VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

ANALÝZA PROTOKOLŮ PRO KOMUNIKACI V ENERGETICKÝCH SÍTÍCH

AN ANALYSIS OF SMART GRID COMMUNICATION PROTOCOLS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. LUKÁŠ SOBOTKA**
AUTHOR

VEDOUCÍ PRÁCE — **Doc. Ing. ONDŘEJ RYŠAVÝ, Ph.D.**
SUPERVISOR

BRNO 2018

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav informačních systémů (UIFS) Akademický rok 2018/2019

Zadání diplomové práce

Student: **Sobotka Lukáš, Bc.**
Program: Informační technologie Obor: Počítačové sítě a komunikace
Název: **Analýza protokolů pro komunikaci v energetických sítích**
An Analysis of Smart Grid Communication Protocols
Kategorie: Počítačové sítě

Zadání:

1. Seznamte se s problematikou a protokoly používanými v komunikaci kritických energetických infrastruktur, především DNP3 a IEC 60870-5-104.
2. Proveďte průzkum dostupných simulačních nástrojů a zhodnoťte jejich využitelnost ve vašem projektu.
3. Navrhněte architekturu a vytvořte simulační prostředí pro experimenty s vybranými protokoly.
4. Po dohodě s vedoucím navrhněte metodu, která bude schopná detekovat vybrané anomálie v komunikaci vybranými protokoly.
5. Vytvořte prototyp navrženého řešení.
6. Navrhněte vhodné testovací úlohy a ověřte funkčnost řešení v rámci vytvořeného simulačního prostředí.
7. Zhodnoťte vlastnosti vytvořeného řešení a navrhněte případné pokračování projektu.

Literatura:

- Clarke, Gordon. Practical Modern SCADA Protocols: DNP3, 60870.5 and Related Systems. Oxford: Newnes, 2004. 544 p. ISBN: 978-0-7506-5799-0.
- Knapp, Eric D. - Langill, Joel T. Industrial Network Security: Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems. Syngress; 2 edition, 2014. 460 p. ISBN: 978-0124201149.
- Radvanovsky, R. - Brodsky, J. Handbook of SCADA/Control Systems Security. CRC Press, 2013. 383 p. ISBN 978-1482209402.
- Garitano I. - Uribeetxeberria R. - Zurutuza U. A Review of SCADA Anomaly Detection Systems. In: Soft Computing Models in Industrial and Environmental Applications, 6th International Conference SOCO 2011. Advances in Intelligent and Soft Computing, vol 87. Springer, Berlin: Heidelberg, 2011. pp 357-366. ISBN 978-3-642-19643-0.

Podrobné závazné pokyny pro vypracování práce viz http://www.fit.vutbr.cz/info/szz/

Vedoucí práce: **Ryšavý Ondřej, doc. Ing., Ph.D.**
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 22. května 2019
Datum schválení: 31. října 2018

Abstrakt

Tato diplomová práce se zabývá bezpečností průmyslových systémů SCADA, které se používají v energetických infrastrukturách. Popisuje architekturu těchto systémů a detailně analyzuje dva komunikační protokoly – DNP3 a IEC 60870-5-104. Dále se v práci rozebírá analýza anomálií a bezpečnostních hrozeb systémů SCADA. Hlavním cílem je navrhnout a implementovat takový systém, který bude schopný detekovat anomálie zachycené na tomto systému. K získání referenčních dat a k testování slouží simulační prostředí daných protokolů.

Abstract

This work deals security of SCADA industry systems which are used in energetic networks. It describes architecture of those systems and also analyze in details two communication protocols – DNP3 and IEC 60870-5-104. Next part is devoted to the analysis of anomaly and security threats which can be happen in SCADA systems. The main goal of this work is design and implementation of system which will be able to detect some of threats or anomalies. Also is necessary to propose simulation environment for testing.

Klíčová slova

monitorování, průmyslové systémy, SCADA, DNP3, IEC 60870-5-104, bezpečnost, detekce hrozeb, energetická síť

Keywords

monitoring, industrial systems, SCADA, DNP3, IEC 60870-5-104, security, threat detection, power grid

Citace

SOBOTKA, Lukáš. *Analýza protokolů pro komunikaci v energetických sítích.* Brno, 2018. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Doc. Ing. Ondřej Ryšavý, Ph.D.

Analýza protokolů pro komunikaci v energetických sítích

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci na téma Analýza protokolů pro komunikaci v energetických sítích vypracoval samostatně pod vedením pana docenta Ondřeje Ryšavého, Ph.D. Další informace mi poskytli zaměstnanci firmy Greycortex. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Lukáš Sobotka
22. května 2019

Poděkování

Rád bych poděkoval svému vedoucímu diplomové práce docentu Ondřeji Ryšavému, Ph.D. zvláště za trpělivost, vedení a podporu, kterou mi poskytl v průběhu tvorby práce.

Obsah

Kapitola 1

Úvod

SCADA systémy se hojně využívají nejen v energetických, ale také v průmyslových sítích například pro přenos informací ze sensorů či ovládání různých vypínačů. Komunikace je založená na protokolech, které nejsou z hlediska historického ani z hlediska samotného návrhu dostatečně zabezpečené proti možným útokům. Tato práce se detailněji věnuje dvěma protokolům – DNP3 a IEC 60870-5-104, které rozebírá, a pro které následně navrhuje metody detekcí možných útoků a anomálií.

Práce začíná popisem SCADA systémů, v rámci kterého je rozebráno, co jsou systémy SCADA a z jakých částí se skládají. V této kapitole je také popsán vývoj napříč jednotlivými generacemi a v neposlední řadě bezpečnostní hrozby a několik úspěšných útoků, které v minulosti byly namířeny proti SCADA sítím. Následuje část, která se věnuje protokolům DNP3 a IEC 60870-5-104 a jejich vnitřním strukturám, které jsou stěžejní pro správné parsování. Další kapitola, která uzavírá teoretickou část, je zaměřená na detekci anomálií. Je hned několik možných kritérií, na základě kterých je možné anomálie dělit – podle charakteru anomálie, metody učení či způsobu hodnocení.

Cílem práce je navrhnout a implementovat systém, který bude schopný pro zmíněné protokoly provádět detekci hrozeb a anomálií. Takový systém je navržený v páté kapitole, kde jsou rozebrány jednotlivé části systému a způsoby, jakým spolu komunikují. V následující kapitole jsou podrobně rozebrané atributy, které budou analyzovány, a konkrétní metody, které byly zvolené pro implementaci detekce anomálií. V poslední části, která se věnuje vyhodnocení testů, je popsána úspěšnost jednotlivých detekčních metod.

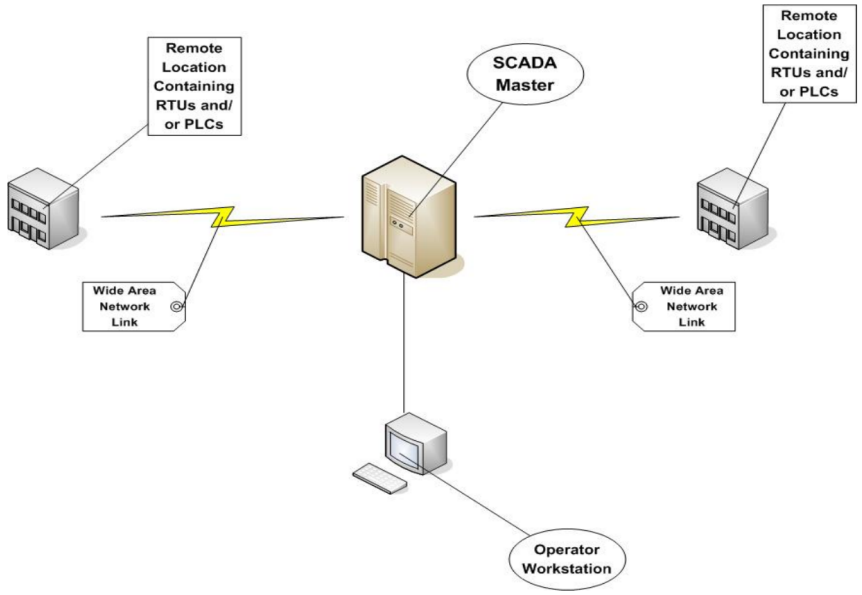
Kapitola 2

Systémy SCADA

SCADA (Supervisory Control And Data Acquisition) jsou systémy pro monitorování a ovládání průmyslových zařízení v reálném čase. Systém zprostředkovává přenos dat mezi čidly v systému, centrálním počítačem a počítačem operátora. Dále přenáší a shromažďuje informace na centrální místo, kde může případně i upozornit na nebezpečí. SCADA systémy se využívají v mnoha oborech jako jsou telekomunikační, energetické, chemické, potravinářské průmysly a mnoha dalších. [20] [10]

2.1 Složení SCADA systémů

Typická architektura SCADA systémů je zobrazena na obrázku 2.1.



Obrázek 2.1: Typické rozložení SCADA systémů [20]

2.1.1 Čidla a akční členi

Zdroj dat poskytují čidla a senzory, mezi které patří například vodoměry, elektroměry, teploměry, tlakoměry atd.

Akčními členy se rozumí zařízení, které reagují na příkazy operátora. Zde se nachází servomotory, rozvaděče, vypínače atd. [20]

2.1.2 RTU a PLC jednotky

Jedná se o samostatné jednotky pro získávání dat a řízení – zprostředkovávají centrálnímu počítači komunikaci s čidly a akčními členy. Typicky obsahují operační systém reálného času, ovladač pro komunikaci s MTU (viz níže 2.1.4) a ovladač pro komunikaci s čidly a akčními členy.

PLC (Programmable Logic Controller) je malý průmyslový počítač, který původně neobsahoval komunikaci. Obsahoval program vykonávající smyčku – načtení dat, provedení operace, odeslání výsledku na výstup.

RTU (Remote Terminal Units) jednotky byly na rozdíl od PLC navržené s podporou komunikace, ale nebyly snadno programovatelné.

Dnes jsou rozdíly mezi RTU a PLC čím dál menší, postupně dochází ke splývání. [10]

2.1.3 Systém komunikace

Komunikační síť slouží k poskytování prostředků, pomocí kterých mohou být data přenášena mezi RTU a MTU jednotkami. Pro komunikaci je možné použít kabel, telefonní linku či rádio. Kabelové propojení je použité ve firmách, kde není třeba překonávat velké vzdálenosti. Pro spojení jednotek na větší vzdálenosti bývá použita telefonní a rádiová komunikace.

Historicky se vytvářely sítě speciálně určené pro systémy SCADA, ale s rozvojem podnikových LAN či WAN počítačových sítí se komunikace postupně integruje do podnikových počítačových sítí. [20]

2.1.4 Master Terminal Unit (MTU)

Centrální server či skupina serverů poskytující rozhraní pro HMI. Server zpracovává informace přicházející z RTU/PLC jednotek, a zároveň jim přeposílá příkazy a zajišťuje data pro HMI, aby byla čitelná pro lidského operátora. Komunikace mezi MTU a HMI probíhá zásadně na LAN/WAN sítích. [20]

2.1.5 Human Machine Interface(HMI)

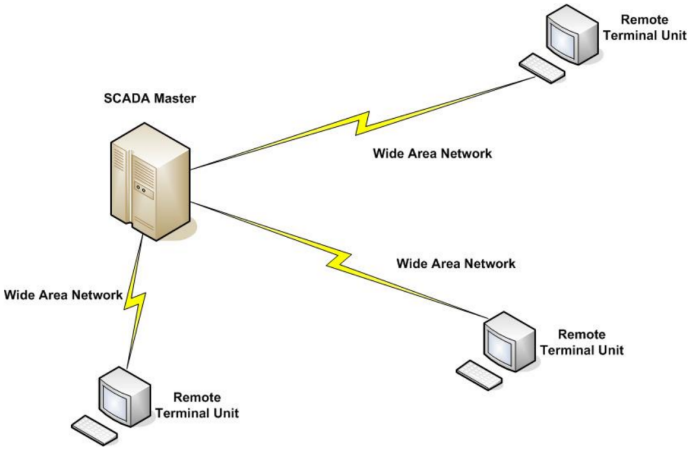
Někdy byl označován také jako MMI (Man Machine Interface). Jedná se o software typicky s grafickým uživatelským rozhraním. Tento software se nachází na pracovní stanici operátora, která je propojena sítí s MTU. Operátorovi zobrazuje informace o stavu procesu a také mu umožňuje zadávat příkazy. Obvykle také zobrazuje trendy – průběhy vybraných veličin. [10]

2.2 Vývoj SCADA systémů

První SCADA systémy vznikaly v 60. letech 20. století. Původní myšlenka zahrnovala pouze průmyslové použití. Systémy byly nákladné na realizaci, provoz i obsluhu. V dnešní době se stávají tyto systémy běžnějšími a proto také i levnějšími. [10]

2.2.1 První generace – ostrovní systémy

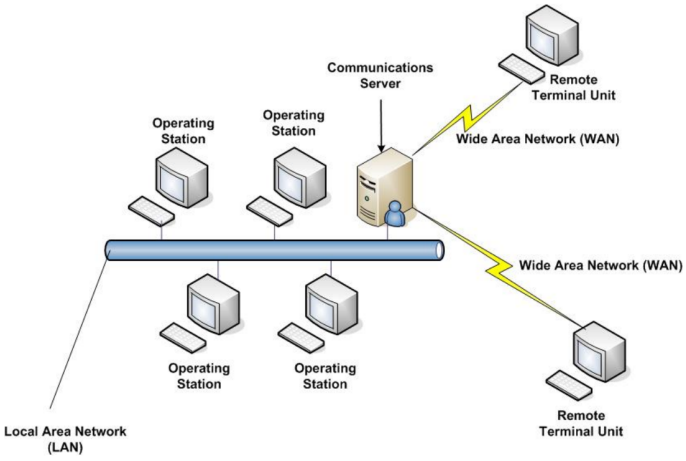
První koncepty SCADA systémů, které lze vidět na obrázku 2.2, byly centralizované, izolované a jednoúčelové. Každý SCADA systém byl samostatný bez žádného propojení s dalšími systémy. Komunikační síť sloužila pouze k účelu komunikace mezi MTU a RTU. Používané protokoly jsou dnes již zastaralé – nízká funkcionalita, vysoká režie. Použité protokoly a software byly proprietární (uzavřené). Systémy byly proto limitované výrobcem. [20] [35]



Obrázek 2.2: Systém SCADA první generace – ostrovní systémy

2.2.2 Druhá generace – distribuované systémy

SCADA systémy druhé generace (viz obrázek 2.3) používaly LAN technologii, díky které mohly probíhat distribuované zpracování informací na vícero stanicích. Každá stanice měla specifickou funkci, informace byly sdíleny s ostatními stanicemi v reálném čase. Zavedením distribuovaného systému se zlepšila i výkonost – celkový výpočetní výkon byl větší než u jednoho procesoru. Snaha o miniaturizaci systému vedla dále k menším, spolehlivějším a levnějším, energeticky úspornější systémům v porování s první generací. Použité protokoly byly většinou proprietární. [20] [35]

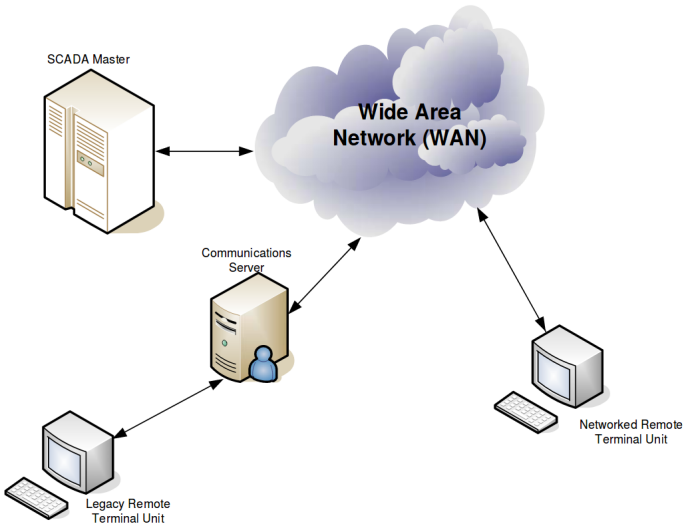


Obrázek 2.3: Systém SCADA druhé generace – distribuované systémy

2.2.3 Třetí generace – síťové systémy

Tato generace, jejíž schéma je na obrázku 2.4, je podobná té předchozí – jsou zde skupiny MTU stanic, které se starají o výpočet, RTU jednotky využívající proprietární protokoly. Ovšem hlavní změnou je důraz na využití otevřených standardů, které využívají IP protokolu. Uživatelsky přívětivější je také přidání „off-the-system“ systému, který zjednodušuje přidání zařízení třetích stran.

Využití IP protokolu snižuje některá omezení oproti předchozím generacím, například dovoluje řídícím stanicím komunikovat přímo s RTU jednotkami – systém je robustní vůči výpadku komunikačního serveru. [35] [20]



Obrázek 2.4: Systém SCADA třetí generace – síťové systémy

2.2.4 Čtvrtá generace – Internet věcí (IoT)

Nejnovější generace SCADA systémů (zobrazena na obrázku 2.5) využívá pro komunikaci technologii IoT a pro výpočetní úlohy komerční cloudové služby. Díky tomu je systém jednodušeji spravovatelný. Také se zlepšila přístupnost dat a dostupnost, zvětšila se flexibilita a škálovatelnost. [35]

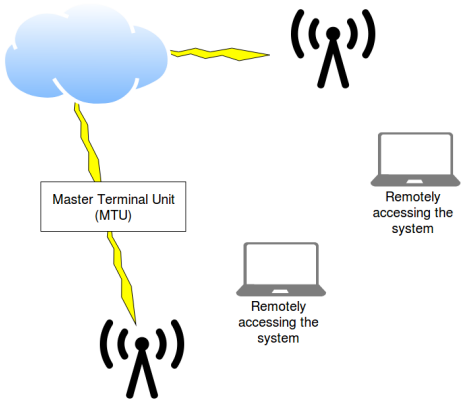
2.3 Bezpečnostní hrozby

První dvě generace SCADA systémů byly odolné proti rozsáhlým útokům – jednalo se o oddělené či uzavřené systémy. S třetí generací, která začala výrazně využívat IP protokol, začalo výrazně přibývat útoků na tyto systémy. [31]

Dalším příčinou oslabení systémů řízení provozu je požadavek na funkčnost v časově citlivém prostředí. Ke splnění tohoto kritéria je třeba, aby byla zařízení jednoduchá, navržená pouze pro plnění požadovaných funkcí. Proto většina zařízení nemá vlastní paměť či potřebný výpočetní výkon k provádění bezpečnostních operací. [32]

Několik dalších omezení řídících systémů v reálném čase: [32]

- slabá či žádná autentizace
- žádné řízení uživatelských účtů



Obrázek 2.5: Systém SCADA čtvrté generace – internet věcí

- absence hierarchie privilegií
- žádný audit přihlašování
- nevykonává se žádný uživatelský program, ale jednoduchá logická operace
- zařízení nejsou navrhovaná s ohledem na bezpečnost
- proprietární protokoly zpomalují integraci bezpečnostních nástrojů

Systémy SCADA čelí mnoha hrozbám, které se dají rozdělit do několika skupin. Mezi ně patří například neoprávněný přístup k řídícímu softwaru (k HMI jednotkám) a to prostřednictvím neoprávněných uživatelů nebo prostřednictvím změn vyvolaných virem či jinými softwarovými hrozbami. [31]

Další skupina hrozeb je založená na přístupu k paketům v síti systému SCADA. Jak bylo již zmíněno, v mnoha případech není protokol zabezpečen žádnou kryptografickou funkcí, která by útočníkovi znemožnila řídit zařízení SCADA odesláním příkazů přes síť. Zde může útočník využít například virtuální privátní sítě (VPN), u kterých uživatelé často mylně předpokládají, že vždy nabízí dostatečnou ochranu. [31]

Tyto hrozby mohou vést k následujícím útokům: [20]

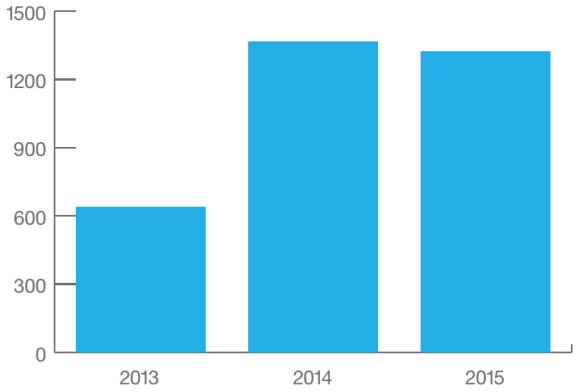
- zachycení logovaných dat – neautorizované získání firemních dat
- změna logovaných dat – nesprávná reakce operátora kvůli podvrženým datům
- smazání či změna dat uložených v databázi – ztráta firemních dat
- logování stisknutých kláves – získání přihlašovacích údajů
- napadnutí systému trojanem – převzetí kontroly nad systémem
- použití DoS útoku – ztráta dat způsobená výpadkem

2.3.1 Proběhlé útoky

Z dostupných dat od společnosti IBM vyplývá, že útoků na průmyslové kontrolní systémy čím dál více přibývá. Z grafu na obrázku 2.6 je patrné, že mezi roky 2013 a 2015 se počet útoků zdvojnásobil. [31] Stoupající tendence dále pokračovala i v roce 2016, kdy došlo k nárůstu útoků dokonce o 110% [6]

Transsibiřské plynovody

Jeden z prvních zaznamenaných útoků se stal roku 1982, kdy způsobil masivní exploze na transsibiřském plynovodu. Výbuch odpovídající síle 3000 tun TNT byl největší nejadernou



Obrázek 2.6: Počet útoků provedených na průmyslové kontrolní systémy mezi lednem 2013 a srpnem 2015 zachycených bezpečností oddělením společnosti IBM [31]

explozí na Zemi, kdy i výsledný požár byl vidět z vesmíru. Příčinou tohoto útoku je neznámý trojan vložený do systému SCADA. [27]

Íránské jaderné centrifugy

Stuxnet reprezentuje jeden z milníků vývoje malware – poprvé v historii využívá čtyř různých zranitelností nultého dne, podvrhuje dva digitální certifikáty, přeprogramovává PLC jednotky a aktivně se skrývá před operátorem. Stuxnet je velmi komplexní a složitý – pro jeho vývoj bylo zapotřebí značné množství zdrojů. Proto se nepředpokládá, že by se výrazně objevily podobné hrozby. Nicméně zdůraznil, že pokusy přímého útoku na kritické infrastruktury jsou reálnou hrozbou. [19]

Samotný útok lze rozdělit do třech fází. V první fázi malware opakovaně napadal a případně se replikoval na stroje se systémem Microsoft Windows. V druhé fázi hledal program Siemens Step 7, což je nástroj pro programování průmyslových kontrolních systémů. V posledním kroku zkompromitoval jednotky PLC, na které nahrál vlastní řídící program aniž by to operátor mohl zaregistrovat. [19]

Podle podrobných výzkumů byl za primární cíl označené zařízení ve městě Natanz provozující centrifugy obohacování uranu íránského jaderného programu. Dalším možným cílem je íránská jaderná elektrárna Búšéhr. Množství útoků také se odhaduje, že hlavním cílem útoků byl Írán. [12]

Kapitola 3

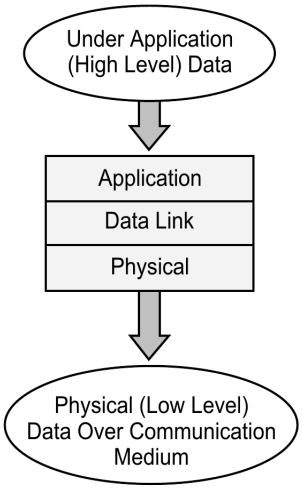
Protokoly

V SCADA systémech lze komunikaci rozdělit do dvou částí. Jednotky RTU přijímají příkazy k řízení akčních členů, nastavují úrovně analogových výstupů a odpovídají na žádosti. Tím poskytují MTU jednotce informace o stavu a hodnotách analogových datech. Tyto informace jsou identifikovány jen na základě jedinečné adresy, z kterých byly poslány.

RTU jednotka nemá informace o tom, které z parametrů jsou reálně monitorovány, pouze monitoruje jednu či více veličin a ukládá získaná data do lokálního úložiště. Teprve až MTU stanice určuje význam jednotlivých hodnot – pro konkrétní RTU jednotku má definované pořadí hodnot a co tyto hodnoty znamenají. [20]

3.1 Enhanced Performance Architecture (EPA)

Analyzované protokoly DNP3 a IEC 60870-5-104 byly vyvíjeny ve stejném čase. I přesto, že DNP3 přebírá některé definice IEC standardu, stále se ale jedná o dva odlišné protokoly. Oba protokoly využívají EPA architekturu (Enhanced Performance Architecture), která je definovaná jako podmnožina ISO/OSI modelu. Tento model, znázorněný na obrázku 3.1, využívá tři vrstvy ISO/OSI modelu – vrstva aplikační, linková a fyzická. [15] [3]



Obrázek 3.1: EPA model používaný protokoly DNP3 a IEC 60870-5-104 [15]

Pseodo-transportní vrstva

Na obrázku 3.1 není zařazená pseodo-transportní vrstva, která je obsažená v protokolu DNP3. Nachází se mezi vrstvou linkovou a aplikační. Tato vrstva provádí fragmentaci uživatelských dat do jedné nebo více rámců, které následně vysílá na linkovou vrstvu, a zároveň převádí data z přijatých rámců linkové vrstvy na uživatelská data. To umožňuje přenos větších datových bloků a tím využití pokročilých funkcí RTU a posílání zpráv přesahujících právě maximální délku rámce. [3] [15]

3.2 Distributed Network Protocol version 3 (DNP3)

Protokol definuje pravidla, jak komunikují RTU a MTU zařízení mezi sebou. Původně byl vytvořený společností Harris Control Division jako uzavřený protokol pro elektrotechnický průmysl. Od roku 1993 je ale ve vlastnictví skupiny DNP User Group a zároveň se změnou majitele byl uvolněn k volnému použití. Nejen díky tomu se stal postupem času oblíbeným jak u amerických výrobců, tak i po celém světě a to nejen v elektrotechnickém průmyslu. [15] [3]

DNP3 je založen na objektovém modelu, který významně snižuje bitové mapování dat, které jsou tradičně vyžadovány jinými méně objektově orientovanými protokoly. Obě komunikující zařízení musí použít knihovnu společných objektů. [16]

3.2.1 Základy komunikace

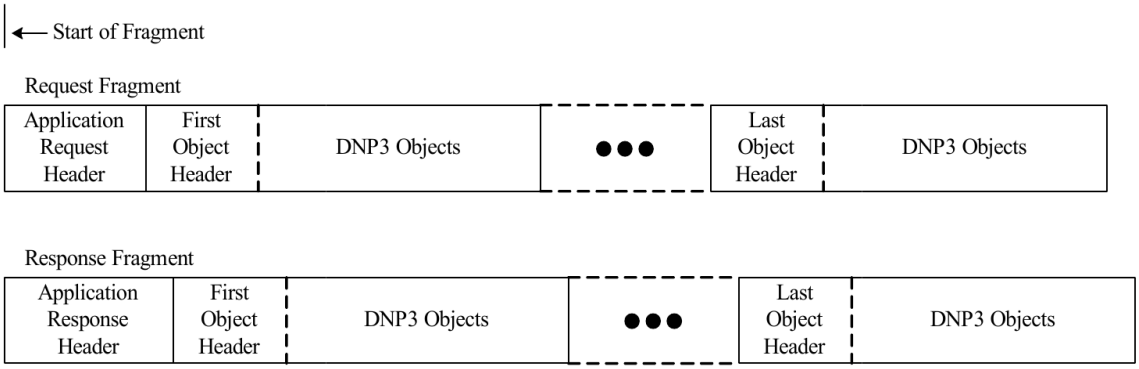
Protokol DNP3 používá základních 27 kódů funkcí. Některé z kódů umožňují požádat či přijmout informace o stavu, jiné kódy nastavit či změnit konfiguraci na vzdálených zařízeních. Většinu kódů iniciuje MTU, nicméně existuje také kód, který umožňuje vzdálenému zařízení samostatně reagovat pomocí tzv. nevyžádané zprávy. Díky tomuto kódu je možné bezprostředně informovat MTU o případných událostech vzniklých na vzdáleném zařízení. [16]

3.2.2 Hlavičky zpráv

Jak je možné vidět na obrázku 3.2, struktury fragmentů požadavků a odpovědí na aplikační vrstvě jsou si značně podobné. Obě dvě hlavičky obsahují dva byty. První se označuje jako „Application Control“, který poskytuje informace potřebné pro rozložení a znovu sestavení zpráv o více fragmentech. Tento byte je rozdělen na 4 bity – pole FIR definuje první fragment, pole FIN definuje poslední fragment, pole CON definuje vyžádání kontrolní zprávy a pole UNS definuje nevyžádané zprávy. Další 4 bity prvního bytu označují sekvenci, o kolikátý fragment v pořadí se jedná – s přibývajícími fragmenty se čítač inkrementuje a provede se operace modulo 16.

Druhý byte v hlavičce je „Function Code“, neboli kód funkce, který identifikuje účel zprávy. Zprávy požadavků generované MTU jsou v rozsahu $\langle 0; 128\rangle$ a zprávy odpovědí generované vzdáleným zařízením jsou v rozsahu $\langle 129; 255\rangle$

Hlavička požadavku je složena ze dvou výše uvedených bytů, nicméně odpovědní hlavička obsahuje kromě bytu kódu funkce ještě další dva byty – interní indikace. Bity v těchto dvou bytech indikují určité stavy a chybové podmínky uvnitř odbočky. [4]



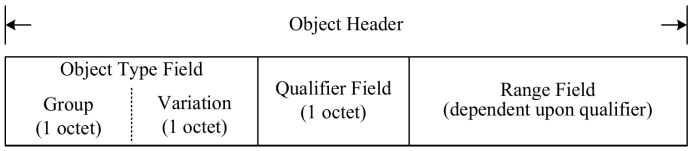
Obrázek 3.2: Schéma struktury zprávy pro požadavky a odpovědi [4]

3.2.3 Hlavičky objektů

Prováděná akce je specifikovaná v hlavičce zprávy, hlavička objektů definuje jaké datové typy požaduje a jaké očekává. Samotný objekt se nachází pouze ve odpovědní zprávě – v požadavku se žádná hodnoty nepřenáší, není proto třeba. Strukturu celé hlavičky je možné vidět na obrázku 3.3.

První byte hlavičky definuje skupinu objektu. Ta specifikuje jaká data či hodnoty budou zahrnuty ve zprávě. Druhý byte označuje variaci objektu, která definuje datový formát objektu – například udává, že se aktuální hodnota analogového signálu bude přenášet jako 16-bit či 32-bit integer. Dvojice skupina a variace objektu jednoznačně definuje strukturu DNP3 objektu a zdroj dat.

Poslední částí hlavičky je bytová hodnota „Qualifier field“ udávající velikost poslední položky „Range field“. „Qualifier field“ je rozdělen na tři pole. První jednobitová hodnota RES nemá zatím využití. Následující tříbitový prefixový kód určuje případnou hodnotu prefixu před každým objektem DNP3, který se řídí záhlavím objektu. Hodnoty prefixu jsou buď indexy nebo velikosti objektů. Zbývající čtyři bity určují specifikátor rozsahu, který udává zda je použitý „Range field“, případně co obsahuje a jak velký je. [4]



Obrázek 3.3: Schéma struktury hlavičky objektu [4]

3.2.4 Nevyžádané odpovědi

Nevyžádané odpovědi („Unsolicited responses“) jsou zprávy, které jsou spontánně posílány ze vzdáleného zařízení bez předchozího požadavku řídící jednotky. Tato metoda je výhodná v systémech, kde se nachází velké množství vzdálených jednotek a jedením řídícím zařízením. [4]

3.3 IEC 60870-5-104

Protokol IEC 60870-5-104 (dále bude použita zkratka IEC 104) byl publikovaný roku 1988 a od té doby stále vyvíjený skupinou IEC Technical Committee 57. Protokol je rozšířením pro standard IEC 60870-5-101, který je založený na modelu EPA (viz 3.1) a který využívá sériovou komunikaci. IEC 104 rozšiřuje tento model tak, aby bylo možné využít pro komunikaci síťový TCP/IP protokol. Aplikační vrstva obou protokolů zůstává tedy stejná, rozdíl je až na nižší transportní a síťové vrstvě. Struktura dat přenášená na aplikační vrstvě je zobrazena na obrázku 3.4. [24]

<table>
<tr><td>ASDU</td><td colspan="2">ASDU Field</td></tr>
<tr><td rowspan="4">Data Unit</td><td rowspan="2">Data Type</td><td>Type ID (TI)</td></tr>
<tr><td>VSQ</td></tr>
<tr><td colspan="2">Cause of Transmission</td></tr>
<tr><td colspan="2">Common Address</td></tr>
<tr><td rowspan="3">Information Object</td><td colspan="2">Address</td></tr>
<tr><td colspan="2">Information Object</td></tr>
<tr><td colspan="2">Timestamp</td></tr>
</table>

Obrázek 3.4: Struktura datové jednotky protokolu IEC 104 [15]

Zprávy protokolu IEC 104 jsou v čistém textu bez jakéhokoli mechanismu ověřování, dále i použitý TCP/IP protokol má sám o sobě problémy se zabezpečením. Proto byla bezpečnost IEC 104 označena jako problematická. [24] [30]

Protokol je široce používán pro energetické sítě v evropských zemích a Číně. [24]

Kapitola 4

Detekce anomálií

Obecně se detekce anomálií dá definovat jako technika používaná k identifikaci neobvyklých vzorů – instance dat neodpovídají žádným způsobem očekávanému chování, protože se značně liší od většiny ostatních údajů, nějakým způsobem nezapadají do běžného chování [14] [11] .

4.1 Typy anomálií

Typicky jsou data spojena s nějakým druhem problému, jako jsou například bankovní podvody, strukturální vady, zdravotní problémy, poruchy zařízení atd. Určení druhu problému je důležité pro definici povahy dané anomálie – po té je možné vybrat, které datové body mohou být považovány za anomálie, a určit správnou detekční metodu. [14]

Anomálie lze rozdělit do tří základních skupin [11]:

4.1.1 Bodové anomálie

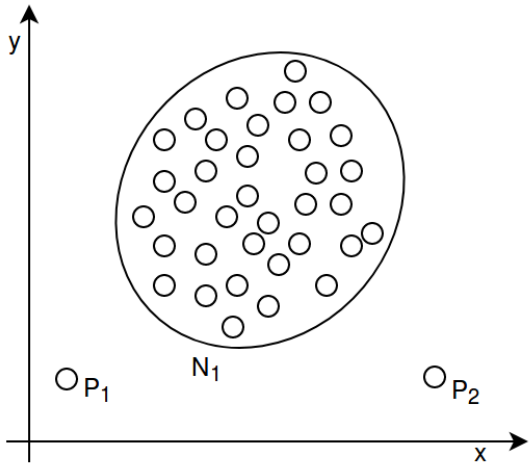
Pokud jednotlivou instanci dat lze považovat za anomálii v porovnání se ostatními daty, nazveme instance jako bodovou anomálii. Jedná se o nejjednodušší typ anomálie a většina výzkumu na detekci anomálií se zaměřuje na tento typ. [11]

Příklad této anomálie je uveden na obrázku 4.1. V běžném životě představuje tuto anomálii například detekce podvodů s kreditními kartami, kde může být vyhodnocena transakční anomálie v rámci výrazně vyšší výběru financí než je u konkrétní kreditní karty obvyklé.

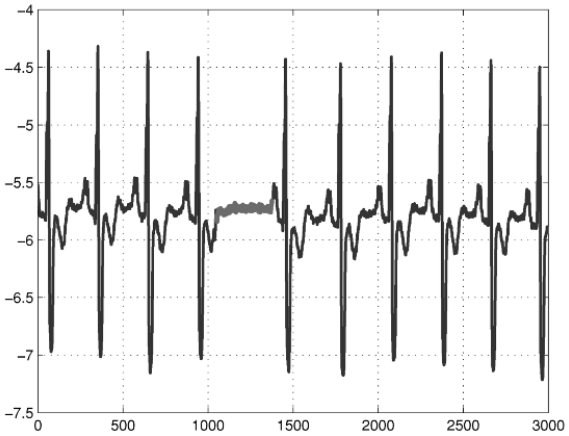
4.1.2 Kolektivní anomálie

Kolektivní anomálií je možné nazvat sekvenci datových instancí, která vykazuje anomálii v rámci celé datové sady. Jednotlivé datové instance nejsou anomálie samy o sobě, ale jejich výskyt spolu jako kolekce je abnormální. [11]

Výskyt kolektivní anomálie je možný vidět na obrázku 4.3, na kterém je zobrazen graf EKG lidského srdce. Hodnoty v době 1000-1400 lze považovat za kolektivní anomálii, jelikož přibližně stejná hodnota existuje po abnormálně dlouhou dobu. Avšak tato hodnota se v rámci datové sady vyskytuje každou periodu srdeční aktivity a sama o sobě není považována za anomálii. Na rozdíl od bodové anomálie, kterou je možné detekovat v jakýchkoliv datových sadách, pro detekci kolektivní anomálie je nutné mít přizpůsobenou datovou sadu, ve kterých jsou datové instance spojeny. [11]



Obrázek 4.1: Bodová anomálie je patrná na bodech P_1 a P_2. Tyto body leží mimo hranici definovanou třídou N_1, která specifikuje oblast normálních bodů.



Obrázek 4.2: Kolektivní anomálie je patrná na grafu křivky EKG lidského srdce [11].

Metody detekce kolektivních anomálií jsou nejčastěji používány u sekvenčních, grafových či prostorových dat [11].

4.1.3 Kontextové anomálie

Anomálie jsou označovány za kontextové (někdy též jako podmíněné), pokud je instance dat abnormální v konkrétním kontextu, tedy lokálnímu okolí dané instance. Samotný kontext musí být obsažený v datové sadě a musí být specifikovaný jako součást problému. [11]

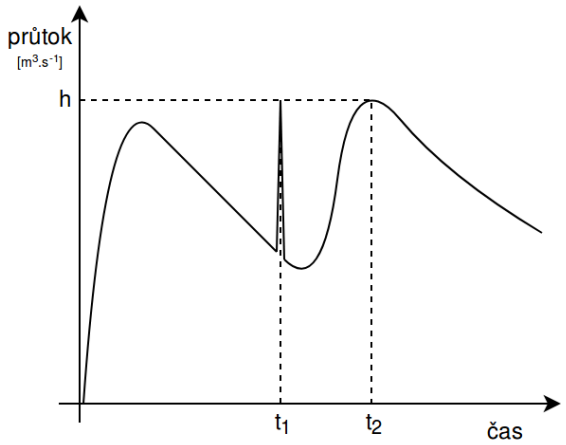
Každá instance dat je definována pomocí následujících dvou sad atributů [11]:

- Kontextové atributy – používají se k určení kontextu (nebo sousedství) pro tuto instanci. Například v prostorových datových sadách se jedná o souřadnice udávající pozici. V datech časových řad je kontextovým atributem čas, který definuje pozici instance v rámci datového setu.

- Atributy chování – definují nekontextové vlastnosti instance. Například v datové sadě popisující vývoj vodočtu na řece během roku je atributem chování odečet v jeden konkrétní den.

Anomálie jsou vyhodnocovány pomocí hodnot atribut chování ve specifickém kontextu. Datová instance může být vyhodnocena jako kontextová anomálie v jednom kontextu, ale instance se stejnými hodnotami u atributů chování v rámci druhém kontextu může být považována za normální. Tato vlastnost je klíčem k identifikaci kontextových a behaviorálních atributů pro techniku detekce kontextové anomálie.

Bodovou anomálii nebo kolektivní anomálii lze vyjádřit kontextovou anomálií. Tuto transformaci je možné provést, pokud datová instance obsahuje atribut, který může být použitý jako kontext. [11]



Obrázek 4.3: Kontextová anomálie v čase t_1. Instance dat se stejným průtokem by v rámci kontextového okna v čase t_2 nebyla vyhodnocena jako kontextová anomálie.

Příkladem reálného světa může být již zmíněný vodočet na řekách. Na obrázku 4.3 je graf průtoku v závislosti na čase. Z něj je patrné, že hodnota h s časem t_1 v rámci daného kontextu je možné považovat za abnormální. Nicméně v čase t_2 už není možné považovat tu samou hodnotu h za anomálii. Obě dvě datové instance mají stejné hodnoty atributů chování, ale kontext se liší.

Kontextové anomálie jsou nejčastěji používány u prostorových datových sad nebo u datových sad s časovými řadami [11].

4.2 Hodnocení detekčních metod

Rozdíl detekčních metod je také ve výstupu, jakým je hlášeno hodnocení, zda-li se jedná o anomálii nebo ne. K tomu jsou využívány standardně dva typy – bodování a značkování.

Prvním způsobem je tzv. bodování, kde každé datové instanci v datové sadě je přiřazené ohodnocení. Tím může být například pravděpodobnost, s jakou daná instance spadá do třídy anomálií. Vznikne tedy množina instancí s ohodnocením (v případě pravděpodobnosti v rozsahu $\langle 0, 1 \rangle$), z které je možné na základě definovaného prahu filtrovat jen ty nejvážnější anomálie.

Výstupem značkování jsou přímo třídy, kterými je ohodnocená každá datová instance. Většinou se jedná rozdělení dat do dvou tříd – normální data a anomálie. Nevýhodou tohoto typu oproti bodování je absence informace o závažnosti anomálie. [11]

4.3 Typy detekčních metod dle učení

Obecně, pro zajištění co nejlepších výsledků detekčních metod, je důležité vytvořit co nejobsáhlejší sadu dat s definovanými třídami pro jednotlivé instance dat. Ovšem získání takovýchto dat je poměrně náročné – je nutné pokrýt všechny možné třídy, které budou reprezentovat dostatečně přesná data. Často je také zapotřebí manuální zásah analytika, který data správně rozdělí. Obvykle je snazší získání normální datové třídy než definování datových tříd anomálií, které pokrývají všechny možné typy anomálního chování. V některých případech ani nelze definovat všechny třídy anomálií, jelikož v době trénování ještě nemusí být všechny známé. [11]

Typy detekčních metod lze rozdělit do níže popsaných tří kategorií podle toho, jaké třídy obsahuje trénovací datová sada . [11]

4.3.1 Učení s učitelem (Supervised Anomaly Detection)

Pro metody učení s učitelem je nutné definovat tréninkové datové sady, v kterých jsou definované datové instance pro normální i anomální třídy. Pomocí této datové sady je následně typicky vytvořen prediktivní model pro normální a anomální třídu. U nové instance dat bez specifikované třídy dojde k porovnání s modelem, na základě kterého se určí třídy daná instance náleží. [11]

Existují dva hlavní problémy metod učení s učitelem. Prvním z nich je množství datových instancí pro anomální třídu v trénovacích sadách, kterých bývá výrazně méně v porovnání s množstvím instancí normální třídy. Druhým problémem je získání přesných a reprezentačních vzorků datových instancí, které bývá náročné, a to zejména pro třídu anomálií. Proto byly navržené různé metody, které jsou schopné ze vzorku dat normální třídy uměle vytvořit anomálie tak, aby byla vytvořená potřebná trénovací sada. [11]

Shrnutí výhod a záporů učení s učitelem [11]:

+ metody této skupiny jsou přesné v detekci anomálií
+ možné využití v případech, kdy jsou anomálie výrazně častější než normální data
− v některých případech nemožné ohodnotit data
− náročné definovat data všech možných normálních a abnormálních tříd

4.3.2 Částečné učení s učitelem (Semisupervised Anomaly Detection)

Metody založené na částečném učení s učitelem využívají tréninková data, kde jsou označené datové instance pouze pro normální třídu. Jelikož se k trénování modelu nevyužívají data označená třídou anomálií, je tato metoda častěji používaná než metody učení s učitelem. Aby tyto metody mohly být využité k detekci, je třeba vytvořit model pro třídu odpovídající normálnímu chování a následně tento model použít k detekci anomálií v datech testu. [11]

Do kategorie detekce anomálií pomocí částečného učením s učitelem je zařazené také několik metod, u kterých je pro trénování důležité mít správně označené datové instance

třídy anomálií. Nicméně tyto metody nejsou příliš používány, a to zejména pro již zmíněnou obtížnost získání datové sady, která obsahuje každé možné anomální chování. [11]

Shrnutí výhod a záporů částečného učení s učitelem [11]:

+ více rozšířené než metody využívající učení s učitelem
+ není nutné mít označené datové instance anomálií
– náročné definovat data všech možných normálních tříd

4.3.3 Učení bez učitele (Unsupervised Anomaly Detection)

Metody zařazené do kategorie učení bez učitele jsou nejrozšířenější díky jejich zásadní vlastnosti – pro získání prediktivního modelu nevyžadují trénovací sady s definovanými třídami. Při detekci tyto metody předpokládají, že výskyty instancí dat normální třídy jsou v trénovacích sadách mnohem častější než anomálie. Pokud se ale v testovacích datech nevyskytují žádné anomálie, jsou tyto metody velmi náchylné na počet falešných poplachů (normální datové instance nesprávně označené jako anomálie).

V rámci kategorie učení bez učitele lze také využít některé metody částečného učení s učitelem (popsaných v sekci 4.3.2), kde se pro trénování využijí neoznačené datové sady. Adaptace těchto metod může být funkční pouze za předpokladu, že v datových sadách je velmi málo anomálií. Díky nízké četnosti budou mít anomálie pro trénování modelu malý význam, a proto bude model vůči několika málo anomáliím imunní. [11]

Shrnutí výhod a záporů učení bez učitele [11]:

+ nejjednodušeji použitelné
+ není třeba značených datových sad pro trénování modelu
– vychází z předpokladu, že běžné příklady jsou mnohem častější než abnormální

4.4 Metody detekce anomálií

4.4.1 Klasifikační metody

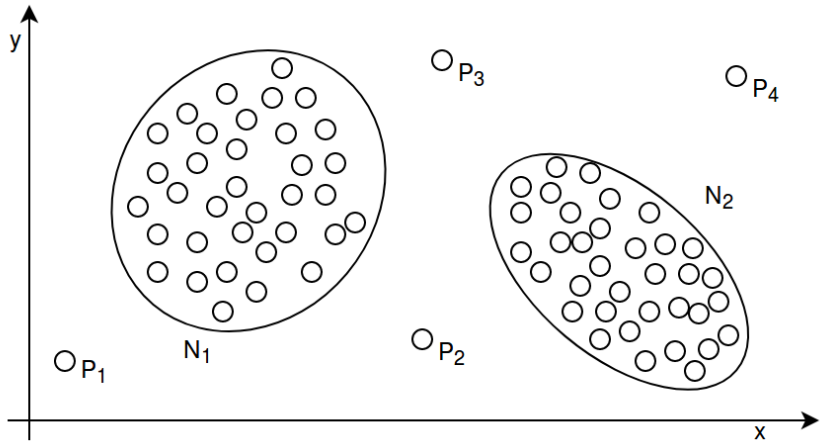
Všechny metody detekce anomálií založené na klasifikaci mají společné dvě fáze – fáze trénování a testování. V první fázi se model neboli klasifikátor učí z dostupných označených datových sad. V testovací fázi následně probíhá určování tříd datových instancí z testovací sady na základě naučeného modelu z první fáze. Aby klasifikační metody správně detekovaly možné anomálie konkrétní datové instance, je nutné zajistit učení modelu z trénovací datové sady, která bude pokrývat okolí dané instance v rámci normální i anomální třídy.

Klasifikační metody jsou rozděleny do dvou různých kategorií, na základě četnosti tříd, kterými jsou označeny instance testovací sady. První kategorií jsou metodu vícetřídní klasifikace, které předpokládají výskyt vícero normálních tříd v označených instancích testovacích dat. Klasifikátor se pro datové instance snaží rozlišit, zda patří mezi normální třídy. Pokud instance nenáleží do žádné normální třídy je označena jako anomálie. Na obrázku 4.4 v grafu zobrazena prostorová data, která mají definované dvě normální třídy N_1 a N_2.

Do druhé kategorie spadají jednotřídní klasifikátory. Pro tyto metody je nutné mít v trénovacích sadách označené instance pouze jednou třídou. Takovéto metody se v rámci učení snaží určit hranice normálních instancí použitím jednotřídních klasifikačních algoritmů (například One-Class SVM). Jakákoliv testovací data, které nespadají do určené

hranice, je označena jako anomální. Příklad jednotřídní datové sadu je na obrázku 4.1, kde je definována jedna normální třída N_1. [11]

Do skupiny klasifikačních metod spadají metody neuronových sítích [29] či bayesovských sítích [36], metody podpůrných vektorů [33] a metody na bázi definovaných pravidel [18]. [11]



Obrázek 4.4: Vícetřídní klasifikační metody označují datové instance do více normálních tříd.

Klíčové vlastnosti klasifikačních metod jsou následující [11]:

+ robustní technika při rozlišování mezi instancemi patřícími do různých tříd
+ rychlá testovací fáze – dochází pouze k porovnání s naučeným modelem
– obtížné získat datové sady s dostatečně přesnými instancemi popisujících normální třídy, na čemž závisí úspěšnost detekce

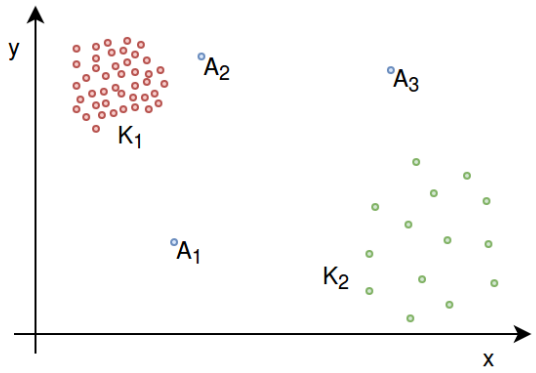
4.4.2 Metody založené na detekci nejbližších sousedů

Metody založené na detekci nejbližšího souseda jsou založené na myšlence, že normální datové instance se nachází v blízké vzdálenosti od jejich sousedů, na rozdíl od anomálií, které se vyskytují daleko od nich. Proto je pro použití těchto metod nutné definovat vzorce pro výpočet vzdálenosti nebo podobnosti mezi dvěma datovými instancemi. Tento výpočet je možné provést několika způsoby – pro spojité hodnoty je často využívaná euklidovská vzdálenost, pro kategorické atributy se používá metoda jednoduchého srovnávacího koeficient (SMC). Pokud datové instance obsahují struktury různých datových typů, jsou obyčejně porovnávány jednotlivé položky samostatně. [11]

Vzorec pro výpočet vzdálenosti (případně podobnosti) nemusí být nutně definovaný jako metrika – je nutné zachovat axiomy nezápornosti, totožnosti a symetrie, nicméně axiom trojúhelníkové nerovnosti už platit nemusí. Metody založené například konceptu nejbližšího souseda lze obecně rozdělit do dvou částí podle významu jejich hodnocení [11]:

- metody, v kterých je k ohodnocení využívaná vzdálenost datové instance ke svému nejbližšímu sousedovi
- metody využívající k ohodnocení relativní hustotu každé datové instance

Do skupiny metod založených na konceptu nejbližších sousedů patří algoritmus k-nejbližších sousedů [17] či algoritmy založené na hustotě výskytů instancí [13]. Graf znázorňující anomálie detekované pomocí těchto metod je zobrazen na obrázku 4.5



Obrázek 4.5: Metody založené na konceptu nejbližších sousedů jsou schopné detekovat anomálie A_x, které nenáleží do normálních tříd K_1 a K_2. Ačkoliv by bod A_2 nebyl považován za anomálii v rámci normální třídy N_2, je v tomto příkladu označen jako anomálie, jelikož se zvažuje lokální hustota kolem analyzované instance.

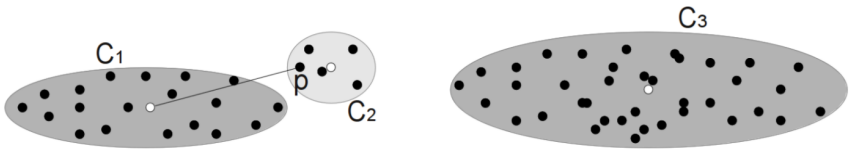
Výhody a nevýhody metod pracujících s nejbližšími sousedy [11]:

+ základní výhodou metod je možné učení bez učitele
+ metody s částečným učením získávají ještě lepší výsledky při detekci anomálií
+ jednoduchá aplikace na instance s různými datovými typy – stačí nadefinovat výpočet vzdálenosti
– metody učení bez učitele vrací chybné ohodnocení pro datové sady, v kterých instance normální třídy nemají dostatečný počet sousedů nebo v kterých naopak výskyty anomálií mají dostatek sousedů
– trénovací datový set metod částečného učení musí vždy obsahovat dostatečný počet instancí pro definici normálních tříd
– výpočetně náročné při získávání vzdálenosti datové instance

4.4.3 Metody shlukování

Metody shlukování jsou založené na seskupování podobných datových instancí do klastrů. Nejčastěji se jedná o techniky s učením bez učitele. Metody shlukování zaměřené na detekci anomálií je možné rozdělit do dvou skupin na základě vstupních předpokladů, které je třeba splnit pro trénovací sadu dat. Pravidlo první skupiny metod shlukování říká, že instance dat normální třídy se nachází blízko jejich geometrického středu klastru, naproti tomu instance anomální třídy jsou daleko od geometrického středu nejbližšího klastru. Nevýhodou této skupiny jsou anomálie v datech, které jsou si polohou blízké takovým způsobem, že samy vytvoří klastr. Tyto metody označí celý klastr za normální třídy, a proto nejsou schopné takové anomálie detekovat. Druhá skupina metod je založená na pravidle definujícím, že datové instance normální třídy náleží do velkých a hustých klastrů, zatímco anomálie patří buď k malým nebo nepříliš hustým klastrům.

Příklad dat vhodných pro detekci anomálií metodami shlukování jsou popsán s obrázkem 4.6. Pro správnou detekci dat je nutné vybrat metody ze skupiny, kterým odpovídá charakteristika detekované anomálie. [11]



Obrázek 4.6: Jako bodové ohodnocení datové instance p je vypočtená vzdálenost mezi bodem p a geometrickým středem klastru C_1. V tomto případě jsou klastry C_1 a C_2 považovány za velké a klastr C_2 za malý [9].

Shrnutí výhod a záporů metod shlukování [11]:

+ metody shlukování mohou pracovat v režimu učení bez učitele
+ fáze testování je velmi rychlá – malý počet klastrů, s kterými se každá datová instance porovnává
– rychlost metod shlukování je závislá na algoritmech detekce klastrů normálních tříd
– největším problémem je výpočetní složitost pro shlukování dat, zvláště pokud se používají algoritmy se složitostí $O(N^2d)$

Mezi nejznámější metody shlukování patří metoda hledání lokálních odlehlých hodnot pomocí shlukování (tzv. CBLOF) [26] a metoda K-means [25].

Metody shlukování jsou dost podobné s metodami založené na konceptu nejbližších sousedů. Nicméně hlavním rozdílem mezi těmito dvěma skupinami detekce anomálií je způsob získání bodového ohodnocení. Metody shlukování hodnotí každou instanci na základě jednoho klastru na rozdíl od metod založených na konceptu nejbližších sousedů, u kterých je ohodnocení datové instance určeně pomocí více ostatních instancí v jejím lokálním okolí. [11]

4.4.4 Statistické metody

Všechny statistické detekční metody definují anomálii jako označení datových instancí, které jsou částečně nebo úplně irelevantní, jelikož nejsou generovány stochastickými modely. Na tomto základě vznikl předpoklad, díky kterému mohou statistické metody fungovat – datové instance normálních tříd se vyskytují v oblastech s vysokou pravděpodobností stochastického modelu na rozdíl od anomálie, které se vyskytují v oblastech s nízkou pravděpodobností. [11]

Metody obvykle vytváří statistický model, který popisuje chování normálních tříd. Při fázi testování se tomuto modelu předloží neoznačená instance dat, pro kterou se určí jestli náleží do modelu nebo ne. Datové instance, které mají nízkou pravděpodobnost, že mohou být generovány naučeným modelem, jsou označené jako anomálie. Obecně jsou statistické metody rozděleny do dvou základních skupin – parametrické a neparametrické metody. [11]

Parametrické metody, které nutně vyžadují znalost základního rozložení datových sad, využívají například algoritmů regresivní analýzy či Gaussovských modelů. Pro neparametrické statistické modely není nutně předem známá struktura modelu, hodnocení probíhá

pouze na základě testovací datové instance. Mezi neparametrické metody patří například algoritmy využívající principy histogramů [23] nebo metody jádrových odhadů hustoty [34]. [11]

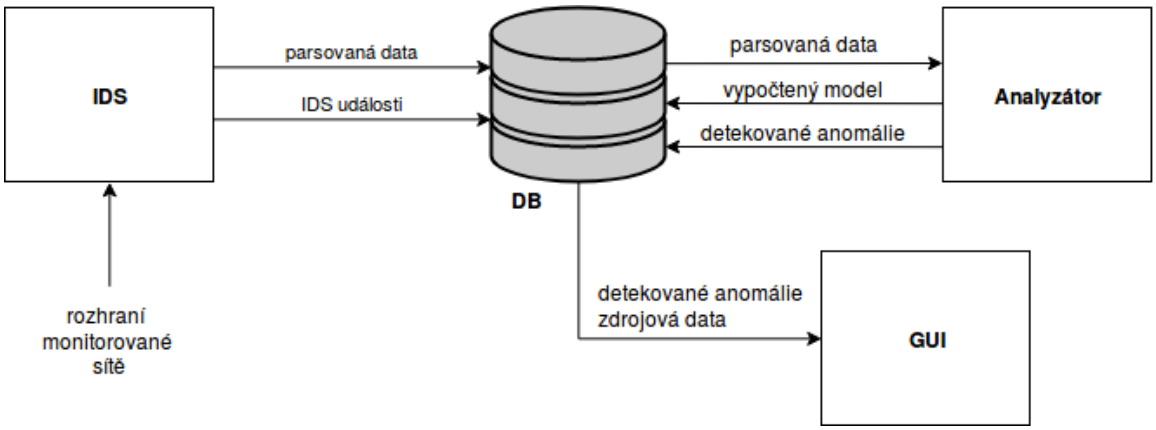
Vlastnosti statistické metod jsou následující [11]:

+ detekce anomálií pomocí statistických metod je matematicky podložena, pokud platí předpoklad základní distribuce dat
− metody spoléhají, že datové instance jsou generována na základě určitého rozdělení
− existuje několik hypotéz, že zvláště pro více dimenzionální datové sady je netriviální určit nejlepší statistickou metodu
− metody využívající histogramy jsou jednoduché na implementaci, ale nejsou schopné zachytit anomálie u více dimenzionálních dat

Kapitola 5

Návrh systému

V následující kapitole budou popsány jednotlivé moduly, z kterých je sestaven analyzátor pro detekování anomálií. Tyto moduly jsou zobrazeny na obrázku 5.1 společně se základním rozhraním, které využívají pro komunikaci.



Obrázek 5.1: Návrh architektury systému pro detekci anomálií.

5.1 Intrusion Detection System – IDS

5.1.1 Existující nástroje

Pro monitorování síťového provozu se nejčastěji využívají dva nástroje – Suricata a Snort. Oba dva projekty mají hodně společného: jsou multiplatformní, open-source, implementované v jazyce C, poskytují podporu IPv6 protokolu, detekují hrozby na základě pravidel, atd. Výběr nástroje proběhl na základě provedených testů [8], z kterých lépe vyplynula Suricata vyvíjená organizací Open Information Security Foundation.

I když je Suricata v porovnání s nástrojem Snort náročnější na paměť a procesorový čas, v testech vykazovala menší množství zahozených paketů. Suricata je totiž rozdílná svou architekturou, která byla navržena pro vícevláknové zpracování paketů, díky které se hodí pro analýzu vysokorychlostní sítě. Mezi dalšími výhodami Suricaty je podpora reputací IP adres, silná podpora komunity či automatická detekce protokolů v rámci předzpracování dat, díky které jsou aplikovaná konkrétní pravidla pouze pro daný protokol.

```
drop tcp $HOME_NET any -> $EXTERNAL_NET any
(
        msg:"ET TROJAN Likely Bot Nick in IRC (USA +..)";
        flow:established,to_server;
        flowbits:isset,is_proto_irc;
        content:"NICK ";
        pcre:"/NICK .*USA.*[0-9]3,/i";
        reference:url,doc.emergingthreats.net/2008124;
        classtype:trojan-activity;
        sid:2008124;
        rev:2;
)
```

Obrázek 5.2: Ukázka detekčního pravidla Suricaty. První část, akce, je zobrazená červenou barvou, hlavička je odlišená zelenou barvou a poslední část, obsah pravidla, je vykreslena modře.

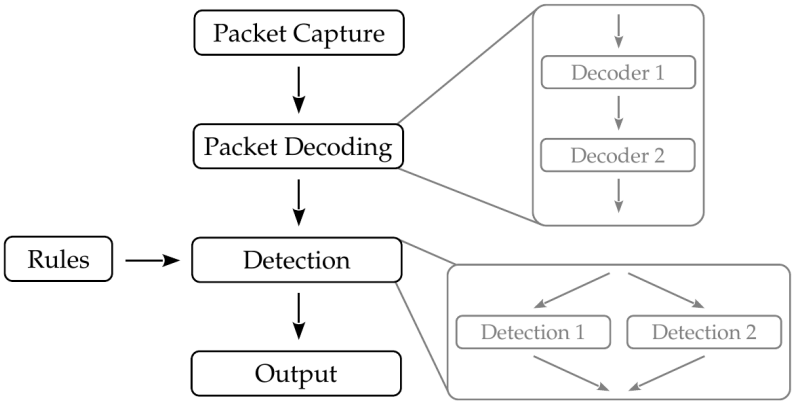
5.1.2 IDS pravidla

Nespornou výhodou obou nástrojů také možnost importovat velkou část pravidel, které slouží k detekci hrozeb. Ukázka detekčního pravidla je na obrázku 5.2 a obecně se skládají z třech hlavních částí [7]:

- akce – definuje co se stane v případě shody signatury
- hlavička – určuje protokol, IP adresu, porty a směr pravidla
- obsah pravidla – blíže specifikuje pravidlo (obsah, popis, vyhledávaný vzor)

5.1.3 Popis zpracování síťového provozu

Zpracování vstupních dat pomocí Suricaty lze rozdělit na čtyři základní části – zachycení paketů ze rozhraní sítě, dekódování hlaviček paketů, detekce hrozeb a výstup. Schéma je na obrázku 5.3.



Obrázek 5.3: Schéma zpracování paketů nástrojem Suricata. [21]

Na začátku celého procesu dojde k zachycení paketu ze síťového rozhraní. Následně dekódovací funkce čte pakety a ukládá dekódovaná data do interní reprezentace paketu. V rámci paketu je tato funkce volána vždy jen jednou.

Při úspěšném dekódování hlaviček procházejí pakety detekcí hrozeb. Ta porovnává, zdali jsou definovaná pravidla shodná s interní reprezentací paketu. Pro zlepšení efektivity vyhodnocování pravidel je možné využít způsob paralelizace tak, aby jeden paket byl zpracováván několika detekčními moduly naráz.

V poslední části je definovaný způsob výstupu celého procesu, kdy dochází k uložení dekódovaných dat a detekovaných událostí. [21]

Vstupem tohoto modulu je provoz zachycovaný na síťovém rozhraní. Výstupem jsou naparsované protokoly a události detekované IDS pravidly. Tyto data se průběžně ukládají do lokální databáze. Naparsovaný protokol DNP3 ve formátu JSON je možné vidět v příloze A, ukázka naparsovaného protokolu ve formátu IEC 104 je v příloze B.

5.1.4 Implementované změny

Suricata se vydává pod open-source licencí, a proto je její kód volně dostupný ke stažení a úpravám. Díky tomu je možné implementovat potřebné části, které nejsou standardně v upstreamové verzi. Pro správné fungování celého systému bylo nutné provést několik změn.

Nové výstupní rozhraní

Nejzásadnější úprava Suricaty se týká výstupního modulu, který je typicky směrovaný do souboru či syslog démona. Implementace nového rozhraní umožní ukládat toky a události přímo do databáze.

Nový parser

Oficiální verze Suricaty parsuje pouze protokol DNP3, proto je nutná neimplementovat parser protokolu IEC 104.

Přidání časové značky

Aplikační data analyzovaných toků nemusí nutně obsahovat časovou značku o odeslání či přijetí, proto je při parsování přidána položka `detected_time`, která udává čas, kdy jsou protokoly parsovány.

Úprava frekvence výstupu

Výstupní modul Suricaty čeká na ukončení toku, aby při provádění výstupních operací byl tok zapsán jako jeden celek. U dlouho trvajících toků je nastaven timeout na pět minut, tzn. že každých pět minut než je tok ukončen se spouští výstupní modul pro parsovaná data. Tato perioda byla stanovená na jednu minutu, aby každou minutu byly nová data uložena.

5.2 Datové úložiště

5.2.1 Výběr technologie

Jako úložiště dat bude sloužit databáze PostgreSQL. Jedná se o volně dostupný, open-source, objektově-relační databázový systém s širokou komunitní podporou. Tento systém je dlouhodobě široce používaný i pro velké komerční projekty a díky tomu je architektura systému odladěná a zaručuje spolehlivost a integritu dat. Databázová transakce také plně podporují vlastnost ACID pro zachování konzistentního stavu.

5.2.2 Požadavky

Podle schématu na obrázku 5.1 rozdělující jednotlivé moduly bude potřebné uložit do databáze následující data:

- IDS události
- parsovaná data – informace o dekódovaných protokolech
- model zachycující dlouhodobý stav sítě
- události analyzátoru

5.2.3 ER diagram

Na základě těchto požadavků byl vytvořen relační model databáze, který je popsán ER diagramem na obrázku 5.4.

5.2.4 Popis tabulky

Následující části jsou vysvětlené logické významy tabulek, jejich položky a vzájemné vazby mezi tabulkami.

Tabulka toků – `flow`

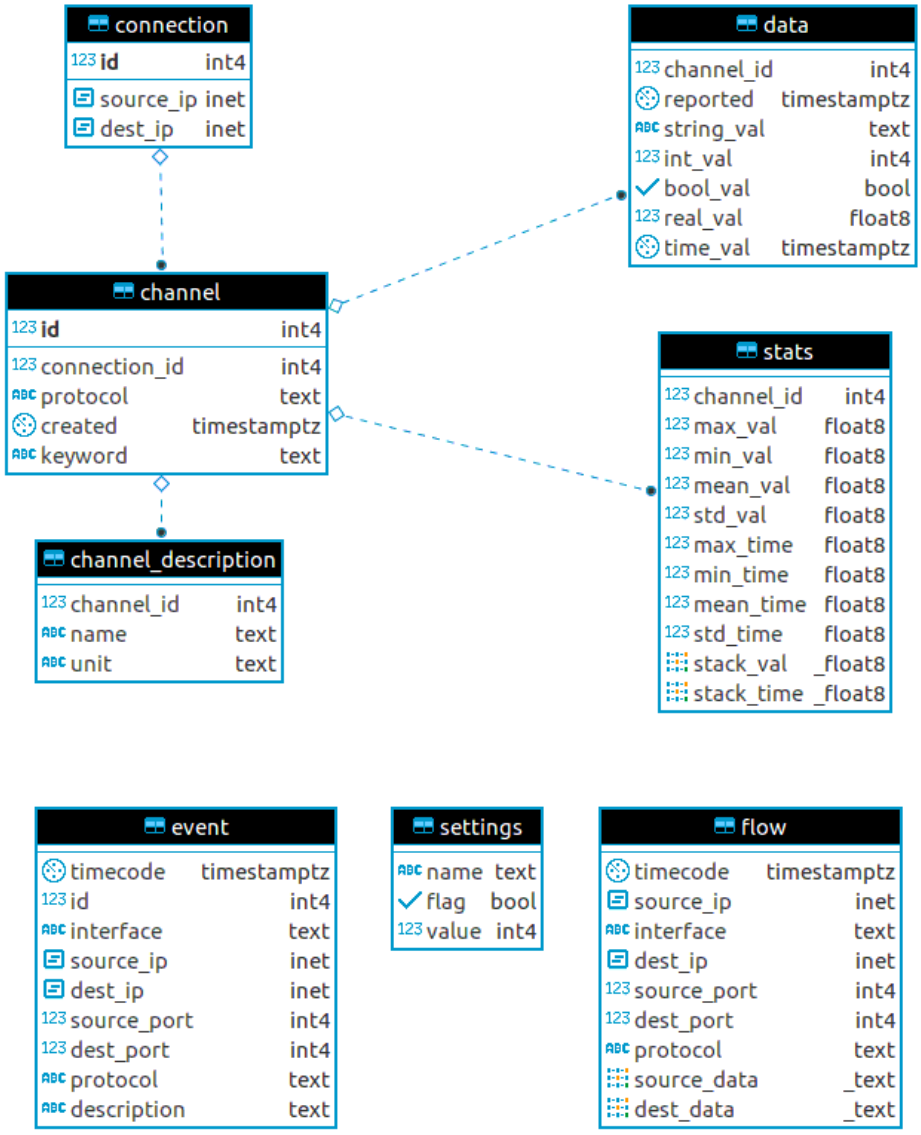
Jedním z výstupů Suricaty jsou naparsované toky, které jsou zdrojem datových instancí. Veškeré informace o tocích ukládá Suricata do této tabulky, z které si následně toky načítá analyzátor.

Tabulka spojení – `connection`

Pojem spojení jsem definoval jako komunikaci mezi dvěma hosty na portech analyzovaných protokolů. Tabulka spojení uchovává IP adresy dvojic hostů. Do této tabulky se vloží nový záznam vždy, když bude zachycen do té doby neexistující komunikace na protokolu DNP3 či IEC 104.

Tabulka kanálu – `channel`

Tabulka kanálu uchovává informace o jednotlivých datových kanálech, které jsou definované jako unikátní trojice protokolu komunikace, spojení a klíčového slova. V rámci datového kanálu jsou přenášené datové instance, které k sobě logicky náleží – vždy pochází ze stejného zdroje (např. konkrétní teplotní čidlo).



Obrázek 5.4: ER diagram pro navržený databázový model.

Klíčové slovo je unikátní identifikátor, který pro daný protokol rozeznává jednotlivé datové toky v komunikaci mezi dvěma hosty. Na základě konfigurace SCADA systému jsou klíčová slova dynamicky tvořená podle zpracovávaného protokolu:

- **IEC 104** – klíčové slovo je hodnota, kterou specifikuje položka `IOA`. Tato položka slouží právě k definování jednoznačných adres v rámci konkrétního systému. Hodnota je definovaná jako jednobytové číslo.
- **DNP3** – klíčové slovo je text odpovídající vzoru „Obj.X Var.Y [Z]“, kde hodnoty X, Y a Z definují unikátní datový kanál v daném systému. Objekt X a variace Y určují o jaký datový typ se jedná. V případě, že je zde více datových toků se stejným datovým typem, index Z jednoznačně tyto datové kanály oddělí.

Datová tabulka – `data`

Datová tabulka je vázána na specifický kanál, pro který ukládá hodnoty datových instancí v závislosti na čase. Díky této tabulce je možné prezentovat historický vývoj kanálu.

Tabulka statistik – `stats`

V této tabulce jsou uloženy veškeré potřebné údaje pro algoritmy detekující anomálie. Jsou zde pro každý kanál uložené statistické údaje a dvě pole, kde je uložených posledních N hodnot pro opětovné naučení detekčního modelu.

Tabulka událostí – `event`

Suricata na svém výstupu mj. předává události pro toky, které detektory Suricata společně s definovanými pravidly označily jako možné hrozby. Tyto události společně s událostmi generovanými analyzátorem (nové objevy, anomálie) se ukládají do tabulky `event`.

Tabulka s popis kanálu – `channel_description`

Tabulka `channel_description` slouží při prezentaci výsledků, kde si může uživatel definovat pro každý kanál název a jednotku veličiny, díky čemu je interpretace dat zřetelnější.

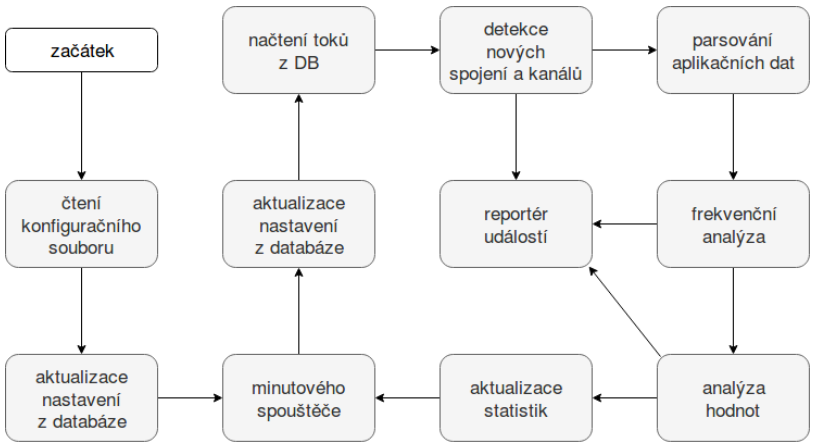
Tabulka nastavení – `settings`

V této tabulce se nachází nastavení, kterým může uživatel ovlivnit chování analyzátoru.

5.3 Analyzátor

Analyzátor je aplikace, jejíž hlavní úlohou je analyzovat příchozí datové instance získané nástrojem Suricata a reportovat výsledky analýzy. Jedná se o aplikace implementovanou v jazyk Java.

5.3.1 Průběh analyzátoru



Obrázek 5.5: Na diagramu je v blocích naznačený průběh aplikace.

Jednotlivé logické části aplikace jsou zobrazené v diagramu na obrázku 5.5. Při spuštění je vyžadován jeden parametr, který definuje cestu ke konfiguračnímu souboru – v něm je v JSON formátu uložená konfigurace pro připojení k databázi. Schéma konfiguračního souboru je ve výpisu 5.1. Jakmile jsou získány přístupové údaje k databázi proběhne ověření údajů a načtení nastavení z databáze. Tím proběhne inicializační fáze a může se překročit do hlavní části programu, která je pomocí nastaven časovače spouštěná patnáctou vteřinu každé minuty.

```
{
  "database": {
    "host": "server",     // hostname nebo IP adresa databazoveho serveru
    "port": "5432",       // port k~pripojeni databaze
    "name": "tidb",       // jmeno databaze
    "user": "tidb",       // jmeno databazoveho uzivatele
    "password": "12345"   // heslo databazoveho uzivatele
  },
  "trainingMode": false  // defaultni nastaveni trenovaciho rezimu
}
```

Výpis 5.1: Schéma konfiguračního souboru

Na začátku hlavní smyčky programu dojde k aktualizaci nastavení, po které se načítají toky ze zpracovávané minuty. Z těchto toků jsou extrahovány spojení a kanály, které se následně porovnávají s již existujícími. Pokud je objevené nové spojení či kanál, vytvoří se odpovídající události.

V další fázi se zpracovávají aplikační data. Nejprve dojde k naparsování dat, které je popsané v sekci 5.3.2. Získaná data se analyzují pomocí metod pro detekci anomálií – první proběhne frekvenční analýza, poté se analyzují hodnoty pomocí statistické metody a následně se porovnávají s modelem klasifikační metody.

Detekčních metod jsou podrobněji popsány v kapitole 6.

5.3.2 Parsování dat

Suricata do databáze ukládá aplikační data ve JSON formátu. Cílem procesu parsování je z těchto aplikačních získat identifikátor datového kanálu, datový typ a aktuální hodnotu. Jak jsou generovány identifikátory je popsané v sekci 5.2.4, proto se na umístění identifikátorů a hodnot.

IEC 104

Data protokolu IEC 104 se mohou nacházet jak v zdrojových, tak i cílových aplikačních datech. V JSON objektu (ukázková v příloze B) je položka `payload`, která obsahuje pole s datovými instancemi. Každý prvek pole může představovat jiný kanál, tudíž má svůj identifikátor a hodnotu – identifikátor je číselná hodnota pod položkou `IOA`. Hodnota může nabývat dvou datových typů – *integer* nebo *double*. Pokud se jedná o datový typ *double*, je hodnota uložená pod klíčem `FLT`. Pro položky celočíselných hodnot existuje více klíčů – položka je definována jedním z následujících klíčů `SIQ, DIQ, VTI, BSI, NVA, SVA, COI, BCR, QOI, SCO, DCO, RCO`.

Některé z instancí dat mohou obsahovat časové značky pro synchronizaci času. Tato značka pod položkou `cp56` je součástí prvku pole stejně jako hodnoty dat. Tato značka udává

přesný čas zdroje. Pokud je tato značka detekována je použita místo času `detected_time` přidávaného při parsování protokolu.

DNP3

Protokol DNP3 datové hodnoty pouze v cílových aplikačních datech, jelikož se jedná vždy o odpovědi. Datové objekty v aplikačních JSON datech (ukázka v příloze A) se nachází pod položkami `application` a `objects`. Stejně jako u předchozího protokolu se zde nachází pole, avšak na rozdíl od IEC 104 reprezentuje každý prvek datový objekt. V příloze C je seznam veškerých datových objektů, které analyzátor dokáže zpracovat. Seznamu je logicky rozdělený podle významu dat a nachází se v něm popis jednotlivých objektů, velikosti dat objektů a datové typy, na které jsou jednotlivé objekty převedeny.

Klíčová slova pro nalezení hodnoty jsou rozdělená podle toho, do které skupiny datové objekty náleží:

- binární objekty – reprezentují datový typ *boolean* a jsou definovány klíčovým slovem `state`
- objekty čítačů – ukládají celočíselné datové typy *integer* pod klíčem `count`
- objekty analogových vstupů/výstupů – hodnoty pod klíčem `value` mohou nabývat datových typů *integer* nebo *double*
- časové objekty – reprezentují čas v epochální formě pod klíčovým slovem `relative_time_ms` pro relativní čas a `timestamp` pro absolutní čas

5.3.3 Výstup analyzátoru

Z pohledy výstupních dat má analyzátor dva cíle – příprava dat pro jejich zobrazení v GUI a zajištění persistence dat potřebných pro detekční metody. Ke splnění obou cílů využívá databázové úložiště.

Do tabulky `data` (sekce 5.2.4) ukládá analyzátor naparsovaná surová data společně s datací jejich výskytu, což umožňuje pro každý kanál historický pohled na vývoj hodnot. Druhým typem dat pro GUI jsou události, které se ukládají do tabulky `event` (sekce 5.2.4).

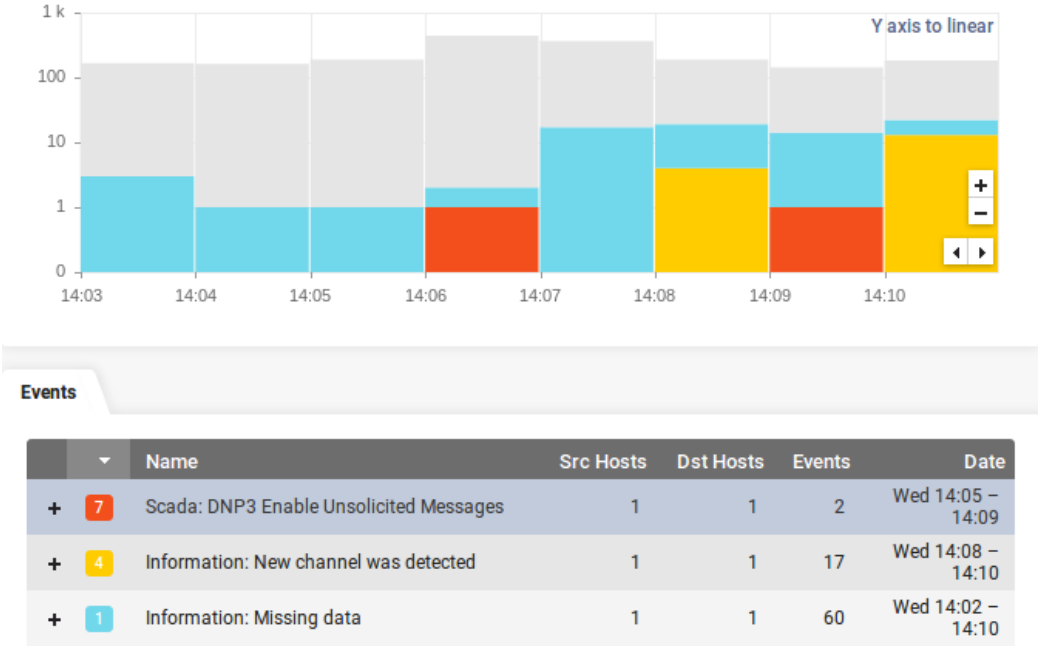
Data potřebná pro správnou funkci detekčních metod jsou uložená v tabulce `stats` (sekce 5.2.4). Nicméně při běhu analyzátoru se do této tabulky v rámci úspor diskových operací nezapisuje, data zůstávají uložená pouze v paměti. V případě ukončení analyzátoru se data z paměti uloží do tabulky statistik.

5.4 GUI

Pro reprezentaci dat uživateli je použitý nástroj Apache Tomcat. Jedná se o volně dostupný otevřený webový server a servlet kontejner založený na jazyce Java, javových servletech, JSP (Java Server Pages) a EJB (Enterprise JavaBeans). Uživatelské rozhraní je provedené formou webové aplikace, na které se nachází tři stránky.

5.4.1 Události

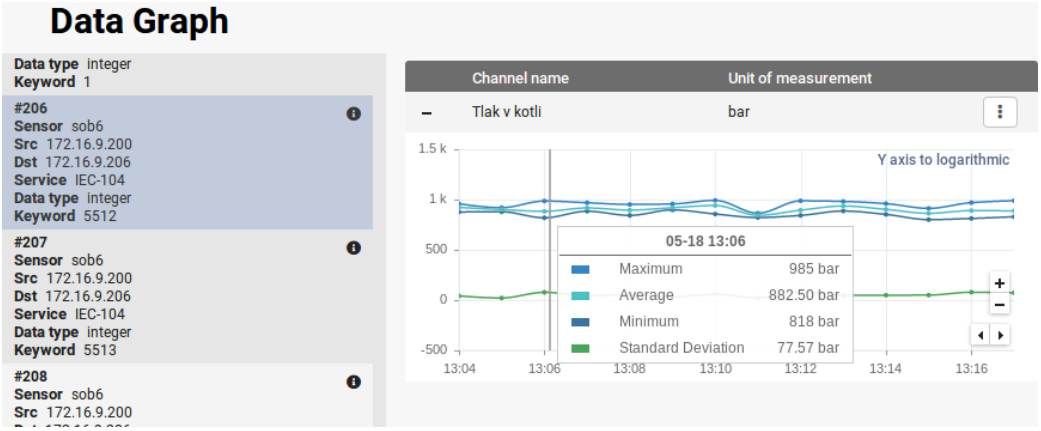
Stránka s události zobrazuje na grafu po minutách seskupené četnosti událostí v závislosti na čase. Pod grafem se nachází tabulka událostí, které může dále poskytnou bližší informace a odůvodnění událostí.



Obrázek 5.6: Screenshot stránky zobrazující detekované událostmi.

5.4.2 Kanály

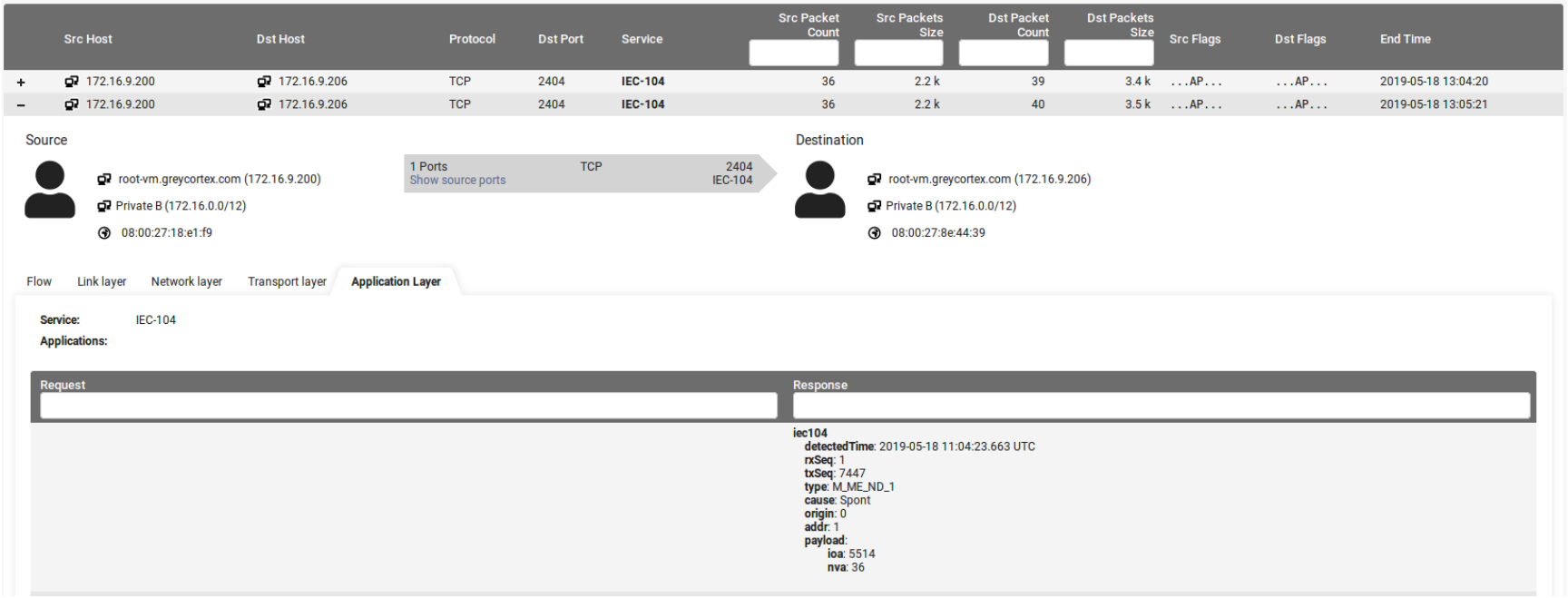
Další stránka zobrazuje podrobné informace o jednotlivých kanálech. Zároveň, pokud si uživatel vybere některý z kanálů, mu bude vykreslen graf s vývojem hodnot v průběhu času. Místo grafu si může uživatel vybrat také tabulku, v které mu budou zobrazena ta naparsovaná data.



Obrázek 5.7: Stránka zobrazuje graf historického vývoj hodnot a jejich základních statistik v závislosti na čase.

5.4.3 Aplikační data

Poslední stránka prezentuje toky, které Suricata ukládá do databáze. Jedná se o analyzátorem ještě nezpracovaná data.



Obrázek 5.8: Stránka zobrazuje toky v takové formě, jak je Suricata ukládá.

Kapitola 6

Návrh detekčních metod

V kapitole 4 jsou popsána různá rozdělení metod detekujících anomálie. Cílem práce je sestavit takový systém, který je schopný se přizpůsobit chování provozu v předem nespecifikované průmyslové síti. Z tohoto důvodu je nutné vybírat mezi metodami, které spadají do kategorie učení bez učitele (viz sekce 4.3.3). Díky tomu nebude nutný manuální zásah, který by nutil uživatele označovat třídami obsáhlé datové instance.

Stejně jako se detekční metody skládají ze dvou fázích, bude i aplikace pracovat ve dvou režimech – trénování a testování. Jelikož jsou anomálie detekovány na základě modelu, je nutné, aby analyzátor nejprve vytvořil model specifický pro konkrétní provoz na dané síti (fáze trénování). Při trénování musíme předpokládat, že v síti je normální provoz bez výskytu anomálií. Při zapnutí režimu testování nově naučený model začíná analyzovat síťovou komunikaci. Dochází k označování nově příchozích datových instancí – instance, které nenáleží naučenému modelu, jsou označené anomální třídou. V průběhu času analyzátor přizpůsobuje model aktuálnímu provozu tak, aby mohl reagovat na případné změny trendu.

Níže jsou popsané atributy a metody, které byly zvolené pro detekci anomálií konkrétních atributů.

6.1 Typy datových analýz

6.1.1 Frekvenční analýza

Komunikace v průmyslových systémech SCADA je vysoce periodická s opakujícími se zprávami [22]. Na základě tohoto předpokladu probíhá detekce, která analyzuje průběh komunikace z hlediska času. Je pravděpodobné, že dojde k malým odchylkám způsobených zatížením sítě či reakční dobou vzdálených zařízení, což by měla detekční metoda zohledňovat.

Vstupními daty frekvenční analýzy je tedy množina časů, kdy přichází jednotlivé datové instance. Pro jednodušší analýzu nedochází k analýze příchozích časů, ale tato data jsou převedena na periody udávající časové rozestupy mezi výskyty hodnot. Teoreticky by periody měly být díky vlastnostem SCADA systému v rámci datových instancí stejné.

Pro detekci anomálií v periodách příchozích dat je zvolená metoda Gaussovského modelu blíže popsaná v sekci 6.2.1.

6.1.2 Analýza hodnot

Analýza hodnot slouží pro případnou detekci podvržení zprávy útočníkem. Vytvořený model uchovává informace o vlastnostech hodnot, které vzdálené zařízení posílá. Analyzovaná data se neskládají z řetězců či struktur, ale z jednoduchých datových typů, které lze převést na numerické hodnoty. Díky tomu lze zjednodušit analýzu pouze nad číselnými datovými řadami.

V tomto typu úloh jsou implementovány dvě metody pro detekci anomálií – statistická metoda Gaussovského modelu (sekce 6.2.1) a klasifikační metoda One-Class SVM (sekce 6.2.2).

Tato analýza by měla detekovat také např. podvržení hodnot, které by mohly následně například ovlivnit rozhodnutí operátora. Výrazné vychýlení hodnoty by mohlo znamenat případně i problémy se senzorem samotným.

6.1.3 Detekce pomocí signatur

Pro detekci útoků a anomálií je také možné využít IDS systém, který monitoruje síťový provoz a snaží se odhalit podezřelé aktivity na základě signatur. Tento způsob jako jediný z navrhovaných nevyužívá model provozu sítě. Detekce probíhá na základě aktuálním paketu, který se porovnává se vzorem signatury. Tato detekce není tedy závislá na stavu.

Pomocí tohoto detektoru můžeme detekovat následující problémy – neoprávněné příkazy od neznámého hosta či z vnější sítě, nesprávný kontrolní součet, kód neznámého objektu či funkce, atd.

6.2 Použité metody

6.2.1 Metoda Gaussovského modelu

Metoda Gaussovského modelu fungují za předpokladu, že datové instance mají normální, neboli Gaussovské rozdělení. Ohodnocení je založené na vzdálenosti analyzované instance dat k odhadovanému průměru. Pro stanovení anomálií se použije práh. Existují mnohé techniky, které počítají vzdálenost k průměru a prahu různými způsoby. Často používaná a zároveň jednoduchá metoda detekce anomálií označuje třídou anomálií všechny datové instance, pro které je jejich vzdálenost větší než 3σ vzdálené od průměru μ, kde σ je směrodatná odchylka. V okolí $\mu \pm \sigma$ se nachází více než 99,7% datových normálních instancí. [11]

Pro snížení četnosti falešných poplachů anomálií je do detekčního algoritmu Gaussovského modelu přidána podmínka variačního koeficientu. Ten je spočítán jako podíl směrodatné odchylky σ a průměrné hodnoty μ. Jestliže je variační koeficient nižší než 40%, předpokládají se příliš odlehlá data, a proto nelze považovat výstup Gaussovského modelu za relevantní.

Výpočet průměru je prováděn na základě metody jednoduchého klouzavého průměru, z kterého je následně dopočítána směrodatná odchylka. Použitím této metody dochází k vyhlazení dat a také průměr více odpovídá aktuálnímu trendu.

6.2.2 One-Class SVM

Jedná se o klasifikační techniku (viz sekce 4.4.1, které k natrénování stačí neoznačená data. Výsledek testování datové instance říká, zda daná instance náleží do naučeného modelu.

Tato metoda je implementovaná s použitím knihovny Weka[1], která je zaměřená na algoritmy strojového učení a dolování dat.

V rámci inicializace One-Class SVM klasifikátoru bylo nutné zvolit parametry pro výpočet modelu. Na testovacích datových sadách byly provedené výpočty, při kterých byl model vyhodnocován pomocí křížové validace. Na základě testování byla ze čtyř možných variant vybrána funkce radiálních bází s hodnotou parametru gamma 0.0012. Za těchto podmínek vracel model nejvíce positivních výsledků.

6.2.3 Metody založené na pravidlech

Díky nástroji Suricata je možné použít pravidla k detekci nejen anomálií, ale i možným útokům na SCADA systémy. Tato metoda funguje jednoduchým způsobem, kdy je sepsané pravidlo, které se porovnává s nově příchozím paketem a pokud dojde k úspěšnému porovnání, je zahlášená anomálie. Existují již sady pravidel pro nástroj SNORT [28], které je ovšem možné použít i v Suricatě. V rámci tohoto projektu byla pravidla přidána a v případně shody se vygeneruje událost.

[1] Knihovna Weka

Kapitola 7

Výsledky testování

Testování probíhalo na třech virtuálních strojích – jeden stroj zachytával síťový provoz, který probíhal mezi dalšími dvěma virtuálními stroji. Komunikace na protokolech DNP3 a IEC 104 byla generovaná pomocí simulátoru Axon Test, který jako jediný vyšel

7.1 Testovací data

Pro testování je nutné získat referenční data, ve kterých bude zachycená komunikace protokolů DNP3 a IEC 60870-5-104.

Veřejné kolekce dat

Nejsnazší cestou, kterou se lze dostat k požadovaným datům, jsou veřejné kolekce dat. Často jsou volně dostupné, takže není problém získat komunikaci různých protokolů.

Prvním zdrojem je databáze komunikačních protokolů, která se nachází na stránkách programu Wireshark [2]. Tento nástroj slouží k monitorování a analýze síťového provozu. Nicméně zde jsou uložené pouze ukázky jednotlivých případů, nenalezneme zde delší komunikaci.

Společnost Netresec [1], která se zabývá vývojem softwaru pro forenzní analýzu síťového provozu, poskytuje další zdroj dat. Ta již nejsou pouze ukázková, ale jedná se o zachycení reálného provozu.

Další zdroj komunikace je publikované v repozitáři [1], ve které se nachází oba protokoly.

Pomocí veřejných kolekcí lze získat různorodou škálu dat, které perfektně poslouží k testování parserů. Nicméně nejsou vhodné pro testování detekčních metod, protože se jedná o data zachycená po krátkou dobu.

Simulátory

Druhým způsobem jak získat potřebná data je použití simulátorů k tomu určených. Díky vlastnímu provozu lze zachytit dlouhotrvající toky či rovnou testovat v reálném čase. Na druhou stranu se jedná pouze o simulátory a v některých případech se mohou výrazně odlišovat od typického provozu.

V rámci této práce bylo vyzkoušeno více simulátorů, každý má své výhody a nevýhody. Jejich srovnání je shrnuté v tabulce 7.1. Ze všech vyzkoušených nástrojů byl pro simulaci komunikačních protokolů zvolen nástroj Axon Test – zásadní vlastností možnost testování

[1] https://github.com/automayt/ICS-pcap

dlouhodobého provoz a možnost simulace obou protokolů. Nicméně i intuitivní ovládání a snadná konfigurace jsou výhodami tohoto nástroje.

Výrobce	Název simulátoru	DNP3	IEC 60870 -5-104	Omezení
Axon Group	Axon Test[2]	✓	✓	45 denní trial verze, omezený počet zařízení.
FreyrSCADA	DNP3 Simulator[3]	✓		Trial verze pouze na 15 minut, žádné jiné omezení.
FreyrSCADA	IEC 60870-5-104 Simulator[4]		✓	
RocyLuo	IEC104TCP[5]		✓	Bez grafického rozhraní, bez manuálu, složitá konfigurace.
Mitra	Simulator IEC 870-5-104[6]		✓	Trial verze pouze na 5 minut.
Graemef	jdnp3[7]	✓		Neintuitivní GUI rozhraní, bez manuálu, složitá konfigurace.

Tabulka 7.1: Srovnání simulačních nástrojů

7.2 Návrh systému

Systém byl implementován podle návrhu popsaném v kapitole 5. Jednotlivé moduly samostatně fungují a správně komunikují přes navržené rozhraní. S návrhem systému tedy žádný problém nebyl.

7.3 Generované události

Nové spojení a kanál

Události nové spojení a nový kanál ve všech testech fungují přesně a spolehlivě.

Nepravidelný čas příchozích dat

Událost generovaná Gaussovským modelem měla stoprocentní úspěšnost detekce anomálie. Události byly ovšem generované, i když se o anomálii nejednalo. Nicméně míra nesprávně určených anomálií je necelé 0.4% ze všech testovaných.

Chybějící data

Tato událost vznikla na základě testů, u kterých se generovalo velké množství událostí hlásící nepravidelný čas příchozích dat. Ukázalo se, že velké množství těchto událostí bylo

[2] http://www.axongroup.com.co/en/productos/axon-test/
[3] http://freyrscada.com/dnp3-ieee-1815-Server-Simulator.php
[4] http://freyrscada.com/iec-60870-5-104.php
[5] https://github.com/RocyLuo
[6] http://www.mitraware.com/
[7] https://sourceforge.net/projects/jdnp3/

generováno správně – problém se nacházel i simulátoru, jak potvrdila i analýza pomocí nástroje Wireshark. Simulátor sice posílá data periodicky, ale někdy se stává, že některou periodu vynechá. Proto vznikla tato metoda, která občasný výpadek zachytí a zahlásí jej jako událost s nízkou závažností.

Anomálie hodnoty na základě statické analýzy

Událost je opět generovaná Gaussovským modelem se stoprocentní úspěšností detekce anomálií. Opět je tu ale problém s nahlášenými událostmi, které anomálie nejsou. Většinou se jedná o krajní hodnoty konkrétního kanálu. Chybně zahlášených anomálií dat, které spadají do normální třídy, je přibližně 0.1%.

Anomálie hodnoty na základě One-Class SVM modelu

Události generované na základě naučeného One-Class SVM modelu. Úspěšnost detekce anomálií závisí na vstupním parametru gamma. Nicméně pro datové řady s různým chováním je velmi obtížné nalézt univerzální parametr, který by vyhovovaly všem řadám. Finálním hodnota byla získána automatickým testováním, z kterého byla vybrána hodnota, které vracela nejvíce správně označených tříd. Bohužel i tato získaný parametr, který je přizpůsobený zpracovávaným řadám, má vysokou četnost falešných poplachů – 34% datových instancí bylo označeno jako anomálie.

Detekční pravidla IDS

Detekční pravidla nebyla všechna otestovaná, protože se mi nepodařilo pro některá pravidla získat data, která by se s pravidly shodovala. Pravidla, která bylo možné otestovat, se chytla vždy na poprvé.

Kapitola 8

Závěr

V úvodu této práce jsme se seznámili se systémy SCADA, které se používají při monitorování a ovládání průmyslových zařízení kritických energetických infrastruktur. Zvláštní pozornost byla zaměřená na protokoly DNP3 a IEC 60870-5-104. Pro úspěšné vypracování této práce je zásadní pochopení vnitřních struktur těchto protokolů, abychom byli schopní provést analýzu, na jejíž základě je možné definovat datové typy a hodnoty. Dále nám pomohla určit možné způsoby detekce anomálií a hrozeb.

K pochopení vlastností SCADA systémů výrazně pomohlo použití simulátoru Axon Test, který jako jediný ze všech testovaných splňoval základní požadavek na bezplatný provoz obou protokolů. Díky simulátoru bylo možné nadefinovat dlouhodobé testy detektorů. Avšak i tento simulátor se nakonec ukázal je nedostatečný, jelikož nesplňuje periodické vlastnosti SCADA systému. Bohužel je velmi obtížné sehnat odchycenou komunikaci, jelikož se často jedná právě o kritické infrastruktury, o kterých nikdo nechce zbytečně zveřejňovat bližší informace.

V této práci jsou navržené celkem tři přístupy detekce anomálií. První přístup je založený na předem definovaných pravidlech, které detekují možné hrozby. Pokud jsou pravidla správně sestavená, je tato metoda velmi úspěšná. Další dva přístupy využívají klasifikační a statistické metody založené na učení bez učitele, kterými detekují anomálie v přenášených hodnotách a časové anomálie při nedodržení periodických vlastností. Testy klasifikační metody One-Class SVM byly neúspěšné – i přes automaticky nastavené parametry pro nejlepší detekci má metoda 34% chybně označených tříd. Naopak statistická metoda měla vynikající výsledky při detekci anomálií a zároveň nízký výskyt falešných poplachů.

Další výzkum detekce hrozeb a anomálií navrhuji zaměřit na analýzu funkcí protokolu DNP3. Ve fázi učení je možné sledovat kódy funkcí a skupiny a variace objektů z aplikačních dat komunikace. V testovací fázi bude takto naučený model porovnávat a vyhodnocovat nově příchozí zprávy. V případě, že se objeví zpráva s příkazem či objektem, který ještě nebyl v předešlé komunikaci zachycen, bude tato zpráva označená za anomálii.

Literatura

[1] Public PCAP files for download.
URL http://www.netresec.com/?page=PcapFiles

[2] SampleCaptures.
URL https://wiki.wireshark.org/SampleCaptures

[3] *650 series, DNP3 Communication Protocol Manual*. ABB AB, Feb 2011.

[4] IEEE Standard for Electric Power Systems Communications-Distributed Network Protocol (DNP3). *IEEE Std 1815-2012 (Revision of IEEE Std 1815-2010)*, Oct 2012: s. 1–821, doi:10.1109/IEEESTD.2012.6327578.

[5] DNP3 Application Note AN2013-004bValidation of Incoming DNP3 Data. Aug 2014.
URL https://www.dnp.org/Portals/0/PublicDocuments/DNP3AN2013-004bValidationofIncomingDNP3Data.pdf

[6] Attacks Targeting Industrial Control Systems (ICS) Up 110 Percent. Apr 2017, [Online; navštíveno 9.01.2018].
URL https://securityintelligence.com/attacks-targeting-industrial-control-systems-ics-up-110-percent/

[7] 4.1. Rules Format. Jun 2018.
URL https://suricata.readthedocs.io/en/suricata-4.1.4/rules/intro.html

[8] Albin, E.; Rowe, N. C.: A realistic experimental comparison of the Suricata and Snort intrusion-detection systems. In *2012 26th International Conference on Advanced Information Networking and Applications Workshops*, IEEE, 2012, s. 122–127.

[9] Amer, M.; Goldstein, M.: Nearest-Neighbor and Clustering based Anomaly Detection Algorithms for RapidMiner. 08 2012, doi:10.5455/ijavms.141.

[10] Balda, P.: SCADA a HMI systémy. ZČU v Plzni, Duben 2007, [Online; navštíveno 2.01.2018].
URL http://vendulka.zcu.cz/Download/Free/IRS1/IRS1-08_SCADA_HMI.pdf

[11] Chandola, V.; Banerjee, A.; Kumar, V.: Anomaly detection: A survey. *ACM Comput. Surv.*, ročník 41, 2009: s. 15:1–15:58.

[12] Chen, T. M.: Stuxnet, the real start of cyber warfare? [Editor's Note]. *IEEE Network*, ročník 24, č. 6, November 2010: s. 2–3, ISSN 0890-8044, doi:10.1109/MNET.2010.5634434.

[13] Chepenko, D.: A Density-based algorithm for outlier detection. Sep 2018.
URL https://towardsdatascience.com/density-based-algorithm-for-outlier-detection-8f278d2f7983

[14] Choudhary, P.: Introduction to Anomaly Detection. Feb 2017.
URL https://www.datascience.com/blog/python-anomaly-detection

[15] Clarke, G. R.; Reynders, D.; Wright, E.: *Practical modern SCADA protocols: DNP3, 60870.5 and related systems*. Engineering : instrumentation & control, Elsevier, 2004.

[16] DenHartog, M.: DNP3 Tutorial - Learn the Industry-Standard SCADA Protocol. Aug 2012, [Online; navštíveno 14.01.2018].
URL http://www.dpstele.com/pdfs/white_papers/dnp3_tutorial.pdf

[17] Dudani, S. A.: The Distance-Weighted k-Nearest-Neighbor Rule. *IEEE Transactions on Systems, Man, and Cybernetics*, ročník SMC-6, č. 4, April 1976: s. 325–327, ISSN 0018-9472, doi:10.1109/TSMC.1976.5408784.

[18] Duffield, N.; Haffner, P.; Krishnamurthy, B.; aj.: Rule-Based Anomaly Detection on IP Flows. 05 2009, s. 424 – 432, doi:10.1109/INFCOM.2009.5061947.

[19] Falliere, N.; Murchu, L. O.; Chien, E.: W32. stuxnet dossier. *White paper, Symantec Corp., Security Response*, ročník 5, č. 6, Feb 2011.
URL http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf

[20] Fischler, M. A.; Bolles, R. C.: Supervisory control and data Acquisition (SCADA) system. *TECHNICAL INFORMATION BULLETIN 04-1*, Říjen 2004.
URL https://scadahacker.com/library/Documents/ICS_Basics/SCADA%20Basics%20-%20NCS%20TIB%2004-1.pdf

[21] Ghafir, I.; Prenosil, V.; Svoboda, J.; aj.: A Survey on Network Security Monitoring Systems. 08 2016.

[22] Goldenberg, N.; Wool, A.: Accurate modeling of Modbus/TCP for intrusion detection in SCADA systems. *International Journal of Critical Infrastructure Protection*, ročník 6, č. 2, 2013: s. 63–75.

[23] Goldstein, M.; Dengel, A.: Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012: Poster and Demo Track*, 2012: s. 59–63.

[24] Han, G.; Xu, B.; Suonan, J.: IEC 61850-Based Feeder Terminal Unit Modeling and Mapping to IEC 60870-5-104. *IEEE Transactions on Power Delivery*, ročník 27, č. 4, Oct 2012: s. 2046–2053, ISSN 0885-8977, doi:10.1109/TPWRD.2012.2209685.

[25] Hartigan, J. A.; Wong, M. A.: Algorithm AS 136: A K-Means Clustering Algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, ročník 28, č. 1, 1979: s. 100–108, ISSN 00359254, 14679876.
URL http://www.jstor.org/stable/2346830

[26] He, Z.; Xu, X.; Deng, S.: Discovering cluster-based local outliers. *Pattern Recognition Letters*, ročník 24, č. 9, 2003: s. 1641 – 1650, ISSN 0167-8655, doi:https://doi.org/10.1016/S0167-8655(03)00003-5.
URL http://www.sciencedirect.com/science/article/pii/S0167865503000035

[27] Ian, P.: *Konec poplašných zpráv o Modré planetě*. Nakladatelství Fragment, 2017, ISBN 9788025324240.

[28] Lin, J.: docker-snort/snortrules-snapshot-2972/rules/protocol-scada.rules. https://github.com/John-Lin/docker-snort/blob/master/snortrules-snapshot-2972/rules/protocol-scada.rules, 6 2015.

[29] Maind, S. B.; Wankar, P.; aj.: Research paper on basic of artificial neural network. *International Journal on Recent and Innovation Trends in Computing and Communication*, ročník 2, č. 1, 2014: s. 96–100.

[30] Maynard, P.; McLaughlin, K.; Haberler, B.: Towards Understanding Man-In-The-Middle Attacks on IEC 60870-5-104 SCADA Networks. In *Proceedings of the 2Nd International Symposium on ICS & SCADA Cyber Security Research 2014*, ICS-CSR 2014, UK: BCS, 2014, ISBN 978-1-78017-286-6, s. 30–42, doi:10.14236/ewic/ics-csr2014.5.
URL https://doi.org/10.14236/ewic/ics-csr2014.5

[31] McMillen, D.: Security attacks on industrial control systems. Oct 2015, [Online; navštíveno 14.01.2018].
URL https://public.dhe.ibm.com/common/ssi/ecm/se/en/sel03046usen/global-technology-services-ibm-security-services-se-research-report-sel03046usen-20170906.pdf

[32] Oman, P.; Phillips, M.: Intrusion Detection and Event Monitoring in SCADA Networks. *IFIP International Federation for Information Processing, Critical Infrastructure Protection*, ročník 253, 2008: s. 161–173, doi:10.1007/978-0-387-75462-8_12.

[33] Pupale, R.: Support Vector Machines(SVM) - An Overview. Jun 2018.
URL https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989

[34] Raschka, S.: Kernel density estimation via the Parzen–Rosenblatt window method. 07 2014, doi:10.13140/2.1.3720.8962.

[35] Sajid, A.; Abbas, H.; Saleem, K.: Cloud-Assisted IoT-Based SCADA Systems Security: A Review of the State of the Art and Future Challenges. *IEEE Access*, ročník 4, 2016: s. 1375–1384, ISSN 2169-3536, doi:10.1109/ACCESS.2016.2549047.

[36] Vomel, J.: Úvod do bayesovských sítí. Oct 2008.
URL staff.utia.cas.cz/vomlel/slides/presentace-medic-info-new-1250.pdf

Příloha A

Aplikační data protokolu DNP3

Aplikační data uložená ve formátu JSON, tak jak je Suricata ukládá do databázového uložiště. Jsou zde definovány tři různé typy dat:

- *double* – kanál definovaný skupinou 32, variací 8 a indexem 0 přenáší hodnotu 1.0
- *timestamp* – kanál definovaný skupinou 51, variací 1 a indexem 0 přenáší čas ve formátu unixové epochy 1516732837324
- *integer* – kanál definovaný skupinou 22, variací 6 a indexem 1 přenáší aktuální hodnotou čítače 63


```
{
  "response": {
    "type": "response",
    "detected_time": "2019-05-06 07:40:58.977 UTC",
    "control": {
      "dir": false,
      "pri": true,
      "fcb": false,
      "fcv": false,
      "function_code": 4
    },
    "src": 4,
    "dst": 3,
    "application": {
      "control": {
        "fir": true,
        "fin": true,
        "con": true,
        "uns": false,
        "sequence": 2
      },
      "function_code": 129,
      "objects": [
        {
          "group": 32,
```

```
        "variation": 8,
        "qualifier": 40,
        "prefix_code": 2,
        "range_code": 8,
        "start": 0,
        "stop": 0,
        "count": 1,
        "points": [
          {
            "prefix": 0,
            "index": 0,
            "online": 1,
            "restart": 0,
            "comm_lost": 0,
            "remote_forced": 0,
            "local_forced": 0,
            "over_range": 0,
            "reference_err": 0,
            "reserved0": 0,
            "value": 1.000000,
            "timestamp": 1516732837324
          }
        ]
      },
      {
        "group": 51,
        "variation": 1,
        "qualifier": 7,
        "prefix_code": 0,
        "range_code": 7,
        "start": 0,
        "stop": 0,
        "count": 1,
        "points": [
          {
            "prefix": 0,
            "index": 0,
            "timestamp": 1516732837324
          }
        ]
      },
      {
        "group": 22,
        "variation": 6,
        "qualifier": 40,
        "prefix_code": 2,
        "range_code": 8,
        "start": 0,
```

```
          "stop": 0,
          "count": 1,
          "points": [
            {
              "prefix": 1,
              "index": 1,
              "online": 1,
              "restart": 0,
              "comm_lost": 0,
              "remote_forced": 0,
              "local_forced": 0,
              "rollover": 0,
              "discontinuity": 0,
              "reserved0": 0,
              "count": 63,
              "timestamp": 1516732837324
            }
          ]
        }
      ],
      "complete": true
    }
  }
}
```

Příloha B

Aplikační data protokolu IEC 60870-5-104

Aplikační data uložená ve formátu JSON, tak jak je Suricata ukládá do databázového uložiště. Jedná se o celočíselná data s identifikátory 1 a 1001.


```
{
  "iec104": {
    "detected_time": "2019-04-29 17:20:09.760 UTC",
    "rx_seq": 1,
    "tx_seq": 16811,
    "addr": 1,
    "origin": 0,
    "type": "M_DP_TB_1",
    "cause": "Spont",
    "payload": [
      {
        "ioa": 1,
        "diq": 2,
        "cp56": "2019-04-29T17:20:09.680+00:00"
      },
      {
        "ioa": 1001,
        "diq": 1,
        "cp56": "2019-04-29T17:20:09.680+00:00"
      }
    ]
  }
}
```

Příloha C

Seznam objektů protokolu DNP3 [5][4]

Group	Variation	Type	Description	Size	Parsed data type
			Binary		
1 (0x01)	1 (0x01)	Static	Binary Input - Packed Format	1 bit	boolean
1 (0x01)	2 (0x02)	Static	Binary Input - Status with Flags	1 octet	boolean
2 (0x02)	1 (0x01)	Event	Binary Input Event	1 octet	boolean
2 (0x02)	2 (0x02)	Event	Binary Input Event - with Absolute Time	7 octets	boolean
2 (0x02)	3 (0x03)	Event	Binary Input Event - with Relative Time	3 octets	boolean
3 (0x03)	1 (0x01)	Static	Double-bit Binary Input - Packed Format	2 bits	boolean
3 (0x03)	2 (0x02)	Static	Double-bit Binary Input - Status with Flags	1 octet	boolean
4 (0x04)	1 (0x01)	Event	Double-bit Binary Input Event	1 octet	boolean
4 (0x04)	2 (0x02)	Event	Double-bit Binary Input Event with Absolute Time	7 octets	boolean
4 (0x04)	3 (0x03)	Event	Double-bit Binary Input Event with Relative Time	3 octets	boolean
10 (0x0A)	1 (0x01)	Static	Binary Output - Packed Format	1 bit	boolean
10 (0x0A)	2 (0x02)	Static	Binary Output - Status with Flags	1 octet	boolean
11 (0x0B)	1 (0x01)	Event	Binary Output Event - Status	1 octet	boolean
11 (0x0B)	2 (0x02)	Event	Binary Output Event - Status with Time	7 octets	boolean
12 (0x0C)	1 (0x01)	Command	Binary Output Command - Control Relay Output Block	11 octets	boolean

Pokračování na další stránce...

Tabulka C.1 – *Pokračování předchozí stránky...*

Group	Variation	Type	Description	Size	Parsed data type
12 (0x0C)	2 (0x02)	Command	Binary Output Command - Pattern Control Block	11 octets	boolean
12 (0x0C)	3 (0X03)	Command	Binary Output Command - Pattern Mask	n bits	boolean
13 (0x0D)	1 (0x01)	Event	Binary Output Command Event - Command Status	1 octet	boolean
13 (0x0D)	2 (0x02)	Event	Binary Output Command Event - Command Status with Time	7 octets	boolean
Counters					
20 (0x14)	1 (0x01)	Static	Counter - 32-bit with Flag	5 octets	integer
20 (0x14)	2 (0x02)	Static	Counter - 16-bit with Flag	3 octets	integer
20 (0x14)	3 (0x03)	Static	Counter - 32-bit with Flag and Delta	5 octets	integer
20 (0x14)	4 (0x04)	Static	Counter - 16-bit with Flag and Delta	3 octets	integer
20 (0x14)	5 (0x05)	Static	Counter - 32-bit w/o Flag	4 octets	integer
20 (0x14)	6 (0x06)	Static	Counter - 16-bit w/o Flag	2 octets	integer
20 (0x14)	7 (0x07)	Static	Counter - 32-bit w/o Flag and Delta	4 octets	integer
20 (0x14)	8 (0x08)	Static	Counter - 16-bit w/o Flag and Delta	2 octets	integer
21 (0x15)	1 (0x01)	Static	Frozen Counter - 32-bit with Flag	5 octets	integer
21 (0x15)	2 (0x02)	Static	Frozen Counter - 16-bit with Flag	3 octets	integer
21 (0x15)	3 (0x03)	Static	Frozen Counter - 32-bit with Flag and Delta	5 octets	integer
21 (0x15)	4 (0x04)	Static	Frozen Counter - 16-bit with Flag and Delta	3 octets	integer
21 (0x15)	5 (0x05)	Static	Frozen Counter - 32-bit with Flag and Time	11 octets	integer
21 (0x15)	6 (0x06)	Static	Frozen Counter - 16-bit with Flag and Time	9 octets	integer
21 (0x15)	7 (0x07)	Static	Frozen Counter - 32-bit with Flag and Time and Delta	11 octets	integer
21 (0x15)	8 (0x08)	Static	Frozen Counter - 16-bit with Flag and Time and Delta	9 octets	integer
21 (0x15)	9 (0x09)	Static	Frozen Counter - 32-bit w/o Flag	4 octets	integer
21 (0x15)	10 (0x0A)	Static	Frozen Counter - 16-bit w/o Flag	2 octets	integer
21 (0x15)	11 (0x0B)	Static	Frozen Counter - 32-bit w/o Flag and Delta	4 octets	integer
21 (0x15)	12 (0x0C)	Static	Frozen Counter - 16-bit w/o Flag and Delta	2 octets	integer
22 (0x16)	1 (0x01)	Event	Counter Event - 32-bit with Flag	5 octets	integer
22 (0x16)	3 (0x03)	Event	Counter Event - 16-bit with Flag	3 octets	integer
22 (0x16)	4 (0x04)	Event	Counter Event - 32-bit with Flag and Delta	5 octets	integer
22 (0x16)	2 (0x02)	Event	Counter Event - 16-bit with Flag and Delta	3 octets	integer

Pokračování na další stránce...

Tabulka C.1 – *Pokračování předchozí stránky...*

Group	Variation	Type	Description	Size	Parsed data type
22 (0x16)	5 (0x05)	Event	Counter Event - 32-bit with Flag and Time	11 octets	integer
22 (0x16)	6 (0x06)	Event	Counter Event - 16-bit with Flag and Time	9 octets	integer
22 (0x16)	7 (0x07)	Event	Counter Event - 32-bit with Flag and Time and Delta	11 octets	integer
22 (0x16)	8 (0x08)	Event	Counter Event - 16-bit with Flag and Time and Delta	9 octets	integer
23 (0x17)	1 (0x01)	Event	Frozen Counter Event - 32-bit with Flag	5 octets	integer
23 (0x17)	2 (0x02)	Event	Frozen Counter Event - 16-bit with Flag	3 octets	integer
23 (0x17)	3 (0x03)	Event	Frozen Counter Event - 32-bit with Flag and Delta	5 octets	integer
23 (0x17)	4 (0x04)	Event	Frozen Counter Event - 16-bit with Flag and Delta	3 octets	integer
23 (0x17)	5 (0x05)	Event	Frozen Counter Event - 32-bit with Flag and Time	11 octets	integer
23 (0x17)	6 (0x06)	Event	Frozen Counter Event - 16-bit with Flag and Time	9 octets	integer
23 (0x17)	7 (0x07)	Event	Frozen Counter Event - 32-bit with Flag and Time and Delta	11 octets	integer
23 (0x17)	8 (0x08)	Event	Frozen Counter Event - 16-bit with Flag and Time and Delta	9 octets	integer
Analog Input					
30 (0x1E)	1 (0x01)	Static	Analog Input - 32-bit with Flag	5 octets	integer
30 (0x1E)	2 (0x02)	Static	Analog Input - 16-bit with Flag	3 octets	integer
30 (0x1E)	3 (0x03)	Static	Analog Input - 32-bit w/o Flag	4 octets	integer
30 (0x1E)	4 (0x04)	Static	Analog Input - 16-bit w/o Flag	2 octets	integer
30 (0x1E)	5 (0x05)	Static	Analog Input - Single-prec. FP with Flag	5 octets	float
30 (0x1E)	6 (0x06)	Static	Analog Input - Double-prec. FP with Flag	9 octets	float
31 (0x1F)	1 (0x01)	Static	Frozen Analog Input - 32-bit with Flag	5 octets	integer
31 (0x1F)	2 (0x02)	Static	Frozen Analog Input - 16-bit with Flag	3 octets	integer
31 (0x1F)	3 (0x03)	Static	Frozen Analog Input - 32-bit with Time-of-Freeze	11 octets	integer
31 (0x1F)	4 (0x04)	Static	Frozen Analog Input - 16-bit with Time-of-Freeze	9 octets	integer
31 (0x1F)	5 (0x05)	Static	Frozen Analog Input - 32-bit w/o Flag	4 octets	integer
31 (0x1F)	6 (0x06)	Static	Frozen Analog Input - 16-bit w/o Flag	2 octets	integer
31 (0x1F)	7 (0x07)	Static	Frozen Analog Input - Single-prec. FP with Flag	5 octets	float
31 (0x1F)	8 (0x08)	Static	Frozen Analog Input - Double-prec. FP with Flag	9 octets	float
32 (0x20)	1 (0x01)	Event	Analog Input Event - 32-bit	5 octets	integer
32 (0x20)	2 (0x02)	Event	Analog Input Event - 16-bit	3 octets	integer

Pokračování na další stránce...

Tabulka C.1 – *Pokračování předchozí stránky...*

Group	Variation	Type	Description	Size	Parsed data type
32 (0x20)	3 (0x03)	Event	Analog Input Event - 32-bit with Time	11 octets	integer
32 (0x20)	4 (0x04)	Event	Analog Input Event - 16-bit with Time	9 octets	integer
32 (0x20)	5 (0x05)	Event	Analog Input Event - Single-prec. FP	5 octets	float
32 (0x20)	6 (0x06)	Event	Analog Input Event - Double-prec. FP	9 octets	float
32 (0x20)	7 (0x07)	Event	Analog Input Event - Single-prec. FP with Time	11 octets	float
32 (0x20)	8 (0x08)	Event	Analog Input Event - Double-prec. FP with Time	15 octets	float
33 (0x21)	1 (0x01)	Event	Frozen Analog Input Event - 32-bit	5 octets	integer
33 (0x21)	2 (0x02)	Event	Frozen Analog Input Event - 16-bit	3 octets	integer
33 (0x21)	3 (0x03)	Event	Frozen Analog Input Event - 32-bit with Time	11 octets	integer
33 (0x21)	4 (0x04)	Event	Frozen Analog Input Event - 16-bit with Time	9 octets	integer
33 (0x21)	5 (0x05)	Event	Frozen Analog Input Event - Single-prec. FP	5 octets	float
33 (0x21)	6 (0x06)	Event	Frozen Analog Input Event - Double-prec. FP	9 octets	float
33 (0x21)	7 (0x07)	Event	Frozen Analog Input Event - Single-prec. FP with Time	11 octets	float
33 (0x21)	8 (0x08)	Event	Frozen Analog Input Event - Double-prec. FP with Time	15 octets	float
34 (0x22)	1 (0x01)	Static	Analog Input Deadband - 16-bit	2 octets	integer
34 (0x22)	2 (0x02)	Static	Analog Input Deadband - 32-bit	4 octets	integer
34 (0x22)	3 (0x03)	Static	Analog Input Deadband - Single-prec. FP	4 octets	float
Analog Output					
40 (0x28)	1 (0x01)	Static	Analog Output Status - 32-bit with Flag	5 octets	integer
40 (0x28)	2 (0x02)	Static	Analog Output Status - 16-bit with Flag	3 octets	integer
40 (0x28)	3 (0x03)	Static	Analog Output Status - Single-prec. FP with Flag	5 octets	float
40 (0x28)	4 (0x04)	Static	Analog Output Status - Double-prec. FP with Flag	9 octets	float
41 (0x29)	1 (0x01)	Command	Analog Output Command - 32-bit	5 octets	integer
41 (0x29)	2 (0x02)	Command	Analog Output Command - 16-bit	3 octets	integer
41 (0x29)	3 (0x03)	Command	Analog Output Command - Single-prec. FP	5 octets	float
41 (0x29)	4 (0x04)	Command	Analog Output Command - Double-prec. FP	9 octets	float
42 (0x2A)	1 (0x01)	Event	Analog Output Event - 32-bit	5 octets	integer
42 (0x2A)	2 (0x02)	Event	Analog Output Event - 16-bit	3 octets	integer
42 (0x2A)	3 (0x03)	Event	Analog Output Event - 32-bit with Time	11 octets	integer

Pokračování na další stránce...

Tabulka C.1 – *Pokračování předchozí stránky...*

Group	Variation	Type	Description	Size	Parsed data type
42 (0x2A)	4 (0x04)	Event	Analog Output Event - 16-bit with Time	9 octets	integer
42 (0x2A)	5 (0x05)	Event	Analog Output Event - Single-prec. FP	5 octets	float
42 (0x2A)	6 (0x06)	Event	Analog Output Event - Double-prec. FP	9 octets	float
42 (0x2A)	7 (0x07)	Event	Analog Output Event - Single-prec. FP with Time	11 octets	float
42 (0x2A)	8 (0x08)	Event	Analog Output Event - Double-prec. FP with Time	15 octets	float
43 (0x2B)	1 (0x01)	Event	Analog Output Command Event - 32-bit	5 octets	integer
43 (0x2B)	2 (0x02)	Event	Analog Output Command Event - 16-bit	3 octets	integer
43 (0x2B)	3 (0x03)	Event	Analog Output Command Event - 32-bit with Time	11 octets	integer
43 (0x2B)	4 (0x04)	Event	Analog Output Command Event - 16-bit with Time	9 octets	integer
43 (0x2B)	5 (0x05)	Event	Analog Output Command Event - Single-prec. FP	5 octets	float
43 (0x2B)	6 (0x06)	Event	Analog Output Command Event - Double-prec. FP	9 octets	float
43 (0x2B)	7 (0x07)	Event	Analog Output Command Event - Single-prec. FP with Time	11 octets	float
43 (0x2B)	8 (0x08)	Event	Analog Output Command Event - Double-prec. FP with Time	15 octets	float
Timestamp					
50 (0x32)	1 (0x01)	Info	Time and Date - Absolute Time	6 octets	timestamp
50 (0x32)	2 (0x02)	Info	Time and Date - Absolute Time and Interval	10 octets	timestamp
50 (0x32)	3 (0x03)	Info	Time and Date - Absolute Time at Last Recorded Time	6 octets	timestamp
50 (0x32)	4 (0x04)	Info	Time and Date - Indexed Absolute Time and Long Interval	11 octets	timestamp
51 (0x33)	1 (0x01)	Info	Time and Date CTO - Absolute Time, Synchronized	6 octets	timestamp
51 (0x33)	2 (0x02)	Info	Time and Date CTO - Absolute Time, Unsynchronized	6 octets	timestamp
52 (0x34)	1 (0x01)	Info	Time Delay Coarse	2 octets	integer
52 (0x34)	2 (0x02)	Info	Time Delay Fine	2 octets	integer
Classes					
60 (0x3C)	1 (0x01)	Info	Class Objects - Class 0 Data		none
60 (0x3C)	2 (0x02)	Info	Class Objects - Class 1 Data		none
60 (0x3C)	3 (0x03)	Info	Class Objects - Class 2 Data		none
60 (0x3C)	4 (0x04)	Info	Class Objects - Class 3 Data		none
Devices					
80 (0x50)	1 (0x01)	Static	Internal Indications - Packed Format	2 octets	integer