BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2018
http://excel.fit.vutbr.cz

Bezpečnosť SCADA systémov a protokol DLMS

Ján Pristaš*

Abstrakt
SCADA (Supervisory Control And Data Acquisition) systém je typ architektúry riadiaceho systému využívajúceho počítače, sieťové prepojenie a rôzne vzdialene riadené objekty. Ide teda o vzdialené riadenie a zber dát. Vo všeobecnosti systém obsahuje riadiacu stanicu a niekoľko vzdialených staníc, ku ktorým sú pripojené rôzne riadené objekty. Objekty môžu byť merače spotreby energie, rôzne typy snímačov ap. Ako u väčšiny odvetví počítačových sietí ani tieto systémy sa nevyhnú hrozbe rôznych útokov, ktoré môžu systému spôsobiť veľké škody. Táto práca sa zaoberá vytváraním/simuláciou rôznych druhov útokov na testovacie siete a zisťovaním reakcií systému na ne. Súčasťou práce je zhodnotenie použiteľnosti rôznych simulačných nástrojov na prevádzku SCADA systémov, vytvorenie simulačného programu a následné vytvorenie testovacieho prostredia, ktoré bude schopné simulovať jednotlivé typy útokov. Výstupom práce je datová sada vo formáte .pcap, ktorá bude obsahovať jednotlivé útoky spolu s popisom reakcií systému a možnosťami detekcie.

Kľúčové slová: Počítačová komunikácia — Bezpečnosť — SCADA — DLMS/COSEM

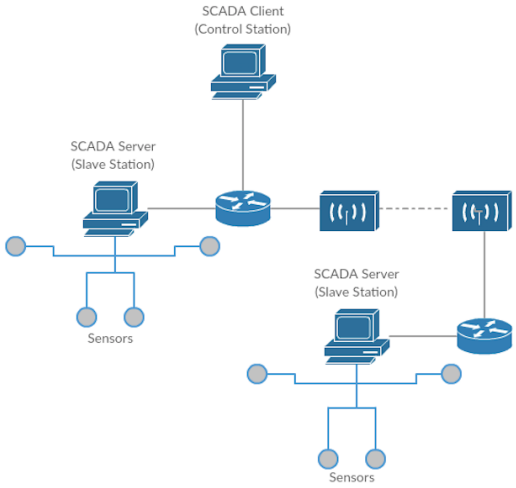
Priložené materiály: Kód na stiahnutie

*xprist06@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Úvod do problematiky

[Motivácia] SCADA systémy sú dnes veľmi využívané mnohými spoločnosťami. Najmä v rôznych výrobných (výrobné linky, baliace linky, skladové systémy) a energetických (elektrárne, teplárne, výmenníkové stanice) závodoch, ale aj v technológiach budov (vzduchotechnika, zabezpečenie, dochádzkové systémy) a v ekológií (emisný monitoring, čističky odpadných vôd). V Českej republike sú SCADA systémy využívané napríklad spoločnosťami ako RWE, EON alebo skupina ČEZ. Útoky na takéto veľké komplexy môžu mať obrovský dopad nielen na spoločnosť ako takú, ale aj na bežných užívateľov. Napríklad na jar roku 2000 sa bývalý zamestnanec istej austrálskej softvérovej spoločnosti uchádzal o zamestnanie v miestnej samospráve, avšak jeho žiadosť bola zamietnutá. Následne nespokojný uchádzač pomocou rádiového vysielača vzdialene infikoval kontrolný systém čistenia odpadných vôd a zmenil elektronické údaje pre konkrétne kanalizačné čerpacie stanice. To malo za následok poruchy v prevádzke systému a následné vypustenie viac ako 950 000 litrov odpadných vôd do blízkych riek a parkov, čo malo veľký efekt na miestny ekosystém [1, p. 3-20]. Počítačové útoky sú v dnešnej dobe skutočnou hrozbou. Jedným z hlavných dôvodov ohrozenia SCADA systémov je ich postupný prechod nad IP vrstvu. Systémy sa stále rozrastajú a vzdialenosť jednotlivých koncových staníc od riadiacej je čoraz väčšia, čo znemožňuje systému komunikovať na fyzickej vrstve a je potrebné prepojenie cez verejnú sieť. Na jednej strane je to užívateľsky veľmi prívetivé. Je možné vzdialene sledovať a riadiť veľmi veľké množstvo zariadení. Taktiež je možné centrálne vykonávať rôzne aktualizácie ap. Avšak prepojenie cez IP siete v značnej miere zjednodušuje útočníkom napadnúť a infikovať danú sieť. Preto je potrebné komunikáciu v sieti neustále monitorovať (flow monitorovanie) a mať k dispozícií nástroje (rôzne sondy ap.), ktoré sú schopné zachytiť neštandardnú komunikáciu v sieti a včas varovať pred bezpečnostným incidentom.

[Definícia problému] SCADA systémy sa skladajú z dvoch častí: strany klienta a strany serveru. Strana klienta zastrešuje riadiacu centrálu, ktorá vzdialene



Obrázok 1. Ukážka topológie SCADA systému

monitoruje a riadi pripojené prvky. Strana servera je vzdialená stanica, ktorá obsahuje pripojené rôzne inteligentné meracie zariadenia (snímače, teplomery, elektromery ap.), ktoré sú vzdialene čítané alebo riadené (nastavované) stranou klienta. V SCADA systémoch sú úlohy klienta a servera presne naopak ako je štandardne používaný model komunikácie klient-server. Na obrázku 1 je ukážka jednoduchej topológie SCADA systému.

V počítačových sieťach obecne existuje mnoho rôznych typov útokov. Avšak pri SCADA systémoch sa väčšina z nich neberie do úvahy nakoľko je pri nich predpoklad dostatočného zabezpečenia a konfigurácie firewallov na prepúšťanie iba najdôležitejšej komunikácie. To znamená, že očividné "diery" do systému boli uzavreté. Napriek tomu zostáva niekoľko typov útokov, ktorým sú SCADA systémy ohrozené - *útok cez komunikačné kanály, útok na server od riadiacej stanice, útok cez koncové zariadenie, útoky "zvnútra"*. Útoky môžu byť typu neoprávneného prístupu do systému, neoprávnený prístup s platným overením inej stanice, zasielanie nevalidných/neoprávnených príkazov, šírenie rôznych typov malware do systému, zasielanie neplatných/podvrhnutých odpovedí. Prípadne je možnosť fyzického napadnutia koncovej stanice a priameho napadnutia/infikovania systému. Medzi hrozby tiež patria neoprávnené akcie od obsluhujúceho personálu.
[Moje riešenie] Cieľom mojej práce je naštudovať dostupnosť a použiteľnosť simulačných a emulačných nástrojov na simuláciu prevádzky SCADA systémov. Zameral som sa na komunikačný protokol DLMS/-COSEM, ktorý sa využíva predovšetkým v energetike (elektrárne). Následne som na základe nových poznatkov vytvoriť prostredie schopné simulovať jednotlivé typy útokov a testovať reakcie systému na ne. Súčasťou toho je aj program napísaný v jazyku C++, ktorý simuluje riadiacu stranu v systéme. Umožňuje zasielať serveru rôzne zmenené/neštandardné správy a sledovať reakcie systému na ne. Výstupom práce je datová sada vo forme .pcap, ktorá obsahuje rôzne typy útokov spolu s popisom možností ich detekcie a prevencie pre nimi.
[Prínos práce] Táto práca môže pomocť mnohým spoločnostiam predísť nežiadúcim ohrozeniam ich systémov a efektívne sa brániť väčšine typov známych útokov na ich siete. Výsledky práce sú súčasťou projektu IRONSTONE[1] vo výzkumnej skupine NES@FIT.

2. Bezpečnosť SCADA systémov

Pri posudzovaní bezpečnosti a bezpečnostných rizík v SCADA systémoch sa rozlišuje medzi dvoma typmi útokov. Útoky cez SCADA kanále (SCADA channels) a útoky cez podporné kanále (maintenance channels) [2].

SCADA kanále sú komponenty slúžiace na primárne účely SCADA systémov - zber, prenos, skladovanie a spracovanie údajov a informácií z komunikácie medzi jednotlivými komponentami systému. Medzi typické komponenty SCADA systémov patria:

- Systém riadenia distribúcie - súbor aplikácií na sledovanie a riadenie systému
- SCADA servery
- Vzdialené koncové zariadenia
- Inteligentné meracie zariadenia
- Ochranne články

Podporné kanále sú systémy slúžiace na inštaláciu a údržbu vyššie spomenutých súčastí systému. Taktiež slúžia na sprostredkovanie komunikácie medzi nimi. Typicky to sú:

- Inžinierske stanice
- Spúšťacie (commissioning) servery
- Servery na synchronizáciu času v systému
- Monitorovacie a logovacie servery (SNMP, syslog)

Hlavným dôvodom rozlišovania medzi jednotlivými súčasťami systému je to, že každý so sebou nesie rozličné bezpečnostné riziká. Komunikácia cez SCADA kanály je väčšinou typu počítač-počítač a prenašajú sa iba prevádzkové data, nie konfiguračné zmeny alebo binárky. Narušenie alebo zneužitie sa v monitorovanej komunikácií dá iba ťažko skryť. To znamená, že je väčšina útokov relatívne rýchlo detekovateľná. Avšak pri podporných kanáloch je to o niečo komplikovanejšie. Autorizované zmeny na

[1] Projekt Ironstone http://www.fit.vutbr.cz/units/UIFS/grants/index.php.cs?id=1101

jednotlivých komponentoch vykonané pracovníkom spoločnosti sa moc nelíšia od neautorizovaných zmien vykonaných útočníkom. A pretože podporné služby vyžadujú privilegovaný prístup do systému, je to veľmi lákavé pre útočníkov, ktorí chcú získať permanentnú kontrolu nad systémom. Je ale možné spoľahlivo monitorovať aj túto časť siete, avšak iba ak sa všetky podporné procesy vykonávajú "disciplinovane".

Každá spoločnosť čelí rôznym hrozbám v závislosti na technológií, ktorú využívajú. Avšak vo všeobecnosti je iba niekoľko možností, ako je možné SCADA systémy narušiť. Medzi ne patria:

- Útok cez SCADA kanál
- Útok cez podporu SCADA serverov
- Útok cez podporu koncových zariadení
- Útok "zvnútra"

Táto práca je zameraná najmä na útoky cez SCADA kanál a cez podporu koncových zariadení. Z prvého typu sa venujem útokom ako je neautorizovaný prístup do systému s platnými oprávneniami, zasielanie príkazov od neautorizovaných hostov a pripojenie do systému cez inú kontrolnú stanicu. Z druhého typu sa venujem neautorizovaným zmenám v koncových zariadeniach a prieniku do systému cez nové sieťové zariadenie.

Pri neautorizovanom prístupe do systému existuje niekoľko možností. Jednou z nich je prístup cez koncové zariadenie. Avšak ak je systém dostatočne zabezpečený, útok nemá na systém príliš veľký vplyv. Jediné, čo môže útočník robiť, je zasielať riadiacemu stredisku nesprávne namerané hodnoty. Keď je napadnuté iba jedno zariadenie, efekt útoku je minimálny. Pokiaľ sa ale útočníkovi podarí pripojiť do systému cez vlastné riadiace rozhranie, môže v podstate vykonávať akékoľvek úkony, ktoré aj bežny pracovník. Útočníci môžu napríklad zasielať riadiace príkazy koncovým zariadeniam, čo môže spôsobiť vážne škody na systéme. Taktiež je možné komunikáciu po ceste odchytiť a zmeniť určité hodnoty v príkazoch, ktoré budú mať stále platné oprávnenia. Systém si tak vôbec nemusí uvodomiť, že mu prišla zlá odpoveď alebo príkaz. To je jedno z hlavných rizík pri prechode SCADA systémov nad IP vrstvu. Ďalší typ útoku, ktorému sa venujem, je prístup do siete cez novo vytvorené zariadenie. Útočník môže vložiť do siete vlastné zariadenie, čím získa jednoduchý prístup do siete. To sa týka najmä strany serveru a koncových staníc, nie riadiacej stanice, nakoľko sú koncové stanice menej strážené [2].

3. Testovacie prostredie

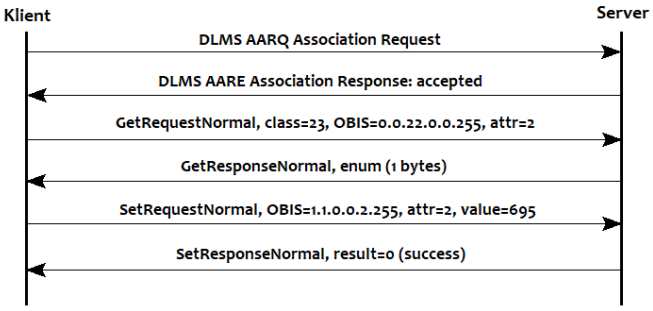
Podobne ako pri väčšine systémov v počítačových sieťach, aj pre SCADA systémy existuje velký počet rozličných komunikačných protokolov, každý so svojimi špecifikáciami. Táto práca je zameraná na komunikačný protokol DLMS/COSEM. Ide o štandard medzinárodnej elektrotechnickej komisie IEC 61334-4-41 [3].

3.1 Komunikačný protokol DLMS/COSEM

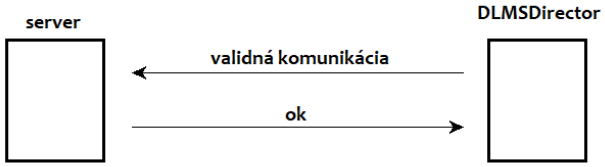
DLMS/COSEM je súbor štandardov pre výmenu údajov o spotrebe v energetike (elektrina, voda, plyn). Protokol pozostáva z niekoľkých častí, pričom každá špecifikuje určitú časť problematiky, ktorú rieši. DLMS (Device Language Message Specification) je špecifikácia aplikačnej vrstvy, nezávislá od nižších vrstiev, určená na podporu prenosu správ do a z koncových meracích zariadení. Cieľom je poskytnúť interoperabilné prostredie pre výmenu dát. COSEM (COmpanion Specification for Energy Metering) je špecifikácia na meranie spotreby energie. Ide o model rozhrania na komunikáciu s meracím zariadením s využitím objektovo-orientovaného prístupu. COSEM modeluje fyzické zariadenie ako súbor logických jednotiek, pričom každé má svoj jednoznačný identifikátor. Informácie obsiahnuté v jednotlivých logických jednotkách sú modelované objektami rozhrania. Objekty rozhrania sú špecifické pre danú doménu merania. Informácie ktoré uchovávajú sú usporiadané v atribútoch. Atribúty predstavujú vlastnosti a stav objektu pomocou ich hodnôt. Môžu obsahovať logické meno objektu, hodnotu, stav ap. Jednotlivé atribúty sú vždy špecifické pre každý typ zariadenia. Spoločný atribút je ale logické meno (OBIS kód), ktorý jednoznačne identifikuje zariadenie. Model COSEM umožňuje identifikáciu, vyhľadávanie a interpretáciu informácií uchovávaných v akomkoľvek meracom zariadení. Na obrázku 2 je ukážka komunikácie protokolu DLMS/COSEM medzi klientom a serverom. Ukážka obsahuje niekoľko bežných príkazov. Ako prvé načítanie informácií o jednotlivých pripojených meračoch (Association Request/Response). Druhá je žiadosť o navrátenie hodnoty druhého atribútu objektu s OBIS kódom 0.0.22.0.0.255. Ide o hodnotu prenosovej rýchlosti (baudrate) objektu typu `IecHdlcSetup`. Nakoniec je zaslaný príkaz o nastavenie druhého atribútu objektu s OBIS kódom 1.1.0.0.2.255. Je to atribút nesúci údaje o spotrebe v objekte typu `Data`.

3.2 Simulátory SCADA systémov

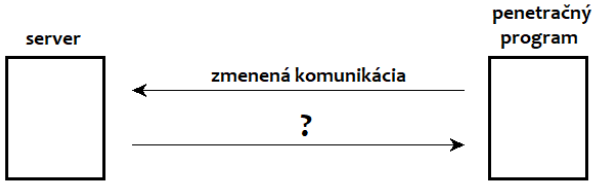
Po naštudovaní protokolu DLMS/COSEM som sa zameral na rôzne priemyselné simulátory a emulátory



Obrázok 2. Ukážka komunikácie protokolu DLMS/COSEM



Obrázok 3. Vzorový systém



Obrázok 4. Testovací systém

prevádzky SCADA systémov. K dispozícií som mal niekoľko nástrojov z ktorých som si vybral program DLMS Director od fínskej spoločnosti GuruX Ltd[2]. Nástroj je typu open source a je veľmi dobre spracovaný. Funguje však iba ako strana klienta, tzn. umožňuje iba vzdialené čítanie a nastavovanie hodnôt zo servera. Rovnaká spoločnosť ale poskytuje aj open source knižnicu[3] pre jazyk C++. Súčasťou knižnice je vzorový program pre stranu servera, ktorý dokáže dobre komunikovať s programom DLMS Director. Pomocou zachytenej komunikácie medzi oboma programami bolo vytvorených niekoľko testovacích .pcap súborov. Komunikácia obsahovala bežné príkazy na vzdialené čítanie, synchronizáciu hodín, vzdialené nastavenie hodnôt ap. Zachytávanie prebiehalo pomocou nástroja Wireshark. Testovanie preukázalo, že programi sú schopné vytvoriť komunikáciu odpovedajúcu štandardom protokolu DLMS/COSEM.

Pomocou programu DLMS Director bola vytvorená datová sada, ktorá slúžila ako vozorvý príklad pri ďalšom testovaní. Ukážku systému je možné vidieť na obrázku 3. Na základe tohto systému bol vytvorený testovací systém, pomocou testovacích programov, ktoré budú bližšie popísané v ďalšej časti tejto práce. Validácia systému preukázala, že odpovedá vzorovému systému a komunikácia odpovedá štandardu DLMS/COSEM, čo ho umožnilo použiť na testovanie rôznych útokov a sledovať reakcie systému. Od reálneho systému je možné očakávať obdobné reakcie na jednotlivé útoky. Ukážka prepojenia je na obrázku 4.

3.3 Testovací program

Protokol DLMS/COSEM a SCADA systémy vo všeobecnosti majú pevne danú štruktúru komunikácie, každý príkaz má štandardom určenú odpoveď. Na základe toho je chovanie systému predikovateľné a tým pádom aj ľahko odsledovateľné. Táto vlastnosť zjednodušila vytvorenie testovacieho prostredia a komunikácia bola jednoduchšie simulovateľná.

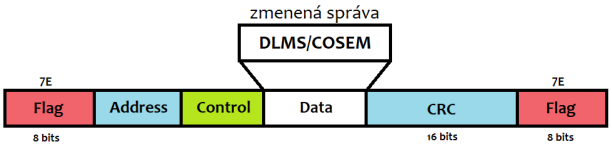
Pri vytváraní testovacieho prostredia pre jednotlivé útoky bol navrhnutý a implementovaný program v jazyku C++, ktorý v systéme simuluje stranu klienta. Na rozdiel od samotného protokolu DLMS/COSEM, ktorý pracuje na 7. (aplikačnej) vrstve TCP/IP modelu, program funguje na 4. (transportnej) vrstve. Vďaka tomu sa nemusí starať o konkrétne špecifikácie protokolu, stačí mu iba vytvoriť spojenie pomocou protokolu TCP a môže komunikovať. Je to veľmi podobné open source programu `tcpreplay`, avšak s pár rozdielmi. `Tcpreplay` dostáva na vstupe priamo zachytený .pcap súbor a je schopný odfiltrovať potrebnú komunikáciu a vykonať požadované zmeny. Program využívaný v tejto práci dostane na vstupe už zmenené príkazy v binárnej podobe a odošle ich. Práca s .pcap súbormi a zmeny v príkazoch sú vykonané ručne. Tento program je teda o niečo jednoduchší ako `tcpreplay`, ale výsledok v komunikácií je prakticky rovnaký.

Vstupom programu je binárny súbor obsahujúci príkazy, ktoré sa budú zasielať serveru. Po zaslaní správy čaká program na odpoveď. Výstupom programu je novo-vytvorený binárny súbor obsahujúci jednotlivé odpovede od servera. Odpovede sú následne porovnané so vzorovou komunikáciou a vyhodnocuje sa reakcia servera.

Protokol DLMS/COSEM využíva na prenos príkazov protokol HDLC, ktorý slúži na spoľahlivý prenos dát po sieti. Súčasťou datového rámca protokolu sú dva byty vyhradané pre tzv. kontrolný súčet CCITT-CRC16. Ukážka rámca je na obrázku 5. Pre každý zmenený rámec je potrebné vypočítať jeho kontrolný súčet. Ten sa počíta z celého rámca, okrem prvého a posledného (režijného) bytu. Ak by totiž súčet nesú-

[2]GuruX Ltd. `http://www.gurux.fi`

[3]C++ knižnica `https://github.com/Gurux/Gurux.DLMS.cpp`



Obrázok 5. HDLC zapúzdrenie

hlasil s dátami, strana príjemcu správu zahodí a viac nekomunikuje.

Výhoda práce programu na nižšej vrstve je v tom, že nie je potrebné poznať štruktúru protokolu a program je možné univerzálne použiť na simuláciu strany klienta pre akýkoľvek komunikačný protokol pracujúci nad prenosovým protokolom TCP. Program ale samotnú L7 komunikáciu nevytvára. Využíva zaslané príkazy z už odchytenej komunikácie uloženej v súboroch vo formáte .pcap. Po vytvorení jednotlivých zmien v príkazoch je možné sledovať reakciu druhej strany na neočakávaný príkaz/odpoveď a porovnať ho so vzorovou komunikáciou. Následne je možné získavať nové poznatky o chovaní systému v prípade rôznych typov narušení.

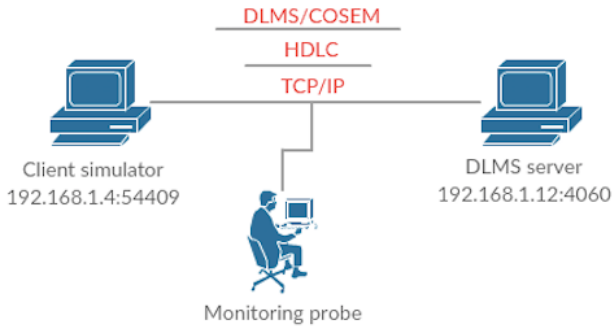
Samotné testovanie vždy prebieha v niekoľkých fázach:

1. Vygenerovanie validnej komunikácie - pomocou strany klienta a servera je vygenerovaná, a zachytená validná komunikácia, ktorá bude slúžiť ako vzorová
2. Úprava komunikácie zo strany klienta - podľa typu testovania je vykonaná zmena v príkazoch. Zmena je v správe odosielanej klientom serveru
3. Vygenerovanie upravenej komunikácie - jednotlivé príkazy zo vzorovej komunikácie spolu so zmenami sú programom klienta opätovne zaslané serveru
4. Vyhodnotenie chovania zariadenia - zachytená komunikácia je porovnaná s validnou (vzorovou) komunikáciou z bodu 1. Následne je vyhodnotená reakcia systému na zmenu v príkazoch

4. Výsledky testov

Pri testovaní bol použitý vyššie spomínaný program simulujúci stranu klienta a program pre server poskytovaný spoločnosťou GuruX. Bola vytvorená testovacia topológia medzi dvoma zariadeniami, pričom na jednom bol spustení program klienta a na druhom program servera. Ukážka topológie je na obrázku 6.

Implementácia programu servera bola po testovaní prehlásená za validnú s tým, že program sa chová a na jednotlivé správy reaguje ako reálne zariadenie. Preto bolo možné program použiť ako vzorový a po testovaní očakávať obdobné reakcie aj od reálneho zariadenia. Využitie simulačného programu má rovnako niekoľko



Obrázok 6. Testovacia topológia

výhod. Jednou z hlavných je to, že reálne zariadenie je veľmi obmedzené, hlavne čo sa týka hardwarových prostriedkov, ktoré má k dispozícii. Má obmedzenú hlavne operačnú pamäť a procesor. Nie je tak možné mať jeden univerzálny prístroj, pomocou ktorého sa bude dať vykonať veľká škála testov. V programe je ale možnosť si jednotlivé zariadenia ľubovolne nakonfigurovať podľa potreby a ďalej testovať.

Testovanie prebiehalo pomocou už zachytenej komunikácie, ktorá je uložená v súboroch typu .pcap. Jednotlivé príkazy klienta boli na binárnej úrovni čiastočne pozmenené a opätovne zasielané serveru. Následne boli sledované reakcie serveru na tieto zmeny. Zmeny boli zamerané najmä na `get` a `set` príkazy, spolu so zasielaním nesprávnych hodnôt, zmeny v informáciach o pripojených objektoch a neautorizovaný prístup k serveru s platnými autentizačnými údajmi. Zachytené .pcap súbory obsahujúce jednotlivé typy zmien a narušení boli uložené do github repozitáru[4].

4.1 Testovanie

Test č.1

Zmena: Data obsahovali zlú veľkosť pri otváraní asociácie. 11 bytov namiesto 9 bytov: 09 → 0b.
Reakcia: Server si zmenu vôbec nevšímal a zaslal spätne potvrdzovaciu správu o zistení deviatich pripojených zariadení.

Test č.2

Zmena: Dĺžka OID prvého požiadavku bola zle zadaná. 9 bytov namiesto 7 bytov: 07 → 09.
Reakcia: Server si opäť zmenu nevšimol. Žiadosť o asociáciu prijal a odpoveď obsahovala OID zmenené na správne.

Test č.3

Zmena: Dĺžka OID prvého požiadavku bola zle zadaná. 3 byty namiesto 7 bytov: 07 → 03.

[4]Github https://github.com/janpristas/bakalarska-praca

Reakcia: Reakcia bola rovnaká ako pri predchádzajúcom teste.

Test č.4

Zmena: Zlý typ paketu. Pôvodná správa bola typu `Get-Request` a bola následne zmenená na `Set-Request`: c0 → c1. Zmena bola vykonaná iba v poli obsahujúcom typ. Zvyšok paketu ostal pôvodný pre `Get-Request`.
Reakcia: Server prijal žiadosť na nastavenie hodnoty, vykonal zmenu a odpovedal potvrdzovacou správou o úspešnom nastavení.

Test č.5

Zmena: Zlý typ paketu. Namiesto `Get-Request` sa poslalo `Set-Response`: c0 → c5.
Reakcia: Server nevedel ako má zareagovať a na správu neodpovedal.

Test č.6

Zmena: Chybný OBIS kód požadovaného objektu. Požaduje sa 0.0.40.0.0.1 namiesto 0.0.40.0.0.255: ff → 01.
Reakcia: Server žiadosť prijal a pokúsil sa vrátiť požadovanú odpoveď. Odoslal ale prázdnu správu, nakoľko požadovaný objekt nepoznal.

Test č.7

Zmena: Zmena požadovaného atribútu objektu. Požaduje sa atribút 10 namiesto 1: 02 → 0a.
Reakcia: Server na zmenu reagoval štandardnou odpoveďou a vrátil hodnotu požadovaného atribútu. Tento test bol zameraný skôr na typ útoku, kedy útočník zachytí prebiehajúcu komunikáciu, čiastočne zmení obsah príkazu, ale ponechá ho validný a pošle pôvodnému adresátovi. Server správu príjme a odpovie, ale riadiacej stanici príde iná odpoveď akú požadovala. Ak nie je odpoveď dostatočne overená, že je to naozaj tá, ktorá bola očakávaná, môže to znamenať napríklad zlú informovanosť o nameraných hodnotách.

Testovanie autentizácie bolo zamerané hlavne na sledovanie prebiehajúcej komunikácie, kde bolo overované, že autentizačné kľúče sú prenášané v nešifrovanej podobe. Útočník si tak môže jednoducho ochytiť komunikáciu, prečítať si kľúč a vytvoriť svoje vlastné spojenie so systémom.

Testovanie preukázalo, že server sa pokúša na príkaz vždy odpovedať, nehľadiac na to, či požadovaný objekt alebo atribút pozná. Spojenie ukončuje iba v prípade nevalidného príkazu. Veľmi nebezpečný je tiež nešifrovaný prenos hesla po sieti, čo umožňuje útočníkom jednoduchý prístup k autentizačným údajom systému.

5. Záver

[Zhrnutie práce] SCADA systémy sú v dnešnej dobe veľmi rozšírené a stále pribúdajú nové spočnosti, ktoré ich využívajú. Avšak rovnako narastá aj počet útočníkov a hrozba, ktorej musia čeliť. Je preto veľmi potrebné a dôležité vedieť jednotlivé útoky včas detekovať a systém pred nimi chrániť. Táto práca môže pomocť pri skúmaní reakcií systému na jednotlivé útoky a tiež pri vývoji rôznych typov monitorovacích zariadení a sónd, ktoré budú schopné útoky včas odhaliť pri sledovaní prebiehajúcej komunikácie. Súčasťou práce je zhodnotenie rôznych simulačných nástrojov na prevádzku SCADA systémov využívajúcich komunikačný protokol DLMS/COSEM. Následne bol vytvorený program v jazyku C++ spolu s postupom vytvorenia simulačného prostredia, kde je možné jednotlivé útoky vytvárať a monitorovať reakcie systému na ne. Výstupom je datová sada vo formáte .pcap obsahujúca jednotlivé útoky spolu s popisom chovania systému. Záver práce je venovaný popisu detekcie útokov a spôsobu prevencie pred nimi.

Poďakovanie

Rád by som sa poďakoval pánovi Ing. Petrovi Matouškovi Ph.D., M.A., za odborné rady a vedenie pri vypracovaní tejto práce.

Literatúra

[1] Karen Scarfone Keith Stouffer, Joe Falco. Guide to industrial control systems (ics) security, 2011.

[2] Eric D. Knapp. *Industrial Network Security: Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems*. Syngress, 2011. ISBN: 1597496456.

[3] International Electrotechnical Commission. *IEC 61334-4-41:1996*. 1996. ASIN: B000Y2LTQI.