Software Security testing

Institute for Software Technology
Research Group for Software Engineering
University of Koblenz-Landau

Supervisor: Mr. Sven Peldszus

Author: Bhupender Kumar Saini (219100887)

OUTLINE

UNIVERSITÄT KOBLENZ · LANDAU

1. Introduction
2. Risks and Observations
3. CIA triad
4. Security Requirements
5. Challenges in Security Testing
6. Techniques of Security testing in SDLC
7. Risk-Based Security Testing
8. Model Based Security Testing
9. Code-Based Testing and Static Analysis
10. Penetration testing
11. Vulnerability Scanning
12. Fuzz Testing
13. Selection Criteria for Security Testing tools
14. Mindset for Security testing
15. Conclusion
16. References

INTRODUCTION

UNIVERSITÄT KOBLENZ · LANDAU

What is Security testing ?

*"Software security is simply about the process of **building secure software** by designing software to be secure, **emphasizing** that the software is secure, and **educating** software architects, developers, and users about **how to build and use secure things**." - G. McGraw [1]*

*Software Security is the software's **ability to highly resist**, **tolerate**, and **recover** from cases that strongly **threaten** the product." - Julia Allen[2]*

Why Security testing ?

*Helps in **identifying weaknesses and security risks at earlier stage.***

RISKS AND OBSERVATIONS

UNIVERSITÄT KOBLENZ · LANDAU

Risks

Loss of **ability**

Loss of access to business-critical information

Loss of **customer trust**

Damage to company **reputation**

Lead to **fines and sanctions**

Observations

46% of businesses lost sensitive data.

90% of businesses admitted security incident

Enterprises lose **half a million US dollars** on average due to security breaches

https://media.kaspersky.com/pdf/it-risks-survey-report-cost-of-security-breaches.pdf

CIA TRIAD

UNIVERSITÄT KOBLENZ · LANDAU

Information is not **disclosed** to **unauthorized** individuals, processes, or devices.

Confidentiality

Data is **unchanged, not modified or destroyed.**

Integrity

Guarantees **timely and reliable access** to data and information services.

Availability

Ref : Michael Felderer et al. A classification for model-based security testing". In: Advances in System Testing and Validation Lifecycle (VALID 2011)] (2011), pp. 109-114.

REQUIREMENTS FOR SECURITY TESTING

UNIVERSITÄT KOBLENZ · LANDAU

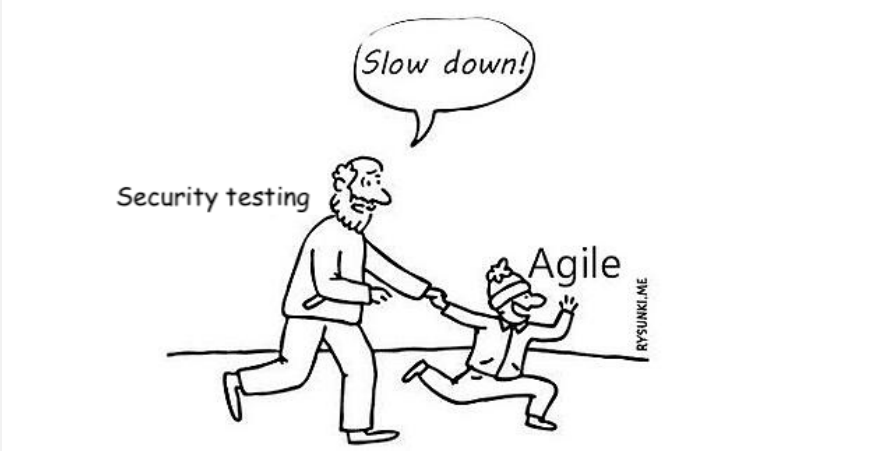
- Identification Requirements
- System Maintenance Security Requirements
- Survivability Requirements
- Security Auditing Requirements
- Privacy Requirements
- Non-repudiation Requirements
- Intrusion Detection Requirements
- Integrity Requirements
- Immunity Requirements

Ref: Donald Firesmith et al. "Engineering security requirements."

CHALLENGES IN SECURITY TESTING

UNIVERSITÄT KOBLENZ · LANDAU

- **Fast paced Development :**
  - Fast paced development can lead to bypassing security requirements.
- **Risks of Using Open/Commercial Source Components :**
  - Using without/little knowledge can lead to vulnerability and unwanted complexity.
- **Vulnerabilities in Code :**
  - Developed Application can have vulnerabilities due to weakness in languages.
  - Can create challenges in applying security policies



C language **buffer overflow error, format string vulnerability, integer errors.**

Ref: https://americansecuritytoday.com/secure-software-development-challenges-and-considerations/

CHALLENGES IN SECURITY TESTING

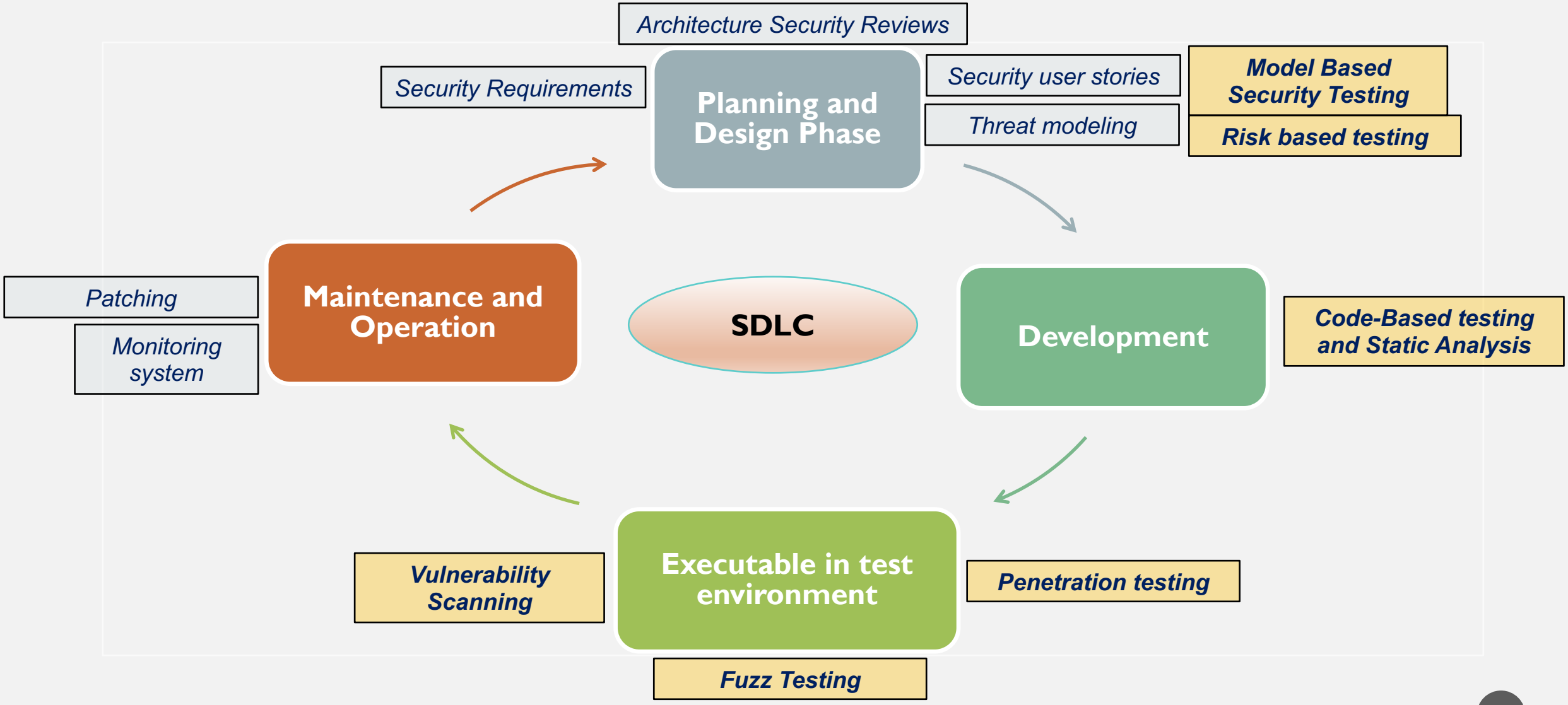
UNIVERSITÄT KOBLENZ · LANDAU

- **Lack of AppSec Planning :**
  - Can lead to unmanageable security issues and unclear requirement .

Ref: https://speakerstudio.typepad.com/.a/6a0134828b4555970c0134884f221f970c-pi

TECHNIQUES OF SECURITY TESTING IN SDLC

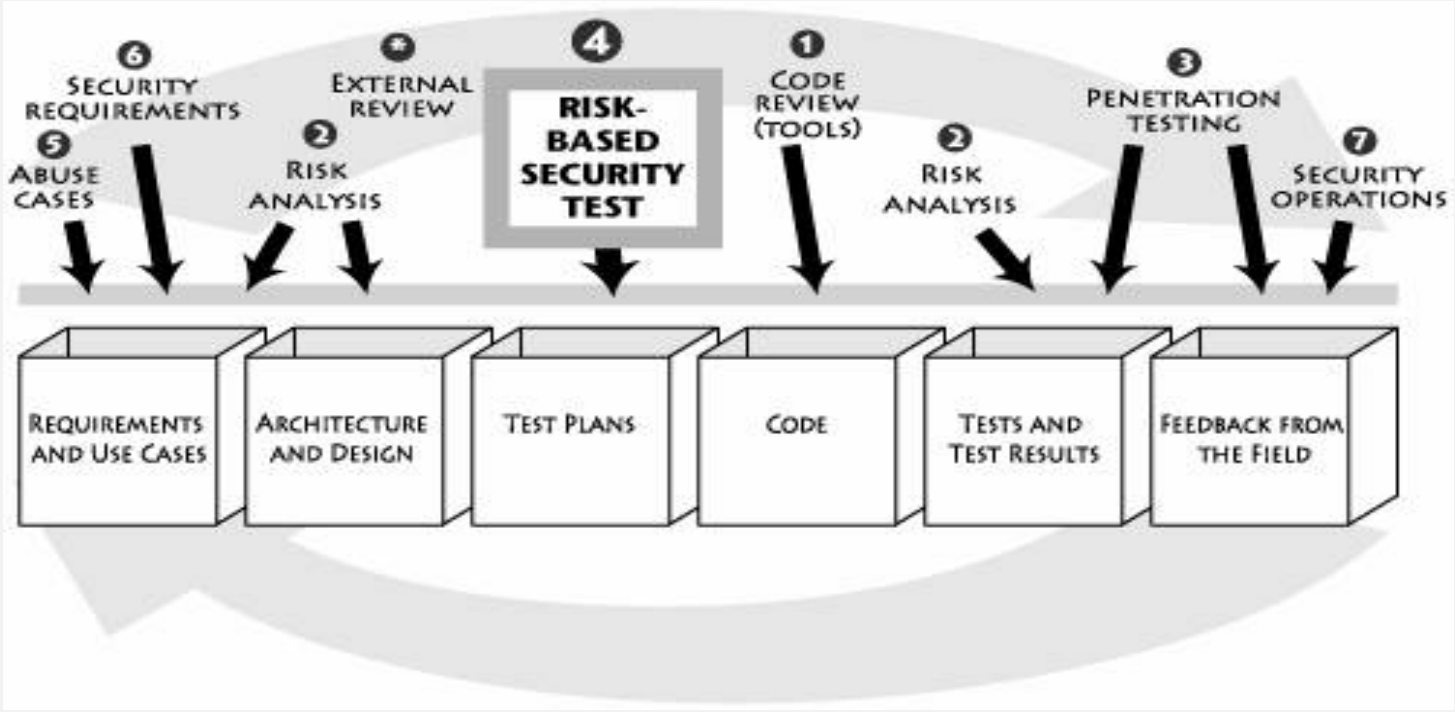
UNIVERSITÄT KOBLENZ · LANDAU



RISK-BASED SECURITY TESTING

UNIVERSITÄT KOBLENZ · LANDAU

- Optimizing process by **risk analysis**.
- Introduces the notion of **risk values**.
- Enable to **weigh test scenarios**.
- Emphasis on **negative requirements** or **abuse cases**.
- Provide **evidence** on the **functional correctness** of countermeasures and **absence of Known vulnerabilities**.
- Discovering **unknown vulnerabilities** at early stage.

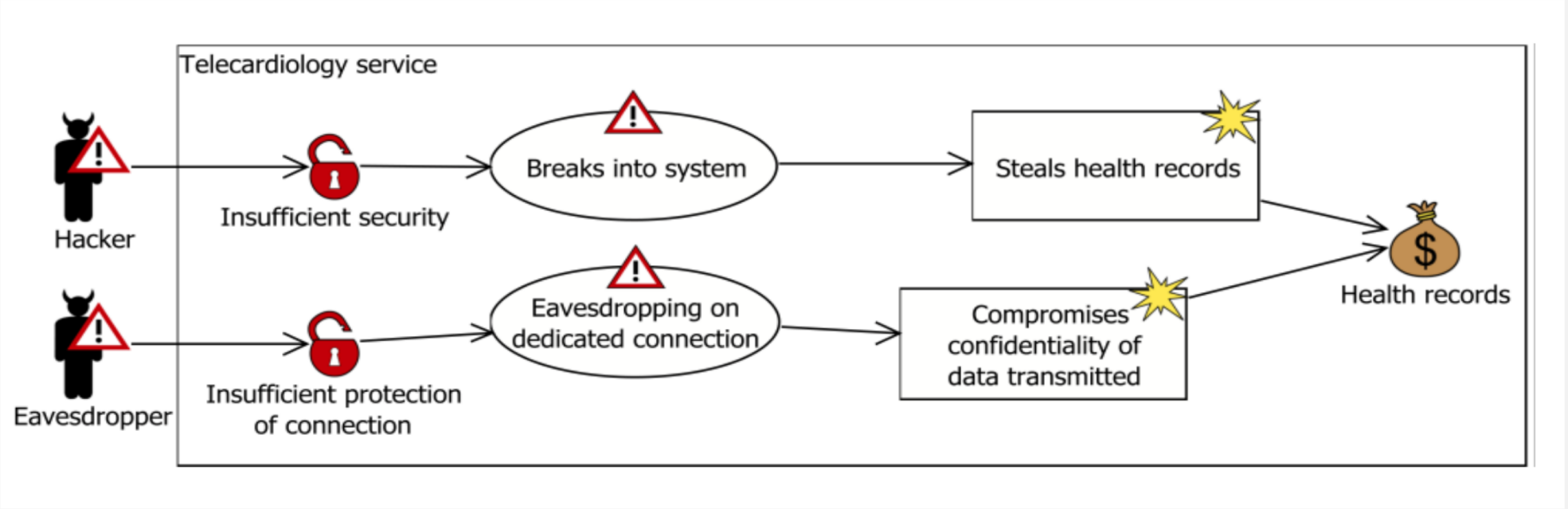


Ref: IEEE Security & Privacy magazine co-authored with Bruce Potter [Potter and McGraw 2004].

RISK-BASED SECURITY TESTING

UNIVERSITÄT KOBLENZ · LANDAU

- CORAS model-based method for security risk analysis.



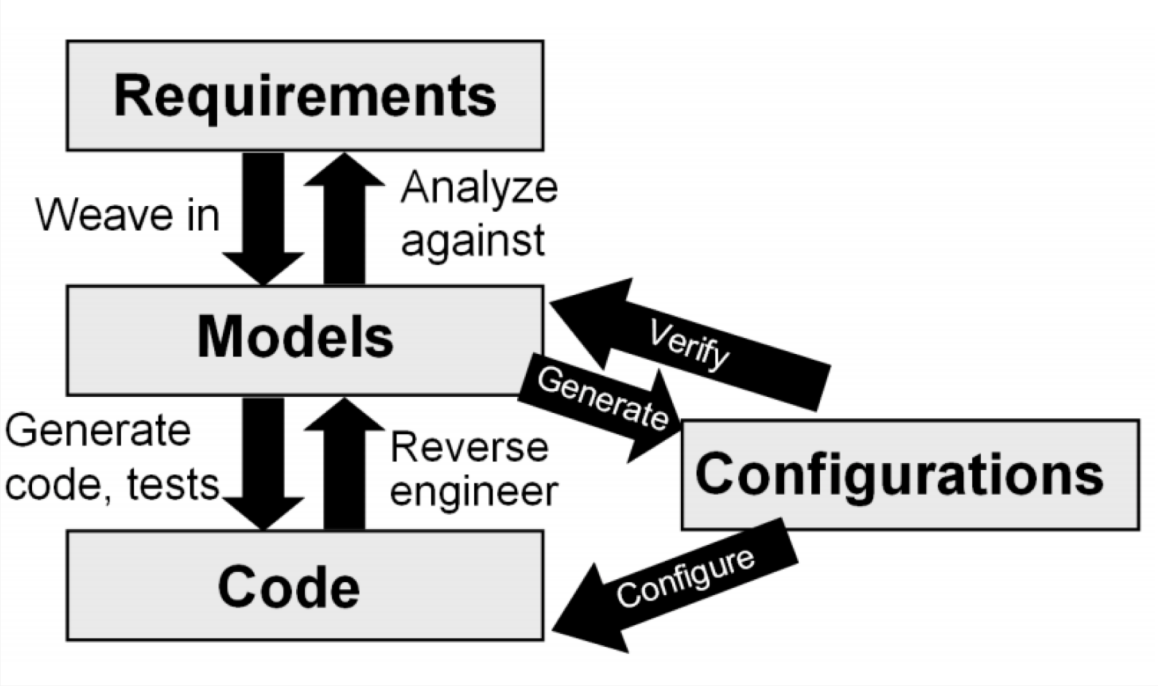
Initial threat diagram: deliberate actions

Ref: CORAS-handbook-v1.0

MODEL BASED SECURITY TESTING

UNIVERSITÄT KOBLENZ · LANDAU

- Relies on models.
- Generating **test cases automatically** from models(set of models).
- **Early vulnerability detection**.
- UMLsec modelling language for security testing.
- Reveal attacks which are visible only at model level. Mutations testing can reveal those.

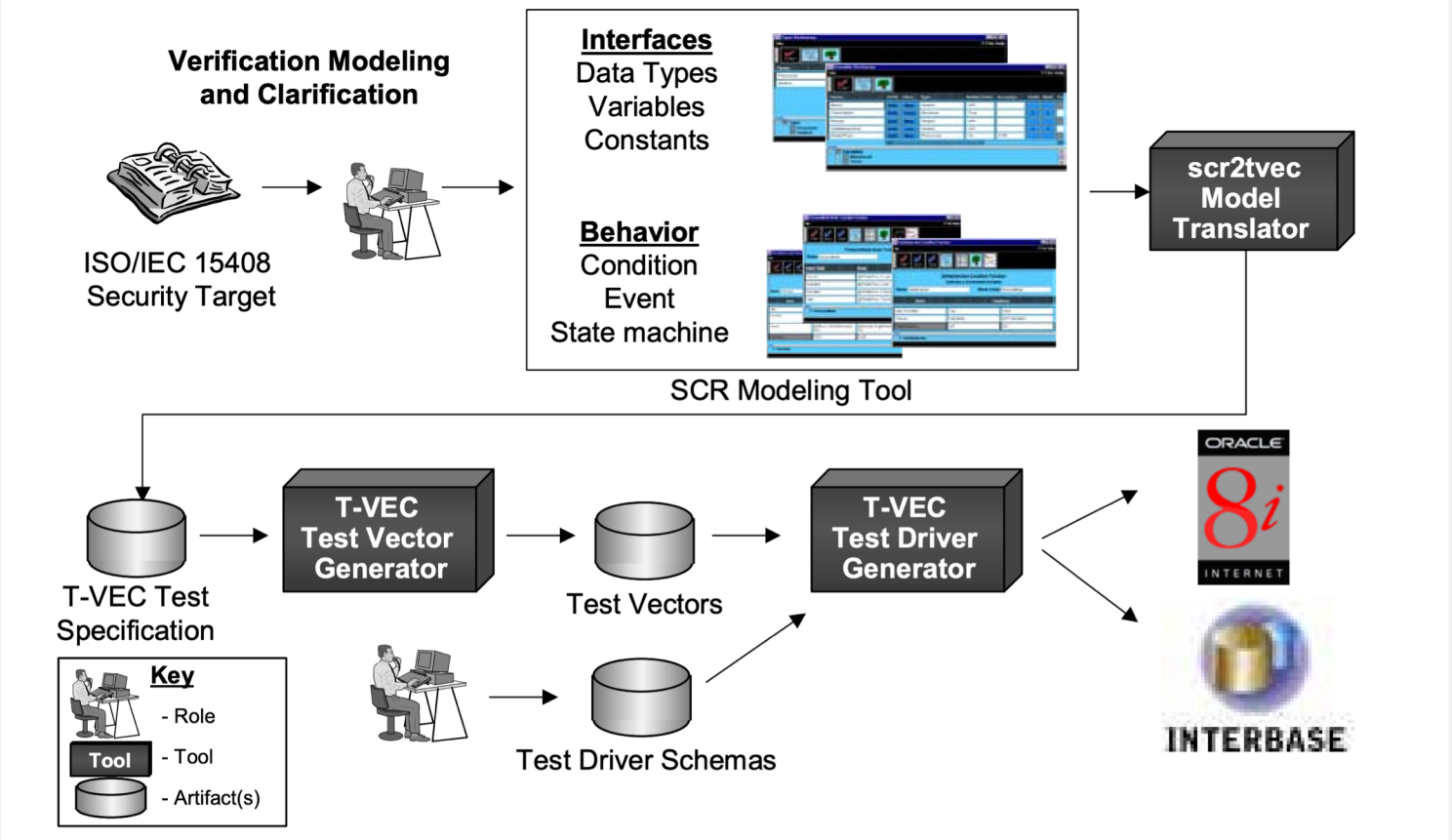


Ref *: Jan Jurjens. Model-based security testing using umlsec: A case study".In: Electronic Notes in Theoretical Computer Science 220.1 (2008)
**: Guido Wimmel and Jan Jurjens. Specication-based test generation for security-critical systems using mutations". In: International Conference on Formal Engineering Methods. Springer. 2002, pp. 471-482.

MODEL BASED SECURITY TESTING

UNIVERSITÄT KOBLENZ · LANDAU

- Developing models of security function specification to generate automatic test vector and test driver generation.
- And, then by using schemas, generating test cases.



Model-based Approach to Security Test Automation

Ref : Mark Blackburn et al. Model-based approach to security test automation". In: Proceeding of Quality Week 2001. 2001

CODE-BASED TESTING AND STATIC ANALYSIS

UNIVERSITÄT KOBLENZ · LANDAU

- Analyzing code **without execution (White Box security testing).**
- Powerful in detecting vulnerabilities in source code.
- **Strength :**
  - Reveal concurrency problems, time bombs, logic bombs, flawed business logic and many more.
  - Back doors, trojans, and other forms of malicious code can be exposed.
  - Scales well – can ran on lots of software
  - Real time feedback to developers.
  - Early bugs
- **Weakness:**
  - False positive
  - Infeasible to detect run-time errors
  - Unable to detect authentication, cryptography, access control issues.

CODE-BASED TESTING AND STATIC ANALYSIS

UNIVERSITÄT KOBLENZ · LANDAU

- **Principle**: Deducing, Data flow analysis(DFA), Control Flow Graph (CFG), Taint analysis and Lexical analysis.
- DFA tries to collect **run-time information** of data in software **in a static state**(Wögerer-2005).
- DFA common terms are **basic block** (the code), **Control Flow Analysis** (the flow of data) **and Control Flow Path** (the path the data takes).
- **Lexical analysis**: preprocess and tokenize of source and then match token stream against a library of vulnerable constructs.
- **Tools** : ITS4, FlawFinder RATS , Veracode

```
$a = 0;
$b = 1;

if ($a == $b)
{ # start of block
  echo “a and b are the same”;
} # end of block
else
{ # start of block
  echo “a and b are different”;
} # end of block
```

PHP basic block

Ref: https://owasp.org/www-community/controls/Static_Code_Analysis

Ref: Wögerer W. A survey of static program analysis techniques. Tech. rep., Technische Universität Wien; 2005 Oct 18.

PENETRATION TESTING

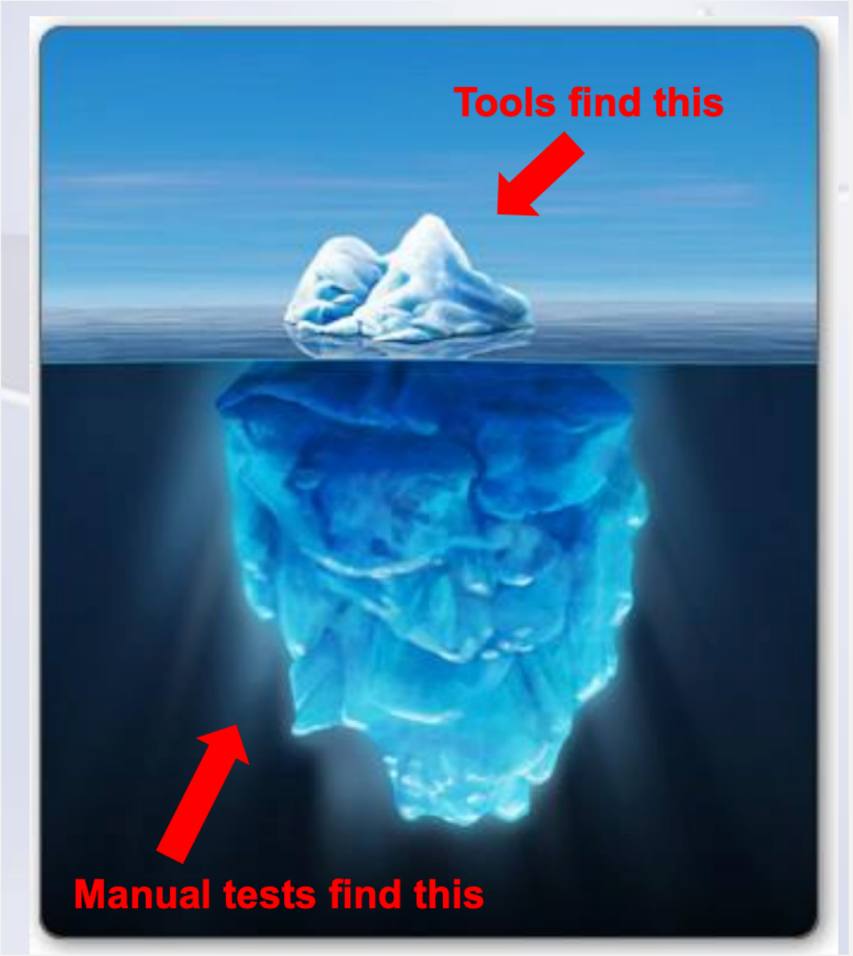
UNIVERSITÄT KOBLENZ · LANDAU

- Tester attempts **to compromise the security features** of a system **based on their understanding** of the system design and implementation.
- **Hacking own system before someone else.**
- **Types :** black-box, white-box, and grey-box
- **Strength :**
  - Can identify a range of vulnerabilities including high-risk weaknesses.
  - Unable to reveal both unknown and known hardware or software flaws.
  - Enables a proactive security approach.
- **Weakness:**
  - Demands deep knowledge and imaginations.
  - Can give false sense of security.

PENETRATION TESTING

UNIVERSITÄT KOBLENZ · LANDAU

- Popular penetration testing methodologies are :
  - OSSTMM (Open Source Security Testing Methodology Manual)
  - OWASP (Open Web Application Security Project)
  - NIST (National Institute of Standards and Technology)
  - PTES (Penetration Testing Methodologies and Standards)
  - ISSAF (Information System Security Assessment Framework)
- **Tools:** Wireshark, Kali linux, Metasploit, OSWAP, Ettercap, Nmap etc.



VULNERABILITY SCANNING

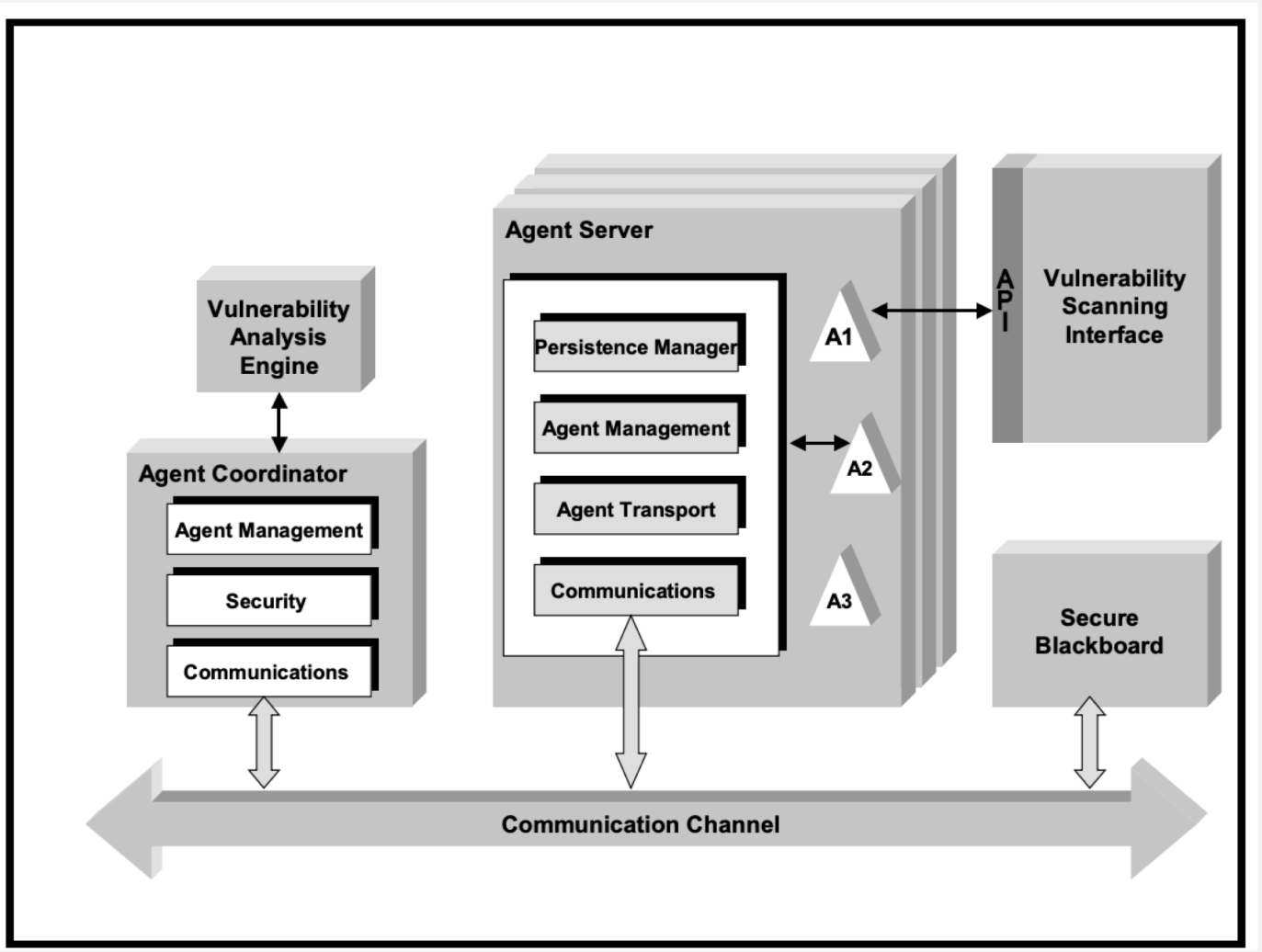
UNIVERSITÄT KOBLENZ · LANDAU

- Automatically scans for software security risks, testing space and known defects.
- For ex. network port scanning reveal vulnerable ports.
- Vulnerability scanning look for known vulnerability signature or pattern.
- **Strength** :
  - Faster result because of automated scanning tools.
  - Can be performed any time.
  - Save time.
- **Weakness**:
  - Only identifies known vulnerabilities.
  - False Positive.

VULNERABILITY SCANNING

UNIVERSITÄT KOBLENZ · LANDAU

- **Tools**:
  - Security Prole Inspector (SPI)
  - Internet Security Scanner (ISS)
  - Security Analysis Tool for Auditing Network (SATAN)
  - Tiger
  - Sscan
  - Nmap
  - COPS
  - Tripwire.
- Jeffrey W. Humphries et at. proposed vulnerability scanning system using secure mobile agents that is easy to customize and is also resistant to attack.

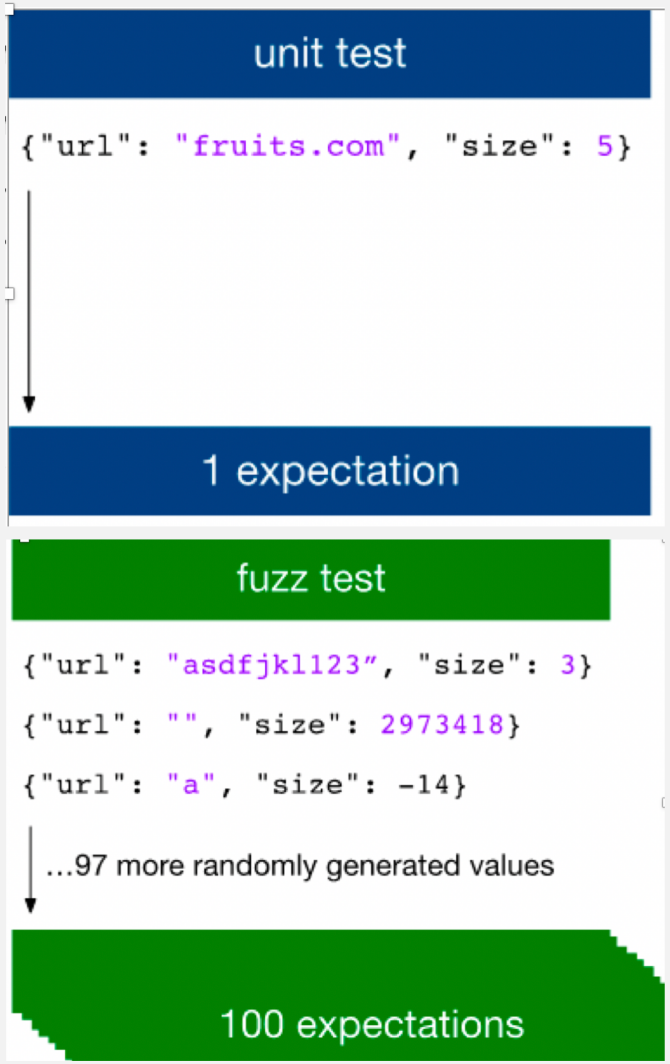


Ref*: Jerey W Humphries, Curtis A Carver Jr, and Udo W Pooch. Secure mobile agents for network vulnerability scanning". In: Proceedings of the Software Security Testing 21 2000 IEEE Workshop on Information Assurance and Security. 2000, pp. 6-7.

FUZZ TESTING

UNIVERSITÄT KOBLENZ · LANDAU

- Injecting **semi-random/random data** into a **SUT** in search of unexpected behavior.
- **Strength :**
  - Finds flaws which are difficult to get via logical methods.
  - Very less computational cost and time.
  - Best for SQL injection, buffer overflow, denial of service(DOS) and cross site scripting.
- **Weakness:**
  - Demands deep knowledge and imaginations.
  - Can give false sense of security.

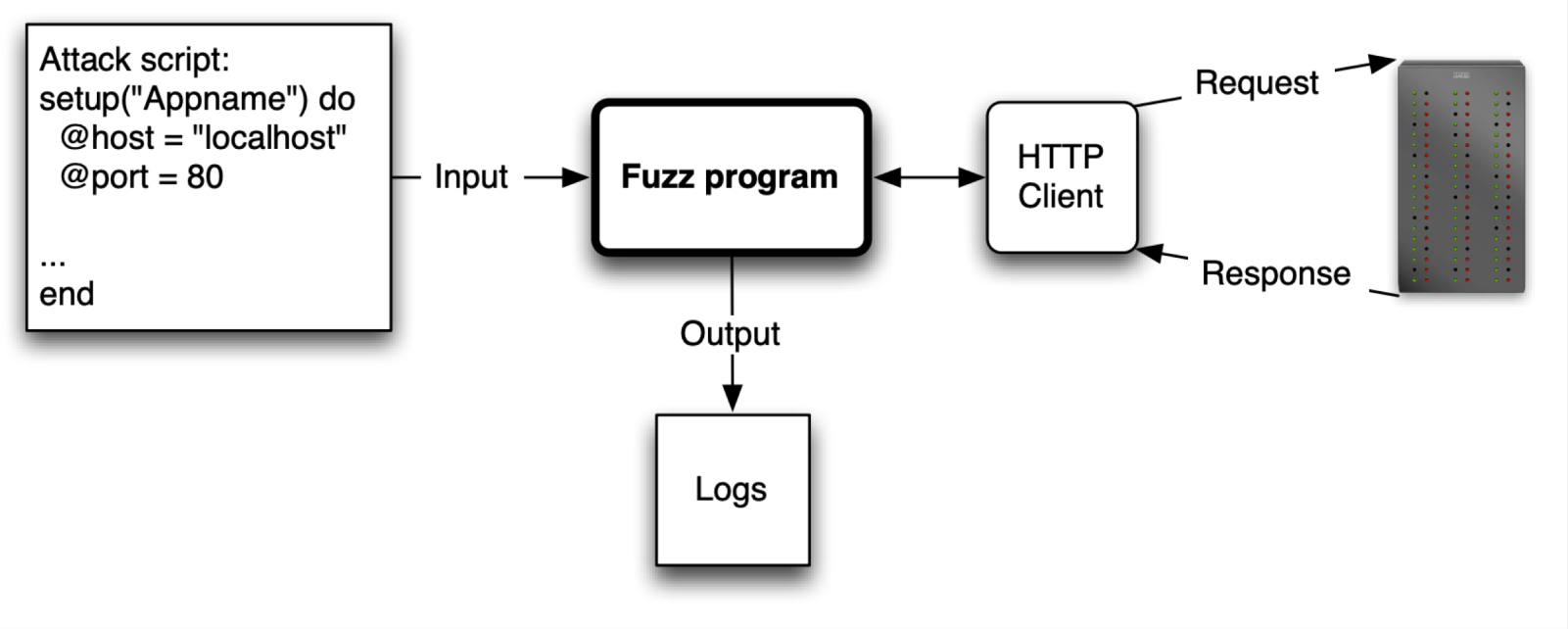


https://freecontent.manning.com/writing-fuzz-tests-in-elm/

FUZZ TESTING

UNIVERSITÄT KOBLENZ · LANDAU

- **Tools:**
  - JBroFuzz
  - WSFuzzer
  - American fuzzy lop
  - Radamsa - a flock of fuzzers
  - Microsoft SDL MiniFuzz File Fuzzer
- Rune Hammersland et al. proposed method to (semi) automatic generation of pseudo random test data (fuzzing) for web application.



Ref : Hammersland, Rune, and Einar Snekkenes. "Fuzz testing of web applications." *Retrieved* 7.16 (2008): 2010.

SELECTION CRITERIA FOR TOOLS

UNIVERSITÄT KOBLENZ · LANDAU

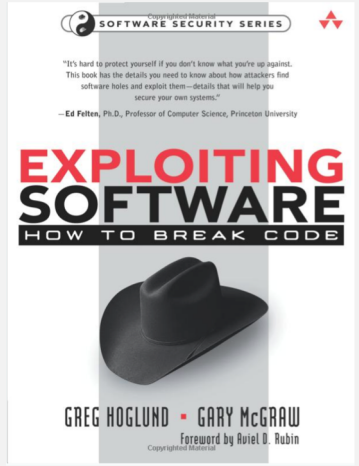
- **Attack surface** : Total vulnerability that can be exploited.
- **Application type** : Web based, Desktop, Mobile app, Cloud based, firmware etc.
- **Quality of results and usability :** False positive rate, scalability, maintainability etc.
- **Supported technologies :** programming languages, interfaces etc.
- **Performance and resource utilization :** Man power, computational power etc.
- **Costs for licenses, maintenance, and support**

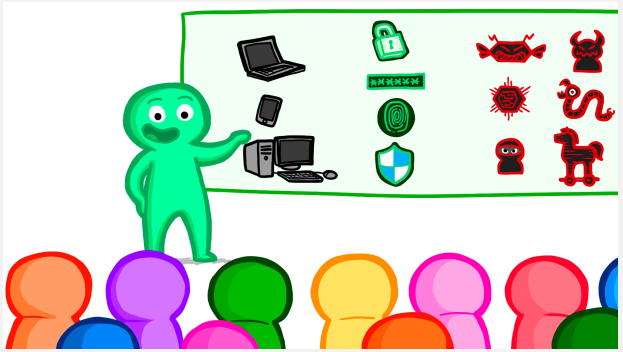
Ref: Michael Felderer et al. Security testing: A survey". In: Advances in Computers. Vol. 101. Elsevier, 2016

MINDSET FOR SECURITY TESTING

UNIVERSITÄT KOBLENZ · LANDAU

- Security mindset makes better tester.
- Some key aspect required for security testing are :
  - **Extensive knowledge of the domain.**
  - **Imagination**
  - **An attacker mindset**





https://www.csforallteachers.org/event/teaching-cybersecurity-csp-or-any-cs-class-introducing-security-mindset

Ref: Sara Hooshangi, Richard Weiss, and Justin Cappos. "Can the security mindset make students better testers?" In: Proceedings of the 46th ACM Technical Symposium on Computer Science Education. 2015

CONCLUSION

UNIVERSITÄT KOBLENZ · LANDAU

- Organizations have to **focus equally or more** on the security aspect of the applications.
- **Test as early as possible.**
- **Adopt to systematic approaches**/frameworks of security testing.
- **Utilize tools** in fast paced SDLC.
- More **focused research on latest technologies** like ubiquitous computing, semantic web, HTML5.
- **Automation** of security testing **demands** focus, especially **usability aspect**.
- Vulnerability not only security flaws, it can be **unexpected but intentional misuse of application**.
- Tester have to be **creative** and **have mindset of attacker**.
- Fully integrated testing tool for model based testing is still a long way.

REFERENCES

UNIVERSITÄT KOBLENZ · LANDAU

1. G. McGraw. Software security". In: IEEE Security Privacy 2.2 (2004)
2. Jason Bau et al. \State of the art: Automated black-box web application vulnerability testing"
3. https://www.stickyminds.com/article/shift-left-approach-software-testing
4. https://americansecuritytoday.com/secure-software-development-challenges-and-considerations/
5. Michael Felderer et al. A classification for model-based security testing". In: Advances in System Testing and Validation Lifecycle (VALID 2011)] (2011), pp. 109-114.
6. Ina Schieferdecker, Juergen Grossmann, and Martin Schneider. Model based security testing".
7. Donald Firesmith et al. "Engineering security requirements."
8. Jan Jurjens. Model-based security testing using umlsec: A case study".In: Electronic Notes in Theoretical Computer Science 220.1 (2008).
9. Jerey W Humphries, Curtis A Carver Jr, and Udo W Pooch. Secure mobile agents for network vulnerability scanning". In: Proceedings of the Software Security Testing 21 2000 IEEE Workshop on Information Assurance and Security. 2000, pp. 6-7
10. Wögerer W. A survey of static program analysis techniques. Tech. rep., Technische Universität Wien; 2005 Oct 18.
11. Guido Wimmel and Jan Jurjens. \Specification-based test generation for security-critical systems using mutations". In: International Conference on Formal Engineering Methods. Springer. 2002, pp. 471-482.
12. Jerey W Humphries, Curtis A Carver Jr, and Udo W Pooch. Secure mobile agents for network vulnerability scanning". In: Proceedings of the Software Security Testing 21 2000 IEEE Workshop on Information Assurance and Security. 2000, pp. 6-7.
13. Hammersland, Rune, and Einar Snekkenes. "Fuzz testing of web applications." *Retrieved* 7.16 (2008): 2010.
14. Petar Tsankov, Mohammad Torabi Dashti, and David Basin. SECFUZZ: Fuzz-testing security protocols". In: 2012 7th International Workshop on Automation of Software Test (AST). IEEE. 2012, pp. 1-7.
15. Sara Hooshangi, Richard Weiss, and Justin Cappos. "Can the security mindset make students better testers?" In: Proceedings of the 46th ACM Technical Symposium on Computer Science Education. 2015

UNIVERSITÄT KOBLENZ · LANDAU

THANK YOU !