VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

EMULACE ÚTOKŮ NA ŘÍDICÍ KOMUNIKACI SCADA/ICS

EMULATION OF ATTACKS ON SCADA/ICS COMMUNICATION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE **Bc. PETER GROFČÍK**
AUTHOR

VEDOUCÍ PRÁCE **Ing. PETR MATOUŠEK, Ph.D., M.A.**
SUPERVISOR

BRNO 2022

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav informačních systémů (UIFS) Akademický rok 2021/2022

Zadání diplomové práce

Student:	**Grofčík Peter, Bc.**
Program:	Informační technologie a umělá inteligence
Specializace:	Počítačové sítě
Název:	**Emulace útoků na řídicí komunikaci SCADA/ICS** **Emulation of Attacks on SCADA/ICS Communication**
Kategorie:	Počítačové sítě

Zadání:

1. Seznamte se s útoky na řízení průmyslových procesů SCADA/ICS na základě doporučené literatury.
2. Proveďte rešerši existujících datových sad s útoky na řídící systémy SCADA/ICS. Popište základní vektory útoků.
3. Pomocí dostupných nástrojů implementujte typické útoky na řídicí komunikaci pro vybrané SCADA/ICS protokoly. Vytvořte datové sady obsahující normální komunikací i emulované útoky.
4. Navrhněte a implementujte metodu pro detekci anomálií pomocí statistických metod. Ověřte přesnost detekce na vytvořených datových sadách.
5. Diskutujte výsledky experimentů a použitelnost metody pro praktické nasazení.

Literatura:

- Knapp ED, Langill JT. Industrial network security. Securing critical infrastructure networks for smart grid, SCADA, and other industrial control systems. Syngress; 2015.
- McCarthy J, Powell M, Stouffer K, Tang C, Zimmerman T, Barker W, Ogunyale T, Wynne D, Wiltberger J. Securing Manufacturing Industrial Control Systems: Behavior Anomaly Detection. Technical Report NISTIR-8219. National Institute of Standards and Technology; 2018.
- Databáze útoků na ICS systémy ATT&CK for Industrial Control Systems, viz https://collaborate.mitre.org/attackics, 2021.
- Keith Stouffer, Victoria Pillitteri, Suzanne Lightman, Marshall Abrams, Adam Hahn. Guide to Industrial Control Systems (ICS) Security. NIST-SP 800-82r, NIST, 2015.
- Maynard, Peter & Mclaughlin, Kieran & Sezer, Sakir. (2018). An Open Framework for Deploying Experimental SCADA Testbed Networks. 89-98. 10.14236/ewic/ICS2018.11.
- MATOUŠEK Petr, RYŠAVÝ Ondřej, HAVLENA Vojtěch and GRÉGR Matěj. Flow based monitoring of ICS communication in the smart grid. *Journal of Information Security and Applications*, vol. 2020, no. 54, pp. 102535-102535. ISSN 2214-2126.

Při obhajobě semestrální části projektu je požadováno:

- Body 1 až 3.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce:	**Matoušek Petr, Ing., Ph.D., M.A.**
Vedoucí ústavu:	Kolář Dušan, doc. Dr. Ing.
Datum zadání:	1. listopadu 2021
Datum odevzdání:	18. května 2022
Datum schválení:	11. října 2021

Abstrakt

Cieľom mojej diplomovej práce je emulácia vhodných sieťových útokov na riadiacu komunikáciu ICS/SCADA systémov s následným návrhom a implementáciou ich detekcie. V prvej časti práce sa venujem kategorizácii a oboznámeniu sa s jednotlivými typmi útokov na priemyselné siete. Pomocou klasifikovaných kategórii a rešerše dostupných dátových sád sa v nasledujúcej kapitole venujem popisu výberu vhodných útokov, ktoré majú aspoň nepriamy súvis s riadiacou komunikáciou a zároveň sú prevediteľné na vitruálnych zariadeniach komunikujúcich protokolom IEC104, ktorý som si pre prácu vybral. Takto vybrané útoky potom nasledovne vykonám a zaznamenám ich priebehy to sád PCAP súborov, ktoré tvoria vstup pre nasledovnú časť týkajúcu sa detekcie anomálií pomocou štatistických metód.

Abstract

The goal of this master's thesis is to emulate suitable network attacks on the control communication of ICS/SCADA systems with a subsequent design and implementation that can detect them. The first part of the work consists of categorization and acquaintance with individual types of attacks on industrial networks. Using classified datasets combined with a research of available datasets, in the next chapter I describe the selection of suitable attacks, which are at least indirectly related to control communication and are also feasible on virtual devices that are communicating using the IEC104 protocol, which I chose for my work. I then perform the selected attacks and record their progress in a set of PCAP files. Those files form the input for the next part that concerns anomaly detection using statistical methods.

Kľúčové slová

ICS, SCADA, Emulácia útokov, MITRE, IEC 104, DNP3, Modbus, Ettercap, Detekcia anomálii, Štatistické metódy

Keywords

ICS, SCADA, Emulation of attacks, MITRE, IEC 104, DNP3, Modbus, Ettercap, Anomaly detection, Statistical methods

Citácia

GROFČÍK, Peter. *Emulace útoků na řídicí komunikaci SCADA/ICS*. Brno, 2022. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Petr Matoušek, Ph.D., M.A.

Emulace útoků na řídicí komunikaci SCADA/ICS

Prehlásenie

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne pod vedením pána Ing. Petra Matouška. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Peter Grofčík
23. mája 2022

Poďakovanie

Rád by som poďakoval Ing. Petrovi Matouškovi za jeho odborné rady, čas a vynaložené úsilie počas konzultácií pri vedení mojej práce.

Obsah

Kapitola 1

Úvod

1.1 Motivácia

Scada systémy sa používajú nielen v priemyselných procesoch, ale aj v rôznych experimentálnych prevádzkach, či citlivých až život ohrozujúcich prostrediach. Jednotlivé systémy väčšinou predpokladajú vysoké zabezpečenie internej siete pre neautorizovaný vzdialený prístup, na základe čoho však riadiace protokoly spravidla neobsahujú vysokú úroveň zabezpečenia, čo otvára dvere pre útočníka, ktorý je schopný získať prístup do lokálnej siete riadiacich zariadení. Hlavnou motiváciou práce, je tak vytvorenie súhrnného popisu vykonateľných útokov a ich emuláciu s následnou detekciou pre odhalenie vybraných útokov, ktoré by mohol útočník s prístupom do riadiacej siete vykonať. Práca tak pomôže priblížiť jednotlivé slabšie zabezpečené časti komunikácie a poskytne možnosť ich prípadnej detekcie.

1.2 Postup práce

Cieľom práce je zoznámiť sa a priblížiť jednotlivé sieťové útoky, ktoré sú realizovateľné na riadiacej komunikácii SCADA/ICS systému. Vytvoriť všeobecný návod resp. súhrn a kategorizáciu útokov zvolených na základe doporučenej literatúry a rešerše prevedenej na poskytnutých dátových sadách a k nim dostupných technických správ. Následne emulácia jednotlivých útokov na riadiacu komunikáciu zvoleného protokolu IEC 104 [6] spolu s presným popisom konkrétnych vykonaných útokov na štandardnú komunikáciu. Každý vykonaný útok je zaznamenaný v dátovej sade pre ďalšie skúmanie a následnú experimenty s implementovanými štatistickými metódami pre detekciu anomálií na vytvorených dátových sadách.

1.3 Rozdelenie práce

V druhej kapitole sa nachádza kategorický popis útokov vytvorený pomocou databázy MITRE. V kapitole je vytvorený všeobecný súhrn efektov a účelov útokov jednotlivých kategórií, ktoré dopomôžu k zúženiu výberu útokov podľa potrebného účinku a možností zvoleného komunikačného protokolu. Výber konkrétnych útokov spolu s dôvodmi, cieľmi a všeobecným popisom sa nachádza v tretej kapitole. Na tento výber som využil aj poskytnuté datasety v kombinácii s možnosťami použitých virtuálnych zariadení komunikujúcich protokolom IEC 104. Presný popis emulovaných útokov sa nachádza v kapitole štyri. Oproti

predchádzajúcej kapitole už obsahuje konkretizovaný popis upravených parametrov či vykonaných operácii pre úspešnú realizáciu vhodne vybraných útokov.

1.4 Ciele

Cieľom tejto diplomovej práce je oboznámiť sa s typickými útokmi na riadenie priemyslových procesov SCADA/ICS. Následne pomocou voľne dostupných nástrojov implementovať a emulovať útoky, ktoré je možné viesť na zvolený priemyselný protokol IEC 104. Jedným z hlavných výstupov práce je tak vytvorený dataset odchytenej komunikácie zariadení komunikujúcich protokolom IEC 104 pod vedenými vhodnými útokmi. V neposlednom rade je ďalším cieľom navrhnúť a otestovať implementáciu detekcie anomálií v zaznamenaných datasetoch pomocou zvolených štatistických metód. Ďalší výstup tak tvorí samotný popis použitia týchto metód na odhalenie týchto anomálií vrátane zhodnotenia presnosti jednotlivých detekcií a použiteľnosti zvolených metód pre praktické nasadenie.

Kapitola 2

Útoky a taktiky na riadiace systémy ICS/SCADA

V tejto kapitole sa zameriam na súhrnnú kategorizáciu útokov na riadiace systémy SCADA/ICS spolu s popisom ich efektov a požiadaviek podľa jednotlivých kategórií. Jednotlivé útoky môžu principiálne spadať pod viacero kategórií, a to z dôvodu viacnásobného efektu útoku či rozdielneho využitia pri komplexnejšom type útoku. Účelom tejto kapitoly je špecifikácia útokov na riadenie priemyselných systémov a vyčlenenie potencionálne detekovateľných taktík a techník, ktoré môžu byť detekované z komunikácie priemyselného protokolu v sieti.

Pre kategorizáciu útokov vychádzam z databáze MITRE[1]. Jedná sa o celosvetovo dostupnú vedomostnú základňu, popisujúcu techniky a praktiky, ktoré môže útočník vykonať počas bežnej prevádzky v sieti ICS. Táto znalostná základňa, nazvaná *ATT&CK*, je používaná ako základ pre vývoj špecifických modelov, detekciu či testovanie hrozieb a metodológií v systémoch SCADA/ICS.

2.1 Počiatočný prístup (Initial Access) [TA0108]

Kategória zahŕňa techniky, ktoré neútočia na riadiacu komunikáciu ako takú, ale usilujú o získanie prístupu k dôležitým zariadeniam SCADA systému či samotnej lokálnej siete, na ktorej zariadenia komunikujú. Techniky môžu zneužívať aj bežne aktívne webové aplikácie na zariadeniach systémov ICS na získanie nepretržitého prístupu k nim. Typickým predstaviteľmi tak môžu byť útoky:

- *Remote Services* [T0886], pri ktorom útočník usiluje o využitie služieb aktívnych na jednotlivých riadiacich zariadeniach za účelom získania vzdialeného prístupu do siete, prenos citlivých dát či zamedzenia prístupu pre autorizovaných užívateľov. Napadnutou službou môže byť napríklad aj služba SSH [5], ktorá patrí medzi najbežnejšie na akýchkoľvek sieťových zariadeniach. Tím *Sandworm*[2] použil na natívne nástroje vzdialeného prístupu na priame pripojenie na pracovné stanice operátora, ktoré zneužili na ovládanie samotných komponentov ICS.
- *Wireless Compromise* [T0860], pri ktorom útočník usiluje o získanie neautorizovaného prístupu do siete s riadiacimi zariadeniami pomocou kompromitovaného bezdrôtového

[1] https://collaborate.mitre.org/attackics/index.php/Main_Page
[2] https://collaborate.mitre.org/attackics/index.php/Group/G0007

zariadenia. *Wireless Compromise* je tiež typickým predstaviteľom otváracích útokov, keďže v prípade úspešnosti je útočníkovi umožnené pokračovať ľubovoľným útokom na komunikáciu v rámci lokálnej siete.

Kategória tak pokrýva útoky vykonateľné na riadiace systémy, avšak nie priamo na riadiacu komunikáciu SCADA/ICS systému. Útoky sú využiteľné na získanie prístupu do lokálnej siete, ale využívajú služby aktívne na konkrétnych zariadeniach a nie riadiaci protokol ako taký.

2.2 Vykonanie (Execution) [TA0104]

Do tejto kategórie spadajú útoky, ktoré usilujú o rozšírenie získaného prístupu v riadiacom systéme. Jednotlivé útoky môžu byť opäť vedené pomocou otvorených služieb, alebo už aj samotného komunikačné protokolu riadiaceho systému. To je však možné len za predpokladu, že daný protokol podporuje funkcionalitu potrebnú pre samotný útok. Typickými predstaviteľmi sú útoky:

- *User execution* [T0863], ktorý tiež predpokladá interakciu s užívateľom potrebnú pre spustenie škodlivého kódu. Pre získanie rozšíreného prístupu môže byť napríklad využitý akýkoľvek protokol určený na prenos súborov či mailov. Útočník sa môže pokúsiť podvrhnúť škodlivú aplikáciu či jej súčasť, ktorá po spustení sprístupní dovtedy nedostupné dáta pre ostatných používateľov v sieti. Napríklad softvér *Bad Rabbit*[3] je zamaskovaný za inštalačný program Adobe Flash. Po otvorení súboru sa infikovaný počítač začne zamykať.

- *Change operating mode* [T0858], ktorý môže využívať samotný riadiaci protokol. Požadovaným účinkom útoku je zmena operačného módu kontrolera, čím môže dôjsť ku kompletnej zmene prenášanej komunikácie. Zariadenie môže byť prevedené z prevádzkového módu na zastavenie či vyresetovanie poskytovaných objektov. Operačné módy a ich zmena však musí byť podporovaná zariadením a v prípade využitia riadiaceho protokolu musí aj ten podporovať príkazy určujúce ich zmenu. Existujúci softwave *Triton*[4] má schopnosť zastaviť alebo spustiť program prostredníctvom protokolu TriStation.

2.3 Pretrvanie (Persistence) [TA0110]

Útoky klasifikované pod túto kategóriu spočívajú v úprave funkcionality programu či inkriminovaného zariadenia, ktoré v sieti komunikuje. Útokmi tak dochádza k zmene funkcií zariadenia ako takého, takže tieto útoky opäť nepatria pod útoky zamerané priamo na riadiacu komunikáciu.

Typickým predstaviteľom je útok *Modify program* [T0889], v ktorom sa útočník snaží upraviť, či pridať program na zariadení. Útočník môže zariadeniu podvrhnúť program, ktorý zmení fungovanie zariadenia či spôsob, akým komunikuje so svojím okolím v sieti. Pre funkčnosť útoku je však nutné, aby cielené zariadenie daný podvrhnutý program stiahlo zo vzdialeného miesta či priamo od útočníka s prístupom do lokálnej siete. softvér Stuxnet[5]

[3] https://collaborate.mitre.org/attackics/index.php/Software/S0005
[4] https://collaborate.mitre.org/attackics/index.php/Software/S0013
[5] https://collaborate.mitre.org/attackics/index.php/Software/S0010

infikuje PLC odlišným kódom v závislosti od charakteristík cieľového systému. Infekčná sekvencia pozostáva z dátových blokov, ktoré po stiahnutí do PLC upravia jeho chovanie v sieti.

Každý ďalší útok v tejto kategórii pracuje na rovnakom princípe podvrhnutia škodlivého obsahu. Líšia sa tak iba v tom, na ktorú časť zariadenia sú namierené. Podvrhnutý program tak môže ešte napríklad cieliť na citlivé informácie o účtoch v danej sieti, na infikovanie projektových súborov (inicializačných objektov, programových organizačných jednotiek) či samotný firmware zariadenia.

2.4 Eskalácia privilégii (Privilege Escalation) [TA0111]

Kategória samotná predstavuje útok, resp. využitím softvérových zraniteľností sa môže útočník pokúsiť o *eskaláciu privilégii* [T0890]. Pôvodne tak útočník má nižšie privilégiá a len základný prístup funkcionalite cieleného zariadenia. Využitím slabých miest v programe zariadenia môže získať prístup k cennejším informáciám či dokonca k možnosti zmeniť hodnoty dôležitých zdrojov zariadenia.

Okrem samotnej eskalácie privilégií môže útočník využiť aj takzvaný *hook*[6] prístup v aplikačnom programovacom rozhraní (API), pomocou ktorého môže útočník využiť opakovateľne použiteľné systémové prostriedky, resp. upraviť samotné odkazy na API funkcie uložené v import address table (príklad pre Windows prostriedky). Vyššie spomenutý softvér Triton 2.2 využíva zraniteľnosť firmvéru Tricon a umožňuje zneužiť nezabezpečené systémové volanie na získanie privilégií supervízora.

2.5 Vyhnutie (Evasion) [TA0103]

Techniky spadajúce do tejto kategórie spočívajú v maskovaní prístupu na zariadenie či úprave samotného stavu zariadenia. Útočník sa snaží zmeniť funkcionalitu zariadenia pomocou štandardných prostriedkov vo svoj prospech. Táto funkcionalita však nepredstavuje zmenu komunikácie zariadenia s okolitým prostredím, ale zmenu (resp. zmiernenie) obranných schopností samotného zariadenia. Medzi typické útoky patria:

- *Masquerading* [T0849], pomocou ktorého sa útočník snaží zamaskovať spustiteľné súbory, aby sa vyhol podozreniu kontrolného technika resp. administrátora. Útok však nemusí cieliť na program uložený na zariadení v sieti, ale môže spočívať v zamaskovaní zariadenia samotného útočníka (resp. program). Útočník sa tak môže tváriť ako legitímne zariadenie a napríklad pomocou korektných dotazov prinútiť jednotlivé stanice v sieti k zmene. Softvér Ekans[7] sa maskuje ako platný spustiteľný súbor s názvom *update.exe*. Tvári sa tak nenápadne, pretože existuje mnoho legitímnych aktualizácií, ktoré majú na pozadí rovnaký názov aktívneho procesu.

- *Change operating mode* [T0858]. Už spomenuté korektné dotazy môžu obsahovať legitímne žiadosti na zmenu operačného módu dotazovaného zariadenia. V tom momente sa jedná o doplnenie útokom *Change operating mode*, ktorým sa útočník snaží prepnúť dotazované zariadenie pomocou prepnutia do vhodnejšieho operačného módu. Tieto módy nie sú štandardizované, ale z hľadiska funkcionality to môžu byť napríklad

[6] Termín zahŕňa celý rad techník používaných na zmenu alebo rozšírenie správania OS, aplikácii či iných softvérových komponentov zachytením udalostí či správ prenášaných medzi softvérovými komponentami.

[7] https://collaborate.mitre.org/attackics/index.php/Software/S0017

módy pre povolenie zmeny interného programu, štandardnej prevádzky zariadenia či povoleného vzdialeného prístupu. Takáto zmena módu tak môže útočníkovi sprístupniť funkcionalitu zariadenia, ktorú by za bežných okolností nebol schopný zneužiť z dôvodu jej neaktívnosti.

2.6 Objavenie (Discovery) [TA0102]

Táto kategória zaberá špeciálne miesto v akejkoľvek sieťovej komunikácii, nie len v ICS/SCADA systémoch. Nezaoberá sa útočením na žiadnu konkrétnu komunikáciu. Jej hlavným cieľom je zber informácií o v sieti zapojených zariadeniach. Takto zozbierané informácie sú pre útočníka užitočné hlavne pri výbere cieľa jednotlivých nasledujúcich útokov, jeho funkcie v rámci siete či aktívnych služieb na cieľovom zariadení. Užitočné útoky spadajúce do tejto kategórie sú:

- *Network connection enumeration* [T0840], pri ktorom útočník využitím štandardných nástrojov (napr. Netstat[8]) objavuje do siete pripojené zariadenia. Po objavení zariadenia môže ďalej skúmať aktívne služby (porty), štandardné informácie o softvéru na zariadení či postavenie zariadenia v sieti pomocou ďalších nástrojov ako napríklad *nmap*[9]. Napríklad aj softvér Industroyer[10] obsahuje modul, ktorý objavuje všetky pripojené sieťové adaptéry, aby určil ich masky podsiete TCP/IP.
- *Network sniffing* [T0842]. Funkcia zariadenia v sieti môže byť však často nejasná zo samotných aktívnych služieb, či takto získané štandardné informácie o zariadení nemusia stačiť k rozhodnutiu jeho postavenia v sieti (komunikácii). Aj pre to sa môže útočník pokúsiť o nadväzujúci útok z tejto kategórie, *Network sniffing*. Z odchytenej komunikácie môže byť vo väčšine komunikačných protokolov pomerne jednoduché odhadnúť funkciu (master/slave) či prípadne samotné určenie zariadenia (senzor, riadiaca stanica,...). Tieto informácie môžu zohrať rolu pri rozhodovaní útočníka o forme či cieli útoku podľa požadovaného efektu. V najlepšom prípade (pre útočníka) môže samotná odchytená komunikácia obsahovať nezašifrované citlivé informácie (prihlasovacie údaje, kritické hlásenia o stave a podobne).

2.7 Postranný postup (Lateral Movement) [TA0109]

Techniky spadajúce pod túto kategóriu využívajú vedľajšie aktívne služby na zariadeniach. Nezahŕňa tak žiadny útok s priamym efektom na riadiacu komunikáciu, ale usiluje sa o zneužitie menej bezpečnej a aktívnej služby na zariadení pre získanie vyšších privilégií. Typickými predstaviteľmi sú:

- *Exploitation of remote services* [T0866]. Útok funguje principiálne rovnako ako už spomenutý *Remote services* v časti *Počiatočný prístup* (sekcia 2.1). Hlavný rozdiel je v komplikovanosti útoku ako takého. *Remote services* priamo zneužíva samotnú aktívnu vzdialenú službu na získanie prístupu do prostredia ICS systému, zatiaľ čo jeho rozšírená varianta sa snaží zneužiť služby, implementačné chyby, či samotný operačný systém na aktiváciu služby, či funkcionality na nej, ktorá by následne mohla

[8] https://en.wikipedia.org/wiki/Netstat
[9] https://linux.die.net/man/1/nmap
[10] https://collaborate.mitre.org/attackics/index.php/Software/S0001

byť skompromitovaná. Tento útok sa samozrejme rovnako radí do kategórie pre *Počiatočný prístup*.

- *Default credentials* [T0812]. Útok je pomerne jednoduchý a z princípu spočíva len vo využití implicitných predvolených prihlasovacích údajov. Aj keď medzi základné pravidlá platí zmena takýchto údajov v momente inicializačnej konfigurácie zariadenia, tak nie je zaručené, že ich užívateľ aj skutočne zmení. Zároveň môže byť samotné zariadenie navrhnuté tak, že nie je možné zmeniť predvolené administrátorské prihlasovacie údaje a môže existovať len voľba na ich deaktiváciu po pridaní nových, ktorú užívateľ nevykonal. Softvér Stuxnet[11] používa predvolené implicitné heslo na databázovom serveri softvéru WinCC ako jeden z mechanizmov používaných na rozšírenie do systémov.

2.8 Zbieranie (Collection) [TA0100]

Kategória sa ako prvá venuje útokom na komunikáciu ako takú. Nemusí sa však stále jednať len o riadiacu komunikáciu ICS systému. Účelom každého útoku je práve zber prenášaných informácií, ale tie však nie vždy musia byť prenášané priamo v riadiacej komunikácii. Typickými predstaviteľmi sú:

- *Network sniffing* [T0887], ktorý som už priamo predstavil v sekcii 2.6.
- *Men in the Middle* [T0830], ktorý funguje v pasívnom móde a iba zbiera odchytenú komunikáciu. Neútočí tak priamo na komunikáciu za účelom jej zmeny, ale pracuje len ako pozorujúci účastník. Napríklad modul softvéru VPNFilter[12] upravuje IP tabuľky zariadenia za účelom presmerovania komunikácie určenej pre port 80 na svoju lokálnu službu, ktorá odpočúva na porte 8888.
- *Program upload* [T0845], pri ktorom útočník podvrhne škodlivý program cielenému zariadeniu, napríklad pomocou využitia vedľajšej aktívnej služby (sekcia 2.7). Tento program má nasledovne jediný účel, študovať a zbierať dôležité informácie o napadnutom systéme a následne zabezpečiť prístup pre útočníka k zozbieranej kolekcii. Vyššie spomenutý softvér *Triton* 2.2 poskytuje možnosť upravenia *payload*[13], ktoré môže doplniť o nahranie samotného programu.

2.9 Príkaz a kontrola (Command and Control) [TA0101]

Techniky spadajúce pod túto kategóriu spočívajú v zneužití príkazov, ktoré je útočník schopný odoslať na cielené zariadenie, a to využitím štandardizovaných služieb a k nim otvorených portov. Medzi typické techniky spadajú:

- *Commonly used port* [T0885], ktorý vykonáva priamo vyššie spomenuté. Cieľom takto odosielaných príkazov je vyhnutie sa detekcii v samotnom systéme či vyhnutie sa detekcii firewallmi. Tieto príkazy môžu, ale nemusia byť príslušné použitému protokolu

[11] https://collaborate.mitre.org/attackics/index.php/Software/S0010
[12] https://collaborate.mitre.org/attackics/index.php/Software/S0002
[13] Časť prenášaných dát, ktorá predstavuje skutočnú zamýšľanú správu

alebo portu. Reálnym príkladom je ruská skupina Dragonfly[14], ktorá riadiacimi príkazmi cez štandardné porty (445,139,137 a 138) útočila na vládne subjekty a viaceré sektory kritickej infraštruktúry v USA a časti energetického sektora v Turecku a Švajčiarsku.

- *Standard application layer protocol* [T0869], ktorý môže práve použitím podobných príkazov cez štandardné aplikačné protokoly založiť spojenie na cielené zariadenie, ktoré mu ďalej umožní vykonávať príkazy či kontrolovať stav ICS systému z pohľadu kompromitovaného zariadenia. Rovnaké protokoly môžu byť tiež využité na vytvorenie spojenia so vzdialeným miestom mimo lokálnej siete. Skupina Hexane[15], ktorá sa zameriavala na organizácie ICS v sektore ropy a zemného plynu používala na získanie kontroly protokoly HTTP [2] a DNS [9].

2.10 Zabránenie funkčnosti odpovedať (Inhibit Response Function) [TA0107]

Jedná sa o najrozsiahlejšiu kategóriu z pohľadu počtu možných útokov, ktorými útočník môže zabrániť cielenému zariadeniu odpovedať na legitímne dotazy. Veľký počet možných útokov je spôsobený práve rozsiahlymi možnosťami, ktorými je možné tento cieľ dosiahnuť z pohľadu útočníka. Medzi najznámejších predstaviteľov patrí:

- *Denial of service*, ktorým útočník zahlcuje rozhranie či samotný výpočetný výkon cieleného zariadenia. Zahltenie dosahuje odosielaním veľkého počtu neoprávnených dotazov z útočiaceho zariadenia, čím môže na obeti spôsobiť nemožnosť prijatia legitímneho dotazu či môže spomaliť pracovný výkon zariadenia natoľko, že to následne nie je schopné na dotaz odpovedať v určitom časovom intervale. Softvér Backdoor.Oldrea[16] spôsobil občasné zlyhanie viacerých bežných platforiem OPC a odmietnutie služieb aplikácií, ktoré sú závislé na komunikácii OPC.

- *Men in the Middle* [T0830], ktorý tentokrát však už nie v pasívnej forme. Z hľadiska útokov na ICS systémy tiež dostal viaceré názvy podľa cieľa, strany komunikácie či funkcionality, ktorú napáda. Z princípu sa tak napríklad útoky:

  - *Block command message* [T0803], ktorý blokuje príkazové správy/správu,
  - *Block reporting message* [T0804], ktorý zamedzuje prenos ohlasovacích správ,

  líšia iba v konkrétnej správe, ktorú útočník zablokuje. Samozrejme na najnižšej úrovni môžu útoky obsahovať oveľa komplexnejšie rozdieli, avšak z principiálneho hľadiska blokujú dve rozdielne typy prenášaných správ.

  Najjednoduchším príkladom je *Network Allowlists*[17], ktorého podvrhnutím je útočník schopný zamedziť prístup pre pôvodne povolené sieťové prvky, čím zamedzí (zablokuje) príkazové či ohlasovacie správy z požadovaného zariadenia.

[14] https://collaborate.mitre.org/attackics/index.php/Group/G0006
[15] https://collaborate.mitre.org/attackics/index.php/Group/G0005
[16] https://collaborate.mitre.org/attackics/index.php/Software/S0003
[17] https://collaborate.mitre.org/attackics/index.php/Mitigation/M0807

2.11 Párové riadenie procesu (Impair Process Control) [TA0106]

Útoky, ktoré táto kategória pokrýva, usilujú o získanie kontroly nad bežiacimi procesmi. Kontrolou je v tomto prípade myslený prístup k zmene funkcionality zaradení či zmene prenášanej informácie tak, aby bola táto funkcionalita pozmenená. Medzi predstaviteľov kategórie patria napríklad:

- *Unauthorized command message* [T0855], ktorým môže útočník vyvolať zmenu objektu na cieľovom zariadení či vyvolať vykonanie inštrukcie mimo určenej funkcionality zariadenia. Správy *Command* sú v sieťach ICS využívané ako priame pokyny na ovládanie systémových zariadení. Takto podvrhnutý neautorizovaný príkaz môže tiež spôsobiť efekty popísané v sekcii 2.12. Pri incidente v roku 2015 skupina Sandworm[18] vyslala neoprávnené príkazy ističom rozvodní po získaní kontroly nad pracovnými stanicami operátora a získala prístup ku klientskej aplikácii systému riadenia distribúcie (DMS).
- *Modify parameter* [T0836], ktorý je opäť variantou útoku MITM. Jeho účelom je priamo meniť parametre jednotlivých objektov v ohlasovacích správach komunikácie ICS či hodnoty parametrov príkazov posielaných v systéme ICS. Efekt útoku je tu opäť rozmanitý, keďže závisí na veľkosti zmeny a mieste uskutočnenia. Už spomenutý softvér Stuxnet 2.3 je naprogramovaný posielať sieťové zhluky, ktorých údaje v rámcoch sú pokynmi pre pohon frekvenčného meniča. Pohony s frekvenčným meničom pozostávajú z parametrov, do ktorých je možno zapísať nové hodnoty pre zmenu správania zariadenia, a ktoré je možné diaľkovo konfigurovať cez Profibus.

2.12 Vplyv (Impact) [TA0105]

Impact je skôr doplňujúca kategória, ktorá klasifikuje útoky podľa účinku v samotnom systéme. Tieto účinky môžu začínať manipulovaním viditeľných informácií a končiť až na škodách na majetku. Väčšina z útokov v tejto kategórii (napríklad):

- *Denial of control* [T0813],
- *Loss of availability* [T0826],
- *Loss of Control* [T0827],
- *Damage to Property* [T0879],

typicky predstavujú samotný priamy či vedľajší efekt útoku z ľubovolnej kategórie. *Denial of control* spôsobuje znemožnenie kontroly nad ICS systémom. Operátor v takom prípade stratí kontrolu nad vybranými vzdialenými stanicami, čo môže byť napríklad účinkom DOS útoku. Do príkladov sa tak radia už vyššie spomenuté nástroje ako Stuxnet 2.3, Industroyer 2.6 či Ekans 2.5.

[18] https://collaborate.mitre.org/attackics/index.php/Group/G0007

2.13 Zhrnutie

V kapitole je priblížené kategorické rozdelenie útokov na ICS/SCADA systémy. Každá kategória je špecifická určitým typom efektu, ktorý je pri každej v skratke popísaný. Tiež sa v každej kategórii nachádzajú v skrátenej verzii popísané typické útoky do nej spadajúce, čo pomôže pre ujasnenie rozdielov jednotlivých kategórii. Toto delenie mi v ďalšej kapitole pomôže k zúženiu výberu jednotlivých útokov, ktoré sa skutočne priamo či okrajovo týkajú riadiacej komunikácie v systémoch ICS/SCADA. Pre voľbu útokov sa ako najzaujímavejšie kategórie javia: Discovery, Collection, Inhibit response function a Impair process control.

Útok	Kategória	Odhalenie z komunikácie ICS protokolu
Remote Services	Initial Access	Nie
Wireless Compromise		Nie
Change operating mode	Execution Evasion	Nie
Masquerading	Evasion	Áno
Network connection enumeration	Discovery	Nie
Network sniffing	Discovery Collection	Áno
Exploitation of remote services	Lateral Movement	Nie
Men in the Middle	Collection Inhibit Response Function Impair Process Control	Áno
Program upload	Collection	Áno
Denial of service	Inhibit Response Function	Áno
Block command message		Áno
Block report message		Áno
Unauthorized command message	Impair Process Control	Áno
Modify parameter		Áno
Denial of control	Impact	Áno
Loss of availability		Áno

Tabuľka 2.1: Prehľad útokov vhodných na monitorovanie v sieti.

Kapitola 3

Výber útokov pomocou rešerše datasetov

V kapitole sa zameriam na výber vhodných útokov na komunikáciu protokolu IEC 104. Na výber jednotlivých aplikovateľných typov útokov popísaných v kapitole 2 som použil poskytnuté datasety, ktoré obsahujú dáta a krátky popis pre útoky prevedené na protokoly DNP3 [18] a Modbus [1]. Pri selekcii som sa však tiež musel prispôsobiť možnostiam použitých virtuálnych zariadení, samotného protokolu IEC 104 a samozrejme relevantnosti útoku pre komunikáciu ako takú. Preskúmanie datasetov mi však pomohlo pri výbere útokov vhodných na emuláciu a vytvorenie datasetu pre detekciu anomálií. Útoky spracované v datasetoch názvami neodpovedajú kategorizovaným útokom popisovaným v databáze MITRE. Z ich popisuje je však možné odvodiť ich zaradenie, čo prispeje k sumarizácií bežne spracúvaných útokov v ICS/SCADA sieťach. Výstupom je tak zúžená množina útokov, ktoré by mohli byť emulovateľné v ďalšej kapitole mojej práce.

Útoky z kategórií *počiatočný prístup*, *vykonanie (execution)* a *eskalácia privilégii* sú zamerané na získanie prístupu do siete či využitie rôznych služieb aktívnych na zariadeniach. Tieto útoky sa tak netýkajú priamo riadiacej komunikácie v SCADA systémoch, ale zneužívajú vedľajšie služby, ktoré by mohli byť na niektorých zariadeniach aktívne. Využitie týchto útokov na virtuálne zariadenia simulujúce protokol IEC 104, ktoré som pri práci použil, by tak nepatrili k relevantným, pretože by skôr využili zraniteľnosti na strane fyzického zariadenia použitého na ich virtualizáciu.

Kategórie *pretrvanie* a *postranný prístup* pokrývajú útoky, ktoré sú používané na podvrhnutie programového obsahu, využitie predvolených prihlasovacích údajov či získanie vzdialeného prístupu k zariadeniam opäť nepatria k útokom na protokol ako taký, a zároveň využívajú služby, ktoré samotné virtuálne zariadenia nepodporujú, resp. ich nepodporuje protokol IEC 104.

Pri výbere útokov z ostatných kategórií som sa inšpiroval práve poskytnutými datasetmi. Všeobecne sa majorita útokov obsiahnutá v skúmaných datasetoch týka práve útokov z posledných štyroch kategórií, keďže práve tie primárne útočia na riadiacu komunikáciu ako takú, a to bez využitia vedľajších služieb. Medzi hlavné datasety, ktoré som skúmal patria:

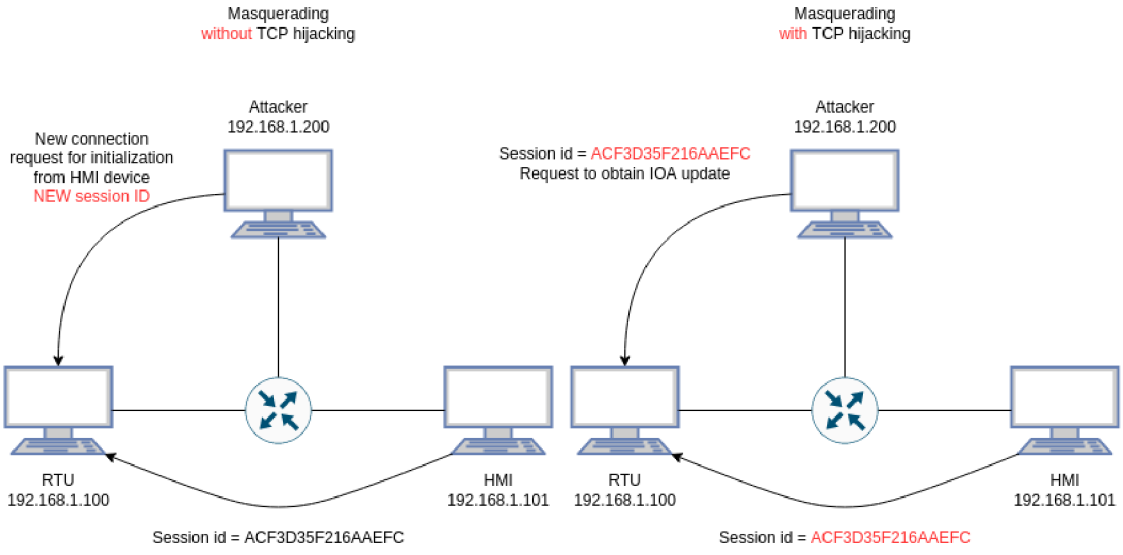
- Real-Time and Interactive Attacks on DNP3 dataset [16],
- Cyber-security Modbus ICS dataset [3],
- Industrial Control System Traffic Data Sets for Intrusion Detection Research [10].

3.1 Vybrané útoky z kategórie Discovery

Útoky z tejto kategórie som vybral z vlastnej iniciatívy a nie z rešerše jednotlivých datasetov. Tieto útoky totiž taktiež neútočia na komunikáciu protokolu IEC 104, no nepriamo s ňou súvisia či sú súčasťou efektu niektorých z útokov. Z tejto kategórie som vybral dva útoky:

- *Enumerácia sieťových pripojení* [T0840] je typický útok na štýl objavenia a získania informácií o zariadeniach, no pre moje účely hlavne službách resp. protokoloch, ktorými konkrétne zariadenia v sieti komunikujú. Pomocou tohto útoku je tak útočník schopný odhadnúť prípadný cieľ ďalšieho útoku či prípadne zistiť konkrétnu úlohu zariadenia v sieti. Útočník na to môže použiť ľubovoľné voľne dostupné nástroje ako napríklad *netstat*[1].
- Odpočúvanie prevádzky na sieti [T0842], ktorým môže útočník zachytávať správy prenášané v komunikácii SCADA systémov. Odchytené správy obsahujú nešifrované hodnoty atribútov jednotlivých staníc, ktoré je tak schopný sledovať a odhadnúť aktuálny stav systému počas bežnej prevádzky, či identifikáciu dôležitých udalostí, ktoré v systéme môžu nastať. Takto odchytené správy je tiež možné použiť pre bližšiu identifikáciu funkcií jednotlivých zariadení v sieti. Tieto informácie môžu pre útočníka slúžiť na deterministický výber cieľového zariadenia, na ktoré by mohol chcieť zaútočiť, či na voľbu správneho typu alebo scenára útoku pre jednotlivé zariadenia.

3.2 Maskovanie (Masquerading) [T0849] - kategória Evasion



Obr. 3.1: Princíp verzií útoku Masquerading

Podstata útoku spočíva v podvrhnutí falošného zariadenia, ktoré je schopné korektne komunikovať s riadiacimi zariadeniami v ICS/SCADA sieti. Efekty takéhoto útoku sú rozmanité, keďže nie je priamo špecifikované, o akú výmenu správ sa pri útoku jedná. Vo všeobecnosti sa dá útok viesť dvoma spôsobmi.

[1] https://man7.org/linux/man-pages/man8/netstat.8.html

1. Samotný útok tak môže byť vedený z nového podvrhnutého zariadenia, ktoré má prístup do siete (obrázok 3.1 - *New session*). V tomto prípade sa útočník pokúsi nadviazať nové spojenie s cieľovým zariadením a korektnými správami s ním komunikovať, čo sa dá bližšie špecifikovať ako podvrhnutie zariadenia ako takého.

2. Druhý prípad spočíva vo využití metódy *TCP-hijacking*, pri ktorej útočník získa prístup k aktívnej relácii medzi dvoma riadiacimi zariadeniami (obrázok 3.1 - *Session id = ACF3D35F216AAEFC*). Pri tomto prípade sa tak útočník môže vydávať za už existujúce zariadenie v sieti a v jeho mene odosielať podvrhnuté správy na cieľové zariadenie.

Attack 2 Masquerading
2.1 Connect to the slave and masquerade as a master device
2.2 Use TCP hijacking to steal the existing service instance connection from the target master
2.3 Masquerade as the target slave and accept the connection from the target master

Obr. 3.2: Prehľad masquerading útokov z datasetu venovanému protokolu DNP3 [16]

Z referovaných datasetov sa oň pokúšal práve *Industrial Control System (ICS) Cyber Attack Datasets* v oboch formách. Na obrázku 3.2 sa nachádzajú vybrané útoky, z ktorých útok 2.1 popisuje prvú variantu vytvorenia nového spojenia s podvrhnutou základňou a útoky 2.2 a 2.3 zneužívajú aktívnu reláciu medzi korektnými zariadeniami.

3.3 Man in the middle [T0830] - kategória Collection

Ako som už spomenul v sekcii 2.8, útoky spadajúce do tejto kategórie sa venujú práve zbieraniu informácií o požadovaných zariadeniach v ICS/SCADA systéme. Hlavným dôvodom (resp. útokom) tejto kategórie je bezpochyby útok *Men in the Middle*. Tento útok je prakticky vykonávaný vo všetkých vyššie spomenutých datasetoch, takže jeho výber bol pre mňa samozrejmosťou.

Efekty a účel útoku sú však rozsiahle, a tak je v rámci kategorizácie spomenutej v kapitole 2 rozdelený na niekoľko rozdielnych útokov spadajúcich pod rôzne kategórie. Jeho všeobecnú verziu tak spomínam hlavne preto, že žiadny z referovaných datasetov nepoužíval jeho špecifickejšie rozdelenie. Ako príklad môže poslúžiť rozdelenie variant MITM útoku zo skúmaného datasetu na obrázku 3.3. Samotné varianty z obrázku pre útok MITM tak pokrývajú útoky:

- Network sniffing [T0842] - zachytenie komunikácie (5.1)
- Block reporting/command message [T0804]/[T0803]- prerušenie komunikácie doplnené o *Injection útok* (5.2),
- Modify parameter [T0836] - zmena parametrov objektov (5.3-9).

Attack 5 MITM	
5.1	Intercept and read all DNP3 communication between the targets
5.2	Intercept DNP3 messages and inject DNP3 communication between the targets
5.3	Intercept DNP3 messages and update function code
5.4	Intercept DNP3 messages and update binary status object
5.6	Intercept DNP3 messages and update counter object data point
5.7	Intercept DNP3 messages and delete binary object
5.8	Intercept DNP3 messages and delete binary object data point
5.9	Intercept DNP3 messages and insert binary object data point

Obr. 3.3: Prehľad MITM útokov z datasetu venovanému protokolu DNP3 [16]

3.4 Vhodné útoky z kategórie Inhibit Response Function

Výber útokov z tejto kategórie bol hlavne pre možnosti virtuálnych zariadení pre mňa pomerne jasný. Zariadenia totiž nepodporujú funkcionalitu vzdialeného reštartu či zastavenia činnosti na diaľku. Pre doplnenie sa však aj mňou vybrané útoky objavili v skúmaných datasetoch. Vybrané útoky tejto kategórie sú:

- Block Command Message [T0803],
- Block Reporting Message [T0804],
- Denial of Service [T0814].

Útok *Denial of Service* je typickým predstaviteľom tejto kategórie a jeho najzákladnejšie implementácie spočívajú v záplave zariadenia pomocou veľkého počtu paketov *SYN* protokolu TCP [14], či *Echo requestov* protokolu ICMP [12]. Jeho priamym efektom však nie je zmena či narušenie samotnej riadiacej komunikácie, ale zahltenie cieľového zariadenia, ktoré tak ideálne nebude schopné odpovedať na autorizované požiadavky ostatných zariadení v sieti. Nepriamo tak s riadiacou komunikáciou súvisí, pričom jeho efektom je destabilizovanie bežnej prevádzky z alebo do cieľového zariadenia. Útok je tak vhodným kandidátom na emuláciu v mojej práci. V skúmaných datasetoch bol v majoritnej miere používaný práve v jeho spomenutej základnej forme (datasety [3] a [10]). Jedine v datasete [16] vytvorenému pomocou frameworku na protokol DNP3 sú popísané ďalšie varianty tohto útoku, ktoré spočívajú v záplave korektnými dotazmi protokolu DNP3.

3.5 Modify parameter [T0836] - kategória Impair Process Control

Útoky, ktoré pokrýva táto kategória, primárne vychádzajú z štandardného sieťového útoku MITM. Do kategórie tiež spadá útok priamo na firmware zariadenia, či útok hrubou silou. Pri použitých virtuálnych zariadeniach však napadnutie firmwaru pre mňa neprichádza do

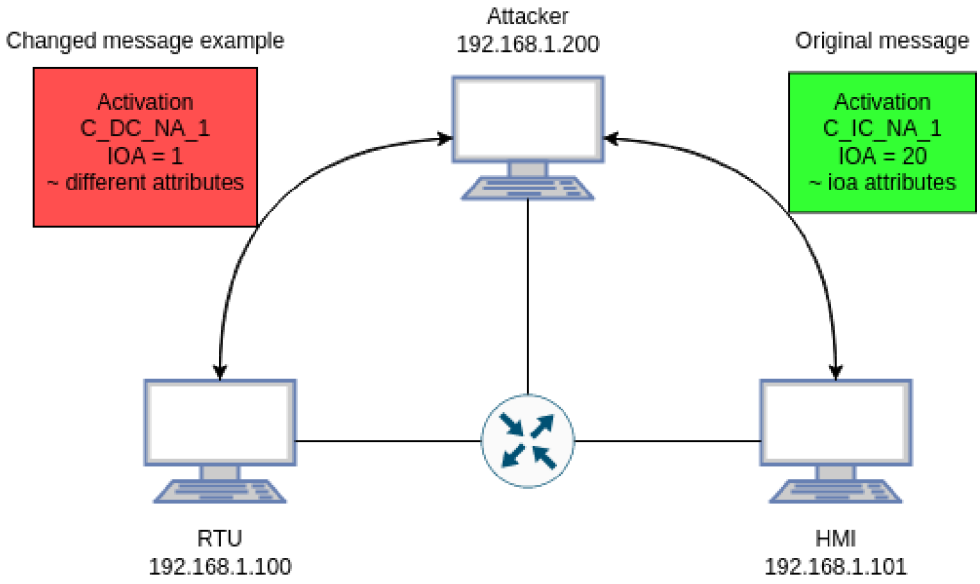
úvahy. Rovnako tak pre útok hrubou silou na komunikáciu protokolu IEC 104 neexistuje korektná varianta, keďže z princípu neexistuje účel pre odhadovanie akýchkoľvek prenášaných parametrov. Všetky odchytené správy protokolu sú v nešifrovanej forme, a teda útočník v strede komunikácie je priamo schopný ich voľne čítať. Okrem nich sem tiež spadá útok na odchytávanie a blokáciu správ, ktoré som už primárne spomenul v časti 3.3 a 3.4.

Najobšírnejším predstaviteľom je práve útok *Modify parameter*. V každom zo študovaných datasetov bol v nejakej forme spracovaný, pričom vždy predstavoval väčšiu časť scenárov útoku MITM (*Modify parameter* nikdy nebol spomenutý samostatne). Ako príklad je na obrázku 3.4 možno vidieť krátky popis týchto scenárov z datasetu útokov na protokol DNP3, pričom okrem prvého sa všetky dajú zaradiť pod práve spomínanú zmenu parametru.

Attack 5 MITM
Intercept and read all DNP3 communication between the targets
Intercept DNP3 messages and update binary object data point
Intercept DNP3 messages and update counter object data point
Intercept DNP3 messages and delete binary object
Intercept DNP3 messages and delete binary object data point

Obr. 3.4: Krátky popis MITM útoku z datasetu pre protokol DNP3

Parameter je totiž v kontexte viacerých protokolov príliš relevantný pojem. Pre protokol IEC 104 sa dá za parameter prenášanej správy považovať ktorákoľvek z hodnôt v originálnej správy z obrázku 3.5. Útočník tak môže zmeniť obyčajný dotaz pre získanie aktuálnych hodnôt objektov na zmenu hodnoty konkrétneho z nich, čo je porovnateľné so zmenou paketu protokolu DNP3 pri útoku pre odstránenie binárneho objektu. Oveľa jednoduchšie porovnanie je však zmena IOA atribútov (protokol IEC 104) voči zmene binárneho objektu (protokol DNP3).



Obr. 3.5: Príklad možných zmien v správe

3.6 Nekategorizované útoky z datasetov

V tejto časti sa budem v skratke venovať útokom, ktoré nie sú priamo zaradené v kategóriách MITRE. Jedná sa o útoky, ktoré nemali priame určenie efektu na priemyselnú sieť SCADA, alebo majú podobný účinok ako samotná úprava parametrov spomenutá vyššie. V kategóriách MITRE nezastávajú vlastné miesto z rovnakého dôvodu. Ich použitie môže byť súčasťou ľubovoľného útoku s konkrétne určeným efektom, ktorý samostatne priamo určený nemajú. K ich zahrnutiu do môjho výberu ma viedol práve fakt, že sa často vyskytovali v skúmaných datasetoch ako samostatne spracované útoky.

Z nekategorizovaných útokov som vybral práve *Replay* a *Injection*. Oba sa objavili naraz alebo aspoň na striedačku vo všetkých skúmaných datasetoch, pričom by mali byť prevediteľné aj na komunikáciu protokolu IEC 104. Na obrázku 3.6 je možné vidieť skrátený popis oboch útokov prevedených v datasete pre protokol DNP3. Z ich popisu je tiež možno vidieť, že samotný replay útok je súčasťou injection útoku na zariadenie vedúce komunikáciu, čo predstavuje podobné kríženie útokov pre docielenie požadovaného efektu, ktoré popisuje samotné delenie v kategóriách MITRE.

Attack 1 Injection Attacks
1.1 Replay a previously collected message into the master via injection
1.2 Replay a previously collected message into the master via injection and acknowledge any response from target
1.3 Inject a malicious command into the slave (Freeze Objects)
Attack 3 Replay
3.1 Replay previously captured DNP3 messages to the slave

Obr. 3.6: Nekategorizované útoky z datasetov

3.6.1 Injekčný útok (Injection)

Efekty útokov Injection, ktoré sú spracované v skúmaných datasetoch, sú blízko spojené s efektmi útoku MITM, ktorého boli aj často súčasťou. Vo veľkej časti prípadov tiež predstavovali len mierne odlišný spôsob útoku *Úprava parametrov*, kedy útočník vygeneruje mierne zmenenú odchytenú správu, ktorou doplní bežnú prevádzku. Z tohto pohľadu by sa pre moju prácu dal tento útok zaradiť pod MITRE kategóriu *Impair Process Control* 2.11. Rovnako tak jeho použitie na komunikáciu protokolu IEC 104 by malo za následok úpravu hodnôt IOA objektov na zariadení RTU.

Pri použitých virtuálnych zariadeniach som však narazil na problém pri jeho vykonaní. Zariadenia totiž striktne kontrolujú hodnoty *RX* a *TX*, ktorých zmenu by útok musel vykonávať v každom ďalšom prenášanom pakete komunikácie. Predstavovali pre vykonávané útoky oveľa väčší problém, ako som čakal, a bez neustálej zmeny tychto hodnôt v každom ďalšom pakete komunikácie virtuálne zariadenia padajú z dôvodu internej ochrannej implementácie. Pre túto komplikáciu som tak nakoniec tento útok z vykonávaných vyradil, keďže som neprišiel na rozumnú verziu riešenia tochto problému. Dovolil som si ho však spomenúť hlavne z dôvodu, že jeho využitie je pomerne obšírne a mohlo by byť pri komunikácii iných protokolov užitočné.

3.6.2 Opakovanie (Replay)

Efekt Replay útoku v skúmaných datasetoch nebol nijak závažný. Vo všetkých sa autori pokúsili o odoslanie odchyteného paketu v malom množstve opakovaní. Typicky sa jednalo o jedno zopakovanie pre útok cieľového paketu. Efekt na zariadenia či samotnú komunikáciu tak nikdy nebol majoritný a vo všetkých datasetoch tak následne sledovali iba jeho duplikovaný odchyt na zariadeniach kam smeroval.

Z pohľadu teoretického efektu by som však opäť útok zaradil do kategórie *Impair Process Control* 2.11, keďže v prípade v prípade správy požadujúcej zvýšenie či zníženie hodnoty (príkaz bez konkrétnej hodnoty), by útok replay opäť predstavoval určitú formu úpravy parametrov na zariadení RTU, čo by znamenalo úpravu hodnôt prenášaných v komunikácii v cyklických dotazoch. V použitej štandardnej komunikácii protokolu IEC 104 sa však takýto príkaz nenachádza, a rovnako tak už teraz viem, že nebude možné zopakovať odchytený paket veľakrát, keďže v prípade jeho obdržania zariadením po určitom časovom úseku by virtuálne zariadenia opäť padali pre nekonzistentné *RX* a *TX* hodnoty. Jeho výsledný efekt pri emulácii tak tiež nebude predstavovať majoritný zásah do samotnej komunikácie.

3.7 Zhrnutie

V kapitole som sa venoval výberu vhodných útokov, ktoré v prvom rade spĺňajú aspoň nepriamy súvis s útokmi na riadiacu komunikáciu ako takú. Vynechal som útoky využívajúce vedľajšie služby, ktoré vo väčšej miere závisia na samotnom hardware a nastavení zariadenia ICS/SCAD systému. Tieto útoky by totiž nepriniesli relevantnejšie výsledky k mojej práci, pretože na emuláciu používam virtualizované zariadenia komunikujúce protokolom IEC 104. V realite totiž zariadenia bežia nad hardvérom zariadení Raspberi-Pi, ktorý neodpovedá hardvéru reálnych zariadení. Pri výbere som si dopomohol dostupnými datasetmi, ktoré mi priblížili jednotlivé typy útokov a ich možnosti v rámci riadiacich protokolov ICS/SCADA systémov. Tieto poznatky sú v kapitole zaznamenané a ako ďalšie sa budem venovať samotnej emulácii a popisu útokov na virtuálnych zariadeniach.

Útok	Výskyt		
	Dataset [16]	Dataset [3]	Dataset [10]
Network Connection Enumeration	✗	✗	✗
Network Sniffing	✓	✓	✓
Masquerading	✓	✗	✗
Block reporting message	✓	✓	✗
Modify parameter	✓	✓	✓
Denial of Service	✓	✓	✗
Injection	✓	✗	✓
Replay	✓	✗	✓

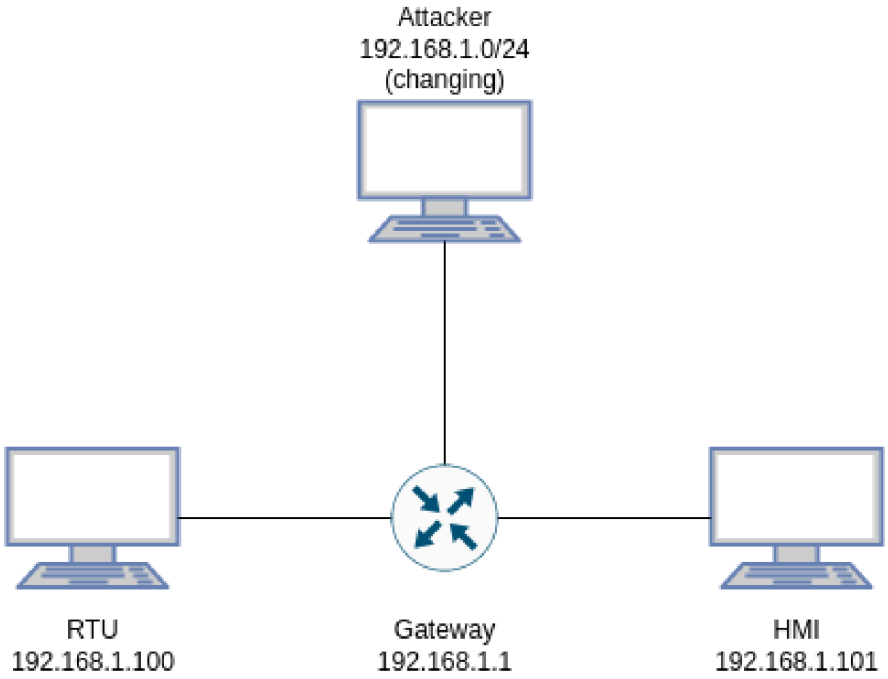
Tabuľka 3.1: Prehľad útokov podľa výskytu v datasetoch

Kapitola 4

Útoky na riadiaci systém IEC 104

Kapitola je zameraná na presný popis jednotlivých emulovaných útokov na riadiacu komunikáciu protokolu IEC 104. Dôvodom ich popisu je následná emulácia na virtualizovaných zariadeniach a vytvorenie datasetu odchytených komunikácií pod útokom. Vytvorený dataset následne podrobím analýze a v poslednej časti práce ich použijem pre vyhodnotenie detekcie pomocou implementovaných štatistických metód.

Pre vykonanie útokov som použil virtuálne zariadenia z upravenej verzie ICS TestBed frameworku [8], ktorá je dôkladnejšie popísaná v technickom reporte [7]. Framework poskytuje možnosti pre vytvorenie virtuálnych zariadení, ktoré komunikujú protokolom IEC 104. V upravenej verzii je mnou vytvorená databáza jednotlivých *Information Object Address* objektov a ich hodnôt, ktorá umožňuje zariadeniam reagovať aj na základe predošlej komunikácie a nie len na aktuálne obdržanú správu ako tomu je v základnej verzii.



Obr. 4.1: Sieťové zapojenie virtuálnych zariadení

Všetky útoky spomenuté nižšie v tejto kapitole som vykonal na zapojení zariadení z obrázka 4.1. Pre zapojenie som použil dva základné typy zariadení, ktoré framwork poskytuje:

- Human machine resource (HMI) - zariadenie riadiace prebiehajúcu komunikáciu,
- Remote terminal unit (RTU) - zariadenie poskytujúce dáta (stavy) objektov.

Komunikácia tak prebieha formou dotaz-odpoveď, pričom úlohou HMI zariadenia je riadenie komunikácie formou dotazov na jednotlivé objekty, ktoré spravuje RTU stanica a je tak schopná poskytovať o nich informácie, či na základe korektného dotazu upraviť atribúty jednotlivých objektov. Už spomenuté objekty sú z reality informácie z fyzických senzorov, spínačov a podobných zariadení, ktorých chovanie môže byť podmienené udalosťou na fyzickom zariadení, na ktoré sú napojené. Framework je však zameraný na komunikáciu medzi riadiacimi prvkami, a teda akékoľvek zmeny na objektoch môžu byť vyvolané jedine dotazom z HMI zariadenia, čo je pre štandardnú komunikáciu a útoky na ňu dostatočné.

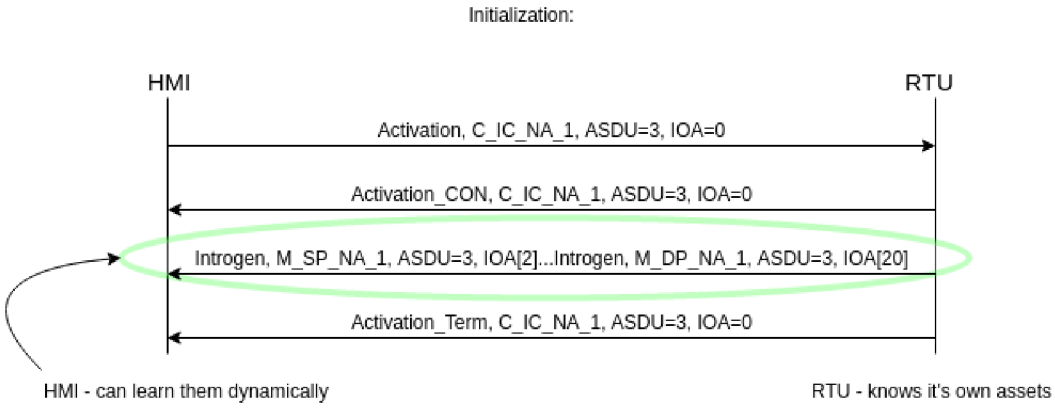
4.1 Štandardná komunikácia

Jedná sa o komunikáciu medzi zariadeniami v čase kedy nedochádza k žiadnemu útoku. Zariadenia medzi sebou cyklicky komunikujú bez vstupu útočníka. Nepatrí tak pod útoky ako také, avšak poslúži ako trénovacia sada pre detekciu anomálií v poslednej časti práce.

Štandardná komunikácia, na ktorú som vykonal útoky spomenuté nižšie (resp. komunikácia medzi zariadeniami HMI a RTU) sa dá rozdeliť na tri základné časti:

- inicializácia,
- cyklická komunikácia,
- dotazy na úpravu.

V inicializačnej fáze zobrazenej na obrázku 4.2 sa tak stanica HMI ako iniciátor dotazuje o informácie z RTU stanice o poskytovaných objektoch. Aj keď rovnaký dotaz (C_IC_NA_1) je následne použitý a prakticky kompletne tvorí cyklickú časť komunikácie tak v tejto fáze sa stanica HMI prvým dotazom naučí, ktoré IOA objekty a informácie o nich mu RTU stanica poskytuje.



Obr. 4.2: Inicializačná/cyklická časť komunikácie

Cyklickú komunikáciu kompletne tvorí dotaz z obrázka 4.2, avšak v tejto fáze už stanica HMI pozná poskytované IOA objekty a opakovanými dotazmi po určitom časovom intervale len obnovuje informácie o nich (o ich vnútornom stave). Stanica HMI je však počas cyklickej komunikácie schopná reagovať negatívne v prípade, že by obdržala informácie o IOA objekte, ktorý neexistoval v momente inicializácie.

```
41 10.029175444  192.168.1.10   192.168.1.12   104asdu   82 <- I (2,31) ASDU=10 C_SC_NA_1 Act    IOA=2
47 10.033351119  192.168.1.10   192.168.1.12   104apci   72 <- S (34)
51 20.029494077  192.168.1.10   192.168.1.12   104asdu   82 <- I (3,34) ASDU=10 C_SC_NA_1 Act    IOA=13
57 20.039124781  192.168.1.10   192.168.1.12   104apci   72 <- S (37)
59 30.031989301  192.168.1.10   192.168.1.12   104asdu   82 <- I (4,37) ASDU=10 C_DC_NA_1 Act    IOA=1
70 30.081254621  192.168.1.10   192.168.1.12   104apci   72 <- S (40)
73 40.032341905  192.168.1.10   192.168.1.12   104asdu   82 <- I (5,40) ASDU=10 C_DC_NA_1 Act    IOA=14
```

Obr. 4.3: Príklad sekvencie HMI dotazov

Posledná časť je tvorená sekvenciou dotazov od stanice HMI, ktorých úlohou je zmeniť atribúty jednotlivých IOA objektov (obrázok 4.3). Tieto dotazy môžu byť vyvolané napr. operátorom, ktorý sa prostredníctvom stanice HMI snaží o zmenu na dotazovaných objektoch. V modelovanej štandardnej komunikácii je táto sekvencia však opäť cyklická a predpokladá stav kedy raz za hodinu dôjde k obnoveniu predvolených hodnôt nastavených dotazmi zo stanice HMI po inicializačnej fáze.

4.2 Enumerácia sieťových pripojení [T0840]

Ako som už vyššie spomínal, tento útok nespadá pod útoky smerujúce priamo na prenášanú komunikáciu, ale má za cieľ zozbierať úvodné informácie o komunikujúcich zariadeniach. Na jeho vykonanie som použil štandardné nástroje ako *arp-scan*[1], ktorý využíva protokol ARP [11] na objavenie dostupných zariadení a *nmap*[2], ktorý naopak zisťuje informácie o konkrétnom zariadení, či zariadeniach. Ich použitím je útočník schopný dostať sa k základným informáciám o dostupných zariadeniach v sieti a vybrať si predbežný cieľ útoku.

Na obrázku 4.4 sa nachádza výstup vykonaného skenu pomocou protokolu ARP. Vo výstupe možno vidieť objavené zariadenia, ktoré odpovedajú topológii na obrázku 4.1. Okrem toho je možno vidieť základný názov (resp. výrobca) dostupných zariadení spolu s ich IP [13] a MAC [17] adresami.

```
peter@peter-Lenovo-ideapad-320-15IKB:~$ sudo arp-scan 192.168.1.1-192.168.1.255
Interface: wlp3s0, datalink type: EN10MB (Ethernet)
Starting arp-scan 1.9 with 255 hosts (http://www.nta-monitor.com/tools/arp-scan/)
192.168.1.1     f8:1a:67:63:d1:46       TP-LINK TECHNOLOGIES CO., LTD.
192.168.1.100   b8:27:eb:02:b1:7e       Raspberry Pi Foundation
192.168.1.101   b8:27:eb:13:93:8e       Raspberry Pi Foundation

3 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9: 255 hosts scanned in 3.121 seconds (81.70 hosts/sec). 3 responded
```

Obr. 4.4: Vykonaný ARP sken v lokálnej sieti virtuálnych zariadení

Následným skenom pomocou utility NMAP je útočník schopný zistiť samotné aktívne služby na zariadení. Obrázok 4.5 obsahuje náhľad na sken vykonaný priamo na port *2404*. Tento port je totiž prednastaveným portom vo virtuálnych zariadeniach pre príjmanie správ. Kompletný arp sken pre zariadenie sa nachádza v prílohe (obrázok D.1) hlavne z dôvodu

[1] https://linux.die.net/man/1/arp-scan
[2] https://linux.die.net/man/1/nmap

nekompaktnosti. Z neho je možno vidieť, že zariadenie poskytuje viacero služieb, ktoré sú však menej relevantné pre moje útoky. Otvorené porty totiž odpovedajú štandardným portom využívaným zariadeniami *Raspberry-Pi*[3], ktoré som používal na virtualizáciu zariadení komunikujúcich protokolom IEC 104. Aj keď by tieto otvorené služby mohli otvoriť dvere ďalším útokom mimo kategórií, ktoré som pre prácu vybral, tak by ich využitie však nemuselo odpovedať možnostiam útokov na reálne zariadenia.

```
peter@peter-Lenovo-ideapad-320-15IKB:~$ nmap -A -T4 192.168.1.101 -p 2404

Starting Nmap 7.60 ( https://nmap.org ) at 2021-11-07 12:43 CET
Nmap scan report for 192.168.1.101
Host is up (0.011s latency).

PORT     STATE  SERVICE VERSION
2404/tcp closed iec-104

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.71 seconds
peter@peter-Lenovo-ideapad-320-15IKB:~$ nmap -A -T4 192.168.1.100 -p 2404

Starting Nmap 7.60 ( https://nmap.org ) at 2021-11-07 12:43 CET
Nmap scan report for 192.168.1.100
Host is up (0.0071s latency).

PORT     STATE SERVICE  VERSION
2404/tcp open  iec-104?

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.12 seconds
```

Obr. 4.5: Vykonaný NMAP sken portu 2404 v lokálnej sieti virtuálnych zariadení

Vhodný nasledujúci útok je tak samotné odpočúvanie prevádzky na zvolených zariadeniach, pretože útočník zatiaľ môže len vedieť určiť relevantné cieľové zariadenia. Stále tak potrebuje zistiť ich jednotlivé role v riadiacej komunikácii spolu s objektmi, či používanými typmi dotazov a odpovedí.

4.3 Odpočúvanie prevádzky na sieti [T0842]

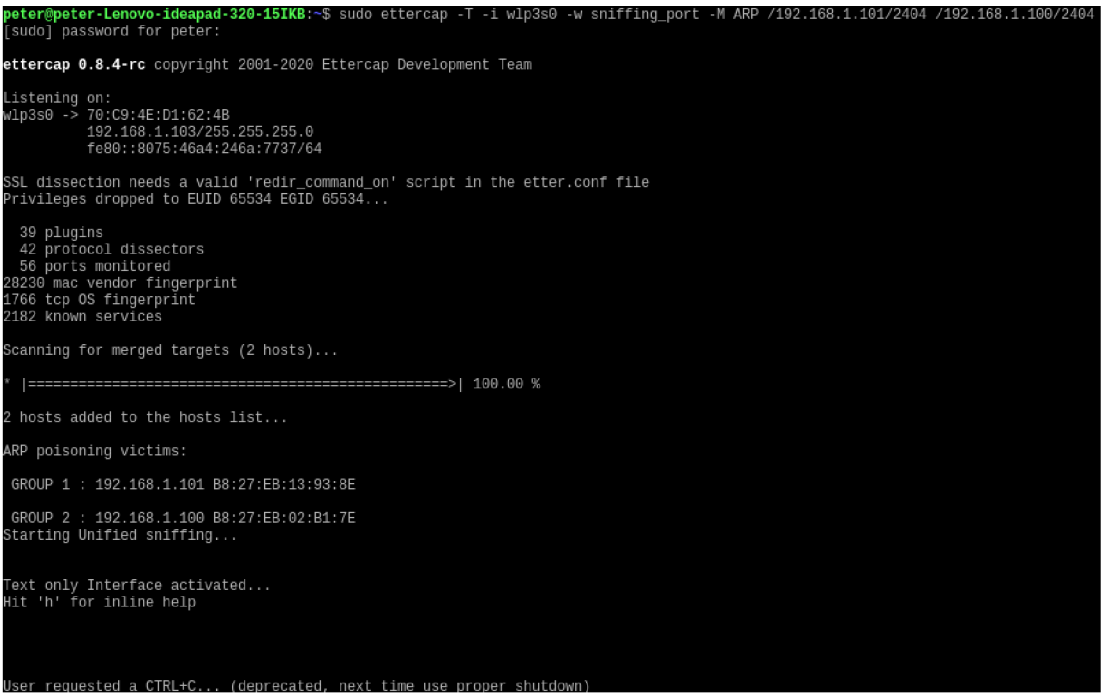
Hlavný účel odpočúvania prevádzky je pre útočníka opäť zbieranie informácií o komunikujúcich zariadeniach v sieti. Pomocou predchádzajúcej enumerácie zariadení si útočník vybral potencionálne cieľové zariadenia, pričom teraz potrebuje zistiť ich role v rámci riadiacej komunikácie.

Na samotný útok (ale aj nasledujúce) som využil program *Ettercap*[4]. Program umožňuje využitie základných otváracích útokov v spojení s dodatočnými možnosťami. V práci som neskôr využil možnosť vytvorenia vlastných pluginov pre vytvorenie jednotlivých útokov. Ich využitie programom je popísané v prílohe A. Tento útok však nevyžaduje žiadne špeciálne súčasti pre samotný Ettercap.

Na obrázku 4.6 sa nachádza jednoduchý príklad využitia program na odpočúvanie prevádzky medzi dvomi zariadeniami na konkrétnom porte (2404). Na pozadí program používa podvrhnuté odpovede pre ARP protokol, ktorý nanúti zariadeniam v sieti zmenu namapovaných MAC adries. Vo výsledku sú tak správy určené zariadeniam HMI a RTU preposlané na zariadenie útočníka. Odpočúvanie prevádzky je tak teoreticky len dôsledkom útoku *arp-spoof*, ktorý Ettercap vykoná v základnom režime. Tento otvárací útok budem aj

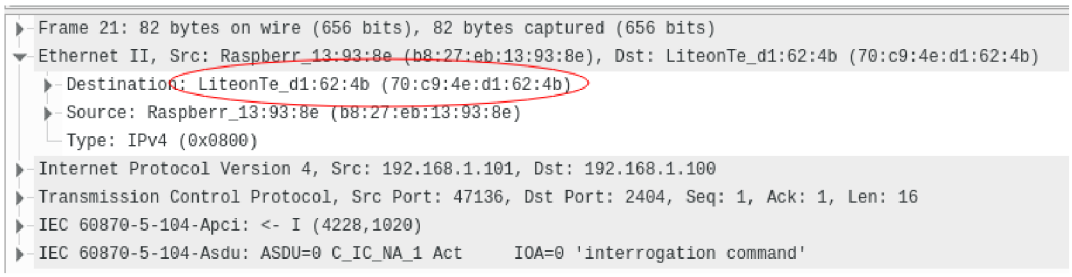
[3] https://www.raspberrypi.org/
[4] https://www.ettercap-project.org/

ďalej využívať v ďalších útokoch vykonaných cez Ettercap v spojení s doprogramovanými pluginmi.



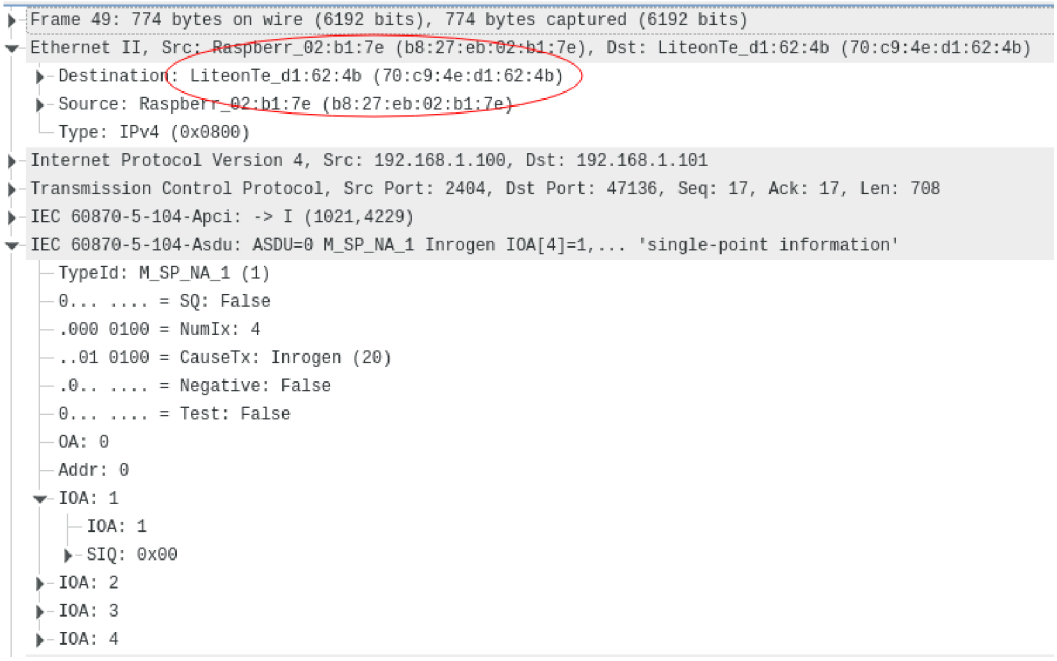
Obr. 4.6: Príklad využitia Ettercapu na odpočúvanie prevádzky

Ako som už spomenul, tak efektom útoku je presmerovanie komunikácie na zariadenie útočníka, ktorý však komunikáciu nijak nemení, ale snaží sa zozbierať z nej podstatné informácie. Na obrázku 4.7 je možné vidieť efekt útoku v zmene MAC adresy cieľového zariadenia na útočníkovu. Najpodstatnejšia informácia získaná útokom je práve pridelenie role z obsahu takto prijatých správ. V komunikácii protokolu IEC 104 je práve stanica HMI iniciátorom spojenia, a tak je z takto odchytenej správy jasné, že úlohu stanice HMI v sieti plní Raspberry-Pi zariadenie s IP adresou 192.168.1.101.



Obr. 4.7: Útočníkom prijatá správa z HMI zariadenia

Rovnaký efekt má útok na správy smerované zo zariadenia RTU (obrázok 4.8). Útočník je tak schopný určiť role oboch komunikujúcich zariadení v sieti. Okrem toho vie z tejto správy určiť aj hodnoty prenášaných objektov, či platných operácii s nimi. V podstate je takto schopný sledovať kompletnú štandardnú komunikáciu popísanú v sekcii 4.1.



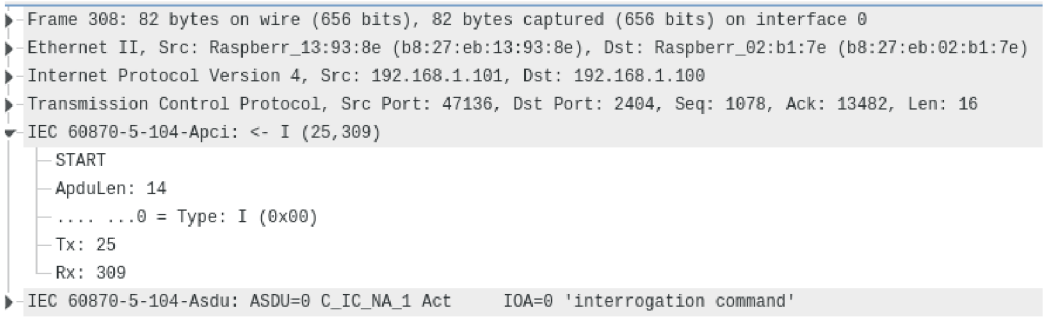
Obr. 4.8: Útočníkom prijatá správa zo zariadenia RTU

Takto odchytená komunikácia otvára pre útočníka možnosť upraviť či blokovať jednotlivé správy podľa ním požadovaného efektu na celkovú komunikáciu. Okrem toho je tiež schopný sledovať časové intervaly jednotlivých správ, pomocou ktorých môže špecifikovať ideálny čas útoku na cieľové zariadenie. V realite totiž môže cyklická forma požiadaviek z HMI zariadenia znamenať interakciu operátora s ovládačom, ktorá môže byť špecifická iba v určitom časovom rozmedzí. Útočník môže na ďalšie útoky zvoliť čas mimo prevádzkovú dobu operátora a zvýšiť tak svoje šance na časovo rozsiahlejší efekt samotného útoku.

4.4 Blokácia hlasovacích správ [T0804]

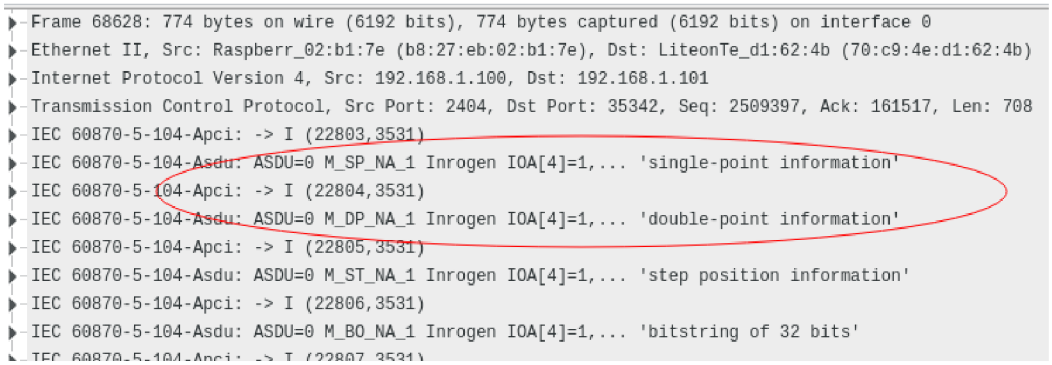
V ponímaní protokolu IEC 104 sa za ohlasovacie správy považujú práve odpovede smerované na HMI zariadenie na ohlásenie stavu objektov obsluhovaných zariadením RTU. Môže sa tak jednať o správu z obrázka 4.8, ale tiež o správu ohlasujúcu vykonanie príkazu na zmenu smerovaného zo zariadenia HMI. Z princípu tak efektom tohto útoku je zamedziť doručenie tejto správy na HMI zariadenie, čím mu útočník odoprie zistenie aktuálneho stavu sledovaného objektu.

Najjednoduchšou variantou tohto útoku by bolo odchytenie takejto odpovede a jej následné zahodenie, avšak toto implementácia virtuálnych zariadení, ktoré som využíval tak úplne nepovoľuje. Na obrázku 4.9 sú zobrazené hodnoty *Tx* a *Rx*, ktoré virtuálne zariadenia pri komunikácii kontrolujú. Samotné zahodenie správy by tak vyvolalo na zariadení chybu a akýkoľvek ďalší prenos by bol prerušený, a to až do jeho manuálneho reštartu. Zahodenie správy ako takej by tak bolo možné len v prípade, že by útočník následne prepisoval tieto hodnoty na správnu postupnosť bez jednej vynechanej správy. Taký útok by bol však nenápadný len do momentu jeho ukončenia, kedy by opäť došlo k detekcii nesprávnych hodnôt.

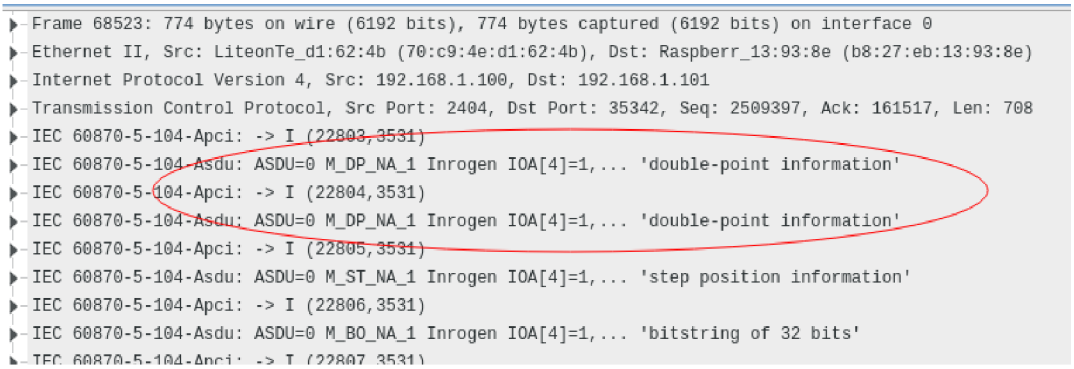


Obr. 4.9: Náhľad na hodnoty Tx a Rx

Pre dosiahnutie efektu blokovania správy som tak prišiel s vhodnejšou alternatívou. Útočník teda môže pozmeniť ohlasovaciu správu, v ktorej sa nachádza viacero prenášaných ASDU správ. Pre dosiahnutie efektu zablokovania tak namiesto jej zahodenia, prepíše kompletný obsah prenášaného ASDU na informácie obsiahnuté v nejakom ďalšom ASDU spolu so zmenou *Tx* a *Rx* hodnôt v prislúchajúcej APCI časti tak, aby ich hodnoty boli v postupnosti zachované a zariadenia tak nedetekovali nedoručenú informáciu. Pre túto variantu útoku je na obrázku 4.10 zobrazená odoslaná ohlasovacia správa a na obrázku 4.11 správa, ktorú obdržalo HMI zariadenie.



Obr. 4.10: Ohlasovacia správa vygenerovaná na zariadení RTU



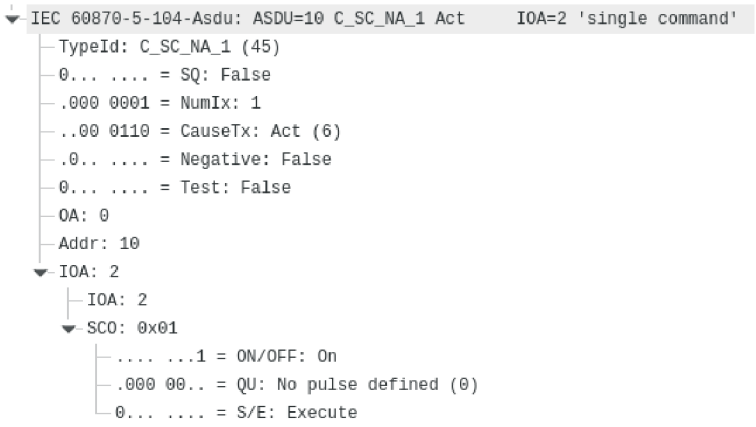
Obr. 4.11: Ohlasovacia správa obdržaná na zariadení HMI

Útok je aplikovateľný na ľubovoľné dve ASDU, avšak samozrejme so zachovaním ich veľkosti, či korektnou úpravou korekčných súčtov paketu. Pri konkrétnom útoku som zamenil informácie o *Single point*, pre všetky dostupné informačné objekty duplikáciou správy o ich *Double point* informáciách. Po dobu trvania útoku tak stanica HMI neobdrží informáciu o aktuálnych hodnotách, čo môže potencionálne slúžiť na prekrytie ich zmeny útočníkom pomocou útoku na úpravu parametrov.

4.5 Úprava parametrov [T0836]

Úprava parametrov je útok, ktorý môže byť vedený na obe strany komunikácie medzi zariadením HMI a RTU, no je však potrebné myslieť na korektnosť úpravy prenášaného atribútu. Na obrázku 4.12 je príklad prenášaného ASDU z paketu, ktorý bol odoslaný ako požiadavka zo strany HMI zariadenia. V prípade zmeny hodnoty *IOA* na hodnotu, ktorú zariadenie RTU nepozná resp. nespravuje prvok s danou hodnotou, by zariadenie RTU, ktoré takúto požiadavku obdrží, odpovedalo chybovou správou (Unknown IOA). Útok je v takomto prípade síce úspešný, keďže sa podarí upraviť komunikáciu medzi zariadením HMI a RTU, avšak nie je veľmi nenápadný a efekt útoku na komunikáciu či zariadenie sa dá porovnať s blokáciou ohlasovacích správ, kde legitímny požiadavok nedorazí na zariadenie RTU.

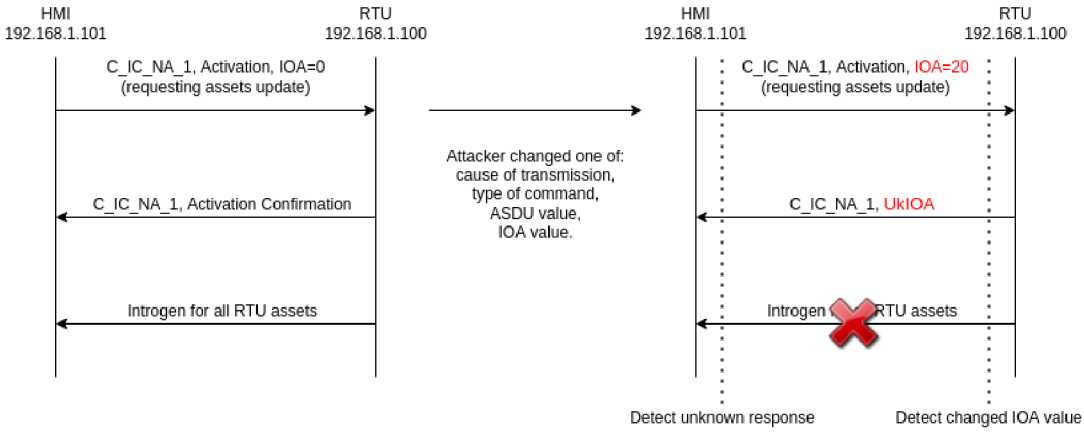
V prípade zmien parametrov jednotlivých IOA objektov k podobnej detekcii nedôjde, samozrejme za predpokladu dodržania formátu prenášanej hodnoty. Avšak v prípade zmeny hodnoty *CauseTx* z *Act (6)* na *Spont (3)* by správa nedodržiavala štandardnú špecifikáciu protokolu IEC 104 a pre zariadenie RTU by nebola ani korektná.



Obr. 4.12: Štandardný single-point dotaz

4.5.1 Úprava na neznámy atribút

Ako som už vyššie spomenul, tento scenár je ľahko detekovateľný hlavne z dôvodu, že zariadenie RTU odpovie na dotaz so zmenenou hodnotou IOA chybovou správou (obrázok 4.13). Pre tento scenár som vybral správu cyklicky odosielanú správu z HMI zariadenia, ktorou sa v štandardnej komunikácii HMI dotazuje aktuálnemu stavu objektov spravovaných zariadením RTU.



Obr. 4.13: Príklad scenára neznámeho atribútu

V tomto scenári som zmenil hodnotu objektu IOA samotného z hodnoty 0 na hodnotu 20. Na obrázku 4.14 je zobrazený dotaz zo zariadenia HMI s korektnou hodnotou a následkom útoku bola kompletne rozdielna odpoveď zo zariadenia RTU ohlasujúca chybný dotaz (obrázok 4.15). Útok ako taký je síce úspešný, no nie je diskrétny. Zabránil obdržaniu korektného požiadavku RTU stanicou, a zároveň zamedzil stanici HMI obdržanie cyklickej aktualizácie o stave objektov spravovaných zariadením RTU. Útok je však ľahko odhaliteľný aj na základe majoritnej zmeny reakcie RTU stanice a to na oboch zariadeniach za predpokladu, že by útočník nemenil aj odpoveď generovanú zariadením RTU, čo by však znemožnilo jednoduchú detekciu len na stanici HMI.

```
▸ Frame 72699: 84 bytes on wire (672 bits), 84 bytes captured (672 bits) on interface 0
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 192.168.1.101, Dst: 192.168.1.100
▸ Transmission Control Protocol, Src Port: 35718, Dst Port: 2404, Seq: 332500, Ack: 5158636, Len: 16
▸ IEC 60870-5-104-Apci: <- I (7368,16038)
▾ IEC 60870-5-104-Asdu: ASDU=0 C_IC_NA_1 Act     IOA=0 'interrogation command'
    TypeId: C_IC_NA_1 (100)
    0... .... = SQ: False
    .000 0001 = NumIx: 1
    ..00 0110 = CauseTx: Act (6)
    .0.. .... = Negative: False
    0... .... = Test: False
    OA: 0
    Addr: 0
  ▾ IOA: 0
      IOA: 0
      QOI: Station interrogation (global) (20)
```

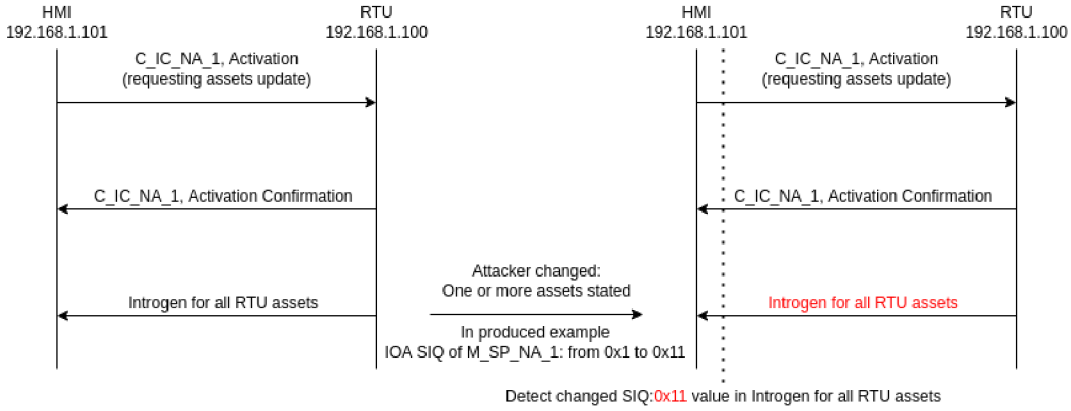
Obr. 4.14: Pôvodná správa pred útokom (Zmena na neznámy atribút)

```
72699 75362.382968… 192.168.1.101   192.168.1.100   104asdu   84 <- I (7368,16038) ASDU=0 C_IC_NA_1 Act     IOA=0
72701 75362.420614… 192.168.1.100   192.168.1.101   104asdu   84 -> I (16038,7369) ASDU=0 C_IC_NA_1 UkIOA_NEGA IOA=20
72703 75372.383227… 192.168.1.101   192.168.1.100   104asdu   84 <- I (7369,16039) ASDU=0 C_IC_NA_1 Act     IOA=0
72704 75372.460289… 192.168.1.100   192.168.1.101   104asdu   84 -> I (16039,7370) ASDU=0 C_IC_NA_1 UkIOA_NEGA IOA=20
72706 75382.383484… 192.168.1.101   192.168.1.100   104asdu   84 <- I (7370,16040) ASDU=0 C_IC_NA_1 Act     IOA=0
72707 75382.492703… 192.168.1.100   192.168.1.101   104asdu   84 -> I (16040,7371) ASDU=0 C_IC_NA_1 UkIOA_NEGA IOA=20
```

Obr. 4.15: Náhľad odpovedí zariadenia RTU (Zmena na neznámy atribút)

4.5.2 Zmena odpovede zariadenia RTU

Tento scenár spočíva v zmene hodnoty pod zvoleným IOA konkrétneho typu, v komunikácii smerujúcej z HMI na zariadenie RTU. Útok je tak opäť zameraný na cyklickú časť komunikácie, avšak tento krát na odpoveď generovanú zariadením RTU (aktualizácia hodnôt zdrojov). Scenár však už nemení ľahko detekovateľný atribút ako v prvom prípade. Dochádza k zmene samotnej dátovej hodnoty pod požadovaným, resp. cieľovým typom informácie (obrázok 4.16).



Obr. 4.16: Zmena hodnoty konkrétneho prvku v odpovedi zo zariadenia RTU

V odchytenom príklade som sa ako útočník rozhodol zmeniť hodnotu *SIQ* pre IOA s indexom 1, a to v pakete obsahujúcom informáciu *M_SP_NA_1* (single point). Samotný útok upravuje iba odpovede odoslané zo zariadenia RTU a nenarúša žiadnu ďalšiu komunikáciu. Paket na obrázku 4.17 bol vygenerovaný ako odpoveď na zariadení RTU, a zároveň na ňom aj priamo odchytený. Na obrázku 4.18 je ten istý paket, avšak po jeho doručení na HMI zariadenie, a samozrejme po jeho upravení útočníkom po ceste.

```
▸ Frame 257: 774 bytes on wire (6192 bits), 774 bytes captured (6192 bits) on interface 0
▸ Ethernet II, Src: Raspberr_13:93:8e (b8:27:eb:13:93:8e), Dst: LiteonTe_d1:62:4b (70:c9:4e:d1:62:4b)
▸ Internet Protocol Version 4, Src: 192.168.1.100, Dst: 192.168.1.102
▸ Transmission Control Protocol, Src Port: 2404, Dst Port: 49608, Seq: 14150, Ack: 1062, Len: 708
▸ IEC 60870-5-104-Apci: -> I (326,27)
▾ IEC 60870-5-104-Asdu: ASDU=0 M_SP_NA_1 Inrogen IOA[4]=1,... 'single-point information'
    TypeId: M_SP_NA_1 (1)
    0... .... = SQ: False
    .000 0100 = NumIx: 4
    ..01 0100 = CauseTx: Inrogen (20)
    .0.. .... = Negative: False
    0... .... = Test: False
    OA: 0
    Addr: 0
  ▾ IOA: 1
      IOA: 1
    ▸ SIQ: 0x00
  ▾ IOA: 2
      IOA: 2
    ▸ SIQ: 0x01
```

Obr. 4.17: Odpoveď vygenerovaná zariadením RTU

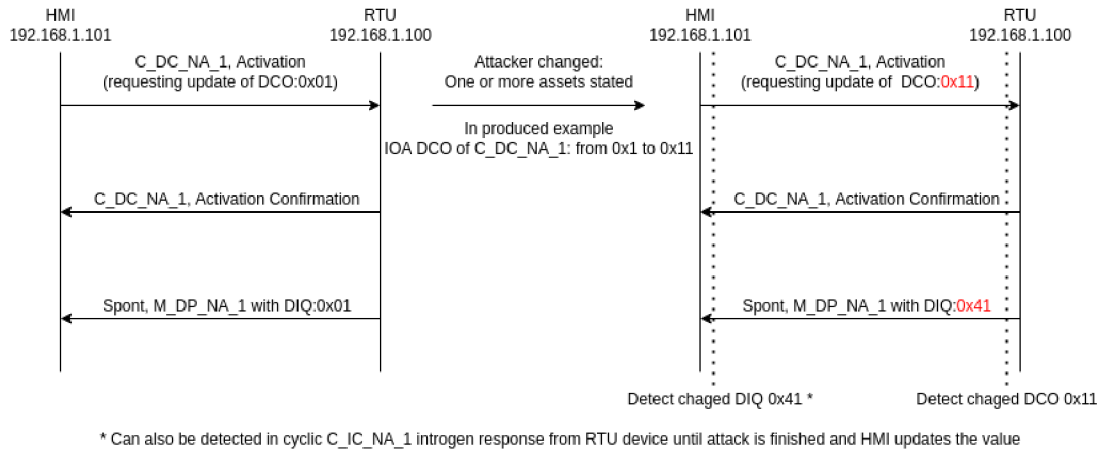
Samotný útok je podstatne diskrétnejší ako útok v prvom scenáre (Úprava na neznámy atribút). Jeho účinok sa prejavuje len dovtedy, pokým je útok resp. útočník aktívny, a teda ho vedie na komunikáciu smerujúcu na HMI zariadenie. Zmena hodnoty SIQ je tak v komunikácii prítomná len do momentu, kým útočník mení odchytávanú komunikáciu *on-fly*. Najjednoduchším spôsobom detekcie je manuálne porovnanie hodnôt odchytených na zariadení HMI a RTU, čo však pri takom množstve prenášaných správ nie je vôbec efektívne.

```
▸ Frame 258: 762 bytes on wire (6096 bits), 762 bytes captured (6096 bits) on interface 0
▸ Ethernet II, Src: LiteonTe_d1:62:4b (70:c9:4e:d1:62:4b), Dst: AskeyCom_f3:8a:5e (e8:39:df:f3:8a:5e)
▸ Internet Protocol Version 4, Src: 192.168.1.100, Dst: 192.168.1.102
▸ Transmission Control Protocol, Src Port: 2404, Dst Port: 49608, Seq: 14150, Ack: 1062, Len: 708
▸ IEC 60870-5-104-Apci: -> I (326,27)
▾ IEC 60870-5-104-Asdu: ASDU=0 M_SP_NA_1 Inrogen_TEST IOA[4]=1,... 'single-point information'
    TypeId: M_SP_NA_1 (1)
    0... .... = SQ: False
    .000 0100 = NumIx: 4
    ..01 0100 = CauseTx: Inrogen (20)
    .0.. .... = Negative: False
    1... .... = Test: True
    OA: 0
    Addr: 0
  ▾ IOA: 1
      IOA: 1
    ▸ SIQ: 0x11
  ▾ IOA: 2
      IOA: 2
    ▸ SIQ: 0x01
```

Obr. 4.18: Odpoveď obdržaná zariadením HMI

4.5.3 Zmena požiadavku HMI zariadenia

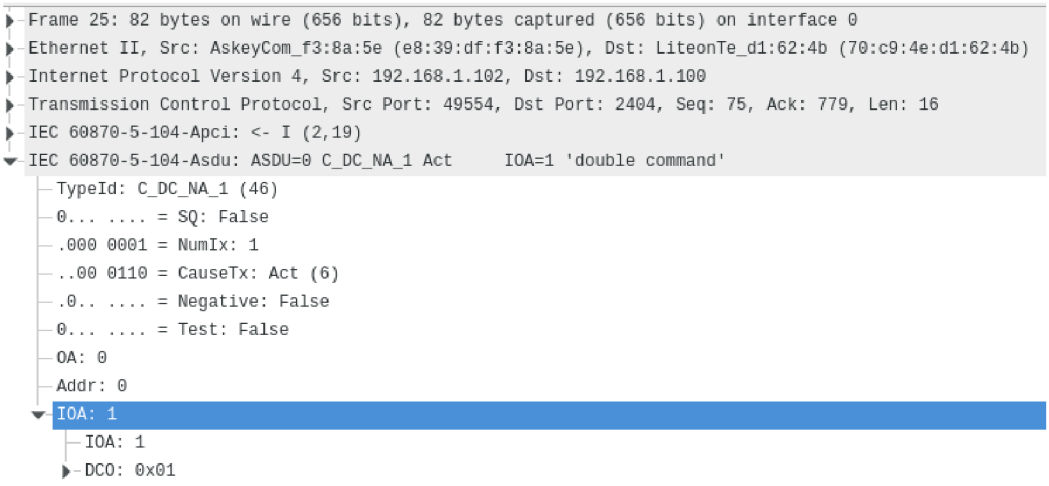
Tento scenár nepriamo útočí na samotné nastavené hodnoty aktív zariadenia RTU, a to úpravou aktuálne prenášaných požiadavok HMI zariadenia o zmenu akýchkoľvek prenášaných hodnôt v rámci cieleného typu dotazu. Zmena je znázornená na obrázku 4.19, a spočíva v jednoduchej úprave hodnoty konkrétneho typu ako v predošlom scenáre, avšak na strane komunikácie smerujúcej na zariadenie RTU.



Obr. 4.19: Zmena konkrétnej hodnoty v dotaze HMI zariadenia

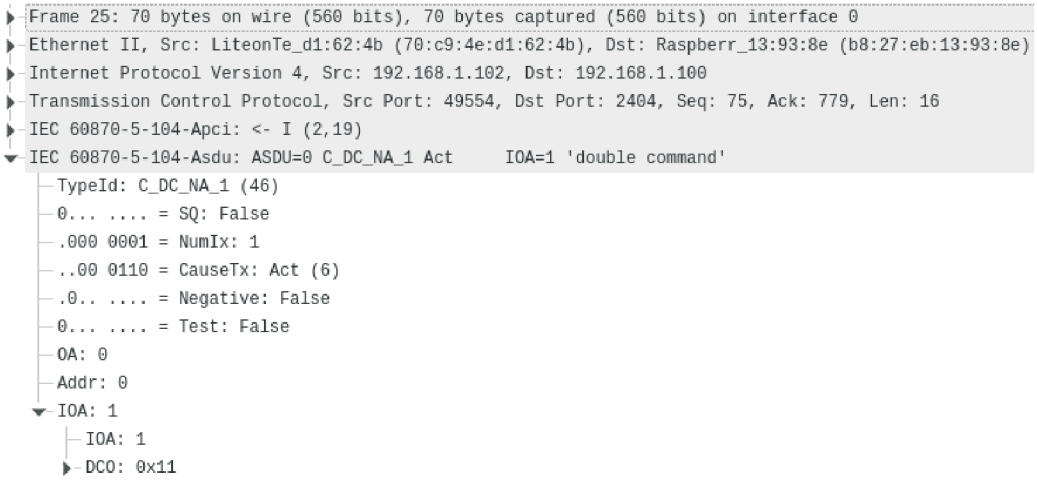
Scenár je na virtuálnych zariadeniach možný vďaka mojej dodatočnej implementácii dátovej štruktúry opísanej v už spomenutej technickej správe [7], pretože bez nej framework využíval na vytváranie komunikácie iba statické príkazy. Zariadenia si tak neviedli databázu aktuálnych hodnôt, čo znamenalo, že sa nedala simulovať úprava preposielaných hodnôt bez statickej úpravy v kóde frameworku.

Pre vykonanie útoku som sa rozhodol zmeniť hodnotu prvku DCO pre IOA s indexom 1 v dotaze generovanom zariadením HMI (*C_DC_NA_1*). Obrázok 4.20 znázorňuje pôvodný vygenerovaný paket, ktorý bol cieľom tohto útoku. Scenár tvorí kombináciu z prvých dvoch, kedy teda dochádza k zmene komunikácie na strane z HMI zariadenia, ale k zmene dochádza priamo v hodnote samotného prvku IOA, ktorá nie je na cieľovom zariadení RTU automaticky detekovaná ako neznáma.



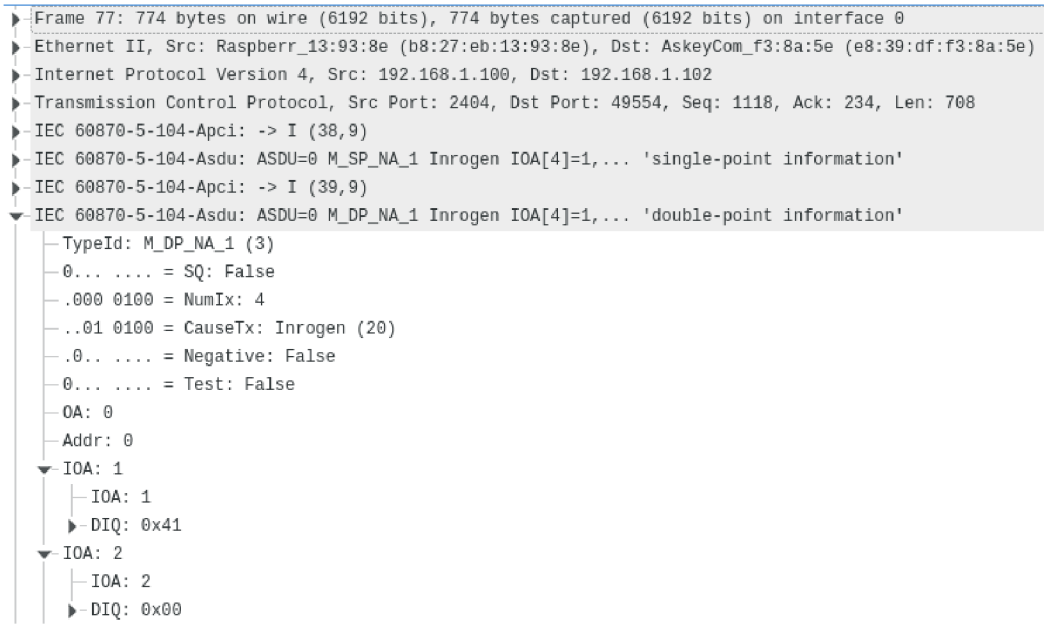
Obr. 4.20: Paket generovaný ako dotaz zariadením HMI

Pôvodná hodnota *0x01* je pri útoku zmenená na *0x11* (obrázok 4.21). Samotná zmena je prvý z efektov útoku, pričom ten je v komunikácii prítomný, len pokiaľ útočník aktívne mení obsah odchytenej správy podobne ako pri zmene odpovede zariadenia RTU.



Obr. 4.21: Paket obdržaný zariadením RTU

Tento krát sa však nejedná o dotaz na zistenie aktuálnych hodnôt poskytovaných prvkov, ale o požiadavku na zmenu hodnoty uloženej pod IOA. Aplikovaním zmeny hodnoty popísanej vyššie dôjde k simulácii zmeny hodnoty na prvku zariadenia RTU. Použitím internej implementácie z použitého frameworku je tak výslendá uložená nová hodnota (*0x41*), pretože samotná požiadavka, ktorú zariadenie RTU obdržalo bola v jeho očiach stále oprávnená. Nedošlo tak k detekcii a následnému odoslaniu negatívnej odpovedi. V následnej cyklickej komunikácii tak zariadenie RTU oznamuje novú hodnotu prvku pre dotazujúce sa HMI zariadenie, a to aj určitú dobu po ukončení samotného útoku. K prepisu danej hodnoty na korektnú totiž dôjde len v prípade, že o ňu legitímne HMI zariadenie požiada. K náprave tak dôjde až po určitej dobe prvým útočníkom nezmeneným dotazom z obrázku 4.20.

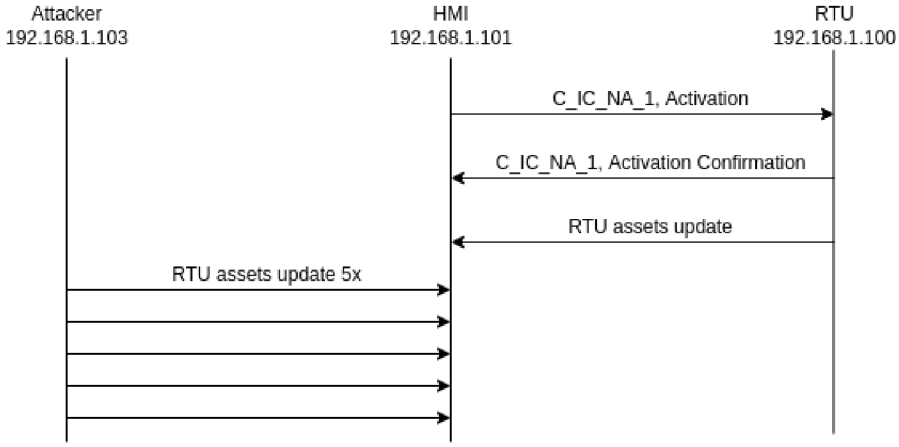


Obr. 4.22: Paket posielaný zariadením RTU v cyklickej komunikácii (obsahuje podvrhnutú hodnotu)

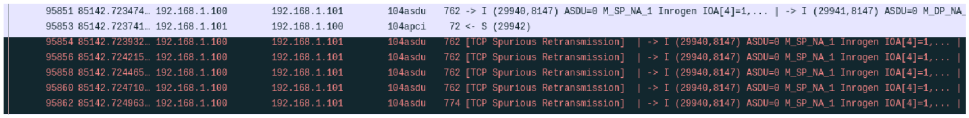
4.6 Replay útok

Útok vychádza primárne z rešerše dostupných dátových sád, keďže si sám o sebe nie je zaradený do MITRE kategórii. Vždy sa tam totiž nachádza len ako súčasť iných konkretnejších útokov, z ktorých niektoré som v práci už spracoval v mierne inej forme. V rešeršovaných datasetoch je cielený na obe koncové zariadenia komunikácie IEC 104 protokolu a to formou jednorázového zopakovania odchyteného paketu smerujúceho na zariadenie, ktoré je cieľom útoku. Požadovaným účinkom by mala byť detekovateľná duplicita a prípadná desinchronizácia TCP spojenia.

Útok som tak spracoval a viedol na obe koncové zariadenia, rovnako ako tomu bolo v skúmaných datasetoch. Na obrázku 4.24 sa nachádza duplikovaný paket, ktorý som si vybral za cieľ pre útok cielený na komunikáciu v smere na HMI zariadenie, kde je tak zároveň aj detekovateľný. Cielený paket oznamujúci aktuálny stav objektov zariadenia RTU je zduplikovaný a preposlaný päť krát na zariadenie HMI (obrázok 4.23).

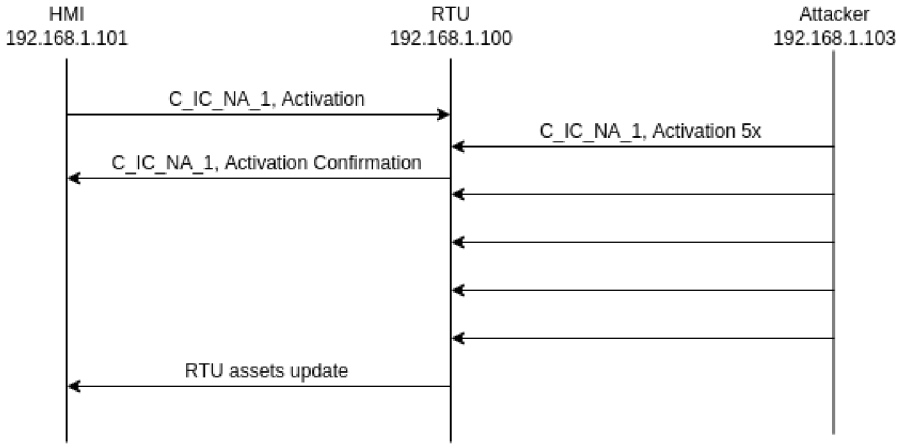


Obr. 4.23: Scenár replay útoku na komunikáciu smerujúcu na zariadenie HMI



Obr. 4.24: Náhľad na duplikovaný paket smerujúci na zariadenie HMI

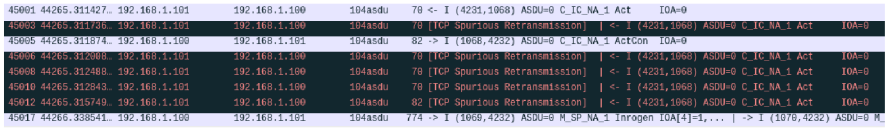
Pri vedení útoku na komunikáciu v smere na zariadenie RTU som sa rozhodol pre duplikáciu *Activation* príkazu *C_IC_NA_1*, ktorý je v štandardnej komunikácii základom cyklickej časti. Opäť dôjde k jeho duplikácii päť krát (obrázok 4.26) ale pre zmenu je detekovateľný na zariadení RTU.



Obr. 4.25: Scenár replay útoku na komunikáciu smerujúcu na zariadenie RTU

Pri oboch prípadoch je nutné aby duplikované pakety dorazili na cieľové zariadenie v čo najkratšom čase po skutočnom pakete. Pri útoku na komunikáciu smerovanú na zariadenie HMI (duplikovaní informácií o objektoch) existuje väčšia časová rezerva ako pri variante útoku na opačný smer komunikácie. Tento fakt je spôsobený tým, že v predošlom scenáre duplikujem posledný prenášaný paket v cyklickej postupnosti. Ďalší cyklus štandardnej komunikácie nastane približne za desať sekúnd, čo tvorí veľkú časovú rezervu pre duplikáciu.

Pri útoku na opačný smer však duplikujem prvý paket cyklu, čo si vyžaduje obdržanie duplicít ideálne pred počiatkom odoslania aktualizácie o objektoch RTU stanice. Dôvodom sú práve hodnoty RX a TX spomenuté vyššie, ktoré by v prípade neskoršieho obdržania duplicitného paketu zhodili virtuálne zariadenie.

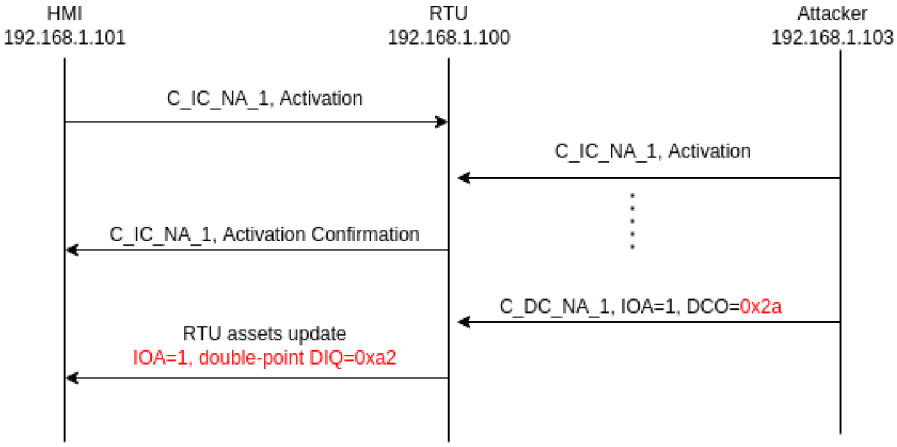


Obr. 4.26: Náhľad na duplikovaný paket smerujúci na zariadenie RTU

Pri oboch útokoch som sa rozhodol pre päť-násobnú duplikáciu, aj keď v skúmaných datasetoch duplikovali paket iba raz. Pri práci používam staršie smerovacie a virtuálne koncové zariadenia, čo má za následok náhodné ojedinelé duplicity TCP segmentov (primárne spôsobené smerovacím zariadením). Chcel som tak umelo vytvoriť rozdiel medzi reálnym útokom a náhodnou anomáliou v sieti.

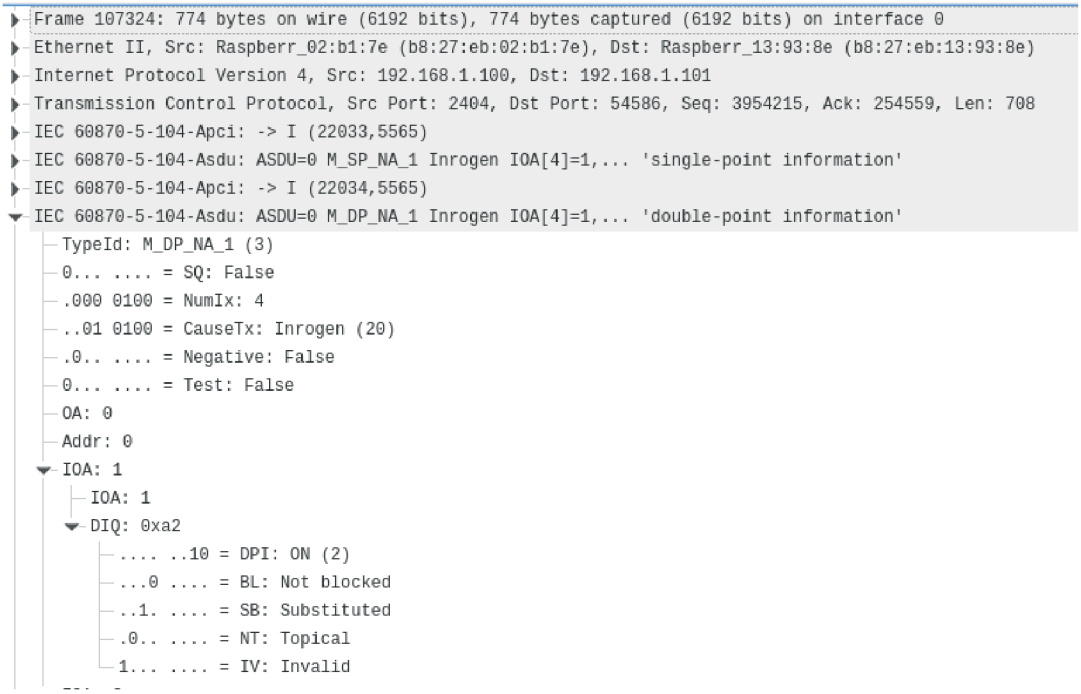
4.7 Maskovanie [T0849] - podvrhnutie zariadenia

Účelom útoku je z útočníkovho hľadiska podvrhnúť zariadenie, ktoré sa správa legitímne a komunikuje korektným spôsobom. Z teoretického hľadiska je ho možné vykonať aj na už existujúcej relácii komunikácie, avšak kvôli implementácii virtuálnych zariadení som nebol schopný tento spôsob útoku vykonať. Pre konkrétny útok som tak bol úspešný len pripojením nového zariadenia a jeho použitím podvrhnúť dáta, ktoré nesúhlasia so štandardnou komunikáciou medzi zariadeniami v sieti.



Obr. 4.27: Komunikácia podvrhnutým zariadením

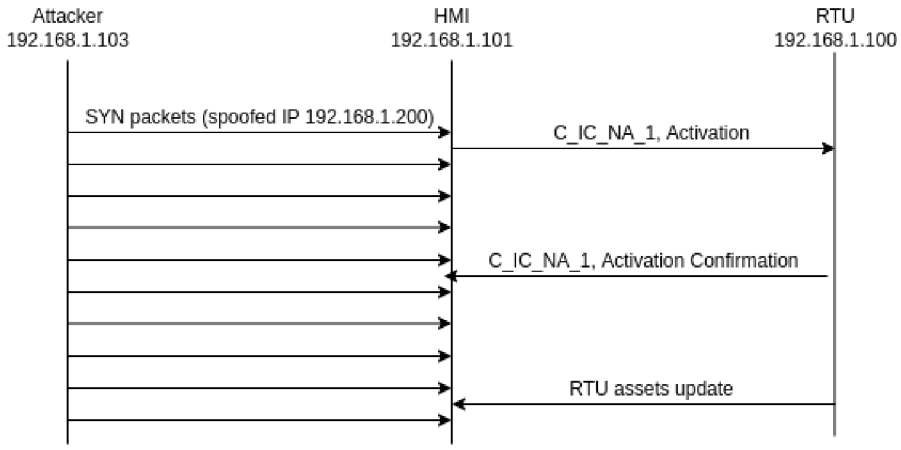
Pri útoku sa tak útočník vydáva za legitímne HMI zariadenie, ktoré sa pripojí na zariadenie RTU a snaží sa cyklicky dotazovať objektom, ktorými RTU disponuje. Okrem toho útočník prevedie dotaz na *double-point* hodnotu IOA objektu 1 (žiadosť na zmenu hodnoty, obrázok 4.27). Žiadosťou vyvolaná zmena sa následne reflektuje do hodnoty, ktorú zariadenie RTU predáva jednak útočníkovi, ale aj každému legitímnemu zariadeniu HMI, ktoré je naň v sieti pripojené (obrázok 4.28).



Obr. 4.28: Hodnota prenášaná ako update legitímnemu HMI zariadeniu

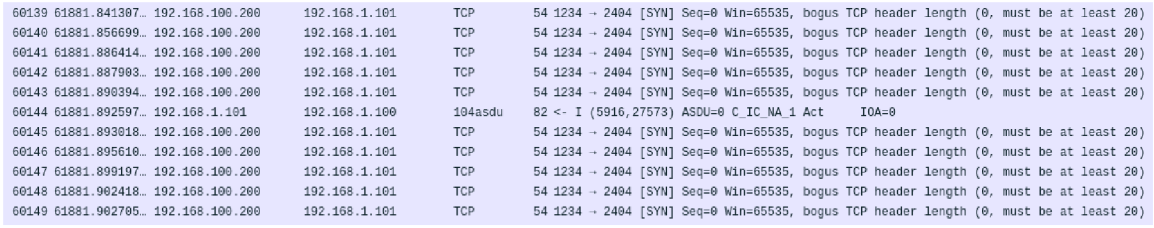
4.8 Odmietnutie služby [T0814] - SYN flood útok

Odmietnutie služby (DOS) patrí k menej špecifickým útokom, ktorého cieľom je zaťaženie cieľového zariadenia veľkým množstvom neoprávnených požiadavok. Požadovaným efektom z pohľadu útočníka je znefunkčniť zariadením poskytované služby pre legitímne zariadenia v sieti. Prevedenie útoku spočíva v zaplavení zariadenia pomocou veľkého množstva neúplnej trojfázovej synchronizácie (obrázok 4.29). Jeho prevedenie môže byť cielené na oba typy zariadení (RTU aj HMI), pričom jeho ideálnym výsledkom je prerušenie IEC 104 komunikácie medzi nimi.



Obr. 4.29: Komunikácia počas podvrhnutých SYN paketov

Pakety generované počas útoku na HMI zariadenie sú zobrazené na obrázku 4.30. V prípade, že je za cieľ zvolené zariadenie RTU, však nedochádza k žiadnej zmene parametrov útoku. Záplavu som počas útoku na obe zariadenia viedol na aktívny port *2404*, keďže ako som už vyššie spomínal, ostatný aktívne služby na Raspberry zariadeniach nemusia byť relevantné pre reálne SCADA systémy.



Obr. 4.30: Príklad z počiatku generovania syn flood záplavy

Vo výsledku bol útok úspešný, avšak ako som očakával, spôsobil kompletný výpadok aktívnej služby na virtualizovaných zariadeniach. Počas vedenia útoku došlo v priebehu niekoľkých sekúnd k uzavretiu spojenia, z dôvodu internej implementácie použitého frameworku. Zariadenia totiž nestíhali spracovať obrovské množstvo paketov smerujúcich na službu virtualizačnej aplikácie, a tak nikdy nezaznamenali legitímnu odpoveď v časovom intervale (timed-out).

4.9 Zhrnutie

Súbor	Packet c.	IEC 104 packet c.	Dĺžka komunikácie	No. devices
HMI_standard	176685	88645	51 h	2
RTU_standard	176703	88725	51 h	2
value_change_HMI	130240	60928	35 h	2
value_change_RTU	133414	60964	35 h	2
value_change2_HMI	267099	133885	77 h	2
value_change2_RTU	267014	133924	77 h	2
report_block_HMI	165471	42976	24 h	2
report_block_RTU	165282	43113	24 h	2
replay_HMI	121112	53013	29 h	2
replay_RTU	121434	55093	29 h	2
masquerating_HMI	259681	67951	39 h	2
masquerating_RTU	262252	69450	39 h	3
dos_targetHMI_HMI	92084	29984	17 h	2
dos_targetHMI_RTU	60147	30071	17 h	2
dos_targetRTU_HMI	66251	33144	19 h	2
dos_targetRTU_RTU	81571	33069	19 h	2

Tabuľka 4.1: Odchytené dáta.

V kapitole som detailne priblížil jednotlivé emulované útoky spolu s popisom ich efektu na riadiacu komunikáciu. Výstupom tejto kapitoly je tak hlavne návod na ich replikáciu v ľubovoľnej štandardnej komunikácii a samotné datasety *PCAP* súborov, ktoré vo väčšom časovom intervale zachytávajú jednotlivé vykonané útoky na zariadeniach HMI a RTU. Tieto súbory a základné parametre zaznamenanej komunikácie sú priblížené v tabuľke 4.1. Tieto dáta mi v ďalšej časti budú tvoriť predlohu pre návrh a implementáciu detekcií anomálií pomocou štatistických metód v ICS/SCADA riadiacej komunikácii. Rovnako tak ich použijem pre vyhodnotenie presnosti detekcie navrhnutej implementácie.

Súbor	Štart		Koniec		Trvanie[min]
	Packet	Čas[s]	Packet	Čas[s]	
value_change_HMI	37850	37251	45923	44031	113
value_change_RTU	88654	86093	97164	93343	120
value_change2_HMI	72311	74122	74411	80972	114
value_change2_RTU	72315	74122	74415	80972	114
report_block_HMI	68523	36903	84518	44804	131
replay_HMI	44889	44201	62794	52561	139
replay_RTU	95844	85142	104949	89402	71
masquerating_RTU	107237	58197	116322	61233	50
dos_targetHMI_HMI	61881	60139	92082	61951	Dropped
dos_targetRTU_RTU	65910	68151	81567	68213	Dropped

Tabuľka 4.2: Zachytené útoky v súboroch.

Pre účeli dohľadania vykonaných útokov k popisu dopĺňam druhú tabuľku 4.2, ktorá obsahuje špecifikované časy a trvania útokov v jednotlivých *PCAP* súboroch. Takmer každý útok predstavoval maximálne dve varianty podľa cieľovej strany komunikácie, a tak je ich možné detekovať vždy na zariadení, ktoré predstavovalo prijímateľa na strane komunikácie.

Kapitola 5

Detekcia pomocou štatistických metód

Kapitola je zameraná na návrh a implementáciu detekcie jednotlivých emulovaných útokov popísaných v kapitole 4 pomocou vybraných štatistických metód. Súčasťou kapitoly sú experimenty, ktorými overím presnosť použitých metód.

Pred samotnou implementáciou som musel vyriešiť reprezentáciu dát jednotlivých komunikácii tak aby bola použiteľná pre detekciu čo najväčšieho množstva emulovaných útokov. Ako prvá možnosť sa ponúkla extrakcia dát v ponuke nástroja Wireshark do formátu *CSV*, ktorá by postačovala na určitú časť útokov. Avšak napríklad *Block reporting message* upravuje časť dát, ktorú Wireshark extrahuje v zhluku pod kolónkou *Info*. Preto som sa rozhodol použiť reprezentáciu, ktorú mi poskytol skript *IEC 104toflow.sh*, ktorý bol vytvorený a použitý pre roztriedenie prenášanej komunikácie protokolu IEC 104 pre štatistické profilovanie[1] výskumnou skupinou *NES@FIT*. Skript vytiahne všetky podstatné dáta komunikácie po paketoch a zaznačí ich rozčlenené vo formáte *CSV* vrátane kolónky *Info*. Z tej vytiahne a rozčlení údaje pre jednotlivé prenesené *ASDU*, ktoré pridá vždy na nový riadok spolu s informáciou, ku ktorému paketu sú pridelené. Toto mi umožnilo v prípade potreby využiť na detekciu ďalšie parametre prenášanej komunikácie protokolu IEC 104. Skript som si však musel z dôvodu kompatibility veľmi mierne upraviť (volania doinštalovaných aplikácií v Linuxe a podobne), preto ho prikladám medzi odovzdanými súbormi.

5.1 Vybrané štatistické metódy

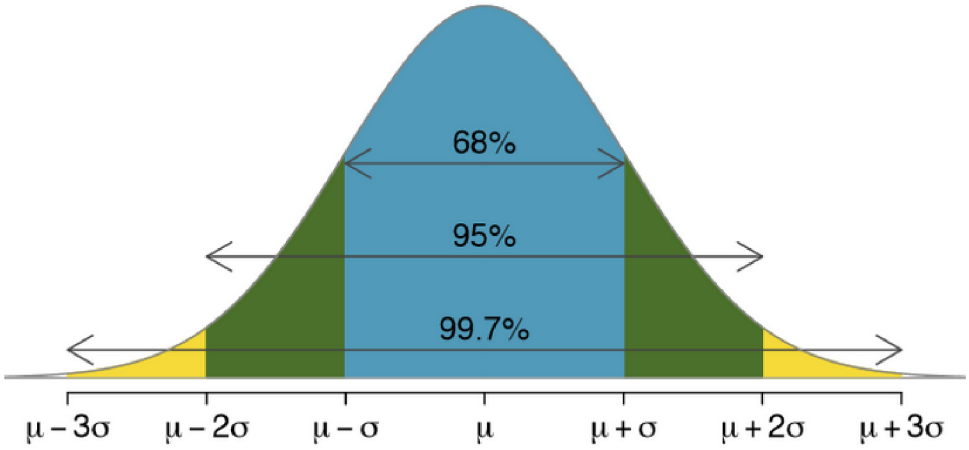
Mojím zámerom bolo vybrať dve štatistické metódy, pomocou ktorých by sa dali detekovať emulované útoky z kategórie 4. Jedna z metód mala byť základného štatistického typu a druhá má obsahovať určitý typ strojového učenia na základe vstupných dát. Na detekciu by mohla úplne stačiť jedna správne zvolená metóda, ale takto zvolené metódy môžem porovnať aj v presnosti medzi sebou a nie len v presnosti na vytvorených dátových sadách.

5.1.1 Three-Sigma

Metóda *Three-Sigma* [15] (resp. pravidlo) vychádza z predpokladu, že v prípade normálne rozdeleného (Gaussovo rozdelenie) štatistického súboru sa majorita (99.7%) sledovaných hodnôt mala nachádzať do troch smerodajných odchýlok (3σ) od aritmetického priemeru

[1] https://github.com/nesfit/bonnet-ics-library

súboru hodnôt. Zvyšné hodnoty sa pri normálnom rozdelení považujú za zanedbateľné, avšak v prípade potreby je možné zvýšiť toleranciu k extrémnym hodnotám navýšením násobku smerodajnej odchýlky. V prípade použitia na súbore dát, ktorý nevychádza z normálneho rozdelenia môže byť však pravdepodobnosť extrémov mnohonásobne vyššia, preto môže byť navýšenie tolerancie nutnou súčasťou pri niektorých datasetoch.



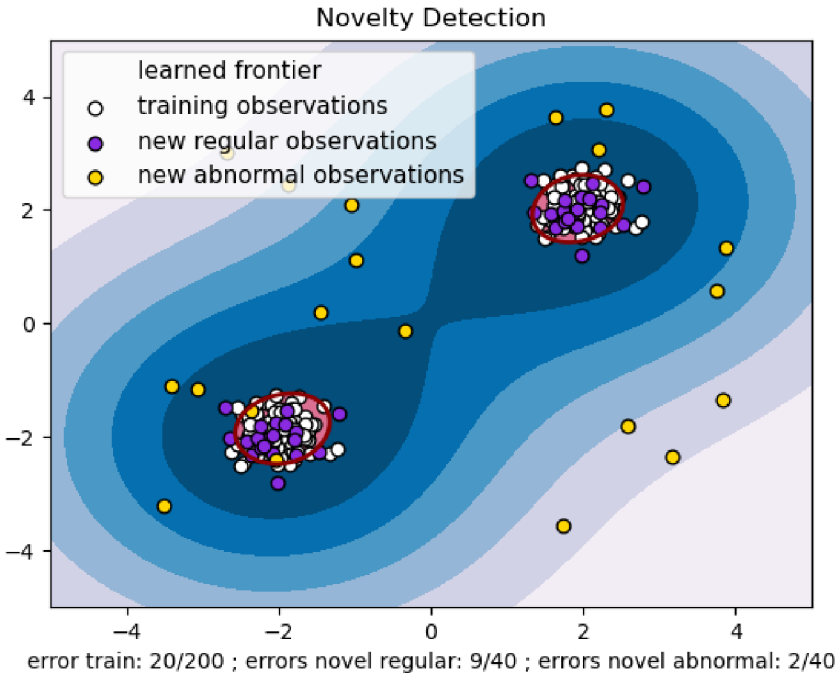
Obr. 5.1: Graf normálneho rozdelenia so smerodajnými odchýlkami (sigma)

Využitie pravidla pre detekciu bude pomerne priamočiare a na podobnom princípe ako natrénovanie modelu pre klasifikáciu pozitívnych a negatívnych dát. Presnosť detekcie bude navyše závisieť na zvolených parametroch vytiahnutých z komunikácie protokolu IEC 104. Z princípu pôjde o naučenie (resp. zaznamenanie) zvolených parametrov (vo forme štatistického súboru) tak, aby čo najmenej hodnôt spadalo nad tri smerodajné odchýlky dát. Následným porovnaním rovnakých parametrov anomálnej funkcie voči takto vytvorenému modelu, by malo byť možné detekovať anomálne fungovanie v komunikácii.

5.1.2 Support Vector Machines (SVM)

SVM je metóda strojového učenia s učiteľom. Základným stavebným kameňom podporných vektorov je lineárny klasifikátor dvoch tried pomocou klasifikovaných trénovacích dát. Cieľom metódy je nájsť nadrovinu, ktorá je schopná optimálne rozdeliť trénovacie dáta do pozitívnych a negatívnych kategórii (polovíc priestoru delených rovinou). Na popis nadroviny špecifikujúcej kategóriu postačujú body ležiace na okraji hraničného pásma, ktorých je zvyčajne málo, a práve tieto body sa nazývajú podpornými vektormi.

Metódy SVM majú veľa variant, no zaujímavá z pohľadu detekcie anomálnej sieťovej komunikácie je práve varianta *One-Class SVM* [4]. Jedná sa už o metódu strojového učenia bez učiteľa, ktorá sa učí rozhodovaciu funkciu pre detekciu (respektíve rozhodovanie) o klasifikácii nových dát ako podobné alebo odlišné od tých v trénovacej sade.



Obr. 5.2: Novelty (nové dáta) - príklad detekcie z natrénovaného modelu

5.2 Vhodné parametre na detekciu

Pri výbere parametrov som musel zobrať v úvahu aj efekty útokov z kapitoly 4, ktoré chcem detekovať. Napríklad pre detekciu útoku *Úprava parametrov* 3.5 by mohlo stačiť sledovať veľkosť jednotlivých paketov za predpokladu, že útočník zmení prenášané dáta nápadnejším spôsobom ako útok, ktorý som spracoval. Majorita agresívnych útokov, ktoré som emuloval veľkosť samotných paketov nemení a tak by táto hodnota sama o sebe nepredstavovala validný parameter pre detekciu.

Majoritná časť agresívnych emulovaných útokov (Modify parameter, Replay, Masquerading, DOS) však generuje zmenu v počte prenášaných paketov (viz. kapitola 4). Samotný počet však tiež nestačí práve napríklad pre útok *Úprava parametrov*. ktorý aj pri drastickejšej úprave prenášaných paketov nespôsobí zmenu ich počtu. Najvhodnejším parametrom pre obe zvolené metódy sa tak ponúka typický prístup k prenášaným dátam akejkoľvek sieťovej komunikácie, ktorým je veľkosť prenášaných dát v určitom časovom intervale. Pri správne zvolenom intervale tak veľkosť prenášaných dát počas spomenutých útokov stúpne či klesne.

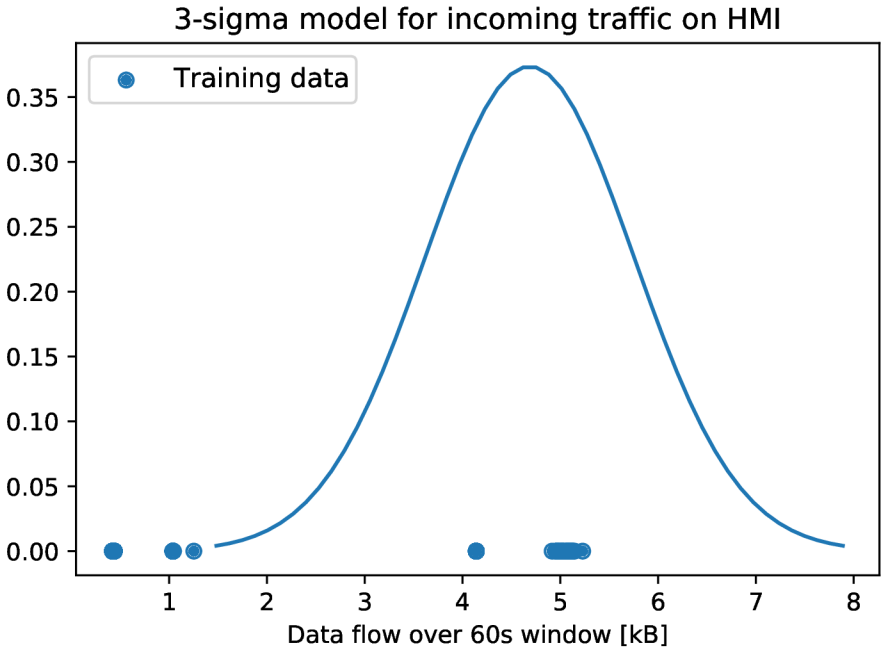
Samotná veľkosť prenášaných dát postačuje pre metódu *Three-Sigma*, avšak pre metódu *One-Class SVM* je potrebný druhý parameter. Ako vhodným doplňujúcim parametrom je počet prenesených paketov vo zvolenom časovom okne.

5.3 Validácia metód

Pri detekcii som sa rozhodol pre separáciu jednotlivých strán komunikácie. Obe strany sa totiž celkovo líšia vo veľkosti celkových prenesených dát, čo by mohlo pri detekcii spôsobiť

väčšie problémy skreslením extrémnych (okrajových) hodnôt. Pri oboch modeloch som pracoval s prichádzajúcou komunikáciou na obe komunikujúce zariadenia, keďže všetky agresívne útoky sú primárne viditeľné práve na zmenách v prichádzajúcej komunikácii. Validácia štandardných (trénovacích) dát prichádzajúcich správ na jednom zariadení je samozrejme rovnaká ako validácia tých odchádzajúcich na zariadení z opačnej strany komunikácie. V prípade potreby je vo vytvorenom *python* skripte možnosť upraviť nosnú podmienku na detekciu z odchádzajúcich dát.

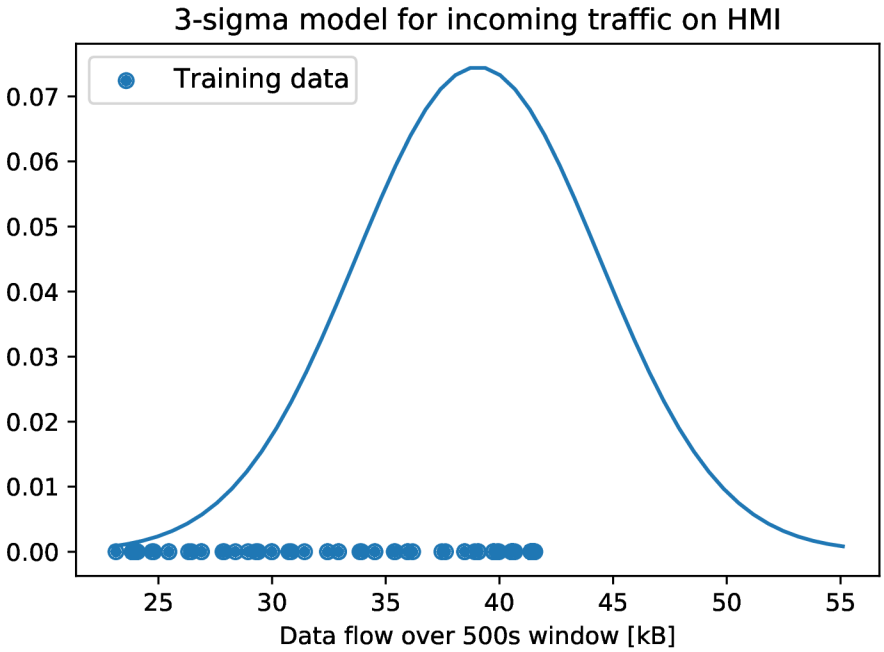
5.3.1 Metóda Three-Sigma



Obr. 5.3: Three-sigma so zle zvoleným oknom

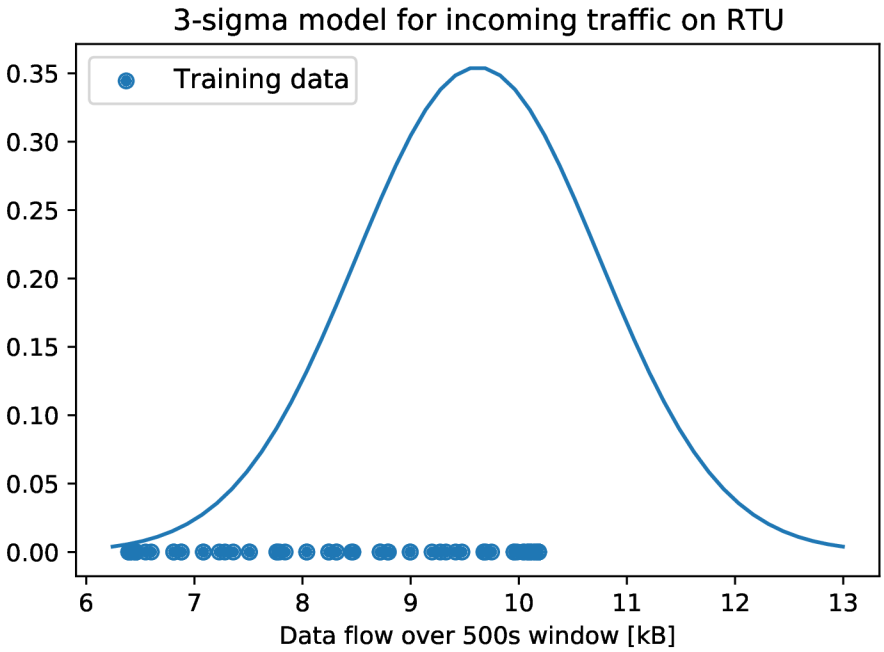
Pre správne fungovanie modelu *Three-Sigma* pri použití parametrov spomenutých v sekcii 5.2 je potrebné zapracovať vlastnosti štandardnej komunikácie pri zvolení sledovaných časových okien. Z obrázka 5.3 je možné vidieť, že pri voľbe časového okna o veľkosti šesťdesiat sekúnd sa priamo v trénovacej komunikácii nachádza oblasť s menšou kapacitou prenesených dát. Pri detekcii je tak vysoko pravdepodobné, že by sa podobná sekvencia objavila v skúmanej komunikácii, pričom by sa stále jednalo o oprávnenú komunikáciu v sieti (*false-positive* detekcia).

Dôvodom spomenutých falošných detekcii bola časť štandardnej komunikácie, v ktorej HMI zariadenie požaduje úpravu niektorých parametrov na zariadení RTU (4.1). Tá pokrýva úsek približne dvesto štyridsiatich sekúnd, a preto som bol nútený zvoliť dostatočne veľké okno, ktoré by ju zaiste celú pokrylo spolu s dostatočne veľkým objemom dodatočných dát tak, aby som dostatočne zväčšil extrémne minimum nevyhovujúce pravidlu *Three-Sigma*. Ako vhodným oknom sa ukázala veľkosť päťsto sekúnd (obrázok 5.4), ktoré toto minimum posunulo do ohraničenia tromi smerodajnými odchýlkami..



Obr. 5.4: Three-sigma na strane príchodzej komunikácie na zariadenie HMI (vhodné okno)

Na obrázku 5.5 sú zobrazené dáta komunikácie smerovanej zo zariadenia HMI na RTU, ktoré sú spracované s rovnako veľkým oknom ako opačná strana komunikácie. Pre separáciu anomálii v štandardnej komunikácii by tak mohla byť zvolená veľkosť okna dostatočnou.

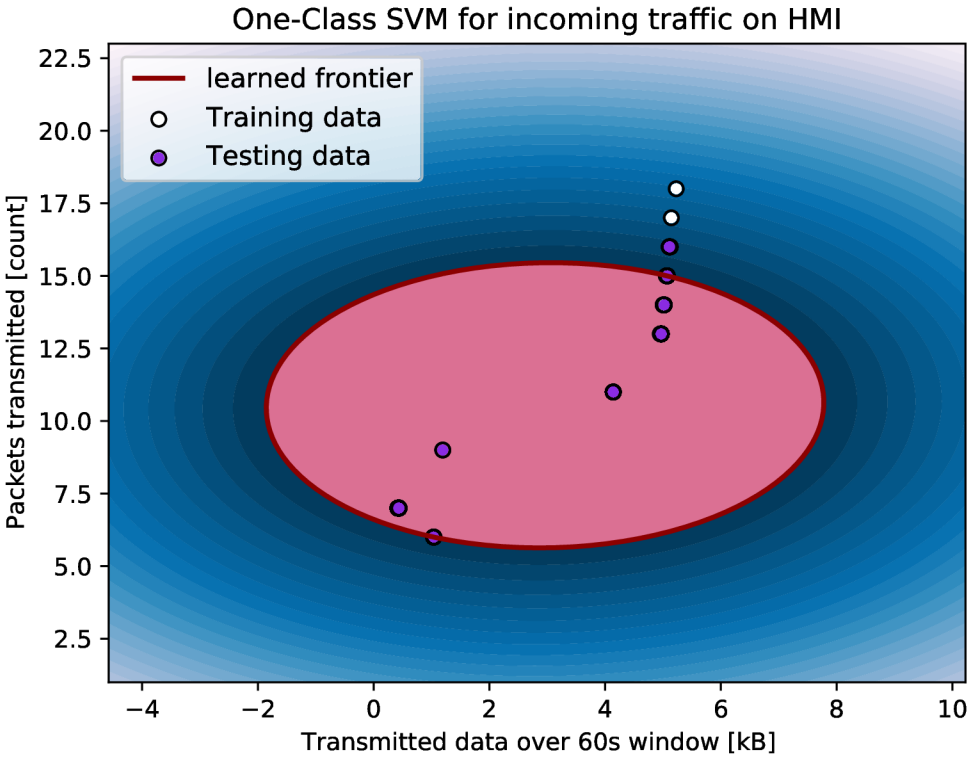


Obr. 5.5: Three-sigma na strane príchodzej komunikácie na zariadenie RTU (vhodne okno)

Už pri porovnaní grafického zobrazenia rozdelených strán komunikácie je jasne vidno rozdiel vo veľkosti prenesených dát na oboch stranách komunikácie, a preto bolo pre detekciu útokov jej rozdelenie správnym krokom. Z oboch obrázkov je tiež možné vyčítať, že zatiaľ čo majorita komunikácie v časových oknách je v okolí 40kB a 10kB (aritmetický priemer je v ich okolí), tak na oboch stranách dochádza k jej drastickému zníženiu v určitom čase. Pre správne vyhodnotenie útokov tak môže byť potrebná manipulácia s časovým oknom smerom nahor, čím môžem sledované dáta priblížiť bližšie k forme normálneho rozdelenia a posunúť tak hranicu rozhodnuteľnosti pre anomálnu komunikáciu. Táto úprava je však validná len pre útoky, ktoré sú vykonávané po dlhšiu časovú dobu. Extrémne krátke útoky na jeden paket či pár sekundový časový interval by sa totiž stratili vo zväčšenom časovom okne.

5.3.2 One-Class SVM model

Jedná sa o metódu strojového učenia bez učiteľa (unsupervised) ako som už spomenul vyššie. Preto som pre validáciu modelu zvolil dodatočné rozdelenie komunikácie na trénovacie a testovacie dáta v pomere sedem ku trom. Na každom ďalšom zobrazenom grafe sa tak budú nachádzať už minimálne dva typy dát pre trénovacie a testovacie dáta. Štandardná komunikácia v jednotlivých oknách sa však líši rádovo v desiatkach až stovkách bajtov, čo však nie je možné zobraziť korektne v rozmedzí približne 10kB (rozpätie dát v oknách celej komunikácie).

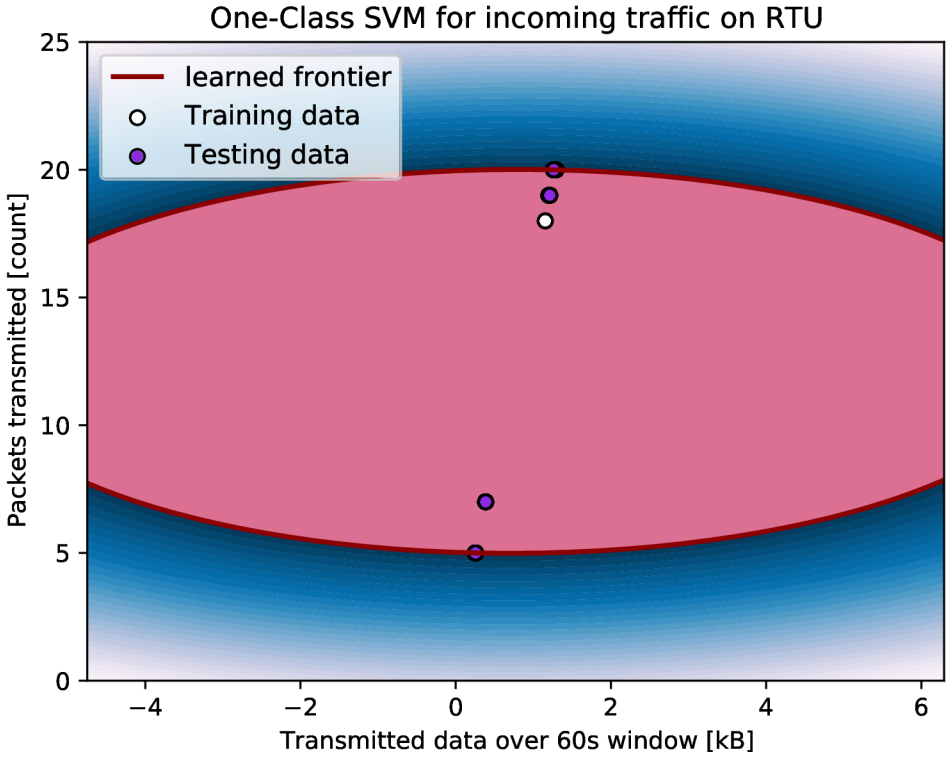


Obr. 5.6: One-Class SVM na strane príchodzej komunikácie na zariadenie HMI

Pri zobrazení dát rádovo v kilobajtoch je tak zobrazenie naučenej rozhodovacej funkcie oveľa prijateľnejšie, avšak veľa bodov štandardnej, trénovacej a neskôr komunikácie s ano-

málnou časťou sa tak v grafe pri tejto mierke prekrýva. Prioritne som tak zvolil zobrazenie testovacích dát nad trénovacími, pričom dáta s anomáliami budú zobrazené za všetkými ostatnými. Tie totiž v každom skúmanom súbore predstavujú len určitú časť dát. Vyhodnotenie celého súboru by tak takmer kompletne prekrylo zobrazenie trénovacích a testovacích dát, pričom tie relevantné pre nález anomálie budú viditeľné aj bez toho aby boli v popredí (budú rozdielne od štandardných).

Na obrázku 5.4 je zobrazené vyhodnotenie trénovacích a testovacích dát komunikácie smerujúcej na HMI zariadenie. Zo zobrazených dát je tiež možno vidieť, že už vo fáze trénovania a testovania modelu dochádza k vyhodnoteniu okrajových hodnôt za *false-positive*. Aj keď sa na prvý pohľad môže zdať, že model nie je správny, a že vyhodnocuje veľkú časť štandardnej komunikácie za nevyhovujúcu, tak tomu tak nie je. Opäť je totiž potrebné pripomenúť skreslenie v zobrazovaných dátach, ktoré sú rovnaké či približne rovnaké a rovnako ich počet. Vo veľkej časti štandardnej cyklickej komunikácie pri staticky zvolenom okne totiž nájdeme rovnaký počet prenesených paketov a približne rovnako veľké prenášané pakety. V realite totiž pri vyhodnotení modelu vyšla približne $97,39\%$ presnosť na testovacích dátach, čo znamená že body nachádzajúce sa za hranicou rozhodovacej funkcie tvoria len približne $2,61\%$ z celej komunikácie.



Obr. 5.7: One-Class SVM na strane príchodzej komunikácie na zariadenie HMI

Pri vyhodnotení dát z komunikácie smerujúcej na zariadenie RTU (obrázok 5.7) je možné vidieť, že rozmedzie veľkosti prenášaných dát je oveľa menej rozmanité ako tomu bolo pri komunikácii smerujúcej na zariadenie HMI. Na prvý pohľad sa tak môže zdať, že je naučená funkcia o niečo presnejšia, avšak presnosť klasifikácie na trénovacích dátach vyšla približne 95.86%. Táto presnosť by sa dala zvýšiť nastavením väčšej tolerancie k extrémom, avšak pri experimentoch sa to ukázalo ako zlý nápad. Zvýšenie tolerancie totiž

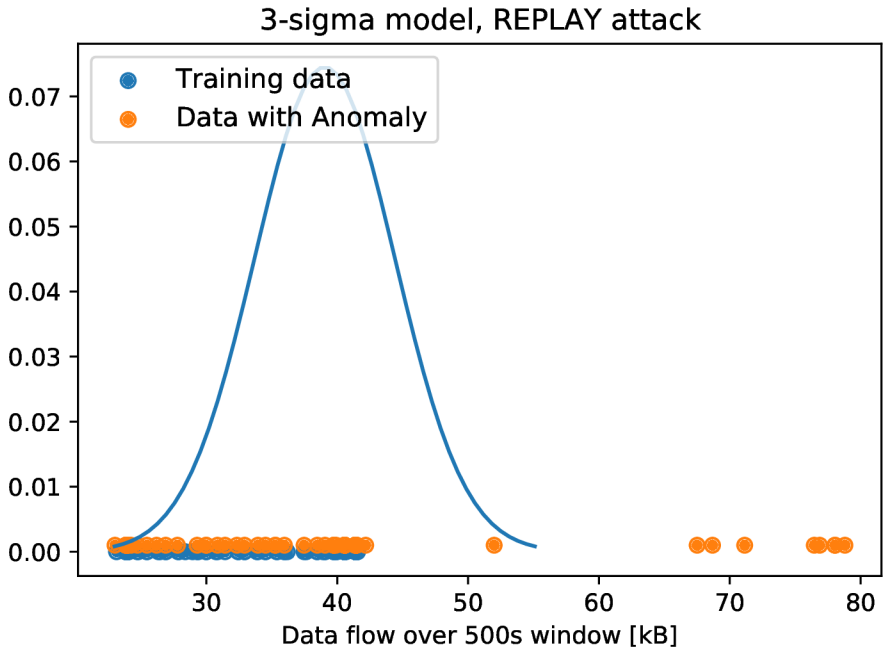
viedlo k pomerne veľkému množstvu *false-negative* predikcií, kedy anomália v komunikácii bola nesprávne klasifikovaná pod štandardnú.

5.4 Detekcia útokov

Sekcia je venovaná vyhodnoteniu detekcie anomálií (útokov) emulovaných v kapitole 4 pomocou implementovaných štatistických metód popísaných na začiatku kapitoly. Pre detekciu metódou *Three-Sigma* som pri každom útoku vyhodnocoval Veľkosť prenesených dát v stanovenom časovom okne. Využitie tohto parametra komunikácie vyzeralo od začiatku ako validné rozhodnutie, a očakával som problém len s útokom *Block report message*. Ten totiž nijak nezasahuje do veľkosti prenášaných dát (kapitola 4.4). Použitie hodnôt prenášaných typov ASDU ako som učinil neskôr pri *One-Class SVM* by mohlo byť riešením pri tomto type útoku, avšak nie pri jeho konkrétnej emulovanej verzii. Pri prevedenom útoku dochádza k zmene hodnoty typu asdu z jedna na tri (v decimálnej reprezentácii), čo je skutočne veľmi malý rozdiel, ktorý by pri žiadnej veľkosti okna neprekročil hodnotu troch odchýliek. Preto pri tomto útoku prezentujem len neúspešnú detekciu metódou *Three-Sigma* za použitia rovnakých parametrov ako pri ostatných skúmaných súboroch.

5.4.1 Replay

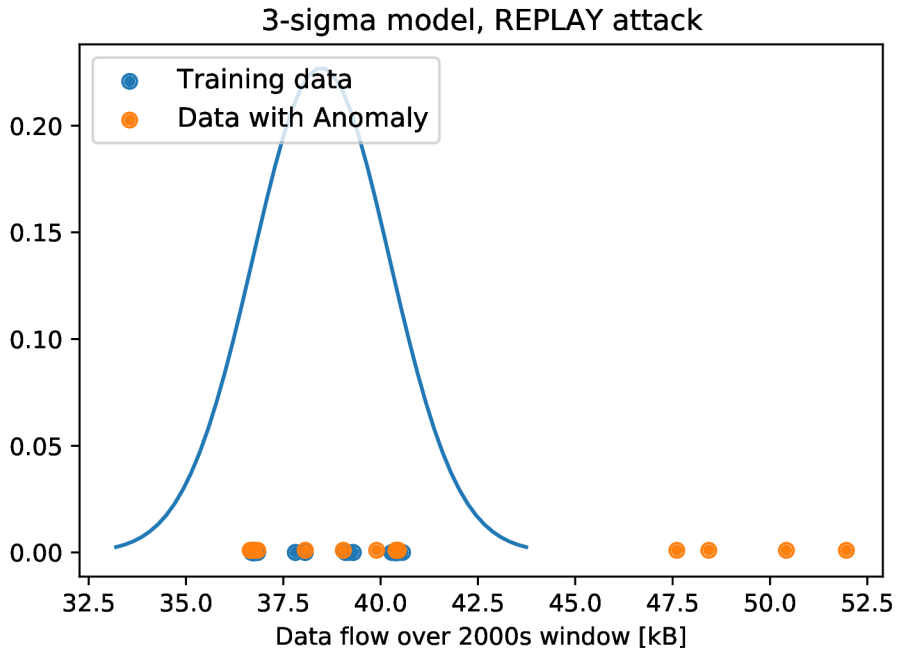
Replay útok som viedol pri emulácii na obe strany komunikácie, preto som aj vyhodnocoval detekciu anomálií na súboroch odchytených na oboch zariadeniach. Na obrázku 5.8 je možné vidieť, že časť komunikácie s anomáliou presiahla hranicu prijateľnosti pravidlom Three-Sigma v maximálnych hodnotách, čo odpovedá zväčšeniu prenesených dát v počas útoku. Bod zobrazený v hodnote približne 52kB naznačuje, že by sa mohlo jednať o *false-negative* detekciu ale nie je tomu úplne tak.



Obr. 5.8: Three-Sigma, útok Replay na komunikáciu (RTU->HMI)

Metódou detekované intervaly pre anomáliu sú <85000s, 88000s> a <88500s, 89500s)>. Detekcia je pomerne presná až na odchýlku spôsobenú päťsto sekundovým oknom. Útok je v súbore vedený v intervale <85142s, 89402s>, ktorý približne zodpovedá detekovanej časti. Dôvod nespojitosti detekovanej časti je pomerne prostý. Emulovaný útok nemá približne v tom čase na komunikáciu žiadny efekt, keďže sa tam nachádza cyklická časť komunikácie o približnej dĺžke dvestodvadsať sekúnd, ktorá neobsahuje útokom zasiahnutý paket (Replay nemá cieľ).

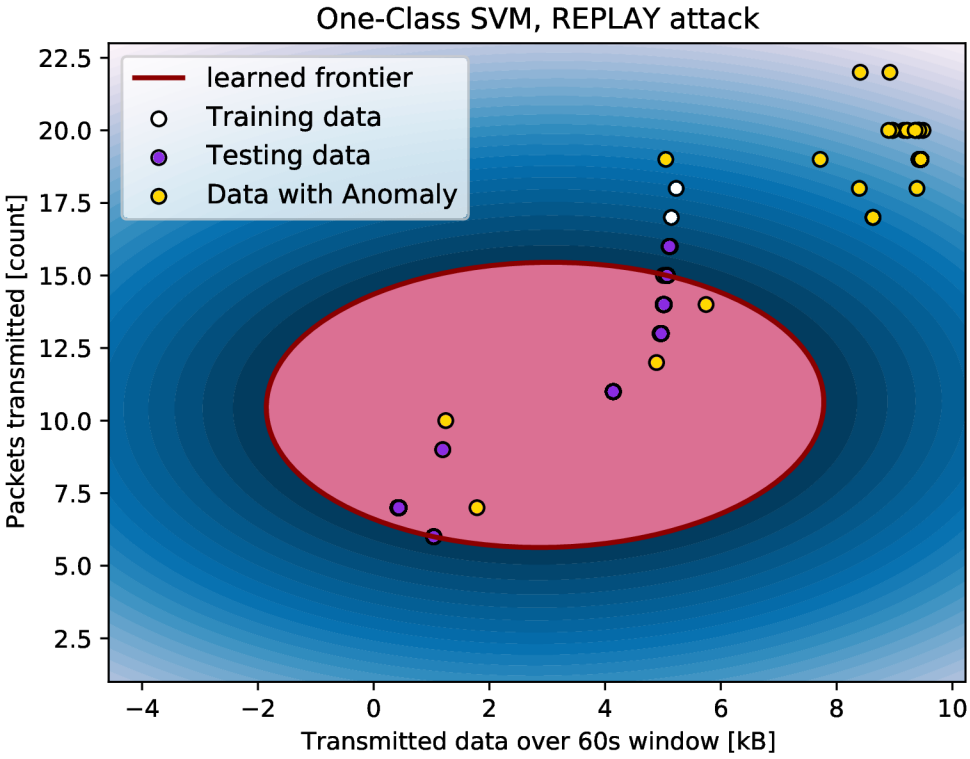
Pri útoku z opačnej strany komunikácie (obrázok 5.9) je situácia pri detekcii pomerne podobná. Detekovaný interval je však tento krát spojitý, keďže korektná detekcia nastala až po zvýšení časového okna, pri ktorom bola nezasiahnutá časť komunikácie dostatočne prekrytá so zasiahnutou. Avšak celková komunikácia príchodzích paketov na zariadenie RTU má skôr riadiaci ako informačný charakter, čo sa vyznačuje veľkým počtom pomerne malých paketov (vrátane paketu zasiahnutého útokom). Útok tak negeneruje dodatočné pakety v dostatočne veľkej miere pre detekciu pri časovom okne päťsto sekúnd. Všetky časové okná tak spadali pri detekcii do kategórie korektnej komunikácie o mierne zvýšenej veľkosti oproti pôvodnej štandardnej komunikácii. Detekcia tak bola úspešná až po zvýšení časového okna na hodnotu dvoch tisíc sekúnd. Detekovaný časový interval bol <44033s, 52033s>, čo je opäť približne správny odhad s mierne väčšou odchýlkou ako na opačnej strane komunikácie, keďže emulovaný útok prebiehal v intervale <44201s, 52561s>.



Obr. 5.9: Three-Sigma, útok Replay na komunikáciu (HMI->RTU)

Detekcia pomocou modelu *One-Class SVM* bola vo všetkých prípadoch presnejšia ako pomocou Three-Sigma pravidla, čo bolo z veľkej miery spôsobené tým, že bol model schopný korektného vyhodnotenia na časovom okne šesťdesiat sekúnd. Detekované intervaly anomálie na zariadení HMI (obrázok 5.10) boli <85140s, 88260s> a <88560s, 89340s>. Detekované intervaly predstavujú takmer presné určenie zasiahnutej časti komunikácie vrátane vynechaného okna pre komunikáciu bez paketu, ktorý bol cieľom útoku. Pri modeli je tiež možné

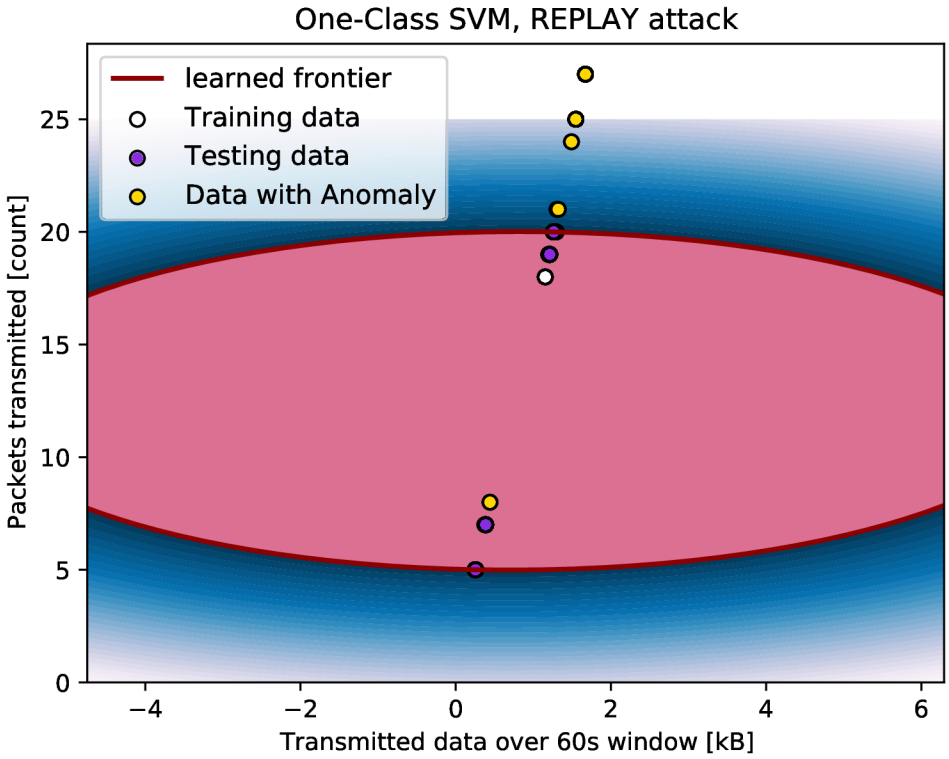
sledovať aktuálnu presnosť na nových testovaných dátach, ktorá bola približne 94.12%, čo je možné vypočítať z predpokladu, že celá skúmaná komunikácia s anomáliou je korektná (ground truth). To značí, že približne pri 5.88% šesťdesiat sekundových oknách model vyhodnotil komunikáciu za anomálnu. Detekované intervaly, ktoré predstavujem však obsahujú len približne 3.71% šesťdesiat sekundových okien celej skúmanej komunikácie. Tie je možné vidieť v pravom hornom rohu zobrazeného grafu. Zvyšných 2.17% okien však model tiež označil za anomálne. Dôvodom je jednoduchá filtrácia *false-positive* hodnôt, ktoré som k modelu po jeho vyhodnotení doplnil. Už pri testovacej sade nebol model presný na 100%, čo bolo spôsobené lokálnymi extrémami v spracovávaných dátach, ktoré model označil za anomálne už vo fáze validácie. K podobným extrémom tak dochádza aj počas časti s korektnou komunikáciou v súbore obsahujúcom akýkoľvek útok. K ich dodatočnému odfiltrovaniu však postačí predpoklad, že žiadny vedený útok nebude mať dĺžku trvania kratšiu ako je veľkosť časového okna. Na ich odfiltrovanie z výslednej detekcie tak postačí ignorovať detekované intervaly, ktoré spolu netvoria súvislý časový úsek väčší ako je veľkosť použitého časového okna (šesťdesiat sekúnd). Všetky takto ignorované detekcie sa v každom grafu modelu nachádzajú v blízkosti rozhodovacej funkcie a miesto ich výskytu je v približne tej istej časti ako tomu bolo pri validácii modelu.



Obr. 5.10: One-Class SVM, útok Replay na komunikáciu (RTU->HMI)

Detekované intervaly anomálií na zariadení RTU (obrázok 5.11) boli <44193s, 46053s>, <46353s, 49893s> a <50193s, 52593s>. Medzi intervalmi je tristo sekundová nespojitosť, v ktorej sa nenachádza žiadny z efektov útoku. Detekcia je tak podstatne presnejšia ako tomu bolo pri metóde Three-Sigma. Model je tak očividne schopný pracovať aj s menšími výkyvmi v rozdieloch sledovaných dát a nezvyšovanie tolerancie k extrémom tak bol dobrý krok za cenu menšej presnosti na malých nespojitých intervaloch (false-positive). Presnosť

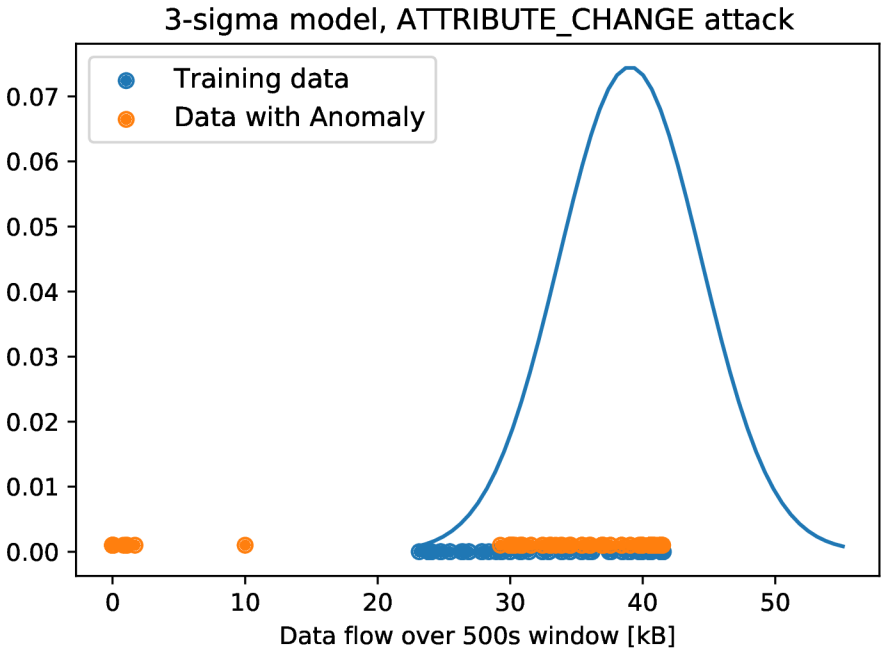
modelu na skúmanej komunikácii bola tento krát 88.53%, čo síce vyzerá ako vážnejší pokles v jeho presnosti, avšak pre jej výpočet pri detekcii je použitý na celé dáta pozitívny ground truth. Detekcia modelu je tak síce korektná no percentuálna presnosť je znížená o časť detekovanej anomálie, ktorá je v tomto prípade 4.45%. Anomália v súbore bola detekovaná korektne a model má tak na skúmaných dátach teoretickú presnosť 92.98%, čo je len mierny pokles oproti presnosti na validačných dátach.



Obr. 5.11: One-Class SVM, útok Replay na komunikáciu (HMI->RTU)

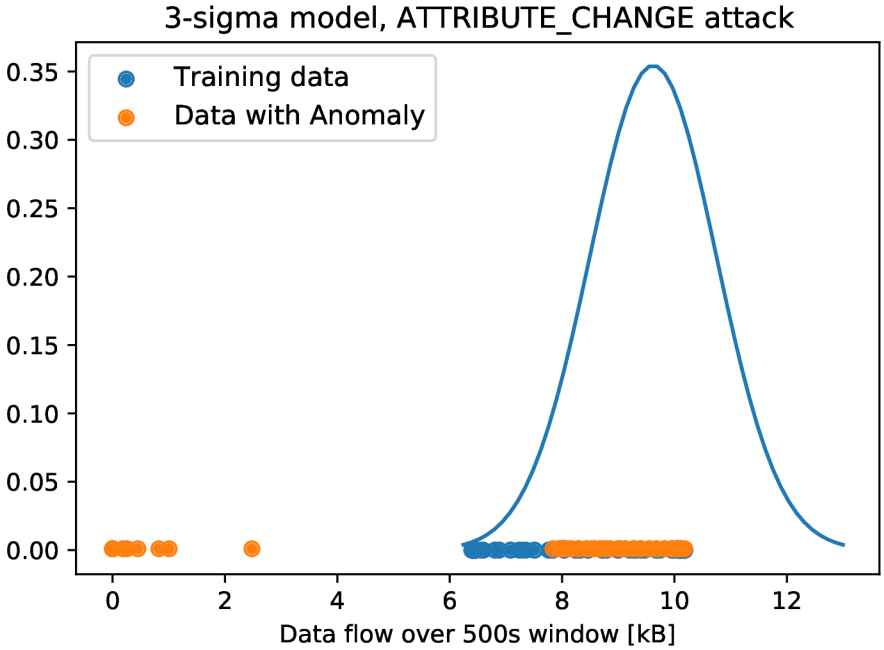
5.4.2 Úprava parametrov (Modify parameter)

Útok v práci emulujem na obe strany komunikácie v troch rôznych scenároch, keďže jeho možnosti sú pomerne rozsiahle. Jeho menej nápadné scenáre však upravujú interné informácie konkrétnych IOA. Bohužiaľ sa extrakcia týchto údajov v použitom skripte nenachádza. Rovnako tak som neprišiel na ďalšiu možnosť extrahovať tieto dáta z PCAP súboru do akéhokoľvek rozumného formátu, čo znemožnilo zahrnúť detekciu týchto scenárov v mojej práci z časových dôvodov. K ich detekcii by pre model *One-Class SVM* by postačovalo upraviť trénovací parameter na hodnoty prenášaných IOA. Pre detekciu som však mohol zahrnúť nápadnejší scenár zmeny hodnoty samotnej IOA, ktorým opäť dôjde k zmene veľkosti prenášanej komunikácie, keďže naň zariadenie RTU reaguje oznamovacou správou o neznámom použitom parametre.



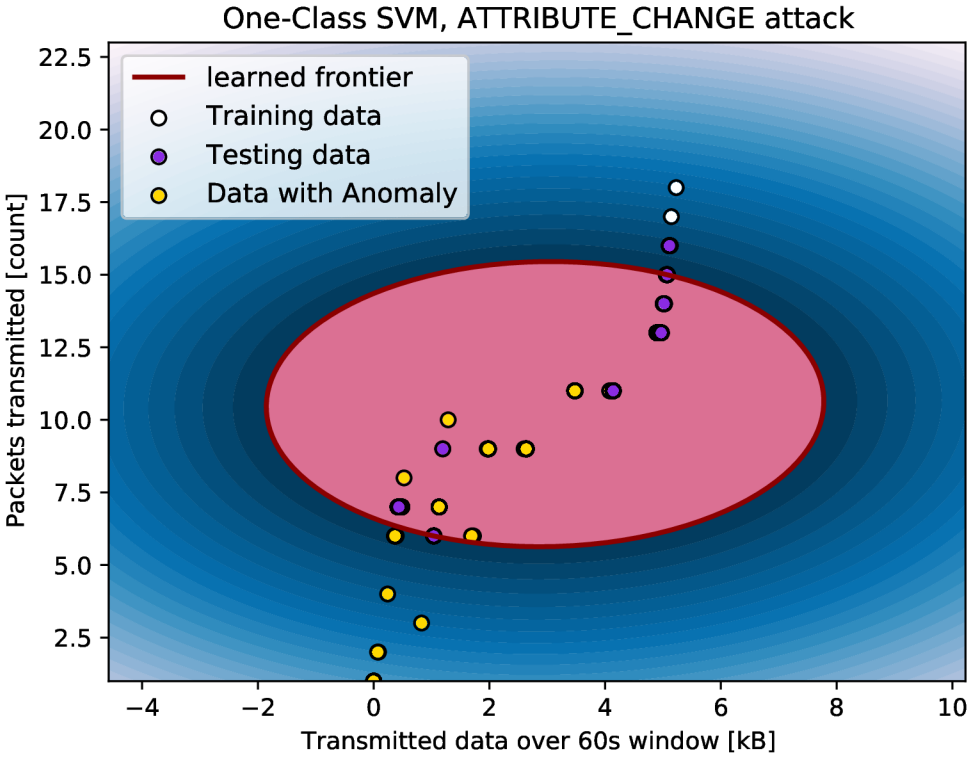
Obr. 5.12: Three-Sigma, útok Modify parameter na komunikáciu (RTU->HMI)

Anomália generovaná útokom bola metódou *Three-Sigma* na zariadení HMI detekovaná v intervale <74000s, 81000s>, v ktorom bol útok približne vedený. Pri detekcii bolo detekované drastické zmenšenie obdržaných dát zo zariadenia RTU oproti štandardnej komunikácii, ktorú je možno vidieť na ľavej strane grafu v obrázku 5.12.



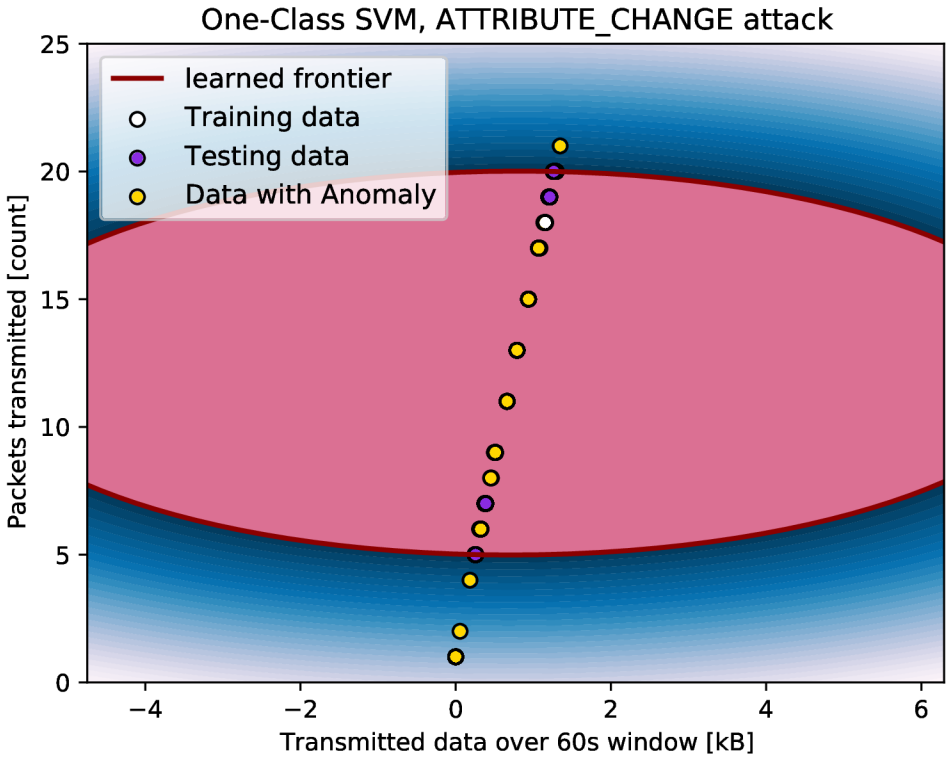
Obr. 5.13: Three-Sigma, útok Modify parameter na komunikáciu (HMI->RTU)

Aj keď sa môže zdať, že pre tento scenár nebude detekcia anomálie na zariadení RTU z príchodzej komunikácie možná, tak opak je pravdou (obrázok 5.13). Útok má totiž ešte ďalší efekt na veľkosť prenášanej komunikácie práve pri scenáre nápadnej zmeny parametru. Po korektnej odpovedi zo zariadenia RTU totiž pri štandardnej komunikácii zariadenie HMI odpovedá potvrdzovacím APCI, ktorým oznamuje zariadeniu RTU korektné obdržanie požadovaných ASDU. Keďže napadnutým paketom je žiadosť zariadenia HMI o aktuálne hodnoty prostriedkov zariadenia RTU, tak po ich neobdržaní zariadenie HMI toto potvrdzovacie APCI neodošle, čo sa prejaví v znížení veľkosti komunikácie obdržanej na zariadení RTU. Detekovaný interval anomálie je rovnaký ako na zariadení HMI, čo je spôsobené identifikáciou toho istého útoku v kombinácii s rovnakým reálnym štartovacím časom odchytávania komunikácie a rovnakou veľkosťou použitého okna pri detekcii.



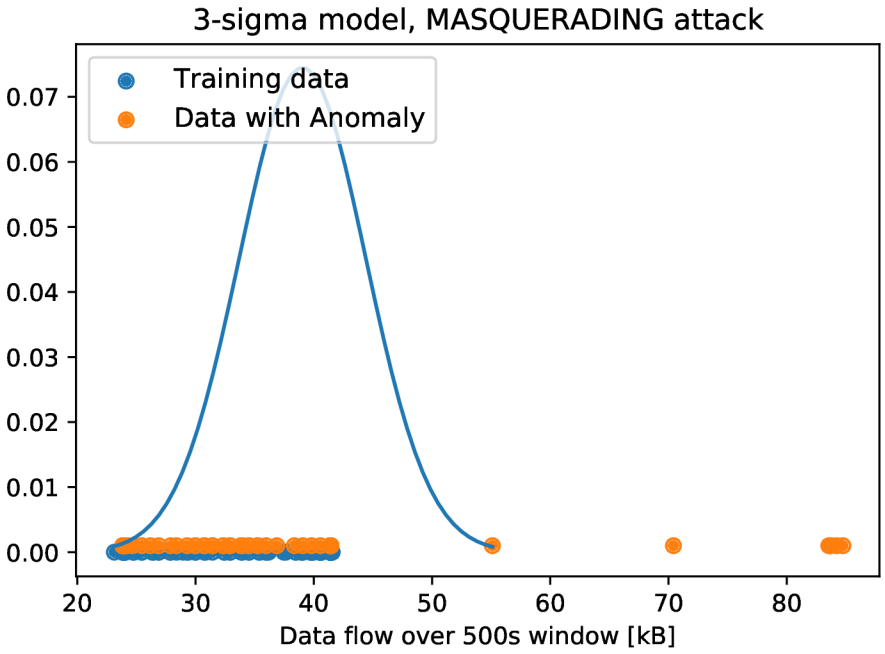
Obr. 5.14: One-Class SVM, útok Modify parameter na komunikáciu (RTU->HMI)

Modelom *One-Class SVM* bola detekcia na zariadení HMI (obrázok 5.14) opäť o niečo presnejšia na intervaloch <74160s, 75360s>, <75360s, 78960s> a <79080s, 80880s>. Na zariadení RTU (obrázok 5.15) však došlo k rozdeleniu prostredného intervalu na <75360s, 77640s> a <77760s, 78960s>, ktoré tak vytvorilo ďalšie okno označujúce korektnú komunikáciu. Nejedná sa však o *false-negative* detekciu. Okno bolo zrejme spôsobené krátkou stratou spojenia zariadenia, ktoré na komunikáciu útočilo, pretože sa v intervale skutočne nachádza nezasiahnutý paket. Na zariadení HMI nebol tento fakt detekovaný, pretože zvýšenie hodnoty v intervale o jeden paket neposunulo hodnotu prenášaných dát cez rozhodovaciu hranicu. Na zariadení RTU sú však aj v štandardnej komunikácii prijímané pakety len s malou veľkosťou (menšia rozmanitosť). Obdržanie korektnej správy tak posunulo daný interval tesne za rozhodovaciu funkciu do oblasti korektnej komunikácie. Namiesto piatich paketov bolo obdržaných sedem čo tiež mierne zvýšilo celkovú veľkosť komunikácie v okne.



Obr. 5.15: One-Class SVM, útok Modify parameter na komunikáciu (HMI->RTU)

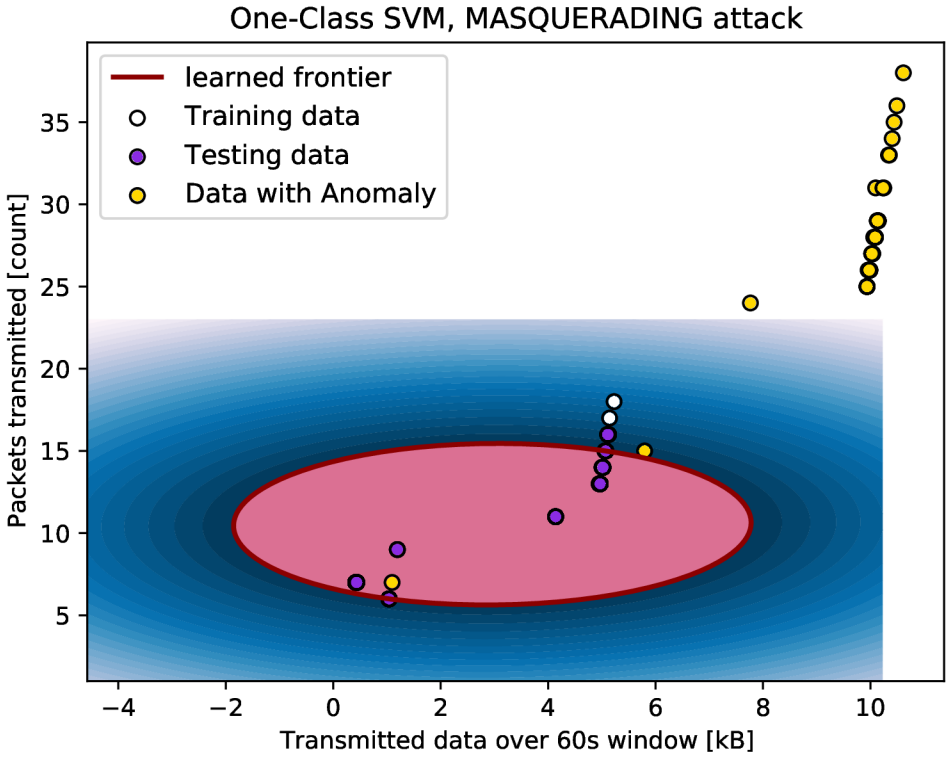
5.4.3 Maskovanie (Masquerading)



Obr. 5.16: Three-Sigma, útok Masquerading (nové HMI zariadenie)

Efekt emulovaného útoku *Maskovanie* je modelmi možné odhaliť len na zariadení RTU. Efektom viditeľným na zariadení HMI je totiž opäť zmena parametrov pod IOA, ktoré sa mi nepodarilo z odchytenej komunikácie extrahovať.

Interval anomálie detekovaný metódou *Three-Sigma* je <58023s, 61023s>. Hodnota medzi 50kB a 60kB (obrázok 5.16) sa dá považovať za *false-negative* detekciu, ktorá je spôsobená ukončením útoku počas odpovedajúceho intervalu. Útok síce generuje podstatne väčšiu veľkosť obdržaných dát na zariadení RTU, avšak v dotyčnom intervale bol vedený len približne pätinu časového okna. Opačným prípadom je hodnota v okolí 70kB. V odpovedajúcom intervale totiž útok začínal približne po prvých sto sekundách. Nejedná sa tak v ani jednom prípade o časť, ktorá by obsahovala zníženie veľkosti prenášaných dát prislúchajúcej štandardnej komunikácii.



Obr. 5.17: One-Class SVM, útok Masquerading (nové HMI zariadenie)

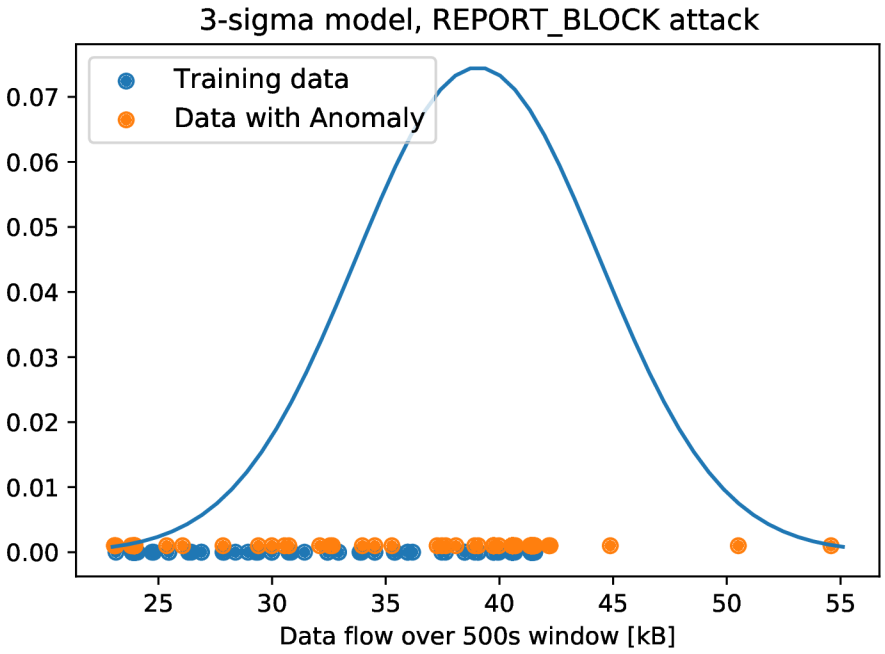
Pozitívna detekcia modelom *One-Class SVM* bola opäť určená presnejším intervalom <58163s, 61163s>. Ani táto metóda sa však nevyhla dvom rozdielnym detekciám patriacim začiatku a koncu útoku. Útok ako taký však generuje podstatne väčšie množstvo dát ako ostatné, pričom hodnoty generovaných dát približne odpovedajú spojeniu s legitímnym zariadením. Posun v grafe je samozrejme spôsobený väčším počtom pripojených zariadení v reálnom čase.

5.4.4 Blokácia ohlasovacích správ (Block report message)

Pre detekciu tohto útoku som musel použiť iné parametre komunikácie, keďže medzi jeho efekty nepatrí úprava veľkosti prenášaných dát vo forme, v ktorej som ho v práci emuloval. Útok je možné detekovať kombináciou hodnôt typu prenášaných ASDU a ich počtu v ča-

sovom okne, Pričom jeho detekcia je sťažená veľmi malou zmenou hodnoty ASDU typu v paketoch, na ktoré bol útok cielený. V decimálnej reprezentácii sa jedná o zmenu jednotky na trojku, čo v porovnaní s ostatnými hodnotami z intervalu <5, 100> predstavuje veľmi malú zmenu v prípade, že by model nepočítal so samotnými ASDU ale len ich reprezentáciou pomocou niektorej logickej operácie v rámci paketu v ktorom sa vyskytli. Na obrázku 5.19 je tak možno vidieť podstatne väčší rozptyl dát ako pri iných útokoch, ktorý je z časti spôsobený väčším objemom použitých dát.

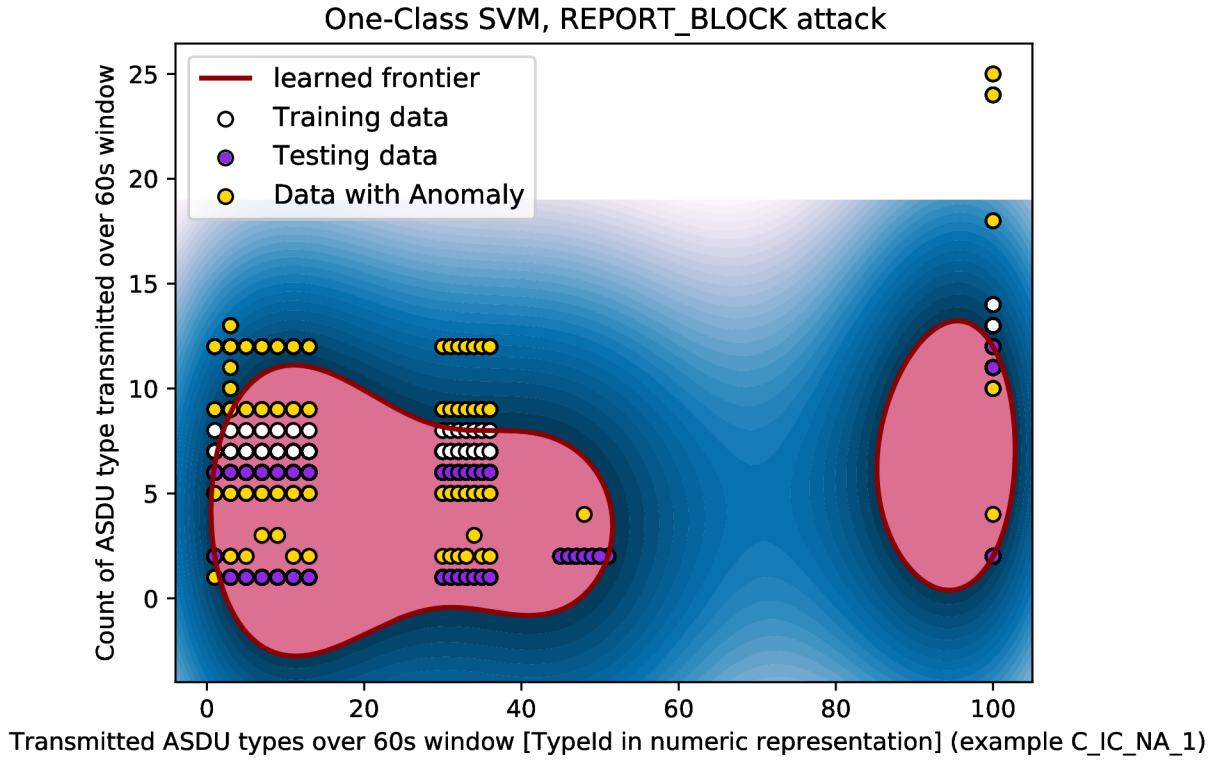
Ani tento zvolený prístup však nebol pre metódu *Three-Sigma* validný, pretože sama metóda nedokáže pracovať s viacerými parametrami naraz a tak malá zmena sa aj pri zvýšenom časovom okne nikdy nedostala za hranicu 3σ od aritmetického priemeru hodnôt ASDU v decimálnej reprezentácii. Pre jednotnosť a zaujímavosť však prezentujem graf na obrázku 5.18, ktorý prezentuje veľkosť prenesených dát v jednotlivých časových oknách. Oproti predchádzajúcim útokom sú na ňom zobrazené mierne zvýšené hodnoty v určitých časových oknách komunikácie. V odchytenom súbore sa totiž objavujú menšie anomálie, ktoré sú pravdepodobne spôsobené smerovacím zariadením. Jedná sa totiž o hodnoty, ktoré reflektujú výskyt opakovaného prenosu dát. Tento prenos však nie je spôsobený žiadnym útokom na komunikáciu. K ich výskytu došlo z dôvodu oneskoreného doručenia.



Obr. 5.18: Three-Sigma, útok Block report message na komunikáciu (RTU->HMI)

Intervalov detekovaných metódou *One-Class SVM* bolo v tomto prípade oveľa viac ako pri iných útokoch a nie všetky v komunikácii detekovali emulovaný útok. Detekcia anomálie v sieti bola však pri každom opodstatnená. Okrem tých, ktoré správne označili približný čas útoku v intervale <36902s, 44822s> sa pri vyhodnotení objavili ďalšie dva nespojité intervaly: <84722s, 85142s> a <86102s, 86222s>. V tomto čase sa totiž v komunikácii začala objavovať anomália, ktorá bola tvorená korektnými odpoveďami zo zariadenia RTU. Túto správu však zariadenie v daných časoch neposielalo a pripisujem ich staršiemu smerovaciemu zariadeniu, ktoré som pri práci používal. Tieto správy sa totiž začali objavovať mimo času

kedy k nim dochádza pri štandardnej komunikácii zariadení a zároveň v odchytených dátach chýba požiadavka na ich obdržanie, ktorá je nutnosťou. Bez nej zariadenie RTU nemá dôvod vygenerovania odpovede a na samotných virtualizovaných zariadeniach by tak automaticky dochádzalo k pádu aplikácie. Časť týchto chýb môže byť spojená so samotným odchytom a uložením odchytenej komunikácie. Tieto anomálie je možné vidieť na obrázku 5.19, kde sú ich predstaviteľmi bežne používané hodnoty ASDU o počte dvanásť za časové okno (generujú vodorovnú čiaru nad rozhodovacou funkciou).



Obr. 5.19: One-Class SVM, útok Block report message na komunikáciu (RTU->HMI)

K podobným chybám došlo v súbore aj počas útoku, čo v tom čase však spôsobilo opačný efekt a rozdelilo dva intervaly na dvojice. Dogenerované pakety tak spätne zvýšili počty ASDU v dvoch šesťdesiat sekundových intervaloch. Reálny efekt útoku, o ktorého detekciu som sa snažil je však možno vyčítať z bodov pre hodnoty ASDU (1 a 3) v ľavom hornom a spodnom rohu. Pri hodnote tri došlo k nárastu oproti hodnote jedna, ktorá sa dostala takmer k počtu nulových výskytov v detekovaných intervaloch.

5.5 Zhrnutie

V kapitole som sa venoval implementácii detekcii anomálii pomocou štatistických metód *Three-Sigma* a *One-Class SVM*. Jedným z výstupov kapitoly je tak samotná implementácia pre konkrétne scenáre útokov, pomocou ktorej sa mi podarilo detekovať anomálie v komunikácii protokolu IEC 104. Výsledky jednotlivých detekcií sú zhrnuté v tabuľke 5.1. Detekcia anomálií v datasete emulovaných útokov bola použitými metódami pomerne presná

a odchýlky v nich boli viazané na zvolené časové okno, ktoré som používal na rozčlenenie jednotlivých načítaných parametrov.

Útok	Interval útoku[s]	Metóda	Zariadenie	Detekcia[s]
Replay	HMI 85142-89402 RTU 44201-52561	Three-Sigma	HMI	85000-88000 88500-89500
			RTU	44033-52033
		One-Class SVM	HMI	85140-88260 88500-89500
			RTU	44193-46053 46353-49893 50193-52593
Modify parameter	74122-80972	Three-Sigma	HMI	74000-81000
			RTU	74000-81000
		One-Class SVM	HMI	74160-75180 75360-78960 79080-80880
			RTU	74160-75180 75360-77640 77760-78960 79080-80880
Masquerading	58197-61233	Three-Sigma	RTU	58023-61023
		One-Class SVM	RTU	58163-61163
Block report message	36903-44804	Three-Sigma	HMI	N/A
		One-Class SVM	HMI	36902-38402 38642-39782 39842-41702 41762-42242 42482-42622 42902-43502 43682-44342 44522-44822

Zariadenie - Detekcia z komunikácie odchytenej na zariadení

Tabuľka 5.1: Súhrn detekovaných anomálií

Metóda *Three-Sigma* má presnosť úzko spojenú s rozdelením hodnôt vo vstupnom súbore a dokáže pracovať s vyhodnocovaním iba na základe jedného parametra. Pri detekcii emulovaných útokov boli jej výsledky pomerne presné, avšak pri jej použití je nutné nastaviť časové okno na základe samotnej štandardnej komunikácie, ktorá môže mať iné časové rozdiely medzi jednotlivými paketmi či zhlukmi paketov. Korektná detekcia je tiež možná iba v prípade, že útok dostatočne upraví samotnú prenášanú komunikáciu. Malé odchýlky v dátach sú pri tejto metóde nedetekovateľné. To je spôsobené samotným určením metódy, ktorá dosahuje najpresnejšie výsledky v prípade normálneho rozdelenia vstupného súboru dát. Bežná štandardná komunikácia sa však málokedy podobá normálnemu rozdeleniu, čo detekciu metódou sťažuje a malé zmeny sa tak nikdy nedostanú za hodnotu 3σ od aritmetického priemeru hodnôt. Tieto fakty značia, že metóda nie je príliš praktická pre nasadenie

a môže skôr slúžiť ako doplnková pre agresívne typy útokov, ktoré generujú skutočne nadmerné množstvo dát.

One-Class SVM na druhú stranu dokáže pracovať s viacerými parametrami komunikácie naraz. Pre detekciu som síce tiež používal časové okno o veľkosti šesťdesiatich sekúnd, avšak nebolo s ním pri rôznych útokoch nutné manipulovať. Viaceré parametre v kombinácii so strojovým učením bez učiteľa umožňujú tomuto modelu naučiť sa rozhodovaciu hranicu v v skúmaných dátach priamo zo štandardnej komunikácie bez ľudského zásahu. Rozhodovacia funkcia v dvoch dimenziách tiež umožňuje metódu naučiť správne klasifikovať všetky časti v štandardnej komunikácii. Pri metóde *Three-Sigma* som bol nútený použiť časové okno o veľkosti päťsto (dvetisíc pri jednom útoku) sekúnd, pretože len vtedy došlo k správnemu zmiešaniu jednotlivých úsekov cyklickej komunikácie s väčšou a menšou hodnotou prenesených dát. Toto pri metóde *One-Class SVM* nie je nutné, keďže pre správnu funkcionalitu nepotrebuje na vstupe hodnoty blízke normálnemu rozdeleniu. Rozdiel medzi výsledkami metód je najviac vidieť v časti 5.4.4, ktorá sa týka menej nápadného útoku. Anomália tak bola odhalená aj z veľmi malej zmeny v hodnote konkrétneho ASDU, čo by metóda *Three-Sigma* nedokázala detekovať ani v prípade, že by komunikácia v čase veľkosťou plne odpovedala normálnemu rozdeleniu. Rovnako tak je pri odhaľovaní každej anomálie možno vidieť, že metóda *One-Class SVM* skutočne odhalí časti, v ktorých sa prejavili efekty jednotlivých útokov. Takto vyčlenené časti sa pri detekcii prejavili nespojitosťou intervalov v mieste ich výskytu (približne dvesto sekundová oblasť).

Kapitola 6

Záver

ICS/SCAD patria medzi celosvetovo používané systémy v priemyselných budovách. Využívané sú na kontrolu priemyselných automatizovaných strojov, pričom k ich správe je využívaná sieťová komunikácia pomocou špecializovaných ICS protokolov. Bezpečnosť týchto riadiacich sietí je patrí medzi dôležité sféry, keďže efekty úspešných útokov môžu v reálnom svete spôsobiť vážne komplikácie. Niektoré komunikačné protokoly používané v týchto sieťach však majú svoje zraniteľnosti, ktoré môžu byť útočníkom využité v prípade prekonania zabezpečenia prístupu do siete.

6.1 Výsledky práce

V práci som sa zameral práve na možnosti útokov na riadiace procesy priemyselných sietí ICS/SCADA. Pre oboznámenie sa s možnosťami útokov mi poslúžila databáza MITRE, ktorej popis sa nachádza v kapitol 2. Pre doplnenie znalostí využitia jednotlivých útokov som bližšie skúmal niekoľko datasetov (kapitola 3) za účelom zistenia ich prevedenia a efektov na protokoly.

Väčšiu časť práce som venoval implementácii vhodných útokov na virtuálne zariadenia, ktoré používajú na komunikáciu protokol IEC 104. Výstupom tejto časti boli datasety štandardnej komunikácie a komunikácie doplnenej o emulované útoky na virtualizovaných zariadeniach. Bližší popis týchto mnou upravených virtuálnych zariadení sa nachádza v technickej správe [7]. Na emuláciu jednotlivých útokov som použil voľne dostupné nástroje a vlastnú implementáciu vo forme pluginov pre nástroj Ettercap. Jednotlivé útoky sú bližšie popísané v kapitole 4 a použitie vytvorených pluginov je popísané v prílohách práce. Pri výbere som sa tiež zameral na viditeľnosť útokov v samotnej komunikácii protokolu a vynechal tak útoky, ktoré používajú zraniteľnosti na samotných zariadeniach. Dôvodom na to bolo jednak zameranie práce na útoky na konkrétny protokol priemyselnej komunikácie a využitie virtuálnych zariadení. Zariadenia použité na virtualizáciu (Raspberry Pi) neodpovedajú reálnym ICS zariadeniam. Využitie zraniteľnosti na fyzických zariadeniach by tak pre prácu neprinieslo relevantné výsledky.

V poslednej časti som sa venoval implementácii zvolených štatistických metód pre detekciu anomálií vo vytvorenom datasete pre emulované útoky. Popis metód spolu s vyhodnotením presnosti a použiteľnosti sa nachádza v kapitole 5. Táto kapitola tvorí ďalší z výstupov práce, ktorým je práve zhodnotenie presnosti zvolených metód v kombinácii s voľbami ich parametrov pre detekciu anomálií v komunikácii protokolu IEC 104. Na detekciu som vytvoril skript v programovacom jazyku *python*, ktorý však pracuje na základe

vygenerovaných CSV súborov štandardnej a anomálnej komunikácie. Jeho použitie na ďalšom datasete protokolu IEC 104 je samozrejme možné a postačí tak premenovať vstupné súbory.

Literatúra

[1] Dube, D. a Camerini, J. *MODBUS Application Protocol*. Internet-Draft draft-dube-modbus-applproto-00. Internet Engineering Task Force.

[2] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L. et al. *Hypertext Transfer Protocol* [Hypertext Transfer Protocol – HTTP/1.1]. 2616. Jún 1999. 1-176 s.

[3] Frazão, I., Abreu, P., Cruz, T., Araújo, H. a Simões, P. Cyber-security Modbus ICS dataset. In:. IEEE Dataport, [Navštívené: 25. November 2021]. Dostupné z: https://dx.doi.org/10.21227/pjff-1a03.

[4] Lamrini, B., Gjini, A., Daudin, S., Armando, F., Pratmarty, P. et al. Anomaly Detection Using Similarity-based One-Class SVM for Network Traffic Characterization. In:. August 2018.

[5] Lonvick, C. M. a Ylonen, T. *The Secure Shell (SSH) Connection Protocol* [The Secure Shell (SSH) Connection Protocol]. 4254. Január 2006.

[6] Matoušek, P. *Description and analysis of IEC 104 Protocol*. 2017. 38 s.

[7] Matoušek, P. a Grofčík, P. *ICS Virtual Testbed*. 2021. 36 s.

[8] Maynard, P., McLaughlin, K. a Sezer, S. An Open Framework for Deploying Experimental SCADA Testbed Networks. In:. 5th International Symposium for ICS & SCADA Cyber Security Research, 2018.

[9] Mockapetris, P. *Domain names - implementation and specification* [Domain names - implementation and specification]. 1035. November 1987. 1-54 s.

[10] Morris, T. a Gao, W. Industrial Control System (ICS) Cyber Attack Datasets. In:. [Navštívené: 15. November 2021]. Dostupné z: https://sites.google.com/a/uah.edu/tommy-morris-uah/ics-data-sets?authuser=0.

[11] Plummer, D. C. *An Ethernet Address Resolution Protocol* [An Ethernet Address Resolution Protocol]. 826. November 1982.

[12] Postel, J. *Internet Control Message Protocol* [Internet Control Message Protocol]. 792. September 1981. 1-21 s.

[13] Postel, J. *Internet Protocol* [Internet Protocol]. 791. September 1981. 1-45 s.

[14] Postel, J. *Transmission Control Protocol* [Transmission Control Protocol]. 793. September 1981. 1-85 s.

[15] Pukelsheim, F. *The Three Sigma Rule*. 2. 1994. 88-91 s.

[16] Rodofile, N. R., Radke, K. a Foo, E. Real-Time and Interactive Attacks on DNP3. In:. [Navštívené: 1. December 2021]. Dostupné z: https://github.com/qut-infosec/2017QUT_DNP3.

[17] Sivabalan, S., Boutros, S., Shah, H. C., Aldrin, S. a Venkatesan, M. *Media Access Control (MAC) Address Withdrawal over Static Pseudowire* [Media Access Control (MAC) Address Withdrawal over Static Pseudowire]. 7769. Február 2016. 1-10 s.

[18] Transmission, I. P. . E. S., Committee, D., Committee, I. P. . E. S. S., Electrical, I. of, Engineers, E. et al. *1815-2012 - IEEE Standard for Electric Power Systems Communications-Distributed Network Protocol (DNP3)*. IEEE.

Príloha A

Inštalácia nástroja Ettercap (plugin import)

Príkaz na jednoduchú inštaláciu potrebných závislostí: Command to easily install required dependencies on debian or debian based distro:

- apt-get install build-essential debhelper bison check cmake flex groff libbsd-dev libcurl4-openssl-dev libgeoip-dev libgtk-3-dev libltdl-dev libluajit-5.1-dev libncurses5-dev libnet1-dev libpcap-dev libpcre3-dev libssl-dev

A.1 Požadované programy

- c - prekladač
- flex - lex-kompatibilný analyzátor pre *.l súbory
- bison - yacc-kompatabilný analyzátor pre *.y súbory +
- cmake

A.2 Požadované knižnice

- libpcap >= 0.8.1
- libpcap >= 0.8.1
- libnet >= 1.1.2.1 (>= 1.1.5 for IPv6 support)
- openssl >= 0.9.7
- libpthread
- zlib
- libgeoip
- CMake 2.8
- Curl >= 7.26.0 to build SSLStrip plugin

A.3 Inštalácia

1. git clone https://github.com/Ettercap/ettercap.git
2. cd ettercap
3. copy - požadované (vytvorené) plug-in do plug-ins priečinku
4. mkdir build
5. cmake ..
6. sudo make install

Ettercap aplikácia sa objavý v systéme s predinštalovanými základnými, ale aj novo vytvorenými pluginmy v momente kedy inštalácia prebehne v poriadku. Pridané pluginy sú využitelné z grafického rozhrania aplikácie, ale aj z konzoly použitím špecifických názvov pluginov nalinkovaných do Ettercap programu.

Ako príklad, príkaz:

- *sudo ettercap -T -q -i wlp3s0 -P COT_change_104 -M ARP /192.168.1.100/ /192.168.1.102/"*

spustí MITM útok medzi dvoma zariadeniami v lokálnej sieti s použitím pluginu *COT_change_104*.

Príloha B

DOS SYN-flood

Táto príloha je venovaná krátkemu popisu (README) na spustenie jednoduchého *SYN-flood* útoku pomocou vytvoreného programu v jazyku C++ a RAW socketov.

./dosattack [-d zariadenie] [-i IP adresa] [-p port]

Pre útok je minimálne potrebný parameter pre IP adresu a port. Zariadenie nie je nutné špecifikovať, avšak v prípade jeho použitia je parameter pre IP adresu ignorovaný. Pri jeho použití dochádza k útoku na IP adresu špecifikovaného zariadenia podľa obrázka 4.1.

- -d zariadenie, špecifikuje zariadenie HMI alebo RTU zo štandardnej komunikácie v kapitole 4:
  - 1 - HMI
  - 2 - RTU
- -i IP adresa, špecifikuje IPv4 adresu zariadenia
- -p port, špecifikuje port služby (typu TCP)

Príloha C

Skript na detekciu

Príloha je venovaná krátkemu popisu k skriptu vytvorenému pre detekciu anomálií v odchytených datasetoch v jazyku *python*.

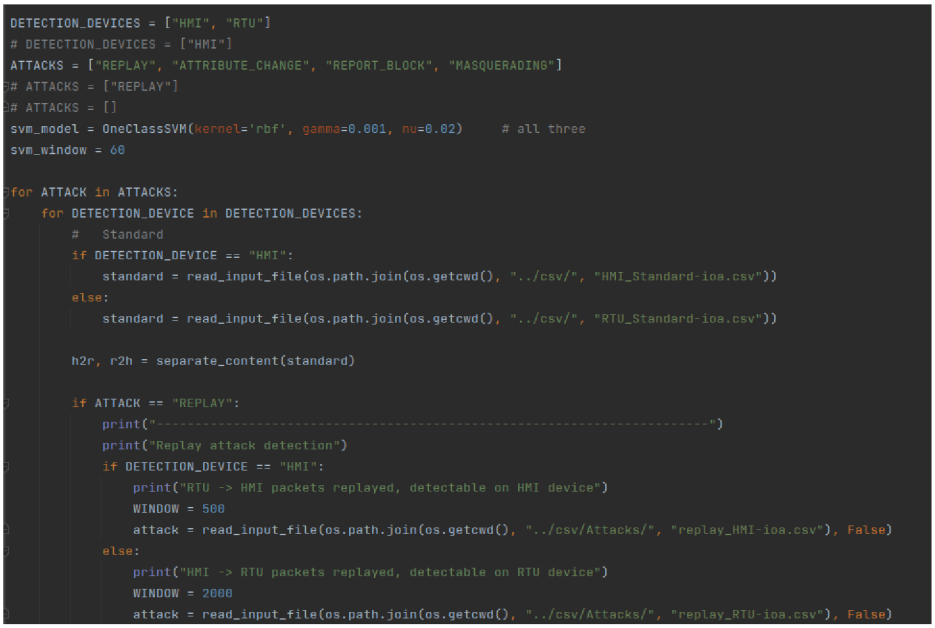
Pre detekciu útokov emulovaných v práci postačí utilitu spustiť ako bežný program v jazyku python (ideálne verzie 3.6).

python3.6 statistical_detection.py

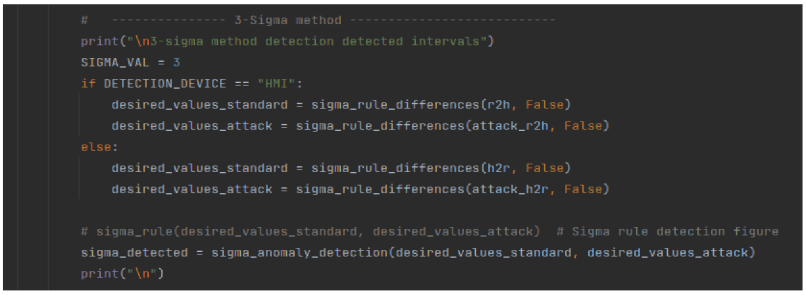
Skript však predpokladá dostupnosť CSV súborov, ktoré sú súčasťou odovzdaného balíka presne v odovzdanom nadadresáry. Mimo základné knižnice je nutné mať doinštalované knižnice:

- matplotlib,
- numpy,
- scipy,
- sklearn.

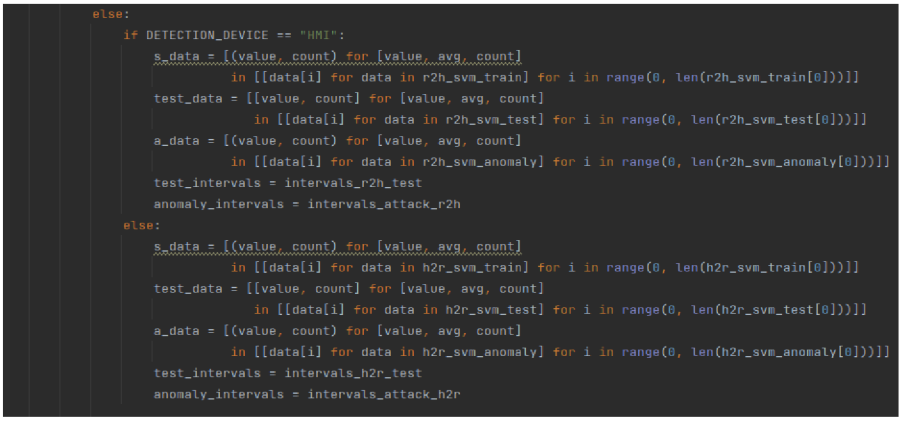
Skript pre účely práce však nebolo potrebné prispôsobiť užívateľskému vstupu a tak v prípade nových súborov je nutné pridať ich načítanie priamo do kódu. V prípade potreby je tiež nutné upraviť na začiatku súboru IP address oboch zariadení pre funkčnú separáciu dvoch strán komunikácie. K detekcii dochádza zo strany príchodzej komunikácie a pre jej prepnutie na odchodziu je potrebné prehodiť podmienku v ďalších dvoch častiach (Three-Sigma a One-Class SVM) na $! = "HMI"$. Call funkcií *sigma_rule* a *SVM_plot* je v hlavnej časti primárne zakomentovaný a v prípade jeho odkomentovania dôjde k vykreslovaniu výsledkov detekcii do grafu použitého pri práci.



Obr. C.1: Oblasť pre doplnenie načítania súborov



Obr. C.2: Oblasť pre doplnenie načítania súborov

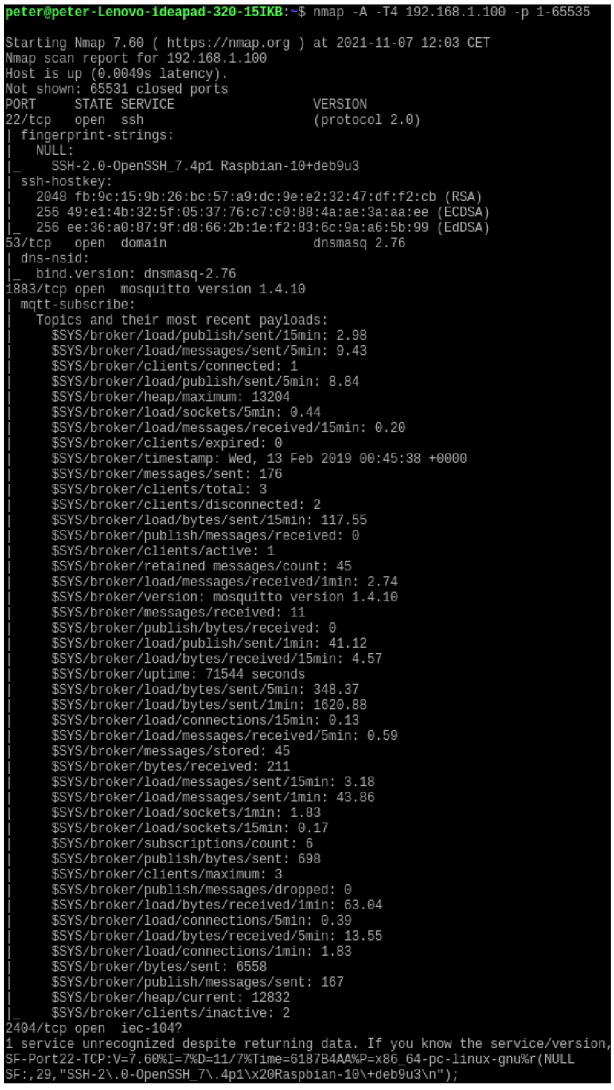


Obr. C.3: Oblasť pre doplnenie načítania súborov

Príloha D

Nadrozmerné obrázky

Táto príloha je všeobecne venovaná nadrozmerným obrázkom či tabuľkám, ktoré sú z hladiska práce podstatné no nepoužiteľné priamo v texte.



Obr. D.1: Vykonaný nmap scan všetkých portov v lokálnej sieti virtuálnych zariadení