Security-oriented software testing

Questões de Segurança em Engenharia de Software (QSES)
Mestrado em Segurança Informática
Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto

Eduardo R. B. Marques, edrdo@dcc.fc.up.pt

Security testing

- How does security testing differ from standard program testing?
  - **Security features** must be tested with regard to possible adversarial actions **=>** guided by the requirements posed to security-minded code.
  - Other features must be tested in respect to **“unintended behavior” =>** guided by the possibility of common vulnerabilities in standard code
- What are general recommended practices? How can standard or specific testing approaches help?
- As an introductory discussion, let’s have a look at some of the activities in the BSIMM Security Testing touchpoint.

BSIMM - Security Testing

- *[ST1.1: 100] “Ensure QA supports* ***edge/boundary value condition testing****”*
  - “*The QA team goes beyond functional testing to perform basic adversarial tests and probe simple* ***edge cases and boundary conditions****, no attacker skills required.”*
- *[ST1.3: 88] “Drive tests with* ***security requirements and security features.****”*
  - *“For the most part, security features can be tested in a fashion similar to other software features.”*
- **Standard software testing practices apply. We will see a few important approaches in this sense.**

BSIMM - Security Testing (cont.)

- *“[ST3.4: 3]* ***Leverage coverage analysis****”*
  - *“Testers measure the code coverage of their security tests. Code coverage analysis drives increased security testing depth.”*
- ***“****[ST2.5: 12]* ***Include security tests in QA automation”***
  - *“Security tests run alongside functional tests as part of* ***automated regression testing.*** *In fact, the same automation framework houses both, and security testing is part of the routine.”*
- **Again, standard software testing practices apply.**
- [ST2.5: 12] mentions regression testing ? **Q:** Are you familiar with it?

A note on regression testing

- **Regression testing:** testing if updated software still behaves the same (in regard to unchanged requirements) after a change in its code (bug fix, new feature) or after integration with an updated version of an external component.
- Ideally, a regression test suite is composed of a minimal set of “core tests” that always run in automated fashion whenever an update takes place.
- In a large code base, determining which tests should be part of (or removed from) the regression suite is not straightforward. Running regression test suites (if too big) for instance on every build may also be quite costly.

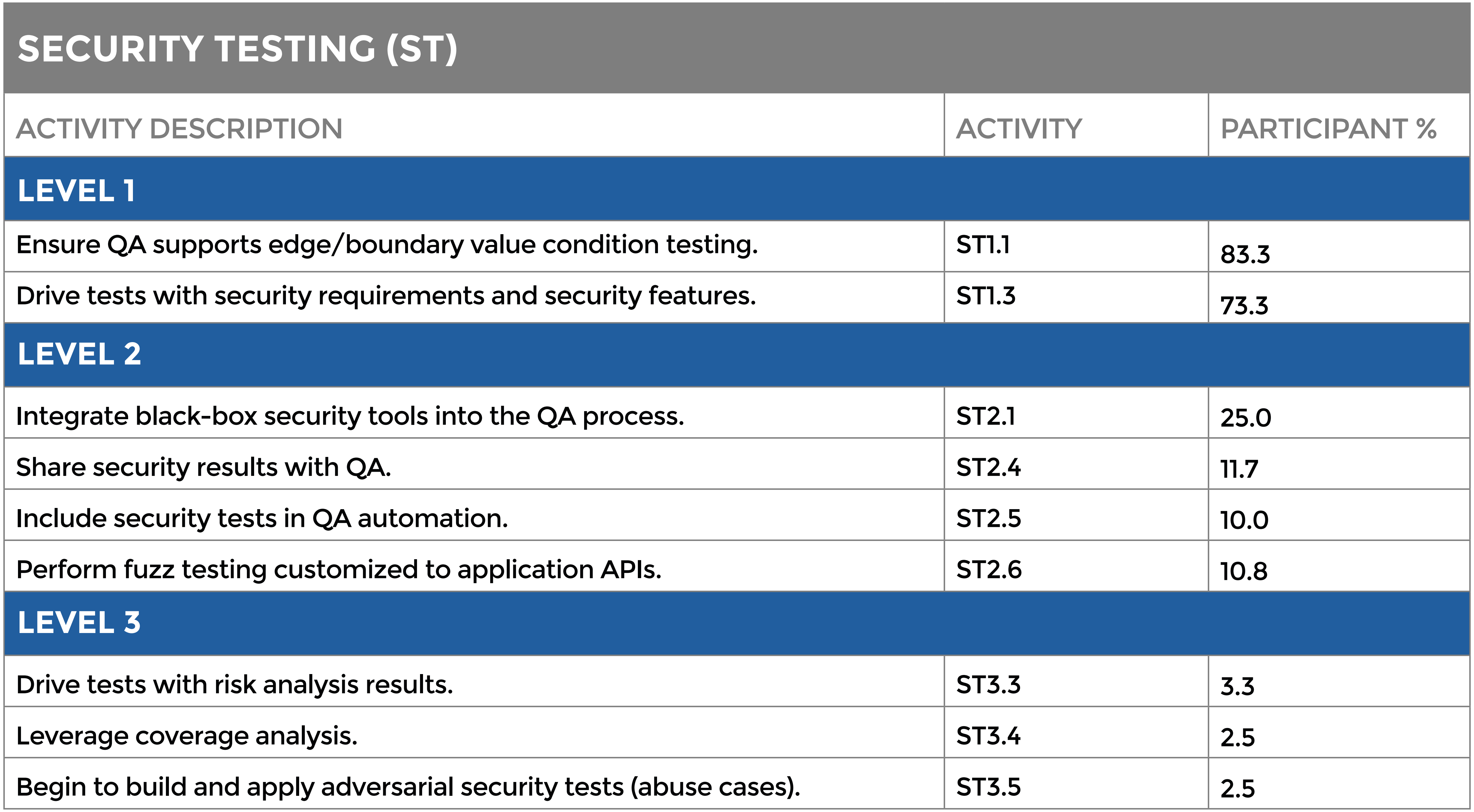
BSIMM - Security Testing (cont.)

- *[ST2.1: 30] “Integrate* ***black-box security tools*** *into the QA process.”*
  - “*The organization uses one or more black-box security testing tools as part of the QA process. Such tools are valuable because they encapsulate an* ***attacker’s perspective****, albeit generically”.*
  - Some commercial frameworks are mentioned in the text combining static/ dynamic analysis, pen-testing and **fuzz testing.**
- *[ST2.6: 13] Perform* ***fuzz testing customized to application APIs.***
  - *“Test automation engineers or agile team members customize a* ***fuzzing framework*** *to the organization’s APIs.The fuzzing framework has a built-in understanding of the application interfaces it calls into.”*
- **Fuzz testing? A common practice in security-oriented testing, but not in standard program testing. We will see what it means.**

BSIMM - Security Testing (cont.)

- *“[ST3.3: 4] Drive tests with **risk analysis** results.”*
  - *“Testers use architecture analysis results to direct their work […] Adversarial tests like these can be developed according to risk profile, with high-risk flaws at the top of the list.”*
- *“[ST3.5: 3] Begin to build and **apply adversarial security tests (abuse cases)**.”*
  - *“Testing begins to incorporate test cases based on abuse cases”*
- These are important aspects mentioned earlier in the course. They require expertise and overall understanding of:
  - the software architecture and design
  - the inherent attack surface
  - the SDLC process in place

BSIMM testing activities — state of practice (2018)



SECURITY TESTING (ST)

ACTIVITY DESCRIPTION	ACTIVITY	PARTICIPANT %
LEVEL 1		
Ensure QA supports edge/boundary value condition testing.	ST1.1	83.3
Drive tests with security requirements and security features.	ST1.3	73.3
LEVEL 2		
Integrate black-box security tools into the QA process.	ST2.1	25.0
Share security results with QA.	ST2.4	11.7
Include security tests in QA automation.	ST2.5	10.0
Perform fuzz testing customized to application APIs.	ST2.6	10.8
LEVEL 3		
Drive tests with risk analysis results.	ST3.3	3.3
Leverage coverage analysis.	ST3.4	2.5
Begin to build and apply adversarial security tests (abuse cases).	ST3.5	2.5

- Source: **BSIMM 9** - Gary McGraw, Ph.D., Sammy Migues, and Jacob West
- “[the] *result of a multiyear study of real-world software security initiatives We present the BSIMM8 model as built directly out of data observed in 109 software security initiatives”*