UNIVERSITATEA POLITEHNICA DIN BUCUREȘTI
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DEPARTAMENTUL DE CALCULATOARE

Teză de doctorat

în Calculatoare și Tehnologia Informației

Securitatea în sistemele de operare Linux din automobile

Jan Alexandru Văduva

Coordonator științific:
Prof. dr. ing. Răzvan-Victor RUGHINIȘ

BUCUREŞTI

2021

Universitatea POLITEHNICA București

Abstract

Securitatea în sistemele de operare Linux din automobile

de Jan Alexandru Văduva

Coordonator științific

Profesor Răzvan-Victor RUGHINIȘ

Departmentul de Calculatoare

Abordările auto moderne sunt compuse dintr-o multitudine de module extrem de complexe, cu funcționalități extinse, care sunt interconectate printr-o mare varietate de opțiuni de rețea, majoritatea informațiilor referitoare la acestea fiind păstrate secrete de industria auto. În ultimii ani s-a dovedit că acest tip de perspectivă devine extrem de problematică atunci când expuneți aceste module lumii exterioare transformându-le în ținte cu o valoare mare expuse atacurilor cibernetice la distanță. Cel mai rău scenariu ar fi atunci când atacatorul obține acces complet la sistemele critice din jurul mașinii, cum ar fi motorul, sistemul de frânare sau direcția.

Această teză propune o schimbare de perspectivă în care mai mulți actori, cunoștințele mulțimii și securitatea vitală se află la baza proiectării arhitecturii. Pentru a îndeplini acest obiectiv, am propus un nou model de arhitectură cu dispozitive centrale și marginale care au cerințe de securitate și suport de apărare împotriva defectelor diferite. Pentru arhitectura concepută, prezentăm funcționalități de îmbunătățire a securității pentru a o spori conform noii paradigme ce vizează securitate încă din faza de proiectare. Prima soluție pe care o prezentăm este o tehnologie bazată pe ARM TrustZone, urmată de o componentă de analiză binară și un sistem de detectare a intruziunilor pe sistemul de operare gazdă. În plus, prezentăm o tehnică capabilă să atenueze anumite atacuri și să asigure integritatea aplicațiilor critice pe baza analizei codului sursă a actualizărilor înainte de a le aplica. În cele din urmă, am evaluat implementările propuse și am demonstrat capacitățile acestora de a atenua vectorii de atac potențiali disponibili în sistemul de operare din automobile.

CUPRINS

MULȚUMIRI

Îmi este greu să scriu mulțumiri deoarece consider că nu există cuvinte capabile să exprime nivelul de recunoștință pe care îl am pentru cei care m-au susținut de-a lungul procesului de redactare a acestei teze. Procesul s-a dovedit cu adevărat consumator de timp și cred că rezultatele finale au beneficiat mult de pe urma unei comunități formate din profesori, colegi, prieteni și familie. Este o onoare pentru mine să am ocazia să-mi exprim în scris mulțumirile față de acești oameni care m-au susținut și inspirat pe tot parcursul procesului.

Sunt profund recunoscător coordonatorului meu Răzvan Rughiniș, membrilor comisiei mele de teză Răzvan Deaconescu, Emil Slușanschi, Catalin Leordeanu și altor colaboratori de la Universitatea Politehnica din București Nicolae Țăpuș, Mihai Chiroiu, Dragos Niculescu, Alexandru Radovici, Ștefan-Dan Ciocîrlan care m-au ghidat în mod constant spre direcția corectă.

Sunt recunoscător colegilor mei din grupul Systems Daniel Rosner, Iulia Florea, Ioana Culic, Eduard Stăniloiu, Răzvan Nițu, Flavia Oprea, Octavian Grigorescu, Cristian Trancă, Lucian Mogoșanu, Florin Stancu, Mihai Carabaș, Cristian Pătru care mi-au furnizat informații utile de cercetare și comentarii la lucrările publicate.

Sunt norocos să lucrez și cu Daniel Pirjan, Ovidiu Mihalachi, Bogdan Mirea, Andreea Guran, Cristi Irimia, Dragos Ducu, Tudor Ciochină, Lucian Perisoara care au fost o sursă constantă de încurajare și experiență. De asemenea, trebuie să-i mulțumesc lui Daniel Bornaz, care a avut răbdarea să-mi modeleze abilitățile de Linux și inginerie și m-a ajutat de-a lungul procesului.

Le mulțumesc tuturor studenților cu care am colaborat, am făcut schimb de idei și am petrecut un timp minunat. Sunt foarte fericit că i-am întâlnit, există o mulțime de oameni, foarte greu de menționat pe toți, dar pot numi aici grupul UPBDrive unde încercăm să construim o mașină de curse Formula Student, grupurile de USO și RL, Radu Chișcariu, Vlad Pașca sau Ștefan Dascălu.

Aș dori, de asemenea, să le mulțumesc soției, părinților, fratelui și chiar noului meu copil născut, care au oferit sprijinul și încurajarea lor de-a lungul anilor de studiu, cercetare și scriere pentru teză.

Mulți alții au contribuit la dezvoltarea ideilor în acest proces, au oferit sprijin sau sfaturi. Trebuie să le mulțumesc tuturor pentru că am făcut toate acestea plăcute chiar și în cele mai grele momente.

Chapter 1. Înțelegerea securității auto

În acest capitol voi defini și descrie contextul și tehnologiile utilizate. De asemenea, va conține principalele întrebări și contribuții de cercetare furnizate pentru domeniul securității auto, amenințărilor sale, vulnerabilității și contramăsurilor.

1.1 Context

Aici discutăm despre schimbările pe care open source le-a implicat diferitelor industrii și companii, împreună cu avantajele sale:

- Cost minim pentru cercetare și dezvoltare pentru noi inovații
- Avantaj pentru o productivitate mai bună a dezvoltării
- Integrarea sau implicarea utilizatorilor sau a părților interesate în procesul de dezvoltare
- Calitate mai bună a produselor și satisfacții ale clienților
- Oferă acces pentru a coopera pentru inovații interne și externe

De asemenea, oferim un argument pentru rolul sursei deschise în companiile auto și, în același timp, avantajele unei astfel de tranziții pentru toți acționarii implicați:

Înțelegere a modului de funcționare a mașinii: industria auto ne oferă câteva produse incredibile, cu o mulțime de componente electronice interconectate și puține informații despre modul în care funcționează. Învățând modul în care componentele interacționează unul cu celălalt sau cu lumea exterioară, vom fi mai buni la determinarea și repararea problemelor.

Lucrul la sistemele electrice auto: Vehiculele devin mai electrice decât mecanice, de obicei sunt cunoscute doar de către reprezentanți și alți acționari auto. Înțelegerea modului în care funcționează aceste componente electronice din acele informații sau din informațiile pe care producătorii de automobile le externalizează poate ajuta la rezolvarea problemelor și la furnizarea de instrumente pentru a le investiga.

Modificare a mașinii: Înțelegerea modului în care funcționează mașina ajută nu numai la posibilitatea de a oferi îmbunătățiri, ci și la adăugarea unui suport suplimentar pentru perifericele terțe, care ar putea funcționa perfect.

Descoperire a componentelor nedocumentate: majoritatea componentelor disponibile în mașini sunt nedocumentate sau dezactivate. Învățarea modului de utilizare a acestora este benefică și îmbunătățește funcționalitatea și potențialul mașinii.

Validarea securității vehiculului: cu câțiva ani în urmă, până la articolul din 2010, care prezintă exploatările unei mașini, siguranța vehiculului nu a explicat amenințările la adresa securității. După acel eveniment, toată lumea a început să se uite la o mașină ca la orice sistem de desktop. Acest lucru înseamnă, de asemenea, că avem nevoie de validarea și auditul mașinilor noastre, deoarece acestea sunt cele care îngrijesc întreaga noastră familie în jurul orașelor și în afara lor, la o gamă întreagă de viteze. Știind mai multe despre cum să-ți spargă mașina, îi ajută pe toți să afle despre vulnerabilitățile lor și, în cele din urmă, să ajute la îmbunătățirea standardelor de siguranță.

Ajutor pentru industria auto: cunoștințele acumulate oferă îndrumări pentru descoperirea vulnerabilităților și soluții pentru protejarea acestui ecosistem. Acest lucru se poate traduce ca beneficii pentru industria auto, ajutându-i să ofere implementări pentru practicile de securitate și pentru alți cercetători, oferind o modalitate de comunicare a constatărilor acestora.

1.2 Domeniul tezei

Scopul acestei teze este de a analiza implicațiile și impactul metricilor de securitate aplicate industriei auto, una în care până în 2010 securitatea nu era considerată o preocupare. Când discutăm despre metricile de securitate, discutăm despre mai multe categorii. Există mai multe modalități de clasificare a acestora. O astfel de metodă este responsabilitatea nivelului de maturitate pentru procesele care contribuie la securitatea sistemului, alta este reprezentată de gradul de prezență pentru o anumită caracteristică de securitate.

Pe lângă metricile de securitate a informațiilor, alte domenii, cum ar fi cel financiar, au reușit să facă progrese în ceea ce privește metodele cantitative de măsurare a riscurilor, ceea ce pentru securitatea informațiilor ar însemna ipoteze realiste și rezultate fiabile. Dezvoltarea în domeniul măsurilor și metricilor de securitate este benefic pentru domenii de la proiectare și implementare până la operațiuni. Cu toate acestea, măsurarea securității este o problemă greu de rezolvat. Alte dovezi sunt reprezentate de faptul că, metricele de securitate la nivel de întreprindere, au fost incluse în lista problemelor dificile menținută de consiliul de cercetare INFOSEC.

Trebuie să facem pentru securitatea informațiilor ceea ce au făcut quant-urile pentru industria financiară. Trebuie să fim capabili să înțelegem, să punctăm, să împachetăm, să cuantificăm, să măsurăm și să putem, de asemenea, să tranzacționăm riscurile într-un mod eficient, similar cu modul în care evoluează piețele financiare. O soluție rapidă a acestei probleme nu va fi ceva la care ne așteptăm, de asemenea, atunci când acest lucru se va întâmpla cel mai probabil vor exista câteva aspecte care vor rămâne câteva activitat ușor de atins. Unii dintre factorii care au un impact al acestui progres sunt:

- Nu sunt disponibili estimatori buni pentru securitatea sistemului.
- Multă dependență de intrările subiective, parte din ele generate de utilizatori.
- Mijloace de măsurare extinse și uneori false.
- Lipsa unei înțelegeri adecvate a modului de funcționare a mecanismelor de securitate.

Cu toate acestea, în ultima vreme pentru industria auto paradigma a început să se schimbe de când WEC.29 al UNECE a început să facă procesul de certificare pentru un sistem de management al securității cibernetice obligatoriu și a forțat practic industria să implementeze ISO / SAE 21434 [130]. Toate acestea, deoarece nu există o formulă matematică care să poată fi utilizată pentru a obține nivelul de încredere cerut de fiecare dintre cei care lucrează cu componente mai mici, ce ar putea fi validate formal sau semiformale în funcție de cerințele componentei. Deci problema se învârte în jurul componentelor pe care ar trebui să ruleze aplicația, în funcție de nivelul de securitate necesar.

1.3 Întrebări de cercetare

După cum s-a menționat deja în numeroase lucrări de cercetare și chiar în mass-media, în ultima perioadă vehiculul disponibil astăzi este foarte diferit de cel pe care îl aveam acum 10 ani, nu numai asta, ci și tehnologia implicată în jurul său s-a schimbat și a fost încorporată în interior. Internetul a ajuns la vehicul și un număr mare de servicii vor fi disponibile în interiorul mașinii noastre în anii următori.

După cum se indică în [20], vehiculele disponibile astăzi pe șosea au între 30 și 100 de computere în interior, numite și ECU-uri în industria auto. Acest lucru este chiar mai mare decât

majoritatea startup-urilor și companiilor mici, dar fără suport de securitate. Numărul de linii din interiorul vehiculelor se învârte astăzi în jurul a sute de milioane și, din moment ce putem avea o vulnerabilitate disponibilă pentru fiecare 10 linii de coduri și faptul că aceste vehicule vor fi conectate prin internet la infrastructură și la alte vehicule. Acest lucru face ca securitatea să fie imperativă și, așa cum este documentat într-un raport de securitate din 2016 până în 2019, numărul problemelor de securitate raportate în interiorul vehiculelor a crescut de șapte ori.

Pentru a face față acestor provocări, am pornit de la următoarele întrebări de cercetare:

- Q1 Cât de pregătită este industria auto pentru a trece de la securitate prin obscuritate la securitate prin design?
  - a De ce avem nevoie de trecerea la securitate prin design??
- Q2 Cum ar arăta o arhitectură digitală auto în care am putea aplica securitatea din etapa de proiectare cu mai mulți actori, cunoștințele unor mulțimi variate și securitatea vitală cu noi servicii și tehnologii integrate?
  - a Ce tip de senzori și servicii pot fi conectate într-o astfel de arhitectură?
  - b Cum putem face acest proces de integrare mai ușor de înțeles de către indivizi non-tehnici?
- Q3 Ce îmbunătățiri ar putea fi utilizate pentru a crește securitatea din faza de proiectare existentă în industria auto?

1.4 Contribuțiile tezei

Această teză a început cu o analiză care validează problemele disponibile în interiorul vehiculelor existente datorită paradigmei de securitate prin obscuritate, care a fost promovată până acum în industrie. Am validat aceste probleme construind, pe baza hardware-ului Raspberry Pi, o platformă care ar putea fi utilizată ca platformă deschisă care să faciliteze interacțiunea cu vehiculul.

Pe această platformă am reușit să integrăm platforma web Wyliodrin pentru a facilita integrarea noilor senzori și funcționalități. Din moment ce am vrut să construim un banc de test care să simuleze o arhitectură securizată din etapa de proiectare, am ajuns la o problemă secundară a domeniului, faptul că schimbarea de la securitate prin obscuritate la securitate prin proiectare ar necesita interacțiunea cu tehnologiile pentru grupuri de indivizi care nu au un background tehnic. Am reușit să rezolvăm acest lucru cu ajutorul lui Wyliodrin, care a fost, de asemenea, utilizat ca mecanism de extindere a funcționalităților într-un mod ușor de înțeles și programat de către acești indivizi. Am dus platforma și mai departe și am integrat un companion Android pentru a facilita interacțiunea și, de asemenea, a încorpora puterea smartphone-ului în ecosistem.

Deoarece toate aceste funcționalități se dovedesc complexe pentru sistemul de operare Raspbian, a fost necesară optimizarea, pentru aceasta am folosit o distribuție Linux bazată pe proiectul Yocto. Bineînțeles că am ales ca punct de pornire distribuția Automotive Grade Linux (AGL), care are și suport hardware pentru Raspberry Pi și am adăugat suportul Wyliodrin necesar în plus. Tot ce ni s-a cerut să facem dacă a fost furnizată o nouă funcționalitate a fost să ne asigurăm că sistemul de operare avea și dependențele de bază integrate în interior.

În acest moment am reușit să stabilim arhitectura digitală deschisă pe care am putea să o folosim ca banc de testare. Acum trebuie să asigur ecosistemul. Pentru a face acest lucru, ne uităm la sistemul securizat de actualizare a software-ului auto și am analizat mecanismul de probă de execuție de încredere pentru introspecția sistemului implementat prin apeluri TrustZone SMC care

s-au dovedit a nu fi clasificate în industrie pentru scopurile noastre. Am propus o întărire a securității în care folosim analiza binară și un sistem de detectare a intruziunilor gazdă responsabil pentru detectarea anomaliilor. La aceasta am adăugat, de asemenea, o analiză statică a codului, unde oferim un mecanism de notare a impactului pe suprafața de atac a oricărei actualizări utilizate pentru sistem. Cu toate acestea am reușit să construim un banc de testare minim arhitectural pentru securitatea din etapa de proiectare și detaliile lucrării vor fi prezentate în secțiunile următoare.

Pentru a rezuma contribuțiile și a le potrivi cu întrebările de cercetare definite în capitolul anterior, oferim următorul rezumat:

A1 Cu ajutorul muncii desfășurate în [3] cu hardware-ul Raspberry Pi, vă prezentăm starea sistemelor așa cum sunt, cât de ușor poate fi exploatată o vulnerabilitate și consecințele acestei acțiuni, ceea ce ar însemna și obținerea securității prin design așa cum este descris în stadiul tehnicii [211].
   a. O analiză a stadiului tehnicii [211] asupra procesului de standardizare care se face în acest moment a prezentat o gravă lipsă de securitate și o lipsă de conștientizare cu privire la impactul noilor tehnologii asupra industriei auto.

A2 Pornind de la lucrările de la A1, unde am dezvoltat un sistem centralizat bazat pe Raspberry Pi, o platformă hardware disponibilă pe scară largă, am început să construim o arhitectură digitală auto în care imităm mai mulți actori, cunoașterea mulțimii și integrăm o serie de noi servicii și tehnologii. Pe platforma hardware am configurat o distribuție Linux bazată pe proiectul Yocto personalizat [118] și [119] pentru a avea control deplin asupra capabilităților hardware și am adăugat extensii de top pentru control și personalizare mai ușoare. Pentru extensibilitate și ușurință în utilizare, ne-am oprit peste Wyliodrin și am adăugat suport pentru senzori suplimentari de controlat ca parte a [117], am adăugat suport pentru recunoașterea vorbirii ca parte a [115] și pentru a controla cu ușurință toate acestea am definit un Android însoțitor cu [116].
   a. Integrăm noi tipuri de senzori și servicii cu ajutorul muncii dezvoltate în contribuțiile [115], [116] și [120] descrise mai sus.
   b. Facilităm procesul de integrare pentru părțile interesate cu diverse experiențe, cu ajutorul contribuțiilor disponibile în [117], [118] și [119], de asemenea, descrise în detalii în secțiunea A2.

A3 Am investigat sistemul securizat de actualizare a software-ului auto și am aruncat o privire asupra mecanismului probelor de execuție de încredere [111] pentru introspecția sistemului implementat prin apeluri TrustZone SMC care s-au dovedit a nu fi clasificate în industrie pentru scopurile noastre. Am propus o soluție de întărire a securității atât cu o soluție de analiză binară [109], cât și cu un sistem de detectare a intruziunilor gazdelor [110], ambele oferind o îmbunătățire atât pentru introspecția statică, cât și dinamică a sistemului responsabil pentru detectarea anomaliilor. La aceasta am adăugat și analiza statică a codului cu [121] unde oferim un mecanism de notare a impactului pe suprafața de atac a oricărei actualizări folosite în sistem.

1.5 Structura tezei

În această secțiune, parte a capitolului 1, prezentăm structura tezei și o scurtă prezentare generală a modului în care ne-am abordat obiectivele în fiecare capitol și ce contribuții au fost incluse în acestea, împreună cu modul în care capitolul influențează obiectivele tezei.

Această teză este structurată în șapte capitole, fiecare abordând unul dintre cele patru obiective principale: identificarea cât de pregătită este industria pentru o trecere spre securitate prin proiectare de la securitate prin obscuritate; să identifice modul în care o arhitectură digitală care a integrat mai mulți senzori și servicii ar putea aplica securitatea din etapa de proiectare și modul în care aceasta ar putea fi integrată în cadrul procesului de dezvoltare pentru indivizii non-tehnici; identificarea îmbunătățirilor care ar putea fi realizate pentru a extinde securitatea existentă a sistemului.

În capitolul 2 oferim un studiu al literaturii existente privind cercetarea științifică și standardele de securitate elaborate și aplicabile în industria auto și modul în care acestea au implicații asupra tuturor proceselor din industrie. Pe baza acestui fapt, am prezentat cum ar putea arăta un ciclu de viață sigur al dezvoltării software-ului și implicațiile acestora asupra tuturor fazelor de dezvoltare pentru o arhitectură hardware de referință. Această lucrare este tratată într-un singur articol.

În capitolul 3, deturnăm sistemele ECU din interiorul vehiculului și le conectăm la lumea exterioară, permițând posibilitatea de a accesa vehiculul de la distanță într-un mod sigur, printr-o interfață ușor de utilizat, utilizând software-ul open source și platforma hardware Raspberry Pi. Această lucrare este acoperită într-un articol publicat.

În capitolul 4, integrăm proiectul Yocto și oferim un sistem de operare configurabil personalizat pentru platforma hardware, încă unul potrivit cerințelor auto și integrăm platforma web Wyliodrin pentru a integra cu ușurință suportul hardware nou și, de asemenea, pentru a ajuta persoanele non-tehnice să înțeleagă a noțiunilor teoretice de informatică cu o curbă de învățare mai mică și să poată oferi aplicații legate de nevoile lor. Această lucrare este acoperită într-un articol și două cărți care au fost publicate în timpul lucrării pentru teza de doctorat.

În capitolul 5, oferim asistență pentru un serviciu care furnizează comenzi vocale ca modalitate de extindere a serviciilor acceptate și oferim un suport însoțitor Android nu numai pentru a facilita interacțiunea cu bancul de testare, ci și pentru a oferi extensibilitate pentru acesta. Această lucrare acoperă trei articole pe care le-am publicat.

În capitolul 6 luăm ceea ce am construit în capitolele anterioare și asigurăm mecanisme pentru securizarea acestui ecosistem pe care l-am construit. Folosim analize de execuție de încredere, binare și de apel de sistem atât pentru timpul de construire, cât și pentru evaluarea timpului de rulare a sistemului de operare și, de asemenea, oferim un mecanism pentru analiza actualizărilor înainte de a le aplica în sistem. Toate acestea acoperă patru articole publicate pe care le includem în această teză.

Capitolul 7 reprezintă concluzia acestei lucrări oferind o imagine de ansamblu asupra tuturor contribuțiilor și definind spațiul explorator viitor și modul în care lucrarea prezentă ar putea fi îmbunătățită.

Chapter 2. Stadiul domeniului

În acest capitol discutăm principalele activități științifice și de standardizare desfășurate în domeniul auto, o analiză a celor mai semnificative programe de cercetare dezvoltate sub umbrela Comisiei Europene, care a oferit baza cercetărilor efectuate în industria auto. Pe baza acestei cercetări, prezentăm un caz de utilizare pentru un proces de dezvoltare a software-ului specific auto, care implică o metodologie TARA modificată aplicată dezvoltării unei arhitecturi hardware de referință, în care securitatea și siguranța joacă un rol central.

Privind industria automobilelor de-a lungul istoriei, siguranța existenței sale a fost considerată una dintre cele mai importante caracteristici, lăsând securitatea aproape neadresată - în special pentru componentele software intensive care utilizează securitatea prin obscuritate ca opțiune implicită.

Astăzi privim vehiculele ca fiind doar o componentă a ecosistemului de mobilitate conectată și orașe inteligente și, în acest context, vehiculul conectat este considerat cea mai critică componentă, cu o multitudine de puncte de atac disponibile, dar și din sistemele și infrastructura backend. Atât de mult, încât reglementarea CEE-ONU a considerat securitatea cibernetică pe primul loc pentru agenda auto pentru peste 60 de piețe, pe măsură ce devine relevantă omologarea de tip.

După cum este descris de proiectul HoliSec [127], ingineria de securitate reprezintă o disciplină de inginerie preocupată de securitatea unui sistem, aceasta include toate procesele, de la proiectarea sistemului până la implementare și întreținere; cu toate acestea, așa cum menționează ei înșiși, mecanismele de securitate a datelor pentru sistemele încorporate legate de siguranță sunt punctul cheie pentru securitatea auto și necesită cercetări și investigații.

Odată cu problemele au sosit soluțiile disponibile și au fost proiectate arhitecturi interne adecvate, cele mai multe dintre acestea ținând cont de faptul că ECU-urile interne din mașină sunt limitate la preț. Deoarece au existat atât de multe soluții și articole bune scrise pe partea mecanismului de protecție, am decis să le prezint în Figura 2.1 în scopul lizibilității:

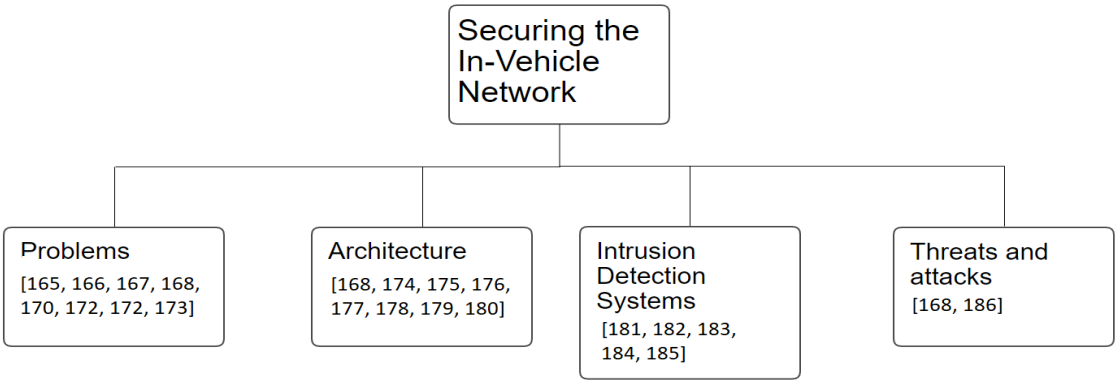

Figura 2.1. Mecanisme de protecție pentru rețelele din vehicul

Rezumatul lucrărilor a fost că mecanismele tradiționale care au oferit rezultate bune în alte industrii ar putea fi utilizate cu succes și în industria auto. Codurile de autentificare ar putea fi utilizate pentru a asigura integritatea traficului, în timp ce firewall-urile ar putea filtra traficul extern. Sistemele de detectare a intruziunilor ar putea oferi informații despre activități neobișnuite și chiar le pot preveni, certificatele pot fi utilizate pentru a identifica vehiculele și alte obiecte, iar multe alte opțiuni ar putea fi considerate mecanisme de protecție viabile. De asemenea, deși doar

două lucrări de referință au fost menționate pe cifra atașată, numărul atacurilor asupra vehiculelor a crescut de șapte ori din 2016.

2.1 Analiza literaturii de standardizare

Metodele generale și stabilite sunt aplicate unui număr considerabil de cazuri de utilizare și aplicații, metode precum cadrul dezvoltat de proiectul EVITA [128]. Lucrările de standardizare sunt, de asemenea, în desfășurare în mai multe subdomenii vehiculare din cadrul de securitate pentru serviciile V2X [129] sau ingineria securității cibernetice [130] sau ciclul de viață al dezvoltării software-ului sigur [131] și altele.

De exemplu, Administrația Națională pentru Siguranța Traficului pe Autostrăzi (NHTSA) a definit specificul reglementărilor privind securitatea traficului din cadrul Departamentului SUA pentru transport [132] în care au fost proiectate cele cinci niveluri de automatizare a conducerii. O altă societate din SUA care a devenit internațională acum este SAE International, standardele sale SAE J3061 [134], SAE J1939 [135], SAE J1709 [136] definesc cele mai bune practici din industrie în ceea ce privește securitatea. AUTomotive Open System Architecture (AUTOSAR) cu standardul său AUTOSAR WP-X-SEC [133] definește interoperabilitatea software-ului între ECU-uri într-un mod sigur și sigur.

Activitățile din Sistemul inteligent de transport (STI) sunt răspândite în jurul diferitelor grupuri care colaborează între ele, cum ar fi Comitetul European pentru Standardizare (CEN), care oferă CEN / TC 278 [145], Institutul European de Standarde în Telecomunicații (ETSI), care oferă o multitudine de standarde: TR 102 893 [137], EN 302 665 [138], TS 102 731 [139], TS 102 941 [140], TS 102 940 [141], TS 103 097 [142], TS 102 165 [173] care vor fi discutate mai în detaliu în rândurile următoare și Car2Car Communication Consortium (C2C-CC) care a colaborat îndeaproape cu ETSI și a furnizat ETSI TC ITS [143] și ISO / TC 204 [144]. Nu în ultimul rând, Institutul de ingineri electrici și electronici (IEEE) definește comunicarea între vehiculele în sine, dar și cu infrastructura în sine, cu ajutorul unor standarde precum IEEE 1609 [153] care reprezintă accesul wireless în mediul vehiculului (WAVE) și standardele IEEE AVB [155] pentru tehnologia rețelei Ethernet precum IEEE 802 11.p [154], IEEE 802.1AS [157], IEEE 802.1BA [126], IEEE 802.1Qav [158], IEEE 802.1Qat [159] și de desigur standardele tradiționale de tehnologie WLAN, cum ar fi IEEE 802.11 a, b, g, n.

Siguranța funcțională este acoperită de Comisia Electrotehnică Internațională (IEC) și de IEC 61508 [146] și de ISO / IEC 27034 [147] de ansamblu și concepte privind tehnicile de securitate, precum și de Organizația Internațională pentru Standardizare (ISO) și ISO 26262 [149] standard împreună cu o mână de altele: ISO / DIS 13400 [150], ISO 15118 [151], ISO 15765 [152] care definesc comunicarea în interiorul vehiculului excepție fiind ISO / TC 22 [148] care gestionează comunicația externă și zona de control în colaborare cu ETSI TC ITS.

O privire mai atentă asupra programelor Comisiei Europene pentru cercetare și inovare în domeniul ITS putem observa că majoritatea rezultatelor lor afectează ETSI și, de asemenea, au ajuns în cadrul ISO pentru standardizare. Din acest minim putem enumera proiecte precum:

- CARONTE (Crearea unei agende pentru cercetarea siguranței transporturilor) [160] care încearcă să clasifice cercetarea pentru securitatea transporturilor.
- HEAVENS (Vulnerabilități HEAling pentru îmbunătățirea securității și siguranței software-ului) [161] au furnizat arhitectura de bază pentru domeniul ITS și cazurile sale de

utilizare și cerințele de securitate derivate din acestea, aceste rezultate ajung în standarde precum SAE J3061, AUTOSAR WP_X_SEC și NHTSA.

- HoliSec (Abordare HOLIstic pentru îmbunătățirea securității datelor) [127] care a furnizat modele, metode, procese, instrumente și linii directoare pentru atingerea obiectivului „securității prin proiectare" oferind în același timp seminarii și ateliere pentru creșterea cunoștințelor și conștientizării cu privire la tema securității.
- SeFram [162] o continuare a SIGYN I și II a furnizat o analiză aprofundată a diagnosticării la distanță securizate, rezultând un protocol verificat formal cu impact în specificațiile ETSI ITS.
- SESAMO (SEcurity and Safety modeling) [163] a abordat problemele convergente ale securității și siguranței în sistemele încorporate cu impact ridicat în definițiile standard.
- EVITA (E-safety Vehicle Intrusion protected Applications) [164] soluții hardware axate pe protecția comunicațiilor intra-vehicul (V2X), au dus la dezvoltarea HSM, un coprocesor criptografic ECU, care este acum standardul în hardware proiecta.

Acest lucru ne face să ajungem la concluzia că, pentru domeniul auto, standardele și metodele stabilite furnizate de proiecte sunt în frunte, oferind în același timp informații despre lipsa de securitate a sistemelor auto.

2.2 Provocari tehnice si dinamica lor

Standardele domeniului vehiculelor se aplică pentru producătorii disponibili în mai multe țări cu prevederi legale diferite, care ar putea fi, de asemenea, conflictuale. Acesta este motivul pentru care avem nevoie de soluții generale de securitate care se pot adapta în funcție de context. De asemenea, deoarece în domeniul auto avem o mare varietate de tehnologii, protocoale și servicii cu timp, resurse și constrângeri de securitate diferite. Vom discuta în rândurile următoare complexitatea acestui sistem capabil să furnizeze procesare QM (Managementul calității), ASIL-B și ASIL-D (Automotive Safety Integrity Level), inclusiv redundanța multidimensională și provocările cu care se confruntă.

Revizuirea lucrărilor menționate mai sus care discută diverse secțiuni din domeniul auto ne-a oferit următoarea listă de provocări care trebuie luate în considerare de un model de amenințare:

- Încredere și confidențialitate: de exemplu, proprietarul vehiculului nu este de încredere de către producător, nici șoferul de către proprietar;
- Comunicare externă: DSRC (IEEE 802.11p), WLAN (IEEE 802.11a, b, g, n), comunicație celulară (GSM, GPRS, 3G, 4G, 5G), Bluetooth sau NFC;
- Securitate hardware și sistem de operare (OS): preferință pentru hardware ieftin „de la raft", multe sisteme de operare rulând direct pe microprocesoare;
- Autentificare, interoperabilitate și mobilitate: deoarece distanțele mari sunt parcurse în perioade scurte de timp, întreținerea și întreținerea ar trebui să rămână posibile;
- Rețeaua integrată în vehicul: un ECU de un euro cu diferite tipuri de rețea, cum ar fi rețeaua de controlor (CAN), rețeaua locală de interconectare (LIN), transportul sistemelor orientate către mass-media (MOST), FlexRay și Automotive Ethernet sau chiar rețelele de senzori fără fir (WSN) ;
- Interacțiunea dintre securitate și siguranță: un vehicul nesigur nu poate fi sigur, deoarece o vulnerabilitate poate afecta siguranța și invers;

- Cerințe în timp real: securitatea trebuie proiectată pentru a perturba la cald aplicațiile în timp real de la vehicul;
- Origini ECU: ECU-uri terțe care oferă atât soluții hardware, cât și software fără o arhitectură de securitate bine definită;
- Actualizare și întreținere software: un vehicul durează aproximativ 20 de ani, astfel încât mecanismele și soluțiile de securitate pe toată durata de viață ar trebui să dureze aceeași perioadă de timp, de asemenea compatibilitatea este importantă între diferiți furnizori;
- Cerințe legale: diferite țări ar putea avea cerințe și legislație specifice țării, totuși toate vehiculele ar trebui să poată participa la viitoarele sisteme de comunicații.

O metodologie aplicabilă care îndeplinește toate considerațiile de mai sus este tehnica de analiză și evaluare a riscurilor (TARA) [197].

2.3 Ciclul de viață al dezvoltării software securizate

În acest capitol este prezentată o analiză pentru domeniul Ciclului de viață al dezvoltării securității (SDL), cu accent pe adaptările sistemelor electrice și / sau electronice auto conform standardelor aplicabile domeniului, cum ar fi ISO 27034 [188], SAE J3061 [131] sau ISO / SAE DIS 21434 [130].

Inspirat de metricele și procesele menționate mai sus, am adaptat un proces similar, adaptat pentru sistemele specifice auto și compatibil cu Ghidurile SAE J3061 Cybersecurity Secure Software Development Cycle Life (SDLC) și ISO / SAE 21434: Vehicule rutiere - Conform inginerie cibernetică, un standard furnizarea de linii directoare pentru armonizarea proceselor de dezvoltare a securității cibernetice auto, bazate pe nivelurile ASIL, este introdusă de ISO 26262 [149]. Procesul constă în trei faze: concept, dezvoltarea produsului și producție și operare.

Fiecare proces SDL profită de revizuirea codului și de analiza statică pentru a detecta defectele de securitate. Efectuarea modelării amenințărilor și a evaluării riscurilor sunt obligatorii în timpul fazei conceptului, pe lângă testarea dinamică în timpul fazei de implementare a produsului. În continuare vom prezenta pe scurt câteva instrumente care pot fi utilizate în diferite etape ale SDL.

Analiza statică examinează codul sursă și îi testează punctele slabe fără a-l executa. Lucrarea de teză a lui Patrik Hellström [194] reprezintă o analiză atât a instrumentelor comerciale, cât și a instrumentelor de analiză statică open source și oferă o lucrare de ultimă generație pentru acest domeniu, inclusiv instrumente precum Fortify, Ounce, Coverity, Klockwork Insight și CodeSonar.

În ceea ce privește modelarea amenințărilor și evaluarea riscurilor conform modelului de securitate HEAVENS [161] și SAE J3061 [134], TARA [197] este preferată cu TMEA și ATA aplicate pentru analiza amenințărilor și o combinație și EVITA [128] și HEAVENS [161] pentru risc evaluare oferind în cele din urmă o mai bună înțelegere arhitecturală.

Testarea dinamică, inclusiv penetrarea și testarea fuzz, examinează codul sursă și îi testează punctele slabe în timp ce îl execută. Aici au fost evidențiate următoarele instrumente CERT Basic Fuzzing Framework (BFF), SDL MiniFuzz File Fuzzer și SDL Regex Fuzzer.

2.4 Prezentări în architectura de securitate

După cum sa menționat în secțiunile anterioare, pe baza standardelor și a specificațiilor domeniului auto, am identificat cele mai aplicabile mecanisme software pentru securitatea auto, unele dintre ele fiind deja validate în alte industrii.

Când discutăm despre sistemele de operare din industria auto, trebuie să avem în vedere faptul că discutăm mai multe opțiuni și chiar mai multe pachete pe același hardware. Ca parte a acestei lucrări, ne concentrăm doar pe Linux și Android, deoarece cele bazate pe AUTOSAR și RTOS au de obicei dimensiuni mai mici și unele dintre ele sunt chiar verificate formal pentru a îndeplini cerințele ASIL-D. Deci, pentru fiecare opțiune de sistem de operare disponibilă, trebuie să ne asigurăm că nucleul este protejat împotriva următoarelor clase de exploatare:

- Atacuri bazate pe stive
- Deborduri de tampon
- Programare orientată pe returnare
- Atacuri de integritate control-flux.

Pentru a vă proteja împotriva celor menționate mai sus exploatează următoarele mecanisme de întărire:

- Protectoare pentru stive
- Randomizarea aspectului de adresă-spațiu
- Asigurați-vă că nicio pagină de memorie nu trebuie să poată fi scrisă și executabilă în același timp

O privire mai atentă asupra tipurilor de memorie cu care interacționează aceste două sisteme de operare, pe baza proprietăților relevante pentru domeniul auto, cum ar fi confidențialitatea, integritatea și disponibilitatea (CIA), relevă următoarele:

- Universal Flash Storage (UFS): este opțiunea implicită pentru bootarea binarelor specifice hardware-ului pentru hardware-ul de referință prezentat anterior. Respectă integritatea și disponibilitatea ultimelor două proprietăți, în timp ce confidențialitatea trebuie investigată pe baza hardware-ului selectat. Suportul de încărcare sigur trebuie activat.
- NOR Flash: poate fi utilizat de bootloader-ul din prima etapă pentru a minimiza timpul de boot din cauza întârzierii timpului de pornire UFS. Similar cu UFS, îndeplinește integritatea și disponibilitatea, în timp ce confidențialitatea trebuie investigată pe baza hardware-ului selectat. Asistența de încărcare sigură trebuie activată atunci când este utilizată.
- LPDDR4 / 5: utilizat în principal pentru partajarea informațiilor între VM-uri sau cu AUTOSAR. În cazul în care datele partajate sunt temporare, CIA nu este necesară, dar dacă datele sunt procesate, atunci ar putea fi necesară CIA. Datele criptografice și suportul pentru memoria HSM ar putea fi necesare pentru a asigura confidențialitatea.

Alte tipuri de memorie, cum ar fi cardul SD și USB, nu sunt acceptate și, prin urmare, nu se recomandă utilizarea acestora.

O analiză a mecanismelor de control al accesului relevă faptul că SELinux este mecanismul preferat utilizat de Android, în timp ce pentru Linux nu există încă un mecanism definit, deși grupurile de lucru Automotive Grade Linux (AGL) încearcă să standardizeze SMACK, dar utilizarea SELinux pentru ambele ar fi reciproc benefică luând în considerare cunoștințele Linux

în afara domeniului încorporat. O scurtă prezentare generală a celor două mecanisme relevă următoarele:

- SMACK (Kernel de control acces obligatoriu simplificat)
  - Guvernată de o politică central
  - Proiectat având în vedere sistemele încorporate și pentru securizarea dispozitivelor conectate la Internet.
  - Configurare și întreținere simple
- SELinux
  - Complex datorită granularității ridicate.
  - Fără acces implicit și reguli necesare.
  - Regulile trebuie create și încărcate în sistemul de operare pentru a specifica drepturile de acces permise

Alte măsuri de securitate ar putea implica o arhitectură sigură orientată spre servicii cu suport firewall, filtrare de pachete și conturarea traficului. Izolarea domeniului ar trebui folosită, de asemenea, pe baza spațiilor de adrese, a mașinilor virtuale și a hardware-ului, alături de clasificarea izolării între diferite caracteristici ale aplicațiilor, toate păzite de un sistem de detectare a intruziunilor.

Această lucrare prezintă amenințări la adresa securității și implicațiile acestora, cu accent pe importanța securității în domeniul vehiculelor și modul în care au fost abordate diferite aspecte ca parte a unui număr de proiecte de cercetare. A fost proiectată o arhitectură de referință și s-au făcut propuneri și analize cu privire la mecanismele de securizare a acesteia. Am discutat despre soluțiile tradiționale care s-au dovedit eficiente în alte domenii de aplicare și chiar au fost prezentate noi soluții specifice.

Scopul acestei lucrări a fost de a prezenta o sinteză a domeniului cu speranța că va fi oferită o mai bună înțelegere a domeniului vehiculului, iar alții vor fi inspirați pentru a continua cercetarea, deoarece există multe de investigat în ceea ce privește securitatea internă a unui vehicul, făcând ca asigurarea de mașini și trafic mai sigure și mai sigure să fie obiectivul suprem.

Chapter 3. Siguranța vehiculului

În acest capitol voi prezenta primii pași ai acestei teze în care vom arunca o privire asupra vehiculelor actuale, o rețea mare de unități electronice de calcul (ECU) conectate nu numai între ele, ci și cu lumea exterioară. Interacționăm cu tehnologiile auto, le prezentăm vulnerabilitățile și sporim securitatea și siguranța oferind informații relevante printr-o interfață definită de utilizator. În același timp, documentăm procesele, infrastructura, tehnologiile și componentele hardware pe care le-am folosit în acest proces. Facem acest exercițiu pentru a înțelege mai bine cum arată suprafața de securitate și de atac a unui vehicul.

3.1 Contextul siguranței vehiculului

Aici ne bazăm munca pe prezentarea generală a contextului actual de piratare a vehiculelor și ne propunem să oferim explicații de ce este relevant acest proiect, cum este soluția semnificativă, care este abordarea noastră față de problemă și cazurile de utilizare pe care le avem în vedere.

Conceptul de hacking auto a devenit o componentă importantă a unui vehicul, chiar și FBI a emis avertismente legate de implicarea riscului de hacking auto. Vehiculul modern include peste 100 de dispozitive electronice conectate prin mai multe rețele specifice industriei, conform unui raport IEEE Spectrum [1] din 2009. Acest lucru este similar sau chiar mai puternic decât un computer personal și fiecare componentă poate fi accesată prin stabilirea unei legături cu rețelele disponibile, comunicarea cu motorul, afișajul unității centrale și altele. După cum sa documentat în [2] „industria auto a produs unele vehicule uimitoare, dar a lansat puține informații despre ceea ce le face să funcționeze. Înțelegerea modului în care vehiculul comunică vă va ajuta să diagnosticați și să depanați problemele auto. Pe măsură ce vehiculele au evoluat, acestea au devenit mai puțin mecanice și mai electronice. Din păcate, aceste sisteme sunt de obicei închise mecanicii. În timp ce reprezentanții au acces la mai multe informații decât puteți obține în mod obișnuit, producătorii auto în sine externalizează piese și necesită instrumente proprii pentru diagnosticarea problemelor. ”

Industria auto din Uniunea Europeană și din Statele Unite a ajuns la aceeași concluzie, aceea de a urma aceleași protocoale și standarde, indiferent de implementare. Acest lucru a permis utilizarea suportului generic și a colaborării, astfel încât dezvoltarea ulterioară a avut loc pe funcționalitatea de bază a unui vehicul.

3.2 Technologiile disponibile

Pentru a începe implementarea acestui proiect, primul pas necesar a fost cel care a necesitat înțelegerea protocoalelor și interfețelor specifice vehiculului. Primul pe care îl vom prezenta aici este portul OBD, o interfață care a devenit obligatorie pentru toate vehiculele disponibile. A fost introdus mai întâi în SUA în 1996, apoi în 2003 în Europa și până în 2008 în China și este acum alegerea principală pentru comunicarea cu vehiculul.

Într-un vehicul, portul OBD efectuează autodiagnosticare pentru a furniza informațiile solicitate de atelierele de reparații pentru identificarea problemelor la mașină. Acest timp de informații este disponibil prin coduri de diagnosticare a erorilor (DTC), mesaje care permit verificarea rapidă a defecțiunilor și identificarea de la ECU-uri din jurul mașinii.

Celălalt element important disponibil într-un automobil este protocolul serial CAN, dezvoltat de Bosch și utilizat pe scară largă în industria auto. Priza OBD-II oferă suport pentru acest protocol pentru a oferi diagnostic. Este descris în ISO 11898 [12] și ISO 15765 [2] și a devenit standardul de facto pentru vehicule în 2001 pentru piața europeană și în 2008 pentru cel american.

Ca structură internă avem o serie de sisteme încorporate diferite care comunică prin CAN. Implementarea lor este diferită și pot exista mai multe rețele și tipuri de comunicații care includ viteze diferite de magistrală de la viteză mică la viteză mare.

Ca parte a suportului hardware, am folosit nu doar o placă Raspberry Pi, ci și câteva componente suplimentare, cum ar fi un chipset GSM numit SIM5218 un microcontroler de interfață OBD numit ELM327 și un dispozitiv USB2CAN. Deoarece există atât de multe componente care compun stiva software, mai multe detalii sunt disponibile în hârtie, împreună cu o diagramă pentru

o trecere mai rapidă. O scurtă prezentare a stivei de software este vizibilă și în următoarele secțiuni ale capitolului.

3.3 Architectura proiectului

Aici descriem caracteristicile disponibile în prezent ale proiectelor, cele care sunt implementate până acum și prezentăm modalitățile în care acestea sunt puse împreună pentru a lucra pentru a oferi rezultatele documentate. Din perspectiva evoluției proiectului, putem descrie patru etape:

1. Implementarea inițială: include configurarea sistemului de operare Raspberry Pi (OS), picătura Digital Ocean și implementarea Ansible

2. Integrare de bază: include OBD / CAN, SMS MT / MO, GSM 3G AT, S-GPS / A-GPS, Wi-fi și suport Bluetooth adăugat.

3. Dezvoltare: unde am făcut API-ul Python, baza de date și implementarea sistemului de autentificare

4. Testarea end-to-end: cu client Android, interfață web pentru utilizator și suport de testare în viața reală adăugat deasupra.

Această etapă a început cu selectarea setării. Pentru placa de dezvoltare Raspberry Pi, am folosit sistemul de operare Raspbian, cel acceptat oficial de Raspberry Pi Foundation. În interior am instalat suportul Ansible ca manager de configurație pentru toate software-urile și serviciile instalate și configurate în imaginea sistemului de operare care se află pe placa SD a plăcii de dezvoltare.

Cu software-ul disponibil și configurat în consecință, configurarea hardware poate fi conectată în interiorul vehiculului utilizând dispozitivul ELM327 prin portul de diagnosticare de la bord. Vom folosi această configurație împreună cu codul sursă de dezvoltare pentru a valida cazurile noastre de utilizare.

Odată cu configurarea hardware completă și la locul său, utilizatorul se poate conecta la aplicația smartphone sau la interfața serverului web și se poate înregistra pentru un cont. Cu contul disponibil, asocierea este necesară pentru a asocia atât dispozitivul, cât și contul și după aceasta se efectuează câteva teste inițiale pentru a valida comunicarea full duplex între cele două.

Toți acești pași sunt esențiali nu numai pentru validarea și testarea cazurilor de utilizare, ci și pentru implementarea etapelor următoare ale soluției. Deși nu toate sunt disponibile din primele interacțiuni ale proiectului, acestea sunt obligatorii pentru configurarea soluției finale.

Pentru a îndeplini cerințele de securitate ale proiectului, soluțiile utilizate aici au fost atent selectate și configurate pentru a se întâlni cu cele mai bune practici din domeniul securității asociat industriei.

Toate comunicațiile fără fir sunt protejate cu WPA2 folosind chei de 256 de biți și autentificarea realizată pe baza unei combinații de nume de utilizator și parolă, cu o cerință minimă pentru lungimea parolei de opt caractere. Aceasta este de fapt cea mai de bază formă de securitate, deoarece este destul de generală și puternic validată.

Mergând mai departe abordăm următorul subiect în care securitatea este importantă, și anume partea SMS, în care comenzile sunt trimise către vehicul.

O altă piesă importantă a puzzle-ului care trebuie rezolvat este reprezentată de securitatea API-ului RESTful, unde, pe lângă securitate, trebuie să fie susținută și consistența și integritatea

datelor. Acest lucru este asigurat cu caracteristici precum: autentificarea utilizatorului, autorizarea utilizatorului, criptarea, controlul accesului pe rol, validarea datelor, integritatea și coerența datelor.

În plus, trebuie să asigurăm securitatea serverului care conține suportul pentru API-ul RESTful și baza de date. Trebuie protejat cu mecanisme de autentificare prin funcționalitatea HTTPS, care oferă criptare cu ajutorul certificatelor SSL. De asemenea, avem suport pentru un firewall acolo cu politică DROP implicită pe toate pachetele care nu corespund regulilor definite. Am definit acolo, excepții pentru protocolul SSH (22 / TCP), API-ul RESTful și interfața utilizatorului web expuse prin HTTPS (443 / TCP), de asemenea, pe Raspberry Pi toate porturile de intrare sunt închise pentru măsurători de siguranță.

Nu în ultimul rând, configurarea vehiculului, aici nu vorbim despre securitate, ci mai mult despre siguranța mașinii, pe care o asigurăm cu ajutorul redundanței atât pentru comunicația Wi-fi, cât și pentru modulul GSM sau GPS.

Chapter 4. Educație deschisă IoT

În acest capitol voi discuta despre modul în care putem elimina decalajul de cunoștințe pentru părțile interesate din domeniul automobilelor prin utilizarea tehnologiei și voi descrie modalități de a o face accesibilă și a impactului său responsabil. Voi prezenta un set de instrumente care ar putea fi utilizate atât de indivizi tehnici, cât și non-tehnici, deoarece schimbarea paradigmei către securitate prin proiectare trebuie înțeleasă la toate nivelurile.

Deoarece există o lipsă de conștientizare cu privire la impactul tehnologiei asupra autovehiculelor, propunem mai multe metodologii pentru a contribui la sensibilizarea tuturor părților interesate, precum instituțiile de reglementare, publicul larg și inginerii implicați în industria auto. Pentru aceasta, primul pas este prezentarea unui set de tehnologii și modul în care acestea ar putea fi aplicate pentru a reduce decalajul de cunoștințe.

Într-o manieră similară, folosim aceste tehnologii pentru a dezvolta în continuare arhitectura digitală a autovehiculelor pe care am început-o în capitolul anterior și nu numai că oferim informații cu privire la starea vehiculului într-un context sigur și sigur, ci și oferim mecanisme pentru mai multe părți interesate pentru a diagnostica și depana diverse probleme. aspecte ale mașinii, păstrând în același timp legătura cu trecerea de la mecanică la electronică și de la închisă la deschisă și externalizată.

4.1 Wyliodrin facilitator de technologie

În zilele noastre, industria auto conține o mulțime de sisteme electrice și electronice care rulează software și sunt conectate la Internet [201]. Această schimbare de tendință înseamnă că sunt necesare cunoștințe de calculator pentru interacțiunea cu vehiculul. Acest lucru este aplicabil pentru un număr mare de categorii de părți interesate care nu dispun de abilități tehnice. Schimbarea tendinței de la abilitățile de software, hardware și tehnologie, fiind o activitate de dezvoltator către publicul larg, este vizibilă și în alte industrii, de la artă la inginerie mecanică și chimică. Chiar și universitățile non-tehnice au introdus cursuri de informatică unde învață cum să interacționeze și să construiască un sistem de infotainment sau dispozitive similare. Astfel de exemple sunt arhitectura, ingineria civilă și chiar universitățile de film [202] [203], unde încurajează studenții să

includă tehnologii de ultimă generație în activitățile de zi cu zi. Faptul că au implementat acest lucru într-un mod de succes deschide calea și pentru automobile.

Atunci când interacționați cu software, hardware și concepte tehnologice în general, nu este o muncă ușoară din cauza mai multor motive, cum ar fi: concepte greu de înțeles, o curbă de învățare în general și o infrastructură uriașă cu care să interacționați, conținând mai multe ECU-uri și senzori disponibili. În mare parte, există dificultăți în aplicarea conceptelor de tehnologie în activitățile lor de zi cu zi. Wyliodrin poate ajuta cu aceste probleme ghidând părțile interesate într-o manieră practică pentru a construi sistemele de care au nevoie pentru a le ajuta să se relaționeze mai bine cu tehnologia.

Wyliodrin este o platformă care permite părților interesate să dezvolte și să interacționeze cu sisteme care sunt legate de domeniul lor de lucru. Cu ajutorul acestei platforme, părțile interesate pot vedea cum este proiectat un subsistem din vehicul, pot identifica abilitățile de programare necesare pentru a le construi și cum pot interacționa cu acesta. Placa oferă suport atât pentru software cât și pentru hardware și oferă un bun candidat pentru cerințele unei arhitecturi de proiectare sigure, care este atât sigură, cât și ușor de interacționat de către toate părțile interesate disponibile în industrie.

Atunci când predăm cunoștințe și programare în computer, abordarea variază în funcție de vârsta și antecedentele părților interesate. În timp ce unii au capacitatea de a înțelege și de a aplica concepte abstracte în exemple practice, alții trebuie să fie implicați în timpul procesului. Drept urmare, oferim o gamă largă de materiale și funcționalități de platformă pentru a alege de la proiectare.

Scratch este un limbaj de programare bazat pe blocuri vizuale, utilizat pe scară largă pentru predarea programării pentru adolescenți și copii. A fost dezvoltat în cadrul MIT Media Lab și de atunci a fost folosit în mare măsură pentru a construi aplicații prin glisarea și plasarea blocurilor pe un tablou de bord, ca într-un puzzle, evitând greșelile de configurare. Toate aceste blocuri sunt convertite în limbaje de programare standard. Poate fi integrat pe platforme terță parte sau din programul ID Scratch Studio, o platformă web.

O altă platformă similară este Blockly, dezvoltată de Google, care este la fel de bine convertită în limbaje de programare standard precum Javascript, Python, Dart sau cod PHP. Avantajul acestui limbaj este faptul că permite generarea de blocuri noi și integrarea lor înapoi în platformă [204].

Arduino este o placă de microcontroler dezvoltată în principal în jurul microcontrolerului ATmega328 care vine, de asemenea, împreună cu propriul IDE. Este cea mai utilizată platformă hardware și software capabilă să integreze mai mulți senzori și actuatori sub cei 14 pini digitali de intrare / ieșire din care 6 suportă ieșiri PWM și 6 intrări analogice. Placa este, de asemenea, echipată cu un antet USB și ICSP pentru programare, o mufă de alimentare, un buton de resetare și un oscilator de cristal de 16 MHz. Este utilizat pe scară largă de pasionați și începători din întreaga lume pentru învățarea prototipurilor dn nu numai datorită plăcii de dezvoltare rezistente la manipulare și ieftine și mediului de dezvoltare ușor de utilizat, ci și datorită multitudinii de biblioteci disponibile care expune funcții pentru controlul pinilor și perifericelor conectate . Aceste avantaje au făcut din Arduino un candidat bun pentru mediul universitar și, în special, pentru arhitectura digitală auto construită cu ajutorul Wyliodrin.

Nu în ultimul rând, am folosit Raspberry Pi, un mic computer cu o singură placă, dezvoltat în scopuri educaționale și care alimentează majoritatea proiectelor de comunitate electronică do it yourself. Fundația Raspberry Pi, fundația care guvernează dezvoltarea consiliului de dezvoltare

Raspberry Pi își propune să promoveze educația în informatică în regiunile nedezvoltate cu ajutorul unei platforme accesibile și ușor de utilizat.

Odată cu anotimpurile învățate din lumea Arduino, Rasperry Pi vine, de asemenea, echipat cu un antet mare GPIO cu 26, respectiv 40 pini, o bază de date mare de demonstranți, făcându-l popular în sectorul educațional căruia Arduino i s-a adresat [205] [206].

Placile de extensie Grove reprezintă un kit care reprezintă o componentă complementară care ar putea interacționa cu plăcile Arduino și Rasperry Pi și asigură interacțiunea cu componentele electrice compatibile Grove; senzori și actuatori care ar putea fi conectați direct la pinii disponibili Grove, astfel încât să poată apărea conexiuni și erori incorecte, cum ar fi un scurtcircuit [207].

Arhitectura și cadrul descris mai sus au fost concepute având în vedere scopuri educaționale și au fost optimizate pentru implicarea maximă și cel mai bun progres general al grupului, luând în considerare și complexitatea aplicațiilor dezvoltate și clasificarea obținută de părțile interesate pentru îndeplinirea sarcinilor. Rezultatele obținute după interacțiunea cu cadrul descris mai sus sunt comparate cu rezultatele obținute anul trecut de aceleași părți interesate într-un context similar.

Părțile interesate sunt studenți la inginerie mecanică cu un background auto și au făcut aceleași exerciții folosind demonstratorul Wyliodrin ca și anul trecut. Diferența dintre utilizarea configurării demonstratorului Wyliodrin și interacțiunea directă a anului anterior cu senzorii în mod direct a fost vizibilă, avantajul utilizării Wyliodrin fiind faptul că abstractizează și simplifică interacțiunea [209] [210]. Comparând cele două scenarii: anul trecut proiectele nu au fost finalizate, concentrarea fiind pierdută pe diferite aspecte, deoarece cu soluția furnizată toată lumea a reușit să finalizeze aceleași proiecte și a demonstrat rate de angajament mai mari.

Faptul că cadrul a permis, de asemenea, trecerea de la un limbaj de programare vizual, cum ar fi Blockly la unul Python, a fost foarte util și le-a permis să învețe un limbaj de programare de nivel înalt. De exemplu, am început interacțiunea doar cu casele de utilizare Blockly care generează cod Python din aceasta. Următorul pas a fost generarea unui cod Python defect și furnizarea mijloacelor pentru a le rezolva. Până la sfârșitul interacțiunii, 80% dintre părțile interesate au reușit să scrie aplicații Python complexe care pot interacționa cu mai mulți senzori. La momentul examenului, 70% dintre ei erau capabili să controleze și să schimbe date între perifericele conectate în timp ce interacționau cu o interfață web, care a fost un salt important față de experiența din anul precedent.

Cu acest proiect am reușit să proiectăm o platformă care a permis părților interesate să dobândească abilități inginerești și să le aplice fără efort în activitățile lor. Acest tip de abordare pentru industria auto s-a dovedit a fi benefic și în tranziția de la o paradigmă precum securitatea prin obscuritate către securitatea prin proiectare ar face din Wyliodrin un bun exemplu al modului de simplificare și posibilitate a unei conversii mai ușoare.

4.2 Yocto Project aplicat în ecosistemul industriei

Yocto Project a fost lansat inițial în martie 2011 și a devenit rapid standardul pentru furnizarea de distribuții și binare Linux personalizate și previzibile. A fost utilizată ca parte a acestei teze în același scop, deoarece aveam nevoie de predictibilitate și de o distribuție Linux care să corespundă nevoilor noastre, care să conțină doar cerințele minime pentru job.

Scopul nostru a fost de a obține un sistem ușor de personalizat, în care să putem include funcționalități pentru caracteristicile și senzorii pe care i-am putea aduce în partea superioară a

sistemului de bază, portând funcționalitatea pe o platformă hardware complet nouă și practic orice funcționalitate la care ne-am putea gândi. Pentru o mai bună înțelegere a modului în care a fost folosit proiectul Yocto ca parte a acestei lucrări, voi încerca să prezint un pic arhitectura și suportul disponibil în secțiunile viitoare.

Fluxul de lucru precizează pe scurt faptul că dezvoltatorii trebuie să precizeze arhitectura, patch-urile, politicile și detaliile de configurare înainte de a construi. Când începe procesul de construire, codul sursă este preluat și descărcat dintr-o locație specifică definită în metadate. Cu codul sursă disponibil local, motorul de construcție aplică patch-urile deasupra, configurează și compilează software-ul. Apoi îl instalează într-o locație temporară înainte de a-l împacheta în formatul preconfigurat recunoscut de managerul de pachete selectat. După aceasta se efectuează verificări QA și sănătate pentru a valida procesul de compilare și când totul arată bine, rezultatele sunt generate de fluxuri de pachete binare pentru a crea sistemul hardware și imaginile SDK.

Pe lângă toate cele prezentate în subcapitolul anterior, industria a început să adauge suport de metadate pentru a-și configura propria versiune a produsului în mod similar cu modul în care lucrurile se întâmplă în nucleul Linux cu module de nucleu, diferența fiind că suportul este inclus în straturi cu funcționalități diferite, fie acestea straturi BSP cu suport pentru diverse platforme hardware sau straturi care includ diverse funcționalități, cum ar fi în timp real, virtualizare și suport pentru securitate.

Pe lângă straturile prezentate mai sus, există mult mai multe straturi disponibile, cu ajutorul cărora dezvoltatorii ar putea folosi. Unele dintre ele sub umbrela Proiectului Yocto, alte întreținute de companii private sau persoane fizice. Pentru un nou venit în acest proiect, toate acestea s-ar putea dovedi foarte multe, cu atât mai mult cu cât interacțiunea cu codul sursă al kernel-ului Linux, dar Fundația Linux și întreținătorii proiectului se străduiesc să ajute în acest sens.

Pe lângă munca de furnizare a funcționalităților și suportului ca parte a straturilor, atât industria, cât și comunitatea open source au văzut valoarea proiectului și au început să ofere suport pentru distribuțiile de referință. Una dintre primele a fost OpenStack menționată mai sus, dar există, de asemenea, o mulțime de alternative care oferă nu numai modalități de interacțiune cu distribuția Linux, ci și noi instrumente de suport și opțiuni de configurare pentru funcționalitatea de bază.

Unele dintre aceste distribuții sunt Carrier Grade Linux (CGL), o distribuție Linux care a început ca o documentație de referință pentru industria de telecomunicații, o linie directoare pentru funcționalitățile pe care ar trebui să le aibă un sistem de operare de tip carrier, bazat pe Linux, acum este o combinație de straturi de proiect Yocto care ar putea fi folosit ca punct fix și ușurează mult procesul pentru o companie.

O altă inițiativă similară, pornită într-un fel de la inițiativa CGL, deoarece a fost inițiată de același manager al Linux Foundation, este Automotive Grade Linux (AGL), care oferă o distribuție Linux de referință care a ajuns chiar și în mașini precum Toyota și se află în dezvoltare continuă de către toți principalii jucători din industria auto.

Acesta a fost suportul pe care am început să-l folosesc, deoarece a sprijinit și placa Raspberry Pi când am luat o decizie cu privire la un sistem de operare care ar putea fi configurat în funcție de cerințele mele fluctuante. S-a dovedit a fi cea mai bună decizie, deoarece rezultatul final a fost nu numai ușor de configurat, ci și rezultatul final a folosit mult mai puține resurse, care au fost utilizate cu adăugarea de noi funcționalități pentru asigurarea unui adevărat sistem deschis, configurabil, o simulare a unui proiectați arhitectura digitală auto care a fost folosită ca banc de testare pentru această teză.

Chapter 5. Prototiparea industrială

În acest capitol voi continua discuția despre reducerea decalajului de cunoștințe pentru părțile interesate din domeniul automobilelor prin utilizarea tehnologiei și voi prezenta seturi de prototipuri ca modalități de a-l face accesibil și a impactului său responsabil. Rolurile prototipurilor sunt gestionate atât de indivizi tehnici, cât și non-tehnici, așa cum sa subliniat deja în capitolul anterior.

După cum sa menționat deja, există o lipsă de conștientizare cu privire la impactul tehnologiei asupra autovehiculelor și am furnizat instrumente și prototipuri pentru a face față acestei formări de lacune de cunoștințe și pentru a contribui la sensibilizarea tuturor părților interesate, cum ar fi: instituțiile de reglementare, inginerii și generalul public implicat în industria auto. Ca al doilea pas al acestui proces este prezentarea unui set de prototipuri care combină noile tehnologii și le aplică pentru a obține rezultate aceleași sau chiar mai bune în scenarii industriale. O privire rapidă asupra acestor prototipuri și putem concluziona cu ușurință că aplicația tehnologică prezentată ar putea fi aplicată și în industria auto, pentru a reduce decalajul de cunoștințe.

5.1 Serviciu de recunoaștere vocală

Munca pentru serviciul de recunoaștere a vorbirii, activată în contextul auto, a început de la cercetările efectuate în cadrul lucrării în care am proiectat și dezvoltat o mănușă robot inteligentă Haptic (IHRG) adecvată pentru reabilitarea pacienților cărora li s-a diagnosticat un accident cerebrovascular. Proiectarea și implementarea soluției IHRG au fost efectuate în laboratorul nostru. Acest lucru este potrivit având în vedere că pentru pacienți se vor simți mai confortabil dacă își fac exercițiile medicale acasă. IHRG propus are un mare avantaj - posibilitatea de a specifica comenzi vocale, ajutând pacientul să facă un număr extins de exerciții medicale.

În articol ne-am propus să restabilim funcțiile motrice pierdute după un traumatism major ca urmare a accidentelor vasculare cerebrale și a bolilor cerebrovasculare (AVC) în care, ca urmare, o parte a corpului controlată de zona afectată a creierului nu poate funcționa corect [213] . Am extins acea implementare pentru funcționalitatea de recunoaștere a vorbirii vehiculului auto, care, în schimb, este capabilă să ofere interacțiune cu diverse ECU-uri de la vehicul, cum ar fi sistemul de infotainment, lumini sau altele.

În ultimul timp, literatura disponibilă a prezentat numeroase tehnici și concepte care permit evaluarea domeniului cu implementări diferite, deși articolul inițial început ca o aplicație pentru sectorul medical nu este o cerință. Apoi numeroase studii [214] [215] au permis reproducerea cinematicii mâinii umane cât mai mult posibil cu ajutorul dezvoltării structurilor cinematice. Acum, același nucleu poate fi aplicat într-un mod similar cu o funcționalitate principală de sprijin pentru șofer, care este necesară pentru a ține mâinile pe volan și ochii la drum.

Proiectarea soluției este realizată pe baza aceleiași platforme hardware Raspberry Pi cu ajutorul extensiilor Arduino Mega 2560 care simulează un controler ECU care determină activitatea motoarelor mici. Pentru a face calculul specific de recunoaștere vocală, a fost utilizată platforma hardware Raspberry Pi. Folosind un microfon, șoferul poate trimite comenzi vocale

către echipament, care vor deduce acțiunile declanșate pentru o anumită perioadă de timp [216] [217].

Au fost efectuate teste ample și rezultatele au oferit o rată de detectare foarte bună, de la 200 de teste, 97,5% au returnat cu exactitate rezultatul corect pentru comenzile simple dintr-un singur cuvânt. Cu toate acestea, a fost observată o rată bună de detecție de 95% pentru constructele mai complexe. Unul dintre cei mai importanți factori care influențează procesul de recunoaștere este zgomotul. Ori de câte ori zgomotul este menținut la un nivel minim, aproape de zero, rezultatele recunoașterii vorbirii sunt la o calitate bună; indiferent dacă nivelul de zgomot crește, nivelul de precizie pentru recunoașterea vorbirii scade aproape exponențial. Un alt factor important pentru acuratețea detecției este legat de utilizarea platformei hardware Raspberry Pi și de suportul său de calitate audio, care se dovedește a avea o rată de detecție mai mică decât atunci când testele au fost efectuate pe un computer personal.

Pentru a îmbunătăți precizia rezultatelor, putem încerca să modificăm valorile înregistrate ca intrare sau modificând scorul Viterbi. O altă modalitate de îmbunătățire a preciziei este prin faza de antrenament a modelului acustic. Odată cu creșterea colectării datelor, se atinge un punct de echilibru în care precizia rămâne neschimbată, acesta fiind considerat punctul maxim.

Pentru a oferi un model acustic utilizabil de mai mulți șoferi, va fi obligatoriu ca fiecare șofer să contribuie cu propria intrare în sistem. Va fi interesant în acel moment să comparăm soluția noastră cu o soluție similară deja disponibilă în Automotive Grade Linux (AGL) numită Alexa, care este furnizată de Amazon și este, de asemenea, integrată în mediul Yocto.

O detectare foarte bună a fost obținută atât pentru comenzi simple, cât și pentru comenzi complexe. Au fost analizate și alte aspecte diferite pentru a identifica influența lor asupra procesului de recunoaștere.

5.2 Companion Android pentru Wyliodrin

Piața sistemelor încorporate este în continuă creștere. Sunt prezenți peste tot în jurul nostru în diferite dispozitive, de la dispozitive obișnuite la automobile sau sisteme inteligente de încălzire. Lucrarea actuală descrie procesul de dezvoltare a unei aplicații Android care permite utilizatorului să interacționeze cu sistemul încorporat prin Internet, utilizând un telefon mobil. În acest fel, utilizatorii pot utiliza valorile citite de senzorii telefonului în proiectele lor și îi pot controla de la distanță în timp real.

Internetul obiectelor (IoT) este un concept care își propune să se conecteze la Internet cât mai multe dispozitive posibil și să permită controlul acestora de la distanță, utilizând aplicații speciale. Creșterea numărului de sisteme încorporate va cauza o dificultate în modul în care acestea pot fi controlate dacă fiecare tip de dispozitiv va avea un mod diferit de interacțiune și control.

Dispozitive precum Raspberry Pi și Intel Galileo ajută la dezvoltarea IoT oferind oricui este interesat un mod simplu de a dezvolta sisteme încorporate care vor realiza funcțiile dorite. Conectarea acestor dispozitive la internet este partea ușoară, provocarea este, desigur, dezvoltarea de aplicații specifice pentru controlul de la distanță a fiecăruia dintre ele.

Wyliodrin este o platformă web care oferă utilizatorilor un mod simplu de a dezvolta aplicații software complexe care pot fi executate pe dispozitive încorporate fără a necesita cunoștințe avansate de programare. Aplicațiile create utilizând acest sistem rulează pe dispozitive în mod independent, fără interacțiunea utilizatorului [218] [219].

Deoarece smartphone-urile sunt atât de utilizate pe scară largă, s-a părut firesc să încerce să capteze atenția utilizatorului smartphone-ului către IoT și dezvoltarea sistemelor încorporate. Combinarea smartphone-ului dvs. cu Wyliodrin reprezintă punctul de plecare perfect pentru controlul de la distanță a sistemelor încorporate prin Internet. De asemenea, aplicația mobilă poate utiliza toate funcțiile furnizate de dispozitivele mobile, cum ar fi citirea valorilor senzorilor telefonului și trimiterea către sistemele încorporate.

Lucrarea actuală descrie procesul de dezvoltare a unei aplicații Android care comunică cu sistemele încorporate care execută aplicații create folosind Wyliodrin. Această aplicație mobilă va trimite acțiunile utilizatorului și datele de pe telefon, cum ar fi valorile senzorilor, către sistemele încorporate.

Aplicația obținută în timpul procesului descris este funcțională și poate fi utilizată pentru a controla dispozitivele încorporate și permite utilizatorului să interacționeze cu acestea. După cum sa descris anterior, primul ecran al aplicației este ecranul de conectare. În acest ecran, utilizatorul poate alege să scaneze un cod QR care îi va permite să se conecteze în sistem.

După ce autentificarea a reușit, utilizatorul este redirecționat către ecranul principal, unde poate vizualiza lista tablourilor de bord salvate și poate alege să creeze unul nou sau să șteargă unul apăsând lung pe acesta. Dacă utilizatorul alege un tablou de bord din listă sau apasă butonul Nou tablou de bord, aplicația îl va redirecționa către ecranul tabloului de bord

. Lista elementelor care pot fi adăugate la tabloul de bord poate fi văzută într-un meniu glisant, în partea stângă a ecranului. După alegerea unui element din listă, utilizatorul trebuie să aleagă proprietățile pentru noul element. Pentru a edita sau șterge un element, utilizatorul trebuie să îl apese lung și aplicația va afișa un meniu pop-up.

Tabloul de bord poate fi utilizat pentru a controla elemente precum un sistem de încălzire simplu. Tabloul de bord arată temperatura curentă în habitaclu folosind un termometru și istoricul valorilor primite folosind un grafic. Utilizatorul poate seta temperatura dorită în cameră folosind o bară de căutare și poate porni și opri sistemul folosind un buton de comutare.

Pe scurt, Wyliodrin COMPANION este o aplicație Android care permite interacțiunea cu sistemele încorporate de la distanță prin Internet; poate trimite date și comenzi către sisteme și poate primi informații care pot fi vizualizate într-unul din formularele disponibile.

Aplicația oferă o serie de elemente utilizate pentru trimiterea comenzilor (cum ar fi butoanele, butoanele de comutare), trimiterea datelor (cum ar fi senzorii telefonului, bara de căutare) și afișarea informațiilor primite (precum termometre, vitezometre, grafice). Aplicația utilizează protocolul XMPP într-un mod care asigură o comunicare rapidă și sigură între dispozitivele mobile și încorporate, așa cum se întâmplă în sistemul Wyliodrin [218].

Chapter 6. Architectura digitală securizată pentru automobile

În acest capitol am investigat mecanismele care ar putea fi implementate pe arhitectura digitală auto dezvoltată până acum, pentru a oferi un sistem de securitate pe mai multe niveluri, rezumând lucrările care au fost publicate în [109], [110], [111] și [121].

Am început această lucrare aruncând o privire asupra mediului de execuție de încredere, descris în subcapitolul 6.1 cu detalii. În mod ideal, am dori să rulăm fiecare aplicație într-un astfel de mediu și să garantăm securitatea execuției, dar, din păcate, există o serie de penalități de timp și performanță care nu sunt considerate acceptabile în toate contextele. Cu ajutorul mecanismului [111] sprobes reușim să facem o introspecție a registrelor sistemului și a paginilor de memorie prin apeluri TrustZone SMC pentru a preveni interacțiunile rău intenționate. Din păcate, rezultatele obținute de performanță și limitările soluției care au necesitat interogarea continuă a sistemului, împreună cu sancțiunile mediului de încredere, în general, s-au dovedit a nu fi suficient de calificate pentru scopurile noastre.

Pentru a contracara limitarea soluției anterioare pentru acele aplicații care nu pot fi executate într-o execuție de încredere, am ales o abordare sistemică, propunând o soluție de întărire a securității cu ajutorul [109] și [110] pe care le-am descris în subcapitolul 6.2 din teză. Mai întâi oferim un sistem de detectare a intruziunilor gazdă [110], care este o opțiune de analiză mai discretă bazată pe interogarea apelurilor de sistem declanșate de o aplicație împreună cu o soluție de analiză binară [109] care are o cheltuială redusă fiind numită o singură dată pentru un binar punându-l într-una dintre multiplele categorii disponibile. Doar una dintre aceste categorii este benignă, restul sunt rău intenționate pentru simplificarea procesului de analiză post în caz de vulnerabilități.

Acum, că avem un sistem sigur, dorim, de asemenea, să ne asigurăm că orice actualizare furnizată nu afectează negativ securitatea, așa că am adăugat și analiza statică a codului cu [121] unde oferim un mecanism de notare a impactului pe suprafața de atac a o actualizare folosită. Această parte a lucrării este descrisă în subcapitolul 6.3 cu detalii suplimentare privind implementarea, rezultatele și evaluarea soluției. Și acesta este un tip de analiză unică, cu costuri reduse la sistem și validare suplimentară a securității, cu adevărat util, deoarece actualizările over-the-air sunt un lucru obișnuit în industria auto.

6.1 Execuție de încredere

Vom merge mai departe și vom încerca să descriem unul câte unul fiecare articol de lucru și vom încerca să oferim pentru fiecare rezultatele și evaluarea corespunzătoare contextului nostru. Pentru a oferi actualizări sigure, trebuie să avem securitate hardware HSM conform prevederilor ISO 26262 [149]. În plus față de hardware, trebuie să ne construim suportul de execuție de încredere. Pentru a oferi o execuție de încredere, una dintre opțiunile oferite de industrie este utilizarea arhitecturii complexe bazate pe hipervizor în mai multe etape, similară cu cea descrisă în stadiul tehnicii, în care hipervizorul este utilizat pentru validarea interacțiunilor. Aceasta vine cu o cheltuială uriașă, deci alternativa este mecanismele de sondare care asigură o cheltuială mai mică. În următoarele rânduri ale acestui subcapitol vom investiga această nouă soluție alternativă la problema de execuție de încredere.

Pe sistemele de bază, sunt necesare mecanisme pentru a menține securitatea pe tot parcursul vieții vehiculului. Există problema escaladării privilegiilor pe un astfel de sistem și de aceea începem prezentarea cu soluția care folosește suportul de execuție de încredere numit TrustZone în arhitecturi ARM și ideea de a proteja sistemul de operare de rootkit-uri cu o cheltuială minimă.

Pentru aceasta am definit un număr de 5 reguli de la S1 la S5 care trebuie implementate și respectate pentru a proteja sistemul. Am folosit un mecanism de introspecție bazat pe apeluri SMC furnizate de TrustZone pentru a analiza într-o lume sigură instrucțiunile care afectează registrele și paginile de memorie, fără interacțiuni rău intenționate în procesul de analiză. Acest mecanism a fost validat într-un mediu sintetic de către un grup de cercetători și, deoarece rezultatele inițiale s-au dovedit a fi încurajatoare, am decis să-l implementăm pe un kernel Linux 4.9 și pe cel mai recent firmware arm-trust.

Rezultatele implementării mecanismului sprobes au arătat o cheltuială generală de aproximativ 10% pentru un sistem cu 279 de sarcini rulate și un număr de 5611 instrucțiuni suplimentare executate, totuși o astfel de implementare nu ar fi fost acceptată în nucleul Linux deoarece modifică componentele de bază care sunt extrem de sensibile la schimbări specifice precum a noastră și, din păcate, cheltuielile cu timpul și performanța, și anume camera de vizualizare din spate de 2 secunde a standardelor precum NHTSA (Administrația Națională pentru Siguranța Traficului pe Autostrăzi) este ridicată în comparație cu cerințele din domeniul auto.

Nu putem bloca modul de atac al atacatorului de a manipula nucleul din interior. Deși am putea limita posibilitățile sale de a avea acces la root. Această punere în aplicare cu apelurile SMC s-a dovedit a fi problematică, întrerupând unele funcționalități și nu adăugând stratul impresionant de protecție cu puține cheltuieli generale. Deși nucleul Linux nu este perfect și mulți îl consideră un megalitic și depreciat, cu siguranță nu este un sistem lipsit de apărare și se dovedește foarte greu să fie spart. Un mecanism de sondare sau întrerupere pentru protejarea modificărilor din registre, astfel încât manipularea să poată fi efectuată automat atunci când instrucțiunile sunt executate la cele mai mici niveluri, ar putea constitui îmbunătățiri suplimentare.

Deși mecanismul SPROBES sună ca o opțiune bună care ar putea foarte bine să facă parte din funcționalitățile kernel-ului Linux, cel mai probabil acest lucru nu ar fi posibil. Deoarece ar necesita o serie de modificări care s-ar dovedi, de asemenea, provocatoare pentru comunitatea Linux. Deși, teoretic, mă bucur de propunerea elegantă SPROBES, în practică s-ar putea dovedi că prezintă destul de multe probleme, făcând din soluția hipervizorului o abordare mai robustă și mai sigură.

Deși cu limitări care nu sunt acceptabile pentru automobile, SPROBES se dovedește a fi un bun punct de plecare pentru o soluție de securitate a nucleului Linux în timp real. Acest lucru a fost validat deja în produsele Samsung KNOX pentru Android. Acest tip de soluție se dovedește a fi util mai ales pentru dispozitivele încorporate bazate pe arhitectura ARM cu suport TrustZone. Orice alt mecanism de izolare hardware ar trebui să implementeze o soluție similară, în special pentru specificațiile hardware..

6.2 Îmbunătățirea securității

După ce am îmbunătățit securitatea la nivel de kernel și firmware cu [111], este timpul să trecem în arhitectură în spațiul aplicației sistemului de operare unde prezentăm lucrarea care a fost publicată în două articole de cercetare [109] și [110] care implică ambele inteligență artificială pentru a evalua securitatea unui sistem de operare. Una este responsabilă pentru analiza în timp de execuție a apelurilor de sistem și cealaltă a caracteristicilor statice care interacționează cu un binar în scopul identificării și clasificării incertitudinii și inconsecvenței în sistemele de operare. Ambele lucrări fac parte dintr-o abordare sistemică pentru securitate, care întărește apărarea cibernetică a sistemului nostru și detectează atacuri și exploatări.

În mare parte, principala tehnică de detectare a anomaliilor în interiorul unui sistem este reprezentată de „detectarea abuzului” în care o semnătură digitală a unui atac este trasă după ce a avut loc și inclusă ca o actualizare în interiorul dispozitivului de securitate. Deși este foarte eficientă împotriva amenințărilor cunoscute, această strategie nu este potrivită pentru a prinde un vector de atac de zi zero. Rezultate mai bune sunt oferite atunci când se construiește o linie de bază de „comportament normal” și se marchează orice activitate care nu se încadrează în linia de bază. Pentru un sistem de detectare bazat pe secvențe de apeluri de sistem, sistemul de detectare a intruziunilor captează orice secvență anormală și o marchează ca nelegitimă. Același lucru este valabil și pentru problema clasificării pentru binare bazate pe benigne și diferite clase malware ca vector de caracteristici numerice folosind apeluri API generate atunci când aplicația apelează o funcție disponibilă într-un alt modul.

Lucrarea prezentată aici se bazează pe un corp larg de literatură și încearcă să abordeze problema existenței anomaliilor în interiorul unui sistem de operare, fie sub forma unei intruziuni, fie a unei activități malware. Familia malware include viruși, troieni, truse de rădăcină, viermi, roboți, backdoors, adware, spyware, ransomware și reprezintă o amenințare în creștere rapidă pentru sistemele de operare disponibile. Reprezintă o piață de miliarde de dolari, care are o creștere exponențială a cantității de activități malware, raportate de companiile antivirus.

Deoarece soluția anterioară nu îndeplinea complet cerințele de timp și performanță cerute de industrie, am trecut la o soluție HIDS bazată pe apeluri de sistem, o soluție de securitate adaptivă care construiește o linie de bază pentru ceea ce înseamnă comportamentul normal / benign și semnalează orice abatere. Această soluție este construită pentru a păstra securitatea bazei de date care conține politici de securitate, deoarece este un punct major de vulnerabilitate.

Acest lucru s-a dovedit vulnerabil mai ales în 2019, când baza de date menținută de Honda le-a permis operatorilor să vadă ce sisteme din rețea erau vulnerabile la defectele de securitate, expunând datele sistemului de 134 de milioane de angajați. Un lucru similar s-a întâmplat cu baza de date configurată necorespunzător din India, care a permis divulgarea informațiilor personale și despre vehicul a 452 mii de utilizatori și lista continuă.

Folosim pentru implementare o instanță a bazei de date MySQL pe care o monitorizăm într-un container. Aici generăm eșantioane benigne utilizând mysqlslap și eșantion rău intenționat, care include modele rău intenționate, cum ar fi amprentarea serviciilor, injecția sql sau atacurile cu forță brută folosind sqlmap. Astfel reușim să captăm 4.732.254 apeluri de sistem dăunătoare și 35.351.981 apeluri de sistem benigne.

Aceste eșantioane sunt apoi analizate folosind metoda tf-idf (termen frecvență - frecvență inversă a documentului, un algoritm utilizat în NLP cu succes) pentru a construi un dicționar de apeluri de sistem cărora le atribuim ca frecvență de greutăți, la care se adaugă ID-ul firului, legat dinamic biblioteci, valoarea returnată sau durata apelului de sistem. Toate aceste proprietăți formează o matrice rară care reprezintă starea containerului într-un anumit moment.

Nucleul de decizie se ia pe baza algoritmului de pădure aleatorie, care sa dovedit a fi foarte eficient în astfel de probleme de clasificare. În analiză, viteza a fost o calitate importantă și am observat că este influențată de dimensiunea tamponului de evenimente (complexitatea urmelor). Cele mai bune rezultate au fost obținute pentru un buffer cu o dimensiune de 250 apeluri de sistem: 95% precizie și 0,4 ms timp de execuție.

Pe măsură ce dimensiunea bufferului crește, numărul eșantioanelor disponibile pentru modulul de învățare va scădea în consecință, afectând precizia generală. De exemplu, în timp ce se utilizează un tampon de dimensiune 250, rezultă un total de 160.336 de monștri rău

intenționați, plus benigni / normali. În timp ce T1 a avut o performanță mai bună, T2 a obținut o precizie comparabilă folosind mai puține eșantioane de învățare decât T1.

Următorul pas este analizarea binarelor pentru a identifica existența diferitelor tipuri de malware în sistemul de operare, deoarece un astfel de sistem (similar cu mediul Android) permite adăugarea de aplicații deasupra acestuia și acest lucru a dus la atacuri precum ransomware-ul WannaCry sau NotPetya (ambele bazate pe EternalBlue exploit la implementarea protocolului Server Message Block (SMB) care permite executarea unui cod arbitrar pe computerele țintă) care a afectat mii de servere ale companiilor de transport precum Deutsche Bahn, Maersk sau FedEx. Pentru aceasta am analizat eficiența clasificării algoritmilor aleatori de pădure și rețele neuronale. Inițial este necesar să construim un set de date reprezentativ, așa că construim o cutie de nisip Cuckoo pentru a surprinde comportamentul binarelor și a furniza rezultatele ca intrare în algoritmi unul câte unul, la final comparăm eficiența lor în analiză.

Alte atacuri ransomware specifice Linux: Tycoon (și pentru Windows, exploatează conexiuni nesecurizate de protocol desktop la distanță (RDP)), QNAPCrypt (exploatează conexiuni proxy SOCKS5) sau Erebus (exploatează probleme de securitate precum DirtyCOW - permite proceselor să scrie doar pentru a citi fișiere datorită exploatării unei condiții de rasă din kernel-ul Linux, funcționează din copy-on-write, de unde și numele - sau anumite versiuni ale Apache 1.3.36, PHP 5.1.4).

În acest caz, setul de date include 10.000 de monștri, dintre care 5022 sunt benigni, 4978 de programe malware din care 901 sunt generate sintetic de noi. În urma analizei, sunt extrase 7101 caracteristici care ar putea ajuta la identificarea aplicațiilor malware, caracteristici pe care am reușit să le reducem la 78 folosind atât o analiză automată, cât și una manuală. Printre caracteristici se numără un număr de biblioteci dinamice, adrese, dimensiuni sau funcții (dar și entropia fiecărei secțiuni a executabilului). Putem lista accesul la bibliotecile Binder (pe partea tranzacției, răspuns, eliberare), biblioteci Widget, rețea biblioteci sau interacțiuni cu biblioteci standard precum libc sau openssl (libssl).

În urma acestei reprezentări sub forma unui set de caracteristici extrase atât în analiza statică a binarului, cât și în timpul rulării, procesul de clasificare se realizează cu cei doi algoritmi menționați mai sus împărțind setul de date în 80% antrenament și 20% testare. Validarea încrucișată este utilizată pentru a evalua modelele alese și în cele din urmă algoritmul forestier aleatoriu este cel care prezintă cele mai bune rezultate (97% față de 96%). Ca o notă laterală: utilizarea numai a caracteristicilor statice, deși scade procentul de precizie, îl menține la o valoare care poate fi luată în considerare (95%).

Prezentul studiu demonstrează aplicabilitatea tehnicilor de învățare automată pentru dezvoltarea unei întăriri mai puternice a securității pe baza instrumentelor dezvoltate mai sus. Pentru ambele instrumente reușim să construim puncte de plecare bune cu seturile de date care s-ar putea dovedi utile pentru lucrările viitoare pe această temă.

Pentru sistemul de analiză a aplicației, am reușit să găsim o acuratețe bună cu algoritmul Random Forest, am construit un model bun care ar putea fi scriptat și, de asemenea, numărul de caracteristici extrase ar putea oferi o perspectivă mai bună despre software-ul rău intenționat pe o analiză mai profundă, chiar dacă se face manual.

Pentru sistemul HIDS am putea îmbunătăți rata de clasificare încercând să surprindem secvența apelurilor de sistem (folosind învățarea multiplă [71], tipărirea frecventă a modelelor [72] sau abordările seriilor temporale [73]) sau prin captarea argumentelor lor [74] , [75] și grupându-le. Intuitiv, o anumită secvență de apeluri de sistem ar putea îmbunătăți problema clasificării, dar aceasta necesită o analiză suplimentară.

6.3 Analiza statică a codului

Pentru a menține un mediu sigur, avem nevoie de un mecanism de evaluare a securității actualizărilor aplicate. Pentru aceasta, vă propunem o analiză a suprafeței de atac și un mecanism de notare a impactului la actualizare bazat pe aceasta. În rândurile următoare voi descrie această soluție pe care am protejat-o în articolul [121] ca exemplu.

Majoritatea vulnerabilităților sunt create prin apelarea greșită a unei funcții dintr-o bibliotecă sau apelarea unei funcții vulnerabile [199]. Acest proiect își propune să automatizeze descoperirea acestor tipuri de vulnerabilități prin utilizarea unei abordări de învățare profundă. Setul de date utilizat pentru abordarea acestei probleme este compus din diferite programe care conțin vulnerabilități. Folosim analizoare statice pe ele pentru a produce etichete pe care le folosim pentru a ne instrui instrumentul de evaluare, pentru a oferi o eficiență de clasificare a vulnerabilităților și a aduna diferite stiluri de programare.

Având sistemul protejat de soluțiile descrise mai sus, ne-am îndreptat atenția asupra analizei actualizărilor care sunt adăugate peste sistemul stabilizat. Ne propunem să îmbunătățim rezultatele obținute pentru detectarea vulnerabilității codului prin implementarea unei metode de detectare a vulnerabilității pe codul C / C ++, bazată pe normalizarea lexicului codului sursă al proiectului cu ajutorul abstractizării și apoi clasificarea utilizând algoritmi de învățare profundă dacă codul are impact suprafața de atac a sistemului sau nu.

Procesul propus are patru etape care includ: construirea unui scrapper de cod de la GitHub, analiza statică pentru a produce etichete utilizate pentru anonimizarea codului, instruirea algoritmilor de învățare profundă și analizarea rezultatelor acestora.

Utilizăm instrumente extra-clang pentru a analiza nodurile contextuale AST (Arborele sintaxei abstracte) atunci când facem analize semantice. Pentru fiecare nod analizat păstrăm locul unde este declarat pentru prima dată (pentru a face diferența între aceleași nume de variabile cu domeniu / scop diferit) cu ajutorul clasei ASTContext oferite de clang. Adunăm toate blocurile C ++: Declarații variabile, Declarații tip, Declarații etichetă, Declarații spațiu nume, Aliasuri spațiu nume, Declarații șablon încorporat, Constante enum etc. Menținem granularitatea la nivel de fișier, astfel încât să putem identifica apeluri API ilegale. La sfârșitul transformării, convertim fișierul într-un dicționar de jetoane aranjat în ordinea de analiză, care poate fi utilizat pentru analize pe baza algoritmilor definiți, pe care îl configurăm pentru a identifica configurațiile optime. În cazul LSTM bidirecțional, am aranjat rezultatul sub forma unei matrici uriașe, cu o netezime diferită, până când am descoperit-o pe cea optimă în ceea ce privește timpul și precizia, dar și pe baza compatibilității cu viitoarele versiuni de LLVM și Clang.

Reușim să îmbunătățim detectarea vulnerabilităților cu ajutorul setului de date construit, procesul de anonimizare și implementarea evaluării bazate pe crossentropia binară Adam. Obținem acest lucru comparând mai multe tipuri de rețele neuronale pentru a identifica cea mai eficientă abordare în ceea ce privește precizia și viteza, inclusiv: memoria pe termen scurt (LSTM), LSTM bidirecțională, perceptron multistrat și rețelele neuronale convoluționale (CNN). Dintre acestea, am obținut cea mai bună precizie de 96% cu algoritmul bidirecțional LSTM, dar și celelalte au fost destul de apropiate.

Pentru a verifica cât de bine se comportă implementarea, îl analizăm pe setul de date SARD unde normalizăm fișierele pentru a analiza numai utilizarea vulnerabilă / bună a API-urilor, ceea ce înseamnă că nu vom furniza ca intrare restul fișierelor la cod sursă de normalizare. După transformarea codului în jetoane, împărțim setul de date în 2 grupuri: 30% testare și 70% antrenament. De asemenea, transformăm toate etichetele diferite de zero în eticheta 1, ceea ce

înseamnă vulnerabil. Există 128.198 de exemple în setul de date. Rezultatul de 96% este o îmbunătățire de 6% față de rezultatele obținute de VulDeePecker pe setul de date SARD, care include programe care conțin 126 de vulnerabilități identificate printr-un ID CWE.

Există multe compilatoare open-source care ar sprijini modificările pentru a ne permite să normalizăm programele. Am folosit clang pentru că putem obține un instrument mult mai rapid cu acesta, dar GCC funcționează și el. Pentru un timp mai ușor de realizare a instrumentelor, putem analiza doar AST generat de un compilator și îl putem folosi ca intrare, fără pierderi de informații. Modul în care am proiectat acest program este de a face posibilă adăugarea numărului de linie la vulnerabilitate în viitor pentru a-l integra într-un IDE.

O a doua concluzie pe care o putem trage este rigiditatea utilităților disponibile în producție și dificultatea de a modifica și a înțelege mecanismele acestor utilități. O astfel de sarcină mărește foarte mult timpul de producție al produselor derivate bazate pe acestea, iar un astfel de instrument ar putea ajuta la corecția mai rapidă a apariției vulnerabilităților în lanțul de producție.

Chapter 7. Concluzii

Această teză se concentrează pe definirea unui spațiu explorator pentru o arhitectură auto sigură prin proiectare, construită pornind de la o arhitectură disponibilă a vehiculului în care securitatea nu a fost considerată necesară. Includem în această teză contribuțiile originale distribuite în cinci capitole după cum urmează: în capitolul 2 discutăm stadiul tehnicii oferind o imagine de ansamblu asupra modului în care ar arăta o unitate centrală de calcul auto care susține securitatea prin proiectare; în capitolul 3 exploatăm un vehicul pentru a-i expune vulnerabilitățile și a oferi funcționalități de actualizare în capitolele 4 adăugând suport pentru Wyliodrin și proiectul Yocto pentru o interacțiune mai ușoară și în capitolul 5 extindem arhitectura cu Android și adăugăm un serviciu de recunoaștere a vorbirii ca modalitate pentru a extinde funcționalitatea sistemului; în capitolul 6 îmbunătățim securitatea cu ajutorul mecanismelor de întărire a securității de execuție de încredere bazate pe analiza de apel binar și sistem și analiza statică a codului actualizărilor înainte de a le aplica.

Rezultatele noastre publicate în domeniile educației deschise IoT, prototipurilor industriale, siguranței și securității auto au primit feedback excelent din partea comunității științifice. Spațiul explorator pentru domeniul securității auto se dovedește a fi cuprinzător și versatil, cu un număr tot mai mare de contribuții până în prezent. În rândurile următoare voi încerca să ofer câteva concluzii lucrării mele.

7.1 Prezentare generală a tezei

Lucrările pentru această teză au început când am încercat să aflăm mai multe despre vehiculele de astăzi, domeniul este o sursă foarte închisă și un fel de elitist, nu s-a oferit acces la resurse, iar vânzătorii tradiționali pentru OEM-uri au fost, de asemenea, în afara limitelor. Am fost interesați să trecem de la codul proprietar la open source pentru a studia problemele de securitate într-un mod controlat.

Prima problemă întâlnită a fost accesul dificil la interfețele, senzorii sau elementele de control ale unei mașini. S-a creat soluția INVICTUS [3], o soluție care reușește să conecteze un Raspberry Pi la rețeaua CAN a unei mașini folosind un controler ELM327 pentru comunicarea cu

interfața OBD-II prezentă pe mașină. Motivația a fost să aducă cea mai folosită platformă de prototipare, Raspberry Pi, în legătură cu domeniul auto. Am reușit să-l conectăm la internet și la rețeaua GSM făcând vehiculul în sine conectat la internet. Pentru a furniza acest lucru a fost necesară multă inginerie inversă.

Apoi, ne-am confruntat cu problema ușurinței de dezvoltare a acestei platforme hardware, care pentru moment necesită o prezență fizică. Deci, cu [117] am venit cu soluția de conectare a plăcii la platforma de dezvoltare online Wylodrin. Lucrând la această platformă, am ajuns la problema secundară care nu vă afectează direct, ci afectează comunitatea: nu toți pasionații de automobile au competențe de programare și electronică. Așadar, pe lângă conectarea la platforma Wyliodrin, am creat o extensie pentru a ajuta oamenii să conecteze senzori și dispozitive suplimentare la Raspberry Pi fără a fi nevoie de cunoștințe electronice și am folosit opțiunea de programare a blocurilor pentru persoanele fără abilități avansate de programare. Motivația a fost utilizarea unei soluții open-source pentru a ajuta cât mai mult din comunitatea auto.

Cu toate acestea, Raspberry Pi este o platformă care acoperă doar un caz de utilizare al platformei digitale auto pe care am prezentat-o în stadiul actual al tehnicii [211], o arhitectură care cuprinde opțiuni multiple de sisteme de operare și niveluri de securitate mixte. Astfel am ajuns la problema aducerii unui produs industrial în arhitectura dvs. și păstrarea în același timp a evoluției făcute anterior. Soluția a venit prin lucrarea [120] care reușește să integreze un dispozitiv industrial, care are conexiuni similare cu extensia pentru Raspberry Pi, în cadrul platformei Wyliodrin.

Putem spune că în acest moment am rezolvat problema platformei hardware pentru interacțiunea cu ECU-urile din domeniul auto. Următoarea problemă pe care am avut-o a fost cu alegerea unui sistem de operare. Am găsit soluția folosind Yocto Project [118] și [119]. Este cea mai utilizată distribuție pe platforme hardware, susținută de Intel și de aproape toată lumea, avem sistemul de operare Linux Automotive Grade ca punct de plecare pentru munca noastră și oferă predictibilitatea și configurabilitatea de care avem nevoie pentru job.

Odată ce am ales sistemul de operare, am trecut la dezvoltarea unui control mai înalt și oferim accesibilitate utilizatorului. Am cercetat câmpul recunoașterii comenzilor vocale prin hârtie [115], dar considerăm că cel mai mare beneficiu este integrarea telefonului mobil cu aplicația Wyliodrin prin hârtie [116]. În acest fel am putut accesa datele senzorilor dispozitivului auto de pe telefonul mobil, nu mai necesită o conexiune fizică la interfața OBD-II, sau căutarea prin informații incomplete furnizate de monitorul dispozitivului, sau conectarea la centralizatorul furnizat anterior sau accesând chiar și o pagină web. Avantajul este accesul la un nou caz de utilizare puternic în platforma noastră digitală auto. De asemenea, am oferit controlul mașinii în cadrul aplicației, dar mai ales o nouă cale către funcții care profită și de datele telefonului mobil sau de dispozitivele legate de aceasta, cum ar fi deblocarea și pornirea vehiculului în funcție de locația dispozitivului mobil sau date de venit din brățara de fitness. Motivația a fost să creăm o arhitectură automobilistică completă, care ar putea fi sub controlul nostru, open source și cu suport complet pentru monitorizarea interacțiunilor.

O nouă oprire în călătoria noastră acum că am reușit să dezvoltăm pe deplin un sistem de la sursă deschisă de jos în sus, rezolvând problema sursei închise cu care am început. Ne dăm seama că nu mai este cazul securității prin obscuritate în domeniul nostru de testare și în domeniul auto în general și acum trebuie să ne asigurăm că ceea ce am făcut, această tranziție de la codul proprietar la open source are securitate. Deci, trebuie să revenim la întregul sistem în aceeași abordare de jos în sus și să ne asigurăm securitatea și întărirea dispozitivului.

Începem acest lucru cât de jos putem, ignorând platforma hardware pentru început, deoarece majoritatea atacurilor la acest nivel sunt laterale și dacă persoana se află deja în mașină

pentru a face asta, nu are prea mult sens în cazul nostru. Deci, pornim de la nivelul sistemului de operare și considerăm că una dintre marile probleme în sistemele auto este escaladarea privilegiilor, care în scenariul nostru mixt al lumii este fundamental, este un gând să aprinzi luminile și altul să controlezi direcția . Astfel ne-am ocupat de protecția sistemului de operare împotriva rootkit-urilor prin muncă [111]. Sprobes [111] este un concept teoretic dezvoltat pentru kernel-ul Linux versiunea 2.6 și validat într-un mediu sintetic. De asemenea, le-am folosit și le-am implementat într-un scenariu specific auto, din păcate, rezultatul sa dovedit a nu fi suficient de convingător pentru contextul industrial pe care l-am solicitat.

Trecând mai departe prin dezvoltare, un pas a fost actualizarea codului. Problema este cât de siguri suntem că noul cod încărcat nu este cumva vulnerabil. Am abordat această problemă în [121], care analizează vulnerabilitățile unui cod sursă folosind o tranzacție de cod token și am testat trei tipuri de algoritmi de învățare automată pentru a clasifica vulnerabilitatea, ajungând la o precizie de 96,27%. Am dorit o confirmare că noul cod pe care îl veți încărca pe platformă nu crește suprafața atașării și nu a avut niciun impact asupra securității ca atare.

Acum putem considera sistemul de operare securizat, dar trebuie totuși să securizăm codul de servicii externe disponibil în formă binară, ceva obișnuit în industria auto. Problema este cum putem verifica aplicațiile binare pe care nu avem acces la codul sursă. Soluția vine prin muncă [109]. Repetăm lucrarea anterioară [121], dar de această dată încercăm algoritmi ML diferiți pentru a detecta binare vulnerabile care ajung la 95% folosind pădure aleatorie. Motivația este de a verifica binarul pentru vulnerabilități.

Acuratețea de 95% nu este aceeași cu 96,27%. Înaintând, am decis că avem nevoie și de o soluție pentru a proteja procesele în curs de desfășurare și în special baza de date vulnerabilă a politicilor de securitate. Prin lucrare [110] propunem un HIDS bazat pe apeluri de sistem folosind un arbore de decizie. Motivația este îmbunătățirea sistemului. În cele din urmă avem arhitectura prezentată în [211] oferind securitate prin design.

7.2 Contribuții

Lucrarea prezentată în această teză este elaborată în domeniul securității sistemelor de operare cu aplicabilitate în industria auto. Teza conține contribuții originale la domeniile educației deschise IoT, prototipurilor industriale, siguranței și securității auto, permițând cercetări și dezvoltări suplimentare pentru industria auto.

În timp ce lucram la teză, mai ales în timpul prototipării manuale, am observat nu numai faptul că un anumit set de abilități tehnice erau necesare pentru dezvoltarea acestor sisteme. În articolele [115], [116], [117], [118], [119] și [120] prezentate în capitolele 6 și 11 este prezentat un rezumat al abilităților, împreună cu modul în care acestea pot fi integrate în cadrul sistemului de învățământ convențional încorporat. 7. Aici puteți găsi, de asemenea, opțiunile tehnologice și implementările practice.

Când discutăm despre automobile, ne imaginăm o sursă apropiată și o comunitate foarte elitistă. Cu munca pe care am făcut-o în [120] am reușit să demonstrăm că până și hardware-ul industrial poate deveni accesibil persoanelor care nu sunt atât de tehnice și toate acestea într-o manieră mai sigură, permițând totuși programarea cu ajutorul limbajelor de programare grafice, astfel încât oamenii tehnici să poată interacționa și configura în funcție de nevoile lor și să poată înțelege tehnologia. Sperăm să putem face același lucru pentru domeniul auto, ca să oferim nu numai un mediu mai sigur, ci și unul mai accesibil pentru utilizatorul non-tehnic cu care să

interacționeze. Toate acestea, deoarece în era vehiculelor autonome și zburătoare, toată lumea ar trebui să poată înțelege tehnologia din jur și chiar să își personalizeze plimbările în funcție de nevoile și dorințele lor. Intenția mea este de a oferi un cadru adecvat și metrici și metodologie corespunzătoare pentru a respecta atât aspectele educaționale, cât și cele de securitate ale lucrării.

În primul capitol introduc noțiunile, agenții, factorii și atributele implicate și utilizate în domeniul auto care sunt luate în considerare pentru această lucrare. Prezent stadiul tehnicii și caracteristicile generale ale domeniului securității auto, de asemenea până la sfârșitul capitolului prezint câteva modele de securitate care pot fi implementate în cadrul deja disponibil pentru identificarea cerințelor de securitate ale sistemelor auto. Facem o comparație a instrumentelor, metodelor și proceselor disponibile pentru evaluarea amenințărilor și riscurilor.

Contribuție 1: Siguranța automobilelor

Articol - Invictus: Open-Source Solution for an Intelligent Vehicle Security System

Aceasta este o explorare a aspectelor critice de siguranță și fiabilitate ale unei arhitecturi de rețele interne a vehiculului și prezintă o arhitectură funcțională, studii de caz de utilizare și experimente de hacking pe unul dintre cele mai utilizate protocoale de rețea din interiorul vehiculului, rețeaua CAN. Mi-am concentrat munca aici pe exploatările ECU bazate pe CAN, deoarece din analiza noastră aceasta este cea mai răspândită comunicare în interiorul vehiculului, dar în acest capitol trecem, de asemenea, peste acea muncă și discutăm despre exploatările disponibile în toate protocoalele de comunicare interne și externe. Discut, de asemenea, despre utilizarea pe scară largă a exploitelor Linux pe care le avem pe părțile vehiculului în infotainment în vehicul (IVI) și Autonom Drive (AD). Această parte nu este trecută cu vederea, dar s-au făcut mult mai multe cercetări în această privință și le analizăm doar pe cele mai relevante pentru industria auto. Nu discut alte sisteme de operare disponibile, cum ar fi QNX sau Android, deoarece acestea nu fac parte din munca noastră, ci discutăm de exemplu, deoarece cele mai puternice ECU-uri bazate pe Linux interacționează cu acesta și chiar controlează siguranța vehiculului, șoferului, pasagerilor și pietonilor. Prin aceste experimente practice pe care le-am realizat sau documentat reușim să oferim o măsurare în timp real, una pe care o putem compara cu rezultatele pe care le obținem prin soluțiile noastre furnizate.

Contribuția 2: Securitatea automobilelor

Articol - Adrem: System call based intrusion detection framework

Articol - Applications of machine learning in malware detection

Articol - Vulnerability analysis pipeline using compiler based source to source translation and deep learning

Articol - Observations over SPROBES mechanism on the TrustZone architecture

Trecând mai departe în secțiunea următoare discutăm despre soluțiile de rulare. În capitolul 4 discut soluțiile bazate pe învățarea automată pe care le-am validat în articolele [109] și [110] în care studiem clasificările. În articolul [110] discutăm despre modul în care putem folosi apelurile de sistem pentru a identifica activitățile dăunătoare și în articolul [109] discutăm despre modul în care interacțiunea cu diferite biblioteci de sistem poate ajuta la clasificarea malware-ului în categorii. Cu aceste două soluții oferim un mijloc de identificare și clasificare a interacțiunilor rău intenționate prin seturile de date pe care le-am folosit și le-am furnizat în munca noastră.

În capitolul 5 avem un alt tip de soluții de evaluare a runtime, care nu este o soluție externă, dar bazată pe suportul intern al hardware-ului și funcționalitățile acestuia, este cazul articolului [111] sau a codului sursă disponibil ca pentru lucrarea descrisă în articolul [121]. Pentru munca din [111] am folosit suportul ARM TrustZone și kernel-ul Linux extins și firmware-ul de încredere pentru a oferi un nou tip de sonde, sigure numite sprobe care inspectează registrele și paginile

pentru a proteja sistemul de operare împotriva rootkiturilor sau amenințări similare și mențineți sistemul de operare liber de contaminare chiar și atunci când contaminarea este deja ambalată în interiorul sistemului. În timp ce lucrați în [121], am construit o bază de date de cod sursă în care conținutul codului sursă a fost abstractizat pentru a aplica algoritmi de învățare automată de top care sunt instruiți pentru a detecta problemele CVE sau CWE sub forma vulnerabilităților codului sursă.

Contribuție 3: Educație deschisă IoT

Articol - Teaching computer engineering concepts to non-technical students
Carte - Learning Embedded Linux Using the Yocto Project
Carte - Linux: Embedded Development

Interacțiunea cu diagnosticul la bordul vehiculului (OBD-II) oferă acces la vehicule către o întreagă categorie de persoane și acesta este doar unul dintre punctele de intrare în interiorul arhitecturii vehiculului. Trebuie să ne asigurăm că aceste tipuri de interacțiuni, de la persoane complementare industriei auto, nu pun în pericol șoferii, pasagerii și ceilalți participanți la trafic. Ajutați oamenii care nu sunt tehnici să interacționeze cu dispozitive încorporate, să dobândească abilități de inginerie computerizată interacționând cu programarea cu ajutorul Scratch și Blockly plus o extensie hardware Raspberry Pi. Aspectele tehnice se învârt în jurul utilizării Wyliodrin Lab, o platformă de extensie dezvoltată intern și o interfață web pentru controlul spațiului de lucru al elevului, inspectarea progresului acestora și evaluarea acestora. Cu ajutorul celor două cărți îi învăț pe oamenii tehnici cum să interacționeze cu Yocto Project, un proiect de colaborare de referință care ajută dezvoltatorii să creeze și să interacționeze cu sisteme de operare personalizate bazate pe Linux, indiferent de arhitectura hardware. Descoperim nu numai ce este proiectul Yocto și cum să-l folosim, ci și modul în care proiectul Yocto influențează industrii precum auto, IoT și industria open source în general.

Contribuție 4: Prototipare industrială

Articol - Hardware design and implementation of an Intelligent Haptic Robotic Glove
Articol - Android application that controls Internet of Things devices Wyliodrin COMPANION

Proiectat și implementat o mănușă robotizată pentru a ajuta persoanele care au suferit un accident să recupereze motricitatea în degete. Mănușa este activată de comenzi vocale predefinite și este concepută pentru a ajuta personalul medical să ajute pacienții să se recupereze mai repede, de aici am extrapolat o soluție care simulează un ECU auto conectat la diverși senzori controlați cu același cadru de recunoaștere a vorbirii. Dezvoltă o aplicație Android pentru a controla și interacționa cu dispozitivele încorporate de la distanță, însoțitoare de Wyliodrin Studio IDE, utilizată pentru programarea dispozitivelor încorporate de studenți non-tehnici cu Scratch și limbaje de programare grafice similare.

În concluzia tezei mele, discut cum toate aceste contribuții sunt benefice pentru un sistem de operare mai robust și un domeniu auto. Discutați despre modul în care pentru un ecosistem cu adevărat sigur avem nevoie de soluții aplicate la mai multe niveluri de interacțiune a sistemelor, cele care oferă valori care pot fi observate, validate și îmbunătățite în timp. Soluție dinamică care nu numai că poate asigura, ci și menține securitatea unui vehicul, deoarece acesta rămâne pe drumuri timp de cel puțin 10 sau 20 de ani.

7.3 Eforturii viitoare

În călătoria de a schimba un ecosistem cu accent pe securitate prin obscuritate către unul care oferă securitate prin proiectare, trebuie să se facă mai multe direcții ale contribuțiilor viitoare. Am grupat contribuțiile noastre în patru categorii numite educație deschisă IoT, prototipare industrială, siguranță auto și securitate auto și există o mulțime de direcții de urmat, dar principalele au implicat îmbunătățirea arhitecturii auto pe care am proiectat-o.

Vrem mai întâi să ne concentrăm asupra transformării automate a procesului de integrare pentru a facilita sprijinirea noilor componente și servicii. Acest lucru se poate face cu tehnologiile și suportul hardware deja disponibile și are rolul de a simplifica interacțiunea, dar nu aduce multă valoare adăugată, îmbunătățindu-se în principal pe stadiul tehnicii.

O contribuție valoroasă pe care am vrea să o urmărim este analiza intrinsecă a datelor de pe bancul de testare pe care l-am construit. Aducerea mai multor asistențe specifice auto ar fi o bună contribuție valoroasă pentru asta, extinzând și testul în acest proces.

Contribuții mai punctuale și, de asemenea, perspicace ar oferi un mecanism de grupare a informațiilor firului ca parte a studiului HIDS, am putea folosi aici un algoritm similar cu ceea ce am folosit pentru analiza binară în care am definit clase de suport rău intenționat și benign. Contribuții similare ar putea fi furnizate din extinderea componentelor statice de analiză a codului sursă, unde am putea permite integrarea intrărilor mai complexe. Mai multe detalii sunt disponibile în fiecare capitol.

References

[1] Robert N. Charette, “This car runs on code,” IEEESpectrum, Feb 01, 2009. [Online]. Available: https://spectrum.ieee.org/transportation/systems/this-car-runs-on-code. [Accessed Aug 16, 2019].

[2] T.C.ISO/TC-22. Iso 15765: Road vehicles – diagnostic communication over controller area network (docan). ISO, 2004-2016.

[3] Cristian Lupu, Alexandru Jan Văduva, Răzvan Rughiniș. “Invictus: Open-source solution for an intelligent vehicle system,” 2016 15th RoEduNet Conference: Networking in Education and Research (RoEduNet NER, (15):1–5, 2016.

[4] Alexandru Radovici, Ioana Culic, “Online Platform for Embedded Devices”, The 10th International Scientific Conference eLearning and software for Education Bucharest, April 24-25, 2014.

[5] Lukas Zöscher, Jasmin Grosinger, Raphael Spreitzer, Ulrich Muehlmann, Hannes Gross, Wolfgang Bösch, “Concept for a security aware automatic fare collection system using HF/UHF dual banc RFID transponders”, 45th European Solid-State Device Research Conference (ESSDERC), 2015.

[6] Sastry Duri, Marco Gruteser, Xuan Liu, Paul Moskowitz, Renald Perez, Moninder Singh, Jung-Mu Tang, “Framework for security and privacy in automotive telematics”, WMC '02 Proceedings of the 2nd international workshop on Mobile commerce, 2002.

[7] Christoph Busold, Ahmed Taha, Christian Wachsmann, Alexandra Dmitrienko, Hervé Seudié, Majid Sobhani, Ahmad-Reza Sadeghi, “Smart keys for cyber-cars: secure smartphonebased NFC-enabled car immobilizer”, CODASPY '13 Proceedings of the third ACM conference on Data and application security and privacy, 2013.

[8] Răzvan Tătăroiu, Dan Tudose, “Remote Monitoring and Control of Wireless Sensor Networks”, 17th International Conference of Control Systems and Computer Science CSCS17, vol. 1, pp. 187-192, 2009.

[9] S. Craig, The Car Hacker’s Handbook – “A Guide for the Penetration Tester”, San Francisco: No Starch Press, 2016.

[10] Dr. Charlie Miller, Chris Valesek, “Remote exploitation of an Unaltered Passenger Vehicle”, Aug 10, 2015.

[11] Sasan Jafarnejad, Lara Codeca, Walter Bronzi, Raphael Frank, Thomas Engel, “A Car Hacking Experiment: When Connectivity meets Vulnerability”, IEEE Globecom Workshops (GC Wkshps), 2015.

[12] T. C. ISO/TC 22, ISO 11898: Road vehicles -- Controller area network (CAN), ISO, 1993-2015.

[13] E. Electronics, “ELM327 OBD to RS232 Interpreter Manual,” ELM Electronics – Circuits for the Hobbyist, 2016.

[14] SIMCom, SIM5218 Serial AT Command Manual, 1.21 ed., Shanghai: SIM Tech, 2011.

[15] E. Electronics, “ELM327 OBD to RS232 Interpreter Manual,” ELM Electronics – Circuits for the Hobbyist, 2016.

[16] X. Ge, H. Vijayakumar, and T. Jaeger. “SPROBES: Enforcing kernel code integrity on the trustzone architecture”, In Proceedings of the 2014 Mobile Security Technologies (MoST) workshop, 2014.

[17] R. Toulson and T. Wilmshurst, “Fast and Effective Embedded Systems Design: Applying in ARM mbed”, Oxford: Newnes, 2012
[18] CAN in Automation. “Challenges in automotive applications”, 2005. [Online]. Available: http://www.cancia.org/applications/passengercars/challenge.html. [Accessed Aug 16, 2019].
[19] Jon Erickson, “Hacking: The Art of Exploitation”, 2nd edition, No Starch Press, Inc., 2007.
[20] Karl Koscher, Alexei Czeskis, Franziska Roesner, Shwetak Patel, and Tadayoshi Kohno, Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage “Experimental Security Analysis of a Modern Automobile”, USENIX Security Symposium, 2010.
[21] Stephen Checkoway, 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.
[22] F. Simonot-Lion, “In-car embedded electronic architectures: how to ensure their safety”, presented at the 5th IFAC Int. Conf. Fieldbus Systems and Their Applications (FeT 2003), Aveiro, Portugal, 2003.
[23] P. Kleberger, T. Olovsson, and E. Jonsson, “Security aspects of the in-vehicle network in the connected car,” in Intelligent Vehicles Symposium (IV), 2011 IEEE. IEEE, 2011, pp. 528–533.
[24] I. Studnia, V. Nicomette, E. Alata, Y. Deswarte, M. Kaˆaniche, and Y. Laarouchi, “Survey on security threats and protection mechanisms in embedded automotive networks,” in Dependable Systems and Networks Workshop (DSN-W), 2013 43rd Annual IEEE/IFIP Conference on. IEEE, 2013, pp. 1–12.
[25] M. Wolf, A. Weimerskirch, and T. Wollinger, “State of the art: Embedding security in vehicles,” EURASIP Journal on Embedded Systems, vol. 2007, no. 1, p. 074706, 2007.
[26] T. Garfinkel, M. Rosenblum et al., “A virtual machine introspection-based architecture for intrusion detection.” in NDSS, 2003.
[27] P. D. Noble, “When virtual is better than real [operating system relocation to virtual machines],” in Hot Topics in Operating Systems, 2001. Proceedings of the Eighth Workshop on. IEEE, 2001, pp. 133–138.
[28] L. Litty, H. A. Lagar-Cavilla, and D. Lie, “Hypervisor support for identifying covertly executing binaries.” in USENIX Security Symposium, 2008, pp. 243–258.
[29] M. I. Sharif, W. Lee, W. Cui, and A. Lanzi, “Secure in-vm monitoring using hardware virtualization,” in Proceedings of the 16th ACM conference on Computer and communications security. ACM, 2009, pp. 477–487.
[30] X. Jiang and X. Wang, “out-of-the-box monitoring of vm-based highinteraction honeypots,” in Recent Advances in Intrusion Detection. Springer, 2007, pp. 198–218.
[31] M. Rosenblum and T. Garfinkel, “Virtual machine monitors: Current technology and future trends,” Computer, vol. 38, no. 5, pp. 39–47, 2005.
[32] K. Nance, B. Hay, and M. Bishop, “virtual machine introspection,” IEEE Computer Society, 2008.
[33] B. D. Payne, M. Carbone, M. Sharif, and W. Lee, “Lares: An architecture for secure active monitoring using virtualization,” in Security and Privacy, 2008. SP 2008. IEEE Symposium on. IEEE, 2008, pp. 233–247.

[34] A. Seshadri, M. Luk, N. Qu, and A. Perrig, “Secvisor: A tiny hypervisor to provide lifetime kernel code integrity for commodity oses,” ACM SIGOPS Operating Systems Review, vol. 41, no. 6, pp. 335–350, 2007.
[35] R. Riley, X. Jiang, and D. Xu, “Guest-transparent prevention of kernel rootkits with vmm-based memory shadowing,” in Recent Advances in Intrusion Detection. Springer, 2008, pp. 1–20.
[36] X. Zhang, L. van Doorn, T. Jaeger, R. Perez, and R. Sailer, “Secure coprocessor-based intrusion detection,” in Proceedings of the 10th workshop on ACM SIGOPS European workshop. ACM, 2002, pp. 239–242.
[37] N. L. Petroni Jr, T. Fraser, J. Molina, and W. A. Arbaugh, “Copilota coprocessor-based kernel runtime integrity monitor.” in USENIX Security Symposium. San Diego, USA, 2004, pp. 179–194.
[38] J. Wang, A. Stavrou, and A. Ghosh, “Hypercheck: A hardware-assisted integrity monitor,” in Recent Advances in Intrusion Detection. Springer, 2010, pp. 158–177.
[39] PaX Team, “Documentation for the PaX project - overall description,” 2008. [Online]. Available: https://pax.grsecurity.net/docs/pax.txt. [Accessed Aug 16, 2019].
[40] S. Andersen and V. Abella, “Data execution prevention. Changes to functionality in microsoft windows xp service pack 2, part 3: Memory protection technologies,” 2004.
[41] OpenBSD, “OpenBSD Innovations,” 2008. [Online]. Available: http://www.openbsd.org/innovations.html. [Accessed Aug 16, 2019].
[42] H. Shacham, “The geometry of innocent flesh on the bone: Return-into-libc without function calls (on the x86),” in Proceedings of the 14th ACM conference on Computer and communications security. ACM, 2007, pp. 552–561.
[43] T. Wang, K. Lu, L. Lu, S. Chung, and W. Lee, “Jekyll on ios: when benign apps become evil,” in USENIX Security Symposium, 2013, pp. 559–572.
[44] ARM Inc., “ARM Security Technology Building a Secure System using TrustZone Technology,” 2009. [Online]. Available: http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.prd29-genc-009492c/index.html. [Accessed Aug 16, 2019].
[45] ARM Inc., “ARM OS use of translation tables”, 2018 [Online]. Available: https://developer.arm.com/products/architecture/cpu-architecture/a-profile/docs/100940/latest/os-use-of-translation-tables. [Accessed Aug 16, 2019].
[46] J. Winter, “Trusted computing building blocks for embedded linux-based arm trustzone platforms,” in Proceedings of the 3rd ACM workshop on Scalable trusted computing. ACM, 2008, pp. 21–30.
[47] Azab, Ahmed M., et al. "Hypervision across worlds: Real-time kernel protection from the arm trustzone secure world." Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2014.
[48] ARM Inc., “SMC CALLING CONVENTION System Software on ARM Platforms”, 2018 [Online]. Available: http://infocenter.arm.com/help/topic/com.arm.doc.den0028b/ARM_DEN0028B_SMC_Calling_Convention.pdf. [Accessed Aug 16, 2019].
[49] A. Wespi, M. Dacier, and H. Debar. Intrusion detection using variable-length audit trail patterns. In Recent Advances in Intrusion Detection (RAID), Toulouse, France, October 2000.

[50] Christopher Kruegel, Darren Mutz, Fredrik Valeur, Giovanni Vigna, On the Detection of Anomalous System Call Arguments, University of California, Santa Barbara, Department of Computer Science, 2003.
[51] Chih-Fong Tsai, Yu-Feng Hsu, Chia-Ying Lin, Wei-Yang Lin, Intrusion detection by machine learning: A review, Elsevier, 2009.
[52] Warrender, C., S. Forrrest, and B. Pearlmutter. Detecting intrusions using system calls: Alternative data models. In: IEEE Symposium on Security and Privacy, Oakland, CA, 1999, pp. 133–145.
[53] Liu, Z., G. Florez, and S.M. Bridges. A comparison of input representations in neural networks: A case study In intrusion detection. In: Proceedings of the 2002 International Joint Conference on Neural Networks, Honolulu, HI, 2002.
[54] Liao, Y.H. and V.R. Vemuri. Use of K-nearest neighbor classifier for intrusion detection. Computers & Security, 2002.
[55] Chebrolu, S., A. Abraham, and J.P. Thomas. Feature deduction and ensemble design of intrusion detection systems. Computers & Security, 2005.
[56] Gideon Creech, Developing a high-accuracy cross platform Host-Based Intrusion Detection System capable of reliably detecting zero-day attacks, PhD dissertation, The University of New South Wales, Canberra, 2013.
[57] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Logstaff, “A Sense of Self for Unix process”, Proceedings of 1996 IEEE Symposium on Computer Security and Privacy 120-128, 1996.
[58] Amr S. Abed, Charles Clancy, David S. Levy, Intrusion Detection System for Applications Using Linux Containers, Department of Electrical & Computer Engineering, Virginia Tech, Blacksburg, VA, USA, 2015.
[59] Ye, N., X.Y. Li, Q. Chen, S.M. Emran, and M.M Xu. Probabilistic techniques for intrusion detection based on computer audit data. IEEE Transactions on Systems, Man and Cybernetics—Part A: Systems and Humans, 2001.
[60] Lee, W. and W. Fan. Mining system audit data: Opportunities and challenges. SIGMOD Record, 2001.
[61] Steven A. Hofmeyr, Stephanie Forrest, and Anil Somayaji. Intrusion detection using sequences of system calls. Journal of Computer Security (JCS), 6(3):151{180, 1998.
[62] Scikit-learn “Scikit-learn User guide”, 2017 [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html. [Accessed Mar, 2017].
[63] Docker, “Docker containers: How secure are they?” 2013 [Online]. Available: http://blog.docker.com/2013/08/containers-docker-how-secure-are-they. [Accessed Mar, 2017].
[64] Sysdig, “Sysdig Documentation”, 2017 [Online]. Available: https://docs.sysdig.com/. [Accessed Mar, 2017].
[65] Sysdig, “Sysdig vs dtrace vs strace: A technical discussion”, 2017 2017 [Online]. Available: https://sysdig.com/blog/sysdig-vs-dtrace-vs-strace-a-technical-discussion/. [Accessed Mar, 2017].
[66] Scikit-learn “Scikit-learn Getting Started”, 2017 [Online]. Available: https://scikit-learn.org/stable/getting_started.html. [Accessed Mar, 2017].
[67] Github, “Test_db repository”, 2017 [Online]. Available: https://github.com/datacharmer/test_db. [Accessed Mar, 2017].

[68] MySQL, “Mysqlslap User manual”, 2017 [Online]. Available: https://dev.mysql.com/doc/refman/5.6/en/mysqlslap.html. [Accessed Mar, 2017].
[69] Sqlmap, “Sqlmap User manual”, 2017 [Online]. Available: http://sqlmap.org. [Accessed Mar, 2017].
[70] Al-Sakib Khan Pathan, The State of the Art in Intrusion Prevention and Detection, CRC Press, Taylor & Francis Group, LLC, ISBN 978-1-4822-0351-6, 2014
[71] Scikit-learn “Scikit-learn User guide”, 2017 [Online]. Available: http://scikit-learn.org/stable/modules/manifold.html. [Accessed Mar, 2017].
[72] Apache, “Frequent pattern mining – RDD-based API”, 2017 [Online]. Available: https://spark.apache.org/docs/latest/mllib-frequent-pattern-mining.html. [Accessed Mar, 2017].
[73] Ali Jalali and Sujay Sanghavi. Learning the Dependence Graph of Time Series with Latent Factors, University of Texas at Austin, 1 University Station Code:C0806, Austin, TX 78712 USA, Jun 9, 2011.
[74] C. Kruegel, D. Mutz, F. Valeur, and G. Vigna. On the detection of anomalous system call arguments. In European Symposium on Research in Computer Security, Gjvik, Norway, October 2003.
[75] G. Tandon and P. Chan. Learning rules from system call arguments and sequences for anomaly detection. In ICDM Workshop on Data Mining for Computer Security (DMSEC), pages 20-29, 2003.
[76] Sin Yeung Lee, Wai Lup Low, Pei Yuen Wong. Learning Fingerprints for a Database Intrusion Detection System. In 7th European Symposium on Research in Computer Security (ESORICS), 2002.
[77] Niels Provos. Improving host security with system call policies. In USENIX Security Symposium, Washington, DC, USA, August 2003.
[78] David Wagner and Drew Dean. Intrusion detection via static analysis. In IEEE Symposium on Security and Privacy, Oakland, CA, May 2001.
[79] R. Sekar, M. Bendre, P. Bollineni, and D. Dhurjati. A fast automaton-based method for detecting anomalous program behaviors. In IEEE Symposium on Security and Privacy, 2001.
[80] T. Garnkel, B. Pfar, and M. Rosenblum. Ostia: A delegating architecture for secure system call interposition. In USENIX Security Symposium, Washington, DC, USA, August 2003.
[81] H. Feng, J.T. Gin, Y. Huang, S. Jha, W. Lee, and B. P. Miller. Formalizing sensitivity in static analysis forintrusion detection. In IEEE Symposium on Security and Privacy, 2004.
[82] Debin Gao, Michael K. Reiter, and Dawn Song. Gray-box extraction of execution graphs for anomaly detection. In ACM conference on Computer and Communications Security (CCS), pages 318{329, Washington, DC, October 2004.
[83] Merkel, D.: Docker: lightweight linux containers for consistent development and deployment. Linux J. 2014(239), 2 (2014)
[84] Yeung, D.Y., Ding, Y.: Host-based intrusion detection using dynamic and static behavioral models. Pattern Recogn. 36(1), 229–243 (2003)
[85] Warrender, C., S. Forrrest, and B. Pearlmutter. Detecting intrusions using system calls: Alternative data models. In: IEEE Symposium on Security and Privacy, Oakland, CA, 1999, pp. 133–145.

[86] Taesoo Kim, Nickolai Zeldovich, Practical and effective sandboxing for non-root users. In: 2013 USENIX Annual Technical Conference (USENIX ATC '13), USENIX Association, 2013.
[87] A. Wespi, M. Dacier, and H. Debar. Intrusion detection using variable-length audit trail patterns. In Recent Advances in Intrusion Detection (RAID), Toulouse, France, October 2000.
[88] Towards data Science, "The Random Forest Algorithm," 2017 [Online]. Available: https://towardsdatascience.com/the-random-forest-algorithm-d457d499cd. [Accessed Aug 16, 2019].
[89] XGBoost, "XGBoost Documentation," 2017 [Online]. Available: http://xgboost.readthedocs.io/en/latest/model.html. [Accessed Aug 16, 2019].
[90] Felan Carlo C. Garcia and Felix P. Muga II. Random forest for malware classification. CoRR, abs/1609.07770, 2016.
[91] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. CoRR, abs/1603.02754, 2016.
[92] S. Hahn, M. Protsenko, and T. Muller. Comparative evaluation of machine learning based malware detection on android. In M. Meier, D. Reinhardt, and S. Wendzel, editors, Sicherheit 2016 - Sicherheit, Schutz und Zuverl•assigkeit, pages 79{88, Bonn, 2016. Gesellschaft fur Informatik e.V.
[93] M. S. Alam and S. T. Vuong 2013. Random forest classification for detecting android malware. IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, Beijing, 2013 pp. 663-669, 2013.
[94] B. Sun, Q. Li, Y. Guo, Q. Wen, X. Lin, and W. Liu. Malware family classification method based on static feature extraction. 2017 3rd IEEE International Conference on Computer and Communications (ICCC), Chengdu, 2017, pp. 507-513, 2017.
[95] A. Wespi, M. Dacier, and H. Debar. Intrusion detection using variable-length audit trail patterns. In Recent Advances in Intrusion Detection (RAID), Toulouse, France, October 2000.
[96] Jonathan Crowe. New "ovidiy stealer" malware makes credential theft cheap and easy, 2017 [Online]. Available: https://blog.barkly.com/ovidiy-stealer-credential-theft-malware. [Accessed Aug 16, 2019].
[97] Ali Jalali and Sujay Sanghavi. Learning the Dependence Graph of Time Series with Latent Factors, University of Texas at Austin, 1 University Station Code:C0806, Austin, TX 78712 USA, Jun 9, 2011.
[98] Anton Ivanov and Orkhan Mamedov. The return of mamba ransomware, 2017 [Online]. Available: https://securelist.com/the-return-of-mamba-ransomware/79403/. [Accessed Aug 16, 2019].
[99] Andy Green. "Dnsmessenger: 2017's most beloved remote access trojan (rat)," 2017 [Online]. Available: https://blog.varonis.com/dnsmessenger-2017s-beloved-remote-access-trojan-rat/. [Accessed: 17- Nov- 2017].
[100] Proofpoint, "Adylkuzz cryptocurrency mining malware spreading for weeks," 2017 [Online]. Available: https://www.proofpoint.com/us/threatinsight/post/adylkuzz-cryptocurrency-mining-malware-spreading-for-weeks-via-eternalblue-doublepulsar. [Accessed Aug 16, 2019].

[101] Sandbox, “Free Automated Malware Analysis Service powered by Falcon Sandbox,” 2017 [Online]. Available: https://reverse.it/. [Accessed: 17- Nov- 2017].
[102] VirusShare, “VirusShare Research,” 2017 [Online]. Available: https://virusshare.com/. [Accessed: 10- Jan- 2018].
[103] Freewareles, “Freeware Files Free Software Downloads and Reviews,” 2017 [Online]. Available: https://freewareles.com/. [Accessed: 26- Nov- 2017].
[104] CNET, “Windows PC Software Free Downloads and Reviews,” 2017. [Online]. Available: http://download.cnet.com/windows/. [Accessed: 25- Nov- 2017].
[105] M. Baykara and B. Sekin. A novel approach to ransomware: Designing a safe zone system. In 2018 6th International Symposium on Digital Forensic and Security (ISDFS), pages 1-5, March 2018.
[106] Q. Chen and R. A. Bridges. Automated behavioral analysis of malware: A case study of wannacry ransomware. In 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), pages 454-460, Dec 2017.
[107] Cockoo, “Cockoo Sandbox,” 2010. [Online]. Available: https://cuckoosandbox.org. [Accessed: 14- Nov- 2017].
[108] CNET, “CNET News,” 2017. [Online]. Available: https://www.cnet.com/news/wannacry-wannacrypt-uiwix-ransomware-everything-you-need-to-know. [Accessed: 14- Nov- 2017].
[109] JA Văduva, VR Pașca, IM Florea, R RUGHINIȘ. Applications of Machine Learning in Malware Detection. In 2019 eLearning & Software for Education Vol. 2, April 2019.
[110] JA Văduva, RE Chișcariu, I Culic, IM Florea, R RUGHINIȘ. ADREM: System Call Based Intrusion Detection Framework. In 2019 eLearning & Software for Education Vol. 1, Jan 2019.
[111] JA Vaduva, S Dascalu, IM Florea, I Culic, R Rughinis. Observations over SPROBES Mechanism on the TrustZone Architecture. 2019 22nd International Conference on Control Systems and Computer Science (CSCS), pages 317-322, May 2019.
[112] Ravinder R. Ravula, Kathy J. Liszka, and Chien-Chung Chan, Learning Attack Features from Static and Dynamic Analysis of Malware, International Joint Conference on Knowledge Discovery, Knowledge Engineering, and Knowledge Management, pages 109-125, 2011.
[113] Mcafee, “Reports: a good decade for cybercrime,” 2010 [Online]. Available: http://www.mcafee.com/us/resources/reports/rp-good-decade-for-cybercrime.pdf. [Accessed Aug 16, 2017].
[114] Messagelabs, “Reports: 2010 forensics evaluation,” 2011 [Online]. Available: http://www.messagelabs.com/mlireport/MLI_2011_01_January_Final_en-us.pdf. [Accessed Aug 16, 2017].
[115] N Popescu, D Popescu, A Cozma, AJ Vaduva. Hardware design and implementation of an Intelligent Haptic Robotic Glove. 2014 International Conference and Exposition on Electrical and Power Engineering (EPE), pages 174-177, October 2014.
[116] EA Stoican, A Radovici, A Văduva. Android application that controls Internet of Things devices Wyliodrin COMPANION. 2015 14th RoEduNet International Conference-Networking in Education and Research (RoEduNet NER), pages 233-237, September 2015.
[117] JA VADUVA, A RADOVICI, I CULIC. Teaching computer engineering concepts to non-technical students. 2019 " CAROL I" National Defence University Publishing House.
[118] Alexandru Vaduva. Learning Embedded Linux Using the Yocto Project. Ed. Packt Publishing Ltd, June 2015.

[119] Alexandru Vaduva. Linux: Embedded Development. Ed. Packt Publishing Ltd, September 2016.
[120] I Culic, A Radovici, L Moraru, C Radu, JA Vaduva. Porting JerryScript to NXP Rapid Prototyping Kit, 2020 eLearning & Software for Education, April 2020.
[121] JA Vaduva, S Dascalu, I Culic, A Radovici, R Rughinis. Vulnerability analysis pipeline using compiler based source to source translation and deep learning. 2020 eLearning & Software for Education, April 2020.
[122] NSA, "NSA's Center for Assured Software. Juliet Test Suite C/C++," 2019. [Online]. Available: https://samate.nist.gov/ SARD/around.php#juliet_documents. [Accessed Jun 8, 2019].
[123] CERN, "Guide for common Vulnerabilities using C language," 2009. [Online]. Available: https://security.web.cern.ch/security/ recommendations/en/codetools/c.shtml. [Accessed Jun 8, 2019].
[124] M. Stevanovic. Advanced C and C++ Compiling. Apress, 2014. ISBN 9781430266686.
[125] C. manning Publications. Deep Learning with Python, Francois Chollet, 2018: Python. Deep learning with Python. Bukupedia, 2018.
[126] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems, pp. 1097–1105, 2012.
[127] Christian Sandberg, HoliSec Automotive Security and Privacy Holistic Approach to Improve Data Security, March 2003.
[128] EVITA Project, "E-safety Vehicle Intrusion Protected Applications (EVITA)". [Online]. http://www.evita-project.org/. [Accessed Nov 14, 2016]
[129] ETSI. Intelligent Transport Systems (ITS); Security; Threat, Vulnerability and Risk Analysis (TVRA). Technical Report TR 102 893, v1.1.1. 650 Route des Lucioles, F-06921 Sophia Antipolis Cedex, France: ETSI, Mar. 2010.
[130] ISO/SAE DIS 21434. Road vehicles — Cybersecurity engineering. Standard. International Organization for Standardization, under development.
[131] SAE J3061 Cybersecurity Guidebook for Cyber-Physical Vehicle Systems. Standard. Vehicle Cybersecurity Systems Engineering Committee, 2016.
[132] NHTSA. National Highway Traffic Safety Administration (NHTSA). [Online]. http://www.nhtsa.gov. [Accessed Dec 19, 2016].
[133] AUTOSAR Consortium. AUTomotive Open System ARchitecture. [Online]. https://www.autosar.org/. [Accessed Nov 14, 2016].
[134] SAE J3061 Cybersecurity Guide-book for Cyber-Physical Automotive Systems. Standard. Vehicle Electrical System Security Committee, 2016.
[135] SAE International. 'Serial Control and Communications Heavy Duty Vehicle Network - Top Level Document Superseding J1939 JUN2012'. In: SAE International, Aug. 2013. [Online]. https://saemobilus.sae.org/content/J1939_201308. [Accessed Nov 14, 2016].
[136] SAE International. 'European Brake Fluid Technology - J1709_200607, cancelled Jul 2006'. In: SAE International, Jul. 2006.
[137] ETSI. Intelligent Transport Systems (ITS); Security; Threat, Vulnerability and Risk Analysis (TVRA). Technical Report TR 102 893, v1.1.1. 650 Route des Lucioles, F-06921 Sophia Antipolis Cedex, France: ETSI, Mar. 2010.

[138] ETSI. Intelligent Transport Systems (ITS); Communications Architecture. European Standard EN 302 665, v1.1.1. 650 Route des Lucioles, F-06921 Sophia Antipolis Cedex, France: ETSI, Sept. 2010.
[139] ETSI. Intelligent Transport Systems (ITS); Security; Security Services and Architecture. Technical Specification TS 102 731, v1.1.1. 650 Route des Lucioles, F-06921 Sophia Antipolis Cedex, France: ETSI, Sept. 2010.
[140] ETSI. Intelligent Transport Systems (ITS); Security; Trust and Privacy Management. Technical Specification TS 102 941, v1.2.1. 650 Route des Lucioles, F-06921 Sophia Antipolis Cedex, France: ETSI, May 2018.
[141] ETSI. Intelligent Transport Systems (ITS); Security; ITS communications security architecture and security management. Technical Specification TS 102 940, v1.3.1. 650 Route des Lucioles, F-06921 Sophia Antipolis Cedex, France: ETSI, April 2018.
[142] ETSI. Intelligent Transport Systems (ITS); Security; Security header and certificate formats. Technical Specification TS 103 097 v1.3.1. 650 Route des Lucioles, F-06921 Sophia Antipolis Cedex, France: ETSI, Oct. 2017.
[143] ETSI. CYBER; Methods and protocols; Part 1: Method and pro forma for Threat, Vulnerability, Risk Analysis (TVRA). Technical Specification TS 102 165-1, v5.2.3. 650 Route des Lucioles, F-06921 Sophia Antipolis Cedex, France: ETSI, Oct. 2010.
[144] ISO/TC 204: Intelligent transport systems. Standard ISO/TC 211 - ISO/TC 204 WG: GIS-ITS under development. International Organization for Standardization.
[145] CEN/TC 278 Intelligent transport systems. Standard under development. European Committee for Standardization.
[146] IEC 61508: Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 1: General requirements. Standard IEC 61508-1:2010. International Electrotechnical Commission
[147] ISO/IEC 27034: Information technology — Security techniques — Application security — Part 1: Overview and concepts. Standard ISO/IEC 27034-1:2011. International Organization for Standardization.
[148] ISO/TC 22: Road vehicles. Standard under development. International Organization for Standardization.
[149] ISO 26262 road vehicles functional safety part 1–10. Standard. International Organization for Standardization, 2011.
[150] ISO/DIS 13400-1: Road vehicles — Diagnostic communication over Internet Protocol (DoIP) — Part 1: General information and use case definition. Standard. International Organization for Standardization.
[151] ISO 15118: Road vehicles — Vehicle to grid communication interface — Part 1: General information and use-case definition. Standard 15118-1:2019. International Organization for Standardization.
[152] ISO 15765: Road vehicles — Diagnostic communication over Controller Area Network (DoCAN) — Part 2: Transport protocol and network layer services. Standard 15765-2:2016. International Organization for Standardization.
[153] IEEE. 1609 WG - Dedicated Short Range Communication Working Group. Oct. 2013. [Online]. https://standards.ieee.org/develop/wg/1609_WG.html. [Accessed Nov 14, 2016].
[154] 802.11p-2010 - IEEE Standard for Information technology-- Local and metropolitan area networks-- Specific requirements-- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular

Environments. Jun. 2010. [Online]. https://standards.ieee.org/standard/802_11p-2010.html. [Accessed Nov 14, 2016].
[155] L. L. Bello. 'Novel trends in automotive networks: A perspective on Ethernet and the IEEE Audio Video Bridging'. In: Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA). Sept. 2014, pp. 1–8. DOI: 10.1109/ETFA.2014.7005251.
[156] IEEE-Standards Association. 802.1BA-2011 - IEEE Standard for Local and metropolitan area networks–Audio Video Bridging (AVB) Systems. Tech. rep. 2011.
[157] IEEE-Standards Association. P802.1AS - Standard for Local and Metropolitan Area Networks – Timing and Synchronization for Time-Sensitive Applications. Tech. rep. 2011.
[158] IEEE-Standards Association. 802.1Qav-2009 - IEEE Standard for Local and metropolitan area networks– Virtual Bridged Local Area Networks Amendment 12: Forwarding and Queuing Enhancements for Time-Sensitive Streams. Tech. rep. 2009.
[159] IEEE-Standards Association. 802.1Qat-2010 - IEEE Standard for Local and metropolitan area networks–Virtual Bridged Local Area Networks Amendment 14: Stream Reservation Protocol (SRP). Tech. rep. 2010.
[160] CARONTE Project. Creating an Agenda for Research On Transportation sEcurity. URL: http://www.caronte-project.eu/ (visited on 14th Nov. 2016).
[161] HEAVENS: HEAling Vulnerabilities to ENhance Software Security and Safety. [Online]. http://www.vinnova.se/sv/Resultat/Projekt/Effekta/HEAVENS-HEAling-Vulnerabilities-to-ENhance-Software-Security-and-Safety/. [Accessed Nov 25, 2016].
[162] SeFram. Security framework for vehicle communication (SeFram). [Online]. https://research.chalmers.se/en/project/6225. [Accessed June 11, 2020].
[163] SESAMO. Security and Safety Modeling (SESAMO). [Online]. http://www.sesamo-project.eu. [Accessed Dec 19, 2016].
[164] EVITA Project. E-safety Vehicle Intrusion Protected Applications (EVITA). [Online]. http://www.evitaproject.org/. [Accessed Nov 14, 2016].
[165] D. K. Nilsson and U. E. Larson. 'Simulated Attacks on CAN Buses: Vehicle Virus'. In: Proceedings of the 5th IASTED International Conference on Communication Systems and Networks. AsiaCSN '08. Anaheim, CA, USA. Palma de Mallorca, Spain: ACTA Press, 2008, pp. 66–72. ISBN: 978-0-88986-758-1.URL:http://portal.acm.org/citation.cfm?id=1713277.1713292.
[166] M. Wolf, A. Weimerskirch and C. Paar. 'Security in Automotive Bus Systems'. In: Workshop on Embedded IT-Security in Cars. Bochum, Germany, Nov. 2004.
[167] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway et al. 'Experimental Security Analysis of a Modern Automobile'. In: 2010 IEEE Symposium on Security and Privacy (SP). IEEE, 2010, pp. 447–462. ISBN: 1424468949. DOI: 10.1109/SP.2010.34.
[168] T. Hoppe and J. Dittmann. 'Sniffing/Replay Attacks on CAN Buses: A simulated attack on the electric window lift classified using an adapted CERT taxonomy'. In: Proceedings of the 2nd Workshop on Embedded Systems Security (WESS). Salzburg, Austria, 2007.
[169] J. D. Howard and T. A. Longstaff. 'A Common Language for Computer Security Incidents'. In: Sandia Report: SAND98-8667 (1998). [Online]. http://www.cert.org/research/taxonomy_988667.pdf. [Accessed Nov 14, 2016].
[170] A. Lang, J. Dittmann, S. Kiltz and T. Hoppe. 'Future Perspectives: The Car and Its IP-Address — A Potential Safety and Security Risk Assessment'. In: Proceedings of the 26th

International Conference on Computer Safety, Reliability, and Security (SAFECOMP ’07). SAFECOMP ’07. Nuremberg, Germany, Sept. 2007, pp. 40–53.

[171] T. Hoppe, S. Kiltz and J. Dittmann. ‘Security Threats to Automotive CAN Networks – Practical Examples and Selected Short-Term Countermeasures’. In: Computer Safety, Reliability, and Security. Ed. by M. D. Harrison and M.-A. Sujan. Lecture Notes in Computer Science 5219. Springer Berlin Heidelberg, Jan. 2008, pp. 235–248. ISBN: 978-3-540-87697-7, 978-3-540-87698-4. URL: http://link.springer.com/chapter/10.1007/978-3-540-87698-4_21 (visited on 19th Dec. 2016).

[172] T. Hoppe, S. Kiltz and J. Dittmann. ‘Automotive IT-Security as a Challenge: Basic Attacks from the Black Box Perspective on the Example of Privacy Threats’. In: Computer Safety, Reliability, and Security. Ed. by B. Buth, G. Rabe and T. Seyfarth. Lecture Notes in Computer Science 5775. Springer Berlin Heidelberg, Jan. 2009, pp. 145–158. ISBN: 978-3-642-04467-0, 978-3-642-04468-7. URL: http://link.springer.com/chapter/10.1007/978-3-642-04468-7_13 (visited on 19th Dec. 2016).

[173] D. K. Nilsson, U. E. Larson, F. Picasso and E. Jonsson. ‘A First Simulation of Attacks in the Automotive Network Communications Protocol FlexRay’. In: Proceedings of the International Workshop on Computational Intelligence in Security for Information Systems (CISIS’08). Ed. by E. Corchado, R. Zunino, P. Gastaldo and Á. Herrero. Vol. 53. Advances in Intelligent and Soft Computing. 10.1007/978-3-540-88181-0_11. Springer Berlin / Heidelberg, 2009, pp. 84–91. URL: http://dx.doi.org/10.1007/978-3-540-88181-0_11.

[174] M. L. Chávez, C. H. Rosete and F. R. Henríguez. ‘Achieving Confidentiality Security Service for CAN’. In: Proceedings of the 15th International Conference on Electronics, Communications and Computers, 2005. CONIELECOMP 2005. Feb. 2005, pp. 166–170. DOI: 10.1109/CONIEL.2005.13.

[175] D. K. Nilsson, U. E. Larson and E. Jonsson. ‘Efficient In-Vehicle Delayed Data Authentication Based on Compound Message Authentication Codes’. In: Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th. 2008, pp. 1–5. DOI: 10.1109/VETECF.2008.259.

[176] A. Groll and C. Ruland. ‘Secure and Authentic Communication on Existing In-Vehicle Networks’. In: 2009 IEEE Intelligent Vehicles Symposium. 2009, pp. 1093–1097. DOI: 10.1109/IVS.2009.5164434.

[177] H. Oguma, A. Yoshioka, M. Nishikawa, R. Shigetomi, A. Otsuka and H. Imai. ‘New Attestation-Based Security Architecture for In-Vehicle Communication’. In: IEEE Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. New Orleans, Louisiana, 2008, pp. 1–6. DOI: 10.1109/GLOCOM.2008.ECP.369.

[178] C. Szilagyi and P. Koopman. ‘A Flexible Approach to Embedded Network Multicast Authentication’. In: 2nd Workshop on Embedded Systems Security (WESS). 2008.

[179] S. Schulze, M. Pukall, G. Saake, T. Hoppe and J. Dittmann. ‘On the Need of Data Management in Automotive Systems’. In: 13. Fachtagung des GI-Fachbereichs "Datenbanken und Informationssysteme" (DBIS). Vol. 144. Gesellschaft für Informatik (GI). Münster, Germany, Mar. 2009.

[180] H. Schweppe, Y. Roudier, B. Weyl, L. Apvrille and D. Scheuermann. ‘Car2X Communication: Securing the Last Meter - A Cost-Effective Approach for Ensuring Trust in Car2X Applications Using In-Vehicle Symmetric Cryptography’. In: 2011 IEEE Vehicular Technology Conference (VTC Fall). 2011, pp. 1–5. DOI: 10.1109/VETECF.2011.6093081.

[181] U. E. Larson and D. K. Nilsson. ‘Securing Vehicles against Cyber Attacks’. In: CSIIRW ’08: Proceedings of the 4th annual workshop on Cyber security and information intelligence research. CSIIRW ’08. Proceedings of the 4th annual workshop on Cyber security and information intelligence research: developing strategies to meet the cyber security and information intelligence challenges ahead. New York, NY, USA: ACM, 2008, 30:1–30:3. ISBN: 978-1-60558-098-2. DOI: 10.1145/1413140.1413174. URL: http://doi.acm.org/10.1145/1413140.1413174.

[182] T. Hoppe, S. Kiltz and J. Dittmann. ‘Adaptive Dynamic Reaction to Automotive IT Security Incidents Using Multimedia Car Environment’. In: Proceedings of the 4th International Conference on Information Assurance and Security (ISIAS ’08). Sept. 2008, pp. 295–298. DOI: 10.1109/IAS.2008.45.

[183] T. Hoppe, S. Kiltz and J. Dittmann. ‘Applying Intrusion Detection to Automotive IT — Early Insights and Remaining Challenges’. In: Journal of Information Assurance and Security 4.3 (2009), pp. 226–235. URL: http://www.mirlabs.org/jias/hoppe.pdf (cit. on pp. 57, 61, 63).

[184] M. Müter, A. Groll and F. C. Freiling. ‘A Structured Approach to Anomaly Detection for In-Vehicle Networks’. In: 2010 Sixth International Conference on Information Assurance and Security (IAS). Atlanta, GA, Aug. 2010, pp. 92–98. DOI: 10.1109/ISIAS.2010.5604050.

[185] M. Müter and N. Asaj. ‘Entropy-Based Anomaly Detection for In-Vehicle Networks’. In: 2011 IEEE Intelligent Vehicles Symposium (IV). Baden-Baden, Germany, June 2011, pp. 1110–1115. DOI:10.1109/IVS.2011.5940552.

[186] R. Brooks, S. Sander, J. Deng and J. Taiber. ‘Automobile Security Concerns’. In: Vehicular Technology Magazine, IEEE 4.2 (June 2009), pp. 52–64. ISSN: 1556-6072. DOI: 10.1109/MVT.2009.932539.

[187] G. Graham and B. Cohen. Microsoft Security Development Lifecycle Adoption. Microsoft, Edison Group. Sept. 2013.

[188] ISO/IEC 27034: Information technology — Security techniques — Application security — Part 1: Overview and concepts. Standard ISO/IEC 27034-1:2011. International Organization for Standardization.

[189] OWASP (Open Web Application Security Project). Open Software Assurance Maturity Model (SAMM), Version 1.0. Mar. 2009. URL: http://www.opensamm.org/downloads/ SAMM-1.0.pdf (visited on 2nd Dec. 2019).

[190] G. McGraw, S. Migues and J. West. BSIMM-V (Building Security In Maturity Model, Version 5.0). Oct. 2013. [Online]. http://www.bsimm.comm/download/dl.php. [Accessed Nov 22, 2019].

[191] OWASP (Open Web Application Security Project). Comprehensive, Lightweight Application Security Process (CLASP). [Online]. http://www.owasp.org. [Accessed Oct 28, 2019].

[192] Cisco Systems, Inc. Cisco Secure Development Lifecycle (CSDL). URL: [Online]. http://www.cisco.com/web/about/security/cspo/csdl/index.html. [Accessed Oct 28, 2019].

[193] Foundstone (McAfee). Secure Software Development Lifecycle (SSDLC). 2008. [Online]. http://www.mcafee.com/us/resources/data- sheets/foundstone/ds- secure- software-devlife-cycle.pdf. [Accessed Nov 22, 2019]

[194] P. Hellström. Master’s Thesis: Tools for Static Code Analysis: A Survey. 2009. DOI: LIU-IDA/LITHEX-A--09/003--SE.

[195] B. De Win, R. Scandariato, K. Buyens, J. Grégoire and W. Joosen. 'On the secure software development process: CLASP, SDL and Touchpoints compared'. In: Information and software technology 51.7 (2009), pp. 1152–1171.
[196] The STRIDE Threat Model. https://msdn.microsoft.com/en-us/library/ee823878(v=cs.20).aspx. Accessed: 2016-11-25 (cit. on p. 74).
[197] G. Macher, E. Armengaud, E. Brenner and C. Kreiner. 'A Review of Threat Analysis and Risk Assessment Methods in the Automotive Context'. In: International Conference on Computer Safety, Reliability, and Security. Springer. 2016, pp. 130–141.
[198] H. Bidgoli. Handbook of Information Security, Threats, Vulnerabilities, Prevention, Detection, and Management. Handbook of Information Security. Wiley, 2006. ISBN 9780470051214.
[199] C. Cummins, P. Petoumenos, A. Murray, and H. Leather. Compiler fuzzing through deep learning. In Proceedings of the 27th ACM SIGSOFT International Symposium on Software Testing and Analysis, pages 95–105. ACM, 2018.
[200] M. Dowd, J. McDonald, and J. Schuh. The art of software security assessment: Identifying and preventing software vulnerabilities. Pearson Education, 2006.
[201] Lea, Perry. Internet of Thing for Architects, Packt Publishing, 2018
[202] Baker, Jean; Smart Board in the Music Classroom, Music Educators Journal, 2007, Volume 93, Issue 5, pp. 18-19
[203] Smith, David; Smart Clothes and Wearable Technology, AI&Society: Journal of Knowledge, Culture and Communication, 2007, Volume 22, Issue 1, pp. 1-3
[204] Fraser, Neil; Ten things we've learned from Blockly, IEEE Blocks and Beyond Workshop, 2015
[205] Richardson, Matt; Wallace, Shawn. Getting Started with Raspberry Pi, Maker Media, Sebastopol, California, 2013
[206] Raspberrypi, "Raspberrypi Education," 2019 [Online]. https://www.raspberrypi.org/education/. [Accessed Feb 7, 2019].
[207] SeeedStudio "Grove System," 2019. [Online]. http://wiki.seeedstudio.com/Grove_System/. [Accessed Feb 7, 2019].
[208] Kasser, Joseph; Tran, Xuan-Linh; Matisons, Simon; Prototype Educational Tools for Systems and Software (PETS) Engineering, Engineering Education for a Sustainable Future: Proceedings of the 14th Annual Conference for Australasian Association for Engineering Education and 9th Australasian Women in Engineering Forum, 2003, pp. 532-543
[209] Crawley, Edward; Malmqvist, Johan; Ostlund, Soren; Brodeur, Doris; Edstrom, Kristina. Rethinking Engineering Education, Springer International Publishing, Switzertland, 2014
[210] Heydt, Gerald; Vittal, Bijay; Feeding our profession [power engineering education], IEEE Power and Energy Magazine, 2003, Volume 99, Issue 1, pp. 38-45
[211] Thomas Rosenstatter, A State-of-the-Art Report on Vehicular Security, April 2017.
[212] ARM Inc., "Interaction of Normal and Secure worlds," 2019. [Online]. https://developer.arm.com/documentation/100935/0100/Interaction-of-Normal-and-Secure-worlds-?lang=en. [Accessed Feb 17, 2020].
[213] G. Grebenstein, "A Method for Hand Kinematic Designers", ICABB, Venice, Italy, 2010.
[214] M.C. Carrozze, F. Vecchi, F. Sebastiani, G. Cappiello,S. Roccella, M. Zecca, R. Lazzarini and P Dario, "Experimental Analysis of an Innovative Prosthetic Hand with Proprioceptive Sensors," Proc. IEEEIntl. Conf. Rob. Aut., Taipei, Taiwan, pp. 2230-2235, 2003.

[215] A. Wege and G. Hommel, “Development and control of a hand exoskeleton for rehabilitation of hand injuries," IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, Canada, 2005.
[216] Beigi, Homayoon, “Fundamentals of Speaker Recognition”, New York: Springer, 2011
[217] S. E. Levinson, L. R. Rabiner and M. M. Sondhi, "An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition", in Bell Syst. Tech. Jnl. v62(4), pp1035--1074, April 1983
[218] A. Radovici, I. Culic, “Open cloud platform for programming embedded systems”, Networking in Education and Research, RoEduNet,vol. 12, pp. 1-2, 2013.
[219] A. Radovici, I. Culic, I. Bocicor, A. Armean, M. Ene, “ Platformă educațională care permite programarea dispozitivelor embedded din Cloud – Wyliodrin”, Conferinţa Naţională de Învăţământ Virtual, 9th edition, vol. 9, pp. 57-61, 2013.
[220] Peter Saint-Andre, Kevin Smith, and Remko Troncon. XMPP: The Definitive Guide. 2009.