Raport științific

Proiect de cercetare PN-III-P1-1.1-PD-2016-1198
Creșterea securității și evaluarea vulnerabilităților pentru rețele standardizate utilizate în industrie

Etapa Nr.1 Analiza vulnerabilităților din protocoale pentru rețele standardizate în industrie

Autor

ș.l.dr.ing. Pal-Ștefan MURVAY

Universitatea Politehnica Timișoara

Noiembrie 2018

Prezentul raport prezintă activitatea științifică desfășurată în cadrul proiectului de cercetare *PN-III-P1-1.1-PD-2016-1198, Creșterea securității și evaluarea vulnerabilităților pentru rețele standardizate utilizate în industrie (Security enhancements and vulnerability assessment for industry-standard networks - SEVEN)* aferentă etapei nr. 1, *Analiza vulnerabilităților din protocoale pentru rețele standardizate în industrie*.

În cadrul raportului vom reda sintetic rezultatele obținute, detalii tehnice suplimentare fiind disponibile în lucrările asociate. Rezultatele științifice obținute în cadrul acestei etape sunt prezentate în 2 lucrări acceptate și prezentate în cadrul unor conferințe de specialitate în domeniul securității. Articolele publicate, disponibile pe pagina web a proiectului (http://www.aut.upt.ro/~pal-stefan.murvay/projects/SEVEN/publications.html), sunt:

- Pal-Stefan Murvay, Bogdan Groza, *Practical security exploits of the FlexRay in-vehicle communication protocol*, The 13th International Conference on Risks and Security of Internet and Systems (CRISIS 2018), 2018
- Pal-Stefan Murvay, Bogdan Groza, *A brief look at the security of DeviceNet communication in industrial control systems*, Proceedings of the Central European Cybersecurity Conference 2018, pp. 5:1-5:6, ACM, 2018.

Activitatea principală prevăzută în această prima etapă a proiecului a constat în analiza unor protocoale de comunicare industriale standardizate utilizate în diverse sectoare industriale în vederea identificării de vulnerabilități care ar putea să fie exploatate pentru realizarea unor atacuri. Au fost alese pentru studiu două protocoale de comunicare. Primul dintre ele este FlexRay, protocol dezvoltat și folosit pentru industria auto în comunicarea pe rețelele interne ale autovehiculelor. Cel de-al doilea protocol, DeviceNet, este utilizat în sisteme industriale de control. În secțiunile următoare sunt detaliate rezultatele obținute în urma analizei protocoalelor anterior menționate.

1 Analiza vunlerabilităților protocolului FlexRay

Protocolul de comunicare FlexRay a fost dezvoltat penrtu domeniul automotive având ca țintă aplicații în care este necesară asigurarea transmiterii la timp a mesajelor. Astfel de aplicații implementează funcționalități responsabile de siguranța ocupanților autovehicolului precum break-by-wire sau steer-by-wire. Utilizarea protocolului FlexRay cu precădere în aplicații responsabile de siguranță în exploatarea autovehiculelor a fost factorul determinant în alegerea acestuia pentru analiză.

FlexRay face parte din categoria de protocoale time-triggered in care transmiterea de mesaje este organizată în cadru unui ciclu de comunicare. Cicli de comunicare sunt repetitivi și păstrează aceeași lungime. În cadrul unui ciclu de comunicare FlexRay sunt definite două segmente în care nodurile pot transmite mesaje: segmentul static și segmentul dinamic. Segmentul static este împărțit în sloturi care pot fi alocate nodurilor din rețea pentru transmiterea de mesaje. Un nod poate transmite pe parcursul unui slot static doar dacă acesta i-a fost alocat. Segmentul dinamic a fost conceput pentru a permite transmiterea mesajelor cauzate de apariția unor evenimente, mesaje a căror transmise nu trebuie să fie ciclică.

Protocolul FlexRay se folosește de layerul fizic și de data-link layer. Aceste două elemente sunt implementate de către un transceiver FlexRay și respectiv controller-ul FlexRay. Transceiverul transformă informația din forma ei digitală în semnale fizice transmise pe cele două linii de comunicare folosite îndeplinind în același timp și funcția inversă de traducere a semnalelor fizice în formă digitală. Controllerul este responsabil de implementarea protocolului de comunicare.

În ceea ce privește securitatea, protocolul FlexRay nu prevede funcționalități intrinseci pentru a asigura acest aspect al comunicării. Dat fiind acest fapt, am analizat posibilitățile de atac asupra comunicării folosind protocolul FlexRay. În acest scop am evaluat două tipuri de abordări pentru implementarea unor atacuri. Prima abordare presupune generarea de semnale pe canalul de comunicare, de la nivelul aplicației, doar prin intermediul transceiverului. A doua abordate se folosește de ajutorul controllerului pentru a genera și transmite mesaje. Folosind cele două abordări au fost identificate o serie de atacuri de tip DoS (Denial of Service) și spoofing (falsificarea de mesaje).

Identificarea atacurilor s-a realizat prin studiul specificației urmând mai apoi implementarea atacurilor pe un sistem experimental. Sistemul experimental, ilustrat în Figura 1, constă dintr-o rețea FlexRay cu 4 noduri și o componentă folosită pentru analiza traficului. Fiecare din nodurile rețelei FlexRay a fost implementat folosind o placă de dezvoltare EVB9S12XF512E echipată cu microcontroller S12XF512 și transceivere FlexRay TJA1080A. Pentru analiza traficului a fost utilizat un echipament PicoScope conectat la un PC.

Prezentăm în continuare detaliile ce fac posibile atacurile identificate precum și informații despre modul în care acestea au fost implementate pe platforma experimentala realizată pentru acest scop.

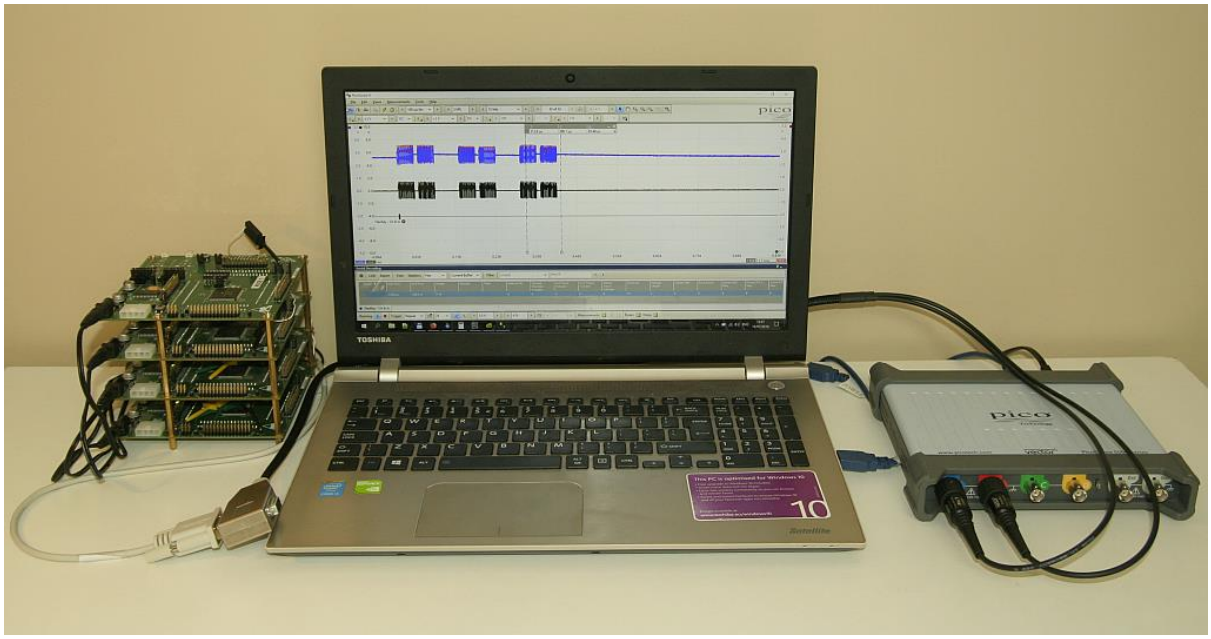

Figura 1. Sistemul experimental utilizat în realizarea atacurilor asupra protocolului FlexRay, conform cu [1]

1.1 DoS pentru întreaga comunicare

Prima variantă de atac identificată este prevenirea completă a transmiterii de mesaje. Acest atac este făcut posibil de sistemul de detecție a erorilor implementat de protocolul FlexRay. În cazul în care frame-urile folosite ca referință pentru sincronizare nu sunt recepționate corect pe parcursul unui număr configurabil de cicli de comunicare nodurile încetează transmisia de mesaje.

Acest tip de atac poate fi realizat folosind oricare din cele două abordări menționate anterior. Folosind controlul direct al transceiverului se poate genera un nivel dominant constant (corespunzător fie pentru 0 logic fie pentru 1 logic) pe canalul de comunicare. generând coliziuni și făcând imposibilă interpretarea corectă a datelor transmise. Efectul final este pierderea sincronizării și, în cele din urmă, încetarea comunicării. Cea de-a doua variantă de implementare a atacului se folosește de controllerul FlexRay pentru a rezerva toate slot-urile de transmisie și a le ocupa cu frame-uri. Rezultatul, similar primei abordări, este incapacitatea nodurilor legitime de a transmite mesaje fără a cauza coliziuni fapt care duce la pierderea sincronizării și oprirea comunicării. O variantă optimizată a atacului constă în ocuparea ilegală doar a sloturilor rezervate pentru frame-urile folosite pentru sincronizare.

Efectele atacului sunt redate în Figura 2 ce ilustrează cele trei variante ale atacului. Zona marcată cu fundal verde conține un ciclu de comunicare în cazul ăn care atacul nu este prezent. În toate cele trei cazuri atacul începe odată cu zona redată pe fundal roșu ce marchează durata de timp necesară pentru a determina terminarea comunicării. După cum se poate observa, cea

de-a treia variantă de atac prezintă cel mai mic impact asupra aspectului unui ciclu normal de comunicare.

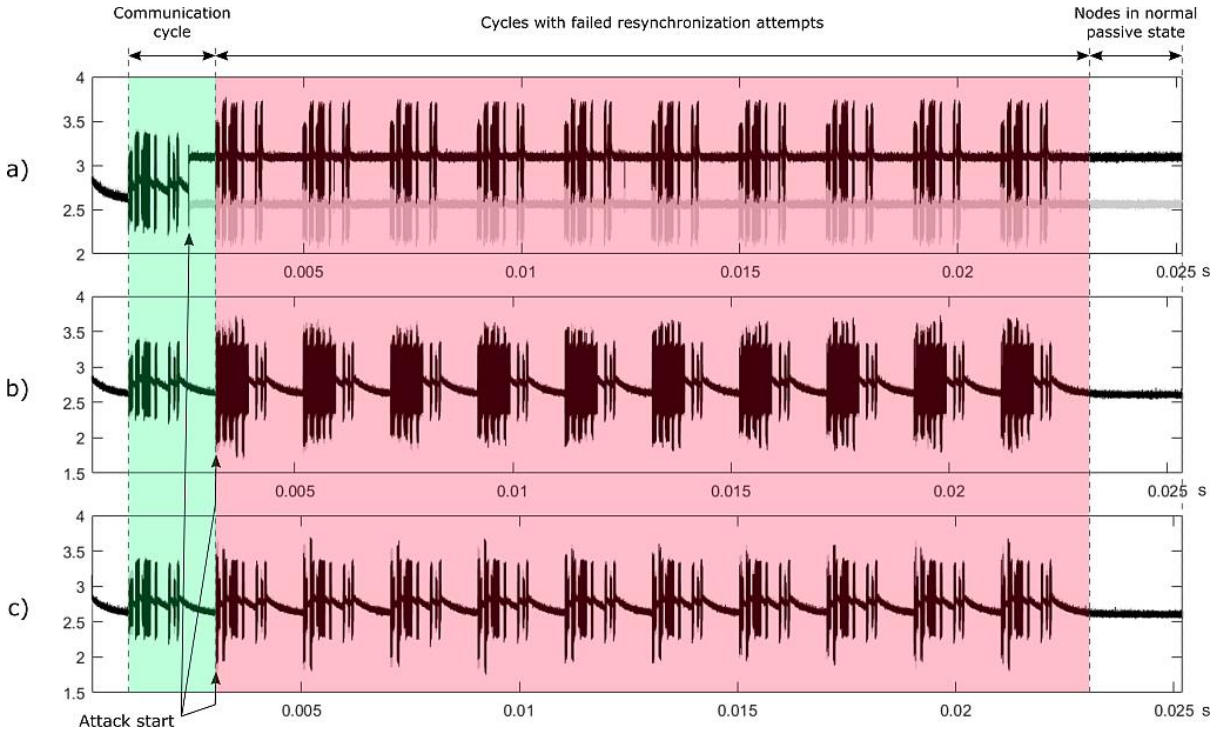

Figura 2. Atacul de tip DoS pentru întreaga comunicare: a) folosind transceiverul, b) folosind controllerul pentru a ocupa toate sloturile, c) folosind controllerul în varianta optimizată, conform cu [1]

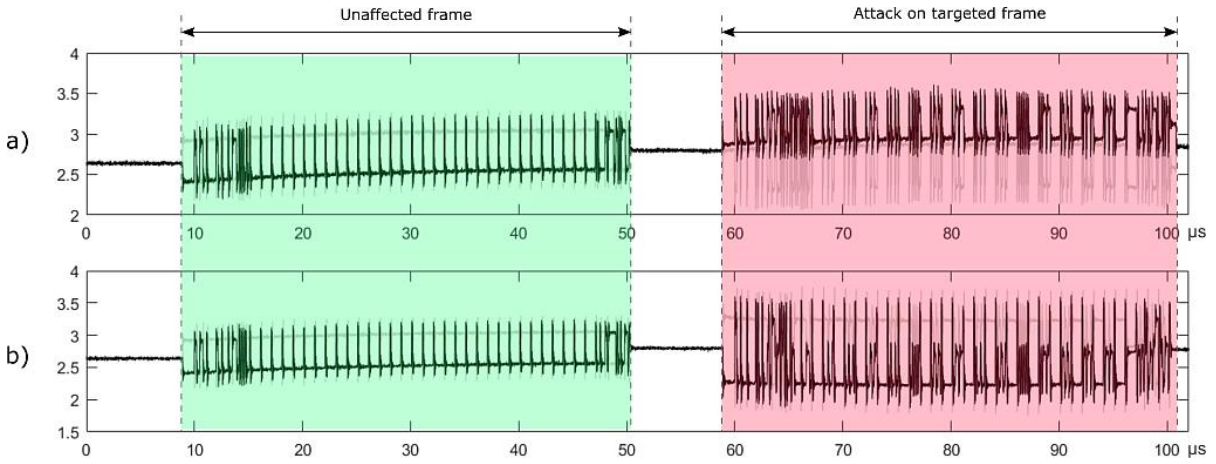

Figura 3. Atacul de tip DoS direcționat: a) folosind transceiverul, b) folosind controllerul, conform cu [1]

1.2 DoS direcționat

Atacul de tip DoS direcționat presupune împiedicarea recepționării corecte pentru anumite mesaje fără a duce neapărat la sistarea completă a transmisiunilor în rețea. Pentru implementarea atacului folosind doar transceiverul FlexRay este necesară implementarea la nivelul aplicației a unei rutine care să identifice în ciclul de comunicare intervalul în care se transmite mesajul țintă pentru generarea adecvată a coliziunilor. În varianta de atac ce folosește

controllerul, atacatorul trebuie doar să ocupe cu un mesaj propriu slotul alocat mesajului țintă. Pentru mesajele transmise în segmentul dinamic nu se poate cunoaște momentul transmisiei. Astfel pentru a împiedica transmisia unor mesaje din acest segment atacatorul trebuie să asigure transmisia continua a mesajului de atac rezultând într-o serie suplimentară de transmisii pe lângă cele care colizionează cu mesajul țintă.

Efectele atacului în cele două abordări sunt ilustrate în Figura 3 ce reprezintă pe fundal verde forma unui mesaj normal, iar pe fundal roșu forma mesajului atacat.

1.3 Falsificarea mesajelor

Falsificarea mesajelor presupune transmiterea de către atacator a unor mesaje menite să pară că sunt transmise de către unul din nodurile legitime ale rețelei. În ceea ce privește mesajele transmise în segmentul static falsificarea nu este posibilă atâta timp cât nodurile legitime transmițătoare ale acestor mesaje sunt active în rețea. Motivul este dat de specificația FlexRay care obligă nodurile pentru care un slot static a fost alocat să transmită în fiecare ciclu un mesaj în acel slot. În ceea ce privește sloturile dinamice, falsificarea de mesaje este posibilă atâta timp cât mesajul legitim nu este transmis. Transmiterea mesajului legitim odată cu cel falsificat va produce coliziuni și imposibilitatea interpretării corecte a mesajului.

2 Analiza vulnerabilităților protocolului DeviceNet

DeviceNet este un protocol folosit în sisteme industriale de control. Face parte dintr-o familie de 4 protocoale, Common Industrial Protocol (CIP), devoltate pentru a servi în același tip de sisteme. CIP reprezintă specificația comună a celor 4 protocoale ce implementează layerele superioare ale modelului OSI. Pentru layerul fizic, data-link și rețea fiecare din cele 4 protocoale prezintă o implementare diferită. DeviceNet folosește protocolul Controller Area Network (CAN) pentru a implementa layerul fizic și data-link.

Binecunoscut datorită utilizării sale pe scară largă în domeniul automotive, CAN a fost studiat în ceea ce privește securitatea, fiind cunoscut faptul că este vulnerabil la atacuri de tip DoS și spoofing. În studiul protocolului DeviceNet ne-am propus identificarea vulnerabilităților specifice layerelor superioare adăugate peste CAN pornind de la vulnerabilitățile cunoscute pentru CAN în general.

DeviceNet implementează un model de comunicare bazat pe conexiuni. Pentru a putea accesa rețeaua, un nod DeviceNet trebuie mai întâi să își asocieze o adresă unică numită MAC ID. Această operație se face verificând dacă adresa dorită nu este deja folosită de alt nod al

rețelei. Doar daca MAC ID-ul este unic în rețea nodul asociat are voie să continue comunicarea. Fiecare nod DeviceNet este modelat ca o colecție de obiecte care pot fi implicate în comunicare. Pentru a putea comunica intre ele obiectele aflate în diferite noduri ale rețelei trebuie să stabilească conexiuni. Stabilirea de conexiuni între obiecte ale diferitelor noduri este necesară pentru a permite transmiterea datelor relevante pentru controlul proceselor. Există două tipuri de conexiuni ce pot fi stabilite într-o rețea DeviceNet:

- *Conexiuni explicite* –folosite în general pentru transmisiuni generale legate de protocol
- *Conexiuni I/O* –folosite pentru a transporta informații specifice folosite pentru controlul proceselor

Pentru stabilirea de conexiuni într-o rețea DeviceNet există trei metode posibile:

- *Unconnected Message Manager (UCMM)* – un obiect special care, daca este implementat, permite stabilirea de conexiuni explicite prin folosirea unor mesaje specifice care pot fi transmise în afara unei conexiuni.
- *Obiect Conexiune existent* – pe baza unei conexiuni existente se pot stabili conexiuni noi adresând clasa din care face parte obiectul conexiune utilizat.
- *Set de conexiuni predefinit* – în cazul în care resursele nodului sunt reduse se poate implementa setul de conexiuni predefinit ce conține o conexiune explicită și 4 conexiuni I/O. Pentru a folosi aceste conexiuni, un nod master trebuie să obțină controlul asupra setului de conexiuni.

În urma analizei protocolului DeviceNet au fost identificate o serie de atacuri care, după tipologie, se încadrează în două categorii: atacuri de tip DoS și spoofing (falsificare). Acestea sunt descrise în cele ce urmează.

2.1 Atacuri DoS

1.Prevenirea accesului la rețea. Acest atac presupune intervenția atacatorului în procesul de verificare a unicității MAC ID-ului prin transmiterea unui răspuns la cererea de verificare emisă de nodul țintă. Recepția unui răspuns la cererea de verificare a prezenței de MAC ID-uri duplicate face ca nodul emitent al cererii să înceteze comunicarea în așteptarea unei reconfigurări pentru setarea unei valori noi pentru MAC ID.

2.Prevenirea conexiunilor prin UCMM. O cerere de conexiune prin UCMM poate să primească un răspuns pozitiv, caz în care conexiunea este considerată stabilită, sau un răspuns ce indică o eroare în procesul de stabilire a conexiunii. Un atacator care dorește să prevină

stabilirea de conexiuni prin această metodă poate răspunde înaintea nodului legitim indicând inițiatorului conexiunii prezența unei erori.

3.Închiderea conexiunilor prin UCMM. O conexiune deja stabilită poate să fie închisă folosind serviciul de închidere a conexiunilor prin intermediul UCMM. Un atacator poate să folosească această funcționalitate pentru a închide orice conexiune stabilită astfel.

4.Prevenirea și închiderea conexiunilor prin clasa de conexiune. Asemănător atacurilor asupra stabilirea de conexiuni prin UCMM sunt posibile atacuri care să prevină stabilirea de conexiuni sau închiderea celor existente folosind clasa de conexiune în cazul utilizării unei conexiuni deja existente.

5.Epuizarea conexiunilor posibile. Fiecare nod dintr-o rețea DeviceNet poate stabili un număr limitat de conexiuni (27) conform specificației protocolului. Ocuparea tuturor conexiunilor face ca nodul să nu poată stabili noi conexiuni. Astfel, un atacator ar putea preveni un nod să stabilească conexiuni legitime forțând stabilirea falsă a tuturor conexiunilor posibile pentru acel nod. Pentru o rețea cu 64 noduri (maximul permis de protocol), în care comunicarea se face folosind o rata de 125 kbps, un atacator ar putea stabili toate conexiunile posibile în cel mult 1.8s.

6.Prevenirea utilizării setului de conexiuni predefinit. În cazul nodurilor care implementează setul de conexiuni predefinit, utilizarea acestora este condiționată de dobândirea de către un nod master a accesului la setul de conexiuni al unui nod slave. Pentru a preveni utilizarea acestor conexiuni, un atacator va obține accesul la setul de conexiuni și nu îl va ceda. În cazul în care un nod legitim a obținut deja accesul la setul de conexiuni predefinit atacatorul poate falsifica mai întâi mesajul de eliberare a controlului asupra acestor conexiuni.

7.Perturbarea transmisiunilor fragmentate. Mesajele care sunt prea lungi pentru a fi transmise într-un singur frame de CAN sunt fragmentate și transmise folosind protocolul DeviceNet pentru fragmentarea mesajelor. Protocolul presupune ca fragmentele să fie transmise în ordine însoțite de un număr indicând numărul fragmentului. Transmiterea unui fragment cu un număr care nu urmează în ordine crescătoare după fragmentul precedent face ca nodul receptor să considere transmisia încheiată și să șteargă fragmentele primite anterior. Astfel, un atacator poate preveni recepția corectă a mesajelor fragmentată transmțând un fragment cu un număr de ordine greșit.

2.2 Mesaje falsificate

1.Modificarea caracteristicilor conexiunii. Fiecare conexiune are un set de parametrii ce definesc caracteristicile transmisiunulor ce au loc folosind acea conexiune. Acești parametrii pot fi modificați prin mesaje specifice. Un atacator poate falsifica astfel de mesaje modificând parametrii esențiali în comunicare. Spre exemplu, unul din parametrii asociați unei conexiuni definește frecvența transmiterii mesajelor. Modificarea frecenței pentru mesaje ce transportă date necesare unui algoritm de reglare poate afecta eficiența acestui algoritm.

2.Falsificarea datelor transmise. Marea parte amesajelor transmise întro rețea DeviceNet transportă date folosite în procesul de control (feedback dat de senzori sau comenzi pentru actuatoare). Prin falsificarea mesajelor un atacator poate injecta valori false ca feedback de la senzori sau comenzi pentru actuatoare afectând rezultatul procesului.

3 Concluzii

Rezultatele științifice obținute în această etapă a proiectului de cercetare demonstrează încă odată existența vulnerabilităților în protocoalele de comunicare folosite în aplicații din diferite sectoare ale industriei. Am identificat vulnerabilități noi precum și efectele pe care poate să le aibă exploatarea acestora de către atacatori în cazul a două protocoale de comunicare: FlexRay și DeviceNet. Rezultatele obținute sunt detaliate în două lucrări științifice [1], [2] ce au fost prezentate în conferințe de specialitate în domeniul securității.

În continuare cercetarea în cadrul proiectului se concentrează pe dezvoltarea de mecanisme de protecție pentru vulnerabilitățile protocoalelor. Sunt avute în vedere atât mecanisme necesare protocoalelor studiate în prima fază a proiectului cât și altor protocoale de comunicare din industrie care prezintă vulnerabilități.

4 Referințe

[1] Pal-Stefan Murvay, Bogdan Groza, *Practical security exploits of the FlexRay in-vehicle communication protocol*, presented at The 13th International Conference on Risks and Security of Internet and Systems (CRISIS 2018), 2018

[2] Pal-Stefan Murvay, Bogdan Groza, *A brief look at the security of DeviceNet communication in industrial control systems*, Proceedings of the Central European Cybersecurity Conference 2018, pp. 5:1-5:6, ACM, 2018.