



Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Méréstechnika és Információs Rendszerek Tanszék

# Kiberfizikai rendszerek modellalapú kiberbiztonsági analízise

DIPLOMATERV

*Készítette*  
Csernátony Döme Máté

*Konzulens*  
Dr. Vörös András

2024. május 20.

# Tartalomjegyzék

<b>Kivonat</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>1. Bevezetés</b>	<b>1</b>
<b>2. Háttérismerekek</b>	<b>3</b>
2.1. Kiberbiztonsághoz kapcsolódó alapfogalmak . . . . .	3
2.2. Elektronikus vezérlőegységek kiberbiztonsága . . . . .	4
2.3. Autóipari kiberbiztonsági szabályozások és szabványok . . . . .	6
2.3.1. UN ECE R155 . . . . .	6
2.3.2. ISO/SAE 21434 . . . . .	6
2.3.2.1. Követelmények a tervezési fázisra . . . . .	7
2.3.2.2. Követelmények a fenyelgetéselemzésre és kockázatértékelésre . . . . .	7
2.4. Fenyelgetésmodellezési keretrendszerek és módszerek . . . . .	8
2.4.1. CIA és AAA . . . . .	8
2.4.2. STRIDE . . . . .	9
2.5. Autóipari rendszerek általános tervezése és modellezése . . . . .	10
2.5.1. V-modell . . . . .	10
2.5.2. UML és SysML . . . . .	11
2.6. Felhasznált eszközök . . . . .	11
2.6.1. Papyrus . . . . .	11
2.6.2. Acceleo . . . . .	11
2.6.3. Eclipse Modelling Framework . . . . .	12
2.6.4. Xtend . . . . .	12
2.6.5. Sirius . . . . .	12
<b>3. Kapcsolódó tanulmányok</b>	<b>13</b>
3.1. Fenyelgetésmodellezés . . . . .	13
3.2. Kiberbiztonság és üzembiztonság kapcsolata . . . . .	15
3.3. Támadási fa generálás . . . . .	17
<b>4. A kiberbiztonsági analízis metodológiája</b>	<b>19</b>
4.1. Áttekintés . . . . .	19
4.2. Termékleírás és fenyelgetésmodell származtatása . . . . .	20
4.2.1. Károkzások azonosítása és attributálása . . . . .	20
4.2.2. Értékek azonosítása és attributálása . . . . .	21
4.2.3. Fenyelgetésmodell származtatása . . . . .	22
4.3. Fenyelgetésmodellezés . . . . .	22
4.3.1. Dependenciák meghatározása . . . . .	22
4.3.2. Támadási fák inicializálása . . . . .	22

4.3.3. Támadási fák szerkesztése . . . . .	23
4.4. Dokumentumok generálása és manuális analízis . . . . .	24
<b>5. A kiberbiztonsági analízis megvalósítása</b>	<b>26</b>
5.1. Áttekintés . . . . .	26
5.2. Kiberbiztonsági modell származtatása . . . . .	26
5.2.1. Kiberbiztonsági profil . . . . .	27
5.2.2. Fenyelgetésmodell generátor . . . . .	28
5.3. Fenyelgetésmodellező eszköz . . . . .	29
5.3.1. Metamodel . . . . .	30
5.3.2. Fenyelgetésmodell szerkesztő felület . . . . .	31
5.3.3. Támadási fák inicializálása . . . . .	32
5.3.4. Támadási fa szerkesztő felület . . . . .	33
5.3.5. Dokumentum generátor . . . . .	35
<b>6. Esettanulmány</b>	<b>36</b>
6.1. Károkozások és értékek azonosítása . . . . .	37
6.2. Fenyelgetésmodell származtatása . . . . .	40
6.3. Fenyelgetésmodellezés . . . . .	40
6.4. Támadási fa analízis . . . . .	41
6.5. Manuális analízis . . . . .	47
<b>7. Összegzés</b>	<b>49</b>
<b>Irodalomjegyzék</b>	<b>50</b>

## HALLGATÓI NYILATKOZAT

Alulírott *Csernátony Döme Máté*, szigorló hallgató kijelentem, hogy ezt a diplomatervet meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forráskat (szakirodalom, eszközök stb.) használtam fel. minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy autentikált felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Budapest, 2024. május 20.

---

*Csernátony Döme Máté*  
hallgató

# Kivonat

A modern gépjárművekben egyre jelentősebb szerepet kapnak a számítástechnikai megoldások. Ma egy prémium személyautó közel 150 elektronikus vezérlőegységgel (ECU) és számos kommunikációs sínnel rendelkezik.

Ezek a feltételek egyrészről lehetővé tették komplexebb üzembiztonsági (safety) megoldások és fejlett vezetéstámogató rendszerek (ADAS) használatát, másrészről viszont a járműbe integrált elektronikai eszközök és azoknak az infokommunikációs hálózatokhoz való csatlakozása megnövelte a lehetséges kiberbiztonsági fenyegetések számát.

Az elmúlt években a járművek ellen elkövetett kibertámadások száma évről évre folyamatosan növekedett és ez a szám hálózati kommunikációban résztvevő járművek terjedésével csak tovább fog növekedni.

Az autóiparban a kockázatalapú biztonság-kezelés terjedt el, mint kiberbiztonsági alapelvek, amely a felfedezett fenyegetésekhez, a megállapított kockázat alapján határozza meg az egyes védelmi mechanizmusok szükségességét.

Ezen fenyegetések, kockázatok és védelmi mechanizmusok megállapítására vonatkozó előírások már megtalálhatóak a modern autóipari szabványok közt, viszont az alkalmazásuk még további támogatást igényel. Ebben nyújthatnak segítséget a már elterjedt általános IT biztonsági keretrendszerök, valamint az autóiparban már régóta jelenlévő üzembiztonsági elemzés eszközei.

A feladatom célja, hogy adjak egy metodológiát amely segíti az autóipari rendszerek tervezési fázisban történő kiberbiztonsági analízisét, valamint megvalósítsak egy eszközt, ami minél magasabb szintű automatizálással teszi lehetővé az elemzés elvégzését.

# Abstract

Electrical and Electronic (E&E) solutions are playing an increasingly important role in modern automotives. Today, a premium car contains around 150 electronic control units (ECU) and several communication buses.

On the one hand, these conditions enabled the use of more complex safety solutions and advanced driver assistance systems (ADAS), but on the other hand, the electronic devices integrated in the vehicle and their connection to infocommunication networks increased the number of possible cyber security threats.

In recent years, the number of cyber attacks against vehicles has increased year by year, and this number will only continue to increase with the spread of vehicles participating in network communication.

In the automotive industry, risk-based security management has spread as a basic cyber security principle, which determines the need for individual protection mechanisms for discovered threats based on the established risk.

Provisions for establishing these threats, risks and defense mechanisms can already be found in modern automotive industry standards, but their application still requires further support. The general IT security frameworks that are already widespread, as well as the operational safety analysis tools that have been present in the automotive industry for a long time, can help in this.

The goal of my task is to provide a methodology that helps the cyber security analysis of automotive systems in the design phase, as well as to implement a tool that enables the analysis to be carried out with the highest possible level of automation.

## 1. fejezet

# Bevezetés

A járműelektronika, mint olyan elektronikus rendszer amely járművek belsejének valamelyik részén kerülnek integrálásra egy adott feladat ellátására, már a járművek történelmének korai szakaszában is fellelhetőek. Ez a kezdetekben, az 1930-as években, csak egy fedélzeti rádió volt. Később, az 1960-as évektől bővült a gyújtási rendszer elektronikai alapokra helyezésével. Napjainkban pedig már, a technológia és a félvezetők folyamatos fejlődésének köszönhetően, a járműelektronikai megoldások egyre több és több feladatot látnak el.

Ha csak belegondolunk, hogy milyen elektronikai eszközök lehetnek egy modern gépjárműben akkor hamar észrevesszük, hogy az ablaktörlő, az oldalsó ablakok és ülések mozgatása, környezetvédelmi szempontokból a motort szabályozó különböző érzékelők, kamerarendszerek, vagy a napjaink premium járműveiben már érintőképernyős fedélzeti számítógépek, és akár a szervokormányok elektromos rássegítése is mind ilyen elektronikai rendszerek használatával történnek.

Ezeket a beágyazott rendszereket általánosan elektronikai vezérlőegységeknek (ECU, electronic control unit) nevezzük.

Az elmúlt 10-20 év technológiai fejlődése forradalmi változásokat hozott az autóiparban. Először megjelentek a hibrid, majd a teljesen elektromos hajtásláncú járművek. Részarányuk az eladásokban drasztikusan növekedett. Bizonyos adatok szerint évente akár 50%-kal is több került forgalomba. Ezek a fejlesztések elsősorban környezetvédelmi szempontok miatt történtek, a kőolaj és egyéb nem megújuló energiaforrástól való elszakadást szolgálják. Ennek a forradalomnak egyik nagy eredménye, ami dolgozatom témájához is kapcsolódik, az a tendencia, miszerint már a járművet működtető legkomplexebb mechanikai folyamatokat is villamossági alapokra terelték át.

A másik, szintén forradalminak nevezhető változás a fejlett vezetéstámogató rendszerek (ADAS, advanced driver assistance systems) területén jelentkezett. Egymásután jelennek meg újabb és újabb megoldások, melyek mára már teljes önvezetésre képes járműveket eredményezett. Ez bizonyos szempontból összefügg az előzővel, hiszen a komplex jármű folyamatok átterelését villamossági alapokra, részben az önvezetésre való törekvésnek is köszönhető. Azonban érdemes kiemelni, hogy már az önvezetés nélkül is, a jármű évjáratától és felszereltségtől függően, találhatunk rengeteg vezetéstámogató rendszert. Ezek a rendszerek már olyan komplexitással rendelkeznek, hogy a jármű különböző komponensei akár különböző beszállítótól is érkezhetnek, és mégis koordinált feladat végrehajtást képesek ellátni. Ezeknek a rendszereknek a feladata lehet egyrészről kényelmi (pl. tempomat, parkolás segítő, stb.) vagy biztonsági (pl. vonalkövetés, holttérfigyelő, fáradtságérzékelő, stb.).

Bár ezeknek a rendszereknek az ismertetése egy külön fejezetet is megérne, dolgozatom szempontjából az a fontos, hogy ezeknek a rendszereknek a jelenléte és az

elektronikai megoldások terjedése olyan kockázattal jár, amelynek kezelésére az autóipar és járműgyártás rugalmatlan struktúrái még viszonylag korlátozottan vannak felkészülve, ez pedig a kiberbiztonsági kockázatoknak a megjelenése.

A biztonsági kockázatok kezelése nem új doleg az autóiparban, hiszen az üzembiztonság már több mint egy évtizede szabványosítva (ISO 26262, 2011) működik. A kiberbiztonság még csak pár éve került szabványosításra (ISO 21434, 2021). Ez azt jelenti, hogy ezeknek a rendszereknek a kiberbiztonsági elemzése még sokkal kevesebb magas érettségű technikával rendelkezik, mint az üzembiztonsági elemzések.

Üzembiztonság esetén a kockázatok már korai fázisban felmérésre kerülnek és azokhoz a fejlesztési időt megelőzően, már a tervezési fázisban meghatározzák a mitigációkat. Ez annyit tesz, hogy már az egyes komponensek tervezésekor, lehetnek azok rendszer-, szoftver- vagy hardverszintű hibák, a kezdeti architektúra is úgy van meghatározva, hogy ezeknek a potenciális hibáknak az előfordulása minimális legyen.

Ezzel szemben a kiberbiztonság egy fiatal terület a kiber-fizikai rendszerek világában. Az általános IT rendszereknél viszont ez is rendelkezik egy pár évtizedes múlttal. Itt jellemzően már kész integrált rendszereknek történik az elemzése, felméri a támadó potenciális belépési pontjait, a lehetséges céljait, majd ezekre határoznak meg további szoftveres (pl. tűzfal) vagy hardveres (pl. DMZ) védelmeket. Ezt a folyamatot nevezik általánosságban fenyegetésmodellezésnek (threat modelling). Még szintén fontos megemlíteni a monitorozást és az utánkövetést, hiszen újabb és újabb sérülékenységek kerülhetnek elő a termék életciklusa során, amelyeket utólag kell javítani ezeknél a rendszereknek.

Ezzel együtt is a korábban említett ISO 21434 szabvány tesz több ajánlást az IT rendszerek biztonsági elemzésére felhasznált módszerek adaptálására egy az üzembiztonsághoz hasonló kockázatalapú tervezési fázisú felmérésre. Ezt nevezik Threat Analysis and Risk Assessment-nek, vagy röviden TARA-nak.

Dolgozatomban először egy olyan eljárást javaslok, amely az autóiparra már jellemző modellekből kiberbiztonsági elemzésre alkalmas modelleket készít. Ez az eljárás automatikusan származtatja a rendszermodellből a fenyegetésmodellett. Ezt követően ennek a származtatott modellnek az elemzésére fejlesztek egy eszközt, ami egyrérszről követi az ISO 21434-ben definiált TARA követelményeit és ajánlásait, másrérszről pedig felgyorsítja a kiberbiztonsági mérnökök munkáját támadási fák valamint dokumentumok automatikus generálásával.

A *Háttérismertetek* fejezetben bemutatom az autóipari kiberbiztonság szabályozási területét, bevezetem a szükséges fenyegetésmodellezési fogalmakat valamint bemutatom a megvalósításhoz használt eszközöket.

A *Kapcsolódó tanulmányok* fejezetben a már a témaiban létező és általam elemzett kutatásokat mutatom be, ismertetve azok alkalmazhatóságát a dolgozatom célja elérésében. Ugyancsak taglalom a megközelítésükben lévő hiányosságokat.

A *kiberbiztonsági analízis metodológiája* a modellező eszköz alkotóelemeivel, azok működésével, valamint a használatuk bemutatásával foglalkozik.

A *modellező eszköz megvalósítása* az eszközök felhasznált technológiákról és az azokban lévő architektúralis megoldásokról, valamint döntésekéről szól.

Az *Esettanulmány* című fejezet az eljárást egy példán való bemutatása, az analízis végrehajtása, valamint az eredmények értelmezése.

Végül pedig az *Összegzés* alatt lesznek találhatóak az elért célok kiértékelése, alkalmazási lehetőségek és bővítési lehetőségek. A dolgozat az *Irodalomjegyzékkel* zárul, ahol a használt források találhatóak.

## 2. fejezet

# Háttérismerekek

A célja ennek a fejezetnek, hogy összefoglaljam a témám értelmezéséhez szükséges alapismereteket, fogalmakat és bemutassam a használt technikai eszközöket. Ebben a fejezetben leírtak lesznek szükségesek ahhoz, hogy a későbbi fejezeteket teljességiükben lehessen értelmezni.

Az első fejezet tartalmazza a későbbiekben használt szakszavakat az elterjedt szakirodalmakban található leírások alapján.

Ezután következik az autóipari kiberbiztonság szabályozási környezete, azon belül is elsősorban az ISO/SAE 21434 szabvány, ami lefedi a terület alapvető irányelveit, valamint ad egy kezdetleges metodológiát a kiberbiztonsági kockázatelemzésre.

Ezt követi egy leírás az IT biztonság területén már ismert fenyegetésmodellezés technikákról és keretrendszerkről.

Végül pedig a munkám során alkalmazott eszközök és technológiák rövid ismertetése olvasható.

### 2.1. Kiberbiztonsághoz kapcsolódó alapfogalmak

Ebben a fejezetben található minden továbbiakban nem általánosnak vehető, az autóipari és a kiberbiztonsági területeken használt szakszavaknak a definíciói. Ezek a leírások elsősorban a már elérhető kutatásoknak, szabályozásoknak és szabványoknak a szójegyzékére építenek.

- **Termék (product / item):** Egy önmagában is értelmezhető és értékesíthető rendszer, amelynek a biztonságát biztosítani kell. Ez lehet egy komponens vagy komponensek csoportja és egy jármű-szintű funkcionálitást valósít meg, pl. kormányrendszer, fékrendszer, szoftver-frissítési infrastruktúra
- **Komponens (component):** Logikailag és/vagy technikailag szeparálható elem
- **Úthasználó (road user):** Személy aki valamilyen formában az utat használja, pl. gyalogos, autóvezető, utas, stb.
- **Kiberbiztonsági terv (cybersecurity concept):** Kiberbiztonsági követelményei egy terméknek, az üzemeltetési környezetnek, támogató információk a mitigációkhöz
- **Kiberbiztonsági specifikáció (cybersecurity specification):** Részletesebb kiberbiztonsági követelmények allokálva az architektúrális tervre
- **Kiberbiztonsági cél (cybersecurity goal):** Magas-szintű kiberbiztonsági követelmény

- **Kiberbiztonsági állítás (cybersecurity claim):** Állítás egy kockázatról
- **Mitigáció (mitigation / cybersecurity control):** Kockázatmódosító intézkedés
- **Érték (asset):** Egy tárgy ami értékkel rendelkezik
- **Kiberbiztonsági tulajdonság (cybersecurity property):** Egy attribútum amelyet meg kell védeni, pl. sérтetlenség, bizalmasság, elérhetőség
- **Károkozás (damage scenario):** Egy kedvezőtlen következmény amely hatással van az úthasználóra
- **Fenyegetés (threat scenario):** Egy lehetséges kompromittálása valamilyen érték kiberbiztonsági tulajdonságának, amely egy károkozásnak vezethet (pl: egy szoftveres sérülékenység kihasználása)
- **Támadási útvonal (attack path):** Események (pl: egy fenyegetés megvalósulása) egymás utáni bekövetkezése, amely egy károkozást realizálnak
- **Támadási lépés (attack step):** Az egyes események amelyeknek az egymás utáni sorozata egy támadási útvonalat realizál

## 2.2. Elektronikus vezérlőegységek kiberbiztonsága

A kiberbiztonság területét a védett eszközök, az üzemeltetési környezet, valamint a védelmi megoldások szerint három fő csoportra lehet osztani. Ezek között vannak átfedések és az élő rendszerek valahol ezen halmazok metszeteiben helyezhetőek el, azonban érdemes ezeknek a fogalmaknak is az ismertetése ahhoz, hogy az autóipari termékek kiberbiztonságát könnyebben el tudjuk helyezni.

Az egyik ilyen csoport az általános IT (Information Technology) Security amely hálózati infrastruktúrák védelméről szól. Ilyenkor a védett eszközök lehetnek szerverek, routherek, számítógépek, tárolt adatok stb.

Az OT (Operational Technology) Security már tovább bővíti a hálózati eszközöket a jellemzően gyártósori vagy erőművi beágyazott rendszerek védelmével amelyek lehetnek PLC-k vagy más irányítórendszerek.

A harmadik csoport pedig az IoT (Internet of Things) security lenne, ide tartozhat egyrészről a külön vett PLC-k biztonsága, valamint a háztartási eszközöktől (okos hűtő, sütő, stb.), okos otthon rendszereken át (redőny, klíma, stb.), egészen az ipari termékek biztonságáig amelyek végül járművek, repülők vagy más rendszerek működtetéséért felelnek.

Az elektronikus vezérlőegység (electronic control unit, ECU) a tágabb értelemben vett IoT security területéhez tartoznának.

Az elektronikus vezérlőegységekből egy prémium személyautóban akár 100-150 db is létezhet, ezek különböző feladatokat láthatnak el kezdve az ülésmagasság állításától a motor részeinek szabályozásáig. Ezek a vezérlőegységek kommunikációs buszokon keresztül kommunikálnak egymással, ezek a buszok lehetnek logikailag vagy akár fizikailag szegmentáltak. Egy ilyen vezérlőegység látható a 2.1 ábrán, a kommunikációs sínek pedig megtékinthetőek egy modern Bentley keresztmetszeti képén a 2.2 ábrán.

Egyes vezérlőegységek képesek lehetnek vezetéknélküli kommunikációra is, fedélzeti számítógép esetén ez lehet WiFi, GPS, Bluetooth, NFC stb., ami nagyban megnövelte a



**2.1. ábra.** Referencia kép egy elektronikus vezérlőegységről[19]



**2.2. ábra.** Egy Bentley keresztmetszeti ábrázolásán látható kommunikációs sínek[4]

kiberbiztonsági kockázatok mértékét a modern járművekben és lehetővé teszi azok működésének a távolról történő manipulálását.

Ami szintén fontos az az over-the-air szoftverfrissítés. Ma már nem arra rendezkedik be az autóipar, hogy a járműbe integrált eszközökön a szoftverfrissítést a szervizből fogják elvégezni, hanem arra, hogy azt akár távolról, otthonról a garázsborlásban is lehessen.

Ezek a feltételek jelentik a megnövekedett fontosságát a járművek kiberbiztonságának, azonban érdemes még kiemelni azokat a támadómodelleket amelyek jellemzőek lehetnek az autóiparban, ezzel is felkészülve a lehetséges károkra amelyektől védenénk a rendszert.

A bűnözői attitűd lehet egyrérször jellemző, az autólopások már az elektronikus eszközök integrációját megelőzik, azonban azoknak a jelenléte nem minden esetben jelent magasabb fokú biztonságot. Az új elektronikus rendszerek például lehetővé tették azt is, hogy a támadó a fényszóró leszerelésével belépési pontra tegyen szert a járműben, majd azon keresztül oldja fel a zárákat és indítsa el a járművét. Erről bővebben lehet olvasni a Gareth Halfacree cikkében [3].

Szintén érdekes és már régóta az autóipar része a tuningolás, ami a vezérlőegységek mennyisége és azoknak a belső paramétere miatt ez még nagyobb kontrollt adhat egy

jogosulatlan felhasználónak. Állíthat egyrészről akár kormányzási érzet paramétereket, de módosíthat kibocsátási vagy motor működést limitáló értékeket is.

A fizetős extráknak a jailbreak-elése már játék konzolok és telefonok esetén ismert, azonban mióta egyes autógyártók előfizetéssel tettek olyan extrákat mint a hátsó ülésfűtés és hasonlók, azóta ez itt is egy természetesen illegális, kártékony tevékenység ami egy támadó célja lehet az ECU ellen. Erről egy példát lehet olvasni Arthur Parkhouse cikkében [14].

Szintén fontos kiemelni a vírusok (malware, ransomware) telepítését is elektronikus vezérlőegységekre, hiszen a távoli szoftverfrissítés támogatása lehetővé teszi az ezen támadások végrehajtását is.

Végül amit fontos észben tartani, hogy az ezeken az eszközökön tárolt adatokat is fontos védeni, hiszen egy fedélzeti számítógép potenciálisan tárolhat személyes adatokat (personal identifiable information, PII) vagy akár felhasználóneveket, jelszavakat is navigációs és más elérhető alkalmazásokhoz. De maga a tárolt szoftver se kerülhet ki nyilvánosan hiszen az elősegítené a támadóknál a sérülékenységek keresését, azokra automatizált megoldások fejlesztését.

## 2.3. Autóipari kiberbiztonsági szabályozások és szabványok

Az autóipari kiberbiztonság területe ellentétben az üzembiztonsággal vagy a bővebb IT biztonsággal még csak pár éves múltra tekint vissza, emiatt az itt alkalmazandó szabályozások és szabványok még csak az első változatukban kerültek kiadásra.

### 2.3.1. UN ECE R155

Az első specifikusan autóipari szabályozás az Egyesült Nemzetek által kiadott 155-ös számú szabályozás a kiberbiztonságról és a kiberbiztonság kezelő rendszerekről és járművek engedélyeztetésének kapcsolatáról (UN ECE R155[13]). Ennek a szabályozásnak kell megfelelnie a járműgyártóknak és beszállítóiknak az összes 2024 után megjelenő járműmodell engedélyeztetéséhez.

Ez a szabályozás már tartalmazza az igényt a kockázat-alapú kiberbiztonsági kezelés szükségességrére. Ami annyit tesz, hogy a biztonsági szolgáltatásokat az alapján kell meghatározni, hogy egy kiberbiztonsági fenyegetés esetleges bekövetkezése mekkora hatással lenne a védendő autóipari termékre.

Szintén már megtalálhatjuk az autóipari termékek életciklusának különböző szakaszain más-más fenyegetésekre lehet számítani, és ennek megfelelően más követelmények is lesznek érvényesek a termékre.

A dokumentum a továbbiakban követelményeket határoz meg, hogy milyen folyamatokon kell keresztülmennie egy autóipari terméknek, ahhoz, hogy az a közúti használatra engedélyt kapjon.

### 2.3.2. ISO/SAE 21434

A másik, már technikaibb szintű, szintén 2021-es megjelenésű, irányadó szabvány az autóipari kiberbiztonsági mérnökségről szóló ISO/SAE 21434 "Road vehicles - Cybersecurity engineering"[6]. Ezt a szabványt közösen fejlesztette és adta ki 2021 augusztusában az International Standards Organization (ISO) és a Society of Automotive Engineers (SAE).

Ez a szabvány kezdett el követelményeket megfogalmazni az autóipari rendszerek (E/E) kiberbiztonsági kockázatkezelésének menetére, valamint a biztonság fejlesztésére és

kezelésére. A felépítése emlékezetheti az olvasóját a már jóval ismertebb ISO 26262 "Road vehicles - Functional safety" szabványra, amely ugyanazon termékek üzembiztonságának a kezelésére és elemzésére fókuszál.

A szabvány először a tervezési, fejlesztési, gyártási, üzemeltetési, karbantartási és kivezetési fázisokra fogalmaz meg követelményeket, valamint tartalmaz egy fenyegetés elemző és kockázat értékelő eljárást amelynek a Threat Analysis and Risk Assessment (TARA) nevet adták.

Továbbá tartalmaz más követelményeket a kiberbiztonsági elvárások kezelésére különböző menedzsment és organizációs szintekre, azonban ezek ismerete nem tartozik a témám látókörébe.

Dolgozatom kifejezetten a tervezési fázishoz tartozó kockázatelemzés végrehajtására vonatkozó követelményeket veszi alapul. A későbbi bemutatásuk során felfedezhető lesz, hogy a kockázatelemzés iteratív használatának szükségessége az életciklus különböző fázisaiban, azonban a termék üzemeltetési környezetére vonatkozó védelmet ebben a tervezési fázisban határozzuk meg. Ezzel elkerülve a magasabb költségű utólagos fejlesztéseket.

### 2.3.2.1. Követelmények a tervezési fázisra

A tervezési fázis egy autóipari termék életciklusában a kiindulópont. Az ebben a fázisban végzett kiberbiztonsági tevékenységek célja, hogy (i) definiálásra kerüljen az elemzendő termék, a környezete és interakciói, (ii) meghatározzák a kiberbiztonsági célokat és állításokat valamint, hogy (iii) elkészüljön a kiberbiztonsági terv.

A **termék definíciója** tartalmazza a termék határait, feladatait, valamint az előzetes architektúrát. Célunk itt az, hogy összegyűjtsük az elemzéshez szükséges információkat.

A **kiberbiztonsági célok és állítások** meghatározásához szükséges a TARA elvégzése, aminek az eredményeképp születnek meg, az egyes kockázatok kezelésére vonatkozó döntések, amelyek alapján eldönthetjük, hogy a kockázathoz egy célt vagy állítást kell megfogalmaznunk. A cél fogja meghatározni a magas-szintű követelményt amit a termék fejlesztése során figyelembe kell venni, míg az állítás azt határozza meg, hogy az adott kockázat mitigálása valamilyen okból már teljesült vagy a teljesülése szükségtelen.

Ezután készülhet el a **kiberbiztonsági terv**, amelyben az egyes mitigációkat határozzuk meg a célok elérésére, a célokat tovább finomítjuk követelményekké, majd azokat allokálhatjuk a termékre vagy egyes komponensekre.

### 2.3.2.2. Követelmények a fenyegetéselemzésre és kockázatértékelésre

A TARA bemenete a termék definíció, és ez alapján lehet elvégezni a hét lépésből álló kockázatelemzési eljárást aminek a kimenete az egyes fenyegetések kockázati értékkel, valamint az azok kezeléséről szóló döntés.

Az első lépése a kockázatelemzésnek az **érték azonosítás**. Ennek a lépésnek két célja van. Az egyik, hogy a lehetséges károkozásokat azonosítsuk és azok segítségével az egyes értékeket is meghatározzuk, a másik pedig, hogy az értékekhez *kiberbiztonsági tulajdonságokat* rendeljünk. A károkozások tartalmazhatják a kár körülírását, a releváns értékeket és a kapcsolatot a járműfunkcionalitás és a kedvezőtlen következmény között. Az értékek azonosítására használhatjuk továbbá a termékleírást, a *fenyegetések* definiálását vagy már létező katalógusokat.

A kockázatelemzés második lépése a **hatásértékelés**. Itt a célunk az egyes lehetséges károkozásokat és azok következményeit valamilyen keretrendszer mentén értékelni. Egy lehetőség, amit több szabvány is említ, az az SFOP alapú értékelés. Az SFOP négy dimenziót határoz meg amiben el kell végezni az értékelést. Ezek az üzembiztonsági hatás (safety), gazdasági hatás (financial), üzemeltetési hatás (operational), valamint az adatvédelmi hatás (privacy).

A kockázatelemzés harmadik lépése a **fenegetések azonosítása**, amelyekhez hozzá kell rendelni a támadott értéket, annak a kompromittált *kiberbiztonsági tulajdonságát*, valamint a kompromittálás okát. A szabvány szerint ezek azonosítására lehet egyrészről csoportos, brainstorming alapú vagy szisztematikus, keretrendszer által meghatározott módszerekkel alkalmazni. Az utóbbi esetben javasolt valamilyen ismert fenyegétesmodellezési megközelítést használata. Néhány felsorolt példa ezekre az EVITA, TVRA, PASTA és a STRIDE.

A negyedik lépés a **támadási útvonal elemzés**. Az elemzés során a szabvány szerint top-down vagy bottom-up megközelítést is használhatunk. Előbbi esetben támadási fák, támadási gráfokat, utóbbi esetben már ismert sérülékenységekre alapulót.

Az ötödik lépés a **támadás megvalósíthatóságának vizsgálata**, ahol több már letező keretrendszert alkalmazhatunk az egyes támadási útvonalak kiértékelésére.

A hatodik lépés a **kockázatiérték meghatározás**. Itt egy egytől ötig terjedő skálán értékeljük a fenyegétesi szcenáriókat a hatásértékek és a megvalósíthatósági értékek alapján.

A hetedik és egyben utolsó lépés pedig a **kockázatkezelési döntés**, amikor az egyes kockázatok kezeléséről hozhatunk döntést. A kockázatokat elkerülhetjük, csökkenthetjük, megoszthatjuk, valamint megőrizhetjük.

Jól látható, hogy ezek a követelmények elég általánosak, sok döntési jogosultságot helyez a folyamatot vezető személyekre, emellett viszont magas szinten jól körülírt követhető lépéseket határoz meg amelyek megfelelnek más szabályozások feltételeinek és képes eljuttatni a mérnököt a konkrét megvalósítandó intézkedések meghatározásához.

## 2.4. Fenyegétesmodellezési keretrendszer és módszerek

A fenyegétesmodellezés egy olyan folyamat, aminek segítségével azonosítani tudjuk a lehetséges fenyegéteseket, valamint segítenek azok értékelésében.

Ez a folyamat már viszonylag régóta elterjedt a kiberbiztonsági szakmában és támogató jellegű kapcsolatban áll a kockázatelemzésekkel. Amíg a fenyegétesmodellezés célja a fenyegétesek meghatározása, a kockázatelemzés az ami segít nekünk a feltárt fenyegétesek kezelésének prioritálásában vagy esetenként az egyes fenyegétesek elhagyásában.

Tágabb értelemben akár a kockázatelemzést is vehetjük a fenyegétesmodellezés részének, azonban az ISO/SAE 21434 szabványban leírt folyamat is különválasztja azokat és a fenyegétesmodellezést kifejezetten a fenyegétesek meghatározására javasolja.

### 2.4.1. CIA és AAA

Bár még nem is egy teljes fenyegétesmodellezési keretrendszer a CIA háromszög vagy CIA triad, mégis a legtöbb kiberbiztonsági elemzés az ezen betűszó által kifejezett modellt alkalmazza.

Már korábban beszélünk kiberbiztonsági tulajdonságokról, itt a CIA által definiáltak használjuk, ezek a bizalmasság (confidentiality), sérteletlenség (integrity) és elérhetőség (availability).

Szintén előfordul ennek a modellnek a bővítése egyéb tulajdonságokkal, ilyenek a szoftverbiztonság esetén használt AAA modell elemei amelyek az egyediség (authenticity), engedélyezhetőség (authorizability), valamint az elszámoltathatóság (accountability).

Adott értéknek a tulajdonságait meghatározhatjuk az alábbi kérdések megválaszolásával:

- **Bizalmasság:** Harmadik fél szerezhet-e tudomást az értékről, annak tartalmáról?
- **Sérteletlenség:** Az érték módosulása vezethet-e nem várt következményekhez?

- **Elérhetőség:** Az érték hiánya vezethet-e nem várt következményekhez?
- **Egyediség:** Kell-e az érték eredetét biztosítani felhasználása előtt?
- **Engedélyezhetőség:** Szükséges-e az adott értékhez való hozzáférés korlátozása?
- **Elszámoltathatóság:** Szükséges-e az adott értékhez való hozzáférések, módosulások visszakövethetősége?

A továbbiakban ezeket a modelleket fogom alkalmazni a kiberbiztonsági tulajdonságokként, azonban ezek módosíthatók, elhagyhatóak, cserélhetőek és bővíthetőek felhasználási környezetüktől függően. Az én esetben a CIA elegendő lesz, hiszen autóipari beágyazott rendszerekben a szoftver szinten azon tulajdonságok relevánsabbak. Az üzenet egyediségének hamisítása már az eredeti üzenet egyfajta sérülését vonja magával, ez a sér-tetlenség tulajdonság által már kezelve van. Az engedélyezhetőség pedig már bonyolultabb operációs rendszerek használatánál kerül elő, ami külön felhasználókat és hozzáféréseket tud definiálni. Ez bizonyos formában az autóiparban is fellelhető, de nem abban a komplexitásban mint Linux vagy Windows alapú rendszereknél. Az elszámoltathatóság szintén problémás mivel ennek a biztosítása, a beágyazott rendszerek limitált hardvererőforrásai miatt nem tud azzal a granularitással létezni, ahogy a hozzáférések számon lennének tartva IT rendszereknél.

#### 2.4.2. STRIDE

A STRIDE egy modell, amely számítógépes kiberbiztonsági fenyegetések azonosítására lett kifejlesztve a Microsoft által 1999-ben. A nevét a hat fenyegetéstípusról kapta, ezek és a jelentésük:

- **Spoofing:** Megszemélyesítés, amikor a rendszer hamisan érzékeli az információ kül-dőjének a kilétéét
- **Tampering:** Valamilyen információ megváltoztatása
- **Repudiation:** Annak az állítása, hogy valamit nem te csináltál vagy nem is történt meg
- **Information disclosure:** Egy támadó képes hozzáférni olyan információhoz amire nincs jogosultsága
- **Denial of Service:** Erőforrások túlterhelése miatt szolgáltatás elérhetetlenné tétele
- **Elevation of privilege:** Egy támadó képes olyan művelet elvégzésére, amire nincs felhatalmazva

Ez első ránézésre egy jó lehetséges kategorizálást ad meg nekünk fenyegetésekhez, valamint kibővíthető ezek kapcsolata az azonosított értékekhez és kiberbiztonsági tulajdon-ságaikhoz. Tehát az egyes fenyegetés típusok egy bizonyos tulajdonság sérülését célozzák.

Ebből jól látható, hogy az egyes értékekhez a tulajdonságaik alapján már azonosíthatjuk az azok kompromittálását célzó lehetséges fenyegetéseket.

Ezekből én mivel csak a CIA tulajdonságait használom, az én esetben a Tampering, Disclosure és Denial fenyegetések lesznek a tulajdonságokból származtatva.

Spoofing	Egyediség (authenticity)
Tampering	Sérhetetlenség (integrity)
Repudiation	Letagadhatatlanság (non-repudiability)
Information disclosure	Bizalmasság (confidentiality)
Denial of Service	Elérhetőség (availability)
Elevation of privilege	Engedélyezhetőség (authorizability)

**2.1. táblázat.** Fenyegesek kapcsolata kiberbiztonsági tulajdonságokkal

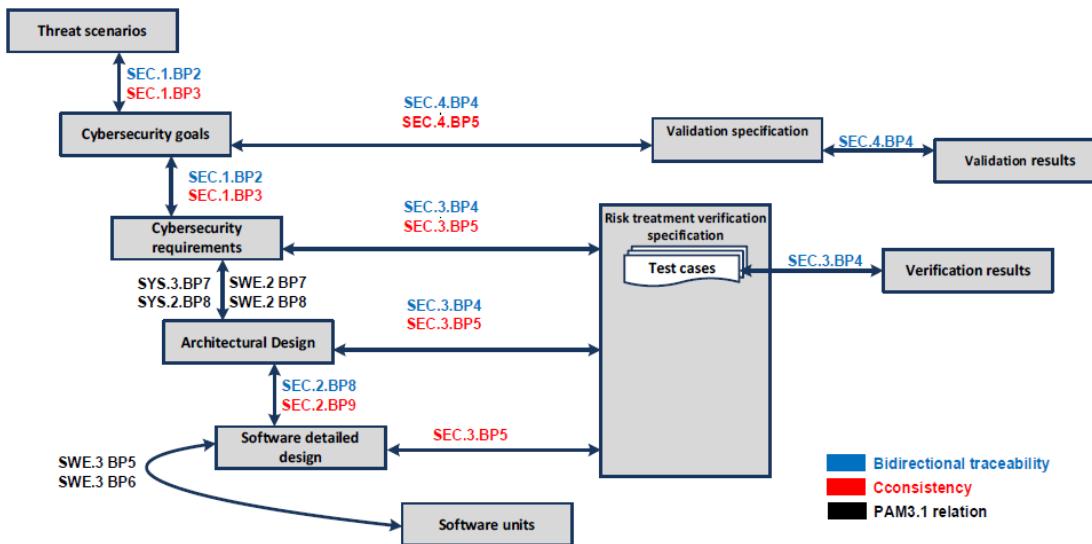
## 2.5. Autóipari rendszerek általános tervezése és modellezése

A diplomamunkám sajátossága abból adódik, hogy amíg az általános IT rendszereknek vagy azok egyes elemeinek az architektúrális tervezése kevésbé jellemző, addig a kiberfizikai rendszereknél, azon belül is a járműveknél, a rendszer komplexitása és az üzembiztonság kritikussága miatt, nagy hagyománya van ezeknek az átfogó dokumentálására már a tervezés kezdeti szakaszában.

### 2.5.1. V-modell

A V-modell egy szoftverfejlesztési folyamat amelyet az ASPICE szabvány adaptál az autóiparban. Lényegében arról szól, hogy a fejlesztés V alakban történik, ahol bal oldalt fentről lefelé történik a tervezés és a fejlesztés, a jobb oldalon pedig minden lépéshoz tartozik egy verifikációs vagy validációs lépés.

Nemrégiben kapott az ASPICE[1] szabvány egy kiegészítést a kiberbiztonsági mérnöki folyamatokhoz amelyeket részben az ISO 21434 is definiált. Ezekről egy összefoglaló a 2.3 ábrán látható.



**2.3. ábra.** Az ASPICE javaslata kiberbiztonsági folyamatokra[1]

Szintén érdemes itt megjegyezni, hogy az általam javasolt metodológia egyfajta vissza-csatolást (feedback) tenne lehetővé az *Architectural design* és a *Threat scenarios* lépések között. De erről később bővebben lesz szó.

### **2.5.2. UML és SysML**

A komplex E/E architektúrák esetén, mint amilyenek az autóipari beágyazott rendszerek, jellemző valamelyen formában a rendszermodellek jelenléte és karbantartása a termék életciklusa alatt. Erre elsősorban a SysML (System Modeling Language) van használva, ami egy bővítése az UML-nek (Unified Modeling Language).

Az UML egy általános felhasználású grafikus modellezési nyelv, amelynek célja a rendszerek specifikálása, a felépítésük leírása, vizualizálása és dokumentálása a fejlesztésben érdekelt minden résztvevő számára.

Az UML több diagram típust különböztet meg, azokat elsősorban két kategóriába sorolhatjuk, az egyik a strukturális a másik pedig a viselkedési diagramok. A strukturális diagramok közé tartozik a csomag, a komponens, az objektum, az osztály, a kompozit, a profil és a telepítési diagramok. A viselkedési diagramok közé pedig az aktivitás, az állapotgép, a használati eset, a kommunikációs, a szekvencia, és az időzítési diagramok.

Az UML szintén támogatja a modellezési nyelvnek egy adott doménre való szabását. Ezt profilok definiálásával lehet megtenni. A profilokra érdemes úgy gondolni, mint az UML egyfajta bővítményei, amelyeket bizonyos modell elemekre tudunk rászabni.

A SysML az UML nyelvi elemei egy részhalmazának további nyelvi elemekkel bővített verziója. Ezeket a bővítéseket egy SysML profillal implementálják és elsősorban ezt a nyelvet használják a komplex hardver-szoftver rendszerek modellezésére.

Az én megoldásom az alap UML bővítéseképp tartalmaz egy kiberbiztonsági profilt, mivel a SysML bővítményei nem voltak elegendők az autóipari rendszerek kiberbiztonsági elemzésére, tervezésére. A termék leírására, valamint a kockázatelemzés érték és károkozás definíciós szakaszára egy használati eset (use case) és egy komponens (component) diagramot használlok.

## **2.6. Felhasznált eszközök**

### **2.6.1. Papyrus**

A Papyrus egy nyílt forráskódú UML 2 modellező eszköz. Ezt az eszközt fogom használni az értékek definiálására egy komponens diagramon valamint a károkozásokat egy használati eset diagramon keresztül.

Ebben az eszközben definiálok továbbá egy kiberbiztonsági profilt, ami bővítményeket tartalmaz a komponensekhez és a használati esetekhez.

### **2.6.2. Acceleo**

Az Acceleo egy nyílt forráskódú Model-2-Text (M2T) eszköz, amellyel a Papyrus-ban definiált modellekkel fogom generálni a kiberbiztonsági elemző eszköz kiinduló modelljét. Más szóval ezzel az eszközzel származtatom a rendszermodellból a kiberbiztonsági modellt.

Azért eset a választásom erre az alkalmazásra az Xtend helyett, mivel az integrációja a Papyrus eszközzel sokkal jobban támogatott, illetve mivel nincs szükség nagy komplexitású kódgenerálásra.

### **2.6.3. Eclipse Modelling Framework**

Az Eclipse Modelling Framework, vagy röviden EMF egy modellezési keretrendszer ami arra ad támogatást, hogy könnyen lehessen modellező eszközöket, majd ahhoz kódgenerátor alkalmazásokat fejleszteni.

Az EMF támogatja modellek definiálását, majd azokból automatikusan Java kódot származtat, ezzel elősegítve a modell könnyebb transzformációját, módosítását és abból való származtatást.

Az EMF szintén ad egy automatikusan generált kezelőfelületet a definiált modell szerkesztésére, tartalommal feltöltésére.

Ezt a keretrendszert használtam a kiberbiztonsági elemző eszköz metamodelljének definiálására, továbbá az eszköz kezelő felülete is a generált szerkesztő felületre épül.

### **2.6.4. Xtend**

Az Xtend egy Java alapú programozási nyelv amelyet elsősorban kódgenerálási célokra lehet használni.

Ezt a nyelvet használtam arra, hogy elkészítsem először a modellből való generálását a szabványos dokumentumoknak, majd a támadási fák inicializálását is.

### **2.6.5. Sirius**

A Sirius egy nyílt forráskódú szoftverprojekt amelynek a célja, hogy könnyen lehessen grafikus felületeket létrehozni domén specifikus modellekhez Eclipse-ben.

Ez a keretrendszer volt használva a támadási fák megjelenítéséhez és azok szerkesztésére használt grafikus felület létrehozására.

### 3. fejezet

## Kapcsolódó tanulmányok

Ennek a fejezetnek célja, hogy összefoglaljam a diplomamunkámhoz előzetesen elvégzett kutatómunka során megismert tanulmányok eredményeit, problémáit, valamint a lehetséges alkalmazásukat.

A kutatómunkámat három témaban végeztem, az első a fenyelgetésmodellezés (threat modeling) területe volt. Ezen kutatásokon keresztül ismertem meg a fenyelgetések felmérése során felmerülő problémákat, megoldásuknak módjait, azok lehetséges megjelenítését és modellezését.

A második téma az autóipari biztonsági elemzések területén készült munkák kutatása volt, hogy meg tudjam ismerni, hogy egy elemzés során milyen komponensekre és azoknak mely attribútumaira kell fókuszálni. Szintén megismertem olyan metodológiákat és best practice-eket, amelyeket az általam kidolgozott metodológiámban is fel tudtam használni.

A harmadik a legspecifikusabb téma azon kutatási eredmények megismerése, amelyek támadási fák (attack trees), illetve támadási gráfok (attack graphs) generálásáról szóltak. Ezek a publikációk az üzembiztonság területén elterjedt hibafa (fault tree) analízis eszközének adaptációjai a kiberbiztonsági terület támogatására.

### 3.1. Fenyelgetésmodellezés

Az első téma a Karahasanovic et al. [8] "Adapting Threat Modelling Methods for the Automotive Industry" című munkája illeszkedik. Ez két fenyelgetésmodellezési keretrendszeret mutat be. Az egyik az Intel-hez köthető TARA (Threat Agent Risk Assessment), ami nem összekeverendő az azonos rövidítéssel fémjelzett Threat Analysis and Risk Assessment metodológiával amit az ISO 21434 definiál. A másik pedig, a sokkal ismertebb Microsoft által fejlesztett STRIDE.

Az előbbi a grafikus modellezési technikák helyett egy könyvtárakon alapuló fenyelgetés elemzést mutat be. Az elemzés során három könyvtárat használnak, egyik a lehetséges támadó ágenseket, másik az általuk véghezvihető támadásokat a harmadik pedig, a jellemző támadási felületeket gyűjti össze. Ezen könyvtárakból határoz meg egy olyan részhalmazt, ami az autóipari rendszerek ellen alkalmazható.

A második technika a támadó-centrikusság helyett inkább szoftver-centrikus irányt követi, ami egy fehér doboz vizsgálatot tesz lehetővé a rendszeren. Ez, a későbbiekben még előforduló Data Flow Diagramok használatára mutat be egy példát amelyben a szoftver komponensek közti kommunikációt modellez a és így tud egy támadási útvonalat végigkövetni.

Az előbbi technika gyengesége az, hogy csak magas szinten definiálja a fenyelgetések, ami nem eléggyes a védelmi mechanizmusok meghatározására. Utóbbi ezzel szemben alkalmas arra, viszont nem lehet vele rendszer szintű védelmet modellezni, valamint

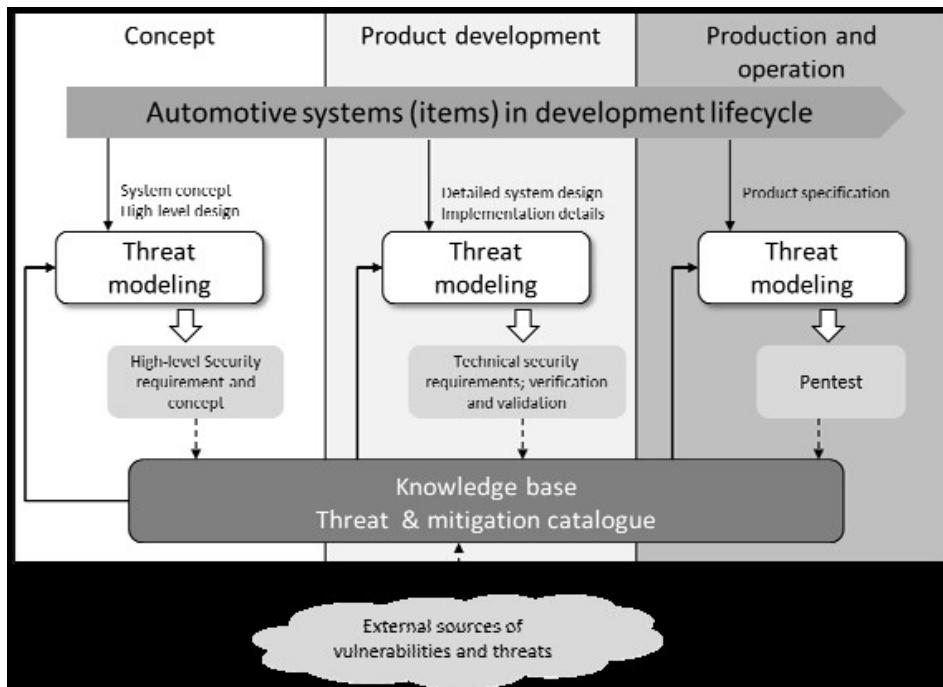
tervezési fázisban is nehezen alkalmazható.

Ma et al.[18] "Threat Modeling for Automotive Security Analysis" című munkája, a fenyegetésmodellezést egy sokkal gyakorlatiasabb módon közelíti meg, nem feltétlenül a technikai részekre koncentrál, hanem a termékfejlesztési életciklust és a már meglévő üzembiztonsági analíziseket is figyelembe veszi.

Helyesen fogalmazza meg jelzi a szükségét az elemzésszintekre bontásának, valamint hasonlóan az üzembiztonsághoz, egy funkcionális kiberbiztonsági tervnek, és egy termék-specifikációval kiegészített technikai kiberbiztonsági tervnek a szükségességét. Ezeket aztán lebontja és egyrészt a tervezési fázisban, magas szintű követelmények azonosításához, másrészt a rendszermodellt használva a termékfejlesztési szakasz bemeneteként, harmadrészt a gyártási fázisban használja.

Ez a munka tartalmaz még egy esettanulmányt, amely egy jármű utasterének a biztonsági analízisén vezet végig, a Microsoft Threat Modelling Tool használatával, ami a STRIDE keretrendszerre épít, data flow diagramokat használ és modell alapon generál lehetséges fenyegetéseket.

Konklúzióként emeli ki az igényt, hogy a biztonsági analízis modellezési paradigmáit integrálni kell a rendszermodellezés paradigmáival, ezzel biztosítva, hogy az analízis a változások során napra kész maradjon. A termékfejlesztési fázisokat és kiberbiztonsági elemzéseket a 3.1 ábrán részletezik.



**3.1. ábra.** Fenyegetésmodellezés a termékfejlesztési életciklusokban[18]

Ehhez a területhez tartozik még Vivek et al.[16] "Automotive system threat modelling" című esettanulmánya, amelyben egy tetszőleges autóipari komponensre, a kiértékeléshez egy módosított STRIDE modellt használ, valamint a lehetséges fenyegetések feltárására az ISO 21434-ben definiált kockázatelemzés kezdeti lépései alkalmazza.

### 3.2. Kiberbiztonság és üzembiztonság kapcsolata

Ezzel a témaival kapcsolatban Dantas et al. [17] "Security engineering for ISO21434" című munkája mutatja be, hogy a szabványosított kockázatelemzésre, hogyan lehet már meglévő technikákat alkalmazni, valamint, hogy a kockázatelemzés, hogyan illeszkedik be a már létező folyamatokba.

A dokumentum részletesen elemzi a kiberbiztonság szerepét az autóiparban, valamint a szoftverfrissítés fontosságát az esetleges sérülékenységek javításában. Szintén ki van emelve a folyamatok rendszeres és folyamatos auditálásának és kiértékelésének a szüksége. Ezekhez jelzi a lehetőséget különböző domén-specifikus nyelvek használatának lehetőségét és automatizálás integrálását, illetve modelellenőrző rendszerek bevezetését.

Van még szó az üzembiztonság területén alkalmazott FTA (Fault Tree Analysis) és FMEA (Failure Modes and Effects Analysis) technikák felhasználásáról a támadási útvonalak elemzésében, illetve idéz több más tanulmányt és keretrendszer amelyek szintén ezen alkalmazásokat ösztönzik.

Böhner et al.[11] "Extending software architectures from safety to security" című munkája az üzembiztonságra kidolgozott architektúrát terjeszti ki a kiberbiztonsági kockázatok kezelésére. Helyesen hívja fel a figyelmet arra, hogy a kiberbiztonsági elemzések alapjául használt CIA triádból kettő, az integritás (integrity) valamint a rendelkezésre állás (availability) az üzembiztonság területén már a véletlenszerű hibák esetére alkalmazva vannak.

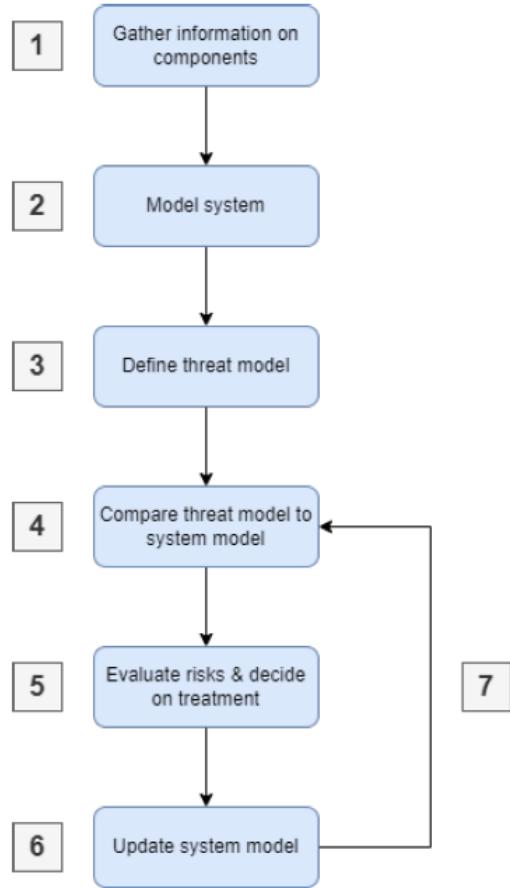
Említésre kerülnek még olyan megoldások, amelyek mind a két területen csökkentik a kockázatot. Ilyenek a memória particionálás illetve az üzenetek védelme a módosítás ellen. Bemutatja, hogy a kiberbiztonság esetén a szándékossan okozott hibákat kell figyelembe vennünk, ellentétben az üzembiztonság véletlenszerűen előforduló hibái helyett. A kockázat csökkentő intézkedések alkalmazása pedig szignifikánsan tudja mind a két fajta biztonság hatékony szolgáltatását.

Chulp et al.[15] "ThreatGet: Toward automated attack tree analysis for automotive cybersecurity" című munkája egy ThreatGet nevezetű kiberbiztonsági kockázatelemző eszköz működési elvét mutatja be amely a bécsi egyetemen készült. Ez az eszköz gyakorlatiasan írja le a tervezési fázisban elvégzendő kockázatelemzés menetét. Az eszköz az ASPICE-szal ellentétben már vizsgálja a folyamat lépései közti feedback lehetőségét és egy külön modellt is használ kockázatelemzésre amelyet össze hasonlítana a meglévő rendszermodellel, ahogy az a 3.2 ábrán látható.

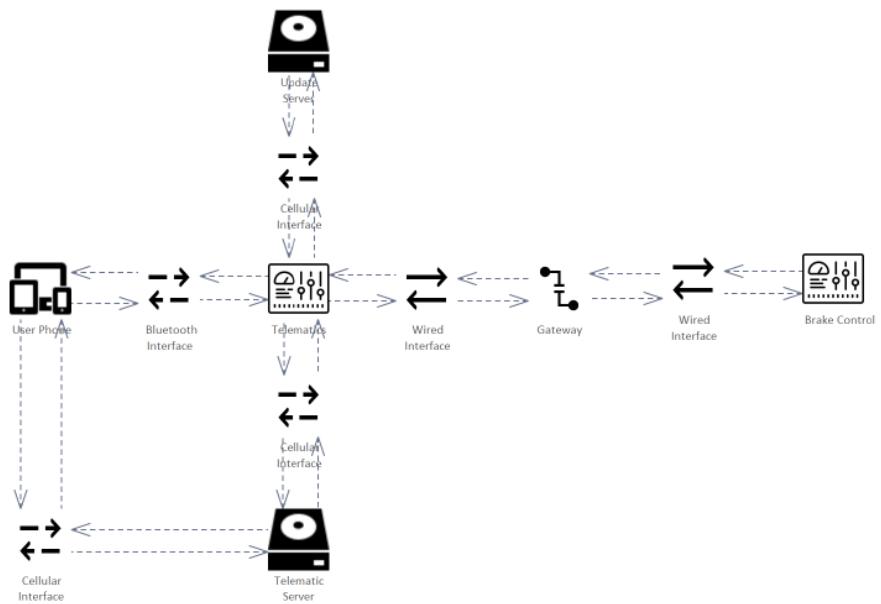
Ebben a munkában továbbá hasonlóan az én megoldásomhoz a STRIDE fenyegetés-modellezési keretrendszer valamint a CIA triad közös használata történik. Szintén érdemes kiemelni a fenyegetésmodellezésre jellemzően használt Data Flow diagram kiegészített formáját amelyet Extended Data Flow diagramnak neveznek. Ezzel egy részről kompozíciók modellezését teszik lehetővé, másrészről pedig értékek (asset) megjelenítéséről is gondoskodnak. Ez a modell már sokkal alkalmasabb komplex kiber-fizikai rendszerek elemzésére az általános Data Flow diagrammakkal szemben. Ez a diagram a 3.3 ábrán látható.

Ebben a kutatásban hangzik el először az automatizált támadási fa generálásának fogalma, valamint a tanulmány támadási gráfokat is definiál. Jól használja fel az Extended Data Flow Diagram rendszermodelljét támadási fák és gráfok generálására amelyekből támadási utakat vezet le amelyek alkalmasak lesznek valódi kockázatok meghatározására.

Magukról a támadási fákról alkotott modell pedig a 3.4 ábra mutatja be. Ezen nehézen látható, hogy az Extended Data Flow diagramból (ami a 3.3 ábrán látható) lenne származtatva a modell, valamint a végeredmény egy absztrakt megfogalmazású és nem kifejezetten az egyes komponensekre irányuló támadásokat jelenít meg a támadás lépése-

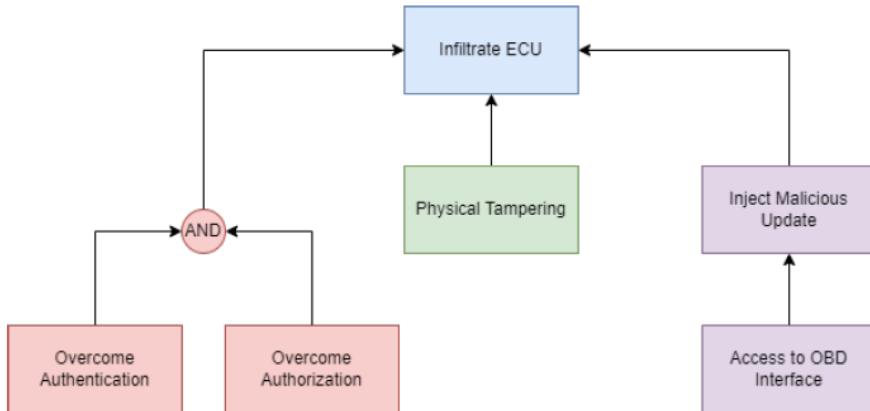


**3.2. ábra.** Fényezésmodellezés folyamata a tervezési fázisban[15]



**3.3. ábra.** Extended Data Flow diagram grafikus megjelenítése[15]

íként. Szintén a dependenciák is kevésbé használják ki a logikai kapuk nyújtotta felépítés lehetőségeit.



**3.4. ábra.** Támadási fák grafikus megjelenítése[15]

### 3.3. Támadási fa generálás

Sowka et al.[7] "*A review on automatic generation of attack trees and its application to automotive cybersecurity*" című publikációja különböző alkalmazásait értékelte az automatikus támadási fáknak az autóipari kiberbiztonság doménjében.

Az írás ad egy általános áttekintést a terület fontosságáról, a szabályozási környezet aktuális helyzetéről majd összehasonlította a különböző elérhető megoldásokat az adott problémára.

A Salfer et al.[12] "*Efficient attack forest construction for automotive on-board networks*" című munkája által bemutatott módszer egy magas fokú modellezett megoldás alapján való támadási utak előállítását határozza meg.

Kifejezetten érdekes, ahogy felépíti a metamodelljét amiben definiál egy rendszer és egy támadó modellt is. A rendszermodell meghatározza az elektronikus vezérlő egységeket, szoftvereket, kommunikációs hálózatokat és értékeket (*asset*), a támadó modell pedig tartalmazza a tudást, motivációt, amelyek aztán a támadások megvalósíthatóságának értékelésében játszanak szerepet.

Szintén elemzi ezeknek a támadási utaknak az alkalmazását a rendszer kiberbiztonsági (penetrációs) tesztelésénél és jól ismeri fel, hogy ez egy magasszintű white-box tesztelésben lehetne felhasználható.

Karray et al.[9] "*Cyber-security of connected vehicles: contributions to chance the risk analysis and security of in-vehicle communications*" kutatása sokban épít az előző bekezdésben említettre, azonban itt nem lehet egy explicit támadómodellről beszélni. A rendszermodell használ bizonyos tulajdonságokat amelyek jelzik a támadó szükséges tudását vagy belépés szükségét, viszont kevesebb feltevést használ a támadások meghatározásánál.

Végül pedig Bryans et al.[2] "*A template-based method for the generation of attack trees*" kutatását néztem meg amely kombinálja a modell alapú valamint a könyvtár alapú automatizált generálást, azaz a modell és template alapú támadási fa generálást. A Chulp et al.[15] kritikája alapján is ez volt kiemelve mint legérettebb megoldás, valamint ez is az egyik legfiatalabb. A támadó modell helyett előredefiniált támadási mintákat használ fel, amelyek segítségével rekurzívan tud a fa leveleiből kifejteni komplexebb támadásokat egyfajta bottom-up megközelítésben. A rendszermodell szemben a template-ekkel sokkal egyszerűbb, nem különít el ECU funkcionalitást vagy értékeket, emiatt ez a része nem lesz

használható a munkám során, viszont ez teszi lehetővé teszteléskor a black-box megközelítést, valamint akár teszt kódok és eszközök integrációjával is lehetne használni deszkriptív template-ek esetén.

## 4. fejezet

# A kiberbiztonsági analízis metodológiája

A diplomamunkám legfontosabb része a metodológia előállítása volt. A metodológia az ami biztosítja azoknak a céloknak az elérését, hogy az analízis (i) teljes körű legyen, (ii) megismételhető legyen és (iii) már létező információkra építsen, azon túl, hogy az eredménye és használata minél átláthatóbb legyen a stakeholderek és az elemző mérnök számára is.

Ez a metodológia kapcsolja össze az autóiparra jellemző rendszermodellek a fenyégetésmodellekkel. Ennek segítségére és támogatására készült el a kapcsolódó modellező eszköz és ennek az eredménye az egyik legfontosabb előállított értéke a kiberbiztonsági mérnök feladatkörnek.

Az említett fejezetben a példáimat egy tetszőleges személygépjármű kormányrendszerének az aktuációs (actuation: mozgásba hoz, működtet) funkcionálisára készítettem el.

Az Áttekintés fejezet tartalmaz egy magas szintű végigvezetést a bemenetektől a kiimenetig és a közte megtett lépésekéről.

A *Termékleírás és fenyégetésmodell származtatása* mutatja be a kiindulómodell elkezszítésének lépéseit valamint, hogy abból, hogyan állítjuk elő a fenyégetés modellt.

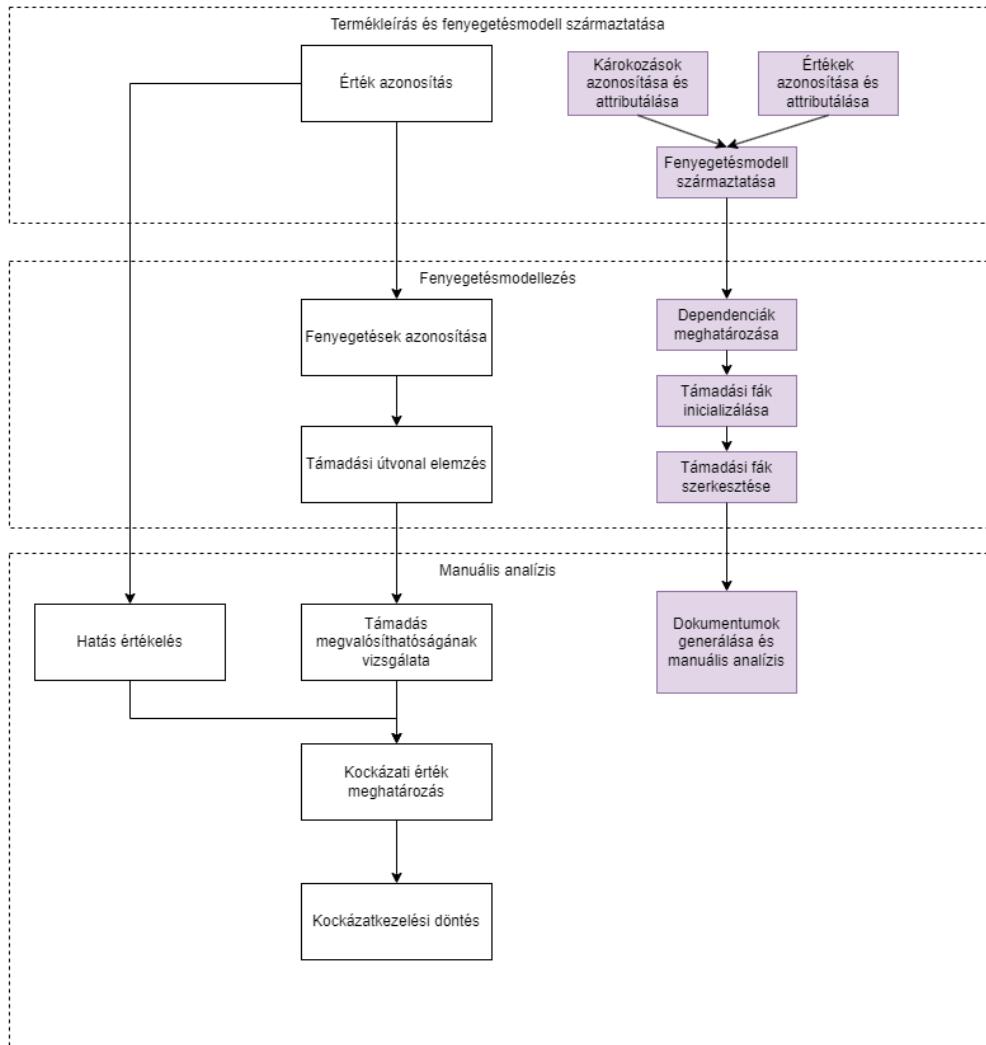
A *Fenyégetésmodellezés* fejezetben láthatóak a további lépések a fenyégetésmodell specifikálására, valamint bemutatja a fenyégetésmodellből előállított támadási fák konstrukcióját és annak szerkesztésének lépéseit.

A *Dokumentumok generálása és manuális analízis* fejezetben pedig találhatóak a szabvány által előírt kimenetek előállítása és az elemző eszköz használatát követő folyamatok.

### 4.1. Áttekintés

A metodológiám fő feladata a *Háttérismerekek* fejezetben található *Követelmények a fenyégetéselemzésre és kockázatértékelésre* részben leírtakat követve kialakítsam azt a lépéssorozatot amelyet az általam fejlesztett eszköz támogatásával végre lehet hajtani és el lehet jutni egy általános autóipari modellből a kockázatelemzés eredményéig.

A lépésekéről egy áttekintés a 4.1 ábrán látható. Itt egyrérszről a fehér hátterű négyzetekben az ISO 21434 által definiált lépések láthatóak amelyek bővebb leírása a *Háttérismerekek* részben található. Másrérszről pedig a lila hátterű négyzetekben az ebben a fejezetben bemutatott lépések láthatóak. Így adható egy egyszerűbb áttekintés a két módszertan közti kapcsolatról és azok közti fedettségről.



**4.1. ábra.** Metodológia áttekintése

## 4.2. Termékleírás és fenyelgetésmodell származtatása

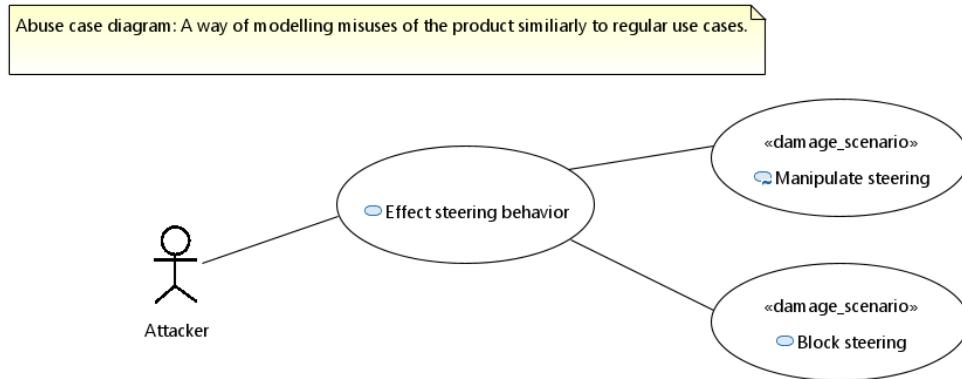
Ez a fejezet mutatja be az első lépéseit a kockázatelemzési folyamatnak. Itt lesz szükségünk a kiinduláskor rendelkezésünkre álló termékleírást (ami a rendszermodell egy részhalmaza) bővíteni kiberbiztonsági attribútumokkal majd abból származtatni egy olyan új modellt ami a kiberbiztonsági elemzésre alkalmas lesz.

Ehhez a rendszermodellünkben két diagram típusra lesz szükség. Az egyik a viselkedési diagramok közé sorolható használati eset (use case) diagram, a másik pedig egy strukturális kategóriába sorolható komponens diagram.

### 4.2.1. Károkozások azonosítása és attributálása

A használati eset diagramot a továbbiakban nevezzük *kihasználási eset* (*abuse case*) diagramnak. Ez annyiban módosítja az eredeti diagram használatát, hogy amíg a szintaktikailag és szemantikailag ugyanaz, tartalmilag nem a felhasználó és rendszer közti kapcsolatokat keressük, hanem a támadó lehetséges motivációját és amit ténylegesen meg is tud tenni a rendszerrel.

A kihasználási esetek azonosításában már tudjuk használni a CIA triát alkalmazását is. A bemutatott eljárásban, veszünk egy funkcionálitást amit a vizsgált termék ellát és azt finomítjuk tovább az egyes kiberbiztonsági tulajdonságoknak a sérülése szerint. Egy példa a 4.2 ábrán látható.



**4.2. ábra.** Példa kihasználási eset diagramra

A példán jól látható, hogy amíg az "Effect steering behavior" elvárt működésnek is tekinthető lenne egy kormányrendszer esetén, addig amikor ez egy támadó lehetséges céljai közé tartozik akkor már nevezhető ez egy kihasználási esetnek. Ezt tudjuk tovább finomítani aszerint, hogy mely kiberbiztonsági tulajdonság sérül.

Az eljárásban használt tulajdonságok a bizalmasság, sérтetlenség és elérhetőség, ezek bővebben a *Háttérismerekek* fejezetben kerültek bemutatásra.

A "Manipulate steering" esetében az aktuáció sérтetlenségi attribútuma sérül, a "Block steering" esetén pedig az elérhetősége. Bizalmassági tulajdonsága nincsen az aktuációtól, hiszen ennek a működtetése kívülről nem igényel semmilyen személyes adatot, szellemi terméket vagy kriptográfiai információt.

Az ISO 21434 szerint mivel ez a kihasználási eset (i) összeköt egy funkcionálitást egy nem kívánt következménnyel, valamint (ii) a funkcionálitáson keresztül más értékekhez is köthető lesz, emiatt jelölhetjük ezt egy *károkozásnak* (*damage scenario*).

#### 4.2.2. Értékek azonosítása és attributálása

A komponens diagram lesz, az ISO 21434 megnevezése szerint, a termékleírás (item definition). Ennek célja, hogy meghatározzuk a (i) termék határait, (ii) a termék funkcionálitását, illetve a (iii) kezdeti architektúrát.

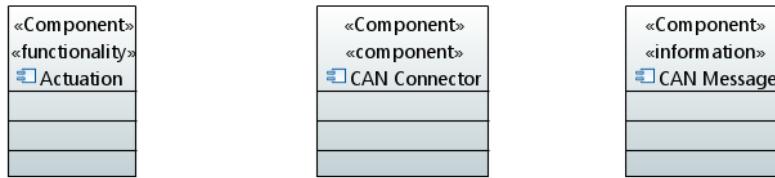
A termék határai, vagy kiberbiztonsági fogalommal támadási felületet (attack surface), a felvett HW komponensek formájában vagy a beérkező üzenetek (információk) formájában értékként vehetünk fel, a funkcionálitással egyetemben. Az előbbi kettő szintén analizálható a kiberbiztonsági tulajdonságaik alapján (pl: bizalmass információt szállít-e egy CAN üzenet), a funkcionálitásnál erre nincsen szükség hiszen a károkozásokból származtatható lesz egy funkcionálitás védendő tulajdonsága.

A kezdeti architektúra abban az értelemben van figyelembe véve, hogy a termékleírás maga egy komplexebb rendszermodell esetére is alkalmazható, ahol a rendszermodell egyes komponenseit jelöljük fel az előbbi három sztereotípia egyikével.

Erre példa a 4.3 ábrán látható.

A termékleírásban látható, hogy az Actuation mint funkcionálitás jelenik meg, a CAN csatlakozó mint HW komponens, a buszon szállított és a csatlakozón beérkező CAN üzenetek pedig mint információ.

**ISO21434-WP-09-01 Item definition:** Component diagrams are common in the automotive industry to conceptualize parts of the system in pre-development. As the TARA is a primarily concept phase activity we can use already existing component diagrams and add desired stereotypes to different components.



**4.3. ábra.** Példa termékleírásra

#### 4.2.3. Fenyelgetésmodell származtatása

Ebből a két diagram típusból, ami bármely autóipari termék rendszermodelljéhez könnyen integrálható, most már tudunk származtatni olyan modellt amely a fenyelgetésmodellezési eljárásunkhoz szükséges. Ehhez nincsen másra szükségünk mint, hogy készítsünk egy kivonatot az előbb feljelölt információkból, amelyet aztán a fenyelgetésmodellező eszközünk fel tud használni bemenetként.

Valamint fontos kiemelni az előző pontokban leírtak végrehajtásával egyben végrehajtottuk az ISO 21434 által leírt *Érték azonosítás* lépését is a kockázatelemzésnek.

### 4.3. Fenyelgetésmodellezés

Már az előző fejezetben leírtak is fenyelgetésmodellezésnek nevezhető, az még elsősorban a fenyelgetésmodell előkészítését és a rendszermodellből való származtatását végezte el.

A következő fejezetben leírtak mutatják be magát a kiberbiztonsági analízisnek a menetét, aminek az eredménye a manuális analízisre alkalmas információk előállítása.

#### 4.3.1. Dependenciák meghatározása

Első lépésként meg kell határoznunk a dependenciákat a károkozások, funkcionálitás és értékek között. Ezt egyszerűen megtehetjük, először a károkozásokat rendeljük hozzá egy-egy funkcionálitáshoz, majd pedig a funkcionálitásokhoz hozzárendeljük, hogy mely szoftveres információktól és mely hardver komponensektől függ.

Ennek a modellje a 4.4 ábrán látható.

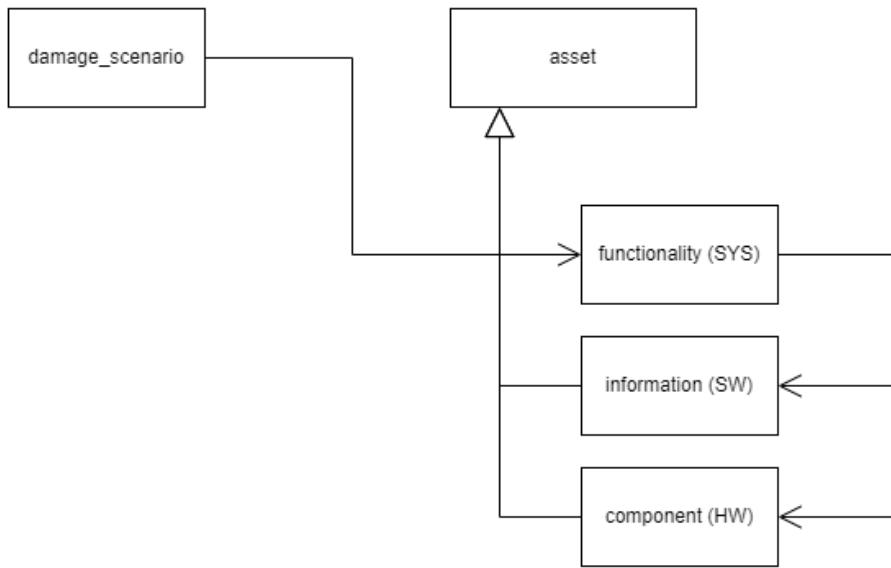
A korábbi példa esetében a két károkozásunk ("Manipulate steering", illetve "Block steering") az Actuation funkcionálitásnak a kártékony használatáról szólnak, ezzel már felvehető kapcsolat van a Károkozás és Funkcionálitás közt.

Ezután határozhatjuk meg a további dependenciákat, a funkcionálitás eléréséhez szükséges egyrésről egy CAN csatlakozóhoz való hozzáférés (ez történhet a CAN buszon keresztül más ECU-n keresztül, vagy állóhelyzetben, olyan kialakítás mellett akár egy saját eszköz csatlakoztatásával).

Másrészről pedig az aktuáció az alapján fog történni, hogy a kormányrendszer milyen üzeneteket kap a CAN busról. Ilyen befolyásoló tényező lehet a jármű sebessége például.

#### 4.3.2. Támadási fák inicializálása

A definiált következő lépése a szabványos kockázatelemzésnek a *Fenyelgetések azonosítása* és a *Támadási útvonal elemzés*.



**4.4. ábra.** Károkozások, funkcionalitás és értékek összerendelése

Az előbbi elvégzésére a STRIDE keretrendszer fogom alkalmazni, amely az egyes fenyelgetés típusokat rendeli a kiberbiztonsági tulajdonságokhoz, a keretrendszer bemutatása a *Háttérismerekek* fejezetben található.

Az utóbbinak pedig ezzel együtt el tudjuk végezni az előkészítését, ahol a generált fenyelgetéseket a meghatározott dependenciák szerint rendezzük.

#### 4.3.3. Támadási fák szerkesztése

A támadási fák olyan irányított aciklikus gráfok amelyekben minden páratlan szinten egy fenyelgetés vagy pszeudo-fenyelgetés található, minden páros szintjén pedig egy logikai kapu amely lehet *ÉS* vagy *VAGY* típusú. A gyökér fenyelgetést nevezhetjük rendszerszintű fenyelgetésnek, a levélben lévőket pedig szoftver- vagy hardverszintű fenyelgetésnek. Ezek közé pszeudo-fenyelgetéseket helyezhetünk el, amelyek csoportosíthatnak szoftver- és/vagy hardverszintű fenyelgetéseket.

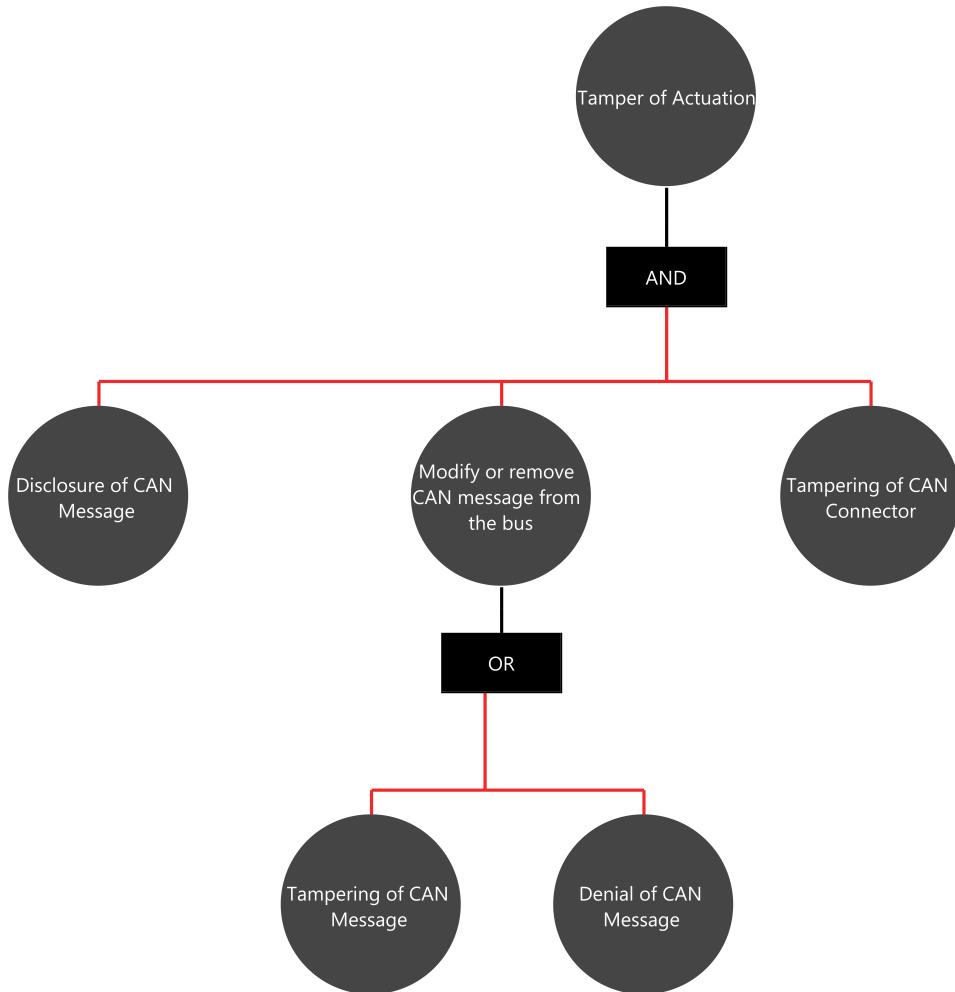
További korlátozás a gráffal szemben, hogy minden fenyelgetésnek 0 vagy egy kapuja lehet mint gyermek elem, illetve egy kapunak 1 vagy több gyermek fenyelgetése lehet. A fenyelgetés felfele nulla vagy egy kapuhoz csatlakozhat (ez már következik a gráf aciklikusságából), a kapunak felfele pedig pontosan egy fenyelgetése kell, hogy legyen. Ebből következik, hogy egy fenyelgetés gyökérnek számít, ha nincs felette kapu, pszeudónak ha felette és alatta is van kapu, levélnek pedig, ha nincs alatta kapu.

Ezeknek a fáknak a konstrukcióját úgy végezzük el, hogy minden károkozáshoz egy támadási fát rendelünk, annak a gyökér eleme a hozzá rendelt funkcionalitás valamint a károkozásban sérült tulajdonság alapján meghatározott fenyelgetés. A levél elemeket szintén a funkcionalitáshoz rendelt értékekből és azok tulajdonságaiból kaphatjuk meg.

Ezután adhatunk hozzá a fához kapukat és határozhatjuk meg az összefüggéseket a szoftver- és hardverszintű fenyelgetések valamint a rendszerszintű fenyelgetés között.

Egy példa erre a fejlesztett elemző eszközben a 4.5 ábrán láthatunk.

Itt látható, hogy a támadási végrehajtásához szükséges egyrésről a CAN üzenet tartalmának megismerése, ez alapján tud a támadó saját üzeneteket konstruálni, megtalálni a címezhető ECU-kat, stb.



**4.5. ábra.** Példa a támadási fára a "Manipulate steering" károkozáshoz

Másrészről szükséges a CAN csatlakozó ellen a Tampering, ami a sértetlenség tulajdonság sértése. Erre példa, hogy amennyiben módosítani akarjuk a kerekek állását, azt abban az esetben befolyásolhatjuk, ha van hozzáférésünk a CAN buszhöz amire az ECU csatlakoztatva van.

Utolsó sorban pedig a CAN üzenetnek módosíthatjuk a tartalmát vagy annak a továbbítását tudjuk blokkolni bizonyos esetekben, ezek közül bármelyik fenyegetés jelenléte a rendszerünkben okozhatja a rendszerszintű fenyegetés jelenlétét is.

Ennek a támadási fa konstrukciónak az elvégzése minden károkozásra elegendő ahhoz, hogy teljesítsük a *Támadási útvonal elemzés* lépését a szabványosított eljárásnak.

#### 4.4. Dokumentumok generálása és manuális analízis

Jelenleg a fenyegetésmodellünk tartalmaz információt a rendszerre releváns károkozásokról, funkcionálitásról, értékekről, fenyegetésekéről valamint a fákból származtatható a támadási útvonalakról.

Ennek a modellnek a segítségével fogunk tudni elvégezni minden analízist amely, pusztán a jellegéből fakadóan, már manuális analízis elvégzését igényli.

Az egyik ilyen a hatás értékelés, ez a károkozások listázásával előállítható. A másik a támadás megvalósíthatóságának a vizsgálata, amelyben a támadási útvonalak és támadási lépésekkel kell kiértékelni a megvalósíthatóságuk szerint.

A támadási útvonal egy minimális halmaza a levéleseményeknek amelyeknek a teljesülése (rendszerben egy időben való jelenléte) vezet a gyökér esemény, vagy rendszerszintű fenyelgetés jelenlétéhez. A támadási lépések az egyes levél fenyelgetések lesznek.

Példa a hatásérték analízisre a 4.1 táblázatban látható, a támadási megvalósíthatóság elemzésre pedig a 4.2 táblázatban.

A táblázat oszlopai a határérték analízis esetén az SFOP keretrendszert használják, a támadási megvalósíthatóságnál pedig az Attack Potential Based megközelítést.

Damage Scenario	Safety	Financial	Operational	Privacy	Impact
Manipulate steering					
Block steering					

**4.1. táblázat.** Generált hatásérték analízis dokumentum

Attack Paths	ET	SE	KoI	WoO	Eq	Attack Feasibility
Attack Path - Manipulate steering						
-> Tampering of CAN Message						
-> Tampering of CAN Connector						
-> Disclosure of CAN Message						
Attack Path - Manipulate steering						
-> Denial of CAN Message						
-> Tampering of CAN Connector						
-> Disclosure of CAN Message						

**4.2. táblázat.** Generált támadás megvalósíthatósági elemzés dokumentum

Ezek a táblázatok már alkalmaskak a megfelelő szakértők bevonásával az elemzés elvégzésére. Ezen eredményéből már meghatározható a kockázati érték és a kockázatkezelési döntés a stakeholder-ek által.

## 5. fejezet

# A kiberbiztonsági analízis megvalósítása

A következő fejezetnek a célja, hogy bemutassa az előző fejezetek által körülírt metodológia végrehajtását segítő fenyelgetésmodellező eszközt, valamint a rendszermodellek származtatására készített szkriptet.

Az első alfejezetben látható egy áttekintés majd két fő részre osztva láthatóak a további alfejezetek. A két fő rész egyike a rendszerből kiberbiztonsági modell transzformálást mutatja be és az ezt segítő eszközököt. A másik pedig az EMF alapon implementált modellező eszközt amely a kiberbiztonsági modell alapján képes a manuális analízist támogatni dokumentumok generálásával valamint a támadási fák szerkesztésével.

### 5.1. Áttekintés

Egy áttekintése az elemző eszköz részeinek megtekinthető a 4.1 ábrán.

Itt először kiemelném a fenyelgetésmodell és a rendszermodell közös halmazában lévő tartalmakat. Ez a rendszermodell egy Papyrus projekt formájában készült el. Tartalmazza a *Kihasználási eset diagramot* és a *Termékleírást*. Ezek tartalmazzák az értékeket, funkcionálitást és a lehetséges károkozásokat, amelyekből két dokumentum állítható majd elő, amelyek három ISO 21434 workproduct-ot fednek.

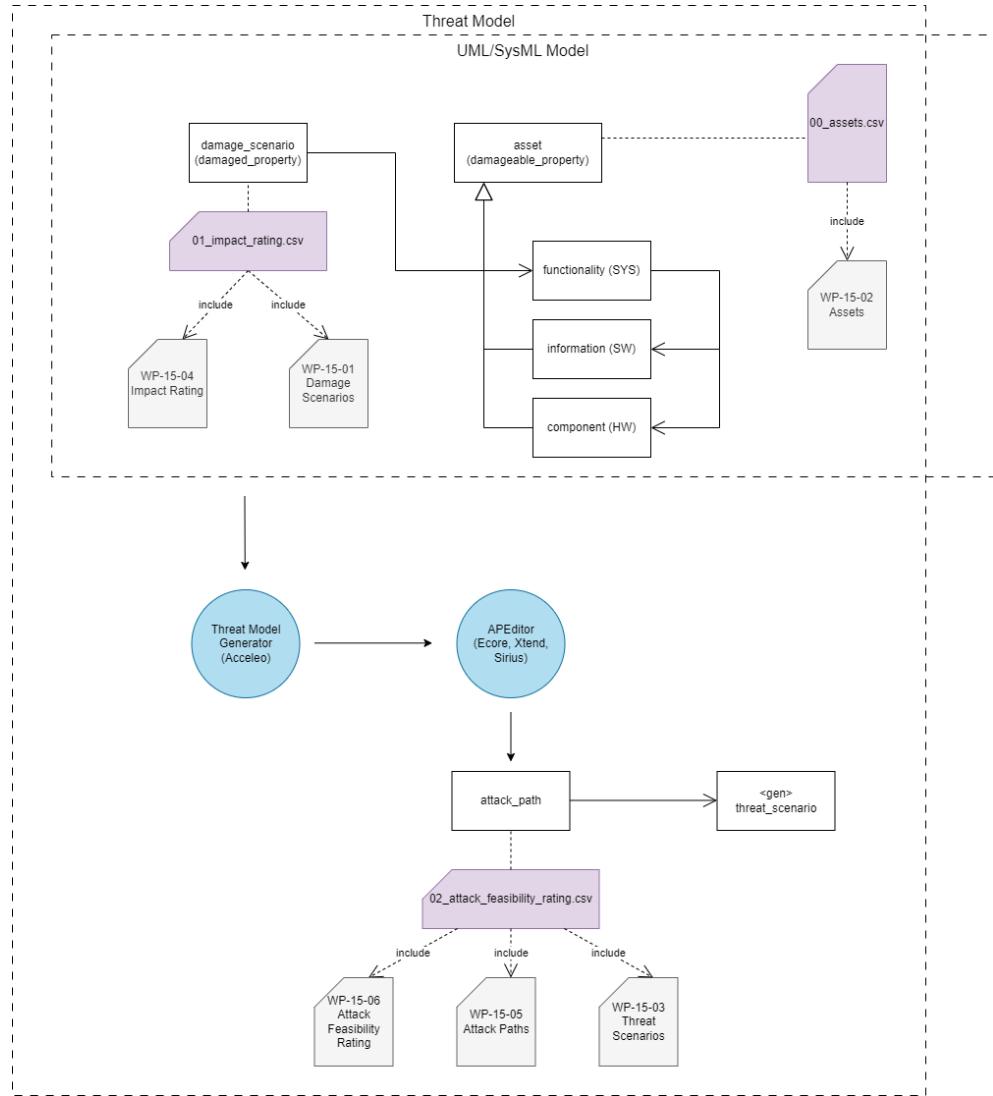
A következő elem a Threat Model Generator ami egy Acceleo script a Papyrus modellező eszközökhöz integrálva és a cybersecurity profil alapján állít elő egy kezdeti modell fájlt amelyet a fenyelgetésmodellező eszközzel fogunk tudni megnyitni.

A fenyelgetésmodellező eszköz az APEditor (Attack Path Editor) amelyben lehetőséünk van a termékleírás és kihasználási diagram tartalmai közti dependenciák meghatározására, valamint támadási fák és fenyelgetések automatikus származtatására.

A támadási fák elkészítése után fogjuk tudni a modellből származtatni a támadás megvalósíthatóság értékelés dokumentumot ami további három ISO 21434 workproductot fed le.

### 5.2. Kiberbiztonsági modell származtatása

A kiberbiztonsági modell származtatása egy fontos lépése a kockázatelemzési folyamatnak. Előkészíti a fenyelgetésmodellünket illetve teremt egyfajta nyomon követhetőséget a rendszermodell és a kiberbiztonsági modell között.

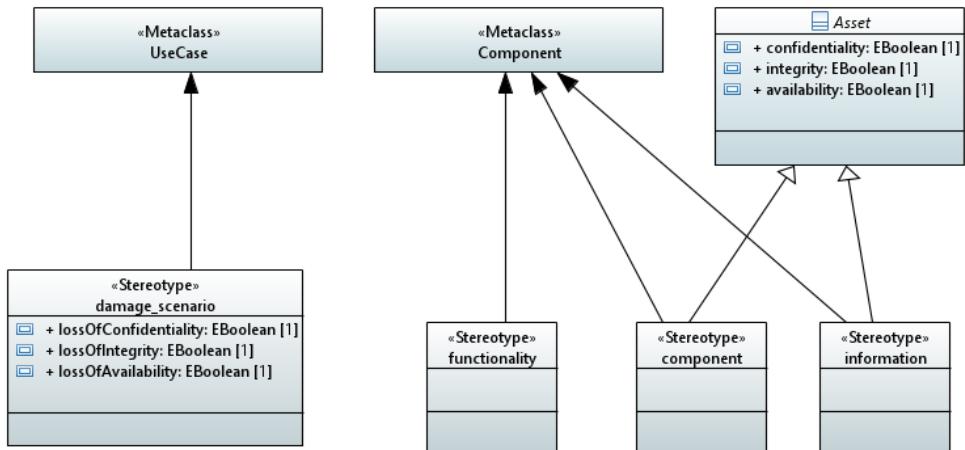


**5.1. ábra.** Megvalósítás áttekintése

### 5.2.1. Kiberbiztonsági profil

A profilban négy sztereotípia került meghatározásra. Egyrészről a UseCase UML metaosztályát bővíti a damage\_scenario sztereotípia amellyel a károkozásokat tudjuk jelölni. A károkozásokhoz fel lett véve atttribútumként, hogy azt mely kiberbiztonsági tulajdonság sérülése okozta.

A Component UML metaosztályt bővíti a functionality, component illetve az information sztereotípia. Itt a functionality jelenti a rendszerszintű funkcionalitást, ennek nincsenek az UML profilban attribútumai mivel azokat majd a kapcsolódó károkozások alapján fogjuk tudni meghatározni az elemzés későbbi fázisában. A component jelzi a HW komponenseket (pl. csatlakozó, modulok, áramkörök), az information pedig a SW szintű információkat (pl. szoftver, kriptográfiai adatok, üzenetek). Az utóbbi kettő az Asset (érték) absztrakt osztályból van származtatva, ebből öröklik a kiberbiztonsági tulajdonságaikat.

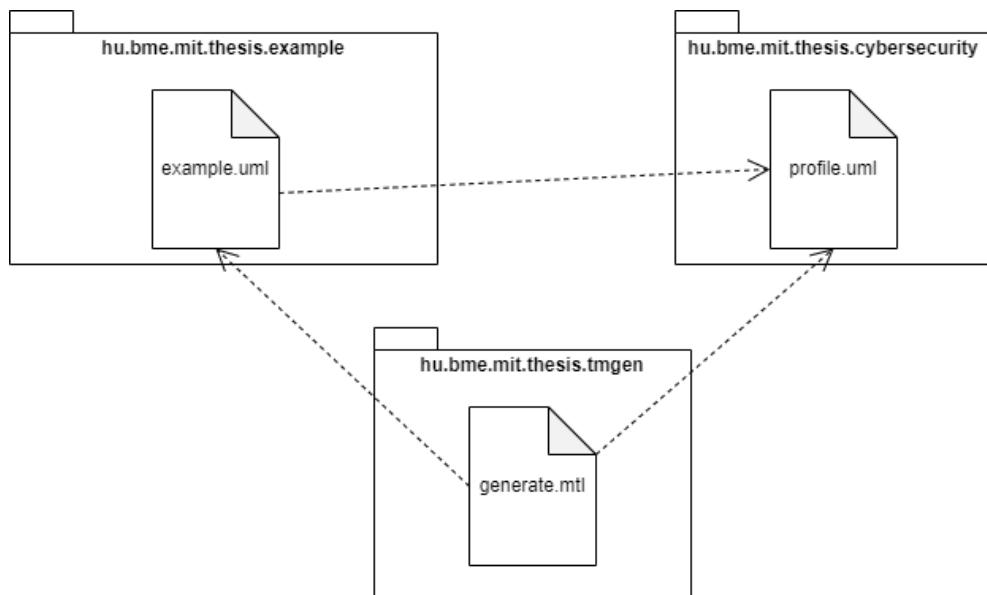


**5.2. ábra.** Kiberbiztonsági profil (UML)

### 5.2.2. Fenytegetésmodell generátor

A fenytegetésmodell generátor egy Acceleo nyelven írt szkript, amelynek a feladata a Papyrus-ban szerkesztett .uml fájloknak a beolvasása és a tartalmuk alapján egy .apeditor fájlnak a generálása, amelyet a fenytegetésmodellező eszközben lehet tovább szerkeszteni.

A projekt felépítése és a dependenciák a 5.3 ábrán láthatóak.



**5.3. ábra.** Threat Model Generator dependenciái

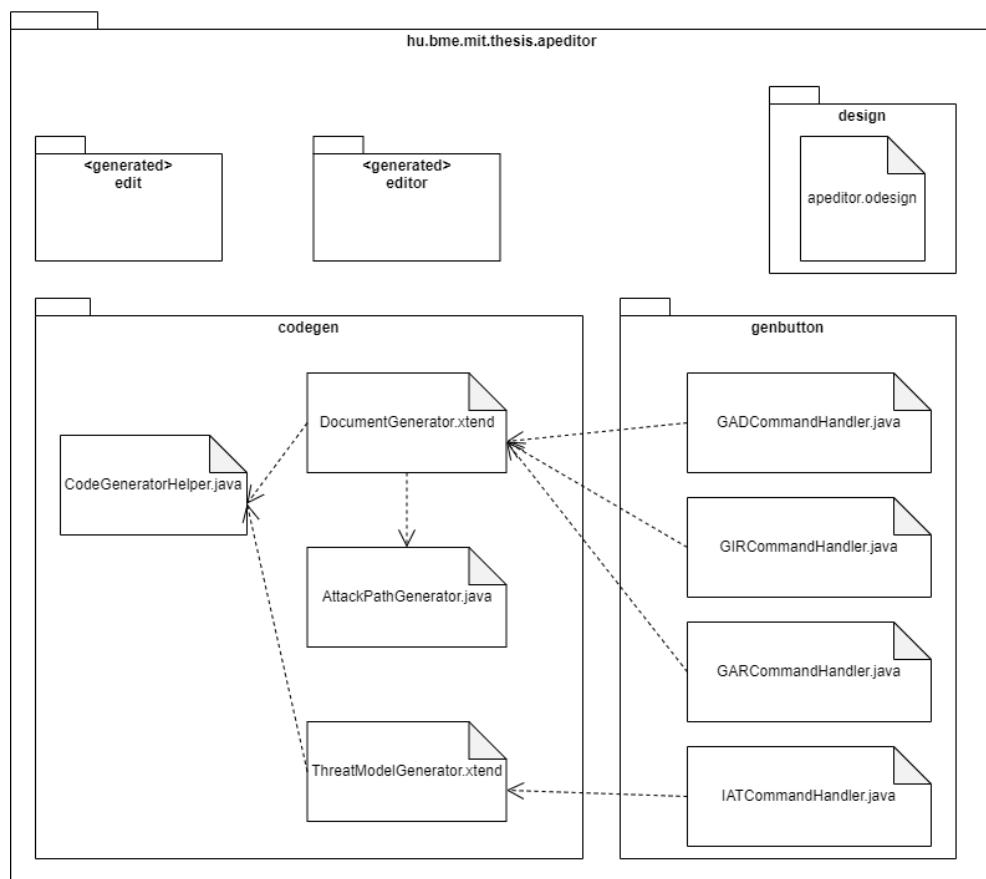
A **hu.bme.mit.thesis.example** projekt tartalmazza a kihasználási eset diagramot és a termékleírást, valamint applikálja a **hu.bme.mit.thesis.cybersecurity** projektben definiált kiberbiztonsági profil sztereotípiáit.

A **hu.bme.mit.thesis.tmgen** tartalmazza a **generate.mtl** fájlt ami egy Acceleo nyelven írt Model-To-Text generátor. Ez fogja előállítani a projekt *generated* mappájába az **example.apeditor** fájlt, amelyet át lehet majd másolni a fenytegetésmodellező eszköz *runtime* környezetében létrehozott projektekbe.

### 5.3. Fenyedegetésmodellező eszköz

A fenyedegetésmodellező eszköz teszi lehetővé a károkozások, funkcionálitások és értékek közti dependencia beállítását, a támadási fák szerkesztését, illetve a szabványos dokumentumok generálását.

Ennek a projektnek az áttekintése a 5.4 ábrán látható.



5.4. ábra. APEditor projekt felépítése és dependenciái

A **hu.bme.mit.thesis.apeditor** projektben található az *apeditor.ecore* fájl ami a metamodellt tartalmazza, illetve az Eclipse Modelling Framework által generált fájlokat.

A **hu.bme.mit.thesis.apeditor.edit** illetve a **hu.bme.mit.thesis.apeditor.editor** projektek szintén az EMF által lettek generálva, ezek adják meg a modell szerkesztő felületének az alapjait.

A **hu.bme.mit.thesis.apeditor.design** tartalmaz egy Sirius keretrendszer használó *apeditor.odesign* fájlt amely a támadási fáknak a grafikus megjelenítését és szerkesztő felületét írja le.

A **hu.bme.mit.thesis.apeditor.genbutton** tartalmazza a plugin.xml-t amelyben az APEditor felhasználói felületén megjelenítendő gombok vannak leírva, valamint a gombok megnyomásával futtatott .java fájl is itt kerül összelinkelésre. A négy .java fájl a különböző funkcionálitását definiálják.

- **GADCommandHandler.java** Asset Definition workproduct generálását viszi végez, hez a **hu.bme.mit.thesis.apeditor.codegen** projektben definiált funkcionálitás aktiválásával

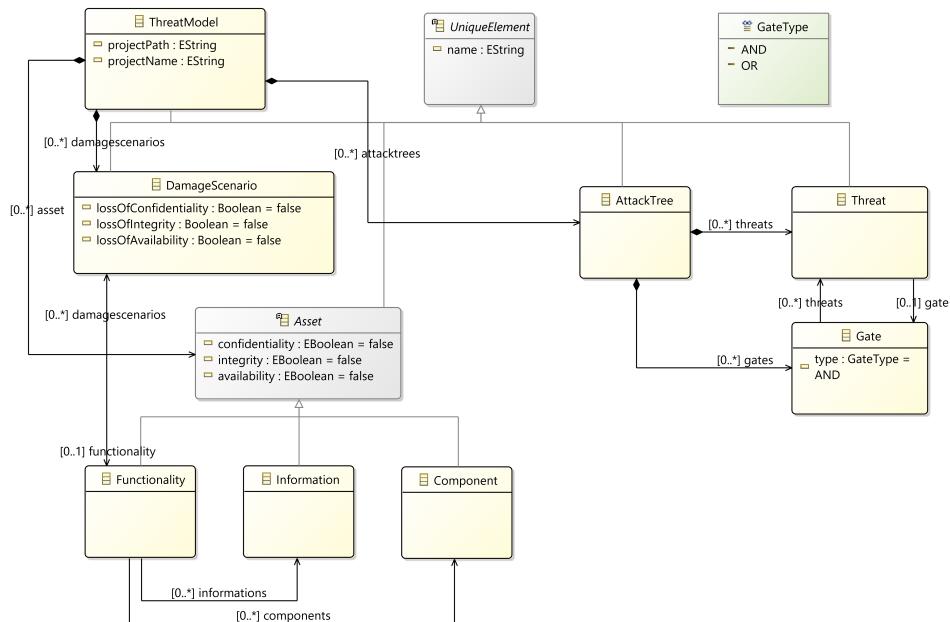
- **GIRCommandHandler.java** Impact Rating workproduct generálását viszi véghez a **hu.bme.mit.thesis.apeditor.codegen** projektben definiált funkcionális aktiválásával
- **GARCommandHandler.java** Attack Feasibility Rating workproduct generálását viszi véghez a **hu.bme.mit.thesis.apeditor.codegen** projektben definiált funkcionális aktiválásával
- **IATCommandHandler.java** Inicializálja a támadási fákat az **hu.bme.mit.thesis.apeditor.codegen** projektben definiált funkcionális aktiválásával

Végül az **hu.bme.mit.thesis.apeditor.codegen** tartalmazza a kódgeneráláshoz szükséges osztályokat és függvényeket. Itt egyrészről a *DocumentGenerator.xtend* fájl tartalmazza azokat az Xtend-ben megírt függvényeket, amelyek a szabványos workproduct-ok előállításához szükségesek, másrészt az Attack Feasibility Rating generálásakor, használja az *AttackPathGenerator.java* függvényeit amelyek a támadási fákat fejti ki támadási útvonalakká amelyek támadási lépésekkel állnak.

Az *ThreatModelGenerator.xtend* a modellből generálja le a támadási fák kezdeti állapotát és aztán azokat írja bele a modellfájlba. Ezzel bővíti a szerkesztett modellt.

Mindkét Xtend fájl a *CodeGeneratorHelper.java* fájlt használja még a fájlba író és fájlt olvasó műveletek elvégzésére.

### 5.3.1. Metamodel



5.5. ábra. apeditor.ecore fájlban definiált metamodel

A metamodel az ISO 21434 szabványban definiált különböző elemeket és megnevezéseket használja. Az elemzésben felhasznált elemek között a szabvány alapján határoztam meg a kapcsolati függőségeket és ezeket tartalmazza a *ThreatModel* objektum.

Az előzetes fenyergetésmóddal származtatása fogja előállítani a **DamageScenario** és **Asset** listákat amelyeket fel is vesz a modellből generált **ThreatModel** objektum alá.

A generált modellben minden **DamageScenario** összerendelhető lesz egy **Functionality**-vel, ez a rendszerszintű **Asset**.

A dependenciákat a szintén generált **Information** (SW-szintű érték) és **Component** (HW-szintű érték) objektumoknak a **Functionality**-hez való rendelésével tettem lehetővé.

A támadási fáknak az inicializálása egy egy **AttackTree** létrehozásával történik az egyes **DamageScenario**-khoz. A **Threat**-ek az alapján vannak generálva, hogy a **DamageScenario**-hoz tartozó **Asset**-ek milyen kiberbiztonsági tulajdonságokkal rendelkeznek. A **Gate** objektumok pedig manuálisan kerülnek létrehozásra a támadási fa szerkesztő felületen.

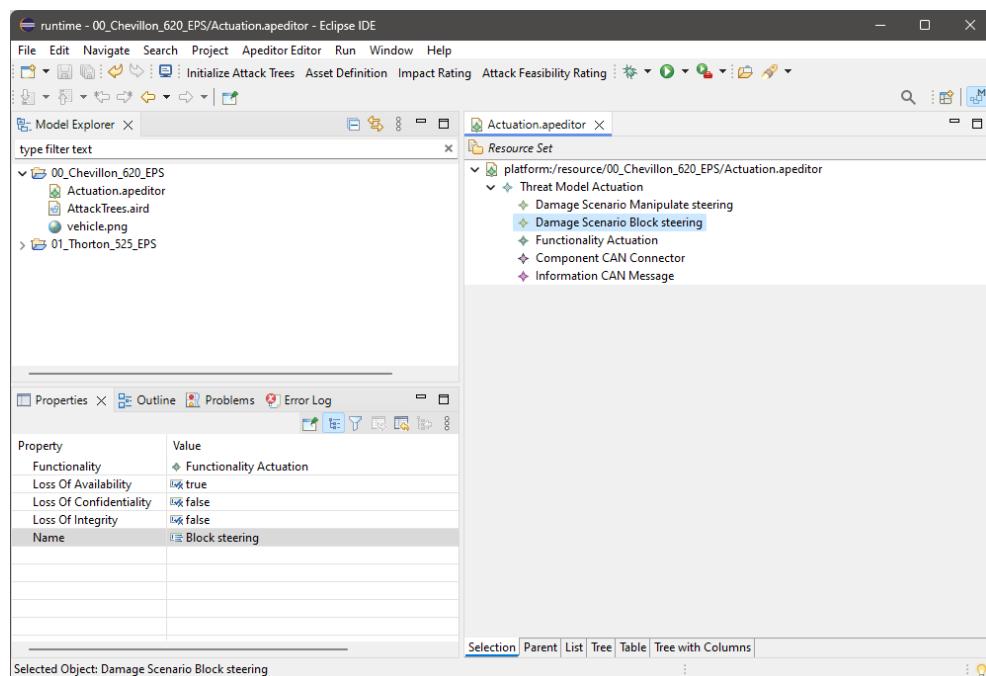
Továbbá a **GateType** enumeráció teszi lehetővé a kapuk, a **UniqueElement** pedig az összes egyedi elem megkülönböztetését.

A **ThreatModel** rendelkezik egy `projectPath` és  `projectName` property-vel, ezeket az értékeket a dokumentumok generálásához használjuk.

### 5.3.2. Fenyedegetésmo dell szerkesztő felület

A fenyedegetésmo dell szerkesztő felülete szolgál arra, hogy a dependenciákat meghatározza a SW- és HW-szintű értékek, valamint a funkcionalitás között, illetve, hogy allokáljuk a károkozásokhoz a funkcionalitást.

Ez a felület az Ecore modellből Eclipse Modelling Framework segítségével kerül generálásra. Megtekinthető a 5.6 ábrán.



**5.6. ábra.** Fenyedegetésmo dell szerkesztő felület

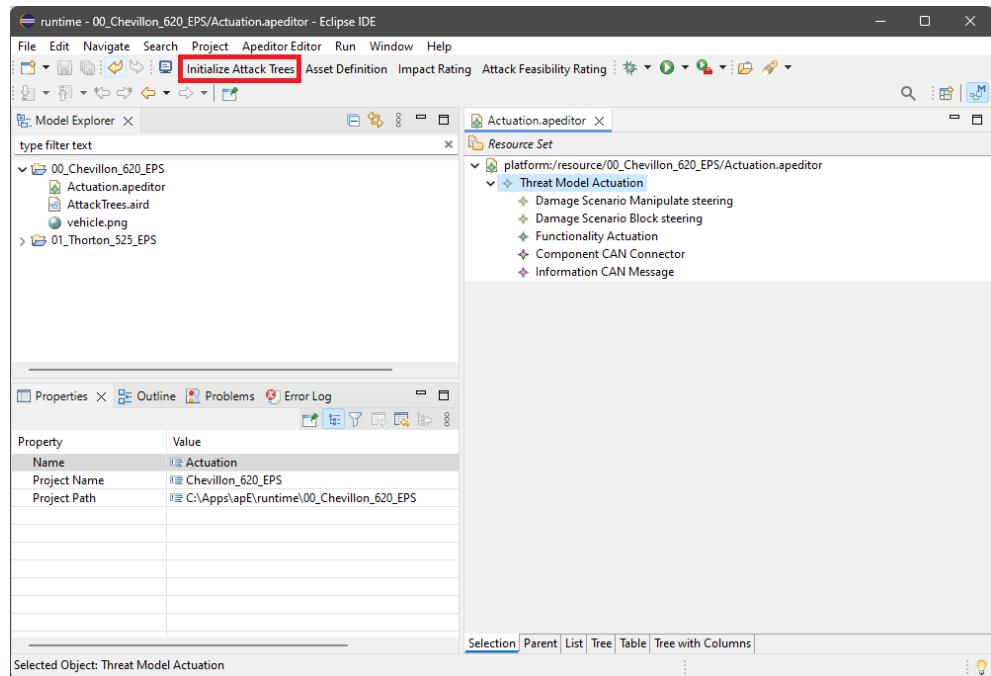
Az ábrán látható a Model Explorer-ben az éppen megnyitott projekt, valamint az abban lévő fenyedegetésmo dellek a .apeditor fájlkiterjesztésekkel, illetve a támadási fa diagramok lesznek láthatóak a .aird fájl megnyitásával.

A jobb oldalon lehet böngészni és felvenni új elemeket a fenyedegetésmo dellbe, illetve itt kell kiválasztani a szerkesztendő elemet.

Bal alul pedig a properties fünlél látjuk a metamodelben definiált attribútumokat és szerkeszthetjük a felvett értékeket (value).

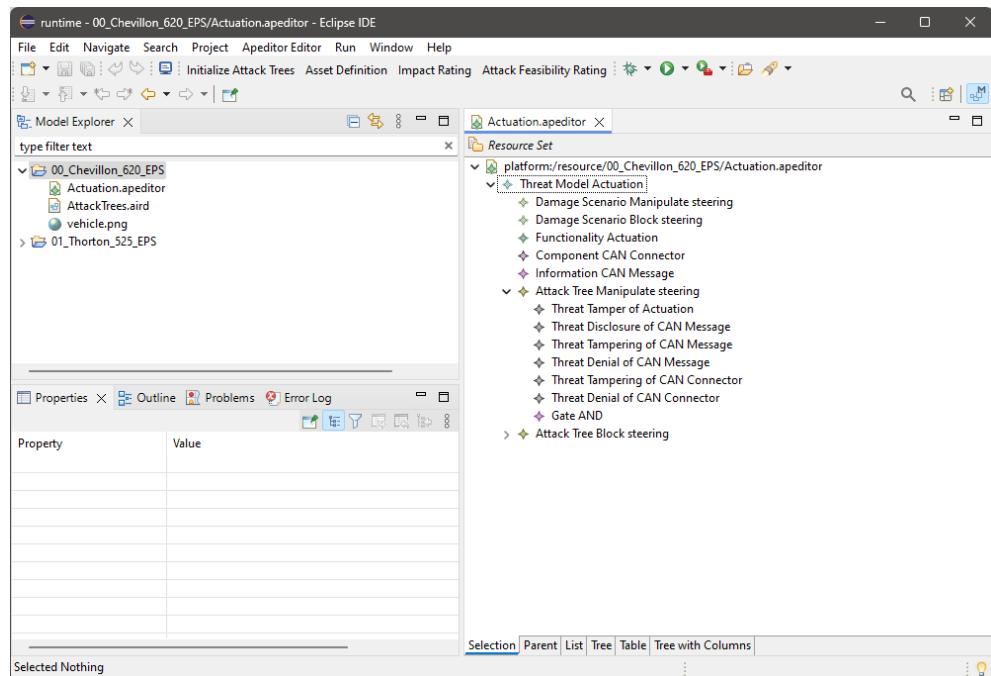
### 5.3.3. Támadási fák inicializálása

A fenyegetésmoell kiválasztásával, a projekt útvonal és nevek beállítását követően el is végezhetjük a támadási fák inicializálását az 5.7 ábrán látható gomb megnyomásával.



**5.7. ábra.** Támadási fák inicializálását végző kiterjesztés

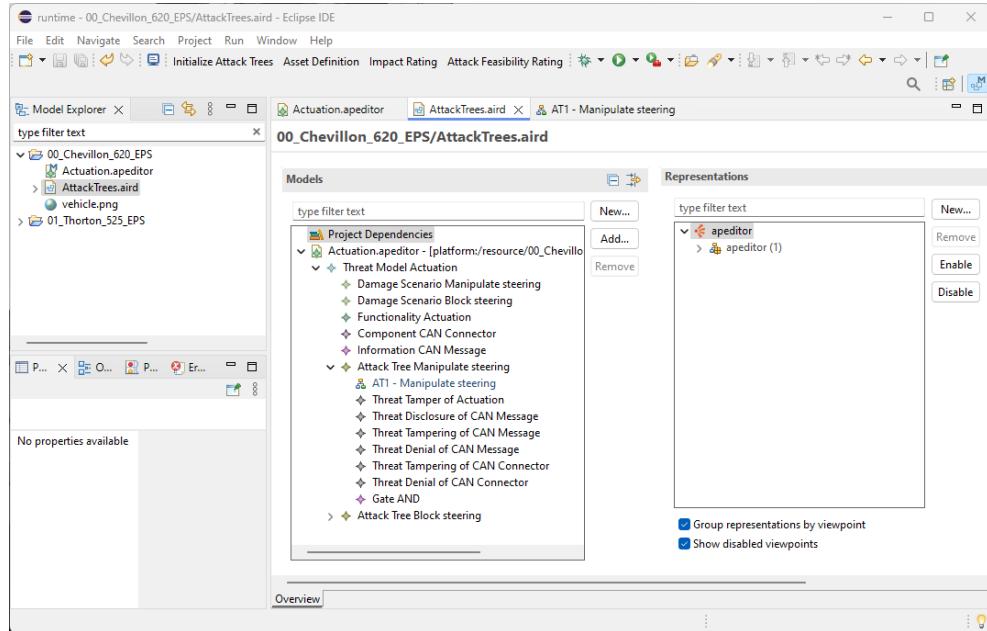
Ez a kiterjesztés fogja használni a kódgenerátor plugin **ThreatModelGenerator.xtend** osztályában implementált funkcionalitást meghívni, amely a projekt útvonalon található fenyegetésmoell fájlt feltölti a tartalma alapján előálló támadási fákkal. Az inicializálás eredménye a 5.8 ábrán látható.



**5.8. ábra.** Támadási fák inicializálásának eredménye

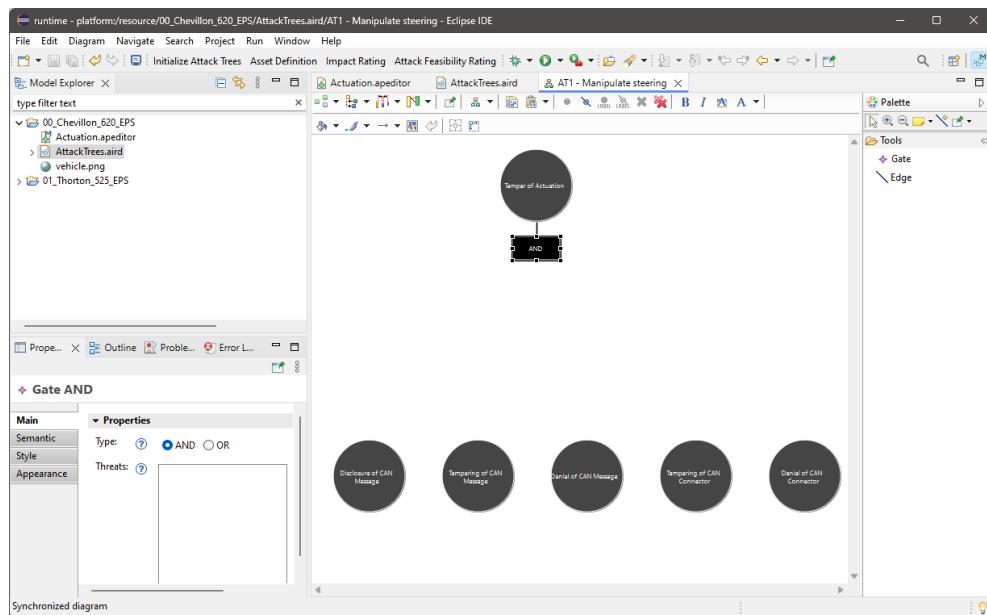
### 5.3.4. Támadási fa szerkesztő felület

A szerkesztő felületet az Eclipse Sirius keretrendszerére alapozva készítettem el. A projektben létrehozott .aird fájakra kattintva lehet a támadási fákhoz új reprezentációt létrehozni, ennek az eredménye a 5.9 ábrán látható.



5.9. ábra. Reprezentáció létrehozása

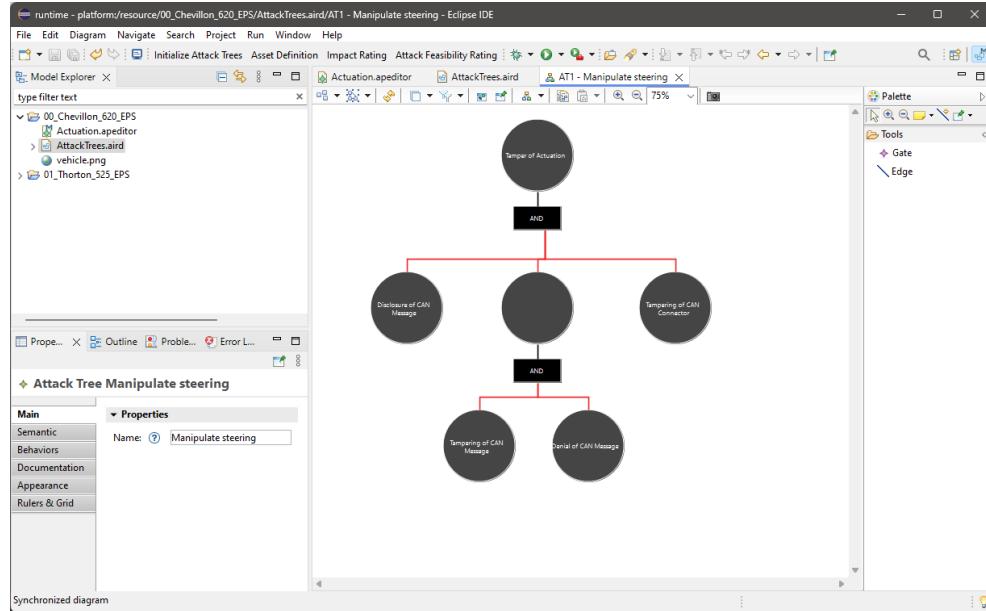
A létrehozás automatikusan meg is fogja nyitni a szerkesztőfelületet (5.10 ábra) amelyen megfigyelhető a gyökér elemhez automatikusan létrehozott első kapu, amelynek típusa a properties fülön állítható.



5.10. ábra. Támadási fa szerkesztő felület (kezdeti)

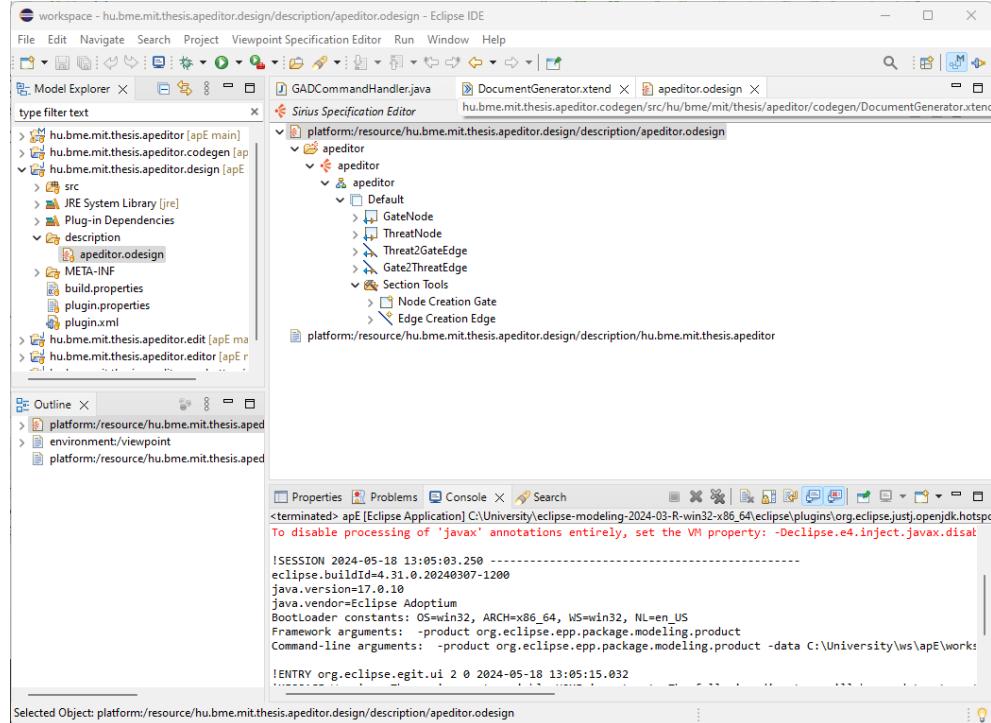
A jobb oldalt a Tools menü alatt találhatóak az új modellelemek létrehozásához szükséges eszközök. Potenciálisan vagy új kapukat fogunk létrehozni vagy a kapuk bemenetére

kötünk fenyegetéseket. Szintén ilyenkor van lehetőség törölni olyan fenyegetéseket amelyek nem relevánsak a támadási fához. Egy kész hibafa megtekinthető a 5.11 ábrán.



**5.11. ábra.** Támadási fa szerkesztő felület (befejezett)

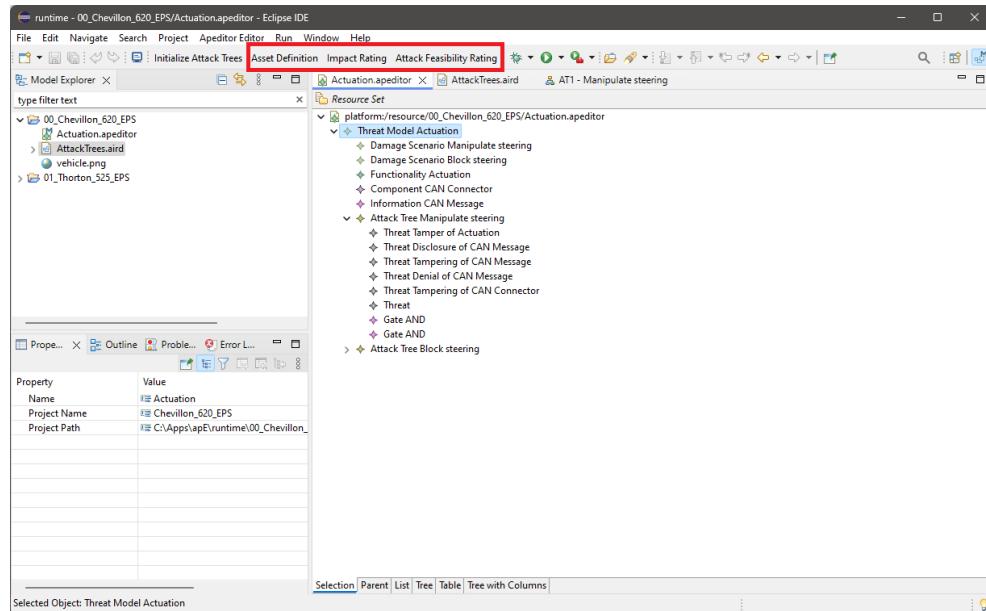
A Sirius leírófájl benne a grafikusan megjelenítendő elemekkel pedig a 5.12 ábrán látható.



**5.12. ábra.** Támadási fa szerkesztő leírófájlja

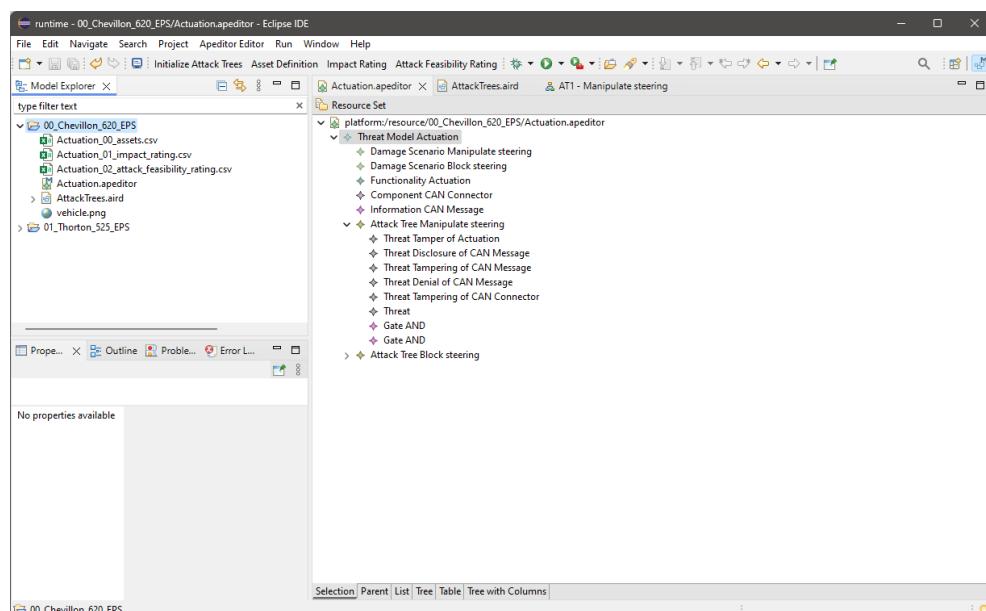
### 5.3.5. Dokumentum generátor

A dokumentumgenerálást szintén egy plugin-ként definiált gomb megnyomásával lehet elvégezni, miután a mérnök kiválasztotta a fenyelőmodellt és beállította helyesen a projekt útvonalat és neveket.



**5.13. ábra.** Dokumentumok generálását végző kiterjesztés

A generálás elvégzése után a projekt alatt megjelennek a manuális analízis elvégzéséhez használható .csv fájlok.



**5.14. ábra.** Generált dokumentumok

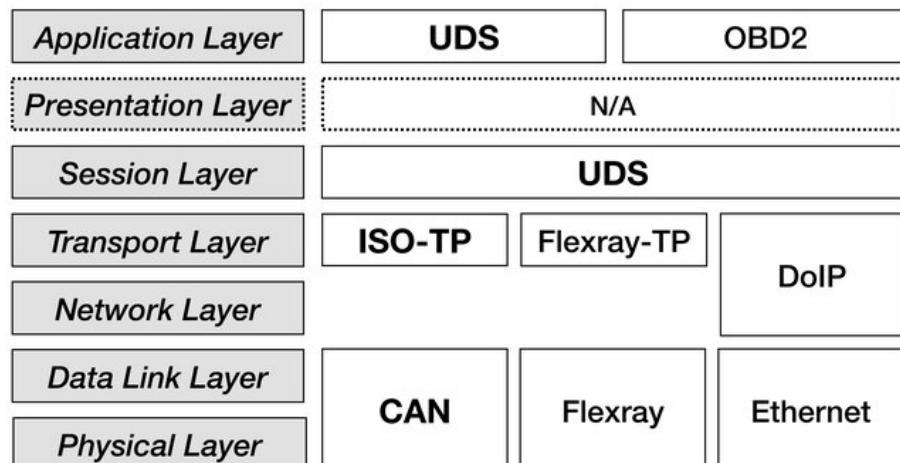
## 6. fejezet

# Esettanulmány

Az esettanulmányomnak a célja, hogy egy példán keresztül bemutassam az általam fejlesztett elemző eszköz valamint a rendszermodellből való származtatásnak a folyamatát.

Az esettanulmányom tárgyaként az elektronikus vezérlő egységek (ECU) diagnosztikai szolgáltatását választottam. Ennek oka egyrészről az, hogy az elemzett diagnosztikai szolgáltatások specifikációja publikusan elérhető az ISO 14229 "Road vehicles - Unified Diagnostic Services (UDS)" [5] szabványban. Másrészről ez egy olyan potenciális belépőpontja lehet a támadóknak amely nem igényel semmiféle fizikai hozzáférést a járműhöz. Amennyiben távolról sikeresen hozzáfér és átveszi az irányítását a támadó egy elektronikus vezérlő egységnek, ami képes V2X vagy más vezetéknélküli szolgáltatásra, onnantól kezdve a jármű többi részén lévő diagnosztikai szolgáltatások is elérhetővé válhatnak számára.

Az ISO 14229 [5] definiálja az Unified Diagnostic Services (röviden UDS) protokollt ami egy alkalmazásrétegbeli szolgáltatás. Általános IT biztonsági szempontból, ehhez legközelebbinek a TLS és HTTPS protokollok lennének mondhatók.



6.1. ábra. ISO/OSI modell autóipari kommunikációs protokollokra [10]

Az esettanulmányomban a szolgáltatások közül kettővel, a diagnosztikai autentikációval, valamint a software frissítéssel fogok foglalkozni, mivel ezek a kiemelten kiberbiztonsági relevanciával rendelkező szolgáltatások.

A diagnosztikai autentikációt két módon lehet implementálni. Az egyik a **Security Access (0x27)** a másik pedig a korszerűbb **Authentication (0x29)** lenne.

A **Security Access**-szel az úgynevezett challenge-response autentikációt lehet megvalósítani, amelynek a menete először egy valamilyen ismeretlen ECU-n belüli információ-nak (*seed*) a szolgáltatása a diagnosztikai tesztelő részére, aki ezt az információt valamilyen közös titok (pl: szimmetrikus kulcs) használatával előállít egy kulcsot, amelyet elküld az ECU-nak. Az ECU a fogadó oldalon ellenőrzi, hogy a kulcs ami érkezett az megegyezik azzal amit ugyanezen közös titok és algoritmus használatával állított elő a *seed* lekérése után. Amennyiben a kulcsok megegyeznek az ECU a felhasználót átengedi magasabb biztonsági szinthez ezzel hozzáférést biztosítva korábban nem elérhető diagnosztikai szolgáltatásokhoz.

Az **Authentication** már egy jóval komplexebb, tanúsítvány alapú autentikációt tesz lehetővé. Ennek a bemutatása túl mutat a jelen esettanulmányon.

A szoftverfrissítés egy több lépésből álló szekvenciája diagnosztikai rutin hívásoknak, amelyeket a szabvány "Upload download functional unit" fejezetében fejt ki.

Annak a folyamata, hogy egy új szoftver kerüljön fel az ECU-ra jellemzően a következő lépésekkel áll:

- **RequestDownload (0x34)**: Előfeltételek ellenőrzése, letöltési kérelem jóváhagyása
- **TransferData (0x36)**: Adatok feltöltése
- **RequestTransferExit (0x37)**: Érkezett adatok ellenőrzése, jóváhagyás, szoftver indíthatóvá tétele

Ezt a folyamat természetesen bővíthető további vagy ezeket megelőző lépésekkel, illetve az egyes rutinok belső működése sincs szabályozva. Maga a keret viszont egy szoftverfrissítésnek az jellemzően ilyen diagnosztikai szolgáltatások futtatásával és kérések kiszolgálásával történnek az elektronikus vezérlőegységeken.

Érdemes kiemelni itt, hogy ez a frissítési metódus eleinte a járműszervizben történt volna a szerező által, de a napjainkban már ezeknek a távoli elérése, jellemzően egy külön erre integrált elektronikus vezérlőegység által is lehet lefolytatva. Ezeknek sokban kényelmi, de szintén szoftveres sérülékenységek vagy hibák javítása szempontjából is fontos szerepük van.

A protokoll egyébként tartalmaz olyan szolgáltatásokat mint a **Routine Control (0x31)** amellyel a beszállító és a vevő az eszközre specifikus, nem szabványos diagnosztikai szolgáltatásokat is definiálhat. Ilyenek lehetnek kulcs és tanúsítványkezelő rutinok, illetve gyártósoron használt szolgáltatások.

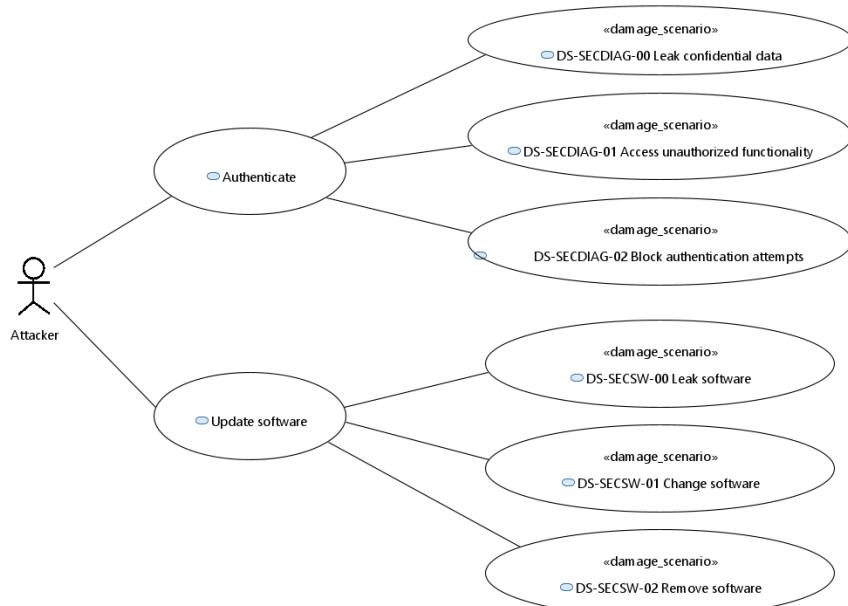
## 6.1. Károkozások és értékek azonosítása

A 6.2 ábrán láthatóak a potenciális károkozások. A korábban meghatároztuk a két fő szolgáltatást ami az esettanulmányban elemzésre kerül, majd ezekhez a CIA triát mentén való elemzés szerint meghatározásra kerültek a potenciális károkozások. Továbbá egy egy azonosító is került a károkozások nevébe, valamint a felvett sztereotípiával az attributálást is el lehetett végezni.

- **DS-SECDIAG-00 Leak confidential data:** Amennyiben az autentikációs szolgáltatás bizalmassága elveszik, abban az esetben a támadónak lehetősége lesz a bizalmás diagnosztikai adatok kiolvasására
- **DS-SECDIAG-01 Acces unauthorized functionality:** Az autentikációs szolgáltatás integritásának a megtörése vezethet olyan szolgáltatások elérhetővé válásához amelyek normális esetben nem lennének elérhetők

- **DS-SECDIAG-02 Block authentication attempts:** Az autentikációs szolgáltatás elérhetőségének a blokkolásával korlátozható jogosult felhasználóknak a korlátozott szolgáltatásokhoz való hozzáférése
- **DS-SECSW-00 Leak software:** A szoftver kiszivárogtatásával lehetőség adódik a szoftverben sérülékenységek keresésére, annak kihasználását könnyebbé téve
- **DS-SECSW-01 Change software:** A szoftver megváltoztatásával kártékony szoftver is feljuttatható az elektronikus vezérlőegységre
- **DS-SECSW-02 Remove software:** A szoftver elérhetőségének korlátozásával az elektronikus vezérlőegység teljes funkcionalitása blokkolhatóvá tehető

Abuse case diagram: A way of modeling misuses of the product similarly to regular use cases.



**6.2. ábra.** Diagnosztika kihasználási esetei

A termékleíráshoz funkcionalitásokat definiáltuk, kellenek még az ezek által dependált értékek. HW-es komponensből egyedül a CAN csatlakozó hozható fel hiszen azon fog keresztül utazni a bejövő UDS üzenet. SW-szintű értékekből vehetjük a CAN üzenetet amiben utazik az UDS üzenet, maga az UDS üzenet, valamint az az által szállított adatok. Ez a szállított adat a szoftverfrissítés esetén a szoftver lesz, egyébként pedig diagnosztikai adatok lehetnek elérhetőek kiolvasásra az autentikációt követően.

A termékleírás ábrázolva a 6.3 ábrán látható. A sztereotípiák felvétele után volt lehetőségem az attributálásra, ezt a következőképpen tettem meg:

- **Diagnostic Authentication:** A károkozásokból származtathatóan a CIA minden tulajdonsága vonatkozik a funkcionálisra
- **Software Update:** A károkozásokból származtathatóan a CIA minden tulajdonsága vonatkozik a funkcionálisra
- **CAN Connector**

Sértetlenség: Egy CAN csatlakozó integritását megsértve, kártékony eszközökkel kerülhet kapcsolatba az elemzés tárgyaként szereplő ECU

Elérhetőség: Egy CAN csatlakozó elérhetőségét megsértve, a beérkező üzenetek hiányára kell felkészülni

- **CAN Message**

Bizalmasság: Az üzenet tartalmazhat bizalmas információt amely segíthet további támadások esetén

Sértetlenség: Az üzenet tartalma változtatható, ezzel a fogadó félnél kiváltott reakció is

Elérhetőség: Az üzenet blokkolható, a hiánya a fogadó félnél nem zárható ki

- **UDS Message**

Bizalmasság: Az üzenet tartalmazhat bizalmas információt amely segíthet további támadások esetén

Sértetlenség: Az üzenet tartalma változtatható, ezzel a fogadó félnél kiváltott reakció is

Elérhetőség: Az üzenet blokkolható, a hiánya a fogadó félnél nem zárható ki

- **Software**

Bizalmasság: A szoftver a fejlesztő cég szellemi tulajdona, ennek kiszivárgása egyrészről anyagi kárt, másrészről további támadások elvégzését is segítheti

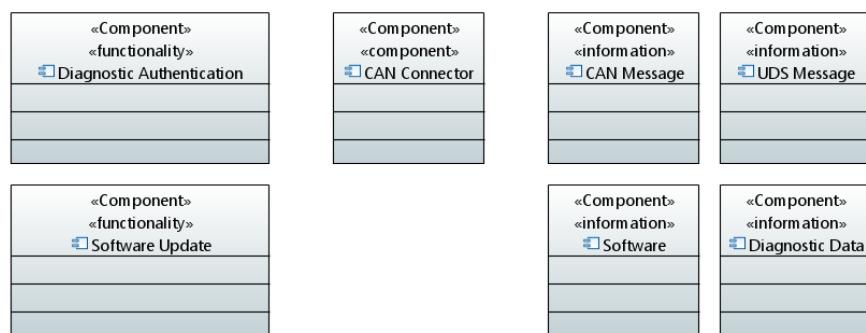
Sérhetlenség: A szoftver változtatható, ezzel a futtató ECU működése is

Elérhetőség: A szoftver törölhető, hiánya az ECU biztonságos működését befolyásolhatja

- **Diagnostic Data**

Bizalmasság: Az üzenet tartalmazhat bizalmas információt amely segíthet további támadások esetén

ISO21434-WP-09-01 Item definition: Component diagrams are common in the automotive industry to conceptualize parts of the system in pre-development. As the TARA is a primarily concept phase activity we can use already existing component diagrams and add desired stereotypes to different components.



**6.3. ábra.** Diagnosztika termékleírása

## 6.2. Fenyelésmódell származtatása

Az Acceleo szkript futtatása a következő fájlt hozza létre Diagnostics.apeditor néven:

```
<apeditor:ThreatModel xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:apeditor="hu.bme.mit.thesis.apeditor">
<damagescenarios name="DS-SECSW-00 Leak software" lossOfConfidentiality="true" lossOfIntegrity="false" lossOfAvailability="false"/>
<damagescenarios name="DS-SECSW-01 Change software" lossOfConfidentiality="false" lossOfIntegrity="true" lossOfAvailability="false"/>
<damagescenarios name="DS-SECSW-02 Remove software" lossOfConfidentiality="false" lossOfIntegrity="false" lossOfAvailability="true"/>
<damagescenarios name="DS-SECDIAG-00 Leak confidential data" lossOfConfidentiality="true" lossOfIntegrity="false" lossOfAvailability="false"/>
<damagescenarios name="DS-SECDIAG-01 Access unauthorized functionality" lossOfConfidentiality="false" lossOfIntegrity="true" lossOfAvailability="false"/>
<damagescenarios name="DS-SECDIAG-02 Block authentication attempts" lossOfConfidentiality="false" lossOfIntegrity="false" lossOfAvailability="true"/>
<asset xsi:type="apeditor:Functionality" name="Software Update"/>
<asset xsi:type="apeditor:Functionality" name="Diagnostic Authentication"/>
<asset xsi:type="apeditor:Component" name="CAN Connector" confidentiality="false" integrity="true" availability="true"/>
<asset xsi:type="apeditor:Information" name="CAN Message" confidentiality="true" integrity="true" availability="true"/>
<asset xsi:type="apeditor:Information" name="UDS Message" confidentiality="true" integrity="true" availability="true"/>
<asset xsi:type="apeditor:Information" name="Software" confidentiality="true" integrity="true" availability="true"/>
<asset xsi:type="apeditor:Information" name="Diagnostic Data" confidentiality="true" integrity="false" availability="false"/>
</apeditor:ThreatModel>
```

## 6.3. Fenyelésmodellezés

A generált fájl innentől beimportálható és megnyitható olyan Eclipse környezettel amelyhez az elemzőeszköz plugin fájljai hozzá vannak adva.

A dependenciák meghatározásánál a károkozásokhoz hozzátudjuk rendelni a funkcionáliszt viszonylag egyértelműen, maga az azonosítójuk is jelzi melyik károkozás melyik funkcióhoz tartozik.

Továbbá minden funkcionáliszt tartozik a **CAN Connector** mint HW komponens valamint a **CAN Message** és **UDS Message** mint információ. Amiben a két funkcionális eltér, hogy ameddig a **Software Update** esetén az érkező üzenetek a **Software**-t tartalmazzák, addig az autentikáció után elsősorban a kimenő **Diagnostic Data** az érték amit védeni akarunk.

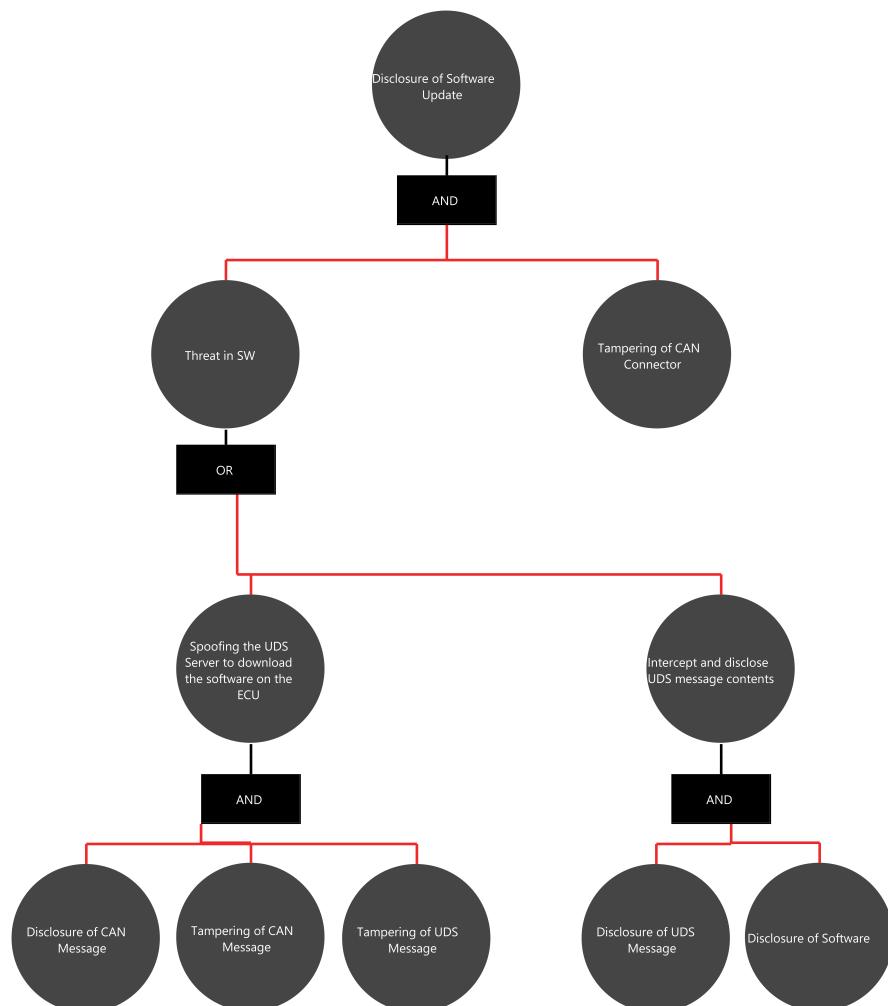
Ebből generált majd megszerkesztett hibafák és elemzésük alább megtekinthető.

## 6.4. Támadási fa analízis

Ahogy az a 6.4 ábrán látható a gyökér fenyelgetés a **Software Update** bizalmasságának sértését okozó *Disclosure of Software Update* lesz. Ez azt jelenti, hogy ezt a funkcióit vagy az ehhez tartozó diagnosztikai rutinokat olyan módon használják amellyel bizalmas információkhoz juthat a támadó.

Először levontam a következetést, hogy egyrészről minden lehetséges támadás azzal fog kezdődni, hogy a támadó valahogyan sérti az integritását a CAN busnak vagy a rendszer csatlakozóján vagy az ahhoz kapcsolódó elemek módosításával. Ez jelenthet fizikai hozzáférést vagy amennyiben egy csatlakoztatott eszköz felett szerez irányítást a támadó, abban az esetben ez akár távolról is elvégezhető.

Másodsorban két támadási irányt különítettem el, az egyik esetben a támadó megpróbál olyan diagnosztikai szolgáltatásokat használni amelyekkel a telepített szoftver kinyerhető lenne az eszközből. A másik esetben az éppen folyamatban lévő szoftverfrissítésnél érkező üzenetek lehallgatásával és azok tartalmának visszafejtésével próbálhat a támadó a szoftverhez hozzájutni. Előbbi esetben egy CAN üzenet lehallgatásával (disclosure) hozzájut a CAN ID-khoz és paraméterekkel, majd módosított üzeneteket injektál amelyekkel a szoftver letöltését kezdeményezheti. A másik esetben UDS üzenet lehallgatás és a szoftver visszafejtése történne.



**6.4. ábra.** Támadási fa a "Leak software" károkozáshoz

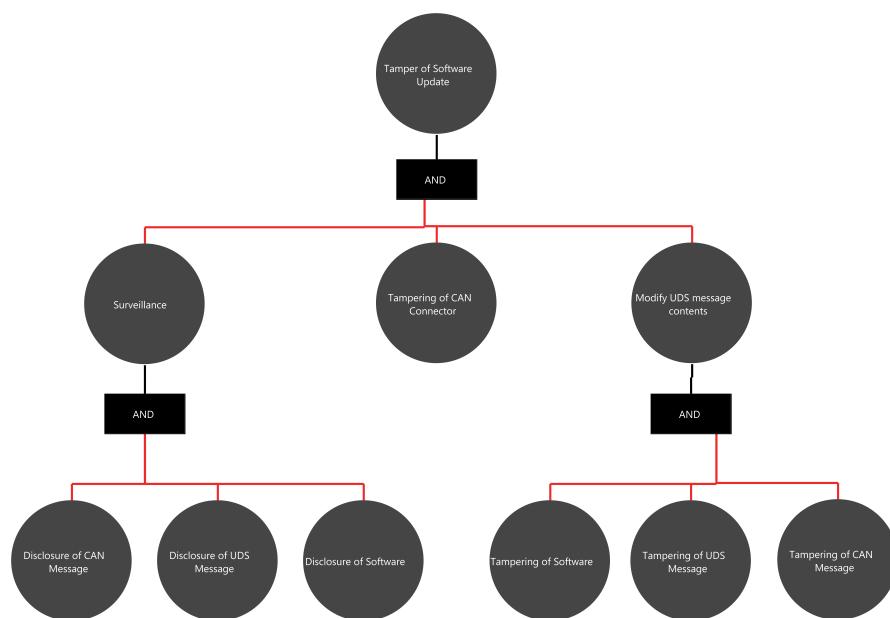
A következő támadási fa a *Change software* károkozásra készült és az 6.5 ábrán látható.

Itt egyrészről különvettem egy *Surveillance* pszeudo-fenyegést, amely maga alá gyűjt azokat a fenyegéseket amelyek önmagukban nem képesek a károkozást okozni, azonban ezt is kénytelenek előzetesen megtenni. A feltételezés az az, hogy a támadónak egy szoftver frissítési folyamatot le kell hallgatnia ahhoz, hogy a sajátját is véghez vigye. Itt egyrészről arról van szó, hogy a támadó azonosítja az ECU-kat a CAN és UDS üzenetek metaadatai alapján, valamint a szoftver felépítését is szükséges azonosítania ahhoz, hogy végül egy saját szoftvert is képes legyen az eszközre feltölteni.

Emellett egyértelműen szükséges a HW oldaláról a CAN busz integritásának megváltoztatása, hogy a támadó képes legyen a saját üzeneteit elküldeni a célpont ECU-nak.

Végül maga a szerkesztett vagy saját szoftver feltöltés lesz a következő lépés, ez magával vonja minden az egyedi CAN és UDS üzenet illetve szoftver létrehozását a támadó oldalról, úgy, hogy azt a célpont ECU elfogadja.

Megfigyelhető, hogy itt egyébként nincs szükség erre a szegmentálásra, hiszen minden atomi fenyegésnek jelen kell lennie a rendszerben a rendszerszintű fenyegés aktiválásához, viszont ebben a formában átláthatóbb és funkcionálisan nincs hatással az elemző eszköz működésére ez a formátum.

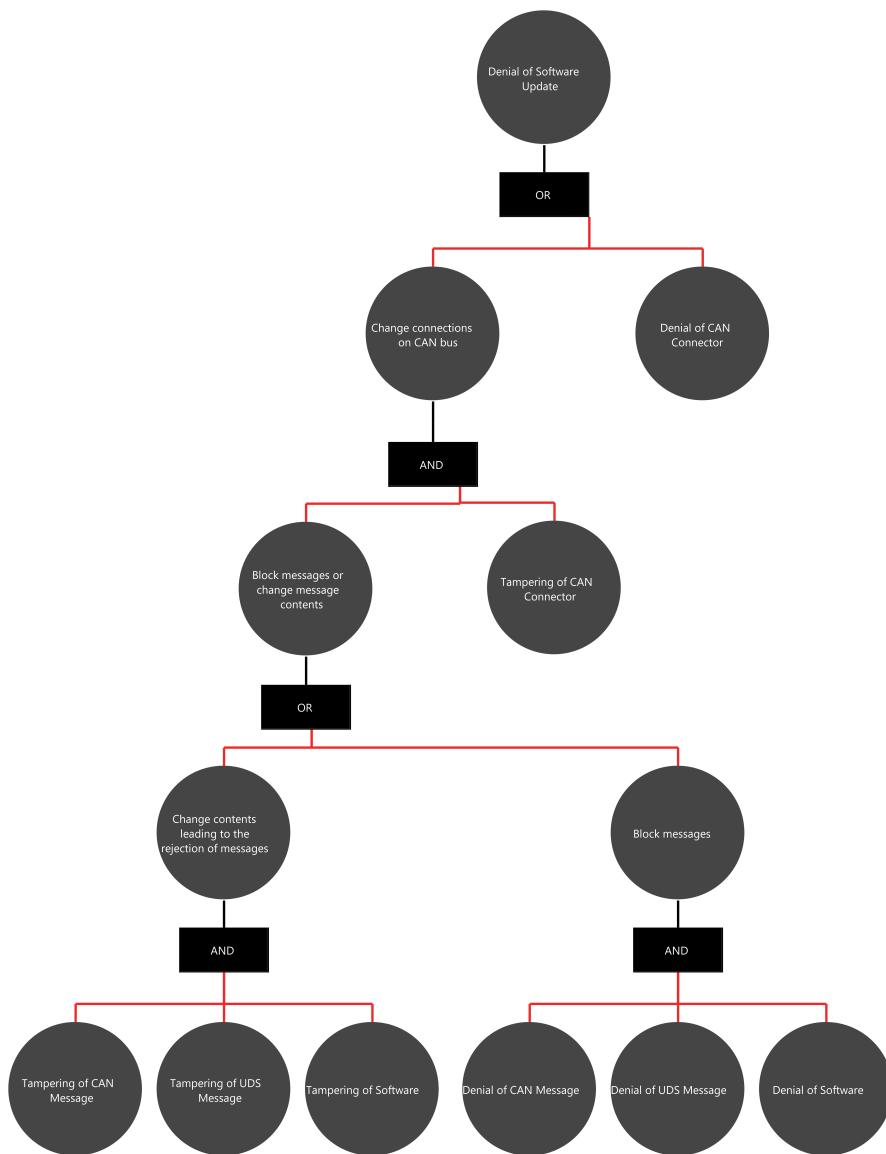


**6.5. ábra.** Támadási fa a "Change software" károkozásra

A szoftver elérhetőségének módosítására az 6.6 ábrán láthatjuk a támadási fát. Ez a támadás már sokkal több módon elvégezhető, a kár kisebb a többi esethez képest azonban az elvégzéséhez szükséges tudás is a támadó részéről.

Egyrészről az első szinten külön vettem a HW-es kapcsolat megszüntetését amely egy fizikai hozzáférést jelentene a járműhöz attól ami potenciálisan távolról is elvégezhető.

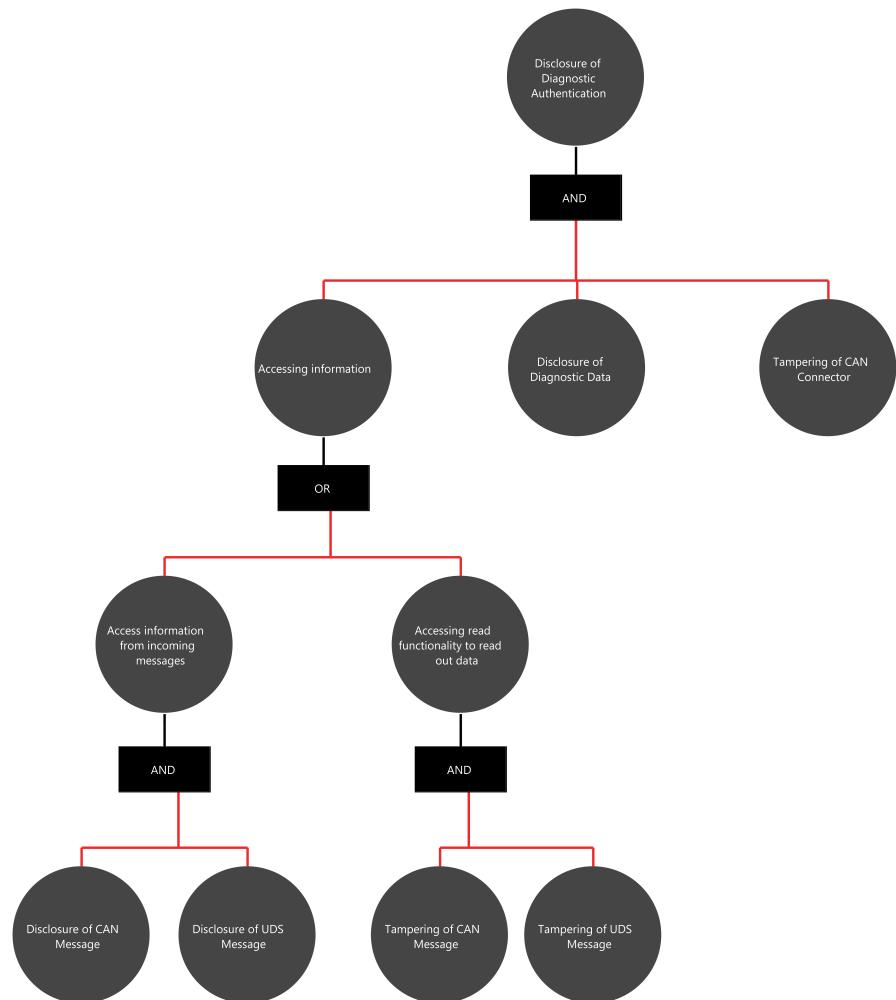
A távolról elvégezhető esetben lehetőség szerint a támadó egyrészről sértenie kell a CAN busz integritását másrészről vagy az érkező üzenetek blokkolását kell elérnie, ezzel a szoftver funkcionalitását blokkolva vagy azokat úgy módosítva, hogy utána a célpont ECU ne fogadja el azokat (pl. szálított adat módosítása úgy hogy a CRC nincs módosítva egyenesen vezetne az üzenet eldobásához üzembiztonsági elvárások miatt).



**6.6. ábra.** Támadási fa a "Remove software" károkozáshoz

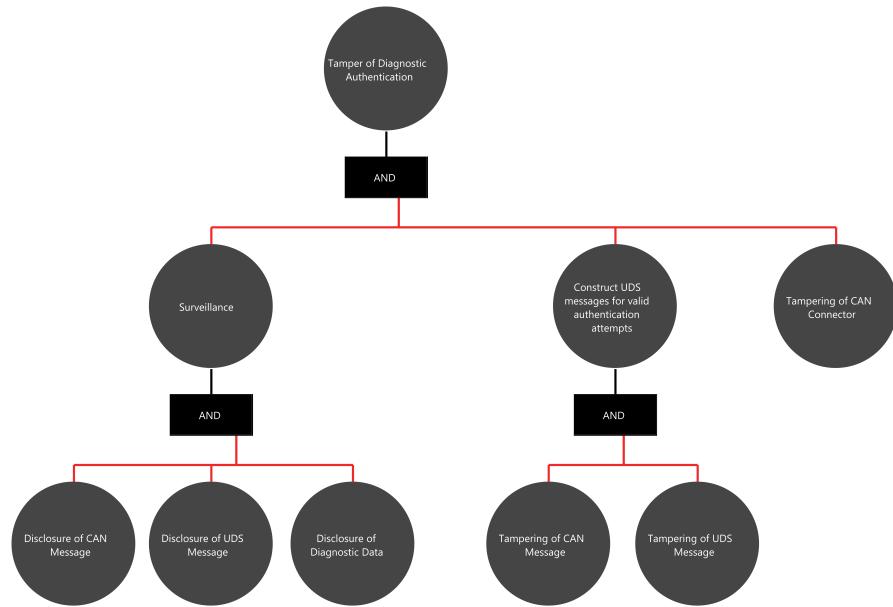
Áttérve a diagnosztikai adatok szivárogtatására az 6.7 ábrán, itt is alapvetésnek vesszük a CAN busz integritásának változását valamint magát az adatszivárgást.

A következő szinten két utat különböztetünk meg az információkhöz való hozzáférésnél, egyik esetben a bejövő üzenetek lehallgatása és visszafejtése lehet a támadó megoldása. A másik bonyolultabb esetben olyan üzeneteket kell a támadónak konstruálnia amelyeket a célpont ECU tud értelmezni valamint, megfelelőnek és jogosultnak ítélni.



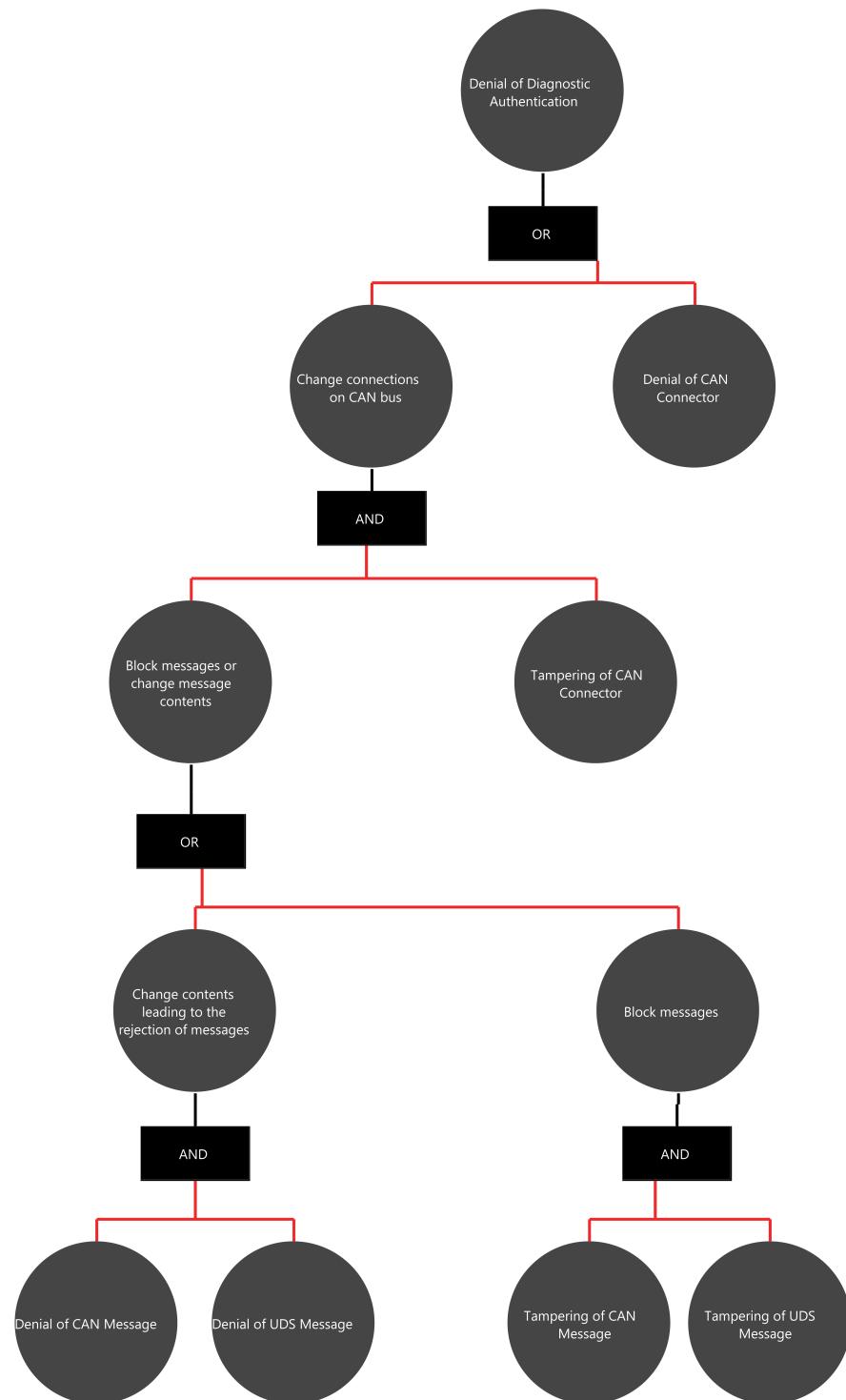
**6.7. ábra.** Támadási fa a "Leak confidential data" károkozáshoz

Az 6.8 ábrán, azt a támadási fát keressük amelyikkel a támadó hozzáférhet korlátozott elérésű diagnosztikai szolgáltatásokhoz. Hasonlóan a szoftver megváltoztatásához itt is szükségeltetik egy lehallgatási lépés, valamint ezen túl a megfelelő üzenetek konstruálása amely különböző mitigációk segítségével jól védhető.



**6.8. ábra.** Támadási fa a "Access unauthorized functionality" károkozáshoz

Az 6.9. ábrán, az autentikációs lépések blokkolását okozó támadási fa látható. Ez is megegyezik a szoftver frissítés elérhetőségét sértő fához. Ennek az oka ezúttal is az, hogy a műveletet minden minden diagnosztikai szolgáltatás, ugyanazon lépésekkel kell megtenni ezen az absztrakciós szinten.



**6.9. ábra.** Támadási fa a "Block authentication attempts" károkhoz

Összességében elmondható, hogy a támadási fák konstruálása helyes, ezt bizonyítja az is, hogy a diagnosztikai rutinok támadási fái ugyanazon mintákat követik hiszen ugyan

azon értékeket használja a két szolgáltatás. Amennyiben a szoftvert és a diagnosztikai adatot egy absztraktabb *adat* meghatározással jelöltük volna a termékleírásban úgy elég lett volna kevesebb támadási fa azonban, ezeknek a külön vétele a manuális analízishez szükséges.

Szintén elmondható, hogy az eljárás gyors és ezáltal könnyen megismételhető. Ezen támadási fáknak a konstruálása ugyanazon atomi és legfelső eseményekből áll így a kontextus váltás a külön fák konstruálásakor jóval egyszerűbb.

Végül pedig a teljességet is teljesíti a támadási fa, minden termékleírásban valamint fenyelőmodellben meghatározott érték elleni fenyelő szerepel valamely támadási fában egyrészről, másrészről pedig megteremtettünk egy lekövethetőséget (traceability) a SW-, illetve HW-szintű fenyelők valamint a károkozások között.

## 6.5. Manuális analízis

Az utolsó lépések az elemzésnek a manuális analízis. Itt először a hatásérték elemzést, utána pedig a támadás megvalósíthatóság elemzést végezzük el.

Ahogy az a 6.10 ábrán látható a SFOP értékek mentén meg lettek határozva a hatás-értékei az egyes károkozásoknak.

A legkomolyabb hatások a szoftver megváltoztatásával, törlésével, valamint a korlátozott hozzáférésű funkcionálitások használata lettek. Ezek értelemszerűen nagyobb kockázatot is jelentenek mint mondjuk az autentikációnak a blokkolása. Az elemzés az ISO 21434 [6] által ajánlott és annak a függelékében lévő példa elemzések mintájára készültek el.

6.10. ábra. Hatásérték elemzés

Az 6.11 ábrán látható az értékelés az Attack Potential Based megközelítést használja, ez szintén az egyik ajánlása az ISO 21434 [6] szabványnak, valamint szintén annak a függelékében lévő minta alapián készült el ez az elemzés is.

Ezen kiértékelések történhetnek egy szakértő által egyénileg, több stakeholder bevonásával illetve behatolástesző mérnökök támogatásával. A kiértékelés a támadó képességei és a rendelkezésre álló információk alapján ad egy becslést a támadási utak megvalósíthatóságára.

The screenshot shows a Microsoft Excel spreadsheet titled "Diagnostics\_02\_attack\_feasibility\_rating". The data is organized into several sections:

- Project Details:** Project name is Thornton\_525\_EPS, version WP-15-03, and date WP-15-05.
- Approach Guideline:** Describes the attack potential-based approach and the need to determine attack feasibility for each step and path. It specifies elapsed time (1 day/1 week/1 month/6 months/More), specialist level (Layman/Proficient/Expert/Multiple experts), knowledge of the item or component (Public/Restricted/Confidential/Strictly confidential), and equipment (Window of Opportunity/Unlimited/Easy/Moderate/Difficult).
- Attack Path Data:** A table listing various attack paths, their elapsed time, specialist expertise, knowledge of the item or component, window of opportunity, equipment, and final attack feasibility rating. The data includes:
 

Attack Path	Elapsed Time	Specialist Expertise	Knowledge of the Item or Component	Window of Opportunity	Equipment	Attack Feasibility Rating
Attack Path - DS-SECSW-00 Leak software	1 week	Proficient	Confidential	Moderate	Specialized	12
i-> Disclosure of CAN Message	1 day	Layman	Public	Easy	Specialized	7
i-> Tampering of CAN Connector	1 day	Layman	Public	Easy	Standard	6
i-> Tampering of UDS Message	1 week	Proficient	Confidential	Moderate	Specialized	12
i-> Tampering of CAN Message	1 day	Proficient	Public	Moderate	Specialized	9
Attack Path - DS-SECSW-00 Leak software	1 day	Proficient	Restricted	Moderate	Specialized	11
i-> Tampering of CAN Connector	1 day	Layman	Public	Easy	Standard	6
i-> Disclosure of UDS Message	1 day	Layman	Public	Easy	Specialized	7
i-> Disclosure of Software	1 day	Proficient	Restricted	Moderate	Specialized	11
Attack Path - DS-SECSW-01 Change software						

**6.11. ábra.** Támadás megvalósíthatóság elemzés a "Leak software" károkozáshoz

Ezzel befejezve a manuális analízisek eredményeinek a szabvány által leírt kombinációja alapján fog a kiberbiztonsági mérnök előállni a kockázati értékekkel amelyek mentén lehet eldönteni mely támadások azok amelyek valóban nagy kockázattal járnak és amelyekre védelmet kell biztosítani valamelyen formában.

## 7. fejezet

# Összegzés

A feladatom összességében sikeresnek mondható és a feladatkiírásban szereplő pontokat sikeresen teljesítette illetve valódi ipari használatra is alkalmasnak mondható.

A dolgozatom elején bemutattam az irodalomban elérhető modellalapú és modellekkel együtt használható kiberbiztonsági analíziseket. Ezekből az analízisekből egyrészről a fenyelgetésmo dellnek a rendszermodellből való származtatását, illetve az üzembiztonsági analízisekből adaptált támadási fa modell alkalmazását használtam fel a saját elemzésem elvégzésére.

Megvizsgáltam a rendszermodellek felhasználási lehetőségeit és egy fejlesztett szkript segítségével automatikusan képes voltam fenyelgetésmo delleket a rendszermodellből. Itt egyrészről a használati eset diagramokból tudtam a károkozási eseteket előállítani, valamint a komponens diagramban készült termékleírásból tudtam egy kezdeti fenyelgetésmo dellt előállítani amely alkalmas volt elemzések elvégzésére.

A javasolt analízis megközelítés az irodalomkutatás valamint ipari ismeretek alapján az ISO 21434 [6] Threat Analysis and Risk Assessment lett. Ennek az elvégzésére elkészült egy elemző eszköz amely Eclipse környezethez plugin formájában került implementálásra. Ez alkalmazás lehetővé tesz további automatizálási lehetőségeket is az elemzés elvégzése során.

Elvégeztem egy esettanulmányt az ISO 14229 [5] szabvány által definiált két diagnosztikai szolgáltatásra amelynek az eredményeit bemutattam.

Az eszköz maga teljesíti az ISO 21434 szabvány követelményeit, alkalmas az integrációra bármilyen rendszermodellező eszközzel amelyben definíálható a kiberbiztonsági profil valamint a modellből szöveg generálásának lehetősége. Az elemzés elvégzése nagy mértékben gyorsított a nem modellalapú implementációkhoz képest és segít is a megértetésben a grafikus ábrázolása a támadási útvonalaknak. Az elemzések könnyen módosíthatóak, elvégezhetőek többször valamint tovább finomíthatóak a termékleírás bővítésével és új értékek bevonásával. Szintén fontos kiemelni, hogy ezen elemzések elvégzése kevesebb emberi erőforrást igényel, hiszen a támadási fák összeállítása egy specializált szaktudás amelyet később szakértőkkel lehet ellenőriztetni, nincs igény azoknak a folyamatos jelenlétére a fák konstruálása során.

Fejlesztési lehetőségeknek megemlíteném egyrészről a manuális analízisnek a modellező eszközben való integrálását, ezzel csökkentve a különböző eszközökön végzett munkát. Szintén érdemesnek tartanám a mitigációk felvételét az egyes fenyelgetésekhez, ezzel akár csökkentve a lehetséges támadási útvonalak számát. Utolsósorban pedig fontosnak tartanám az egyes fenyelgetésekhez valós sérülékenység adatbázisoknak (pl. CVE) vagy más keretrendszerrel (pl. MITRE) való megfeleltetését, ezzel is lehetővé téve alaposabb elemzéseket.

# Irodalomjegyzék

- [1] ASPICE: Automotive spice® for cybersecurity process reference and assessment model, 2021.
- [2] Nguyen H. N. Sabaliauskaite G. Shaikh S. Zhou F. Bryans J., Liew L. S.: A template-based method for the generation of attack trees, 2020.
- [3] Gareth Halfacree.
- [4] JOHN PEARLEY HUFFMAN: It takes a lot of wiring to keep a modern vehicle moving (witness this bentley's harness), 2016.
- [5] ISO: Iso 14229 "road vehicles — unified diagnostic services (uds)", 2020.
- [6] ISO/SAE: Iso/sae 21434 "road vehicles - cybersecurity engineering", 2021.
- [7] Hesamaldin Jadidbonab Paul Wooderson Hoang Nguyen Kacper Sowka, Vasile Palade: A review on automatic generation of attack trees and its application to automotive cybersecurity, 2023.
- [8] Almgren M. Karahasanovic A., Kleberger P.: Adapting threat modeling methods for the automotive industry, 2017.
- [9] Khaled Karray: Cyber-security of connected vehicles : contributions to enhance the risk analysis and security of in-vehicle communications, 2020.
- [10] Ramin Tavakoli Kolagari Markus Zoppelt: Reaching grey havens: Industrial automotive security modeling with sam, 2019.
- [11] Alexander Much Martin Böhner, Alexander Mattausch: Extending software architectures from safety to security, 2015.
- [12] Claudia Eckert Martin Salfer, Hendrik Schweppe: Efficient attack forest construction for automotive on-board networks, 2014.
- [13] United Nations: Un regulation no. 155 - cyber security and cyber security management system, 2021.
- [14] Arthur Parkhouse.
- [15] Christoph Schmittner Sebastian Chlup, Korbinian Christl: Threatget: Towards automated attack tree analysis for automotive cybersecurity, 2022.
- [16] Tanvi Tirthani Yashodhan Vivek: Automotive system threat modeling, 2022.
- [17] Dr. Vivek Nigam Yuri Gil Dantas, Dr. Harald Ruess: Security engineering for iso21434, 2020.

- [18] C. Schmittner Z. Ma: Threat modeling for automotive security analysis, 2016.
- [19] ZF: Electronic control units.  
URL [https://www.zf.com/products/en/cars/products\\_65835.html](https://www.zf.com/products/en/cars/products_65835.html).