

Ohjelmiston uhkamallinnus – kevyt uhkakysymyslista

Tämä lomake auttaa tunnistamaan ja kirjaamaan ohjelmistoon liittyviä keskeisiä tietoturvauhkia. Tarkoituksena on kuvata mitä voi mennä pieleen, mihin se kohdistuu ja mitä siitä seuraa käytännössä.

Uhkat kirjataan dokumenttiin *5. Uhkat ja riskit -taulukko*.

Miten tätä dokumenttia käytetään?

Käy jäljempänä olevat aihealueet läpi yksi kerrallaan ja keskustele aiheesta apukysymysten avulla. Kun tunnistatte uhkan, kirjatkaa se dokumenttiin *5. Uhkat ja riskit -taulukko* välilehdelle 1. Uhkat (tunnistus):

1. **Mitä voi mennä pieleen / tilanne** → Kuvaa lyhyesti uhka tai ei-toivottu tilanne.
2. **Mihin kohdistuu** → Kerro mihin osaan ohjelmistoa tai tietoon uhka liittyy.
3. **Mitä käytännössä tapahtuu, jos uhka toteutuu** → Kuvaa konkreettinen seuraus.

Täydellisyyttä ei tarvitse tavoitella, vaan tärkeintä on tunnistaa ja kuvata keskeiset riskit mahdollisimman konkreettisesti.

Tässä vaiheessa ei vielä arvioida riskin vakavuutta tai todennäköisyyttä, vaan keskitytään siihen, mitä voi mennä pieleen.

Seuraava vaihe: riskien arviointi

Kun uhkat on tunnistettu ja kirjattu dokumentin *5. Uhkat ja riskit -taulukon* välilehdelle 1. Uhkat (tunnistus), arvioidaan niiden merkitys siirtymällä saman dokumentin välilehdelle 2. Riskit (arviointi), jonne tunnistetut uhkat näkyvät valmiiksi siirtyneinä.

Tässä vaiheessa arvioidaan kunkin uhkan vaikutus ja todennäköisyys, määritetään riskitaso sekä sovitaan tarvittavat toimenpiteet. Arviointi tehdään dokumentin *4. Yksinkertainen ohjelmistoriskien arviointimalli* -ohjeen mukaisesti.

Yleisimmät uhkatoimijat

Erilaisilla toimijoilla voi olla erilaisia tavoitteita, osaamista ja toimintatapoja. Alla olevia esimerkkejä voi käyttää keskustelun tukena, kun pohditaan, ketkä voisivat olla kiinnostuneita järjestelmästä ja millaisia uhkia siitä voisi seurata.

Voitte pohtia esimerkiksi:

- Kuka voisi olla kiinnostunut järjestelmästä tai sen tiedoista?
- Mitä toimija voisi yrittää saavuttaa?
- Miten se voisi yrittää toimia?
- Mitä se yrittäisi tehdä ensimmäisenä?
- Mitä järjestelmän osaa tai tietoa se voisi tavoitella?

Uhkatoimija	Tyypillinen tavoite	Resurssit ja osaaminen	Tyypillinen toimintatapa
Satunnainen hyökkääjä	Kokeilu, uteliaisuus, näkyvyys	Matala	Käyttää valmiita työkaluja ja etsii helposti löydettäviä heikkouksia
Kyberrikollinen	Rahallinen hyöty	Keskitaso–korkea	Haittaohjelmat, tunnusten kalastelu, kiristys, tietojen varastaminen
Sisäinen toimija	Henkilökohtainen hyöty, virhe tai vahinko	Vaihtelee	Käyttää olemassa olevia käyttöoikeuksia
Kilpailija	Liiketoiminnallinen hyöty	Keskitaso	Tiedon kerääminen, vakoilu
Aktivistit tai ideologinen ryhmä	Vaikuttaminen, näkyvyys	Vaihtelee	Palvelun häirintä, tietojen julkaiseminen, mainehaitta
Järjestäytynyt rikollisuus	Taloudellinen hyöty	Korkea	Pitkäkestoinen ja suunnitelmallinen hyökkäys
Valtiollinen toimija	Tiedustelu, vaikuttaminen, pitkäaikainen pääsy	Erittäin korkea	Pitkäkestoinen ja huomaamaton toiminta
Toimittaja tai kumppani	Ei yleensä haitallinen, mutta voi aiheuttaa riskin	Vaihtelee	Virheet, väärät käyttöoikeudet, heikot käytännöt

Uhka-aihealueet ja apukysymykset

1. Pääsy ja identiteetti

Tässä kohdassa tarkastellaan, kuka pääsee järjestelmään ja millä ehdoilla. Tavoitteena on tunnistaa tilanteet, joissa joku voi päästä sisään väärin tai saada liikaa oikeuksia.

- Kuka pääsee järjestelmään ja miten tunnistautuminen tehdään?
- Voiko joku esiintyä toisena käyttäjänä tai palveluna?
- Voiko käyttöoikeuksia korottaa tai kiertää?
- Mitä tapahtuu, jos tunnukset tai API-avaimet vuotavat?
- Onko käyttöoikeudet rajattu minimiin vai annetaanko liikaa oikeuksia?
- Onko eri roolit eroteltu riittävän selkeästi?

2. Tiedon eheys

Tässä tarkastellaan, voiko järjestelmässä olevaa tietoa muuttaa luvatta tai virheellisesti. Tavoitteena on varmistaa, että tieto pysyy oikeana ja luotettavana.

- Voiko käyttäjä tai järjestelmä muuttaa tietoa luvatta?
- Onko syötteiden validointi kunnossa?
- Voiko integraatio syöttää virheellistä tai haitallista dataa?

lista dataa?

- Voiko tietoa muuttaa järjestelmän sisällä väärin?
- Voiko dataa poistua tai korruptoitua vahingossa?
- Luotetaanko johonkin tietolähteeseen liikaa ilman tarkistuksia?
- Voiko virhetilanne paljastaa tietoa tai rikkoa logiikkaa?

3. Tiedon luottamuksellisuus

Tässä arvioidaan, voiko tieto päätyä väärin käsiin. Erityisesti tarkastellaan arkaluonteista ja liiketoiminnan kannalta kriittistä dataa.

- Voiko arkaluonteinen tieto vuotaa ulkopuolisille?
- Onko data suojattu siirrossa ja tallennuksessa?
- Näkykö liikaa tietoa rajapinnoissa tai virheilmoituksissa?
- Vuotaako tietoa lokitietoihin?
- Onko käyttöliittymässä näkyvissä tietoa, jota käyttäjän ei pitäisi nähdä?
- Voiko tietoa yhdistämällä päätellä jotain arkaluonteista?

4. Saatavuus

Tässä tarkastellaan, pysyykö järjestelmä käytävissä ja kestääkö se häiriöitä. Tavoitteena on tunnistaa tilanteet, joissa palvelu estyy tai hidastuu.

tuu merkittävästi.

- Mitä tapahtuu, jos palvelu kaatuu?
- Voiko järjestelmää kuormittaa tarkoituksella?
- Onko yksittäisiä kriittisiä riippuvuuksia?
- Onko varmistukset ja palautus mietitty?
- Kuinka nopeasti palvelu voidaan palauttaa?
- Voiko jokin ulkoinen palvelu estää toiminnan?

5. Lokitus ja jäljitettävyys

Tässä varmistetaan, että tapahtumista jää riittävä jälki. Tavoitteena on, että poikkeamat voidaan havaita ja selvittää jälkikäteen.

- Tallennetaanko kriittiset tapahtumat lokiin?
- Voiko joku toimia ilman, että siitä jää jälkeä?
- Voiko lokitietoa muuttaa tai poistaa?
- Riittävätkö lokit tapahtumien selvittämiseen?
- Seurataanko epäilyttävää toimintaa aktiivisesti?
- Onko hälytykset määritelty?
- Voiko käyttäjä kiistää tehneensä toimenpiteet?
- Onko tapahtumat sidottu yksilöitävään käyttäjään luotettavasti?

6. Integraatiot ja riippuvuudet

Tässä tarkastellaan ulkoisia yhteyksiä ja riippuvuuksia. Tavoitteena on tunnistaa riskit, jotka syntyvät järjestelmän ulkopuolelta.

- Mihin ulkoisiin palveluihin ollaan riippuvaisia?
- Voiko integraatio syöttää haitallista dataa?
- Onko selvää, missä kohtaa luotetaan ulkopuoliseen järjestelmään ilman varmistusta?
- Mitä tapahtuu, jos integraatio ei toimi?
- Onko rajapinnat suojattu riittävästi?
- Voiko integraation kautta päästä sisään järjestelmään?
- Onko järjestelmässä taustaprosesseja tai ajastettuja toimintoja, joilla on laajat oikeudet? Voiko niitä käyttää väärin tai ohittaa?

7. Väärinkäyttö ja poikkeavat tilanteet

Tässä mietitään, miten järjestelmää voisi käyttää väärin. Tavoitteena on tunnistaa hyökkääjän näkökulmasta mahdolliset väärinkäytöt.

- Miten järjestelmää voisi käyttää väärin?
- Voiko käyttäjä kiertää liiketoimintalogiikkaa?
- Voiko rajoituksia (esim. määrät, nopeus) kiertää?
- Voiko toimintoja käyttää eri tarkoitukseen kuin on suunniteltu?
- Mitä hyökkääjä yrittäisi tehdä ensimmäisenä?
- Mikä on pahin realistinen väärinkäyttöskenaariorio?