VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

IDENTIFIKACE ÚTOKŮ U DOMÁCÍCH IOT SÍTÍ

IDENTIFYING ATTACKS IN HOME IOT NETWORKS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **TOMÁŠ MATUŠ**
AUTHOR

VEDOUCÍ PRÁCE **doc. Ing. ONDŘEJ RYŠAVÝ, Ph.D.**
SUPERVISOR

BRNO 2023

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Zadání bakalářské práce

146950

Ústav:	Ústav informačních systémů (UIFS)
Student:	**Matuš Tomáš**
Program:	Informační technologie
Specializace:	Informační technologie
Název:	**Identifikace útoků u domácích IoT sítí**
Kategorie:	Počítačové sítě
Akademický rok:	2022/23

Zadání:

1. Navrhněte a realizujte experimentální prostředí pro domácí síť IoT založené na použití nástroje Home Assistant.
2. Navrhněte způsob monitorování tohoto prostředí pomocí síťových toků a logů z nástroje HA.
3. Vyzkoušejte s použitím nástrojů penetračního testování různé vybrané útoky na tento systém.
4. Pro provedené útoky analyzujte nasbíraná monitorovací data.
5. Navrhněte detekční metodu pro vybrané útoky.
6. Vhodným způsobem zhodnoťte navrženou metodu detekce útoků.

Literatura:

- V. Sivaraman, H. H. Gharakheili, C. Fernandes, N. Clark and T. Karliychuk, "Smart IoT Devices in the Home: Security and Privacy Implications," in *IEEE Technology and Society Magazine*, vol. 37, no. 2, pp. 71-79, June 2018, doi: 10.1109/MTS.2018.2826079.
- F. Corno and L. Mannella, "A Threat Model for Extensible Smart Home Gateways," *2022 7th International Conference on Smart and Sustainable Technologies (SpliTech)*, 2022, pp. 1-6, doi: 10.23919/SpliTech55088.2022.9854235.
- Martin Lundgren; Ali Padyab, "Security and Privacy of Smart Homes: Issues and Solutions," in *Security and Privacy in the Internet of Things: Architectures, Techniques, and Applications* , IEEE, 2022, pp.235-260, doi: 10.1002/9781119607755.ch9.

Při obhajobě semestrální části projektu je požadováno:
První tři body zadání.

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

Vedoucí práce:	**Ryšavý Ondřej, doc. Ing., Ph.D.**
Vedoucí ústavu:	Kolář Dušan, doc. Dr. Ing.
Datum zadání:	1.11.2022
Termín pro odevzdání:	10.5.2023
Datum schválení:	28.10.2022

Abstrakt

Cílem této bakalářské práce je návrh a realizace experimentálního prostředí pro domácí síť IoT založené na použití nástroje Home Assistant, testování různých vybraných útoků na tento systém pomocí penetračních nástrojů, návrh způsobu monitorování tohoto prostředí pomocí logů z nástroje Home Assistant, analýza nasbíraných monitorovaných dat, návrh detekční metody pro vybrané útoky a zhodnocení navržené metody detekce útoků.

Abstract

The aim of this bachelor thesis is to design and implement an experimental environment for a home IoT network based on the use of Home Assistant, testing various selected attacks on this system using penetration testing tools, designing a way to monitor this environment using logs from Home Assistant, analysis of collected monitoring data, design of detection method for the selected attacks and evaluation of the proposed attack detection method.

Klíčová slova

Internet věcí, Home Assistant, Zigbee, monitorování, detekce anomálií

Keywords

Internet of Things, Home Assistant, Zigbee, monitoring, anomaly detection

Citace

MATUŠ, Tomáš. *Identifikace útoků u domácích IoT sítí.* Brno, 2023. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce doc. Ing. Ondřej Ryšavý, Ph.D.

Identifikace útoků u domácích IoT sítí

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana docenta Ondřeje Ryšavého. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Tomáš Matuš
16. května 2023

Poděkování

Rád bych touto cestou vyjádřil poděkování panu doc. Ing. Ondřeji Ryšavému, Ph.D. za jeho cenné rady, doporučení a trpělivost při vedení mé bakalářské práce.

Obsah

1 Úvod **3**

2 Internet věcí a standardizace zabezpečení **4**
2.1 Kybernetická bezpečnost v oblasti IoT 5
2.2 Doporučená základní kritéria IoT zařízení 6

3 Testovací prostředí a používané technologie **8**
3.1 Laboratoř 8
3.2 Aplikace pro správu chytré sítě 9
3.3 Komunikační protokol Zigbee 11

4 Testování vytvořeného prostředí **15**
4.1 Využité nástroje 15
4.2 Testování z vnější sítě 16
4.3 Testování sítě zevnitř 17

5 Metody pro analýzu aktivity zařízení **21**
5.1 Připravené datasety 21
5.2 Metoda OneClassSVM 22
5.3 Metoda K-Means 23
5.4 Metoda IsolationForest 25
5.5 Metoda LocalOutlierFactor 26
5.6 Porovnání výsledků jednotlivých metod 28
5.7 Detekce útoku na Zigbee můstek 28

6 Nástroj pro monitorování aktivit **30**
6.1 Použité technologie 30
6.2 Prototyp 30
6.3 Vyhodnocení úspěšnosti 32

7 Závěr **33**

Literatura **35**

A Obsah SD karty **37**

Seznam obrázků

3.1 Schéma laboratorní sítě 8
3.2 Zobrazení hvězdicové topologie 13
3.3 Zobrazení stromové topologie 13
3.4 Zobrazení smíšené topologie 14

4.1 Souhrn výsledků spuštěných Nessus testů na webové rozhraní Home Assistant 16
4.2 Zachycené pakety zobrazené ve Wireshark 17
4.3 Souhrn výsledků spuštěných Nessus testů na webové rozhraní Zigbee můstku 18
4.4 Záznam o detekci pohybu 19
4.5 Záznam o nedostupnosti můstku 20

5.1 Detekce pomocí metody OneClassSVM 23
5.2 Detekce pomocí metody K-Means 24
5.3 Detekce nad daty se dvěma vlastnostmi pomocí K-Means 25
5.4 Detekce pomocí metody IsolationForest 26
5.5 Detekce pomocí metody LocalOutlierFactor 27
5.6 Detekce DoS útoku 29

Kapitola 1

Úvod

V posledních několika letech došlo k masivnímu rozšíření Internetu věcí (IoT) a díky nárůstu cenově dostupných chytrých zařízení výrazně vzrostlo jejich nasazení nejen ve výrobních halách nebo kancelářských prostorách, ale téměř v každé oblasti života. Stále častěji se nasazují například ve zdravotnických přístrojích a ulehčují tak práci zdravotníkům, kteří díky těmto technologiím mohou spolehlivěji monitorovat stav pacientů a rychleji reagovat na případné potíže.

Obecně využívání chytrých zařízení dává uživatelům jisté pohodlí. Toto však s sebou přináší i poněkud stinnou stránku rychlého rozšíření technologií a tou jsou kybernetické hrozby. Právě kvůli velmi rychlému rozšíření a v počátku nedostačujícímu zabezpečení nebo chybějící standardizaci již došlo k několika hrozbám. Jednou z nich je například incident, kdy útočník dokázal využitím chytrých žárovek Philips Hue[1] proniknout do vnitřní sítě [6].

Je tedy zřejmé, že zabezpečení zařízení připojených k internetu nelze brát na lehkou váhu a je nutné zajistil jejich odolnost vůči známým, i v budoucnu novým, útokům.

Tato práce se skládá celkem ze 7 kapitol. Úvod sloužící jako nastínění problematiky a důležitosti jejího řešení. V kapitole 2 je obecně popsáno využití chytrých zařízení a důležitost zabezpečení jejich kvalitní funkce. V kapitole 3 je popsáno pracovní prostředí včetně využitých technologií a použitých senzorů. Poté v kapitole 4 jsou popsány možné přístupy k testování tohoto prostředí. Představeny jsou využité nástroje, které toto testování umožňují a jsou popsány výsledky jednotlivých testů. V kapitole 5 je popsána příprava datasetu, výběr metod pro jejich analýzu, porovnání a zhodnocení jejich výsledků. Kapitola 6 pojednává o prototypu aplikace, která slouží k jednoduché analýze připravených dat a detekci atypického chování jednotlivých zařízení. V závěru v kapitole č. 7 jsou celkově zhodnoceny dosažené výsledky v rámci této práce a jsou také zmíněny možnosti pro budoucí vylepšení prototypu.

[1] https://www.philips-hue.com/cs-cz

Kapitola 2

Internet věcí a standardizace zabezpečení

V obecné rovině představuje Internet věcí (IoT) jakoukoliv věc, která může být bezdrátově připojena k internetové síti. Je to systém počítačů a chytrých zařízení či strojů, které jsou schopny vzájemně komunikovat a spolupracovat bez asistence člověka. V podstatě jde o digitalizaci, která z původně hloupých zařízení dělá zařízení chytrá, což umožňuje využití elektroniky v každodenním životě. V současné době představuje propojené věci, které jsou vybaveny senzory, softwarem a dalšími technologiemi, které jim umožňují pracovat s daty. Konektivity může být dosahováno prostřednictvím Wi-Fi nebo 5G sítí, ovšem dnes jsou především využívány speciálně vyvinuté standardy, navržené pro levný provoz a jednoduchý přenos dat po síti.

S rozšířením IoT je úzce spjato několik faktorů. Připojení k internetu a cloudu je nyní dostatečně rychlé, aby bylo možné odesílat a přijímat obrovské objemy dat. V současné době existuje několik možných způsobů jak jednotlivá chytrá zařízení připojit do sítě, například pomocí Wi-Fi nebo 4G a 5G sítí, či použitím standardů Zigbee, Z-wave a Bluetooth LE.

Snížením ceny nových typů senzorů bylo možno nasadit IoT zařízení do běžného života. Nejsou tak součástí pouze velkých průmyslových celků, ale dostávájí se i do běžných domácností. Lidé používají ve svých domovech sítě IoT prostřednictvím chytrých přepínačů, senzorů a bezpečnostních zařízení. Monitorují a řídí věci, jako je osvětlení, klimatizace, bezpečnostní systémy a další spotřebiče. K nasazování dochází také ve městech, kde tato zařízení monitorují dopravu a shromažďují data o dopravních zácpách a snaží se tak proativně řídít chod města.

S rostoucím množství nasazených senzorů vzniká potřeba analyzovat velké množství dat. K usnadnění tohoto úkolu bývá využíváno strojového učení a umělé inteligence, která je schopna tato data efektivně zpracovávat.

Kromě výhod a široké škály využití je ovšem nutné zmínit také rizika, která jsou s masovým rozšířením Internetu věcí spojená. Hlavním rizikem a současně nejslabším bodem celé myšlenky je bezesporu bezpečnost. S rostoucím počtem elektroniky připojené k internetu současně narůstá prostor pro útočníky, kteří se jej mohou snažit napadnout. Zabezpečení Internetu věcí se proto stalo důležitou tématikou, kterou je třeba aktivně řešit. V současnoti se tímto zabývají specializované týmy a instituce.

Jednou z těchto intitucí je Národní institut standardů a technologie (NIST) a jejich program *Cybersecurity for IoT*, který se zaměřuje na posílení kybernetické bezpečnosti v oblasti

IoT. Tento program se zabývá problematikou spojenou s rychlým rozvojem a nasazením zařízení IoT, které představují nové hrozby a zranitelnosti v kybernetickém prostoru.

Cílem programu je vytvořit soubor osvědčených postupů, standardů a doporučení, která pomohou výrobcům chytrých zařízení posílit jejich kybernetickou bezpečnost. Zahrnuty jsou všechny aspekty kyberbezpečnosti IoT, tedy hardware i software.

2.1 Kybernetická bezpečnost v oblasti IoT

Rizika kybernetické bezpečnosti a ochrany soukromí u IoT zařízení lze rozdělit do tří hlavních skupin [12].

1. **Ochrana zabezpečení zařízení** – zabránění tomu, aby zařízení bylo použito k útokům. Mezi možné útoky patří distrtibuované útoky typu DDoS mířené na jiné organizace, odposlouchávání síťového provozu a nebo napadení jiných zařízení ve stejné síti. Tento cíl se vztahuje na všechna IoT zařízení.

2. **Ochrana a zabezpečení dat** – nutnost chránit důvěryhodnost, integritu a dostupnost dat včetně osobních údajů, která jsou shromažďována, ukládána, zpracovávána nebo přenášena při komunikaci. Tento cíl se vztahuje na každé zařízení internetu věcí s výjimkou těch, která nevyžadují ochranu žádných dat.

3. **Ochrana soukromí osob** – chránit soukromí jednotlivců ovlivněné zpracováním osobních údajů nad rámec rizik řízených prostřednictvím ochrany zařízení a zabezpečení dat. Tento cíl se vztahuje na všechna zařízení internetu věcí, která zpracovávají osobní údaje nebo která přímo či nepřímo ovlivňují uživatele.

Každý z cílů navazuje na ty předchozí a nijak je nenahrazuje a nepopírá jejich potřebu. Pro splnění každého z cílů je potřeba zahrnout řešení pro jednotlivé oblasti zmírnění rizik. Tyto oblasti jsou popsány níže a každá z nich definuje aspekt v kybernetické bezpečnosti nebo ochrany soukromí, které jsou z hlediska rizikových faktorů považovány za nejvýznamnější.

Oblasti rizik pro cíl 1, Ochrana zabezpečení zařízení

- **Správa zařízení** – uchovávat aktuální a přesný soupis všech používaných IoT zařízení a jejich relevantních vlastností v průběhu jejich životnosti.

- **Správa zranitelností** – identifikace a eliminace známých zranitelností v softwaru a firmwaru IoT zařízení s cílem snížit pravděpodobnost a snadnost jeho zneužití.

- **Řízení přístupu** – zabránění lidem nebo jiným počítačovým zařízení k neoprávněnému a nesprávnému fyzickému nebo logickému přístupu k používání a správě IoT zařízení.

- **Detekce bezpečnostních incidentů zařízení** – monitorování a analýza aktivity IoT zařízení a vyhledávání incidentů, které se týkají jeho zabezpečení.

Oblasti rizik pro cíl 2, Ochrana zabezpečení dat

- **Ochrana dat** – zabránění přístupu k datům při přenosu a manipulaci s nimi, která by mohla odhalit citlivé informace nebo umožnit manipulaci či narušení provozu zařízení.

- **Detekce incidentů v oblasti zabezpečení dat** – monitorování a analýza aktivity IoT zařízení a vyhledávání incidentů, které se týkají bezpečnosti dat.

Oblasti rizik pro cíl 3, Ochrana soukromí osob

- **Řízení toku informací** – udržování aktuálního mapování životního cyklu informací o osobních údajích. Udržovány jsou všechny relevantní faktory o zpracovávání osobních údajů, jako například strany provádějící zpracování, prvky, které zpracovávají nebo jaké akce provádí.
- **Oprávnění ke zpracování osobních údajů** – zajištění oprávnění pro zpracování osobních údajů, aby se zabránilo neoprávněnému zpracování osobních údajů.
- **Informované rozhodování** – umožnění zodpovědným osobám pochopit účinky zpracování osobních údajů a interakce se zařízením. Ty se pak podílí na rozhodování o zpracování osobních údajů a řeší případné problémy.
- **Detekce narušení soukromí** – monitorování a analýza aktivity IoT zařízení a vyhledávání známek narušení soukromí uživatelů.

2.2 Doporučená základní kritéria IoT zařízení

Následující základní kritéria doporučuje NIST pro definování výsledků kybernetciké bezpečnosti, které se očekávají od vývojářů a od spotřebitelských produktů IoT. Většina těchto kritérií se týká přímo IoT zařízení a očekává se, že budou splněna buď softwarovými nebo hardwarovými prostředky. Některá kritéria se vztahují spíše na vývojáře zařízení. U těchto kritérií se očekává, že budou splněna prostřednictvím činností a podporována spíše tvrzeními a důkazy od vývojáře [13].

- **Konfigurace produktu** – konfigurace IoT zařízení je měnitelná. Uživatel může své zařízení nakonfigurovat tak, aby se vyhnul konkrétním hrozbám, o kterých ví na základě svých znalostí. Zároveň existuje možnost jak jednoduše obnovit zabezpečené výchozí nastavení a veškeré tyto změny mohou provádět pouze autorizované osoby.
- **Ochrana dat** – zařízení a jeho součásti chrání uložená a přenášená data pomocí zabezpečovacích prostředků před neoprávněným přístupem, zveřejenění nebo neoprávněnými úpravami. Umožňuje také vymazat nebo znepřístupnit uložená data, která jsou shromážďována. Zachování důvěrnosti, integrity a dostupnosti dat je základem kybernetické bezpečnosti zařízení internetu věcí.
- **Řízení přístupu k rozhraní** – každá komponenta zařízení omezuje logický přístup k protokolům a službám, které jsou používány na místních nebo síťových rozhraní a umožňuje přístup pouze oprávněným osobám, službám a ostatním komponentám chytrého zařízení. Každá komponenta může mít přístup pouze k rozhraním, která jsou nezbytná pro provoz a jsou zavedena opatření pro kontrolou přístupu. Přístup ke všem ostatním rozhraní je znemožněn. Zároveň zařízení musí udržovat správné řízení přístupu po zapnutí během počátečního připojení a při obnovení připojení po odpojení nebo výpadku.

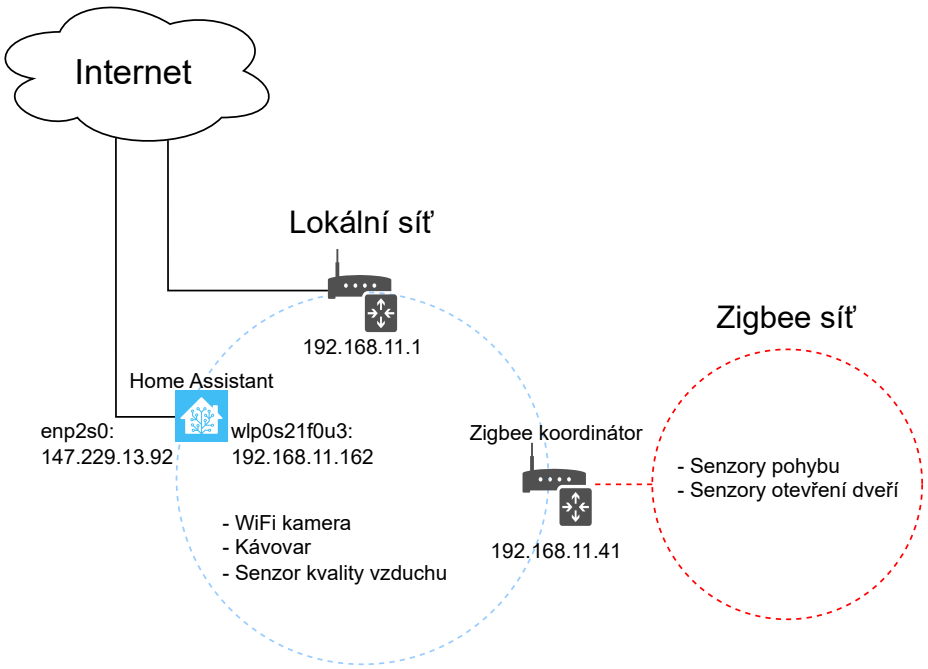
- **Aktualizace software** – v softwaru, který je součástí zařízení mohou být průběžně nalezeny nové zranitelnosti. Oprávněné osoby nebo služby tak musí být schopny pomocí bezpečného mechanizmu provést aktualizaci. Zařízení je schopno příjmout, verifikovat a aplikovat ověřené aktualizace. V případě dostupné aktualizace dojde k upozornění správce a nebo může být automaticky aplikována.
- **Povědomí o kybernetické bezpečnosti** – zařízení podporuje detekci kybernetických bezpečnostních incidentů. Ochranu dat a zajištění správné funkčnosti lze podpořit schopností upozornit správce, když zařízení začne fungovat neočekávaným způsobem, což může znamenat, že došlo k pokusu o neoprávněný přístup, nahrání malwaru, vytvoření botnetu, chybě softwaru zařízení nebo k jiným typům akcí, které nebyly iniciovány uživatelem nebo zamýšleny vývojářem.
- **Dokumentace** – informace týkající se jednotlivých součástí zařízení a jeho bezpečnosti jsou shomažďovány jak během vývoje, tak i v následném životním cyklu. Dokumentace by měla obsahovat očekávaný způsob využívání, zásady pro bezpečný životní cyklus, požadavky zařízení, očekávané vstupy a výstupy, předpokládaná délka života a doba trvání podpory zařízení ze strany výrobce. Součástí by měl být i způsob, jakým je uživatel informován v případě nalezení nových zranitelností.
- **Informovanost o zařízení** – vývojář zařízení vytváří povědomí a nabízí uživatelům informace týkající se kybernetické bezpečnosti produktu. Nabízí manuál jak se zařízením pracovat, jak jej spravovat a udržovat ve správné funkci a jak bezpečně používat zařízení, aby se dosahlo nejlepších výsledků v oblasti kybernetické bezpečnosti.

Kapitola 3

Testovací prostředí a používané technologie

3.1 Laboratoř

Prostředí pro experimentování se zařízeními pro chytrou domácnost se nachází v samostatné síti a to za účelem minimalizovat okolní komunikaci a usnadnit tak cílené odposlouchávání komunikace sledovaných zařízení. Pro ovládání a koordinaci chytrých zařízení je zde využita aplikace Home Assistant. Jelikož spousta chytrých senzorů není přímo připojena do lokální internetové sítě, ale využívají vlastní síť Zigbee 3.3, je zapotřebí připojit bránu, která funguje jako koordinátor a zprostředkovává vytvoření sítě Zigbee. Tato brána je připojena do lokální sítě LAN, a tím vytváří jednotný spoj pro komunikaci chytrých senzorů využívajících Zigbee a ostatními zařízeními připojenými k internetu.



Obrázek 3.1: Schéma laboratorní sítě

Schéma použitého laboratorního prostředí je ukázáno v obrázku 3.1. Vyobrazena je zde především využitá lokální internetová síť a také bezdrátová síť Zigbee. Dále zde jsou zakresleny hlavní zařízení, která umožňují funkci sítí a jejich vzájemnou komunikaci.

Parametry použité lokální sítě jsou následující. Rozsah dostupných IP adres: 192.168.11.0/24, brána sítě je dostupná na 192.168.11.1 a adresa pro broadcast[1] je 192.168.11.255. Ostatní IP adresy jsou z rozsahu 192.168.11.2 - 192.168.11.254. Přístup pro ostatní zařízení do této laboratorní sítě je možný i pomocí připojení přes WiFi s SSID[2] NES@FIT IoT.

Počítač, na kterém je zajištěn provoz aplikace Home Assistant je kromě lokálního přístupu dostupný i na veřejné adrese 147.229.13.92 a k webovému rozhaní je možné přistoupit vzdáleně přes adresu https://ha.nesad.fit.vutbr.cz/. Vzdálený přístup k webovému rozhraní je zabezpečený a ke komunikaci je použit aplikační protokol HTTPS[3], tedy HTTP protokol rozšířený o šifrování pomocí TLS[4]. Do IoT sítě je připojen pomocí WiFi s IP adresou 192.168.11.162. Home Assistant je nainstalován pomocí kontejneru a je použita instalace typu *Home Assistant Supervised*, která je popsána v kapitole 3.2.

Připojený koordinátor sítě Zigbee se nachází na adrese 192.168.11.41. Použitým koordinánorem je Sonoff Zigbee Bridge[5]. Koorinátor vytváří samostatnou bezdrátovou síť a propojuje ji s lokální internetovou sítí. Do Zigbee sítě se pak připojují některé ostatní senzory, mezi které patří využitý senzor na snímání pohybu intenzity světla Aquara Illuminance and Motion Sensor E1[6] a magnetický senzor pro detekci otevření dveří Aquara E1 Door & Window Sensor[7].

3.2 Aplikace pro správu chytré sítě

Trh se zařízeními pro chytrou domácnost je velmi pestrý a nachází se zde mnoho výrobců. Bývá zvykem, že každý výrobce pro své produkty vyvíjí vlastní, ať už webovou nebo ve většině případů mobilní aplikaci. Tyto aplikace slouží pro prvotní párování, nastavení a monitorování zařízení. Některé navíc nabízí možnost vytvářet vlastní automatizace. V rámci jedné aplikace, tedy zařízení od jednoho výrobce, tak vzniká možnost jejich propojení a lze vytvořit systém, kde na základě výstupů jednoho zařízení mohou jiná měnit svůj stav. Například rozsvícení světel při zaznamenání pohybu pohybovým senzorem. Problémem tohoto přístupu je, že zde vzniká izolace zařízení, protože různí výrobci a jejich aplikace podporují pouze omezený sortiment a vzájemně jsou mezi sebou nekompatibilní. Tím omezují využití možností, které chytrá zařízení nabízí.

Home Assistant

Použitou platformou pro správu chytrých zařízení v síti je Home Assistant. Jedná se o volně dostupný software, který je navržen tak, aby umožnil centralizované řízení a monitorování chytré domácnosti. Home Assistant je rozšiřitelný a díky velké základně aktivních uživatelů a vývojářů nabízí široké množství jak oficiálních integrací, tak i neoficiálních rozšíření pro různá zařízení od jakýchkoliv výrobců. Právě díky této modularitě Home Assistant

[1] https://cs.wikipedia.org/wiki/Broadcast
[2] https://cs.wikipedia.org/wiki/SSID
[3] https://en.wikipedia.org/wiki/HTTPS
[4] https://en.wikipedia.org/wiki/Transport_Layer_Security
[5] https://zigbee.blakadder.com/Sonoff_ZBBridge.html
[6] https://zigbee.blakadder.com/Aqara_RTCGQ15LM.html
[7] https://zigbee.blakadder.com/Aqara_MCCGQ14LM.html

představuje jednotný a univerzální systém pro interakci s chytrou domácností a eliminuje nesmyslnou nekompatibilitu zařízení od různých výrobců. Umožňuje tak univerzální propojení a komunikaci mezi všemi připojenými zařízeními.

Pro využití Home Assistantu je potřeba jej nejprve nainstalovat. Zbůsobů instalace je dostupných hned několik[8], avšak nároky na hardware jsou minimální a pro plně funkční instalaci je dostačující i pouhé Raspberry Pi 4[4]. Po instalaci je uživatelské rozhraní dostupné prostřednictvím webové aplikace využitím internetového prohlížeče, ale také jako aplikace pro Android a iOS. Rozhraní poskytuje přehlednou a snadno použitelnou platformu, nabízí pohodlné ovládání a monitorování zařízení a funkce v domácnosti. Dále Home Assistant umožňuje vytvářet pokročilé automatizace a skripty, které na základě určitých podmínek dovedou vyvolat různé události, nebo měnit nastavení či informovat uživatele o daném stavu.

Díky schopnosti ovládat několik zařízení od různých výrobců, která by za normálních okolností nebyla schopna spolu komunikovat, má Home Assistantu velkou výhodu a dává tak základ pro široké využití všech možností, která zařízení pro chytrou domácnost poskytují.

Architektura a rozšíření Home Assistantu

Jak už bylo zmíněno, Home Assistant je možné rozšiřovat o nové funkce přidáváním balíčků. Home Assistant se z pohledu jeho architektury dělí na dvě části. Ta hlavní, která představuje jeho jádro, se jmenuje *Home Assistant Core*. Druhou částí je *Home Assistant Supervisor*, který je dostupný pouze u některých typů instalace a slouží jako správce rozšíření, která jsou instalována využitím *Docker*[9] kontejnerů.

- **Home Assistant OS** (HAOS) – je plnohodnotný operační systém optimalizovaný pro potřeby Home Assistantu a používá Docker pro správu jednotlivých kontejnerů. Ve výchozím nastavení nasazuje Home Assistant Supervisor jako kontejner [3].
- **Home Assistant Supervised** – jde o alternativní způsob *Supervisor* instalace, který je složitější na instalaci a místo instalace plného operačního systému nebo plnohodnotné virtualizace využivá Docker pro svou kontejnerizaci. To znamená, že Home Assistant Core i Supervisor jsou spuštěny ve vlastním kontejneru. Má pevně dané požadavky[10] a uživatel je zodpovědný, že na hostitelském operačním systému jsou nainstalovány požadované závislosti a jsou udržovány jejich správné verze.

Každá ze dvou částí Home Assistantu přináší vlastní rozšíření. V případě Home Assistant Core se těmto rozšířením říká *integrace* [3]. Každá integrace definuje určitou doménu IoT zařízení. Například integrace pro světlo *light*. Ta definuje data a jejich formát, ve kterém jsou v Home Assistantu dostupná. Zároveň poskytuje služby pro jejich ovládání. Dalším typem jsou integrace, které slouží pro interakci s externím zařízením. Příkladem této integrace je *Zigbee Home Automation*. Ta je propojena se Zigbee koordinátorem a jsou skrze ni pro interakci dostupné ostatní doménové integrace.

Rozšíření dostupná díky Home Assistant Supervisor jsou nazývány *addons*. Ty na rozdíl od integrací umožňují rozšířit funkcionality Home Assistantu přidáním nové aplikace

[8] https://www.home-assistant.io/installation/
[9] https://www.docker.com/
[10] https://github.com/home-assistant/architecture/blob/master/adr/0014-home-assistant-supervised.md

od třetí strany. Příkladem může být rozšíření, které přidá plnohodnoté vývojové prostředí Visual Studio Code[11] nebo taky aplikace Frigate[12] pro správu IP kamer.

Třetím zdrojem, ze kterého je možné získat nová rozšíření je *Home Assistant Community Store* (HACS). Jde o komunitní „obchod“, který obsauje integrace, které, zatím, nejsou dostupné z oficiálních zdrojů. Protože se jedná o neoficiální zdroj, tak tato rozšíření nejsou testována vývojáři Home Assistantu a jeho týmy nenesou zodpovědnost a nenabízí podporu pro jakákoliv rozšíření z tohoto zdroje. Protože po instalaci jsou rozšíření autorizována a mohou přistupovat do aplikačního programovacího rozhraní Home Assistantu, je dobré dbát na bezpečnost při instalaci rozšíření z tohoto zdroje.

Zabezpečení vzdáleného přístupu

Vzdálený přístup do Home Assistantu je velmi užitečná funkce pro ovládání a pro přehled o dění v systému z libovolného místa. Přístup umožňuje monitorovat stav domova a provádět stejné akce jako kdyby se uživatel nacházel doma. Může tedy zkontrolovat, zda jsou vypnuta všechna světla, jsou zavřeny vstupní dveře a nebo kontrolovat stav termostatu. Zároveň umožňuje přístup k bezpečnostním prvkům jako jsou například bezpečnostní kamery nebo ke kontrole stavu alarmu. Lze také nastavit generování upozornění na základě různých událostí a uživatel je tak může dostávat na svém mobilním telefonu, což mu umožní rychle reagovat na důležité události, jako například detekce kouře nebo pohyb v neautorizovaných oblastech.

Ačkoliv vzdálená správa přináší spoustu výhod, je důležité věnovat pozornost jejímu správnému zabezpečení. Protože během přihlašování dochází k přenosu hesel a jiných citlivých dat, tak se nesmí stát, že by tato data byla přenášena v nezabezpečené formě po veřejném internetu. Existuje několik možností jak zajistit bezpečné připojení do domácí sítě, respektive do Home Assistantu. Nejjednodušším řešením je pravděpodobně služba poskytovavá přímo od zakladatelů Home Assistantu, *Home Assistant Cloud*. Jde o zpoplatněné cloudové řešení, které po registraci a povolení v nastavení Home Assistantu umožní bezpečné připojení, šifrované pomocí SSL certifikátů získaných od certifikační autority Let's Encrypt[13]. Druhou, už trochu složitější možností je nainstalování rozšíření *Nginx Proxy Manager*. Pro jeho funkci je zapotřebí mít vlastní doménu, na směrovači přesměrovat porty 80 a 443 a po prvotním nastavení automaticky zajišťuje platné SSL certifikáty pro zabezpečení komunikace od stejné certifikační autority jako Home Assistant Cloud.

3.3 Komunikační protokol Zigbee

Zigbee je standardizovaná bezdrátová síť navržená tak, aby umožnila nízkoenergetický způsob komunikace mezi zařízeními, především u Internetu věcí (IoT). Zigbee vychází ze standardu IEEE 802.15.4, který definuje fyzickou a MAC vrstvu pro nízkoenergetické osobní sítě s nízkou rychlostí. Zigbee pak navíc definuje specifikace síťové vrstvy pro hvězdicovou, stromovou a meshovou topologii sítě a poskytuje rámec pro programování aplikací v aplikační vrstvě [10]. Vývoj standardu Zigbee je zastřešen seskupením mnoha firem pod asociací Connectivity Standards Alliance[14].

[11] https://github.com/hassio-addons/addon-vscode
[12] https://docs.frigate.video/integrations/home-assistant/
[13] https://letsencrypt.org/
[14] https://csa-iot.org/

Zigbee pracuje ve frekvenčním pásmu 2,4 GHz a v závislosti na okolním prostředí a výkonu vysílače může disponovat dosahem až 100 metrů. Klíčovou výhodou tohoto protokolu je jeho nízká spotřeba energie. Díky tomuto je vhodým řešením pro zařízení, která jsou dlouhodobě napájena z baterie. Na druhou stranu Zigbee praucuje na nízké přenosové rychlosti a je tedy vhodný pouze pro aplikace s malým přenosem dat.

Typy zařízení v síti Zigbee

Ne u všech zařízení musí být žádoucí, aby byla schopna provádět hybridní funkci koncového zařízení a směrovače. Například u snímačů, které jsou pro jednoduchou montáž často napájeny z vnitřní baterie, by toto vedlo ke zvýšení nároků na hardware senzoru a také ke zvýšení jeho spotřeby energie. V síti Zigbee se tedy mohou nacházet různé typy zařízení, které můžeme dělit podle fyzických a logických vlastností [8].

Fyzické typy zařízení

- **Zařízení s plnou funkcionalitou** podporují všechny dostupné operace v rámci standardu, včetně směrovacího mechanismu, koordinačních úkonů a aktivnímu naslouchání. Typicky se jedná o světla nebo zásuvky, tedy zařízení, která jsou vždy připojená k elektrické síti a naslouchají v síti Zigbee.
- **Zařízení s redukovanou funkcionalitou** mají omezenou výpočetní kapacitu a tak neprovádí aktivní směrování paketů. Musí tedy mít ve svém dosahu zařízení s plnou funkcionalitou, která tuto činost obstarávají. Jejich hlavním účelem je vyhledat síť, po které mohou přenést data, kontrolovat jestli data nečakají na odvysílání a samotné vysílání dat do koordinátoru. Pokud není zapotřebí přenášet data, tak se pro omezení spotřeby enerige přepínají do režimu spánku.

Logické typy zařízení

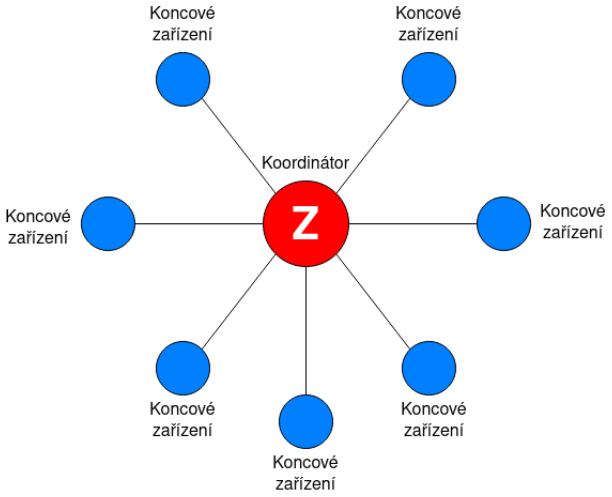
- **Zigbee koordinátor** představuje hlavní bod Zigbee sítě. Slouží pro vytvoření této sítě, nastavení velikosti přenášených paketů, využívané topologie a dalších parametrů. V jedné Zigbee síti se vždy nachází právě jeden koordinátor a zároveň také může fungovat jako brána (můstek) do ostatních sítí, například do lokální internetové sítě.
- **Zigbee směrovače** jsou typicky zařízení připojená přímo do elektrické sítě a jsou schopna vykonávat dvojí funkcionalitu jak aplikačního zařízení, tak i směrovače. Směrovače jsou přechodná zařízení, která směrují přenášená data od zdroje k jejich cíli. Jde o zařízení schopná rozšířit dosah sítě dále než je samotný koordinátor schopen pokrýt.
- **Koncová zařízení** obsahují omezenou funkcionalitu jim dostačující pro komunikaci v síti. Jedná se o zařizení s redukovanou funkcionalitou.

Topologie využívané v Zigbee

Zigbee podporuje celkově 3 typy síťové topologie [8]:

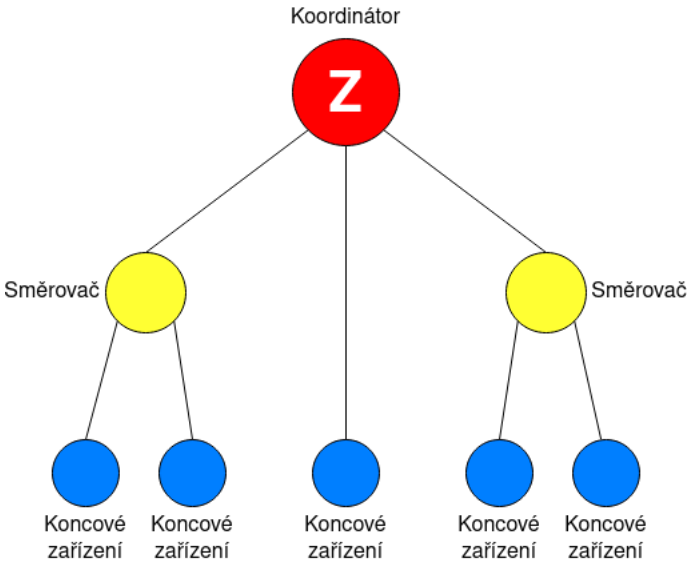
- **Hvězdicová topologie** je sestavena z koordinátoru nacházejícího se ve středu vytvořené sítě a ostatní zařízení jsou umístěna okolo tohoto bodu. V této topologii se všechna zařízení připojují přímo ke hlavnímu koordinátoru, který je jediným bodem

topologie, se kterým mohou komunikovat. Veškerá komunikace s ostatními zařízeními musí tedy probíhat přes koordinátor.



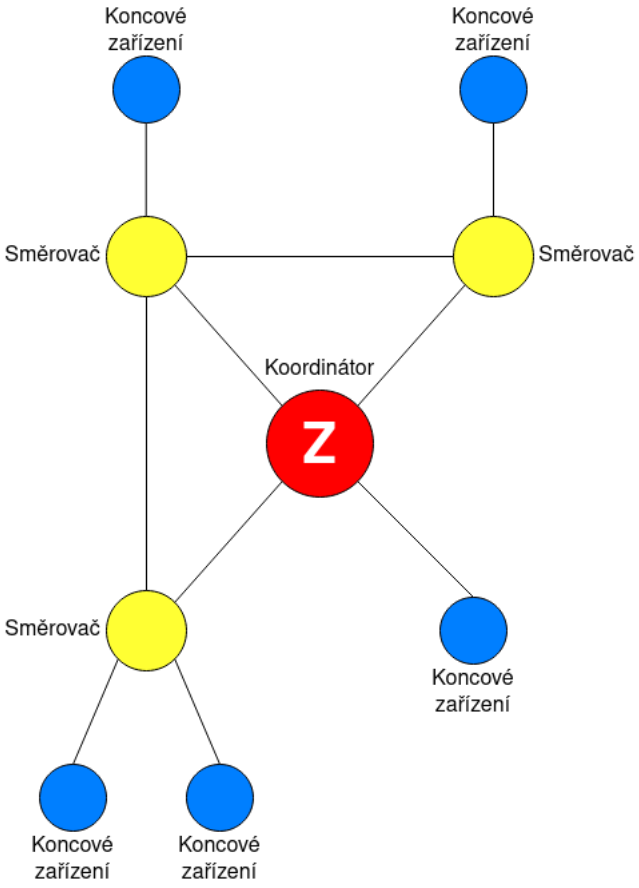
Obrázek 3.2: Zobrazení hvězdicové topologie

- **Stromová topologie** se skládá z kořene, jehož pozici v případě Zigbee sítě zaujímá koordinátor. Na tento kořen se pak v hierarchii připojují další směrovače, kteří tuto síť rozšiřují a koncová zařízení představují listy. Směrovače představují vnitřní uzly stromu. V obrázku 3.3 je zobrazeno propojení jednotlivých uzlů. Koncová zařzení mezi sebou nemohou napřímo komunikovat, a proto jsou připojena ke koordinátoru nebo k jednomu ze směrovačů. Každý uzel může komunikovat se svým rodičovským uzlem a právě přes tento uzel probíhá komunikace s ostatními uzly. Nevýhodou stromové topologie je, že v případě výpadku jednoho z vnitřích uzlů dojde i k odpojení jeho dceřiných uzlů, protože zanikne jedinná cesta, kterou byly schopny komunikovat.



Obrázek 3.3: Zobrazení stromové topologie

- **Smíšená topologie** využívá koordinátor, několik směrovačů a jedno nebo více koncových zařízení. V tomto typu zapojení jsou koordinátor a směrovače schopny komunikovat s ostatními uzly v jejich dosahu. Pokud ale cílové zařízení v jejich dosahu není, tak je posláno na jeden ze směrovačů v dosahu, který data přepošle k danému cílovému zařízení. Výhodou smíšené topologie je, že dokáže dynamicky měnit svou velikost. To znamená, že jednotlivé směrovače mohou být průběžně přidávány a odebírány [8]. Přidání nových koncových zařízení je pak stejně jednoduché, neboť po připojení může komunikovat s jakýmkoliv směrovačem v jeho dosahu [8]. Výhodou této topologie je její schopnost sebeopravy. V případě, že dojde k výpadku jednoho směrovacího uzlu, tak nedochází k výpadku podobně jako u stromové topologie. Zařízení pro svou komunikaci využijí další dostupnou cestu, dokud existují. Propojení zařízení v této topologii je zobrazeno na obrázku 3.4.



Obrázek 3.4: Zobrazení smíšené topologie

Kapitola 4

Testování vytvořeného prostředí

Tato kapitola je zaměřena na celkovou bezpečnost sítě a zařízení Internetu věcí. Cílem tohoto testování je zhodnotit kvalitu zabezpečení, identifikovat možné bezpečnostní chyby a odhalit potenciální zranitelnosti v rámci IoT infrastruktury. Snahou je zajistit, že připojená zařízení jsou odolná proti útokům a jsou schopna zachovat svou integritu, důvěryhodnost a dostupnost.

Při identifikaci možných útoků se v tomto případě dá k testování přistupovat ze dvou pohledů. Z pohledu vnějšího útočníka snažícího se připojit do webového rozhraní Home Assistantu z internetu a nebo naopak pohled „zevnitř“, kdy útočník má přístup do lokální sítě a je schopen se přímo připojovat k jednotlivým chytrým zařízením.

V kapitole jsou popsány použité nástroje a následně jsou zhodnoceny výsledky jejich testů. Na konci této kapitoly je navrhnut a realizován jednoduchý styl útoku, který dokáže být velmi účinný a jeho použití je schopno dočasně odstavit využívanou síť Zigbee.

4.1 Využité nástroje

K samotnému testování bylo použito několika nástrojů. Prvním z nich je nástroj pro skenování potenciálních zranitelností a bezpečnostních chyb v počítačových sítích Nessus[1]. Jeho cílem je poskytnout rychlý a efektivní způsob indentifikace slabých míst v infrastruktuře. Tento nástroj provádí různé techniky skenování. Mezi ty hlavní patří například skenování portů a služeb, které na nich naslouchají, skenuje operační systém a kontroluje přítomnost jeho známých zranitelností [9]. Po dokončení kontroly jsou výsledky dostupné k analýze v poskytnutém uživatelském rozhraní.

Další možná rizika popisuje projekt a komunita OWASP (Open Web Application Security Project)[2]. Jedná se souhrn nejobvyklejších bezpečnostních rizik, která se často vyskytují ve webových aplikacích. Cílem projektu je zvýšit povědomí o rizicích v (nejen) webových aplikací, a proto pravidelně vydává soupis zvaný OWASP Top 10 [15], který popisuje 10 nejčasateji odhalených bezpečnostních rizik. Mezi ně například patří:

- Injektování kódu (např. SQL injection nebo Cross-site scripting)
- Nesprávné řízení přístupu
- Chybná autentizace a správa relace

[1] https://www.tenable.com/products/nessus
[2] https://owasp.org/

- Expozice citlivých dat

Jeden z nástrojů, který se zaměřuje na vyhledávání těchto chyb je Nikto[3]. Nikto je otevřený a volně dostupný software, který umožňuje automatizovanou kontrolu webových aplikací a provádí mnoho sofistikovaných testů.

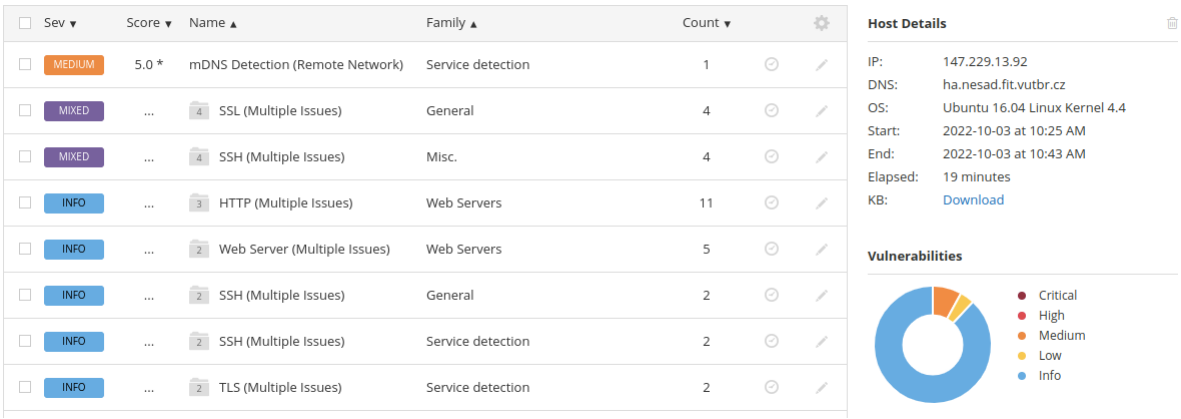
4.2 Testování z vnější sítě

Při testování z vnějšku je důležité se zaměřit na zabezpečení webového rozhraní aplikace Home Assistant. To právě často bývá veřejně dostupné s pomocí webového serveru a umožňuje tak jednoduchou vzdálenou správu a přehled pokud se zrovna uživatel nenachází ve stejné síti.

Nessus nabízí několik přednastavených testů a v tomto případě byl použit základní sken sítě (Basic Network Scan) a test webové aplikace (Web Application Tests). Testovaný Home Assistant je dostupný na adrese https://ha.nesad.fit.vutbr.cz/.

Výsledky testů

Spuštěné Nessus testy neodhalily žádnou zásadní zranitelnost. U většiny výsledků jde spíše pouze o informativní a málo důležité nálezy, které nepopisují bezpečnostní problémy. Nachází se zde i několik výstupů s nízkým nebo středním varováním, ale jedná se o upozornění ohledně dostupného SSH[4] serveru. Ten jeho konfigurací povoluje používání algoritmů pro zabezpečení komunikace, které jsou dnes považovány za nedostačující.



Obrázek 4.1: Souhrn výsledků spuštěných Nessus testů na webové rozhraní Home Assistant

Z popsaných nálezů vyplývá, že se spíše jedná o doporučení pro zvýšení bezpečnosti testovaného serveru a netýkají se přímo samotného Home Assistantu. Dá se tedy považovat, že přístup z vnějšku s využitím platných SSL certifikátů k Home Assistantu je zabezpečený na dostatečné úrovni.

Žádná rizika definovaná projektem OWASP nebyla programem Nikto nalezena.

[3] https://cirt.net/nikto2
[4] https://www.openssh.com/

4.3 Testování sítě zevnitř

Z pohledu vnitřní sítě se útočníkovi nabízí mnohem více potenciálních způsobů útoku. Tím, že se nachází ve stejné síti, tak má přímý přístup k chytrým zařízením. Pak tedy může odposlouchávat provoz, a pokud jsou data v přenášených paketech nezašifrována, tak lze velmi jednoduše vyčíst co se v dané komunikaci odehrávalo. Další možností jsou přímé útoky na chytrá zařízení, která většinou bývají za účelem snížit spotřebu energie ochuzena o některé funkce a mají slabší výkon. Takovým základním scénářem je například Denial of Service (DoS) útok, díky kterému může dojít k přetížení zařízení a tím jej dočasně odstavit z provozu.

Nessus testy na Home Assistant

V případě spouštění Nessus testů na lokální IP adresu serveru se v podstatě výsledek neliší od testů z vnějšku a opět se jedná pouze o informační nebo varování s nízkou prioritou.

Odposlouchávání komunikace

Druhý test se zaměřil na sledování komunikace mezi Home Assistantem a Zigbee můstkem. Cílem tohoto testu bylo zjistit, jestli jsou data přenášená formou prostého textu nebo jestli jsou zašifrována. V případě, že by přenášená data nebyla jakkoliv zabezpečena, tak může být ze zachycené komunikace velmi jednoduše vyčteno co bylo v daný moment po síti přenášeno.

Po zahájení odposlouchávání komunikace pomocí Wireshark[5] bylo v pravidelných, zhruba třísekundových intervalech na jednom ze senzorů stisknuto tlačítko, což vyvolá událost, která je následně zobrazena v aktivitě tohoto senzoru.

No.	Time	Source	Destination	Protoc	Length	Info
90	8.076384	192.168.11.41	192.168.11.162	TCP	68	8888 → 39516 [PSH, ACK] Seq=286 Ack=40 Win=5435 Len=14
91	8.077671	192.168.11.41	192.168.11.162	TCP	100	8888 → 39516 [PSH, ACK] Seq=300 Ack=40 Win=5435 Len=46
94	8.196571	192.168.11.41	192.168.11.162	TCP	56	8888 → 39516 [ACK] Seq=346 Ack=44 Win=5431 Len=0
104	11.282855	192.168.11.41	192.168.11.162	TCP	68	8888 → 39516 [PSH, ACK] Seq=346 Ack=44 Win=5431 Len=14
105	11.283098	192.168.11.41	192.168.11.162	TCP	100	8888 → 39516 [PSH, ACK] Seq=360 Ack=44 Win=5431 Len=46
108	11.522722	192.168.11.41	192.168.11.162	TCP	56	8888 → 39516 [ACK] Seq=406 Ack=48 Win=5427 Len=0
113	12.438124	192.168.11.41	192.168.11.162	TCP	60	8888 → 39516 [PSH, ACK] Seq=406 Ack=57 Win=5418 Len=6
114	12.440247	192.168.11.41	192.168.11.162	TCP	69	8888 → 39516 [PSH, ACK] Seq=412 Ack=57 Win=5418 Len=15
116	12.445291	192.168.11.41	192.168.11.162	TCP	57	8888 → 39516 [PSH, ACK] Seq=427 Ack=57 Win=5418 Len=3
117	12.446939	192.168.11.41	192.168.11.162	TCP	57	8888 → 39516 [PSH, ACK] Seq=430 Ack=57 Win=5418 Len=3
119	12.448622	192.168.11.41	192.168.11.162	TCP	57	8888 → 39516 [PSH, ACK] Seq=433 Ack=57 Win=5418 Len=3
120	12.450814	192.168.11.41	192.168.11.162	TCP	57	8888 → 39516 [PSH, ACK] Seq=436 Ack=57 Win=5418 Len=3
122	12.452253	192.168.11.41	192.168.11.162	TCP	56	8888 → 39516 [PSH, ACK] Seq=439 Ack=57 Win=5418 Len=2
123	12.455000	192.168.11.41	192.168.11.162	TCP	57	8888 → 39516 [PSH, ACK] Seq=441 Ack=57 Win=5418 Len=3
125	12.455785	192.168.11.41	192.168.11.162	TCP	57	8888 → 39516 [PSH, ACK] Seq=444 Ack=57 Win=5418 Len=3
126	12.457078	192.168.11.41	192.168.11.162	TCP	106	8888 → 39516 [PSH, ACK] Seq=447 Ack=57 Win=5418 Len=52
130	12.463567	192.168.11.41	192.168.11.162	TCP	56	8888 → 39516 [ACK] Seq=499 Ack=71 Win=5404 Len=0
131	12.489222	192.168.11.41	192.168.11.162	TCP	66	8888 → 39516 [PSH, ACK] Seq=499 Ack=71 Win=5404 Len=12
133	12.545692	192.168.11.41	192.168.11.162	TCP	56	8888 → 39516 [ACK] Seq=511 Ack=75 Win=5400 Len=0
147	14.122933	192.168.11.41	192.168.11.162	TCP	74	8888 → 39516 [PSH, ACK] Seq=511 Ack=75 Win=5400 Len=20
148	14.123360	192.168.11.41	192.168.11.162	TCP	93	8888 → 39516 [PSH, ACK] Seq=531 Ack=75 Win=5400 Len=39
149	14.123395	192.168.11.41	192.168.11.162	TCP	56	8888 → 39516 [PSH, ACK] Seq=570 Ack=75 Win=5400 Len=1
160	14.324615	192.168.11.41	192.168.11.162	TCP	56	8888 → 39516 [ACK] Seq=571 Ack=79 Win=5396 Len=0
170	17.133839	192.168.11.41	192.168.11.162	TCP	61	8888 → 39516 [PSH, ACK] Seq=571 Ack=79 Win=5396 Len=7

Obrázek 4.2: Zachycené pakety zobrazené ve Wireshark

[5] https://www.wireshark.org/

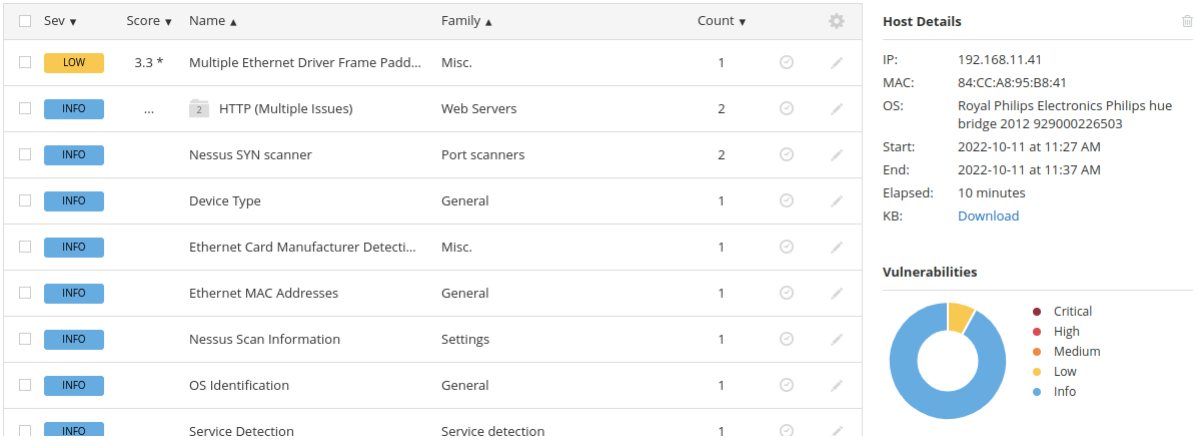
V obrázku číslo 4.2 jde vidět, že se v komunikaci pravidelně zobrazuje několik paketů s odpovídajícím intervalem tří sekund. Po nahlédnutí do jednotlivých paketů se ale nejedná o komunikaci v prostém textu a data jsou tak nečitelná. K šifrování je využitý 128 bitový klíč, který je vygenerován po nainstalování rozšíření pro komunikaci se Zigbee koordinátorem.

Odposloucháváním komunikace se tedy nedá jednoduše rozpoznat co přesně přenos obsahoval. Ze zaznamenané komunikace jde získat informace kdy byla nějaká událost vyvolána a je možné, že by se na základě velikosti přenesených dat daly od sebe odlišit různé události. Touto problematikou jsem se v této práci dále nezabýval.

Webové rozhraní Zigbee můstku

Webové rozhraní pro úpravu nastavení můstku není v tomto případě zabezpečeno jakýmkoliv typem autentizace, což můžeme vnímat jako potenciální hrozbu. Zařízení však umožňuje nastavit přihlašovací heslo.

Spuštěné testy Nessus neodhalily žádné závažné nebezpečí.



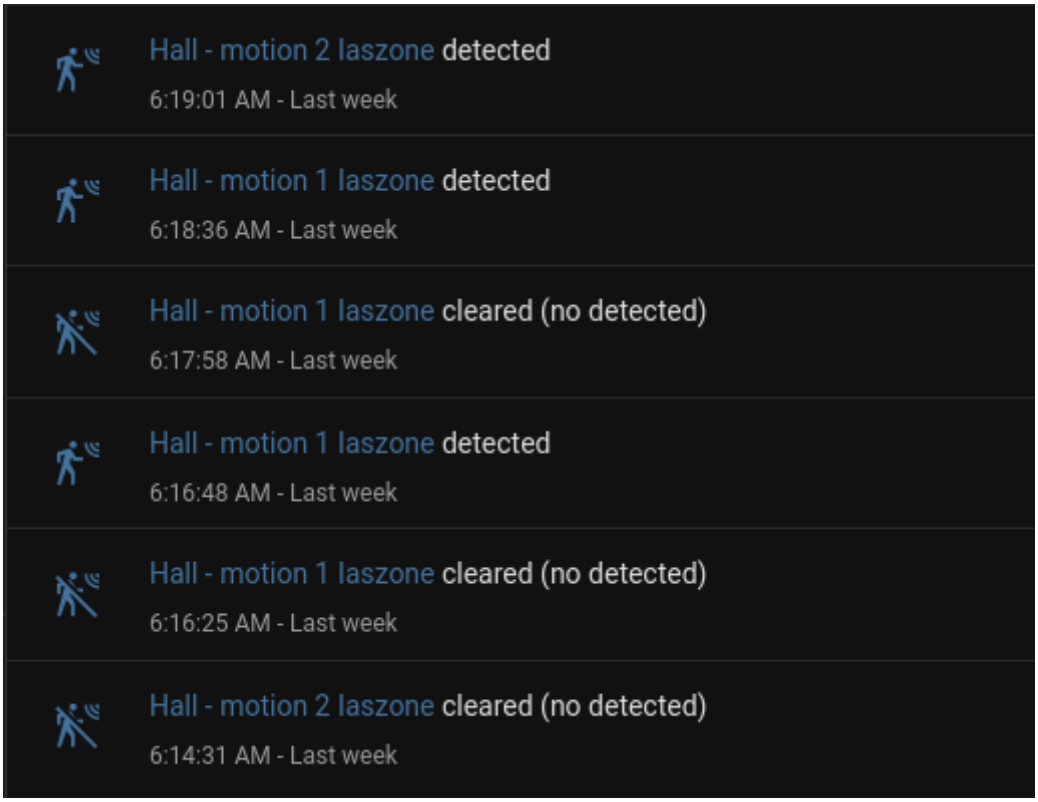
Obrázek 4.3: Souhrn výsledků spuštěných Nessus testů na webové rozhraní Zigbee můstku

Na druhou stranu sken pomocí Nikto vypsal celkem 214 nálezů. Většinou se jedná to injektování kódu a po bližším pohledu na některá varování se nálezy nepodařilo potvrdit, protože webové rozhraní na některé pokusy reagovalo přesměrováním na jiný pohled a pravděpodobně kvůli tomuto byl test vyhodnocen jako hrozba.

DoS útok na Zigbee můstek

Další pokus využívá výše zmíněný DoS útok. Využitý program se jmenuje Slowloris [14]. Tento útok je zaměřen na samotný Zigbee můstek s cílem ho zahltit a znemožnit tak aby mohl přeposílat informace od senzorů do Home Assistantu. Slowloris lze spouštět s parametrem `-s`, který stanoví počet využitých schránek na odesílání požadavků.

Před spuštěním útoku vše funguje jak má a po průchodu chodbou všechny senzory zaznamenají pohyb a událost je zaznamenána v Home Assistantu.

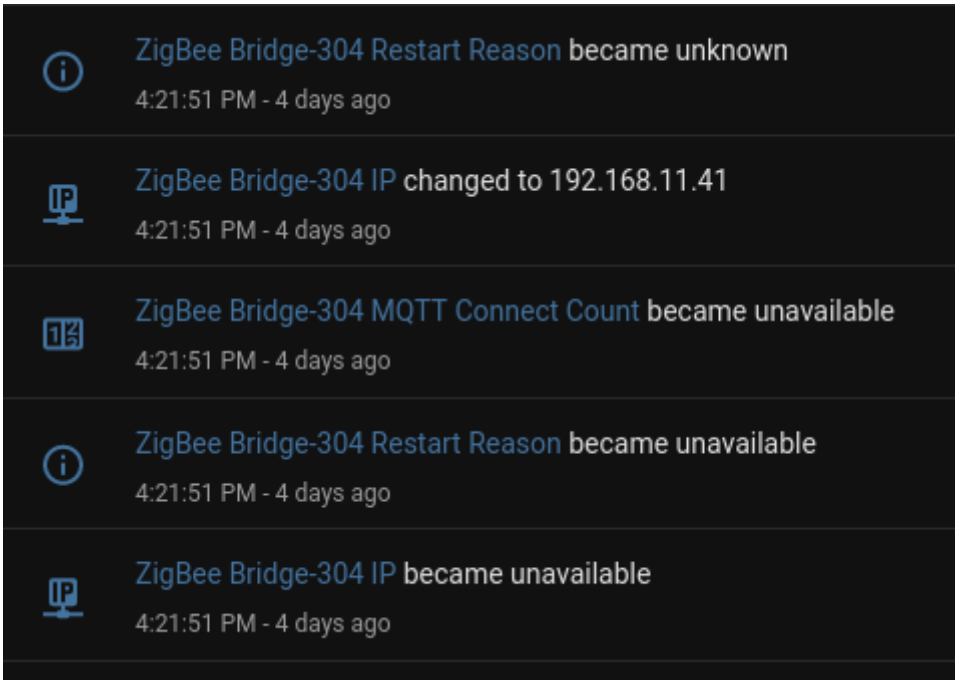


Obrázek 4.4: Záznam o detekci pohybu

Postup při provádění tohoto útoku je následovný. Nejprve je spuštěn program `slowloris` s daným parametrem. Poté je vyvoláno několik událostí průchodem po chodbě, kde senzory pohybu zaznamenají pohyb a událost je při správné funkci zaznamenána jak je ukázáno v obrázku 4.4. Po ustálení průchodu po chodbě senzory opět vyvolají událost, a to že už není žádný pohyb na chodbě. Nakonec je spuštěný útok ukončen. Po ukončení testu trvá Zigbee můstku zhruba 5 až 10 minut než se vrátí do normální funkce, protože ještě vyčkává na zahájené požadavky od dříve spuštěného útoku. Pro urychlení tohoto obnovení je můstek po ukončení útoku restartován.

Spuštěním útoku s parametry `slowloris -s 5` se značně zpomalila responzivita webového rozhraní cíleného můstku, ale zprávy od senzorů téměř vždy dorazily a jsou zobrazeny v záznamech Home Assistantu. Někdy zpráva nebyla doručena, protože na krátkou dobu byl můstek v Home Assistantu zobrazen jako nedostupný.

Po spuštění s `slowloris -s 10` se výrazně zvýšil interval kdy byl můstek nedostupný a kvůli tomu byla většina zpráv ze senzorů ztracena. Při spuštění s parametry `slowloris -s 15` a `slowloris -s 20` už je můstek pouze nedostupný a žádné přicházející zprávy od pohybových senzorů nejsou přeposílány dále do Home Assistantu. Po úplném zahlcení můstku se v záznamech pouze opakovaně ukazuje, že je můstek nedostupný.



Obrázek 4.5: Záznam o nedostupnosti můstku

Výsledkem tohoto testu tedy je, že je velmi snadné použitý Zigbee můstek zahltit požadavky. Zahlcení má fatální následky a je dostačující na znemožnění sběru zpráv od senzorů připojených do Zigbee sítě. V tomto případě se pak útočník může volně pohybovat po monitorovaných prostorách bez jakéhokoliv záznamu o pohybu.

Kapitola 5

Metody pro analýzu aktivity zařízení

Tato kapitola se skládá celkem z 5 podčástí. Nejprve je zde popsána struktura dat získaných ze záznamů o aktivitách z Home Assistantu. Dále jsou pak jednotlivě rozebrány zvolené metody, které jsou využity pro analýzu nashromážděných dat a zhodnocení jejich výsledků.

Snahou této části je vyhledávat anomálie v chování jednotlivých zařízení v čase. Anomálie jsou neobvyklé či atypické vzory, které se liší od běžného chování.

Celkově jsou zde vyzkoušeny 4 metody, které slouží k vyhledávání anomálií. Pro každou použitou metodu je obecně nastíněn princip její funkce a pro usnadnění orientace ve výsledcích jsou přidány grafy, ve kterých je vyobrazen průběh aktivity zařízení v čase. Červeně zvýrazněné body v těchto grafech představují hodnoty, které příslušná metoda označila jako abnormální. Všechny metody jsou testovány na stejných datasetech.

Po zhodnocení výsledků detekce jednotlivých metod bude zvolena ta nejvhodnější, která bude následně použíta v aplikaci pro automatické detekování anomálií.

5.1 Připravené datasety

Analyzovaná data pochází ze záznamu o aktivitách zařízení připojených k Home Assistantu. Tyto záznamy obsahují posledních 10 dní aktivit všech zařízení. Každý záznam se skládá z nekolika různých informací, přičemž mezi ty společné patří název a ID zařízení, typ události, čas, kdy se událost odehrála a stav, do kterého se zařízení přepnulo. Struktura je zobrazena v tabulce 5.1.

název zařízení	**čas odehrání události**	**stav**
Hall - motion 1 Occupancy	2023-03-13T11:45:07.663665+00:00	on
Hall - motion 2 Occupancy	2023-03-13T11:45:12.115921+00:00	on
305 - door Opening	2023-03-13T11:45:15.475044+00:00	on
Hall - motion 1 Occupancy	2023-03-13T11:45:37.475044+00:00	off
Hall - motion 2 Occupancy	2023-03-13T11:45:42.118125+00:00	off
coffee machine READY LED	2023-03-13T11:59:20.017418+00:00	on
coffee machine READY LED	2023-03-13T12:00:05.969534+00:00	off

Tabulka 5.1: Struktura dat z Home Assistantu

Pro analýzu je z těchto dat vybrán počet událostí v čase a jednotlivá zařízení jsou odlišena podle jejich jména. Aby byl časový průběh vyvolaných událostí lépe viditelný, tak je časová osa rozdělena na diskrétní úseky po 30 minutách a každý úsek obsahuje počet v něm odehraných událostí. Vybrané vlastnosti z dat jsou tedy den, časový úsek a počet zaznamenaných událostí v příslušném časovém úseku.

Tabulka 5.2 ukazuje strukturu datasetu, který představuje vstupní hodnoty pro jednotlivé metody. Z důvodu, že metody nedokáží pracovat s textovým vyjádřením vlastností dat, musí být převedeny do číselné podoby. Číslo u dne v týdnu představuje v kolikátem dnu se událost odehrála. Pro hodiny je použit 24 hodinový formát a čísla jsou tak v intervalu 0 až 23. Poslední sloupec obsahuje počet zaznamenaných událostí pro daný časový úsek.

den v týdnu	**hodina**	**počet událostí**
4	12	5
4	13	10
4	14	12
4	15	7
4	16	9
4	17	3
4	18	5

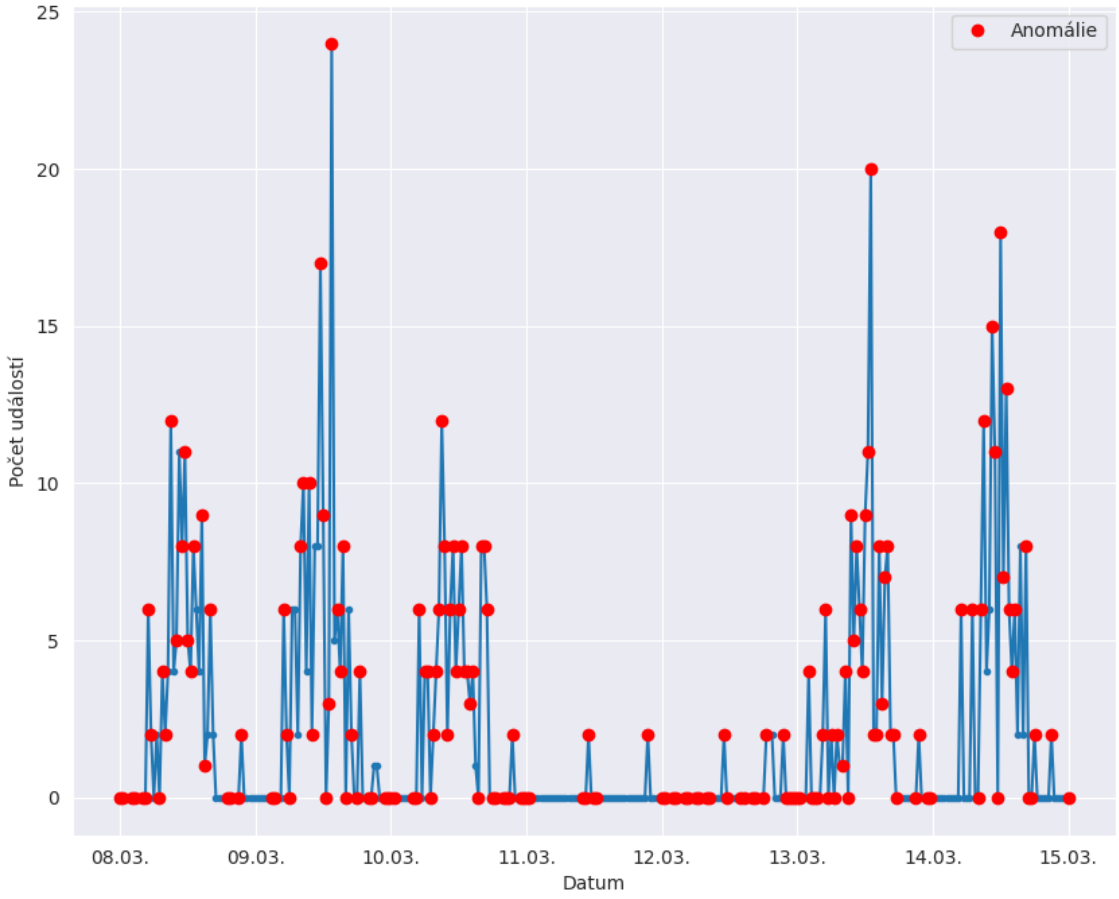
Tabulka 5.2: Struktura datasetu

5.2 Metoda OneClassSVM

Jedná se o algoritmus z kategorie učení bez učitele. V trénovací části tato metoda definuje hranici, která vyhovuje většině trénovacích bodů. Jestliže daný datový bod spadá do této hranice, tak je vyhodnocen jako normální, a pokud do této hranice nespadá, tak je označen jako odlehlý, tedy anomálie.

Metoda OneClassSVM je známá svou citlivostí na odlehlé hodnoty, a proto se při detekci příliš neosvědčila. Pro správné použití je potřeba velmmi pečlivě nastavit její parametr ν, aby se dosáhlo přesnějších výsledků. Parametr ν vyjadřuje horní hranici podílu chyb v trénovacích datech a dolní hranici podílu podpůrných vektorů. Implementace v knihovně `scikit` dovoluje hodnoty v intervalu $\nu\epsilon\,(0, 1]$ [1].

Výsledky analýzy této metody jsou zobrazeny v grafu na obrázku 5.1. Na první pohled je patrné, že metoda označila obrovský počet bodů jako anomálie. Výsledky jsou nepřesné a téměř každá změna v počtu aktivit za daný úsek je nesprávně vyhodnocena. OneClassSVM se tak jeví jako nevhodná volba pro analýzu použitých dat.



Obrázek 5.1: Detekce pomocí metody OneClassSVM

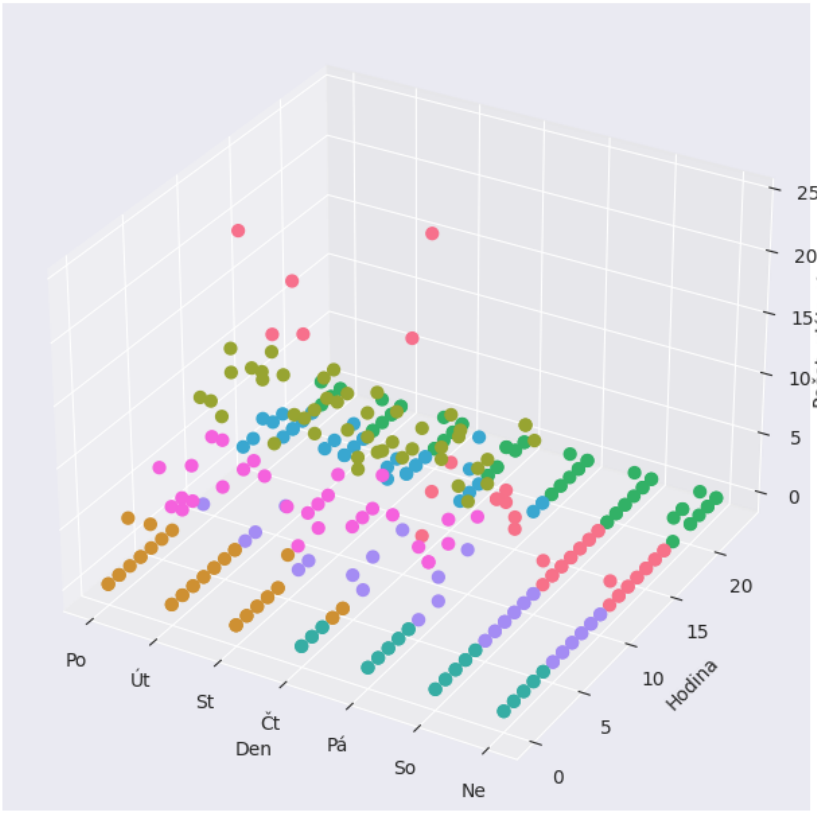
5.3 Metoda K-Means

K-Means clustering je jednou z populárních metod ze třídy učení bez učitele. Ve vstupních datech vyhledává shluky podobných bodů, a tím v nich zkoumá jejich vzory. Počet shluků k, na která data dělí, se zvolí před spustěním této metody. Výběr vhodného počtu shluků je velmi důležitý pro správnou funkci metody a má velký vliv na výsledek. Pokud je zvolené k příliš nízké, shluky budou obsahovat příliš mnoho různorodých bodů a naopak pokud je k příliš vysoké, tak mohou vzniknout shluky, které jsou příliš specifické a nemají velký význam [11].

Princip funkce K-Means spočívá v tom, že nejprve je zvoleno k shluků, do kterých se budou jednotlivé body dělit. Pak se zvolí *centroidy*, tedy k bodů, které představují výchozí středy pro jednotlivé shluky. V dalším kroku se vypočítají vzdálenosti každého bodu od každého centroidu, a každý bod je přiřazen do shluku k nejbližšímu centroidu. Po přiřazení všech bodů se pro každý shluk vypočítají nové centroidy. Tento algoritmus se vykonává tak dlouho, dokud při přepočítávání centroidů nedojde ke změně vůči předchozí iteraci nebo pokud se provede předem určený počet iterací [2, 11].

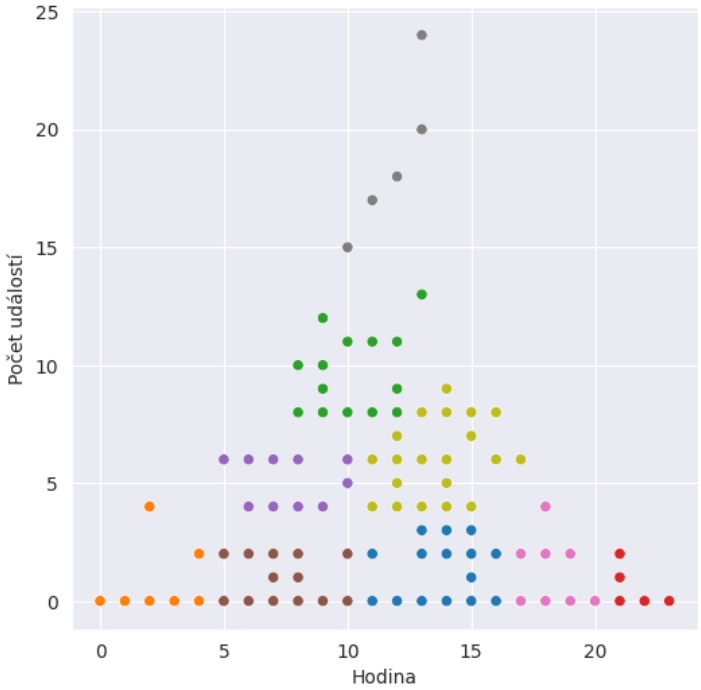
Graf v obrázku 5.2 zobrazuje výsledné shluky metody K-Means. Protože vstupní data obsahují tři vlastnosti, tak jsou body vyobrazeny ve 3D prostoru. V místech, kde je zařízení více aktivní, tak vznikají rozeznatelné shluky, ale v místech, kde zařízení nevyvolalo žádné

události jsou shluky spíše složeny ze stejných hodnot v rozmezí několika hodin. Především se tak děje během víkendu kdy obvykle není v tomto prostředí žádná lidská aktivita. Experimentálně bylo v tomto případě zvoleno $k = 9$ shluků, ale vhodý počet shluků se může měnit v závislosti na celkové aktivitě zařízení.



Obrázek 5.2: Detekce pomocí metody K-Means

Po odstranění složky, která vyznačuje den v týdnu, je zobrazen graf v obrázku 5.3. Tento graf vhodněji vyobrazuje aktivitu zařízení a umístění shluku podle počtu událostí v čase je díky tomu lépe viditelné.



Obrázek 5.3: Detekce nad daty se dvěma vlastnostmi pomocí K-Means

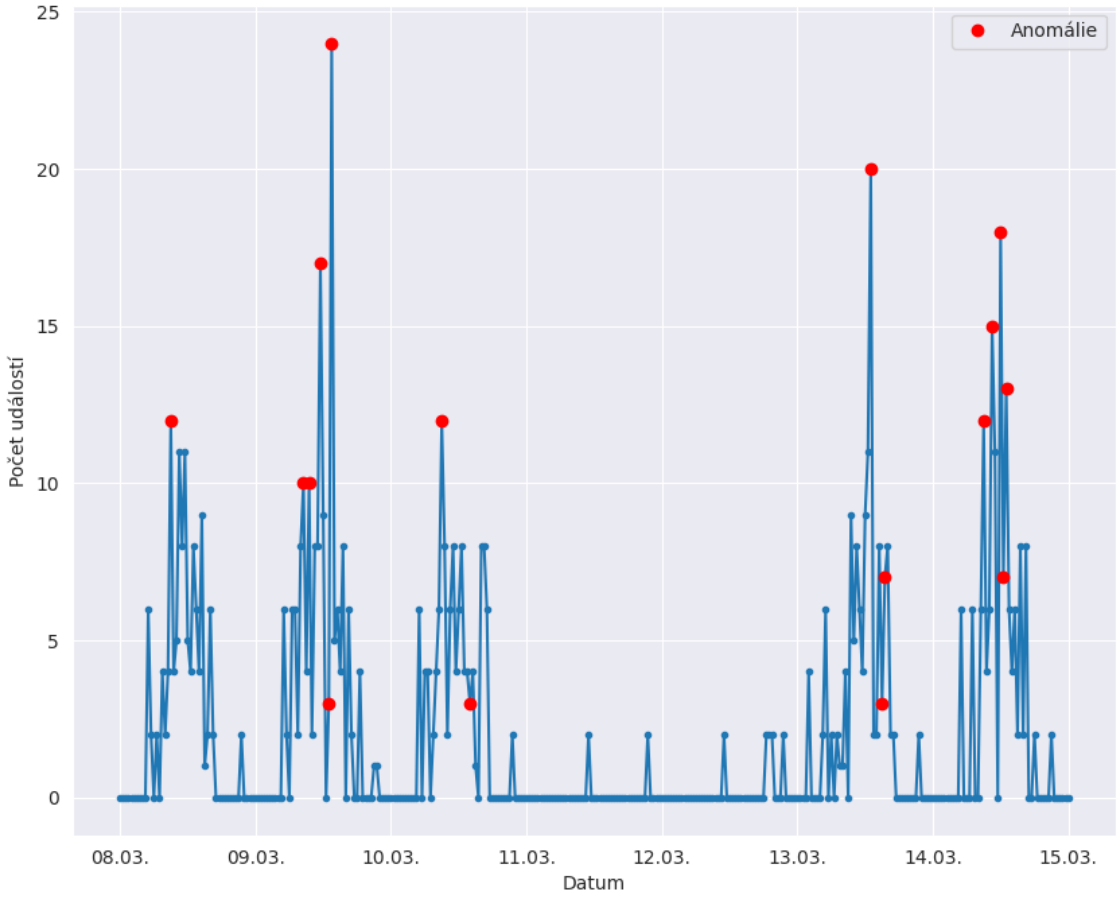
5.4 Metoda IsolationForest

Metoda *Isolation Forest* patří do třídy metod učení bez učitele. Tato metoda je založena na předpokladu, že se v datech vyskytuje pouze malé množství bodů, které jsou rozdílné, a díky tomu je jednoduší je oddělit od ostatních bodů. Jak napovídá název tohoto algoritmu, tak jeho funkce spočívá v *izolaci* jednotlivých bodů. Tímto pojmem rozumíme postupné oddělování bodu od ostatních, dokud se v odděleném prostoru nenachází jako jediný.

Princip izolace je prováděn náhodným zvolením jedné z vlastností dat a náhodým výběrem rozdělovací hodnoty, která náleží do intervalu mezi minimem a maximem zvolené vlastnosti [7]. Tímto náhodným výběrem si metoda nad daty rekurzivně vytváří oblasti, které jsou reprezentovány binárním stromem[1]. Délka cesty od kořene stromu k listu odpovídá počtu provedených rozdělení pro daný bod. Výhodou náhodného výběru je, že cesty od kořene k listu jsou výrazně kratší u anomálií. Když tedy strom vyprodukuje kratší délku cesty pro konkrétní body, je vysoce pravděpodobné, že se jedná o anomálii [7].

V grafu na obrázku 5.4 je vyobrazen výsledek detekce zvoleného zařízení. Červené značky opět zobrazují body, které metoda vyhodnotila jako anomálie. V tomto případě metoda správně označila vysoké odskoky v počtu událostí jako anomálie a ve většině případů rozeznala normální provoz. Vyskyuje se zde malé množství falešně pozitivních varování, která jsou pravděpodobně způsobena razantními skoky v aktivitě zařízení mezi sousedícími půlhodinovými intervaly. Nejde ale o výraznou nepřesnost, spíše o zanedbatelnou odchylku.

[1] https://www.tutorialspoint.com/data_structures_algorithms/tree_data_structure.htm



Obrázek 5.4: Detekce pomocí metody IsolationForest

5.5 Metoda LocalOutlierFactor

Tato metoda pro detekci anomálií je postavena na principu lokálních hustot vzorku, kde lokalita je dána k nejlbližšími sousedy. Vzdálenost těchto sousedů je využita k odhadnutí hustoty. Porovnáváním lokální hustoty bodu s jeho nejbližšími sousedy lze identifikovat oblasti s podobnou hustotou. Body s podstatně nižší hustotou než jejich sousedé jsou považovány za odlehlé [5]. Algoritmus *LocalOutlierFactor* (LOF) se skládá z několika kroků:

- Pro každý bod v datech se vypočítá jeho dosažitelnost (*Reachability Distance*, RD) vůči jeho `k`-nejbližším sousedům. Tato hodnota se vypočítá jako maximum z Euklidovské nebo Manhattanské vzdálenosti mezi bodem a jeho nejbližším sousedem a vzdálenosti nejbližšího souseda k tomu nejvzdálenějšímu [5].

$$RD_k(A, B) = max(K - vzdálenost(B), vzdálenost(A, B))$$

- Pro všechny body se vypočítá hustota místní dosažitelnosti nebo-li *Local Reachability Density* (LRD). Ta je definována jako inverzní hodnota průměrné dosažitelnosti bodu od jeho sousedů. Nízká hodnota LRD implikuje, že nejbližší shluk sousedů je poměrně vzdálený od tohoto bodu [5].

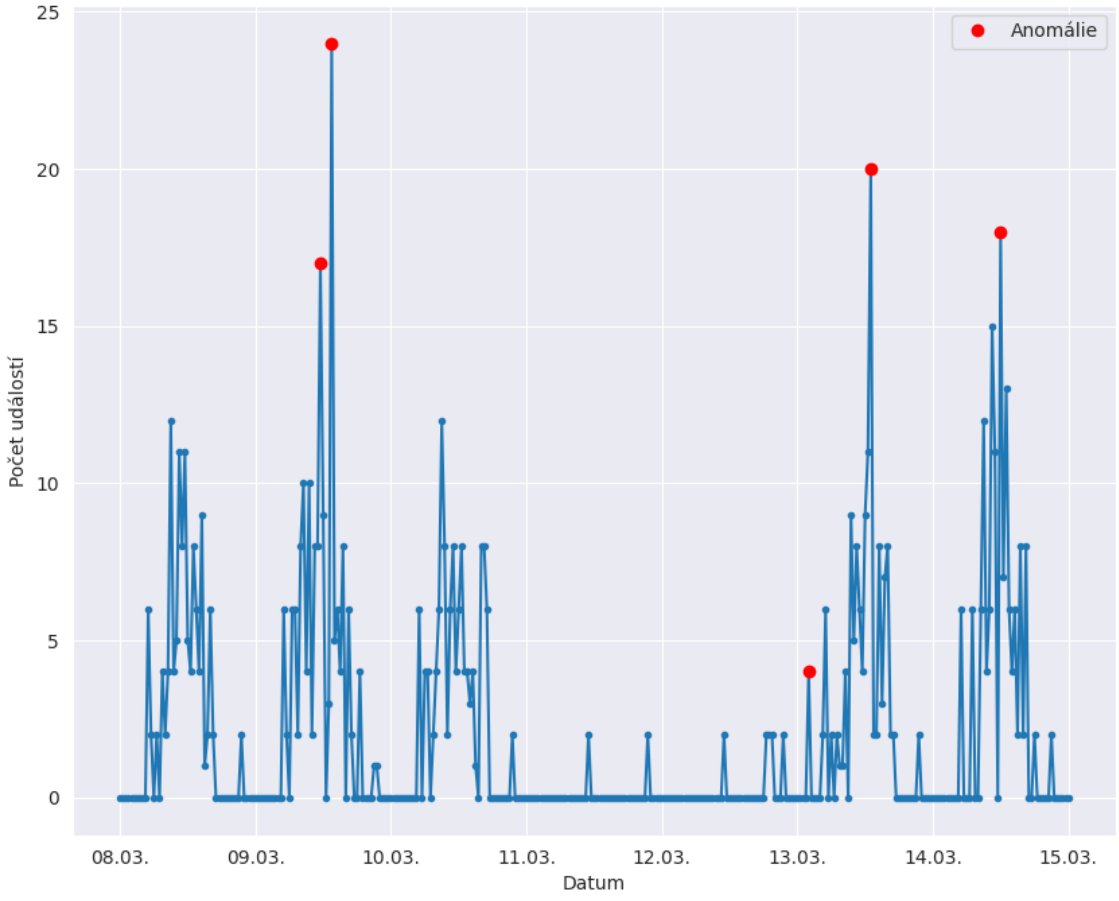
$$LRD_k(A) = \frac{1}{\sum_{B \epsilon N_k(A)} \frac{RD_k(A,B)}{||N_k(A)||}}$$

- Finální hodnota LOF je pak vypočítána jako poměr průměrné hustoty místní dosažitelnosti okolních bodů s hustotou místní dosažitelnosti bodu.

$$LOF_k(A) = \frac{\sum_{B \epsilon N_k(A)} LRD_k(B)}{||N_k(A)||} \cdot \frac{1}{LRD_k(A)}$$

Pokud bod není odlehlý, tak jeho hodnota LRD je podobná průměrnému LRD a jeho LOF se blíží k 1. Naopak pokud se jedná o odlehlý bod, tak je jeho hodnota LOF vysoká.

V obrázku 5.5 je ukázán graf s výsledky po využití této metody. Z výsledků je patrné, že metoda správně označila vysoké výkyvy v aktivitě a v místěch normálního provozu prakticky neobjevují nepravdivě označené body. Z otestovaných metod se LocalOutlierFactor ukázal jako nejvíce vhodný pro detekci anomálií. Výsledky obsahují malý počet falešně pozitivních bodů a přesnost je u středně a víče aktivních zařízení na dobré úrovni.



Obrázek 5.5: Detekce pomocí metody LocalOutlierFactor

5.6 Porovnání výsledků jednotlivých metod

Výše byly popsány výsledky jednotlivých metod. Všechny metody dospěly k určitému závěru. Jako nejméně vhodná se ukázala metoda OneClassSVM, která nedokázala rozeznat určité vzory v chování zařízení a s velkou nepřesností označila spoustu bodů jako anomálie. Výsledné shluky u metody K-Means nemají adekvátní vypovídací schopnost. Vzniklé shluky jsou příliš rozptýleny, a proto se i tato metoda jeví jako nevhodná. Třetí posuzovaná metoda IsolationForest už dosáhla částečně použitelných výsledků. Na rozdíl od OneClassSVM dokázala rozeznat podobnosti v chování zařízení a jako anomálii označila viditelně odlehlé body. Nicméně výsledky jsou pořád částečně nepřesné tím, že označila v bežném provozu relativně velký počet anomálií. Jako nejvhodnější metoda se ukázala LocalOutlierFactor. Nejlépe ze všech metod označila viditelné výkyvy v aktivitě a v místech, kde nedochází k výrazným výkyvům nezaznamenala žádnou neobvyklost.

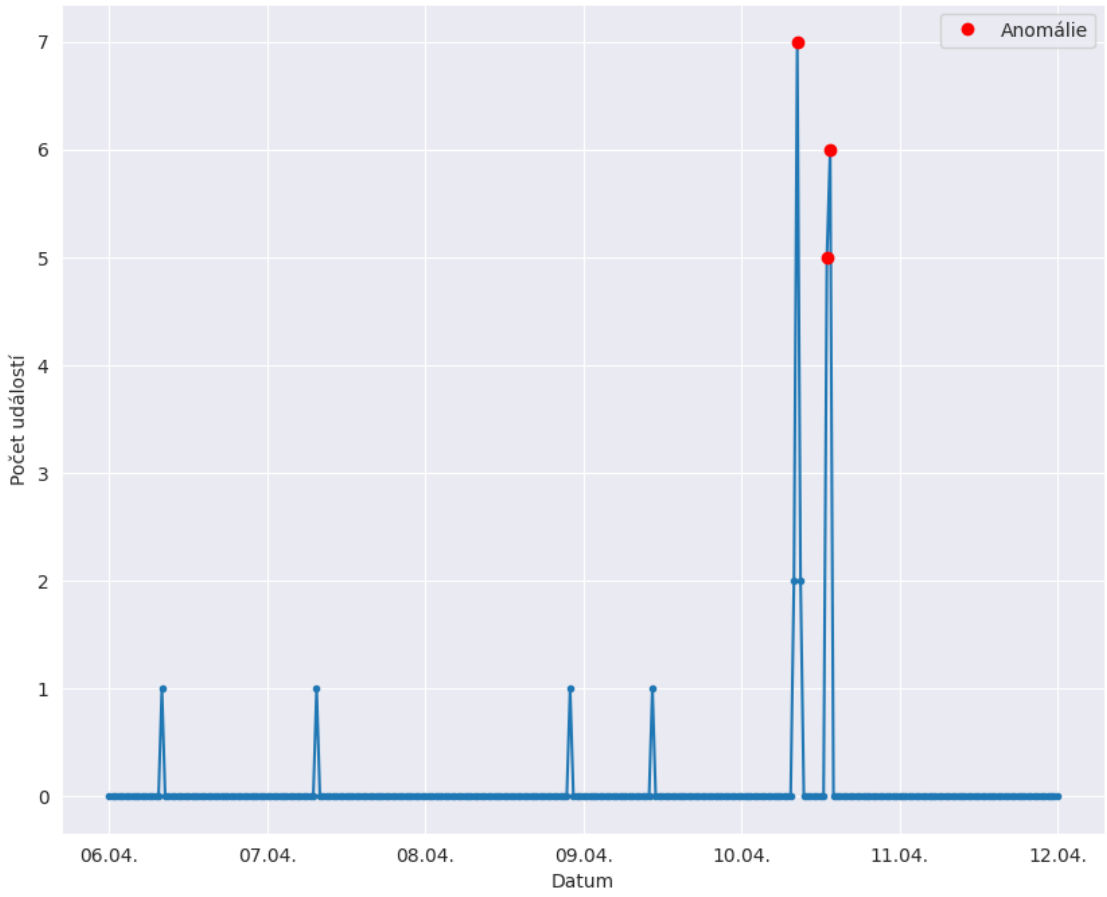
Po zhodnocení výsledků vyzkoušených metod je pro další analýzu zvolena metoda LocalOutlierFactor, protože je schopna nejlépe rozpoznat obvyklé vzory v chování zařízení a její výsledky obsahují nejnižší počet nesprávně vyhodnocených bodů.

5.7 Detekce útoku na Zigbee můstek

V předchozí kapitole byl popsán realizovaný DoS útok na Zigbee můstek. V této části bude provedena detekce tohoto útoku za pomocí vybrané metody LocalOutlierFactor a vyhodnocena úspěšnost správné detekce časových úseků, kdy byl útok proveden.

Z grafu v obrázku 5.6 jednoznačně vyplývá, že metoda správně označila oba úseky, ve kterých útok probíhal. Zahlcení můstku způsobuje, že opakovaně dochází k odpojení a tím vzroste počet zaznamenaných událostí. Za normální funkce na Zigbee můstku nejsou po většinu času zaznamenávány události, pouze občas dojde v nepravidelných intervalech například k ověření IP adresy nebo ke změně počtu připojených zařízení v Zigbee síti. Tyto úseky metoda vyhodnotila správně jako obvyklý stav a nejsou falešně označené jako anomálie.

Metoda LocalOutlierFactor se tedy v tomto jednoduchém pokusu jeví jako funkční a vhodná pro další použití.



Obrázek 5.6: Detekce DoS útoku

Kapitola 6

Nástroj pro monitorování aktivit

Součástí této práce je vytvoření prototypu aplikace, která je dostupná jako rozšíření do Home Assistantu. Po instalaci si aplikace automaticky vyžádá data ze záznamů Home Assistantu a může nad nimi provádět analýzu použitím vybrané metody. Z vyzkoušených metod je zvolena metoda LocalOutlierFactor, protože její výsledky se po experimentování v minulé kapitole 5 ukázaly jako nejlépe použitelné. V této kapitole jsou zmíněny použité technologie při tvorbě prototypu a jsou popsány problémy, na které se během vývoje a nasazení aplikace narazilo.

6.1 Použité technologie

Aplikace je implementována v jazyce Python a použita je verze `3.10`. Tento jazyk byl zvolen, protože je populární pro zpracování dat a datovou analýzu a také nabízí knihovny pro vykreslování grafů. Pro zpracování dat je využita knihovna `pandas`. Použitá metoda LocalOutlierFactor pro detekování anomálií je implementována v knihovně `sklearn`. Pro vykreslení grafů je použita knihovna `plotly` a pro vytvoření jednoduchého webového rozhraní je použita knihovna `dash`. Pro instalaci jednotlivých knihoven je použit program `pip` a všechny potřebné závislosti jsou sepsány v souboru `requirements.txt`. Tyto závislosti jde pak jednoduše nainstalovat příkazem `pip install -r requirements.txt`.

6.2 Prototyp

Popis funkce

Po spuštění je potřeba nejprve získat data o aktivitách zařízení komunikujících s Home Assistantem. Ta jsou získána odesláním požadavku na aplikační programovací rozhraní (API). Požadavek je zaslán na koncový bod API `/api/logbook/<čas začátku>?end_time=<čas konce>`, kde je nutné specifikovat dva parametry. Paremetry `čas začátku` a `čas konce` ohraničují časový interval, ze kterého mají zaznamenané události pocházet. Home Assistant uchovává události za posledních 10 dní, a proto jsou ohraničující parametry nastaveny tak, aby odpovídaly tomuto omezení.

Po získání kladné odpovědi a tedy obdržení záznamů o aktivitách zařízení může být zahájena příprava dat pro následnou analýzu. Obdržené záznamy jsou ve formátu `JSON` a s využitím knihovny `pandas` jsou převedeny do datového rámce, který je potřebný pro další použití funkcí pro zpracovávání touto knihovnou. Ze vzniklého rámce jsou odstraněny některé

nepotřebné údaje a zachovány jsou pouze ID zařízení, jméno zařízení v Home Assistantu, stav a čas, kdy byla aktivita zaznamenána.

Datový rámec obsahuje předpřipravené záznamy o všech aktivitách a je tedy potřeba tyto záznamy seskupit podle jednotlivých zařízení, ke kterým patří. Seskupení je provedeno metodou `groupby` a následně jde k záznamům pro jednotlivá zařízení přistupovat pomocí metody `get_group(<název zařízení>)`.

Seskupený rámec obsahující záznamy o aktivitách jednoho zařízení je následně převzorkován a časové úseky jsou rozděleny po půl hodinách, kde pro každý úsek je vypočítán počet zaznamenaných aktivit. Časové úseky ve zkoumaných 10 dnech, kde není zaznamenána žádná aktivita, jsou doplněny nulou. Ze zaznamenaného času je určeno, o který den v týdnu se jedná a data jsou připravena ve formátu, ve kterém jsou předána metodě pro detekci anomálií. Předávaný formát dat obsahuje tři vlastnosti a to den v týdnu, čas a počet událostí.

Použitá metoda LocalOutlierFactor následně přizpůsobí svůj model předaným trénovacím datům. Výsledkem její analýzy je pole obsahující příznaky, kde hodnota příznaku 1 značí, že se jedná o normální bod a hodnota -1 znamená, že byl bod vyhodnocen jako anomálie.

Rámec obsahující aktivitu zařízení a výsledek detekce je nakonec použit pro vykreslení grafu. Graf, stejně jako grafy v kapitole 5, zobrazuje chování vybraného zařízení v čase a jsou zvýrazněny body, které detekční metoda vyhodnotila jako anomálie. Graf je navíc interaktivní a je možné v něm libovolně přibližovat vybrané oblasti. Zároveň je také možné najetím myši zobrazit hodnoty vybraného bodu.

Nasazení aplikace

Prototyp je možné nainstalovat jako rozšíření do Home Assistantu. Protože se jedná o prototyp, tak není dostupný v některém z repozitářů, ale instaluje se jako lokálního rozšíření. Kvůli tomuto je třeba Home Assistantu poskytnout všechny zdrojové soubory a soubory potřebné k sestavení a konfiguraci aplikace. Rozšíření jsou nasazována jako Docker kontejner, a proto se musí nejprve sestavit obraz pro tento kontejner. Postup pro jeho sestavení se nachází v souboru `Dockerfile` a ostatní informace o rozšíření, včetně názvu a nastavení, jsou obsaženy v souboru `config.yaml`.

Pro umožnění komunikace s API je v konfiguračním souboru nastaven parametr `home_assistant_api: true`, který zajistí, že je v kontejneru nastavena environmentální proměnná `SUPERVISOR_TOKEN`. Tato proměnná obsahuje token, který je využíván pro autorizaci při každém zasílání požadavku na vnitřní API Home Assistantu.

Pro přístup k webovému rozhraní rozšíření jsou v konfiguračním souboru nastaveny parametry `host_network: true` a `ports: 8050/tcp: 8050`. Ty zajistí, že je pak možné aplikaci využívat na portu 8050. K přístupu není využit `ingress`, který Home Assistant nabízí a komplikace s jeho využitím jsou popsány v další části této kapitoly.

Kromě nasazení přímo do Home Assistatnu je možné aplikaci spustit i na jiném počítači. V tomto případě není k dispozici vnitřní autorizace skrze proměnnou `SUPERVISOR_TOKEN`, ale je zapotřebí vygenerovat přístupový token s dlouhou životností. Token lze vygenerovat v nastavení přihlášeného uživatele v Home Assistantu a následně je potřeba tento token přidat do souboru `token` v kořenovém adresáři aplikace. Aby aplikace věděla kam zasílat požadavky, tak je nutné ji spustit s parametrem `--address <adresa Home Assistantu>`.

6.3 Vyhodnocení úspěšnosti

Knihovnu pro vytvoření jednoduchého webového rozhraní `dash` jsem zvolil, protože umožňuje rychlý vývoj a je navržena přímo pro vizualizaci dat na webové stránce. Zároveň se prezentuje svou jednoduchostí při nasazení nebo při vložení stránky do jiné, větší, aplikace. Díky tomu, že se jedná o knihovnu pro jazyk `Python`, tak bylo snadné propojit webovou část aplikace se samotným zpracováváním dat a detekcí anomálií. Funkce pro vytváření grafů jsou kompatibilní s knihovnou `pandas` a umí nativně pracovat s datovými rámci, které knihovna `pandas` využívá.

Během testování nasazené aplikace na větších datech z reálného provozu a více ruzných typů zařízení se ukázalo, že výsledky neobsahují vysoký počet nesprávně označených anomálií. Zvolená metoda se osvedčila pro detekci větších výkyvů v chování zařízení. Na druhou stranu v případě, kdy došlo k nechtěnému odpojení senzoru, tak metoda nebyla schopna určit, že i neobvyklá nečinnost zařízení může být považována za anomálii.

Problémy při nasazení aplikace do Home Assistantu

Bohužel se ukázalo, že vestavění aplikace do Home Assistantu nebude tak jednoduché, jak se původně zdálo. Pro správné vestavění externí aplikace do Home Assistantu se využívá `ingress`. To umožňuje vložit webové rozhraní aplikace přímo do rozhraní Home Assistantu a zároveň, protože se jedná o vnitřní komunikaci mezi Home Assistantem a nasazeným rozšíření, zajišťuje zabezpečení přístupu včetně autentizace k vestavěné aplikaci. I přes veškerou snahu správně nasadit aplikaci za Home Assistant se nepodařilo dosáhnou správné funkce. Spuštění rozšíření proběhne bez chyb, ale z neznámého důvodu dochází k problémům při zobrazení rozhraní tohoto rozšíření. Pravděpodobně se jedná o nesprávné směrování a problémy s URL adresou požadavků. Ani po úpravě parametrů pro nastavení základní cesty URL, kde je `dash` aplikace umístěna, se tento problém nepodařilo vyřešit. Výsledné nasazení tedy vyžaduje externí přístup mimo webové rozhraní Home Assistantu, vyžaduje otevření nového portu a přístup není zabezpečen skrze nutnost přihlášení do Home Assistantu.

Kapitola 7

Závěr

Cílem bakalářské práce bylo nastudovat obecné zabezpečení domácích IoT sítí, navrhnout a realizovat experimentální prostředí pro domácí IoT sít, seznámit se s používanými technologiemi jako jsou Zigbee a Home Assistant a seznámit se s využitými senzory v experimentálním prostředí. Po vytvoření prostředí byly otestovány vybrané nástroje pro penetrační testování a byly realizovány vybrané útoky na některé prvky tohoto systému. Byl navržen způsob monitorování tohoto prostředí využitím záznamů o aktivitě zařízení v Home Assistantu. Na základě těchto dat bylo připraveno několik datasetů a byly otestovány zvolené metody pro detekci anomálií v těchto datasetech. Porovnáním jednotlivých metod byla zvolena ta nejvhodněší, která následně byla použita ve vytvořeném prototypu aplikace, který slouží k automatizaci průběžné analýzy aktivit zařízení.

V úvodní kapitole 1 bylo nastíněno téma bakalářsné práce, úvod do problematiky zabezpečení IoT sítí způsobený masivním rozšířením chytrých zařízení a s ním spojené komplikace. Čtenář zde byl seznámen s incidentem, kdy bylo chytré osvětlení zneužito pro umožnění přístupu utočníka do domácí sítě a byla zdůrazněna důležitost zabezpečení všech zařízení připojených k internetu a jejich aktivní ochraně před možnými útoky.

V kapitole 2 byl obecně představen internet věcí. Byly zde popsány konktrétní příklady jeho nasazení a využití v každodenním životě. Zmíněny byly některé využívané standardy, které umožňují komunikaci zařízení s okolním světem. Byla zde řešena důležitost bezpečnosti a ochrany chytrých zařízení, kterou se zabývají specializované týmy a instituce a také zde byly rozebrány jednotlivé oblasti hlavních skupin v rizicích kybernetické bezpečnosti. V závěru této kapitoly byla stanovena základní kritéria IoT zařízení a doporučeny byly postupy pro důkladné zabezpečení chytrých zařízení.

Kapitola 3 popisovala vytvořené testovací prostředí a používané technologie pro komunikaci nasazených zařízení. Představena byla architekura vytvořené sítě včetně grafického znázornění. Byla představena jednotlivá zařízení, která hrají důležitou roli ve funkci této laboratoře a způsob jejich připojení k internetu. Dále byl představen nástroj pro správu této chytré sítě *Home Assistant*, rozebrána byla jeho role a funkce, které nabízí. Detailně byla popsána jeho architektura, zbůsoby instalace nových rozšíření a byly zmíněny některé způsoby pro zabezpečení vzdáleného přístupu k webovému rozhraní. Nakonec byl popsán protokol Zigbee pro komunikaci IoT zařízení. Popsány byly typy zařízení, které se v Zigbee síti vyskytují a jaké úkoly vykonávají a byly zde představeny síťové topologie, které Zigbee využívá pro efektivní pokrytí.

V kapitole 4 jsem se věnoval testování vytvořeného prostředí. Zaměřena byla na celkové zabezpečení sítě a cílem testování bylo zhodnotit kvalitu jejího zabezpečení, odhalit potenciální zranitelnosti v infrastruktuře a indetifikovat možné bezpečnostní chyby. Popsány

byly využité nástroje pro penetrační testování a byly popsány nejčastejší bezpečností rizika. Testovány byly způsoby útoků v závislosti na tom, kde se útočník nachází, tedy uvnitř nebo vně sítě. Nakonec byl popsán útok, který zahltil Zigbee můstek a znemožnil tak jeho správnou funkci.

Metody pro analýzu aktivit zařízení byly popsány v kapitole 5. Celkově byly vyzkoušeny 4 metody, které slouží pro vyhledávání anomálií. Byla zde představena struktura dat získaných z Home Assistentu a způsob jejich úpravy pro následné využití detekčních metod. U každé z metod byl popsán princip jejich funkce a výsledek jejich analýzy byl zobrazen formou grafu. Na základě porovnání metod byla zvolena ta nejvhodnější a následně byla otestována provedením detekce DoS útoku.

V 6. kapitole byl popsán postup vývoje prototypu rozšíření do Home Assistantu pro detekování anomálií. Zmíněny byly použité technologie a knihovny pro zpracování dat a vytvoření základního webového prostředí. Následně byla popsána funkce a způsob, jakým komunikuje s aplikačním programovacím rozhraním Home Assistantu a jak jsou získaná data zpracována. Vysvětlen byl postup možného nasazení výsledné aplikace. V závěru byly popsány komplikace při nasazení aplikace do Home Assistantu.

Další navazující práce by se mohly zaměřit na rozšíření způsobů detekce anomálií. Otestovat by se mohly další metody, které by citlivěji detekovaly dlouhodobé změny v chování zařízení nebo metody schopné detekovat náhlý přechod do nečinnosti v případě, že dojde k výpadku spojení se zařízením. Případně z dostupných metod by mohla být zvolena nejvhodnější metoda pro detekci anomálií v chování daného zařízení. Výsledný prototyp může být rozšířen o schopnost aktivně upozorňovat uživatele v případě nalezené anomálie.

Literatura

[1] *Overview of outlier detection methods*. 2023. Dostupné z: https://scikit-learn.org/stable/modules/outlier_detection.html#outlier-detection.

[2] *Scikit userguide - Clustering*. 2023. Dostupné z: https://scikit-learn.org/stable/modules/clustering.html.

[3] CONTRIBUTORS, H. A. *Home Assistant Operating System and Integrations*. 2023. Dostupné z: https://github.com/home-assistant/operating-system#readme.

[4] GIENCKE, A. *Home Assistant & Raspberry Pi 4: How to Get Started*. 2019. Dostupné z: https://all3dp.com/2/home-assistant-raspberry-pi-tutorial/.

[5] JAYASWAL, V. *Local Outlier Factor (LOF) — Algorithm for outlier identification*. 2020. Dostupné z: https://towardsdatascience.com/local-outlier-factor-lof-algorithm-for-outlier-identification-8efb887d9843.

[6] KRAUS, R. *Flickering Philips Hue smart bulb? There could be a hacker in the house*. Dostupné z: https://mashable.com/article/philips-hue-smart-bulb-security-vulnerability-hackers.

[7] LIU, F. T., TING, K. M. a ZHOU, Z.-H. Isolation forest. In: IEEE. *2008 eighth ieee international conference on data mining*. 2008, s. 413–422.

[8] PARNEET DHILLON, H. S. A Review Paper on Zigbee (IEEE 802.15.4) Standard. 2014. Dostupné z: https://www.ijert.org/a-review-paper-on-zigbee-ieee-802-15-4-standard.

[9] ROGERS, R. *Nessus network auditing*. Elsevier, 2011.

[10] SAFARIC, S. a MALARIC, K. ZigBee wireless standard. In: IEEE. *Proceedings ELMAR 2006*. 2006, s. 259–262.

[11] SINAGA, K. P. a YANG, M.-S. Unsupervised K-means clustering algorithm. *IEEE access*. IEEE. 2020, sv. 8, s. 80716–80727.

[12] STANDARDS, N. I. of a TECHNOLOGY. *Considerations for Managing Internet of Things (IoT) Cybersecurity and Privacy Risks*. Washington, D.C.: U.S. Department of Commerce. Dostupné z: https://doi.org/10.6028/NIST.IR.8228.

[13] STANDARDS, N. I. of a TECHNOLOGY. *Recommended Criteria for Cybersecurity Labeling for Consumer Internet of Things (IoT) Products*. Washington, D.C.: U.S. Department of Commerce. Dostupné z: https://doi.org/10.6028/NIST.CSWP.02042022-2.

[14] Yaltirakli, G. *Slowloris*. 2015. Dostupné z: https://github.com/gkbrk/slowloris.

[15] Zavodsky, P., Klimes, V., Duracinska, Z., Prokop, M., Rejthar, E. et al. *OWASP Top 10 2013 - Czech*. 2013. Dostupné z: https://owasp.org/www-pdf-archive/OWASP_Top_10_-_2013_Final_-_Czech_V1.1.pdf.

Příloha A

Obsah SD karty

- **prototyp** – složka obsahující zdrojové soubory, všechny ostatní soubory potřebné pro sestavení prototypu, python notebook s porovnáním všech metod a použité datasety.
- **README.md** – návod k obsluze prototypu.
- **xmatus37_bp.pdf** – text práce.
- **text** – složka obsahující zdrojové soubory textu práce.