INF226 – Software Security

Håkon Robbestad Gylterud

2019–08–28

Prepared statements

Without prepared statements

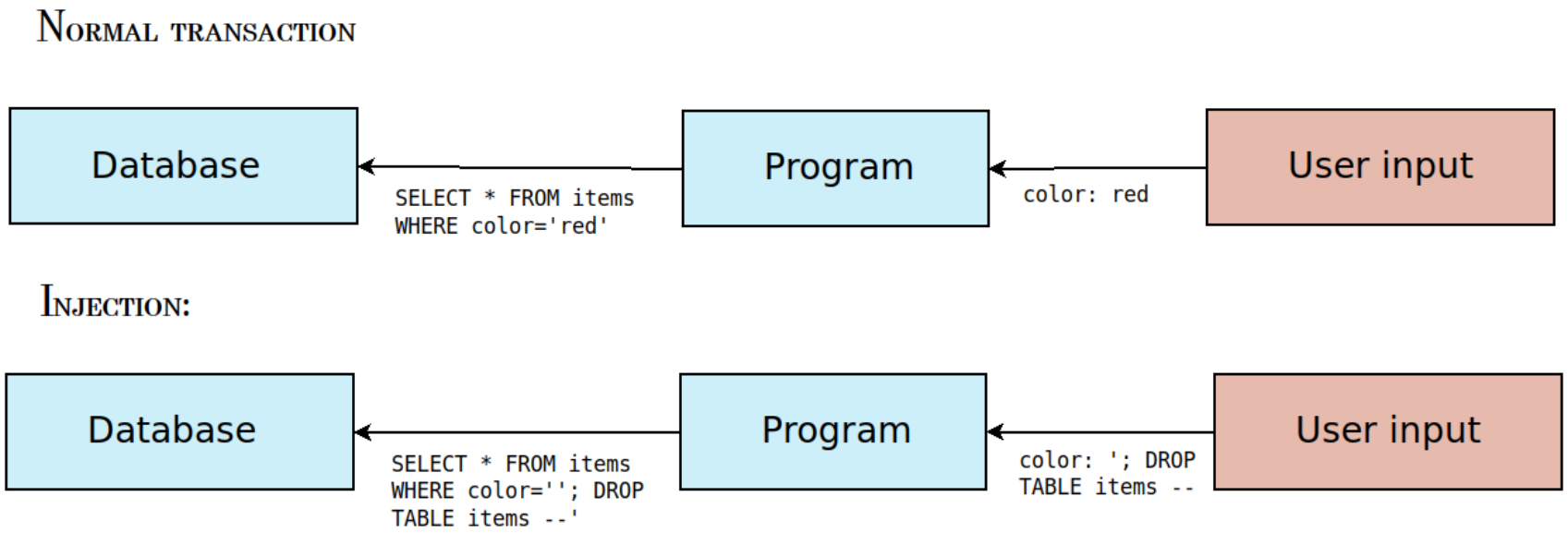


Figure 1: SQL injection

With prepared statements

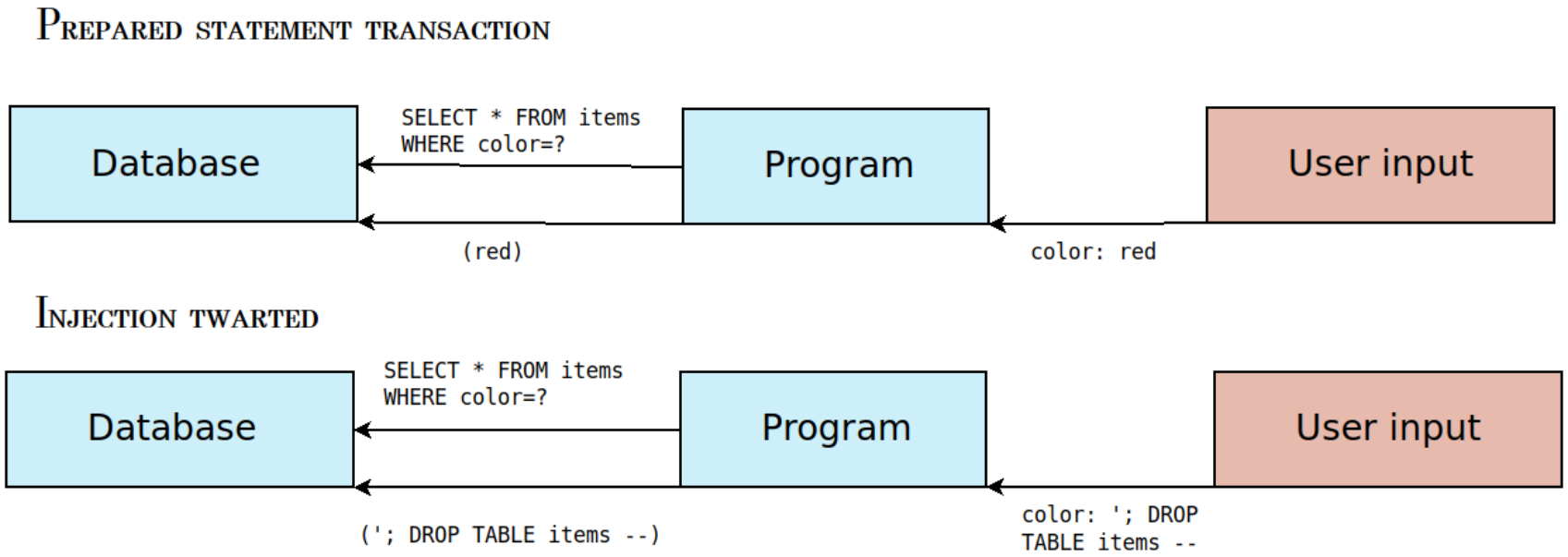


Figure 2: Preparared statement

Threats to software

STRIDE

Praerit Garg and Loren Kohnfelder at Microsoft, has suggested a classification of threats to software, named STRIDE.

- Useful to spark your imagination when developing a threat model.
- List is neither exhaustive, nor orthogonal.

STRIDE

- **S**poofing
- **T**ampering
- **R**epudiation
- **I**nformation disclosure
- **D**enial of service
- **E**levation of priviledge

Spoofing

Wikipedia's definiton:

> *A **spoofing attack** is a situation in which a person or program successfully masquerades as another by falsifying data, to gain an illegitimate advantage.*

SMS sender spoofing and phishing

NRK reported this summer:

> *Posten har i sommer fått daglige henvendelser fra kunder i forbindelse med svindelforsøk. Irene Kalstad Aase fikk SMS om en pakke som var på vei, og opplevde at svindlerne forsøkte å trekke penger fra kontoen hennes.*

URL spoofing

Example:

`https://paypal.com-us.cgi-bin-usrweb.a891u2basdas90···`

URL spoofing

Example: https://www.apple.com/

URL spoofing

Example: https://www.apple.com/

The letters in this address are actually cyrillic Unicode letters. The `L` in the above address is actually `CYRILLIC SMALL LETTER PALOCHKA`

In fact the address is: `https://www.xn--80ak6aa92e.com/`, but (some) browsers will display this as Unicode characters (punycode).

Tampering

Example

Open Wifi networks injecting ads, malware, etc into webpages.

Example

SQL injection to tamper with the database.

Repudiation

repudiate: *to refuse to acknowledge.*

Repudiation

repudiate: *to refuse to acknowledge.*

Example

E-mails can be spoofed. Imagine an untoward e-mail was sent from one politician to another. Does the sender have repudiation?

Information disclosure

Example

Debug info on production systems.

Example

Passwords and other secrets being logged as parts of requests.

Denial of service

1. Flooding the service.
2. Exploiting vulnerability.
3. Hijacking (e.g. DNS).

Elevation of priviledge

Example

Students and school administration work on the same platform. Insecure access control allows students to get adminstration priviledge and change their grades.

STRIDE

- **Spoofing**: Transmissions with intentially mislabeled source.
- **Tampering**: Modification of persistent data or data in transport
- **Repudiation**: Denial of having performed unauthorized operations, in systems where these operations cannot be traced.
- **Information disclosure**: Access to data in an unauthorized fasion.
- **Denial of Service**: Rendering a service unaccessible to intended users.
- **Elevation of priviledge**: Non-priviledged users gaining access to priviledged operations and data.

Exercise

How can a successful buffer overflow exploit be used by an attacker to effect each of these threats?
How about SQL injection?

Ranking threats (DREAD)

- Damage potential
- Reproducibility
- Exploitability
- Affected users
- Discoverability

(What about attacker incentives?)

Functional decomposition and threat model

Rough sketch of a system

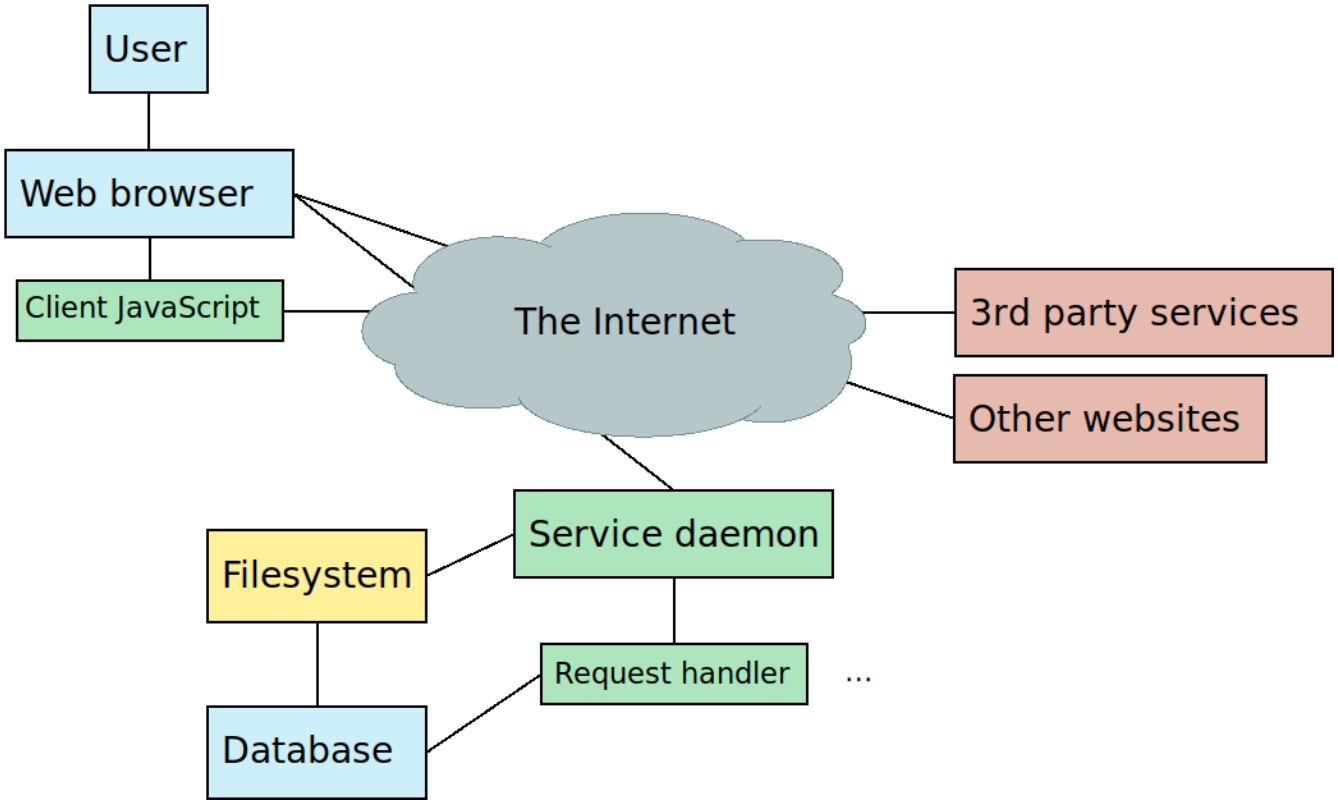


Figure 4: Drawing out the software components of a simple website

Functional decomposition

The functional decomposition of the system contains:

- An overview of components of the system.
- A detailed map of comunication between components.
- Description of the function of each of the components.

Threat model

The threat model explicates our assumptions about the system:

- What threats (STRIDE) applies to each component?
- What are the trust relationships between components?
- Which threats apply to each relationship?

Trust and boundaries

Each service / program / function:

- Takes input from different sources
- Gives output to different destinations

Some of these connections represent security boundaries.

- Receiving: Can I trust the data from this sender?
- Sending: Can I trust the recipient with this data?

Trust

Trust is not binary:

I can trust HTTP requests enough to respond with public information, but not private information.

Trust is not linear:

- I can trust requests from Bob to acces Bob's inbox.
- I can trust requests from Alice to acces Alice's inbox.

Transport security

The first level of security is transport security:

- Secrecy
- Authenticity
- Integrity

We have well established cryptographic mechanisms to achieve this.

Do not roll your own crypto.

Transport security

The only first level of security is transport security:

Transport security

The only first level of security is transport security:

- Example: SQL injections can be sent over HTTPS.

Trust boundary

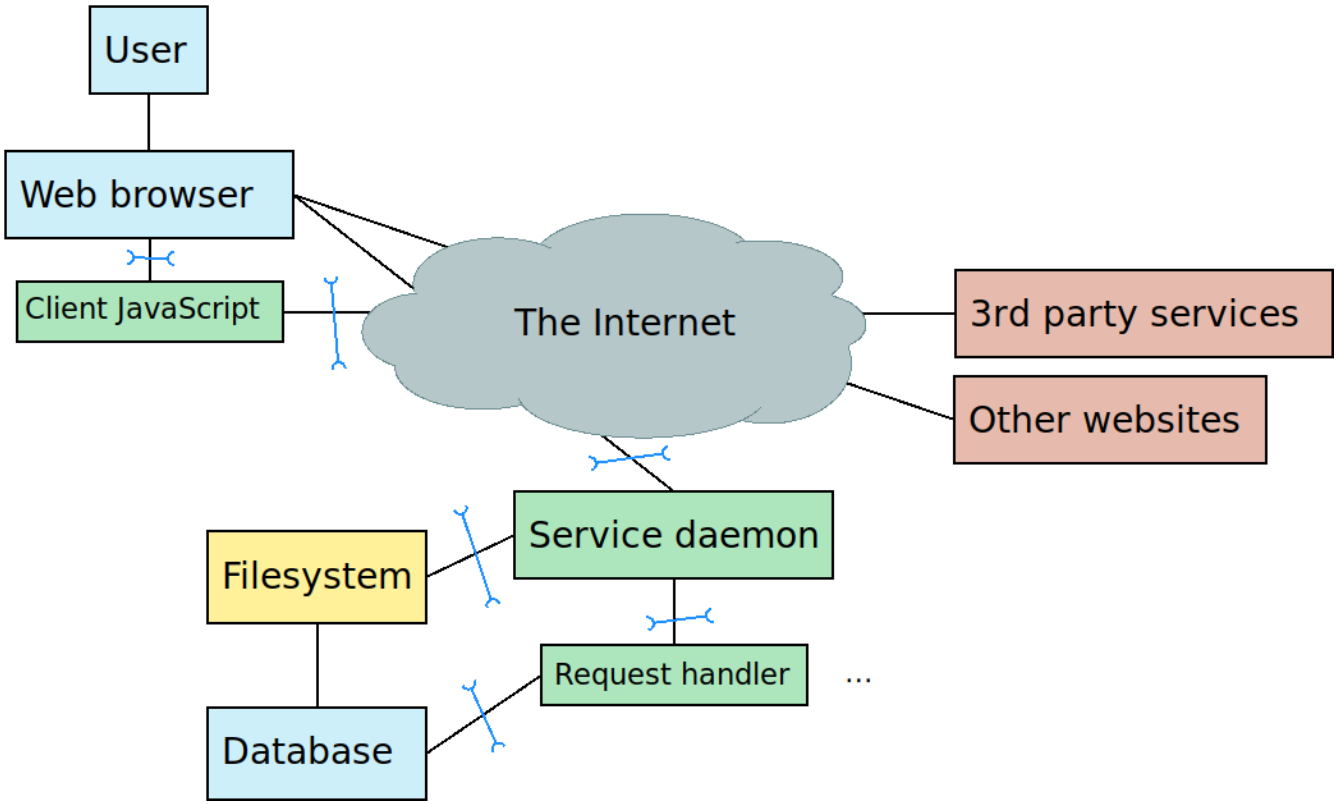
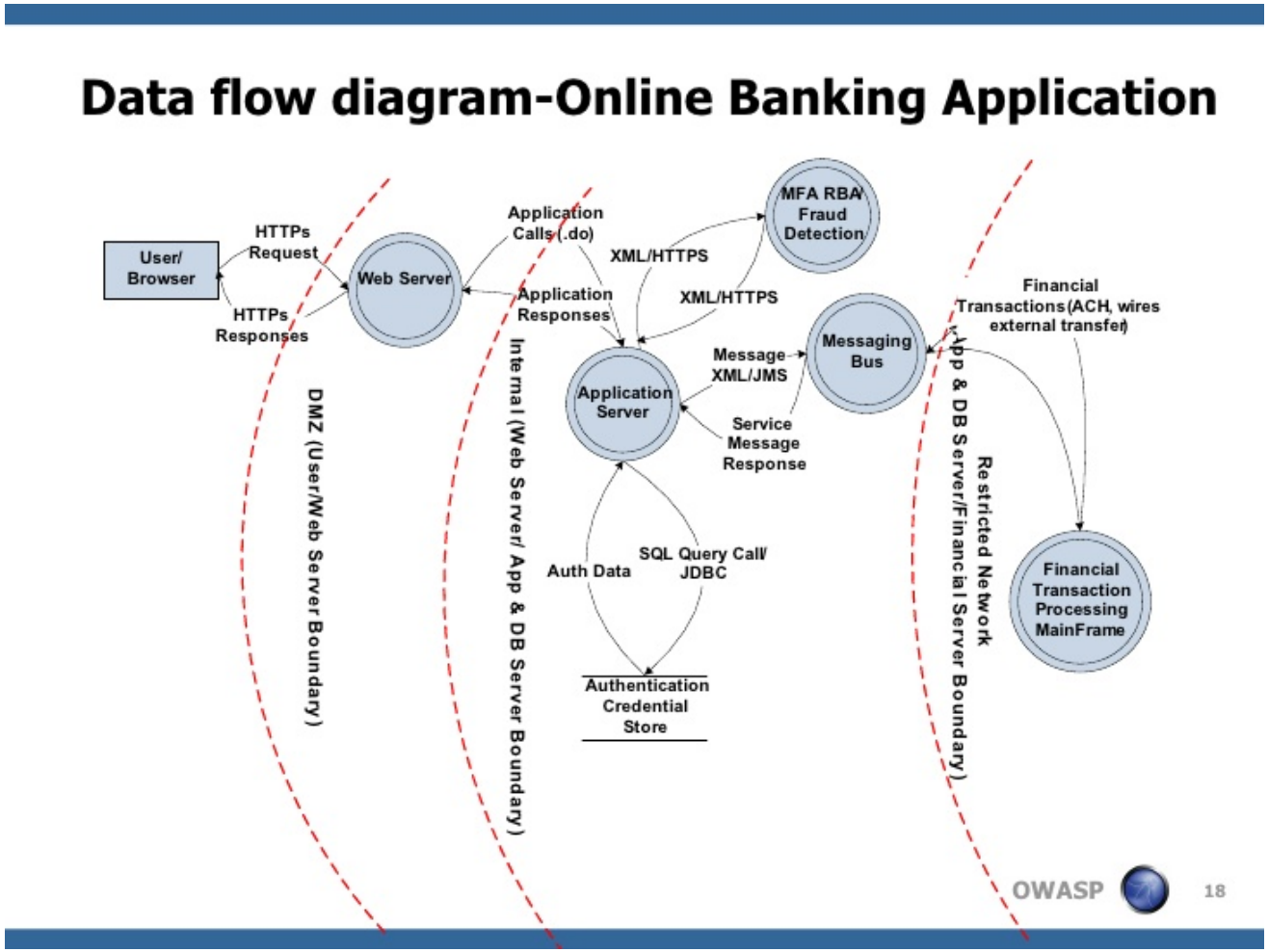


Figure 5: When data is communicated between components with different trust, it crosses a boundary.

OWASP online banking example



Defence in depth

Analyse what happens in case a given security mechanism fails:

- When one mechanism fails other mechanism should:
  - mitigate the failure
  - or at least detect the failure.

Defence in depth

Analyse what happens in case a given security mechanism fails:

- How are other parts affected?
- When one mechanism fails other mechanism should:
  - mitigate the failure
  - or at least detect the failure.

- Are there any *linchpins*?

Next time

Reflections on trusting trust:

- Ken Thompson's acceptance speech for the 1983 Turing Award.
- Ken Thompson and Dennis Ritchie got the prize for their work on UNIX.

Will put out links to reading about this.