Vzdrževanje:

- HW je poceni, vzdrževanje je drago; imeti nekoga, ki se spozna, je drago
- ni nujno da z napadi želimo ukrasti neke podatke, ampak lahko je naš cilj samo onemogočiti neke service, da nekomu povzročimo downtime in s tem (denarno) škodo

Ranljivosti z napadi preko klienta:

- kako vem, da je klient res klient:
  - pove nam neke credentiale (geslo, certifikat)
- certifikati:
  - napiše javni ključ, kdo si, kdo ti je dal certifikat, tej CA moramo zaupati
- spletne strani so večinoma dinamične
- ko kliknemo reklame, gremo na neko drugo spletno stran
- kot napadalec lahko damo v reklamo neko svojo kodo, ki se bo izvedla na glavni spletni strani - XSS

XSS:

- ves čas lahko dodajamo nek content na neke spletne strani
- v input lahko damo JS + HTML kodo, ki se bo potem izvedla vsakemu, ki bo to pogledal in ti bo npr. kupilo 10 pajkic
- zaščita:
  - omejujemo, kam ven lahko skačemo, ali zunanji povezavi zaupamo
- tipi:
  - reflected XSS:
    - napadalec pošlje ostalim uporabnikom povezavo do varne strani, ki naloži dodatno kodo (npr. v reklami ali iz query parametra, ki se direktno prikaže na spletni strani)
    - strežnik servira dodatne parametre ostalim uporabnikom
    - uporabnik mora eksplicitno kliknit gor
    - reflected = ker ti napiše na site tisto, kar je narobe sparsano (npr. <b> tag)
    - v query parametru lahko damo <script> tag
    - zaščita: sanitiziramo uporabniški vnos
    - npr. v source od slike damo neko svojo domeno; npr. v src od slike zapremo <img> tag in damo <script> tag naprej
    - lahko ukrademo cookie (cookie = lokalno shranjene informacije, npr. session, add to cart)

        - uporabnik se bo iz svoje mašine povezal do zlonamerne domene, ki ji bo ukradla cookije (session)
        - napadalec mora ditribuirati svoj link; ni hostano na nekem strežniku
    - stored XSS:
        - napadalec shrani uporabniški vnos na strežnik (npr. comment na objavi, profili, forumi)
        - strežnik servira ta uporabniški vnos ostalim uporabnikom

CSRF:

- cross site request forgery
- z domene B delamo poizvedbe na domeno A
- npr. uporabnik gre na facebok.com, ta dela poizvedbe na facebook.com in bo uporabniku izlgledalo čisto legit, hkrati pa bo domena B lahko kradla vnesene podatke, ker je v bistvu MITM
- domena B bi lahko kradla cookieje, kar na srečo ni omogočeno
- CORS forgery:
    - chrome ti reče, da tega ne dovoli
    - na svoji spletni strani povemo, katere domene so dovoljene (tvoje zunanje domene, na katerih je nek API hostan)
    - če dovoliš vse domene (*) imaš lahko problem, ker napadalec z XSS lahko naredi povezavo na svojo domeno
- kako pošiljamo argumente v HTTP request:
    - pri GET v query parametre v URL
    - pri POST v body (JSON)
    - da pridemo do nekega dela API-ja, gremo včasih samo na domena/api/neki1/neki2

Zaščita:

- sanitizacija nizov:
    - textContent atribut - pove browserju, da naj tam ne poriva kode noter
    - npr. `<` se prtvori v `<`
- CORS, da ne pustimo povezav na katerokoli domeno:
    - včasih naredimo, da je API na drugi domeni, ker je to mogoče lažje in potem ostane tako
- HttpOnly piškotki:
    - ne moreš jih brati iz JavaScripta
- Secure piškotki - samo preko https
- SameSite piškotki - samo iz te domene

CSRF žeton:

- za avtentikacijo, da pokažemo, da smo to res mi
- ko gremo na stran in se avtenticiramo, nam bo zgeneriralo CSRF žeton, s katerim se bomo naprej avtenticirali
- če napadalec naredi session hijack (vzame cookie-je), ne bo imel pravilnega tokena in se ne bo mogel pretvarjati, da je jaz

SSRF:

- napad preko proxy strežnika:
  - preko proxy strežnika imamo dostop do vseh zalednih sistemov, če nimamo zero trust modela (ker mislimo, da si v notranji mreži itak vsi zaupamo in ko si enkrat noter imaš lahko dostop do vsega)
  - napad preko absolutizacije relativnih poti: probamo dostopati do /etc/passwd
- zaščite:
  - izogibamo se proxyjem z dinamičnimi poizvedbami (ne daš user inputa direktno v poizvedbo)
  - raznih image-ov ne loadamo direktno iz poti
  - politika omrežnega dostopa: ne želimo pustiti, da se vsi serverji vidijo med sabo na internem omrežju
- ne uporabljati reverse proxyja za vse, problematični postanejo, ko so dinamični
- ne 100% zaupati AWS ali Cloudflare, da te bojo vedno zaščitili
- samo interno dostpona S3 instanca