Lecture 1

zaterdag 12 november 2022 17:58

web applications are often attacked nowadays

GET → parameter in URL → more easily logged (in particular: in browser history) → do not use for password or other sensitive information
POST → parameter not in URL → somewhat hidden

HTTP is sessionless / stateless → requests are treated independently

↓

connections are not maintained

↓

cookies are stored and sent by the client and contain state (identifier)

↓

can be stolen

SSL secures connection, but not the application

security is difficult / impossible to measure / quantify

SQL injection → string including user input is treated as command

more generally: command injection

cross site scripting (XSS)

↳ a site provides code to client which is provided by an attacker

- reflected → code from URL / get parameter
- stored → code stored by webapp

problem: lack of input sanitization

can be used to steal sessionID

just frame oversite

avoid XSS

- signature-based filter e.g. block <, >, script, <\
- sanitization
- truncating input