Overview

- Security mindset
- CIA triad
- Secure development lifecycle
- Common attacks on the websites
Why Security

We worry about security when we have something of value and there is a risk it could be harmed.

Security Mindset

- **Threats**: who are the bad actors?
- **Vulnerabilities**: what can possibility they exploit?
- **Risk**: if threats succeed to exploit a vulnerability, what is that attack/risk going to be?
Example: Airport Wi-Fi

![Free WiFi]

Example: vulnerability

![Bicycle]
Example: attack

Software Security
The Threat Landscape

• Beyond 430 million unique pieces of malware exist;
• A new zero-day vulnerability was discovered, on average, once each week;
• Above %75 of all legitimate websites have unpatched vulnerabilities that can be exploited. And, %15 of them are critical vulnerabilities.

Symantec: internet security report 2016

Threats

• **Cybercriminals**: want to profit from our sensitive data for financial gain.

• **Hacktivists**: activists who do not like something you are or something you do. E.g., Edward Snowden.

• **Nation-states**: countries do it for political advantage or for spying.
Vulnerability

The weak points in software that can lead to security concerns.

https://nvd.nist.gov reports different common vulnerabilities

Attack

When threats uncover the vulnerability, conduct research about it, and exploit it to launch their schemes.
The alternatives

• Make threats go away
• Reduce vulnerabilities
  – Confidentiality
  – Integrity
  – Availability

Confidentiality

• Encryption
• Access control (rules and policies)
  – Based on identity, role
• Authentication
  – password, card, finger print
• Authorization
• Physical security
  – Locked windowless rooms, faraday cages
Integrity

• Backup
• Checksum
• Data correcting codes

Availability

• Physical protection
• Computational redundancies
Example

- Confidentiality:
  - encryption of traffic data,
  - time out for invalid inputs,
  - return invalid card, retain stolen card, use of TAN in net-banking
- Integrity:
  - consistency of data during transmission
- Availability:
  - diverse network, fair resource sharing

What should we do

- Detection
- Response
- Recovery
The role of software security

Software security is the property of software, and is an engineering discipline.

It is not only about reactive technologies like firewalls, intrusion detection systems, and antivirus engines.

See the book "Software Security" by Gary McGraw

Security Development Lifecycle

[Diagram showing the Security Development Lifecycle with stages: Training, Requirements, Design, Implementation, Verification, Release, Response]

https://www.microsoft.com/en-us/SDL
Training

All stakeholders should stay informed about security basics and recent trends in the field.

Core secure trainings:
- Threat modeling
- Secure design, coding, and testing
- Privacy

Requirements

- Security requirements
  - Specifying minimum requirements
  - Specifying and deploying a vulnerability tracker
- Quality gates/bug bars
  - Defining criteria for acceptable level of security
- Risk assessment
  - Identifying the functional aspects of the software that requires deep review.
**Design**

- Design requirements
- Attack surface reduction (ASR)
  
  Big Attack Surface = Big Security Work = Big Security Problems

- Threat modeling

---

**Implementation**

- Use approved (e.g. static analysis) tools and libraries
- Deprecate unsafe functions
**Verification**

- Dynamic analysis
- Fuzz testing
- Threat model and ASR review

**Release**

- Incident response plan
  - Sustained engineering resources
  - On-call contacts with decision-making authority
  - Service plans for codes inherited from others
- Final security review (FSR)
  - Passed
  - Passed with exceptions
  - FSR with escalation
- Certifying release and archive
Response

- Execute incident response plan

Web Attacks
SQL Injection

Execution

Consequences:
- Read or write data
- Get privileged access

Malicious code

1

Cross Site Scripting (XSS)

1- Alternatively, attacker may send to the victim a link containing a malicious script e.g. in the URL (known as reflected or non-persistent XSS)

1- Inject malicious script in the website

2- Malicious script executes at each visit

3- Attacker receives private data
The Meaning of Cookie

Information which a website requests or maintains about specific users visiting the website:

- how and when a visit happened,
- authentication information.

Same Origin Policy
Session Hijacking

Steal the session
- MIM attack
- XSS

Impersonate to be "John"

Protecting Cookies

- HttpOnly
  - This flag makes the cookie inaccessible to JavaScript

- Secure flag
  - A cookie will be sent only over a https connection
Cross Site Request Forgery (CSRF)

1. User types a URL into their browser.
2. The browser sends a request to the server.
3. The server sends a response back to the browser.

Free Apple chocolate
Cross Site Request Forgery (CSRF)

If you’re interested in a

– BSc thesis,
– seminar project, or
– part time work on security-related projects

@ghafarii

scg.unibe.ch