Inf2- SEPP Lecture 25: Security and Privacy

Cristina Adriana Alexandru

School of Informatics
University of Edinburgh

In the previous lecture

NFRs

- Reliability
- Availability

This lecture

Security

- Definition and relationship to other non-functional requirements
- Why important?
- Types of security attacks
- Some attacks and defences
- Some defences in detail:
  - Authentication
  - Authorization
  - Encryption
- Design guidelines for security

This lecture

Privacy

- Definition and relationship to security
- Why important?
- Data protection laws
- Privacy policy

Security

Security is *"a system attribute that reflects the ability of the system to protect itself from malicious internal and external attacks."* (Sommerville SE)

It is part of dependability: the ability to provide services that can defensibly be trusted within a time-period, and it is also related to other attributes of dependability:

- *Reliability*
- *Availability*
- *Safety* (ability to operate without catastrophic failure)
- *Resilience* (ability to recover)

Security is also related to *privacy*, but more later . . .

Why is security important?

In the 1990s, the emergence of the Internet led to the proliferation of attacks to online systems.

Effects: unhappy customers, legal action, loss of business revenue, even getting out of business, recovery taking a lot of time and effort.

Human, social and organisational failings can lead to security vulnerabilities (e.g. easy to guess passwords) but . . .

such failings are often a sign of poor system decisions (e.g. frequent password changes), and problems in SE activities!

Types of security attacks

- *Interception*: attacker gaining access to an asset on the system
- *Interruption*: attacker making a system unavailable
- *Modification*: attacker tampering with a system asset
- *Fabrication*: attacker inserting false information into the system

Injection attacks (Type: fabrication)

Attacker imputting malicious code or database commands using an input field, causing damage or the leaking of data to them.

Examples:

- Buffer overflow (C/C++): inputting a string with executable instructions which overwrites memory and transfers control to the malicious code.
- SQL poisoning attacks (systems with SQL database): inputting part of an SQL command to obtain confidential information.

Avoided through: input validation.

Cross-site scripting attacks (Type: fabrication)

Another form of injection attack

Attacker adding malicious JavaScript code to a webpage (e.g. through web form) which - when loaded by a client may:

- Steal user information
- Direct user to another website to capture personal information or display advertisments
- Facilitate hijacking attack (see next slide)

Avoided through: input validation, checking inputs from database before providing results in webpage, HTML 'encode' command to make information non-executable.

Session hijacking attacks (Type: interception)

Attacker stealing a session cookie and using it to impersonate user and carry out user actions (active session hijacking) or follow user to detect valuable information (passive session hijacking).

Can be achieved through:

- Cross-site scripting
- (Server-client) Traffic monitoring

Likelihood reduced through:

- Traffic encryption (https and not http)
- Multi-factor authentication (see below)
- Short timeouts on sessions

Denial-of-service attacks (Type: interruption)

Attacker making system unavailable to users.

Some types:

- Distributed denial-of-service (DDOS) attacks: sending large amount of requests for service to a web application, which denies access to legitimate users. Solution: specialist software.
- User lockout attacks: using policy of locking users out after number of failed attempts and the user's email address used as username to lock user out of account. Some solutions:
  - Temporary lockouts
  - IP address tracking and locking after number of failed attempts

Brute force attacks (Type: interception)

Attacker using information like username and repeatedly generating passwords to try to gain access to system.

Vulnerable if: not locking accounts after failed validation, users choosing weak passwords.

Solutions: locking accounts after failed validation (but may attract denial-of-service attacks), insisting on strong passwords, two-factor authentication (see below).

Some defences in detail

- Authentication
- Authorization
- Encryption

Authentication

Process of making sure a system user is who they claim to be.

Authentication approaches:

- Knowledge-based authentication: providing secret, personal information when logging in, e.g. passwords.
- Possession-based authentication: use of a physical device e.g. mobile phone for authentication
- Attribute-based authentication: using biometric attribute of user e.g. fingerprint, face recognition, retina scan.

Multi-factor authentication uses several approaches from above for increased security.

Authentication via federated identity

"Login with Google", "Login with Facebook" etc., i.e. using external system for authentication.

Advantages for users:

- Single set of credentials
- Credentials stored in a single place (more secure)

But . . . some people have privacy concerns.

Advantages for businesses:

- Not needing to maintain databases with user information.
- Potentially using additional information on the users from the identity provider (IMPORTANT! With users' approval).

Authorization (for multiuser systems)

Using the users' identity to control user access to system resources.

Business systems define user rights through an access control policy. This policy is important:

- From a legal perspective, to conform to data protection rights
- From a technical perspective, can help set up the access control scheme for the system

Most file and database systems use access control lists: tables linking users with resources and actions

Encryption

"Process of making a document unreadable by applying an algorithm transformation" including a secret key to it (Sommerville ESP). Decoding requires reverse transformation.

Practically uncrackable, but things change with new technology.

- Symmetric: same key used for encoding and decoding. Problem: securely sharing encryption key.
- Assymetric: each user having a public and a private key; encrypting using one, decrypting using the other.

Key management essential for businesses. Key management systems used, but keys should also be changed regularly and timestamped.

Design guidelines for security

1. *Base security decisions on an explicit security policy*: high-level statement about the 'what' of security for the organisation (not the 'how').
2. *Use defense in depth*: in critical systems, using a combination of security techniques and not just one.
3. *Fail securely*: when failure occurs, don't use less secure fallback procedures.
4. *Balance security and usability*: security and usability contradictory, and impacting usability may result in users affecting security.
5. *Log user actions*: can help recover from failures, detect attacks and how access was gained, attacker deterrent

Design guidelines for security

6. *Use redundancy and diversity to reduce risk*: more versions of the system or data, not relying on the same platform.
7. *Specify the format of system inputs*
8. *Compartmentalise your assets*: not providing users access to all information in the system by using compartments.
9. *Design for deployment*: providing support for deployment to avoid human mistakes introducing security vulnerabilities.
10. *Design for recovery*: assume failure can always occur, think about recovering system to secure operational state.

Privacy

"Social concept that relates to the collection, dissemination and appropriate use of personal information held by a third party" (Sommerville ESP)

Relationship with security:

- Privacy requires security, but security does not guarantee privacy (e.g. Facebook).
- Not all security attacks breach privacy (e.g. denial-of-service).

Why is privacy important?

- Possibility for legal action from customers or data regulators if not conforming to privacy regulations
- For business products, business customers require privacy guarantees
- Leaking personal information can ruin a company's reputation and lose its customers.

Data protection laws

Many countries have data protection laws limiting the collection, dissemination and use of personal data for the purposes it was collected (e.g. GDPR for companies holding data on EU citizens)

Central notions:

- Data subjects (individuals whose data is affected) having right to: access data, correct errors in it, consent for its use, require its deletion.
- Data controllers (managers of the data), responsible for storing data securely in a location covered by data protection legislation, providing subjects access to it, using it only for consented purpose.

Privacy policy

Legal document outlining how personal and sensitive information is collected, stored, managed.

Should:

- Not be part of long 'terms and conditions' but more accessible to users.
- Be auditable to check that it conforms to the data protectin laws of where software is sold.
- Be reviewable by users; possibility for them to say 'no' about storing different information.

Reading

Essential: Sommerville ESP Chapter 7 on Security and Privacy

Essential: Sommerville SE Chapter 13 on Security Engineering introduction and sections 13.1 and 13.4

Recommended: the rest of the Sommerville SE Chapter 13 on Security Engineering