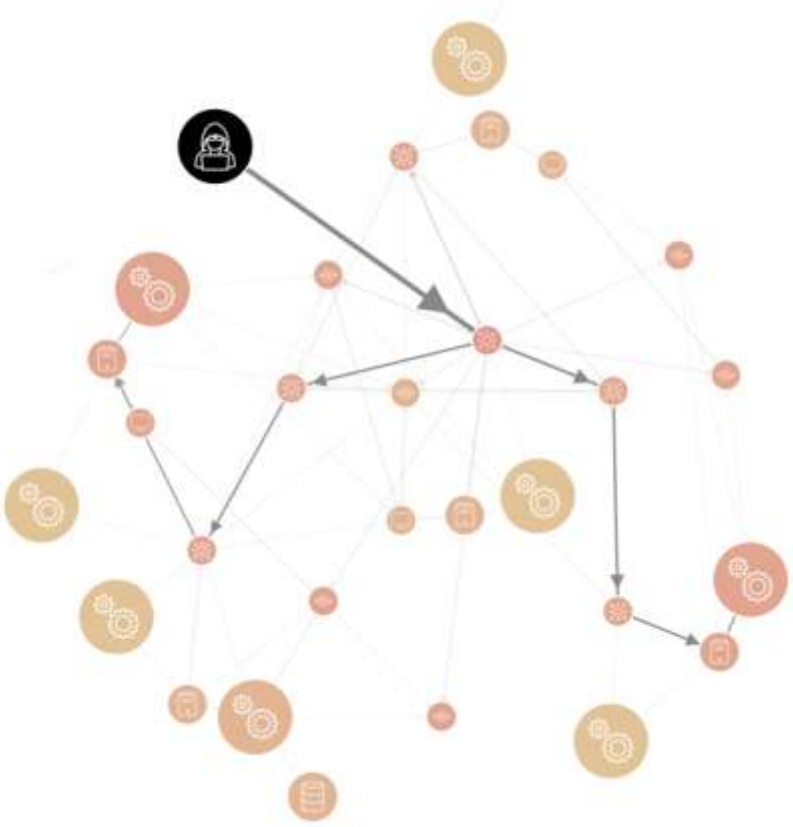
Hotmodellering och -simulering för fordons-IT (förstudie)

Publik rapport



Författare: Robert Lagerström
Datum: 2017-06-07
Projekt inom Fordons IT-säkerhet och integritet

FFI Fordonsstrategisk Forskning och Innovation
VINNOVA Energimyndigheten TRAFIKVERKET
FKG SCANIA VOLVO

Innehållsförteckning

1 Sammanfattning

Den ökande graden av datorisering och nätverksuppkoppling gör moderna fordon sårbara för cyberattacker. Det är, på grund av systemens ökande komplexitet, mycket svårt att manuellt kartlägga den stora mängden attackvägar som kan utnyttjas av potentiella angripare, och vilka konsekvenser attacker längs de olika vägarna kan få. En klar förståelse för hotbilden är en förutsättning för ett effektivt försvar. Programvarubaserade verktyg för hotmodellering och -simulering kan användas för att bedöma sannolikheten att en angripare lyckas nå fram till olika delar av fordonssystemet. Dessa verktyg ger därmed en god bild av säkerheten av ett system, liksom de säkerhetspåverkande effekter systemförändringar leder till. Idag finns det emellertid inga verktyg för hotmodellering och -simulering för fordon.

Denna förstudie har undersökt förutsättningarna för att söka ett större projekt inom ramen för Vinnova FFI. Där målet skulle vara att ta fram ett hotmodellerings- och simuleringsspråk för fordons-IT.

Universitetslektor Robert Lagerström på Kungliga Tekniska Högskolan har lett förstudien tillsammans med start-up-företaget foreseeti och Scania. Under förstudien har vi identifierat och diskuterat projektet med ytterligare partners, tagit fram början till ett ramverk för språket, samt skrivit projektansökan. Förstudien har pågått under våren 2017 och huvudleverabeln är således en projektansökan till Vinnova FFI som skickats in juni 2017. Nya parter som tillkommit under processen är Volvo Cars och F-Secure.

2 Executive summary in English

Increasing use of computers and networking lead to increasing vulnerability of modern vehicles to cyber attacks. Due to the increasing complexity of systems, it is very difficult to manually map the large number of attack paths available to potential attackers. A clear understanding of the threats and vulnerabilities is a prerequisite for an efficient defense. Software tools for threat modeling and attack simulations can today be used to assess the probability that an attacker will reach the various parts of the vehicle system. These tools thus provide a useful evaluation of the security of a system, as well as the effects of system changes on the overall security. However, in the automotive domain, there are currently no tools available for threat modeling and simulation.

This preliminary study has investigated the prerequisites for a larger project within the framework of Vinnova FFI, where the goal would be to develop a threat modeling and simulation language for vehicle IT.

Associate professor Robert Lagerström at KTH Royal Institute of Technology has led the project together with the start-up company Foreseeti and Scania. The preliminary study has identified and discussed the project with additional partners, developed the beginning of a framework for the said language, and written the application for the larger project. The preliminary study took place during spring 2017 and the main delivery is a project application for Vinnova FFI, which was submitted in June 2017. New partners added to the application are Volvo Cars and F-Secure.

3 Bakgrund

Den ökande graden av datorisering och nätverksuppkoppling gör moderna fordon sårbara för cyberattacker (Koscher et al., 2010; Checkoway et al., 2011; Petit & Shladover 2015). Det är, på

grund av systemens ökande komplexitet, mycket svårt att manuellt kartlägga den stora mängden attackvägar som kan utnyttjas av potentiella angripare, och vilka konsekvenser attacker längs de olika vägarna kan få. En klar förståelse för hotbilden är en förutsättning för ett effektivt försvar. Programvarubaserade verktyg för hotmodellering och -simulering kan användas för att bedöma sannolikheten att en angripare lyckas nå fram till olika delar av fordonssystemet. Dessa verktyg ger därmed en god bild av säkerheten av ett system, liksom de säkerhetspåverkande effekter systemförändringar leder till. Idag finns det emellertid inga verktyg för hotmodellering och -simulering för fordon.

3.1 Hotmodellering och -simulering idag

Det finns idag ett antal verktyg för hotmodellering och -simulering, dock inga för fordons-IT. Många av de tillgängliga verktygen lider också av väsentliga brister då detta är ett fält som är relativt nytt och omoget. Framtiden ser dock ljus ut om fokus kan läggas på rätt saker.

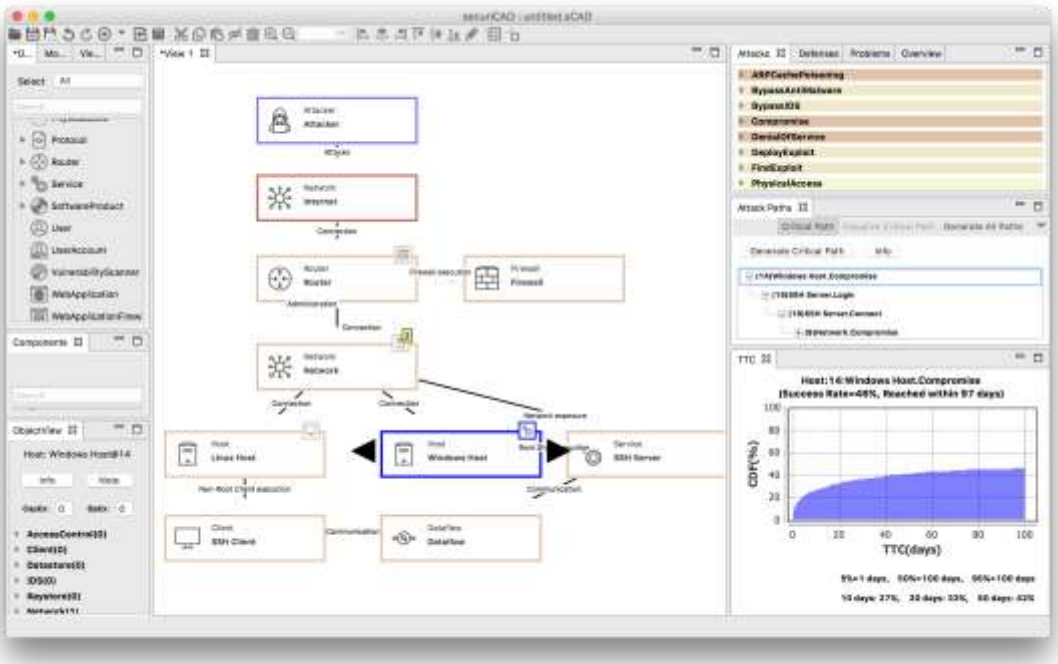
Internationell state of the art

Ett av de mest kända verktygen för hotmodellering är Microsofts Threat Modeler (Shostack, 2008), som används för att på en hög nivå modellera potentiella attacker mot informationssystem. Liksom många andra verktyg saknar Threat Modeler möjligheten till avancerade simuleringar av attacker. Ett par verktyg som omfattar simulering är SkyBox1, RedSeal2 och den akademiska produkten MulVal (Ou et al., 2005). Dessa simuleringar, baserade på attackgrafer, är emellertid endast baserade på kända sårbarheter från officiella sårbarhetsdatabaser, vilket innebär svårigheter att simulera framtida attacker. I designfasen är naturligtvis framtida attacker av stor betydelse. Utöver verktyg finns också metoder för hotmodellering, som STRIDE (Kohnfelder & Garg, 1999), där namnet är en minnesregel för en uppsättning hotklasser: Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service och Elevation of Privilege. En annan metod är OCTAVE (Alberts & Dorofee, 2002) för modellering av hot och identifiering av motåtgärder. I CORAS (Lund et al., 2010) modelleras hotscenarier som riktade acykliska grafer med sannolikheter på kanterna. Ingen av ovanstående metoder tillåter emellertid automatiserad analys eller simulering. UMLSec (Jürgens 2002) är en utveckling av UML för att inkludera säkerhetsrelevant information i UML-modeller. Trike-ramverket (Saitta et al., 2005) genererar attackgrafer automatiskt från krav- och implementeringsmodeller. Modelleringskostnaden är emellertid hög även för medelstora system. Trots att det idag finns flera initiativ med fokus på hotmodellering är dessa oftast antingen generella, med lite stöd för vad som bör modelleras och hur simulering ska genomföras, eller mycket specifika med fokus på olika typer av system och miljöer där fordons-IT inte har fått någon uppmärksamhet alls.

foreseeti securiCAD

Ett av de mest innovativa och sofistikerade, kommersiellt tillgängliga hotmodellerings- och -simuleringsverktygen utvecklas av foreseeti3, ett av Ny Teknik och Affärsvärldens 33-listebolag 2016 (mest lovande unga teknikföretag). Verktyget, securiCAD (Ekstedt et al., 2015), är baserat på probabilistisk inferens i automatgenererade attackgrafer, och använder parallelliserade, GPU-baserade algoritmer för att effektivt beräkna grafer med miljontals attacksteg. Typiska modelleringsobjekt utgörs av Nätverk, Routrar, Brandväggar, Arbetsstationer, Accesskontroll, Användarkonton, etc. Modeller kan variera i storlek från några få objekt till väldigt stora, autogenererade modeller med tio- eller hundratusentals objekt.

[1] www.skyboxsecurity.com
[2] www.redseal.net
[3] www.foreseeti.com



Figur 1. Exempel på en securiCAD-modell i securiCAD:s användargränssnitt. En attackerare på Internet kan i modellen nå arbetsstationen "Windows Host" inom 10 dagar med 27 % sannolikhet (se TTC-fönstret). Den snabbaste vägen är via SSH-servern (se Attack Path-fönstret).

För varje typ av objekt, exempelvis en Arbetsstation (Host i Figur 1), har securiCAD definierat ett antal möjliga attacksteg tillgängliga för en angripare. T.ex. är det nödvändigt att undgå upptäckt av eventuella antivirusprogram (BypassAntiMalware i Figur 1) för att framgångsrikt utnyttja kända sårbarheter på en arbetsstation (Compromise i Figure 1). Objekt har också försvarsmekanismer som försvårar angrepp. Som exempel kan operativsystemen i arbetsstationer implementera Adress Space Layout Randomization (ASLR), vilket minskar möjligheten att utnyttja vissa typer av sårbarheter. Tiden som krävs av en angripare för att lyckas ta över ett objekt varierar således beroende på vilka försvarsmekanismer som är aktiverade.

En securiCAD-modell, som består av många sammanlänkade objekt, kan konverteras till en attackgraf, som på ett koncist vis beskriver alla angriparens möjliga attackvägar, tillsammans med estimat på den tid varje steg förväntas ta. Med grafen kan den sannolika tiden för angriparen att nå varje punkt i modellen beräknas. Denna tidsuppskattning, kallad Time To Compromise (TTC) utgör ett kvantitativt mått på systemets säkerhet.

securiCAD är baserad på många års forskning vid KTH, publicerad i en stor mängd vetenskapligt granskade artiklar t.ex. (Johansson & Johnson, 2005; Sommestad et al., 2010; Holm et al., 2012; Blom et al., 2016; Flores & Ekstedt, 2016, Johnson et al., 2016).

3.2 Hotmodellering och -simulering för fordon

I sin nuvarande form är securiCAD som sagt, liksom de flesta hotmodellerings- och -simuleringsverktyg, avsett för företagsnätverk. Det finns många likheter mellan företagsnätverkssäkerhet och fordonssystemsäkerhet, men också många skillnader. Ett sätt att tydliggöra dessa skillnader och likheter är genom att jämföra de objekt som behöver representeras i modelleringsspråket för respektive domän. Objekt som vi förväntar oss återfinna i båda domäner inkluderar Klienter, Servrar, Dataflöden, Programvaruprodukter och Protokoll. Objekt som är unika för fordonsdomänen inkluderar ECU:er, sensorer och ställdon. Vissa befintliga objekt i företagsdomänen kommer också att behöva anpassas till fordonsdomänen. Ett sådant exempel är Nätverk, som i företagsdomänen representerar ett företags-LAN snarare än en CAN-bus. Även de konnektorer som sammanlänkar objekt kommer att behöva förändras.

4 Syfte

Syftet med förstudien har varit att undersöka förutsättningarna för ett större Vinnova FFI-projekt där ett hotmodellerings- och simuleringsspråk för fordons-IT ska tas fram.

Vi arbetat med att beskriva syftet för det större projektet vi nu ansökt om. Dess syfte är: att skapa ett hotmodellerings- och simuleringsspråk specifikt anpassat för fordons-IT. Projektet kommer att; 1) ta fram ett ramverk anpassat för hotmodellering, 2) med hjälp av ramverket skapa ett hotmodelleringsspråk för fordons-IT, 3) samla information och statistik som attacksimuleringen kan baseras på t.ex. genom penetrationstester och laboratorietester, 4) implementera resultaten i en användarvänlig prototyp, 5) validera på verkliga system, och 5) sprida framgångarna.

5 Mål

Målet med förstudien var att 1) utreda hur det större projektet ska utformas, 2) vilka krav som finns på det hotmodellerings- och simuleringsspråk som föreslås, och 3) vilka intressenter som bör vara med i det större projektet.

De planerade målen i det stora projektet som sökts genom förstudien är att; 1) öka säkerhetskompetensen i fordonsbranschen genom att både tillhandahålla säkerhetsexpertis i projektleverabler och öka kompetensen bland de inblandade parterna, 2) bidra till ökad samverkan mellan parter som normalt sett inte samverkar, och 3) säkerställa att framtidens uppkopplade fordon är säkra och att de på ett tids- och kostnadseffektivt sätt kan utvärderas gällande säkerhet. Den senare punkten kommer att uppnås då projektets huvudleverabel, det domänspecifika språket, fokuserar på att designa säkra system med ett holistiskt angreppsätt.

6 Resultat och måluppfyllelse

Huvudresultatet i förstudien är ett konsortium (Kungliga Tekniska Högskolan, Foreseeti, F-Secure, Scania och Volvo Cars) som tillsammans har sökt det större projektet i juni 2017, en detaljerad plan för detta projekt (ansökan med arbetspaket, budget och tidsplan), och början på det ramverk som kommer att ligga till grund för hotmodellerings- och simuleringsspråket (MAL - Model-based Attack-graph Language).

Förväntade resultat i det större projektet vi sökt baserat på förstudien är ett hotmodellerings- och -simuleringsspråk för fordons-IT. Detta språk avses att vara fritt tillgängligt för fordonsindustri, hotmodelleringsföretag, akademia och andra intresserade avnämare. Vidare avses språket konkretiseras som en modul till securiCAD från foreseeti. Det förväntade resultatet är att modulen kommer att vara en testad och verifierad prototyp, som efter projektet kan kommersialiseras och användas effektivt av fordonsbranschen. Arbetet med att ta fram språket och modulen kommer också att bygga upp säkerhetskompetens generellt och för fordons-IT specifikt för de engagerade parterna i projektet. Inom projektet kommer också flertalet forskningsartiklar att publiceras där metod och modeller presenteras och valideras med användarfall inom projektet, samt att en doktorsavhandling och ca fyra examensarbeten kommer att slutföras under perioden.

Planen är att gå från TRL-nivå 2 till 6. Där KTH ansvarar för de tidigare faserna upp till steg 4 (Teknisk validering i laboratoriemiljö), samt att foreseeti tar ansvar för de senare stegen upp till TRL-nivå 6 (Demonstration i relevant miljö).

Resultaten i projektet förväntas öka kunskapen (väsentligt) genom gemensamt projektarbete, presentationer, artiklar och utbildning. Vi förväntar oss att resultaten kommer att föras vidare till

andra projekt hos de olika parterna för ytterligare utveckling och implementation. Den detaljerade specifikationen av hotmodellerings- och -simuleringsspråket kommer att vara publikt tillgänglig.

6.1 THREAT MOVE-plan

Detta kapitel detaljerar vår plan för det större projektet vi sökt som kallas THREAT MOVE (THREAT MOdeling and simulation of VEhicle IT). Planen och ansökan är vår huvudleverabel, vårt huvudsakliga resultat i förstudien.

Projektinnehåll

THREAT MOVE är uppdelat i åtta arbetspaket: (i) projektledning, (ii) utveckling av ramverk för domänspecifikt språk, (iii) design av domänspecifikt modellerings- och analysspråk för säkerhet i fordons-IT, (iv) implementation, (v) iterativ testning och validering av domänspecifikt språk, (vi) inkludering av Tool chain integration, (vii) fordons säkerhetsparametrar och (viii) kunskapsspridning.

Arbetspaket 1	Projektledning
Ansvarig	Robert Lagerström, KTH
Övriga deltagare	Pontus Johnson, foreseeti Niklas Wiberg, Scania Jörgen Borg, Volvo Cars
Beskrivning av innehåll	Planering – detaljplanering av innehåll i de olika arbetspaketen och dess relation till projektet, samt planering av löpande projektmöten. Styrning – se till att alla aktiviteter utförs i tid och med kvalitet, samt att resultat rapporteras och eventuella hinder hanteras. Definiera och sprida begrepp inom projektet. Synkronisera gemensam prioritering inom projektet. Uppföljning – hålla koll på vilka aktiviteter som genomförts och som är godkända, se till att budgeten följs, samt att resultat är av rätt kvalitet. Rapportering – huvudansvar för att all rapportering till Vinnova kommer in i tid, samt att interna rapporter sprids till rätt personer. Robert Lagerström har huvudansvaret, men övriga deltagare kommer att lägga tid på projektmöten, rapportskrivning etc.
Leverans	Löpande rapportering internt i projektet Rapportering till Vinnova, inkl. Projektplan Projektavtal Startrapport Lägesrapporter Slutrapport

Arbetspaket 2	Ramverk
Ansvarig	Robert Lagerström, KTH
Övriga deltagare	Pontus Johnson, foreseeti
Beskrivning av innehåll	Idag finns det inget bra sätt att specificera ett attacksimuleringsspråk. Istället blir varje implementation unik och svår att anpassa. Syftet med detta arbetspaket är att utveckla det ramverk

	som projektets huvudleverabel (det för fordons-IT domänspecifika språket) ska förhålla sig till. Målet är att ramverket ska skapa rätt förutsättningar för ett flexibelt och användbart hotmodellerings- och simuleringsspråk. Ramverket kommer att specificera vad som går att modellera och simulera på en hög nivå, t.ex. ”assets”, ”attacksteps”, ”relationships”. Robert Lagerström har huvudansvaret för att ta fram ramverket, men foreseeti deltar med mycket tid för att få detta klart tidigt.
Metod/angreppssätt	Samla in och jämföra befintliga ramverk för att i så stor utsträckning som möjligt förhålla oss till dessa. Ostrukturerade intervjuer/workshops för att validera det föreslagna ramverket.
Leverans	Ramverk Forskningsartikel Internrapport

Arbetspaket 3	Domänspecifikt språk
Ansvarig	Robert Lagerström, KTH
Övriga deltagare	Niklas Wiberg, Scania Doktorand, KTH Pontus Johnson, foreseeti Jörgen Borg, Volvo Cars
Beskrivning av innehåll	Huvudleverabel i projektet är ett domänspecifikt språk för modellering och analys av säkerhet i fordons-IT, dvs ett språk för hotmodellering och riskhantering. Språket kommer att innehålla både struktur (klasser, attribut, relationer) för modellering samt intelligens (sannolikheter) för analys. Delar av språket kommer att iterativt testas och valideras i arbetspaket 5 och 7, samt uppdateras med hjälp av indata från arbetspaket 5-7. Språket kommer att specificera vilka t.ex. ”assets”, ”attacksteps”, ”relationships” som är viktiga för denna domän (säkerhet i fordons-IT). D.v.s. t.ex. CAN bus, ECU. KTH tar huvudansvar och lägger mycket tid på utvecklingen av språket, men viktiga krav måste komma från Scania, Volvo Cars och foreseeti. Undersöka om metod och insikter från HEAVENS/HoliSec-projekten kan komplettera och stödja analys och verktyg.
Metod/angreppssätt	Iterativ design med följande delar: Studera befintlig forskningslitteratur, samt rapporter och

	standarder från fordonsbranschen Enkäter och intervjuer med experter Studera systembeskrivningar hos OEMer Testning och validering, enligt arbetspaket 5 & 7
Leverans	Validerat och testat domänspecifikt språk Forskningsartiklar Doktorsavhandling Internrapporter

Arbetspaket 4	Implementation
Ansvarig	Per Eliasson, foreseeti
Övriga deltagare	Pontus Johnson, foreseeti Utvecklare, foreseeti
Beskrivning av innehåll	Ramverket (arbetspaket 2) och det domänspecifika språket (arbetspaket 3) kommer att hållas öppna och tillgängligt för alla intressenter. För att verkligen komma industrin till nytta snabbt kommer foreseeti att utveckla moduler baserade på det domänspecifika språket i deras produkt securiCAD®. Det är viktigt för projektet att KTH, Volvo Cars och Scania kan använda securiCAD med det domänspecifika språket tidigt i detta projekt för att iterativt kunna testa och validera olika lösningsförslag. Därför kommer detta arbetspaket att starta tidigt för att sedan iterativt förbättra implementationen med hjälp av användarkrav och resultat från andra arbetspaket.
Metod/angreppssätt	I implementationen ingår att: Stödja det föreslagna ramverket (arbetspaket 2), inklusive utveckling av kompilator. Implementera det föreslagna domänspecifika språket (arbetspaket 3). Vidareutveckla det grafiska användargränssnittet m.h.a. användarkrav från Scania, Volvo Cars och KTH (arbetspaket 5). Iterativt förbättra verktyget baserat på krav från fordonsindustrin (t.ex. via arbetspaket 5-7). Förfina beräkningsmotorn så att det går att räkna ännu snabbare på ännu större modeller. Utveckla infrastrukturen för filhantering och fleranvändarmodellering.
Leverans	Modul till securiCAD® Internrapporter, t.ex. beskrivning och användarmanual

Arbetspaket 5	Övergripande testning och validering
Ansvarig	Robert Lagerström, KTH
Övriga deltagare	Doktorand, KTH Pontus Johnson, foreseeti Niklas Wiberg, Scania Säkerhetsexperter, Scania

	Jörgen Borg, Volvo Cars Säkerhetsexperter, Volvo Cars
Beskrivning av innehåll	Övergripande testning och validering av det domänspecifika språket och implementationen av språket. I detta arbetspaket kommer det domänspecifika språket användas för att modellera och simulera attacker i fordonssystem hos Scania och Volvo Cars. Detta kommer att ske med hjälp av den implementerade modulen i securiCAD. Vi kommer således att bygga Scania/Volvo-modeller och testa dess säkerhet, d.v.s. en nulägesanalys av systemen, samtidigt som språket och implementation testas och valideras. I de olika iterationerna inom projektet kommer detta arbetspaket att samla in krav för vidareutveckling av språket och implementationen, samt ge Scania och Volvo Cars återkoppling på beslut som fattas när nya tekniker planeras.
Metod/angreppssätt	I praktiken innebär det att information om systemen måste samlas in, antingen automatiskt via datainsamlingsverktyg eller manuellt i workshops och intervjuer. Med hjälp av den informationen byggs sedan modeller som securiCAD kan analysera. Resultaten som kommer ut från verktyget behöver då diskuteras och testas för att se om dessa verkar rimliga. T.ex. via experttester.
Leverans	Modeller Rapport(er) som beskriver utfall av tester (kravspecifikationer för det domänspecifika språket och implementationen) Forskningsartiklar

Arbetspaket 6	Tool chain integration
Ansvarig	Niklas Wiberg, Scania
Övriga deltagare	Pontus Johnson, foreseeti Doktorand, KTH Robert Lagerström, KTH Jörgen Borg, Volvo Cars
Beskrivning av innehåll	Utrusta implementationen av språket/verktyget med förmåga att koppla till andra system via t.ex. LinkedData. LinkedData är en öppen standard och används internt på Scania som en databuss mellan verktyg och datakällor. Detta möjliggör: Att data om t.ex. nätverksstruktur, funktionsarkitektur, och datatrafik, kan importeras i analysverktyget. Spårbarhet i utvecklingsarbetet genom att analysarbetet kan knytas till hur fordonssystemet såg ut vid den punkt där beslut tagits. Vi tror att det skulle kunna bli lagkrav på detta framöver när graden av autonomi i fordon ökar.
Leverans	Intern rapportering, Scania Intern rapportering, Volvo Cars

	Möjlig forskningsartikel

Arbetspaket 7	Fordons säkerhetsparametrar
Ansvarig	Niklas Wiberg, Scania
Övriga deltagare	Pontus Johnson, foreseeti Doktorand, KTH Robert Lagerström, KTH Penetrationstestare, F-Secure Säkerhetsexperter, Scania Jörgen Borg, Volvo Cars Säkerhetsexperter, Volvo Cars
Beskrivning av innehåll	Detta arbetspaket syftar till att välja ut delar av det domänspecifika språket och i detalj specificera värden för dessa utifrån expertbaserad säkerhetsanalys av verkliga fordonssystem. Genom denna analys bygger man viktiga delar av den expertkunskap som behövs i modellen som kvantitativa sannolikhetsvärden. Detta är också ett viktigt komplement till den övergripande testningen och valideringen i arbetspaket 5. Utifrån de viktigaste användarfallen för säkerhet i fordons-IT, t.ex. uppkoppling, cyberfysisk manipulation, stöldskydd och integritetsskydd, kommer viktiga attackvägar att väljas ut och analyseras i detalj. Dessa kan simuleras med hjälp av det domänspecifika språket och securiCAD och de kan penetrationstestas av experter. Simuleringarna i språket kan guida penetrationstestarna och penetrationstesterna kan validera eller ge viktigt input till språket. T.ex. kan parametrar iterativt behöva uppdateras med nya sannolikheter. Penetrationstester och säkerhetsutvärderingar För denna typ av tester brukar en så kallad ”grey box”-metod förespråkas, d.v.s. en metod där man får viss insyn i systemuppbyggnad och skyddsvärda objekt och tillgångar, för att sedan komplettera med penetrationsförsök av helt eller delvis okända element, kopplingar, gränssnitt etc. Det ger en bra balans och effektivitet i ett test, då man kan fokusera på det väsentliga, men ändå använda metodik som en verklig attackerare skulle använda. Sådana metoder kommer huvudsakligen att användas i detta arbetspaket för penetrationstesterna.
Metod/angreppssätt	Viktiga steg i detta arbetspaket är: Identifiera svagheter i fordons-IT Identifiera säkerhetsfunktioner i fordons-IT Utvinna dess säkerhetsegenskaper Sätta och testa parametrar inkl. hårdvara, mjukvara, sensorer, ställdon, och hela ECU-system Testa i labbet med verktyg Testa med penetrationstestare Simulera med securiCAD
Leverans	Rapport(er) som beskriver utfall av tester (kravspecifikation för det domänspecifika språket och implementationen)

	Forskningsartiklar

Arbetspaket 8	Kunskapsspridning
Ansvarig	Pontus Johnson, foreseeti
Övriga deltagare	Dan Johansson, foreseeti Personal, foreseeti Robert Lagerström, KTH Doktorand, KTH Niklas Wiberg, Scania Jörgen Borg, Volvo Cars
Beskrivning av innehåll	Detta arbetspaket syftar till att sprida kunskap om projektet och dess framsteg. Vi avser att synkronisera vår kunskapsspridning och samarbeta med det Vinnova FFI-sponsrade projektet DEx (identifiering, GAP-analyser och spridning av forskningsbehov inom fordons IT-säkerhet och Integritet). Forskningsartiklar i konferenser och tidskrifter Doktorsavhandling vid Kungliga Tekniska Högskolan Rapporter (interna och externa) Seminarier och föreläsningar Modeller (referensmodeller) för fordonssystem Spridning via branschorganisationer Nyheter, bloggar, etc. via sociala medier och projekthemsida
Leverans	Löpande spridning av material till tredje part

Tidsplanering

THREAT MOVE förväntas pågå från oktober 2017 till september 2021, det vill säga fyra år. Arbetet sker iterativt i arbetspaket 3-7, arbetspaket 2 (samt delar av 3) genomförs tidigt i projektet innan huvudaktiviteterna i paket 4-7 startar - då de är beroende av dess resultat. Arbetspaket 1 och 8 pågår parallellt med alla andra arbetspaket då dessa hanterar projektet och kommunicerar resultaten.

Tabell 1. Tidsplanering, ljusblå är mer lågintensivt arbete i iterativa arbetspaket.

	2017	2018				2019				2020				2021		
WP	*Q4*	*Q1*	*Q2*	*Q3*	*Q4*	*Q1*	*Q2*	*Q3*	*Q4*	*Q1*	*Q2*	*Q3*	*Q4*	*Q1*	*Q2*	*Q3*
1																
2																
3																
4																
5																
6																
7																
8																

Tabell 2. Milstolpar.

Milstolpe	Beskrivning	Datum
1	Projektavtal	2017-12-31
2	Ramverk för domänspecifikt språk	2017-12-31
3	Domänspecifikt språk, version 0.1	2018-06-30
4	securiCAD-modul, version 0.1	2018-06-30
5	Rapport – validering / Tool chain integration / parametrar, version 0.1	2018-12-31
6	Domänspecifikt språk, version 0.2	2019-06-30
7	securiCAD-modul, version 0.2	2019-06-30
8	Rapport – validering / Tool chain integration / parametrar, version 0.2	2019-12-31
9	Domänspecifikt språk, version 0.5	2020-06-30
10	securiCAD-modul, version 0.5	2020-06-30
11	Rapport – validering / Tool chain integration / parametrar, version 0.5	2020-12-31
12	Domänspecifikt språk, version 1	2021-06-30
13	securiCAD-modul, version 1	2021-06-30
14	Rapport – validering, version 1	2021-09-30
15	Doktorsavhandling	2021-09-30
16	Slutrapport	2021-09-30

Projektekonomi

Kostnaden för THREAT MOVE beräknas till 15 154 429 kr varav 7 079 900 kr äskas från Vinnova. De huvudsakliga kostnaderna för projektet utgörs av arbetstid för personer från KTH, foreseeti, Volvo Cars och Scania.

Tabell 3. Projektekonomi per arbetspaket och partner.

WP	*Budget per deltagande part*					*Total budget/WP*
	KTH	*foreseeti*	*Scania*	*Volvo Cars*	*F-Secure*	
1	750	250	122,5	52,5		1 175
2	500	475				975
3	1 000	510	122,5	52,5		1 685
4		2 000				2 000
5	1 000	560	700	300		2 560
6	250		350	150		750
7	1 000	700	2100	900	100	4 800
8	534	525	105	45		1 209
TOTALT						15 154 milj. SEK

Projektledning och projektdeltagare

THREAT MOVE leds av Universitetslektor Robert Lagerström (KTH) och bemannas med personal från KTH, foreseeti, F-Secure, Volvo Cars och Scania.

Från KTH deltar även Prof. Pontus Johnson och en ännu inte rekryterad doktorand. Robert Lagerström och Pontus leder forskningsgruppen Software Systems Architecture and Security (SSAS). SSAS har i snart tjugo år forskat på informationssystemarkitektur och -simulering, med ett starkt fokus på informationssäkerhet. Gruppen har publicerat omkring 200 granskade vetenskapliga artiklar i tidskrifter och konferenser. SSAS utgör en av parterna i det Nationella forskningscentret för resilienta informations- och styrsystem, de är representerade i den Kungliga Ingenjörsvetenskapsakademien (IVA), och deltar aktivt i internationell forskning tillsammans med

exempelvis Harvard i USA. Gruppens forskning har utmynnat i spinn-off-företaget foreseeti. Från KTH kommer en doktorand att medverka. Denna kommer att tillsättas när projektet startas.

Från foreseeti deltar Dan Johansson, Pontus Johnson, Per Eliasson, samt flertalet utvecklare. foreseeti grundades 2014 baserat på forskning från KTH, finansierat av riskkapitalbolaget InnoEnergy, under the European Institute of Technology (EIT). Ungefär 15 personer är anställda vid foreseeti, som har ett antal stora kunder inom bland annat finans-, verkstads- och konsultbranscherna. 2016 och 2017 vann foreseeti en plats på Ny Teknik och Affärsvärldens 33-lista över mest lovande unga teknikföretag. foreseeti utvecklar och säljer hotmodellerings och -simuleringsverktyget securiCAD.

F-Secure är ett av världens ledande cybersäkerhetsbolag. Företaget grundades 1988, och är listat på Helsingforsbörsen. F-Secure omsatte 158MEUR förra året och har idag över 1 000 anställda. De har tiotals miljoner konsumentkunder och 100 000 företagskunder. Affärsområdet Cyber Security Services har en portfölj tjänster inom IT-säkerhet, där F-Secure jobbar med världens mest krävande kunder. Testning och utvärdering av system är ett stort område. Från F-Secure kommer Christoffer Jerkeby och Olle Segerdahl att medverka. Båda har lång erfarenhet av penetrationstestning och säkerhetsutvärderingar.

Från Scania, som inte tarvar någon introduktion i detta sammanhang, deltar Niklas Wiberg, systemarkitekt med fokus på cybersäkerhet i fordonens elektriska system och flera av Niklas kollegor.
Avslutningsvis så deltar även Volvo Cars med Jörgen Borg, projektledare och ämnesexpert inom cybersäkerhet inriktad mot fordonets elektriska system samt även andra utvecklare och experter från Volvo Cars.

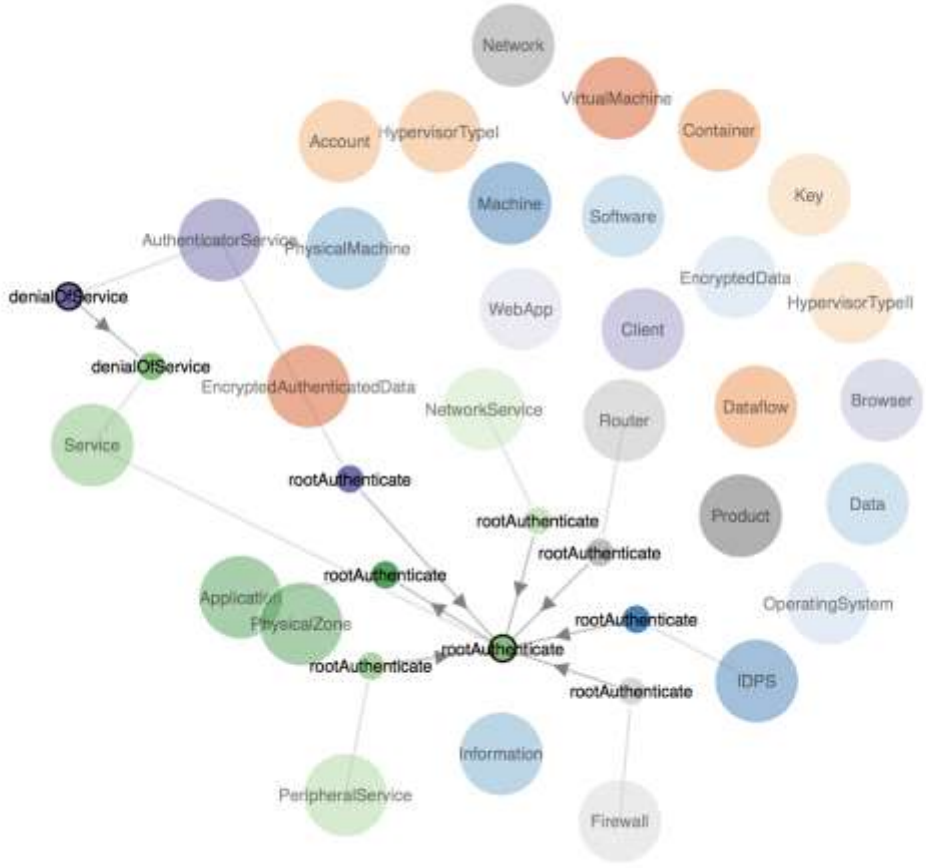
6.2 MAL – Model-based Attack graph Language

Under förstudien har vi börjat skissa på det ramverk som kommer att utgöra grunden för vårt domän-specifika språk för säkerhet i fordons-IT. En början / ett utkast på en akademisk artikel som presenterar MAL finns men är inte redo att skickas in ännu. Följande är taget från sammanfattningen (på engelska):

”Security is a key issue in our connected society. Threat modeling and attack graphs are used in order to analyze and simulate this complex Information and Communication Technology (ICT) environment. However, most of the approaches available are based on General Purpose Languages (GPLs) like Java, making these difficult to modify, extend, and re-use. Studies have shown that Domain-Specific Languages (DSLs) outperform GPLs. This paper presents MAL (Model-based Attack-graph Language), a Domain Specific Language (DSL) for attack simulation.”

Skissen på MAL som sådan finnas bifogat i Appendix I.

En implementation av MAL i java och en grafisk vy finns att se i Figur 2.



Figur 2. En visualisering av en attackgraf på meta-nivå implementerar i java med hjälp av MAL-ramverket.

7 Spridning och publicering

7.1 Kunskaps- och resultatspridning

Hur har projektresultatet använts och spridits?	Markera med X	Kommentar
Öka kunskapen inom området	X	Vi har pratat med flertalet aktörer inom fordonsbranschen och bl.a. medverkat på Vehicle ICT Arena - Innovation Bazaar[4] med både monter och presentation.
Föras vidare till andra avancerade tekniska utvecklingsprojekt	X	Huvudleverabel är en ansökan till ett större projekt.
Föras vidare till produktutvecklingsprojekt		
Introduceras på marknaden		
Användas i utredningar/regelverk/ tillståndsärenden/ politiska beslut		

7.2 Publikationer

Ett grovt utkast till en forskningsartikel finns men är inte inskickad ännu. Denna avser att presentera ramverket som hotmodellerings- och simuleringsspråket kommer att baseras på.

- MAL (Model-based Attack-graph Language): A Domain Specific Language for Attack Simulations, av Robert Lagerström, Pontus Johnson och Mathias Ekstedt.

Utkastet till själva ramverket, MAL (Model-based Attack-graph Language), är publicerat online[5] och bifogat i appendix.

8 Slutsatser och fortsatt forskning

Vår slutsats är att det är värt att fortsätta med denna forskning och innovation genom ett större projektet inom ramen för Vinnova och FFI.

Vi har under förstudien hittat fler intresserade partners och totalt sett blir vi Kungliga Tekniska Högskolan, Foreseeti, F-Secure, Volvo Cars och Scania som söker det större THREAT MOVE-projektet tillsammans.

Vi har också under förstudien planerat och detaljerat THREAT MOVE, se kapitel 6.1.

Samt att vi har en grund för ramverket som THREAT MOVE kan bygga vidare på. Det vi här kallar MAL och som kommer att möjliggöra ett flexibelt hotmodelleringsspråk för fordons-IT. See Appendix I för detaljer.

Fortsättning blir att helt enkelt, om vi beviljas THREAT MOVE, fortsätta utveckla MAL, det fordonsspecifika hotmodellerings- och simuleringsspråket, testa detta, implementera i användarvänlig prototyp, och sprida våra resultat.

[4] https://vehicle.lindholmen.se/evenemang/vehicle-ict-arena-innovation-bazaar-6
[5] https://www.kth.se/profile/robertl/page/the-model-based-attack-graph-language-mal

9 Deltagande parter och kontaktpersoner

Part	Kontakt	Email	Logga
Kungliga Tekniska Högskolan (KTH)	Robert Lagerström	robertl@kth.se	
Scania	Niklas Wiberg	Niklas.Wiberg@scania.com	
Foreseeti	Joakim Nydrén	Joakim.nydren@foreseeti.com	

10 Referenser

Alberts, C. J. och A. Dorofee. Managing information security risks: the OCTAVE approach, Addison-Wesley, 2002.

Blom, Rikard, Matus Korman, Robert Lagerström, och Mathias Ekstedt. “Analyzing attack resilience of an advanced meter infrastructure reference model.” Cyber-Physical Security and Resilience in Smart Grids (CPSR-SG), IEEE, 2016.

Checkoway, Stephen, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno. “Comprehensive Experimental Analyses of Automotive Attack Surfaces.” USENIX Security Symposium. 2011.

Ekstedt, Mathias, Pontus Johnson, Robert Lagerström, Dan Gorton, Joakim Nydrén, och Khurram Shahzad. “Securi CAD by Foreseeti: A CAD Tool for Enterprise Cyber Security Management.” IEEE 19th International Enterprise Distributed Object Computing Workshop, pp. 152-155. IEEE, 2015.

Flores, Waldo Rocha, och Mathias Ekstedt. “Shaping intention to resist social engineering through transformational leadership, information security culture and awareness.” Computers & Security 59: 26-44, 2016.

Holm, Hannes, Mathias Ekstedt, och Dennis Andersson. “Empirical analysis of system-level vulnerability metrics through actual attacks.” IEEE Transactions on dependable and secure computing 9, no. 6: 825-837, 2012.

Johansson, Erik, och Pontus Johnson. “Assessment of enterprise information security-an architecture theory diagram definition.” Proc. of CSER 5, 2005.

Johnson, Pontus, Alexandre Vernotte, Mathias Ekstedt, och Robert Lagerström. “pwnPr3d: an Attack Graph Driven Probabilistic Threat Modeling Approach.” International Conference on Availability, Reliability and Security (ARES), 2016.
Johnson, Pontus, Dan Gorton, Robert Lagerström, och Mathias Ekstedt. “Time between vulnerability disclosures: A measure of software product vulnerability.” Computers & Security 62: 278-295, 2016.
Jürjens, J. “UMLsec: Extending UML for secure systems development,” UML 2002: The Unified Modeling Language. Springer, 2002.
412–425.
Kohnfelder, L. och P. Garg. “The threats to our products.” Microsoft Interface, Microsoft Corporation, 1999.
Kordy, Barbara, Ludovic Piètre-Cambacédès, och Patrick Schweitzer. ”DAG-based attack and defense modeling: Don’t miss the forest for the attack trees.” Computer science review 13:1-38, 2014.
Koscher, Karl, Alexei Czeskis, Franziska Roesner, Shwetak Patel, Tadayoshi Kohno, Stephen Checkoway, Damon McCoy et al. “Experimental security analysis of a modern automobile.” IEEE Symposium on Security and Privacy, pp. 447-462. IEEE, 2010.
Lund, M. S., B. Solhaug, och K. Stølen, Model-driven risk analysis: the CORAS approach. Springer Science & Business Media, 2010.
Ou, X., S. Govindavajhala, och A. W. Appel. “Mulval: A logic-based network security analyzer.” USENIX security, 2005.
Petit, Jonathan, and Steven E. Shladover. “Potential cyberattacks on automated vehicles.” IEEE Transactions on Intelligent Transportation Systems 16, no. 2:546-556, 2015.
Saitta, P., B. Larcom, och M. Eddington. “Trike v. 1 methodology document [draft].” 2005.
Shostack, Adam. ”Experiences threat modeling at microsoft.” Modeling Security Workshop. Dept. of Computing, Lancaster University, UK 2008.
Sommestad, Teodor, Mathias Ekstedt, och Pontus Johnson. “A probabilistic relational model for security risk analysis.” Computers & security 29, no. 6: 659-679, 2010.

11 Appendix

```
grammar MAL;

compilationUnit
	:		categoryDeclaration* 'associations {' associationDeclaration* '}' EOF
	;

// Assets are sorted in categories, probably for UI reasons.
categoryDeclaration
	:		'category' Identifier description? '{' assetDeclaration* '}'
	;

// Declare an asset
assetDeclaration
	:		asset Identifier ('extends' Identifier)? description? rationale? assumptions? '{'
stepDeclaration* '}'
	;

stepDeclaration
	:		attackStepDeclaration
	|		existenceStepDeclaration
	;
```

```
asset
        : 'asset'
        | 'abstractAsset'
        ;

// Here, we define associations
associationDeclaration
        :               Identifier '[' Identifier ']' multiplicity leftRelation Identifier rightRelation multiplicity '['
Identifier ']' Identifier description? rationale? assumptions?
        ;

rightRelation
        :               '-->'
        ;

leftRelation
        :               '<--'
        ;

// 1   An asset must be connected to exactly one of the related assets
// 0-1 An asset can be connected to at most one of the related assets
// 1-* An asset must be connected to at least one of the related assets
// *   An asset can be connected to any number of the related assets
multiplicity
        :               '1'
        |               '0-1'
        |               '1-*'
        |               '*'
        ;

// Declare attack steps
attackStepDeclaration
        :               attackStepType Identifier ttc? description? rationale? assumptions? children?
containedSteps?
        ;

// Declare existence steps
existenceStepDeclaration
        :               existenceStepType Identifier ttc? description? rationale? assumptions?
existenceRequirements? children? containedSteps?
        ;

existenceStepType
        :               'E'
        |               '3'
        ;

// List the existenceRequirements.
existenceRequirements
        :               '<-' Identifier (',' Identifier)*
        ;

// List the children.
children
        :               '->' expressionName (',' expressionName)*
```

```
        ;

// An attack step can have sub-attack-steps only to better organize the code.
containedSteps
        : '{' attackStepDeclaration* '}'
        ;

// The description is meant for the end user to read
description
        : 'info:' StringLiteral
        ;

// The rationale is meant to justify the concept for securiLang designers.
rationale
        : 'rationale:' StringLiteral
        ;

// The assumptions list attack paths we chose to not include, modeling assumptions, etc.
assumptions
        : 'assumptions:' StringLiteral
        ;

StringLiteral
        :       "" StringCharacters? ""
        ;
fragment
StringCharacters
        :       StringCharacter+
        ;
fragment
StringCharacter
        :       ~["\\]
        ;

// The local TTC declaration. (The allowed distributions could be specified here, in the grammar.)

ttc
        : '[' Identifier '(' formalParameters? ')' ']'
        ;

// The attack step(s) whose TTC determine the color of the asset in the UI.
mostImportant
        :       '!'
        ;

// Should the user see this attack step?
visibility
        :       '+'
        |       '-'
        ;

// | OR attack step.
// & AND attack step.
// # defense step.
// t CPT attack step.
```

```
attackStepType
        :               '|'
        |               '&'
        |               '#'
        |               't'
        ;

expressionName
        :               Identifier
        |               ambiguousName '.' Identifier
        ;

ambiguousName
        :               Identifier
        |               ambiguousName '.' Identifier
        ;

Identifier
        :               JavaLetter JavaLetterOrDigit*
        ;

formalParameters
        :               DecimalFloatingPointLiteral (',' DecimalFloatingPointLiteral)*
        ;

DecimalFloatingPointLiteral
        :               Digits '.'? Digits?
        ;

Digits
        :               Digit Digit*
        ;

Digit
        :               [0-9]
        ;



fragment
JavaLetter
        :               [a-zA-Z$_] // these are the "java letters" below 0xFF
        |               // covers all characters above 0xFF which are not a surrogate
                        ~[\u0000-\u00FF\uD800-\uDBFF]
                        {Character.isJavaIdentifierStart(_input.LA(-1))}?
        |               // covers UTF-16 surrogate pairs encodings for U+10000 to U+10FFFF
                        [\uD800-\uDBFF] [\uDC00-\uDFFF]
                        {Character.isJavaIdentifierStart(Character.toCodePoint((char)_input.LA(-2),
(char)_input.LA(-1)))}?
        ;

fragment
JavaLetterOrDigit
        :               [a-zA-Z0-9$_] // these are the "java letters or digits" below 0xFF
        |               // covers all characters above 0xFF which are not a surrogate
```

```
                        ~[\u0000-\u00FF\uD800-\uDBFF]
                        {Character.isJavaIdentifierPart(_input.LA(-1))}?
        |               // covers UTF-16 surrogate pairs encodings for U+10000 to U+10FFFF
                        [\uD800-\uDBFF] [\uDC00-\uDFFF]
                        {Character.isJavaIdentifierPart(Character.toCodePoint((char)_input.LA(-2),
(char)_input.LA(-1)))}?
        ;

//
// Whitespace and comments
//

WS  :  [ \t\r\n\u000C]+ -> skip
    ;

LINE_COMMENT
    :   '//' ~[\r\n]* -> skip
    ;
```