



STP SOFTWARE
TEST PROFESSIONALS
CONFERENCE SPRING 2019




Security from Testers Perspective

Guillermo Skrilec - QAlified | @gskrilec

1

About me

- Computer Systems Engineer
- Master in Management of Tech Companies (MBA)
- SQA Manager & VP of US Operations at GeneXus Consulting
 - + 10 years in SW testing
 - + 70 projects in 6 countries
- CEO at QAlified (spin-off)
- Testing teacher and advisor
- TestingUy community co-organizer
- STWC South America main judge
- 7WCSQ / CAST 2017 speaker
- CSM / ISTQB Test Manager / Quality Systems Manager

2



TestingUy



- Since 2013
- Conference
- Meetups

testinguy.org
@TestingUy



3



4

Software Security

It's not about:

- Network (SSL, Firewall, Ports, ...)
- Obfuscating code
- Enforce policies

Protection of software after it's already built



5

Software Security

It's about building secure software:

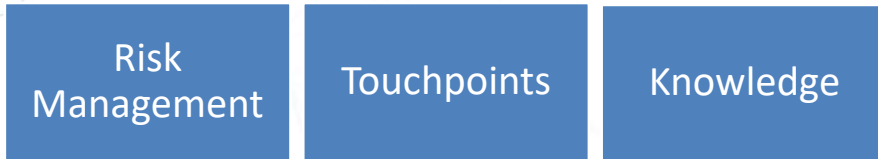
- Designing software to be secure
- Making sure that software is secure
- Educating developers, testers, architects and users about how to build security in

Security is not a set of features



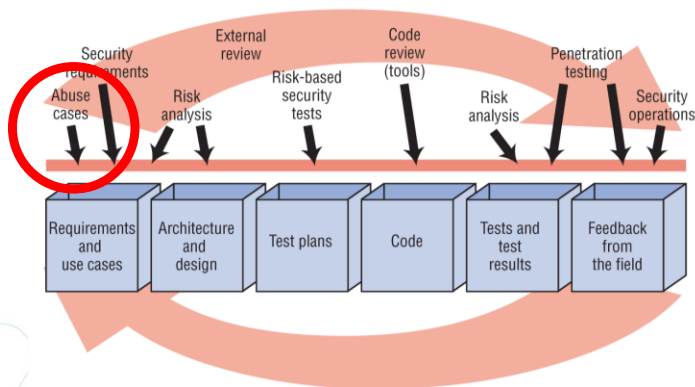
6

3 Pillars



7

Touchpoints



8

Abuse cases

Use cases: The focus is on functionality

If your software is going to be used,
it's going to be abused

Get out your black hat and think like a bad guy



9

Abuse cases

Consider a payroll system that allows a human resources department to control salaries and benefits.

- “The system allows users in the HR management group to view and modify salaries of all employees”
- “The system will only allow a basic user to view his or her own salary”



10

Abuse cases

Think like a potential attacker:

- An attacker is likely to try to gain extra privileges in the payroll system and remove evidence of any fraudulent transaction.
- Might try to delay all the paychecks by a day or two and embezzle the interest that is accrued during the delay.



11

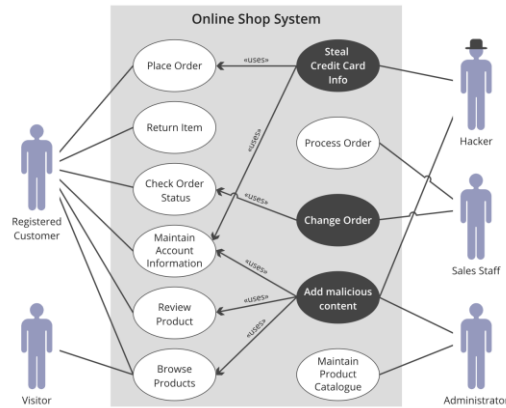
OWASP Cheat Sheet

[https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/Abuse Case Cheat Sheet.md](https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/Abuse%20Case%20Cheat%20Sheet.md)



12

Abuse cases



STP SOFTWARE TEST PROFESSIONALS CONFERENCE SPRING 2019

QAlified Software Quality Assurance

13

Use case family

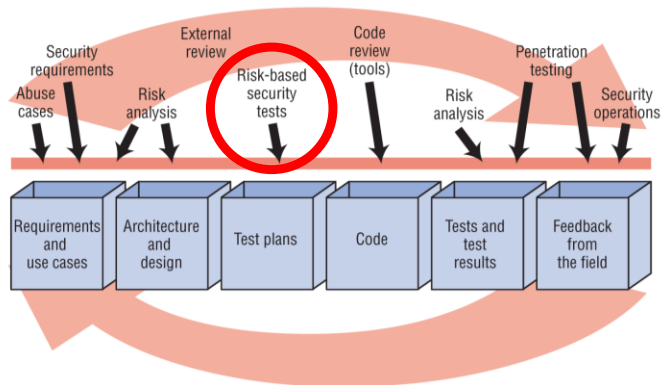
Type	Actors and actions	Example
Use case	Insiders doing appropriate tasks	Customer is paying his bill
Abuse case	Outsiders trying to breach the system	Criminal executes phishing attack to gather credentials
Misuse case	Insiders doing inappropriate tasks intentionally	Administrator stealing log files with credit card info and social security numbers
Confuse case	Insider doing inappropriate tasks unintentionally	User reading malicious e-mail on a PC without antivirus software or malware protection

STP SOFTWARE TEST PROFESSIONALS CONFERENCE SPRING 2019

QAlified Software Quality Assurance

14

Touchpoints



STP SOFTWARE
TEST PROFESSIONALS
CONFERENCE SPRING 2019

QAlified
Software Quality Assurance

15

Risk-based security tests

Security testing must encompass 2 strategies:

1. Functional security testing

Testing security mechanisms to ensure that their functionality is properly implemented

2. Adversarial security testing

Performing risk-based security testing motivated by understanding and simulating the attacker's approach

STP SOFTWARE
TEST PROFESSIONALS
CONFERENCE SPRING 2019

QAlified
Software Quality Assurance

16

Adversarial security testing

Focus: Testing for business logic

- This type of vulnerabilities cannot be detected by a vulnerability scanner and relies upon the skills and creativity of the tester.
- In addition, this type of vulnerabilities is usually one of the hardest to detect, application specific, and one of the most detrimental to the application if exploited.



17

Security Testing vs. Penetration Testing



18

Auction example

The screenshot shows an auction interface. On the left, the current bid is \$1.84 USD with a timer at 00:00:08. The bidder's name is 'clif3232' and there is a 'Bid Now' button. Below this, the value price is \$25.00 and the bid credit is -\$30.00. On the right, a 'Bidding History' table lists recent bids from various users.

Username	Amount	Bid Type
clif3232	\$1.84	Single Bid
S...LONON00	\$1.83	Single Bid
clif3232	\$1.82	Single Bid
...	\$1.81	Single Bid
...	\$1.80	Single Bid
...	\$1.79	Single Bid
...	\$1.78	Single Bid
...	\$1.77	Single Bid
...	\$1.76	Single Bid

At the bottom of the table, it says '11 Recent Bidders' and 'Bid-O-Matic'.

Current winner (username)

The screenshot shows a login page with the heading 'Login' and the instruction 'Please login to your account'. A red error message box states: 'Following errors occurred: Too many login attempts. Please try again in 60 seconds.' Below the error, there is a user input field containing 's_gama', a password field, and a 'Sign In' button. There are also links for 'Remember Me', 'Forgot password?', and 'Forgot username?'. At the bottom, there are links for 'Create an account' and 'Resend activation code'.



Message blocked contact

The screenshot shows a notification dialog box with the title 'Mute notifications' and a toggle switch. The main text reads: 'Block Blocked contacts will no longer be able to call you or send you messages.' Below the text are two buttons: 'CANCEL' and 'BLOCK'.

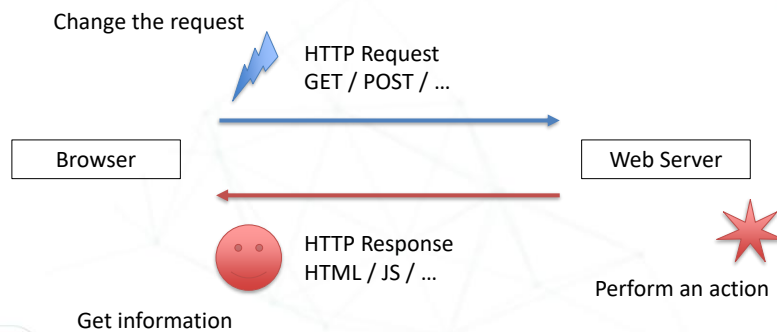
The screenshot shows a contact menu with several options: 'New group', 'New broadcast list', 'Contacts', and 'Settings'. The 'New group' option is highlighted with a red rectangular box.



Testing for business logic

1. Test business logic data validation
2. Test ability to forge requests
3. Test integrity checks
4. Test for process timing
5. Test number of times a function can be used limits
6. Test for the circumvention of workflows
7. Test upload unexpected file types

Web Application



Real examples



23

How to report these bugs?

“Users are not allowed to do that in the browser”

“The users will never do this kind of things”

“The UI prevents these problems”

“Users don’t know how to change the HTTP request”

Solution: Report security problems, not bugs!



24

Security by Design Principles

1. Minimize attack surface area
2. Establish secure defaults
3. Least privilege
4. Defense in depth
5. Fail securely
6. Don't trust services
7. Separation of duties
8. Avoid security by obscurity
9. Keep security simple
10. Fix security issues correctly



25

References

Software Security Book – Gary McGraw

OWASP Testing guide 4.0

https://www.owasp.org/images/5/52/OWASP_Testing_Guide_v4.pdf

OWASP Testing for business logic

https://www.owasp.org/index.php/Testing_for_business_logic

OWASP Security by Design Principles

https://www.owasp.org/index.php/Security_by_Design_Principles

OWASP Abuse Case Cheat Sheet

https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/Abuse_Case_Cheat_Sheet.md



26

