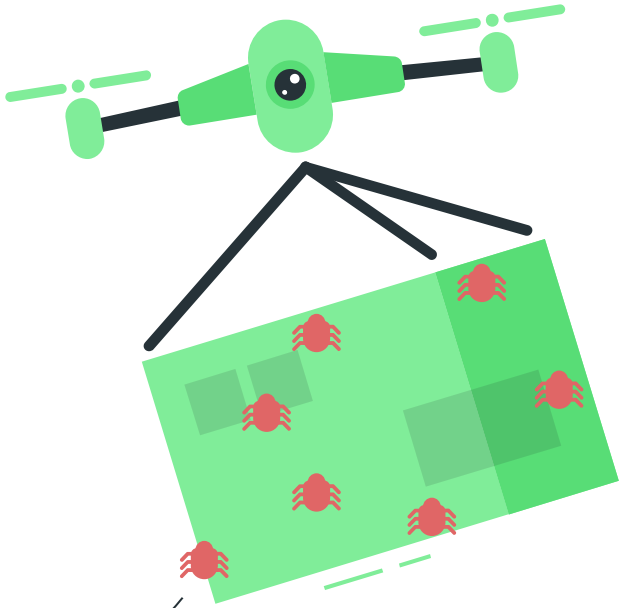
# Software Reuse

**+90%** Software products on the market use OSS components.



**Libraries and frameworks have defects, especially vulnerabilities!**

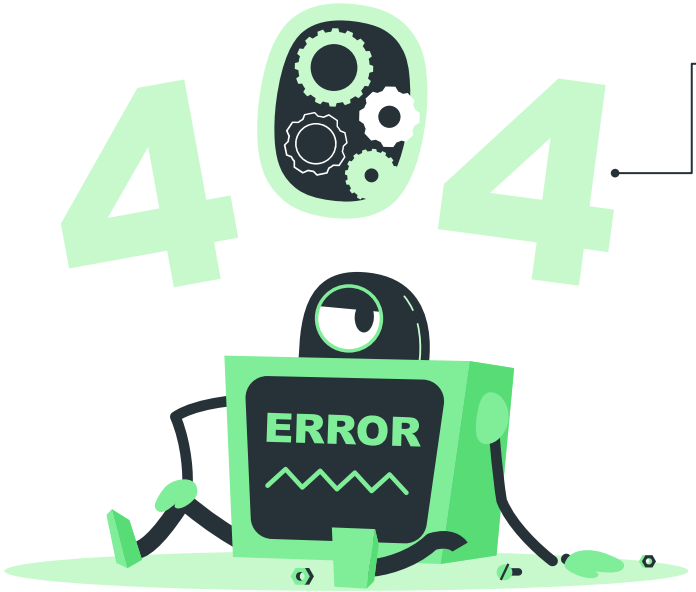
1. Injection
2. Broken Authentication
3. Sensitive Data Exposure
4. XML External Entities (XXE)
5. Broken Access Control
6. Security Misconfiguration
7. Cross-Site Scripting XSS
8. Insecure Deserialization
9. **Using Components with Known Vulnerabilities**
10. Insufficient Logging & Monitoring



# Top-10 Security Risks in Web Applications

OWASP®

# Detecting Vulnerable OSS Components




**Static Analysis**

Quick and easy,
but low precision

*Ponta, Serena & Plate, Henrik & Sabetta, Antonino. (2020).*

***Detection, assessment and mitigation of vulnerabilities in open source dependencies.***

# Detecting Vulnerable OSS Components

**Static Analysis**

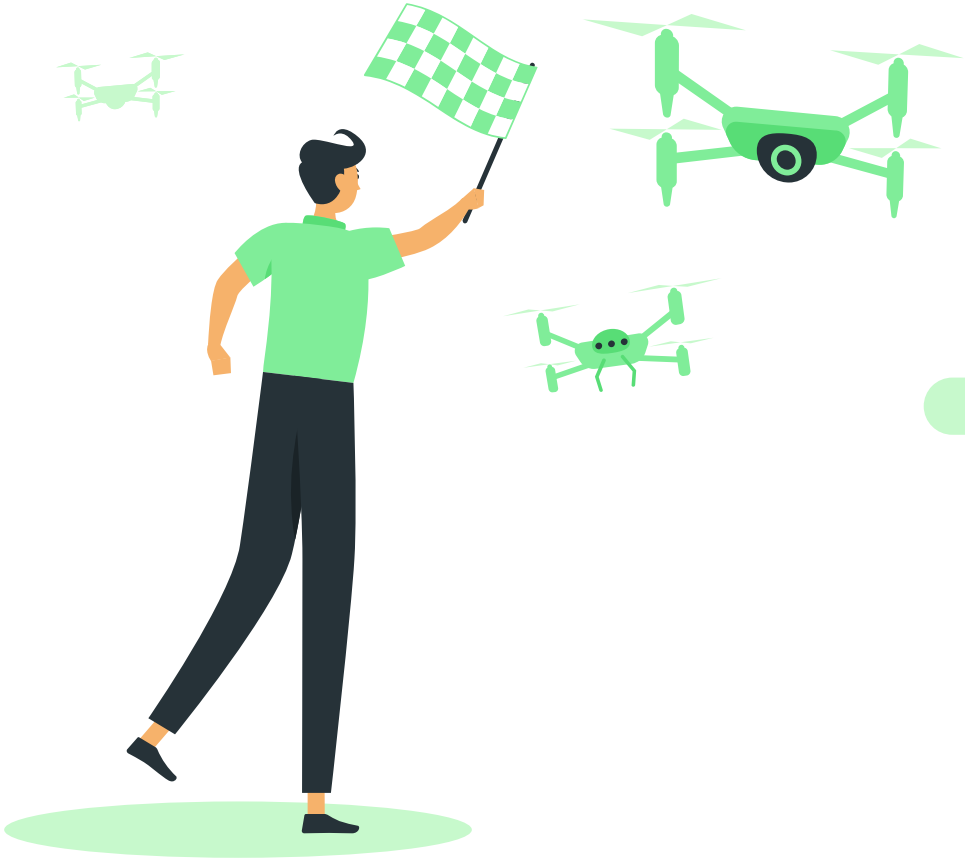
Quick and easy, but low precision


**Static + Dynamic**

More precise, but require good test suites

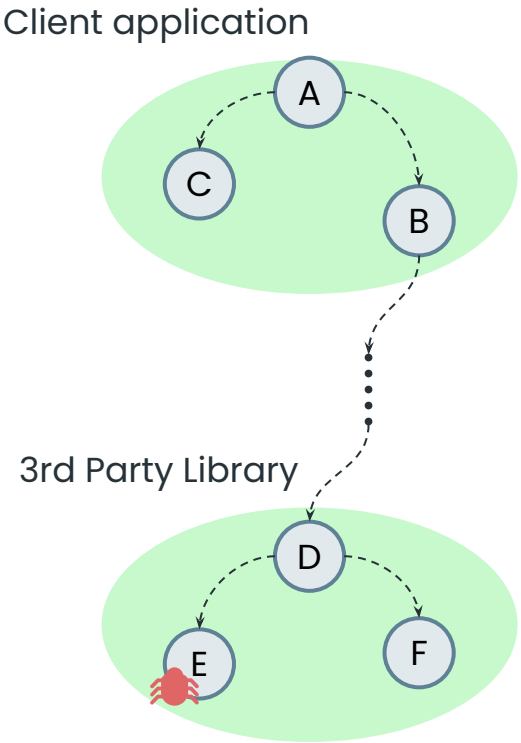
*Ponta, Serena & Plate, Henrik & Sabetta, Antonino. (2018).*

***Beyond Metadata: Code-centric and Usage-based Analysis of Known Vulnerabilities in Open-source Software.***



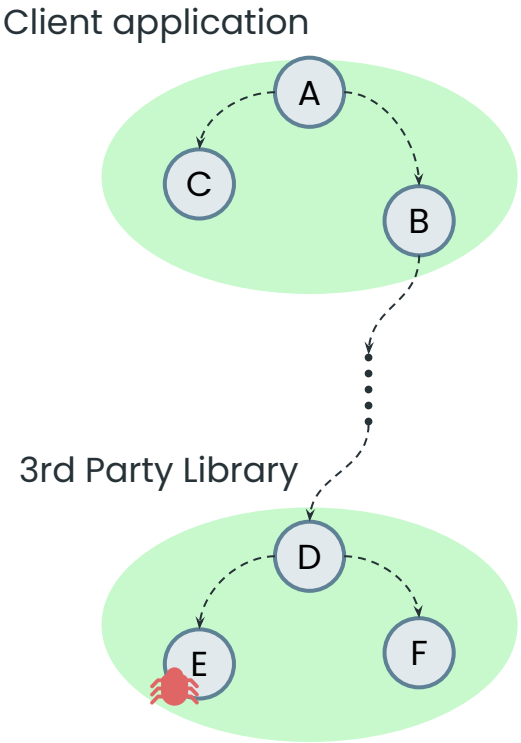
# SIEGE

**S**earch-based automat**I**c
**E**xploit **G**en**E**ration




# SIEGE

**S**earch-based automat**I**c
**E**xploit **G**en**E**ration

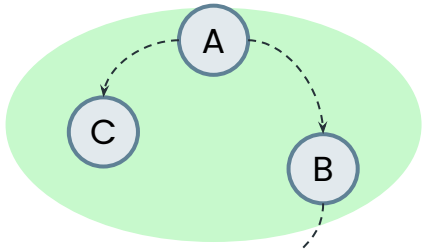



**Are we able to generate a test case that executes the vulnerable components?**

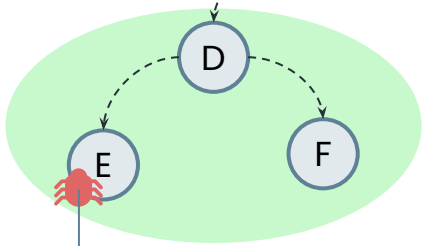
# SIEGE

**S**earch-based automat**I**c
**E**xploit **G**en**E**ration

Client application




3rd Party Library

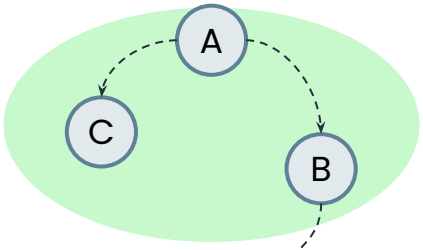



**Are we able to generate a test case that executes the vulnerable components?**

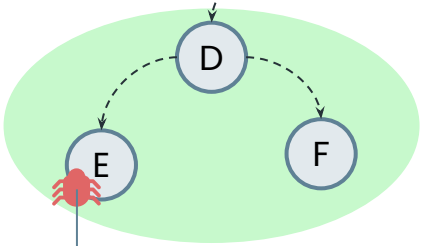
**SIEGE**

Vulnerability location

Client application



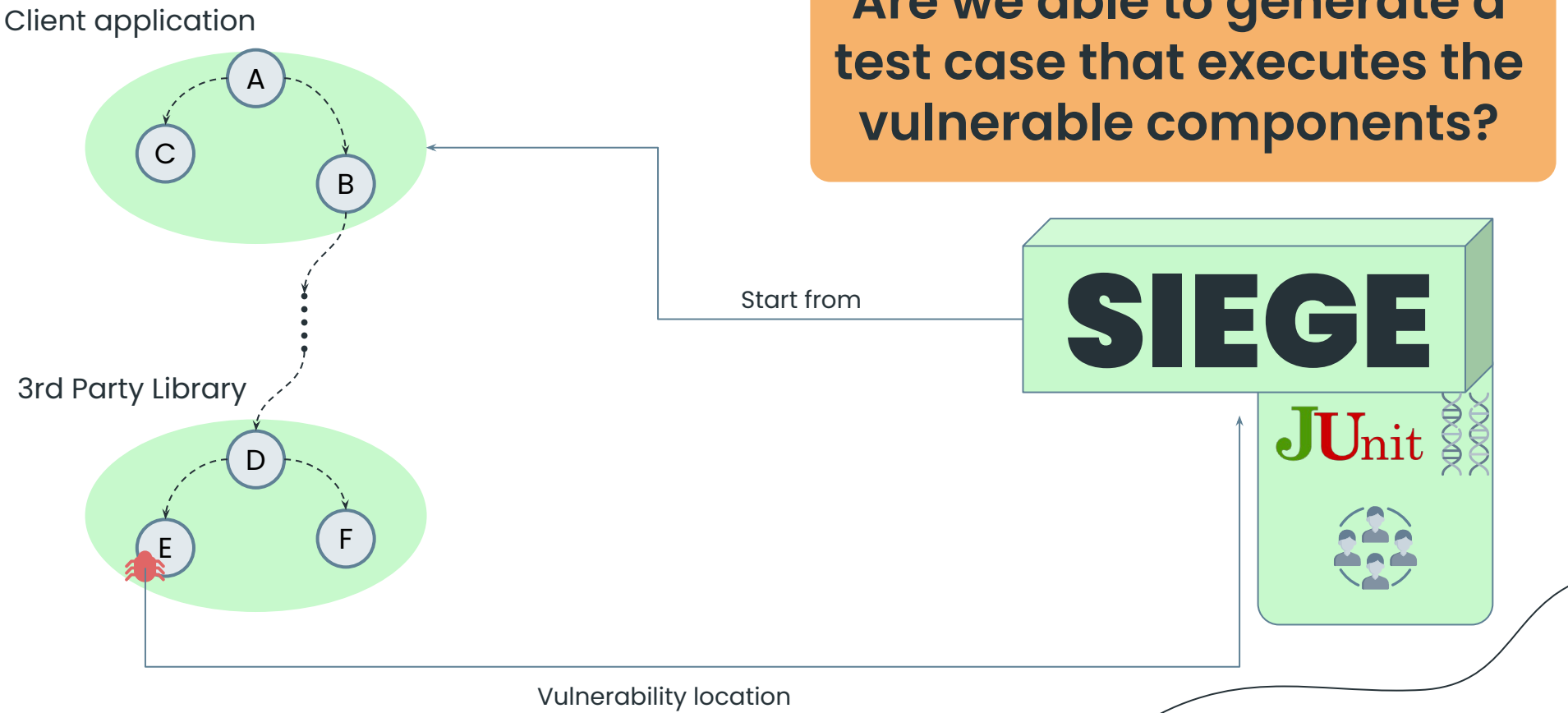

3rd Party Library

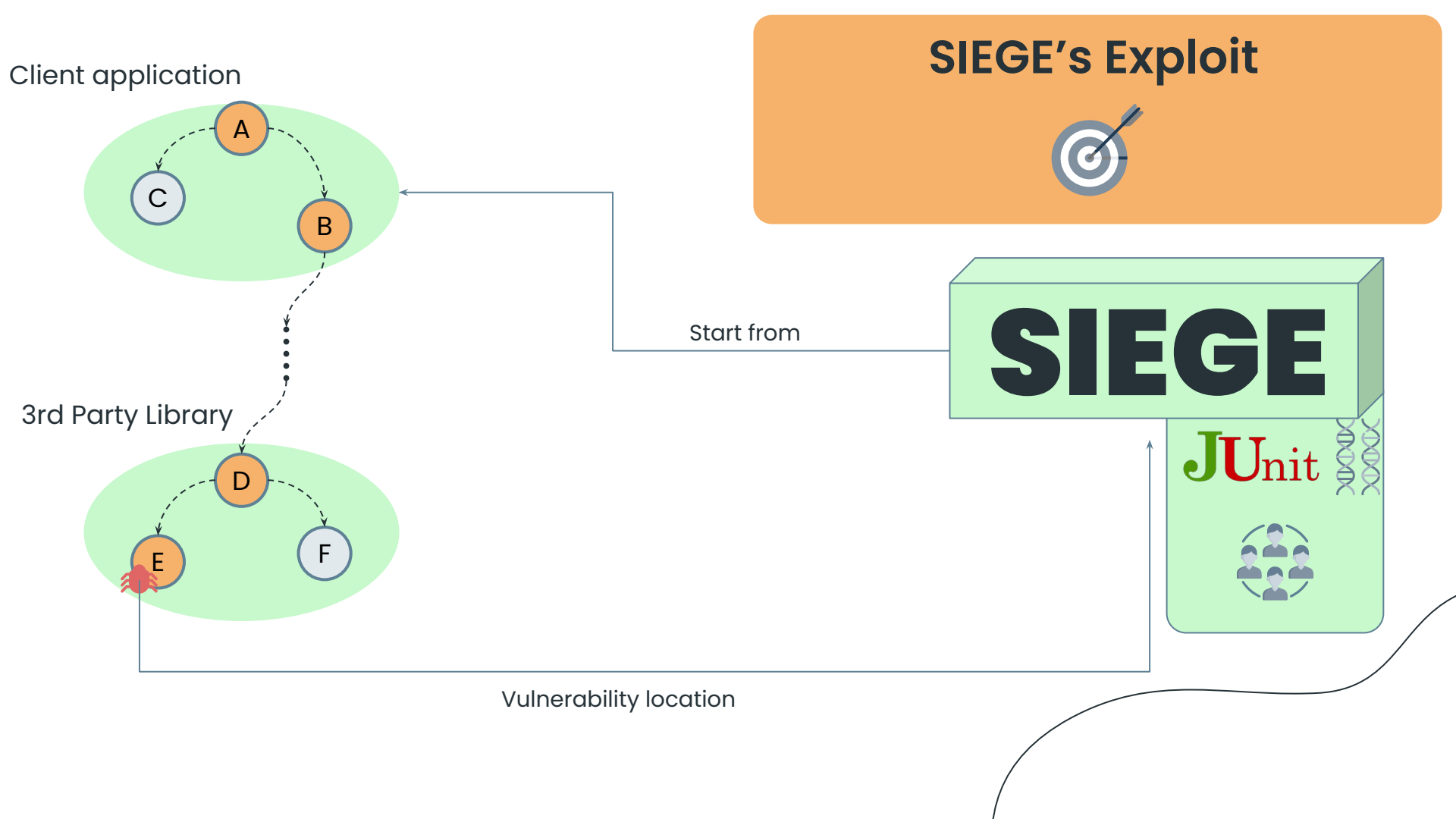



Vulnerability location

**Are we able to generate a test case that executes the vulnerable components?**

**Are we able to generate a test case that executes the vulnerable components?**




SIEGE's Exploit
Client application
A
C
B
3rd Party Library
D
E
F
Start from
SIEGE
JUnit
Vulnerability location

# The Implication

If SIEGE generates an exploit easily then an attacker can do it as well.

# Preliminary Evaluation

**11** Known vulnerabilities of Java libraries

**64%** Successful exploits with 60s budget

**Finding #1**

**The performance are influenced by the specific vulnerability**

**Finding #2**

**The performance are influenced by how the vulnerable component is called**

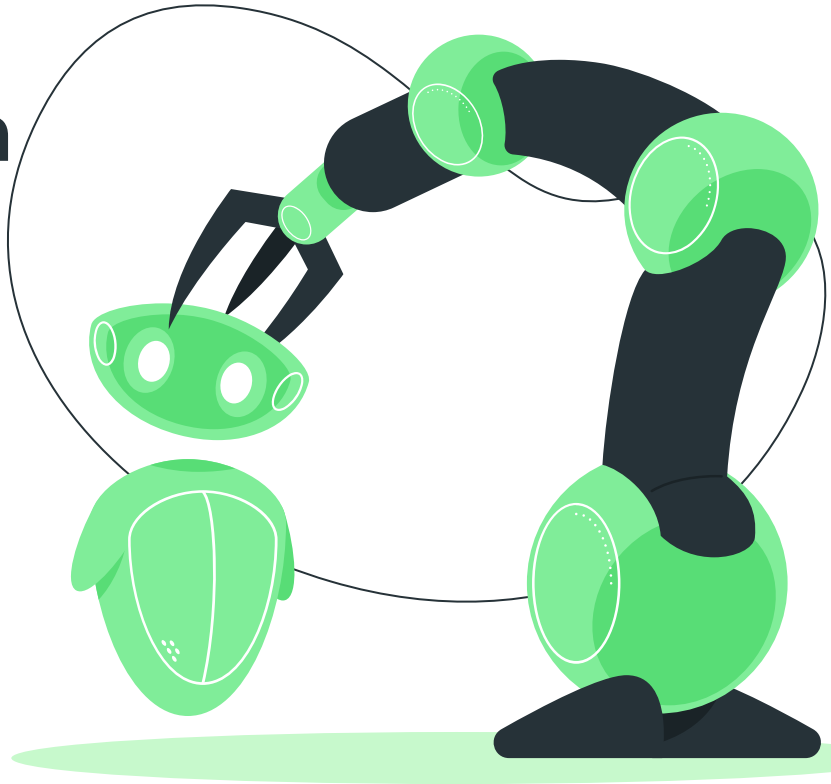
# The Future

- Integration in known vulnerability assessment tools
- Improve and extend the evaluation

eiannone@unisa.it
https://emaiannone.github.io/

# Toward Automated Exploit Generation for Known Vulnerabilities in Open-Source Libraries

**Emanuele Iannone**[*], Dario Di Nucci[†], Antonino Sabetta[‡] and Andrea De Lucia[*]



[*]*SeSa Lab* - University of Salerno, Fisciano, Italy
[†]*Tilburg University*, JADS, 's-Hertogenbosch, The Netherlands
[‡]*SAP Security Research*, France