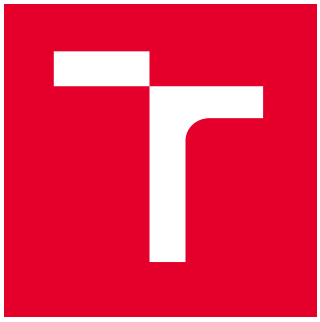


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

BEZPEČNOST PROTOKOLU DLMS/COSEM

DLMS/COSEM PROTOCOL SECURITY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. Tomáš Tomko**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Tomáš Lieskovan**
SUPERVISOR

BRNO 2022

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní program **Informační bezpečnost**

Ústav telekomunikací

Student: Bc. Tomáš Tomko ***ID:*** 203177

Ročník: 2 ***Akademický rok:*** 2021/22

NÁZEV TÉMATU:

Bezpečnost protokolu DLMS/COSEM

POKYNY PRO VYPRACOVÁNÍ:

Cílem diplomové práce je seznámení se s protokolem DLMS/COSEM, seznámení se s bezpečnostními riziky v průmyslových sítích a Cyber Range platformou KYPO. Student zprovozní v laboratorním prostředí platformu KYPO s bezpečnostním scénářem zaměřeným na bezpečnost DLMS/COSEM. Bezpečnostní scénář bude obsahovat odposlech komunikace a využití alespoň dvou bezpečnostních zranitelností.

DOPORUČENÁ LITERATURA:

[1] Gurux DLMS Server: https://www.gurux.fi/Gurux.DLMS.Server

[2] VYKOPAL, Jan, et al. Kypo cyber range: Design and use cases. 2017.

Termín zadání: 7.2.2022 ***Termín odevzdání:*** 24.5.2022

Vedoucí práce: Ing. Tomáš Lieskovan

doc. Ing. Jan Hajný, Ph.D.
předseda rady studijního programu

ABSTRAKT

Cieľom tejto diplomovej práce je oboznámiť čitateľa s protokolom DLMS/COSEM, ktorý sa využíva prevažne v energetike. V práci je predstavený pojem chytrých energetických sietí spolu s ich vývojom a základnou právnou úpravou. V nasledujúcej časti sú prebrané všetky protokoly používané v chytrých energetických sieťach s najväčším dôrazom na spomínaný protokol DLMS/COSEM a jeho bezpečnostné prvky a zraniteľnosti. Ďalej cieľom práce je zoznámiť čitateľa s bezpečnostnými rizikami v priemyselných sieťach a platformou Cyber Range KYPO. Dôležité bolo sa zoznámiť s možnosťami, ktorá táto platforma ponúka. Ďalšia úloha tejto práce je uviesť do prevádzky platformu KYPO v laboratórnom prostredí s bezpečnostným scenárom zameraným na bezpečnosť DLMS/COSEM. Bezpečnostný scenár, ktorý sa nám podarilo vytvoriť zahŕňa zachytávanie komunikácie a zneužitie aspoň dvoch bezpečnostných zraniteľností protokolu DLMS/COSEM. Všetky použité programy a nástroje, ktoré sme pre dosiahnutie cieľov tejto práce potrebovali sme zhrnuli podrobnejšie v samostatnej kapitole, ktorá sa venuje praktickej časti diplomovej práce. V práci je spísaný postup a problémy, na ktoré sme pri plnení zadania práce narazili. Samotný virtuálny scenár, ktorý sme pripravili beží na serveri v priestoroch univerzity VUT v Brne. Na tomto serveri je nainštalovaná Cyber Range platforma KYPO v konfigurácii all-in-one. Vytvorený bezpečnostný scenár s využitím protokolu DLMS/COSEM obsahuje dvoch užívateľov a jedného útočníka, ktorí sa nachádzajú v jednej sieti, tak ako sme to definovali. Na definíciu sieťovej konfigurácie sme použili jednoduchý textový editor a definovali sme ju pre platformu KYPO vo formáte YAML, ktorý slúži na automatizáciu dát vo forme, ktorá je pre človeka ľahko čitateľná. Rovnaký formát YAML sme použili aj pre definíciu playbookov, ktoré nám slúžia ako najjednoduchší spôsob v systéme Ansible na automatizáciu opakujúcich sa úloh. My sme ich využili na inštaláciu balíčkov, aktualizáciu stávajúcich balíčkov a prípravu virtuálneho prostredia na využitie zraniteľností protokolu DLMS/COSEM.

KĽÚČOVÉ SLOVÁ

Bezpečnosť, COSEM, Cyber Range, DLMS, elektromer, KYPO, Openstack

ABSTRACT

The aim of this thesis is to familiarize the reader with the DLMS/COSEM protocol, which is mainly used in the power industry. The thesis introduces the concept of smart grids along with their development and basic legislation. In the following section, all the protocols used in smart energy networks are discussed with most emphasis on the mentioned DLMS/COSEM protocol and its security features and vulnerabilities. Furthermore, the thesis aims to introduce the reader to the security risks in industrial networks and the Cyber Range KYPO platform. It was important to familiarize with the capabilities that this platform offers. The next task of this thesis is to operationalize the KYPO platform in a laboratory environment with a security scenario focused on DLMS/COSEM security. The security scenario that we have managed to create involves the interception of communication and the exploitation of at least two security vulnerabilities of the DLMS/COSEM protocol. All the programs and tools that we have used to achieve the objectives of this thesis have been summarized in more detail in a separate chapter, which is dedicated to the practical part of the thesis. The thesis describes the procedure and the problems that we encountered while completing the thesis assignment. The actual virtual scenario that we prepared runs on a server at the premises of the BUT. On this server is installed Cyber Range platform KYPO in all-in-one configuration. The created security scenario using the DLMS/COSEM protocol contains two users and one attacker located in the same network, as we have defined it. We used a simple text editor to define the network configuration and defined it for the KYPO platform in YAML format, which is used to automate the data in a form that is easy for humans to read. We also used the same YAML format to define playbooks, which serve as the easiest way in Ansible to automate repetitive tasks. We used them to install packages, update existing packages, and prepare the virtual environment to exploit DLMS/COSEM protocol vulnerabilities.

KEYWORDS

COSEM, Cyber Range, DLMS, KYPO, Openstack, Security, Smart meter

TOMKO, Tomáš. *Bezpečnost protokolu DLMS/COSEM*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2022, 102 s. Diplomová práca. Vedúci práce: Ing. Tomáš Lieskovan

Vyhlásenie autora o pôvodnosti diela

Meno a priezvisko autora:	Bc. Tomáš Tomko
VUT ID autora:	203177
Typ práce:	Diplomová práca
Akademický rok:	2021/22
Téma závěrečnej práce:	Bezpečnost protokolu DLMS/COSEM

Vyhlasujem, že svoju záverečnú prácu som vypracoval samostatne pod vedením vedú-cej/cého záverečnej práce, s využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej záverečnej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto záverečnej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávnych dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákonníka Českej republiky č. 40/2009 Sb.

Brno ....

podpis autora*

*Autor podpisuje iba v tlačenej verzii.

POĎAKOVANIE

Chcel by som sa poďakovať vedúcemu mojej diplomovej práce, pánovi Ing. Tomášovi Lieskovanovi, za odborné vedenie, konzultácie, trpezlivosť a návrhy pre túto prácu.

Obsah

Zoznam obrázkov

Zoznam tabuliek

Úvod

V dnešnej dobe množstvo oblastí ľudského života prechádza digitalizáciou. Táto téma sa týka aj energetiky. Digitalizácia prebieha v oblasti elektroenergetiky, plynárenstva či teplárenstva. Moderné technológie umožňujú využívať elektrinu efektívnejšie a môžu tak zaistiť väčšiu stabilitu celého energetického systému. V minulosti sa odčítanie rôznych spotrebných jednotiek vždy vykonávalo tak, že poverená osoba merač osobne skontrolovala, stav zapísala a potom zadala ručne do systému. Takýto postup je náročný na čas a tiež na náklady. Ako oveľa lacnejšia, rýchlejšia a elegantnejšia cesta sa ukazujú odpočty pomocou chytrých senzorov a prostredníctvom IoT. Vďaka tomu je možné odpočty vykonávať vzdialene, presne a prehľadne. Takáto integrácia inteligentných sietí a spotrebičov prináša aj riziko kybernetických útokov a aj tie by som rád v tejto diplomovej práci predstavil. Spolu s útokmi by som rád popísal a uviedol protokoly, ktoré sú pre komunikáciu v elektroenergetike a IoT nevyhnutné, a ktoré sa najviac používajú pričom najväčší dôraz sa kladie na protokol DLMS/COSEM. Príklad využitia tohoto protokolu vo virtuálnom prostredí je obsahom praktickej časti tejto práce.

Progresívny rozvoj v oblasti energetiky by mal priniesť výhody aj samotným občanom. Česká republika sa zaviazala, že dodrží Smernicu Európskeho parlamentu o energetickej účinnosti. Jej hlavným účelom je znižovanie energetickej náročnosti a snaha o energetické úspory vo všetkých oblastiach. Vďaka chytrým senzorom, ktorých údaje si je možné zobraziť v aplikácii, či na portáloch spoločností, sú odberatelia kedykoľvek informovaní o svojom aktuálnom odbere energií. Týmito informáciami sú nepriamo motivovaní k úspore obzvlášť pri nedávnych prudkých nárastoch ceny za energie. Smernica uvádza, že všetky chytré merače tepla, chladu a elektriny, vodomery na teplú vodu a iné, ktoré sú novo inštalované po 25. októbri 2020 by mali mať funkciu diaľkového odčítania. Ostatné staré merače sa majú vymeniť do roku 2027. To je hraničný rok od kedy by sa mala merať spotreba na diaľku u všetkých väčších odberateľov energií.

1 Chytré energetické siete

Energetika sa veľmi rýchlo mení a vyvíja. Prináša to so sebou mnoho výhod, ale aj výziev. Využívať ich a vyrovnávať sa s nimi pomáhajú chytré technológie. Ich nástup umožňuje rozvoj digitalizácie a informačných a komunikačných technológií. To všetko si môžeme predstaviť pod pojmom „chytrých sietí“.

1.1 Vývoj chytrých sietí

Chytré siete (nazývané aj „inteligentné siete“, anglicky „smart grids“) sú elektrické siete schopné efektívne prepojiť správanie a konanie všetkých užívateľov, ktorí sú k nim pripojení. Smart grid sa skladá z prenosových a distribučných sústav, ktoré sú vybavené istým stupňom inteligencie - teda schopnosťou automatizácie, komunikácie a regulácie. [1] Laicky povedané chytré siete prepájajú výrobcov elektriny, prevádzkovateľov sietí, obchodníkov s elektrinou a spotrebiteľov a umožňujú im spolu komunikovať a vzájomne spolupracovať. Výhodou takejto komunikácie a práce s dátami je ekonomicky efektívne využívanie energetickej sústavy, ktoré vedie k nižším stratám a zvyšuje energetickú účinnosť. Spotrebiteľom elektrickej energie napríklad umožňuje ľahšie sledovať svoju spotrebu a tiež využívať elektrinu vo chvíľach, keď je to najvýhodnejšie na základe odpozorovaných výsledkov a prípadne si môže na základe analýzy aj zvoliť rôzne tarify u poskytovateľov. V spojení so systémami inteligentného merania sa inteligentné siete dostávajú k spotrebiteľom a dodávateľom poskytovaním informácií o spotrebe v reálnom čase.

S vývojom vo svete a neustálym progresom vo všetkých smeroch je dôležité zavádzanie inteligentných sietí potrebné pre zabezpečenie spoľahlivej prevádzky elektrizačnej sústavy. Bude to nevyhnutné kvôli rastúcemu podielu zdrojov, ktorých výrobu možno dopredu horšie odhadovať, ako sú napríklad slnečné a veterné elektrárne. Patria sem hlavne dnes už bežné zdroje elektrickej energie z obnoviteľných zdrojov ale aj atypické ako je napríklad výroba elektriny využitím sopečnej energie a jej tepla. Podľa predpovede a nastoleného trendu sa stále viac elektriny tiež bude vyrábať v malých zdrojoch. Preto bude potrebné premeniť doterajší systém riadenia sietí, tak aby bolo možné zladiť objem vyrobenej a spotrebovanej elektriny pre zachovanie stability. Inteligentné siete umožňujú novým subjektom na trhu, ako sú agregátory a spoločnosti poskytujúce energetické služby, ponúkať spotrebiteľom nové typy služieb, čo im umožňuje prispôsobiť si spotrebu a využívať výhody flexibility, ktorú poskytuje sieť.

V prípade bežných občanov alebo odvetví náročných na elektrickú energiu budú ich rozhodnutia ovplyvnené zmenami trhových cien. Títo noví hráči budú

hľadať širšiu škálu modelov a riešení, ako sú v súčasnosti k dispozícii. To by malo posilniť hospodársku súťaž na maloobchodnom trhu, prispieť k posilneniu postavenia spotrebiteľa a stimulovať znižovanie emisií skleníkových plynov a zároveň poskytnúť príležitosť pre hospodársky rast. [2] Tento rozvoj inteligentných sietí vo všeobecnosti posúva priemysel dodávok energie smerom orientovaným viac na služby než na samotnú infraštruktúru.

1.2 Chytré domácnosti

Kým veľkí výrobcovia a spotrebitelia elektriny majú vlastných pracovníkov, ktorí zodpovedajú za komunikáciu s ostatnými účastníkmi energetickej sústavy, v prípade domácností túto prácu zaisťuje domáci systém riadenia spotreby alebo výroby energie. Tento systém je nazývaný ako EMS (angl. Energy Management System). EMS komunikuje s distribučnou sieťou a distribútorovi a dodávateľmi poskytuje informácie o spotrebe energie či dostupnosti dodávky. [1]

Kľúčovú roľu v tomto systéme zohráva chytrý elektromer (angl. smart meter), ktorý zaznamenáva aktuálnu spotrebu elektriny a ďalej umožňuje obojstrannú komunikáciu medzi distribútorom a zákazníkom. Vďaka tomu majú obe strany lepší prehľad o spotrebovanej energie zo strany zákazníka. Zároveň to prináša ďalšie výhody ako napríklad možnosť predchádzať možným problémom v distribučnej sieti. Aby bola domácnosť schopná prispôsobovať svoju spotrebu energie aktuálnej situácii, sú do systému pripojené ďalšie elektrické zariadenia. Nad týmito zariadeniami môže mať kontrolu užívateľ napríklad prostredníctvom chytrého telefónu alebo sa môžu spúšťať automaticky. Túto sústavu nazývame internet vecí (angl. Internet of things alebo skrátene IoT), v nej môžu jednotlivé súčasti systému komunikovať a zdieľať informácie. Každé zariadenie môže byť zároveň šikovným senzorom, ktorý umožňuje monitorovať aktuálnu situáciu. Toto by malo prispieť k ešte efektívnejšiemu využívaniu energie. Vďaka inteligentným meračom môžu koncoví zákazníci získať presné a pravidelné informácie z meraní spotreby energie a môžu im byť účtované poplatky za elektrinu, ktorú skutočne používajú. To pomáha predchádzať nesprávnemu vyúčtovaniu, ktoré bolo veľmi časté.

Z digitalizácie a smart technológií však nemusia ťažiť len spotrebitelia, ale aj energetické firmy. Prevádzkovatelia distribučných sietí môžu vďaka ním ľahšie monitorovať celú sústavu a riadiť toky elektriny, ktoré sa vzhľadom k rozvoju decentralizovaných zdrojov stávajú horšie predvídateľnými. Vďaka využitiu chytrých elektromerov sa digitalizuje aj toto odvetvie a napomáha to lepšie spoznávať potreby zákazníkov a dodávateľom ako im vyhovieť. Na jednej strane to umožňuje zvýšiť dôveru zákazníkov, ktorí sa môžu pohybovať v transparentnejšom prostredí

a mať lepšiu predstavu o spôsobe, akým sami využívajú energiu. Na druhej strane vďaka tomu môžu firmy ponúkať spotrebiteľom služby šité na mieru, čo ponúka nové obchodné príležitosti. Zo strany podnikov to ale vyžaduje, aby si osvojili úplne nové spôsoby práce s dátami, ktorých môžu získavať stále väčšie množstvo. Nadobudnuté dáta sú veľmi kritická oblasť a sú veľkým lákadlom pre potencionálnych útočníkov, ktorí by ich mohli využiť vo svoj prospech. Osobné údaje spotrebiteľa sú chránené pravidlami Európskej únie o spracúvaní a voľnom pohybe údajov, a preto EÚ (Európska únia) prijala sériu opatrení na ich ochranu a dodržaní pravidiel. Okrem ochrany údajov a súkromia sa kybernetická bezpečnosť stále častejšie stáva problémom súvisiacim s inteligentnými sieťami a elektromermi. Európska komisia je odhodlaná v tomto smere zmierňovať všetky riziká a zvyšovať odolnosť voči kybernetickým útokom.

1.2.1 Projekty inteligentných sietí v Európe

Zavádzanie inteligentných sietí patrí medzi jednu z troch prioritných oblastí v rámci Transeurópskych energetických sietí, ktorých cieľom je pomôcť integrovať energiu z obnoviteľných zdrojov, dobudovať európsky energetický trh a umožniť spotrebiteľom lepšiu reguláciu spotreby energie.

Projekty inteligentných sietí, ktoré k tomu prispievajú a majú významný vplyv na trhy s energiou najmenej v dvoch krajinách EÚ, sú označované ako PCI (Projects of Common Interest) a sú považované za kľúčové pre implementáciu cezhraničnej energetickej infraštruktúry v EÚ. O toto označenie si jednotlivé projekty, krajiny môžu zažiadať. Projekty inteligentných sietí, ktoré žiadajú o označenie PCI, a aby mohli byť tak aj vyhodnotené nakoniec rozhoduje Smart Grid Regional Group. Zoznam PCI z roku 2019 obsahuje 6 projektov inteligentných sietí:

- **SINCRO.GRID** (Slovinsko, Chorvátsko) - Integrácia synergických, vyspelých technologických riešení s cieľom zvýšiť bezpečnosť prevádzky slovinskej a chorvátskej elektrickej sústavy súčasne.
- **ACON** (Česko, Slovensko) - Hlavným cieľom ACON (Again COnnected Networks) je podpora integrácie českého a slovenského trhu s elektrickou energiou.
- **Smart Border Initiative** (Francúzsko, Nemecko) - Iniciatíva Smart Border spája politiky navrhnuté Francúzskom a Nemeckom s cieľom podporiť ich mestá a územia v ich stratégiách energetickej transformácie a integrácii európskeho trhu.
- **Danube InGrid** (Maďarsko, Slovensko) - projekt posilňuje cezhraničnú koordináciu správy elektrickej siete so zameraním na inteligentnejšie zhromažďovanie a výmenu údajov.

- **Data Bridge** (Estónsko, Lotyšsko, Litva, Dánsko, Fínsko, Francúzsko) - má za cieľ vybudovať spoločnú sieť, ktorá umožní integráciu rôznych typov údajov (údaje inteligentného merania, prevádzkové údaje siete, údaje o trhu, ...) s cieľom vyvinúť škálovateľné a replikovateľné riešenia pre EÚ.
- **Cross-border flexibility project** (Estónsko, Fínsko) - má za cieľ podporovať integráciu obnoviteľných zdrojov energie a zvýšiť bezpečnosť dodávok prostredníctvom cezhraničného poskytovania flexibilných služieb pre Estónsko a Fínsko.

1.2.2 Smart meter

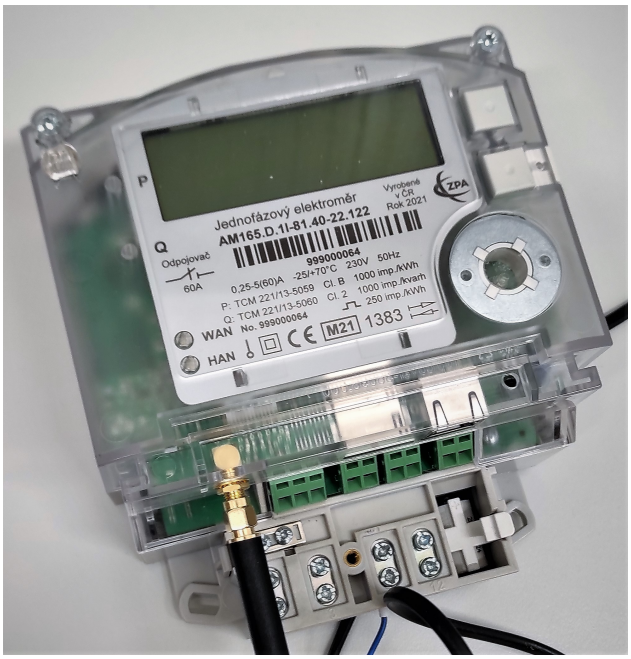
Inteligentné elektromery môžu poskytovať spätnú väzbu o spotrebe energie v reálnom čase a umožňujú spotrebiteľom, ktorí majú záujem, lepšie hospodáriť s jej využitím, šetriť energiou a znížiť svoje účty.

Spotrebiteľom, ktorí chcú byť na trhu s elektrickou energiou aktívnejší, či už sami alebo s pomocou servisnej spoločnosti, môžu inteligentné elektromery využiť ešte viac a môžu im ponúknuť široké spektrum nových možností. Umožňujú im prispôsobiť svoju spotrebu energie rôznym cenám energie počas celého dňa, čo im umožňuje spotrebovať viac počas nižších cenových období a ušetriť tým ďalšie peniaze. Inteligentné elektromery sú dôležité aj pre tých, ktorí vyrábajú elektrickú energiu, napríklad zo solárneho panelu nainštalovaného na streche alebo pomocou vlastnej vodnej elektrárne. Pomocou inteligentného merača môžu zmerať elektrickú energiu svojich domácich dodávok do siete a oznámiť túto dodávku manažérovi siete.[2] V dôsledku toho operátori sietí získavajú lepší prehľad o tom, čo sa deje u spotrebiteľov. Takto môžu lepšie plánovať svoje investície a spravovať svoju infraštruktúru tak, aby reagovala na požiadavky ich zákazníkov. Príklad ako môže taký smart meter vyzerať je na obrázku číslo 1.1.

Aby inteligentné merače fungovali na správne a slúžili tak ako majú vo všetkých smeroch, musia byť vybavené správnymi funkciami. Tie sú uvedené v smernici o elektrickej energii (EÚ)2019/944.

1.3 Kybernetická bezpečnosť kritickej infraštruktúry

Digitalizácia v kritickej infraštruktúre prináša značné riziká. Všetky senzory a všetko počítačové vybavenie pripojené na sieť sa môže stať cieľom pre potencionálnych útočníkov a môžu sa stať obeťou rôznych kybernetických incidentov. Tieto incidenty môžu potencionálne ohroziť bezpečnosť dodávok energie alebo súkromie spotrebiteľských údajov. Pod kybernetickým útokom sa v energetike rozumie



Obr. 1.1: Smart meter

situácia, keď sa hackeri pokúsi získať prístup ku kľúčovým informáciám alebo častiam infraštruktúry, ako sú elektrárne, rozvodné siete alebo riadiace centrá.

Cieľom je narušiť ich funkciu alebo ich úplne ovládnuť, a útočníkom môže byť jednotlivec alebo celá skupina ľudí. S podobnými útokmi sme sa už mohli stretnúť aj napríklad v neďalekej Ukrajine kde po útoku hackerov zostalo až 700 000 ľudí niekoľko hodín bez elektriky. Všeobecne platí, že firmy by mali dáta získané od zákazníkov anonymizovať a agregovať do väčších celkov, aby sa hackerom nemohli do rúk dostať informácie o konkrétnych ľuďoch.

Kybernetická bezpečnosť a výzvy s ňou súvisiace sa vyvíjajú rýchlym tempom, a preto Európska komisia prijala sériu opatrení na jej riešenie. To o čo sa Európska komisia pokúša je to, aby vytvorili komplexný legislatívny rámec. Tento rámec stavia na troch základných pilieroch z legislatívy a tie sú: [3]

- stratégia EÚ pre kybernetickú bezpečnosť (JOIN(2013)01 final), [4]
- smernica o bezpečnosti sieťových a informačných systémov (smernica NIS) (EÚ)2016/1148, [5]
- balík kybernetickej bezpečnosti (JOIN(2017)450 final) zo septembra 2017, ktorý zahŕňa aj zákon o kybernetickej bezpečnosti. [6]

Napriek tomu, že existuje komplexný právny rámec pre kybernetickú bezpečnosť, energetický sektor, a teda primárne kritická infraštruktúra si vyžaduje osobitnú pozornosť pretože čelí výzvam, ktoré nie sú pre kybernetickú bezpečnosť typické.

Tu napríklad patrí aj spracovávanie požiadavkov v reálnom čase, niektoré energetické systémy musia reagovať tak rýchlo, že štandardné bezpečnostné opatrenia, medzi ktoré napríklad patrí autentifikácia príkazu alebo overenie digitálneho podpisu, jednoducho nie je možné použiť kvôli oneskoreniu, ktoré tieto opatrenia spôsobujú a vyžadujú.

Ďalej musíme byť obozretný aj pri systémoch, ktoré sú veľmi silno prepojené či už v jednotlivých štátoch ale aj v rámci Európy alebo sveta. Výpadok v jednej krajine môže viesť k výpadkom prúdu alebo nedostatku dodávok v iných oblastiach a krajinách.

Medzi poslednú výzvu by som zaradil implementáciu nových technológií do starších systémov, ktoré sú v prevádzke. Mnoho prvkov energetického systému bolo navrhnutých a vybudovaných dostatočne dlho predtým, ako sa vôbec dostali do úvahy otázky ohľadom kybernetickej bezpečnosti. A práve tieto systémy potrebujú integrovať najmodernejšie zariadenia na automatizáciu a riadenie do svojich systémov tak aby neboli vystavené kybernetickým hrozbám. Medzi tieto zariadenia patria napríklad inteligentné merače alebo pripojené spotrebiče, a zariadeniami z „internetu vecí“.

Na riešenie vyššie spomenutých problémov Európsky parlament vytvoril Nariadenie Európskeho parlamentu a Rady (EÚ)2019/941 z 5. júna 2019 o pripravenosti na riziká v odvetví elektrickej energie, ktoré ukladá krajinám EÚ povinnosť zahrnúť opatrenia o kybernetickej bezpečnosti do ich národných plánov hodnotenia rizík a ďalej pojednáva aj o zrušení starej smernice 2005/89/ES. Zatiaľ čo Nariadenie Európskeho parlamentu a Rady (EÚ)2019/943 z roku 2019 sa od komisie požaduje, aby vypracovala sieťový kódex o kybernetickej bezpečnosti cezhraničných tokov elektriny. Týmto kódexom sa stanovujú prioritné zoznamy pre rozvoj sieťových predpisov a usmernení pre elektrickú energiu na určité obdobie. Medzi odporúčané oblasti, ktorým sa je potreba venovať v najbližšom období patria podľa kódexu tieto veci:

- cezhraničné hodnotenie a riadenie kybernetického rizika,
- Certifikácia ISO/IEC 27001 alebo dôkaz o rovnocennosti,
- spoločné funkčné a nefunkčné bezpečnostné kontroly a požiadavky,
- systém zabezpečenia a zdieľanie informácií. [3]

2 Komunikačné protokoly v energetike

V tejto kapitole sa budeme venovať komunikačnými protokolmi, ktoré sa bežne používajú v energetických sieťach. Je tu spomenutá krátka charakteristika a základné predstavenie protokolov. Medzi protokoly pre chytré siete sa radia tieto protokoly:

- DLMS/COSEM,
- IEC 61850,
- IEC 60870-5-104,
- Z-wave,
- ZigBee,
- 6LoWPAN.

2.1 DLMS/COSEM

Tomuto protokolu sa budeme, viac venovať v nasledujúcej kapitole a teda tu uvedieme len stručný prehľad. DLMS/COSEM je celosvetový štandard, ktorý slúži ako komunikačný protokol pre smart meters, ktoré slúžia na meranie elektriny, plynu, vody, ... Definuje objektovo orientovaný dátový model, aplikačný protokol a komunikačné profily špecifické pre používané médiá. DLMS/COSEM zahŕňa tri kľúčové komponenty: DLMS (Device Language Message Specification), COSEM (Companion Specification for Energy Metering), OBIS (Object Identification System).

DLMS je protokol aplikačnej vrstvy, ktorý mení informácie uchovávané v objektoch na správy. Táto vrstva reguluje diaľkové odčítanie nameraných hodnôt z meracích zariadení a ich vzdialené ovládanie a tiež ďalšie služby pre meranie akéhokoľvek typu energie.

COSEM je objektový model rozhrania komunikačného zariadenia pre merania akéhokoľvek typu energie. Je to špecifikácia, ktorá poskytuje reprezentáciu funkčnosti meracích zariadení. Model rozhrania používa objektovo orientovaný prístup.

OBIS predstavuje systém, ktorý definuje pomenovanie objektov. OBIS definuje identifikačné kódy (ID), čím poskytuje jedinečný identifikátor pre všetky dáta v meranom systéme. Tieto identifikačné kódy sa používajú pre bežné dátové položky v zariadeniach na meranie energií.

V komunikácii DLMS/COSEM má každá strana, ktorá komunikuje priradenú svoju vlastnú adresu. Adresa klienta ja podľa definície protokolu bajtová hodnota. Hodnota adresy klienta určuje aj skutočnú povahu klienta. Môžeme napríklad uviesť

prípad kde norma uvádza, že klient s hodnotou adresy 16 je verejný klient. Môže ale existovať aj iný druh klientov: systém zberu údajov, výrobca, spotrebiteľ, ... Adresa sa skladá z adresy fyzického zariadenia a adresy logického zariadenia. [7]

2.2 IEC 61850

Súbor noriem IEC 61850 špecifikuje metódy komunikácie a komunikačné protokoly pre oblasť energetiky a elektrických sietí. Podľa IEC 61850 je možné vytvárať flexibilné komunikačné systémy, ktoré vyhovujú súčasným požiadavkám energetického priemyslu, ale rovnako dobre budú schopné vyhovieť aj požiadavkám budúcim.

Súbor noriem IEC 61850 určuje pravidlá pre komunikáciu medzi zariadeniami v rozvodniach a stanovuje požiadavky, ktoré sú na rozvodné siete a zariadenia v nich kladené. Táto norma obsahuje definície komunikačných protokolov ale aj štandardy pre riadiace funkcie použité v nej. IEC 61850 zaisťuje vzájomnú súčinnosť zariadení a systémov v rozvodniach tým, že štandardizuje ich rozhrania, protokoly a dátové modely. Týmto spôsobom sú rozvodne schopné veľmi efektívne znížiť náklady na integráciu zariadení. Veľkou výhodou tohoto súboru noriem je aj to, že nešpecifikuje len protokoly pre komunikáciu medzi zariadeniami vnútri rozvodne, ale aj medzi rozvodňami a prípadne ďalšími členmi elektrickej siete. Umožňuje prenášať dôležité informácie z jednej rozvodne do druhej, čo je obzvlášť dôležité pri výpadku elektrizačnej siete. Ďalej dovoľuje integrovať zariadenia a podsystémy v elektrizačnej sieti a vytvoriť z nich jednotný a úplný riadiaci a komunikačný systém, čo umožňuje jednoduché riadenie technických a ekonomických procesov. [8]

2.2.1 Priebeh komunikácie

Komunikácia prebieha formou publish/subscribe. Dáta od klientov idú do určitého serveru a ten následne rozosiela dáta tým, čo si dáta objednali. Tieto dáta môžu následne cestovať aj režimom multicast. Server posiela dáta všetkým účastníkom siete, ale dáta čítajú iba tí, čo si ich objednali (subscribers).

Architektúra komunikačného systému podľa IEC 61850 je taktiež typu klient/server, ale odstraňuje nevýhody klasickej architektúry klient/server tým, že umožňuje aj klientskym staniciam, aby riadili prenos dát. To dovoľuje presunúť riadiace a komunikačné funkcie bližšie prevádzkovým procesom a prináša do sietí s touto architektúrou veľkú komunikačnú flexibilitu.

Medzi ďalšiu výhodu by sme zaradili aj to, že ku špecifikovaným dátovým modelom serverov, ktoré poskytujú dáta do komunikačnej siete, dáva IEC 61580

prednosť objektovo orientovanému programovania, tak ako je to aj v prípade DLMS/COSEM. Opäť teda platia rovnaké výhody, ktoré patria k objektovo orientovanému programovaniu. Dáta sa posielajú v objektoch. Tieto objekty spájajú dáta a programy, takže všetky informácie a funkcie sa nachádzajú na jednom mieste. Vďaka tomu je pre používateľov aj jednotlivé zariadenia oveľa ľahší prístup k dátam a funkciám s nimi spojenými.

2.3 IEC 60870-5-104

Protokol IEC 60870-5 (známy aj ako IEC 104) je súčasťou normy IEC Telecontrol Equipment and Systems Standard IEC 60870-5, ktorá poskytuje komunikačný profil na odosielanie základných správ diaľkového ovládania medzi dvoma systémami v elektrotechnike a automatizácii energetických systémov. Poskytuje komunikačné profil pre zasielanie základných riadiace správ medzi stále pripojenými systémami. Protokol je definovaný v piatich dokumentoch s označením IEC 60870-5-1 až IEC 60870-5-5. [9]

IEC 60870-5-104 umožňuje komunikáciu medzi riadiacou stanicou a rozvodňou prostredníctvom štandardného modelu siete TCP/IP. Na bezpečný prenos sa používa spojovo orientovaný protokol TCP. IEC 60870-5-104 je rozšírením IEC 60870-5-101 o niekoľko funkcií transportortného protokolu TCP/IP. [10] Jedná sa teda o ďalšie nadstavbu protokolu. Možno teda využívať takmer všetku funkcie známeho protokolu TCP/IP. Najväčšou výhodou normy IEC 60870-5-104 je, že umožňuje komunikáciu prostredníctvom štandardnej siete, ktorá umožňuje simultánny prenos údajov medzi niekoľkými zariadeniami a službami.

2.3.1 Priebeh komunikácie

Komunikácia v protokoloch vychádzajúcich z IEC 60870-5-104 funguje na modeli master/slave. To znamená, že jedno zariadenie (master), pri komunikácii riadi prenos dát a pracuje s nimi a ďalšie zariadenie (slave) dáta poskytuje a pracuje tak ako master určí. Toto sa vykonáva predovšetkým pri komunikácií na zbernici. Master tieto požiadavky na ďalšie zariadenia rozosiela postupne, nie naraz. Každé zariadenie slave reaguje len na dáta, ktorá mu sú určené, ostatné ignoruje.

Tento typ komunikácie má však jednu veľkú nevýhodu, pretože dáta na zbernici môže odpočúvať akékoľvek pripojené zariadenia, čo predstavuje veľké bezpečnostné riziko. Pri komunikácii iba medzi jednotlivými strojmi, však na toto bezpečnostné riziko nemusíme brať príliš veľký ohľad. V rozvodniach je táto bezpečnosť dôležitejšie. Na zlepšenie bezpečnosti slúži aj jedna z implementovaných

funkcií tohoto protokolu. Jednou z takých funkcií je to, že slave môže obsahovať kritické informácie, ktoré je potrebné spracovávať prednostne a tak tieto informácie majú vyššiu prioritu a môžu byť doručené master zariadeniu bez toho aby si o ne vôbec požiadal. V energetických sústavách spravidla sústavu riadi počítač. Ten riadi zvyšné jednotky, ktoré sa starajú o chod elektrární a rozvodní.

Podľa protokolu je každá stanica buď riadiacou stanicou alebo riadenou stanicou. Komunikácia IEC 101/104 sa vymieňa medzi riadenou a riadiacou stanicou. Riadená stanica je monitorovaná alebo riadená hlavnou stanicou (RTU). Riadiaca stanica je stanica, kde sa vykonáva kontrola nadstavieb (SCADA). Typicky sa jedná o PC so systémom SCADA.

2.4 Z-Wave

Z-Wave je jednoduchý štandard pre bezdrôtovú komunikáciu, ktorý umožňuje vzájomnú komunikáciu zariadení domácej siete, ktorú môže tvoriť napríklad niekoľko inteligentných senzorov a meračov.

Protokol Z-Wave je založený na bezdrôtovej technológii, ktorá používa rádiovú frekvenciu. Protokol je primárne určený najmä na ovládanie, monitorovanie a čítanie stavu aplikácií v domácnosti, ktoré spoločne tvoria IoT (Internet of things). Z-Wave podporuje topológiu siete kde komunikuje každý s každým teda tzv. full mesh. To umožňuje mnohým zariadeniam komunikovať medzi sebou súčasne. Z-Wave umožňuje bezpečnú komunikáciu s nízkou spotrebou energie medzi schválenými Z-Wave zariadeniami. Protokol je štandardizovaný a je možné pomocou neho prepojiť zariadenia od rôznych výrobcov. Vďaka Z-Wave je možné pomocou jednej centrálnej jednotky ovládať všetky zariadenia celej domácnosti. Centrálnou jednotkou pritom môže byť počítač alebo aj tablet alebo chytrý telefón.

Medzi hlavné výhody protokolu patrí:

- spoľahlivá a bezpečná bezdrôtová komunikácia,
- kompatibilita zariadení mnohých rôznych značiek,
- jednoduchá inštalácia,
- minimálna spotreba elektrickej energie,
- ovládateľnosť na diaľku cez internet aj lokálne z domu,
- nízke obstarávacie náklady a ľahká rozšíriteľnosť. [11]

Vzorový model zariadení, s ktorými môže protokol komunikovať môžeme vidieť na obrázku číslo 2.1, tieto zariadenia tak spolu tvoria chytrú sieť prostredníctvom zariadení IoT.

Obr. 2.1: Spojenie IoT

2.4.1 Priebeh komunikácie

Na komunikáciu sa používa šírka pásma v rozmedzí 800 - 900 MHz v závislosti od kontinentu. V Európe využíva Z-Wave bezdrôtový prenos na frekvencii 868,42 MHz. Vďaka tejto hodnote je rezistentný voči rušeniu od WiFi signálov na typickýcj frekvenciách 2,4 aj 5 GHz. Bežný dosah riadiacej jednotky je približne 100 metrov vo vonkajšom priestore a 50 m vo vnútri budovy. Keďže je však využívaná topológia siete typu full mesh, jednotlivé zariadenia si odovzdávajú signál aj medzi sebou a tým efektívne dosah navyšujú. Z tejto úvahy môžeme dedukovať aj to, že čím viac máme zariadenie, tým väčší máme dosah. V sieti sa nachádza jeden centrálny prvok na ovládanie, čo môže pri komunikácii s koncovými zariadeniami prinášať určité spomalenie. V porovnaní s mobilnými telefónmi majú produkty Z-Wave 4000 krát nižšie vyžarovanie takže škodlivého vplyvu na zdravie sa netreba báť.

2.5 ZigBee

ZigBee je protokol pre bezdrôtové siete vyvinutý ako otvorený globálny štandard na riešenie jedinečných potrieb nízko nákladových a nízko energetických bezdrôtových sietí IoT. Štandard Zigbee funguje na fyzickej rádiovej špecifikácii IEEE 802.15.4 a funguje v nelicencovaných pásmach 900 MHz a 868 MHz vrátane 2,4 GHz. Špecifikácia 802.15.4, na ktorej Zigbee stack funguje, bola schválená Inštitútom elektrických a elektronických inžinierov (IEEE) v roku 2003. Táto špecifikácia je paketovo orientovaný rádiový protokol určený primárne pre lacné zariadenia napájané z batérie. Protokol umožňuje zariadeniam komunikovať v rôznych sieťových topológiách. [12]

Bežný dosah riadiacej jednotky je 100 metrov vo vonkajšom priestore a 50 m vo vnútri budovy. Keďže však väčšinou využíva topológiu siete typu mesh, jednotlivé zariadenia si odovzdávajú signál aj medzi sebou a tým efektívne dosah navyšujú – čím viac máte zariadenie, tým väčší máte dosah. Protokol Zigbee 3.0 je určený na komunikáciu údajov cez rušivé RF (Radiofrequency) prostredia, ktoré sú bežné v komerčných a priemyselných aplikáciách. Tento štandard dovoľuje všetkým zariadeniam aby boli pripojené do jednej siete bez ohľadu na ich trhové označenie a funkciu.

Medzi hlavné výhody tohoto protokolu patrí:

- podpora viacerých sieťových topológii, ako sú point-to-point, siete typu point-to-multipoint a mesh,
- nízky pracovný cyklus, ktorý poskytuje dlhú životnosť batérie,
- krátka odozva,
- šifrovanie cez AES-128 pre bezpečné dátové spojenie,
- kooperácia zariadení rôznych značiek (kompatibilita musí byť povolená zo strany výrobcov konkrétnych produktov),
- minimálna spotreba elektrickej energie.

2.6 6LoWPAN

6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks) je otvorený štandard definovaný organizáciou Internet Engineering Task Force (IETF) v ich dokumente RFC 6282. Je to nízko-energetická bezdrôtová sieť typu mesh, kde každý uzol má svoju vlastnú IPv6 adresu. To umožňuje uzlu pripojiť sa priamo k internetu pomocou otvorených štandardov. Tento protokol spadá pod rovnakú normu ako protokol ZigBee ale každý z nich má vlastnú nadstavbu. Technológia 6LoWPAN využíva IEEE 802.15.4 na zabezpečenie nižších vrstiev pre svoj systém. Ďalej používa zabezpečenie spojovacej vrstvy algoritmom AES-128, ktoré je definované tiež v už spomínanom IEEE 802.15.4. To má na starosti autentifikáciu a šifrovanie. Ďalšiu bezpečnosť poskytujú už jednotlivé bezpečnostné mechanizmy transportnej vrstvy. [13]

Celkový systém je zameraný na poskytovanie bezdrôtového internetového pripojenia pri nízkom počte prenášaných dát a s nízkym pracovným cyklom. Existuje veľa možnosti aplikácie protokolu, kde nájde svoje využitie. Môžeme spomenúť napríklad oblasť automatizácie, priemyselný monitoring kde je možné aj pripojenie na cloud pre neskoršiu analýzu a nakoniec smart home alebo inteligentné siete v spojení s inteligentnými meračmi. Protokol je primárne navrhnutý pre siete, kde rýchlosť prenosu dát je relevantná ale primárna je spoľahlivosť prenosu dát.

3 DLMS/COSEM

V tejto kapitole bude bližšie teoreticky popísaný komunikačný protokol využívaný v energetickým sieťach známy pod pojmom DLMS/COSEM. Komunikačný protokol DLMS (Device Language Message Specification) je súborom noriem vyvinutých a udržiavaných spoločnosťou DLMS UA (DLMS User Association). DLMS User Association udržiava spojenie typu D s IEC TC13 WG14 zodpovedné za medzinárodné štandardy pre výmenu údajov z meračov a za zavedenie radu IEC 62056. V tejto úlohe poskytuje DLMS UA služby certifikácie údržby, registrácie a zhody pre IEC 62056 DLMS/COSEM. [14]

3.1 Prehľad

Komunikačný protokol DLMS sa často v literatúre objavuje v spojení DLMS/COSEM. DLMS a COSEM sú dve rozdielne veci ale zásadne sa používajú spolu. DLMS/COSEM používa DLMS komunikačný protokol a COSEM rozhranie, ktoré definuje triedy v aplikačnej a transportnej vrstvi protokolu DLMS. DLMS protokol má vlastnú radu noriem iba k využitiu v energetike, táto rada sa nazýva IEC 62056. DLMS UA definuje protokoly do súboru štyroch dokumentov špecifikácie. Tieto dokumenty tvoria komplexný rámec pre porozumenie protokolu. Patrí tu zelená knihy (Green book), žltá kniha (Yellow book), modrá kniha (Blue book) a biela kniha (White book). Modrá kniha opisuje model objektu merača COSEM a systém identifikácie objektov OBIS, zelená kniha popisuje architektúru a protokoly, žltá kniha spracováva všetky otázky týkajúce sa testovania zhody a biela kniha obsahuje slovník pojmov a definícii. Ak výrobok prejde testom zhody uvedeným v žltej knihe, tak následne DLMS UA vydá certifikáciu zhody DLMS/COSEM.

DLMS/COSEM pozostáva z troch kľúčových komponentov:

- COSEM - objektový model schopný popísať prakticky akúkoľvek aplikáciu,
- OBIS - systém pomenovania objektov,
- DLMS - protokol aplikačnej vrstvy, ktorý premieňa informácie uchovávané objektmi na správy.

DLMS/COSEM je možné použiť pre všetky druhy verejných služieb a energetiky, všetky trhové segmenty, všetky aplikácie a takmer všetky komunikačné médiá. [15]

3.1.1 COSEM

Objektový model COSEM popisuje sémantiku jazyka. Triedy tohoto rozhrania, a ich objekty, môžeme jednoducho použiť na modelovanie možných prípadov použitia

pre energetický manažment. To zahrnuje napríklad rôzne merania, obecne je tento model dostatočne všeobecný na modelovanie akejkoľvek aplikácie. Pre tento model platia rovnaké zásady ako pre akékoľvek objektovo orientované programovanie. Medzi základné vlastnosti patrí aj to, že každý aspekt údajov je modelovaný pomocou atribútu a aj to, že objekty môžu mať niekoľko atribútov a tiež metódy na vykonávanie operácií s atribútmi. Objekty v modeli COSEM je možné použiť v kombináciách na modelovanie jednoduchých prípadov použitia, ako je čítanie registrov, alebo zložitejších, ako sú schémy fakturácie alebo správa zaťaženia. Dnes je uvedených 89 tried rozhraní, ale je možné ich mať až 65 535.

3.1.2 OBIS

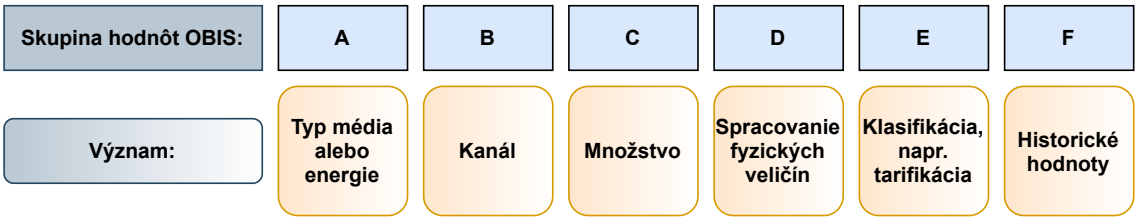
OBIS je systém, v ktorom sa pomenúvajú objekty z modelu COSEM. Kódy OBIS sú špecifikované pre meranie elektrickej energie, plynu, vody, rozdeľovačov nákladov na teplo a tepelnej energie, ako aj pre abstraktné údaje, ktoré nesúvisia s meraným druhom energie. [15] K dispozícii je 281 474 miliárd kódov OBIS, z ktorých 4 398 miliárd je vyhradených na účely normalizácie. Zvyšok je možné použiť na konkrétne účely výrobcu, krajiny alebo akéhokoľvek ekonomického združenia objektov.

Zoznam štandardných kódov OBIS a objektov COSEM pravidelne udržiava DLMS User Association (DLMS UA) a je voľne dostupný na portáli DLMS. DLMS UA je nezisková organizácia, do ktorej sú zapojené energetické spoločnosti, výrobcovia meračov, vývojári a integrátori systémov a výrobcovia čipov. V tabuľke číslo 3.1 sú uvedené kódy OBIS zopár bežných objektov triedy A, ktoré sa nachádzajú na bežných zariadeniach DLMS. Táto tabuľka slúži ako ukážka niektorých typov kódu.

Tab. 3.1: OBIS kódy

Kódy OBIS bežných objektov	**Popis**
0.0.42.0.0.255	Názov logického zariadenia COSEM
0.0.1.0.0.255	Hodiny
0.0.25.0.0.255	Nastavenie TCP
0.0.25.1.0.255	Nastavenie IPv4
0.0.25.2.0.255	Nastavenie MAC
0.0.96.1.0.255	Výrobné číslo
0.0.99.12.0.255	Profil pripojenia
0.1.2.0.0.255	Konfigurácia modemu

Spôsob akým funguje identifikačný systém OBIS kódov môžeme vidieť na obrázku číslo 3.1.



Obr. 3.1: Identifikačný systém OBIS kódov

3.1.3 DLMS

Službami DLMS je určená syntax jazyka. DLMS/COSEM používa systém klient-server. Kde ako koncové zariadenia ako sú inteligentné merače vystupujú ako server a hlavné koncové systémy vystupujú ako klient. Aplikačná vrstva DLMS/COSEM poskytuje služby ACSE na pripojenie klientov a serverov a služby xDLMS na prístup k údajom uchovávaným objektmi COSEM. Služby xDLMS sú pre každý objekt rovnaké, to umožňuje pridanie nových objektov do modelu bez ovplyvnenia aplikačnej vrstvy. Aplikačná vrstva taktiež zostavuje správy, označované ako APDU (Apllication Protocol Data Units) a ďalej podľa potreby aplikuje, kontroluje a odstraňuje kryptografickú ochranu a riadi prenos dlhých správ v blokoch. [15] Správy je potom možné prenášať prakticky cez akékoľvek komunikačné médium. K dispozícii sú rôzne vstavané mechanizmy na optimalizáciu prenosu podľa charakteristík daných médií. DLMS protokol sa môže použiť ako v energetike tak aj v bežnom domácom prostredí. Má však jednu nevýhodu, že treba udržiavať stále a stabilné spojenie komunikácie bez prerušenia. Ďalšou nevýhodou je, ze DLMS pracuje ako nadstavba UDP/IP, pretože potrebuje väčšiu šírku pásma pre komunikáciu.

3.2 Výmena informácii v DLMS/COSEM

Cieľom DLMS/COSEM je špecifikovať štandard pre objektový model rozhrania orientovaného na obchodnú doménu pre meracie zariadenia a systémy, ako aj služby pre prístup k objektom. Špecifikované sú aj komunikačné profily na prenos správ prostredníctvom rôznych komunikačných médií. DLMS/COSEM používa koncepty modelu Open Systems Interconnection (OSI) na modelovanie výmeny informácií medzi meračmi a systémami zberu údajov. [16] Dáta pri komunikácii v DLMS

sú roztriedené do tried. Dáta sú roztriedené podľa hodnôt alebo informácií, ktoré obsahujú. Triedy obsahujú objekty. Objekt je kolekcia atribútu a metód a atribúty obsahujú samotné dáta. Kľúčové charakteristiky, ktoré musia platiť pri výmene informácií pomocou DLMS/COSEM sú tieto:

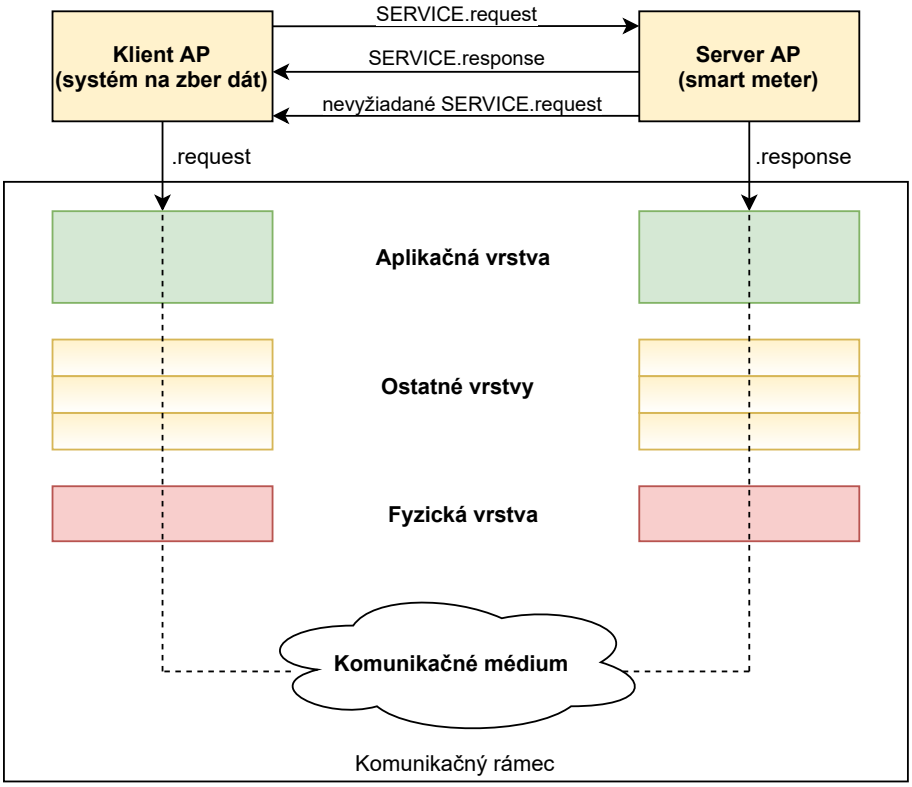
- k meracím zariadeniam majú prístup rôzne strany, nie len klienti,
- sú poskytnuté mechanizmy na kontrolu prístupu k zdrojom chytrých meračov,
- bezpečnosť a súkromie je zaistené použitím kryptografickej ochrany na správy xDLMS a na údaje COSEM,
- v mieste merania môže byť jedno alebo viac chytrých meracích zariadení, hlavne v prípade viacerých meraných veličín. Ak by na mieste bolo viacero chytrých meračov stačí vytvoriť iba jeden prístupový bod,
- výmena údajov môže prebiehať buď diaľkovo alebo lokálne. V závislosti od možností meracieho zariadenia je možné vykonávať lokálnu a vzdialenú výmenu údajov súčasne bez toho, aby sa navzájom rušili,
- v miestnych sieťach (LN - Local Networks), susedských sieťach (NN - Neighbour Network) a rozsiahlych sieťach (WAN - Wide Area Network) je možné použiť rôzne typy komunikačných médií.

Aplikačné funkcie meracích zariadení a systémov zberu dát sú modelované aplikačnými procesmi (AP). Komunikácia medzi AP je modelovaná komunikáciou medzi aplikačnými entitami (AE). Aplikačná entita predstavuje komunikačné funkcie aplikačných procesov. V AP môže existovať niekoľko sád komunikačných funkcií modelu OSI, takže jeden AP môže byť reprezentovaný niekoľkými AE. Každý AE však predstavuje iba jeden AP.

Ako už bolo spomínané vyššie tak výmena údajov medzi systémami zberu údajov a meracími zariadeniami je založená na modeli klient/server, kde systémy zberu údajov zohrávajú úlohu klienta a meracie zariadenia zohrávajú úlohu servera. Klient odosiela servisné požiadavky na server, ktorý odosiela servisné odpovede. Server k tomu ešte môže navyše iniciovať nevyžiadané servisné požiadavky, aby klienta informoval o udalostiach alebo aby odoslal údaje o vopred nakonfigurovaných podmienkach.

Komunikácia klienta so serverom spravidla prebieha distančnou formou pretože dva prístupové body sú umiestnené na dvoch rozličných miestach. Výmena správ sa preto uskutočňuje prostredníctvom súboru protokolov, ako je to znázornené na obrázku číslo 3.2.

Pred samotnou výmenou správ musí dôjsť k naviazaniu spojenia a následne po výmene správ aj k ukončeniu spojenia. Tvorba a priebeh tejto relácie s použitím HLS je zobrazená na obrázku číslo 3.3. Komunikácia protokolom DLMS začína poslaním správy od klienta s príznakom AARQ (Association Request) a následnou odpoveďou

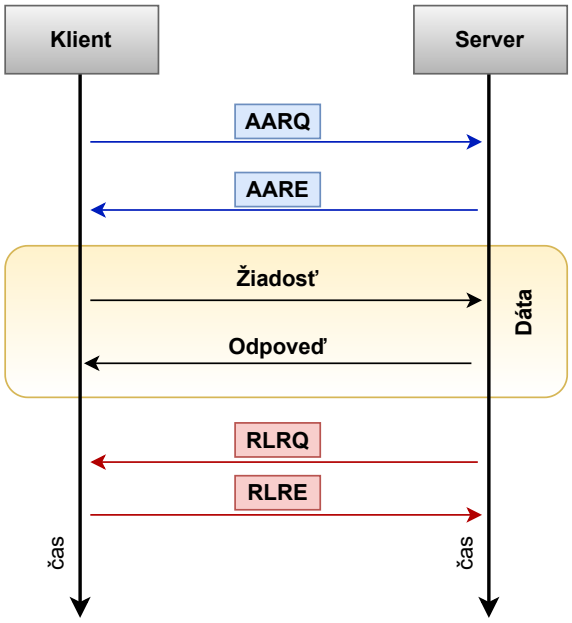


Obr. 3.2: Model klient-server a komunikačné protokoly

serveru s príznakom AARE (Association Response). K ukončeniu spojenia dochádza opäť podľa špeciálnych správ DLMS/COSEM s príznakom AARL (Association Release) a AARE (Association Response).

3.2.1 Autentifikácia

Autentifikácia je jednou z hlavných bezpečnostných vlastností komunikačného systému. Autentifikácia je proces overenia identity subjektu. Po dokončení autentifikácie obvykle nasleduje autorizácie, čo je súhlas, schválenie, umožnenie prístupu či prevedenie konkrétnej operácie daným subjektom. V komunikačných systémoch je autentifikácia entity zásadne dôležitou bezpečnostnou službou. Cieľom autentifikácie entity je zistiť, či je ten kto vystupuje v komunikácii skutočne tým, za koho sa hlási. Aby sa dosiahol tento cieľ, mal by existovať už existujúci vzťah, ktorý spája entitu s tajomstvom. V DLMS/COSEM prebieha autentifikácia počas zakladania logického spojenia medzi aplikačnou entitou klienta a servera. V potvrdených spojeniach sa môže autentifikovať buď klient (jednostranná autentifikácia), alebo klient aj server (vzájomná autentifikácia oboch strán).



Obr. 3.3: Relácia spojenia DLMS/COSEM s HLS

3.2.2 Komunikácia

Komunikácia DLMS protokolu funguje na hierarchickom princípe. DLMS protokol tvorí aplikačnú a niekoľko nižších vrstiev. Užívateľ má priamy prístup k objektom čiže triedam, takže aj k samotným dátam. K týmto dátam sa dostava pomocou ich názvu alebo samotného kódu dát. Užívateľ zadáva aj parameter čo sa s dátami bude diať. Tento parameter bude spracovaný podľa príslušnej triedy, do ktorej patrí a následne bude odovzdaný do aplikačnej vrstvy. Aplikačná vrstva vloží výstup z vrstvy do samotného zariadenia, ktorému dáta patria a celé inštrukcie, čo sa ma diať predá ako informáciu do nižších vrstiev, kde sa dáta už spracovávajú v jednotlivých zariadeniach. Klient a server medzi sebou môžu komunikovať na všetkých vrstvách. Je však potreba najskôr nadviazať tzv. „asociácie“. Asociácia sa vykonáva pri vytváraní spojenia v aplikačnej vrstve a pri asociácii si klient a server stanovuje niektoré komunikačné parametre, ktoré pri celej komunikácii musíme dodržiavať. Po uskutočnení spojenia sa následne môže klient pýtať serveru priamo na dáta v ňom a naopak. [17]

3.2.3 Komunikačný rámec DLMS/COSEM

DLMS/COSEM (IEC 62056-53, IEC 62056-62) je štandardná špecifikácia, ktorá používa COSEM pre zariadenia na modelovanie rozhrania a DLMS na výmenu údajov o tomto meracom zariadení. Obsahuje objektový model, protokol aplikačnej

vrstvy a komunikačné profily na prenos správ. V referenčnom modeli ISO/OSI DLMS komunikuje cez L4-L5 (transportná a relačná vrstva) a COSEM tvorí prezentačnú vrstvu (L6), viac viz v tabuľke č. 3.2.

Tab. 3.2: DLMS a referenčný ISO/OSI model

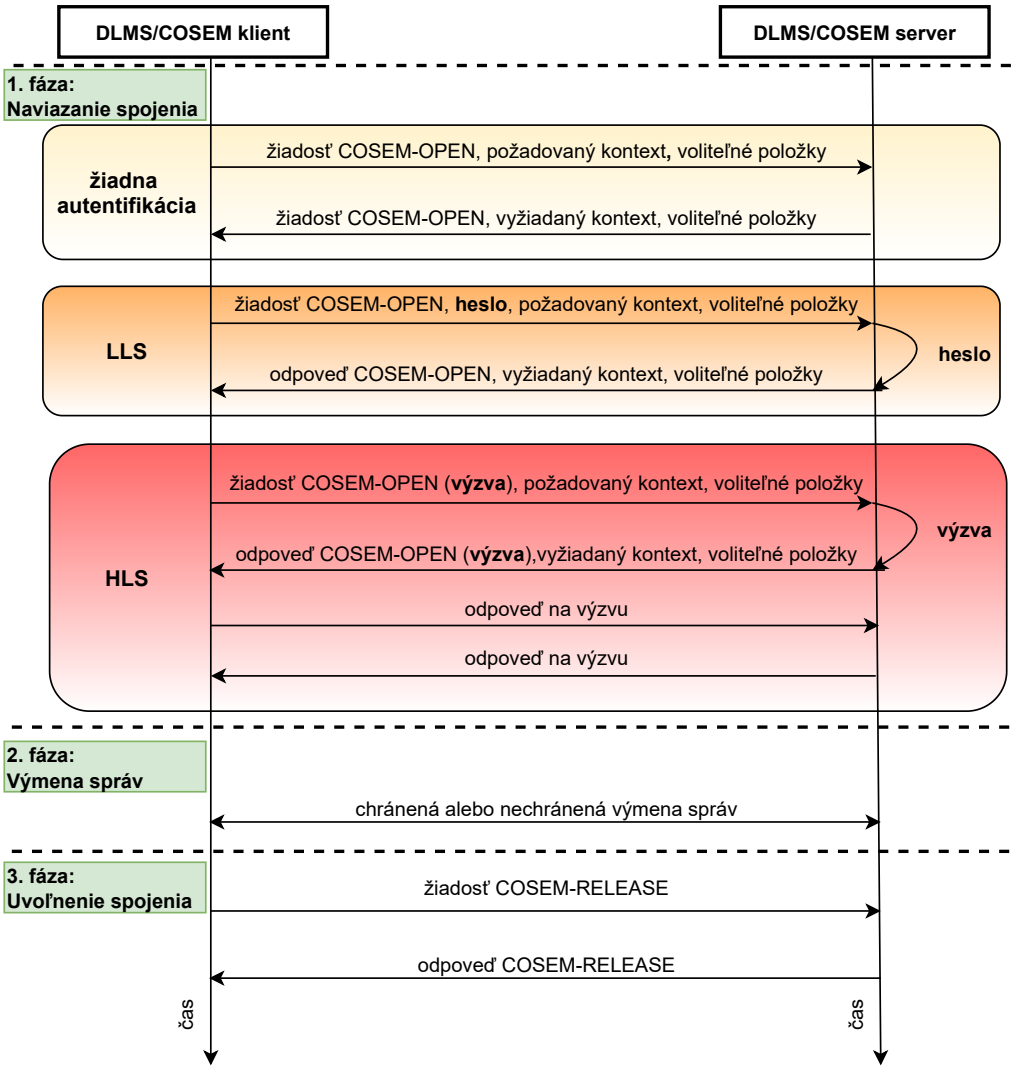
Vrstva	**Funkcia**	**DLMS/COSEM**
Aplikačná	Sieťový proces do aplikácie	Aplikačná
Prezentačná	Reprezentácia údajov, šifrovanie a dešifrovanie	COSEM
Relačná	Komunikácia klientov, správa relácií medzi aplikácií	DLMS
Transportná	Spojenia typu end-to-end, spoľahlivosť a riadenie toku	DLMS
Sieťová	Určenie cesty a logické adresovanie	DLMS
Spojová	Fyzické adresovanie	HDLC,IEC 60256-47
Fyzická	Prenos média, signálu a binárny prenos	Sériové médiá, kábel

3.3 Informačná bezpečnosť v DLMS/COSEM

Ako bolo spomenuté v odseku o autentifikácii tak autentifikácia prebieha počas vytvorenia logického spojenia medzi klientom a serverom. Autentifikačné mechanizmy určujú protokol, ktorý majú dve strany použiť na autentizáciu počas vytvárania spojenia. K dispozícii sú tri rôzne mechanizmy autentifikácie s rôznymi úrovňami zabezpečenia autentifikácie:

1. žiadna bezpečnosť autentifikácie (lowest level security),
2. nízka úroveň bezpečnosti LLS (Low Level Security),
3. vysoká úroveň bezpečnosti HLS (High Level Security).

Systém autentifikácie je ukázaný na obrázku číslo 3.4. Bezpečnosť výmeny správ vo fáze 2 je nezávislá od autentifikácie klient-server počas vytvárania spojenia (1. fáza). Dokonca aj v prípade, že sa neuskutočňuje žiadna autentifikácia klient-server, možno na zaistenie bezpečnosti správ použiť kryptograficky chránené dáta jednotiek aplikačnej vrstvy.



Obr. 3.4: Úrovne zabezpečenia autentifikácie

Žiadne zabezpečenie

Účelom autentizácie bez zabezpečenia je umožniť klientovi získať niektoré základné informácie zo servera. Tento autentifikačný mechanizmus nevyžaduje žiadnu autentifikáciu. Jedná sa o najnižší stupeň zabezpečenia. Klient môže pristupovať k atribútom a metódam objektu COSEM v rámci bezpečnostného kontextu predaného pri nadväzovaní spojenia. Bezpečnostný kontext definuje bezpečnostné atribúty relevantné pre kryptografické transformácie a zahŕňa niekoľko prvkov. Bezpečnostný kontext sa prenáša vo všetkých typoch zabezpečenia a obsahuje napríklad používané bezpečnostné algoritmy, bezpečnostnú politiku, kľúče, inicializačné vektory a certifikáty verejného kľúča alebo aj prístupové práva k metódam. Tie môžu byť RO, WO, RW (Read only, Write only, Read and Write).

LLS

V tomto prípade server vyžaduje, aby sa klient autentifikoval zadaním hesla, ktoré server pozná. Klient odošle tajomstvo (heslo) serveru pomocou primitíva služby COSEM-OPEN.request a následne server skontroluje, či je heslo správne. Ak áno, klient je autentifikovaný a môže byť zriadené spojenie. Ak nie, spojenie bude zamietnuté. Výsledok zriadenia spojenia odošle server späť smerom klientovi pomocou primitíva služby COSEM-OPEN.response spolu s diagnostickými informáciami.

HLS

Ako posledná možnosť je uvedený najvyšší stupeň bezpečnosti, kde sa klient aj server musia úspešne autentifikovať, aby vytvorili spoločné spojenie. Autentifikácia HLS je proces zložený zo štyroch krokov, ktorý je podporovaný službou COSEM-OPEN a metódou *reply_to_HLS_authentication*. Hlavnou časťou, ktorá sa líši v týchto krokoch oproti LLS je to, že prebieha poslanie a prijatie výzvy so žiadosťou o overenie a naviac prejdú viacerými službami kontroly (AARQ (A-Associate Request) a AARE (A-Associate Response)). Existuje niekoľko typov mechanizmov autentifikácie HLS. V niektorých mechanizmoch autentifikácie HLS zahŕňa spracovanie výziev aj použitie tajného kľúča HLS.

3.3.1 Kryptografické algoritmy

Kryptografické prvky sa vo všeobecnosti používajú za účelom ochrany informácii a na rovnaký účel sú použité aj v tomto prípade. Tak ako všade aj tu je hlavnou úlohou dodržať 5 základných služieb: dôvernosť, integrita údajov, autentifikácia, autorizácia a nepopierateľnosť.

Protokol DLMS/COSEM používa širokú škálu protokolov a funkcií k dodržaniu čo najvyššej úrovne bezpečnosti. Ako prvé by sme chceli spomenúť hashovacie funkcie, tie v DLMS/COSEM sú použité na digitálny podpis, dohodu o kľúčoch a autentifikáciu HLS. Symetrická kryptografia je zase využívaná na autentifikáciu komunikujúcich partnerov pomocou autentifikačných mechanizmov HLS, overovanie a šifrovanie správ xDLMS a overovanie a šifrovanie údajov COSEM. Protokol používa aj rôzne módy šifrovanie medzi ktoré môžeme zaradiť napríklad Galois/Counter Mode (GCM) na generovanie MAC (Message Authentication Code), ktoré poskytujú záruku pravosti a integrity. Ďalej môžeme spomenúť, že protokol využíva algoritmus AES (Advanced Encryption Standard) mimo iné aj na tzv. obaľovanie kľúčov. Obaľovanie kľúčov znamená, že zašifrujeme kľúče ktoré sú potrebné na dešifrovanie komunikácie. [18]

Ako ďalšiu kryptografickú komponentu používa DLMS/COSEM inicializačný vektor. Inicializačný vektor je zreťazenie dvoch polí, nazývaných pevné pole a pole vyvolania. Pevné pole identifikuje fyzické zariadenie a pole vyvolania identifikuje sady vstupov. Kryptografia stavia na dvoch ťažko riešiteľných problémoch, na faktorizácii veľkých prvočísel a na probléme diskrétneho logaritmu. Kryptografia založená na eliptických krivkách poskytuje podobnú úroveň zabezpečenia ako RSA (Rivest, Shamir, Adleman), ale s výrazne zníženou veľkosťou kľúčov. Tento typ kryptografie je obzvlášť vhodný pre vstavané zariadenia, a preto bol vybraný aj na použitie v DLMS/COSEM.

Eliptické krivky využíva DLMS/COSEM aj na digitálny podpis v spojení s hashovacími funkciami a to v dvoch prípadoch. V prvom prípade sa jedná o eliptickú krivku P-256 s hashovacím algoritmom SHA-256 a druhá varianta je eliptická krivka P-384 s hashovacím algoritmom SHA-384. Medzi posledný kryptografický by sme chceli uviesť algoritmus DH (Diffie-Hellman). Ten sa v spojení s eliptickou krivkou používa na kľúčovú dohodu dvoch entít a umožňuje im spoločne vypočítať zdieľané tajomstvo. [18]

Celý súhrn kryptografického balíčka, ktorý protokol používa môžeme vidieť v tabuľke č. 3.3

Tab. 3.3: Kryptografický balíček pre DLMS/COSEM

ID bezpečnostného balíka	0	1	2
Názov bezpečnostného balíka	AES-GCM -128	ECDH-ECDSA-AES-GCM-128-SHA-256	ECDH-ECDSA-AES-GCM-256-SHA-384
Šifrovanie	AES-GCM -128	AES-GCM-128	AES-GCM-256
Digitálny podpis	-	ECDSA P-256	ECDSA P-384
Kľúčová dohoda	-	ECDH P-256	ECDH P-384
Hashovacia funkcia	-	SHA-256	SHA-384
Transport kľúča a jeho obalenie	AES-128	AES-128	AES-256

K prostriedkom serverov DLMS/COSEM, atribútom a metódam objektov COSEM, môžu pristupovať klienti DLMS/COSEM v rámci asociácií aplikácií. Počas vytvárania spojenia sa klient a server musia identifikovať. Môže nastať niekoľko možných variánt autentifikácie. Mechanizmy identifikácie a autentifikácie sú viac

popísané vyššie. Po uzatvorení spojenia možno služby xDLMS použiť na prístup k atribútom a metódam objektu COSEM v závislosti od kontextu zabezpečenia a prístupových práv. Jednotky aplikačnej vrstvy nesúce primitíva služieb môžu byť tiež kryptograficky chránené. Požadovaná ochrana je určená kontextom zabezpečenia a prístupovými právami. Na podporu komplexnej bezpečnosti medzi tretími stranami a servermi môžu tieto tretie strany tiež pristupovať k zdrojom servera pomocou klienta ako sprostredkovateľa.

4 Zraniteľnosti DLMS/COSEM

V tejto kapitole je spomenutých niekoľko zraniteľností, ktoré sa pri používaní štandardu DLMS/COSEM môžu vyskytnúť a predstavujú výzvu pre všetkých užívateľov tohoto protokolu. Musíme podotknúť, že nie každá zraniteľnosť je hneď viditeľná a nie každá zraniteľnosť sa dá rýchlo a jednoducho zneužiť a preto nie je nutné hľadieť na protokol DLMS/COSEM ako málo bezpečný, pretože žiadna technológia nie je bez chyby. Samotný štandard neobmedzuje výrobcov ani v tom aby si implementovali vlastné riešenie pre bezpečnosť. Medzi základné zabezpečovacie mechanizmy patrí identifikácia, autentifikácia, autorizácia a šifrovanie.

4.1 Slabiny DLMS/COSEM

4.1.1 Slabá autentifikácia a kryptografické metódy

Prvá slabina sa vyskytuje v bezpečnostnej hlavičke, tá sa odosiela v otvorenom texte aj v chránených APDU, čo spôsobuje, že je náchylná na úpravy. Napríklad zmenou jedného bitu v hlavičke zabezpečenia a odstránením posledných dvanástich bajtov APDU a aktualizáciou jej dĺžky môže útočník premeniť overenú a zašifrovanú APDU na iba zašifrovanú APDU. Ak bezpečnostná politika cieľového zariadenia povoľuje iba šifrované APDU, útočník môže znížiť úroveň ochrany aplikovanej v APDU, čím efektívne spôsobí útok na nižšiu verziu. [19]

4.1.2 Únik informácií

Niektoré služby xDLMS používajú špecifické šifrovanie čoho výsledkom je klasický variant normálneho xDLMS APDU. V tomto prípade je značka, ktorá identifikuje typ APDU, ponechaná v otvorenom texte. [19] To nám odhaľuje, ktorá služba xDLMS sa prenáša v APDU. Vďaka tomu je pre útočníka triviálne zistiť, kedy bol odoslaný požiadavok na získanie, bez ohľadu na silnú ochranu APDU. Tieto informácie by mohli byť použité napríklad v prípade útoku hrubou silou keďže útočník by mal ľahšie určenie, ako dešifrovať APDU.

4.1.3 Pevná veľkosť správy

Ďalšia zraniteľnosť súvisí s pevnými veľkosťami správ, ktoré môžu tiež pomôcť protivníkovi zistiť, aké správy sa prenášajú. Tento problém je obzvlášť akútny pre APDU s pevnou veľkosťou, ktorá je jedinečná medzi vymieňanými správami. To uľahčuje hádanie obsahu v poliach týchto správ, čo je zase užitočné na vykonávanie útokov so známym otvoreným textom.

Okrem toho je možné získať veľa známeho otvoreného textu z iných zdrojov. Dobrým príkladom sú polia s informáciami o používateľovi v odpovediach AARQ. Zistilo sa, že overovacie odpovede HLS 5 sú najspoľahlivejším zdrojom známeho otvoreného textu, ktorý poskytuje až 22 bajtov ľahko dostupných informácií. [19]

4.1.4 Integrita toku správ

V správach nie je zahrnuté nič, čo by klientovi umožnilo prepojiť odoslaný požiadavok s prichádzajúcou odpoveďou. Klient preto nemôže jednoducho rozlíšiť medzi skutočnou odpoveďou a odpoveďou, ktorá bola podvrhnutá prostredníctvom útoku typu man-in-the-middle.

4.1.5 Transportná vrstva DLMS/COSEM

Cieľom metód zabezpečenia transportnej vrstvy v DLMS/COSEM je chrániť údaje pred neoprávneným prístupom. Na to nám slúžia už spomínané tri rôzne kryptografické balíčky pre DLMS/COSEM s označením 0, 1 a 2.

Balíček číslo 0 bol predstavený vo verzii 7.3 Zelenej knihy. Na šifrovanie používa GCM režim (Galois/Counter Mode). V rámci GCM sa ako bloková šifra používa AES. Kľúč, ktorý sa vyžaduje na šifrovanie, má dĺžku 128 bitov. GCM využíva AES na generovanie prúdu kľúčov na šifrovanie XOR medzi prúdom kľúčov a otvoreným textom. Výhodou tejto metódy je, že bloková šifra sa konvertuje na prúdovú šifru. Jednou z nevýhod šifrovania XOR je potenciálny **útok na výmenu správy (Message Replacement Attack)**. Ďalšou dôležitou požiadavkou je, že prúdové šifry nesmú mať možnosť použiť inicializačné vektory dvakrát spolu s tým istým kľúčom. To je jedno z opatrení, ktoré je nutné dodržať aby sa predišlo ďalšiemu typu útoku a tým je **útok prúdovou šifrou (Stream Cipher Attack)**.

Zraniteľnosti popísané platné pre balíček číslo 0 sú platné aj pre balíčky číslo 1 a 2. Hlavným rozdielom medzi bezpečnostnou sadou 0 a sadami 1 a 2 je, že nie sú k dispozícii metódy asymetrickej kryptografie, takže je potrebná dohoda o kľúči offline a aj to môže prispievať ku **slabej náhodnosti pri šifrovacom kľúči (Weak Randomization at the Encryption Key)**. Bezpečnostné balíčky 1 a 2 v Zelenej knihe používajú digitálne podpisy založené na eliptickej krivke (ECDSA) a na dohodu kľúčov Diffie-Hellman s eliptockými krivkami (ECDH). Zvláštnosťou je, že dĺžka overovacej značky generovanej GCM sa medzi súpravami 1 a 2 nelíši. V roku 2005 boli preukázané slabiny autentifikačnej funkcie GCM, keď sa používa s krátkou autentifikačnou značkou, typ slabiny resp. možnosti útoku je v tomto prípade **krátka autentizačná značka (Short Authentication Tag)**. Preto sa dôrazne odporúča používať 128-bitový alebo vyšší autentifikačný tag. [20]

4.1.6 Aplikačná dátová jednotka xDLMS

Po použití kryptografickej ochrany je potrebné údaje zapúzdriť a preniesť. Vo verzii 8 Zelenej knihy DLMS/COSEM ponúka na tento prístup aplikačné dátové jednotky (xDLMS APDU). Ide o zašifrované pakety s použitím globálneho alebo vyhradeného kľúča. APDU sa môžu šifrovať štvoro rôznymi spôsobami.

Prípad 1 opisuje APDU, ktoré sú iba komprimované. Prípad 2 opisuje použitie iba autentifikácie. V prípade 3 sa používa šifrovanie, ale bez autentifikácie. Táto metóda je zraniteľná voči **útokom na výmenu správy (Message Replacement Attack)**. Prípadní útočníci môžu na zašifrovaný text použiť operáciu XOR a týmto spôsobom môžu útočníci manipulovať so správou, čo môže viesť napríklad k falošným údajom o stave merača. V prípade 4 sa používa úplné šifrovanie GCM vrátane overovacej značky. Manipuláciou s nezašifrovaným obsahom hlavičky môžu útočníci tento prístup degradovať na úroveň prípadu 3. Tento útok sa nazýva **útok na zníženie autentifikácie (Authentication Downgrade Attack)** a možným protiopatrením je povoliť použitie iba prípadu 4 vo všeobecnosti.

4.1.7 Úrovne zabezpečenia overovania

Autentifikácia je základnou súčasťou vytvorenia AA (application association – združenie aplikácií), jej cieľom je chrániť informácie inteligentného merača pred neoprávneným prístupom. DLMS/COSEM definuje pre autentifikáciu tri rôzne úrovne zabezpečenia. Jednotlivé úrovne sú viac popísané v predchádzajúcej kapitole. Najnižšia úroveň je bez zabezpečenia, čo znamená, že sa vôbec neposkytuje autentifikácia. A ďalšie dve sa nazývajú LLS a HLS.

V LLS sa používa vopred definované heslo. Posiela sa v otvorenej podobe meraču, ktorý následne môže skontrolovať rovnosť hesla. Útočníci môžu zachytiť balík, a tak získať heslo na overenie, preto by sa LLS mala používať len vtedy, ak sa používa šifrovanie na jednej z nižších úrovní. Aj v takom prípade je však systém stále zraniteľný voči útokom hrubou silou na slabé heslá. V prípade, že útočníci zachytia balík tak môžeme hovoriť o útokoch typu **odhalenie tajomstva LLS (LLS Secret Disclosure**), **útoku hrubou silou na LLS (LLS Brute Force Attack)** a **únik informácií LLS (LLS Information Leakage)**.

High Level Security (HLS) v podstate definuje päť rôznych spôsobov autentifikácie klienta k inteligentnému meraču. Namiesto hesla sa používa všeobecný termín „tajomstvo HLS“. Tajomstvo HLS sa neprenáša v otvorenej podobe. Namiesto toho sa používa metóda výzva-odpoveď, vďaka čomu je HLS zo svojej podstaty bezpečnejší proti úniku hesla ako LLS. V prvých dvoch metódach spôsobu autentifikácie prostredníctvom HLS sa používa iba hash predtým vymeneného

náhodného čísla a tajomstva HLS. Útočník môže obísť autentifikáciu použitím **útoku na opakovanie (Replay Attack)**. V tomto prípade útočník môže zachytiť údaje pri prenose a neskôr ich opakovať alebo upraviť priamo pri prechode sieťovým prvkom. Aby sa zabránilo tomuto typu útoku protokol núti klienta prerušiť vytvorenie AA v prípade, že výzvy servera a klienta sú rovnaké. Ak sa však v rôznych klientoch používa rovnaké tajomstvo HLS, útok sa stále dá vykonať. Útočník musí len mierne upraviť prístup. Na vykonanie útoku je potrebné, aby dvaja klienti, ktorí zdieľajú rovnaké tajomstvo HLS, kontaktovali server (útočníka) takmer súčasne.

Keď útočník zistí takúto situáciu, zhromaždí výzvy oboch klientov, vymení ich a odošle späť klientom, pričom predstiera, že je to výzva servera. V tomto prípade sa útočník môže overiť u oboch klientov ako server. V skutočnosti to znamená, že útočníci by mohli úspešne spustiť pripojenie jedného klienta. Tento prístup sa nazýva **paralelné overovanie relácie (Parallel Session Authentication)**. Tomuto útoku sa dá predísť použitím kryptografického kľúča ako tajomstva HLS pre každého klienta, čo znamená, že tajomstvo je zvolené rovnomerne náhodne a dostatočne dlhé, týmto sa predíde aj slabine protokolu kde je **slabá náhodnosť pri šifrovacom kľúči (Weak Randomization at the Encryption Key)**. Napríklad pri použití tajomstva HLS s dĺžkou 160 bitov je pravdepodobnosť, že dvaja používatelia zdieľajú rovnaké tajomstvo, zanedbateľná. V skutočnosti sa však za tajomstvá často volia veľmi slabé heslá, ktoré zdieľajú často aj veľké skupiny zariadení. [20]

4.2 Najčastejšie útoky

Zraniteľnosť je v informatike označenie pre chybu, ktorá v softwari alebo v hardwari spôsobuje bezpečnostný problém. Ak je napríklad v protokole známa chyba je len otázka času kedy sa ju pokúsi niekto využiť, túto skutočnosť, že je to možné využiť označujeme ako exploit. Útočník môže využiť zraniteľnosť napríklad pre zmocnenie sa prvku alebo čítanie dát alebo v iný svoj prospech.

4.2.1 DoS a DDoS

Medzi najviac viditeľné zraniteľnosti protokolu DLMS/COSEM patrí určite náchylnosť na útoky typu DoS alebo DDoS. Kybernetický útok typu DoS je kybernetický útok, ktorý má za cieľ obmedziť alebo vyradiť služby počítačových systémov v našom prípade priamo elektromer. Spravidla ide buď o generovanie veľkého množstva podvrhnutých požiadaviek s cieľom zahltiť systém alebo prenosovú cestu alebo ide o sofistikovaný útok na slabé miesta v cieľovom systéme alebo prenosovej ceste. Druhú variantu tvorí kybernetický útok typu DDoS. Jedná sa o kybernetický útok typu DoS, ktorý prebieha naraz koordinovane z mnohých

uzlov siete na jeden cieľ a tým ho zahltia požiadavkami a cieľ nie je schopný reagovať na skutočnú nepodvrhnutú správu. Distribuovaný DoS útok má oproti normálnemu útoku niekoľko výhod. Po prvé, väčšie množstvo útočníkov (zombie počítačov v botnete) môže generovať väčšie množstvo sieťovej prevádzky, než jeden počítač, a tým spôsobí väčšie zaťaženie cieľového systému. Po druhé, útok viacerých útočníkov je ťažšie ošetriť napríklad aj preto, že viac jednotlivých útočiacich systémov môže byť menej agresívnych, než keby bol útok vedený len jedným systémom, čo môže sťažiť ich odhalenie . Výhodou je tiež škálovateľnosť útoku, kedy útočník môže ľahko meniť počet útočiacich počítačov, takže napríklad ak obeť zvýši prenosovú kapacitu svojho internetového pripojenia alebo zväčší výpočtový výkon napadnutého počítača, môže útočník jednoducho pridať k útoku ďalšie zombie počítača.

Pokiaľ budeme na elektromer posielať neustále nejaké správy a tým mu zahltíme pamäť tak elektromere dochádza k určitému druhu výnimočného stavu. Na väčšinu dotazov na objekty a ich atribúty (väčšie ako 1) sú prijímané chybové odpovede a elektromer vôbec neodpovedá na validné požiadavky. Dokonca aj požiadavky z koncentrátora na tieto chybové objekty končia chybou. Pri neustálom zasielaní správ a dlhšom generovaní stále ďalších správ nastáva ďaleko závažnejšia situácia. Nakoniec sa stane, že elektromer sa dostane do takého stavu kde zahadzuje všetky TCP spojenia. V takom prípade pravdepodobne dôjde k uzavretiu portu 4059. Jediným funkčným spojením v takom prípade je *ping* a *telnet*. Po zablokovaní spojenia sa pri čítaní z koncentrátora objavuje chybová hláška „UnexpectedResponse“. [21]

4.2.2 Útok hrubou silou (Brute force attack)

Tento typ útoku patrí medzi najznámejšie kryptografické útoky. Útok spočíva v tom, že útočník odošle veľa hesiel alebo prístupových fráz s nádejou, že nakoniec ho uhádne. Útočník systematicky kontroluje všetky možné heslá a prístupové frázy, kým nenájde to správne. Alternatívne sa môže útočník pokúsiť uhádnuť kľúč, ktorý je zvyčajne vytvorený z hesla, pomocou funkcie odvodenia kľúča. Pri hádaní hesiel je táto metóda veľmi rýchla, keď sa používa na kontrolu všetkých krátkych hesiel, ale na dlhšie heslá sa používajú iné metódy, ako napríklad slovníkový útok, pretože vyhľadávanie hrubou silou trvá príliš dlho.

Ako druhú najnižšiu úroveň autentifikácie DLMS/COSEM definuje LLS (Low Level Security). Tu sa používa preddefinované heslo. Heslo sa posiela v jednoduchej podobe do meracieho prístroja (elektromera), ktorý si môže skontrolovať či sa heslo rovná s tým čo má uložené. Útočníci v tomto prípade môžu zachytiť balík a tak získať heslo na autentifikáciu. Preto by sa LLS malo používať iba vtedy, ak sa používa

šifrovanie na nižšej vrstve. Aj v takom prípade je však systém stále zraniteľný voči útokom hrubou silou proti slabým heslám. Protokol bol testovaný na heslá s dĺžkou osem znakov a zistilo sa, že to nie je dostatočne bezpečné. Podľa odporúčaní by sa teda malo použiť heslo s najmenej 14 znakmi, ktoré sa vyberie náhodne z písmen a číslic. Takéto heslo dosahuje 80 bitov Shannonovej entropie. [22]

Stručný prehľad útokov je uvedený v tabuľke č. 4.1

Tab. 4.1: Prehľad útokov a zraniteľností pre protokol DLMS/COSEM

ID	Typ útoku (zraniteľnosti)
1.	Útok na nahradenie správy (Message Replacement Attack)
2.	Útok prúdovou šifrou (Stream Cipher Attack)
3.	Slabá náhodnosť pri šifrovacom kľúči (Weak Randomization at the Encryption Key)
4.	Krátka autentizačná značka (Short Authentication Tag)
5.	Únik informácií (Information Leakage)
6.	Útok na zníženie autentifikácie (Authentication Downgrade Attack)
7.	Odhalenie tajomstva LLS (LLS Secret Disclosure)
8.	Útok hrubou silou LLS (LLS Brute Force Attack)
9.	Únik informácií HLS a LLS (HLS and LLS Information Leakage)
10.	Autentifikácia na základe jednej výzvy (One Challenge Authentication)
11.	Paralelné overovanie relácie (Parallel Session Authentication)
12.	Útok HLS hrubou silou offline (Offline Brute Force HLS Attack)
13.	Útok na opakovanie (Replay attack)
14.	Útok na nedostupnosť (DoS - Denial of Service)

5 Cyber range platformy

Kybernetická bezpečnosť je výzvou dvadsiateho prvého storočia, ktorá si vyžaduje ľudí, ktorí sú vzdelaní a sú odborníci vo svojom vednom odbore. V súčasnej dobe je veľký nedostatok pracovníkov v oblasti kybernetickej bezpečnosti a chýba dostatok odborníkov so zručnosťami, školeniami a povereniami. Štúdie trhu predpovedajú, že táto situácia sa nemá zlepšiť ani v najbližšom období. Tento nedostatok pracovnej sily v oblasti kybernetickej bezpečnosti predstavuje obrovské riziko pre podnikanie, vládu a spoločnosť. Kľúčovým nástrojom a platformou na zníženie rozdielu v zručnostiach a zabezpečenie spoločnosti má byť práve cyber range platforma. Cyber range platformy boli vytvorené na to aby slúžili na poskytovanie národných, regionálnych a globálnych kybernetických cvičení nie len pre študentov ale pre všetkých, ktorí by mali záujem na sebe progresívne pracovať v oblasti bezpečnosti.

5.1 Definícia Cyber range platforiem

5.1.1 Hlavné výhody

Cyber range platformy sú interaktívne, simulované platformy a reprezentácie sietí, systémov, nástrojov a aplikácií. Tieto platformy môžu poskytovať:

- školenie a hodnotenie založené na výkone,
- simulované prostredie, kde môžu tímy spolupracovať na zlepšení tímovej práce a tímových schopností,
- spätnú väzbu v reálnom čase,
- skúsenosti na pracovisku,
- prostredie, v ktorom možno testovať nové nápady a kde tímy môžu pracovať na riešení zložitých kybernetických problémov.

5.1.2 Dôvod vzniku

Cyber range platformy môžu a musia zohrávať ústrednú úlohu pri uľahčovaní a podpore vzdelávania, školenia a certifikácie v oblasti kybernetickej bezpečnosti. Tieto kritické nástroje môžu zahŕňať skutočný hardware a software alebo môžu byť kombináciou skutočných a virtuálnych komponentov, to všetko si stačí len nakonfigurovať napríklad v grafickom prostredí.

Prípadov kde je vhodné použiť túto platformu je mnoho, tak ako aj ľudí čo znej môžu profitovať. V nasledujúcom zozname je uvedených niekoľko ukážkových prípadov kto a ako môže platformu využiť:

1. Pedagógovia, ktorí sa snažia zaviesť základné a pokročilé vzdelávacie kurzy a osnovy v oblasti kybernetickej bezpečnosti.
2. Organizácie alebo jednotlivci, ktorí hľadajú školenie a ďalšie vzdelávanie pre bezpečnostné operácie, analýzy a forenzných špecialistov.
3. Organizácie, ktoré chcú testovať „situačné operácie“ pre nové produkty, verzie softwaru a organizačnú zmenu štruktúry.
4. Organizácie alebo jednotlivci, ktorí hľadajú overenie zručností v oblasti kybernetickej bezpečnosti na vyhodnotenie kandidátov na pozície v oblasti kybernetickej bezpečnosti.
5. Jednotlivci, ktorí hľadajú odbornú prípravu pretože prechádzajú do oblastí a pozícií súvisiacich s kybernetickou bezpečnosťou. [23]

5.2 Komponenty Cyber range

5.2.1 Range Learning Management System

Ústrednou funkciou pre mnohé kybernetické rady je systém riadenia učenia rozsahu, anglicky Range Learning Management System (RLMS). Ako už názov napovedá, systém riadenia učenia rozsahu obsahuje štandardné funkcie platformy systému riadenia učenia (LMS). LMS je softwarová aplikácia alebo webová technológia používaná na plánovanie, implementáciu a hodnotenie špecifického vzdelávacieho procesu. Systém riadenia výučby zvyčajne poskytuje inštruktorovi spôsob, ako vytvárať a poskytovať obsah, monitorovať účasť študentov a hodnotiť výkon študentov. [23]

5.2.2 Orchestračná vrstva

Na základe vstupu z RLMS, orchestračná vrstva spája všetky komponenty technológie alebo služieb z kybernetickej oblasti. Orchestračná vrstva uľahčuje prepojenie základnej infraštruktúry, virtualizačnú alebo izolačnú vrstvu a cieľovú infraštruktúru. Táto vrstva tiež umožňuje dynamickú rozšíriteľnosť cyber range platformy, ktorá podporuje napríklad cloud. [23]

5.2.3 Základná infraštruktúra

Všetky kybernetické platformy sú nad úložiskami, servermi a sieťami. Niektoré platformy sú postavené priamo na fyzickej infraštruktúre (prepínače, smerovače, firewall, ...) v racku. Táto možnosť je drahá a nie je ľahko škálovatelná. Z dôvodov škálovateľnosti, nákladov a rozšíriteľnosti mnohí poskytovatelia volia softwerovo definovanú virtuálnu infraštruktúru. [23]

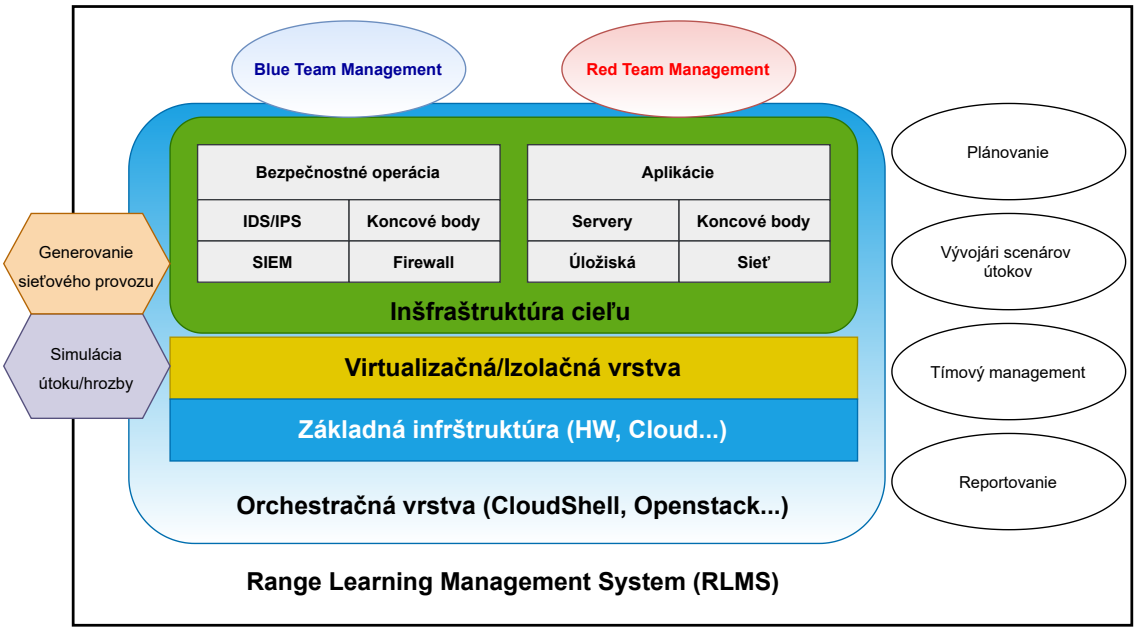
5.2.4 Virtualizačná vrstva

Väčšina cyber range platforiem sa zameriava na určitú úroveň virtualizácie aby sa zmenšila fyzická stopa. Virtualizačný prístup sa tu používa aj vzhľadom na nechcený a nepredvídateľný jitter a oneskorenia v systéme.

5.2.5 Infraštruktúra cieľu

Cieľovou infraštruktúrou je simulované prostredie, v ktorom študenti trénujú. Na základe use casu sa cieľová infraštruktúra môže v niektorých prípadoch zhodovať s reálnou IT infraštruktúrou. Pokročilé cyber range platformy obsahujú profily komerčne dostupných serverov, úložísk, koncových bodov, aplikácií a firewallov. Na základe interakcie študentov RLMS vygeneruje skripty, ktoré inštruujú orchestračnú vrstvu na vytvorenie cieľovej infraštruktúry. Tieto skripty môžu obsahovať informácie o konfigurácii, ktoré sú špecifické pre klienta vrátane rozsahov adries IP, informácií o smerovaní a softwaru koncového bodu. [23]

Zhrnutie a infraštruktúru celého systému platformy cyber range môžeme vidieť na obrázku číslo 5.1.



Obr. 5.1: Štruktúra Cyber Range

Cyber range platformy, ktoré sa spoliehajú na lokálnu hardware infraštruktúru, sú obmedzené množstvom pamäte RAM a miesta na pevnom disku. Tieto platfromy môžeme škálovať len do bodu, než nie sú vyčerpané zdroje. Platformy založené na

cloudoch by sa mali vo všeobecnosti veľmi dobre škálovať, pretože na požiadanie môžu využiť ďalšie systémy poskytovateľov cloudu.

5.3 Typy platforiem

Rozdiely medzi platformami sa stávajú dôležitými pri priraďovaní typu platformy k jednotlivému prípadu použitia jednotlivca alebo organizácie.

5.3.1 Simulačné cyber range

Koncept simulácie spočíva v tom, že sa vždy vytvorí nové sieťové prostredie založené na správaní skutočných sieťových komponentov. Simulácie bežia vo virtuálnych inštanciách a nevyžadujú žiadne fyzické sieťové vybavenie. Tieto šablóny virtuálnych strojov sú štandardizované, a preto sú do istej miery obmedzené v tom, ako presne simulujú skutočnú IT infraštruktúru. Veľkou výhodou simulačného prostredia je rýchlosť konfigurácie a schopnosť používať sieťové a úložné prostriedky. Primárnou nevýhodou simulovanej siete je nepredvídateľná a nereálna latencia a kolísanie výkonu siete.

5.3.2 Overlay cyber range

Overlay cyber range platformy bežia na skutočných sieťach, serveroch a úložiskách. Majú významnú výhodu v oblasti vernosti v porovnaní so simulačnými rozsahmi. Nevýhodou oproti simulačným platformám je to, že prinášajú značné náklady na hardware.

5.3.3 Emulačné cyber range

Emulácia prevádzkuje cyber range na vyhradenej sieťovej infraštruktúre, ktorá sa ako vybudovaná sieť/server/úložisko mapuje na fyzickú infraštruktúru. Emulácia poskytuje skúsenosti s uzavretou sieťou s viacerými vzájomne prepojenými prostrediami. [23]

5.3.4 Hybridné cyber range

Hybridný typ ako už názov napovedá je schopný skombinovať akýkoľvek typ platformy aký bol uvedený vyššie.

6 KYPO

V tejto kapitole je bližšie popísaná vybraná cyber range platforma, ktorá je použitá pri práci a pri bezpečnostných scenároch. KYPO alebo Kryptografický Polygon je cyber range platfroma, ktorá je vyvíjaná už od roku 2013 tímom z Masarykovej univerzity. Táto platforma už bola aj v minulosti použitá na výučbu študentov a školenie odborníkov v oblasti kybernetickej bezpečnosti.

6.1 Predstavenie

Vzhľadom na špecifické využitie platformy KYPO, napríklad vláda, armáda, priemysel, sa mnohé technické detaily považujú za citlivé a nemôžu byť zverejnené. KYPO bolo vytvorené na výskum a vývoj nových bezpečnostných metód, nástrojov a na školenie bezpečnostných tímov a študentov. Poskytuje virtualizované prostredie na vykonávanie zložitých kybernetických útokov proti simulovaným kybernetickým prostrediam. Najväčšiu časť práce vykonávajú pri využívaní platformy organizátori a to najmä vo fáze prípravy cvičenia. [24]

6.2 Architektúra

KYPO je navrhnutý ako modulárny distribuovaný systém. Platforma KYPO využíva cloudové prostredie na dosiahnutie vysokej flexibility a škálovateľnosti. Virtualizácia zase umožňuje opakovane vytvárať plne funkčné virtualizované siete s plnohodnotnými operačnými systémami a sieťovými prvkami, ktoré sú takmer identické so systémami z reálneho sveta. Vďaka svojej modulárnej architektúre je KYPO schopný bežať na rôznych platformách cloud computingu, napr. OpenNebula alebo OpenStack. [24] V tabuľke číslo 6.1 sú uvedené základné požiadavky na cyber range platformu KYPO. Požiadavky sú spísané stručne a prehľadne z dôvodu prehľadnosti.

Platforma sa skladá z piatich hlavných komponentov:

1. Ovládač správy infraštruktúry.
2. Management sandboxu.
3. Úložisko dát pre sandbox.
4. Správa monitorovania.
5. Portál správy platformy.

Portál správy platformy slúži ako hlavný bod interakcie užívateľa. Tieto komponenty spolu interagujú, aby vytvorili a spravovali sandbox (oddelený priestor)

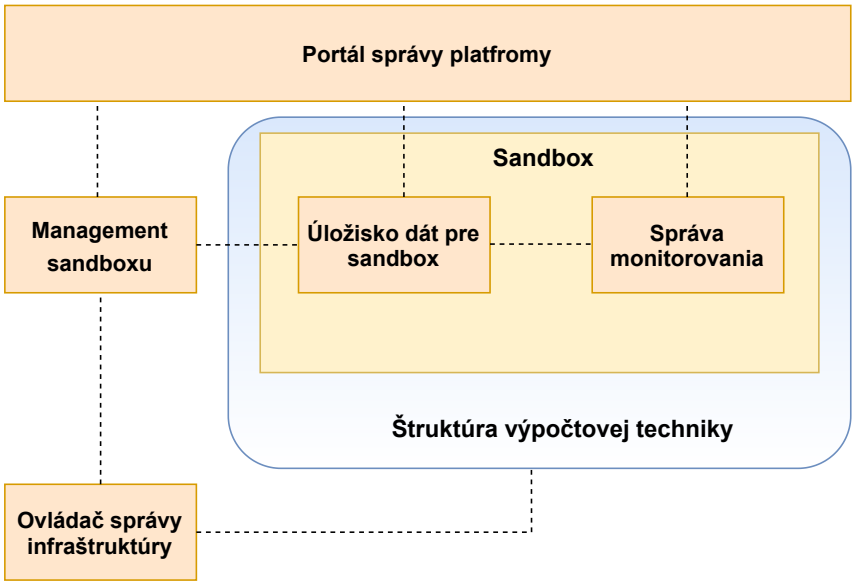
Tab. 6.1: Požiadavky na platformu KYPO

Flexibilita	ľubovolná sieťová topológia, široká škála operačných systémov, rýchla konfigurácia topológii
Škálovateľnosť	z pohľadu počtu uzlov v sieti, výpočtového výkonu, veľkosti siete a šírky pásma, počtu sandboxov a počtu používateľov
Izolácia	rôzne topológie a používatelia platformy by mali mať možnosť byť izolovaní od vonkajšej siete a od seba navzájom
Efektívnosť nákladov	mal by podporovať nasadenie na komerčný bežne dostupný hardvér bez potreby špeciálneho dátového centra
Monitorovanie	poskytnutie dohľadu v reálnom čase, údaje o topológii, údaje o tokoch a zachytených paketov a o protokoloch
Jednoduchý prístup	platforma pre menej skúsených používateľov by mal byť vo forme webového prístupu, pokročilí užívatelia napr. cez SSH
Transparentný prístup	náklady na vývoj a údržbu sú veľké, cieľom je poskytnúť transparentný prístup k platforme vo forme služby
Open Source	platforma opätovne používa vhodné open source projekty a aj jej vydania by mali byť distribuované pod licenciami open source

nachádzajúci sa v základnej infraštruktúre cloud computingu. Spôsob akým sú jednotlivé komponenty previazané môžeme vidieť na obrázku číslo 8.2.

Ovládač správy infraštruktúry

Používa sa na riadenie výpočtovej infraštruktúry, tú tvoria napríklad zariadenia v byte, fyzické stroje alebo aj iné sieťové prvky a hardware. KYPO je navrhnutý tak, aby fungoval na verejnej infraštruktúre cloud computingu, takže sandboxy možno vybudovať bez potreby vyhradenej infraštruktúry. Aplikačné prostredie poskytované ovládačom ponúka služby, ktoré umožňujú spravovať virtuálne stroje a siete jednotným spôsobom. V súčasnosti KYPO funguje na OpenStack. [24]



Obr. 6.1: Komponenty KYPO

Management sandboxu

Používa sa na vytváranie a riadenie sandboxu v základnej výpočtovej infraštruktúre. Počas nasadzovania sandobxu riadi infraštruktúru prostredníctvom ovládača správy infraštruktúry, aby konfiguroval virtuálne stroje a sieť.

Úložisko dát pre sandbox

Spravuje informácie súvisiace s topológiou sandboxu a poskytuje jej všeobecnú abstrakciu. Preto moduly pracujúce s údajmi súvisiacimi so sandboxom nezískavajú informácie priamo z cloudu, ale namiesto toho využívajú úložisko dát pre sandbox. Vždy sa aktualizuje pri nejakej akcii užívateľa, napríklad keď pridáme alebo odoberieme jeden uzol.

Správa monitorovania

Komponent správy monitorovania poskytuje kontrolu nad konfiguráciou a tiež poskytuje aplikačné prostredie, ktoré sprístupňuje získané monitorovacie údaje externým spotrebiteľom. Všetky potrebné informácie o topológii sandboxu sa načítavajú z úložiska dát pre sandbox tak ako bolo spomínané vyššie.

Portál správy platformy

Tento portál sprostredkováva prístup k platforme pre koncových užívateľov tým, že im poskytuje interaktívne vizuálne nástroje. Portál je navrhnutý tak, aby pokrýval

najmä nasledujúce typy interaktívnych služieb:

- Manažment kybernetických cvičení - definujú sa tu bezpečnostné scenáre, správa účastníkov a alokácia zdrojov. Portál správy platformy podporuje automatizáciu týchto úloh.
- Spolupráca - veľa bezpečnostných scenárov je založených na vzájomnej spolupráci, kde účastníci zdieľajú sandbox a spoločne riešia požadované úlohy alebo, naopak, súperia medzi sebou.
- Prístup do sandboxov - umožňuje koncovým užívateľom prihlásiť sa do počítačov v sandboxe prostredníctvom webového klienta vzdialenej pracovnej plochy namiesto nie tak užívateľsky prívetivého SSH spojenia cez príkazový riadok.
- Interaktívne vizualizácie - poskytuje špecializované vizualizačné a interakčné techniky, ktoré sprostredkúvajú dáta a udalosti merané v sandboxoch, ktoré sú dobre využiteľné napríklad pri analýze malwaru.

6.3 Úlohy jednotlivých rolí v platforme

Príprava kybernetického cvičenia je veľmi zložitá úloha. Je potrebné vytvoriť a definovať scenár, zariadiť alokáciu zdrojov, správu užívateľov a podobne. Pre automatizáciu týchto procesov pomocou interakcie s používateľmi je potrebné definovať užívateľské role s jasnými pravidlami prístupu a zodpovednosťou.

6.3.1 Scenárista

Úlohou scenáristu je navrhovať bezpečnostné scenáre so všetkými potrebnými detailmi vrátane definície sandboxu a návrhu webových užívateľských rozhraní pre koncových používateľov zapojených do scenára. Na tejto úrovni sú rozhrania definované ako všeobecné šablóny. Scenáristi majú tiež na starosti správu užívateľov a vyberajú kto sa môže stať sa organizátormi cvičení. [24]

6.3.2 Organizátor

Organizátor je technicky zdatná osoba poverená scenáristom plánovať a pripravovať kybernetické cvičenia alebo experimenty konkrétneho bezpečnostného scenára. Aktivity, ktoré má na starosti pozostávajú z prideľovania sandboxov v cloude, úpravy informačných stránok, konfigurácie bodovacieho subsystému a iných služieb špecifických pre scenár. Taktiež majú možnosť delegovať supervízora cvičenia. [24]

6.3.3 Účastníci

Účastníci predstavujú koncových používateľov zapojených do konkrétneho kybernetického cvičenia alebo experimentu. Využívajú webové užívateľské rozhranie pripravené scenáristami a vykonávajú úlohy predpísané bezpečnostným scenárom. V základnom formáte rozlišujeme medzi bežnými účastníkmi a tými, ktorí majú rozšírené privilégiá dohľadu. Bežní účastníci majú pridelenú len jednu rolu scenára. Napríklad scenár cvičenia definuje rolu útočníka a obrancu. Útočník potom nemá priamy prístup k hostiteľom kontrolovaným obrancom a naopak. Ďalší rozdiel je v prístupe k sandboxom. Bežní účastníci majú prístup len k jednému sandboxu, supervízori majú prístup ku všetkým sandboxom prideleným pre dané cvičenie.

6.4 Typický priebeh cvičenia

Pred samotným začiatkom tréningu je potrebné vykonať niekoľko krokov. Tieto kroky sú zvyčajne rozdelené medzi viacerých používateľov s rolou inštruktora. Celý pracovný postup vytvárania cvičenia je rozdelený na dve časti, jedna z nich je tvorba sandboxov a druhá je už samotné školenie. Definícia sandboxu je adresárová štruktúra uložená ako Git repozitár, ktorý je spravovaný inštruktormi platformy a slúži na definovanie topológie sanboxu a poskytovanie sieťových uzlov. [24]

Postup na vytvorenie sandboxu:

1. Definíciu sandboxu vytvoria inštruktori podľa daného formátu a uložia ju ako Git repozitár.
2. Na portáli KYPO sa záznam o definícii sandboxu vytvorí prostredníctvom stránky zadaním URI príslušného úložiska Git z predchádzajúceho kroku.
3. Z definície sandboxu vytvoreného na portáli KYPO je možné v tomto kroku vytvoriť skupinu (pool).
4. Sandboxy v cloude sa prideľujú do príslušného poolu. V ten moment sa automaticky vykoná niekoľko akcií. Príslušná definícia sandboxu sa stiahne z úložiska Git, analyzuje sa a spracuje sa. A následne sa sandboxy v cloude vytvoria podľa definície.
5. Sandboxy môžu byť použité dvoma spôsobmi. Inštruktor môže pristupovať k virtuálnym počítačom vo vnútri sandboxov pomocou SSH a vykonávať akékoľvek činnosti. Alebo sa používajú ako súčasť školení a ich tvorby.

V ďalšom kroku prebehne vytvorenie školenia. Toto školenie sa vytvára prostredníctvom stránky v platforme kde nasledujeme postup v platforme. Následne vytvoríme inštanciu školenia a pridelíme vybrané sandboxy pre školenie. Každá inštancia školenia má čiastočne vygenerovaný prístupový token, ktorý inštruktor odovzdá účastníkom školenia.

Počas školenia majú jeho účastníci prístup k školeniu pomocou získaného prístupového tokenu. Každému jednému tréningu je priradený konkrétny sandbox a tým môžu pristupovať k virtuálnym počítačom vrámci neho pomocou Sandbox SSH Access,Apache Guacamole alebo klienta Spice. [24] Organizátor tejto inštancie školenia môže v reálnom čase sledovať postup účastníkov školenia a počas školenia môže vidieť ich výsledky.

Po skončení inštancie školenia sú k dispozícii výsledky, ktoré sú pripravené na ďalšie vyhodnotenie. Priradený pool sa odpojí od inštancie školenia a odstráni sa pomocou tlačidla na odstránenie na stránke portálu. Tento krok slúži k tomu aby sa uvoľnili výpočtové zdroje v cloudovom systéme.

7 OpenStack

Vo svete IT existuje mnoho open source projektov a ďalšie stále pribúdajú na dennej báze. Len niektoré z nich sa však vypracujú na takú úroveň, kedy ich začnú rešpektovať a používať aj veľké medzinárodné firmy a korporácie. Podobný príbeh má za sebou aj cloudová platforma OpenStack, ktorá pomáha riadiť stále ďalšie výpočtové cloudy. Používa ju stále viac firiem, medzi najvýznamnejšie patrí napríklad Intel a PayPal.

7.1 Charakteristika

OpenStack je bezplatná, open-source platforma cloud computingu. Väčšinou sa nasadzuje ako IaaS (Infrascture as a service alebo Infraštruktúra ako služba) vo verejných a súkromných cloudoch, kde sú užívateľom poskytnuté virtuálne servery a ďalšie zdroje. Softvérová platforma pozostáva zo vzájomne prepojených komponentov, ktoré riadia rôznorodé hardvérové oblasti spracovania, úložiska a sieťových zdrojov viacerých dodávateľov v celom dátovom centre. Používatelia ho spravujú buď prostredníctvom webového ovládacieho panela, cez príkazový riadok alebo webových služieb. [25]

OpenStack je teda cloudová platforma, ktorá zaisťuje rozdeľovanie virtualizovanej výpočtovej kapacity. Znamená to, že má pod kontrolou zdroje cloudu alebo datacentra. Pri klasickom fyzickom serveri je veľký problém, keď narazíme nedostatok hardvérového výkonu, riešiť sa to dá prakticky len pridaním nového hardwaru. [26] To ale väčšinou nie je príliš efektívne riešenie. To sa dá riešiť virtualizáciou. Nad fyzické servery pribudne hypervízor, ktorý výpočtové prostriedky prerozdeľuje virtuálnym serverom. Toto riešenie je síce efektívnejšie ale na druhú stranu administrátorom a vývojárom aplikácií tým pribúda ďalšia práca. Na správne pochopenie platformy je ale potrebné ešte vysvetliť čo je cloud computing a na čo sa používa.

Pri používaní Openstacku príliš nezáleží na tom, aké servery pod ním bežia. On zdroje abstrahuje, to znamená, že všetky prostriedky priraďuje do takzvaných poolov, odkiaľ potom môžu čerpať všetky virtuálne inštancie. Vďaka tomu je možné aj komplexné systémy oveľa ľahšie riadiť. Stačí OpenStacku povedať „daj mi virtuálny stroj“ a už nie je treba riešiť, na ktorom clusteri alebo fyzickom stroji pobeží.

7.1.1 Cloud computing

Cloud computing je doručovanie výpočtových služieb, vrátane serverov, úložísk, databáz, sietí, softvéru, analytických nástrojov a inteligentných funkcií, cez internet (cloud) a ponúka rýchlejšie inovácie, flexibilitu prostriedkov a cenové výhody. [27] Platí sa len za cloudové služby, ktoré skutočne využívame, to pomáha znižovať prevádzkové náklady, efektívnejšie prevádzkovať infraštruktúru a napomáha k lepšej škálovateľnosti s ohľadom na meniace sa obchodné potreby.

Výhody cloud computingu

Cloud computing ponúka mnoho výhod a veľa z nich už bolo uvedených. Medzi najväčšie z nich ale patrí:

- Náklady - služba eliminuje investičné náklady na nákup hardwaru a sofwaru.
- Globálny rozmer - patrí tu schopnosť škálovateľnosti, znamená to dodať podľa potreby vhodné množstvo IT prostriedkov, napríklad menej alebo viac výpočtového výkonu, úložiska alebo šírky pásma.
- Výkon - služby bežia v sieti zabezpečených dátových centier, ktoré sú pravidelne upgradované na najnovšiu generáciu rýchleho a efektívneho výpočtového hardwaru.
- Rýchlosť - služby cloud computingu sa väčšinou poskytujú ako samoobslužné a na vyžiadanie, tým pádom aj veľké množstvo výpočtových prostriedkov je možné zaistiť za krátky čas.
- Zabezpečenie - poskytovatelia cloudu ponúkajú širokú škálu technológií a kontrolných prvkov, ktoré posilňujú stav zabezpečenia a tým pomáhajú chrániť dáta pred potenciálnymi hrozbami.
- Produktivita - cloud computing odstraňuje potrebu nastavenia a správy HW, rôzne opravy SW priamo na HW riešeniach.
- Spoľahlivosť - uľahčuje a znižuje náklady na zálohovanie dát, zotavenie po havárii a zaistenie prevádzkovej kontinuity, pretože dáta je možné zrkadliť vo viacerých redundantných lokalitách v rámci siete poskytovateľa cloudu.

Nie všetky cloudy sú rovnaké a neexistuje žiadny typ cloud computingu, ktorý by bol vhodný pre všetkých. A preto sa vyvinulo niekoľko rôznych modelov. Pri plánovanom užití cloudu je ako prvé potrebné určiť typ cloudového nasadenia alebo architektúry cloud computingu, v ktorom sa cloudové služby budú implementovať. Na toto existujú tri spôsoby nasadenia cloudových služieb. Patrí tu verejný cloud, privátny cloud a hybridný cloud, ktorý je kombináciou verejného a privátneho cloudu.

7.1.2 Architektúra Openstacku

OpenStack má modulárnu architektúru s rôznymi kódovými názvami svojich komponentov. Celkovo ich má k 6.10.2021 približne 38. Počas plánovacej fázy každého vydania sa komunita zhromaždí na OpenStack Design Summit, aby zostavila plány vydania. [25] V tejto časti som sa rozhodol popísať tie, ktoré využíva KYPO pre svoje riešenie. Platforma KYPO vyžaduje nasledujúce služby OpenStack:

- **Nova** s podporou konzoly SPICE.
- **Neutron** s definovanými internými sieťami a plávajúcimi IP adresami.
- **Keystone**
- **Heat**

Pre nasadenie OpenStack sa odporúča použiť open source projekt Kolla Ansible, ktorý podporuje všetky požiadavky KYPO. [24]

Nova

Nova je výpočtový projekt Openstacku. Nova je projekt, ktorý ponúka spôsob výpočtových inštancií (virtuálnych strojov). Nova podporuje vytváranie virtuálnych strojov a má obmedzenú podporu pre systémové kontajnery. Nova beží ako sada démonov nad existujúcimi Linuxovými servermi na poskytovanie tejto služby. [28] Nova je napísaná v Pythone a používa mnoho externých knižníc Pythonu. Namiesto prechodu na väčšie servery si zaobstaráte viac serverov a jednoducho nainštalujete identicky nakonfigurované služby.

Neutron

Neutron je projekt OpenStacku, ktorý poskytuje sieťovú konektivitu medzi zariadeniami rozhrania spravovanými inými službami OpenStacku ako je napríklad práve Nova. Neutron implementuje OpenStack Networking API. Spravuje všetky sieťové aspekty pre virtuálnu sieťovú infraštruktúru a aspekty prístupovej vrstvy fyzickej sieťovej infraštruktúry v prostredí OpenStack. OpenStack Networking umožňuje projektom vytvárať pokročilé topológie virtuálnych sietí, ktoré môžu zahŕňať služby ako firewall a virtuálnu privátnu sieť (VPN). [28]

Keystone

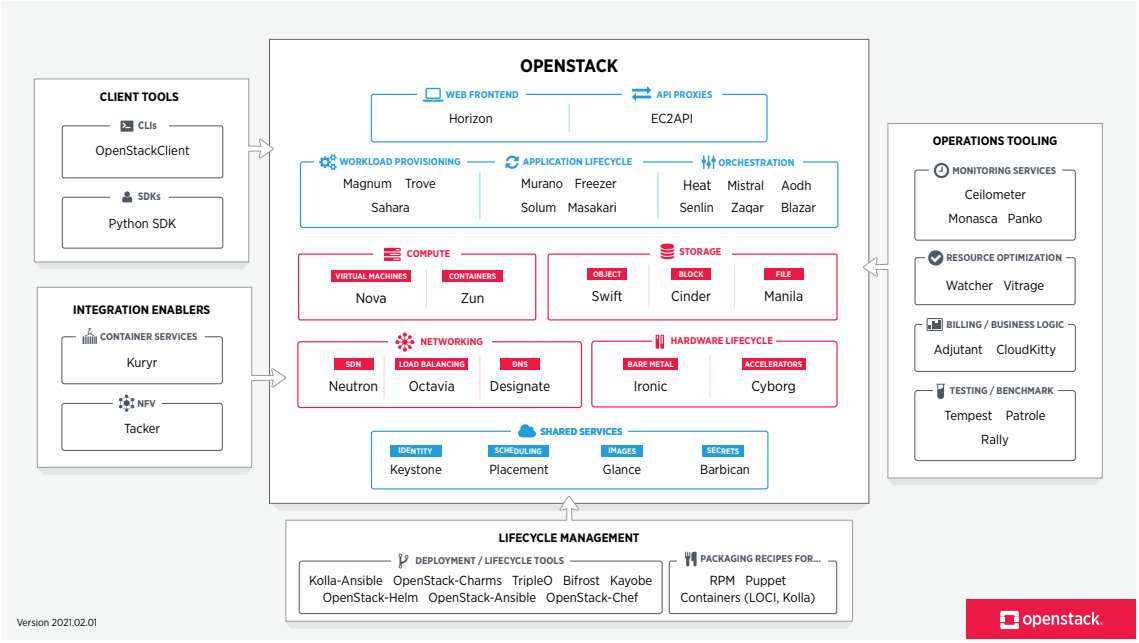
Keystone je služba OpenStacku, ktorá poskytuje API autentifikáciu klienta, zisťovanie služieb a distribuovanú autorizáciu. Ide o spoločný overovací systém v rámci cloudového operačného systému. [28] Keystone sa dá integrovať s adresárovými službami. Podporuje štandardné používateľské meno a heslo ale aj systémy založené na tokenoch.

Heat

Heat je služba na organizovanie viacerých kompozitných cloudových aplikácií. [28] Má za úlohu plánovanie alebo koordináciu prvkov rôznych situácií na dosiahnutie požadovaného cieľa.

7.1.3 Mapa komponentov Openstacku

OpenStack je rozdelený na služby, ktoré nám umožňujú používať cloud computing v závislosti od našich potrieb. Na obrázku číslo 7.1 je uvedená „mapa“ Openstacku. Tá predstavuje akýsi prehľad služby a komponenty Openstacku, aby sme videli, kam sa tieto služby hodia a ako môžu spolupracovať.



Obr. 7.1: „Mapa“ platformy Openstack (Zdroj:<`https://cutt.ly/QYnUNom`>)

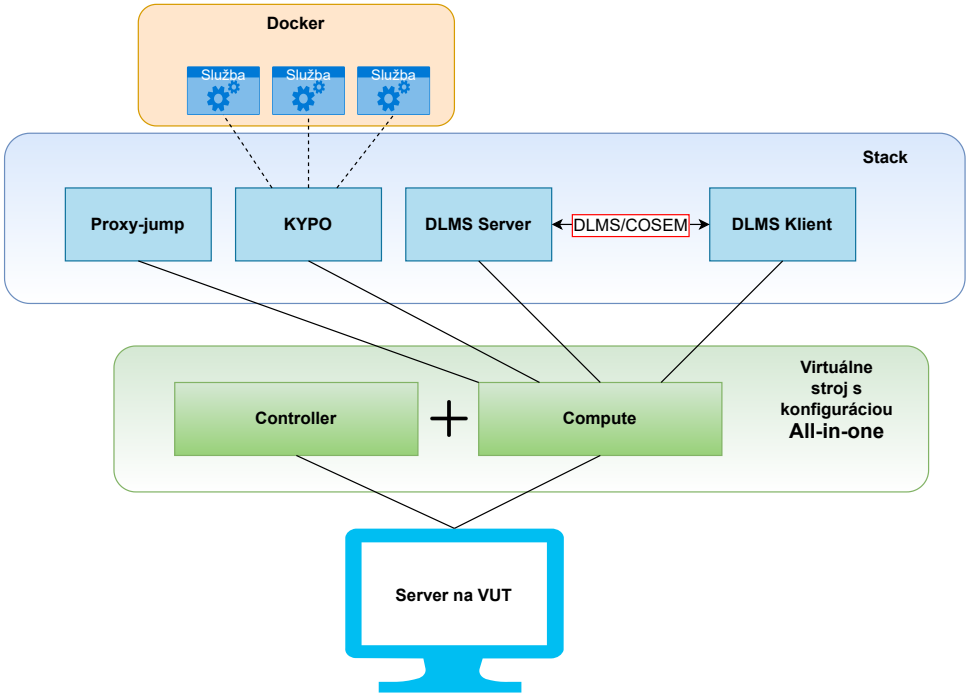
Výzvy pri implementácii

Je nutné si uvedomiť, že OpenStack je komplexná entita, a preto nie je jednoduché ho implementovať a pri implementácii sa stretneme s rôznymi prekážkami. OpenStack je skôr súbor projektov než jeden produkt, a pretože každá z rôznych aplikácií musí byť nakonfigurovaná tak, aby vyhovovala požiadavkám užívateľa. Musíme brať do úvahy aj to, že pri takom veľkom počte komponentov je zložité udržiavať bezchybnú dokumentáciu. Kvôli multiprojektovému vývojovému prístupu OpenStack sa môže stať, že pre zložitosť pri aktualizácii a synchronizácie môže nastať nedostupnosť služieb.

8 Praktická časť

V praktickej časti diplomovej práce bolo úlohou vytvoriť v laboratórnom prostredí platformu KYPO spolu s bezpečnostným scenárom. Tento bezpečnostný scenár, ktorý sa nám podarilo vytvoriť zahŕňa zachytenie komunikácie s využitím protokolu DLMS/COSEM a využitie niekoľkých zraniteľností protokolu DLMS/COSEM. Tento scenár zahŕňa prípravu virtuálneho prostredia na prevedenie útokov pre jednoduchšiu a rýchlejšiu aplikáciu v prípade, že by ju chcel niekto požiť v Cyber Range platforme KYPO. V následujúcej časti tejto práce budú popísané časti potrebné k tomu aby sme mohli daný scenár vytvoriť a komunikáciu simulovať.

Praktická časť práce je vykonávaná na jednom osobnom počítači a inštalačná schéma, ktorá je použitá je zobrazená na obrázku číslo 8.1. Praktický výstup tejto práce bol síce tvorený a ovládaný z osobného počítača ale samotná platforma KYPO a všetky potrebné časti pre jej správnu konfiguráciu a definíciu sú uložené na serveri v priestoroch univerzity VUT v Brne. Z tejto schémy nám vyplýva, že sme pracovali s jedným virtuálnym strojom s prednastavenou konfiguráciou *All-in-one*.



Obr. 8.1: Schéma praktickej časti

Na prevedenie praktickej časti sme si vybrali operačný systém Ubuntu. Ubuntu je kompletná distribúcia operačného systému Linux pre pracovné stanice a servery, založená na linuxovej distribúcii Debian. Držali sme sa odporúčaní od vývojového

tímu z Masarykovej Univerzity a preto bol výber práve tohoto operačného systému nutnosťou. Pre prácu vo virtuálnom prostredí sme zvolili software VMware ale takisto sme pre vzdialený prístup na server použili software PuTTy. Software na prácu vo virtuálnom prostredí sme zvolili len na zásade vlastných preferencií. Na vzdialenú správu serveru sme si zvolili už spomenutý nástroj PuTTy. Na základe pridelenej IP adresy máme možnosť vzdialeného prístupu prostredníctvom protokolu SSH.

8.1 Openstack

Kolla Ansible je realizačný projekt oddelený od projektu Kolla. Kolla Ansible nasadzuje služby a komponenty infraštruktúry OpenStack. Kolla Ansible posiela príkazy na inštaláciu balíčkov. Celú sústava potrebnú na našu prácu si je potreba predstaviť ako na niekoľko virtuálnych vrstiev.

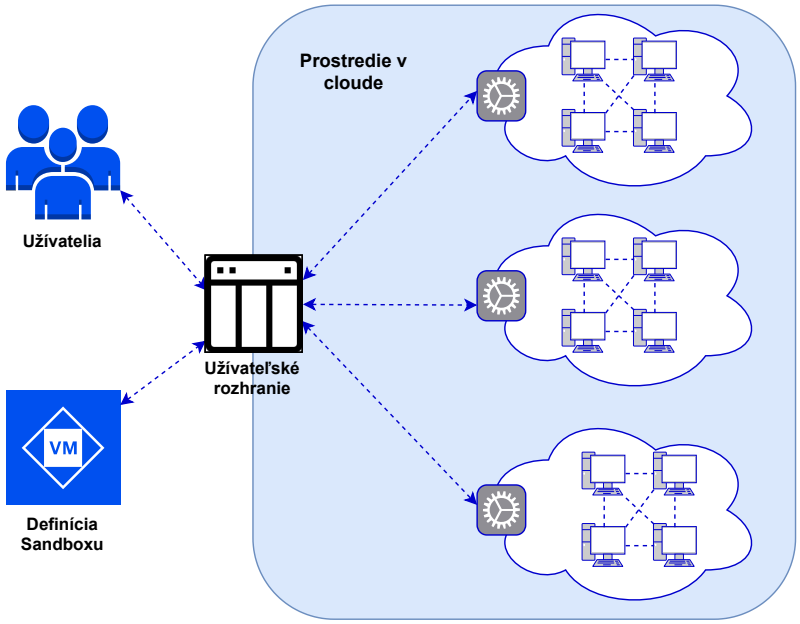
Na našom virtuálnom stroji je nainštalovaný Openstack, čo je vlastne ďalšia vrstva virtualizácie a tam sa vytvorí „zásobník“ služieb. Následne prebehla úprava konfiguračného súboru **globals.yml** a povolenie a definovanie parametrov podľa ktorých sa bude naša platforma chovať. Ďalej tu sú definované IP adresy virtuálnych strojov a parametre projektu Kolla. Sú tu definované aj sieťové rozhrania a IP adresy, ktoré sme museli upraviť podľa našich potrieb.

Engine prostredia KYPO je založený na cloudovej platforme OpenStack. Ten potom riadi veľké skupiny výpočtových, úložných a sieťových zdrojov. Všetky z nich sú spravované prostredníctvom rozhraní API alebo dashboardu. Väčšinou sa nasadzuje ako IaaS vo verejných aj súkromných cloudoch, kde sú používateľom sprístupnené virtuálne servery a ďalšie zdroje.

8.2 KYPO

KYPO ako platforma sa sústreďuje hlavne na dve veci. Na tvorbu simulačných virtuálnych prostredí a na školiace tréningy. Vytváranie a poskytovanie prostredia na simuláciu počítačových infraštruktúr v kontrolovanom cloudovom prostredí s cieľom dosiahnuť vysokú flexibilitu, škálovateľnosť, izoláciu a prenosnosť. Platforma umožňuje vytvárať virtuálne siete s plnohodnotným operačným systémom a sieťovými prvkami, ktoré sú schopné simulovať reálne systémy. Simultánne prebiehajú školenia ako hry kybernetickej bezpečnosti, ktoré sú ešte doplnené hodnotením účastníkov. Ako bolo spomenuté v predchádzajúcej kapitole rola inštruktora vytvorí scenár a účastníci na ňom pracujú a svoje výsledky a pokrok

je možné sledovať v grafoch a tabuľkách. Formát ako platforma funguje môžeme vidieť na obrázku číslo 8.2.



Obr. 8.2: Platforma KYPO

Samotné školenia v KYPO potom prebiehajú v sandboxe, kde účastníci riešia úlohy prezentované v grafickom rozhraní KYPO. Školenie KYPO môže obsahovať aj dotazníky na zber spätnej väzby od účastníkov školenia alebo testy na posúdenie ich vedomostí.

Pre jednoduchšiu interakciu užívateľov KYPO bolo vytvorené grafické používateľské rozhranie. V tomto prostredí je jednoduchý prístup k sandboxom a ďalším funkciám. Predstavuje sprostredkovateľa medzi používateľmi a mikroslužbami, ktoré bežia na pozadí. Celému grafickému rozhraniu hovoríme KYPO portál. Portál je rozdelený do troch okruhov:

1. **Okruh sandboxov** - špecifikuje pokyny pre vytváranie a realizáciu sandboxov.
2. **Okruh školení** - hlavné zameranie na tvorbu a organizáciu školení.
3. **Administračný okruh** - zaoberá sa správou používateľov a ich prístupom do špecifických častí Portálu KYPO na základe prístupových rolí používateľov.

Prístup k portálu KYPO do veľkej miery závisí od role, ktorú ako používateľ máme. Tieto role nám určujú kam môžeme pristupovať a k akým rôznym stránkam agend máme prístup a aké funkcie tam môžeme vykonávať. Portál rozlišuje tri základné role používateľov:

- **Stážisti** - používatelia, ktorí sa zúčastnia školenia a majú len základný prístup potrebný pre vykonávanie úloh.
- **Inštruktori** - používatelia, ktorí sú zodpovední za prípravu a vytváranie školení a príslušných sandboxov.
- **Administrátori** - používatelia, ktorí sú zodpovední za správu celej inštancie KYPO platformy. Môžu mať právu aj na na správu užívateľov.

8.3 Postup riešenia

V tejto časti práce je popísaný základný a stručný postup spolu s prekážkami, s ktorými sme sa stretli pri spustení celej Cyber Range platormy KYPO. Inštalácia a samotné spustenie je veľmi náročné a samotný postup, ktorý je v tejto kapitole popísaný už o niekoľko týždňov alebo aj dní nemusí byť aktuálny pretože jednotlivé balíčky, z ktorých sa celá inštalácia skladá sa rýchlo menia a aktualizujú na nové verzie. Tento postup má čitateľovi priblížiť výzvy s ktorými sa môže užívateľ stretnúť a mal by mu pomôcť pochopiť jednotlivé kroky.

8.3.1 Inštalácia základných balíčkov

Ako prvú vec, ktorú musíme urobiť je priradiť si výpočtové a pamäťové prostriedky pre náš prístup na server. Pre správu virtuálneho stroja používame VMware ESXi. VMware ESXi je hypervízor nezávislý od operačného systému založený na operačnom systéme VMkernel, ktorý spolupracuje s agentmi, ktorí bežia nad ním. ESXi je skratka pre Elastic Sky X Integrated. ESXi je hypervízor typu 1, čo znamená, že beží priamo na systémovom hardvéri bez potreby operačného systému. Prostredie na správu virtuálneho stroja je na obrázku číslo 8.3. Parametre, ktoré sme zvolili my pre náš virtuálny stroj je 10 procesorov, 48 GB pamäte RAM a 200 GB miesta na pevnom disku. Pri tomto kroku by som rád spomenul, že je dôležité naozaj dobre zvoliť výpočtové a pamäťové parametre virtuálneho stroja a skontrolovať si minimálne požiadavky Cyber Range platformy.

Po zadaní parametrov pre tvorbu virtuálneho stroja sme pristúpili k inštalácii operačného systému. Podľa odporúčaní v manuáli od Masarykovej Univerzity sme vybrali operačný systém Ubuntu. Pracovali sme ešte so staršou verziou OS Ubuntu 20.04. Po tomto kroku už nasleduje inštalácia a aktualizácia potrebných balíčkov a softwarového vybavenia.

Obr. 8.3: VMware ESXi

Spustenie platformy Openstack

Na začiatok sme aktualizovali všetky stávajúce balíčky v Ubuntu aby sme pracovali s najaktuálnejšou verziou, ktorá bola momentálne dostupná. Po tomto kroku sme už mohli prejsť k inštalácii balíčkov ako sú „python“, inštalačný nástroj „pip“ a správnu kompatibilnú verziu „Ansible“ a „Kolla Ansible“. Potom sme vytvorili adresár pre Kolla pretože ten sa automaticky nevytvára a skopírovali do neho ostatné súbory, ktoré sa vytvorili pri inštalácii Kolla Ansible, Základnú inštaláciu a prípravu sme urobili príkazmi v tomto poradí.

```
1  # sudo apt update
2  # sudo apt install python3-dev libffi-dev gcc libssl-dev
3  # sudo apt install python3-venv
4  # pip install 'ansible==5.*'
5  # pip3 install kolla-ansible
6  # kolla-ansible install-deps
```

Pri inštalácii Kolla a Kolla ansible sme sa stretávali s častou nekompatibilitou pri nasadzovaní platformy KYPO a preto je nutné si naštudovať dokumentáciu ku platforme KYPO pretože nie všetky komponenty sú spätne kompatibilné s najnovšími verziami softwaru. Potom sme už iba doplnili ďalší balíček „Ansible Galaxy“ a prešli sme na úpravu dôležitého konfiguračného súboru **globals.yml**. Naša definícia súboru globals.yml je zobrazená na obrázku číslo 8.4. Tu ešte chýba zbytok súboru a preto uvádzame ešte ďalšie nastavenia, ktoré sa týkajú sieťových rozhraní a použitej konzole na prístup v platforme KYPO.

```
1  network_interface: "ens37"
2  neutron_external_interface: "ens36"
3  nova_console: "spice"
```

Keďže používame konfiguráciu, ktorá je pripravená pre spustenie celého projektu na jednom stroji tak nie je nutné robiť úpravy na konfiguračnom súbore all-in-one.yml, tu už je všetko pripravené a nám ho stačí len použiť.

```
# Valid options are ['centos', 'debian', 'rhel', 'ubuntu']
kolla_base_distro: "ubuntu"

# Valid options are [ binary, source ]
kolla_install_type: "source"

# Do not override this unless you know what you are doing.
#openstack_release: "xena"

# Docker image tag used by default.
#openstack_tag: "{{ openstack_release ~ openstack_tag_suffix }}"

# Suffix applied to openstack_release to generate openstack_tag.
#openstack_tag_suffix: ""

# Location of configuration overrides
#node_custom_config: "/etc/kolla/config"

# This should be a VIP, an unused IP on your network that will float between
# the hosts running keepalived for high-availability. If you want to run an
# All-In-One without haproxy and keepalived, you can set enable_haproxy to no
# in "OpenStack options" section, and set this value to the IP of your
# 'network_interface' as set in the Networking section below.
kolla_internal_vip_address: "192.168.1.209"
```

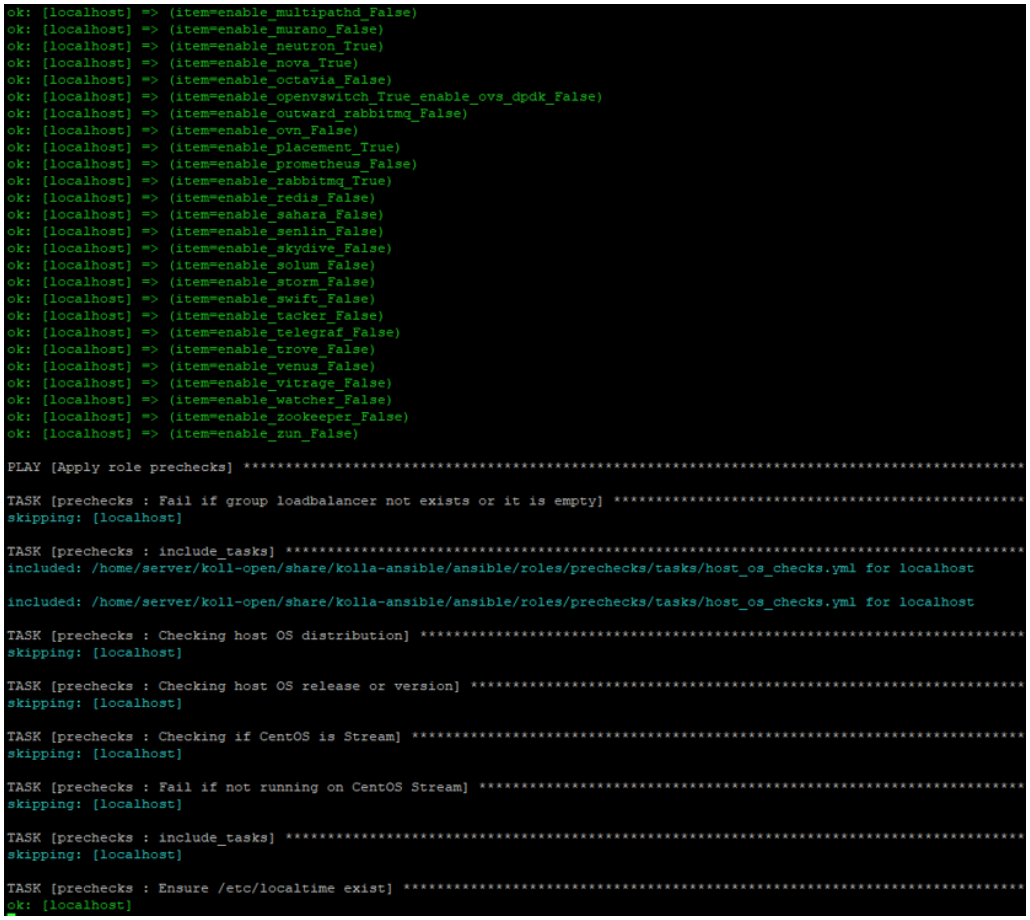
Obr. 8.4: Konfiguračný súbor globals.yml

Po vyplnení konfiguračných súborov sme už generovali heslá s rolami užívateľov a naplnili nimi súbor **passwords.yml**. Pri generovaní hesiel a užívateľov nesmieme zabudnúť na zaškrtnutie políčka „Unrestricted“ pre zabezpečenie neobmedzeného prístupu. Nasadenie platformy Openstack prebieha v 4 krokoch. Patrí tu napríklad kontrola požiadaviek, potrebných kontajnerov a konfiguračných súborov pred nasadením a tá istá kontrola prebieha po zadaní príkazu na nasadenie. Poradie príkazov na nasadenie platformy OpenStack, tak aby všetky kroky prebehli vporiadku je uvedený nižšie.

```
1  # kolla-genpwd
2  # kolla-ansible -i all-in-one bootstrap-servers
3  # kolla-ansible -i all-in-one prechecks
4  # kolla-ansible -i all-in-one deploy
5  # kolla-ansible -i all-in-one post-deploy
```

To ako môže taká kontrola prebiehať môžeme vidieť napríklad na obrázku číslo 8.5, kde je zobrazené ako prebieha kontrola požiadaviek pred nasadením platformy. Pokiaľ všetky tieto kroky prebehnú úspešne tak sa vytvorí aj nový súbor s prístupovými údajmi na platformu Openstack. Pri nasadzovaní platformy boli sa vyskytovali najväčšie problémy a bolo náročné ich odstrániť bola nevyhnutná

práca s vyhľadávaním na webových stránkach Google alebo komunikácia s podporou Masarykovej Univerzity a ich platformy.



Obr. 8.5: Kontrola pred nasadením (prechecks)

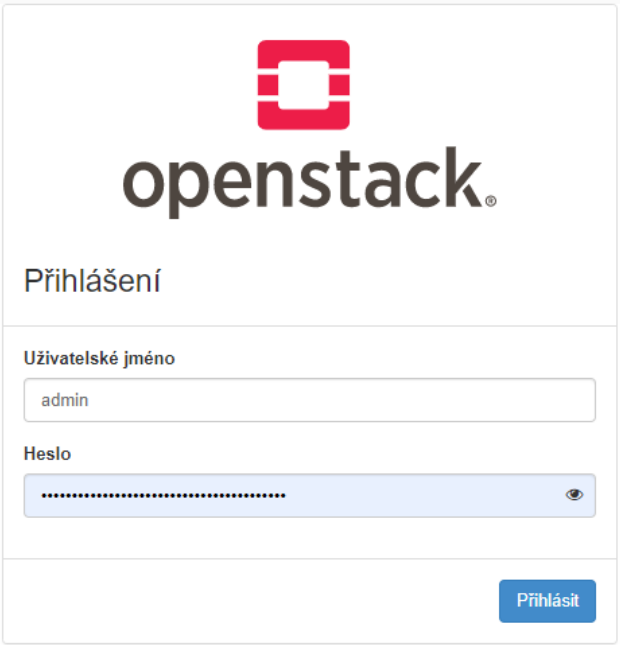
Teraz sme pristúpili k stiahnutiu a pripojeniu podporovaných obrazov na platformu, ktoré nám poslúžia neskôr v platforme KYPO. Pre príklad je uvedený len jeden prípad stiahnutia a pripojenia obrazu.

```
1  # wget https://cloud-images.ubuntu.com/bionic/current/
2    bionic-server-cloudimg-amd64.img -P /tmp/
3  # openstack image create --disk-format qcow2 --container
4   -format bare --public --property os_type=linux --file
5   /tmp/bionic-server-cloudimg-amd64.img ubuntu-bionic-x86_64
```

Nakoniec sme pridali ešte flavors (typy) týchto vytvorených obrazov a prešli sme k vytvoreniu externej siete a poolu adries pre klientov Openstacku.

```
# openstack flavor create --id 1 --ram 16384 --disk 30
  --vcpus 4 standard.large
```

Úvodná obrazovka prihlásenia funkčnej platformy Openstack je na obrázku číslo 8.6.



Obr. 8.6: Úvodná obrazovka platformy Openstack

8.3.2 Základná konfigurácia KYPO

Najskôr sa pripojíme cez webové prostredie na platformu Openstack a následne si na nej vygenerujeme súbor s definovanými rolami pre platformu. Tento súbor zahrnuje role admin, member, reader a heat_stack_owner. Generovanie a stiahnutie súboru sme spravili v prostredí Horizon. Potom je potrebné nainštalovať openssh a nástroj pipenv pre prácu s virtuálnym prostredím. Ďalším krokom je stiahnutie aktuálneho repozitáru zo služby Git pre nasadenie platformy KYPO.

```
# sudo apt install python3-pip openssh-client jq
# sudo pip3 install pipenv
# source app-cred-kypo-openrc.sh
# git clone https://gitlab.ics.muni.cz/muni-kypo-crp/
  devops/kypo-crp-openstack-base.git
# pipenv install
# pipenv shell
```

Po stiahnutí a nasadení repozitáru do nášho stroja musíme upraviť súbor **openstack-defaults.sh** a súbor **extra-vars.yml**. Ten nám pomáha dodatočne nastaviť niekoľko dôležitých parametrov. Súbor extra-vars.yml napĺňame hodnotami zo stiahnutého súboru app-cred-kypo-openrc.sh z prostredia Horizon. Súbor s vyplnenými hodnotami je na obrázku číslo 8.7.

```
# The FQDN or IP address of KYPO CRP.
kypo_crp_host: 172.24.0.19

# The prefix of the sandbox in the OpenStack cloud.
kypo_crp_instance_name: default0

# The maximum transmission unit for KYPO services.
kypo_crp_docker_network_mtu: 1442

# The URL of OpenStack Identity service API.
kypo_crp_os_auth_url: http://192.168.1.209:5000

# The ID of application credentials to authenticate at the OpenStack cloud platform.
kypo_crp_os_application_credential_id: b392f62ceab944729cd23a0153a858e9

# The secret string of `kypo_crp_os_application_credential_id`.
kypo_crp_os_application_credential_secret: vfh0lQ5Pj7CKTlQvOIuXrV9V3rpU125BMsJuvNdmVC489Fe14-t-zqwI4eCKdXrZwpw9qnkrQyRkTMHhOCiO8g

# The KYPO Jump host IP address or hostname.
kypo_crp_proxy_host: 172.24.0.14

# The name of the user on the KYPO Jump host.
kypo_crp_proxy_user: ubuntu

# The list of IP addresses to custom DNS servers.
kypo_crp_dns:
    - 1.1.1.1
    - 1.0.0.1

# The OpenStack console type. One of: novnc, spice-html5
kypo_crp_os_console_type: spice-html5
```

Obr. 8.7: Konfiguračný súbor extra-vars.yml

Potom môžeme pristúpiť k vytvoreniu základnej infraštruktúry KYPO a overeniu dostupnosti nami vytvorených staníc *kypo-head* a *kypo-proxy-jump*.

```
# ./bootstrap.sh public      //tvorba
# ./create-base.sh
# ./ansible-check-base.sh    //overenie
# ./ansible-user-access.sh
```

Nasleduje generovanie SSL certifikátu a jeho súkromného kľúča, aby sme následne mohli pomocou dátového formátu base64 zakódovať tento certifikát s kľúčom a SSH kľúč pre prístup k stanici KYPO Proxy.

```
# openssl req -nodes -new -x509 -keyout kypo.key -out
  kypo.crt -subj /C=AU/ST=Some-State/O=Internet Widgits
  Pty Ltd/CN=172.24.0.28 -addext subjectAltName =
  DNS:172.24.0.28, IP:172.24.0.28
# base64 kypo.crt
# base64 kypo.key
# base64 ../kypo-crp-openstack-base/admin_kypo-base-key.key
```

Zakódované certifikáty a kľúče presunieme do súboru **secrets.yml**. Potom vytvoríme súbor **inventory.ini** a vložíme do neho adresu stanice *kypo-head* a zdroj kde sa nachádza vygenerovaný SSH kľúč pre prístup na túto stanicu. Pre ďalšie pokračovanie je opäť nutná úprava, v tomto prípade je to úprava systémovej premennej s názvom ANSIBLE_ROLES_PATH. Po úprave ďalších súborov nasleduje ešte niekoľko krokov vedúcich k finálnemu spusteniu playbooku, obsahujúci konfiguráciu virtuálnych strojov. Opäť tu prebieha inštalácia nástrojov a vytvorenie CSIRT-MU dummy OIDC issuer, ktorý má na starosti pridelenie oprávnení v platforme KYPO.

```
# ansible-galaxy collection install community.docker
# ansible-galaxy collection install community.postgresql
# ansible-galaxy install -r provisioning/requirements.yml
  -p provisioning/roles_required
# ansible-playbook -i inventory.ini provisioning/docker.yml
  --extra-vars=@extra-vars.yml --extra-vars=@secrets.yml
```

Po tomto kroku sme prešli k tvorbe sandboxovej definície, ktorej predchádza tvorba topológie siete.

8.3.3 Topológia siete sandboxu

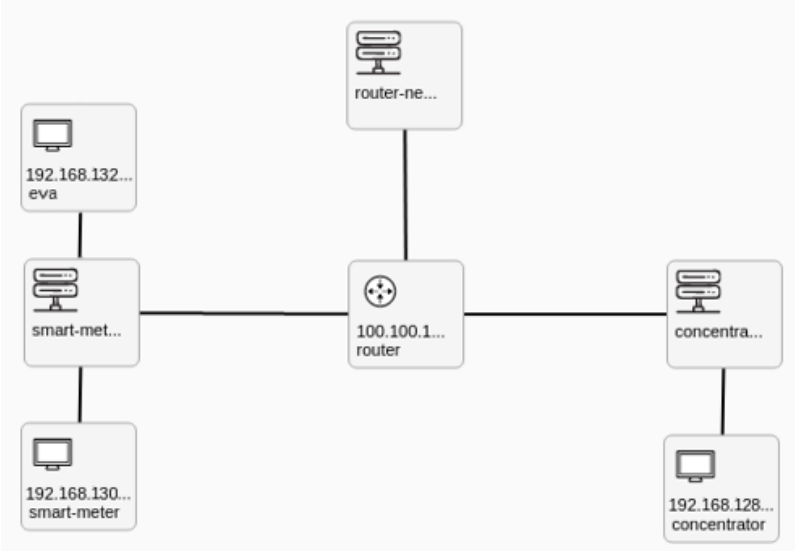
Topológia siete sandboxu je opäť tvorená vo formáte súboru YAML (Yet Another Markup Language). Niekoľko vzorových príkladov takýchto topológii sa nachádza na gitlabe Masarykovej Univerzity a ich Wikipédii. Týmito vzorovými príkladmi sme sa inšpirovali aj pre túto diplomovú prácu a skombinovali a upravili sme si definície topológii tak aby vyhovovali naším potrebám. Úpravu definícií sme robili v jednoduchom textovom editore, ktorý je súčasťou operačného systému hosťovského počítača. Vzor našej definície topológie môžete vidieť na obrázku číslo 8.8. Graficky zobrazená topológia je zase na obrázku číslo 8.9.

V našej topológii virtuálny stroj „eva“ predstavuje škodlivého útočníka. Tento útočník je pre náš scenár dôležitý aby sme prostredníctvom neho mohli uskutočňovať

```
name: xtomko04

hosts:
  - name: smart-meter
    base_box:
      image: ubuntu-focal-x86_64
      man_user: ubuntu
    flavor: csirtmu.tiny1x2

  - name: eva
    base_box:
      image: debian-9-x86_64
    flavor: csirtmu.tiny1x2
    extra:
      memory: 512

  - name: concentrator
    base_box:
      image: ubuntu-focal-x86_64
      man_user: ubuntu
    flavor: csirtmu.tiny1x2

routers:
  - name: router
    base_box:
      image: debian-9-x86_64
      man_user: debian
    flavor: csirtmu.tiny1x2
```

```
networks:
  - name: smart-meter-switch
    cidr: 10.10.20.0/24

  - name: concentrator-switch
    cidr: 10.10.30.0/24

net_mappings:
    - host: smart-meter
      network: smart-meter-switch
      ip: 10.10.20.5

    - host: eva
      network: smart-meter-switch
      ip: 10.10.20.6

    - host: concentrator
      network: concentrator-switch
      ip: 10.10.30.5

router_mappings:
    - router: router
      network: smart-meter-switch
      ip: 10.10.20.1

    - router: router
      network: concentrator-switch
      ip: 10.10.30.1

groups: []
```

Obr. 8.8: Definícia topológie siete



Obr. 8.9: Graficky interpretovaná topológie siete

útoky v tréningovom scenári a využiť tak zraniteľnosti protokolu DLMS/COSEM. PO definovaní sieťovej topológie sme prešli k tvorbe sandboxov.

8.3.4 Vytvorenie sandboxov

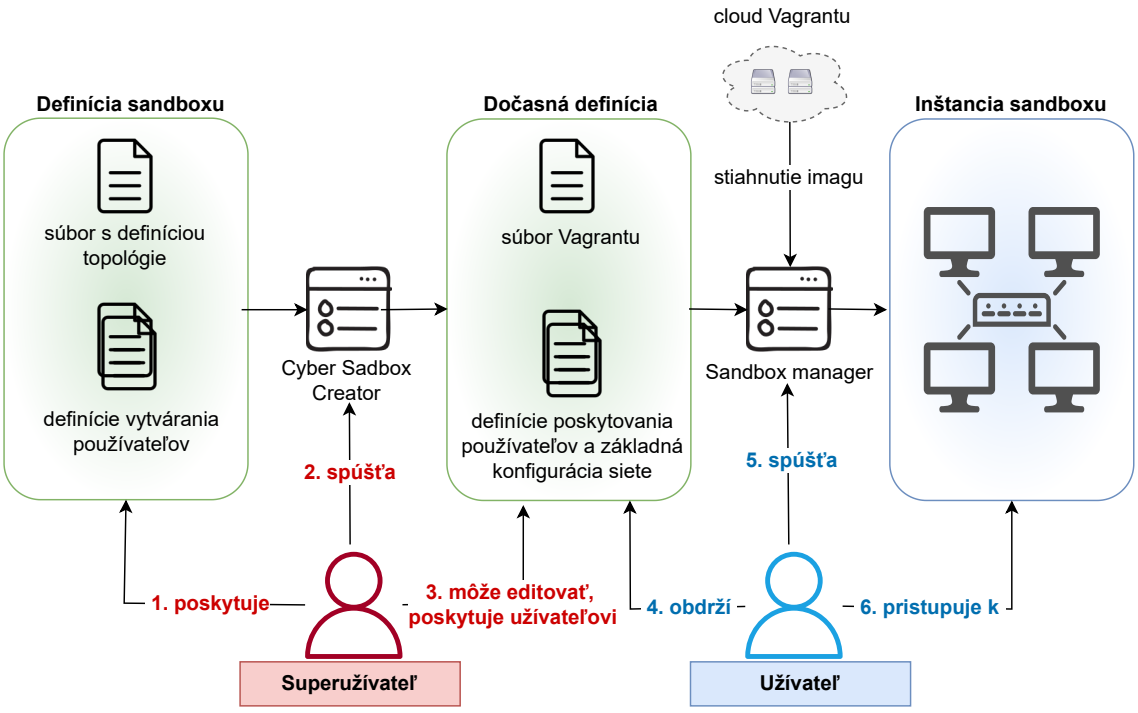
Na vytvorenie sandboxov sme si zvolili Cyber Sandbox Creator (CSC). Momentálne Masarykova univerzita zverejnila najnovšiu verziu a tú používame. Je to v poradí už tretia verzia. Cyber Sandbox Creator je nástroj, ktorý dokáže generovať prenosné definičné súbory a vytvárať virtuálne prostredia pomocou VirtualBoxu, Vagrantu a Ansible z jednoduchej definície topológie YAML. Kombinácia týchto nástrojov umožňuje vytvárať virtuálne počítače prepojené s virtuálnymi sieťami s minimálnym úsilím, a to aj na osobnom počítači.

Pre správne fungovanie a inštaláciu CSC sme museli náš počítač pripraviť podľa postupu, ktorý je dostupný na platforme GitLab spravovaný Masarykovou univerzitou. GitLab je platforma DevOps, ktorá sa dodáva ako jedna aplikácia. Vďaka tomu sa môže GitLab používať na efektívnejší softvérový pracovný postup, ktorý organizácie oslobodí od obmedzení spojených s reťazcom nástrojov.

Prvý použitý nástroj je VirtualBox, ten má veľmi podobnú funkciu ako nami už skôr zvolený VMware. VirtualBox je multiplatformový virtualizačný nástroj distribuovaný pre všetky operačné systémy. Druhý potrebný nástroj je nástroj Vagrant. Vagrant je nástroj na vytváranie a správu prostredí virtuálnych počítačov v rámci jedného pracovného postupu. Vďaka ľahko použiteľnému pracovnému postupu a zameraniu na automatizáciu rovnako znižuje čas potrebný na nastavenie vývojového prostredia. Ako posledný nástroj pre správnu funkciu CSC sme použili nástroj Ansible. Ansible je voľne dostupný softvér, ktorý nám pomáha s tvorbou platformy pre konfiguračnú správu a riadenie počítačov, vykonávanie úloh a správu konfigurácií. Ansible spravuje počítače v sieti pomocou SSH alebo cez PowerShell. Zároveň má aj minimálne nároky na nainštalovaný softvér, na linuxových uzloch mu stačí Python verzie 2.4 alebo vyššej, a na uzloch s MS Windows s PowerShell verziou 3.0 alebo vyššou. To ako spolu jednotlivé komponenty komunikujú a spolupracujú môžeme vidieť na obrázku číslo 8.10. CSC sa skladá z dvoch skriptov: *Sandbox Creator* a *Sandbox Manager*. Oba sú súčasťou dvoj krokového procesu vytvárania inštancie virtuálneho laboratória (inštancie sandboxu). Tieto dva kroky často vykonávajú rôzni používatelia.

Pre spätnú kompatibilitu s CSC a ostatnými nástrojmi je nutné aby aj ostatné nástroje dosahovali určitých minimálnych verzií. Krátky výpis nejmenších možných verzií by mal poskytnúť stručný prehľad:

1. Python >= 3.7
2. VirtualBox >= 6.1
3. Vagrant >= 2.2.5
4. Ansible >= 2.5



Obr. 8.10: Schéma Cyber Sandbox Creator

Nástroj Sandbox Creator

Ako bolo už spomenuté vyššie tak CSC sa skladá z dvoch skriptov a tie sú súčasťou procesu vytvárania inštancie sandboxu. Pod pojmom definícia sandboxu si môžeme predstaviť skupinu súborov, ktoré definujú obsah virtuálneho prostredia. Tieto súbory sú navrhnuté tak, aby sa dali ľahko upravovať a umožňovali prispôsobenie prostredia. [29]

Nástroj Sandbox Creator sa používa v prvom kroku, v ktorom sa definícia sandboxu transformuje na dočasnú definíciu. Priebežná definícia je skupina súborov, ktoré možno priamo použiť na vytvorenie inštancie virtuálneho prostredia. Tieto súbory nie sú určené na úpravu. Možno ich distribuovať na vytvorenie mnohých inštancií virtuálneho prostredia na rôznych počítačoch. Sandbox Manager sa používa na vytvorenie jednej inštancie virtuálneho prostredia (inštancie sandboxu) z dočasnej definície.

Definícia Sandboxov

Rovnaké súbory s definíciou topológie možno použiť na vytvorenie sandboxov v cloudových prostrediach pomocou platformy KYPO Cyber Range Platform. Keďže cloudové prostredia potrebujú trochu odlišnú sadu argumentov, niektoré z nich

CSC ignoruje a niektoré používa len CSC. Radi by sme spomenuli pár dôležitých argumentov a ich krátke vysvetlenie, ktoré sa v definícii sandboxov používajú. Zoznam tých najvýznamnejších je uvedený v tabuľke číslo 8.1.

Tab. 8.1: Argumenty sandboxov

Názov argumentu	**Význam argumentu**
name	Skrátený názov topológie (povinné).
base_box:image	OS, ktorý bude nainštalovaný na počítači (povinné).
base_box:mgmt_user	Používateľ pre správu obrazu OS (nepovinné).
base_box:mgmt_protocol	Komunikačný protokol ssh alebo winrm (nepovinné).
extra, (*cpus*, *memory*)	Špeciálne atribúty (nepovinné).
flavor	Príchuť, definícia pamäte a procesorov (povinné).
routers	Zoznam smerovačov (povinné, môže byť prázdne).
net_mappings	Mapovanie počítačov na sieť (povinné, môže byť prázdne).
host	Názov existujúceho hostiteľa (povinné).
network	Názov existujúcej siete (povinné).
ip	IP adresa hostiteľa v sieti (povinné).
groups	Zoznam ďalších skupín pre ansible (povinné).
nodes	Zoznam názvov zariadení v skupine (povinné).

V tabuľke je spomenutý aj výraz „flavor“, ktorý môžeme doslovne preložiť ako príchuť. Flavor poskytujú rýchly spôsob výberu hardvérových špecifikácií (napríklad počtu procesorov a pamäte) pre virtuálny počítač. Tieto atribúty možno špecifikovať aj samostatne pomocou ďalších atribútov memory (pamäť) a cpus (procesory). Hodnoty pamäte alebo procesorov sú vždy nadradené hodnotám uvedeným vo flavor. Flavor sú vždy potrebné, pretože sú jediným prostriedkom na špecifikáciu pamäte a počtu procesorov v cloudových prostrediach. CSC má na tento účel flexibilnejšie atribúty, takže flavor možno považovať za akýsi núdzový variant, ak je sandbox vytvorený práve v cloude. Cyber sandbox creator podporuje niekoľko typov flavor do veľkosti pamäte približne až 66 GB a maximálny počet procesorov je 16.

Generovanie sandboxu

Po tom ako si zadefinujeme ako bude náš sandbox vyzerať, respektíve po tom ako vytvoríme sieťovú konfiguráciu tak nasleduje krok generovania sandboxu. Na to potrebujeme najskôr vytvoriť „dočasnú definíciu“. Dočasná definícia je výstupom nástroja Cyber Sandbox Creator a slúži ako vstup do nástroja Sandbox Manager,

ktorý zostavuje a konfiguruje virtuálne počítače. Vygenerovanie dočasnej definície spravíme prostredníctvom príkazového riadku na hosťovskom počítači s využitím príkazu create-sandbox. Tento príkaz vygeneruje dočasnú definíciu v adresári sandboxu na rovnakom mieste ako sa nachádza definícia topológie. Ak adresár neexistuje, vytvorí sa.

Sandbox Manager je skript príkazového riadka, ktorý riadi proces vytvárania sandboxu. Po úspešnej inštalácii CSC je správca pieskoviska prístupný z ľubovoľného miesta. Ďalej na správu sandboxu používame aj nástroj pre prácu s virtuálnymi prostrediami, Vagrant. Sandbox Manager nám umožňuje základné operácie s vytvorenými virtuálnymi prostrediami. Patrí sem napríklad vytvorenie, zničenie alebo vypnutie inštancie sandboxu, jeho uspanie a reštart alebo obnovenie do predchádzajúcich nastavení. Po úspešnom vytvorení inštancií sa na jednotlivé virtuálne stroje prihlasuje protokolom SSH a môžeme ich ďalej ovládať rovnako z príkazového riadku.

8.4 Príprava virtuálneho prostredia

8.4.1 Playbook pre Ansible

Software Ansible, ktorý bol už v predchádzajúcej časti raz spomenutý je jedným z najpoužívanejších nástrojov na správu cloudovej a lokálnej infraštruktúry. Slúži ako flexibilný a výkonný nástroj na automatizáciu úloh správy a konfigurácie infraštruktúry. Ansible je primárne určený správcom IT infraštruktúry pre jednoduchú správu ekosystému. Ansible používa koncepty riadiacich a spravovaných uzlov. Z riadiaceho uzla sa pripája k riadeným uzlom a posiela im príkazy a inštrukcie.

Jednotky kódu, ktoré Ansible vykonáva na spravovaných uzloch, sa nazývajú moduly. Každý modul je vyvolaný úlohou a usporiadaný zoznam úloh spolu tvorí **playbook**. Následne my, ako používatelia, píšeme playbooky s úlohami a modulmi, aby sme definovali požadovaný stav systému. Ansible využíva veľmi jednoduchý jazyk YAML na definovanie playbookov v ľudsky čitateľnom dátovom formáte. Ansible nevyžaduje inštaláciu žiadnych ďalších agentov na spravované uzly. Jediné, čo používateľ potrebuje, je terminál na vykonávanie príkazov Ansible a textový editor na definovanie konfiguračných súborov.

Naše virtuálne prostredie sme si teda prispôsobili a pripravili na kybernetické útoky pomocou správneho definovania playbooku. Zadaný adresár na úpravu virtuálneho prostredia musí obsahovať súbor playbook.yml. Súbor playbook.yml je hlavným playbookom poskytovania Ansible a odovzdáva sa Ansible po vytvorení

inštancie virtuálneho počítača. Poskytovanie sa môže vykonávať celé z uvedeného súboru, alebo môže byť rozdelené do viacerých súborov, prípadne rolí ansible zahrnutých v súbore playbook.yml.

Playbooky Ansible sú textové súbory YAML obsahujúce konfiguráciu virtuálnych strojov. Každý playbook sa skladá z jednej alebo viacerých úloh a zároveň môže obsahovať niekoľko atribútov, ktoré môžeme ľubovoľne kombinovať a priraďovať. Playbooky sú najjednoduchším spôsobom v systéme Ansible na automatizáciu opakujúcich sa úloh vo forme opakovane použiteľných a konzistentných konfiguračných súborov. V playbooku musia mať dátové prvky na rovnakej úrovni rovnaké odsadenie, zatiaľ čo prvky, ktoré sú deťmi iných prvkov, musia byť odsadené viac ako ich rodičia.

Inštalácia balíka je jednou z najbežnejších operácií. Inštalácie balíkov sme využili aj my v našej práci pre prípravu virtuálneho prostredia na kybernetické útoky. Ukážka nášho vytvoreného playbooku pre virtuálne prostredie na kybernetické útoky môžeme vidieť na obrázku číslo 8.11.

```
- name: Ansible apt module examples
  hosts: all
  become: true
  tasks:
   - name: Ansible Update Cache and Upgrade all Packages
     register: updatesys
     apt:
       name: "*"
       state: latest
       update_cache: yes

- name: Playbook s prikazem dsniff
  hosts: eva
  become: yes
  become_user: root
  tasks:
   - command: 'apt-get install dsniff -y'

- name: Playbook to set ipforward
  hosts: eva
  become: yes
  become_user: root
  tasks:
   - command: 'sysctl -w net.ipv4.ip_forward=1'
```

```
- name: Playbook to install HPING3-DDOS
  hosts: eva
  become: yes
  become_user: root
  tasks:
   - name: Ansible apt install hping3
     register: updatesys
     apt:
       update_cache: yes
       name:
         - hping3
       state: present

- name: Playbook to install TCPDUMP-REPLY multiple packeges
  hosts: eva
  become: yes
  become_user: root
  tasks:
   - name: Ansible apt to install multiple packages - LAMP - REPLY
     register: updatesys
     apt:
       update_cache: yes
       name:
         - tcpdump
         - tcpreplay
         - bittwist
       state: present
```

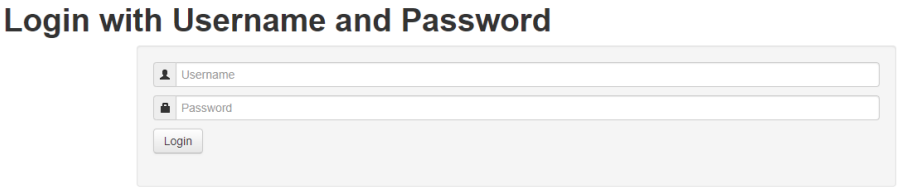
Obr. 8.11: Playbook

Ukážka tejto časti playbooku sa vzťahuje len na prípravu virtuálneho prostredia pre kybernetické útoky a nie je to kompletný playbook pre funkčnú platformu. V kompletnej verzii playbooku sa nachádzajú ďalšie pravidlá a úlohy spolu s definovaním rozsahu IP adries a nastaveniu názvov sieťovým rozhraniam. Ďalej sa tu nachádza nastavenie virtuálnych strojov ako chytrého elektromeru a koncentrátora. Nachádza sa tu aj definovanie klonovania repozitáru z Githubu

komponentu Gurux. Komponent Gurux dlms určuje metódy, ktoré umožňujú komunikáciu pomocou protokolu DLMS.

8.5 Testovací scenár

Posledným krokom celej prípravy je uviesť celý scenár do fungujúceho stavu tréningového prostredia slúžiaceho na precvičenie si kybernetických útokov a zneužitia zraniteľností protokolu DLMS/COSEM. Na tento krok sa musíme prihlásiť do KYPO portálu, ten sa nachádza na adrese kypo head inštancie, ktorú nám pridelil Openstack v portáli Horizon. Zároveň nesmieme zabudnúť tejto IP adrese povoliť riziko certifikátu a následne sa prihlásiť našimi údajmi, ktoré sme si zvolili a vložili do súboru **extra-vars.yml**. Úvodnú obrazovku prihlásenia do KYPO portálu môžete vidieť na obrázku číslo 8.12.



Obr. 8.12: Úvodná obrazovka prihlásenia KYPO portálu

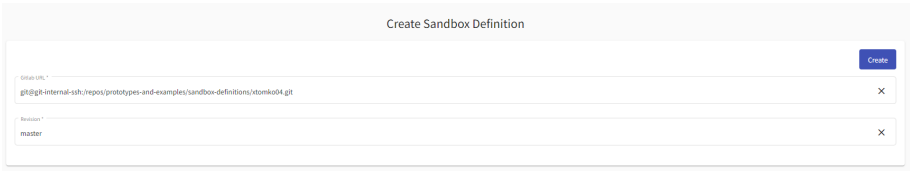
Na samotné spustenie testovacieho scenáru je potreba:

- prihlásiť sa ako admin,
- vloženie sandboxu,
- vytvorenie definície sandboxu cez náš GitHub repozitár,
- alokácia sandboxu a jeho poolu,
- a nakoniec stiahnutie súboru pre SSH prístup a jeho rozbalenie.

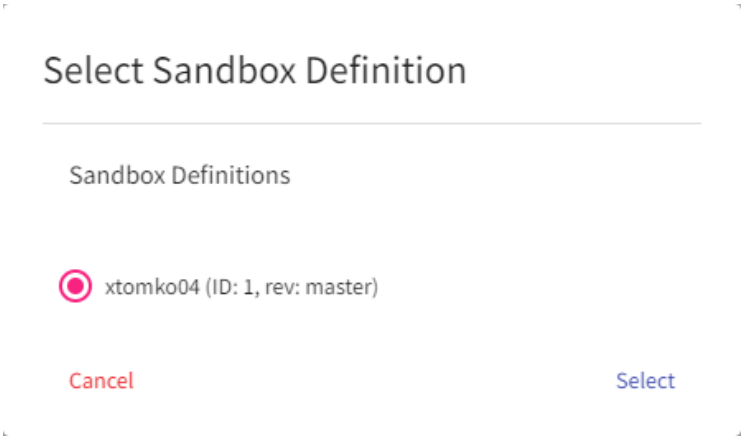
Na vloženie a alokovanie sandbox definície pracujeme s kontajnerom, ktorý hostuje interný repozitár KYPO a tam nahrávame svoju vytvorenú sandbox definíciu. Tú najskôr nahráme na svoj vytvorený verejne dostupný git repozitár a potom ju skopírujeme do interného gitu KYPO. Vloženie a alokovanie tohoto sandboxu môžeme vidieť na obrázku číslo 8.13 respektíve na obrázku číslo 8.14.

Ďalším krokom po alokácii sandox definície je na rade alokácia poolu pre sandbox deiníciu. Tá sa skladá z troch krokov:

1. OpenStack Stage,
2. Networking Ansible Stage,
3. User Ansible Stage.

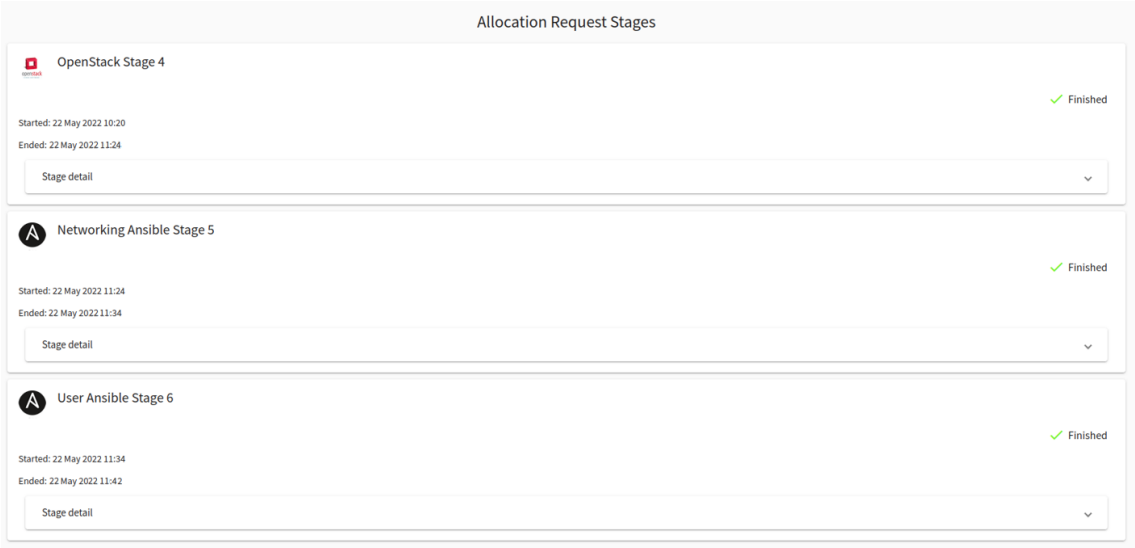


Obr. 8.13: Vloženie sandbox definície



Obr. 8.14: Alokovanie sandbox definície

Úspešnú alokáciu poolu môžeme vidieť na obrázku číslo 8.15.



Obr. 8.15: Alokovanie poolu pre sandbox deiníciu

Po týchto krokoch už môžme do tréningového prostredia pristupovať cez vzdialený prístup protokolom SSH a pracovať s ním podľa našich potrieb.

Pristupovať k jednotlivým prvkom z našej topológie budeme cez SSH kľúč, ktorý máme uložený v priečinku *.ssh* na serveri VUT, z ktorého všetko vykonávame. To aby sme mohli použiť SSH kľúč pre prístup do virtuálnych strojov nám zabezpečuje KYPO platforma. Na tejto platforme máme možnosť si po alokovaní sandboxu a vytvorení poolu stiahnuť súbor s týmito vygenerovanými kľúčmi, **ssh-access.zip**.

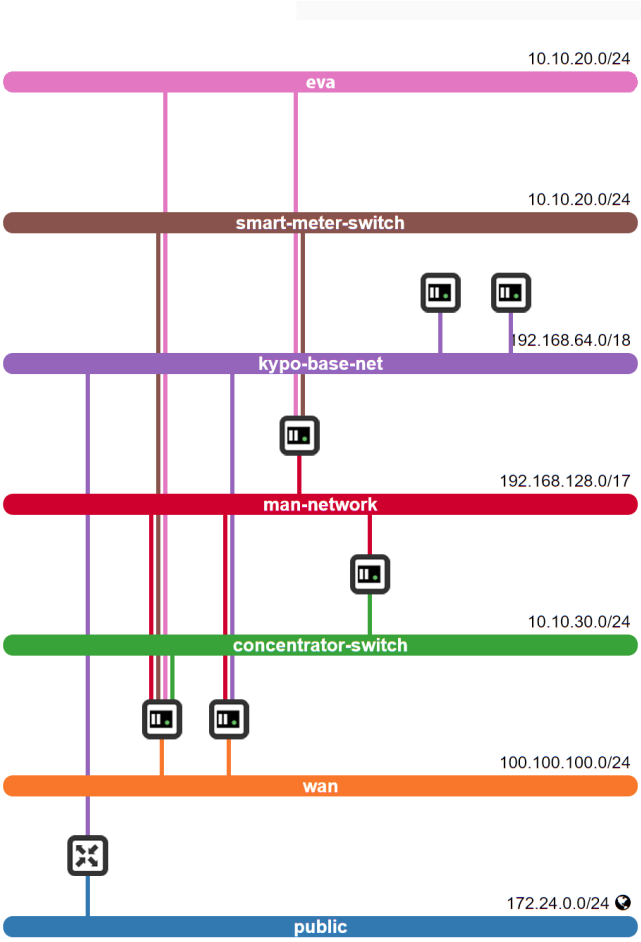
Po stiahnutí súboru je nutné si tento súbor nahrať na server VUT. K tomu vedie mnoho prístupov a my sme si zvolili na prevod použitie programu WinSCP. WinSCP je open source klient na prenos súborov pre Microsoft Windows. Jeho hlavnou funkciou je bezpečný prenos súborov medzi miestnymi a vzdialenými počítačmi. Potom si už tento súbor stačí len rozbaliť do priečinku *.ssh* a vzdialená správa je možná.

Našou úlohou diplomovej práce bola príprava bezpečnostného scenára v laboratórnom prostredí platformy KYPO. To dosiahneme úpravou playbooku tak aby po spustení tréningového prostredia bola stanica (útočník) pripravená vykonávať požadované útoky na zraniteľnosti protokolu. Medzi zraniteľnosti protokolu DLMS/COSEM, ktoré sme sa rozhodli využiť patrí slabá ochrana na DDoS útoky, REPLAY attack a odpočúvanie komunikácie Man-in-the-middle.

Aby sme mohli útoky uskutočniť a demonštrovať pridali sme do topológie útočníka *eva*. Eva je v jednej sieti s chytrým elektromerom a je je prepojená routerom aj na koncentrátor. Celú sieťovú topológiu si môžete pozrieť na obrázku číslo 8.16. Náš útočník v našom scenári disponuje operačným systémom linuxovej distribúcie. Na odpočúvanie komunikácie medzi elektromerom a koncentrátorom sme použili balíček dsniff, ktorý obsahuje **arpspoof**. Nástroj arpspoof je výborný na presmerovanie paketov z cieľového hostiteľa (alebo všetkých hostiteľov) v sieti LAN určených pre iného miestneho hostiteľa falšovaním odpovedí ARP. Takisto ale nesmieme zabudnúť na stroji útočníka zapnúť ipforward aby sme mohli zachytenú komunikáciu preposlať ďalej.

Na využitie zraniteľnosti na REPLAY attack je potreba nainštalovať viacero balíčkov. Patrí medzi ne **tcpdump**, **tcpreplay** a **bittwist**. Pri útoku REPLAY hacker zachytí údaje a opätovne odošle tú istú požiadavku na server, takže to vyzerá, že údaje pochádzajú z legitímneho hosťa. Výsledkom tohoto útoku je že, keď server odošle odpoveď, dostane ju hacker.

Ako posledný útok sme sa rozhodli simulovať DDoS útok, ktorý bude simulovaný len z jednej stanice pomocou nástroja **hping3**. Útok DDOS (Distributed Denial of Service) je podobný útoku DOS, ale je vykonávaný z rôznych uzlov (alebo rôznymi útočníkmi) súčasne. Útoky DDOS bežne vykonávajú botnety. Na prevedenie tohoto



Obr. 8.16: Použitá sieťová topológia pre bezpečnostný scenár

útoku nám v našom scenári slúži nástroj hping3. Nástroj hping3 umožňuje odosielať manipulované pakety. Tento nástroj umožňuje kontrolovať veľkosť, množstvo a fragmentáciu paketov s cieľom preťažiť cieľ a obísť alebo napadnúť firewall. Nástroj hping3 môže byť užitočný na účely testovania bezpečnosti alebo schopností, pomocou neho je možné otestovať to či klient zvládne veľké množstvo paketov. Všetky zraniteľnosti a ich možnosti využitia sú popísané v samostatnej kapitole, kde sú rozpísané väčšie podrobnosti.

Podrobnejší opis prevedenia celého bezpečnostného scenáru je v prílohe, ktorá je súčasťou tejto práce. Tento bezpečnostný scenár je tvorený pre študentov informačnej bezpečnosti pre laboratórne cvičenia. Hlavným zámerom je aby sa študenti zoznámili s možnosťami čo Cyber Range platformy poskytujú a čím môžeme z toho všetci prosperovať. V tomto cvičení je popísaný návod na spracovanie úlohy spolu s teoretickým úvodom a kontrolnými otázkami v závere.

Záver

Diplomová práca obsahuje základnú charakteristiku chytrých sietí a spomína ich historický ale hlavné rýchly vývoj v dnešnej dobe. Obsahuje popis trendov a prvkov, ktoré chytré siete a chytré domácnosti môžu obsahovať. Základný stavebný kameň je v tomto prípade smart meter, ktorého hlavnou úlohou je merať aktuálnu spotrebu daného objektu a komunikovať s dodávateľmi elektrickej energie. To ako komunikácia môže prebiehať a jej jednotlivé prvky sú popísané v ďalšej kapitole. Ako posledná časť prvej kapitoly je zhrnutá problematika kritickej infraštruktúry a jej potreby pre kybernetickú bezpečnosť. Tú sa snaží Európska Únia dosiahnuť komplexným legislatívnym rámcom prostredníctvom smerníc a rôznych nariadení, keďže kybernetické útoky v tomto odvetví sú čoraz početnejšie.

V druhej kapitole je spomenutých 6 protokolov, ktoré sa používajú na komunikáciu v energetických sieťach. Je tu spomenutá krátka charakteristika a základné predstavenie protokolov spolu s priebehom komunikácie. Väčšina týchto protokolov sa používa v spojení s IoT. Najväčší dôraz tejto práce je na protokol DLMS/COSEM a aj preto, je mu venovaná jedna celá kapitola. Tento protokol ja štandardom pre komunikáciu v chytrých energetických sieťach. COSEM alebo Companion Specification for Energy Metering, obsahuje sadu špecifikácií, ktoré definujú dopravné a aplikačné vrstvy protokolu DLMS. Združenie DLMS definuje protokoly do súboru štyroch špecifikačných dokumentov Green Book, Yellow Book, Blue Book a White Book.

V praktickej časti tejto práce práce je popis toho, čo bolo treba spraviť a ako pripraviť prostredie pre vytvorenie kybernetického polygonu (KYPO) a inštalácia so spustením platformy Openstack. Je tu spomenutá celá schéma prevedenia a neskôr aj jej jednotlivé časti a programové vybavenie, ktoré sme použili na uvedenie do prevádzky. Zároveň pri postupe riešenia sú zahrnuté aj možné komplikácie pri vypracovávaní zadanie diplomovej práce. Ako hlavné je treba spomenúť kompatibilitu verzií programov a funkcií, ktorá nás sprevádzala celým procesom pretože sa platforma a jej časti neustále vyvíjajú.

Na záver tejto časti sa venujeme ukážke a popisu prevedeniu útokov v prostredí tréningovej platformy KYPO. Takisto je tu spomenuté čo všetko predchádza príprave tejto časti a ktoré konfiguračné súbory bolo nutné upraviť na vytvorenie sieťovej infraštruktúry a definíciu použitých vytvorených virtuálnych strojov spolu s definíciou prvkov používaných v energetike. Ako poslednú časť tejto kapitoly popisujeme aj súhrn využitých zraniteľností a to ako sme tieto zraniteľnosti zneužili a aké nástroje boli pri tom používané. To nadväzuje na pripravené laboratórne cvičenie, ktoré je súčasťou príloh tejto práce.

Literatúra

[1] *Chytrá elektřina: co jsou to inteligentní sítě a k čemu slouží.* DENKOVÁ, Adéla. Euractiv [online]. 2017 [cit. 2021-10-11]. Dostupné z: <https://euractiv.cz/section/all/linksdossier/chytra-elektrina-co-jsou-to-inteligentni-site-a-k-cemu-slouzi/>

[2] *European commission: Smart grids and meters* European commission [online]. 2014 [cit. 2021-10-12]. Dostupné z: <https://ec.europa.eu/energy/topics/markets-and-consumers/smart-grids-and-meters_en?redir=1/>

[3] *European commission: Critical infrastructure and cybersecurity* European commission [online]. 2019 [cit. 2021-10-15]. Dostupné z: <https://ec.europa.eu/energy/topics/energy-security/critical-infrastructure-and-cybersecurity_en?redir=1>

[4] *JOINT COMMUNICATION TO THE EUROPEAN PARLIAMENT, THE COUNCIL, THE EUROPEAN ECONOMIC AND SOCIAL COMMITTEE AND THE COMMITTEE OF THE REGIONS.* EUR-Lex [online]. 2013 [cit. 2021-10-26]. Dostupné z: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A52013JC0001>

[5] *SMERNICA EURÓPSKEHO PARLAMENTU A RADY (EÚ) 2016/1148.* EUR-Lex [online]. 2016 [cit. 2021-10-26]. Dostupné z: <https://eur-lex.europa.eu/legal-content/SK/TXT/?uri=CELEX%3A32016L1148>

[6] *JOINT COMMUNICATION TO THE EUROPEAN PARLIAMENT AND THE COUNCIL.* EUR-Lex [online]. Brusel, 2017 [cit. 2021-10-26]. Dostupné z: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=JOIN:2017:0450:FIN>

[7] *MATOUŠEK, Petr. Analysis of DLMS Protocol: Technical Report, version 1.0* [online]. Brno, 2017 [cit. 2021-10-25]. Dostupné z: <https://www.fit.vut.cz/research/publication-file/11616/TR-DLMS.pdf.> Technická zpráva. Vysoké učení technické v Brně, Fakulta informačních technologií.

[8] *IEC 61850: soubor norem pro komunikaci v energetice s velkým potenciálem výhod.* Automa [online]. 2010, (03), 10-12 [cit. 2021-10-25]. Dostupné z: <https://automa.cz/cz/casopis-clanky/iec-61850-soubor-norem-pro-komunikaci-v-energetice-s-velkym\-potencialem-vyhod-2010_03_40771_5154/>

[9] *Description and analysis of IEC 104 Protocol* [online]. Brno, 2017 [cit. 2021-10-25]. Dostupné z: <https://www.fit.vut.cz/research/publication-file/11570/TR-IEC104.pdf>. Technická zpráva. Vysoké učení technické v Brně, Fakulta informačných technologií.

[10] *IEC 60870-5-104.* IPCOMM [online]. Nemecko [cit. 2021-10-25]. Dostupné z: <https://www.ipcomm.de/protocol/IEC104/en/sheet.html>

[11] *Z-Wave.* Alza [online]. [cit. 2021-10-26]. Dostupné z: <https://www.alza.cz/slovnik/z-wave-art17515.htm>

[12] *SPORRE, Kyle. Understanding the Zigbee 3.0 Protocol.* Digi [online]. USA, Minnesota, 2018 [cit. 2021-11-06]. Dostupné z: <https://www.digi.com/blog/post/understanding-the-zigbee-3-0-protocol>

[13] *6LoWPAN.* Radiocrafts: Embedded wireless solutions [online]. [cit. 2021-11-07]. Dostupné z: <https://radiocrafts.com/technologies/6lowpan/>

[14] *IEC 62056.* Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2019 [cit. 2021-10-18]. Dostupné z: <https://en.wikipedia.org/wiki/IEC_62056>

[15] *Overview DLMS UA.* Dlms [online]. Švajčiarsko [cit. 2021-10-18]. Dostupné z: <https://www.dlms.com/dlms-cosem/overview>

[16] *Green Book: DLMS/COSEM Architecture and protocols* [online]. 9th. 2019 [cit. 2021-10-19]. Dostupné z: <https://www.dlms.com/files/Green_Book_Edition_9-Excerpt.pdf>

[17] *PECHÁČEK, Jakub. Komunikační protokoly pro chytré sítě* [online]. Liberec, 2019 [cit. 2021-10-20]. Dostupné z: <https://dspace.tul.cz/bitstream/handle/15240/153265/BPJAKUBPECHACEK.pdf?sequence=1&isAllowed=y> Bakalářská práce. Technická univerzita v Liberci.

[18] *DLMS/COSEM: Architecture and Protocols* [online]. 8.1 edition. 2015 [cit. 2021-11-13].

[19] *MEDEIROS, Ibéria, Henrique MENDES a Nuno Ferreira NEVES. Validating and Securing DLMS/COSEM Implementations with the ValiDLMS Framework* [online]. In: . 2018, s. 1-7 [cit. 2021-12-09]. Dostupné z: doi:10.1109/DSN-W.2018.00060

[20] *LURING, Norman, Daniel SZAMEITAT, Stefan HOFFMANN a Gerd BUMILLER.* Analysis of security features in DLMS/COSEM: Vulnerabilities

and countermeasures [online]. In: . Washington, DC, USA: IEEE, 2018, 09.10 2018, s. 1-5 [cit. 2022-01-21]. ISSN 2472-8152. Dostupné z: <doi:10.1109/ISGT.2018.8403340>

[21] *KOHOUT, David. ZÁTĚŽOVÝ GENERÁTOR ZPRÁV DLMS/COSEM* [online]. Brno, 2019 [cit. 2021-12-08]. Dostupné z: <https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=193533>. Bakalárska práca. VUT v Brne. Vedoucí práce Ing. Tomáš Lieskovan.

[22] *N. Luring, D. Szameitat, S. Hoffmann and G. Bumiller, Analysis of security features in DLMS/COSEM: Vulnerabilities and countermeasures,* [online]. 2018 IEEE Power and Energy Society Innovative Smart Grid Technologies Conference (ISGT), 2018, pp. 1-5 [cit. 2021-12-09]. Dostupné z: <https://ieeexplore.ieee.org/document/8403340> , doi: 10.1109/ISGT.2018.8403340.

[23] *The Cyber Range: A Guide: Guidance Document for the Use Cases, Features, and Types of Cyber Ranges in Cybersecurity Education, Certification and Training* [online]. In: , National Initiative for Cybersecurity Education (NICE) Cyber Range Project Team. s. 1-17 [cit. 2021-12-06]. Dostupné z: <https://www.nist.gov/system/files/documents/2020/06/25/The%20Cyber%20Range%20-%20A%20Guide%20%28NIST-NICE%29%20%28Draft%29%20-%20062420_1315.pdf>

[24] *VYKOPAL, Jan, Radek OŠLEJŠEK, Pavel CELEDA, Martin VIZVÁRY a Daniel TOVARNÁK. KYPO Cyber Range: Design and Use Cases* Madrid, Spain: SciTePress, [online]. 2017. p. 310-321. [cit. 2021-12-07]. ISBN 978-989-758-262-2. Dostupné z: doi:<http://dx.doi.org/10.5220/0006428203100321>

[25] *OpenStack.* Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2021-12-08]. Dostupné z: <https://en.wikipedia.org/wiki/OpenStack>

[26] *HANÁK, Jiří. Jak funguje OpenStack a šest důvodů, proč v něm mít cloud.* MasterDC [online]. 2015 [cit. 2021-12-08]. Dostupné z: <https://www.master.cz/blog/co-je-openstack-jak-funguje-vyhody-openstacku/>

[27] *Co je cloud computing?: Průvodce pro začátečníky.* Azure Microsof [online]. [cit. 2021-12-08]. Dostupné z: <https://azure.microsoft.com/cs-cz/overview/what-is-cloud-computing/>

[28] *Openstack* [online]. 2010 [cit. 2021-12-08]. Dostupné z: <https://www.openstack.org/>

[29] *VYKOPAL, Jan, Pavel CELEDA, Pavel SEDA, Valdemar SVABENSKY a Daniel TOVARNAK. Scalable Learning Environments for Teaching Cybersecurity Hands-on. 2021 IEEE Frontiers in Education Conference (FIE)* [online]. IEEE, 2021, 2021-10-13, 1-9 [cit. 2022-03-27]. ISBN 978-1-6654-3851-3. Dostupné z: doi:<http://dx.doi.org/10.1109/FIE49875.2021.9637180>

Zoznam symbolov a skratiek

6LoWPAN IPv6 over Low-Power Wireless Personal Area Networks

AARQ A-Associate Request

ACK Acknowledgement

ACON Again COnnected Networks

AE Aplikačná entita

AES Advanced Encryption Standard

AP Aplikačné procesy

API Aplication interface

APDU Apllication Protocol Data Units

COSEM Companion Specification for Energy Metering

CSC Cyber Sandbox Creator

DDoS Distributed Denial of Service

DH Diffie Hellman

DLMS Device Language Message Specification

DLMS UA Device Language Message Specification User Association

DoS Denial of Service

DH Diffie Hellman

ECDSA Elliptic Curve Digital Signature Algorithm

ECHD Elliptic-curve Diffie–Hellman

EMS Energy Management System

EÚ Európska únia

GB Gigabyte

GCM Galois/Counter Mode

GHz Gigahertz

HDLC High-Level Data Link Control

HLS High Level Security

IaaS Infrastructure as a Service

IANA Internet Assigned Numbers Authority

ID Identifikačný kód

IEEE Electrical and Electronics Engineers

IETF Internet Engineering Task Force

IoT Internet of Things

IT Information Technology

L1-7 Layer 1-7, vrsty v komunikačnom modeli

LLS Low Level Security

LMS Learning Management System

LN Local Networks

MAC Message Authentication Code

MHz Megahertz

NN Neighbour Network

OBIS Object Identification System

OS Operačný systém

OSI Open Systems Interconnection

PC Personal Computer

PCI Projects of Common Interest

RAM Random access memory

RF Radiofrequency

RFC Request for comments

RLMS Range Learning Management System

RO	Read only
RSA	Rivest, Shamir, Adleman
RTU	Remote Terminal Unit
RW	Read and Write
SCADA	Supervisory Control And Data Acquisition
SHA	Secure Hash Algorithm
SSH	Secure shell
SYN	Synchronous
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
VUT	Vysoké učení technické
WAN	Wide Area Network
Wi-Fi	Wireless Fidelity
WO	Write only
YAML	Yet Another Markup Language

A Návod pre využitie zraniteľnosti protokolu DLMS/COSEM v tréningovom prostredí Cyber Range platformy KYPO

A.1 Teoretický úvod

V tomto cvičení sa budeme venovať zneužitiu zraniteľností protokolu DLMS/COSEM v tréningovom prostredí Cyber Range platformy KYPO, ktorá je vyvíjaná na Masarykovej Univerzite v Brne. Na uskutočnenie laboratórnej úlohy budeme potrebovať funkčnú platformu **Cyber Range KYPO**, ktorá bude spustená na vzdialenom serveri pretože vyžaduje viac výpočtového výkonu ako dokáže bežný užívateľ disponovať a k tomu **osobný počítač** s pripojením na internet aby sme k tréningovej platforme mohli pristúpiť.

A.1.1 Cyber Range platformy

Cyber range platformy sú interaktívne, simulované platformy a reprezentácie sietí, systémov, nástrojov a aplikácií. Tieto platformy môžu poskytovať:

- školenie a hodnotenie založené na výkone,
- simulované prostredie, kde môžu tímy spolupracovať na zlepšení tímovej práce a tímových schopností,
- spätnú väzbu v reálnom čase,
- skúsenosti na pracovisku,
- prostredie, v ktorom možno testovať nové nápady a kde tímy môžu pracovať na riešení zložitých kybernetických problémov.

Koncept simulácie spočíva v tom, že sa vždy vytvorí nové sieťové prostredie založené na správaní skutočných sieťových komponentov. Simulácie bežia vo virtuálnych inštanciách a nevyžadujú žiadne fyzické sieťové vybavenie. Tieto šablóny virtuálnych strojov sú štandardizované, a preto sú do istej miery obmedzené v tom, ako presne simulujú skutočnú IT infraštruktúru. Veľkou výhodou simulačného prostredia je rýchlosť konfigurácie a schopnosť používať sieťové a úložné prostriedky.

A.1.2 KYPO

KYPO je navrhnutý ako modulárny distribuovaný systém. Platforma KYPO využíva cloudové prostredie na dosiahnutie vysokej flexibility a škálovateľnosti. Virtualizácia zase umožňuje opakovane vytvárať plne funkčné virtualizované siete

s plnohodnotnými operačnými systémami a sieťovými prvkami, ktoré sú takmer identické so systémami z reálneho sveta. Vďaka svojej modulárnej architektúre je KYPO schopný bežať na rôznych platformách cloud computingu, napríklad ako v našom prípade na OpenStack.

A.1.3 DLMS/COSEM

DLMS/COSEM je celosvetový štandard, ktorý slúži ako komunikačný protokol pre smart meters, ktoré slúžia na meranie elektriny, plynu, vody, ... Definuje objektovo orientovaný dátový model, aplikačný protokol a komunikačné profily špecifické pre používané médiá. DLMS/COSEM zahŕňa tri kľúčové komponenty: DLMS (Device Language Message Specification), COSEM (Companion Specification for Energy Metering), OBIS (Object Identification System).

DLMS je protokol aplikačnej vrstvy, ktorý mení informácie uchovávané v objektoch na správy. Táto vrstva reguluje diaľkové odčítanie nameraných hodnôt z meracích zariadení a ich vzdialené ovládanie a tiež ďalšie služby pre meranie akéhokoľvek typu energie.

COSEM je objektový model rozhrania komunikačného zariadenia pre merania akéhokoľvek typu energie. Je to špecifikácia, ktorá poskytuje reprezentáciu funkčnosti meracích zariadení. Model rozhrania používa objektovo orientovaný prístup.

OBIS predstavuje systém, ktorý definuje pomenovanie objektov. OBIS definuje identifikačné kódy (ID), čím poskytuje jedinečný identifikátor pre všetky dáta v meranom systéme. Tieto identifikačné kódy sa používajú pre bežné dátové položky v zariadeniach na meranie energií.

V komunikácii DLMS/COSEM má každá strana, ktorá komunikuje priradenú svoju vlastnú adresu. Adresa klienta ja podľa definície protokolu bajtová hodnota. Hodnota adresy klienta určuje aj skutočnú povahu klienta. Môžeme napríklad uviesť prípad kde norma uvádza, že klient s hodnotou adresy 16 je verejný klient. Môže ale existovať aj iný druh klientov: systém zberu údajov, výrobca, spotrebiteľ, ... Adresa sa skladá z adresy fyzického zariadenia a adresy logického zariadenia.

A.2 Úloha laboratórneho cvičenia

Úlohou tohoto laboratórneho cvičenia je postupne využiť tri vybrané zraniteľnosti protokolu DLMS/COSEM. Na realizáciu týchto úloh dostane študent pripravený scenár s dopredu definovanou sieťovou topológiou, ktorá obsahuje chytrý **elektromer**, **koncentrátor** a útočníka **eva**. Takisto tieto virtuálne stroje obdržia

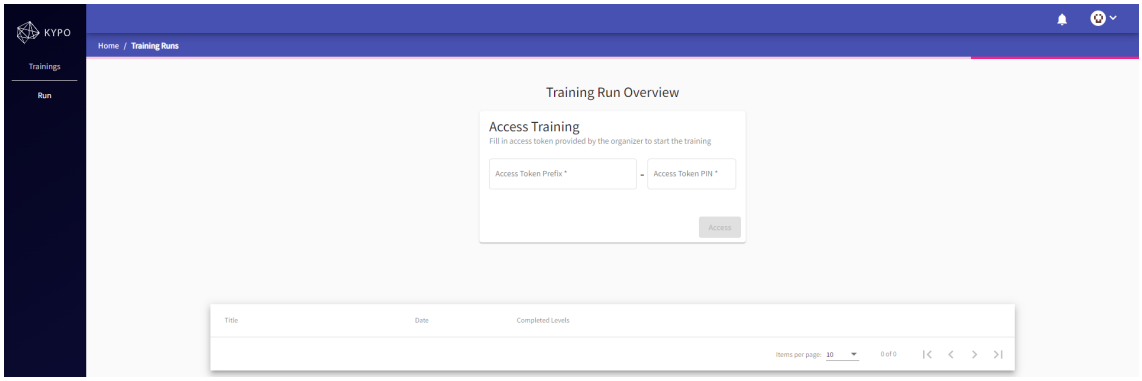
už pripravené a nakonfigurované tak aby sa študent len pripojil prostredníctvom konzole na počítač útočníka a mohol vykonávať útoky. Útočníkov virtuálny stroj a ostatné prvky v sieťovej konfigurácii boli pripravené v súbore **playbook.yml** a **topology.yml**, potrebné nástroje sú už na strane útočníka nainštalované.

Študent teda postupne bude vykonávať tri úkony:

1. Odpočúvanie komunikácia prostredníctvom balíčka *dsniff*.
2. Útok na odoprenie služby DDoS s nástrojom *hping3*.
3. REPLAY attack s využitím balíčkov *tcpdump, tcpreplay, bittwist*.

A.3 Realizácia tréningového scenáru

Študent obdrží prihlasovacie údaje a IP adresu webového serveru, ktorom bude Cyber Range platforma spustená. Študent zadá svoje prihlasovacie údaje do príslušných polí a prihlási sa. Po prihlásení sa do platformy by ste mali vidieť úvodnú obrazovku s možnosťou zadať ID tréningového scenáru aby ste sa mohli laboratórnej úlohy zúčastniť. Úvodná obrazovka by mala vyzerať ako na obrázku číslo A.1



Obr. A.1: Úvodná obrazovka KYPO

A.3.1 Odpočúvanie komunikácia

Na odpočúvanie komunikácie medzi elektromerom a koncentrátorom sme použili balíček dsniff, ktorý obsahuje nástroj **arpspoof**. Nástroj arpspoof je výborný na presmerovanie paketov z cieľového hostiteľa (alebo všetkých hostiteľov) v sieti LAN určených pre iného miestneho hostiteľa falšovaním odpovedí ARP. Takisto ale nesmieme zabudnúť na stroji útočníka zapnúť ipforward aby sme mohli zachytenú komunikáciu preposlať ďalej a následne na ukončenie tohoto útoku nastaviť hodnotu ipforward na hodnotu 0. Štruktúra útoku bude mať následovné parametre.

```
# sysctl -w net.ipv4.ip_forward=1
# arpspoof -i [Interface Name] -t [Victim IP] [Router IP]
# arpspoof -i [Interface Name] -t [Router IP] [Victim IP]
```

Útoky Man in the Middle patria medzi najčastejšie pokusy o útoky na sieti. Používajú sa väčšinou na získanie prihlasovacích údajov alebo osobných informácií, špehovanie obete, sabotovanie komunikácie alebo poškodenie údajov. Aby sme mohli tento útok uskutočniť musíme pracovať s viac ako jedným terminálom. Na to aby sme mohli pracovať v jednom okne pracovať s viacerými terminálmi použijeme nástroj **screen**. Pokiaľ nie je ešte nainštalovaný tak ho na virtuálny stroj nainštalujeme.

```
# sudo apt install screen
# screen
```

Tento nástroj nám umožňuje v jednom terminálovom okne si vytvárať nové terminály z toho istého virtuálneho stroja a vyberať si medzi nimi. Nižšie je uvedených niekoľko najbežnejších príkazov na správu terminálov a prácu medzi nimi:

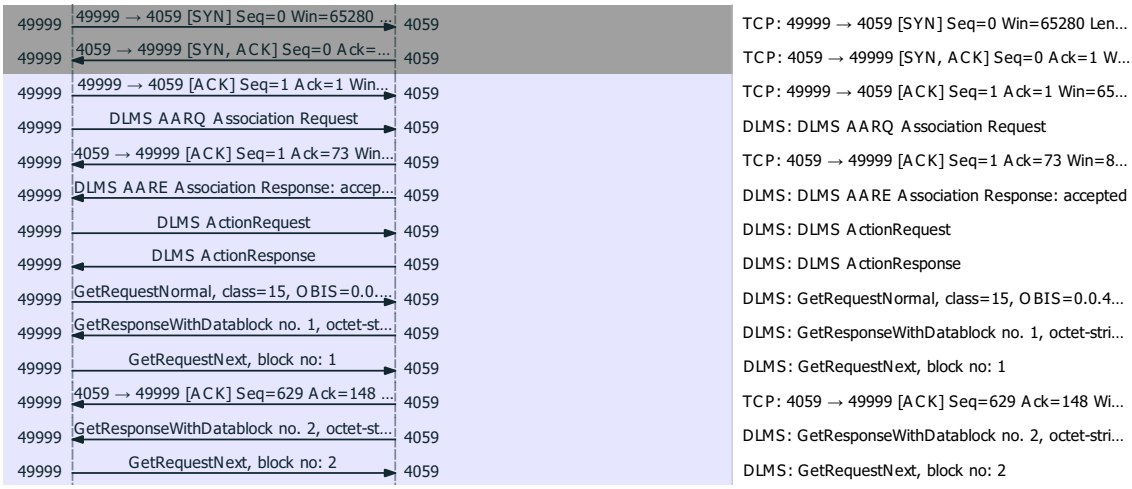
- *Ctrl+a* Vytvoriť nové okno.
- *Ctrl+a 0* Prepnutie na okno 0 (podľa čísla).
- *Ctrl+a S* Rozdeliť aktuálnu oblasť horizontálne na dve oblasti.
- *Ctrl+a Ctrl+a* Prepínanie medzi aktuálnym a predchádzajúcim oknom.
- *Ctrl+a X* Zatvorenie aktuálnej oblasti.

V prvom okne terminálu si spustíme arpspoof kde prvý parameter bude sieťové rozhranie útočníka a ďalší parameter bude IP adresa obete a teda chytrého elektromeru (v našom prípade 10.10.20.5) a ako druhý parameter bude IP adresa routeru v sieti (v našom prípade 10.10.20.1). Pridaním druhého okna terminálu klávesovou skratkou *Ctrl+a* si môžeme spustiť zase druhý príkaz aprspoof s tými istými parametrami ale obrátenými. To znamená, že najskôr použijeme IP adresu routeru a potom IP adresu elektormeru, sieťové rozhranie útočníka zostáva rovnaké.

```
# arpspoof -i eth0 -t 10.10.20.5 10.10.20.1
# arpspoof -i eth0 -t 10.10.20.1 10.10.20.5
```

Priebeh komunikácia môžete zachytiť otvorením okna tretieho terminálu kde zadáte príkaz *tcpdump* s parametrom na ktorom sieťovom rozhraní chcete komunikáciu zachytávať a do akého súboru si chcete záchyt uložiť. Výsledný priebeh zachytenej komunikácie si môžete následne stiahnuť a prezrieť v programe na analýzu sieťového toku napríklad v programe Wireshark. Príklad takej komunikácie je na obrázku číslo A.2.

```
1 # tcpdump -i eth0 -w mitm.pcap
2 # tcpdump -r mitm.pcap
```



Obr. A.2: Príklad komunikácie protokolu DLMS/COSEM

A.3.2 Útok na odoprenie služby DDoS

Útoky DoS sú jedným z najčastejších útokov, ak nie najčastejším. Podstata útoku je taká, že sa vyčerpajú všetky zdroje cieľa, aby ich nemohli používať iní a nebol schopný odpovedať na legitímne požiadavky.

Nástroj s názvom **hping3** útočníkovi umožňuje vytvárať a odosielať vlastné pakety. Vďaka tomu s ním môžeme robiť veľa vecí vrátane prieskumu, prípadne základného zneužitia, v našom prípade ho použijeme na útok DDoS. Existuje viacero druhov útokov DDoS, ale tu sa zameriame na SYN flood. Ten posiela požiadavky na server tak rýchlo, ako len môže. Keď sa tieto požiadavky spracujú, zaberú zdroje servera a znemožnia mu odpovedať na všetky skutočné požiadavky používateľov, ktorí sa ho snažia použiť. Ako prvý krok si musíme zistiť aké porty má má smart-meter otvorené a to spravíme nástrojom **nmap** alebo druhou variantou nástrojom hping3.

```
1  # nmap 10.10.20.5
2  # hping3  --scan 1-65535 10.10.20.5 -S --rand-source
```

Keď zistíme aké porty sú otvorené a vieme IP adresu nášho cieľa a aj našu IP adresu tak môžeme použiť nástroj hping3 a spustiť DDoS záplavový útok

s príznakom SYN paketu. Port 4060 je port, na ktorom bežia služby DLMS a preto útočíme naň.

```
1  # hping3 -S 10.10.20.6 -a 10.10.20.5 -p 4061 --flood
2  # hping3 -S -d 10000 --flood 10.10.20.5 //príznak SYN
3  # hping3 -F -d 10000 --flood 10.10.20.5 //príznak FIN
```

Po spustení jedného z vyššie uvedeného príkazov si môžeme dať požiadavku z koncentrátora aby skúsil znovu vyčítať dáta z chytrého elektromeru, ktorý mu už ale nebude odpovedať pretože je nedostupný a zahltený inými požiadavkami. Príkaz na vyčítanie dát z elektromeru môžeme vidieť nižšie.

```
1  # java -jar target/gurux.dlms.client.example.java-0.0.1
2  -SNAPSHOT.jar -h 10.10.20.5 -p 4061
```

A.3.3 REPLAY útok

REPLAY útok je špecifickejším typom útoku typu man-in-the-middle, takže majú niektoré spoločné črty a preto informácie, ktoré už vieme vďaka MitM útoku tak ich tu použijeme. Pri REPLAY útoku hacker zachytí vaše údaje a opätovne odošle tú istú požiadavku na server, takže to vyzerá, že údaje pochádzajú od vás. Na toto budeme používať balíčky **tcpdump**, **tcpreplay**, a tí ktorí chcú posúvať svoje vedomosti tak si môžu vyhľadať syntax balíčka **bittwist** a použiť namiesto tcpreplay balíček bittwist, ktorý je tiež na virtuálnom stroji už pripravený.

V našom prípade sme zachytili 5 paketov nástrojom tcpdump a následne týchto 5 paketov pošleme na chytrý elektromer z útočníka eva aby si elektromer myslel, že sa jedná o legitímne požiadavky z koncentrátoru.

```
1  # tcpdump -i eth0 -c 5 -w dlms.pcap  //zachytenie paketov
2  # tcpdump -r dlms.pcap  //výpis súboru
3  # tcpreplay -i eth0 -t dlms.pcap  //zopakovanie komunikácie
```

A.3.4 Kontrolné otázky

1. Na čo slúži Cyber Range platforma KYPO?
2. Kde sa najviac používa protokol DLMS/COSEM?
3. Na aké typy útokov je protokol DLMS/COSEM zraniteľný?
4. Aká nástroje boli použité na využitie zraniteľností?
5. Čo musíme pred prevedením útoku zistiť o našom cieli?
6. Ako zistíme či bol útok úspešný?