Ohjelmistokehittäjän tietoturvaopas

Tähän dokumenttiin olen kerännyt ohjelmistokehittäjälle tietoturvan näkökulmasta ohjeita, jotka auttavatohjelmistokehittäjiä tunnistamaan, ehkäisemään ja hallitsemaan yleisiä tietoturvauhkia.

Sisällysluettelo

Perusperiaatteet ja ajattelutapa

Turvallisuusperiaatteet

- **Tietoturva on osa suunnittelua, ei viimeinen lisäys**
  - Tietoturva alkaa vaatimusmäärittelystä
  - Turvallisuusarvioinnit ennen kehitystä
  - Resursoi tietoturvaominaisuudet

- **Kaikkea dataa käsitellään oletusarvoisesti luottamuksellisena**
  - Minimoi käsiteltävä data
  - Rajaa näkyvyys ja pääsy

- **Oletan aina, että järjestelmää yritetään murtaa**
  - Mieti väärinkäyttötilanteet (abuse cases)
  - Suunnittele myös “pahin päivä” (breach mindset)

- **Vähimmän oikeuden periaate (Least Privilege)**
  - Käyttäjät saavat vain tarvitsemansa oikeudet
  - Säännöllinen oikeuksien tarkistus
  - Roolipohjainen pääsynvalvonta (RBAC)

- **Syvä puolustus (Defense in Depth)**
  - Useita turvakerroksia
  - Yhden kerroksen pettäminen ei saa kaataa kaikkea
  - MFA, verkkoeristys, WAF, lokitus, valvonta

- **En luota käyttäjän syötteeseen**
  - Kaikki syötteet validoidaan
  - Kaikki output enkoodataan kontekstiin oikein

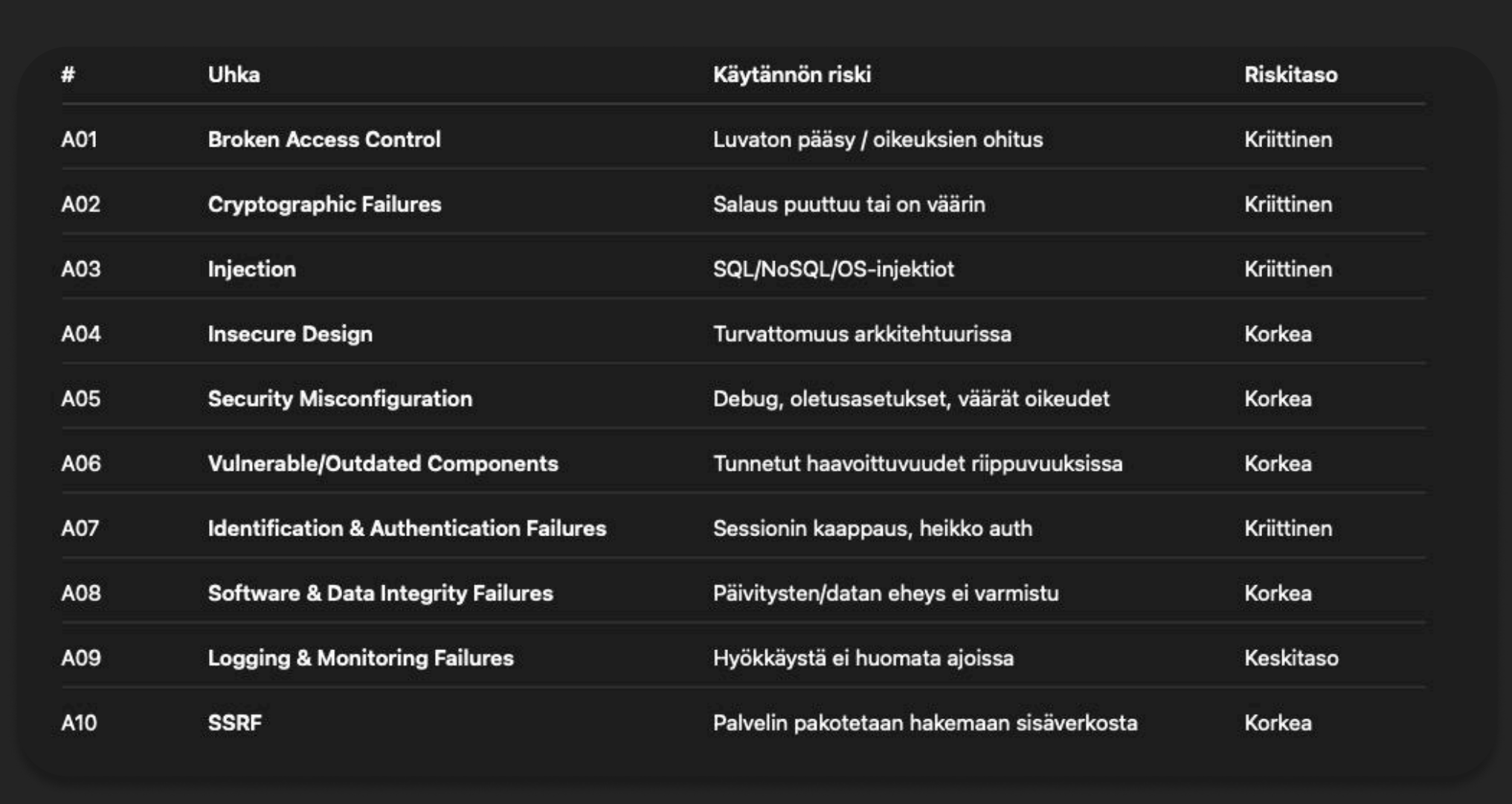
- **Säännöllinen päivitys & patchaus**
  - Riippuvuudet ajan tasalla
  - Turvapäivitykset ASAP
  - Testaa ennen tuotantoon vientiä + rollback

Perusymmärrys

- Ymmärrän mitä web-sovellustietoturva tarkoittaa ja miksi se on olennainen osa ohjelmistokehitystä
- Ymmärrän sisäisen tietoturvan vs ulkoisen tietoturvan eron
- Ymmärrän miksi tietoturva on liiketoimintariski (talous, maine, juridiikka)
- Tunnen keskeiset sääntelyt ja standardit (GDPR, PCI DSS, HIPAA) perusvaatimusten tasolla

Yleisimmät uhat (OWASP-ajattelu)

OWASP Top 10



#	Uhka	Käytännön riski	Riskitaso
A01	**Broken Access Control**	Luvaton pääsy / oikeuksien ohitus	Kriittinen
A02	**Cryptographic Failures**	Salaus puuttuu tai on väärin	Kriittinen
A03	**Injection**	SQL/NoSQL/OS-injektiot	Kriittinen
A04	**Insecure Design**	Turvattomuus arkkitehtuurissa	Korkea
A05	**Security Misconfiguration**	Debug, oletusasetukset, väärät oikeudet	Korkea
A06	**Vulnerable/Outdated Components**	Tunnetut haavoittuvuudet riippuvuuksissa	Korkea
A07	**Identification & Authentication Failures**	Sessionin kaappaus, heikko auth	Kriittinen
A08	**Software & Data Integrity Failures**	Päivitysten/datan eheys ei varmistu	Korkea
A09	**Logging & Monitoring Failures**	Hyökkäystä ei huomata ajoissa	Keskitaso
A10	**SSRF**	Palvelin pakotetaan hakemaan sisäverkosta	Korkea

Yleisimmät uhkat

- Credential stuffing (vuotaneilla tunnuksilla sisään)
- Brute force -hyökkäykset (arvausyritykset)
- SQL Injection (SQLi)
- Cross-Site Scripting (XSS)
- Cookie poisoning / evästeiden manipulointi
- Man-in-the-Middle (MITM)
- Sensitive data exposure
- Insecure deserialization
- DoS / DDoS (saatavuuden kaato)

Ehkäisykeinot

- **Syötteiden validointi**
  - Whitelist-pohjainen validointi
  - Pituusrajat
  - Sallitun merkkijoukon rajaus
  - Tyypin ja formaatin tarkistus

- **Lähdön enkoodaus (Output encoding)**
  - HTML-enkoodaus
  - URL-enkoodaus
  - JavaScript-enkoodaus (kontekstin mukaan)
  - Älä rakenna HTML:ää string-konkatenoinnilla

- **Komentojen ja kyselyiden parametrisointi**
  - Prepared statements
  - Parametrisoidut kyselyt
  - ORM oikein käytettynä

- **HTTPS/TLS**
  - TLS 1.2+ (mieluummin 1.3)
  - HTTP -> HTTPS uudelleenohjaus
  - HSTS tarvittaessa

Secure SDLC

- Tietoturva mukana jo ennen koodausta osana suunnittelua
- Uhkamallinnus ja riskien arviointi tehdään järjestelmällisesti
- Riskit hyväksytään, pienennetään tai dokumentoidaan tietoisesti
- Muutosten vaikutukset tietoturvaan arvioidaan koko elinkaaren ajan

Käytännöllinen kysymys joka viikolle:

“Miten tämän voisi rikkoa?”

Secure Coding

- Syötteet validoidaan sekä client- että server-puolella
- En luota käyttäjän syötteeseen missään tilanteessa
- Hyödynnän frameworkien suojausmekanismeja oikein (mutta en oleta niiden tekevän kaikkea)
- Liiketoimintalogiikka ei ole vain frontendissä
- Käyttöoikeudet tarkistetaan backendissä
- Virheenkäsittely hallittua (ei stack traceja tuotannossa)
- CSRF-suojaus (lomakkeet ja session-pohjainen auth)
- Tiedostolataukset rajattu (koko, tyyppi, sijainti)

Autentikointi & käyttäjät

Salasanahallinta

- **Vahva salasanakäytäntö**
  - Vähintään 12 merkkiä (mielellään 14+)
  - Estä yleisimmät ja vuotaneet salasanat
  - Älä pakota “säännöllistä vaihtoa” ilman syytä (pakota vaihto epäilyssä)

- **Salasanan tallennus**
  - Hash + salt (ja tarvittaessa pepper)
  - bcrypt / PBKDF2 / Argon2
  - Ei MD5, ei SHA-1

- **Salasanan palautus**
  - Resetointi-linkki tokenilla
  - Token vanhenee (esim. 15–30 min)
  - Älä paljasta käyttäjän olemassaoloa
  - Tarvittaessa lisätodennus

Monivaiheinen tunnistautuminen (MFA)

- FIDO2/WebAuthn (paras)
- Authenticator-app (hyvä)
- SMS/email (heikoin, mutta parempi kuin ei mitään)

Sessionien hallinta
- Uniikki session ID
- Session ID vaihtuu loginissa (session fixation -suojaus)
- HttpOnly + Secure + SameSite
- Session timeout / idle timeout

Käyttäjäprofiilien suojaus
- Estä käyttäjä- ja sähköpostienumeraatio
- Samankaltaiset virheilmoitukset (”virhe kirjautumisessa”)
- Rate limiting loginille + lockout-pohdinta

Oikeudet & pääsynhallinta
- Vähimmän oikeuden periaate
- Backend tarkistaa oikeudet aina (ei “frontend piilottaa napin”)
- Admin-toiminnot suojattu erityisen hyvin
- Inaktiiviset ja tarpeettomat käyttäjätilit suljetaan
- Ylläpito-oikeudet tarvittaessa tilapäisiä ja rajattuja

Salaus & avainten hallinta

Datan suojaus

- **Siirrossa oleva data**
  - TLS/HTTPS
  - Sertifikaattien validointi
  - Vahvat salausasetukset

- **Levossa oleva data**
  - AES-256 (yleinen standardi)
  - Arkaluonteiset kentät erikseen salattu
  - Avainten kierto (rotation)
  - Avainten elinkaaren hallinta

- **Eheys ja alkuperä**
  - Digitaalinen allekirjoitus tarvittaessa
  - SHA-256 tai uudempi (eheyden tarkistus)

Avainten hallinta

- Avaimet ja salaisuudet pois koodista (env vars)
- Secrets manager (esim. AWS Secrets Manager tms.)
- Ei hardcodausta
- Ei versionhallintaan
- Pääsynvalvonta ja lokitus salaisuuksille

API-turvallisuus

- **Autentikointi & autorisointi**
  - Tokenit / API-avaimet (riippuen tapauksesta)
  - OAuth 2.0 / OIDC, jos käyttäjälähtöinen integraatio

- **Rate limiting**
  - IP- tai käyttäjäkohtaiset rajat
  - DDoS-suojaus (tarvittaessa)

- **Validointi**
  - Parametrien validointi
  - Pituudet, tyypit, formaatit

- **CORS**
  - Origin-rajaus
  - Metodien rajaus
  - Credential-asetukset oikein

- **SSRF-suojaus**
  - Älä anna käyttäjän määrätä kohde-URL:ää ilman tiukkaa allowlistaa

Data & tietokannat

- Tietokantayhteydet suojattu
- Haetaan vain tarvittavat kentät (data minimization)
- Arkaluonteinen data salattu
- Varmuuskopiointi kunnossa
- Palautuksia testataan (ei vain “otetaan backup”)
- Testidata ≠ tuotantodata

Varmuuskopioiden tavoitteet
- RTO (palautumisaika) määritelty
- RPO (sallitun datan menetyksen määrä) määritelty

Riippuvuudet & kehitysympäristö

Riippuvuuksien hallinta
- Tunnettujen haavoittuvuuksien tarkistus
- Säännöllinen skannaus (SCA)
- SBOM (jos mahdollista)
- Päivitykset hallitusti + rollback

Kolmannen osapuolen koodi
- Lisenssit tarkistettu
- Ylläpidon aktiivisuus
- Yhteisön luotettavuus
- Turvakäytännöt ja CVE-historia

Kehitysympäristö
- OS ja työkalut ajan tasalla
- Palomuuri käytössä
- VPN julkisissa verkoissa
- Tuotantodata ei kehityksessä
- Eri tunnukset eri ympäristöille

Julkaisu & tuotantoympäristö
- HTTPS käytössä
- Oletusasetukset poistettu
- Debug pois tuotannosta
- Portit ja palvelut minimoitu
- Verkon segmentointi tarvittaessa
- Lokitus käytössä (mutta ei arkaluonteista dataa)

Lokitus & monitorointi

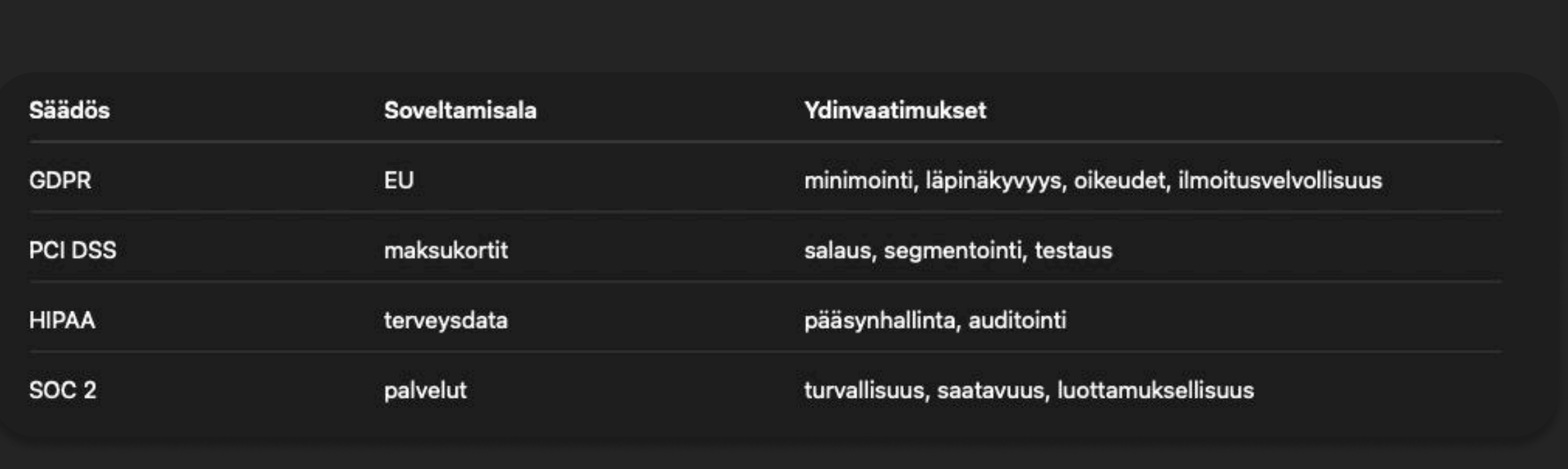
Lokit

- **Merkittävät tapahtumat kirjataan**
  - kirjautumiset, epäonnistumiset, oikeuksien muutokset
  - admin-toiminnot
  - järjestelmävirheet
- Lokien säilytys suunniteltu (esim. 12 kk)
- Lokit suojattu manipuloinnilta
- Hälytykset poikkeamista (anomaly detection)

Monitorointi
- Reaaliaikainen seuranta
- Automaattiset hälytykset
- Brute force ja epänormaalit kirjautumiset tunnistetaan

Tietosuoja & sääntely

Säädökset (yleistaso)



Säädös	Soveltamisala	Ydinvaatimukset
GDPR	EU	minimointi, läpinäkyvyys, oikeudet, ilmoitusvelvollisuus
PCI DSS	maksukortit	salaus, segmentointi, testaus
HIPAA	terveysdata	pääsynhallinta, auditointi
SOC 2	palvelut	turvallisuus, saatavuus, luottamuksellisuus

Tietosuojan toteutus

- Kerätään vain tarpeellinen data
- Käyttötarkoitus määritelty
- Suostumus tarvittaessa
- Poistaminen mahdollista
- Loukkausilmoitusprosessi (esim. 72h) + dokumentointi

Tietoturvapoikkeamiin regointi (Incident response)

- Toimintamalli tietoturvaloukkauksille
- Vastuut ja roolit määritelty
- Viestintämalli valmiina
- Post-mortem analyysi ja parannustoimet

Testaus & dokumentaatio

Turvallisuustestaus

- SAST (staattinen koodianalyysi)
- DAST (dynaaminen testaus)
- SCA (riippuvuuksien skannaus)
- Manuaalinen testaus ja katselmoinnit
- Käyttöoikeusrajat testattu

Dokumentaatio

- Turvallisuusperiaatteet
- Arkkitehtuuri ja tietovirrat
- Riskit ja päätökset
- Poikkeamat ja palautusprosessit

Tarkistuslista ennen tuotantoon siirtämistä

- OWASP-riskit arvioitu
- Riippuvuuksien haavoittuvuudet skannattu
- Koodikatselmointi tehty
- Penetraatiotestaus (tarvittaessa) tehty
- HTTPS/TLS kunnossa
- Salaus siirrossa ja levossa
- Auth + autorisointi kunnossa
- Lokitus ja hälytykset toimivat
- Varmuuskopiot + palautustesti tehty
- Incident response -toimintamalli olemassa
- Tietosuojaseloste ja minimointi kunnossa

Riskinhallinnan prosessi

```
1. Riskien tunnistaminen
   ↓
2. Riskin arviointi (todennäköisyys × vaikutus)
   ↓
3. Mitigoinnin suunnittelu
   ↓
4. Mitigoinnin toteutus
   ↓
5. Seuranta & testaus
   ↓
6. Dokumentaatio & oppiminen
```

Hyödylliset resurssit

- OWASP: https://owasp.org/
- CWE: https://cwe.mitre.org/
- CVE: https://cve.mitre.org/
- NIST Cybersecurity Framework: https://www.nist.gov/cyberframework
- SANS: https://www.sans.org/

Muistiinpanot

- Tietoturva on jatkuva prosessi, ei kertaluontoinen tapahtuma.
- Säännöllinen koulutus ja päivitykset ovat osa tekemistä.
- Turvallisuuteen panostavat yritykset ovat usein kilpailukykyisempiä.
- Turvallisuusongelmista pitää voida puhua avoimesti ilman syyllistämistä.

Tommi Hallila. 2026.