Lab 2 – STRIDE Enumeration

Onderdeel van: Risico Analyse
Locatie in GitLab: `labs/risico-analyse/stride-enumeration/`

Leeruitkomst: LO2 – *Adviseren*
Tijdsinschatting: 3 uur
Benodigdheden:

- Je eigen DFD (gemaakt in Lab 1 – System Modeling)
- Documentatie over STRIDE, HAZOP en PASTA
- Hulpmiddelen voor schema's of tabellen (bijv. Draw.io, Lucidchart, of markdown-tabellen)

Doel

In dit lab leer je bedreigingen systematisch identificeren met behulp van STRIDE. Je oefent met het categoriseren van dreigingen en koppelt ze aan dataflows in je eigen DFD.

Voorbereiding

- Bestudeer de basis van **STRIDE, HAZOP en PASTA** threat modeling frameworks.
- Neem je eigen DFD uit Lab 1 bij de hand.
- Open in GitLab de map:

Opdrachtstappen

Stap 1 – Basiskennis

1. Leg in je eigen woorden uit wat **STRIDE, HAZOP en PASTA** zijn.
    - Beschrijf de **voordelen en nadelen** van elk framework.
    - Schrijf voldoende detail: minimaal **200 woorden per framework**.

Stap 2 – Dreigingen categoriseren

Voor de volgende scenario's:

1. Authenticatiecookie kan worden vervalst
2. API-toegang is onbeveiligd
3. Kernel-level system calls zijn toegankelijk vanaf user-space processen
4. Files op de logserver zijn te bewerken
5. Onbeperkte SQL-injectie is mogelijk via de website

- Plaats elke dreiging in de **meest passende STRIDE-categorie**.
- Beschrijf je keuze en motiveer.
- Geef daarnaast ook **twee alternatieve STRIDE-categorieën** die je zou kunnen kiezen en leg uit waarom.

Stap 3 – Uitvoeren STRIDE op je eigen DFD

1. Maak een overzicht van alle dataflows in je DFD (Lab 1).
2. Schrijf per dataflow in één korte zin welke data erdoorheen gaat.
3. Ga per dataflow alle STRIDE-categorieën langs en geef aan of daar een dreiging kan optreden.
4. Beschrijf kort de mogelijke aanval per dataflow + categorie (alleen beschrijven, **nog niet** inschalen op ernst/impact).
5. Gebruik hierbij tabellen of schema's indien dat overzichtelijker is.

Inleveren

1. Werk je antwoorden direct in **dit bestand** uit, onder het kopje **Antwoorden student**.
2. Voeg eventuele tabellen, schema's, screenshots of afbeeldingen toe in dit bestand, zodat ze zichtbaar zijn in de PDF.
3. Lukt het niet om schema's of afbeeldingen zichtbaar te maken in dit bestand, lever ze dan ook los in via de DLO. Verwijs er in de PDF wél naar (bijvoorbeeld: *"zie bestand stride-tabel.png in de DLO"*).
4. Converteer dit bestand naar **PDF** met het script `pandoc_convert.py`.
5. Lever in de DLO in:
   - de **PDF** (met alle antwoorden, tabellen, screenshots en afbeeldingen erin)
   - eventuele **losse bestanden** die onderdeel zijn van de opdracht (bijvoorbeeld originele schema's of afbeeldingen).

Checklist

- ☑ Frameworks STRIDE, HAZOP en PASTA uitgelegd (minimaal 200 woorden per framework)
- ☑ Alle 5 voorbeeld-dreigingen gecategoriseerd (met 2 alternatieven per stuk)
- ☑ Overzicht dataflows gemaakt
- ☑ Dataflows beschreven met korte zin per flow
- ☑ STRIDE-categorieën toegepast per flow
- ☑ Mogelijke aanvallen beschreven
- ☑ Tabell(en), schema's of screenshots toegevoegd
- ☑ PDF-export ingeleverd via DLO

Toetsing

- **Eis**: Je maakt **100% van de labs**.
- **Beoordeling**: Minimaal **80% moet op niveau** zijn.
- **Niet goed?** > Je krijgt feedback en mag onderdelen herstellen.
- **Op tijd ingeleverd?** > Ja/nee (harde deadline).

Antwoorden student (vul hier je werk in)

feedback ontvangen & mijn verbetering

Ik heb stap 2 verbeterd door alternatieven toe te voegen aan Dreigingen categoriseren

gekregen feedback

Alternatieve categorieën ontbreken of zijn niet beargumenteerd. Vul bij elk scenario minimaal twee alternatieven in met een korte uitleg waarom die ook te verdedigen zijn.

verbeteringen stap 2

1. Authenticatiecookie kan worden vervalst: **Spoofing** door het vervalsen van een Authenticatie cookie doe je je voor als iemand anders. alternatief: **Tampering** de inhoud of MAC adress van de cookie wordt aangepast. dat raakt de integriteit het authenticatie mechanisme. alternatief: **Elevation of Privilege** als de cookie een rol of scope bevat, dan kan er een vervalst admin prodiel hogere rachten geven of toestaan.

2. API-toegang is onbeveiligd: **Information disclosure** als je de API niet bevijligd lek je alle informatie op straat voor indringers. alternatief: **Spoofing** als de indringer toegang krijgt in de API kan hij zich voordoen als een Gebruiker. alternatief: **Tampering** onbeveiligde endpoinnts kunnen configuraties of gegevens aanpassen ongeautoriseerde.

3. Kernel-level system calls zijn toegankelijk vanaf user-space processen: **Elevation of Privilege** als een indringer bij alle admin level commands kan is dit een directe threat en heeft hij hogere Privileges dan mag.
   alternatief: **Tampering** als een hacker de privileged calls misbruikt kan de kernel structuur of beveiligings instellingen worden gewijzigd. Alternatief: **Denial of Service** Door de kernel te overspoelen met zware of foute calls wordt het systeem onbruikbaar.

4. Files op de logserver zijn te bewerken: **Tampering** als de intruder toegang heeft om files te bewerken** of aanpassen telt dit als tampering. Alternatief: **Repudiation** Met gemanipuleerde of gewiste logs kan een dader handelingen ontkennen omdat het bewijs ontbreekt. Alternatief: **Information Disclosure** Logs bevatten vaak tokens, foutmeldingen en persoonlijke informatie die bij leaks kunnen lekken.

5. Onbeperkte SQL-injectie is mogelijk via de website: **Tampering** Door een injectie kan je tables in een datababase bekijken, bewerken of aanpassen. Alternatief: **Information Disclosure** union technieken kunnen de compleete tabellen uitlezen. Alternatief: **Elevation of Privilege** Injectie kan sessies stelen of rechtenkolommen aanpassen en zo hogere applicatierechten geven.

Stap 1 – Basiskennis

STRIDE, HAZOP en PASTA

Dit document beschrijft drie bekende methoden voor het identificeren en analyseren van risico's: **STRIDE**, **HAZOP** en **PASTA**. Voor elk framework geef ik een begrijpelijke uitleg, gevolgd door voordelen en nadelen. Alle tekst is in eigen woorden geschreven en geschikt om als onderdeel van een rapport of studieopdracht te gebruiken.

STRIDE

Wat is STRIDE?
STRIDE is een threat-modelingkader dat systemen systematisch bekijkt aan de hand van zes dreigingscategorieën: **Spoofing** (zich voordoen als iemand anders), **Tampering** (gegevens of processen wijzigen), **Repudiation** (handelingen ontkennen), **Information Disclosure** (onbedoelde of kwaadaardige informatielekken), **Denial of Service** (dienstonderbreking) en **Elevation of Privilege** (verkrijgen van ongepaste rechten). Het wordt veel gebruikt binnen softwareontwikkeling en systeemontwerp om vroeg in het proces mogelijke beveiligingsproblemen aan te wijzen.

Waarom je STRIDE zou gebruiken
STRIDE dwingt ontwikkelteams om op een gestructureerde manier na te denken over bedreigingen: elke component of interactie wordt langs de zes categorieën gelegd. Door die systematiek is het makkelijk om ontbrekende controles of zwakke punten te herkennen en prioriteiten te stellen. STRIDE werkt goed in combinatie met architectuurdiagrammen en use-case analyses, en helpt bij het formuleren van concrete mitigerende maatregelen (bijv. authenticatie, integriteitschecks, logging, encryptie).

Voordelen

- **Breed toepasbaar:** kan ingezet worden in veel soorten IT-projecten en sectoren.
- **Helpt prioriteren:** biedt een duidelijk kader om eerst de meest kritieke dreigingen aan te pakken.
- **Praktisch voor ontwerp:** geeft direct aanknopingspunten voor technische controles en ontwikkelkeuzes.
- **Vroegtijdige detectie:** door vroeg toe te passen voorkom je dure aanpassingen later in de levenscyclus.

Nadelen

- **Tijdsinvestering:** een grondige STRIDE-analyse kost tijd, zeker bij complexe systemen.
- **Statisch karakter:** richt zich voornamelijk op ontwerp en architectuur en kan dynamische runtime-dreigingen missen.
- **Subjectiviteit:** resultaten hangen af van de expertise van de deelnemers verschillende teams kunnen tot andere inschattingen komen.
- **Beperkt buiten IT:** STRIDE is minder geschikt voor fysieke risico's of puur menselijke/sociale dreigingen zoals social engineering.

HAZOP (Hazard and Operability Study)

Wat is HAZOP?
HAZOP is een systematische, methode om veiligheidsrisico's en bedieningsproblemen in (proces)systemen te identificeren. Het wordt traditioneel veel toegepast in procesindustrieën (zoals chemie en farmacie), maar kan ook op andere technische processen worden toegepast. Een HAZOP-sessie bekijkt processtappen één voor één en gebruikt sturende woorden (bijv. "geen", "meer", "minder", "omgekeerd") om afwijkingen en hun oorzaken en gevolgen te analyseren.

Waarom HAZOP gebruiken
Het krachtigste element van HAZOP is de gestructureerde samenwerking: verschillende experts — operators, engineers, veiligheidsspecialisten — analyseren samen en brengen zo praktische kennis en ervaring samen. Door het stapsgewijze karakter komen zowel voor de hand liggende als subtielere

afwijkingen aan het licht. HAZOP genereert vaak concrete aanbevelingen voor procesbeheersing, beveiliging en operationele procedures.

Voordelen

- **Diepgang en systematiek:** HAZOP onderzoekt processen grondig en produceert gedocumenteerde bevindingen.
- **Multidisciplinair:** combineert operationele praktijk met technische expertise, wat de kwaliteit van de analyse verhoogt.
- **Direct toepasbaar:** aanbevelingen zijn vaak operationeel van aard (bedieningsinstructies, alarms, redundant ontwerp).
- **Bewezen effectief:** in veel zware industrieën heeft HAZOP aantoonbaar bijgedragen aan veiliger werken.

Nadelen

- **Resource-intensief:** HAZOP-studies vereisen tijd, voorbereiding en ervaren facilitators kosten kunnen hoog zijn.
- **Complexiteit bij grote systemen:** voor uitgebreide installaties ontstaan lange lijsten met bevindingen die prioritering vereisen.
- **Beperkt cyber-scope:** de methode is primair proces- en fysiek gericht moderne IT-dreigingen vragen aanvullende analyses.
- **Resultaatafhandeling:** zonder goede governance blijven aanbevelingen soms onuitgevoerd.

PASTA (Process for Attack Simulation and Threat Analysis)

Wat is PASTA?
PASTA is een risicogedreven en attacker-gerichte methodologie voor threat modeling. Het bestaat uit zeven stappen: bepalen van bedrijfsdoelstellingen, afbakenen van technische scope, ontleden van applicaties, analyseren van bedreigingen, beoordelen van kwetsbaarheden, modelleren van aanvalsscenario's en tenslotte risico- en impactanalyse. PASTA richt zich erop om dreigingen vanuit het perspectief van een aanvaller te simuleren en koppelt technische bevindingen aan zakelijke impact.

Waarom PASTA gebruiken
PASTA legt expliciet verbinding tussen bedrijfsrisico's en technische kwetsbaarheden, waardoor beslissingen over mitigatie goed te onderbouwen zijn. Door stakeholders uit verschillende disciplines te betrekken (security, development, business) ontstaat een gedeeld begrip van wat belangrijk is voor de organisatie. De methode is schaalbaar: je kunt het proces aanpassen aan de omvang van het project, en de nadruk op aanvalssimulatie levert realistische scenario's die prioritering eenvoudiger maken.

Voordelen

- **Risicogedreven:** focust op dreigingen met de grootste zakelijke impact, niet alleen op technische zwakheden.
- **Stakeholder-alignment:** verbindt technische teams en management waardoor maatregelen bedrijfskritisch worden ingebed.
- **Realistische scenario's:** attacker-georiënteerde modellering maakt het eenvoudiger om concrete mitigaties te kiezen.

- **Schaalbaar en aanpasbaar:** geschikt voor zowel kleine als grote projecten en kan modulair worden toegepast.

Nadelen

- **Complex en arbeidsintensief:** full PASTA doorlopen vraagt veel tijd en ervaring, wat een drempel vormt voor kleinere organisaties.
- **Afstemming nodig:** het integreren van niet-technische stakeholders vergt inspanning en goede communicatie zonder die afstemming verliest de methode effectiviteit.
- **Niet altijd snel toepasbaar:** wanneer snelle, praktische resultaten nodig zijn, kan PASTA te uitgebreid zijn.
- **Afhankelijk van data:** om realistische attack-simulaties te doen zijn vaak gedetailleerde systeem- en dreigingsgegevens nodig.

Conclusie

STRIDE is praktisch en overzichtelijk voor softwaregerichte dreigingen HAZOP is diepgaand en operationeel voor procesveiligheid PASTA is risicogedreven en zakelijk verbonden, geschikt voor organisaties die dreigingen willen kwantificeren vanuit het perspectief van de aanvaller. In moderne omgevingen is het vaak verstandig elementen van meerdere methoden te combineren om zowel technische als operationele risico's en de bijbehorende zakelijke impact te adresseren.

Stap 2 – Dreigingen categoriseren

1. Authenticatiecookie kan worden vervalst: **Spoofing** door het vervalsen van een Authenticatie cookie doe je je voor als iemand anders. alternatief: **Tampering** de inhoud of MAC adress van de cookie wordt aangepast. dat raakt de integriteit het authenticatie mechanisme. alternatief: **Elevation of Privilege** als de cookie een rol of scope bevat, dan kan er een vervalst admin prodiel hogere rachten geven of toestaan.

2. API-toegang is onbeveiligd: **Information disclosure** als je de API niet bevijligd lek je alle informatie op straat voor indringers. alternatief: **Spoofing** als de indringer toegang krijgt in de API kan hij zich voordoen als een Gebruiker. alternatief: **Tampering** onbeveiligde endpoinnts kunnen configuraties of gegevens aanpassen ongeautoriseerde.

3. Kernel-level system calls zijn toegankelijk vanaf user-space processen: **Elevation of Privilege** als een indringer bij alle admin level commands kan is dit een directe threat en heeft hij hogere Privileges dan mag.
   alternatief: **Tampering** als een hacker de privileged calls misbruikt kan de kernel structuur of beveiligings instellingen worden gewijzigd. Alternatief: **Denial of Service** Door de kernel te overspoelen met zware of foute calls wordt het systeem onbruikbaar.

4. Files op de logserver zijn te bewerken: **Tampering** als de intruder toegang heeft om files te bewerken** of aanpassen telt dit als tampering. Alternatief: **Repudiation** Met gemanipuleerde of gewiste logs kan een dader handelingen ontkennen omdat het bewijs ontbreekt. Alternatief: **Information Disclosure** Logs bevatten vaak tokens, foutmeldingen en persoonlijke informatie die bij leaks kunnen lekken.

5. Onbeperkte SQL-injectie is mogelijk via de website: **Tampering** Door een injectie kan je tables in een datababase bekijken, bewerken of aanpassen. Alternatief: **Information Disclosure** union technieken kunnen de compleete tabellen uitlezen. Alternatief: **Elevation of Privilege** Injectie kan sessies stelen of rechtenkolommen aanpassen en zo hogere applicatierechten geven.

Stap 3 – STRIDE op eigen DFD

1. Overzicht dataflows + korte omschrijving

Nr.	Dataflow	Omschrijving
1	Order (Customer > Receive orders)	Klant stuurt bestelling door naar het systeem.
2	Invalid order (Receive orders > Customer)	Klant krijgt bericht terug als de bestelling ongeldig is.
3	Billing information (Receive orders > Invoice)	Factuurgegevens van de bestelling worden verstuurd.
4	Invoice (Invoice > Customer)	Klant ontvangt de factuur.
5	Bank details (Invoice > Payment)	Betaalgegevens worden doorgestuurd naar het betaalsysteem.
6	Payment status (Payment provider > Payment)	Bevestiging of betaling gelukt is.
7	Payment confirmation (Payment > Payment provider)	Betaalopdracht of bevestiging richting de betaalprovider.
8	Customer details (Receive orders > Customers)	Persoonsgegevens van klant (naam, adres) worden opgeslagen.
9	Order details (Receive orders > Orders)	Bestelgegevens worden doorgestuurd naar de orderdatabase.
10	Order details (Orders > Warehouse)	Magazijn krijgt orderdetails voor verwerking.
11	Shipping details (Orders > Shipping)	Verzending ontvangt verzendinformatie.
12	Package + delivery date (Courier <-> Customer)	Pakket en leverdatum tussen klant en koerier.
13	Shipping details (Shipping > Courier)	Verzendinformatie wordt naar de koerier gestuurd.
14	Package handover (Warehouse > Shipping)	Magazijn draagt pakket over aan verzendafdeling.

2. STRIDE-analyse per dataflow

Legenda: S = Spoofing, T = Tampering, R = Repudiation, I = Information Disclosure, D = Denial of Service, E = Elevation of Privilege
(ja = dreiging mogelijk, nee = niet waarschijnlijk)

Dataflow	S	T	R	I	D	E	Voorbeeld aanval
Order (Customer > Receive orders)	ja	ja	ja	ja	ja	ja	Klant stuurt gemanipuleerde order (tampering), of doet zich voor als andere klant (spoofing).
Invalid order (Receive orders > Customer)	nee	ja	ja	ja	nee	nee	Foutmeldingen of responsen manipuleren om systeeminformatie te achterhalen.
Billing information (Receive orders > Invoice)	nee	ja	nee	ja	nee	nee	Factuurdata aanpassen of onderscheppen om klantgegevens te stelen.
Invoice (Invoice > Customer)	ja	ja	ja	ja	nee	nee	Gemanipuleerde factuur die gebruiker naar een nep-betaalpagina stuurt.
Bank details (Invoice > Payment)	ja	ja	nee	ja	nee	ja	Man-in-the-middle steelt bankgegevens aanvaller verkrijgt toegang tot betalingssysteem.
Payment status (Provider > Payment)	ja	ja	ja	ja	ja	ja	Nepstatus terugsturen (spoofing), waardoor het systeem onterecht bestelling verwerkt.
Payment confirmation (Payment > Provider)	ja	ja	ja	ja	nee	ja	Nepbetalingen of gewijzigde bevestigingen doorsturen.
Customer details (Receive orders > Customers)	ja	ja	nee	ja	nee	ja	Persoonsgegevens lekken of worden gewijzigd (tampering / information disclosure).
Order details (Receive orders > Orders)	ja	ja	ja	ja	nee	ja	Order aanpassen waardoor verkeerde producten worden geleverd.
Order details (Orders > Warehouse)	nee	ja	nee	ja	nee	nee	Onderscheppen of aanpassen van ordergegevens verkeerde zendingen.
Shipping details (Orders > Shipping)	nee	ja	nee	ja	nee	nee	Manipulatie van verzendinformatie, pakket naar fout adres.
Package + delivery date (Courier <-> Customer)	ja	ja	nee	ja	ja	nee	Aanvaller doet zich voor als koerier en onderschept pakket of vertraagt levering (DoS).
Shipping details (Shipping > Courier)	nee	ja	nee	ja	nee	nee	Verzendlabel aanpassen zodat pakket verkeerd bezorgd wordt.

Dataflow	S	T	R	I	D	E	Voorbeeld aanval
Package handover (Warehouse > Shipping)	nee	ja	nee	nee	ja	nee	Pakket wordt fysiek onderschept of overdracht wordt verhinderd (Denial of Service/diefstal).

Bronnen

Allen-Addy, C. (2025, 6 maart). Threat Modeling Methodology: STRIDE. Threat Modeling Methodology: STRIDE. https://www.iriusrisk.com/resources-blog/threat-modeling-methodology-stride

3. HAZOP-methode (Hazard and operability). (n.d.). Kennisportaal Klimaatadaptatie. https://klimaatadaptatienederland.nl/kennisdossiers/vitale-kwetsbare-functies/bescherming/keteneffecten/methodes/hazop-methode/

https://www.iriusrisk.com/resources-blog/pasta-threat-modeling-methodologies