SQL Injection: Summary

- Target: web *server* that uses a back-end database
- Attacker goal: inject or modify database commands to either read or alter web-site information
- Attacker tools: ability to send requests to web server (e.g., via an ordinary browser)
- Key trick: web server allows characters in attacker's input to be interpreted as SQL control elements rather than simply as data

CSRF: Summary

- Target: user who has some sort of account on a vulnerable *server* where requests from the user's *browser* to the server have a *predictable structure*
- Attacker goal: make requests to the server via the user's browser that look to server like user *intended* to make them
- Attacker tools: ability to get user to visit a web page under the attacker's control
- Key tricks: (1) requests to web server have *predictable structure*; (2) use of `<IMG SRC=…>` or such to force victim's browser to issue such a (predictable) request
- Notes: (1) do not confuse with Cross-Site Scripting (XSS); (2) attack only requires HTML, no need for Javascript

Stored XSS: Summary

- Target: user with Javascript-enabled *browser* who visits *user-generated-content* page on vulnerable *web service*
- Attacker goal: run script in user's browser with same access as provided to server's regular scripts (subvert SOP = *Same Origin Policy*)
- Attacker tools: ability to leave content on web server page (e.g., via an ordinary browser); optionally, a server used to receive stolen information such as cookies
- Key trick: server fails to ensure that content uploaded to page does not contain embedded scripts
- Notes: (1) do not confuse with Cross-Site Request Forgery (CSRF); (2) requires use of Javascript

Reflected XSS: Summary

- Target: user with Javascript-enabled *browser* who visits a vulnerable *web service* that will include parts of URLs it receives in the web page output it generates
- Attacker goal: run script in user's browser with same access as provided to server's regular scripts (subvert SOP = *Same Origin Policy*)
- Attacker tools: ability to get user to click on a specially-crafted URL; optionally, a server used to receive stolen information such as cookies
- Key trick: server fails to ensure that output it generates does not contain embedded scripts other than its own
- Notes: (1) do not confuse with Cross-Site Request Forgery (CSRF); (2) requires use of Javascript