VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2019 Jan Hudec

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

GENERÁTOR ÚTOKŮ NA SCADA PROTOKOLY
GENERATOR FOR SIMULATION OF SCADA ATTACKS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR
Jan Hudec

VEDOUCÍ PRÁCE
SUPERVISOR
Ing. Petr Blažek

BRNO 2019

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní obor **Informační bezpečnost**
Ústav telekomunikací

Student: Jan Hudec ***ID:*** 195152
Ročník: 3 ***Akademický rok:*** 2018/19

NÁZEV TÉMATU:

Generátor útoků na SCADA protokoly

POKYNY PRO VYPRACOVÁNÍ:

Bakalářská práce je zaměřena na realizace síťových útoků vyskytujících se v systémech SCADA. Cílem bakalářské práce je návrh pracoviště, které bude realizovat útoky na zařízení komunikující protokoly DNP3, IEC 60870 nebo IEC 61850. Dílčím cílem je realizace zařízení vyskytujících se ve SCADA systémech (např. PLC, čidla, elektroměry atd.), na které budou směřovány vytvořené útoky. V teoretické části bakalářské práce nastudujte zvolené SCADA protokoly a útoky na tyto protokoly. V praktické části proveďte návrh pracoviště, implementujte zvolené SCADA útoky a realizujte simulovaná zařízení. Výstupem bakalářské práce bude navržené pracoviště, které bude generovat alespoň pět síťových útoků na minimálně tři simulovaná zařízení výše zmíněnými protokoly.

DOPORUČENÁ LITERATURA:

[1] MAKHIJA, Jay; SUBRAMANYAN, L. R. Comparison of protocols used in remote monitoring: DNP 3.0, IEC 870-5-101 & Modbus. Electronics Systems Group, IIT Bombay, India, Tech. Rep, 2003.

[2] UZAIR, Muhammad. COMMUNICATION METHODS (PROTOCOLS, FORMAT & LANGUAGE) FOR THE SUBSTATION AUTOMATION & CONTROL (Project report of course 586 b) Dostupné z: http://www.eng.uwo.ca/people/tsidhu/Documents/project%20report%20Uzair.pdf

Termín zadání: 1.2.2019 ***Termín odevzdání:*** 12.8.2019

Vedoucí práce: Ing. Petr Blažek
Konzultant:

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

Abstrakt

Bakalářská práce se zabývá realizací generátoru útoků na protokol IEC 61850 s využitím knihovny libpcap. Součástí řešení je simulace zařízení vyskytujících se ve SCADA systémech. Teoretická část přibližuje strukturu protokolů DNP3, IEC 60870 a IEC 61850 a dále popisuje vybrané útoky. V praktické části dochází k realizaci útoků zaměřených na komunikaci mezi uzly SCADA systému, konkrétně na GOOSE a Sampled Values. Simulovaná zařízení generují komunikaci s využitím knihovny libiec61850, na kterou jsou pak útoky cíleny.

Klíčová slova

Bezpečnostní slabiny, IEC 61850, generátor, GOOSE, libpcap, Sampled Values, simulace

Abstract

Output of this bachelor's thesis is an attack generator aimed on the IEC 61850 protocol using the libpcap library. Part of the output is also a simulation of devices which are normally present in SCADA systems. Theoretic part aims to explain protocols DNP3, IEC 60870 and IEC 61850 more in-depth and further focus is on the attacks. Practical part contains attacks implementation that aim on communication between SCADA units, more precisely on GOOSE and Sampled Values frames. Simulated devices generate this communication using the libiec61850 library. Attacks are aimed on that communication.

Keywords

Security weakness, IEC 61850, generator, GOOSE, libpcap, Sampled Values, simulation

Bibliografická citace

HUDEC, Jan. *Generátor útoků na SCADA protokoly* [online]. Brno, 2019 [cit. 2019-08-08]. Dostupné z: https://www.vutbr.cz/studenti/zav-prace/detail/121397. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce Petr Blažek.

Prohlášení autora o původnosti díla

Prohlašuji, že svou bakalářskou práci na téma Generátor útoků na SCADA protokoly jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: 27. července 2019

.....
podpis autora

Poděkování

Děkuji vedoucímu mé bakalářské práce panu Ing. Petru Blažkovi za cenné rady a odbornou pomoc během zpracování této práce. Děkuji také mé rodině za podporu, trpělivost a pochopení.

V Brně dne: 27. července 2019 .....

podpis autora

Obsah

Seznam symbolů a zkratek

APDU	…	Application Protocol Data Unit
ASDU	…	Application Service Data Unit
ASN.1	…	Abstract Syntax Notation One
CDC	…	Common Data Class
CSMA/CD	…	Carrier Sense Multiple Access/Collision Detection
DNP	…	Distributed Network Protocol
EPA	…	Enhanced Performance Model
GOOSE	…	Generic Object Oriented Substation Events
GSSE	…	Generic Substation State Events
HMI	…	Human-Machine Interface
IEC	…	International Electrotechnical Commission
IED	…	Intelligent Electronic Device
IP	…	Internet Protocol
ISO	…	International Organization for Standardization
MMS	…	Manufacturing Message Specification
OSI	…	Open Systems Interconnection
RTU	…	Remote Terminal Unit
SCADA	…	Supervisory Control and Data Acquisition
TLV	…	Type-Length-Value
VLAN	…	Virtual Local Area Network
XML	…	Extensible Markup Language

Seznam obrázků

Seznam tabulek

1. ÚVOD

Ve velkém ekosystému zařízení, kde je potřeba zachovat spolehlivost a funkčnost, hraje klíčovou roli automatizace a efektivní možnost kontroly připojených zařízení. Tyto požadavky plní systém SCADA (Supervisory Control and Data Acquisition), jehož funkcionalita zahrnuje dispečerské řízení a sběr dat.

Výsledkem snahy o jednoduché propojení jednotlivých zařízení vzniklo několik komunikačních protokolů. Důraz byl ovšem kladen na efektivní přenos dat, nikoliv na jejich zabezpečení, proto jsou tyto protokoly více náchylné na cílené útoky. V závislosti na typu útoku a záměrům útočníka se liší rozměr útoku a jeho následky. Záleží také na tom, jak moc ofenzivní útok bude, kdy si jej bezpečnostní složky systému všimnou a jak na něj zareagují. Některé útoky mohou být přehlíženy relativně dlouhou dobu, zvlášť pokud se jedná o pasivní sběr informací a odesílání dat zpět útočníkovi, nebo malware, jehož cílem je šíření systémem. Naopak vysoce ofenzivní útok spoléhá spíše na jeho efektivitu a rychlé odzbrojení cílového systému než na nenápadnost.

Z minulosti lze jmenovat především útoky Shamoon a BlackEnergy. Cílem těchto útoků bylo šířit se v síti, mazat data, provádět špionáž v systému a posílat zpět útočníkovi informace, tudíž byly poměrně komplexním balíkem se širokým spektrem možností. I přesto, že Shamoon nebyl původně cílen přímo na SCADA systémy, byl velikou hrozbou z důvodu mazání dat na cílových stanicích. To znamenalo omezení provozu stanice a následně i části systému nebo systému jako celku z důvodu provázanosti jednotlivých stanic, které na sobě závisí. Výpadek systému velkého rozsahu, ať už v energetice nebo jiném průmyslu, byť i na krátkou dobu, může mít nedozírné následky, jak v podobě ušlého finančního zisku, ukradených informací, nebo v nejhorším případě i ztrátu lidských životů.

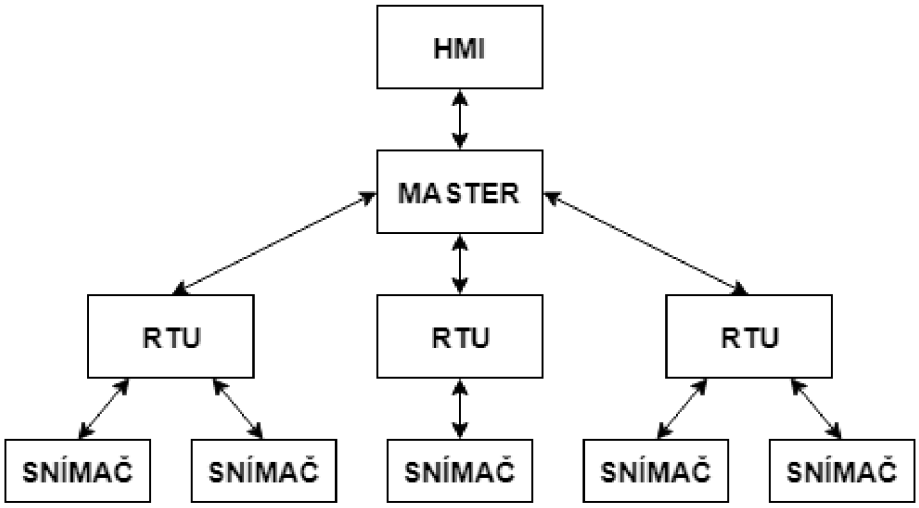
Tato práce se zabývá analýzou bezpečnostních slabin protokolu IEC 61850, návrhem útoků na tyto slabiny a dalšího přiblížení, jak z technické stránky, tak i po stránce provedení. Práce obsahuje popis realizace těchto útoků v simulovaném prostředí včetně jeho popisu a nastavení.

2. POPIS SYSTÉMU SCADA

Systém SCADA (Supervisory Control and Data Acquisition) je síť zařízení, která plní předem stanovenou funkci. Jedná se například o funkce monitorovací, ovládací, ochranné, či měřící. Dohromady zařízení, která jsou na sobě závislá, tvoří jeden funkční celek. Obecné rozvržení systému je znázorněno na *Obr. 2-1*. Komunikace mezi zařízeními probíhá organizovaně na základě určitého protokolu, kterým se řídí celý systém.

Komunikační protokol obecně umožňuje zařízením komunikovat mezi sebou. Tato zařízení však musí daný protokol podporovat. Jakýkoliv rozdíl v implementaci protokolu může vést k nežádoucím chybám v komunikaci. Pokud jsou všechna zařízení od stejného výrobce, většinou nenastane problém, jelikož využívají stejný protokol. Jeden výrobce ovšem nemusí nabízet všechna zařízení, která jsou v systému potřeba, tudíž je nutno využít zařízení od jiného výrobce, kdy může nastat závažný problém v komunikaci mezi stávajícími zařízeními z důvodu odlišného komunikačního protokolu. Z těchto důvodů potřeby otevřeného systému, kde je možno zapojit zařízení od různých výrobců, vznikl standardizovaný komunikační protokol.

Kromě benefitů standardizovaného protokolu, jako je nezávislost na výrobci, otevřenost systému, či dostupnost dokumentací a informací, přináší i nevýhody v podobě větších režií při přenosu dat a možnosti nevyužití plného potencionálu zařízení z důvodu chybějící implementace funkcionality. [1]

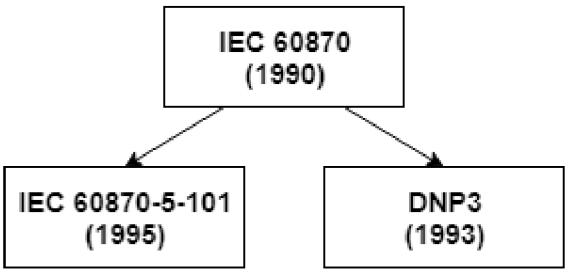


Obrázek 2-1 Zjednodušené schéma systému SCADA.

3. IEC 60870-5

Mezinárodní standard vytvořen technickou komisí 57 spadající do organizace IEC (International Electrotechnical Commission) s primárním zaměřením pro energetické systémy. Zahrnuje dálkové řízení, ochranu a přidružené telekomunikační služby. Je postaven na základě kolekce standardů IEC 60870-5, která poskytuje komunikaci mezi dvěma systémy prostřednictvím jednoduchých zpráv.

Paralelně se tímto protokolem vznikal i protokol DNP3 (Distributed Network Protocol), který byl vyvíjen jinými organizacemi. Založen byl ale taktéž na standardu IEC 60870, proto sdílí určité podobnosti na nižších vrstvách funkcionality. Ve vyšších vrstvách ale pracují naprosto odlišně. Tento protokol má hojné zastoupení především v Evropě. [3] Paralelní vývoj protokolů IEC 60870-5-101 a DNP3 znázorňuje *Obr. 3-1*.



Obrázek 3-1 Vývoj standardů IEC 60780-5-101 a DNP3.

3.1 Specifikace IEC 60870-5

Dělí se na 5 dílčích dokumentů:

- IEC 60870-5-1 – Formáty přenosového rámce – popisuje 4 formáty rámců, každý z nich má jinou míru zabezpečení proti chybám, definuje fixní a proměnnou délku rámců,
- IEC 60870-5-2 – Služby přenosu – definuje pojmy vyvážený a nevyvážený přenos,
- IEC 60870-5-3 – Obecná struktura aplikačních dat – zahrnuje APDU objekt, popisuje obecnou strukturu pro aplikační data a vytváří pravidla pro jednotky aplikačních dat,
- IEC 60870-5-4 – Definice a kódování informačních prvků – definuje objekty reprezentující informaci, které mohou být využity, vytváří stavební bloky, ze kterých lze následně poskládat celý informační objekt,

– IEC 60870-5-5 – Základní aplikační funkce – popisuje funkce nejvyšší vrstvy přenosového protokolu, zahrnuje inicializaci, metody získání dat, synchronizaci a přenos příkazů.

3.2 IEC 60870-5-101

Popis níže uvedeného standardu IEC 60870-5 vychází z [2] a [4].

K přenosu dat využívá architekturu EPA (Enhanced Performance Model), která vychází z modelu ISO/OSI (International Organization for Standardization/Open Systems Interconnection). K přenosu dat využívá asynchronní sériový přenos dat. Vhodný je zejména pro zařízení zapojené jako bod-bod nebo hvězda.

Komunikace je založena na principu nadřízené a podřízené stanice (master-slave), přičemž lze dodatečně nastavit typ výměny informací na vyvážený (komunikaci může zahájit jak nadřízená stanice, tak podřízená stanice) a nevyvážený (komunikaci může zahájit pouze nadřízená stanice).

Ze standardu IEC 60780-5 vychází také funkce, které jsou pro dálkové řízené systémy důležité – dotazování (polling), hlášení na základě výjimky (report by exception) a přiřazení časových razítek.

3.2.1 Dotazování

Nadřízená stanice v pravidelných intervalech rozesílá požadavky na jednotlivé podřízené stanice, které na přijatý dotaz odpoví. Požadavek lze individuálně upravit pro konkrétní stanici. Touto metodou nadřízená stanice sbírá aktuální údaje z připojených zařízení.

3.2.2 Hlášení na základě výjimky

Pokud se na podřízené stanici vyskytne náhlá změna hodnoty nebo chyba, je žádoucí, aby se nadřízená stanice o této skutečnosti dozvěděla co nejdříve. K tomu slouží funkce hlášení na základě výjimky, kdy může podřízená stanice požádat o zahájení komunikace.

Pokud by tato funkce neexistovala, nadřízená stanice by se o dané události dozvěděla až tehdy, kdy by vyslala požadavek na postiženou podřízenou stanici v rámci pravidelného dotazování, tzv. pollingu.

3.2.3 Přiřazení časových razítek

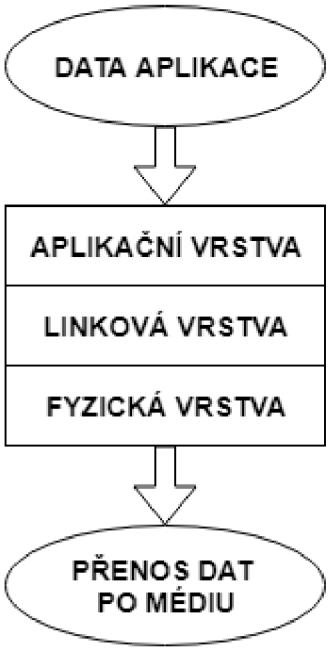
K událostem, které odesílají podřízené stanice, také připojí jedinečný časový identifikátor, který je vitální součástí v případě, že v systému nastane chyba. Podle

tohoto údaje lze přesně dohledat, kdy a u které stanice vznikl problém či nevyžádaná změna.

Při využití tohoto mechanismu je třeba dbát na správnou implementaci, neboť doba vzniku události a doba přijetí zprávy o této události může být zkreslena například vlivem přenosu zprávy po médiu. Musí být zajištěna časová synchronizace mezi stanicemi.

3.3 Architektura EPA

Navržen a vytvořen technickou komisí 57 organizace IEC, je třívrstvým komunikačním modelem vycházející ze sedmivrstvého modelu ISO/OSI. Zahrnuje aplikační vrstvu, linkovou vrstvu a fyzickou vrstvu. Jeho účelem je rychlejší a efektivnější přenos dat. Vizuální interpretace modelu je zobrazena na *Obr. 3-2*.



Obrázek 3-2 Model EPA.

3.3.1 Aplikační vrstva

Pracuje s přijatými daty a s daty pro odeslání zapouzdřenými v objektu ASDU (Application Service Data Unit). Poskytuje základní funkce, které jsou definované v IEC 60870-5-5. Výrobce nemá možnost jakkoliv přidat či upravit definované funkce a objekty reprezentující informaci.

3.3.2 Linková vrstva

Poskytuje spolehlivý přenos dat po fyzickém médiu, kontrolu toku dat a detekci chyb. Pracuje s rámci dat, využíván je rámec typu FT1.2, který je definován v IEC 60870-5-1.

Při vyváženém přenosu může komunikaci zahájit jak nadřízená stanice, tak podřízená stanice. Je nutné implementovat metodu detekce kolizí, například plně-duplexním připojením (RS-232) nebo protokolem CSMA/CD (Carrier Sense Multiple Access/Collision Detection) na fyzické vrstvě.

Nevyvážený přenos je specifický tím, že komunikaci může zahájit pouze stanice nadřízená. Tím odpadá riziko vzniku kolizí.

3.3.3 Fyzická vrstva

Fyzické medium, přes které dochází k samotnému přenosu dat. Přenášenou jednotkou je bit, který je reprezentován signálem. Využívá standardu RS-232, nebo RS-485.

4. DNP3

Počátek vzniku již v roce 1990 společností Westronic, kdy byl veden jako proprietární. V roce 1993 se licence změnila na otevřenou a byl vyvíjen spolkem DNP3 Users Group.

Primárním zaměřením vývoje tohoto standard bylo nasazení do systémů existujících v energetickém průmyslu a zajištění kompatibility zařízení různých výrobců. V současné době je využit i v oblastech plynárenském a ropném průmyslu, v oblastech zpracování odpadů a vod a v bezpečnostních odvětvích. Převážné využití má v Severní a Jižní Americe, Jižní Africe, Asii a Austrálii. V Evropě je jeho největším konkurentem standard IEC 60870-5-101, který je nicméně zaměřen primárně na energetický průmysl.

Oba standardy, DNP3 a IEC 60870-5-101, vychází ze stejného základu, standardu IEC 60870. [3]

Dalšími značnými výhodami jsou:

- dovoluje komunikaci jak typu nadřízený-podřízený (master-slave), tak komunikaci typu bod-bod (peer-to-peer),
- dovoluje určit více nadřízených (master) stanic,
- možnost vyžádat si odpověď pouze se novými daty (pokud má podřízená stanice po opakovaném dotazu stejná data, nepošle žádná data v odpovědi),
- použít Ethernet jako komunikační médium.

4.1 Architektura EPA

Stejně jako standard IEC 60870-5-101, i DNP3 využívá zjednodušený komunikační model EPA. Na rozdíl od IEC 60870-5-101 však implementuje navíc pseudo-transportní vrstvu. *Obr. 4-1* znázorňuje dělení zprávy na jednotlivých vrstvách EPA modelu.

4.1.1 Aplikační vrstva

Specifikuje formát zprávy a dostupné služby. Potvrzuje úspěšně přijaté zprávy a sestavuje zprávy pro odeslání. Pokud je zpráva příliš dlouhá, pošle se sekvenčně více menších zpráv po sobě. Tyto zprávy zároveň indikují, zda jsou poslední zprávou, nebo jestli následují další části. [3]

4.1.2 Pseudo-transportní vrstva

Segmentuje případné velké zprávy do menších celků. Ke každé zprávě vkládá označení, zda se jedná o první nebo poslední rámec přenášené zprávy. Přidává také pořadové číslo zprávy pro určení nedoručených zpráv nebo pro poskládání zprávy, dorazí-li zprávy ve špatném pořadí.

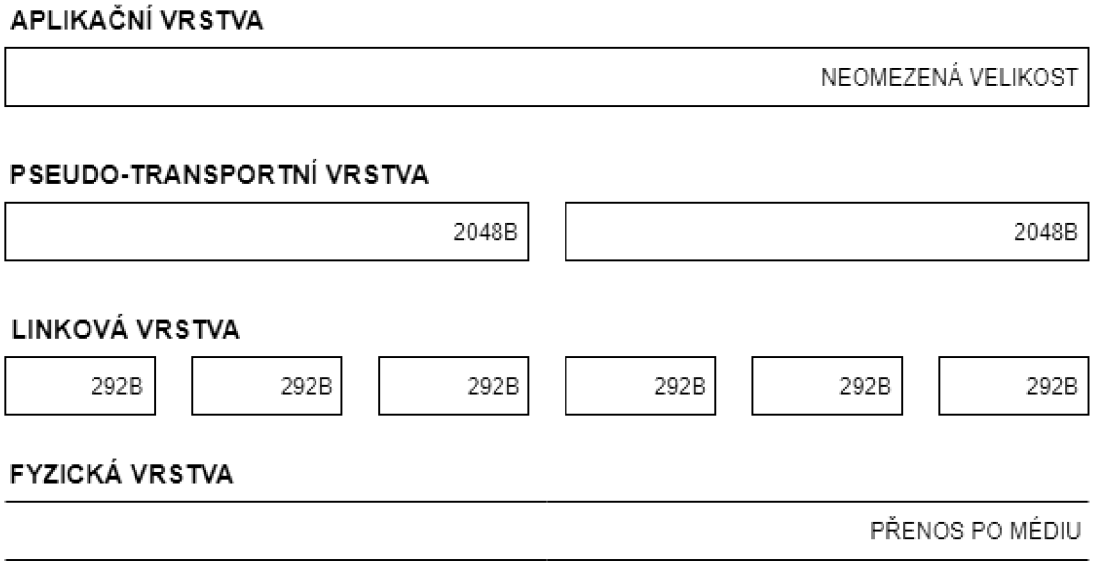
4.1.3 Linková vrstva

Specifikuje formát rámců, DNP3 využívá rámec typu FT3. Do rámce ukládá informaci o zdrojové a cílové adrese. Detekuje případné chyby v přijatých rámcích.

4.1.4 Fyzická vrstva

Spravuje fyzické médium. Poskytuje informaci o aktuálním stavu média, jestli je možno poslat zprávu, či nikoliv. Stará se také o synchronizaci přenosu.

Fyzickou vrstvu mohou představovat sériové linky RS-232 a RS-485, optické a rádiové vlny, nebo Ethernet. [2]



Obrázek 4-1 Sestavení zprávy a velikost datových jednotek.

5. IEC 61850

Poskytuje jednotnou a standardizovanou metodu pro tvorbu komunikačního systému, která je nezávislá na dodavateli inteligentních elektronických zařízení, tzv. IED (Intelligent Electronic Device). Základním cílem byla kompatibilita těchto zařízení od odlišných výrobců. K přenosu zpráv využívá všech vrstev referenčního modelu ISO/OSI.

IED zajišťují ochranu rozvodny, realizují dohled, její automatizaci a v neposlední řadě také měření a regulaci připojených prvků.

„IEC 61580 je jediný standard, který vyhovuje všem požadavkům energetických a rozvodných společností na celém světě na kompatibilitu instalovaného souboru regulačních a řídicích zařízení od různých výrobců.“ [5]

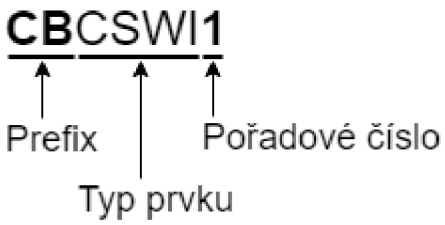
Značnou výhodou oproti předchozím protokolům je fakt, že vychází z technologie Ethernet, tudíž lze využít i mnoho již zavedených nástrojů a zařízení. Nově také využívá logického a snadno čitelného způsobu popisu zařízení vycházející z objektově orientovaného návrhu. Tento postup lze použít k jednoznačnému rozlišení zařízení v systému, jejich funkcí a následně dat, která obsahují. [5]

5.1 Datový model

Adresace zavedených standardů, jako IEC 60870-5-101 a DNP3, je založena na registrech a indexech. Adresa zařízení sama o sobě nemá žádnou výpovědní hodnotu o typu nebo zaměření zařízení. IEC 61850 definuje syntaxi objektů, která má určitou výpovědní hodnotu.

Jako příklad lze uvést pojmenování logického prvku s názvem „CBCSWI1“. „CB“ je předpona, „CSWI“ je typ prvku (circuit switch, přepínač), a „1“ je pořadové číslo prvku. Popis logického prvku je také zobrazen na *Obr. 5-1*.

Jazyk pro popis a konfiguraci objektů vychází ze značkovacího jazyka XML (Extensible Markup Language).



Obrázek 5-1 Příklad pojmenování prvku. [7]

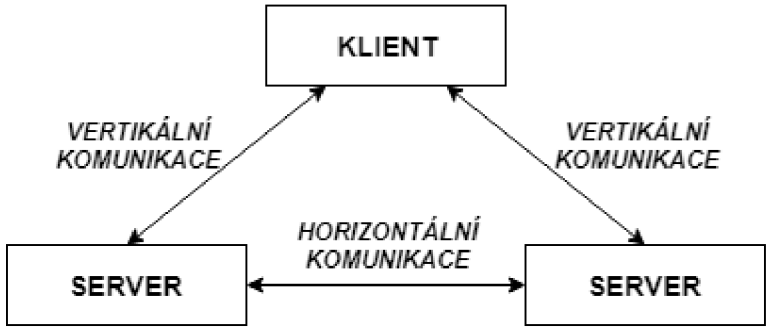
Kompletní struktura systému, jejíž princip adresace je uveden na *Obr. 5-2* níže, je rozdělena do několika částí: [7]

- Fyzické zařízení. Fyzickým zařízením je IED. Fyzické zařízení je označeno IP adresou a unikátním názvem zařízení o maximální délce 10 znaků.
- Logické zařízení. V rámci jednoho fyzického zařízení může existovat několik logických zařízení. Umožňuje fyzickému zařízení chovat se jako výchozí brána a tím zastat funkcionalitu datového koncentrátoru. Seskupuje logické uzly.
- Logický uzel. Reprezentuje samotné prvky. Název logických uzlů bývá odvozen od názvů prvků. Seskupuje datové objekty.
- Datový objekt. Každý datový objekt má unikátní označení, jenž je specifikováno standardem. Jeho strukturu určuje CDC (Common Data Class). Obsahuje set atributů, které jsou dále pojmenována, mají přiřazen datový typ.
- Datový atribut. Uchovává určitou hodnotu. Má specifikován název a datový typ.
- Funkční omezení. Seskupuje datové atributy do skupin podle jejich charakteru.

Obrázek 5-2 Ukázka kompletní adresy objektu. [7]

5.2 Komunikace

Ve standardu IEC 61850 se topologie systému dělí na vertikální a horizontální komunikaci. Rozdíl mezi nimi je uveden na *Obr. 5-3*. Zatímco vertikální komunikace slouží k přenosu zpráv mezi řídícím střediskem a podřízenými jednotkami, horizontální komunikace se využívá pro přenos zpráv mezi IED jednotkami na stejné úrovni. Oba typy jsou detailněji popsány dále.



Obrázek 5-3 Znázornění vertikální a horizontální komunikace.

5.2.1 Komunikace typu klient-server

Tento typ komunikace využívá řídící středisko SCADA systému pro komunikaci s IED jednotkami. Pro přenos zpráv je použit komunikační protokol MMS (Manufacturing Message Specification). Funguje zde mechanismus žádost-odpověď, kdy řídící středisko požádá IED jednotku o vrácení naměřených hodnot, nebo jí pošle řídící příkaz. IED (server) má zároveň možnost využít metody hlášení na základě výjimky, kdy může řídící jednotce oznámit událost i bez předchozího dotazu. [6]

Jedná se o vertikální typ komunikace, který lze využít pouze pro přenos dat, která nejsou časově kritická. Mezi tato data spadá monitoring celého systému, naměřené hodnoty, řídící příkazy a konfigurační soubory.

5.2.2 Komunikace typu vydavatel-odběratel

Tento typ komunikace je využit pro přenos zpráv na jedné úrovni systému, mezi jednotkami IED. Dovoluje mezi jednotkami IED komunikovat mezi sebou a v případě potřeby na základě kritických zpráv i řídit systém. Zprávy se přenáší pomocí skupinového rozesílání a odebírají je jen zařízení, která se k odběru přihlásila. Tento typ komunikace se provozuje na horizontální úrovni.

Zprávy se dělí na dva typy, GOOSE (Generic Object Oriented Substation Events) a GSSE (Generic Substation State Events).

5.2.3 Zpráva typu GOOSE

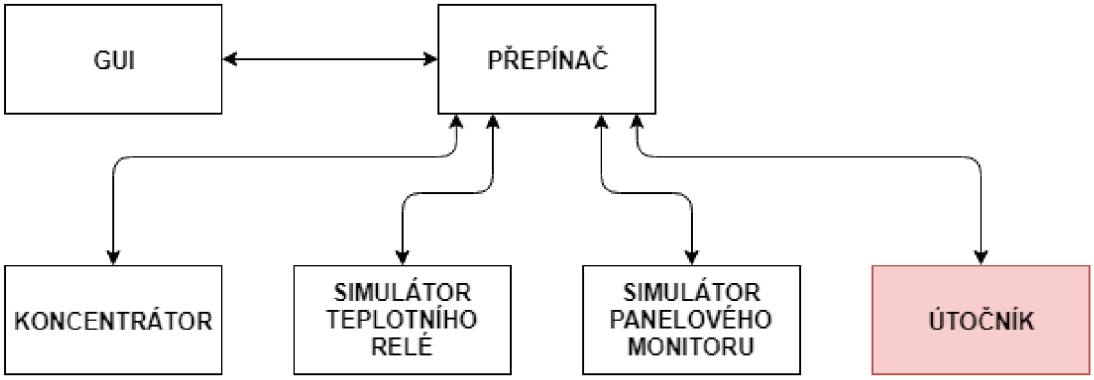
Generická objektově orientovaná událost rozvodny. Přenášená data jsou seskupena do datového objektu. Je určena pro přenos kritických dat, kde jsou kladeny přísné časové požadavky na přenos. Od odeslání do přijetí nesmí uběhnout více jak 4 ms. Má také vyšší prioritu při přenosu, ve frontě v Ethernetové síti s přepínači se přednostně řadí na začátek. Je směrována do specifické multicastové skupiny, zprávu přijímají a zpracovávají pouze zařízení, které mají nastavený odběr zpráv vysílacího zařízení. Typ vydavatel-odběratel (publish-subscribe). [5]

5.2.4 Zpráva typu GSSE

Generická stavová událost rozvodny. Je určena pouze pro přenos stavových dat a není seskupena do datového objektu. Je přenášena pomocí MMS a její přenos trvá déle. Příjemci jsou jasně určeni. Typ klient-server. [5]

6. SIMULOVANÉ PROSTŘEDÍ

Pro virtualizaci simulovaného prostředí bylo využito programu VMware Workstation verze 14, ve kterém bylo spuštěno pět virtuálních strojů. Tři z nich slouží pro vytvoření jednoduché topologie, ve které je realizována komunikace protokolem IEC 61850. Do této topologie je také připojen čtvrtý stroj, který představuje útočníka a ze kterého bylo prováděno testování bezpečnostních slabin zmíněného protokolu. S pomocí posledního virtuálního stroje byly vzdáleně vytvářeny programy na ostatních virtuálních strojích. Topologie je zobrazena na *Obr. 6-1*.



Obrázek 6-1 Topologie simulovaného prostředí.

6.1 Virtualizované stroje

Aby bylo možné efektivně zpracovat a otestovat navržené pracoviště, bylo zvoleno virtuální prostředí. Umožňuje jednoduchou práci s virtuálními stroji, jejich nastavení a přepínání mezi nimi. Veškerá činnost je také omezena pouze na virtuální stroje a nedostane se mimo toto prostředí. Kompletní obrazy virtuálních strojů jsou také součástí přílohy této práce.

6.1.1 Stanice SCADA GUI

Operačním systémem tohoto stroje je Debian 9.4. Obsahuje vývojové prostředí NetBeans IDE 8.2, s jehož pomocí byly vyvíjeny programy pro ostatní virtuální stroje. Všechny zdrojové soubory pohromadě lze nalézt po spuštění vývojového prostředí NetBeans na levé straně mezi projekty, nebo ve složce `/root/NetBeansProjects`. Alternativně jsou také zvlášť součástí přílohy této práce.

Programovací jazyk byl zvolen C++, jelikož knihovna libiec61850, která zprostředkovává dílčí funkce protokolu, je také psána v jazyce C++. Projekty byly

zkompilovány pomocí kompilátoru g++ přímo na cílových stanicích skrze zabezpečené připojení.

6.1.2 Stanice SCADA outstation 1, outstation 2 a koncentrator

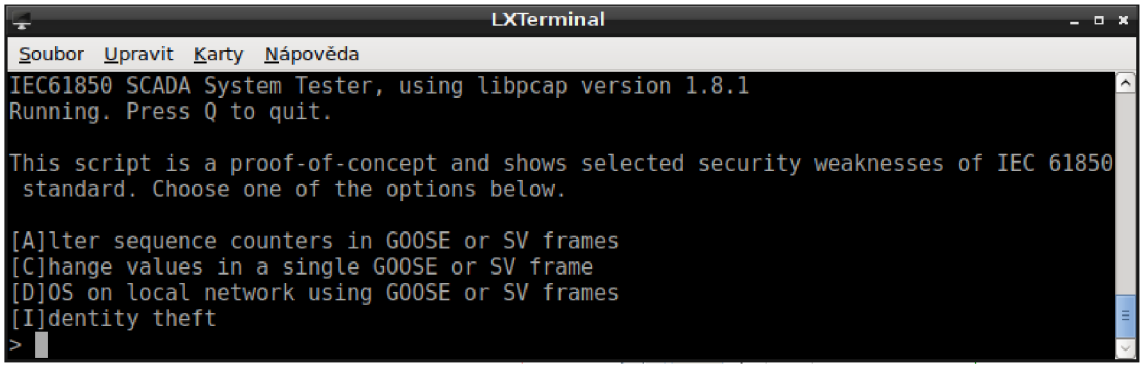
Tyto tři stanice tvoří SCADA systém a generují komunikaci mezi nimi. Stanice outstation 1 a outstation 2 představují IED zařízení, které v rámci simulování reálných zařízení slouží jako generátory dat, které se pak v systému SCADA zpracovávají. Stanice koncentrator je odběratelem těchto dat a zpracovává je. Tyto programy jsou podrobněji rozebrány dále.

Všechny tři stanice mají nainstalován operační systém Debian 9.6 v minimální verzi bez grafického rozhraní, pouze s příkazovou řádkou. Aplikace je možno na každé stanici nalézt ve složce `/root/.netbeans/remote/192.168.116.1xx/debian-Linux-x86_64/root/NetBeansProjects/`.

6.1.3 Stanice Kali Linux

Tato stanice představuje narušitele systému. Je zde k dispozici konzolová aplikace, která umožňuje demonstrovat zneužití bezpečnostních slabin v daném SCADA systému. Hlavní okno konzolové aplikace je zobrazeno na *Obr. 6-2*. Je doporučeno použít zároveň s touto aplikací i další aplikaci pro zachytávání paketů pro názornější ukázku principu útoků, například program Wireshark.

Aplikace se nachází ve složce `/root/.netbeans/remote/192.168.116.200/debian-Linux-x86_64/root/NetBeansProjects/`.



Obrázek 6-2 Hlavní okno aplikace generátoru útoků.

6.2 Vytvořené aplikace

V rámci této práce bylo zapotřebí vytvořit několik aplikací, z nichž každá vykonává dílčí činnost ve výsledném SCADA systému. Jedná se o následující aplikace: simulátor teplotního relé a generování hodnot, simulátor panelového monitoru k měření hladin

nízkého napětí a generování hodnot, koncentrátor, jenž sbírá tyto generované údaje, a souhrnný balíček funkcí, který umožňuje realizovat zmíněné útoky na bezpečnostní slabiny, který bude podrobně rozebrán dále. Aplikace využívají knihovnu libiec61850, která je dostupná z [9]. Součástí jsou také bloková schémata pro lepší představu fungování důležitých částí aplikací.

6.2.1 Konfigurační soubor

Aplikace simulátorů teplotního relé a panelového monitoru používají knihovnu libiec61850 pro implementování funkcí potřebných pro komunikaci protokolem IEC 61850. Proto také vychází z podobného základu a mají možnost nastavení hodnot potřebných pro správnou funkci. Jelikož je těchto hodnot mnoho, bylo lepším řešením vytvořit konfigurační soubor, který bude aplikace po startu načítat a hodnoty z něj číst, než aby uživatel psal nespočet parametrů při spouštění aplikace z příkazové řádky.

Položky v konfiguračním souboru jsou rozděleny do tří kategorií:

- libiec61850 values, což jsou hodnoty, které slouží k nastavení komunikace aplikace protokolem IEC 61850,
- simulator values, hodnoty, které nastavují parametry simulátoru a generování hodnot,
- initial time values, výchozí hodnoty pro časový údaj, jenž je součástí odesílaných rámců v rámci GOOSE a Sampled Values

Konstrukce konfiguračního souboru je typu `promenna=hodnota`, přičemž každá proměnná je na novém řádku. Pomocí křížku (#) je možno řádek zakomentovat a nebude při načítání konfiguračního souboru zpracován. Ty položky konfiguračního souboru, které mají simulátory teplotního relé a panelového monitoru společné, jsou uvedeny v následující *Tab. 6-1*.

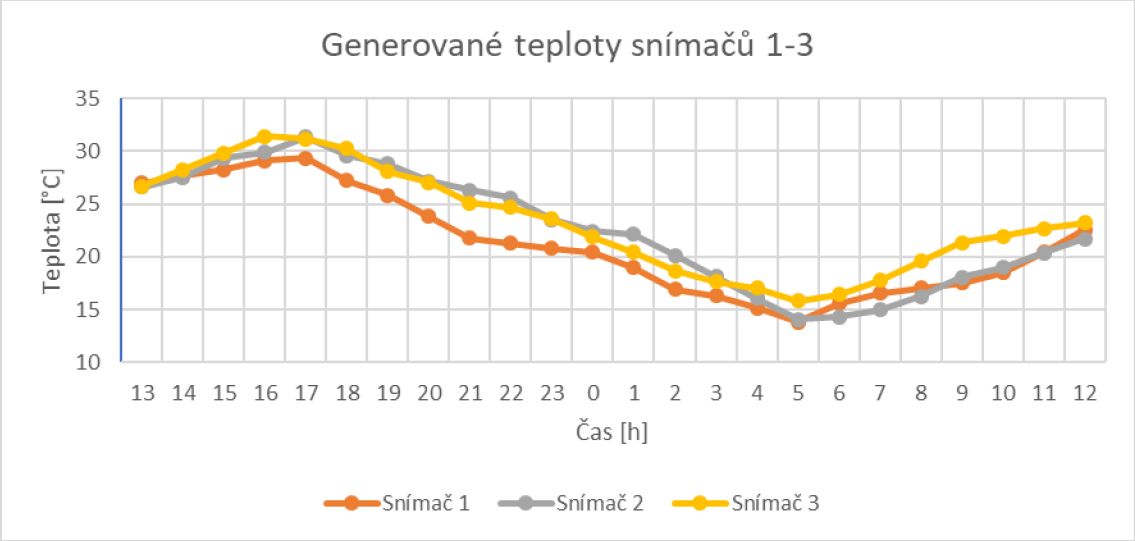
Tabulka 6-1 Výčet společných položek konfiguračního menu.

Název	Hodnota	Vysvětlivka
libiec61850 values		
interface	ens33	Rozhraní, které je využito pro odesílání dat.
vlan_priority	4	Hodnota Priority Code Point, součást IEEE 802.1Q v záhlaví rámce, který udává prioritu zpracování rámce.
vlan_id	0	Označení virtuální LAN sítě, do které rámec patří.
sv_app_id	0x4000	Označení pole APPID v rámci typu Sampled Values.
sv_mac_5	0x00	Hodnota paté dvojice hexadecimálních čísel MAC adresy zařízení.

sv_mac_6	0x00	Hodnota šesté dvojice hexadecimálních čísel MAC adresy zařízení.
sv_ratio	20	Poměr odesílaných GOOSE rámců ku Sampled Values rámcům.
goose_app_id	0x1000	Označení pole APPID v rámci typu GOOSE.
goose_mac_5	0x00	Hodnota paté dvojice hexadecimálních čísel MAC adresy zařízení.
goose_mac_6	0x00	Hodnota šesté dvojice hexadecimálních čísel MAC adresy zařízení.
goose_time_min	25	Interval, ve kterém začne zařízení rozesílat náhlé GOOSE rámce v rámci překročení hodnoty.
goose_time_max	500	Interval periodického odesílání GOOSE rámců.
simulator values		
monitoring_time	1	Konstanta, která je spolu s hodnotou time_factor použita pro výpočet posunu času při každé iteraci.
time_factor	3600	
sensors	3	Počet senzorů, pro které se generují data.
initial time values		
year	2019	Výchozí časová hodnota – rok.
month	4	Výchozí časová hodnota – měsíc.
day	1	Výchozí časová hodnota – den.
hours	13	Výchozí časová hodnota – hodina.
minutes	37	Výchozí časová hodnota – minuta.
seconds	0	Výchozí časová hodnota – sekunda.

6.2.2 Simulátor teplotního relé

Simulátor vychází z reálného zařízení Ziehl TR1200IP, což je teplotní relé, ke kterému je možno připojit až 12 individuálních teplotních čidel typu PT100. Zařízení je schopno měřit aktuální hodnotu na každém čidlu a přímo komunikovat protokolem IEC 61850. V případě náhlého skoku hodnoty, které je vyšší než přednastavená hodnota, zařízení vyšle GOOSE rámec informující o této skutečnosti. Dokumentace zařízení je dostupná z [10]. Simulované zařízení se nachází na stanici SCADA outstation 1. Výstup simulátoru, který byl převeden do grafu, je na *Obr. 6-3*. Graf zahrnuje generování teplot v průběhu jednoho dne pro tři nezávislé teplotní snímače.



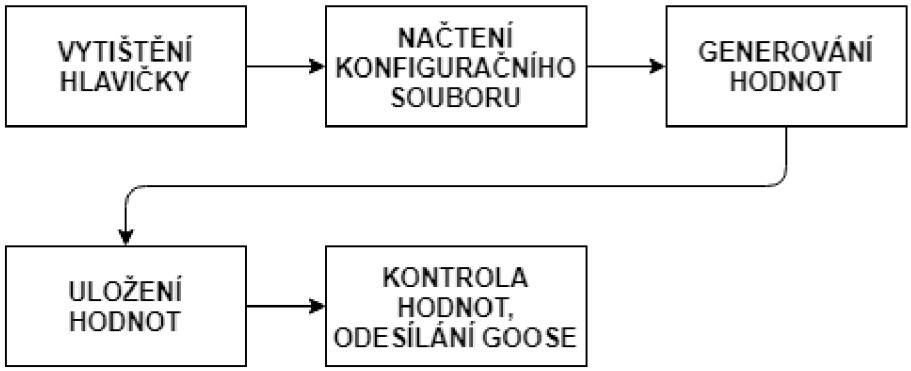
Obrázek 6-3 Výstup simulátoru teplotního relé.

Teplotní relé nabízí podrobné možnosti nastavení zařízení. Ne všechny položky nastavení jsou však potřebné k funkci simulátoru. Potřebné položky nastavení pro simulátor jsou uvedeny v konfiguračním souboru s předdefinovanými hodnotami, který se načítá při spuštění aplikace. Tyto položky jsou uvedeny v *Tab. 6-2*.

Tabulka 6-2 Výčet individuálních položek konfiguračního souboru pro teplotní relé.

Název	**Hodnota**	**Vysvětlivka**
simulator values		
deadband	2.17	Prahová hodnota, po jejímž překročení vygenerovanou hodnotou bude odeslán okamžitý GOOSE rámec.
temp_min	15	Dolní hranice generátoru teplot.
temp_max	30	Horní hranice generátoru teplot.

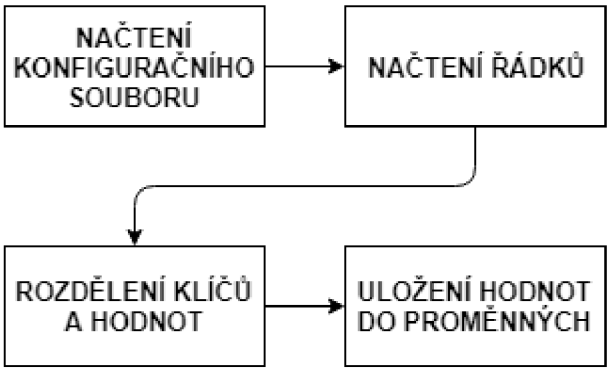
Kód aplikace je tvořen z hlavního souboru main.cpp a souborů vytvořené třídy `ziehl_tr1200ip_simulator.h` a `.cpp`. Soubor `main.cpp` obsahuje funkci pro vytištění hlavičky aplikace, vytvoření objektu třídy, načtení konfiguračního souboru a jednoduchou smyčku. Celek je popsán blokovým schématem na *Obr. 6-4*.



Obrázek 6-4 Blokové schéma simulátoru teplotního relé.

Třída `ziehl_tr1200ip_simulator.cpp` obsahuje několik funkcí:

- konstruktor, který obsahuje pole hodnot teplot přiřazených k denní hodině a inicializace proměnných,
- destruktor, který uvolní alokované místo v paměti pro proměnné,
- funkci pro načtení a zpracování konfiguračního souboru, jehož princip je popsán blokovým schématem na *Obr. 6-5*, ověření načtených dat, zdali jsou v povoleném rozmezí hodnot, připravení ASDU jednotek pro naplnění daty a odeslání,
- funkci pro inicializaci knihovny libiec61850, přiřazení odpovídajících hodnot, vytvoření potřebných objektů a nastavení patřičných údajů,
- funkci pro uložení vygenerovaných hodnot do textového souboru, který se nachází v kořenovém adresáři aplikace, není ovšem dále využit a je určen spíše pro zpětné dohledání hodnot v případě potřeby,
- funkci pro vygenerování hodnot na základě aktuální denní doby (v případě simulátoru zadané v konfiguračním souboru) a generování dalších hodnot vycházejících z hodnot předešlých, kontrola rozmezí hodnot, naplnění objektů knihovny libiec61850 a odeslání rámců, generování hodnot popisuje blokové schéma na *Obr. 6-6*,
- funkci pro přeposílání náhlých GOOSE zpráv s prioritou doručení.



Obrázek 6-5 Blokové schéma zpracování konfiguračního souboru.

Obrázek 6-6 Blokové schéma generování hodnot.

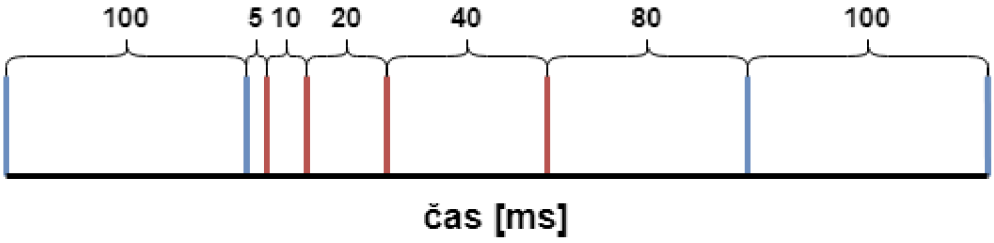
Funkce pro generování hodnot má několik základních bodů, ze kterých vychází. Aby generované hodnoty byly relevantní a jistým způsobem odpovídaly realitě, bylo potřeba vytvořit pole teplot v odpovídající denní hodinu, vůči kterým si pak simulátor načetl výchozí hodnotu podle zadaného časového údaje v konfiguračním souboru. Tyto hodnoty v poli vychází z archivních údajů meteorologické stanice. [11]

V každém dalším průběhu generování teplot generátor vychází z předešlé hodnoty. Rozpětí přírůstku za danou dobu je určeno proměnnými `rand_low` a `rand_high`, jejichž hodnota odpovídá nejnižšímu a nejvyššímu přírůstku teploty za 1 sekundu. Tyto výpočty vychází taktéž ze zmíněných archivních údajů.

Aby limitní hodnoty generátoru odpovídaly nastavenému časovému posunu během generování, jsou násobeny proměnnými `time_factor` a `monitoring_time`. Během denního cyklu se také určuje, jestli se bude vygenerovaný přírůstek teploty přičítat, nebo odčítat, což závisí na faktu, jestli uložené časové údaje generátoru odpovídají noci nebo dni.

Konečná vygenerovaná hodnota se porovnává se zadanou hodnotou `deadband` v konfiguračním souboru. Pokud je přírůstek větší než `deadband`, okamžitě se vyšle

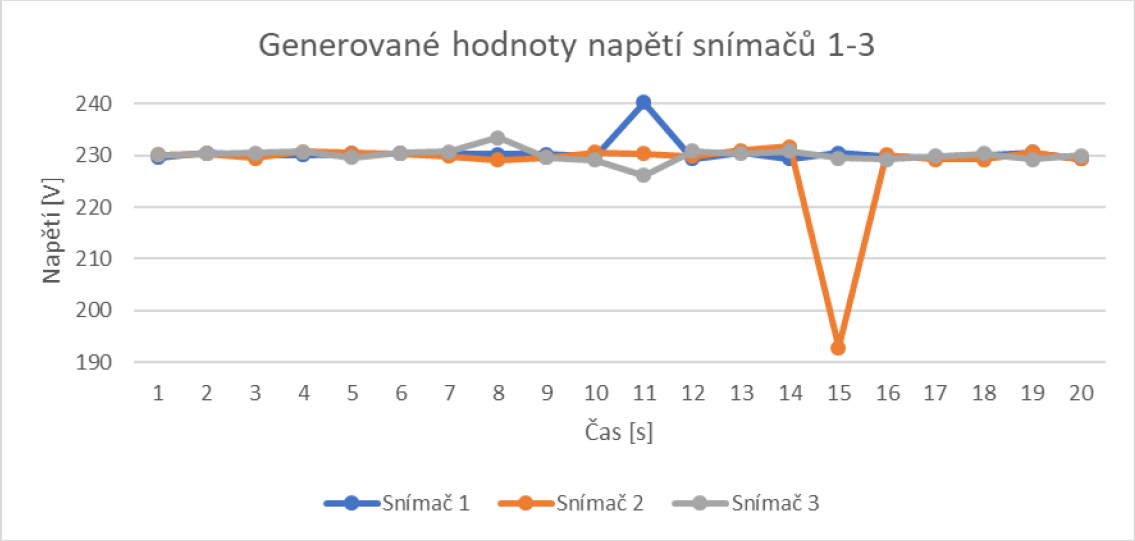
GOOSE rámec. Vysílání toho GOOSE rámce podléhá požadavkům co nejrychlejšího možného doručení pod 4 ms a je k tomu určen speciální postup vysílání. GOOSE rámec se vyšle s intervalem rovnajícím se hodnotě `goose_time_min` a s každým dalším vysláním rámce se tato hodnota zdvojnásobí, dokud nedosáhne opět pravidelného intervalu určeným proměnnou `goose_time_max`. Znázornění toho principu vysílání je zobrazen na *Obr 6-7*. Modře je znázorněno pravidelné vysílání rámců, červeně pak prioritní.



Obrázek 6-7 Vysílání prioritního GOOSE rámce.

6.2.3 Simulátor panelového monitoru

Předlohou tohoto simulátoru bylo existující zařízení od firmy Měřící Energetické Aparáty, a.s. typu MEg44PAN. Zařízení je určeno k měření hladin nízkého napětí a dokáže měřit tři napětí či tři proudy, dále funguje jako elektroměr, provádí funkci analýzy kvality napětí a lze zobrazit i oscilografické záznamy. Nemá sice podporu protokolu IEC 61850, ale podporuje protokol IEC 60870-5-104 a MODBUS. Posílané hodnoty je možno vyčíst a aplikovat do simulátoru protokolu IEC 61850. Simulované zařízení se nachází na stanici SCADA outstation 2. Výstup simulátoru panelového monitoru, převedeného do grafu, znázorňuje *Obr. 6-8*. Graf zahrnuje generování hodnot pro tři nezávislé hladiny napětí během dvaceti sekund. V 11. sekundě byla uživatelem zadána špička pro snímač č. 1, nápodobně pro snímač č. 2 v 15. sekundě.



Obrázek 6-8 Výstup simulátoru panelového monitoru.

Simulátor tohoto zařízení umožňuje generovat až tři nezávislé hodnoty napětí, přičemž si uživatel může zvolit základní hladinu napětí, vnitřní a vnější rozpětí kolísání. Také lze zvolit poměr kolísání mezi vnitřním a vnějším rozpětím. Výchozí hodnotou pro základní hladinu napětí je 230, pro vnitřní rozpětí je to hodnota 231 a pro vnější rozpětí je to hodnota 235. Poměr kolísání mezi vnitřním a vnějším rozpětím je 5 %. Jelikož předešlé vygenerované hodnoty nemají s novou hodnotou žádnou souvislost, je možné v každé iteraci generovat přírůstek a ten přičíst nebo odečíst od výchozí hodnoty napětí.

Uživatel má možnost ručně zadat stisknutím klávesy výkyv v hodnotách napětí, který se počítá jako náhodný přírůstek k vygenerované hodnotě v rozmezí od 0 do 0,2 × vygenerovaná hodnota. Vznikne nová událost a prioritní GOOSE rámec se vysílá stejným způsobem, jako tomu je u teplotního relé.

Simulátor panelového monitoru opět načítá hodnoty z konfiguračního souboru. Základ konfiguračního souboru je totožný s konfiguračním souborem teplotního relé, odlišné položky jsou vypsány v *Tab. 6-3*.

Tabulka 6-3 Výčet individuálních položek konfiguračního souboru pro panelový monitor.

Název	**Hodnota**	**Vysvětlivka**
simulator values		
voltage_center	230	Výchozí hodnota napětí.
voltage_inner_bound	231	Hodnota vnitřního rozpětí.
voltage_outer_bound	235	Hodnota vnějšího rozpětí.
inner_to_outer_ratio	5	Poměr kolísání mezi vnitřním a vnějším rozmezím.

Kód aplikace tvoří soubory `main.cpp` a třída `mega_meg44pan_simulator.h` a `.cpp`. Podobně jako u teplotního relé tvoří soubor `main.cpp` funkce pro výpis hlavičky aplikace a smyčka, ve se které po uplynutí nastavené doby vygenerují nové hodnoty napětí. Rozdílem je zde dodatečná funkce, jež s každou iterací hlavní smyčky zkoumá, jestli uživatel stiskl tlačítko. Pokud ano, nastaví se proměnná `spike` na `true`, vygeneruje se špička v hodnotě napětí a vyšle se GOOSE rámec, který se stejně jako u teplotního relé posílá prioritním způsobem. Kontrola stisknuté klávesy je znázorněna blokovým schématem na *Obr. 6-9*.

Obrázek 6-9 Blokové schéma kontroly stisknuté klávesy.

Soubor `mega_meg44pan_simulator.cpp` téže třídy obsahuje následující funkce:

- konstruktor, který inicializuje proměnné třídy,
- destruktor, který uvolní alokovanou paměť,
- funkci pro načtení konfiguračního souboru, funkcionálně totožná s funkcí načtení konfiguračního souboru u teplotního relé, avšak některé proměnné jsou určeny přímo pro generátor hodnot napětí, dále ověření hodnot, jestli se nachází v povolených rozmezích a vytvoření objektů knihovny libiec61850,
- funkci pro inicializaci objektů knihovny libiec61850,
- funkci pro uložení vygenerovaných hodnot, i když opět není v práci dále využita a slouží v případě nutnosti dohledání určitých hodnot
- funkci pro generování hodnot,
- funkci pro přeposílání náhlých GOOSE zpráv s prioritou doručení.

Funkce pro generování hodnot je z této třídy nejsložitější. Jelikož je možné hodnoty generátoru zadávat vlastní, je potřeba v aplikaci s tímto počítat a vypočítat správné hodnoty.

Porovnáním rozdílu hodnot `voltage_center` a `voltage_inner_bound`, `voltage_center` a `voltage_outer_bound` a jejich následné převedení do absolutní hodnoty se určí vnitřní a vnější rozmezí. V dalším kroku se vypočítá proměnná `multiplier` podělením vnějšího a vnitřního rozmezí. Tato hodnota udává, jakou

hodnotou je třeba vynásobit vygenerované napětí, aby napětí dosáhlo do vnějšího rozmezí, pokud se naplní šance pro tento jev. Zmíněná šance se počítá jako náhodně vygenerované číslo v rozmezí od 0 do hodnoty 100/`inner_to_outer_ratio`, což je další hodnota zadaná v konfiguračním souboru. Pokud se pak šance rovná 1, vygenerovaný přírůstek napětí se vynásobí dříve získanou hodnotou `multiplier` a tím se dostane do vnějšího rozmezí hodnot.

Pokud byl uživatelem zadán příkaz k vytvoření špičky v napětí, vygeneruje se ještě dodatečný přírůstek v rozmezí od nuly do 1/5 hodnoty napětí a ten se následně přičte či odečte od hodnoty napětí, tím se vytvoří špička, která zasahuje i mimo vnější rozmezí. Zvolením konstanty 1/5 jsme zároveň chráněni od toho, aby byla špička příliš markantní a nereálná.

Princip rozesílání GOOSE rámců je totožný s principem uvedeným u simulátoru teplotního relé.

6.2.4 SCADA Koncentrátor

Tato stanice zachytává a zpracovává GOOSE a Sampled Values rámce, které jsou vysílány simulátorem teplotního relé a simulátorem panelového monitoru. Při spuštění aplikace pomocí konzolové řádky je možné přidat parametry, které jsou popsány v následující *Tab. 6-4*.

Tabulka 6-4 Parametry spouštění koncentrátoru.

Název	**Hodnota**	**Vysvětlivka**
interface	ens33	Výchozí rozhraní, na kterém bude probíhat komunikace.
iterations	3	Indikuje počet ASDU v Sampled Values rámci.
tr_sv_app_id	0x4000	Určuje APPID Sampled Values rámců z aplikace teplotního relé.
tr_gs_app_id	0x1000	Určuje APPID GOOSE rámců z aplikace teplotního relé.
ie_sv_app_id	0x4001	Určuje APPID Sampled Values rámců z aplikace panelového monitoru.
ie_gs_app_id	0x1001	Určuje APPID GOOSE rámců z aplikace panelového monitoru.

Aby aplikace byla schopna přijímat rámce typu GOOSE a Sampled Values, je zapotřebí vytvořit přijímač. Přijímači se nastaví, na jakém rozhraní má naslouchat komunikaci. Následně se vytvoří odběratel, který se přiřadí k vytvořenému přijímači.

Poté se odběrateli přiřadí obslužná funkce, která se zavolá v případě, že dorazí rámec jemu určený. Nakonec se spustí samotný přijímač.

Nevýhodou obslužné funkce, alespoň v rámci knihovny libiec61850, je fakt, že je kompletně v režii programátora, a tudíž je zpracování rámců a ošetření případných bezpečnostních slabin závislé na jeho implementaci této funkce. Popis obslužných funkcí v rámci této práce bude rozebrán dále.

Analogicky se podobný postup provede i pro druhý typ rámců. Níže následuje ukázka posloupnosti příkazů pro vytvoření funkčního odběratele Sampled Values rámců.

```
// vytvoreni odberatele
SVSubscriber sv_sub = SVSubscriber_create(NULL, sv_app_id);
// vytvoreni listenera
SVSubscriber_setListener(sv_sub, svUpdateListener, &sv_app_id);
// vytvoreni prijimace
SVReceiver sv_rec = SVReceiver_create();
// nastaveni rozhrani pro prijimac
SVReceiver_setInterfaceId(sv_rec, interface);
// prirazeni odberatele na prijimac
SVReceiver_addSubscriber(sv_rec, sv_sub);
// spusteni prijimace
SVReceiver_start(sv_rec);
...
// ukonceni prijimace a uvolneni pameti
SVReceiver_stop(sv_rec);
SVReceiver_destroy(sv_rec);
```

6.2.4.1 Obslužná funkce pro rámec typu GOOSE

Obslužná funkce si uchovává stavy čítačů `stNum` a `sqNum` z přijatých a zpracovaných rámců. Na základě toho lze rozpoznat, jestli příchozí rámec není duplikátem, starým rámcem nebo rámcem podvrženým.

Po spuštění se čítače inicializují na nulu. Pokud dorazí rámec, zkontroluje se nejdříve jeho APPID, zdali souhlasí s APPID, které má obslužná funkce zpracovávat. Pokud ano, porovnají se hodnoty uložených čítačů s čítači v rámci. V případě, že uložené čítače jsou všechny nulové, značí to počáteční běh funkce a je potřeba je nejprve aktualizovat, což se provede uložením čítačů `stNum` a `sqNum` z přijatého rámce do uložených čítačů. Tímto se načetly první hodnoty, na jejichž základě lze porovnávat každý další rámec typu GOOSE.

Přijatý rámec prochází následující sekvencí podmínek:

- Je `sqNum` rámce stejné jako uložené `sqNum` a zároveň není nulové? Tato situace značí duplikovaný rámec a je proto zahozen.
- Je `stNum` rámce stejné jako uložené `stNum` a zároveň je `sqNum` o jedno větší, než je uložená hodnota? Toto značí legitimní rámec a je zpracován.
- Je `sqNum` nulové a `stNum` větší, než je uložená hodnota? Tato situace může značit změnu hodnot přenášených pomocí GOOSE a tudíž inkrementované `stNum` a vynulované `sqNum`. Pokračuje se do vnořených podmínek.
    - Je `stNum` rámce stejné jako uložená hodnota? Toto značí duplikovaný rámec.
    - Je `stNum` rámce o jedna větší, než je uložená hodnota? Toto značí legitimní rámec s inkrementovaným `stNum`. Je dále zpracován.

V případě, že je rámec zpracován, mimo výpisu jeho hodnot do konzole jsou také aktualizovány uložené hodnoty obslužné funkce, aby bylo možné následující rámec porovnat s aktuálními hodnotami.

Může nastat situace, že čítače `stNum` a `sqNum` odpovídají legitimnímu rámci a je dále zpracován. Ovšem ještě se porovnává uběhnutý čas od přijetí posledního rámce. Pokud je časová prodleva od posledního rámce větší, než je životnost právě přijatého rámce, uvedená v hodnotě `Time Allowed to Live`, je rámec zahozen a čeká se na rámec, který spadá do správného časového intervalu. Je možné, že se rámec zpozdil v síti a hodnoty již nemusejí být aktuální a validní.

6.2.4.2 Obslužná funkce pro rámec typu Sampled Values

Obslužná funkce si stejně jako funkce pro GOOSE rámce uchovává hodnotu posledního přijatého rámce a porovnává proti ní hodnotu čítačů v přijatých rámcích. Nevýhodou pro implementaci sofistikovanějšího zabezpečovacího mechanismu pro tuto obslužnou funkci je fakt, že data zasílaná pomocí Sampled Values jsou uchovávána ve speciálních ASDU jednotkách, kterých může být v jednom Sampled Values rámci více. Pro každou ASDU jednotku, která je v rámci přítomna, se volá obslužná funkce zvlášť do doby, než se zavolá pro poslední ASDU jednotku. V tomto případě je nutné uchovávat nejen hodnotu čítače, ale také index zpracovávané ASDU jednotky, aby jich obslužná funkce nezpracovala více než má, nebo naopak méně. Tato hodnota, kolik ASDU jednotek se má v rámci jednoho Sampled Values rámce zpracovat, je do aplikace předána pomocí parametru při spouštění, viz *Tab. 6-4*.

Přijatý rámec, podobně jako u obslužné funkce pro GOOSE rámec, ověří se APPID a dále prochází řadou podmínek, které vyhodnotí, jestli je rámec legitimní, či nikoliv:

- Je čítač rámce shodný s uloženým čítačem a zároveň je různý od nuly? Platí zároveň podmínka, že počet zpracovaných ASDU jednotek přesáhl

maximální dovolený počet těchto jednotek v rámci jednoho rámce? Pokud ano, je tento rámec duplicitní.

- Je hodnota čítače rámce shodná s uloženým čítačem a zároveň nebyl naplněn maximální počet zpracovaných ASDU jednotek jednoho rámce? Rámec se tedy zdá být validní a pokračuje ve zpracování, inkrementuje se čítač zpracovaných ASDU jednotek.
- Je hodnota čítače rámce o jedna větší, než je hodnota uloženého čítače? Je to tedy nový rámec s novou sadou ASDU jednotek. Resetuje se čítač zpracovaných ASDU jednotek.
- Pokud ani jedna z podmínek výše neplatí, jedná se o první běh programu a je potřeba uložit aktuální hodnoty z rámce.

Pokud rámec projde těmito podmínkami, je dále zpracován a vypsán do konzole.

7. VEKTORY ÚTOKŮ

Kyberkriminalita je neustále se rozvíjející odvětví informačních technologií. Jedná se o kybernetické útoky úzce zaměřené na danou část informačního systému. Cíl útoků se liší, může se jednat o krádež dat, infiltraci do systému, nahrání dat, znemožnění přístupu či poškození softwaru nebo hardwaru. Tyto útoky se velmi liší v měřítku provedení a následcích takového útoku. Mohou je provádět jednotlivci, ale také celé skupiny osob. Cílem mohou být jak individuální zařízení, tak celé systémy.

Obrana proti takovým útokům není jednoduchá, už jen z toho důvodu, že útočník může využít slabin, o kterých správce systému nemusí ani vědět. Obecně platí, že žádný systém nelze považovat za stoprocentně bezpečný. I když v současnosti není známa žádná zranitelnost, neznamená to, že nebude v budoucnu objevena a zneužita.

Základním pravidlem pro účinnou obranu proti kybernetickým útokům je správně a důkladně nastavený systém. Nedílnou součástí je také zavedený mechanismus logování, tedy ukládání údajů o stavu kritických částí systému. Existuje také možnost najmout si tzv. white-hat hackery, specialisty se zaměřením na kybernetickou bezpečnost s účelem penetračního testování systému. Zjištěné poznatky pak nezneužije k nelegálním úkonům, ale předá je zadavateli, který na jejich základě může dále zabezpečit systém.

V rámci systémů SCADA jsou útoky zaměřeny převážně na datový tok a modifikaci posílaných paketů a rámců. Záleží také na tom, jak daleko se v rámci systému útočník dostane, jelikož nemusí mít přístup k veškeré komunikaci.

Dále následuje teoretický popis a rozbor bezpečnostních slabin a možnosti jejich zneužití. Ke každému typu bezpečnostní slabiny bude věnována samostatná podkapitola.

7.1 Struktura rámců GOOSE a Sampled Values

Aby bylo možné se zachycenými rámci jakkoliv manipulovat a měnit jejich data, bylo nejdříve nutné vytvořit tzv. dissector, což je funkce, která na základě znalosti struktury daného rámce a dalších identifikátorů polí dokáže rozčlenit souvislý blok dat rámce do menších logických celků, se kterými pak lze v rámci útoků pracovat.

Struktura GOOSE rámce a Sampled Values rámce je znázorněna na *Obr. 7-1* a *Obr. 7-2*. Rámec se skládá z Ethernetové hlavičky, kde je uvedena zdrojová a cílová MAC adresa zařízení a EtherType, který označuje rámce, které jsou tagované pro VLAN. Dále následuje hlavička IEEE 802.1Q, kde je uveden přímo typ protokolu. GOOSE rámce jsou označeny jako IEC 61850/GOOSE s hexadecimálním vyjádřením 0x88b8, Sampled Values rámce jsou označeny jako IEC 61850/SV (Sampled Value Transmission (0x88ba)). Veškeré ethernetové typy, které organizaci patří a jejich

hexadecimální hodnoty lze nalézt v [12]. Další je hlavička GOOSE části rámce, kde je uvedeno APPID, délka rámce a rezervované pole. Zbytek rámce je kódován podle notace ASN.1 (Abstract Syntax Notation One) a přenášená data tvoří trojici TLV (Type-Length-Value). Type označuje typ hodnoty v následujícím poli, length délku dat v poli, value pak samotná data v poli. [13] Pořadí bajtů dat odpovídá uložení podle Big-Endian, tedy na nejnižší adresu se ukládají nejvíce významné bajty (MSB), za něj ostatní bajty až po nejméně významný bajt (LSB).

Barvy, které označují určité části struktury rámce na *Obr. 7-1*, mají následující význam:

- tmavě modrá barva – hlavička Ethernet II,
- světle modrá barva – hlavička IEEE 802.1Q,
- tmavě oranžová barva – APPID,
- světle oranžová barva – délka rámce,
- černá barva – rezervované pole, aktuálně nevyužito,
- červená barva – ASN.1 identifikátor,
- zelená barva – délka pole,
- šedá barva – hodnota pole.

```
0000   01 0c cd 01 00 00 00 0c 29 38 cc 55 81 00 00 00   ..Í.....)8ÌU....
0010   88 b8 10 00 00 95 00 00 00 00 61 81 8a 80 21 54   .¸.....•.....a...!T
0020   52 31 32 30 30 49 50 4d 45 41 53 2f 4c 4c 4e 30   R1200IPMEAS/LLN0
0030   24 47 4f 24 54 65 6d 70 41 6e 64 52 65 6c 61 79   $GO$TempAndRelay
0040   81 02 02 ee 82 1e 54 52 31 32 30 30 49 50 4d 45   ...î..TR1200IPME
0050   41 53 2f 4c 4c 4e 30 24 54 65 6d 70 41 6e 64 52   AS/LLN0$TempAndR
0060   65 6c 61 79 83 0e 5a 49 45 48 4c 5f 54 52 31 32   elay..ZIEHL_TR12
0070   30 30 49 50 84 08 5d 3f 35 64 41 89 37 0a 85 01   00IP..]?5dA.7...
0080   1a 86 02 01 01 87 01 00 88 01 01 89 01 00 8a 01   ................
0090   03 ab 14 85 01 01 87 05 08 41 82 12 31 91 08 5d   .«.......A..1..]
00a0   03 24 6c 00 00 00 00                              .$l....
```

Obrázek 7-1 Struktura GOOSE rámce.

Tab. 7-1 přiřazuje ASN.1 identifikátory ke konkrétním částem rámce. [13]

Tabulka 7-1 ASN.1 identifikátory GOOSE rámce.

ASN.1 identifikátor	Část GOOSE rámce
0x80	goCBRef
0x81	timeAllowedToLive
0x82	datSet
0x83	goID
0x84	T
0x85	stNum
0x86	sqNum
0x87	Simulation
0x88	confRev
0x89	ndsCom
0x8a	numDatSetEntries
0x8b	allData

Velmi podobnou strukturu má i rámec typu Sampled Values zobrazený na *Obr. 7-2*, kde se liší pouze EtherType a jednotlivé identifikátory polí, které jsou vypsány v *Tab. 7-2*. [14]

```
0000   01 0c cd 04 00 00 00 0c 29 38 cc 55 81 00 00 00   ..Í.....)8ÌU....
0010   88 ba 40 00 00 3d 00 00 00 00 60 33 80 01 01 a2   .º@..=....`3...¢
0020   2e 30 2c 80 0a 54 52 31 32 30 30 49 50 5f 31 82   .0,..TR1200IP_1.
0030   02 01 1c 83 04 00 00 00 01 85 01 00 87 11 01 41   ..............A
0040   61 77 43 00 00 00 00 5d d9 fa dc 00 00 00 00       awC....]ÙúÜ....
```

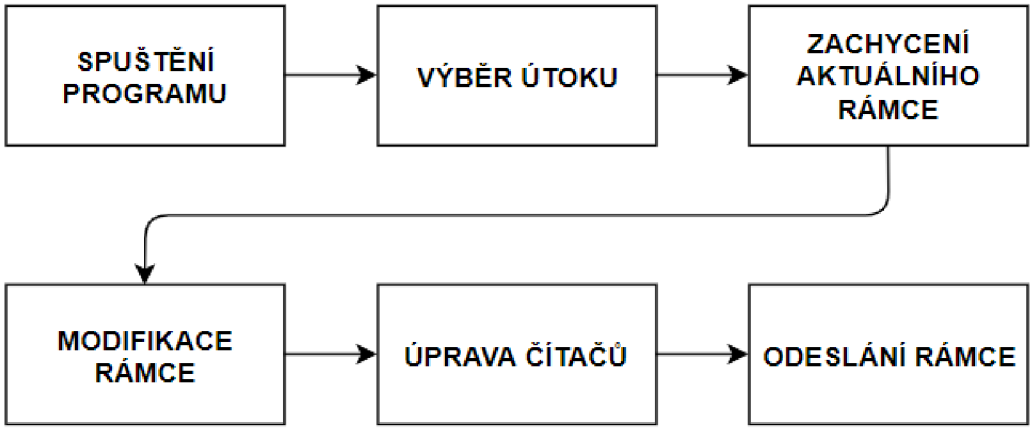
Obrázek 7-2 Struktura SV rámce.

Tabulka 7-2 ASN.1 identifikátory SV rámce.

ASN.1 identifikátor	Část SV rámce
0x60	Začátek SV datové části.
0x80	Počet ASDU jednotek.
0xA2	Začátek ASDU sekvence.
0x30	Začátek konkrétní ASDU.
0x80	svID

0x82	smpCnt
0x83	confRev
0x85	smpSynch
0x87	Datová část ASDU jednotky.

Pro zachycení aktivity na ethernetovém rozhraní a pro zpracování jednotlivých útoků bylo využito knihovny libpcap. S její pomocí bylo možno zachytit GOOSE a Sampled Values rámce a modifikované je zpět poslat do sítě. Knihovna je dostupná z [15]. Obecný popis principu funkčnosti generátoru útoků je zobrazen pomocí blokového schéma na *Obr. 7-3*.



Obrázek 7-3 Princip funkčnosti generátoru útoků.

7.2 Testování chování odběratele při změně hodnot čítačů v GOOSE a Sampled Values rámcích

Komunikace pomocí GOOSE a Sampled Values rámců je založena na periodickém zasílání vydavatelem a zpracování jedním či více odběrateli těchto rámců. Rámce obsahují také hodnoty čítačů, na jejichž základě dokáže odběratel rozpoznat pořadí zachyceného rámce a rozhodnout, jestli rámec bude zpracován, či nikoliv. Rámec typu GOOSE se využívá pro časově kritický přenos dat, např. při náhlé události nebo nečekané změně naměřených dat. Rámec typu SV se využívá pro přenos hromadné kolekce dat od vydavatele k odběrateli.

Rámec typu GOOSE obsahuje čítače dva, `stNum` (status number) a `sqNum` (sequence number). Čítač `stNum` vyjadřuje, jestli se změnily data vyslané vydavatelem. Pokud ano,

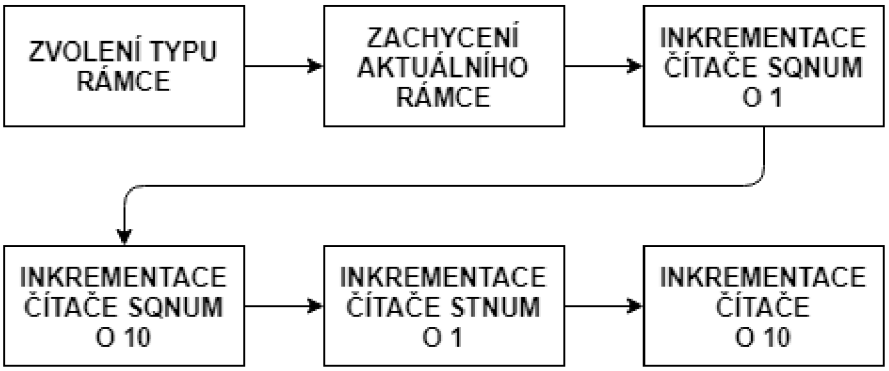
čítač se o inkrementuje o jedna a čítač `sqNum` se vynuluje. Pokud jsou data totožná, inkrementuje se o jedna pouze čítač `sqNum`. Rámec typu GOOSE je vysílán periodicky v určitém intervalu.

Rozesílání rámce typu Sampled Values probíhá obdobně jako u GOOSE rámce, avšak s větším časovým intervalem. Obsahuje jednotlivé datové jednotky ASDU, které mají svůj čítač `smpCnt` (sample count).

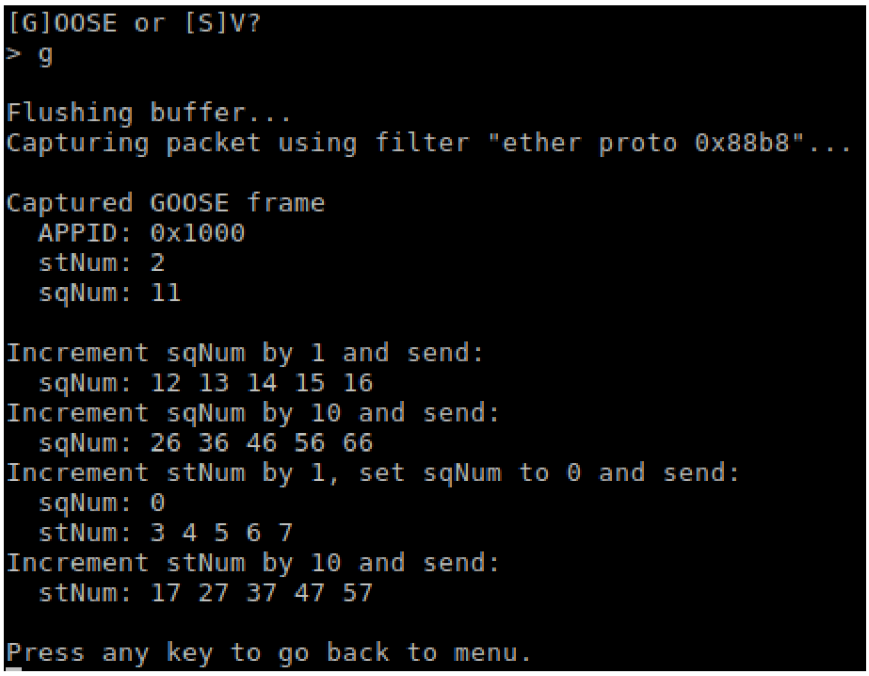
Cílem tohoto útoku je zjistit, jak a bude odběratel zpracovávat rámce, které budou mít upravené hodnoty zmíněných čítačů, zdali je přijme a zpracuje, nebo odmítne zpracovat. Útok je možné spustit klávesou „a“, která zvolí volbu „[A]lter sequence counters in GOOSE or SV frames“ v aplikaci scada_tester. Následuje volba typu rámce, se kterým pak bude aplikace dále pracovat, ten zvolíme pomocí kláves „g“ pro GOOSE rámce nebo „s“ pro Sampled Values rámce. Zbytek útoku je automatizován a není potřeba zásahu od uživatele.

V prvním kroku útoku se zachytí rámec zvoleného typu, do konzole se vytiskne APPID zachyceného rámce a původní hodnoty čítačů `stNum` a `sqNum`. Ve druhém kroku se testuje modifikace hodnoty čítače `sqNum`, která se inkrementuje o 1 a následně pošle zpět do sítě. Třetí krok také testuje modifikaci čítače `sqNum`, ale nyní zvyšuje jeho hodnotu o 10. Čtvrtý krok testuje inkrementaci hodnoty čítače `stNum` o 1. Je nutno také vynulovat hodnotu čítače `sqNum`. Pátý a poslední krok zvyšuje hodnotu čítače `stNum` o 10. Stiskem libovolné klávesy se uživatel dostane zpět do hlavní nabídky aplikace.

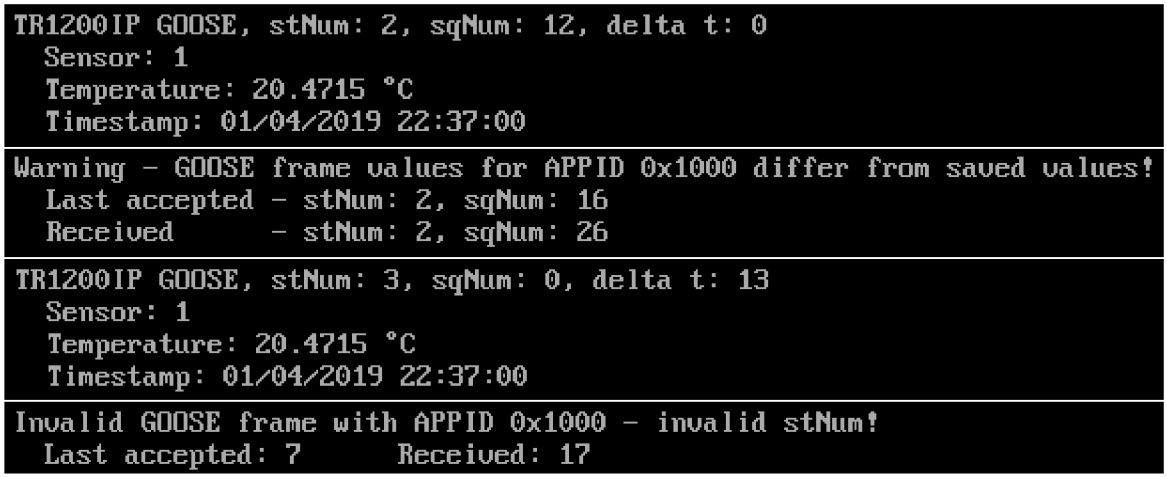
Průběh útoku je znázorněn blokovým schématem na *Obr. 7-4*. Výsledky útoku pro typ rámce GOOSE jsou zobrazeny na obrázcích *Obr. 7-5* a *Obr. 7-6*.



Obrázek 7-4 Blokové schéma útoku změny hodnot čítačů.



Obrázek 7-5 Průběh útoku v aplikaci scada_tester.



Obrázek 7-6 Průběh útoku na straně koncentrátoru.

Pro přehlednost byl výpis průběhu útoku na straně koncentrátoru na *Obr. 7-6* zkrácen a obsahuje vždy pouze první odeslaný rámec z každého kroku útoku.

Závěrem lze konstatovat, že útok byl úspěšný. Rámce, které měly hodnoty čítačů `stNum` a `sqNum` inkrementované vždy pouze o 1 koncentrátor přijal a zpracoval bez problémů. Naopak pokud byly tyto hodnoty zvýšeny více jak o 1, koncentrátor nedokázal tyto rámce správně zařadit a rámce nezpracoval a zahodil.

7.3 Změna hodnot v rámcích GOOSE a Sampled Values

Jakýkoliv zásah do komunikace třetí stranou je neoprávněný a správně by k němu nemělo dojít. Následkem modifikace komunikace může být změna přenášených hodnot, změna cílové adresy, nebo také zakrytí kritických hodnot. Obecný princip modifikace komunikace je uveden na *Obr. 7-7*.

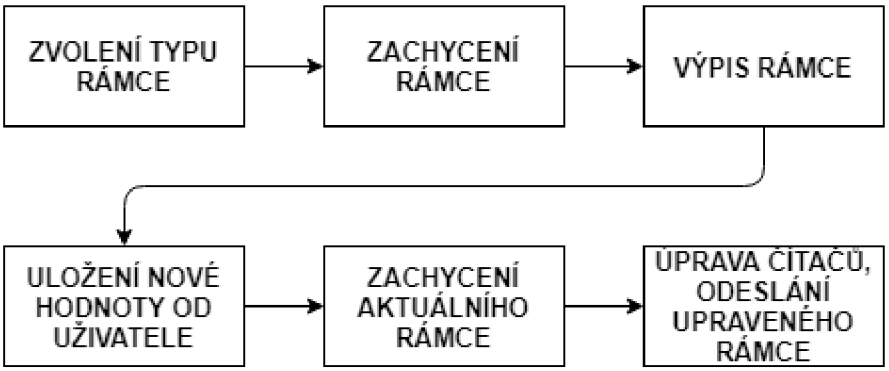
Logické uzly (node, vydavatelé) systému rozesílají rámce typu GOOSE, které odebírají přihlášení odběratelé. Informace obsažené v rámci jsou pro ně určitým způsobem užitečné, buď je sbírají a následně odesílají nadřízené stanici, nebo na jejich základě řídí svá připojená zařízení. Tyto informace jsou obvykle aktuální hodnoty připojených zařízení k danému uzlu (měřící prvky, spínače). Podle jejich hodnot se řídí další části systému a v případě kritických hodnot se mohou třeba také vypnout nebo jinak změnit své chování.

Pro správné provedení útoku je potřeba nejen zachytit rámec, zjistit, na kterém místě v rámci jsou která data uložena, ale také je správně vyčíst, zjistit kolik místa v rámci zabírají, získat novou hodnotu a vložit ji na správné místo ve správném tvaru.

Cílem tohoto útoku je zjistit, jak složité je pozměnit tato data a doručit rámec odběrateli tak, aby jej přijmul a zpracoval.

Obrázek 7-7 Princip modifikace komunikace.

Útok se v aplikaci scada_tester spustí pomocí klávesy „c“, což je volba „[C]hange values in a single GOOSE or SV frame“. Následně je uživatel vyzván k volbě typu rámce. Do konzole se vypíše aktuální zachycený rámec zvoleného typu, včetně všech jeho polí a hodnot. Uživatel má na výběr, které z polí změnit a jakou hodnotou. Po potvrzení nové hodnoty se zachytí nový rámec, aby aplikace zjistila aktuální stav čítačů, jelikož interakce s uživatelem trvala určitou dobu a dříve zachycený rámec již nemusí mít aktuální hodnoty čítačů. Po zjištění aktuálních hodnot čítačů se rámec patřičně upraví, vloží se modifikovaná hodnota a odešle se zpět do sítě. Tento postup znázorňuje blokové schéma na *Obr. 7-8*.



Obrázek 7-8 Blokové schéma útoku změny hodnot.

Součástí útoku byly také pomocné funkce pro převod hodnot rámce z hexadecimálního tvaru, jak jsou reprezentovány v paměti, do správných datových typů, aby bylo možné je správně uložit do proměnných a zobrazit v konzoli. Nápodobně, akorát v opačném pořadí bylo nutné nové hodnoty převést z jednotlivých datových typů do hexadecimálního tvaru. Avšak ne na všechny datové typy lze použít stejné převodní funkce, jelikož např. vícebajtové pole znaků je reprezentováno jako znak po znaku, a tudíž je nutné také převádět znak po znaku, ale kupříkladu celočíselný datový typ integer, v jazyce C++ standardně dlouhý 4 bajty, je nutné převést jako celek.

Na obrázcích *Obr. 7-9* a *Obr. 7-10* je zobrazen výpis rámce z aplikace Wireshark před a po změně hodnoty. Změněná hodnota je zvýrazněna červeně, jedná se o pole „gocbRef“ a bylo změněno z hodnoty „MEg44PANLD0/LLN0GOTemperature“ na „Zmena hodnoty v gocbRef“.

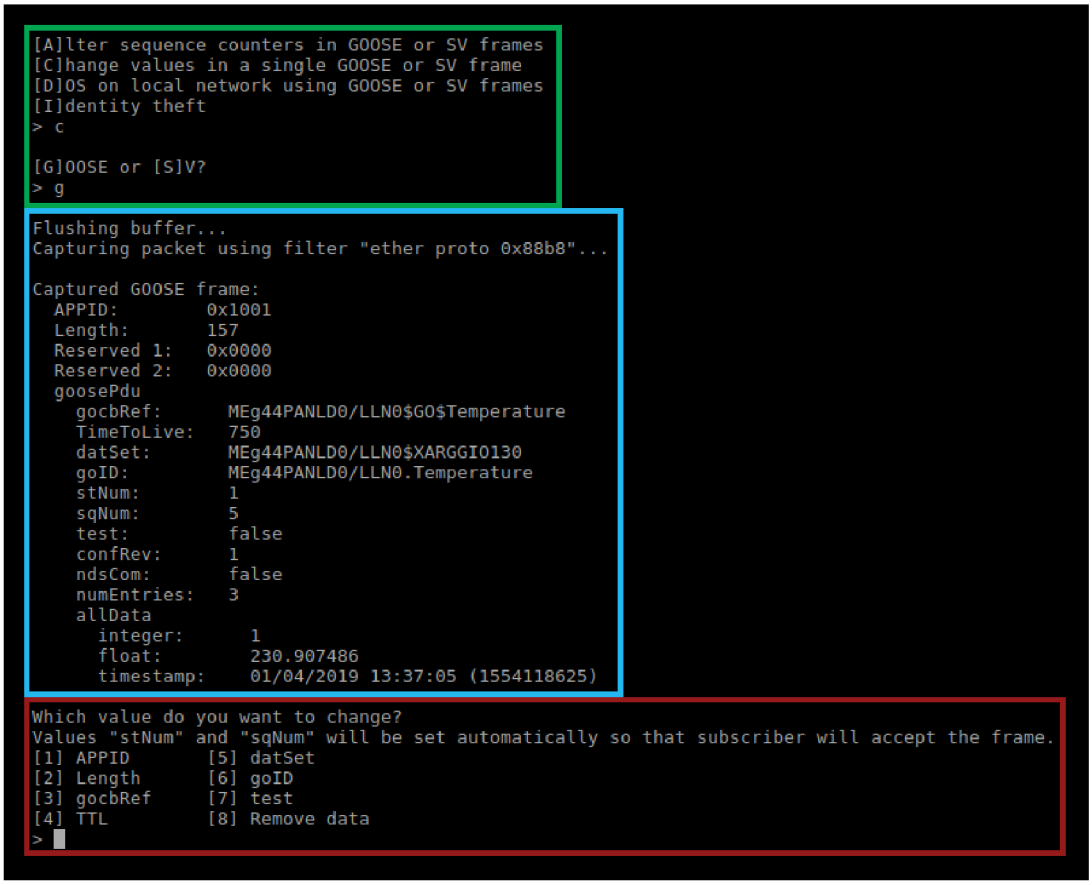
```
GOOSE
  APPID: 0x1001 (4097)
  Length: 157
  Reserved 1: 0x0000 (0)
  Reserved 2: 0x0000 (0)
  goosePdu
    gocbRef: MEg44PANLD0/LLN0$GO$Temperature
    timeAllowedtoLive: 750
    datSet: MEg44PANLD0/LLNO$XARGGIO130
    goID: MEg44PANLD0/LLN0.Temperature
    t: Jun 9, 2019 9:37:00.059999942 UTC
    stNum: 1
    sqNum: 35
    test: False
    confRev: 1
    ndsCom: False
    numDatSetEntries: 3
    allData: 3 items
```

Obrázek 7-9 Změna hodnoty rámce – původní rámec.

```
GOOSE
  APPID: 0x1001 (4097)
  Length: 157
  Reserved 1: 0x0000 (0)
  Reserved 2: 0x0000 (0)
  goosePdu
    gocbRef: Zmena hodnoty v gocbRef
    timeAllowedtoLive: 750
    datSet: MEg44PANLD0/LLNO$XARGGIO130
    goID: MEg44PANLD0/LLN0.Temperature
    t: Jun 9, 2019 9:37:00.059999942 UTC
    stNum: 1
    sqNum: 36
    test: False
    confRev: 1
    ndsCom: False
    numDatSetEntries: 3
    allData: 3 items
```

Obrázek 7-10 Změna hodnoty rámce – modifikovaný rámec.

Na obrázku *Obr. 7-11* je zobrazen průběh změny hodnoty rámce. Zelený rámec označuje hlavní menu volby útoku a následně výběr typu rámce. V modrém rámci je zachycení a výpis rámce daného typu. V červeném rámci pak možnost volby, které pole rámce se bude modifikovat.



Obrázek 7-11 Změna hodnoty rámce v aplikaci scada_tester.

Modifikace rámce při testování útoku byla úspěšná. Upravenou hodnotu bylo možné vidět v zachyceném rámci pomocí aplikace Wireshark. Upravený rámec byl bez problému přijat koncentrátorem a pokud by se změnila jedna z hodnot, které koncentrátor standardně vypisuje do konzole s každým přijatým rámcem, bylo by změnu možno vidět i tam.

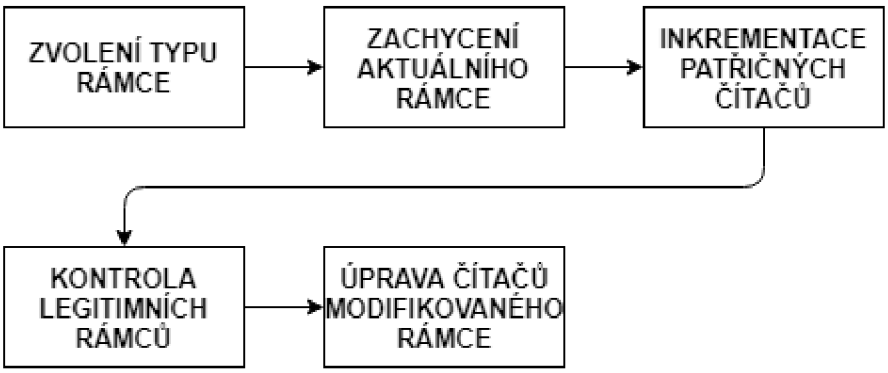
7.4 Krádež identity uzlu

Rámce typu GOOSE a Sampled Values jsou rozesílány broadcastově, tudíž rámec mohou přijmout i zařízení, kterým nebyl určen. V případě, že odběratel nepřijímá takové rámce, jejichž hodnota čítačů je nižší, než jakou má uloženou jako poslední přijatou, tyto rámce nezpracuje a zahodí. Tohoto chování může útočník využít, pokud přijme legitimní rámec, ten si u sebe uloží, inkrementuje mu patřičné čítače v závislosti na typu rámce a takový rámec vyšle zpět do sítě. Nastane situace, kdy odběratel takto upravený rámec přijme a považuje za legitimní a pocházející jako další rámec od vydavatele. Další a opravdu legitimní rámec od patřičného vydavatele tak odběratel nezpracuje a bude jej považovat za duplicitní, jelikož čítače budou mít stejnou hodnotu, jakou už má u sebe uloženou. Takto může útočník ukrást identitu uzlu na libovolně dlouhou dobu za podmínky, že si pravidelně kontroluje rámce od vydavatele, jehož identitu převzal a patřičně upravuje čítače v podvrženém rámci tak, aby byly vždy vyšší, než jsou čítače v rámci od vydavatele.

Ideální stav by byl tehdy, kdy by se útočník dostal přímo mezi inkriminovaného vydavatele a přepínač, ke kterému je vydavatel připojený, takzvaný „muž uprostřed“ (man in the middle). V tomto případě by útočník mohl odchytávat rámce přímo od vydavatele, u sebe je upravit a upravené je dále poslat na přepínač a z něj dále do sítě. To je ovšem složité na provedení, jelikož v případě připojení vydavatele kabelem tato operace vyžaduje fyzický přístup k přepínači a manipulaci s připojením. Tím vzniknou nedoručené rámce, a to může být popud bezpečnostních techniků k provedení kontroly.

Cílem tohoto útoku je po určitou dobu převzít identitu uzlu. Útok je možné spustit volbou „i“, což je „[I]dentity theft“. Tento útok zaručuje, že koncentrátor vždy přijme podvržený rámec namísto legitimního rámce. Na začátku je uživatel opět vyzván ke zvolení typu rámce. Následně aplikace scada_tester zachytne odpovídající rámec a zjistí hodnoty čítačů. Pro rámec typu GOOSE jsou to čítače `stNum` a `sqNum`. Čítač `stNum` je inkrementován o 1, čítač `sqNum` je vynulován. Pro rámec typu Sampled Values je to čítač `smpCnt`, který je inkrementován o 1. Tento náskok je potřeba zachovat, proto je každý legitimní rámec zachycen a hodnoty jeho čítačů jsou kontrolovány. S každou jejich inkrementací je patřičně inkrementován i odpovídající čítač v modifikovaném rámci. Tímto je zaručeno, že koncentrátor bude přijímat a zpracovávat pouze modifikované

rámce a legitimní rámce zahodí, protože pro koncentrátor budou mít neaktuální hodnoty čítačů. Na obrázku *Obr. 7-12* je blokové schéma útoku.



Obrázek 7-12 Blokové schéma útoku krádeže identity.

Na obrázku *Obr. 7-13* je výstup koncentrátoru, kde lze vidět přijaté modifikované rámce a odmítnuté legitimní rámce. Modře jsou zvýrazněné podvržené přijaté rámce, šedě pak legitimní nezpracované rámce.

```
MEg44PAN GOOSE, stNum: 1, sqNum: 10, delta t: 497
  Sensor: 1
  Voltage: 229.179 V
  Timestamp: 01/04/2019 13:37:10

MEg44PAN GOOSE, stNum: 2, sqNum: 0, delta t: 0
  Sensor: 1
  Voltage: 54.0023 V
  Timestamp: 01/04/2019 13:37:10

Warning - GOOSE frame values for APPID 0x1001 differ from saved values!
  Last accepted - stNum: 2, sqNum: 0
  Received      - stNum: 1, sqNum: 11

MEg44PAN GOOSE, stNum: 2, sqNum: 1, delta t: 500
  Sensor: 1
  Voltage: 58.9153 V
  Timestamp: 01/04/2019 13:37:11

Warning - GOOSE frame values for APPID 0x1001 differ from saved values!
  Last accepted - stNum: 2, sqNum: 1
  Received      - stNum: 1, sqNum: 12

MEg44PAN GOOSE, stNum: 2, sqNum: 2, delta t: 498
  Sensor: 1
  Voltage: 52.8331 V
  Timestamp: 01/04/2019 13:37:12
```

Obrázek 7-13 Výstup koncentrátoru s legitimními a podvrženými rámci.

Tento útok je také nezbytnou součástí všech ostatních útoků aplikace scada_tester, jelikož během zpracování a modifikace rámců mohlo dojít k tomu, že čítače rámce by nebyly nadále aktuální, a tudíž by koncentrátor modifikovaný rámec nepřijal.

S výstupem tohoto útoku lze považovat útok za funkční.

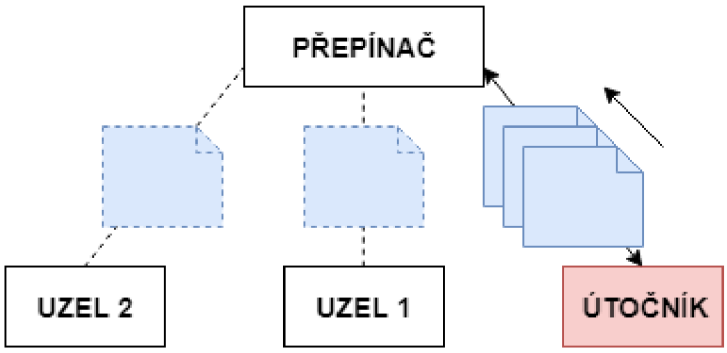
7.5 Záplavový útok

Záplavový útok (flood attack) je jedním z běžnějších typů útoků. Jednou z prerekvizit je přístup k zařízením, na které útočník následně rozesílá zprávy takovým způsobem, že ochromí jednotlivá zařízení, nebo prvky, které síť obsluhují. Cílem může být také zařízení na Internetu, které je veřejně přístupné a útočník tedy nemusí být přítomen přímo v síti společně s cílovým zařízením. Princip útoku je zobrazen na *Obr 7-14*.

Tento typ útoku se také označuje jako „odmítnutí služby“ (denial of service), což je odvozeno ze situace, kdy je cílová destinace pod útokem a není schopna poskytovat službu. Existují i silnější verze tohoto útoku, kdy útočník využívá celého spektra různých zařízení, též zvané botnet, která jsou mnohdy nakažená škodlivým kódem a majitelé těchto zařízení ani netuší, že jsou součástí právě probíhajícího útoku.

Záplavové útoky se liší způsobem zpracování a specializací na určitý druh zařízení či služby, obecný postup je ovšem vždy stejný. Cílem útoku je ochromit zařízení natolik, aby nebyla schopna zpracovávat legitimní provoz v síti. Výsledkem je buď značně zpomalená síť s výpadky zpráv, nebo zcela nefunkční síť, která není schopna zpracovávat provoz v síti.

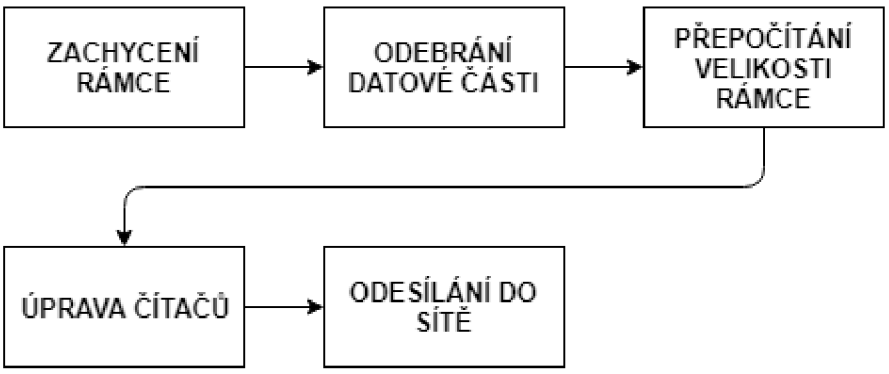
V simulovaném systému SCADA bude útok směřován na konkrétní stanici odběratele s cílem porovnat schopnost stanice zpracovat ostatní legitimní datové toky, které standardně přijímá, během útoku.



Obrázek 7-14 Princip záplavového útoku.

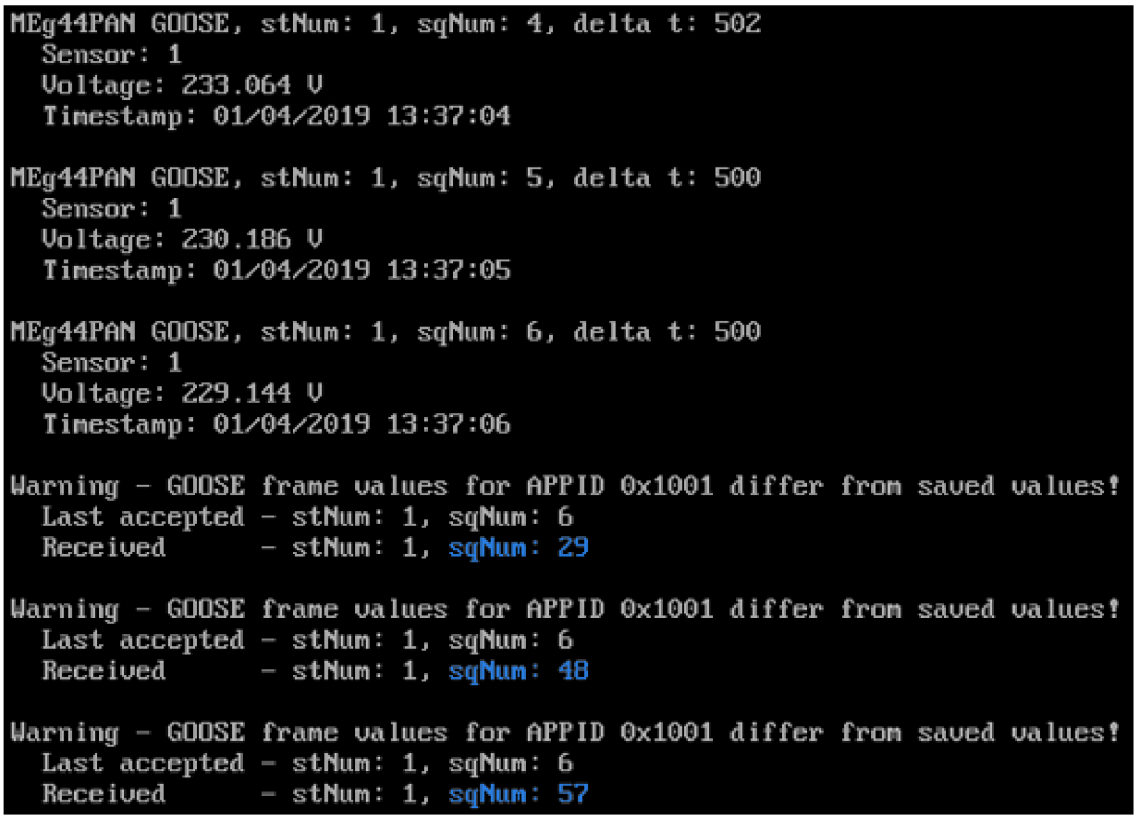
Stisknutím klávesy „d“, což je volba „[D]OS on local network using GOOSE or SV frames“ se spustí záplavový útok. Tento útok je cílen na konkrétní stanici, nikoliv na celou síť. Cílovou stanici zahltí vysokým množstvím malých rámců, které cílová stanice odebírá a tím vyčerpá veškerý její výpočetní výkon a zahltí ji. Stanice nadále není schopná poskytovat službu a je vyřazena z provozu. Díky tomu, že rámce jsou minimální velikosti, nezahltí celou propustnost sítě, ale jen stanice, které jsou přihlášeny k jejich odběru.

Aplikace scada_tester zachytí jeden rámec, ze kterého odstraní datovou část a přepočítá jeho velikost, aby byl rámec stále validní. Tímto se sníží celková velikost rámce, ale je nadále zpracován koncentrátorem. S nižší velikostí je aplikace schopna poslat větší množství rámců a tím zvýšit sílu útoku. Každý rámec má také patřičně upraveny čítače, aby jej koncentrátor považoval za správný. Na obrázku *Obr. 7-15* je blokové schéma záplavového útoku.



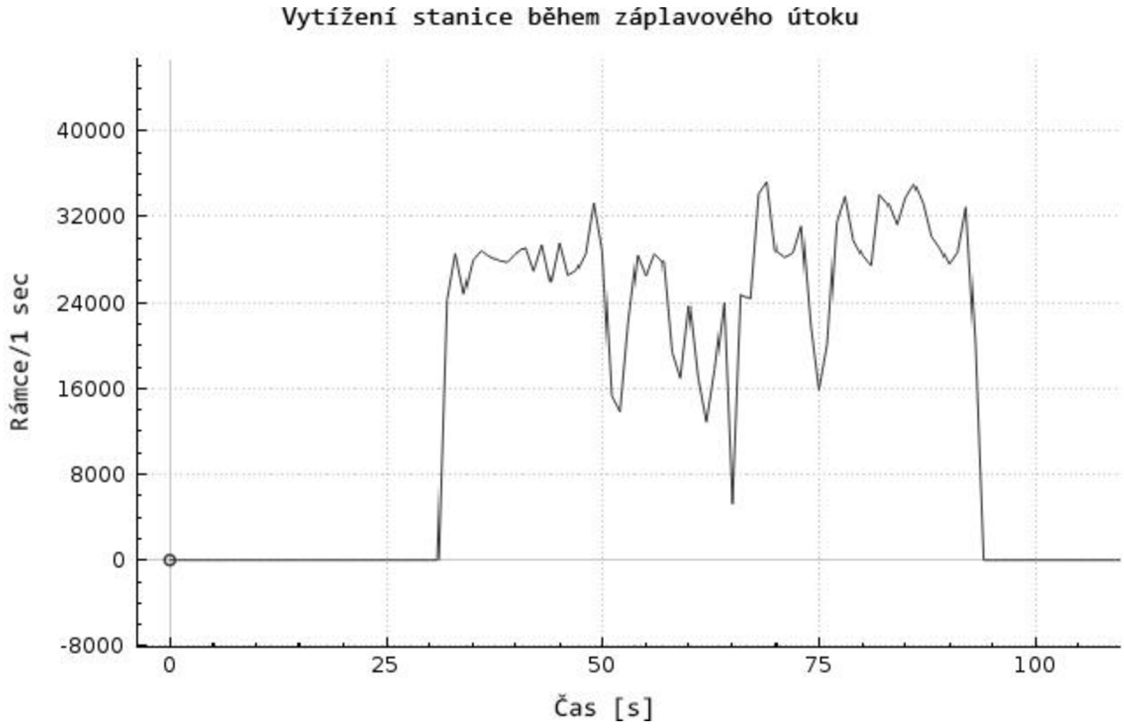
Obrázek 7-15 Blokové schéma záplavového útoku.

Na obrázku *Obr. 7-16* je konzolový výstup koncentrátoru. První tři záznamy značí úspěšné zpracování legitimního datového toku, další tři záznamy značí průběh záplavového útoku. Podle hodnoty čítače `sqNum`, které jsou zvýrazněny modrou barvou, je vidět, že stanice je zahlcená a nestíhá zpracovávat příchozí rámce.



Obrázek 7-16 Průběh záplavového útoku.

Vytížení stanice během záplavového útoku zobrazuje *Obr. 7-17*. Útok začal zhruba v 31. sekundě a trval po dobu 65 sekund. Počet zasílaných rámců dosahoval v průměru 28 tisíc rámců za sekundu, ve špičce až 36 tisíc rámců za sekundu. Hladina zasílaných rámců není stabilní z důvodu nerovnoměrného vytížení hostovského počítače během útoku, na kterém byly virtuální stroje spuštěny, nicméně princip útoku je splněn a cílová stanice pod náporem útoku přestává zpracovávat legitimní rámce. Po ukončení útoku cílová stanice není schopna synchronizace s legitimními rámci a je nutné ji restartovat, aby opět začala legitimní ráme zpracovávat.



Obrázek 7-17 Vytížení cílové stanice během záplavového útoku.

7.6 Shrnutí simulovaných útoků

Zásadní podmínkou pro provedení útoků je schopnost útočníka připojit se do sítě, kde se vyskytuje potřebná komunikace. Další nutností je schopnost zachytit dané rámce. Samotné připojení útočníka a pasivní odposlech nemusí být nutně zaznamenán bezpečnostními opatřeními sítě.

Při změně hodnot čítačů v GOOSE a Sampled Values rámcích a jejich následném odeslání zpět do sítě za účelem narušení správné funkčnosti odběratele nemusí mít útočník příliš velkou znalost o cílovém systému, postačí mu pouze vědět, které APPID konkrétní odběratel odebírá. Poté je možné vyfiltrovat odpovídající rámce, ty upravit a odeslat. Složitost tohoto útoku proto není velká, odhalení útoku taktéž nemusí být jednoznačné. Vše ovšem záleží na provedených bezpečnostních opatřeních a nastavení systému.

Při krádeži identity uzlu, podobně jako u útoku změny hodnot čítačů, postačí útočníkovi znalost zdrojové MAC adresy, která je jediným identifikátorem identity uzlu. V tomto případě však útok může být jednoduše odhalen, trvá-li jeho průběh delší dobu. Rámce vysílané útočníkem vytváří určitou pravidelnost, která může být alarmem pro bezpečnostní systém.

Změna hodnot v rámcích GOOSE a Sampled Values je složitější na provedení, jelikož je nutné útok spojit s krádeží identity, aby byl upravený rámec přijat odběratelem. Další složitější část představuje rozdělení zachyceného rámce na jednotlivé části a jejich následnou změnu, protože je nutné znát strukturu rámce, délku jednotlivých polí v bajtech a způsob uložení dat v rámci. Dopadem tohoto útoku může být vypnutí stanice, cílené spuštění alarmu v případě nečekané hodnoty či jiná řetězová reakce zařízení, které závisí na odběrateli.

Provedení záplavového útoku cíleného na stanici pro útočníka znamená znalost APPID, k jehož odběru je odběratel přihlášen, zachycení takového rámce a úpravu datové části rámce. Je taktéž nutné útok spojit s krádeží identity, aby upravené rámce byly odběratelem zpracovány. Výsledkem rychlého zasílání takto upravených rámců je nadměrné množství rámců v síti a neschopnost odběratele takové množství rámců zpracovat. Zároveň je nemožné tento útok skrýt před bezpečnostním systémem, proto je snaha útoku omezit cílovou stanici ještě předtím, než bude útok detekován a eliminován.

8. ZÁVĚR

Cílem práce bylo seznámení se se systémy SCADA. Konkrétně s topologií takového systému, zařízeními, jenž se v systému vyskytují a také s jejich funkcemi. Na základě znalosti topologie a zařízení bylo možno určit vektory útoků.

Další částí bylo přiblížení tří významných protokolů, a to IEC 60870-5, DNP3 a IEC 61850. Tyto protokoly byly podrobněji rozepsány, větší důraz byl však kladen na poslední jmenovaný protokol, IEC 61850, na který byly také útoky cíleny.

Zvolené útoky vycházely zejména ze slabin protokolu, což je převážně absence šifrování komunikace, tudíž je možné komunikaci využít jako zdroj všech realizovaných útoků. Byla dána přednost útokům krádeže identity, modifikace komunikace mezi zařízeními a úpravám čítačů v rámcích.

Součástí práce byl také návrh simulovaného prostředí vycházejícího ze znalostí nabytých v teoretické části. Pro potřeby práce bylo zvoleno virtuální prostředí, kde se odehrávala veškerá činnost. Tím odpadla potřeba laboratorního vybavení a usnadnila realizaci, manipulaci a nastavení prostředí.

V praktické části se podařilo zachytit komunikaci, která byla filtrovaná pomocí BPF filtru a zachycené rámce rozdělit z jednoho souvislého bloku v paměti na logické celky dat, se kterými se dalo nadále pracovat v rámci útoků.

Dále by realizovány čtyři útoky. Test chování odběratele při doručení rámců s různou hodnotou sqNum a stNum v případě rámce typu GOOSE a s různou hodnotou smpCnt v případě rámce typu Sampled Values. Odběratel se podle očekávání choval dle navržené funkce, kterou knihovna libiec61850 neimplementuje a jakékoliv zpracování rámce musí programátor řešit sám. Dalším útokem byla změna datových polí v rámcích typu GOOSE a Sampled Values. Poměrně složité bylo správně určit místo v paměti, kde daná data začínají, převést je do správného datového typu, upravit je a vložit zpátky. Jakmile tento problém byl vyřešen, nezáleželo už na tom, jaká data a kde je potřeba změnit. Na vkládání dat sloužily tři funkce, které uměly převést daný datový typ (int, float, string) zpět do hexadecimální podoby a vložit je na správné místo. Třetím útokem byla krádež identity, která proběhla bez problému, avšak odběratel může být alarmován faktem, že jsou vždy doručovány dva rámce najednou, jeden podvržený a druhý legitimní. Posledním realizovaným útokem byl záplavový útok s cílem odmítnutí služby, nebo alespoň omezení služby. Pátý útok, jenž měl za cíl změnit identifikátory dat v rámci, vložit dlouhý řetězec znaků, nebo změnit délku datových polí, byl realizován pouze teoreticky z důvodu podcenění časové náročnosti práce.

Každý z realizovaných útoků byl podrobně popsán v samostatné podkapitole včetně vysvětlujících a upřesňujících snímků obrazovky a blokových schémat. Cílů útoků bylo ve virtuálním prostředí dosaženo.

Literatura

[1] MAKHIJA, Jay; SUBRAMANYAN, L. R. Comparison of protocols used in remote monitoring: DNP 3.0, IEC 870-5-101 & Modbus. Electronics Systems Group, IIT Bombay, India, Tech. Rep, 2003.

[2] UZAIR, Muhammad. COMMUNICATION METHODS (PROTOCOLS, FORMAT & LANGUAGE) FOR THE SUBSTATION AUTOMATION & CONTROL (Project report of course 586 b) Dostupné z: http://www.eng.uwo.ca/people/tsidhu/Documents/project%20report%20Uzair.pdf

[3] CLARKE, Gordon R, Deon REYNDERS a Edwin WRIGHT. Practical modern SCADA protocols: DNP3, 60870.5 and related systems. London: Elsevier, 2004. Engineering: instrumentation & control: instrumentation & control. ISBN 075067995.

[4] Komunikační protokoly pro dálkové ovládání IEC/ISO 60870-5 [online]. 2010 [cit. 2018-11-10]. Dostupné z: http://automa.cz/cz/casopis-clanky/komunikacni-protokoly-pro-dalkove-ovladani-iec/iso-60870-5-2010_02_40552_5799/

[5] IEC 61850: soubor norem pro komunikaci v energetice s velkým potenciálem výhod [online]. 2010, 2010(03) [cit. 2018-11-13]. Dostupné z: http://automa.cz/cz/casopis-clanky/iec-61850-soubor-norem-pro-komunikaci-v-energetice-s-velkym-potencialem-vyhod-2010_03_40771_5154/

[6] Learn IEC 61850 configuration in 30 minutes [online]. 2018 [cit. 2018-11-14]. ISBN 978-1-5386-6127-7. Dostupné z: https://ieeexplore.ieee.org/document/8349803

[7] IEC 61850 Communication Networks and Systems In Substations: An Overview for Users [online]. 2009, 2009(01) [cit. 2018-11-18]. Dostupné z: http://www.gegridsolutions.com/multilin/journals/issues/spring09/iec61850.pdf

[8] IDC's Worldwide Quarterly Ethernet Switch Tracker Shows Solid Growth in Q2 2018 While Router Market Sees Mixed Results. In: IDC: The premier global market intelligence firm [online]. Framingham (Massachusetts), 2018, 6 Sep 2018 [cit. 2018-12-10]. Dostupné z: https://www.idc.com/getdoc.jsp?containerId=prUS44262218

[9] Downloads | libIEC61850 / lib60870-5. LibIEC61850 / lib60870-5 | open source libraries for IEC 61850 and IEC 60870-5-104 [online]. [cit. 2018-12-10]. Dostupné z: https://libiec61850.com/libiec61850/downloads/

[10] Pt 100-Temperature relays Type TR1200IP. ZIEHL industrie-elektronik GmbH + Co KG [online]. [cit. 2019-05-23]. Dostupné z: https://www.ziehl.com/en/Temperature-Relays/detail/TR1200IP-40/

[11] Hourly Data Report for May 02, 2018 - Climate - Environment and Climate Change Canada. Home - Canada.ca [online]. [cit. 2019-05-23]. Dostupné z: http://climate.weather.gc.ca/climate_data/hourly_data_e.html?StationID=51459&timeframe=1&StartYear=1840&EndYear=2019&Day=2&Year=2018&Month=5

[12] Standards-oui.ieee.org/ethertype/eth.txt. IEEE - The world's largest technical professional organization dedicated to advancing technology for the benefit of humanity. [online]. [cit. 2019-07-22]. Dostupné z: http://standards-oui.ieee.org/ethertype/eth.txt

[13] MATOUŠEK, Petr. Description of IEC 61850 Communication. Brno, 2018. Dostupné také z: http://www.fit.vutbr.cz/research/pubs/index.php?file=%2Fpub%2F11832%2FTR-61850.pdf&id=11832

[14] BARANOV, Pavel F., Sergey V. MURAVYOV, Almaz O. SULAYMANOV a Lyudmila I. KHUDONOGOVA. Software for Emulating the Sampled Values Transmission in Accordance with IEC 61850 Standard. In: Proceedings of the 2nd International Symposium on Computer, Communication, Control and Automation [online]. Paris, France: Atlantis Press, 2013, 2013, s. - [cit. 2019-07-23]. DOI: 10.2991/3ca-13.2013.116. ISBN 978-90786-77-91-8. Dostupné z: http://www.atlantis-press.com/php/paper-details.php?id=10232

[15] Tcpdump/Libpcap public repository [online]. [cit. 2019-05-23]. Dostupné z: https://www.tcpdump.org/

Seznam příloh